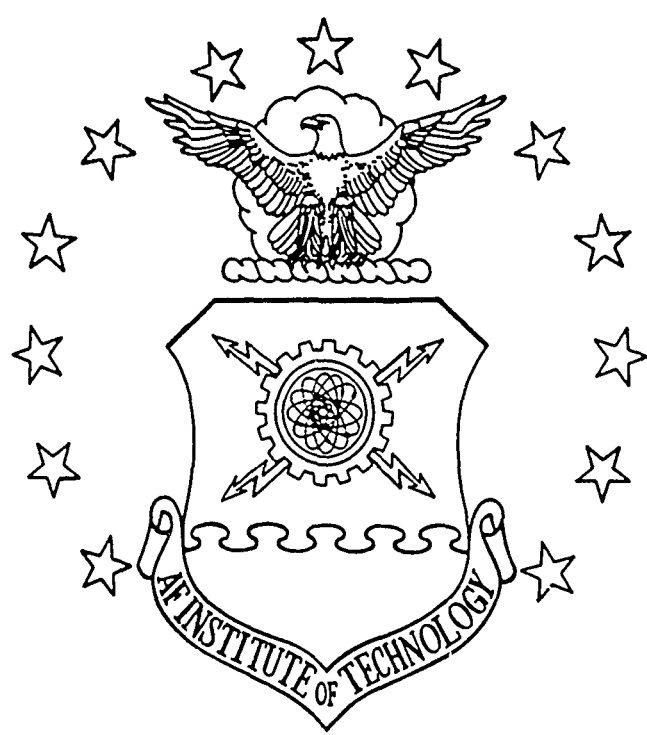




FILE COPY

AD-A205 727



ENHANCED TRACKING OF BALLISTIC
TARGETS USING FORWARD LOOKING
INFRARED MEASUREMENTS

THESIS

David R. Rizzo
Captain, USAF

AFIT/GE/ENG/89M-7

DTIC
ELECTE
29 MAR 1989
S
a
E
D

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved
for public release and sales its
distribution is unlimited.

1 89 3 29 045

AFIT/GE/ENG/89M-7

ENHANCED TRACKING OF BALLISTIC
TARGETS USING FORWARD LOOKING
INFRARED MEASUREMENTS

THESIS

David R. Rizzo
Captain, USAF

AFIT/GE/ENG/89M-7



Approved for public release; distribution unlimited

AFIT/GE/ENG/89M-7

ENHANCED TRACKING OF BALLISTIC TARGETS USING FORWARD LOOKING INFRARED MEASUREMENTS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Guidance and Control

David R. Rizzo, B.S.A.E.
Captain, USAF

March, 1989

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited

Acknowledgments

This thesis, which investigates the tracking capability of the Air Force Weapons Laboratory's high energy laser tracking system for ballistic missiles experiencing plume "pogo" effects, represents a majority of my graduate work at the Air Force Institute of Technology. This research would not have been possible without the support and guidance of many individuals along the way. I thank the Lord for giving me the opportunity to pursue my educational aspirations, and for letting me meet such wonderful people along the way.

Without a doubt, any of the successes associated with this work are directly attributed to Dr. Peter Maybeck. As my advisor, he was a constant source of motivation and inspiration; and as a friend, his ability to help me keep "life" in perspective is greatly appreciated. His dedication, personal concern and understanding are, without question, the reason I was able to complete this work. I appreciate the opportunity to work with such an admirable man, who has earned my complete respect. Additionally, I thank Major Hank Worsley, and Lt Col Zdzislaw Lewantowicz for their review of my research, as well as, their constant support throughout my AFIT experience.

To my friends at school, I say thank you for being my friends. I will never forget the experiences we've shared and the help you've given me. Particularly, Tom and Korina, Steve, Larry, Chuck, and Ron. Without you, AFIT would have been unbearable. Thanks for being there and I'll miss all of you.

A very special thanks to my beautiful wife and best friend, Shelley. Without your love and understanding, completing this program would have been impossible. Your unquestioning confidence in me supercedes my own self-confidence, and for this, your influence on my life has made me a better man. Thank you and I love you.

Finally, to my two Siberian Huskies, Tasha and Meiko, whose daily “walks” have suffered the price for late nights at school. Now that school is over, it’s time to “get back in shape.”

David R. Rizzo

Table of Contents

	Page
Acknowledgments	ii
Table of Contents	iv
List of Figures	viii
List of Tables	x
List of Symbols	xii
Abstract	xvii
 I. Introduction	 1-1
1.1 Background	1-1
1.2 Summary of Previous AFIT Research	1-3
1.3 Objectives	1-8
1.3.1 Exhaust Plume Pogo Effects.	1-9
1.3.2 Implementation of a Rotating Field-of-View.	1-9
1.3.3 Single Filter Benchmarks.	1-10
1.3.4 Single Filter Performance.	1-10
1.3.5 Single Filter Robustness Analysis.	1-10
1.3.6 Multiple Model Adaptive Filtering Benchmarks.	1-11
1.3.7 Multiple Model Adaptive Filtering Performance.	1-11
1.4 Overview of the Thesis.	1-11

	Page
II. Filter Theory	2-1
2.1 Bayesian Multiple Model Adaptive Filtering	2-1
2.2 The Extended Kalman Filter	2-6
2.3 Summary	2-8
III. Truth Model	3-1
3.1 Introduction	3-1
3.2 Dynamics Model	3-3
3.2.1 The Target States.	3-6
3.2.2 The Atmospheric States.	3-11
3.2.3 The Bending/Vibration States.	3-14
3.2.4 Plume Pogo States.	3-16
3.3 The Measurement Model	3-18
3.4 Simulation Space	3-23
3.4.1 Coordinate Frames.	3-23
3.4.2 Target Model.	3-25
3.4.3 Target Scenarios for MMAF Analysis.	3-28
3.4.4 Velocity Projection onto the FLIR Plane.	3-29
3.4.5 Target Image Projection onto the FLIR Plane.	3-31
3.5 Summary	3-33
IV. Filter Models	4-1
4.1 Introduction	4-1
4.2 Dynamics Models	4-1
4.3 The Measurement Model	4-9
4.3.1 Template Generation.	4-10
4.3.2 "Pseudo-Measurements" by Enhanced Correlation.	4-14
4.4 Summary	4-17

	Page
V. Tracking Algorithm	5-1
5.1 Introduction	5-1
5.2 Overview of the Tracking Algorithm.	5-1
5.3 Field-of-View Data Processing	5-3
5.4 Field-of-View Rotation	5-5
5.5 Truth Model Parameters	5-12
5.6 Filter Parameters	5-15
5.7 Tracking Algorithm Statistics	5-16
5.8 Performance Plots	5-17
5.8.1 Plot Designation Codes.	5-23
5.9 Greyscales	5-23
5.10 Summary	5-24
VI. Performance Analysis	6-1
6.1 Introduction	6-1
6.2 8-State Filter Tuning Via Dynamic Trajectory Parameters	6-1
6.3 8-State Filter Benchmarks	6-4
6.4 Rotating Field-of-View Analysis	6-8
6.5 10-State Filter Performance Analysis	6-11
6.6 8-State Filter Benchmark with Bending Removed from the Truth Model	6-18
6.7 10-State Filter Performance with Bending Removed from the Truth Model	6-19
6.8 Robustness and MMAF Discussions	6-21
6.9 Trouble-Shooting	6-22
6.9.1 Analysis of Possible Pogo Sign Errors in the For- tran Code.	6-23
6.9.2 Analysis of Possible Atmospheric Jitter Sign Er- rors in the Fortran Code.	6-27

	Page
6.9.3 Observability Issues.	6-31
6.9.4 Possible Pogo-Jitter Interactions.	6-36
6.10 Summary	6-38
VII. Conclusions and Recommendations	7-1
7.1 Introduction	7-1
7.2 Conclusions	7-1
7.2.1 Filter Tuning Based upon Dynamics Parameters.	7-1
7.2.2 Rotating Field-of-View Results.	7-2
7.2.3 10-State Filter Tracking Performance.	7-2
7.3 Recommendations	7-4
7.3.1 Observability of Previously Used 8-State Filters.	7-4
7.3.2 Remodelling of the Filter Dynamics.	7-5
7.3.3 Continued Characterization of the Plume Pogo.	7-6
Appendix A. Gain Calculation for Pogo Effect	A-1
Appendix B. Filter Tuning Plots via Dynamic Trajectory Parameters	B-1
Appendix C. 8-State Filter Benchmark Plots	C-1
Appendix D. Plots for FOV Rotation Analyses	D-1
Appendix E. 10-State Filter Pogo Performance Plots	E-1
Appendix F. Filter Performance Plots: Bending Removed	F-1
Appendix G. Tuning Plots Used in Jitter Sign Test	G-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure	Page
2.1. Multiple Model Filtering Algorithm	2-3
3.1. The $\alpha - \beta$ Plane	3-7
3.2. Sensor-to-Target Range	3-8
3.3. Plume Pogo Relative to the Equilibrium Point	3-19
3.4. FLIR Image Plane Intensity Function for the Difference Between Two Gaussian Intensity Functions	3-21
3.5. Distribution of Gaussian Intensity Functions	3-26
3.6. Hotspot Ellipsoid of Dispersion on the FLIR Image Plane	3-27
3.7. Geometry Required to Project the Target's Inertial Velocity onto the FLIR Plane	3-30
3.8. Infrared Image Projection Geometry	3-32
4.1. Enhanced Correlator/Linear Measurement Model Data Processing Algorithm	4-12
5.1. 8×8 Field-of-View Filter	5-4
5.2. Field-of-View Rotation Schemes	5-6
5.3. Data Processing Algorithm for a Rotating Field-of-View	5-8
5.4. Ballistic Missile Trajectory in Inertial Space	5-13
5.5. Example of Actual RMS Position Error vs. Filter Computed RMS Position Error Performance Plot (Plot #2)	5-19
5.6. Example of Mean Position Error (Mean $\pm \sigma$ at t_i^-) Performance Plot (Plot #4)	5-20
5.7. Example of Mean Position Error (Mean $\pm \sigma$ at t_i^+) Performance Plot (Plot #6)	5-21

Figure	Page
5.8. Example of Mean Centroid Position Error (Mean $\pm\sigma$ at t_i^+) Performance Plot (Plot #10)	5-22
5.9. Example of a Greyscale Diagram	5-25
6.1. Greyscale Diagram of a NRFOV	6-12
6.2. Greyscale Diagram of a RFOV	6-13
6.3. Greyscale Diagram of a DRFOV	6-14

List of Tables

Table	Page
3.1. Truth Model Scenarios	3-29
5.1. Elemental Filters of Proposed MMAF	5-2
5.2. Greyscale Field Descriptions	5-23
5.3. Greyscale Symbol Key	5-24
6.1. Individual Filter Tuning Parameters	6-2
6.2. 8-State Benchmark Filter Tuning Parameters	6-5
6.3. Truth Model Pogo Scenarios for the Four Benchmark Filters . . .	6-5
6.4. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/ Performance Statistics	6-6
6.5. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/ Performance Statistics	6-7
6.6. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/ Performance Statistics	6-7
6.7. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/ Performance Statistics	6-7
6.8. Individual Filter Tuning Parameters	6-8
6.9. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/ Performance Statistics	6-9
6.10. /BENCHMARK/SINGLE/RFOV/GAIN-HI/FREQ-HI/ Performance Statistics	6-9
6.11. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/ Performance Statistics	6-10
6.12. 10-State Pogo Performance Filter Tuning Parameters	6-15
6.13. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/ Performance Statistics	6-15

Table	Page
6.14. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-HI/ Performance Statistics	6-16
6.15. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/ Performance Statistics	6-16
6.16. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO/ Performance Statistics	6-16
6.17. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/BEND OFF/ Performance Statistics	6-19
6.18. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/ BEND OFF/ Performance Statistics	6-20
6.19. Proposed 10-State Filter Robustness Scenarios	6-23
6.20. Frame Analysis of Mean Errors for Pogo Sign Test	6-24
6.21. Performance Statistics for Simulation #1 of Pogo Sign Test . . .	6-26
6.22. Performance Statistics for Simulation #2 of Pogo Sign Test . . .	6-26
6.23. Frame Analysis of Mean Errors for Atmospheric Jitter Sign Test .	6-28
6.24. Performance Statistics for Simulation #1 of Jitter Sign Test . . .	6-29
6.25. Performance Statistics for Simulation #2 of Jitter Sign Test . . .	6-29
6.26. Performance Statistics for Simulation #1 with Pogo in the Filter	6-37
6.27. Performance Statistics for Simulation #2 with Pogo Removed from the Filter	6-37

List of Symbols

Symbol	Page
FLIR	1-1
IR	1-1
FOV	1-2
EKF	1-3
GM	1-4
MMAF	1-5
CTR	1-7
Hz	1-7
RRFOV	1-8
RFOV	1-9
DFOV	1-10
$\mathbf{x}(t)$	2-2
$\mathbf{u}(t)$	2-2
$\mathbf{w}(t)$	2-2
$\mathbf{z}(t_i)$	2-2
$\mathbf{v}(t_i)$	2-2
$p_k(t_i)$	2-2
$\hat{\mathbf{x}}_{mmaf}(t_i)$	2-2
$\mathbf{A}_k(t_i)$	2-4
$\mathbf{r}_k(t_i)$	2-4
$\mathbf{a}_k$	2-4
$\mathbf{P}_k(t_i^-)$	2-4
$\mathbf{Z}(t_{i-1})$	2-4
$\mathbf{P}_{mmaf}(t_i^+)$	2-5

Symbol	Page
$G(t)$	2-7
$K(t_i)$	2-7
$R(t_i)$	2-7
(t_i^-)	2-8
(t_i^+)	2-8
H	2-8
(t/t_i)	2-8
F	2-8
x_t	3-1
x_{FLIR}	3-1
x_b	3-2
x_c	3-2
x_p	3-2
θ_T	3-2
y_t	3-2
y_a	3-2
y_b	3-2
$x_T(t)$	3-3
Φ_T	3-3
I	3-3
LOS	3-6
F_G	3-10
G	3-10
m_1, m_2	3-10
K_a	3-11
ω_1	3-11
ω_2	3-11

Symbol	Page
$\delta(\tau)$	3-12
$E[\cdot]$	3-12
K_b	3-14
ζ_b	3-14
ω_{nb}	3-14
Δt	3-16
σ_b	3-16
ω_b	3-16
K_p	3-17
ζ_p	3-17
ω_{np}	3-17
I_{max}	3-20
σ_v^2	3-20
σ_{pv}^2	3-20
$z_{jk}(t_i)$	3-22
A_p	3-22
I_1, I_2	3-22
$n_{jk}(t_i)$	3-22
$b_{jk}(t_i)$	3-22
'pv'	3-24
'ppv'	3-24
α	3-24
β	3-24
α'	3-24
β'	3-24
r_o	3-33
$v_{\perp LOS}$	3-33

Symbol	Page
δ	3-33
AR	3-33
v_x	4-2
v_y	4-2
a_x	4-2
a_y	4-2
v_p	4-2
$x_f(t)$	4-3
F_f	4-3
K_{pf}	4-4
ω_{npf}	4-4
ζ_{pf}	4-4
τ_x, τ_y	4-4
$\hat{x}_f(t_i)$	4-5
$\Phi_f(\Delta t)$	4-5
Q_{df}	4-8
NRFOV	4-9
$\hat{I}(t_i)$	4-11
$I(t_i)$	4-11
γ	4-11
$F\{\cdot\}$	4-14
$g(x, y) * l(x, y)$	4-14
$g(x, y)$	4-14
$l(x, y)$	4-14
$G(f_x, f_y)$	4-14
$L^*(f_x, f_y)$	4-14
$h_f[\cdot]$	4-15

Symbol	Page
R_f	4-15
H_f	4-16
$\hat{\theta}_f$	5-7
σ_p	A-1

Abstract

This thesis is an extension of the work performed over the past ten years at the Air Force Institute of Technology (AFIT) towards tracking of airborne targets using forward looking infrared (FLIR) measurements. The research has aimed at replacing a standard correlation tracker with a hybrid Kalman filter/enhanced correlation tracker for implementation in a high energy laser weapon.

This research deviates somewhat from past research at AFIT in that the target trajectory being tracked is modelled as a benign, non-maneuvering, thrusting ballistic missile trajectory at large sensor-to-target ranges. In addition, to capture the characteristic shape of the exhaust plume, the plume is modelled as the difference between two bivariate Gaussian functions with elliptical equal intensity contours. As the missile ascends on its thrusting trajectory, the exhaust plume tends to oscillate (pogo) along the direction of the velocity vector. In this thesis, a second-order Gauss-Markov process is used to model the plume's "pogo" oscillation properties.

The ultimate goal of this research effort is to design a multiple model adaptive filter (MMAF) algorithm composed of elemental filters tuned for varying plume pogo parameters (frequency and amplitude characteristics). This MMAF accounts for atmospheric disturbance effects of the propagating infrared wave fronts, as well as bending/vibrational effects of the optical hardware associated with the FLIR sensor. The bank of filters provide the accurate estimation capability to guide the pointing mechanism of a shared aperture laser/FLIR sensor.

An 8×8 -pixel tracking field of view (FOV) of the FLIR sensor provides the infrared data to the enhanced correlation tracking algorithm. To enhance performance of the tracking algorithm, a FOV rotation scheme is analyzed in an effort to maintain accurate tracking of a plume undergoing the pogo phenomenon. A FLIR

rotation scheme which aligns the diagonal dimension of the 8×8 -pixel tracking window with the missile velocity vector demonstrates a 50% performance improvement over a non-rotating FOV FLIR.

A benchmark of performance involving an eight-state Kalman filter is established in order to compare results from various tracking enhancement techniques. The eight-state filter excludes explicit modelling of the pogo phenomenon, but the pogo effect is compensated by the addition of pseudo-noise in the filter model. To implement the MMAF, a ten-state filter which models the additional two pogo states is analyzed, and results are compared to the eight-state filter benchmark for performance enhancement. The ten-state filter consistently showed an unexpected performance degradation compared to the eight-state filter. Various trouble-shooting techniques are employed to uncover the source(s) of this degradation. Possible problems include: (1) a pogo-atmospheric jitter interaction, (2) poor estimation by the Kalman filter atmospheric jitter model and (3) observability issues of the target dynamics model. Recommendations to overcome these shortcomings are proposed in order to enhance performance of the ten-state filter and eventually implement the MMAF algorithm.

ENHANCED TRACKING OF BALLISTIC TARGETS USING FORWARD LOOKING INFRARED MEASUREMENTS

I. Introduction

The laser beam has had an enormous impact on our present society. From industrial, to medical, to military applications, the laser has received a tremendous amount of attention and investigation. The laser's ability to transmit energy instantaneously onto a target makes it extremely attractive as a potential weapon. With the recent United States' attention on the Strategic Defense Initiative (SDI), the laser beam has become a prime candidate as a potential weapon system.

Critical to the deposition of laser energy is the ability to track a potential target accurately. Precise tracking and laser pointing would enable the beam to concentrate its energy on a small area of the target. This is essential since the amount of energy in a laser beam is limited; and without the accurate tracking system, the laser's destructive effect would be rendered useless. This requirement for accurate tracking is the motivation for this and previous research efforts.

1.1 Background

The Air Force Weapons Laboratory at Kirtland AFB, New Mexico, is presently researching high energy laser weapons to be used against airborne vehicles. The targets are passively tracked by means of a forward looking infrared (FLIR) sensor. This tracker uses a 300 x 500 array of picture elements (pixels) to sense the target's radiated infrared (IR) energy. Each pixel in the array can effectively "see" or detect the target's radiated IR energy through an angle of 20 micro-radians in two orthogonal directions. In actual implementation of the tracking algorithm, an 8 x 8

subset of pixels (a "tracking window") processes the radiated energy. This window defines the tracker's current field-of-view (FOV).

The tracking algorithm will process the FLIR data and detect any angular offset between the target's actual position and the center of the current FOV. The measured offsets are regulated to zero by a pointing controller, thereby causing the target's image to be centered in the FLIR FOV. As the image is centered in the FOV, the laser is automatically pointed at the target, since the inbound IR energy shares the same aperture as the outbound laser energy.

Presently, a correlation algorithm [20] is being used to accomplish the tracking functions. This algorithm compares the current FLIR measurement data to the corresponding data from the previous sample time. By cross-correlating this data, the algorithm generates relative position offsets, since a detected translation in the image is assumed to be a translation of the actual target in the spatial domain. Because the correlation tracker assumes no prior information concerning the type of target to be tracked, it performs reasonably well against a variety of targets, but it does have some inherent weaknesses.

First, in many tracking scenarios, the size, shape, and motion characteristics of the target may be known or can be estimated adaptively on-line. This available information, although not required by the correlation tracker, can be used to enhance the performance of the tracker. Secondly, a time lag is inherent to the correlation tracker. This lag is a combination of the time required to cross-correlate the present image with the previous image, and the finite time necessary to point the tracker at the target. The correlation algorithm provides no means of estimating future target positions. Lastly, the traditional correlation algorithm cannot distinguish between actual target motion and "apparent" target motion caused by identifiable physical phenomena. These phenomena can include atmospheric "jitter" [12, 16], caused by distorted wavefronts of the inbound IR energy; bending/vibration of the optical

hardware or platform [5]; and missile exhaust plume "pogo" effects [10], caused by pressure variations over the time of flight and over the length of the missile's hardbody. These weaknesses in the correlation algorithm motivate the incorporation of Kalman filtering techniques [7] into the tracking system [16].

The target dynamics, atmospheric jitter, optical bending/vibration, and plume pogo effects can be modeled and included in the Kalman filter dynamics model. By assuming that the measurements from the FLIR image plane are a composite sum of these effects and additional noise disturbances, an estimate of the target's actual position can be obtained. By developing an appropriate target dynamics model, this estimate can be propagated forward in time to establish an estimate of target position in the future. The Kalman filter used in this research will model target dynamics, atmospheric jitter, and plume pogo effects (see Section 1.3.4) by explicit states. The filter will not model the bending/vibration phenomenon via explicit filter states, but this effect will be included in the real-world truth model. Tuning of the filter to this truth model will compensate for the reduced-order structure of the filter model.

1.2 Summary of Previous AFIT Research

Over the past nine years, the staff and students at the Air Force Institute of Technology (AFIT) have produced numerous theses and research papers investigating the potential use of Kalman filtering techniques with the Air Force Weapons Laboratory's high energy laser pointing and tracking system. An overview of this work has been presented in previous AFIT thesis research [2, 3, 4, 5, 6, 16, 17, 18, 19, 21, 22, 23, 24], as well as publications [11, 12, 13, 14, 25]. That overview will be reproduced in this section with some modification.

In 1978, Mercier [16] began the study by demonstrating that the Extended Kalman filter (EKF) algorithm could significantly outperform the traditional correlation tracker at design conditions. The target models used in this study were

long-range targets represented as infrared point sources of radiated energy. The target's FLIR plane image was assumed to have a bivariate Gaussian distribution with circular, equal-intensity contours. The filter model consisted of four states — two states representing position of benign target dynamics in each of two FLIR plane coordinate directions and two states representing atmospheric jitter in the same two directions. The position and jitter states were each modelled via a first-order, zero-mean, Gauss-Markov (GM) process. FLIR measurement noise, corresponding to both background clutter effects and internal FLIR noises such as thermal noise and dark current, was considered to be both temporally and spatially uncorrelated. This enhanced correlation algorithm provided an order of magnitude performance improvement over the traditional correlation algorithm when the filter was correctly informed about the tracking environment characteristics. These desirable results led to further research in the area of enhancing the original tracking algorithm.

To accommodate tracking of more maneuverable targets, Harnly and Jensen [3] incorporated velocity and acceleration estimates into the filter structure. They also modelled the FLIR plane image with elliptical equal- intensity contours versus circular contours to account for target shape effects, as well as adaptively estimating the target's shape function. Additionally, a spatially correlated Gaussian measurement noise model was incorporated to represent the correlation distance characteristics of typical background clutter. Finally, they implemented an algorithm to estimate the strength of the Kalman filter's driving noise adaptively as the target performed a maneuver.

The research thus far assumed that the shape of the FLIR plane image was known a priori and could be modelled via a bivariate Gaussian distribution. Research by Singletery [22] and Rogers [21] implemented algorithms which made no such target shape assumptions, but instead produced an estimate of the target's shape via a finite-memory averaging technique which avoids the problem of large memory requirements by using exponential smoothing as an approximation to true

finite-memory averaging. Tracking scenarios which modelled the targets as having multiple hot-spots and several dynamic angular orientations were used in evaluating the algorithms.

Rogers additionally developed an enhanced correlation tracker algorithm which generated "pseudo-measurements" as its output. This algorithm was "enhanced" over the traditional correlator because it compared the current FLIR image with a template instead of the previous image. The template was actually the target shape function estimate described in the preceding paragraph. The pseudo-measurements were position offsets between the target image and the center of the FOV in two orthogonal directions on the FLIR plane. The offsets were then fed into a linear Kalman filter for processing. Since the filter's dynamics model and measurement model were now linear, an extended Kalman filter was no longer required. This model was extremely attractive from a computational loading standpoint, since a linear Kalman filter requires much less computer resource allocation than the EKF. The enhanced correlation tracker was additionally attractive since the performance was comparable to the previously used EKF tracking algorithm in many applications.

Kozemchak [4] and Millner [17] continued the research by testing the EKF algorithm and the linear Kalman filter/enhanced correlation algorithm developed by Rogers with close range, highly maneuverable targets. This research modelled the target dynamics using a first-order Gauss-Markov acceleration process, as well as a constant turn-rate dynamics model. In an effort to maintain lock on harshly maneuvering targets, adaptive estimation of the filter's driving noise strength was again implemented. Performance was good for targets with limited maneuvering capabilities; but when the maneuver exceeded five g's, the filter performance degraded considerably. Ad hoc adaptive compensation techniques were considered, but not thoroughly evaluated.

To overcome this high maneuverability limitation, Flynn [2] investigated multiple model adaptive filtering (MMAF) techniques in the algorithm. See Figure 2.1

in Chapter 2 where the theory of multiple model adaptive filtering is discussed. Suizu [23] followed up the research by Flynn and successfully implemented the MMAF into the algorithm. The MMAF contained a bank of two elemental filters, each tuned for different target maneuvers. One filter was tuned for benign target dynamics and processed measurements from an 8 x 8 pixel FOV. The second filter was tuned for a highly maneuverable target and processed measurements based on a 24 x 24 pixel FOV. The field of view was expanded in the second filter as added insurance in maintaining lock on the harshly maneuvering target. Based upon probabilistic weightings of Bayesian MMAF theory [8:129-136], the tracker performed extremely well, tracking targets whose dynamics ranged from benign maneuvers to 20-g pull-up maneuvers at 20 kilometers. The elemental filters used in the bank were implemented using both the EKF and linear Kalman filter/enhanced correlator with similar results.

Loving [6] continued the MMAF research, adding a third elemental filter to the bank. This additional filter processed measurements from an 8 x 8 FOV array and was tuned for intermediate levels of target maneuverability. The three-bank MMAF showed significant performance over the previously used filters. Additionally, she developed a Maximum a posteriori (MAP) MMAF algorithm for comparison to the Bayesian MMAF. The MAP algorithm uses the same elemental filters as the Bayesian approach; but the MAP filter outputs the estimates of the individual elemental filter with the highest probability weighting, as opposed to the sum of probabilistic weighted estimates which are output by the Bayesian MMAF. Both MMAF techniques performed favorably against a variety of target maneuvers, while no significant performance differences were noted between the Bayesian and MAP comparisons.

Follow-on research by Netzer [18] expanded Loving's analysis with the three-elemental-filter Bayesian MMAF. The existence of steady state bias errors when tracking a target that executed a 20-g turn led to the investigation of multiple model

adaptive filtering based on some elemental filters being tuned for dynamics predominantly in the azimuth or the elevation directions. Using this technique, maneuvers in the x-direction can be distinguished from maneuvers in the y-direction, therefore permitting the tracker to expand its FOV in the critical direction and maintain lock on a maneuvering target while maintaining accurate tracking estimates in the non-critical direction. In addition, since the zero-mean, Gauss-Markov acceleration process might not adequately describe target dynamics for all situations, Netzer suggested using a constant turn rate (CTR) model [15] at close ranges. Although this model was investigated previously by Kozemchak [4], it was never implemented with the enhanced correlation algorithm developed by Rogers [21].

Tobin [24] continued with the recommendations by Netzer, specifically implementing the constant turn rate dynamics model into the elemental filters of the MMAF bank. His results showed that the CTR model exhibited smaller steady state standard deviation errors, while the GM MMAF showed smaller bias errors, but that they both possessed very comparable rms errors. Tobin also investigated the inclusion of two rectangular FOV elemental filters in the MMAF bank, tuned specifically for maneuvers in the x- and y-directions. Results indicated that the tracker maintained lock on the target during a "jink" in the y-direction while maintaining substantially better tracking performance in the x-direction than attainable with an MMAF without any elemental filters tuned for specific directionality of maneuvers.

Leeney [5] continued with the research effort by applying the MMAF algorithm based on Gauss-Markov acceleration models to a truth model where the bending/vibrational effects of a large space structure were modelled. Even though the filter was not provided with the bending/vibrational information, nor were any states augmented to compensate for this effect, the MMAF tracker was able to track a target exhibiting a 10-g maneuver, provided that the level of bending/vibration is on the order of that expected. Leeney also investigated performance enhancement by implementing a 50 Hertz (Hz) sampling rate versus the previously used 30 Hz sample

rate. A slight increase in performance was achieved (average 6% decrease in mean error and variance in both FLIR directions), but a large computer loading penalty was incurred. Additionally, Leeney did a preliminary investigation on a rotating rectangular field-of-view (RRFOV) filter so as to align the "elongated" side with the estimate of the acceleration vector. The intent was to replace the five-elemental-filter MMAF with a four-elemental-filter MMAF. Preliminary results warranted further investigation of the RRFOV filter.

Most recently, Norton [19] continued the investigation of the RRFOV. The choice of a larger filter dynamics driving noise strength ("Q") in the direction of a maneuver proved more important for improved filter performance than field-of-view size. Thus, by maintaining an 8 x 8 pixel rotating FOV (versus a rectangular rotating FOV) and employing a large filter "Q" value in the direction of the maneuver, he was able to improve the filter performance. He investigated a scheme to transform the "Q" matrix mathematically so that the larger entry stays aligned with the acceleration vector, as well as a scheme to simulate a physical rotation of the FLIR plane to keep one axis coincident with the acceleration vector. Separate elemental filters were tuned for varying target dynamics and eventually incorporated into a MMAF bank. Performance characteristics were encouraging enough to adapt this methodology to the current research area.

1.3 Objectives

Previous AFIT research has concentrated on the tracking of airborne targets using FLIR measurements and Kalman filtering techniques. The purpose of this thesis is to continue with this philosophy, but to apply the previous knowledge to the tracking of a ballistic missile target during its boost phase through the atmosphere. Since the linear Kalman filter/enhanced correlator algorithm has proven computationally more beneficial (with comparable performance results) than the EKF operating directly on raw FLIR data, it will be the algorithm of choice for this

thesis effort. Specific objectives and solution methods are outlined below.

1.3.1 Exhaust Plume Pogo Effects. During the thrusting phase of a ballistic missile trajectory, the exhaust plume that is generated inherently "pogos" or oscillates along the longitudinal axis of the missile hardbody. This pogoing will sometimes occlude the missile hardbody, causing a traditional correlation tracker operating on FLIR sensor input (that tracks the plume versus the missile) to provide poor estimates of hardbody location. The FLIR-based tracker will always track the highest intensity of the plume if a simple correlation tracker with no filter is used. The internal filter dynamics model is the means by which separation of hardbody dynamics from plume oscillations can be accomplished. A key element of this thesis will be to model the dynamics of the plume, via a second order Gauss-Markov process [7], in the truth model and eventually in the filter model as well.

1.3.2 Implementation of a Rotating Field-of-View. Based upon the investigations by Norton [19], the concept of the mathematical transformation (rotation) of the "Q" matrix will be applied to the ballistic missile target. The states representing the plume pogo will be aligned along the estimated velocity vector of the hardbody; thus the previously mentioned transformation will be used to determine the components of pogo in the azimuth and elevation directions on the FLIR sensor plane. Also, since the missile's hardbody will be modelled as having identical dynamic characteristics in each of the two directions on the FLIR plane, the direct pre- and post-multiplication of the "Q" matrix by the appropriate transformation matrix need not be employed as was done by Norton [19]. Three "physical" rotation schemes involving the FLIR image plane will be considered. The first scheme involves using an 8 x 8 FOV filter and aligning a single axis of the FLIR plane with the estimated velocity vector of the missile. By aligning one of the FLIR axes with the velocity vector, the FOV will stay oriented with the oscillation of the plume. This scheme will be referred to as the rotating field-of-view (RFOV) filter. The next rotation scheme

will be referred to as the diagonal field-of-view (DFOV) filter, where the diagonal of the 8 x 8 tracking window will be aligned with the estimate of the velocity vector. The motivation behind this scheme is that the "tracking window" oriented in such a fashion will be able to "see" more of the target's intensity image, thus enabling the sensor to obtain more measurement data. The final tracking scheme to be analyzed will be the rectangular rotating field-of-view (RRFOV) filter as initially addressed by Tobin [24] and Leeney [5]. This will be studied to confirm that pixel size is not as important as tuning considerations in filter performance.

1.3.3 Single Filter Benchmarks. To establish single filter benchmarks of performance, the truth model will include the modelling of the pogo effect while the pogo effect will be absent from the filters. A nominal damping ratio representing an underdamped response will be used in the truth model representation of the plume pogo. Since the amplitude and the undamped natural frequency of the pogo oscillation will most likely be the dominant parameters in filter performance [10], nine single filters will be analyzed for a range of predetermined values for the pogo parameters. Each of the different rotation schemes mentioned in Section 1.3.2 will be addressed.

1.3.4 Single Filter Performance. The purpose of this section is to improve the performance of the single filter benchmarks by adding the pogo models to the filter structure. This will increase the dimension of the filter but will give insights into anticipated performance improvements by informing the filter of the pogo phenomenon.

1.3.5 Single Filter Robustness Analysis. The purpose of this objective is to determine the robustness of the best performing rotating filters from the previous section. The tuned rotating filters from Section 1.3.4 will be tested against a truth model where the values of pogo parameters are mismatched with corresponding

pogo parameters in each filter. This study will provide insight into which of the pogo parameters affects filter performance sufficiently to warrant on-line adaptive estimation, as well as explore the possible applicability of MMAF techniques for accomplishing the adaptation.

1.3.6 Multiple Model Adaptive Filtering Benchmarks. Similar to the single filter benchmarks performed in Section 1.3.3, an MMAF benchmark will be established in which the pogo effects are modelled only in the truth model and not the elemental filters of the MMAF bank. The MMAF will be tested against seven different scenarios (involving pogo parameter variations), and the performance results compared to the results of the single filter performance.

1.3.7 Multiple Model Adaptive Filtering Performance. The pogo effect is modelled in both the truth model scenarios and the individual elemental filters of the MMAF. The elemental filters used in the bank will be from Section 1.3.4, and the seven scenarios run in Section 1.3.6 will be repeated. Since each of the elemental filters in the bank is made aware of the pogo phenomenon (with the exception of one elemental filter, to be discussed in Chapter V), this MMAF should outperform all preceding filters.

1.4 Overview of the Thesis.

This chapter has presented a review of the research efforts performed to date in developing an implementable tracking algorithm, and it has also defined the areas to be pursued in this thesis effort. Chapter II introduces the concept of multiple model adaptive filtering which is required for a better understanding of the tracking algorithm. Chapter III develops the dynamics and measurement models used to simulate the real-world environment to evaluate the tracking algorithm's performance. The dynamics and measurement models embedded into the Kalman filter structure are developed in Chapter IV. Chapter V discusses the tracking algorithms used to

incorporate the concepts of Chapters II and IV. Methods for evaluating the tracker's performance are also presented in Chapter V. The results of the Monte Carlo simulations are analyzed in Chapter VI, while Chapter VII presents the final conclusions of this research effort and provides recommendations for further study.

II. Filter Theory

The basic purpose of this chapter is to present the mathematical forms and algorithmic structure of the multiple model adaptive filter (MMAF) and the extended Kalman filter (EKF). This review is necessary for understanding the development and analysis of the tracking algorithm presented in this thesis. Rigorous mathematical developments for the MMAF technique and the EKF can be found in references [8:129-136] and [8:39-59], respectively. It is assumed that the reader already has a basic understanding of linear Kalman filtering techniques [7].

2.1 Bayesian Multiple Model Adaptive Filtering

When dealing with Kalman filter tracking applications, maximum performance is achieved when the parameters of the filter dynamics model match the parameters of the actual target being tracked. In many real world applications, the parameters of interest may be time-varying, and the designer may not have a priori knowledge of the optimal parameter values for a given scenario. Thus, to achieve good filter performance, on-line adaptability is essential. One means of providing this on-line adaptability is by multiple model adaptive filtering as presented in references [5, 6, 8, 18, 19, 23, 24]. For physical problems in which parameters can assume values in a continuous range, it becomes necessary to discretize the parameter space to keep the algorithm computationally tractable. Consider a target which can display K different discrete sets of target dynamics. No one single vector value of parameters, $\mathbf{a}$, is adequate to describe all of the different dynamic scenarios. To achieve maximum performance, it is desired to match the k th possible parameter vector, $\mathbf{a}_k$, where $k = 1, 2, 3, \dots, K$, to the k th target dynamics characteristic. Multiple model adaptive filtering is one way to accomplish this objective.

As developed in [8], consider a system model represented by the following first-order, linear, stochastic differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{a})\mathbf{x}(t) + \mathbf{B}(\mathbf{a})\mathbf{u}(t) + \mathbf{G}(\mathbf{a})\mathbf{w}(t) \quad (2.1)$$

with noise corrupted, discrete-time measurements given by:

$$\mathbf{z}(t_i) = \mathbf{H}(\mathbf{a})\mathbf{x}(t_i) + \mathbf{v}(t_i) \quad (2.2)$$

where:

- $\mathbf{x}(t)$ = n -dimensional system state vector
- $\mathbf{u}(t)$ = r -dimensional deterministic control vector
- $\mathbf{w}(t)$ = s -dimensional white Gaussian, zero-mean
noise vector process of strength $\mathbf{Q}(\mathbf{a})$
- $\mathbf{z}(t_i)$ = m -dimensional measurement vector
- $\mathbf{v}(t_i)$ = m -dimensional discrete-time white Gaussian,
noise vector process of covariance $\mathbf{R}(\mathbf{a})$
- $\mathbf{F}(\mathbf{a})$ = $n \times n$ system plant matrix
- $\mathbf{B}(\mathbf{a})$ = $n \times r$ input distribution matrix
- $\mathbf{G}(\mathbf{a})$ = $n \times s$ noise distribution matrix
- $\mathbf{H}(\mathbf{a})$ = $m \times n$ matrix relating measurements to states.

As mentioned previously, it is necessary to discretize $\mathbf{a}$ into a set of K finite vector values, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$. As depicted in Figure 2.1 [8], the MMAF consists of a bank of elemental Kalman filters, each of which is tuned for a specific dynamic scenario represented by the appropriate vector, $\mathbf{a}_k$, where $k = 1, 2, 3, \dots, K$. Each of the K elemental Kalman filters produces a state estimate which is weighted appropriately using the hypothesis conditional probability $p_k(t_i)$ to produce the state estimate $\hat{\mathbf{x}}_{mmaf}(t_i)$ as a probabilistically weighted sum, where:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a}, \mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_k, \mathbf{Z}_{i-1})p_k(t_{i-1})}{\sum_{j=1}^K f_{\mathbf{z}(t_i)|\mathbf{a}, \mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_j, \mathbf{Z}_{i-1})p_j(t_{i-1})} \quad (2.3)$$

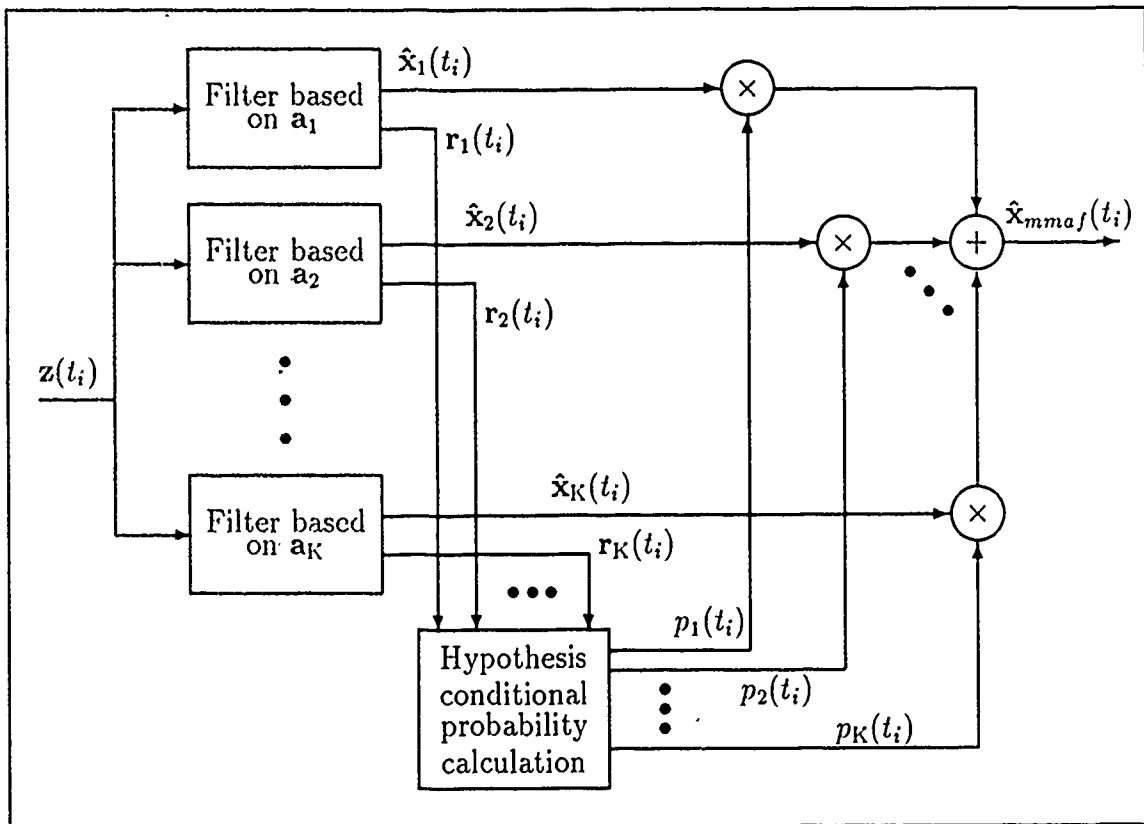


Figure 2.1. Multiple Model Filtering Algorithm

$$f_{z(t_i)|a, Z(t_{i-1})}(z_i|a_k, Z_{i-1}) = \frac{\exp\{\cdot\}}{(2\pi)^{m/2} |A_k(t_i)|^{1/2}} \quad (2.4)$$

$$\{\cdot\} = \left\{ -\frac{1}{2} r_k^T(t_i) A_k^{-1}(t_i) r_k(t_i) \right\} \quad (2.5)$$

$$\begin{aligned} A_k(t_i) &= k\text{th filter's computed residual covariance} \\ &= H_k(t_i) P_k(t_i^-) H_k^T(t_i) + R_k(t_i) \end{aligned} \quad (2.6)$$

$$\begin{aligned} r_k(t_i) &= k\text{th filter's residual} \\ &= [z(t_i) - H_k(t_i) \hat{x}_k(t_i^-)] \end{aligned} \quad (2.7)$$

and

$$\begin{aligned} a_k &= \text{parameter value assumed in the } k\text{th filter} \\ P_k(t_i^-) &= k\text{th filter's computed state error covariance} \\ &\quad \text{before incorporating the measurement at time } t_i \\ Z(t_{i-1}) &= \text{measurement history up to time } t_{i-1}. \end{aligned}$$

This hypothesis conditional probability identifies which of the elemental filters has the greatest probability of providing the best performance at a given time. As can be seen from Equation (2.3), $p_k(t_i)$ is the ratio of a numerator product and a denominator of a sum of such products. The numerator is the k th filter's product of its previous hypothesis probability and the conditional probability density of the current measurement given the k th filter's assumed parameter value and the previous measurement history. The denominator is the sum of the same products for all K elemental filters in the MMAF bank. When the k th filter is the best match for the current target dynamics, that filter will produce the smallest squared residual relative to the filter-computed residual covariance of the K filters. This will cause Equation (2.5) to become a smaller magnitude negative quantity and Equation (2.4) to be larger for the k th filter than for the other $K - 1$ filters. The ratio in Equation (2.3) will now be the largest value for the k th filter, i.e., the filter that best matches the current target dynamics. It is essential that the residuals of the "best-matched" filter be distinguishable from those of the mismatched filters. If this distinction is not

obtainable, large probabilities can be assigned to incorrect models, resulting in poor performance in the MMAF algorithm. To overcome the possibility of such degraded performance, each of the elemental filters should be tuned for best performance against a specific target scenario to match its own internal dynamics model [5, 18]. Additionally, to prevent masking the distinction between good and bad models, the common practice of adding excessive amounts of pseudonoise to compensate for model inadequacies should be minimized. This is an important point, since if the residuals are constantly of the same magnitude, then Equations (2.3) and (2.5) will result in large p_k values associated with the filter with the smallest $|\mathbf{A}_k|$. Because $|\mathbf{A}_k|$ is independent of the residuals and the "correctness" of the K models, such a result would be totally in error [8].

As can be seen in Figure 2.1, each of the K filters processes its own estimates and residuals in parallel. Each filter can also generate its own numerator term out of Equation (2.3). The recursion is then run at each sample time and a $p_k(t_i)$ assigned for each filter. The output of the recursion is the estimate, $\hat{\mathbf{x}}_{mmaf}(t_i^+)$, which is the probabilistically weighted average of the elemental filters' estimates [5:19]:

$$\hat{\mathbf{x}}_{mmaf}(t_i^+) = \sum_{k=1}^K p_k(t_i) \hat{\mathbf{x}}_k(t_i^+) \quad (2.8)$$

The conditional covariance matrix for the MMAF is computed as follows [5]:

$$\mathbf{P}_{mmaf}(t_i^+) = \sum_{k=1}^K p_k(t_i) [\mathbf{P}_k(t_i^+) + \hat{\mathbf{y}}_k(t_i^+) \hat{\mathbf{y}}_k^T(t_i^+)] \quad (2.9)$$

where:

$$\begin{aligned} \hat{\mathbf{y}}_k(t_i^+) &= \hat{\mathbf{x}}_k(t_i^+) - \hat{\mathbf{x}}_{mmaf}(t_i^+) \\ p_k &= k\text{th filter's conditional hypothesis probability} \\ \mathbf{P}_k(t_i^+) &= k\text{th filter's state error covariance matrix after} \\ &\quad \text{incorporating the measurement at time } t_i. \end{aligned}$$

Since the values of $p_k(t_i)$ and $\hat{\mathbf{x}}_{mmaf}(t_i^+)$ depend upon the discrete measurements taken through time t_i , $\mathbf{P}_{mmaf}(t_i^+)$ cannot be computed a priori as is the case for

each of the elemental linear Kalman filters. Note that Equation (2.9) need not be calculated for an on-line implementation of the MMAF.

Finally, the calculated probabilities of Equation (2.3) should involve an artificial lower bound [5, 8, 18]. This lower bound will prevent a mismatched filter's hypothesis conditional probability from converging to zero. If a filter's p_k should reach zero, it will remain zero for all time since it is a function of the previous conditional probability, as depicted in Equation (2.3). This "zeroing" of the hypothesis conditional probability effectively removes that filter from the bank, and can degrade the MMAF's ability to respond to future changes in the true future parameter values. If some future target dynamic scenario matched the model for which the probability was permitted to reach zero, that elemental filter would not be appropriately weighted, and the MMAF estimate would be in error. In previous work, Loving [6] established a lower bound of .001 for $p_k(t_i)$. The use of this lower bound value will be continued in this study.

2.2 The Extended Kalman Filter

An extended Kalman filter (EKF) provides the means by which the states of a nonlinear stochastic system can be estimated. Paralleling the linear Kalman filter, the EKF is composed of a sequence of propagation and update cycles. The extended Kalman filter is a first-order nonlinear filter. The nonlinear dynamics and measurement equations are expanded in a Taylor series about the most recent value of the state estimate [8]. The series is then truncated at first order terms, resulting in the EKF formulation. Since the Taylor series expansion is truncated to first-order terms, the EKF does not produce an optimal state estimate as is the case with the linear Kalman filter [7:231-236]. A complete development of the EKF algorithm can be found in reference [8]. The results of that development are now presented.

Consider a system described by the following nonlinear stochastic differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \quad (2.10)$$

where:

- $\mathbf{x}(t)$ = n -dimensional state vector
- $\mathbf{u}(t)$ = r -dimensional vector of known control inputs
- t = time
- $\mathbf{w}(t)$ = zero-mean, white Gaussian s -vector process of strength $\mathbf{Q}(t)$; independent of $\mathbf{x}(t_0)$
- $\mathbf{G}(t)$ = $n \times s$ noise distribution matrix.

Furthermore, assume that sampled-data measurements are available at discrete time increments and are modeled by the following nonlinear vector function:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \quad (2.11)$$

where:

- $\mathbf{z}(t_i)$ = m -dimensional measurement vector
- $\mathbf{v}(t_i)$ = zero-mean, white Gaussian m -vector process with covariance $\mathbf{R}(t_i)$; independent of both $\mathbf{x}(t_0)$ and $\mathbf{w}(t)$ for all time.

The extended Kalman filter update cycle incorporates the measurement $\mathbf{z}(t_i)$ by:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T\{\mathbf{H}\mathbf{P}(t_i^-)\mathbf{H}^T + \mathbf{R}(t_i)\}^{-1} \quad (2.12)$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\{\mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i]\} \quad (2.13)$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}\mathbf{P}(t_i^-) \quad (2.14)$$

where:

- $\mathbf{P}(t_i)$ = $n \times n$ filter covariance matrix
- (t_i^-) = instant immediately before measurements are incorporated at time t_i
- (t_i^+) = instant immediately after measurements are incorporated at time t_i

$$\mathbf{H} = \mathbf{H}[\hat{\mathbf{x}}(t_i^-), t_i] \equiv \left. \frac{\partial \mathbf{h}[\mathbf{x}, t_i]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t_i^-)} \quad (2.15)$$

The extended Kalman filter propagation cycle propagates the state estimate and state error covariance matrix forward to time t_{i+1} by integrating the following equations from t_i to t_{i+1} , using the results of the update cycle as the initial conditions:

$$\hat{\mathbf{x}}(t/t_i) = \mathbf{f}[\hat{\mathbf{x}}(t/t_i), \mathbf{u}(t), t] \quad (2.16)$$

$$\dot{\mathbf{P}}(t/t_i) = \mathbf{F}\mathbf{P}(t/t_i) + \mathbf{P}(t/t_i)\mathbf{F}^T + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \quad (2.17)$$

where:

- (t/t_i) = estimate at time t given measurements through time t_i

$$\mathbf{F} = \mathbf{F}[\hat{\mathbf{x}}(t/t_i), t_i] \equiv \left. \frac{\partial \mathbf{f}[\mathbf{x}, \mathbf{u}, t]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t/t_i)} \quad (2.18)$$

Note that, for the case of linear vector functions $\mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t]$ and $\mathbf{h}[\mathbf{x}(t), t_i]$, the above propagation/update cycles reduce to the standard linear Kalman filter propagation/update cycles. Since the linear system model is totally representative of the first-order terms of a Taylor series expansion, the EKF propagation/update equations reduce to the linear Kalman filter algorithm [7].

2.3 Summary

This chapter has introduced the concepts of multiple model adaptive filtering and the extended Kalman filter. The intent was to provide some basic insight into

the formulation and applicability of both techniques. A more detailed development can be found in reference [8]. This chapter provides a basic understanding of the theory to be applied to the ensuing tracking algorithm and filter implementation.

III. Truth Model

3.1 Introduction

A "truth model" is an accurate simulation of "real world" effects. The truth model depicts, as best as possible, the dynamic activity of interest of a specified system. It is the standard used to determine the filter's errors and overall performance. More states are generally required to describe the truth model than the model upon which the Kalman filter is based. The less dominant states are normally omitted from the filter structure to accommodate on-line implementation on an operational computer system. One accounts for the decreased filter order by injecting white, Gaussian noise into the model upon which the Kalman filter is based.

For this thesis, the dynamics of the target's image on the FLIR detector plane are a result of true target motion, atmospheric jitter due to distorted infrared wavefronts, bending/vibration of the optical hardware, and pogo effects of the exhaust plume's oscillatory nature. If x_c and y_c represent the distances, measured in pixels, of the apparent image intensity centroid from the center of the FOV in the x and y FLIR plane directions, respectively, then

$$x_c = x_t + x_a + x_b + x_p \cos \theta_T \quad (3.1)$$

$$y_c = y_t + y_a + y_b + x_p \sin \theta_T \quad (3.2)$$

where:

- x_t = component of x_c due to actual target dynamics in the x_{FLIR} direction, measured in pixels
- x_a = component of x_c due to atmospheric jitter in the x_{FLIR} direction, measured in pixels

- x_b = component of x_c due to bending/vibration in the x_{FLIR} direction, measured in pixels
- x_p = component of x_c due to plume pogo along the missile velocity direction, measured in pixels
- θ_T = TTrue target orientation angle (see Section 3.3)
- y_t = component of y_c due to actual target dynamics in the y_{FLIR} direction, measured in pixels
- y_a = component of y_c due to atmospheric jitter in the y_{FLIR} direction, measured in pixels
- y_b = component of y_c due to bending/vibration in the y_{FLIR} direction, measured in pixels

Note that Equation (3.2) has a minus sign before the resolved pogo component. This is because of the defined orientations of the Target and FLIR coordinate frames (see Section 3.4.1). It will be shown that, in order to model $x_t, x_a, x_b, x_p, y_t, y_a$, and y_b adequately, fourteen stochastic differential equations are necessary. Of the seven output states, x_t and y_t each require first-order differential equations; x_a and y_a each require third-order; and x_b, y_b and x_p each require second-order. These differential equations, when arrayed in state-space format, comprise the dynamics portion of the FLIR tracker truth model used in this study. Section 3.2 presents this dynamics model as the augmentation of the deterministic target trajectory component (Section 3.2.1), the atmospheric jitter component (Section 3.2.2), the bending/vibration component (Section 3.2.3), and the plume pogo component (Section 3.2.4).

Following the presentation of the dynamics model, the measurement portion of the FLIR tracker truth model is presented in Section 3.3. Then, to implement the simulation on a digital computer, a "simulation space" model is presented in Section 3.4.

3.2 Dynamics Model

The overall fourteen-state dynamics model is the augmentation of a two-state, deterministic target dynamics model, a six-state, stochastic atmospheric jitter model, a four-state stochastic bending/vibration model, and a two-state, stochastic plume pogo model. This augmented system is described by the following linear, stochastic differential equation:

$$\dot{\mathbf{x}}_T(t) = \mathbf{F}_T \mathbf{x}_T(t) + \mathbf{B}_T \mathbf{u}_T(t) + \mathbf{w}_T(t) \quad (3.3)$$

where:

- $\mathbf{F}_T$ = 14×14 time-invariant truth model plant matrix
- $\mathbf{x}_T(t)$ = 14-dimensional truth model state vector
- $\mathbf{B}_T$ = 14×2 time-invariant truth model distribution matrix
- $\mathbf{u}_T(t)$ = 2-dimensional deterministic input vector
- $\mathbf{w}_T(t)$ = 14-dimensional, zero-mean, white, Gaussian noise vector process with autocorrelation function:

$$E[\mathbf{w}_T(t)\mathbf{w}_T^T(t+\tau)] = \mathbf{Q}_T\delta(\tau). \quad (3.4)$$

The equivalent discrete-time solution [7] to Equation (3.3) is given by:

$$\mathbf{x}_T(t_{i+1}) = \Phi_T(t_{i+1}, t_i)\mathbf{x}_T(t_i) + \mathbf{B}_{Td}\mathbf{u}_{Td}(t_i) + \mathbf{w}_{Td}(t_i) \quad (3.5)$$

where the state transition matrix $\Phi_T(t_{i+1}, t_i)$ is given from solving the differential equation [7:40-41]:

$$\frac{d\Phi_T(t, t_i)}{dt} = \mathbf{F}_T \Phi_T(t, t_i) \quad (3.6)$$

with the initial condition: $\Phi_T(t_i, t_i) = \mathbf{I}$.

and

- $\mathbf{x}_T(t_i)$ = 12-dimensional discrete-time truth model state vector
- $\mathbf{u}_{Td}(t_i)$ = 2-dimensional discrete-time input vector
- $\mathbf{w}_{Td}(t_i)$ = 12-dimensional discrete-time, zero-mean, white Gaussian noise with covariance:

$$\mathbf{Q}_{Td} = \int_{t_i}^{t_{i+1}} \Phi_T(t_{i+1}, \tau) \mathbf{Q}_T \Phi_T^T(t_{i+1}, \tau) d\tau. \quad (3.7)$$

where Q_T is defined in Equation (3.4). The discrete-time input distribution is defined as:

$$B_{Td} = \int_{t_i}^{t_{i+1}} \Phi_T(t_{i+1}, \tau) B_T d\tau \quad (3.8)$$

The internal structure of the discrete-time truth model consists of two target dynamic states (one for each FLIR plane direction), six atmospheric jitter states (three for each direction), four mechanical bending states (two for each direction), and two plume pogo states (oriented along the target's velocity vector). In augmented form, the truth model state vector is given by:

$$x_T = \begin{bmatrix} x_t \\ x_a \\ x_b \\ x_p \end{bmatrix} \quad (3.9)$$

The discrete-time truth model state transition matrix is given as:

$$\Phi_T = \begin{bmatrix} \Phi_{t(2 \times 2)} & | & 0_{(2 \times 2)} & | & 0_{(2 \times 4)} & | & 0_{(2 \times 2)} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ 0_{(6 \times 2)} & | & \Phi_{a(6 \times 6)} & | & 0_{(6 \times 4)} & | & 0_{(2 \times 2)} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ 0_{(4 \times 2)} & | & 0_{(4 \times 6)} & | & \Phi_{b(4 \times 4)} & | & 0_{(2 \times 2)} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ 0_{(2 \times 2)} & | & 0_{(2 \times 6)} & | & 0_{(2 \times 4)} & | & \Phi_{p(2 \times 2)} \end{bmatrix} \quad (3.10)$$

and the discrete-time truth model distribution matrix is given by:

$$\mathbf{B}_{Td} = \begin{bmatrix} \mathbf{B}_{dt(2 \times 2)} \\ - - - \\ \mathbf{0}_{(6 \times 2)} \\ - - - \\ \mathbf{0}_{(4 \times 2)} \\ - - - \\ \mathbf{0}_{(2 \times 2)} \end{bmatrix} \quad (3.11)$$

and the discrete-time truth model white Gaussian noise process is given by:

$$\mathbf{w}_{Td} = \begin{bmatrix} \mathbf{0}_{(2 \times 1)} \\ - - - \\ \mathbf{w}_{da(6 \times 1)} \\ - - - \\ \mathbf{w}_{db(4 \times 1)} \\ - - - \\ \mathbf{w}_{dp(2 \times 1)} \end{bmatrix} \quad (3.12)$$

where:

- $\mathbf{x}_t$ = 2-dimensional target dynamics state vector
- $\mathbf{x}_a$ = 6-dimensional atmospheric jitter state vector
- $\mathbf{x}_b$ = 4-dimensional bending/vibration state vector
- $\mathbf{x}_p$ = 2-dimensional plume pogo state vector
- $\mathbf{w}_{da}(t_i)$ = 6-dimensional discrete-time, white Gaussian noise related to atmospheric states
- $\mathbf{w}_{db}(t_i)$ = 4-dimensional discrete-time, white Gaussian noise related to bending states
- $\mathbf{w}_{dp}(t_i)$ = 2-dimensional discrete-time, white Gaussian noise related to pogo states.

From Equations (3.5) and (3.9) to (3.12), it can be seen that the truth model is in a block diagonal form, permitting the models for target dynamics, atmospheric jitter, bending/vibration, and plume pogo to be evaluated separately. The following subsections provide the details of those individual evaluations.

3.2.1 The Target States. The deterministic target dynamics of the ballistic missile are modelled as they occur on the FLIR image plane. In order to understand how the target states are propagated forward in time, the $\alpha - \beta$ plane must be introduced.

The $\alpha - \beta$ (FLIR image plane) coincides with the array of infrared sensing pixels. The FLIR plane is perpendicular to the sensor-to-target line-of-sight (LOS) vector, and bounded by a finite field-of-view (FOV). If the sensor-to-target range is large, then the FLIR "pseudo" azimuth (α') and the FLIR "pseudo" elevation (β') angles are directly proportional to the linear translational coordinates x_t and y_t on the FLIR plane. Note that the "pseudo" angles are referenced from the current LOS vector, while the true azimuth (α) and elevation (β) angles are referenced from true north and the horizon, respectively [24]. Figure 3.1 illustrates the relevant geometry. Therefore, if α' and β' are measured in micro-radians, and x_t and y_t are measured in pixels, then the pixel proportionality constant (k_p), used in Equations (3.13) and (3.14), is the angular FOV of a single pixel.

The pixel proportionality constant used in this research is on the order of 15 micro-radians/pixel versus the 20 micro-radian/pixel k_p used in previous studies. The reason for the reduction is that, in considering the bending/vibration of a spaceborne optical platform, the sensor-to-target range used in this study is approximately two orders of magnitude greater (see Figure 3.2) than previous work. This increase in range requires a finer resolution FLIR so that the 8×8 tracking window is able to "see" the plume's IR image.

Assuming that α' and β' remain constant over the Δt second sample period,

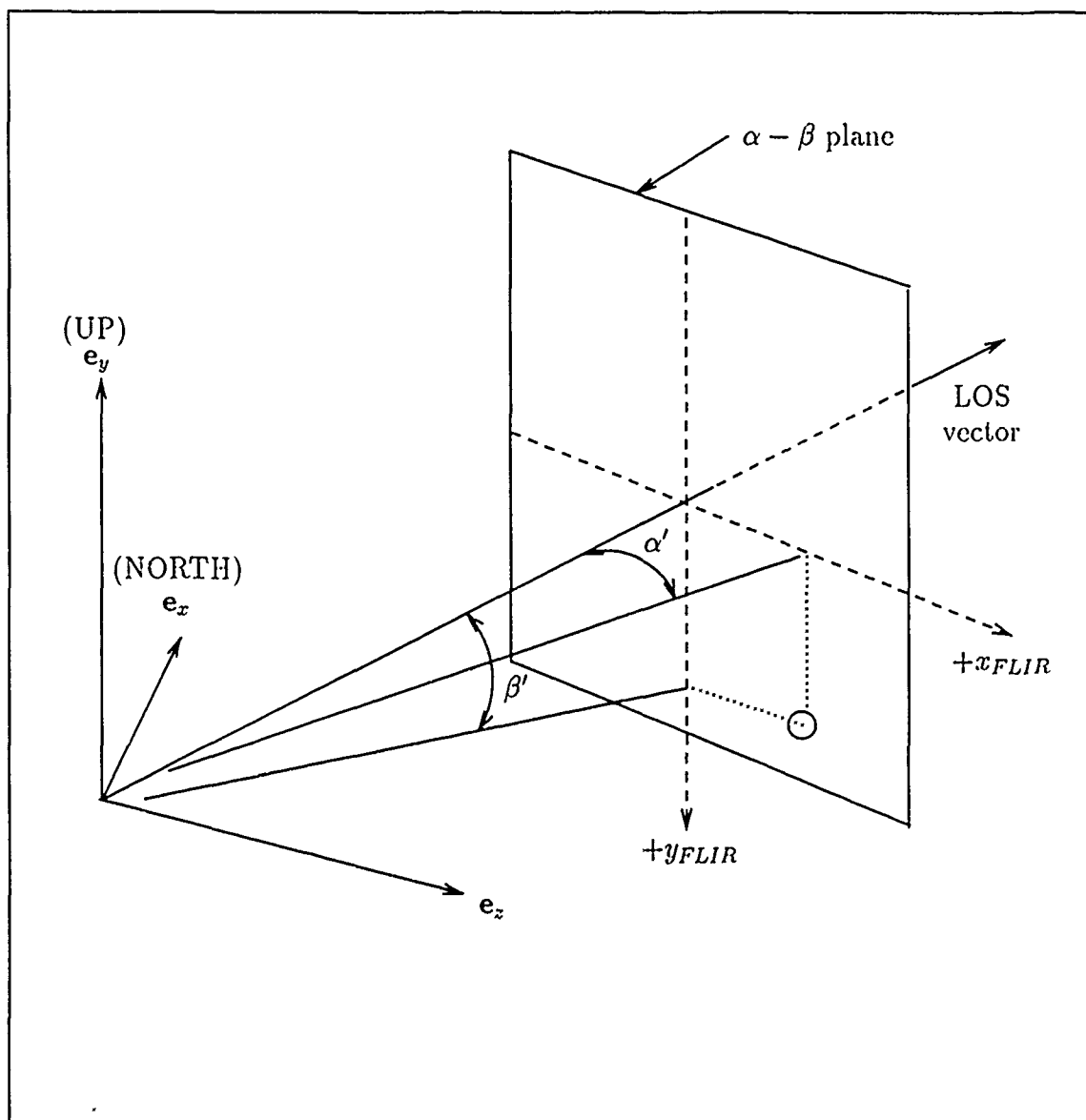


Figure 3.1. The $\alpha - \beta$ Plane

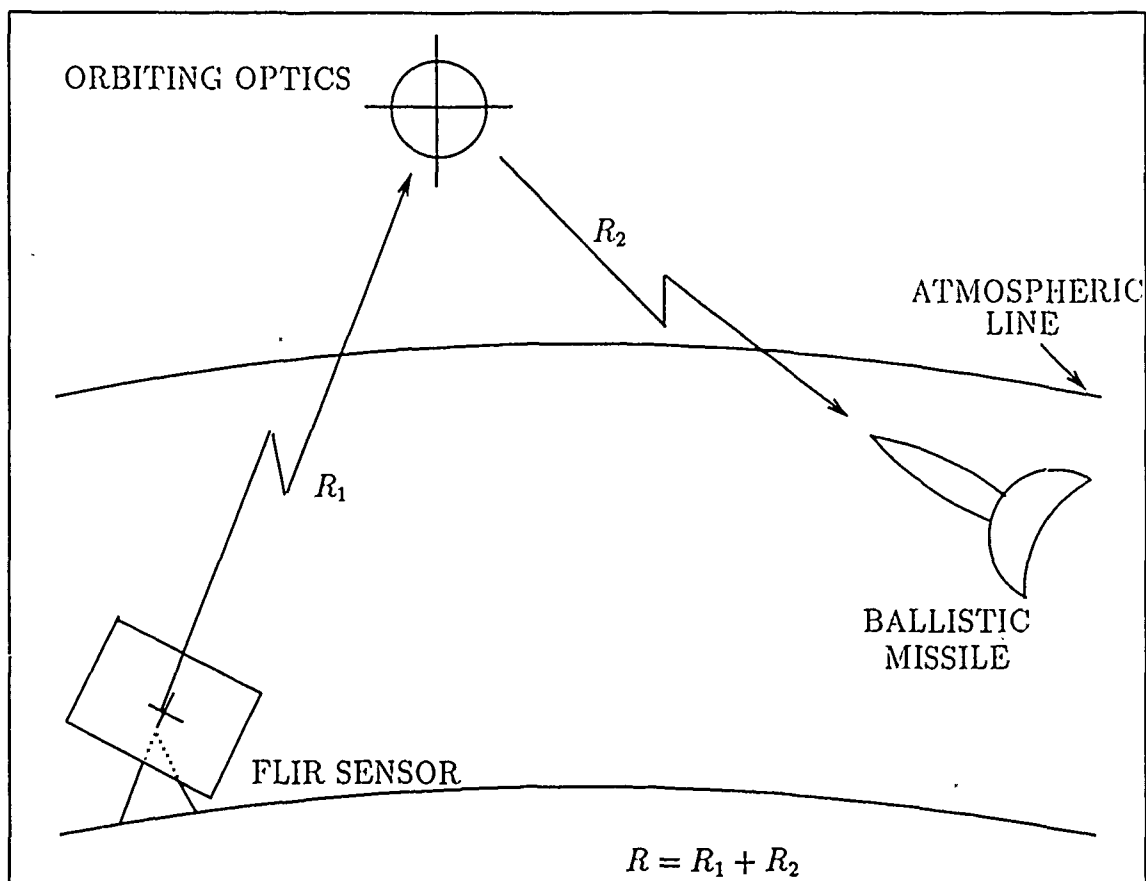


Figure 3.2. Sensor-to-Target Range

then:

$$x_t(t_{i+1}) = x_t(t_i) + \frac{(\dot{\alpha}')(\Delta t)}{k_p} \quad (3.13)$$

$$y_t(t_{i+1}) = y_t(t_i) + \frac{(\dot{\beta}')(\Delta t)}{k_p} \quad (3.14)$$

Arranging the above equations in state space form yields:

$$\begin{bmatrix} x_t(t_{i+1}) \\ y_t(t_{i+1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t(t_i) \\ y_t(t_i) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{k_p} & 0 \\ 0 & \frac{\Delta t}{k_p} \end{bmatrix} \begin{bmatrix} \dot{\alpha}'(t_i) \\ \dot{\beta}'(t_i) \end{bmatrix} \quad (3.15)$$

where:

$$\dot{\alpha}'(t_i) = \frac{d\alpha'}{dt}, \text{ measured in micro-radians/second and constant over the time interval } [t_i, t_{i+1}]$$

$$\dot{\beta}'(t_i) = \frac{d\beta'}{dt}, \text{ measured in micro-radians/second and constant over the time interval } [t_i, t_{i+1}]$$

$$\Delta t = \text{sample time interval, } t_{i+1} - t_i$$

$$k_p = \text{pixel proportionality constant, 15 micro-radians/pixel.}$$

Using these relationships in the block form of the overall truth model, by inspection of Equation (3.10), the upper left hand block is:

$$\Phi_{t_2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.16)$$

and the upper block of Equation (3.11) is:

$$B_{dt_2 \times 2} = \begin{bmatrix} \frac{\Delta t}{k_p} & 0 \\ 0 & -\frac{\Delta t}{k_p} \end{bmatrix} \quad (3.17)$$

and the input vector in Equation (3.5) is given by:

$$u_{Td}(t_i) = \begin{bmatrix} \dot{\alpha}'(t_i) \\ \dot{\beta}'(t_i) \end{bmatrix} \quad (3.18)$$

The minus sign in Equation (3.17) is due to the difference in the y axis orientations between the FLIR plane coordinate frame and the inertial coordinate frame (Figure 3.1). This is inherent to the simulation and provides the correct directions for the truth model and filter model position states.

The truth model missile trajectory used in the simulation is a point mass influenced by a thrust force and a gravitational force, described by the following inverse square-law force field equation in Reference [1]:

$$F_G = \frac{Gm_1m_2}{r^2} \quad (3.19)$$

where:

- F_G = force of attraction between the missile and the Earth
- G = universal gravitational constant
- m_1, m_2 = mass of the Earth and missile, respectively
- r = distance between the Earth's center and the missile center of gravity

For the purposes of this study, all other external forces acting on the missile (atmospheric drag, deterministic solar effects, etc.) are assumed negligible; and the missile is assumed to have constant mass over the simulation interval of ten seconds. To obtain an expression for the missile acceleration, Newton's second law is used:

$$\mathbf{F} = m\mathbf{a} \quad (3.20)$$

where:

- $\mathbf{F}$ = external force(s) acting on a body (missile)
- m = constant mass of the missile
- $\mathbf{a}$ = inertial acceleration of the missile.

From the derived inertial acceleration, the components of the missile's inertial velocity and position are obtained via integration. The deterministic inertial position

and velocity components are used to project the velocity onto the FLIR image plane (see Section 3.4.4), and the resulting FLIR plane position coordinates from the truth model propagation cycle represent the first two states in the truth model state vector of Equation (3.9). Note that the thrust and mass parameters used to describe the simulated ballistic missile are based upon Atlas missile specifications as given by reference [1].

This truth model deterministic trajectory could have been contained in "look-up" tables, where the exact coordinates of the missile's position are stored for every time increment of the simulation. There are two advantages to representing the deterministic truth model in the form of Equation (3.15). First, Equation (3.15) can be substituted back into Equation (3.5) to form a single augmented vector differential equation that defines the truth model. Second, since Equation (3.15) is in state space form, white noise could be added, if desired, to account for non-deterministic type terms such as wind-buffeting or solar effects acting on the missile's hard-body.

3.2.2 The Atmospheric States. Based upon power spectral density characteristics, the atmospheric jitter phenomena can be modelled as the output of a third-order shaping filter driven by white Gaussian noise [24, 25]. With this model, one can identify the effects of the atmospheric disturbance on the FLIR plane image. The Laplace domain representation of this shaping filter is given by [16]:

$$\frac{x_a(s)}{w_a(s)} = \frac{K_a \omega_1 \omega_2^2}{(s + \omega_1)(s + \omega_2)^2} \quad (3.21)$$

where:

- x_a = output of shaping filter, defined in Equation (3.1)
- w_a = zero-mean, scalar, unit-strength white Gaussian noise
- K_a = gain, adjusted for desired atmospheric jitter RMS value
- ω_1 = break frequency, 14.14 rad/sec
- ω_2 = break frequency, 659.5 rad/sec.

The inverse Laplace transform of Equation (3.21) is a third-order, linear differential equation which can be expressed as three, coupled, first-order, linear differential equations in state space format. The atmospheric jitter model in the y_{FLIR} direction can be identically modelled as in the x_{FLIR} direction; therefore the truth model for atmospheric jitter can be expressed in Jordan canonical form as [16] :

$$\dot{\mathbf{x}}_a(t) = \mathbf{F}_a \mathbf{x}_a(t) + \mathbf{G}_a \mathbf{w}_a(t) \quad (3.22)$$

where:

$\mathbf{x}_a(t)$ = 6-dimensional atmospheric state vector

$\mathbf{F}_a$ = 6×6 time-invariant atmospheric plant matrix

$\mathbf{w}_a(t)$ = 2-dimensional, independent, zero-mean, white Gaussian noise process vector with unit strength components and statistics:

$$E[\mathbf{w}_a(t)] = 0$$

$$E[\mathbf{w}_a(t) \mathbf{w}_a^T(t + \tau)] = \mathbf{Q}_a \delta(\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \delta(\tau)$$

$E[\cdot]$ = expected value

where the atmospheric plant matrix is defined as:

$$\mathbf{F}_a = \begin{bmatrix} -\omega_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\omega_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_2 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\omega_2 \end{bmatrix} \quad (3.23)$$

and the noise distribution matrix is:

$$G_a = \begin{bmatrix} \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)^2} & 0 \\ \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)^2} & 0 \\ \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)} & 0 \\ 0 & \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)^2} \\ 0 & \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)^2} \\ 0 & \frac{K_a^2 \omega_1 \omega_2^2}{(\omega_1 - \omega_2)} \end{bmatrix} \quad (3.24)$$

The discrete-time equivalent to Equation (3.22) is the atmospheric jitter partition of the augmented truth model in Equation (3.5) and is given by:

$$x_a(t_{i+1}) = \Phi_a(t_{i+1}, t_i) x_a(t_i) + w_{da}(t_i) \quad (3.25)$$

Hannley and Jensen [3] showed that the state transition matrix in Jordan canonical form for the time-invariant plant matrix F_a of Equation (3.22) is given by:

$$\Phi_a(\Delta t) = \begin{bmatrix} \Phi_{a11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \Phi_{a22} & \Phi_{a23} & 0 & 0 & 0 \\ 0 & 0 & \Phi_{a33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_{a44} & 0 & 0 \\ 0 & 0 & 0 & 0 & \Phi_{a55} & \Phi_{a56} \\ 0 & 0 & 0 & 0 & 0 & \Phi_{a66} \end{bmatrix} \quad (3.26)$$

where:

$$\begin{aligned} \Phi_{a11} &= \Phi_{a44} = \exp(-\omega_1 \Delta t) \\ \Phi_{a22} &= \Phi_{a55} = \exp(-\omega_2 \Delta t) \\ \Phi_{a23} &= \Phi_{a56} = \Delta t \exp(-\omega_2 \Delta t) \\ \Phi_{a33} &= \Phi_{a66} = \exp(-\omega_2 \Delta t) \end{aligned}$$

Furthermore, the six-dimensional, zero-mean, discrete-time, white, Gaussian noise $w_{da}(t_i)$ has characteristics defined by:

$$E[\mathbf{w}_{da}(t_i)] = 0 \quad (3.27)$$

$$E[\mathbf{w}_{da}(t_i)\mathbf{w}_{da}^T(t_i)] = \mathbf{Q}_{da} = \int_{t_i}^{t_{i+1}} \Phi_a(t_{i+1}, \tau) \mathbf{G}_a \mathbf{Q}_a \mathbf{G}_a^T \Phi_a^T(t_{i+1}, \tau) d\tau. \quad (3.28)$$

3.2.3 The Bending/Vibration States. Mechanical bending states were recently added to the truth model in a study conducted by Leeney [5]. The bending model is included to represent vibrational phenomena that exist in the FLIR data when a non-rigid optical platform is involved in collecting the IR image data of the plume (see Figure 3.2). Based on tests conducted for the Air Force Weapons Laboratory, Leeney concluded that the bending phenomenon in both the x and y FLIR directions can be represented by a second-order shaping filter driven by white, Gaussian noise. The Laplace domain transfer function is described as [5]:

$$\frac{x_b(s)}{w_b(s)} = \frac{K_b \omega_{nb}^2}{s^2 + 2\zeta_b \omega_{nb} s + \omega_{nb}^2} \quad (3.29)$$

where:

- x_b = mechanical bending disturbance state shaping filter output for the x direction, similar for the y direction
- w_b = zero-mean, unit strength, white Gaussian noise with an autocorrelation of:
 $E[w_b(t)w_b(t + \tau)] = Q_b \delta(t - \tau); \quad Q_b = 1$
- K_b = gain adjustment to obtain desired root mean squared (RMS) bending output; $K_b^2 = 5 \times 10^{-13}$
- ζ_b = damping coefficient equal to 0.15
- ω_{nb} = undamped natural frequency for bending; $\omega_{nb} = \pi$ rad/sec.

Leeney [5] represented the bending states by a second-order shaping filter, rather than a higher order model. Since this model captures the fundamental fre-

quency of this effect, a second-order model represented a good initial study of filter performance for the bending phenomenon. The linear stochastic differential equation that describes the bending/vibration is:

$$\dot{\mathbf{x}}_b(t) = \mathbf{F}_b \mathbf{x}_b(t) + \mathbf{G}_b \mathbf{w}_b(t) \quad (3.30)$$

where:

- $\mathbf{x}_b$ = 4-dimensional mechanical bending state vector
- $\mathbf{F}_b$ = 4×4 time invariant bending plant matrix
- $\mathbf{w}_b(t)$ = 2-dimensional, zero-mean, white Gaussian noise process with independent components of strength $\mathbf{Q}_b = \mathbf{I}$
- $\mathbf{G}_b$ = 4×2 noise distribution matrix.

The equivalent discrete-time model for Equation (3.30) is of the form:

$$\mathbf{x}_b(t_{i+1}) = \Phi_b(t_{i+1}, t_i) \mathbf{x}_b(t_i) + \mathbf{w}_{db}(t_i) \quad (3.31)$$

where:

$$\Phi_b = \begin{bmatrix} \Phi_{b1} & \Phi_{b2} & 0 & 0 \\ \Phi_{b3} & \Phi_{b4} & 0 & 0 \\ 0 & 0 & \Phi_{b1} & \Phi_{b2} \\ 0 & 0 & \Phi_{b3} & \Phi_{b4} \end{bmatrix} \quad (3.32)$$

and

$$\begin{aligned} \Phi_{b1} &= \exp(-\sigma_b \Delta t) [\cos(\omega_b \Delta t) + \frac{\sigma_b}{\omega_b} \sin(\omega_b \Delta t)] \\ \Phi_{b2} &= \exp(-\sigma_b \Delta t) [\frac{1}{\sigma_b} \sin(\omega_b \Delta t)] \\ \Phi_{b3} &= \exp(-\sigma_b \Delta t) [-1 - (\frac{\sigma_b}{\omega_b})^2] \sin(\omega_b \Delta t) \\ \Phi_{b4} &= \exp(-\sigma_b \Delta t) [\cos(\omega_b \Delta t) - \frac{\sigma_b}{\omega_b} \sin(\omega_b \Delta t)] \end{aligned}$$

- Δt = sample time interval $[t_i, t_{i+1}]$
 σ_b = real part of the root of the characteristic equation in Equation (3.29): $\sigma_b = -0.47124$ rad/sec
 ω_b = imaginary part of the root of the characteristic equation in Equation (3.29): $\omega_b = 3.10605$ rad/sec.

The 4-dimensional, zero-mean, discrete-time, white Gaussian noise process of Equation (3.31) has a 4×4 equivalent, discrete-time covariance matrix defined by:

$$Q_{db} = \int_{t_i}^{t_{i+1}} \Phi_b(t_{i+1}, \tau) G_b Q_b G_b^T \Phi_b^T(t_{i+1}, \tau) d\tau. \quad (3.33)$$

3.2.4 Plume Pogo States. One of the main objectives of this thesis effort is to model the plume pogo phenomenon in the truth model. To avoid possible classification of this study, the assumed model used for plume pogo is a basic second-order Gauss-Markov process [10]. This model was chosen to study the amplitude and frequency characteristics of the oscillatory nature of the plume. It should be noted that unclassified experimental data was unavailable to characterize the pogo phenomenon, i.e. power spectral density plots. However, based upon physical insight and visual observation of the pogo effect, a second-order shaping filter driven by white Gaussian noise was designed as follows [10]:

$$\frac{x_p(s)}{w_p(s)} = \frac{K_p \omega_{np}^2}{s^2 + 2\zeta_p \omega_{np} s + \omega_{np}^2} \quad (3.34)$$

where:

- x_p = plume pogo state shaping filter output along the direction of the velocity vector
- w_p = zero-mean, unit strength, white Gaussian noise with an autocorrelation of:

$$E[w_p(t)w_p(t + \tau)] = Q_p\delta(t - \tau); \quad Q_p = 1$$
- K_p = gain adjustment to obtain desired root mean squared (RMS) pogo amplitude (see Appendix A)
- ζ_p = assumed damping coefficient chosen as 0.05
- ω_{np} = nominal undamped natural frequency for pogo; assumed range is 0.1-10 Hertz, with a nominal value of 1.0 Hertz.

Oscillations due to this effect are modelled along the direction of the ballistic missile velocity vector. The mathematical expression used to describe the pogo effect takes the form of a two-state linear stochastic differential equation described by:

$$\dot{\mathbf{x}}_p(t) = \begin{bmatrix} 0 & 1 \\ -\omega_{np}^2 & -2\zeta_p\omega_{np} \end{bmatrix} \mathbf{x}_p(t) + \begin{bmatrix} 0 \\ K_p\omega_{np}^2 \end{bmatrix} w_p(t) \quad (3.35)$$

where:

- $\mathbf{x}_p(t)$ = 2-dimensional pogo state vector derived from Equation (3.34)
- $w_p(t)$ = 1-dimensional zero-mean, white Gaussian noise of unity strength from Equation (3.34).

To simulate the pogo model on a digital computer, the following equivalent discrete-time model for Equation (3.35) is used:

$$\mathbf{x}_p(t_{i+1}) = \begin{bmatrix} \Phi_{p11}(\Delta t) & \Phi_{p12}(\Delta t) \\ \Phi_{p21}(\Delta t) & \Phi_{p22}(\Delta t) \end{bmatrix} \mathbf{x}_p(t_i) + \mathbf{w}_{dp}(t_i) \quad (3.36)$$

where:

$$\begin{aligned}
\Phi_{p11}(\Delta t) &= \frac{1}{\sqrt{1-\zeta_p^2}} \exp(-\zeta_p \omega_{np} \Delta t) \sin(\omega_{np} \sqrt{1-\zeta_p^2} \Delta t + \arctan[\frac{\sqrt{1-\zeta_p^2}}{\zeta_p}]) \\
\Phi_{p12}(\Delta t) &= \frac{1}{\omega_{np} \sqrt{1-\zeta_p^2}} \exp(-\zeta_p \omega_{np} \Delta t) \sin(\omega_{np} \sqrt{1-\zeta_p^2} \Delta t) \\
\Phi_{p21}(\Delta t) &= \frac{-\omega_{np}}{\sqrt{1-\zeta_p^2}} \exp(-\zeta_p \omega_{np} \Delta t) \sin(\omega_{np} \sqrt{1-\zeta_p^2} \Delta t) \\
\Phi_{p22}(\Delta t) &= \frac{1}{\sqrt{1-\zeta_p^2}} \exp(-\zeta_p \omega_{np} \Delta t) \sin(\omega_{np} \sqrt{1-\zeta_p^2} \Delta t + \arctan[\frac{\sqrt{1-\zeta_p^2}}{-\zeta_p}] + \pi) \\
\Delta t &= \text{sampling interval } [t_{i+1} - t_i] \\
\mathbf{w}_{dp}(t_i) &= \text{2-dimensional, zero-mean, discrete-time, white Gaussian noise} \\
&\quad \text{with independent components and } 2 \times 2 \text{ covariance matrix:}
\end{aligned}$$

$$\mathbf{Q}_{dp} = \int_{t_i}^{t_{i+1}} \Phi_p(t_{i+1}, \tau) \mathbf{G}_p \mathbf{Q}_p \mathbf{G}_p^T \Phi_p^T(t_{i+1}, \tau) d\tau. \quad (3.37)$$

As stated above, $\mathbf{x}_p$ is a 2-dimensional pogo state vector that represents: (1) the position of the plume image intensity centroid along the longitudinal axis of the missile and (2) the plume's velocity along the same axis. The plume "pogos" or oscillates about an equilibrium point also located on the longitudinal axis. The location of this equilibrium point is defined by the initial positions of the two intensity functions in the target coordinate frame (see Section 3.4.1), and remains equidistant from the hardbody's center of mass throughout the simulation.

Figure 3.3 shows the location of the equilibrium point relative to the plume's centroid for a positive and negative pogo. It should be noted that, for this simulation, the velocity vector is assumed to lie coincident with the longitudinal axis of the missile: the angle of attack and side-slip angle are also assumed to be zero throughout the entire simulation.

3.3 The Measurement Model

Target information is obtained by measuring the intensity and location of the target's infrared image on the pixel array of infrared sensitive detectors. This image or "intensity function" is the collective sum of target plume IR radiation, background noise, and sensor noise.

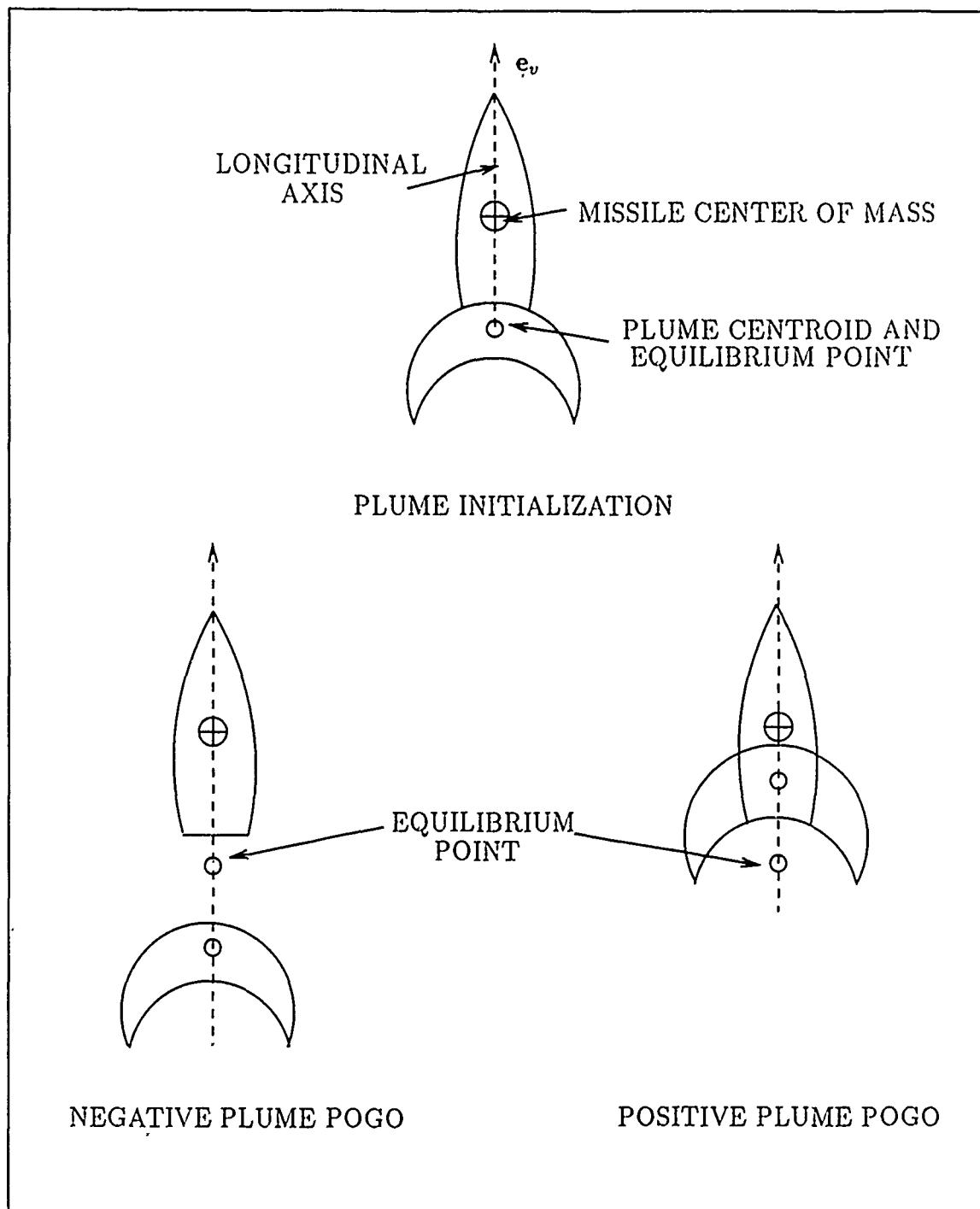


Figure 3.3. Plume Pogo Relative to the Equilibrium Point (Note that the crescent-shaped plume is just one of many equal-intensity contour lines of the actual plume)

Consider the radiated energy from a single intensity function target. The infrared intensity function can be modelled as a bivariate Gaussian distribution with elliptical constant intensity contours. This bivariate Gaussian intensity function is given by the following equation [9, 24]:

$$I[x, y, x_{peak}(t), y_{peak}(t)] = I_{max} \exp\{-0.5[\Delta x \Delta y] \mathbf{P}^{-1} [\Delta x \Delta y]^T\} \quad (3.38)$$

where:

- Δx = $(x - x_{peak}) \cos \theta_T + (y - y_{peak}) \sin \theta_T$
- Δy = $(y - y_{peak}) \cos \theta_T - (x - x_{peak}) \sin \theta_T$
- θ_T = target orientation angle between the projection of the velocity vector perpendicular to the LOS vector and the x axis in the FLIR image plane
- x, y = reference coordinate axes on the $\alpha - \beta$ plane
- x_{peak}, y_{peak} = coordinates of the peak intensity of the single Gaussian intensity function
- I_{max} = maximum intensity of the function
- $\mathbf{P}$ = 2×2 target dispersion matrix whose eigenvalues (σ_v^2 and σ_{pv}^2) define the dispersion of the elliptical constant intensity contours (along the velocity vector and perpendicular to that velocity vector, respectively) in the $\alpha - \beta$ plane (see Sections 3.4.1 and 3.4.5).

The composite FLIR plane image intensity function, for the difference of two individual intensity functions representing a ballistic missile target plume, is shown in Figure (3.4). To form the characteristic shape of a missile plume, the rear individual Gaussian intensity function is subtracted from the forward Gaussian intensity function, and the resulting intensity function is obviously not Gaussian. Since the intensity value on a pixel sensor can not be negative, the simulation software hardcodes any calculated negative intensity values to zero.

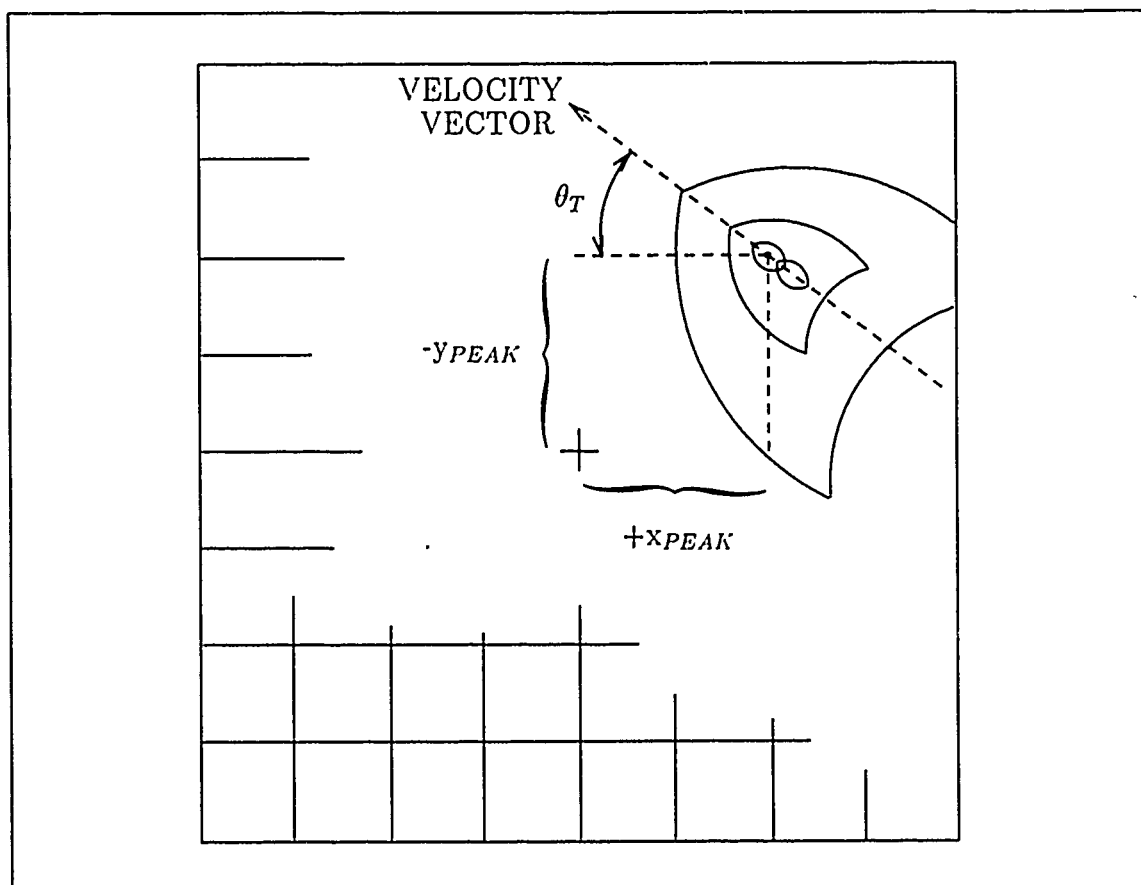


Figure 3.4. FLIR Image Plane Intensity Function for the Difference Between Two Gaussian Intensity Functions (Note the sign convention on the peaks which corresponds appropriately to Figure 3.1)

The intensity measurement produced by each pixel is the average intensity on that pixel that results from the sum of the target's intensity function, spatially correlated background noise, and FLIR sensor noise. The output of the pixel in the j^{th} row and k^{th} column at sample time t_i , is given by:

$$z_{jk}(t_i) = \frac{1}{A_p} \int_{pixel_{jk}} \{ I_1[x, y, x_{peak_1}(t_i), y_{peak_1}(t_i)] - I_2[x, y, x_{peak_2}(t_i), y_{peak_2}(t_i)] dx dy \} + n_{jk}(t_i) + b_{jk}(t_i) \quad (3.39)$$

where:

- $z_{jk}(t_i)$ = output of pixel jk
- A_p = area of one pixel
- I_1, I_2 = intensity function of first and second Gaussian intensity function, respectively
- x, y = coordinates of any point within pixel jk
- x_{peak_1}, y_{peak_1} = coordinates of maximum intensity of the first Gaussian intensity function
- x_{peak_2}, y_{peak_2} = coordinates of maximum intensity of the second Gaussian intensity function
- $n_{jk}(t_i)$ = effect of internal FLIR sensor noise on pixel jk
- $b_{jk}(t_i)$ = effect of spatially correlated background noise on pixel jk .

The sensor error, $n_{jk}(t_i)$, is the result of thermal noise and dark current in the infrared sensitive detectors. This sensor error is assumed to be a corruptive noise which is both temporally and spatially uncorrelated [9].

The background noise, $b_{jk}(t_i)$, is represented as a spatially correlated noise with a radially symmetric, exponentially decaying correlation pattern characterized by a correlation distance of approximately two pixels in the FLIR image plane [3, 24]. Harnly and Jensen [3] simulated this effect by maintaining non-zero correlation coefficients between each pixel and its two nearest neighbors in all directions.

By concatenating all 64 values of b_{jk} (corresponding to an 8×8 -matrix of pixels) into a 64-dimensional vector $\mathbf{b}(t_i)$, the spatially correlated background noise is modelled as:

$$\mathbf{b}(t_i) = \sqrt[4]{\mathbf{R}} \mathbf{b}'(t_i) \quad (3.40)$$

where:

- $\mathbf{R}$ = 64×64 correlation matrix of the discrete, zero-mean, white Gaussian vector noise process $\mathbf{b}(t_i)$
- $\mathbf{b}'(t_i)$ = 64-dimensional, discrete, zero-mean, white Gaussian vector noise process with the correlation matrix: $\mathbf{I}_{(64 \times 64)}$
- $\sqrt[4]{}$ = Cholesky square root.

A detailed development of this spatially correlated noise process and the FLIR sensor noise process can be found in the work of Maybeck, Harnly and Jensen [3, 11]. It is only mentioned briefly in this section for completeness in describing the truth model.

3.4 Simulation Space

To simulate the FLIR tracker's operation on a digital computer, a "simulation space" model is required. As presented by Tobin [24], this simulation space was designed to perform two tasks. First, it generates the propagation of a realistic target trajectory in three dimensional space. Second, the simulation space provides a mathematical means of projecting the target's infrared image and velocity vector onto the FLIR image plane. Each of these tasks is discussed in detail in this section; but first, the pertinent coordinate frames will be presented.

3.4.1 Coordinate Frames. The following coordinate frames are used during the simulation of the FLIR tracker on a digital computer:

Target Frame:

Origin: center of mass of the target

Axes: e_v - along the velocity vector
 e_{pv} - out right side of the target, perpendicular to e_v
 e_{ppv} - vector completing right-hand coordinate set

Note: 'v' - along the velocity vector
'pv' - perpendicular to the velocity vector
'ppv' - perpendicular to both of the above.

Inertial Frame:

Origin: location of the FLIR sensor

Axes: e_x - due north, tangent to earth's surface, defines zero azimuth
 e_y - inertial "up" with respect to a flat earth approximation
 e_z - vector completing right-hand coordinate set, defines 90° azimuth

Note: The azimuth angle (α) is measured eastward from e_x .

The elevation angle (β) is measured "up" from the horizontal plane defined by e_x and e_z .

$\alpha - \beta - r$ Frame:

Origin: center of mass of the target

Axes: e_r - coincident with the true sensor-to-target LOS vector.
 e_α, e_β - define a plane perpendicular to e_r , rotated from the inertial e_x and e_y by the azimuth angle (α) and the elevation angle (β), respectively.

$\alpha - \beta$ (FLIR Image) Plane:

This is the FLIR image plane defined by the e_α and the e_β unit vectors above. The "pseudo" azimuth and elevation angles, α' and β' , measured with respect to the FLIR LOS vector, are linearly proportional to the cartesian coordinates x and y on the FLIR plane. The coordinates

x and y are distances from the center of the FLIR FOV, measured in pixels on the $\alpha - \beta$ plane. Observing the FLIR plane along the LOS vector from the origin of the inertial axis, x is positive to the right and y is positive down. This choice is made to maintain a right-handed coordinate system, with the target's range measured positive away from the sensor. It should be noted that this is also the perspective of the greyscale plots to be discussed in Section 5.8.

The inertial frame and the $\alpha - \beta$ plane are illustrated in Figure 3.1. The target frame is shown in Figure 3.5.

3.4.2 Target Model. The basic target used for this thesis is a planform with two intensity functions. Note the displacement of the two Gaussian intensity function centroids along the e_v direction. These values were chosen based upon the assumption that, in the target frame, the dispersion of the plume in the e_{pv} direction is approximately 20 times the radius of the missile. The centroid of the first intensity function is placed at -65 meters from the center of mass of the missile in order to simulate the composite centroid of the plume being close to the exhaust nozzle of the missile. This is based on an assumption that the distance from the missile's center of mass to the end of the hardbody is on the order of 20 meters. The second intensity function is arbitrarily set at -110 meters to simulate one of many different characteristic plume crescent shapes. By varying the location of the second intensity function centroid, the shape of the plume can be varied, as well as the relative distance of the composite image centroid to the end of the missile hardbody. The centroids of these intensity functions remain fixed in the target frame (if pogo oscillations did not exist; see Section 3.4.5) and are shown in Figure 3.5. As mentioned earlier, the target's angle of attack and sideslip angle are assumed to be zero. These assumptions imply that the semi-major axes of the infrared intensity function ellipses are aligned with the target's velocity vector (see Figure 3.6). As noted by Netzer [18], this simplifies the simulation space geometry without degrading the performance analysis of the tracker.

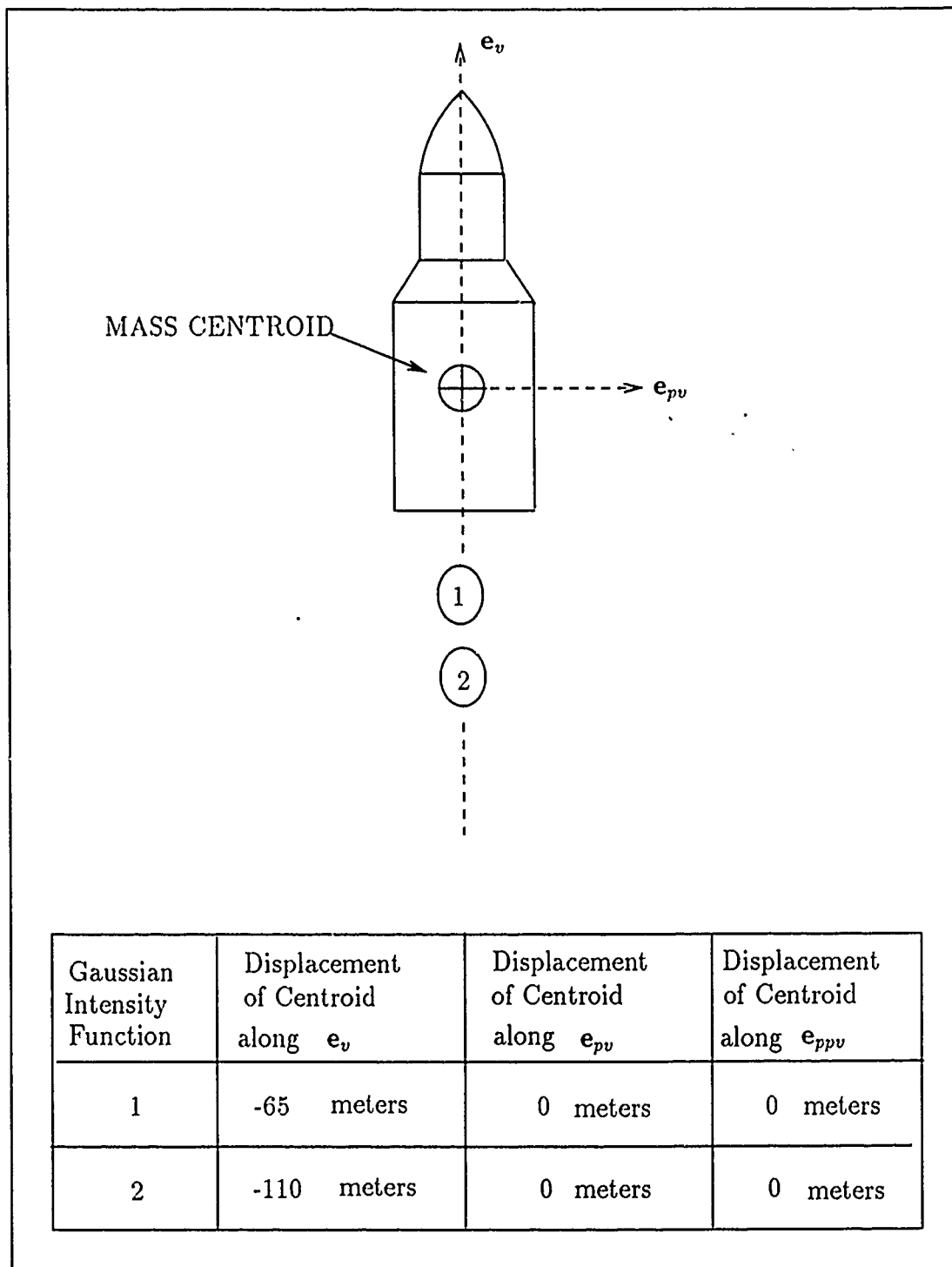


Figure 3.5. Distribution of Gaussian Intensity Functions

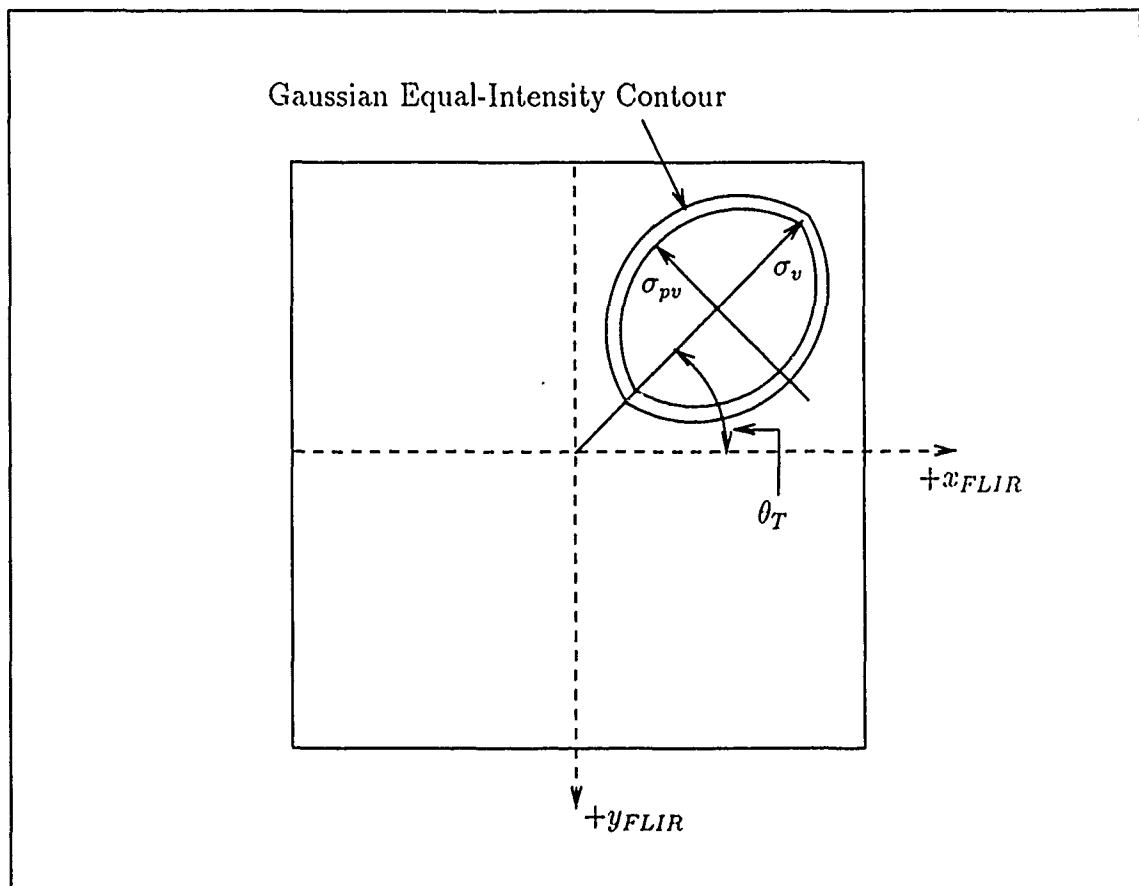


Figure 3.6. Hotspot Ellipsoid of Dispersion on the FLIR Image Plane

3.4.3 Target Scenarios for MMAF Analysis. As mentioned in Section 3.2.1, the trajectory used in this study was a typical thrusting trajectory of an Atlas missile. The initial conditions of the missile in inertial space define the missile orientation in inertial space, as well as on the FLIR image plane. This thesis deals with only one such trajectory, where θ_T , the target orientation angle, is initialized at approximately 60° , and is permitted to decrease due to gravitational effects over the ten second simulation. The reason for working with only one target orientation is that the focus of this research is to characterize the plume pogo effect; therefore various target scenarios are generated to study the varying effects of plume pogo dynamics, particularly in a MMAF application.

It should be noted that, although the pogo scenarios are defined in this section, they were never implemented for reasons to be described in Chapter VI. The scenarios are presented here for completeness, so that a continuing, follow-on study can be performed to research the applicability and adaptability of multiple model adaptive filtering to missile plume pogo effects (see Chapter VII).

Seven truth model scenarios are defined to study the MMAF application to the plume phenomenon. The scenarios involve varying the amplitude and frequency characteristics of the pogo effect in the truth model in an effort to study the tracking ability of the MMAF. As mentioned in Section 3.2.4, much of the information on plume pogo is classified; therefore, nominal ranges of pogo amplitude and pogo frequency were chosen using physical insights and sound engineering judgment. The maximum value (upper bound) for the amplitude characteristic of the pogo effect is chosen at 1.12 pixels and represents the maximum desired RMS pogo value used to adjust the gain in Equation (3.34). This value was chosen because it represents a pogo effect of approximately 20 meters in the target frame, which corresponds to approximately 60% occlusion of an Atlas missile's hardbody by the plume [1]. Based upon this upper bound, the range for pogo amplitude is chosen as 1.12 to 0.0112 pixels on the FLIR image plane, where 0.112 pixels is assumed to be the

Scenario	Amplitude(pixels)	Frequency(Hz)
#1	0.0112	0.1
#2	1.12	10
#3	0.0112	10
#4	1.12	0.1
#5	0.0 $\rightarrow$ 0.0112	0.0 $\rightarrow$ 0.1
#6	0.0112 $\rightarrow$ 1.12	0.1 $\rightarrow$ 10
#7	0.0112 $\rightarrow$ 1.12	10 $\rightarrow$ 0.1

Table 3.1. Truth Model Scenarios

nominal value. The nominal undamped natural frequency for the pogo oscillation is assumed to be about one Hertz; therefore, a range of ten Hertz to 0.1 Hertz was chosen to analyze the MMAF performance. Table 3.1 provides the details of the seven scenarios.

Scenarios #1 through #4 provide MMAF performance statistics for a truth model pogo phenomenon that is fixed at a specific amplitude and frequency throughout the entire simulation. The MMAF results are then directly comparable to a 10-state, single benchmark filter's (to be discussed in Chapter IV) performance for the same scenario. Scenarios #5 through #7 provide MMAF performance statistics for a pogo phenomenon that will vary in amplitude and frequency over the ten second simulation. These scenarios test the adaptability of the MMAF by identifying which of the single filters in the bank has the highest probability weighting at various times during the scenario (see Equation (2.3)). As mentioned above, although these scenarios were not performed in this research effort, they are mentioned here for completeness for possible follow-on work. Also, a suggested five-bank MMAF structure is provided in Chapter V.

3.4.4 Velocity Projection onto the FLIR Plane. The deterministic input vector, $\mathbf{u}_{Td}(t_i) = [\dot{\alpha}'(t_i) \ \dot{\beta}'(t_i)]^T$ in Equation (3.5), is the projection of the target's inertial velocity vector onto the FLIR image plane. Loving [6] demonstrated that this projection is based on the geometry shown in Figure 3.7.

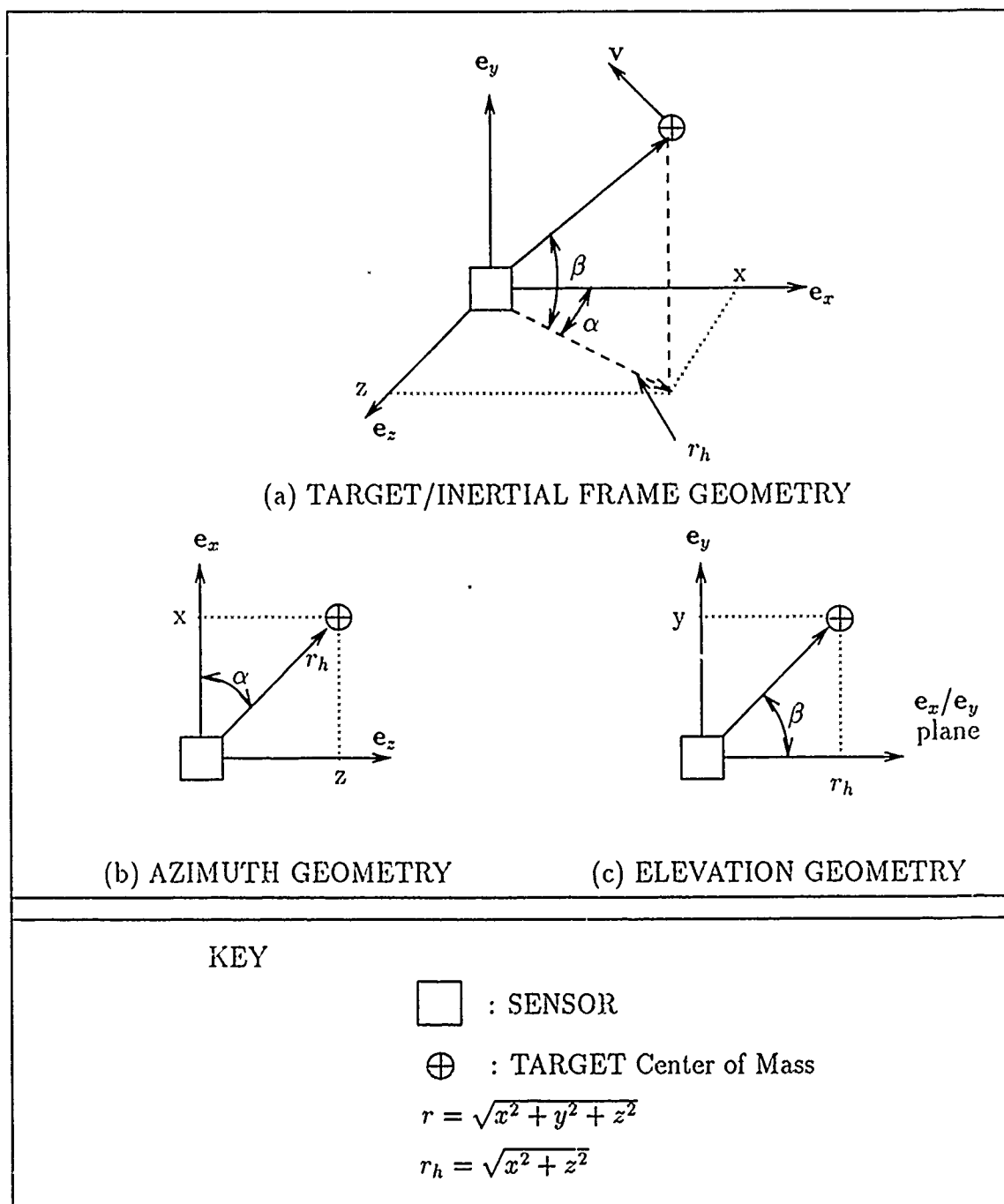


Figure 3.7. Geometry Required to Project the Target's Inertial Velocity onto the FLIR Plane

From Figure 3.7(b), it can be seen that:

$$\alpha(t) = \arctan \left[\frac{z(t)}{x(t)} \right]. \quad (3.41)$$

Taking the time derivative of this equation and realizing that $\dot{\alpha}(t) = \dot{\alpha}'(t)$ yields [17]:

$$\dot{\alpha}'(t) = \dot{\alpha}(t) = \frac{x(t)v_z(t) - z(t)v_x(t)}{x^2(t) + z^2(t)} \quad (3.42)$$

where:

v_x, v_z = components of the target's inertial velocity in the e_x and e_z directions, respectively.

In a similar development from Figure 3.7(c):

$$\beta(t) = \arctan \left[\frac{y(t)}{r_h(t)} \right] \quad (3.43)$$

$$\dot{\beta}'(t) = \dot{\beta}(t) = \frac{r_h(t)v_y(t) - y(t)\dot{r}_h(t)}{r_h^2(t)} \quad (3.44)$$

where:

$$\dot{r}_h(t) = \frac{x(t)v_x(t) + z(t)v_z(t)}{r_h(t)}$$

v_y = component of the target's inertial velocity in the e_y direction.

Equations (3.42) and (3.44) define the deterministic input vector $u_{Td}(t_i)$ in the truth model dynamics difference equation, Equation (3.5).

3.4.5 Target Image Projection onto the FLIR Plane. During the simulation, the target propagates through three-dimensional space; and the output of the infrared sensitive pixels is simulated by projecting the target's two individual intensity functions onto the FLIR image plane. In previous research, the individual intensity functions (hotspots) remained fixed with respect to the target frame, while the orientation and location of the intensity functions on the FLIR plane change as the target translates and changes angular orientation relative to the sensor. In this research,

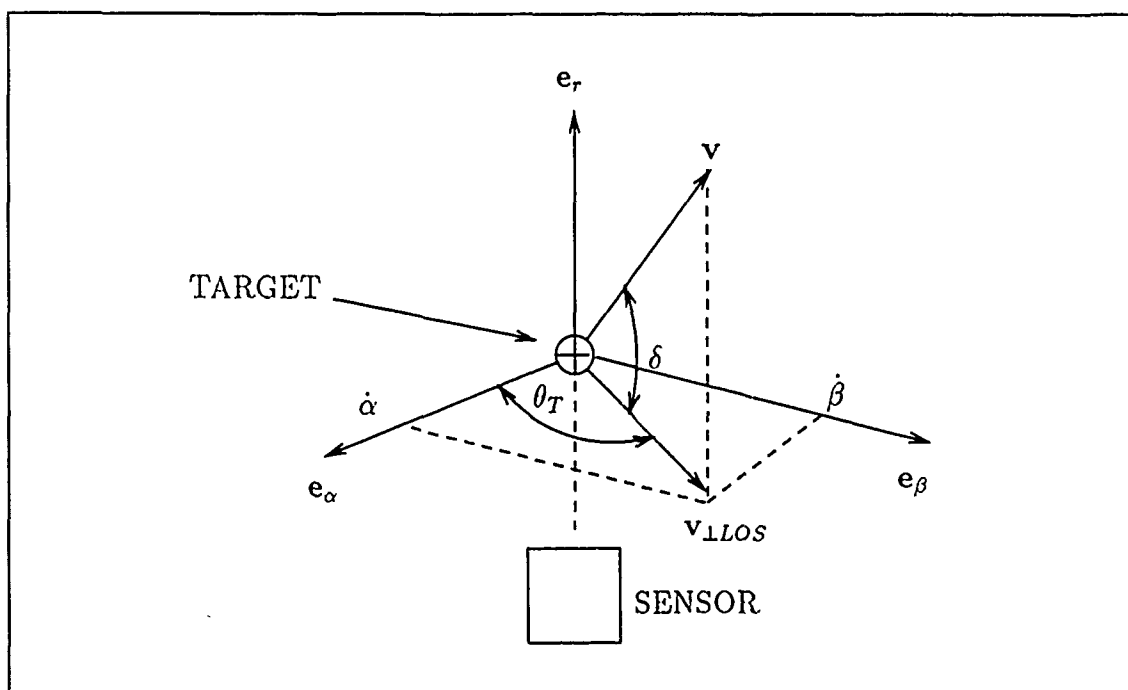


Figure 3.8. Infrared Image Projection Geometry

since the pogo phenomenon is causing the composite image centroid to oscillate along the velocity vector, the individual intensity functions do not remain fixed in the target frame; and correspondingly, this pogo phenomenon produces an additional perturbation to the intensity function projections onto the FLIR plane. For simplicity, the location of each of the individual intensity functions is initialized in the target frame as displacements from the missile's center of mass. To orient them properly in the FLIR coordinate frame, they are rotated by the target orientation angle, θ_T (see Figure 3.6).

Similar to the development in [6, 24], consider the geometry presented in Figure 3.8. This geometry relates the current target image to an initial "reference target" image on the FLIR plane, as seen in Figure 3.5. The reference image is oriented to correspond to the largest apparent planform at a specified range. The current image is defined by [24]:

$$\sigma_{pv} = \sigma_{pvo} \left(\frac{r_o}{r} \right) \quad (3.45)$$

$$\begin{aligned}
\sigma_v &= \left(\frac{r_o}{r} \right) (\sigma_{pvo} + (\sigma_{vo} - \sigma_{pvo}) \cos \delta) \\
&= \sigma_{pv} \left\{ 1 + \frac{v_{\perp LOS}}{v} (AR - 1) \right\}
\end{aligned} \tag{3.46}$$

where:

- $\sigma_{vo}, \sigma_{pvo}$ = the initial dispersions of the target intensity functions along $\mathbf{e}_v$ and $\mathbf{e}_{pv}$ in the target frame of the reference image
- σ_v, σ_{pv} = the current dispersions of the target's image
- r_o = initial sensor-to-target range of the reference image
- r = current sensor-to-target range
- $\mathbf{v}$ = inertial velocity vector of the target
- v = magnitude of $\mathbf{v}$
- $\mathbf{v}_{\perp LOS}$ = projection of $\mathbf{v}$ on the $(\alpha - \beta)$ plane; i.e., the component of $\mathbf{v}$ perpendicular to the LOS vector
- $v_{\perp LOS}$ = magnitude of $\mathbf{v}_{\perp LOS}$: $v_{\perp LOS} = \sqrt{\dot{\alpha}^2 + \dot{\beta}^2}$
- δ = angle between $\mathbf{v}$ and the $(\alpha - \beta)$ plane
- AR = $\frac{\sigma_{vo}}{\sigma_{pvo}}$: aspect ratio of the reference image

Together, Equations (3.45) and (3.46) define the dispersion along the principle axes of the intensity functions' images as seen by the sensor (Figure 3.6).

3.5 Summary

This chapter shows the truth model dynamic system to be the augmentation of a deterministic target trajectory component, a stochastic atmospheric component, a stochastic bending/vibration component, and a stochastic plume pogo component. In the measurement model, the two individual intensity profiles that are differenced are assumed to be described by bivariate Gaussian distributions. To simulate the tracker operation on a digital computer, a "simulation space" has been defined to propagate the target trajectory and to make the required transformations between coordinate frames.

IV. Filter Models

4.1 Introduction

This chapter presents the filter model structure used in this thesis research. The models presented here are also used to define the elemental Kalman filters on which the MMAF is based (see Chapter II). Section 4.2 defines the ten-state filter dynamics model which consists of the target dynamics, atmospheric jitter, and plume pogo stochastic processes. The filter measurement model presented in Section 4.3 describes the enhanced correlator/linear measurement model proposed by Rogers [21].

4.2 Dynamics Models

Previous AFIT research has considered two different methods for representing target dynamics in the Kalman filter update equations. The first method describes the target's acceleration as a zero-mean, first-order Gauss-Markov process; while the second method models the acceleration as a series of constant turn-rate trajectories [15]. The ten-state Kalman filter vector used in this research is defined below as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ v_x \\ v_y \\ a_x \\ a_y \\ x_a \\ y_a \\ x_p \\ v_p \end{bmatrix} \quad (4.1)$$

where the target dynamics, jitter, and pogo states used in this research are modelled as Gauss-Markov processes with:

- x_t = x component of target position
- y_t = y component of target position
- v_x = x component of target velocity
- v_y = y component of target velocity
- a_x = x component of target acceleration
- a_y = y component of target acceleration
- x_a = x component of atmospheric jitter
- y_a = y component of atmospheric jitter
- x_p = pogo position along the target velocity vector
- v_p = pogo velocity along the target velocity vector

Each element in Equation (4.1) is coordinatized in FLIR ($\alpha - \beta$) plane. Note that x_t, y_t, x_a, y_a , and x_p were previously defined in Equations (3.1) and (3.2). Also note that the atmospheric jitter model is reduced from six states as defined in the truth model to two states described here in the filter. The filter jitter model disregards the high-frequency effect of the double pole in Equation (3.21). This reduces the order of the filter model, while still capturing the dominant characteristic of atmospheric jitter. The pogo effect is modelled in the filter identically to the way it is modelled in the truth model. This was done to enhance the performance of the filter as well as making the model more applicable for MMAF implementation by allowing more accurate filter tuning for varying pogo characteristics. Eventually, a lower order model will be implemented in the filter, but for the purpose of characterizing the pogo effect and its applicability for MMAF implementation, the two models will be identical. The filter model is described by the following time-invariant, linear stochastic differential equation:

$$\dot{\mathbf{x}}_f(t) = \mathbf{F}_f \mathbf{x}_f(t) + \mathbf{G}_f \mathbf{w}_f(t) \quad (4.2)$$

where:

$\mathbf{x}_f(t)$ = ten-state filter state vector

$\mathbf{F}_f$ = time-invariant system matrix

$\mathbf{G}_f$ = time-invariant noise distribution matrix

$\mathbf{w}_f(t)$ = zero-mean, white Gaussian noise vector of strength $\mathbf{Q}_f$.

Based upon work done by Millner [17] and Kozemchak [4], the elements of Equation (4.2) are:

$$\mathbf{F}_f = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_a} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\omega_{npf}^2 & -2\zeta_{pf}\omega_{npf} \end{bmatrix} \quad (4.3)$$

$$G_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & G_p \end{bmatrix} \quad (4.4)$$

where $G_p = K_{pf}\omega_{npf}^2$

$$Q_f = \begin{bmatrix} \frac{2\sigma_x^2}{\tau_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2\sigma_y^2}{\tau_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2\sigma_a^2}{\tau_a} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2\sigma_a^2}{\tau_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

where:

K_{pf} = gain adjustment to obtain desired RMS pogo amplitude
(see Appendix A)

ω_{npf} = undamped natural frequency for filter pogo

ζ_{pf} = filter damping coefficient for pogo chosen to be 0.05

τ_x, τ_y = correlation times for the target azimuth and elevation
accelerations

- τ_a = correlation time for the atmospheric jitter position process
 σ_x^2, σ_y^2 = variance and mean-squared value for the target azimuth and elevation accelerations
 σ_a^2 = variance and mean-squared value for the atmospheric jitter position process.

The filter state estimate and error covariance matrix are propagated forward over a sample period as shown by the following equations [7:171-172]:

$$\hat{\mathbf{x}}_f(t_{i+1}^-) = \Phi_f(\Delta t) \hat{\mathbf{x}}_f(t_i^+) \quad (4.6)$$

$$\mathbf{P}_f(t_{i+1}^-) = \Phi_f(\Delta t) \mathbf{P}_f(t_i^+) \Phi_f^T(\Delta t) + \mathbf{Q}_{df} \quad (4.7)$$

where:

- $\hat{\mathbf{x}}_f(t_i)$ = filter's estimate of the 10- dimensional state vector
 $\mathbf{P}_f(t_i)$ = filter's covariance matrix (10 $\times$ 10)
 (t_i^-) = time instant before FLIR measurement is incorporated into the estimate at time t_i
 (t_i^+) = time instant after FLIR measurement is incorporated into the estimate at time t_i
 $\Phi_f(\Delta t)$ = time-invariant state transition matrix associated with propagation over the sample period: $\Delta t = t_{i+1} - t_i$

and the $\mathbf{Q}_{df}$ matrix is the obtained by the following:

$$\mathbf{Q}_{df} = \int_{t_i}^{t_{i+1}} \Phi_f(t_{i+1}, \tau) \mathbf{G}_f \mathbf{Q}_f \mathbf{G}_f^T \Phi_f^T(t_{i+1}, \tau) d\tau \quad (4.8)$$

Based upon the presentation by Netzer [18:47-48], and including the additional pogo state model, $\Phi_f(\Delta t)$ and $\mathbf{Q}_{df}$ are found to be:

$$\Phi_f(\Delta t) = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \Phi_{15} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Phi_{26} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \Phi_{35} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \Phi_{46} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Phi_{55} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_{66} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Phi_{77} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_{88} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_{99} & \Phi_{9,10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_{10,9} & \Phi_{10,10} \end{bmatrix} \quad (4.9)$$

where:

$$\begin{aligned} \Phi_{15} &= \tau_x \left[\Delta t - \tau_x \left(1 - \exp \left\{ -\frac{\Delta t}{\tau_x} \right\} \right) \right] \\ \Phi_{26} &= \tau_y \left[\Delta t - \tau_y \left(1 - \exp \left\{ -\frac{\Delta t}{\tau_y} \right\} \right) \right] \\ \Phi_{35} &= \tau_x \left(1 - \exp \left\{ -\frac{\Delta t}{\tau_x} \right\} \right) \\ \Phi_{46} &= \tau_y \left(1 - \exp \left\{ -\frac{\Delta t}{\tau_y} \right\} \right) \\ \Phi_{55} &= \exp \left\{ -\frac{\Delta t}{\tau_x} \right\} \\ \Phi_{66} &= \exp \left\{ -\frac{\Delta t}{\tau_y} \right\} \\ \Phi_{77} &= \exp \left\{ -\frac{\Delta t}{\tau_a} \right\} \\ \Phi_{88} &= \exp \left\{ -\frac{\Delta t}{\tau_a} \right\} \\ \Phi_{99} &= \frac{1}{\sqrt{1 - \zeta_{pf}^2}} \exp(-\zeta_{pf} \omega_{npf} \Delta t) \sin \left(\omega_{npf} \sqrt{1 - \zeta_{pf}^2} \Delta t + \arctan \left[\frac{\sqrt{1 - \zeta_{pf}^2}}{\zeta_{pf}} \right] \right) \\ \Phi_{9,10} &= \frac{1}{\omega_{npf}} \sqrt{1 - \zeta_{pf}^2} \exp(-\zeta_{pf} \omega_{npf} \Delta t) \sin(\omega_{npf} \sqrt{1 - \zeta_{pf}^2} \Delta t) \end{aligned}$$

$$\Phi_{10,9} = \frac{-1\omega_{npf}}{\sqrt{1-\zeta_{pf}^2}} \exp(-\zeta_{pf}\omega_{npf}\Delta t) \sin(\omega_{npf}\sqrt{1-\zeta_{pf}^2}\Delta t)$$

$$\Phi_{10,10} = \frac{1}{\sqrt{1-\zeta_{pf}^2}} \exp(-\zeta_{pf}\omega_{npf}\Delta t) \sin\left(\omega_{npf}\sqrt{1-\zeta_{pf}^2}\Delta t + \arctan\left[\frac{\sqrt{1-\zeta_{pf}^2}}{-\zeta_{pf}}\right] + \pi\right)$$

and

$$Q_{df} = \begin{bmatrix} Q_{11} & 0 & Q_{13} & 0 & Q_{15} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{22} & 0 & Q_{24} & 0 & Q_{26} & 0 & 0 & 0 & 0 \\ Q_{13} & 0 & Q_{33} & 0 & Q_{35} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{24} & 0 & Q_{44} & 0 & Q_{46} & 0 & 0 & 0 & 0 \\ Q_{15} & 0 & Q_{35} & 0 & Q_{55} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{26} & 0 & Q_{46} & 0 & Q_{66} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{77} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{88} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{99} & Q_{9,10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{10,9} & Q_{10,10} \end{bmatrix} \quad (4.10)$$

where:

$$Q_{11} = \sigma_x^2 \left[\frac{2}{3}\tau_x\Delta t^3 - 2(\tau_x\Delta t)^2 - 4\tau_x^3 \exp\left\{-\frac{\Delta t}{\tau_x}\right\} + 2\tau_x^3\Delta t - \tau_x^4 \exp\left\{-\frac{2\Delta t}{\tau_x}\right\} + \tau_x^4 \right]$$

$$Q_{13} = \sigma_x^2 \left[\tau_x\Delta t^2 + 2\tau_x^2\Delta t \exp\left\{-\frac{\Delta t}{\tau_x}\right\} + \tau_x^3 - 2\tau_x^3 \exp\left\{-\frac{\Delta t}{\tau_x}\right\} - 2\tau_x^2\Delta t \right. \\ \left. + \tau_x^3 \exp\left\{-\frac{2\Delta t}{\tau_x}\right\} \right]$$

$$Q_{15} = \sigma_x^2 \left[-2\tau_x\Delta t \exp\left\{-\frac{\Delta t}{\tau_x}\right\} + \tau_x^2 - \tau_x^2 \exp\left\{-\frac{2\Delta t}{\tau_x}\right\} \right]$$

$$Q_{22} = \sigma_y^2 \left[\frac{2}{3}\tau_y\Delta t^3 - 2(\tau_y\Delta t)^2 - 4\tau_y^3 \exp\left\{-\frac{\Delta t}{\tau_y}\right\} + 2\tau_y^3\Delta t - \tau_y^4 \exp\left\{-\frac{2\Delta t}{\tau_y}\right\} + \tau_y^4 \right]$$

$$Q_{24} = \sigma_y^2 \left[\tau_y\Delta t^2 + 2\tau_y^2\Delta t \exp\left\{-\frac{\Delta t}{\tau_y}\right\} + \tau_y^3 - 2\tau_y^3 \exp\left\{-\frac{\Delta t}{\tau_y}\right\} - 2\tau_y^2\Delta t \right. \\ \left. + \tau_y^3 \exp\left\{-\frac{2\Delta t}{\tau_y}\right\} \right]$$

$$Q_{26} = \sigma_y^2 \left[-2\tau_y\Delta t \exp\left\{-\frac{\Delta t}{\tau_y}\right\} + \tau_y^2 - \tau_y^2 \exp\left\{-\frac{2\Delta t}{\tau_y}\right\} \right]$$

$$\begin{aligned}
Q_{33} &= \sigma_x^2 \left[2\tau_x \Delta t - 3\tau_x^2 + 4\tau_x^2 \exp \left\{ -\frac{\Delta t}{\tau_x} \right\} - \tau_x^2 \exp \left\{ -\frac{2\Delta t}{\tau_x} \right\} \right] \\
Q_{35} &= \sigma_x^2 \left[\tau_x - 2\tau_x \exp \left\{ -\frac{\Delta t}{\tau_x} \right\} + \tau_x^2 \exp \left\{ -\frac{2\Delta t}{\tau_x} \right\} \right] \\
Q_{44} &= \sigma_y^2 \left[2\tau_y \Delta t - 3\tau_y^2 + 4\tau_y^2 \exp \left\{ -\frac{\Delta t}{\tau_y} \right\} - \tau_y^2 \exp \left\{ -\frac{2\Delta t}{\tau_y} \right\} \right] \\
Q_{46} &= \sigma_y^2 \left[\tau_y - 2\tau_y \exp \left\{ -\frac{\Delta t}{\tau_y} \right\} + \tau_y^2 \exp \left\{ -\frac{2\Delta t}{\tau_y} \right\} \right] \\
Q_{55} &= \sigma_x^2 \left[1 - \exp \left\{ -\frac{2\Delta t}{\tau_x} \right\} \right] \\
Q_{66} &= \sigma_y^2 \left[1 - \exp \left\{ -\frac{2\Delta t}{\tau_y} \right\} \right] \\
Q_{77} &= \sigma_a^2 \left[1 - \exp \left\{ -\frac{2\Delta t}{\tau_a} \right\} \right] \\
Q_{88} &= \sigma_a^2 \left[1 - \exp \left\{ -\frac{2\Delta t}{\tau_a} \right\} \right]
\end{aligned}$$

and $Q_{99}, Q_{9,10}, Q_{10,9}, Q_{10,10}$ are determined identically as the discrete-time, white, Gaussian noise covariance matrix pogo components of the truth model by solving Equation (3.37). The equations for these four components are not presented in the text because of their length, but the pogo components of the $\mathbf{Q}_{df}$ matrix are implemented in the software and have been validated based upon the following first-order approximation [7:170-174]:

$$\mathbf{Q}_{dfp}(t_i) \cong \mathbf{G}_{fp}(t_i) \mathbf{Q}_{fp}(t_i) \mathbf{G}_{fp}^T(t_i) [t_{i+1} - t_i] \quad (4.11)$$

where the subscript “ fp ” refers to the partition of the filter model representing the pogo phenomenon. The filters are “tuned” by choosing appropriate values for the correlation times (τ_x, τ_y, τ_a) and the variances ($\sigma_x^2, \sigma_y^2, \sigma_a^2$) corresponding to the ballistic trajectory and various pogo scenarios.

Note that the pointing controller used in this study is considered to be ideal. The dynamics of the pointing mechanism (servo lag, inertia, etc.) are neglected. Netzer [18] demonstrated that the errors resulting from any non-ideal controller dynamics are small and are interpreted by the Kalman filter as atmospheric jitter.

Following each filter propagation cycle, the estimates $\hat{x}_1(t_{i+1}^-)$ and $\hat{x}_2(t_{i+1}^-)$ are used to generate control signals to point the FLIR optical centerline at the target. It should be noted that these control signals are applicable for a non-rotating field-of-view (NRFOV) FLIR. The filter's estimates of the velocity states are used to rotate the field-of-view for the rotating field-of-view (RFOV) FLIR and the diagonal rotating field-of-view (DRFOV) FLIR. Chapters V and VI discuss in more detail the two rotation schemes.

4.3 The Measurement Model

Recall the measurement model of Equation (3.39) for the 14-state truth model. An alternative to this 64-dimensional, non-linear measurement model was developed by Rogers [21] during his AFIT research. Rogers suggested an enhanced correlation algorithm which is implemented to provide measurements to a linear Kalman filter measurement model. This correlation algorithm is "enhanced" over the standard correlation algorithm in the following ways [24]:

1. The current FLIR data frame is correlated with a template (an estimate of the target's intensity function), as opposed to correlation with the previous FLIR data frame.
2. Instead of outputting the peak of the correlation function, the enhanced correlator outputs the center of mass of that portion of the correlation function which is greater than some predetermined lower bound, a technique known as "thresholding". The enhanced correlator does not suffer the problem of distinguishing global peaks from local peaks, as do many conventional "peak-finding" correlation algorithms.
3. Using the enhanced correlation algorithm, the FLIR/laser pointing commands are determined via the Kalman filter propagation cycle as opposed to the output of a standard correlation algorithm.

4. The Kalman filter state estimate $\hat{\mathbf{x}}(t_i^+)$ is used to center the template, so the offsets seen in the enhanced correlation algorithm should be smaller than in the conventional correlator. This increases the amount of "overlap" between the actual FLIR data and the stored template, thus improving performance.

The output of the enhanced correlation algorithm are the two linear offsets x_c and y_c of Equations (3.1) and (3.2). These "pseudo-measurements" are then fed into a linear Kalman filter update cycle. The following two sections present an overview of the enhanced correlation algorithm. A more detailed analysis can be found in [13, 21].

4.3.1 Template Generation. As stated earlier, the template is an estimate of the target's intensity profile. This template is generated by averaging over the N most recent centered intensity functions. The intensity functions are centered on the FLIR plane by the "shifting property" of the Fourier transform, which is the domain in which the correlation is taking place. The memory size N is chosen according to how rapidly the shape functions change. Highly dynamic intensity functions require small values of N , while slowly varying functions can take advantage of large N values.

The premise behind this proposed finite memory filter can involve large memory requirements on a digital computer. To avoid this potential problem, the averaging is approximated by the use of "exponential smoothing". Exponential smoothing has properties very similar to those of finite memory filtering [8], but requires the storage of only one FLIR data frame instead of N frames, thus reducing computer storage requirements significantly. The template is maintained by the exponential smoothing algorithm given by the following equation:

$$\hat{\mathbf{I}}(t_i) = \gamma \mathbf{I}(t_i) + (1 - \gamma) \hat{\mathbf{I}}(t_{i-1}) \quad (4.12)$$

where:

$\hat{\mathbf{I}}(t_i)$ = "smoothed estimate" of the target's intensity function; i.e.,
the template

$\mathbf{I}(t_i)$ = "raw" intensity function from the current FLIR data frame

γ = smoothing constant; $0 < \gamma \leq 1$.

The smoothing constant γ is comparable to the N value of the true finite memory averager. From Equation (4.12), it can be seen that large values of γ tend to emphasize the current data frame and thus correspond to small N values in the finite memory filter. Based on studies by Suizu [23] and Loving [6], a smoothing constant of $\gamma = 0.1$ will be used throughout this thesis effort.

Figure 4.1 on the next page shows the structure of the enhanced correlator/linear measurement model data processing algorithm. This algorithm is strictly for a NRFOV FLIR sensor. Chapter V presents a modification to this algorithm in order to simulate the RFOV and DRFOV sensors. Note that the portion of the algorithm enclosed in dotted lines is the template generation scheme. After the raw FLIR data is transformed into the Fourier domain by a fast Fourier transform (FFT), it is centered on the FLIR plane by shifting it an amount equal to:

$$x_{shift} = \hat{x}_1(t_i^+) + \hat{x}_7(t_i^+) + \hat{x}_9(t_i^+) \cos \hat{\theta}_f \quad (4.13)$$

$$y_{shift} = \hat{x}_2(t_i^+) + \hat{x}_8(t_i^+) - \hat{x}_9(t_i^+) \sin \hat{\theta}_f \quad (4.14)$$

where:

$$\cos \hat{\theta}_f = \frac{\hat{x}_3(t_i^+)}{\sqrt{\hat{x}_3^2(t_i^+) + \hat{x}_4^2(t_i^+)}} \quad (4.15)$$

$$\sin \hat{\theta}_f = \frac{-\hat{x}_4(t_i^+)}{\sqrt{\hat{x}_3^2(t_i^+) + \hat{x}_4^2(t_i^+)}} \quad (4.16)$$

It should be noted that the reason the minus signs are in Equations (4.14) and (4.16) is because of the difference in the defined orientations of the Target and FLIR

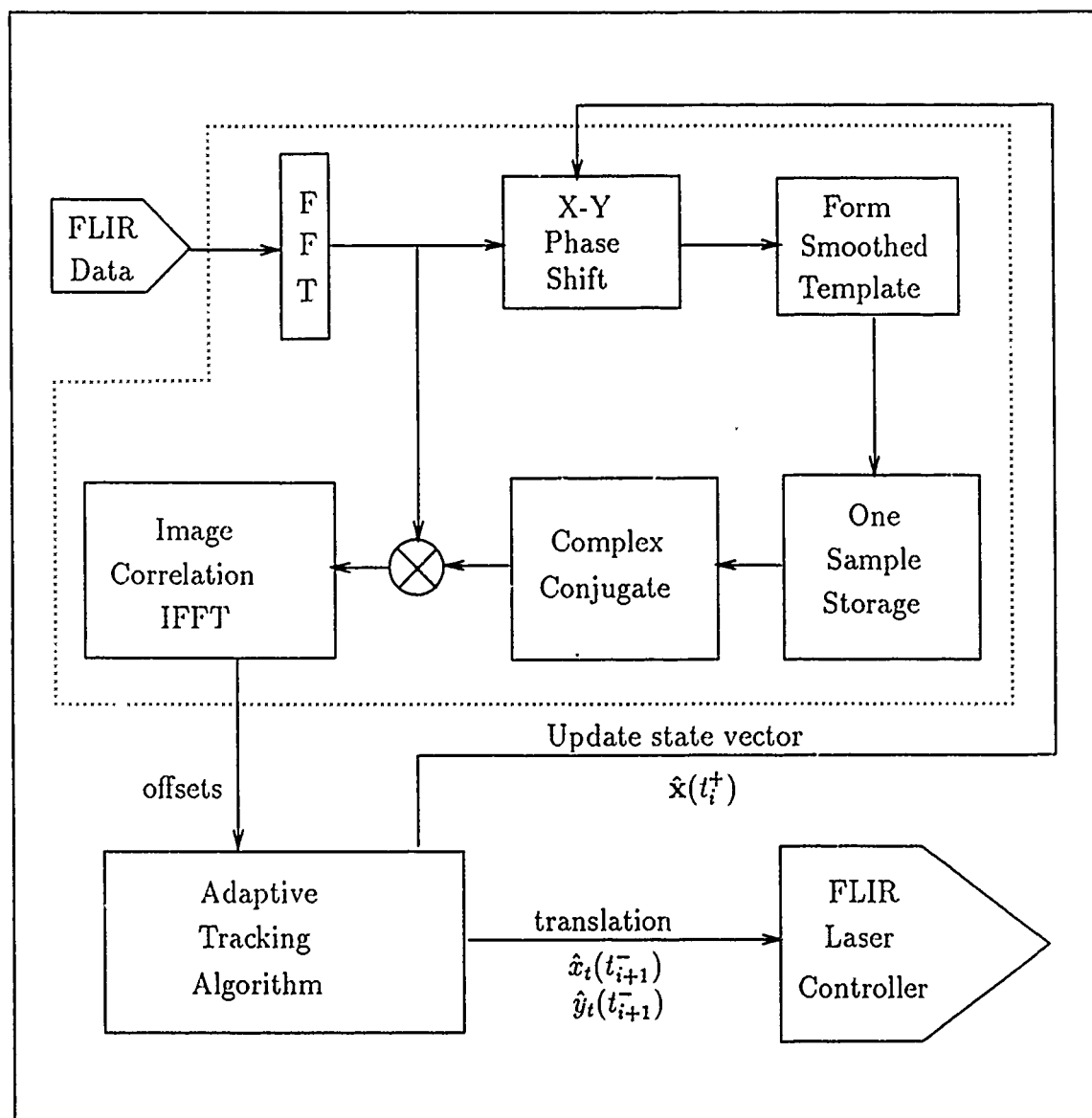


Figure 4.1. Enhanced Correlator/Linear Measurement Model Data Processing Algorithm

coordinate frame axes (see Figure 3.1). This is compatible with the structure of Equation (3.2) of the truth model. It should also be noted that, since the value of the plume pogo is referenced from the pogo "equilibrium point" (see Figure 3.3), the true centroid locations on the FLIR plane as determined by Equations (3.1) and (3.2) actually should include a constant "offset" in both FLIR directions. These offsets are constant and are equal to the distance between the hard-body's center of mass and the pogo "equilibrium point." Similarly, to be absolutely correct, the above shift equations should also consider the effect of these offsets. Since the present filter structure has no way of estimating these offsets based on FLIR data alone, and the major challenge of this thesis is to characterize the pogo phenomenon relative to the pogo "equilibrium point," the offsets were purposely left out of both the truth and filter measurement models.

The above shifts are performed using the shifting property of Fourier transforms, which states that a translational shift in the spatial domain is equivalent to a linear phase shift in the frequency domain [24]. The phase shift is computed as follows:

$$F\{g(x - x_{shift}, y - y_{shift})\} = G(f_x, f_y) \exp\{-j2\pi(f_x \cdot x_{shift} + f_y \cdot y_{shift})\} \quad (4.17)$$

where:

$$\begin{aligned} g(x, y) &= \text{2-dimensional spatial data array} \\ F\{\cdot\} &= \text{Fourier transform operator} \\ G(f_x, f_y) &= F\{g(x, y)\} \end{aligned}$$

After the data is centered by the above phase shift, it is incorporated into the template according to the exponential smoothing algorithm of Equation (4.12). The template is then stored and correlated with the subsequent FLIR data frame to produce the "pseudo-measurement."

4.3.2 "Pseudo-Measurements" by Enhanced Correlation. The enhanced correlator algorithm provides "pseudo-measurements" in the form of position offsets from the centroid of the target intensity image to the center of the FOV. The current FLIR data frame and the stored template are correlated in the Fourier domain space. This cross-correlation is computed by taking the inverse fast Fourier transform (IFFT) of the following equation [21]:

$$F\{g(x, y) * l(x, y)\} = G(f_x, f_y)L^*(f_x, f_y) \quad (4.18)$$

where:

$$\begin{aligned} F\{\cdot\} &= \text{Fourier transform operator} \\ g(x, y) * l(x, y) &= \text{cross correlation of } g(x, y) \text{ and } l(x, y) \\ g(x, y) &= \text{measured target intensity function; the current FLIR data frame} \\ l(x, y) &= \text{expected target intensity function; the template} \\ G(f_x, f_y) &= F\{g(x, y)\} \\ L^*(f_x, f_y) &= \text{complex conjugate of } F\{l(x, y)\} \end{aligned}$$

The Fourier transform, $F\{\cdot\}$, is implemented in the software via the Cooley-Tukey Fast Fourier Transform (FFT) algorithm [24].

Once the IFFT is performed, the correlation function, $g(x, y) * l(x, y)$, is "thresholded" so that any value in the correlation function less than 30% of the function's maximum value is set to zero [6, 18]. The location of the center-of-mass of the "thresholded" function represents the relative displacement between the current FLIR data frame and the template.

As shown in Figure 4.1, the displacements or "offsets" are outputs of the IFFT block and are assumed to be the result of hardbody dynamic activity, atmospheric jitter, plume pogo effect, and measurement noise. Therefore:

$$x_{offset} = x_t + x_a + x_p \cos \theta_f + v_{f1} \quad (4.19)$$

$$y_{offset} = y_t + y_a - x_p \sin \theta_f + v_{f2} \quad (4.20)$$

where:

$$\cos \theta_f = \frac{v_x}{\sqrt{v_x^2 + v_y^2}} \quad (4.21)$$

$$\sin \theta_f = \frac{-v_y}{\sqrt{v_x^2 + v_y^2}} \quad (4.22)$$

Using the state space representation in Equation (4.1), Equations (4.19) and (4.20) can be written in state space form as [8]:

$$\mathbf{z}(t_i) = \mathbf{h}_f[\mathbf{x}_f(t_i), t_i] + \mathbf{v}_f(t_i) \quad (4.23)$$

where:

- $\mathbf{z}(t_i)$ = $[x_{offset}, y_{offset}]^T$; measured in pixels
- $\mathbf{x}_f(t_i)$ = 10-dimensional filter state vector of Equation (4.1)
- $\mathbf{h}_f[\cdot]$ = 2-dimensional, nonlinear, measurement vector function
- $\mathbf{v}_f(t_i)$ = 2-dimensional, discrete-time, zero-mean, white Gaussian measurement corruption noise of covariance $\mathbf{R}_f$;
measured in pixels

Note that because of the pogo states being defined along the velocity vector and being included in the output equations, this measurement model is nonlinear in the filter states and; the extended Kalman filter update cycle described in Chapter II (Equations (2.12)–(2.14)) must be applied. The measurement noise $\mathbf{v}_f(t_i)$ reflects the spatially correlated background noise (Section 3.3), the FLIR sensor noise, and errors due to the FFT/IFFT processes. The covariance matrix associated with this

cumulative error has been found to be [13, 17, 21]:

$$\mathbf{R}_f = \begin{bmatrix} 0.00436 & 0 \\ 0 & 0.00598 \end{bmatrix} \quad (4.24)$$

The linearized $\mathbf{H}_f$ matrix based upon Equation (2.15) is:

$$\mathbf{H}_f = \begin{bmatrix} 1 & 0 & H_1 & H_2 & 0 & 0 & 1 & 0 & H_3 & 0 \\ 0 & 1 & H_4 & H_5 & 0 & 0 & 0 & 1 & H_6 & 0 \end{bmatrix} \quad (4.25)$$

where:

$$H_1 = \left. \frac{x_9 x_4^2}{\sqrt[3]{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.26)$$

$$H_2 = \left. \frac{-x_9 x_4 x_3}{\sqrt[3]{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.27)$$

$$H_3 = \left. \frac{x_3}{\sqrt{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.28)$$

$$H_4 = \left. \frac{-x_9 x_4 x_3}{\sqrt[3]{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.29)$$

$$H_5 = \left. \frac{x_9 x_3^2}{\sqrt[3]{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.30)$$

$$H_6 = \left. \frac{x_4}{\sqrt{x_3^2 + x_4^2}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_f(t_i^-)} \quad (4.31)$$

This completes the structure of the Kalman filter models used in this study. It should be noted that, although the MMAF was not implemented in this thesis effort, the concepts discussed in this chapter are directly applicable to the MMAF. That also includes the data processing of Figure 4.1. See references [19] and [24] for additional details.

4.4 Summary

This chapter has presented the models upon which the Kalman filter is based. The ten-state filter vector includes models to estimate trajectory dynamics, atmospheric jitter, and plume pogo. All of these models are based upon Gauss-Markov stochastic processes. "Pseudo-measurements" are created by correlating the current FLIR data frame with an adaptively constructed template representing the target's infrared intensity profile. Figure 4.1 shows the overall filter processing algorithm for a non-rotating field-of-view (NRFOV) sensor. When pogo states are introduced into the filter state vector, the output model becomes nonlinear in the filter states and requires the use of the extended Kalman filter update cycle when incorporating the "pseudo-measurements" from the data processing algorithm.

V. Tracking Algorithm

5.1 Introduction

This chapter presents the overall tracking algorithm used in this research by combining the principles presented in the preceding chapters. First, an overall view of the algorithm is presented, along with a proposed structure for a MMAF algorithm. Second, the method for field-of-view (FOV) processing is discussed, followed by the different FOV rotation schemes analyzed in this research and the relationship of the rotating FOV to the overall tracking algorithm that was presented in Chapter IV (Figure 4.1). The filter and truth model parameters are then presented before concluding with the tools used to evaluate the performance of the tracking algorithm (i.e., statistical calculations, performance plot formats, and greyscale diagrams).

5.2 Overview of the Tracking Algorithm.

The main objective of this research effort is to design an algorithm to track a ballistic missile accurately when its plume is undergoing pogo (oscillation) along the longitudinal axis. A Bayesian Multiple Model Adaptive Filter (MMAF) tracking algorithm was originally proposed to increase performance over a single filter algorithm, but for the reasons described in Section 6.8, this MMAF was never implemented. This section presents the proposed structure of the MMAF which, once implemented, should demonstrate increased tracking performance over the single filter models which were implemented in this research. As mentioned earlier, the reason for presenting the proposed MMAF structure is for completeness of the objectives described in Chapter I, and for the benefit of suggested continuations of this research effort (see Chapter VII).

The proposed MMAF is composed of five elemental filters based upon a nominal ballistic missile trajectory and varying scenarios for the plume's pogo effect.

Filter	Filter Dimension	Tuning Characteristics
1	8	Tuned for a truth model exhibiting no plume pogo
2	10	Tuned for a truth model exhibiting high frequency and large magnitude pogo
3	10	Tuned for a truth model exhibiting low frequency and large amplitude pogo
4	10	Tuned for a truth model exhibiting high frequency and small amplitude pogo
5	10	Tuned for a truth model exhibiting low frequency and small amplitude

Table 5.1. Elemental Filters of Proposed MMAF

The actual structure of the MMAF can not be determined until the results of a robustness study are performed (Section 1.3.5). Again, for reasons discussed in Section 6.8, a robustness study was inappropriate based upon the results of this thesis research; but a suggested MMAF structure is still presented in this chapter as an expected baseline for future research. The actual research performed in this thesis implemented a tracking algorithm based upon each of the single elemental filters in the proposed MMAF. Each of the elemental filters is based upon an 8×8 pixel FOV and is "tuned" based upon specific dynamics of the plume pogo effect in the truth model. More concerning the FOV data processing is presented in Section 5.3. Table 5.1 presents the proposed structure of the MMAF.

Note that the first filter is an eight-dimensional filter, while the other four are

ten-dimensional filters. The ten-dimensional filters are of the structure defined in Chapter IV, while the eight-dimensional filter is of the same structure without the two additional pogo states. The pogo states are omitted because of the desire to have an elemental filter in the MMAF structure specifically tuned to a scenario where the plume is not "pogoing". This desire is based upon the observed condition that a plume will not pogo until the missile reaches a specific altitude where the pressure gradients along the hardbody are favorable for the pogo phenomenon [10]. The eight-state and ten-state filters will be discussed more fully in Chapter VI, where the specific choice of tuning parameters for each of the filter structures, and the results of their performance analyses are presented.

5.3 Field-of-View Data Processing

In order to process the current data frame to produce the required "pseudo-measurements" for the Kalman filter update cycle, the field-of-view used in this research is represented as an 8×8 data array of pixels. The 8×8 FOV structure was chosen, rather than some larger FOV's that have been investigated in the past [5, 25], based upon the benign missile trajectory used in the simulation. Since the ballistic missile is assumed not to perform maneuvers or "jinks" during its ascent trajectory, the 8×8 FOV tracker, with pixels of length 15 micro-radians on a side, does not lose track on the target and provides accurate tracking estimates in both directions on the FLIR plane. "Staging" events during the ballistic missile ascent can cause large differences in the missile's acceleration characteristics for which an 8×8 FOV tracker might lose lock; but for the purposes of this initial research, such staging was not considered (see Chapter VII).

In addition to staging, the maximum amplitude of the plume's pogo along the longitudinal axis could possibly cause the 8×8 FOV tracker to lose lock on the target. However, based upon the maximum amplitude assumption of 60% occlusion of the missile hardbody (which corresponds to approximately three pixels on the FLIR

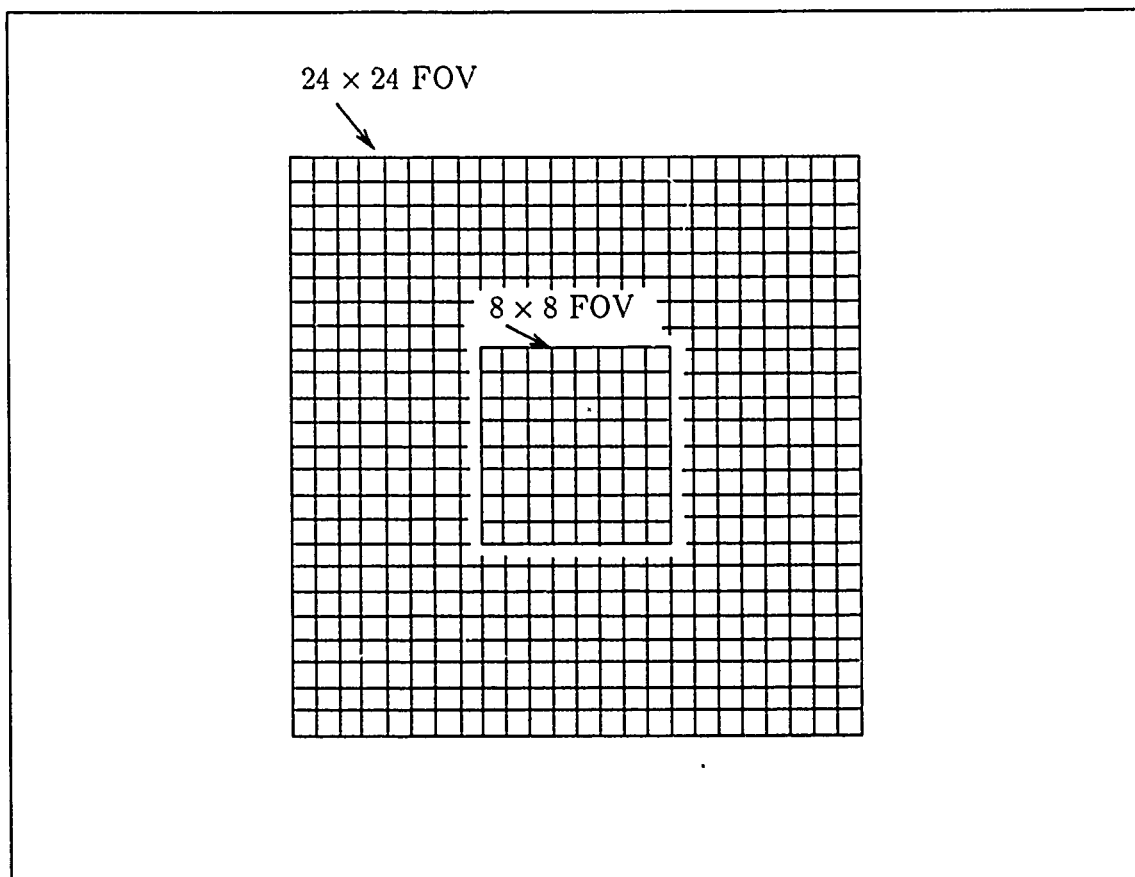


Figure 5.1. 8×8 Field-of-View Filter

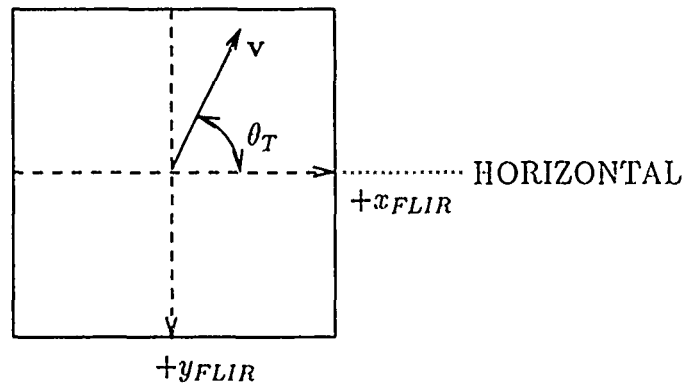
plane. Section 3.4.3), the 8×8 FOV tracker was able to maintain lock on the target. As mentioned in Chapter I, previous research [5, 24] considered a MMAF tracker that included 24×24 , 8×24 , and 24×8 FOV elemental filters for the purposes of tracking highly maneuverable targets. For the benign dynamic characteristics of the ballistic missile hardbody and considering the maximum pogo amplitude (approximately three pixels) of the plume, an 8×8 FOV was found to be appropriate for this research. Figure 5.1 gives a perspective of the 8×8 tracking window against a 24×24 array of pixels. The reason for comparing the relative sizes of the 8×8 FOV to a 24×24 array is that the correlation algorithms of Figures 4.1 and 5.3 process data from a 24×24 array of pixel sensors, where the infrared intensity values of the plume are localized to an 8×8 tracking window.

5.4 *Field-of-View Rotation*

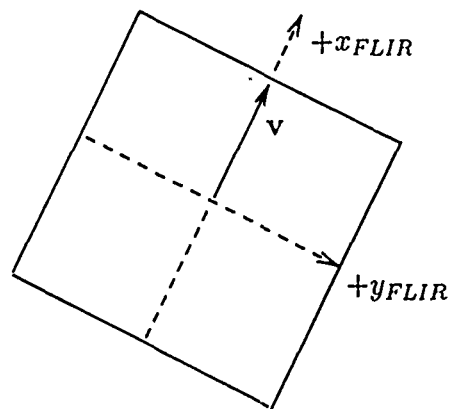
As mentioned in Chapter I, in addition to characterizing the ballistic missile exhaust plume dynamics, rotating the FOV of the FLIR sensor to enhance tracking performance was an additional objective of this research. Three different analyses are conducted to compare performance of a FOV that does not rotate, i.e., the non-rotating field-of-view (NRFOV), a rotating field-of-view (RFOV) implemented by Norton [19], and a diagonal rotating field-of-view (DRFOV). The NRFOV is the standard tracker used in previous studies conducted at AFIT [5, 18, 24] which maintains the x -axis of the FLIR parallel to the local horizon. Norton implemented a RFOV which aligns the x -axis of the 8×8 FOV FLIR parallel to the filter estimated velocity vector. The DRFOV is a rotation scheme which aligns the diagonal of the 8×8 FOV with the filter's estimate of the velocity vector. Each of these rotation schemes is presented in Figure 5.2.

The concept of a RFOV was originally conceived to maintain lock on a highly maneuvering target that could "jink" in either of the two FLIR directions. The idea of a rectangular rotating field-of-view (RRFOV) was first suggested by Leeney [5] using a rectangular 8×24 FOV which was originally implemented by Tobin [24] in a non-rotating scheme. Norton demonstrated that a MMAF algorithm composed of 8×8 FOV filters rotated so that the x -axis is aligned with the velocity vector [19:64-67], as well as adaptively transforming the filter's dynamic driving noise matrix $\mathbf{Q}_{fd}$ so that the target's acceleration distribution corresponds to the direction of the target's "jink" maneuver [19:72-76], would improve tracking performance over algorithms implemented by Leeney [5] and Tobin [24]. Although the adaptive rotation of the $\mathbf{Q}_{fd}$ matrix does not apply to a benign ballistic missile target, because the acceleration is modelled identically in each of the FLIR channels, the rotation of the 8×8 FOV filter is applied in this research effort. Due to time constraints, the implementation of the RRFOV as discussed in Section 1.3.2 was not accomplished. However, it is a suggested topic for future research particularly in the study of a simulated "staging"

(a) NRFOV



(b) RFOV



(c) DRFOV

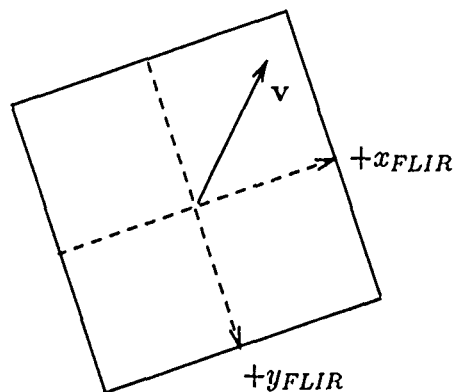


Figure 5.2. Field-of-View Rotation Schemes

event since the sudden change in acceleration may cause an 8×8 filter to lose lock on the target (see Chapter VII).

To implement a rotation scheme in the existing data processing algorithm, whether it be the RFOV, DRFOV or the RRFOV, some modifications are made to the algorithm depicted in Figure 4.1. These modifications are depicted by the "Rotate" blocks in Figure 5.3. The basis of the rotating FOV is the estimate of the target's positive velocity orientation angle (see Figure 5.2: $\hat{\theta}_f$ is the filter's estimate of θ_T):

$$\hat{\theta}_f = \arctan \left[\frac{-\hat{v}_y}{\hat{v}_x} \right] \quad (5.1)$$

Note that the terms in this equation are state estimates of the third and fourth elements of the filter's state vector. Therefore, the filter is capable of estimating the velocity orientation angle in addition to the translational position states. This permits the filter to provide control inputs to the FLIR sensor to perform both a translation and a rotation of the FOV for on-line application of the tracking algorithm. Also note the negative sign on the $\hat{v}_y$ term of Equation (5.1). This notation is consistent with the previously defined coordinate frame of the FLIR (Figure 3.1) used in past research [24:37-38]. Inserting the negative sign in the numerator of Equation (5.1) keeps the filter velocity orientation angle defined as positive in the counter-clockwise direction from the positive x -axis on the FLIR plane (looking at the target along the LOS vector from the origin of the inertial frame); and it produces a direct correlation with the truth model velocity orientation angle θ_T of Figure 5.2, where:

$$\theta_T = \arctan \left[\frac{\dot{\beta}(t)}{\dot{\alpha}(t)} \right] \quad (5.2)$$

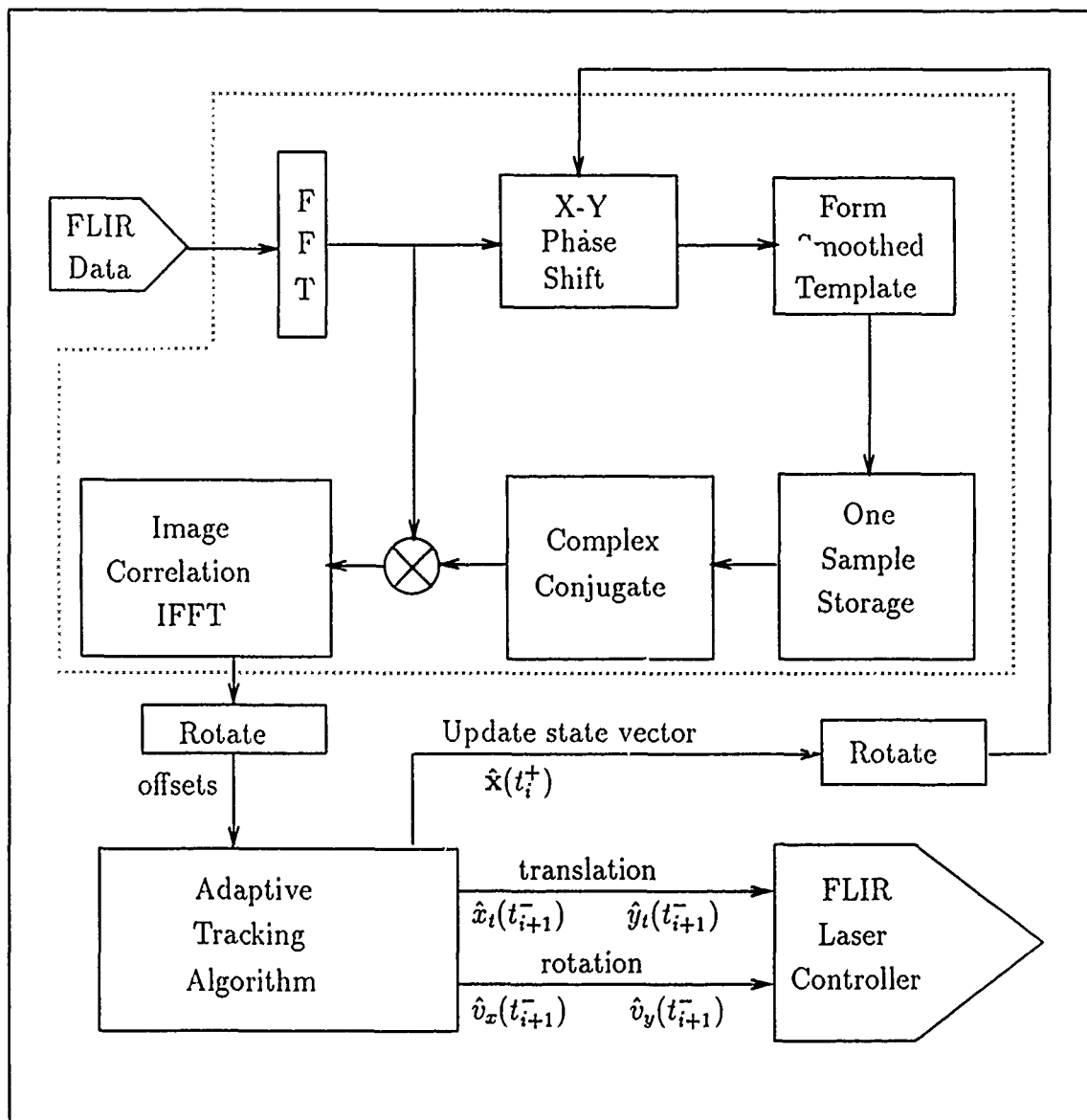


Figure 5.3. Data Processing Algorithm for a Rotating Field-of-View

and the directions of the inertial azimuth (α) and elevation (β) angles are defined in Section 3.4.1.

To simulate the physical rotation of the FLIR sensor, the incoming FLIR data is fed into the data processing algorithm in a rotated coordinate system based on the orientation angle of Equation (5.1). This is simulated by performing a negative rotation, based on a positive velocity orientation angle from the horizontal, on the location and orientation of the individual Gaussian intensity functions. This directly corresponds to a positive rotation of the FOV, which when applied to the FLIR sensor, aligns the FOV with the positive velocity orientation angle. Based upon a development by Millner [17:157-163], the intensity function peaks are transformed from the target frame, where they are positioned with respect to the center of mass of the missile (see Figure 3.6), to the FLIR image plane. Millner first transforms the intensity functions to a plane perpendicular to the LOS vector. The final transformation about the LOS vector, implemented by the following relationships:

$$\cos \theta_T = \frac{\dot{\alpha}(t)}{v_{\perp LOS}} \quad (5.3)$$

$$\sin \theta_T = \frac{\dot{\beta}(t)}{v_{\perp LOS}} \quad (5.4)$$

where all of the terms are previously defined in Chapter 4, moves the intensity function peaks into the proper orientation on the FLIR plane for a non-rotating FOV sensor. The intensity distribution about these peaks was previously defined by Equation (3.38). In simulating the rotating field-of-view, rather than perform the last transformation by θ_T , the velocity orientation angle given in Equation (5.1) is used to rotate the intensity function peaks as if the FOV had been rotated positively in a counter-clockwise direction from a perspective looking out of the sensor along the LOS vector.

Mathematically, the simulation of rotating the input FLIR data is done by first rotating the intensity function peaks defined in Equation (3.38) by the rotation

algorithm produced by Norton [19]:

$$\begin{bmatrix} x'_{peak} \\ y'_{peak} \end{bmatrix} = \begin{bmatrix} \cos \hat{\theta}_f & -\sin \hat{\theta}_f \\ \sin \hat{\theta}_f & \cos \hat{\theta}_f \end{bmatrix} \begin{bmatrix} x_{peak} \\ y_{peak} \end{bmatrix} \quad (5.5)$$

where the primed variables correspond to the rotated coordinate system. The intensity distribution (Equation 3.38) is then oriented about the rotated peaks by negating Millner's final transformation (by θ_T) with the filter's estimate of the velocity orientation angle of Equation (5.1) via the following calculation:

$$I[x', y', x'_{peak}(t), y'_{peak}(t)] = I_{max} \exp[-0.5[\Delta x' \Delta y'] \mathbf{P}^{-1} [\Delta x' \Delta y']^T] \quad (5.6)$$

where:

$$\Delta x' = (x' - x'_{peak}) \cos \Delta\theta + (y' - y'_{peak}) \sin \Delta\theta$$

$$\Delta y' = (y' - y'_{peak}) \cos \Delta\theta - (x' - x'_{peak}) \sin \Delta\theta$$

$$\Delta\theta = \text{the difference between the truth model velocity orientation angle and the filter computed velocity orientation angle, i.e.}$$

$$\Delta\theta = \theta_T - \hat{\theta}_f$$

$$x', y' = \text{rotated coordinates from the original FLIR coordinate frame via the same transformation used in Equation (5.5).}$$

Once the incoming FLIR data is rotated by the above transformations, the data processing algorithm in Figure 5.3 generates the templates in the same manner as was done in Figure 4.1, except that now the measurement data entering the data processing algorithm is in a rotated coordinate system.

Recall the shifts of Equations (4.13) and (4.14). The incoming data is centered via these translational shifts in both azimuth and elevation directions on the FLIR plane. Recall that these shifts are used to center the incoming data on the FLIR FOV so that the offset can be regulated to zero over the ensuing sample period. Note, however, that these shifts are computed in the filter coordinate system, while the

current image data is represented in a rotated coordinate system simulating a RFOV. To implement the RFOV data processing algorithm properly, the translational shifts are transformed into the rotated coordinate frame by the same transformation that was used on the hotspot peaks:

$$\begin{bmatrix} x'_{shift} \\ y'_{shift} \end{bmatrix} = \begin{bmatrix} \cos \hat{\theta}_f & -\sin \hat{\theta}_f \\ \sin \hat{\theta}_f & \cos \hat{\theta}_f \end{bmatrix} \begin{bmatrix} x_{shift} \\ y_{shift} \end{bmatrix} \quad (5.7)$$

where x_{shift} and y_{shift} are given by Equations (4.12) and (4.13).

This transformation is represented by the "Rotate" block directly following the "Adaptive Tracking Algorithm" block of Figure 5.3. The current image data and the filter's estimate of the target centroid (the shifts) are now represented in corresponding coordinate frames, i.e., the rotated coordinate frame, and the template generation proceeds as presented in Section 4.3.1.

The only other modification to the data processing algorithm of Figure 5.3 is the "Rotate block" following the "IFFT" block. Recall that the outputs of the "IFFT" block are the linear offsets between the current data image and the centered template. These offsets are depicted in Equations (4.19) and (4.20) and represent the linear measurements from the enhanced correlator which are fed into the Kalman filter update cycle. Recall, however, that the current states used in the Kalman filter equations are represented in the original unrotated FLIR coordinate frame, while the linear measurement offsets from the enhanced correlator are coordinatized in the rotated frame. Thus, to ensure compatibility of coordinate frames once again, the measurement offsets are transformed by the opposite transformation that was performed on the shifts and the intensity function peaks:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \hat{\theta}_f & \sin \hat{\theta}_f \\ -\sin \hat{\theta}_f & \cos \hat{\theta}_f \end{bmatrix} \begin{bmatrix} z'_1 \\ z'_2 \end{bmatrix} \quad (5.8)$$

where z_1 and z_2 are the components of the two-dimensional measurement vector

in Equation (4.22) in the original coordinate frame, and z'_1 and z'_2 are the linear outputs in the rotated coordinate frame of the data processing algorithm for the rotating FOV.

The above simulation of the rotating field-of-view was originally developed by Norton [19] to align the estimated velocity vector with one axis of the FLIR FOV (the x -axis in this description). It should be noted that the same concepts applied for the RFOV are also applicable to the DFOV and RRFOV. As mentioned earlier, this research implements a DRFOV, as well as the RFOV; and the performance of each is discussed in Section 6.4, along with performance results of a NRFOV filter analysis.

5.5 Truth Model Parameters

The initial conditions of the target's inertial position and velocity vectors (see Figures 3.7 and 5.4 for coordinate system axis definitions) for the nominal ballistic missile trajectory are:

$$\begin{aligned} e_x &= 20,000 \text{ meters} \\ e_y &= 100,000 \text{ meters} \\ e_z &= 2,000,000 \text{ meters} \\ v_x &= -2500 \frac{\text{meters}}{\text{second}^2} \\ v_y &= 4330 \frac{\text{meters}}{\text{second}^2} \\ v_z &= 0 \frac{\text{meters}}{\text{second}^2} \end{aligned}$$

and the components of the acceleration vector are calculated based upon the discussion in Section 3.2.1.

Note the large initial condition in the e_z direction. This initial condition was intentionally made large to simulate the large effective range when considering an orbiting optical platform that is undergoing bending/vibrational effects (see Figure 3.2). Using these initial conditions on the position parameters produces an effective range on the order of 2×10^6 meters. The reason that the e_z direction was

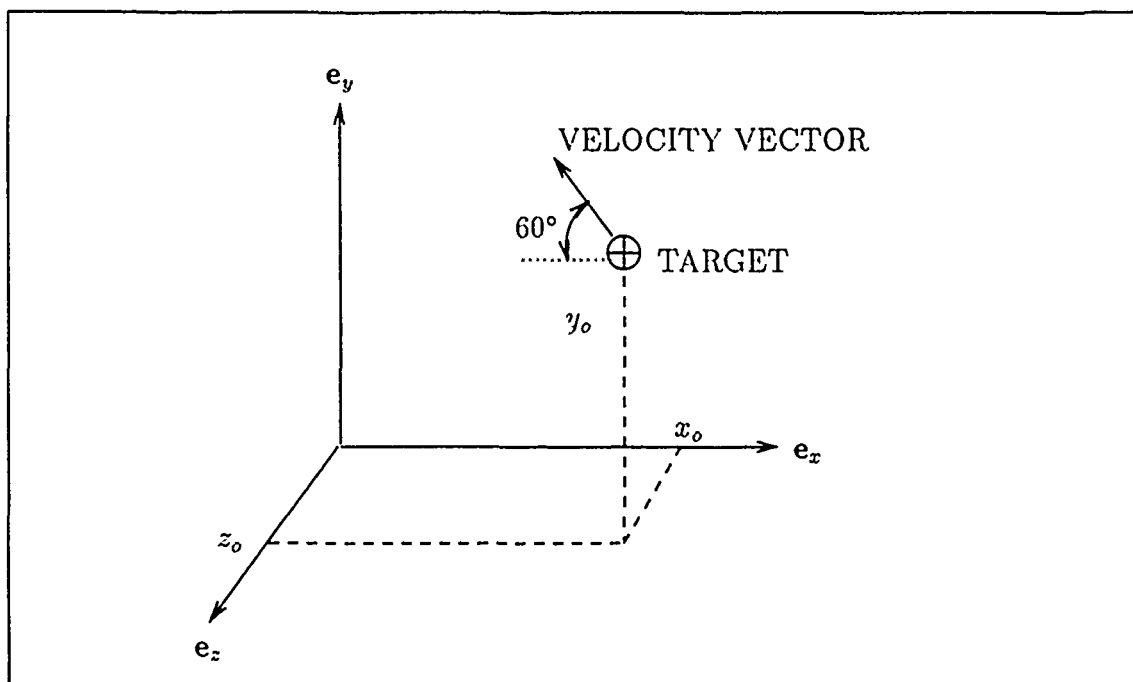


Figure 5.4. Ballistic Missile Trajectory in Inertial Space

chosen to generate a large effective range is because it is a “benign” axis when implementing the ballistic missile trajectory for this particular simulation. Figure 5.4 shows the missile trajectory used in the simulation. All of the missile dynamics are simulated to occur in a plane parallel to the inertial $e_x - e_y$ plane. Therefore, the e_z axis is basically used to scale the desired range when considering an orbiting optical platform in the simulation. The effective range is also used in determining the required pixel proportionality constant discussed in Chapter 3, as well as being involved in the various coordinate frame transformations that have been implemented in the simulation software over the past ten years. The large initial component in e_z does not affect the true missile trajectory as generated by the development in Section 3.2.1, since the only forces acting on the missile are assumed to be the thrust force and the force due to the Earth’s gravitational attraction, where both are simulated to occur in a plane parallel to the $e_x - e_y$ plane.

Also note the relative magnitudes of the two initial velocity components. The

ratio of these two components form the velocity vector orientation angle of the target in an inertial coordinate frame. As mentioned earlier, this orientation angle was chosen to be approximately 60° , as is evidenced by the choice of the initial velocity components in the inertial coordinate frame.

The maximum intensity value of each of the intensity functions in Equation (3.38) is 20 intensity units. The RMS value of v_{jk} , which is the sum total of the spatially correlated background noise (b_{jk}) and the FLIR sensor noise (n_{jk}) in Equation (3.39), is one. This produces a signal-to-noise ratio (SNR) of 20, which, as stated by Tobin [24], is typical of many tracking scenarios of current interest.

One of the assumptions used to generate the size of the plume relative to the diameter of the ballistic missile in the target coordinate frame is that the plume width is on the order of 30 times the diameter of the missile for certain altitudes of interest [10]. To implement an 8×8 FOV tracker for the ballistic missile plume which undergoes the pogo effect, it was desired to "fit" the plume in a 5×5 FOV window under the condition of a non-pogoing plume. A 5×5 window was chosen to permit the 8×8 FOV tracker to maintain lock on the target even when the maximum plume amplitude (approximately three pixels) is reached [10]. Based upon these assumptions and the effective range discussed earlier, the reference hotspot dispersion in the e_{pv} direction of the target frame (Equation 3.46) is chosen to be 1 pixel when projected onto the FLIR image plane; and with an aspect ratio of 1.5, the hotspot dispersion along the e_v direction in the target frame becomes 1.5 pixels when projected onto the FLIR plane. The pixel proportionality constant (k_p) required to meet all of the above criteria is on the order of 15 micro-radians/pixel, as presented in Section 3.2.1.

The variance and mean squared value for the of the atmospheric jitter process in the truth model, given by x_a and y_a in Equations (3.1) and (3.2), is 0.2 pixels² [24]. The truth model parameters used to describe the bending/vibration phenomenon of the optical platform are presented in Section 3.2.3, and the range of parameters used

to study the pogo phenomenon are presented in Section 3.2.4 and Appendix A.

5.6 Filter Parameters

In the tracking simulation, the filter is initialized to zero errors in the position and velocity states at $t = 0$. This is an artificial initial condition intended to allow a first analysis of tracking, devoid of recovery problems associated with poor initial conditions. Non-zero initial errors are introduced in subsequent analyses to investigate the acquisition routine developed by Tobin [24], but for this study the tracker is assumed initially to have perfect knowledge of the target location. The position states x_1 and x_2 are initialized so that the target's center of mass is centered in the FLIR FOV (see Figure 3.1). The velocity states x_3 and x_4 are initialized in accordance with the target's actual inertial position and velocity vectors of Section 5.5, and the transformations of Equations (3.42) and (3.44). The acceleration states x_5 and x_6 are initialized by subtracting the velocity states at $t = 0$ from the velocity states at $t = \frac{1}{30}$ and then dividing by $\frac{1}{30}$. The atmospheric jitter states x_7 and x_8 , as well as the pogo states x_9 and x_{10} are initialized to zero. The initial state covariance matrix, $\mathbf{P}(t_o)$, is given by [24]:

$$\mathbf{P}(t_o) = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 \end{bmatrix} \quad (5.9)$$

Since the MMAF was not implemented during this research, see reference [19] for initial conditions on the hypothesis conditional probabilities for the elemental filters in the MMAF structure, as well as other details of the reacquisition routine to be used should any of the elemental filters diverge during the simulation.

5.7 Tracking Algorithm Statistics

The tracking algorithm performance is evaluated by Monte Carlo simulation techniques [7]. Previous research has demonstrated that ten Monte Carlo runs exhibit sufficient convergence to the actual statistics resulting from an infinite number of runs [2, 3, 16]. Based upon this previous research, ten Monte Carlo runs are used to analyze the tracker's performance in this research effort.

The sample mean errors of the tracking algorithm's estimates are calculated as [24]:

$$\begin{aligned}\bar{E}_{xd}(t_i) &= \frac{1}{N} \sum_{n=1}^N e_{xdn}(t_i) \\ &= \frac{1}{N} \sum_{n=1}^N [x_{dn}(t_i) - \hat{x}_{dnf}(t_i)]\end{aligned}\tag{5.10}$$

where:

$\bar{E}_{xd}(t_i)$ = sample mean error of the x target position estimate at time t_i , averaged over N runs

$e_{xdn}(t_i)$ = error in the x position estimate at t_i during simulation n

$\hat{x}_{dnf}(t_i)$ = estimate of target's x position at t_i during simulation n

$x_{dn}(t_i)$ = truth model value of the target's x position at time t_i during simulation n

N = number of Monte Carlo runs.

The sample variance of the error is given by:

$$\sigma_{xd}^2 = \frac{1}{N-1} \sum_{n=1}^N \{e_{xdn}^2(t_i)\} - \frac{N}{N-1} \bar{E}_{xd}^2(t_i) \quad (5.11)$$

where all the quantities are defined above.

The error committed in estimating the target's position is the most important parameter when evaluating the tracker's performance. The error committed in estimating the location of the centroid of the target's image on the FLIR plane is also of importance, since it provides an indication of how well the algorithm is adaptively identifying the target's shape function and centroid location. The location of the centroid is necessary to center the data accurately on the FLIR plane for use in the template generation scheme discussed in Section 4.3.1.

The above statistics are calculated in both the x and y FLIR plane directions. Both error parameters are calculated before (t_i^-) and after (t_i^+) the Kalman filter update cycle. All of the errors are measured in units of pixels, where each pixel is 15 micro-radians on a side.

The above statistics are reduced even further for easily tabulated scalars versus entire time functions as indicators of performance, by temporally averaging the mean error and standard deviation time histories over the ten second simulation. In actual implementation, the statistics are averaged over the final five seconds of each simulation to ensure steady state performance is reached. These temporal averages provide a measure of comparability between various tracking scenarios studied in Chapter VI, but should be used in conjunction with the actual plots when assessing tracker performance, since some trends in the plotted time histories are not distinguishable from the time-averaged scalars. The data is presented in tabular form in Chapter VI for ease of comparison of each the tracking scenarios.

5.8 Performance Plots

Ten plots are used to assess the filter's performance in this study. They are:

1. True x position rms error vs. filter-computed x position rms error
2. True y position rms error vs. filter-computed y position rms error
3. Mean x target position error, $\pm$ one σ , plotted at all time t_i^-
4. Mean y target position error, $\pm$ one σ , plotted at all time t_i^-
5. Mean x target position error, $\pm$ one σ , plotted at all time t_i^+
6. Mean y target position error, $\pm$ one σ , plotted at all time t_i^+
7. Mean x centroid position error, $\pm$ one σ , plotted at all time t_i^-
8. Mean y centroid position error, $\pm$ one σ , plotted at all time t_i^-
9. Mean x centroid position error, $\pm$ one σ , plotted at all time t_i^+
10. Mean y centroid position error, $\pm$ one σ , plotted at all time t_i^+

Performance plots 1 and 2 indicate the adequacy of the tuning process of the filters by directly comparing the actual true rms error of the filter vs. what the filter "thinks" its error is, i.e., the filter computed rms error. Plots 3 through 6 provide primary tracking performance evaluation, because the state estimates at t_i^- are used to generate control signals to point the FLIR/laser, and the state estimates at t_i^+ provide the best possible filter estimation accuracy. As mentioned earlier, plots 7 through 10 provide information regarding the adequacy of the image centering algorithm to aid in the construction of the template. The " $\pm$ one σ " characteristics of the plots 3 through 10 are equally as important as the mean error characteristics. A tracker with large error standard deviations is ineffective in pointing a laser, since the laser energy will tend to "paint" the target, thus rendering it useless as a weapon. Examples of plots 2, 4, 6 and 10 are shown in Figures 5.5, 5.6, 5.7, and 5.8, respectively.

Toward the end of this research, it was concluded that the ten plots discussed above were not totally adequate to establish firm conclusions in regards to some of the

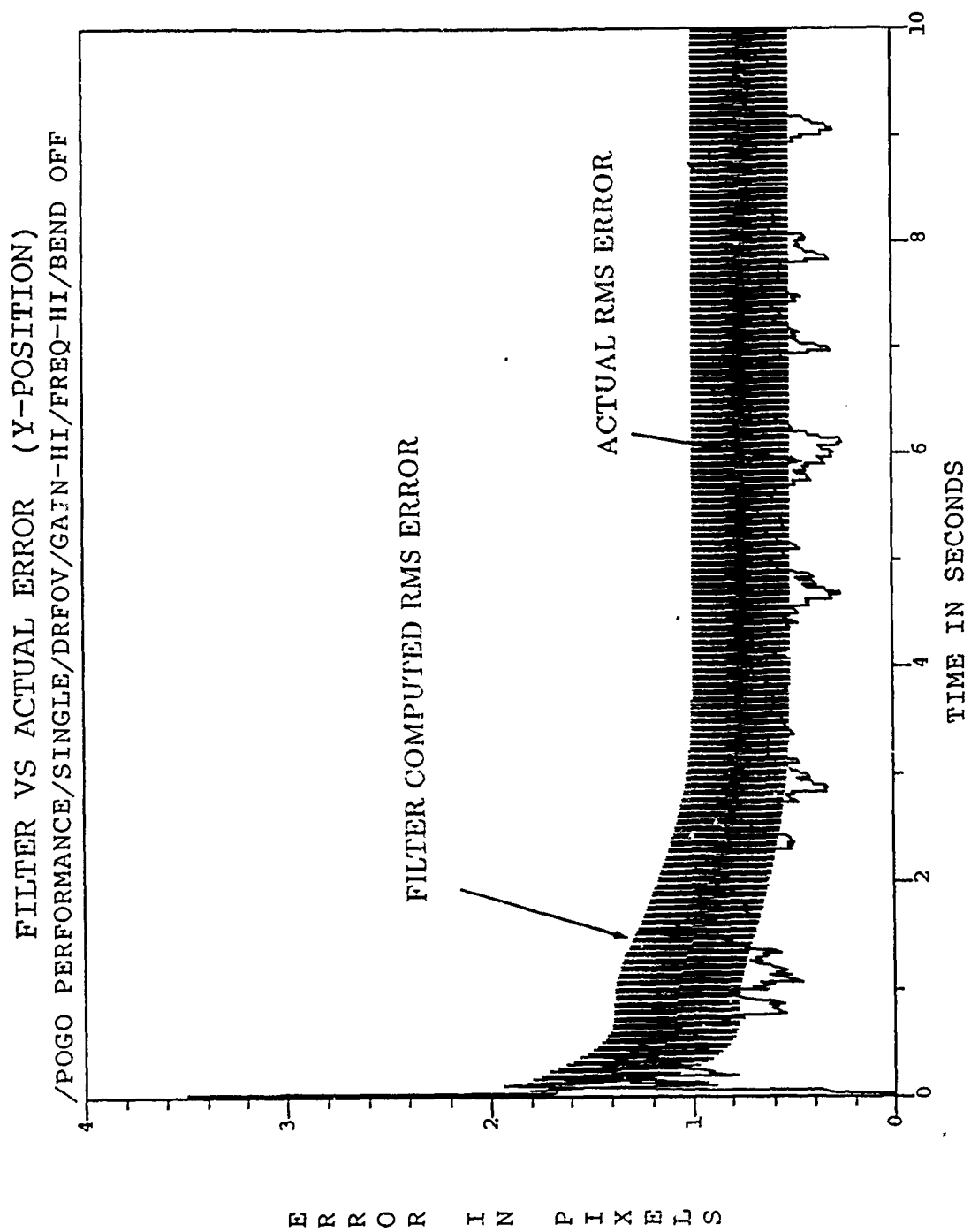


Figure 5.5. Example of Actual RMS Position Error vs. Filter Computed RMS Position Error Performance Plot (Plot #2)

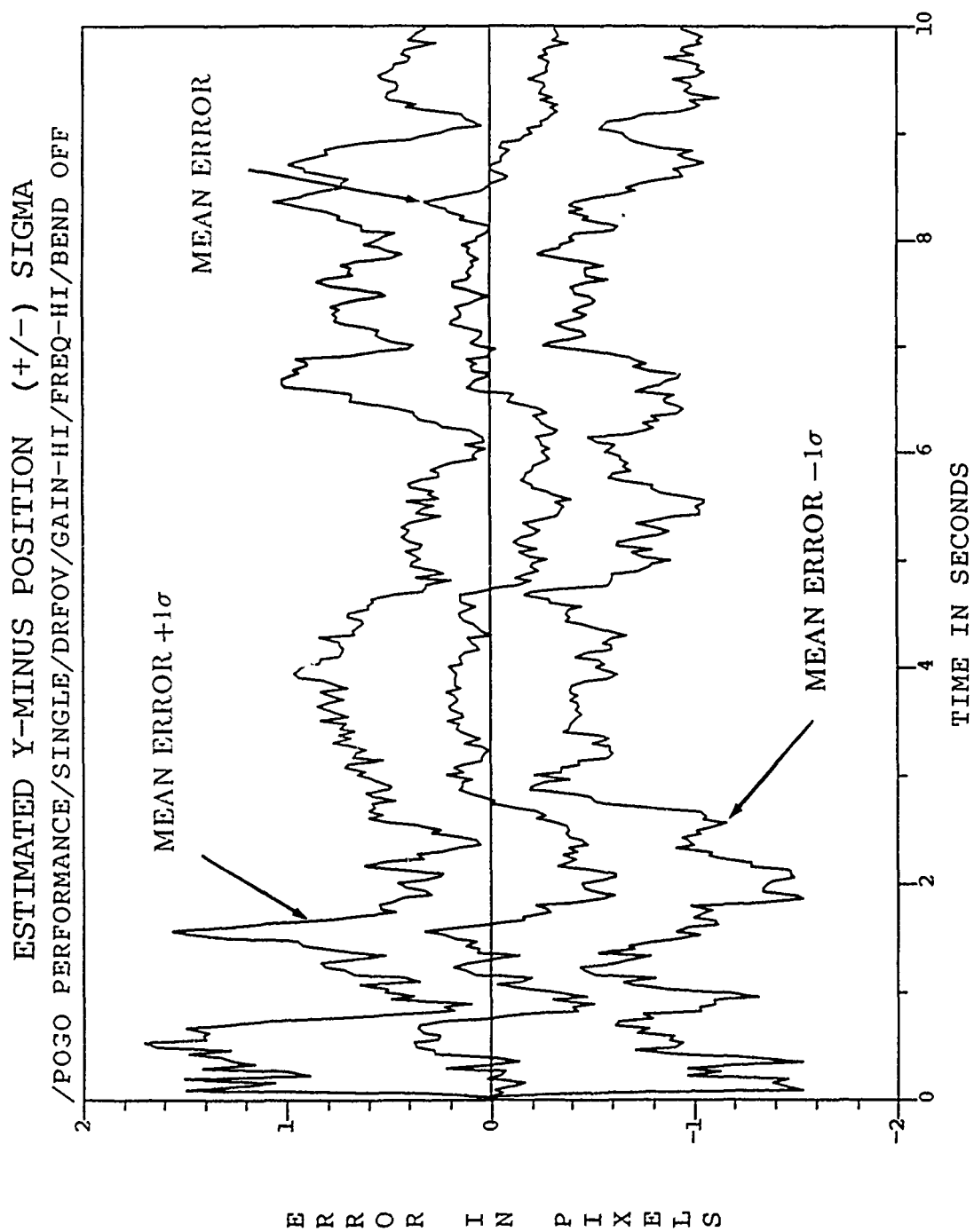


Figure 5.6. Example of Mean Position Error (Mean $\pm\sigma$ at t_i^-) Performance Plot (Plot #4)

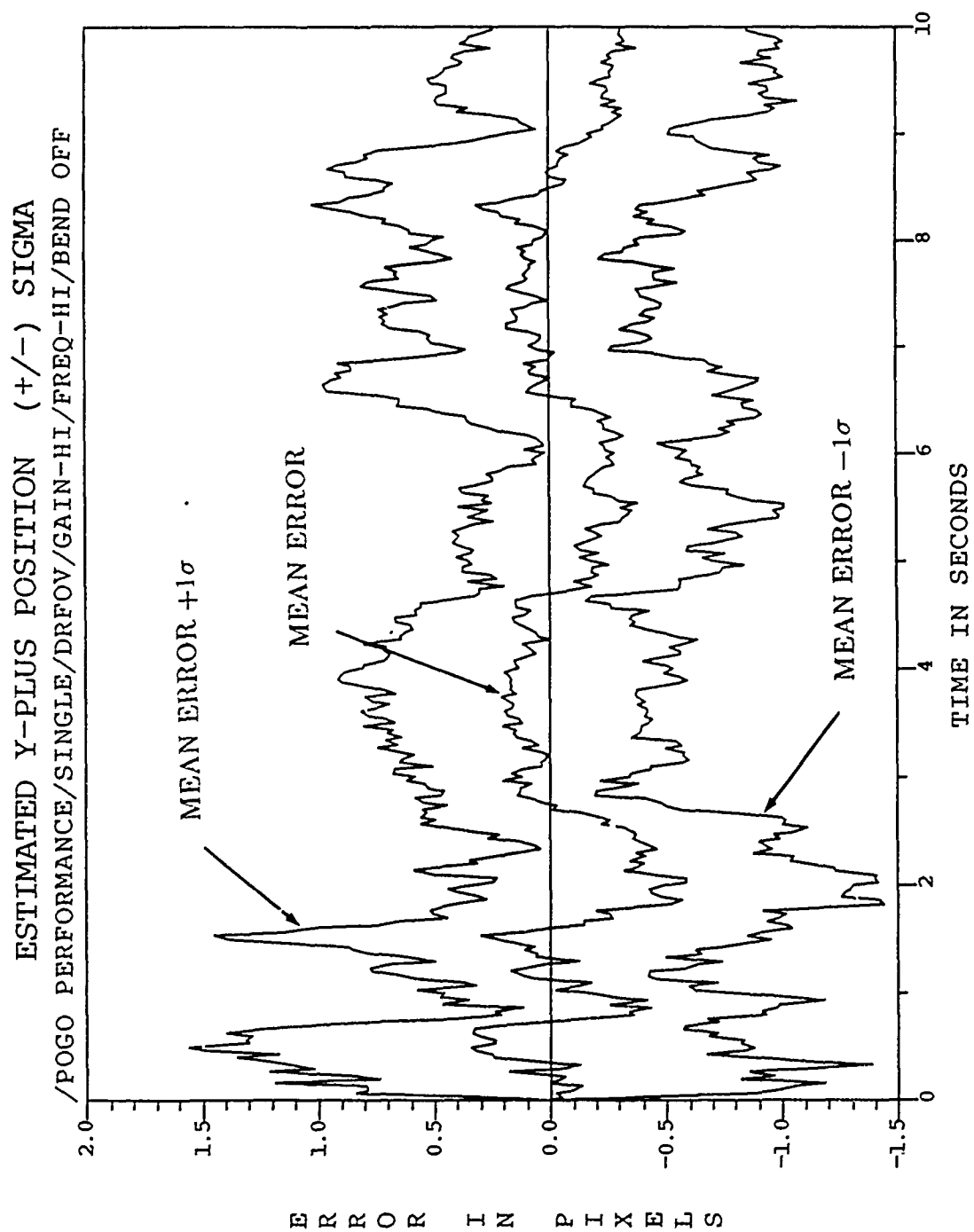


Figure 5.7. Example of Mean Position Error (Mean $\pm \sigma$ at t_i^+) Performance Plot (Plot #6)

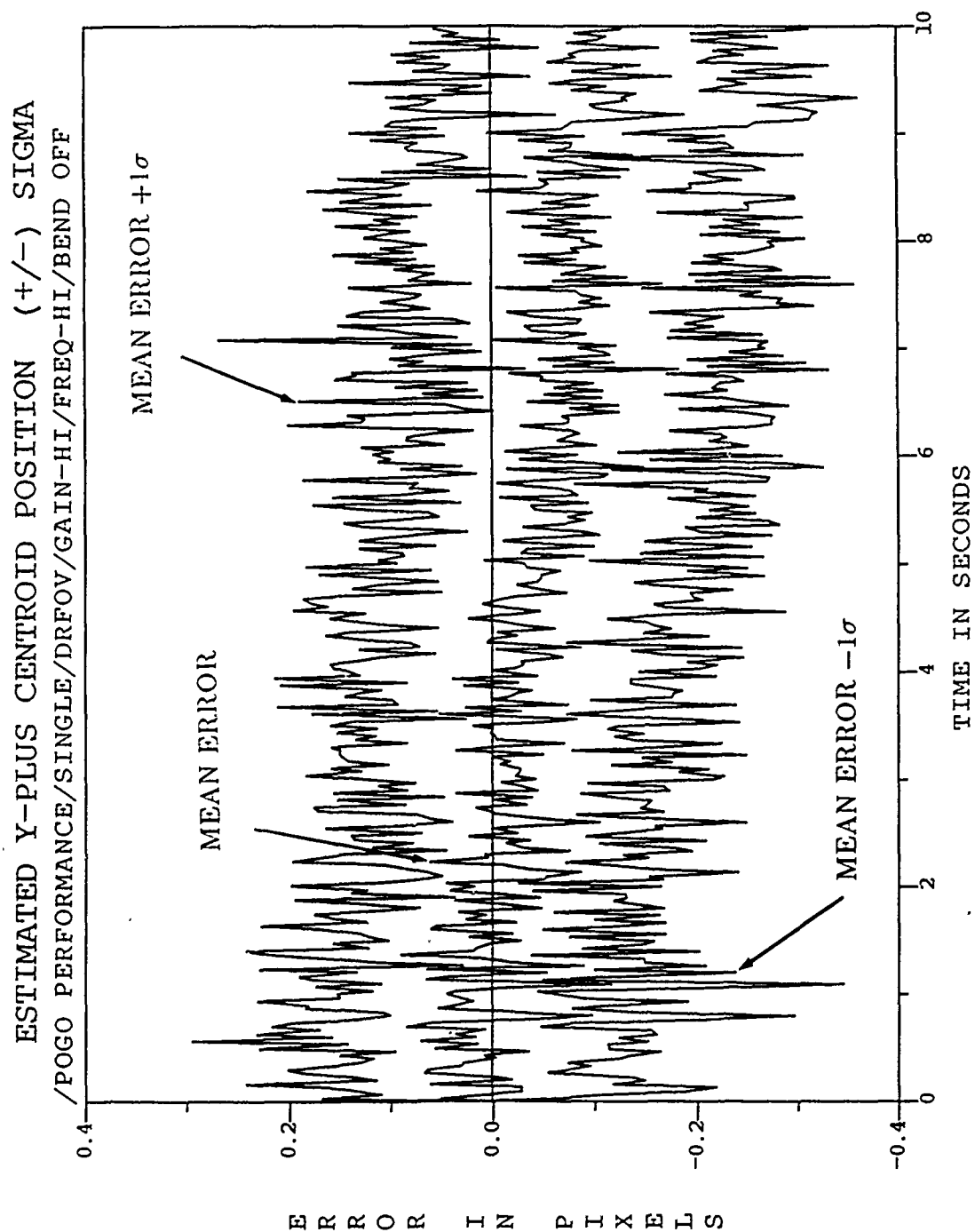


Figure 5.8. Example of Mean Centroid Position Error (Mean $\pm\sigma$ at t_i^+) Performance Plot (Plot #10)

Field	Notation	Description
field 1	BENCHMARK POGO PERFORMANCE	8-state filter analysis 10-state filter analysis
field 2	SINGLE MMAF	single filter analysis multiple model adaptive filter
field 3	NRFOV RFOV DRFOV	non-rotating field-of-view rotating field-of-view diagonal rotating field-of-view
field 4	GAIN-HI GAIN-LO	high amplitude pogo in truth model low amplitude pogo in truth model
field 5	FREQ-HI FREQ-LO	high frequency pogo in truth model low frequency pogo in truth model
field 6	POGO OFF BEND OFF	pogo turned off in truth model bending turned off in truth model

Table 5.2. Greyscale Field Descriptions

performance results during this study. Chapter VII discusses these shortcomings and provides some recommendations to overcome some of the evaluation tool limitations.

5.8.1 Plot Designation Codes. Each of performance plots is labelled with a plot designation code formatted as follows:

/field 1/field 2/field 3/field 4/field 5/field 6

where field 6 is optional and each field is explained in Table 5.2. As an example, consider the following designation code:

/BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/BEND OFF

This signifies the case of an analysis of an 8-state, single, diagonal rotating field-of-view filter where the pogo effect is a high amplitude, high frequency oscillation in the truth model, and where bending is turned off in the truth model.

5.9 Greyscales

The greyscales used in this research are representations of FLIR plane images and filter templates. In the greyscale diagram, each numerical symbol characterizes

SYMBOL	INTENSITY UNITS
0	$0 < I \leq 10$
1	$10 < I \leq 20$
2	$20 < I \leq 30$
3	$30 < I \leq 40$
4	$40 < I \leq 50$
5	$50 < I \leq 60$
6	$60 < I \leq 70$
7	$70 < I \leq 80$
8	$80 < I$

Table 5.3. Greyscale Symbol Key

a specific intensity range. The higher the number in a pixel location, the higher the intensity representation is on that pixel. Table 5.3 is the key to the numbers used in the greyscales. Since the maximum intensity of each of the hotspots is 20, intensity values greater than 20 would seem never to appear. However, the intensity units shown in Table 5.3 have been rescaled to maximize the greyscale's pictorial effect [24], and thus do not have the same meaning as the units originally used to define the 20:1 SNR.

The purpose of the greyscale diagrams is to demonstrate the adaptive identification of the target's intensity shape function in the form of a template. The greyscales are used in Section 6.4 to demonstrate the rotation schemes studied in this thesis. An example of a greyscale diagram is presented in Figure 5.9.

5.10 Summary

This chapter has presented the overall tracking algorithm used in this research by combining the principles presented Chapters II, III, and IV. An overall view of the algorithm was presented, along with a proposed structure for a MMAF algorithm. The method for field-of-view (FOV) processing was discussed, followed by the different FOV rotation schemes analyzed in this research and the relationship of the rotating FOV to the overall tracking algorithm that was presented in Chap-

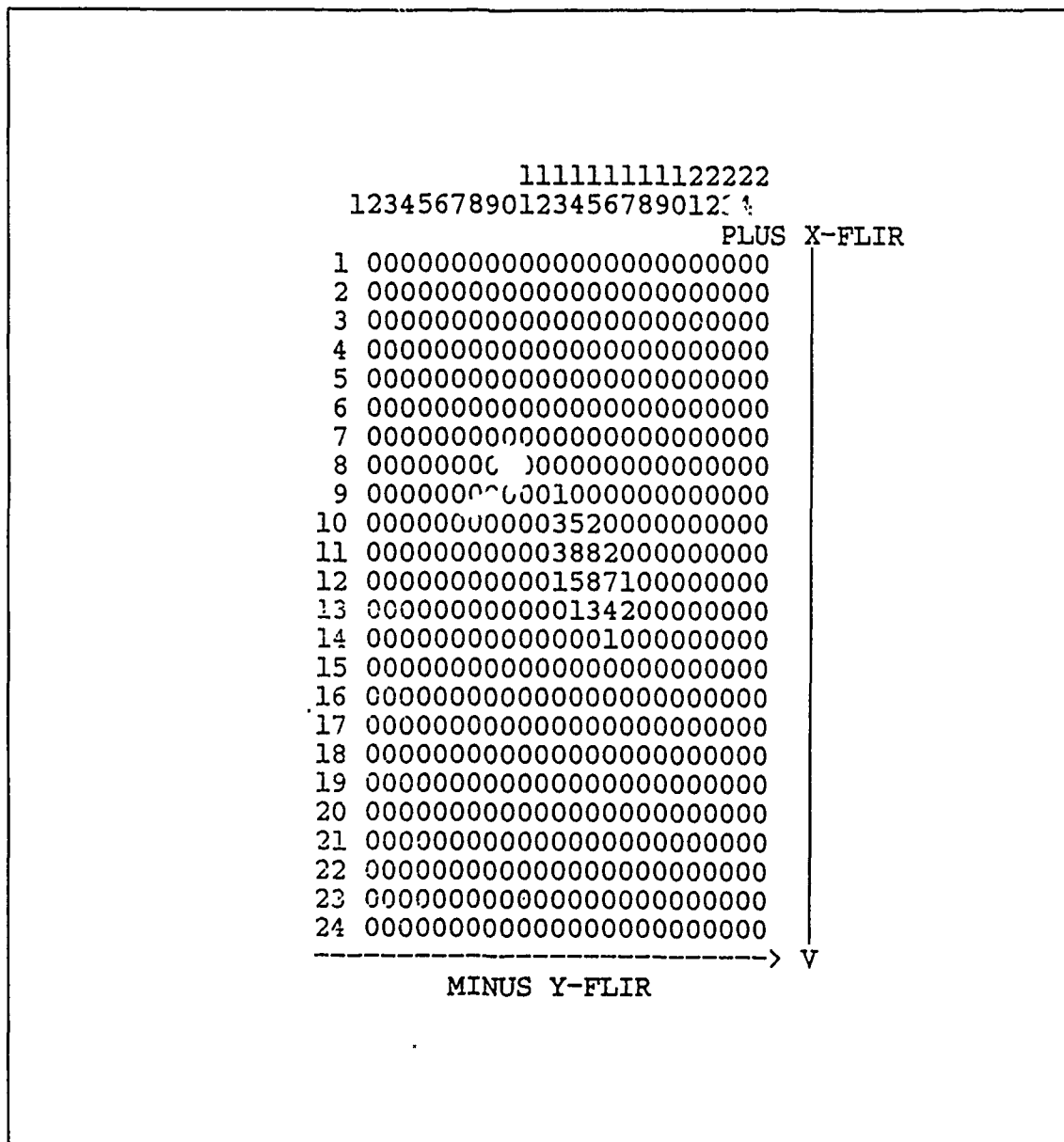


Figure 5.9. Example of a Greyscale Diagram

ter IV (Figure 4.1). The filter and truth model parameters were then presented, before concluding with the tools used to evaluate the performance of the tracking algorithm (i.e., statistical calculations, performance plot formats, and greyscale diagrams) accompanied by examples of each.

VI. Performance Analysis

6.1 Introduction

This chapter presents and analyzes the performance characteristics of the tracking algorithms discussed in Chapter V. Section 6.2 presents a tuning analysis for an 8-state filter before inserting the pogo states into the truth model. Section 6.3 then presents the performance results for four single 8-state filters, each tuned for varying pogo characteristics in the truth model. Once the performance analysis of the tuned 8-state filters is presented, the results of the FOV rotation schemes are given in Section 6.4 and are followed by the 10-state filter performance analyses in Section 6.5. Sections 6.6 and 6.7 present a rework of the 8-state and 10-state filter performance analyses, respectively, for a truth model where the bending phenomenon is removed. The reason for this rework is due to the absence of an expected performance enhancement of the 10-state filter over the 8-state filter in the preceding sections. As mentioned previously, the robustness analysis and the MMAF performance analysis were not implemented as part of this thesis research, and Section 6.8 describes why these analyses were inappropriate. The performance analysis section concludes with some trouble-shooting procedures and results in an attempt to understand the performance inconsistencies.

6.2 8-State Filter Tuning Via Dynamic Trajectory Parameters

As described in Chapter IV, the acceleration models used in the 8-state filter structure are first-order Gauss-Markov processes: the outputs of first-order lags driven by zero-mean, white Gaussian noise. The truth model target trajectory (Figure 5.4) is assumed to be a benign trajectory where no maneuvers or "jinks" are being simulated. The purpose of this section is to describe the tuning procedure and results used to tune the 8-state filter for this benign trajectory, before a plume pogo is implemented into the truth model. The tuning parameters used to tune the

FILTER #	$\sigma_x^2, \sigma_y^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_x, τ_y (seconds)	$\sigma_a^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_a (seconds)
1	250	4	0.2	0.0707
2	2000	8.5	0.2	0.0707
3	530	8.5	0.2	0.0707
4	5	8.5	0.2	0.0707
5	5	8.5	2.15	0.0707

Table 6.1. Individual Filter Tuning Parameters

8-state filter are the variances (σ_x^2, σ_y^2) and correlation times (τ_x, τ_y) of the first-order Gauss-Markov acceleration processes in each FLIR direction; and the variance (σ_a^2) and correlation time (τ_a) for the atmospheric jitter position process. By selecting various values for the variance and correlation time for the first-order acceleration process, the amplitude and rate-of-change characteristics of a variety of targets can be modelled [8:53-56]. Five separate tuning runs are described in this section, and the parameters for each of the 8-state filters are listed in Table 6.1.

Note that the tuning parameters are identical in both the x and y channels, which is characteristic of the target dynamics being modelled equally in both FLIR directions. It should also be noted that the first four filters in Table 6.1 have identical jitter characteristics. These values are based upon previous tuning results for the reduced order jitter model in the filter [18]. The reason for the change in the jitter variance for filter #5 will be explained as the analysis in this section proceeds. The performance plots for each of the five filters in Table 6.1 are found in Appendix B.

To begin the tuning of the 8-state filter for the benign ballistic trajectory, the parameters for filter #1 are chosen based upon the benign trajectory (trajectory #1) used by Leeney [5:66-67] for a small 8×8 FOV filter. Figures B.1-B.10 of Appendix B show the results of this first tuning run. Note that the filter's computed error is underestimating the actual error in both FLIR channels for the benign ballistic trajectory.

To simulate a more benign target, the correlation time of the acceleration

process was increased to 8.5 seconds [5, 10]. Also, in an attempt to tune the filter from the underestimation result shown in Figures B.1 and B.2, the acceleration variance was increased to 2000. The results of this performance run are shown in Figures B.11–B.20. The filter is still underestimating in both FLIR channels, and the actual error is larger than the corresponding plots for filter #1.

The next step in the tuning process assumed that the ballistic missile rate-of-change characteristics ($\tau_x, \tau_y = 8.5$) remain identical to the values used in filter #2. The variance was calculated to be $530 \frac{\text{pixels}^2}{\text{seconds}^4}$ based upon Equation (6.1) [7] and the tuning characteristics used by Leeney for filter #1:

$$Q = \frac{2\sigma^2}{\tau} \quad (6.1)$$

Q is the strength of the white, Gaussian noise driving the first-order lag to produce the acceleration process. Leeney's research showed that filter #1 demonstrated good tuning characteristics for a benign aircraft trajectory. In order to maintain the same value of Q that Leeney used in filter #1 (which is tuned for benign dynamics), the variances (σ_x^2, σ_y^2) of filter #3 are scaled for a benign ballistic trajectory where $\tau_x, \tau_y = 8.5$. This scaling resulted in a value of $530 \frac{\text{pixels}^2}{\text{seconds}^4}$ for the variance in each direction on the FLIR plane for filter #3. The performance plots of filter #3 are presented in Figures B.21–B.30. The tuning plots still demonstrate filter underestimation similar to the plots for filters #1 and #2.

To observe the tuning characteristics at a lower strength of pseudonoise, σ_x^2 and σ_y^2 are reduced to a value of $5 \frac{\text{pixels}^2}{\text{seconds}^4}$; and the correlation times in each direction remain unchanged at 8.5 seconds (filter #4). This model for the trajectory dynamics represents a target that shows very benign maneuverability characteristics. The amplitude parameter of the acceleration is assumed small, and the rate-of-change of the acceleration process remains slow. These target parameters produce a very small Q value, which also implies that the model is assumed to be a very accurate representation of the true target trajectory. Figures B.31–B.40 show the performance plots

for filter #4. This filter continues to show the underestimation qualities of the first three filters studied in this tuning analysis. It should be noted that the actual errors in Figures B.31 and B.32 are lower than the three preceding filters; but likewise, the filter-computed errors are also lower due to the low value of Q used. Lowering the value of Q any further could be damaging to the Kalman filter's estimation properties. By reducing Q , the filter places more emphasis on its internal model than it does the incoming measurements from the enhanced correlator. Because of this consideration, σ_x^2 and σ_y^2 are maintained at $5 \frac{\text{pixels}^2}{\text{seconds}^4}$ and tuning via the jitter variance σ_a^2 is conducted since filter tuning via the target acceleration parameters was unsuccessful. The value for the acceleration process variance was maintained at 5, because this value, along with the correlation time of 8.5 seconds, provided the best performance of all the underestimating filters studied thus far.

Table 6.1 provides the tuning parameters used in Filter #5. The performance plots are represented in Figures B.41–B.50 and demonstrate that the 8-state filter is tuned for the parameters listed in Table 6.1. Tuning via the jitter variance proved successful in tuning the 8-state filter, but the error performance plots for the dynamic states and the centroid states show a degradation compared to the performance plots for the untuned filters. This is because the value of the jitter variance (2.15) used to tune the filter is an order of magnitude greater than the jitter variance representation in the truth model (0.18). This increase in pseudonoise strength is somewhat surprising considering that previous research implemented a jitter variance of 0.2 in the filter to obtain good tuning characteristics. Section 6.9 further discusses this discrepancy between the filter and truth model representations for atmospheric jitter.

6.3 8-State Filter Benchmarks

Based upon the results in Section 6.2, the two state pogo model is added to the truth model structure, and four single filters are tuned for varying pogo

FILTER #	$\sigma_x^2, \sigma_y^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_x, τ_y (seconds)	$\sigma_a^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_a (seconds)
1	5	8.5	2.20	0.0707
2	5	8.5	2.15	0.0707
3	5	8.5	2.10	0.0707
4	5	8.5	2.15	0.0707

Table 6.2. 8-State Benchmark Filter Tuning Parameters

SCENARIO	POGO AMPLITUDE (pixels)	POGO FREQUENCY (Hertz)
1	1.12	10
2	0.012	10
3	1.12	0.1
4	0.012	0.1

Table 6.3. Truth Model Pogo Scenarios for the Four Benchmark Filters

characteristics. The parameter used in tuning the four single filters is the atmospheric jitter variance. By adjusting the jitter variance, the filters are able to capture the pogo effect as atmospheric jitter without corrupting the estimates of the dynamic states. Table 6.2 illustrates the filter parameters used to tune each of the filters for the corresponding truth model pogo scenarios in Table 6.3. Note that the tuning parameters in each of the filters in Table 6.2 are nearly identical. These results indicate that an 8-state filter is fairly robust to the varying levels of pogo amplitude and frequency (at least when the jitter variance has been increased an order of magnitude above the true jitter value), and that an MMAF structure composed of 8-state filters would not be applicable since there are no strong distinguishing characteristics for the four cases. On the other hand, an MMAF composed of 10-state elemental filters (two additional pogo states in the filter structure), which are "tightly" tuned for the varying pogo scenarios in Table 6.3, would be expected to enhance performance over both the single 8-state and single 10-state filters.

The four tuned filters in Table 6.2 establish a benchmark of performance to which other filters will be compared. Specifically, performance statistics of the 10-state filters of Section 6.5 will be compared to the following performance statistics of

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.0070759	.86775
$\hat{y}(t_i^-)$	-.070112	.74936
$\hat{x}(t_i^+)$	-.0040636	.83851
$\hat{y}(t_i^+)$	-.071775	.72038
$\hat{x}_c(t_i^-)$	.013944	1.2062
$\hat{y}_c(t_i^-)$	-.0032691	1.6308
$\hat{x}_c(t_i^+)$	.030104	.24844
$\hat{y}_c(t_i^+)$	-.012197	.17827

Table 6.4. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/
Performance Statistics

the benchmarks. It should be noted that all filters are analyzed using the DRFOV rotation scheme. This decision is based on the results presented in Section 6.4 that indicate it to be the best of the three FOV rotation strategies considered. The performance statistics of all filters are temporally averaged over the last five seconds of the ten second simulation. Mean and 1σ error statistics are measured in units of pixels on the FLIR plane, and the performance statistics for filters #1, #2, #3, and #4 are given in Tables 6.4., 6.5, 6.6, and 6.7, respectively. The performance plots for each of the filters are included in Appendix C. Figures C.1-C.10 correspond to filter #1. Figures C.11-C.20 correspond to filter #2. Filters #3 and #4 relate to the plots in Figures C.21-C.30 and C.31-C.40, respectively.

Note the relative performance improvements in the statistics after the Kalman filter update cycle. In almost all cases, the mean errors after an update show some degradation in performance, but the 1σ parameters show improvement in all channels. The cause of the degradation in the mean statistics after measurement updating is not clear. The improvement in standard deviations after updates indicates a narrowing of the error envelope and is essential for accurate and effective pointing of a laser weapon at the target. The four tuned filters discussed in this section demonstrate fairly equivalent performance, with the exception of the y position of filter #3. Based upon the temporally averaged statistics, this filter has a degraded tracking

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.0015083	.88223
$\hat{y}(t_i^-)$	-.079612	.73758
$\hat{x}(t_i^+)$	.0044635	.85175
$\hat{y}(t_i^+)$	-.081503	.70990
$\hat{x}_c(t_i^-)$	.025630	.89075
$\hat{y}_c(t_i^-)$	-.020411	.81683
$\hat{x}_c(t_i^+)$	.041439	.29739
$\hat{y}_c(t_i^+)$	-.030525	.17981

Table 6.5. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/
Performance Statistics

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.072219	.89999
$\hat{y}(t_i^-)$	-.22353	.93390
$\hat{x}(t_i^+)$	.074845	.87133
$\hat{y}(t_i^+)$	-.22487	.90969
$\hat{x}_c(t_i^-)$	.012314	.88652
$\hat{y}_c(t_i^-)$	-.022698	.81777
$\hat{x}_c(t_i^+)$	.026308	.28814
$\hat{y}_c(t_i^+)$	-.029826	.18076

Table 6.6. /BENCHMARK/SINGLE/DRFOV/GAIN-H Q-LO/
Performance Statistics

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.010234	.86902
$\hat{y}(t_i^-)$	-.075643	.73763
$\hat{x}(t_i^+)$	-.0072553	.83859
$\hat{y}(t_i^+)$	-.077485	.70995
$\hat{x}_c(t_i^-)$	.013227	.87923
$\hat{y}_c(t_i^-)$	-.014780	.81560
$\hat{x}_c(t_i^+)$	.029164	.26840
$\hat{y}_c(t_i^+)$	-.024641	.17300

Table 6.7. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/
Performance Statistics

ROTATION	$\sigma_x^2, \sigma_y^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_x, τ_y (seconds)	$\sigma_a^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_a (seconds)
NRFOV	5	8.5	1.95	0.0707
RFOV	5	8.5	2.45	0.0707
DRFOV	5	8.5	2.20	0.0707

Table 6.8. Individual Filter Tuning Parameters

capability of approximately 60% when compared to the other filters. Filter #3 is having some difficulty in separating the hardbody y dynamics from the other states for a large amplitude, low frequency pogo even though it is estimating the centroid states as accurately as the other three filters. Figures C.34 and C.36 demonstrate that filter #3 is showing larger mean errors at approximately eight seconds into the simulation as compared to the y channel plots of the other three filters. The reason for this discrepancy remains unknown at this time.

6.4 Rotating Field-of-View Analysis

This section compares the two FOV rotation schemes (RFOV, DRFOV) presented in Section 5.4 along with the nominal non-rotating FOV filter. Each of the rotation schemes is tested against an identical truth model scenario so that a valid comparison can be made. The nominal ballistic trajectory is simulated with truth model pogo characteristics representing a large amplitude, high frequency pogo effect. This pogo scenario is considered because it should stress each of the FOV schemes to the limit; i.e., if a filter can show accurate tracking against the pogo parameters at the upper limit of their respective ranges, then it is assumed that the other three pogo scenarios could be tracked equally as well since they are less dynamic. Table 6.8 represents the tuning parameters used in the FOV rotation schemes. Each filter is tuned for the same truth model scenario before the performance analyses are compared.

Appendix D provides the performance plots for each of the rotation schemes tested. Figures D.1-D.10 are the plots for the NRFOV analysis, Figures D.11-D.20

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.050960	.88013
$\hat{y}(t_i^-)$	-.076976	.85183
$\hat{x}(t_i^+)$	-.047905	.75585
$\hat{y}(t_i^+)$	-.078825	.72863
$\hat{x}_c(t_i^-)$	-.029302	1.2136
$\hat{y}_c(t_i^-)$	-.011870	1.6430
$\hat{x}_c(t_i^+)$	-.012656	.26714
$\hat{y}_c(t_i^+)$	-.021950	.28773

Table 6.9. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/
Performance Statistics

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.12915	.85499
$\hat{y}(t_i^-)$	-.13041	.77807
$\hat{x}(t_i^+)$	.13198	.82450
$\hat{y}(t_i^+)$	-.13205	.74899
$\hat{x}_c(t_i^-)$	.14943	1.2134
$\hat{y}_c(t_i^-)$	-.062422	1.6376
$\hat{x}_c(t_i^+)$	.16435	.29877
$\hat{y}_c(t_i^+)$	-.07106	.24470

Table 6.10. /BENCHMARK/SINGLE/RFOV/GAIN-HI/FREQ-HI/ Performance
Statistics

represent the RFOV analysis, and Figures D.21-D.30 represent the analysis of the DRFOV. The main performance indicators are the temporally averaged statistics over the final five seconds of the ten second simulation. These statistics are tabulated similarly to the performance statistics in Section 6.3 and provide a tool for a direct comparison of the three rotation scenarios. Tables 6.9, 6.10, and 6.11 provide the temporally averaged statistics for the NRFOV, RFOV, and DRFOV, respectively.

Comparing the results of these three tables, it is obvious that the DRFOV does outperform the other two rotation schemes, as was initially expected. The DRFOV outperforms the other two in both the mean errors and the 1σ statistics, which indi-

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.0070759	.86775
$\hat{y}(t_i^-)$	-.070112	.74936
$\hat{x}(t_i^+)$	-.0040636	.83851
$\hat{y}(t_i^+)$	-.0071775	.72038
$\hat{x}_e(t_i^-)$	.013944	1.2062
$\hat{y}_e(t_i^-)$	-.0032691	1.6308
$\hat{x}_e(t_i^+)$	.030104	.24844
$\hat{y}_e(t_i^+)$	-.012197	.17827

Table 6.11. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/
Performance Statistics

cates better identification of the missile hardbody location, as well as more accurate pointing of the laser weapon. As an example, the DRFOV outperforms the NRFOV by approximately 85% in the $\hat{x}(t_i^-)$ mean estimate and approximately 2% in the standard deviation of the same estimate. By the DRFOV aligning the diagonal dimension of the 8×8 FOV with the filter estimate of the velocity vector, more FLIR intensity data is available to the enhanced correlation algorithm to generate a better estimate of the template; thus creating more accurate linear offset measurements to the Kalman filter update algorithm. This results in the better performance characteristics over the other two schemes. The RFOV tends to show the least performance benefit of the three schemes. Based upon Figure 5.2, this makes sense intuitively. As the pogo effect causes the plume to oscillate about the x axis on the FLIR plane, the plume actually "pogos" out of the FOV of the FLIR. Less intensity data is available to the enhanced correlation algorithm; thus the degraded performance statistics for the RFOV. The NRFOV does demonstrate good performance, which is based upon the 60° velocity orientation angle of the chosen missile trajectory (at 0° , 90° , etc., it would show the same poorer performance of the RFOV).

As mentioned in Chapter V, the greyscale diagram is a plot which demonstrates the size and shape characteristics of an image on the FLIR plane. Figures 6.1, 6.2, and 6.3 are greyscale diagrams of the ballistic missile plume and represent a NRFOV,

RFOV, and DRFOV, respectively. Note that the difference in the plume orientation is difficult to distinguish between the NRFOV and DRFOV, although Figure 6.3 does demonstrate a more "diagonal-like" orientation. A change of the greyscale's rectangular representation to a "true" square-shaped grid would demonstrate that the DRFOV shows a more prominent alignment of the image semi-major axis with the FOV diagonal. The RFOV of Figure 6.2 does demonstrate that the semimajor axis is aligned with the positive FLIR x , axis as was described in Figure 5.2.

6.5 10-State Filter Performance Analysis

The objective of this section is to demonstrate the performance of four 10-state Kalman filters paralleling the truth model scenarios in Table 6.3. The performance results of each of the four Kalman filters are tabulated, and the results are compared to the four 8-state filters of Section 6.3. Again, as in the previous sections, the filters use the DRFOV rotation scheme as described in Section 6.4. The 10-state Kalman filters used in this study are of the structure presented in Chapter IV, where the pogo model is represented by two additional filter states and corresponds directly to the model used in the truth model. The reason for the identical pogo structures in the filter and truth models is to provide the filter with the actual effects of pogo being simulated (robustness to mismodelling of the pogo phenomenon would be a natural follow-on investigation). It was decided to handle any mistuning via the filter jitter state, as was done in Section 6.3. By using an identical pogo representation in the filter model and truth model, an increase in performance is expected over the 8-state filter where the pogo states were not modelled. Although this tends to make the 10-state filters less robust to varying pogo scenarios (see Table 3.1), it leads perfectly into an MMAF structure that will additionally provide performance enhancement over both a single 10-state and 8-state filter. The tuning parameters for each of the 10-state filters are presented in Table 6.12; and the performance plots for filters #1, #2, #3, and #4 are included in Figures E.1-E.10, E.11-E.20,

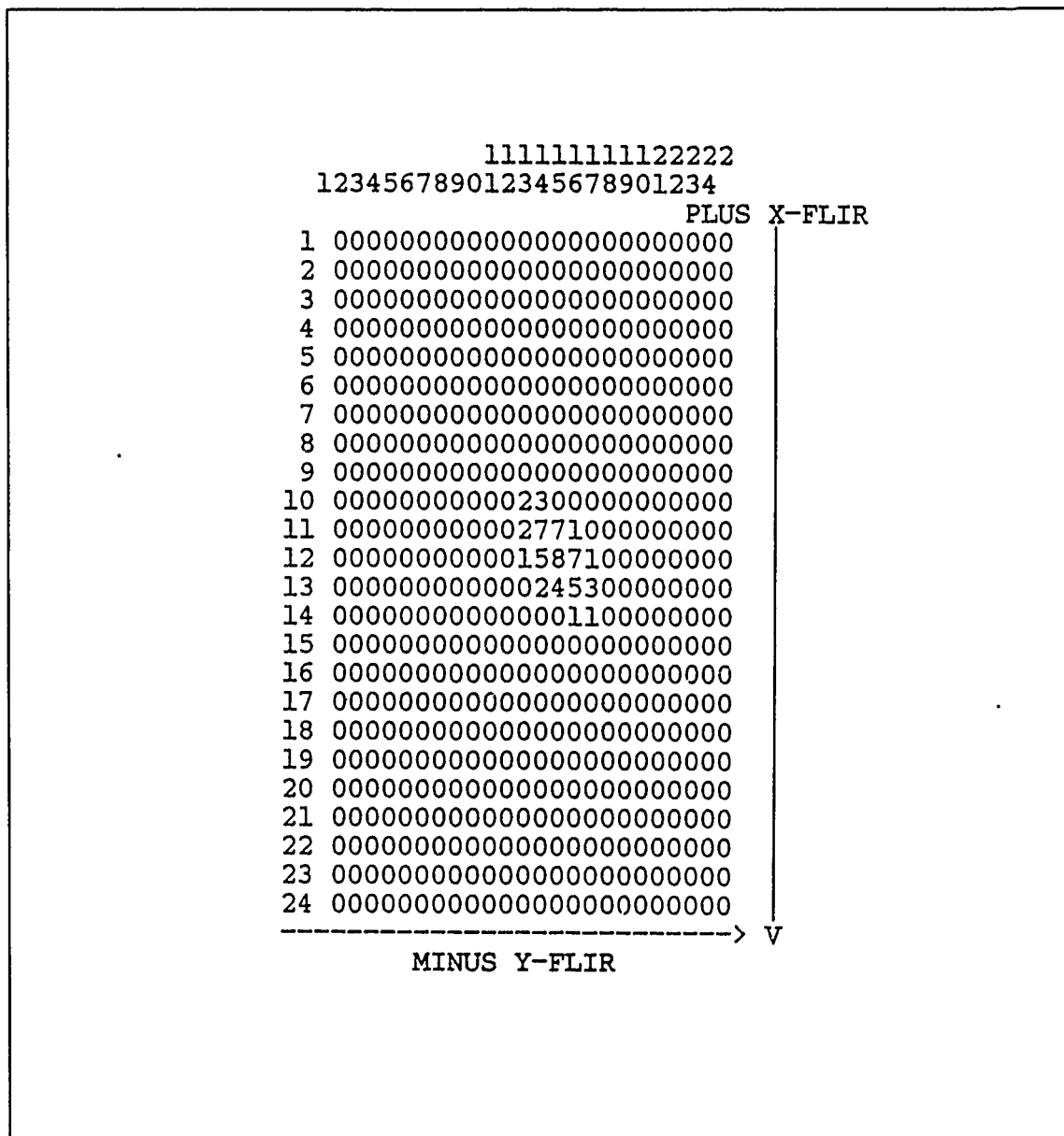


Figure 6.1. Greyscale Diagram of a NRFOV

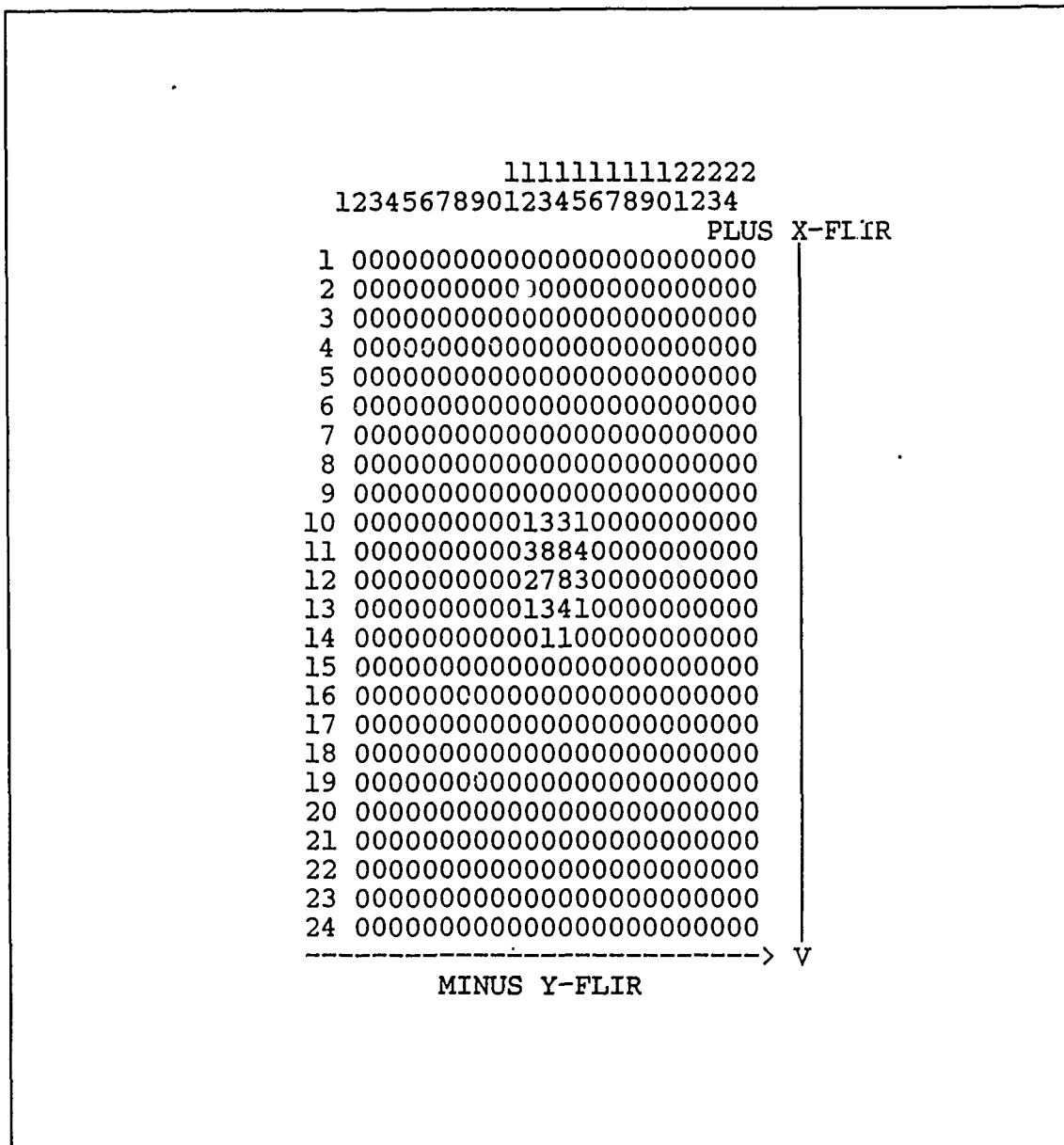


Figure 6.2. Greyscale Diagram of a RFOV

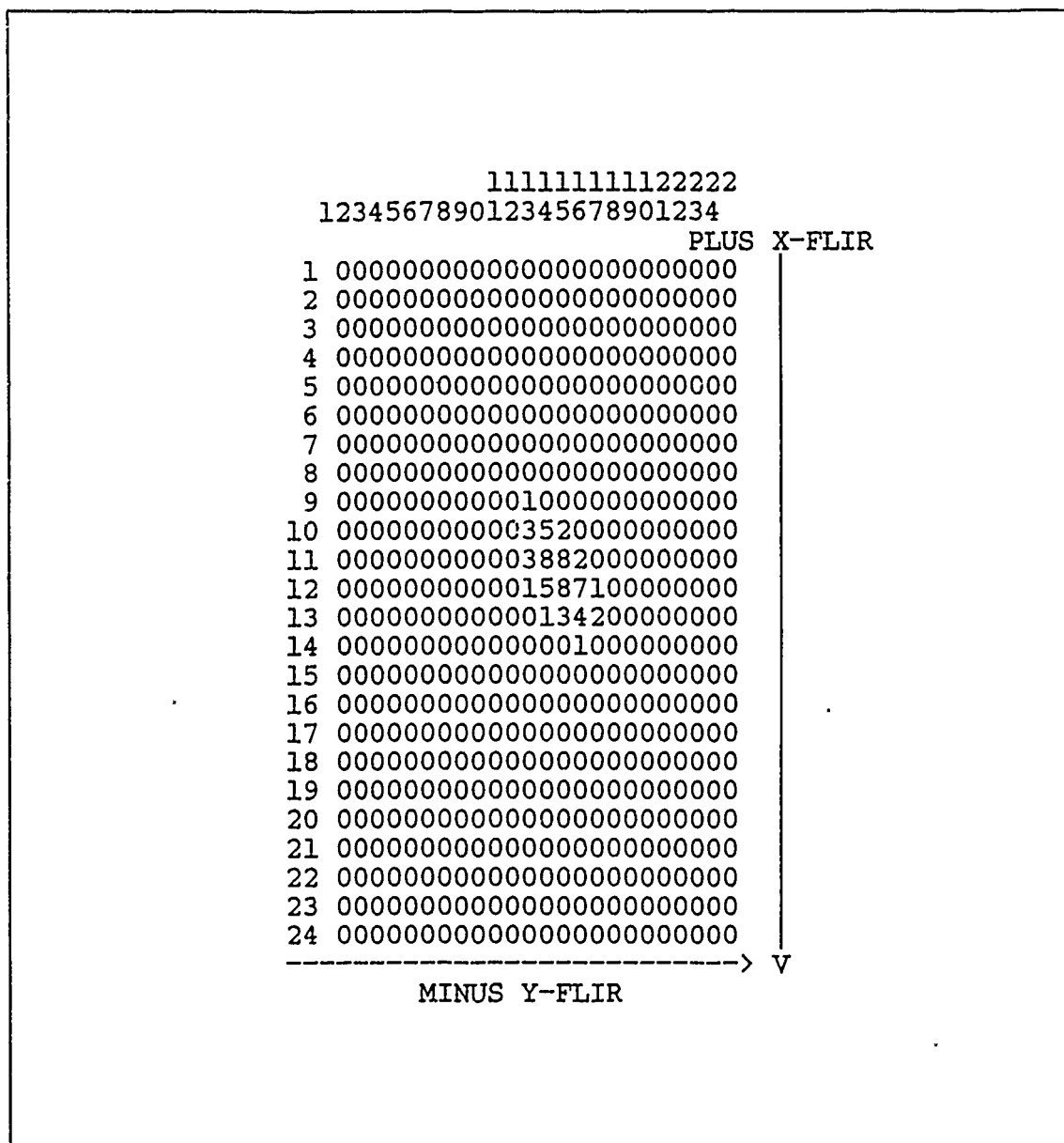


Figure 6.3. Greyscale Diagram of a DRFOV

FILTER #	$\sigma_x^2, \sigma_y^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_x, τ_y (seconds)	$\sigma_a^2 (\frac{\text{pixels}^2}{\text{seconds}^4})$	τ_a (seconds)
1	5	8.5	2.20	0.0707
2	5	8.5	2.15	0.0707
3	5	8.5	1.70	0.0707
4	5	8.5	2.05	0.0707

Table 6.12. 10-State Pogo Performance Filter Tuning Parameters

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.011699	.86265
$\hat{y}(t_i^-)$	-.068184	.74331
$\hat{x}(t_i^+)$	-.0086868	.83259
$\hat{y}(t_i^+)$	-.0069922	.71512
$\hat{x}_c(t_i^-)$	.0073344	1.0123
$\hat{y}_c(t_i^-)$	.0034861	1.1889
$\hat{x}_c(t_i^+)$	.026560	.21228
$\hat{y}_c(t_i^+)$	.0098915	.15534

Table 6.13. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/
Performance Statistics

E.21–E.30, and E.31–E.40 of Appendix E, respectively.

Note that, when comparing the tuning parameters of Table 6.12 and Table 6.2, filters #1 and #2 do not change tuning characteristics when going from an 8-state to a 10-state model. Filters #3 and #4 do show a difference in the tuning characteristics for the 8-state to 10-state models. Filters #3 and #4 are tuned for a low pogo frequency while filters #1 and #2 are tuned for a high pogo frequency. This suggests that the 10-state filters are sensitive to a variation in the pogo frequency, which should be noticeable in a robustness study with the 10-state filters. This sensitivity to frequency changes provides insights into possible elemental filters necessary in the MMAF algorithm, but no firm conclusions can be made until a complete robustness study is performed (see Section 6.8). Tables 6.13, 6.14, 6.15, and 6.16 present the temporally averaged performance results for filters #1, #2, #3, and #4, respectively.

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.00099137	.87757
$\hat{y}(t_i^-)$	-.084805	.73927
$\hat{x}(t_i^+)$	.0029912	.84687
$\hat{y}(t_i^+)$	-.086673	.71175
$\hat{x}_c(t_i^-)$	.024272	.88594
$\hat{y}_c(t_i^-)$	-.025499	.81603
$\hat{x}_c(t_i^+)$	.040221	.28125
$\hat{y}_c(t_i^+)$	-.035495	.17773

Table 6.14. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-HI/
Performance Statistics

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.089741	.94506
$\hat{y}(t_i^-)$	-.28040	.99732
$\hat{x}(t_i^+)$	.092084	.91402
$\hat{y}(t_i^+)$	-.28132	.97042
$\hat{x}_c(t_i^-)$	.015006	.88221
$\hat{y}_c(t_i^-)$	-.055326	.81552
$\hat{x}_c(t_i^+)$	.026931	.27572
$\hat{y}_c(t_i^+)$	-.059738	.16126

Table 6.15. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/
Performance Statistics

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.0037107	.89255
$\hat{y}(t_i^-)$	-.10924	.76164
$\hat{x}(t_i^+)$	-.00083007	.86142
$\hat{y}(t_i^+)$	-.11088	.73316
$\hat{x}_c(t_i^-)$	.0096192	.88558
$\hat{y}_c(t_i^-)$	-.031412	.81304
$\hat{x}_c(t_i^+)$	.024948	.28201
$\hat{y}_c(t_i^+)$	-.040174	.16130

Table 6.16. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO/
Performance Statistics

In comparing Tables 6.13-6.16 to the corresponding tables in Section 6.3, the filters of this section show a mixture of performance improvement and degradation, with no real pattern. There is also a mixed performance comparison between the x and y channels of the same filter. The performance improvements cannot be tied to a particular channel or a particular pogo parameter. For example, in comparing the statistics of the 10-state filter #1 to the 8-state filter #1, the y position channel means demonstrate a performance enhancement for the 10-state filter of about 3% with a 1σ improvement of approximately 1%, while the x position channel means demonstrate a performance degradation of 65 to 100% and a 1σ enhancement of about 1%. Likewise, in comparing the 10-state filter #4 to the 8-state filter #4, the x channel position means demonstrate an enhanced performance of 60 to 90%, while the y channel position means show a degraded performance on the order of 45%. These mixed performance characteristics are consistent when comparing each of the filter statistics in this section to those of Section 6.3. Thus, due to the lack of consistent performance enhancement of the 10-state filter, a robustness analysis for possible MMAF implementation of the 10-state elemental filters was not feasible at this time. At this point, the research therefore altered direction in an attempt to understand the lack of consistent 10-state filter performance enhancement.

After ensuring that the pogo phenomenon was correctly derived (Section 3.2.4) and implemented in the Fortran code, a possible problem for the lack of consistent performance enhancement was attributed to a possible observability problem caused by interaction between the pogo effect and the bending/vibration phenomena modelled in the truth model [10] if the bandwidths of the two processes were too close to one another. The undamped natural frequency of the bending phenomenon is set at one Hertz throughout the simulation, and a possible resonance effect with the pogo phenomenon's frequencies (0.1 and 10 Hertz) may cause the filter difficulty in distinguishing between an optical bending phenomenon and the plume's pogo phenomenon. The next two sections are an attempt to substantiate this claim.

Section 6.6 tunes an 8-state filter for a large amplitude, high frequency truth model pogo phenomenon where all bending effects are removed from the truth model. Section 6.7 tunes the corresponding 10-state filter for the same truth model scenario, and the performance statistics of the 8-state and 10-state filters are once again compared. This process parallels the performance comparisons between the filters in Sections 6.3 and 6.5; but as a start, only one truth model scenario (large amplitude, high frequency pogo) is being implemented. By removing the bending effects, the intent is to isolate the ability to estimate pogo effects (adaptively) in the current filter structure, separating it from a physical phenomenon that might cause observability or distinguishability difficulties. It should be noted that the Fortran code used in this analysis has been modified to include two "flags" that can automatically turn the pogo effect and/or the bending phenomenon on or off in the truth model. This capability was extremely helpful in the trouble-shooting scheme presented in this section, in addition to the trouble-shooting discussed in Section 6.9.

6.6 8-State Filter Benchmark with Bending Removed from the Truth Model

As mentioned in the previous section, the performance analysis of an 8-state filter for a large amplitude, high frequency truth model pogo effect with the bending states removed is presented in this section. The tuning parameters used are identical to the parameters of filter #1 in Table 6.2, with one exception. The atmospheric jitter variance necessary to tune the 8-state filter of this section is lowered to $0.85 \frac{\text{pixels}^2}{\text{seconds}^4}$ as compared to the 2.20 value in Table 6.2. This lowering of the jitter variance makes sense intuitively, since the bending effect was removed from the truth model; thus the filter does not have to use the additional pseudonoise to represent this unmodelled effect. This result was encouraging in the sense that it demonstrates that the bending/vibration phenomenon has a large impact on the tuning of the filter and could possibly be interfering with the pogo phenomenon.

Figures F.1-F.10 of Appendix F show the performance plots for the single 8-

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.0013045	.52118
$\hat{y}(t_i^-)$	-.067579	.63134
$\hat{x}(t_i^+)$	.00028606	.50176
$\hat{y}(t_i^+)$	-.067861	.60457
$\hat{x}_c(t_i^-)$	.015861	1.1899
$\hat{y}_c(t_i^-)$	-.052311	1.6195
$\hat{x}_c(t_i^+)$	.0083355	.27029
$\hat{y}_c(t_i^+)$	-.053648	.17155

Table 6.17. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/BEND
OFF/ Performance Statistics

state filter used in this section. The temporally averaged performance statistics are provided in Table 6.17.

Comparing the performance results of the 8-state filter of this section with the statistics of Table 6.4, shows a dramatic increase in x and y position performance for the filter where the bending is removed from the truth model. This is expected since the filter of this section does not have to concern itself with separating the bending effects from the position states in the truth model. The next section provides the performance results of the 10-state filter, in addition to a comparison of the 8-state filter statistics of this section.

6.7 10-State Filter Performance with Bending Removed from the Truth Model

As previously mentioned, an identical analysis of a 10-state filter is conducted to parallel the analysis in the previous section for an 8-state filter. The 10-state filter of this section shows the same tuning characteristics as filter #1 of Section 6.5, with the exception of the atmospheric jitter variance. The jitter variance required to tune this 10-state filter is $0.7 \frac{\text{pixels}^2}{\text{seconds}^4}$ as compared to the jitter variance of 2.20 for filter #1 of Section 6.5. This lowering of the jitter variance is consistent with the corresponding decrease for the 8-state filter of the previous section. The performance

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.018256	.52291
$\hat{y}(t_i^-)$	-.095694	.61774
$\hat{x}(t_i^+)$	-.016663	.50233
$\hat{y}(t_i^+)$	-.095728	.59076
$\hat{x}_c(t_i^-)$	-.033336	.98328
$\hat{y}_c(t_i^-)$	-.076664	1.1528
$\hat{x}_c(t_i^+)$	-.024408	.25916
$\hat{y}_c(t_i^+)$	-.078125	.15941

Table 6.18. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/
BEND OFF/ Performance Statistics

plots for the 10-state filter of this section are presented in Figures F.11–F.20 of Appendix F. The corresponding performance statistics temporally averaged over the last five seconds of the ten second simulation are presented in Table 6.18.

In comparing the results of Table 6.18 to those of Table 6.17, the mean position errors of all the variables of interest are showing a degradation in performance by the 10-state filter. The 1σ statistics show a mix in performance enhancement and degradation; although overall, the 10-state filter performance consistently shows a performance degradation when considering the worst case tracking scenario, i.e., mean error $+1\sigma$ or mean error -1σ . For example, consider the mean errors of $\hat{y}_c(t_i^+)$ for the 10-state filter of this section to the 8-state filter of the previous section. The 8-state filter demonstrates an enhanced estimation ability over the 10-state filter. But, upon inspection of the respective 1σ statistics, the 10-state filter demonstrates better estimation accuracy over the 8-state filter other than the larger bias just noted. When comparing the worst case scenario as described above, the 8-state filter maintains a performance increase of approximately 5% over the 10-state filter's estimation ability.

The results of the degraded 10-state filter performance are similar to the results presented in Section 6.5; therefore it was deemed unnecessary to compare the 10-

state and 8-state filter performance for the remaining truth model scenarios listed in Table 6.3. From the results of the analysis in this section, it can be concluded that, although the bending phenomenon does have an impact on the tuning characteristics of the 8-state and 10-state filters, it is not the reason for the degraded performance of the 10-state filter. Bending may have some effect on the degraded performance, but something else involving the filter structure seems to be more dominant in degrading the 10-state filter performance from that of the 8-state filter. Based on the results of this analysis, the 10-state filter is showing a decrease in overall performance for every statistic. Since it is expected that the 10-state filter should show a corresponding increase in performance rather than the observed decrease, a possible sign error in the Fortran code was suspected [10]. Section 6.9 discusses the analysis and results of additional trouble-shooting techniques (including possible sign errors in the code) in attempt to explain the degraded 10-state filter performance. Before getting into the trouble-shooting techniques of Section 6.9, Section 6.8 summarizes the "obvious" reasons why the MMAF and robustness analyses were not performed as part of this research.

6.8 Robustness and MMAF Discussions

As stated throughout this thesis, the robustness and MMAF analyses were not implemented as part of this research. Based upon the results of Chapter VI thus far, the overriding reason that the robustness study is not performed is due to the lack of performance enhancement of the 10-state Kalman filters over the corresponding 8-state filters. Without an increase in performance of the 10-state filter, a robustness analysis on a poorer quality filter does not make sense. In addition, the implementation of the MMAF algorithm is directly tied to the results of the robustness analysis, and since the robustness analysis is inappropriate at this time, the MMAF analysis suffers correspondingly. At this point in the research, a single robust 8-state filter outperforms the 10-state filters, and one would likewise expect it to outperform an

MMAF structure composed of inferior quality 10-state elemental filters.

Once the problem of the 10-state filter performance degradation is corrected, a robustness study can determine what elemental filters are appropriate in the MMAF structure. A proposed MMAF structure is provided in Chapter IV. Therefore, for completeness of the objectives described in Chapter I, and for the benefit of follow-on research to implement an MMAF structure, a proposed robustness study for each of the *tuned* single 10-state Kalman filters presented in Section 6.5 is provided in Table 6.19. Note that each of the four *tuned* filters of Section 6.5 are tested against four truth model pogo robustness scenarios that vary in combinations of pogo parameters for which they are not tuned. The results of this analysis will determine the sensitivity of each of the tuned filters to a mismatch in what the filter "thinks" the pogo parameters are, to what the truth model is actually simulating. This would indicate relative merits of including either pogo amplitude, or pogo frequency, or both, in the adaptation of the MMAF algorithm. Also note that a robustness scenario involving pogo amplitude and frequency values corresponding to the median value of the assumed range of pogo parameter values has been added to the analysis. The intent is to determine whether an additional 10-state filter, tuned for these median pogo parameters, is required in the MMAF structure.

6.9 *Trouble-Shooting*

The objective of this section is to determine the possible reasons why the 10-state filters developed in this research do not outperform the corresponding 8-state filters. This section is divided into four subsections which address the analysis and results of (1) possible sign errors in the pogo model, (2) jitter model sign errors, (3) observability issues of the 8-state and 10-state filters, and (4) possible pogo-jitter interactions.

FILTER #	TRUTH MODEL POGO AMPLITUDE (pixels)	TRUTH MODEL POGO FREQUENCY (Hertz)
1	0.112	1
	0.0112	10
	1.12	0.1
	0.0112	0.1
2	1.12	10
	0.112	1
	1.12	0.1
	0.0112	0.1
3	1.12	10
	0.112	1
	0.0112	10
	0.0112	0.1
4	1.12	10
	0.112	1
	1.12	0.1
	0.0112	10

Table 6.19. Proposed 10-State Filter Robustness Scenarios

6.9.1 Analysis of Possible Pogo Sign Errors in the Fortran Code. Based upon the discussion in Section 6.7, the results indicate that a sign error may exist in the Fortran code. This section describes the trouble-shooting performed in determining a possible sign error in the pogo implementation. Two separate checks were implemented to detect such a pogo sign error. The first check involved an individual frame analysis of three separate simulations. All three simulations involved the atmospheric jitter states being removed from the filter and truth models, along with removing the bending states in the truth model. The only states remaining in the filter and truth models were the target dynamic states and the pogo states. The differences in the three simulations are described as follows:

1. In the first simulation, the pogo output of the Kalman filter propagation cycle and the pogo output of the truth model propagation cycle are hard-coded to equal the value 2. The error analysis in this simulation should demonstrate the best performance if the truth model simulation, tracker algorithm, and errors

SIMULATION #	$\hat{x}(t_i^-)$	$\hat{y}(t_i^-)$	$\hat{x}_c(t_i^-)$	$\hat{y}_c(t_i^-)$
1	-.000572681	.000602245	-1.33138	-.642759
2	-.000572681	.000602245	-3.3341	2.8192
3	-.000572681	.000602245	-1.58063	-.211713
SIMULATION #	$\hat{x}(t_i^+)$	$\hat{y}(t_i^+)$	$\hat{x}_c(t_i^+)$	$\hat{y}_c(t_i^+)$
1	1.25099	.808904	.0734272	.264287
2	3.28608	-2.34386	.0848646	.267146
3	1.59888	.395761	.0713057	.190000

Table 6.20. Frame Analysis of Mean Errors for Pogo Sign Test

are being calculated correctly.

2. In the second simulation, the pogo output of the filter propagation was hard-coded to -2, while the truth model remained hardcoded at 2. The error analysis in this simulation should show the worst performance if the truth model simulation, tracker algorithm, and errors are being calculated correctly.
3. The third simulation involved no hardcoding of the filter or truth states. This simulation is added as a control to ensure that nothing completely unexpected happens in simulations #1 and #2.

The intent is to compare the mean errors after the first propagation and update cycles to determine if the variables and errors are being calculated correctly. Table 6.20 presents the mean errors in the filter estimates after the first Kalman filter propagation and update cycle.

Note the error magnitudes in the $\hat{x}_c(t_i^-)$ and $\hat{y}_c(t_i^-)$ centroid estimate channels for simulations #1 and #2. The relative magnitudes of these errors indicate that the pogo signs are implemented correctly in the error calculations of the simulation software. Simulation #2 hard-coded the value of the pogo filter state after a propagation cycle at -2, whereas the corresponding truth model value of the state was hard-coded at 2. Likewise, the pogo filter output of simulation #1 was hard-coded at 2 after a propagation cycle and 2 for the output of the truth model pogo. Since the error calculations coded in the software take the difference between the filter state minus

the truth state, one might expect an error difference of -4 for the centroid errors between simulations #1 and #2. But recalling that the pogo phenomenon is simulated to occur about the estimated velocity vector, the differences between the centroid errors in simulations #1 and #2 are on the order of $-4 \cos \theta_T$ for the x channel centroid errors and $-4 \sin \theta_T$ for the y channel centroid errors, where $\theta_T = 60^\circ$ for the chosen missile trajectory. These results signify that the pogo model is implemented properly and that the errors are being calculated correctly.

In addition, the errors for simulation #2 show the worst performance of the three simulations tested. This is additional evidence that the errors are being calculated correctly and the pogo models are implemented as expected. The errors for simulation #1 closely resemble those of the control simulation #3. This also indicates that the nominal pogo model of simulation #3 is estimating as designed. Evidence that the errors after a Kalman filter update cycle are being calculated correctly is also contained in Table 6.20. Simulations #1 and #3 demonstrate small errors after the Kalman filter update cycle for this particular data frame analyzed. As the simulations continued, the Kalman filter attempted to regulate all of the errors to zero, as would be expected.

The second check to test for pogo sign errors involves two simulations based upon the same concept presented above. The first simulation involves hardcoding the output pogo state from the Kalman filter propagation cycle to equal the output of the truth model's pogo state. The second simulation involves hardcoding the output pogo state from the Kalman filter propagation cycle to equal the negative of the output of the truth model's pogo state. The only difference between the set-up for this check and the previous check is that the pogo representation in the truth model is permitted to propagate as designed, instead of hardcoding it to a value of 2 as was previously done. By allowing the pogo representation in the truth model to propagate naturally, the Kalman filter update cycle is also permitted to react more naturally, since the estimate of the pogo state after the propagation cycle is

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.024647	.12071
$\hat{y}(t_i^-)$	.13471	.21716
$\hat{x}(t_i^+)$	-.022972	.11548
$\hat{y}(t_i^+)$	.13203	.21102
$\hat{x}_c(t_i^-)$	-.025102	.58298
$\hat{y}_c(t_i^-)$	.11998	.99070
$\hat{x}_c(t_i^+)$	-.018504	.22206
$\hat{y}_c(t_i^+)$	.10575	.38650

Table 6.21. Performance Statistics for Simulation #1 of Pogo Sign Test

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.11219	.14661
$\hat{y}(t_i^-)$	-.14732	.22869
$\hat{x}(t_i^+)$	-.10969	.14364
$\hat{y}(t_i^+)$	-.14934	.22409
$\hat{x}_c(t_i^-)$	-.10350	.95367
$\hat{y}_c(t_i^-)$	-.14650	1.6198
$\hat{x}_c(t_i^+)$	-.094226	.36817
$\hat{y}_c(t_i^+)$	-.15686	.61230

Table 6.22. Performance Statistics for Simulation #2 of Pogo Sign Test

not continuously being hardcoded to the same value. The bending states are once again removed from the truth model, and the atmospheric jitter states are removed from both the filter and truth models. The temporally averaged results of ten Monte Carlo runs are collected for each of these simulations and are presented in Tables 6.21 and 6.22.

Based upon a comparison of Tables 6.21 and 6.22, it is obvious that the results from simulation #1 show definite performance improvement over the results of simulation #2 for all statistics collected. This is further evidence that the pogo model is acting correctly in both the truth and filter models. It can be surmised that from the results of this section, the degraded performance of the 10-state filters presented in Section 6.5 is not due to a problem with pogo effect implementation in

the simulation software. The next section performs the same analysis to check for possible sign errors in the atmospheric jitter implementation.

6.9.2 Analysis of Possible Atmospheric Jitter Sign Errors in the Fortran Code. A similar sign analysis to the one from the previous section is conducted for the atmospheric jitter states. The first check involves three simulations in which the pogo states are removed from the filter and truth models. In addition, the bending effects are also removed from the truth model. This first check is similar to the one in the preceding section, in that the mean error statistics are observed over a data frame that includes one filter propagation and update cycle. The three simulations are described as follows:

1. In the first simulation, the jitter output of the Kalman filter propagation cycle and the jitter *outputs* of the truth model propagation cycle are hard-coded to equal the value 4 and 2, respectively. Recall that, when the 6-state atmospheric jitter truth model is transformed to Jordan canonical form, two states per channel become direct contributions to the outputs, which are used as the jitter contributions to the centroid location on the FLIR plane. Thus, the value of 2 is chosen for the hardcoded value of each of these outputs, to correspond to the one-state contribution from the filter's model of the jitter. The error analysis in this simulation should demonstrate the best performance if the errors are being calculated correctly.
2. In the second simulation, the jitter output of the filter propagation was hard-coded to -4, while the truth model values remained hardcoded at 2. The error analysis in this simulation should show the worst performance if the errors are being calculated correctly.
3. The third simulation involved no hardcoding of the filter or truth states. This simulation is added as a control to ensure that nothing completely unexpected happens in simulations #1 and #2.

SIMULATION #	$\hat{x}(t_i^-)$	$\hat{y}(t_i^-)$	$\hat{x}_c(t_i^-)$	$\hat{y}_c(t_i^-)$
1	-.000572681	.000602245	-.000573158	.000602245
2	-.000572681	.000602245	-8.00057	-7.99940
3	-.000572681	.000602245	-1.33132	-.642725
SIMULATION #	$\hat{x}(t_i^+)$	$\hat{y}(t_i^+)$	$\hat{x}_c(t_i^+)$	$\hat{y}_c(t_i^+)$
1	.0198026	.0466537	.0206451	.0485578
2	7.69954	7.72541	.0179048	.0448005
3	1.31882	.696278	.0426211	.0817142

Table 6.23. Frame Analysis of Mean Errors for Atmospheric Jitter Sign Test

The intent is to compare the mean errors after the first propagation and update cycles to determine if the errors are being calculated correctly. Table 6.23 presents the mean errors in the filter estimates after the first Kalman filter propagation and update cycle.

Analyzing the results of the of Table 6.23 in an identical manner to that used for the pogo sign test, it can be concluded that the jitter error calculations are being done correctly. Refer to Section 6.9.1 for the pogo sign test analysis of Table 6.20.

The second sign check for the atmospheric jitter is again done identically to the pogo sign check in Section 6.9.1. Two simulations are again performed. The first simulation involves hardcoding the output jitter state from the Kalman filter propagation cycle to equal the output of the truth model's jitter output. The second simulation involves hardcoding the output jitter state from the Kalman filter propagation cycle to equal the negative of the output of the truth model's jitter outputs. The temporally averaged performance statistics for simulations #1 and #2 are presented in Tables 6.24 and 6.25, respectively.

Again, as in the test for sign errors in the pogo implementation, the direct comparison of Tables 6.24 and 6.25 demonstrate that the filter represented in simulation #1 outperforms the filter represented in simulation #2. On the basis of these results, it can be concluded that the atmospheric jitter error calculations in the code are being performed correctly, and this suspected problem is not the cause of the

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.063826	.29375
$\hat{y}(t_i^-)$	-.035509	.23917
$\hat{x}(t_i^+)$	-.062046	.28502
$\hat{y}(t_i^+)$	-.037160	.23073
$\hat{x}_c(t_i^-)$	-.062593	.98040
$\hat{y}_c(t_i^-)$	-.035654	.90888
$\hat{x}_c(t_i^+)$	-.054385	.25669
$\hat{y}_c(t_i^+)$	-.043256	.17049

Table 6.24. Performance Statistics for Simulation #1 of Jitter Sign Test

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	.10434	1.2789
$\hat{y}(t_i^-)$	-.030109	1.4876
$\hat{x}(t_i^+)$	.10431	1.2218
$\hat{y}(t_i^+)$	-.026815	1.4172
$\hat{x}_c(t_i^-)$	.063816	1.6163
$\hat{y}_c(t_i^-)$	-.0055187	1.6140
$\hat{x}_c(t_i^+)$	.063691	.26885
$\hat{y}_c(t_i^+)$	.0096029	.18104

Table 6.25. Performance Statistics for Simulation #2 of Jitter Sign Test

10-state filter performance degradation observed in the preceding analyses.

During the course of the jitter sign testing, an interesting result was uncovered. The two tuning plots for simulation #1 of Table 6.24 are included in Appendix G as Figures G.1 and G.2. Notice the overestimation that the filter-computed errors are exhibiting. The tuning parameters for this particular simulation are identical to the tuning parameters of the 10-state filter from Section 6.7. Specifically, the variance of the atmospheric jitter process in the Kalman filter is set at a value of $0.7 \frac{\text{pixels}^2}{\text{seconds}^4}$. Note that the actual errors are on the order of 0.3 pixels, which is expected, since the filter's estimate of the jitter was hardcoded to match the truth models values of the jitter. With this in mind, another simulation was run identically to the simulation represented in Figures G.1 and G.2, except that the jitter variance was set to its nominal value used in previous research, i.e., 0.2. Recall that, in Section 6.2, it was stated that the truth model jitter variance is approximately 0.18 and the value used for the filter jitter variance from past research is 0.2. The tuning plots for this simulation are represented in Figures G.3 and G.4, and as expected, the filter shows tuning characteristics similar to previous research [5, 19, 24].

Based upon these results, it was suspected that the jitter model in the Kalman filter may not be estimating the jitter states as well as originally anticipated, and this might be one of the causes for the degraded performance of the 10-state filter. In addition, the tuning problems of Section 6.2 might be related to the same poor estimation problem of the jitter states. Two final simulations were run to gain more insight into this new occurrence. The two simulations are described as follows:

1. The pogo states are removed from the filter and truth model, along with the bending from the truth model. The jitter variance in the filter is set equal to 0.7 and the truth model jitter variance is kept at 0.18. The jitter states in both the filter and the truth models are not hardcoded, permitting them to react as designed.

2. The above simulation as just described is duplicated except the jitter variance is set to the value used in previous research, i.e., 0.2.

The tuning plots for each of these simulations are located in Appendix G. Recall that the tuning plots used in this research are for actual and filter-computed rms errors of the dynamic position states only, and that additional tuning plots for the jitter position errors would be very desirable and should be investigated in follow-on research. Figures G.5 and G.6 correspond to simulation #1, while Figures G.7 and G.8 correspond to simulation #2. The tuning plots for simulation #1 show good tuning results for a jitter variance of 0.7, while the tuning plots for simulation #2 show the typical underestimation qualities as the plots of Section 6.2 consistently demonstrated. The interesting point is that the actual rms errors in each of these four figures is larger than the actual errors of in Figures G.1–G.4. This seems obvious since the Figures G.1–G.4 are the result of hardcoding the filter jitter equal the truth model's representation of the jitter. Based upon this analysis, the possibility exists that, for the ballistic missile trajectory described in this thesis research, the jitter model in the filter is not estimating the jitter states in the truth model as well as initially anticipated. Due to limiting time constraints, this anomaly was not pursued further, but recommendations to understand the jitter estimation properties and how they relate to this thesis research are described in Chapter VII.

6.9.3 Observability Issues. In an attempt to understand as much as possible about the lack of performance enhancement of the 10-state filter before this thesis effort has to be concluded, a stochastic observability test was performed on both the 8-state and 10-state filter structures discussed in Sections 6.6 and 6.7. By investigating whether any of the states in either of the filter models is unobservable in the output, a better decision on where to focus future trouble-shooting can be determined. The fact that some of the states may be unobservable could be one of the causes of the degraded performance. The stochastic observability condition is

given by the following relationship [7]:

$$\alpha \mathbf{I} \leq \sum_{j=i-N+1}^i \Phi^T(t_j, t_i) \mathbf{H}^T(t_j) \mathbf{R}^{-1}(t_j) \mathbf{H}(t_j) \Phi(t_j, t_i) \leq \beta \mathbf{I} \quad (6.2)$$

If there exist positive numbers α and β , and $0 < \alpha < \beta < \infty$, and a positive integer N such that, for all $i \geq N$, the above relationship holds, then the system is said to be stochastically observable. It should be noted that the observability analyses performed in this section are based upon simulations that were done with single precision numerics. This is important when analyzing the eigenvalues of the respective observability matrices. As will be demonstrated shortly, some of the eigenvalues of the observability matrices are negative quantities, which is theoretically impossible. The source of these negative values is contributed to the single precision calculations used throughout the simulations.

The observability matrix (diagonal terms) for the 8-state filter described in Section 6.6 is given as follows:

$$\begin{bmatrix} 68578 & - & - & - & - & - & - & - \\ - & 50000 & - & - & - & - & - & - \\ - & - & 76.198 & - & - & - & - & - \\ - & - & - & 55.556 & - & - & - & - \\ - & - & - & - & .021664 & - & - & - \\ - & - & - & - & - & .015431 & - & - \\ - & - & - & - & - & - & 26710 & - \\ - & - & - & - & - & - & - & 19474 \end{bmatrix}$$

The observability matrix (diagonal terms) for the 10-state filter described in Section 6.7 is given as follows:

68578	-	-	-	-	-	-	-	-	-
-	50000	-	-	-	-	-	-	-	-
-	-	77.455	-	-	-	-	-	-	-
-	-	-	55.98	-	-	-	-	-	-
-	-	-	-	.022535	-	-	-	-	-
-	-	-	-	-	.015895	-	-	-	-
-	-	-	-	-	-	26710	-	-	-
-	-	-	-	-	-	-	19474	-	-
-	-	-	-	-	-	-	-	9156.7	-
-	-	-	-	-	-	-	-	-	8.4749

Note that the observability matrices are in no way diagonal matrices; but for the purposes of emphasis and clarity, only their diagonal terms are depicted. By looking at the diagonal terms of the two observability matrices, the specific states that could cause possible observability problems can be distinguished by small magnitudes relative to the other diagonal entries. To determine if the two system models are, in fact observable, the eigenvalues of each matrix must be positive. The eigenvalues of the 8-state filter model are, in descending order:

95364
60529
.060781
.00050024
.00034823
.00018957
.0000000026546
-.068208

and the eigenvalues for the 10-state filter are:

$$\begin{bmatrix} 99049 \\ 75006 \\ 3.8473 \\ 1.6355 \\ .10869 \\ .00054219 \\ .000000068269 \\ -.00000049120 \\ -.000037323 \\ -.023223 \end{bmatrix}$$

Note that three of the eigenvalues of the 10-state observability matrix are negative and two of them, namely the sixth and seventh eigenvalues, are very small when compared to the remaining five eigenvalues. For a system to be stochastically observable, none of the eigenvalues can be negative or zero; therefore, the 10-state filter is considered unobservable in at most five of its states. As mentioned earlier, the negative eigenvalues must be due to numerics problems for a given application, since negative eigenvalues are theoretically impossible. To determine possible problem states, the diagonal terms of the 10-state observability matrix are investigated. The five smallest diagonal terms in this matrix correspond to the acceleration and velocity states in both the x and y FLIR directions and the pogo velocity state which is oriented along the missile velocity vector.

For the 8-state filter observability matrix, one of the eigenvalues is negative and four (namely, the fourth, fifth, sixth, and seventh) of them are small relative to the others. This indicates that the 8-state filter is also subject to observability problems and that five states are causing the difficulty. Investigating the diagonal

terms of the 8-state observability matrix, the two acceleration states and the two velocity states in each direction on the FLIR plane seem to be the cause, as well as possibly the y channel jitter state. This is physically reasonable, since acceleration is least observable of position, velocity, and acceleration, from position measurements. Moreover, this effect is most pronounced for benign trajectories, as used in this research. An important aspect is that the observability matrix diagonal terms corresponding to the two pogo states are of sufficient magnitude to indicate no heightened observability problems from their addition to the original 8-state model.

Based on the observability results for both filters, there seems to be a possible problem with the acceleration models in each filter. Additionally, the observability problems with the first-order lag acceleration models could also be affecting the unobservability condition of the pogo velocity. At this point, a recommendation is made possibly to model the filter's position as a second-order Gauss-Markov process versus third-order. Possible models for velocity are $\dot{v} = 0 + w$ (essentially constant velocity paths, plus pseudonoise), or $\dot{v} = -\frac{1}{\tau}v + w$ (first-order Gauss-Markov velocity). The reason is that the ballistic trajectory used in this simulation is a very benign trajectory, and the first-order acceleration process may not be modelling this benign trajectory very well since the target's acceleration is not changing during the course of the simulation [10]. The additional acceleration state is anticipated to be difficult to estimate well under these conditions. One benefit to modelling the acceleration as a first-order process is that highly dynamic targets can be tracked accurately. The ballistic missile simulated in this research is benign and experiences no harsh dynamics. Chapter VI re-emphasizes this point for the benefit of future research possibilities.

One last note on the observability of the 8-state filter is that the observability matrix is independent of the truth model trajectory selected in the simulation. Each of the terms in Equation (6.1) are constant matrices, and the only difference between the observability matrix for this 8-state filter and the 8-state filters used in

past AFIT research is the value of the correlation time chosen for the acceleration process. Thus, the possibility exists that the 8-state filters used in past research have had similar observability problems that went unnoticed. Chapter VII also discusses recommended actions to be taken concerning this issue, before continuing study is performed on this research.

6.9.4 Possible Pogo-Jitter Interactions. One final study is performed to assess the possibility of pogo-jitter interactions existing in the filter model and/or the truth model. In addition, this analysis also investigates the proposed feasibility of adding pogo states to the filter for enhanced performance, which is the basis of the 10-state filter. To accomplish these objectives, two simulations are performed that remove jitter states, thereby removing any possible interaction of pogo and jitter phenomena from affecting the results. In the first simulation, the only states included in the filter and the truth model are the pogo states and the dynamic target states. The second simulation removes the pogo states from the filter model, but keeps them in the truth model. Each simulation is performed using ten Monte Carlo runs and a simulation time of ten seconds. Performance statistics are gathered over the last five seconds of the simulation, and the results are presented in Tables 6.26 and 6.27. The same tuning parameters used in Sections 6.6 and 6.7 are duplicated for the two simulations of this section, with the exception of the jitter parameters, which are irrelevant for this analysis.

Comparing the results in Tables 6.26 and 6.27, the filter from simulation #1 that models the pogo in the filter structure shows an average performance enhancement on the order of 50%. Particularly note the improved performance in the tracking accuracies signified by the 1σ statistics. This is a welcomed enhancement, since the applicability for pointing a laser weapon accurately depends on a small tracking "envelope" in order to avoid painting the target.

The results of this analysis demonstrate that a possible pogo-jitter interaction

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.16696	.36679
$\hat{y}(t_i^-)$	.16254	.32098
$\hat{x}(t_i^+)$	-.16351	.35699
$\hat{y}(t_i^+)$	.15990	.30789
$\hat{x}_c(t_i^-)$	-.14366	1.3816
$\hat{y}_c(t_i^-)$	.16443	.22974
$\hat{x}_c(t_i^+)$	-.095352	.27453
$\hat{y}_c(t_i^+)$	.11109	.220111

Table 6.26. Performance Statistics for Simulation #1 with Pogo in the Filter

Temporally Averaged Error Statistic	Mean	1 Sigma
$\hat{x}(t_i^-)$	-.20393	.81056
$\hat{y}(t_i^-)$	.07813	.97758
$\hat{x}(t_i^+)$	-.20213	.77871
$\hat{y}(t_i^+)$	.075085	.92391
$\hat{x}_c(t_i^-)$	-.18583	3.3051
$\hat{y}_c(t_i^-)$	.041323	5.6122
$\hat{x}_c(t_i^+)$	-.17849	.55789
$\hat{y}_c(t_i^+)$	.02898	.42620

Table 6.27. Performance Statistics for Simulation #2 with Pogo Removed from the Filter

does exist in the 10-state filter structure, and that an expected performance enhancement should result when modelling the pogo phenomenon in the Kalman filter model. Further investigation of the pogo-jitter interaction is strongly motivated.

6.10 Summary

This chapter has analyzed several different issues regarding the tracking algorithm of Chapter V. First, the tuning issues with regard to the filter model dynamics state parameters were presented, followed by the 8-state filter benchmark analysis, FOV rotation schemes, and the 10-state filter performance analysis. After the performance degradation results of the 10-state filter were discovered, an analysis of two single filters (8-state and 10-state) was performed with the bending phenomenon removed from the truth model. This modification resulted in continued performance degradation by the 10-state filter, so a comprehensive trouble-shooting plan was implemented to gain insights into the possible reasons for the degraded performance. A pogo-jitter interaction appears to be a substantial cause of this degradation, and it warrants further investigation. Before implementing the trouble-shooting schemes, a discussion regarding the MMAF analysis and robustness studies was provided, although neither was implemented in this thesis research effort.

VII. Conclusions and Recommendations

7.1 Introduction

This chapter summarizes the final conclusions of this thesis and suggests topics for further study. Section 7.2 draws conclusions based upon the results obtained in Chapter VI. Suggestions for continued research on applying the FLIR tracking system to the ballistic missile tracking problem are enumerated in Section 7.3.

7.2 Conclusions

Numerous conclusions have been made throughout the performance analysis of Chapter VI. These conclusions will be reassembled and presented in the following subsections.

7.2.1 Filter Tuning Based upon Dynamics Parameters. As demonstrated in Section 6.2, the tuning of the 8-state Kalman filter was not possible strictly using the tuning parameters of the target dynamics model. Several variations of the parameters were examined with little success. Eventually, to obtain good tuning characteristics on the FLIR position states, the atmospheric jitter variance was used as the dominant tuning parameter. This makes sense, since the unmodelled effects are better "captured" by the filter jitter states than the filter target states, so that the filter target estimates are not severely distorted. Using this parameter, an 8-state Kalman filter model was tuned for truth model scenarios that ranged from a large amplitude, high frequency pogo phenomenon to a simulation with no plume pogo present.

Based upon the observability testing performed in Section 6.9.3, the first-order Gauss-Markov acceleration model may be inappropriate for modelling the benign behavior of a ballistic missile that is not undergoing a "staging" event. The acceleration states of the filter seem to cause an observability problem which may

be overcome by using a second-order Gauss-Markov target position process model versus a third-order model.

7.2.2 Rotating Field-of-View Results. The analysis comparing the various rotating FOV schemes in Section 6.4 proved that the diagonal rotating field-of-view can provide enhanced performance over the non-rotating field-of-view and the rotating field-of-view implemented by Norton [19]. This conclusion is based upon a velocity orientation angle of approximately 60° , which was the chosen trajectory for this research; but it could be easily extended to other target orientations. The conclusion that the DRFOV, properly aligned, can provide enhanced tracking performance of a missile hardbody whose plume is undergoing a pogo phenomenon, makes sense intuitively based strictly on geometry. It is recommended that future research involving plume pogo use the DRFOV tracking scheme.

7.2.3 10-State Filter Tracking Performance. As demonstrated by the results of Chapter VI, the 10-state filters analyzed showed degraded or mixed performance results when compared to corresponding 8-state filters. These poor performance results were the driver for the trouble-shooting analyses conducted in Chapter VI. Initially, the problem was thought to be a truth model pogo-bending interaction based upon each phenomenon's frequency characteristics. The analyses conducted in Sections 6.6 and 6.7 basically concluded that the proposed pogo-bending interaction, if one exists, is not the cause of the degraded performance of the 10-state filter. When the bending phenomenon was removed from the truth model, the 8-state filter continued to outperform the corresponding 10-state filter. The results of these analyses implied that a possible sign error existed in the Fortran code when calculating the error statistics. This conclusion was inferred because the 8-state filter outperformed the 10-state filter in every performance statistic, when the converse was actually expected to occur.

To investigate the possibility of a sign error, the pogo and jitter models were evaluated independently. As concluded in Chapter VI, the individual frame checks, as well as the overall temporally average statistics, proved that both the pogo and jitter models are implemented correctly in the code.

During the sign analysis of the jitter phenomenon, it was discovered that the jitter model imbedded in the Kalman filter may not be estimating as well as anticipated. This result was uncovered when the jitter estimate in the Kalman filter was hardcoded to equal the true jitter output of the truth model. The possible poor estimation properties of the jitter model may have an impact on the inability of the 10-state filter to show improved performance over the 8-state filter.

Additionally, the results of Section 6.9.4 demonstrate that an interaction between the pogo and the atmospheric jitter may also exist. The performance of two filters were compared against a truth model that contained only the two pogo states and the two dynamic states. One filter included the pogo phenomenon in its model while the other filter had no pogo modelling. The jitter model in both filters was removed, and the temporally averaged statistics compared. The filter that modelled the pogo phenomenon showed a performance enhancement of approximately 50% over the filter with no pogo modelling. These results tend to imply that a possible interaction is occurring between the pogo and jitter states in the filter, possibly causing the observed performance degradation of the 10-state filter from that of the 8-state filter. Additionally, the improved performance obtained by modelling the pogo effect in the filter demonstrates the applicability of an eventual MMAF implementation in the tracking algorithm, were it not for the pogo-jitter interaction problem.

Another possible explanation for the decreased performance of the 10-state filter was seen by the observability results of Section 6.9.3. The 10-state filter proved to contain a maximum of five states that are unobservable in the output of the 10-state filter structure. Those states included the two acceleration and velocity states

characterizing the hardbody dynamics and the state representing the pogo velocity. This is physically reasonable, since acceleration and velocity are least observable from position measurements. The fact that the 10-state filter is unobservable in these states can possibly explain the decrease in performance.

In conducting the observability analysis on the 10-state filter, it was discovered that the 8-state filter also demonstrated an observability problem. Based upon the eigenvalue analysis, five states seem to be unobservable in the output of the 8-state filter. These states are, again, the two acceleration and velocity states, as well as possibly the y channel jitter state. Going from the 8-state to the 10-state filter did not increase the number of small (essentially zero) eigenvalues, thus these results tend to imply that the addition of the pogo states to the 10-state filter does not seem to cause a problem in itself, but the first-order Gauss-Markov model for acceleration may not be the best choice to model the benign dynamic characteristics of the ballistic missile. Additionally, the 8-state filters used in previous AFIT research involving the FLIR tracking algorithm may have also suffered from the same observability problems.

Based upon the numerous trouble-shooting results obtained in Section 6.9, various recommendations for future study of the pogo phenomenon are provided in the next section.

7.3 Recommendations

The following recommendations are suggested for further study in applying the FLIR tracking algorithm to the ballistic missile plume problem. Many of the recommendations are based upon the trouble-shooting performed in Chapter VI, with the overall intent of improving the performance of the 10-state filter over the 8-state filter and eventually implementing an MMAF algorithm.

7.3.1 Observability of Previously Used 8-State Filters. Before continuing the study of the ballistic missile undergoing a plume pogo phenomenon, the observabil-

ity of previously used 8-state filters should be investigated. Doing an observability analysis on the filters used in previous work can possibly shed some light on the observability issues of the 8-state filter used in this research. The Fortran code is presently implemented for single precision numerical calculations. For more precise calculations, particularly when studying filter observability, a double precision implementation of the code should eventually be adopted.

7.3.2 Remodelling of the Filter Dynamics. Based upon the observability problems associated with the acceleration states in the 8-state filter for the benign dynamics of this research, the velocity states of the filter should be modelled as the output of a first-order Gauss-Markov process. This may prove to be a more appropriate model for the benign dynamics of the simulated ballistic missile trajectory. The filter dimension will be reduced to six states, for which an observability check can be performed.

In addition to the remodelling of the dynamic states, the output plotting routine should be modified to add tuning and performance plots to analyze the filter jitter estimates, as well as pogo estimates for eventual inclusion of the pogo effect in the Kalman filter model. This modification was not included at the start of the present research effort because the filter jitter was assumed to be adequately tuned and characterized. The results of Section 6.9.2 suggest otherwise, while the analysis performed in Section 6.9.4 suggest that the pogo and jitter phenomena are possibly interacting to cause the performance degradation of the 10-state filter. Having the additional tuning and performance plots will provide a necessary capability to characterize the jitter and pogo estimates in the filter.

Based upon the above suggested modifications, performance of single filters containing the pogo phenomenon can be compared to performance of single filters without the pogo modelling. With the aid of the additional analysis plots, the interactions of all of the filter states can be properly understood and characterized,

and the filters containing the pogo phenomenon should outperform the benchmark filters that do not model that phenomenon. The proposed robustness analyses of Section 6.8 and the MMAF scenarios described in Chapter IV can be implemented to enhance the tracking algorithm further.

7.3.3 Continued Characterization of the Plume Pogo. Once the MMAF is implemented in the tracking algorithm, other performance enhancements and characterizations can be pursued. Testing the performance of the MMAF tracker against differing ballistic missile trajectories, i.e. different velocity orientation angles, can provide insights into the robustness of the MMAF for differing ballistic missile acquisition scenarios.

The tracking algorithm should also be analyzed for simulated staging events. This could entail adding an elemental filter to the MMAF structure to model the changing acceleration characteristics experienced by the missile during the firing of a staging motor. Implementation of a rectangular rotating field-of-view (RRFOV) to perform the tracking during the simulated staging event may prove beneficial.

Additional performance enhancements for accurately estimating the location of the missile hardbody can be achieved through illumination of the hardbody with a low energy laser and observing the speckle of the return. Simulating this effect can provide additional measurement information to help separate the plume centroid location from the missile hardbody to improve the tracking performance of the algorithm still further. Also, the additional information on the location of the hardbody center of mass may provide the filter with the capability of modelling the "offsets" (see Section 4.3.1) between the pogo equilibrium point and the hardbody center of mass.

Recall the R_f matrix of the filter measurement model of Equation (4.23) which represents second-order statistics of errors due to the background noise, FLIR noise, and errors in the correlation algorithm. To derive this result, Rogers [21] devel-

oped software which tested the correlator/centroid algorithm's position estimates in order to characterize the mean and variance of the algorithm's errors. The values needed to describe the error's mean and variance are a function of the template-target separation, the thresholding level used by the correlator, and the variance of the background noise. The model in Equation (4.23) was developed based upon a non-rotating field-of-view FLIR with a 20 micro-radian/pixel proportionality constant and a three hotspot dynamic target simulation. The research work done in this thesis deals with a rotating field-of-view FLIR, a 15 micro-radian/pixel proportionality constant, and a target image simulated by differencing two Gaussian intensity functions, at a range two orders of magnitude greater than previous simulations. Based upon these changes to track a ballistic target relative to the targets simulated by Rogers, it is recommended that a re-evaluation of Equation (4.23) be performed to determine if the existing model remains valid. Specifically, a new R_f should be computed empirically and compared to Rogers' result. Another reason for redoing the Rogers analysis, other than just the changes to the simulation, is to understand why the diagonal terms of the matrix are not equal. Intuitively, if the correlator algorithm performs its template-image correlations identically in both directions of the FLIR plane, then the covariance of the noise associated with the error in each direction should be equal; but as evidenced by Equation (4.23), this is not the case.

Appendix A. Gain Calculation for Pogo Effect

As presented in Section 3.2.4, the Laplace domain transfer function for the ballistic missile plume pogo phenomenon is given by Equation (3.29), and the continuous time state space representation is:

$$\dot{\mathbf{x}}_p(t) = \begin{bmatrix} 0 & 1 \\ -\omega_{np}^2 & -2\zeta_p\omega_{np} \end{bmatrix} \mathbf{x}_p(t) + \begin{bmatrix} 0 \\ K_p\omega_{np}^2 \end{bmatrix} w_p(t) \quad (\text{A.1})$$

where:

- $\mathbf{x}_p(t)$ = 2-dimensional pogo state vector
- $w_p(t)$ = 1-dimensional zero-mean, white Gaussian noise of unity strength
- ω_{np} = undamped natural frequency for pogo effect
- ζ_p = pogo damping coefficient
- K_p = gain adjustment to obtain desired rms pogo amplitude

The output relationship for the pogo phenomenon along the missile velocity vector is:

$$y_p(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_p(t) \quad (\text{A.2})$$

The gain K_p is adjusted to obtain the desired rms pogo amplitude (Table 3.1), which is expressed mathematically as:

$$\begin{aligned} \sigma_p^2 &= E[y_p^2(t)] \\ &= E[x_1^2(t)] \end{aligned} \quad (\text{A.3})$$

where σ_p is the desired rms pogo along the velocity vector of the missile. The continuous time model for the covariance matrix for the state of this system can be written as [7]:

$$\dot{\mathbf{P}}(t) = \mathbf{F}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \quad (\text{A.4})$$

where:

- $\mathbf{P}(t)$ = the continuous time state covariance matrix
- $\mathbf{F}$ = the pogo system plant matrix of Equation (A.1)
- $\mathbf{G}$ = the noise distribution matrix of Equation (A.1)
- Q = unity variance of w_p

When the system reaches steady state, $\dot{\mathbf{P}}(t)$ will equal zero. For the steady state solution, substituting the appropriate values of $\mathbf{F}$, $\mathbf{G}$, and Q into Equation (A.4) yields the following:

$$0 = \begin{bmatrix} P_{12} + P_{21} & P_{22} - \omega_{np}^2 P_{11} - 2\zeta_p \omega_{np} P_{12} \\ P_{22} - \omega_{np}^2 P_{11} - 2\zeta_p \omega_{np} P_{21} & -\omega_{np}^2 P_{12} - \omega_{np}^2 P_{21} - 4\zeta_p \omega_{np} P_{22} + K_p^2 \omega_{np}^4 \end{bmatrix} \quad (\text{A.5})$$

where:

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \quad (\text{A.6})$$

Solving each entry of Equation (A.5) and realizing that the covariance matrix is a symmetric matrix yields:

$$P_{12} = -P_{21} = 0 \quad (\text{A.7})$$

$$P_{22} = \omega_{np}^2 P_{11} \quad (\text{A.8})$$

$$K_p^2 = \frac{4\zeta_p P_{22}}{\omega_{np}^3} \quad (\text{A.9})$$

Substituting Equation (A.8) into Equation (A.9) and recognizing that

$P_{11} = E[x_1^2] = \sigma_p^2$, gives:

$$K_p = 2\sigma_p \sqrt{\frac{\zeta_p}{\omega_{np}}} \quad (\text{A.10})$$

Using the development in this appendix, the value of the pogo gain can now be determined based on a desired rms pogo amplitude.

Appendix B. *Plots for Filter Tuning via Dynamic Trajectory*

Parameters $(\sigma_x^2, \sigma_y^2, \tau_x, \tau_y)$: Discussion in Section 6.2

B.1 Figures B.1-B.10: $\sigma_x^2, \sigma_y^2 = 250, \tau_x, \tau_y = 4, \sigma_a^2 = 0.2$

B.2 Figures B.11-B.20: $\sigma_x^2, \sigma_y^2 = 2000, \tau_x, \tau_y = 8.5, \sigma_a^2 = 0.2$

B.3 Figures B.21-B.30: $\sigma_x^2, \sigma_y^2 = 530, \tau_x, \tau_y = 8.5, \sigma_a^2 = 0.2$

B.4 Figures B.31-B.40: $\sigma_x^2, \sigma_y^2 = 5, \tau_x, \tau_y = 8.5, \sigma_a^2 = 0.2$

B.5 Figures B.41-B.50: $\sigma_x^2, \sigma_y^2 = 5, \tau_x, \tau_y = 8.5, \sigma_a^2 = 2.15$

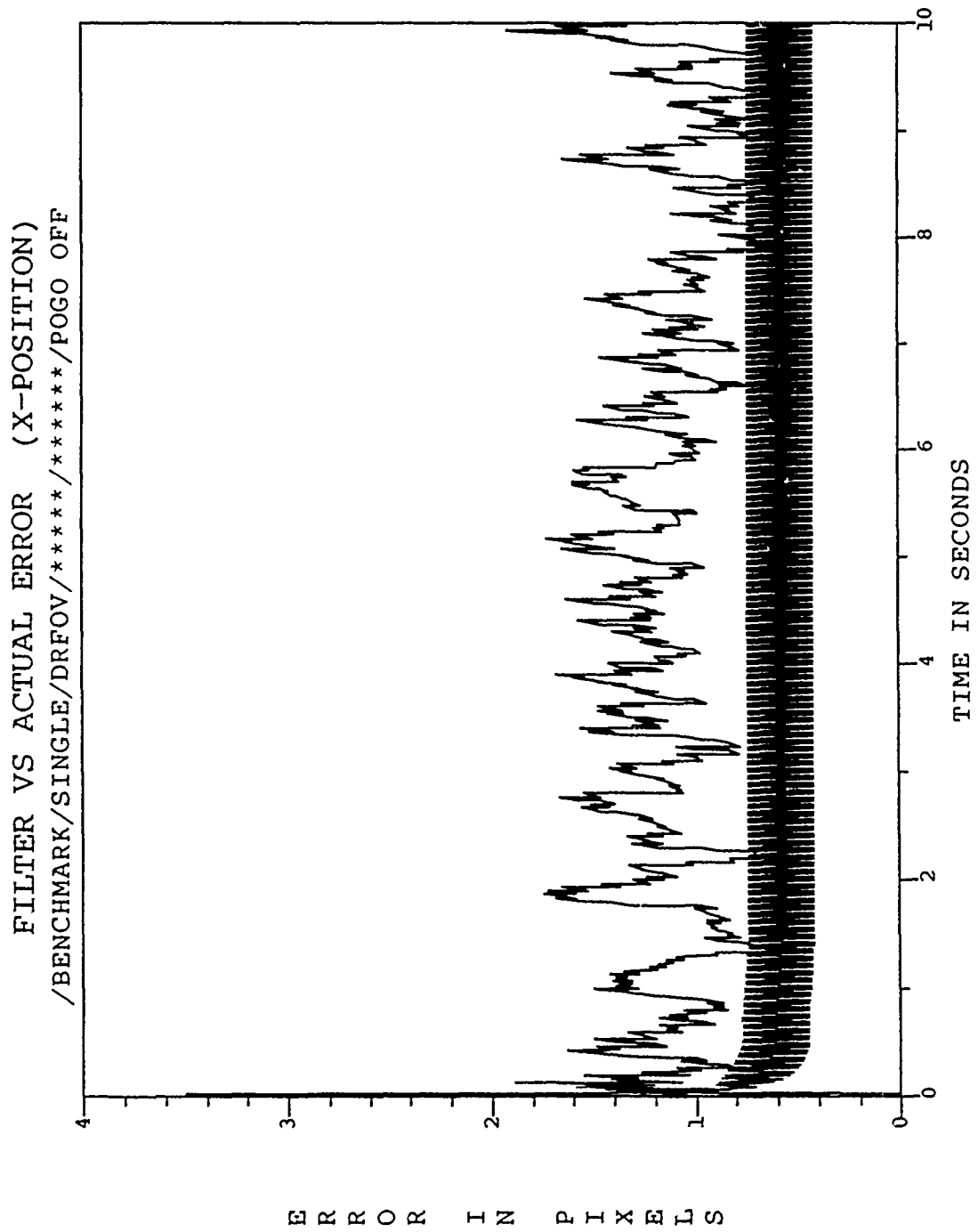


Figure B.1. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

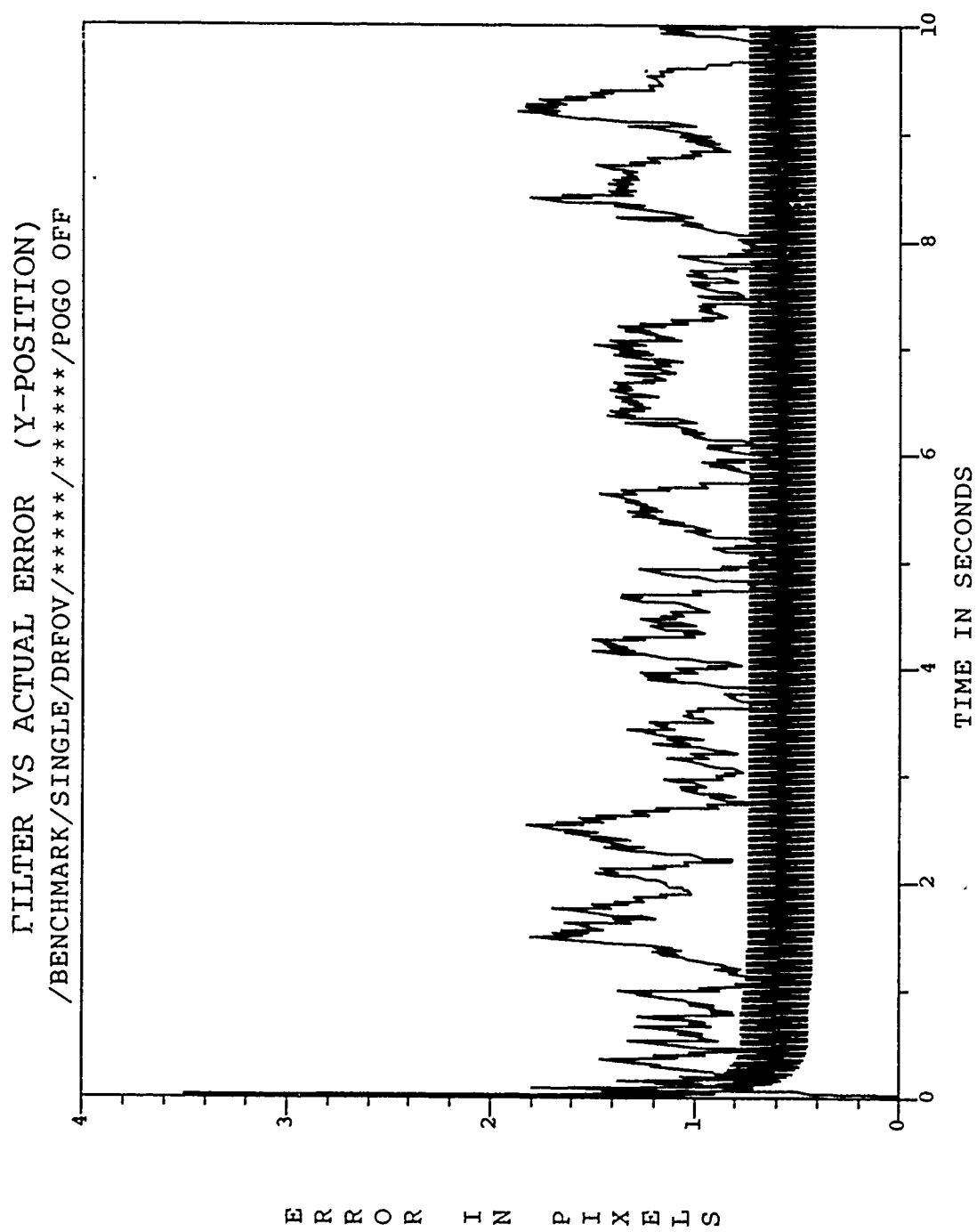


Figure B.2. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

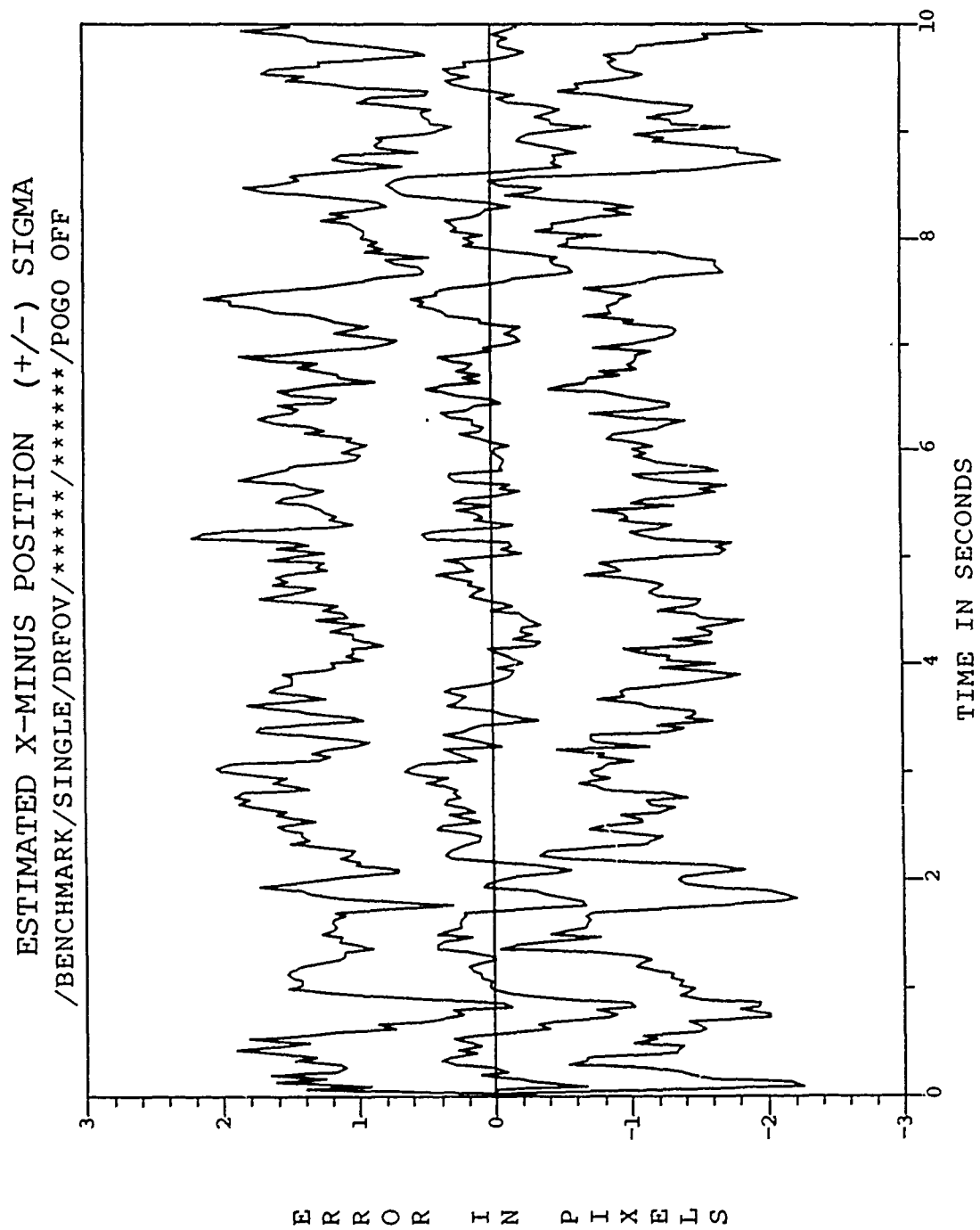


Figure B.3. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

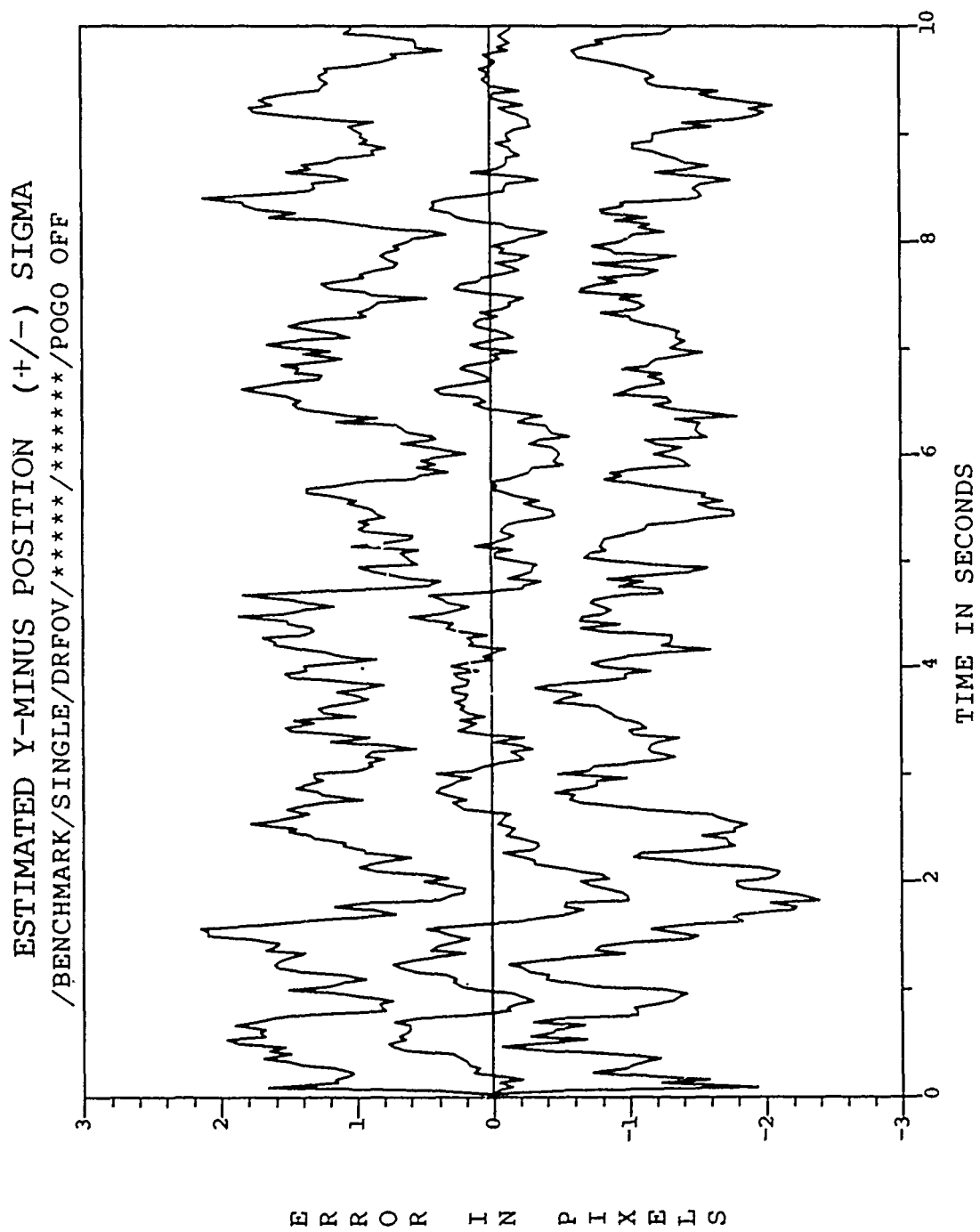


Figure B.4. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

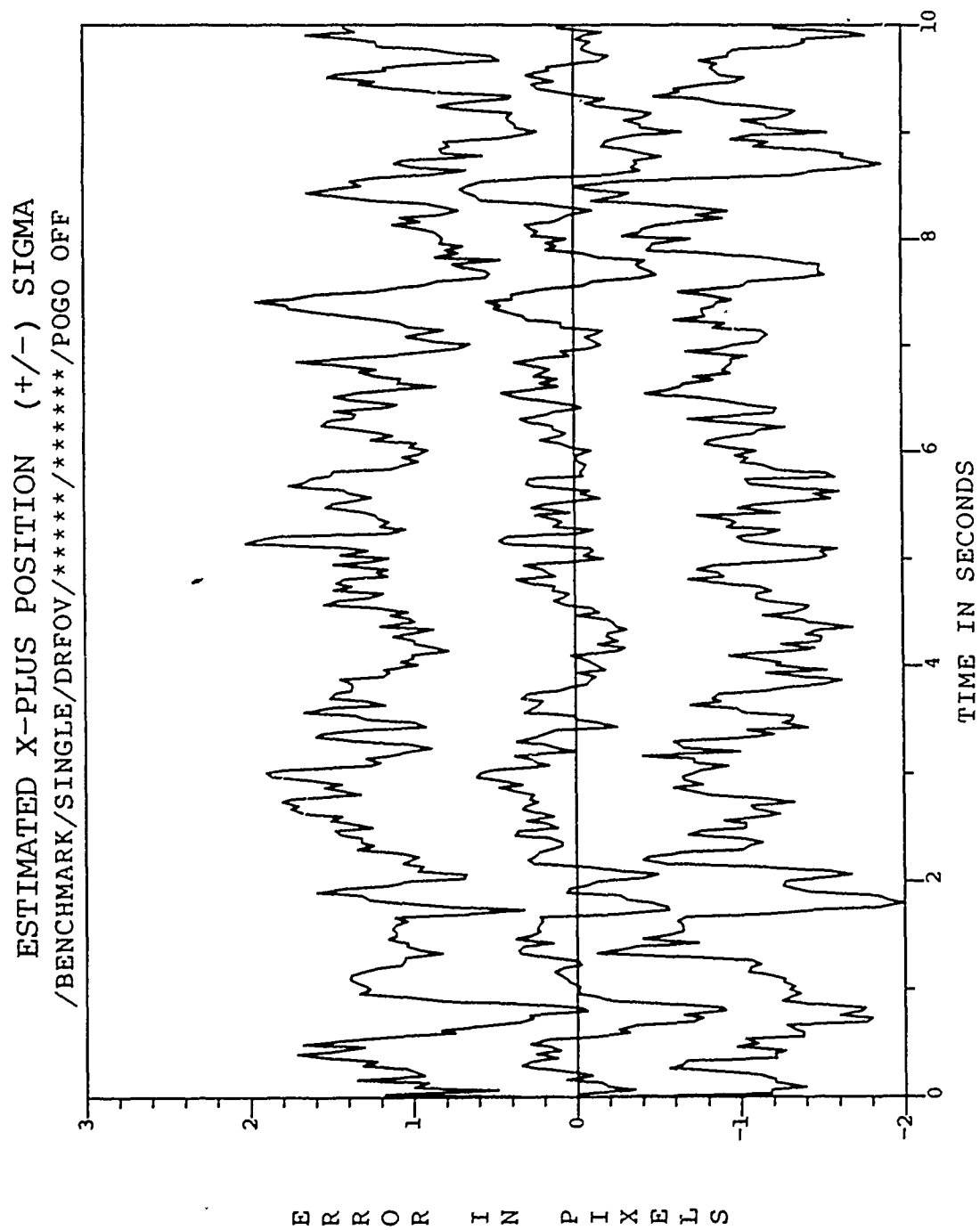


Figure B.5. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

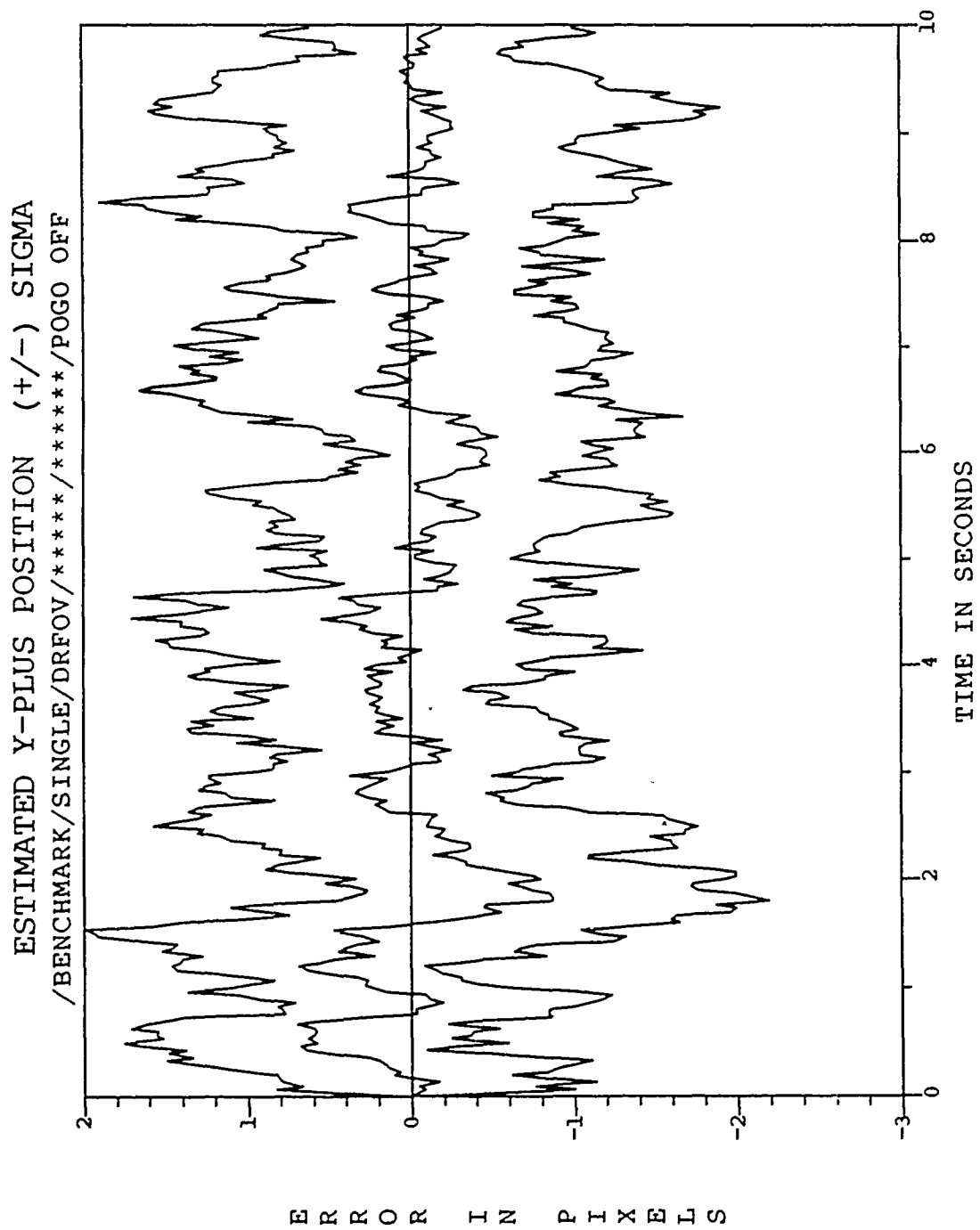


Figure B.6. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

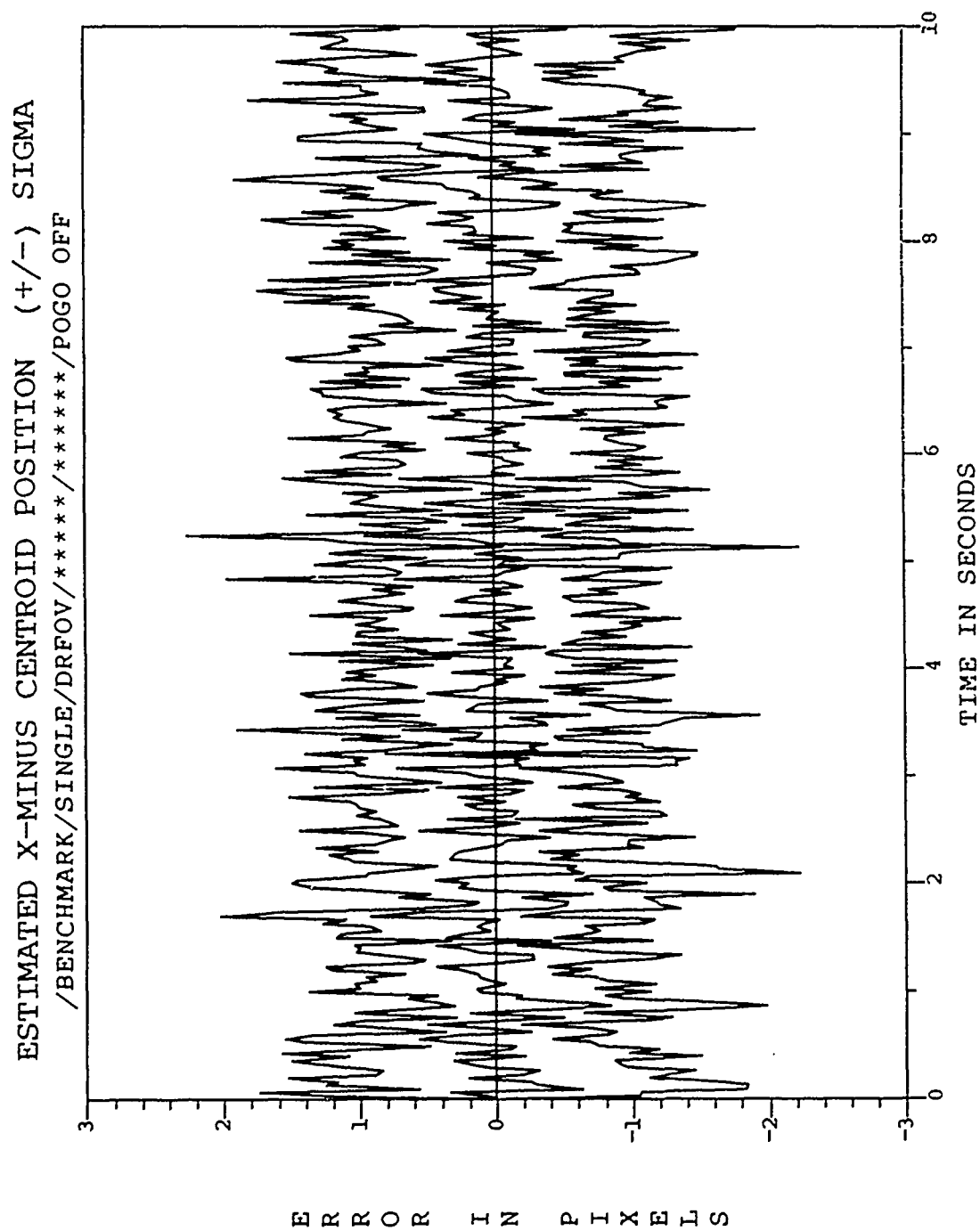


Figure B.7. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

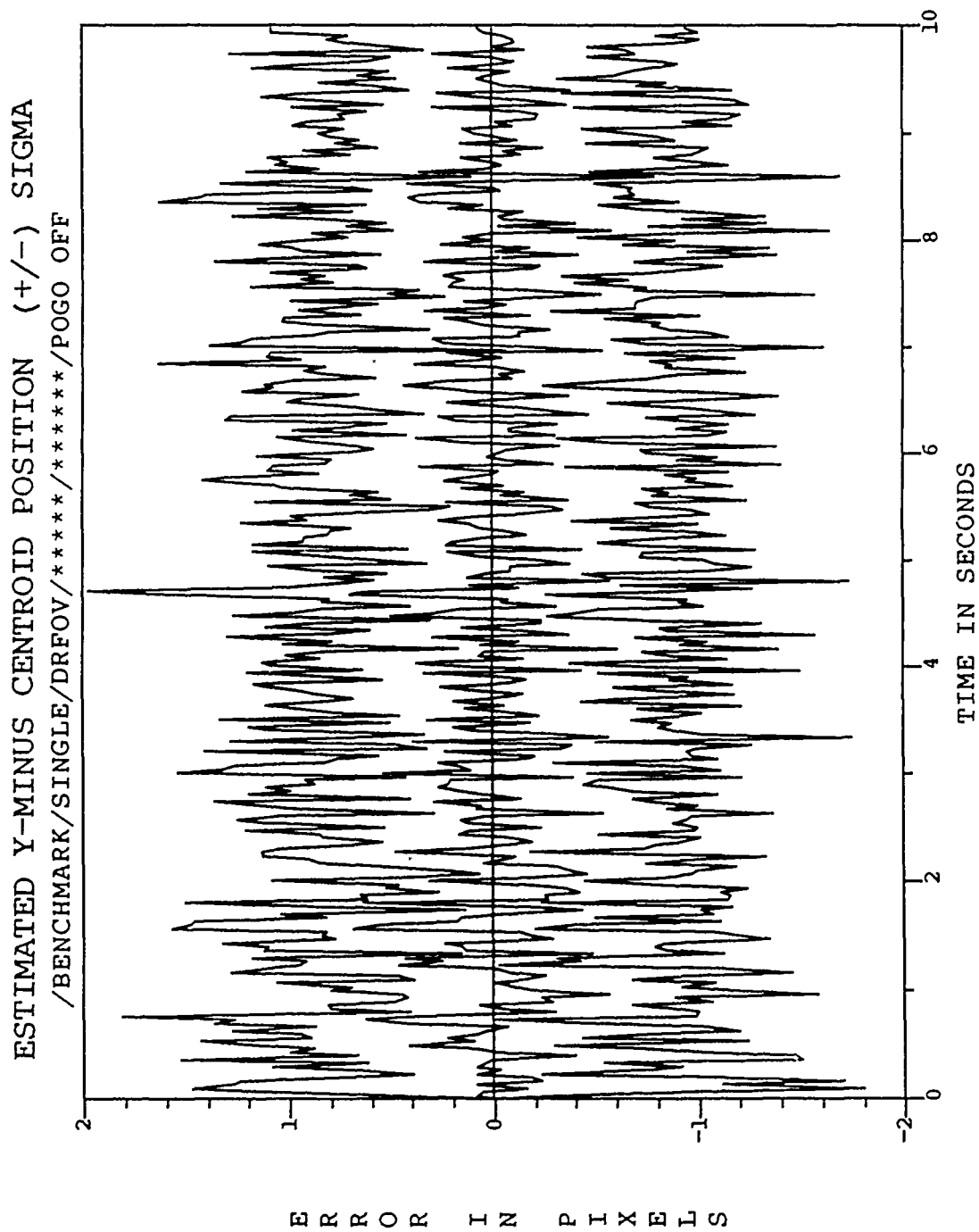


Figure B.8. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

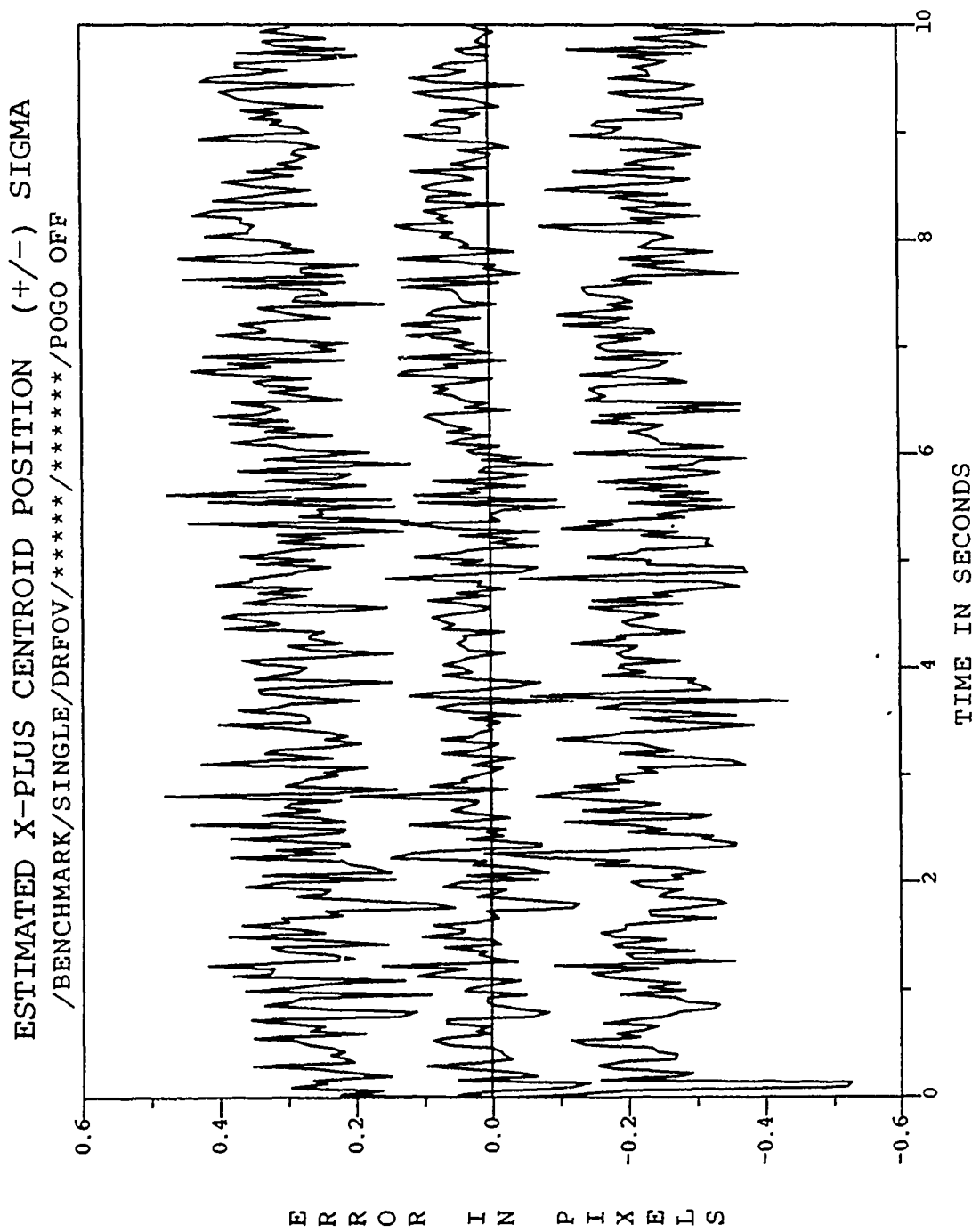


Figure B.9. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

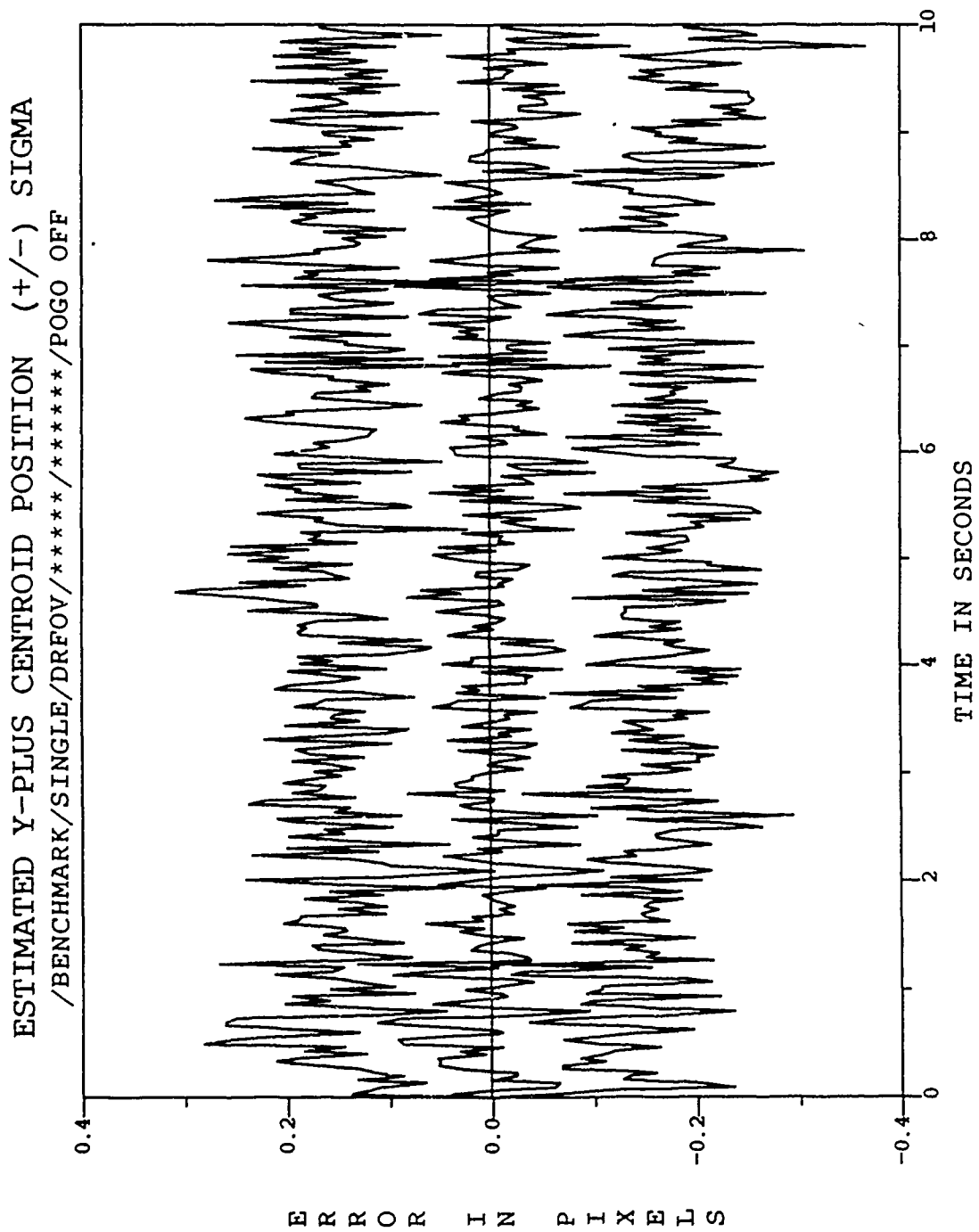


Figure B.10. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

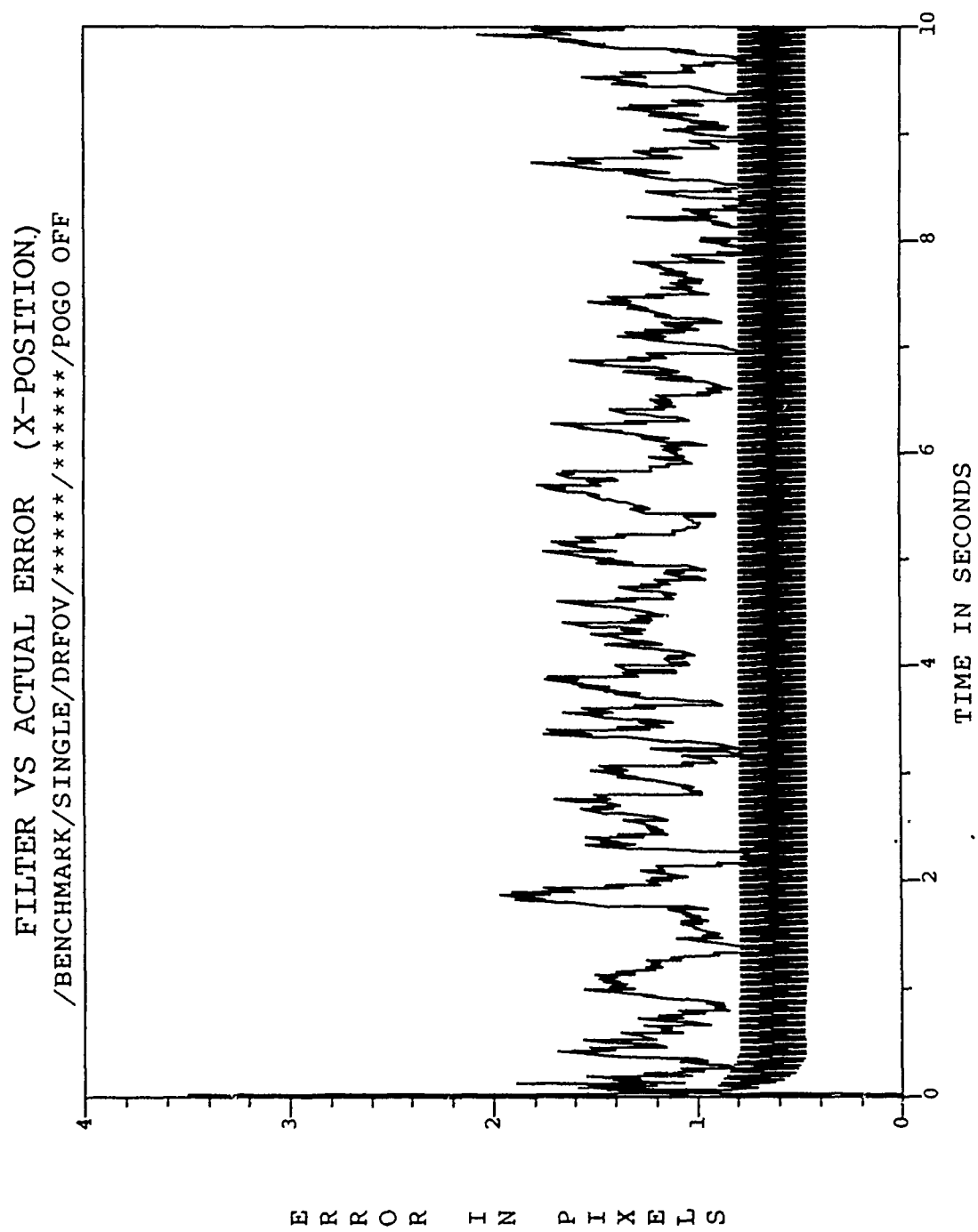


Figure B.11. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

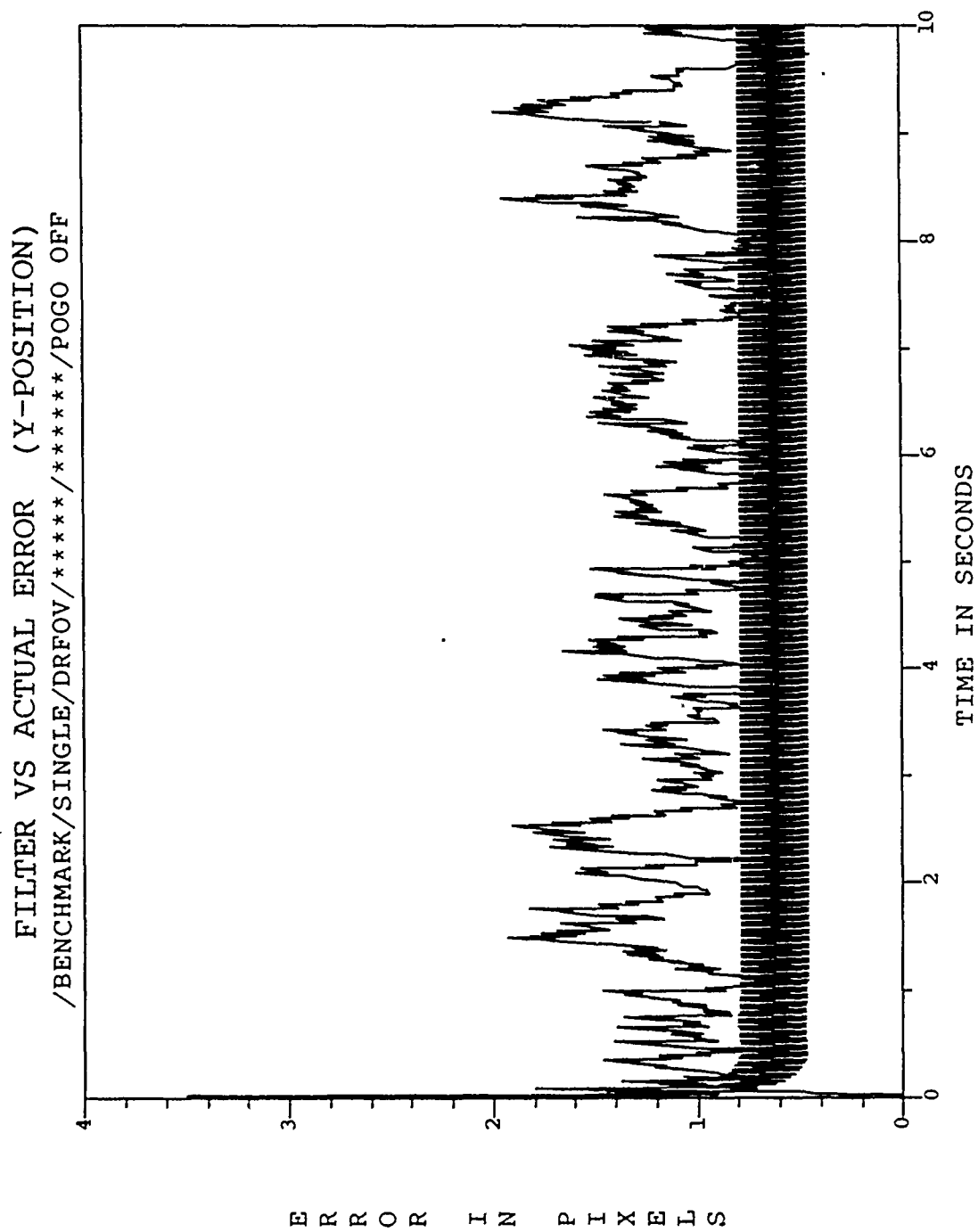


Figure B.12. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

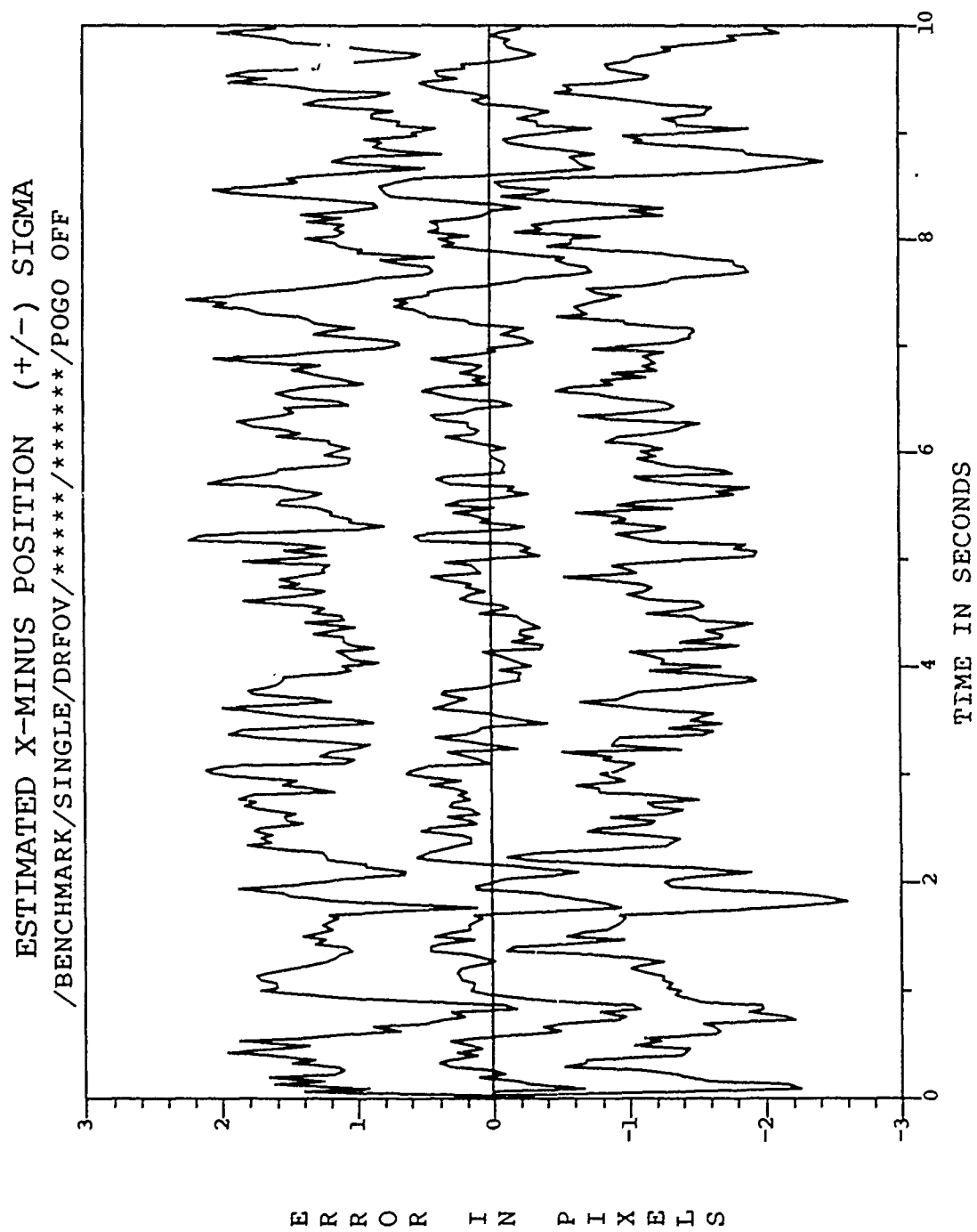


Figure B.13. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

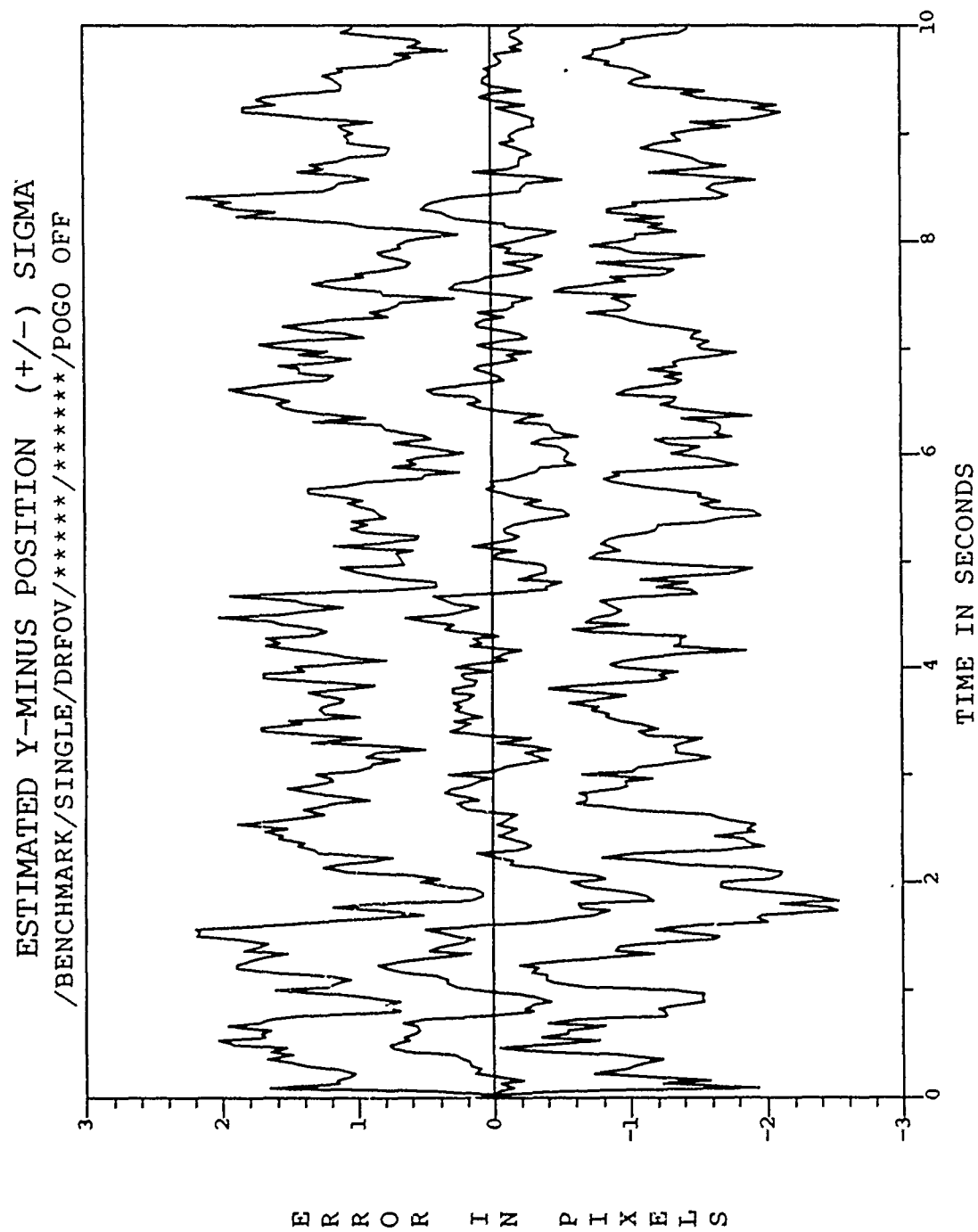


Figure B.14. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

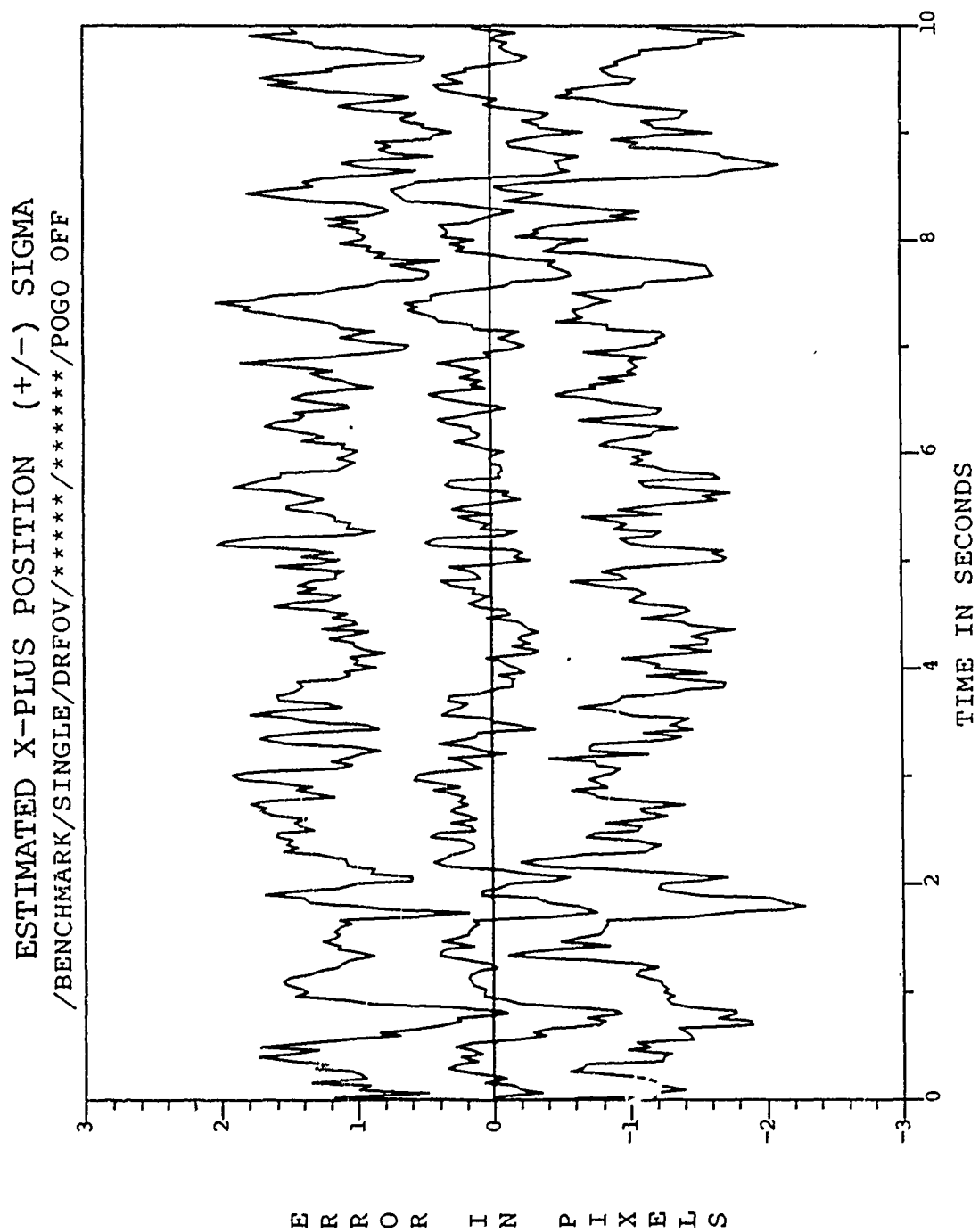


Figure B.15. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

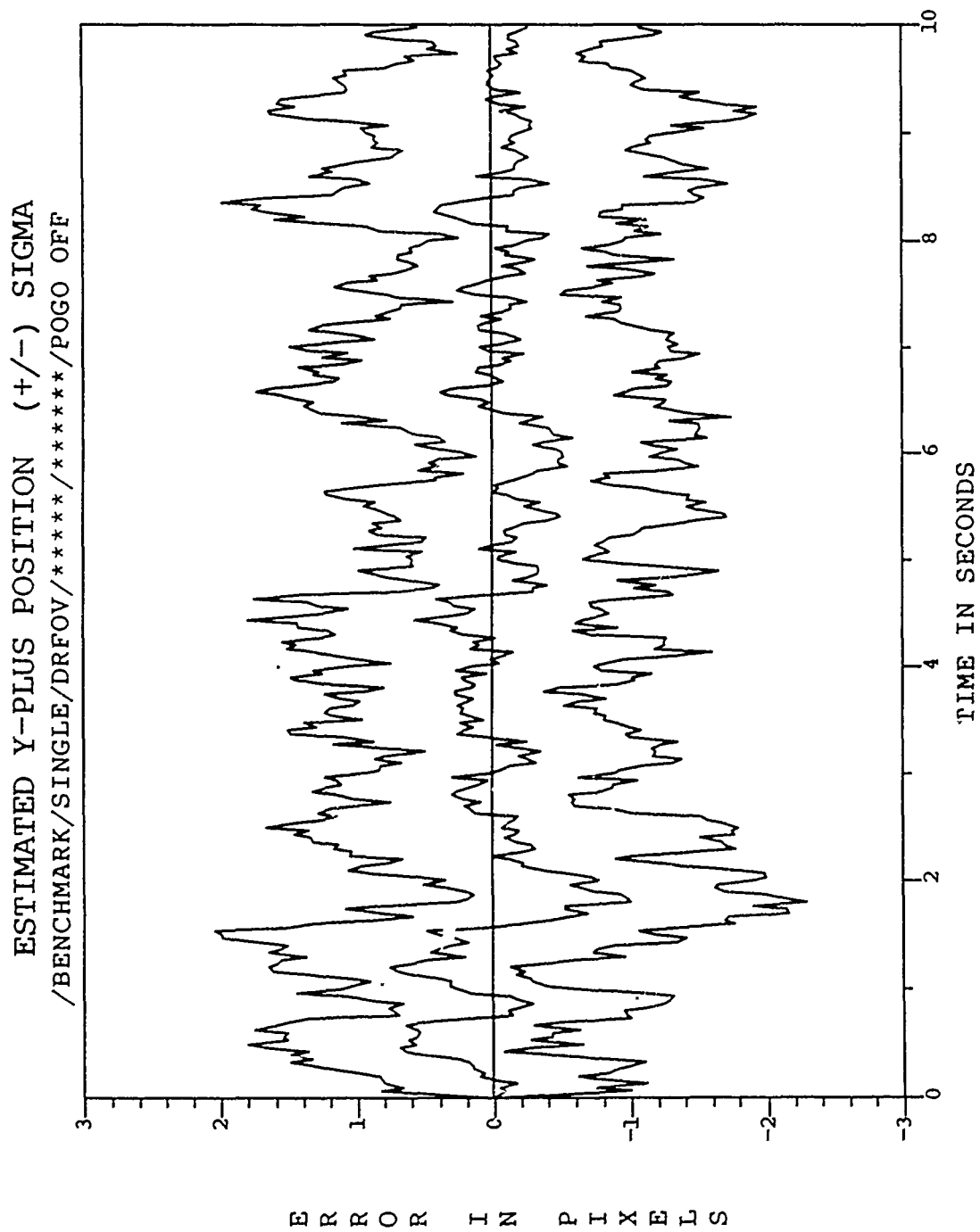


Figure B.16. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

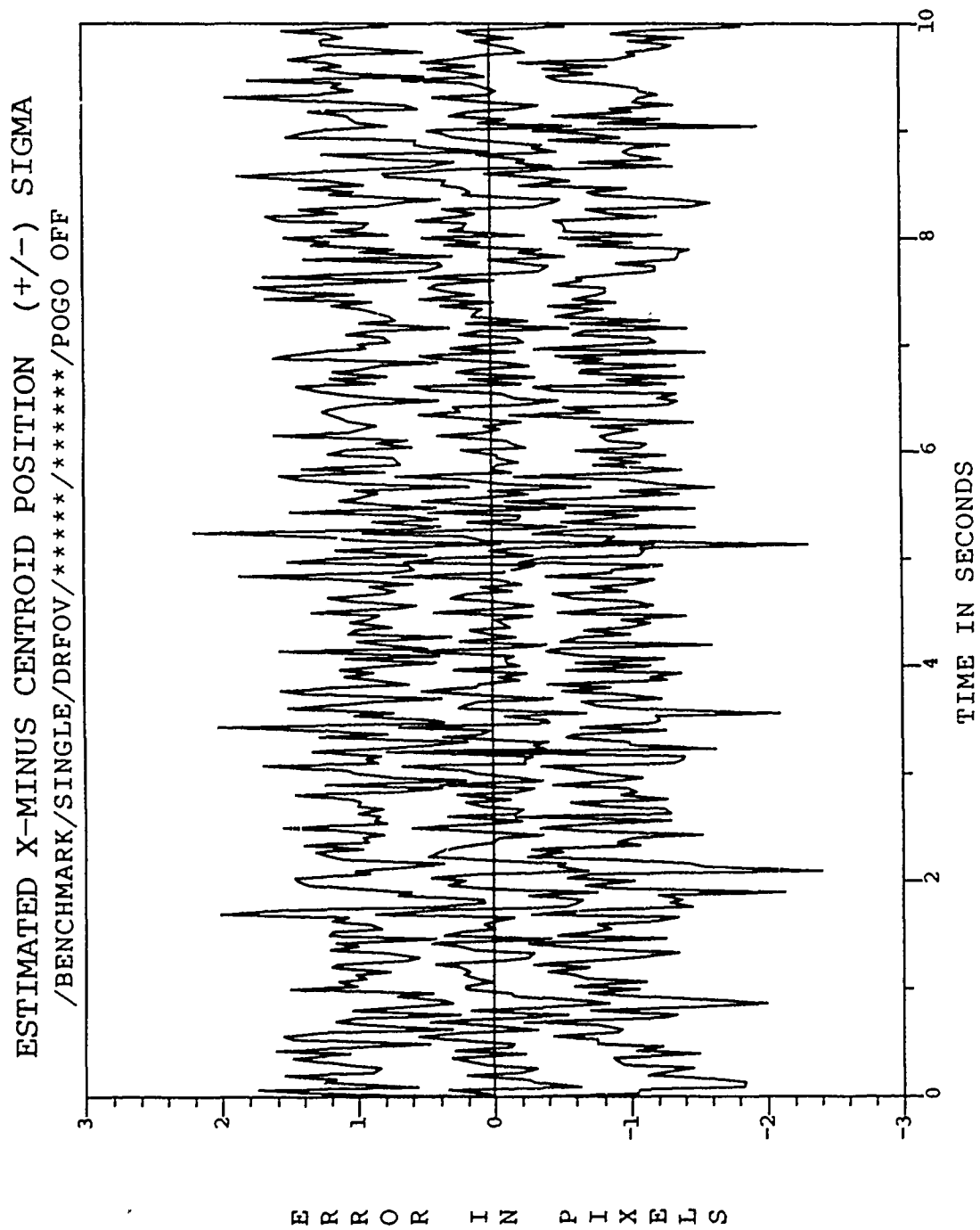


Figure B.17. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

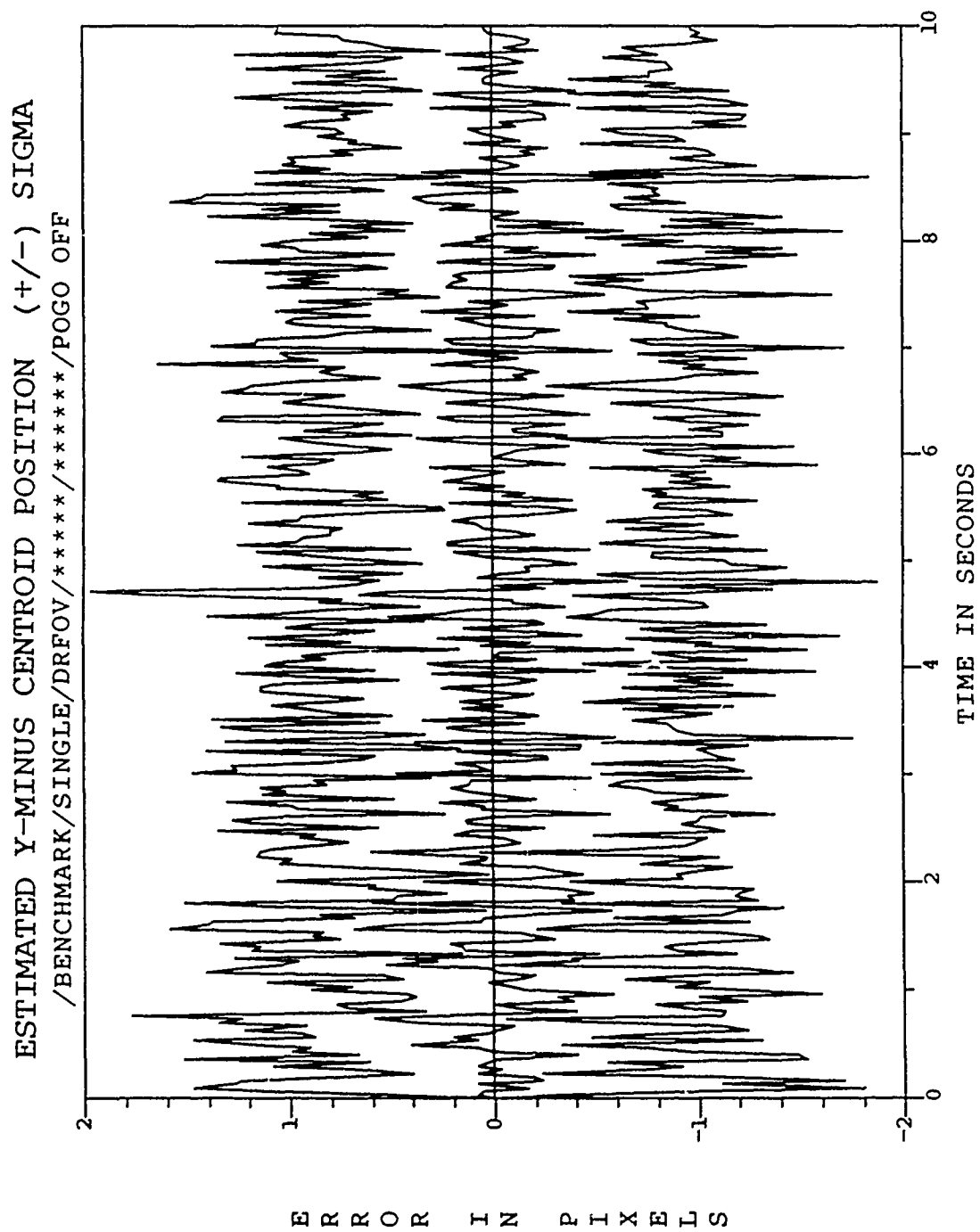


Figure B.18. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

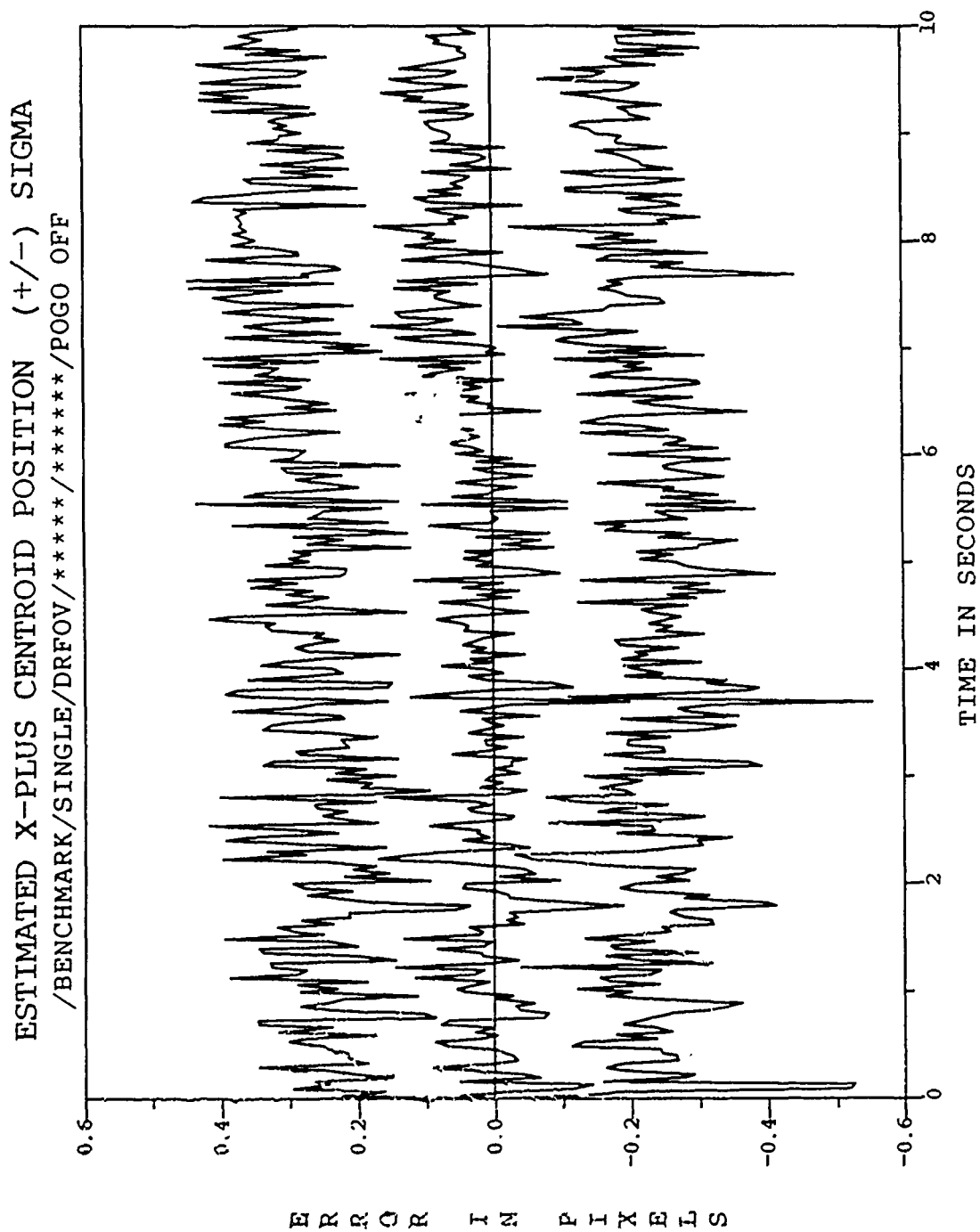


Figure B.19. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

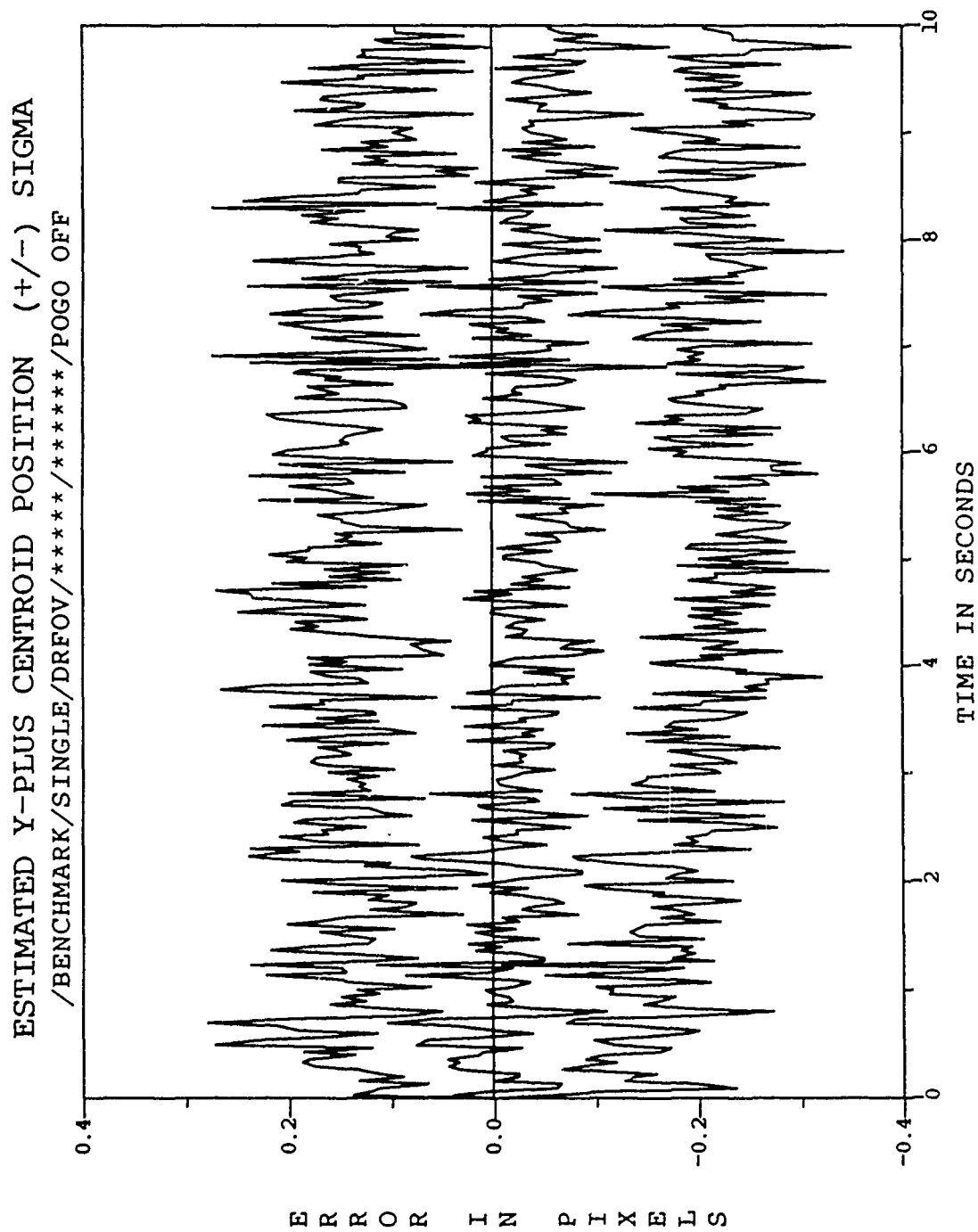


Figure B.20. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

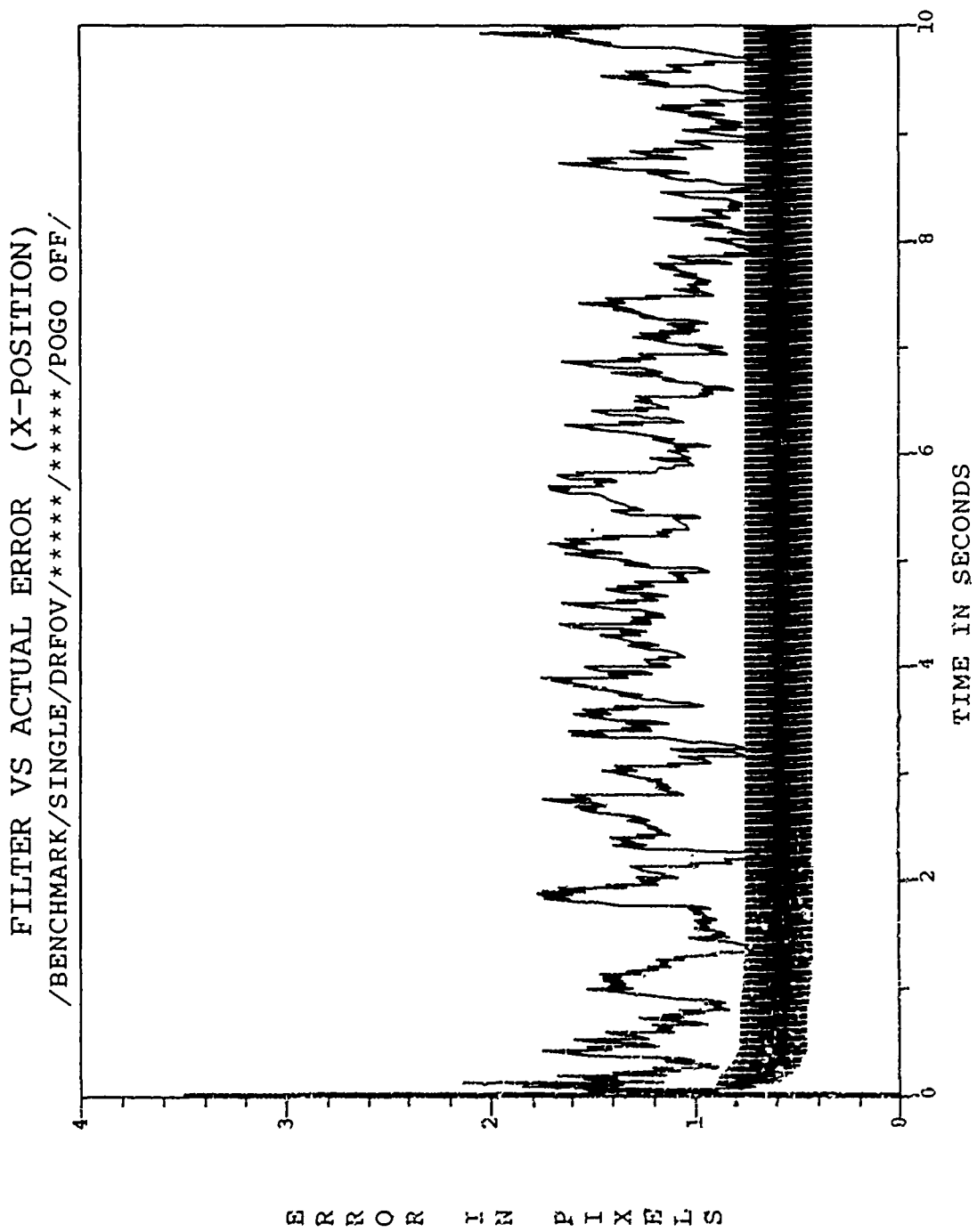


Figure B.21. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

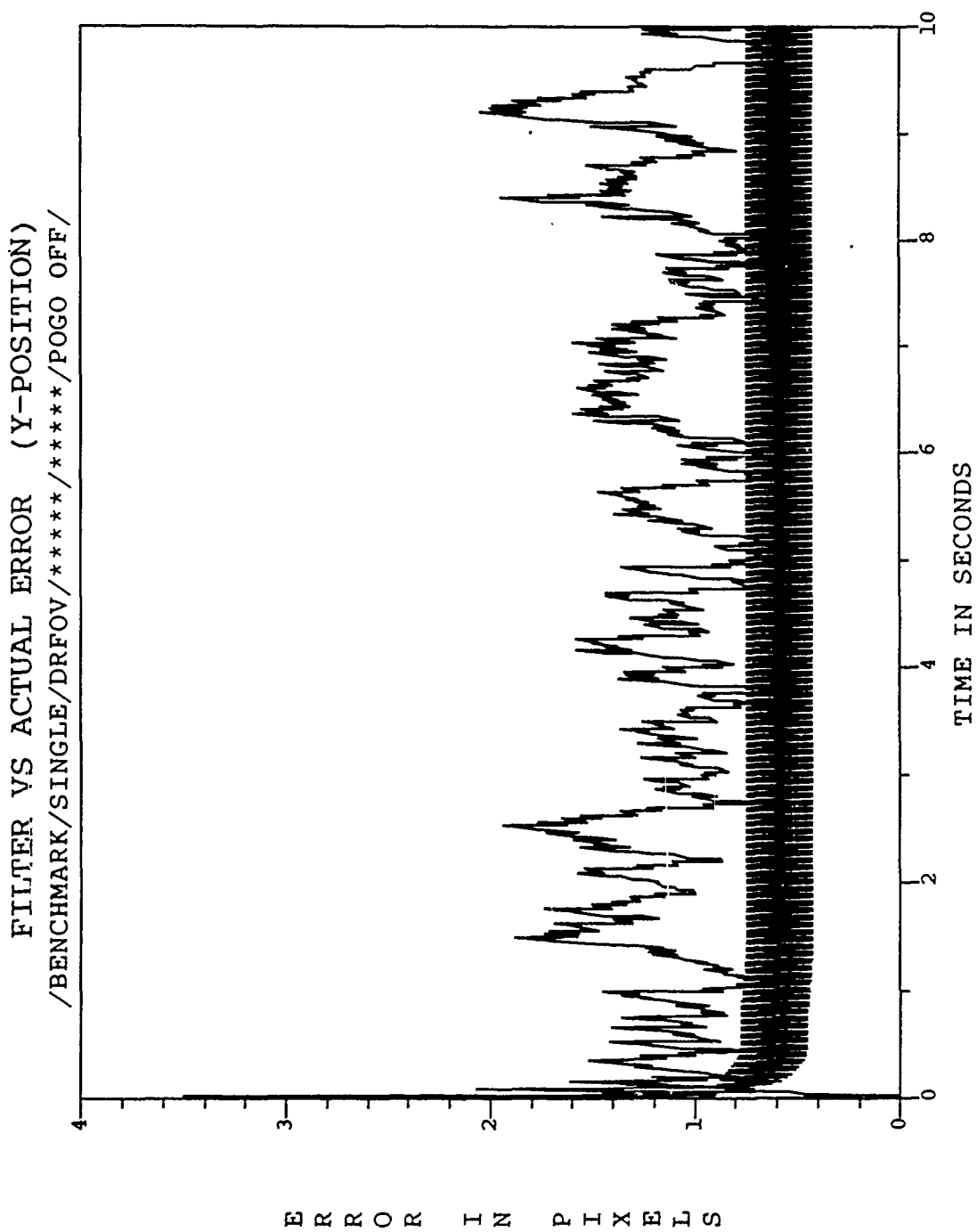


Figure B.22: /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

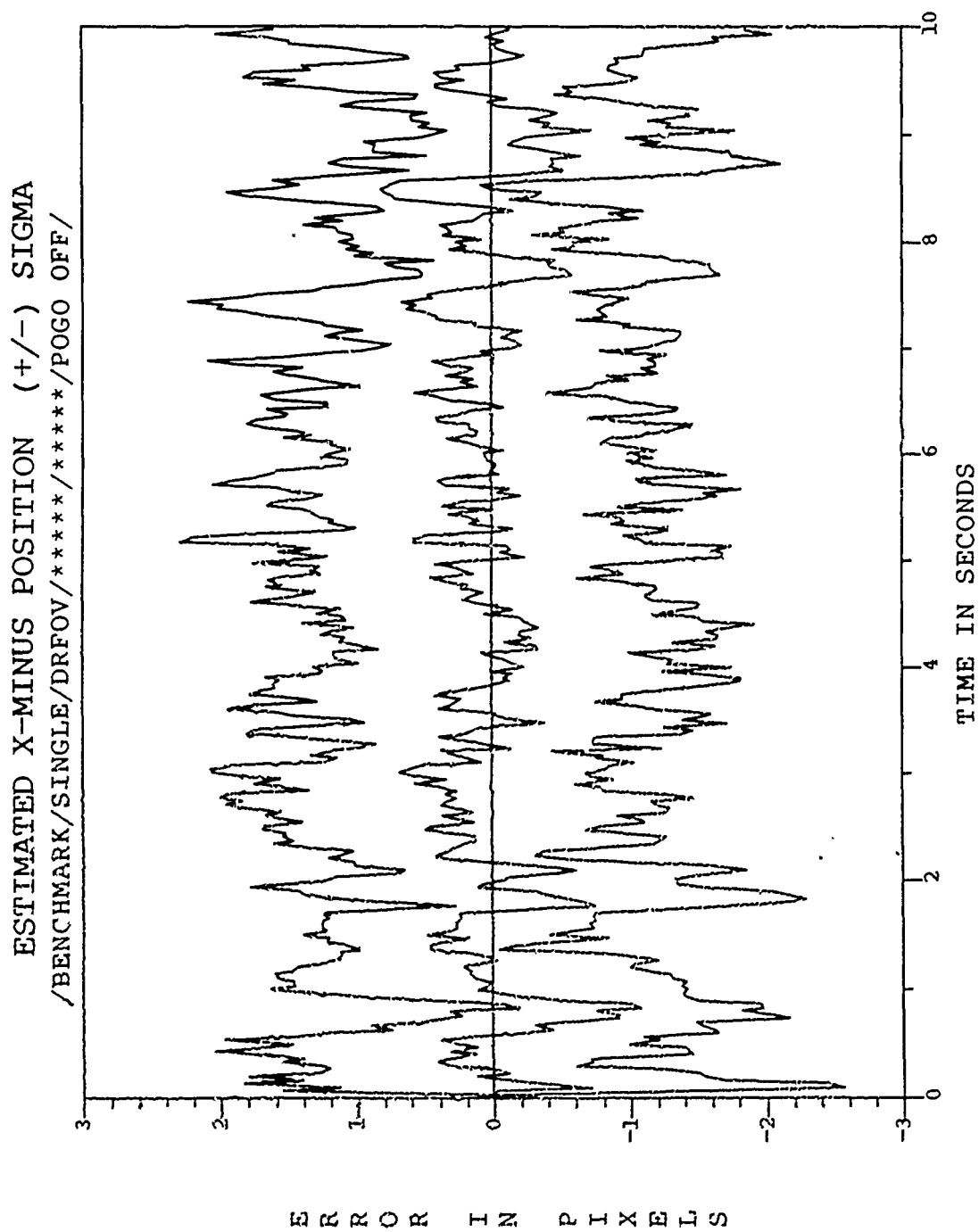


Figure B.23. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

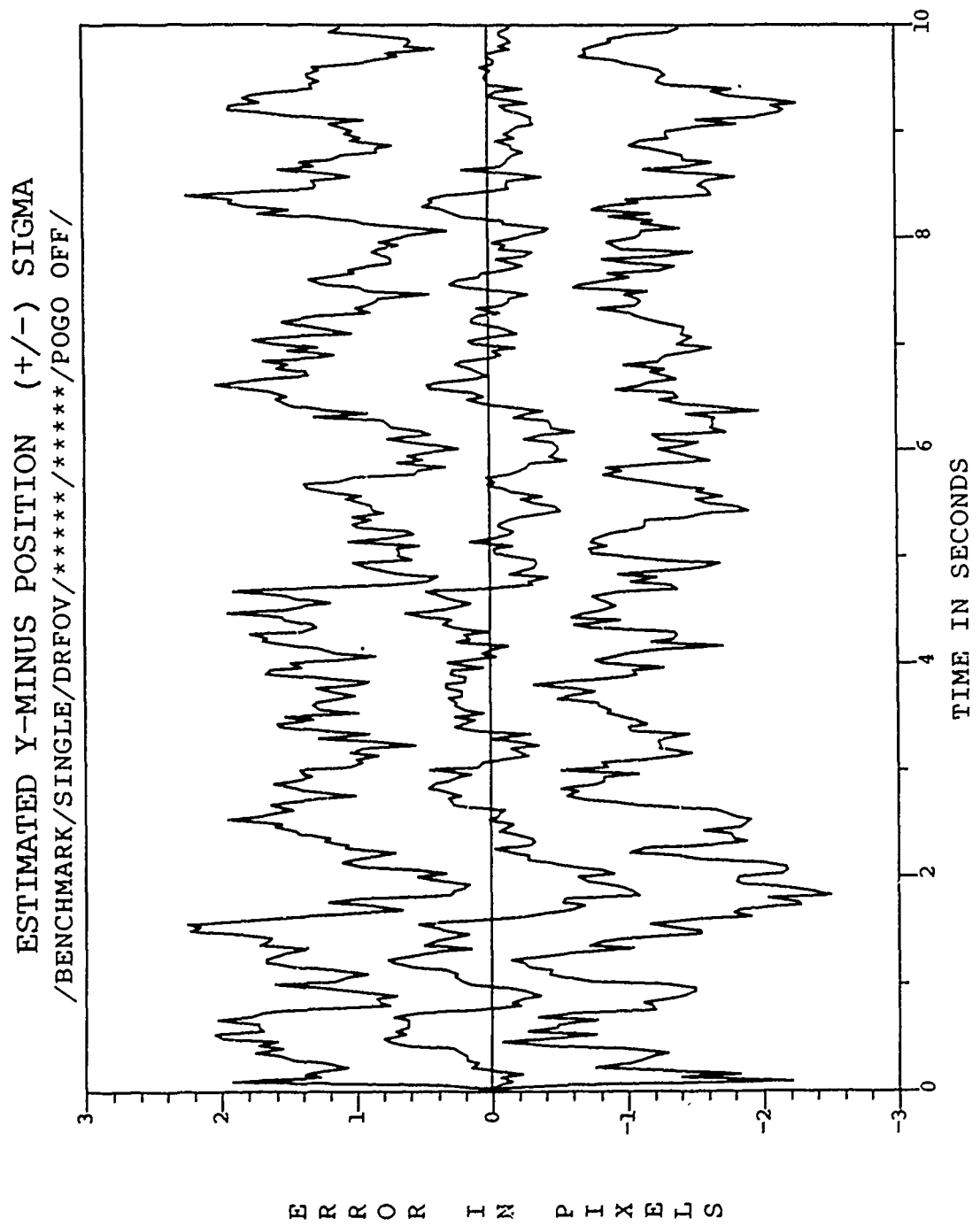


Figure B.24. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

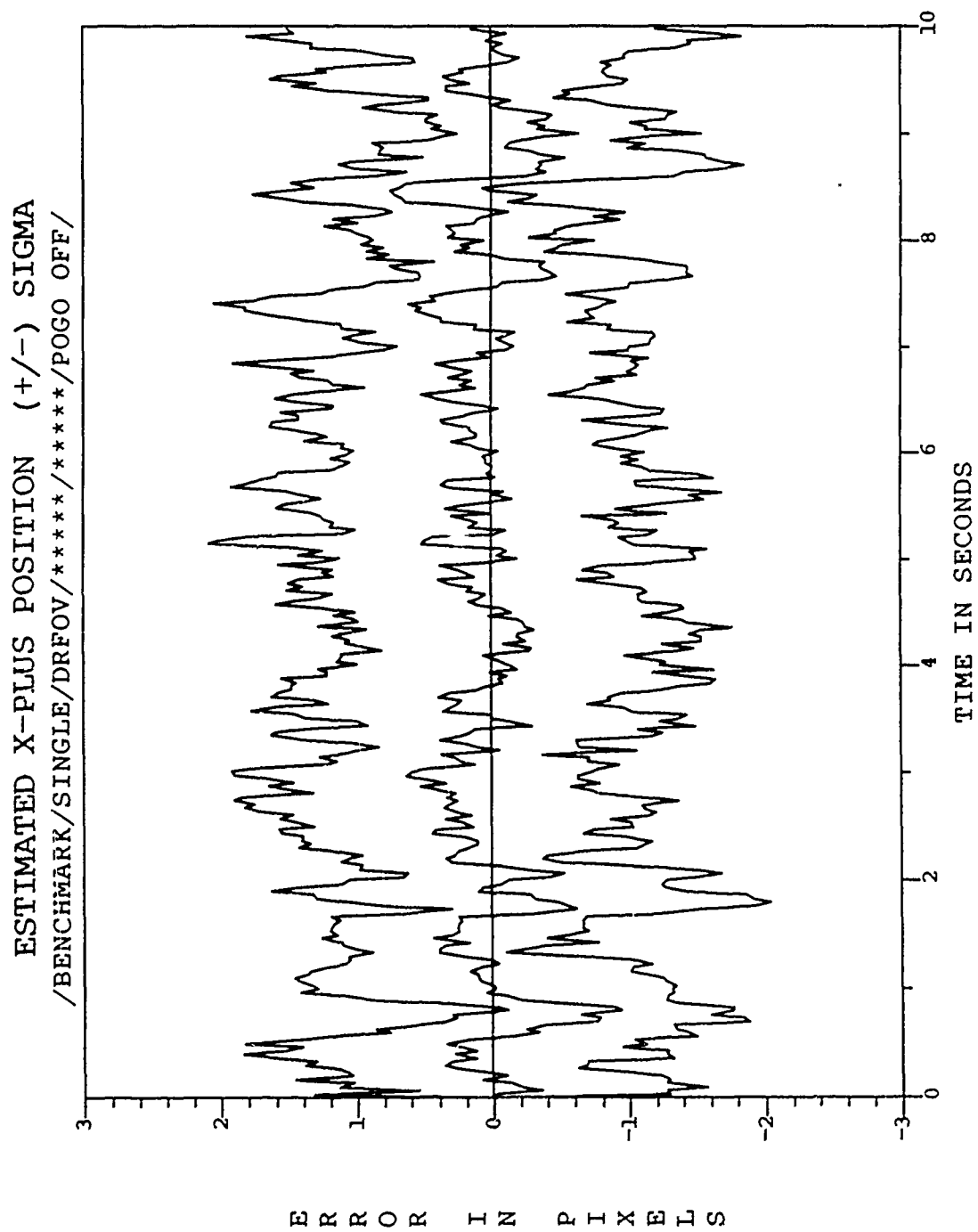


Figure B.25. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

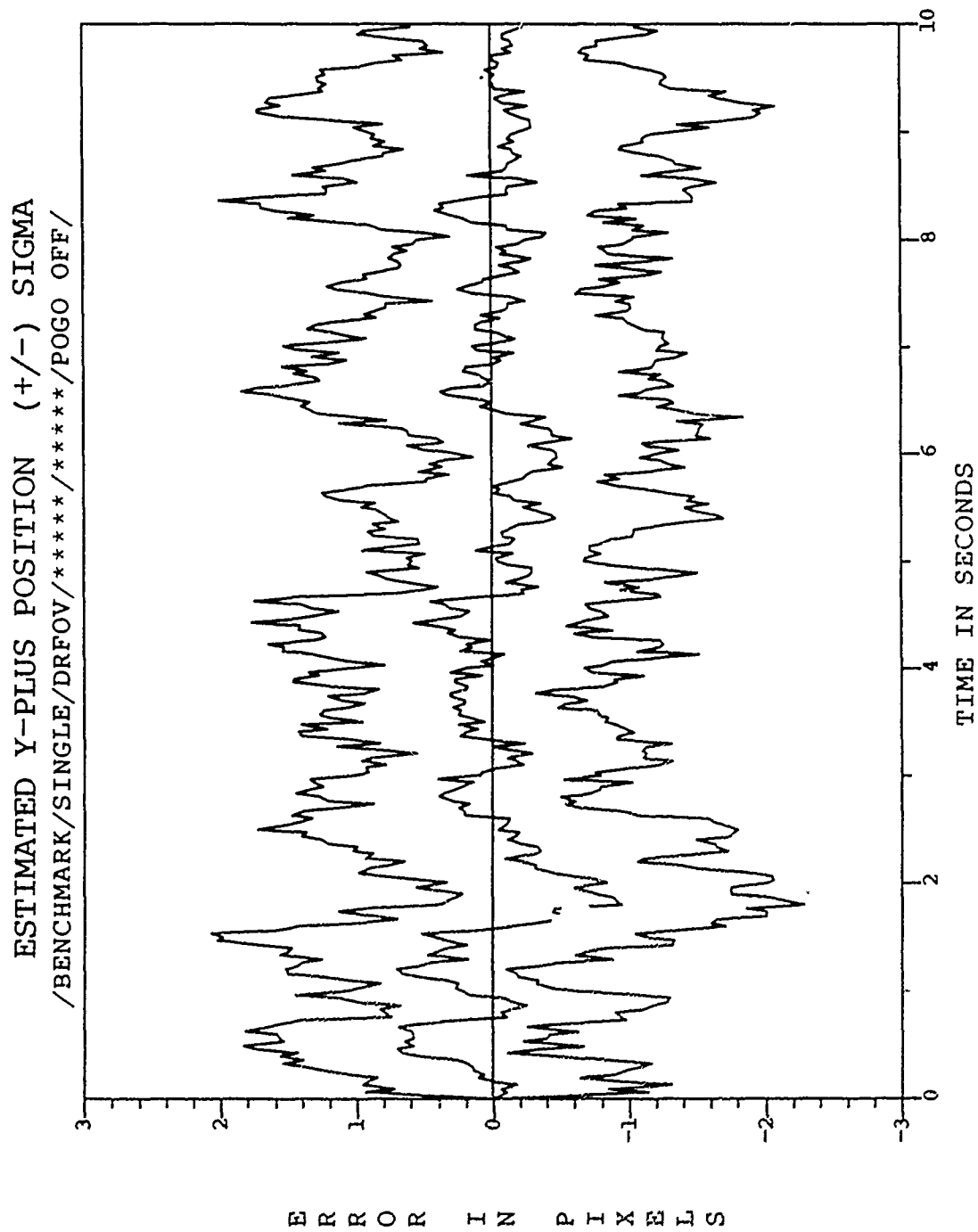


Figure B.26. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

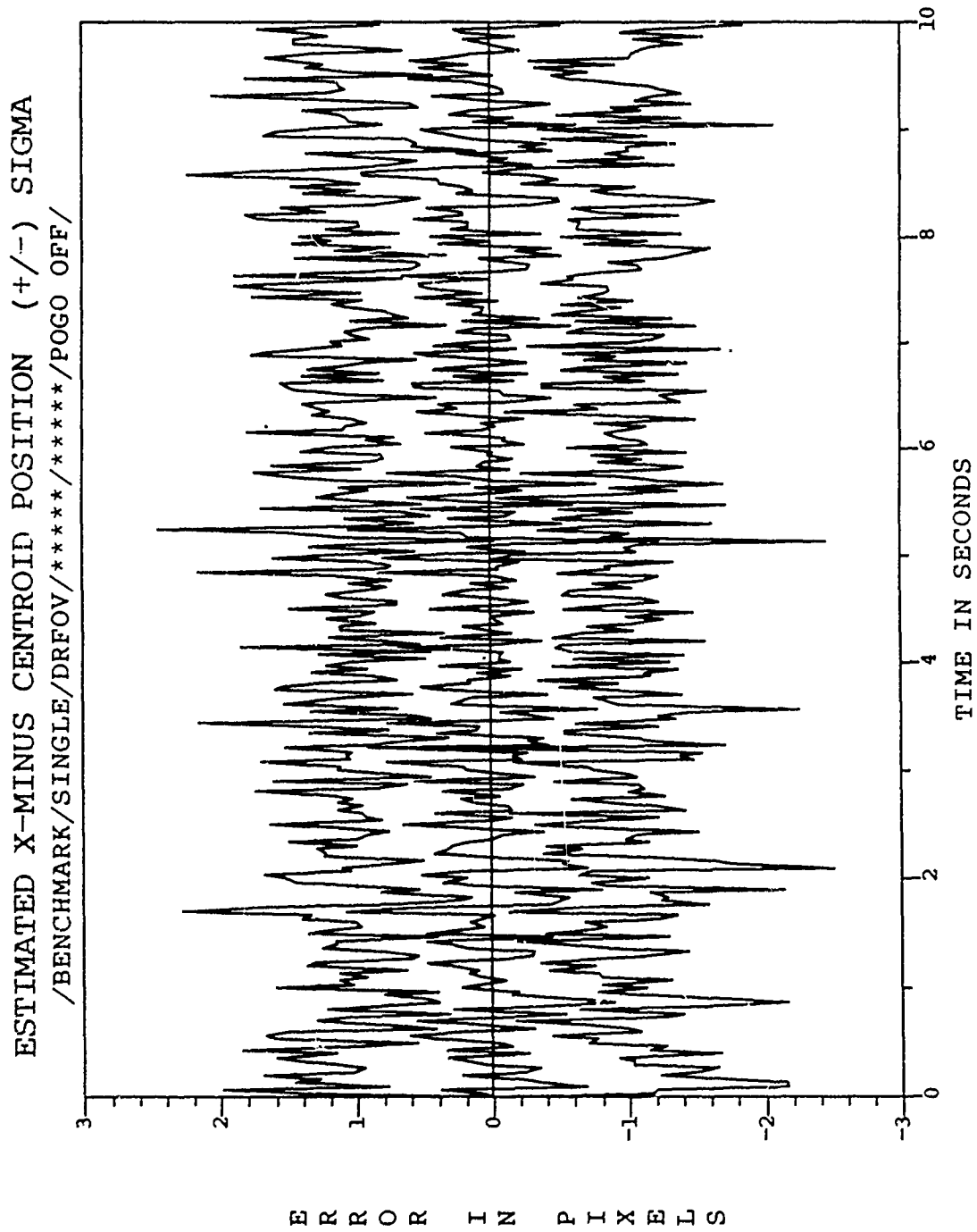


Figure B.27. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

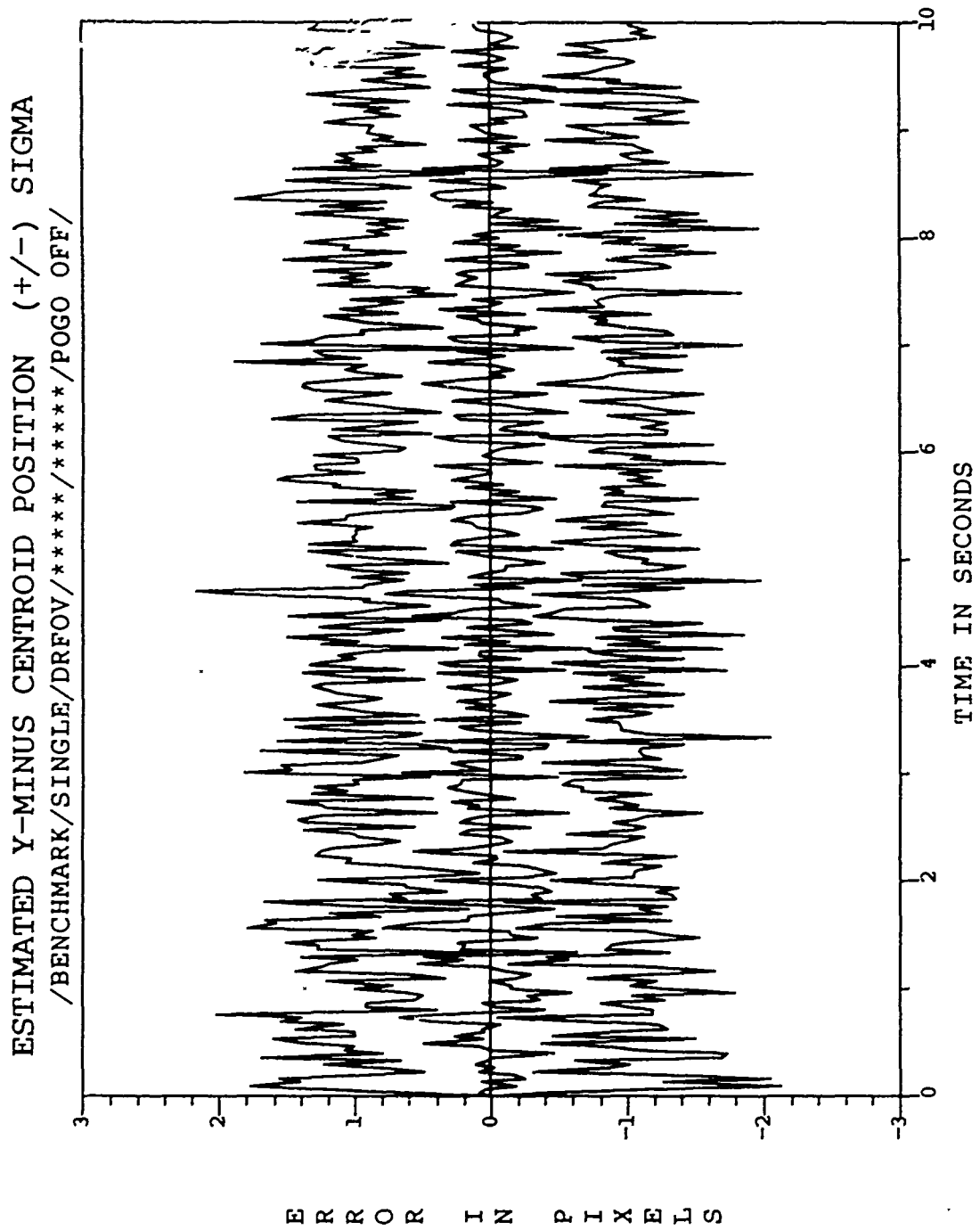


Figure B.28. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

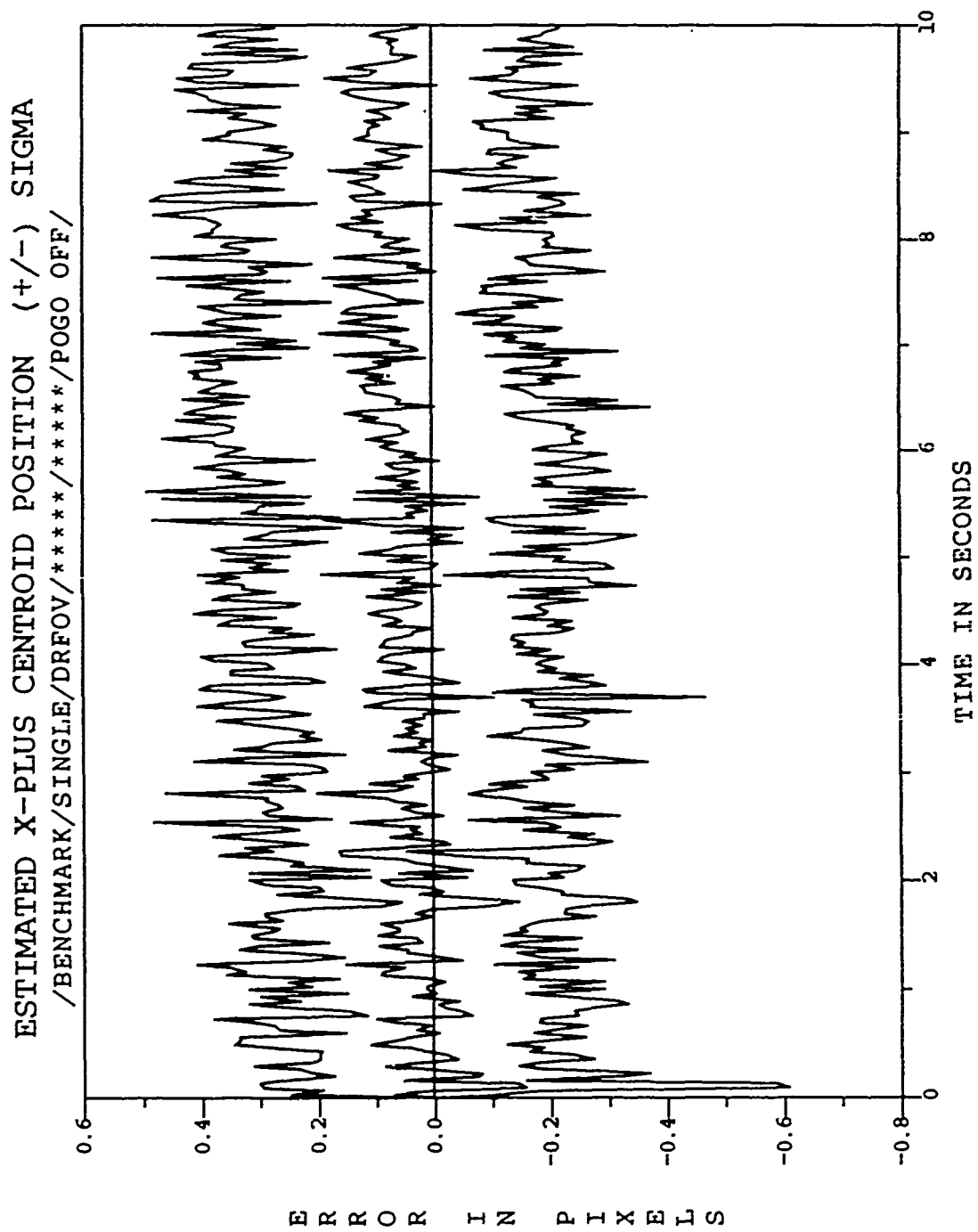


Figure B.29. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

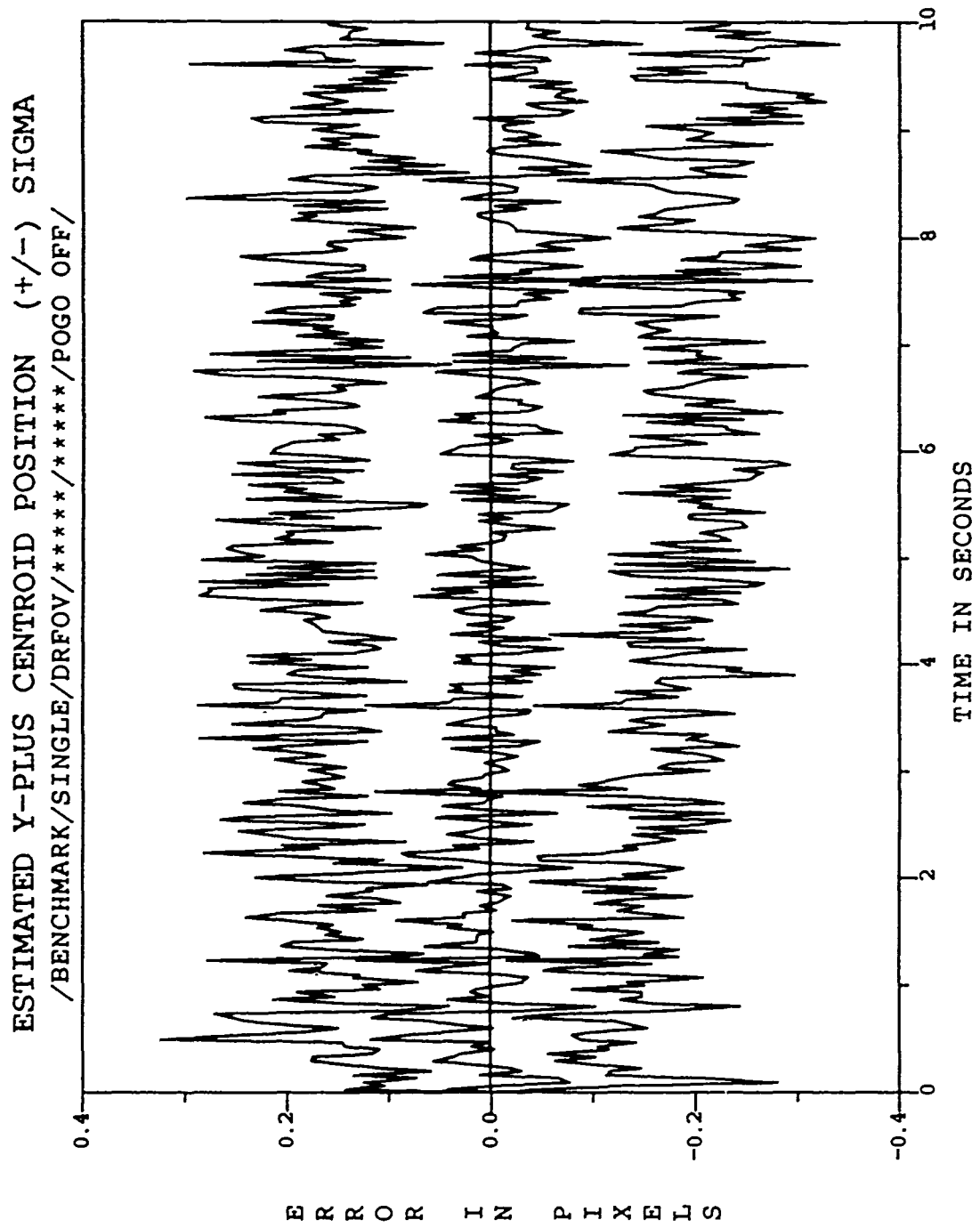


Figure B.30. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

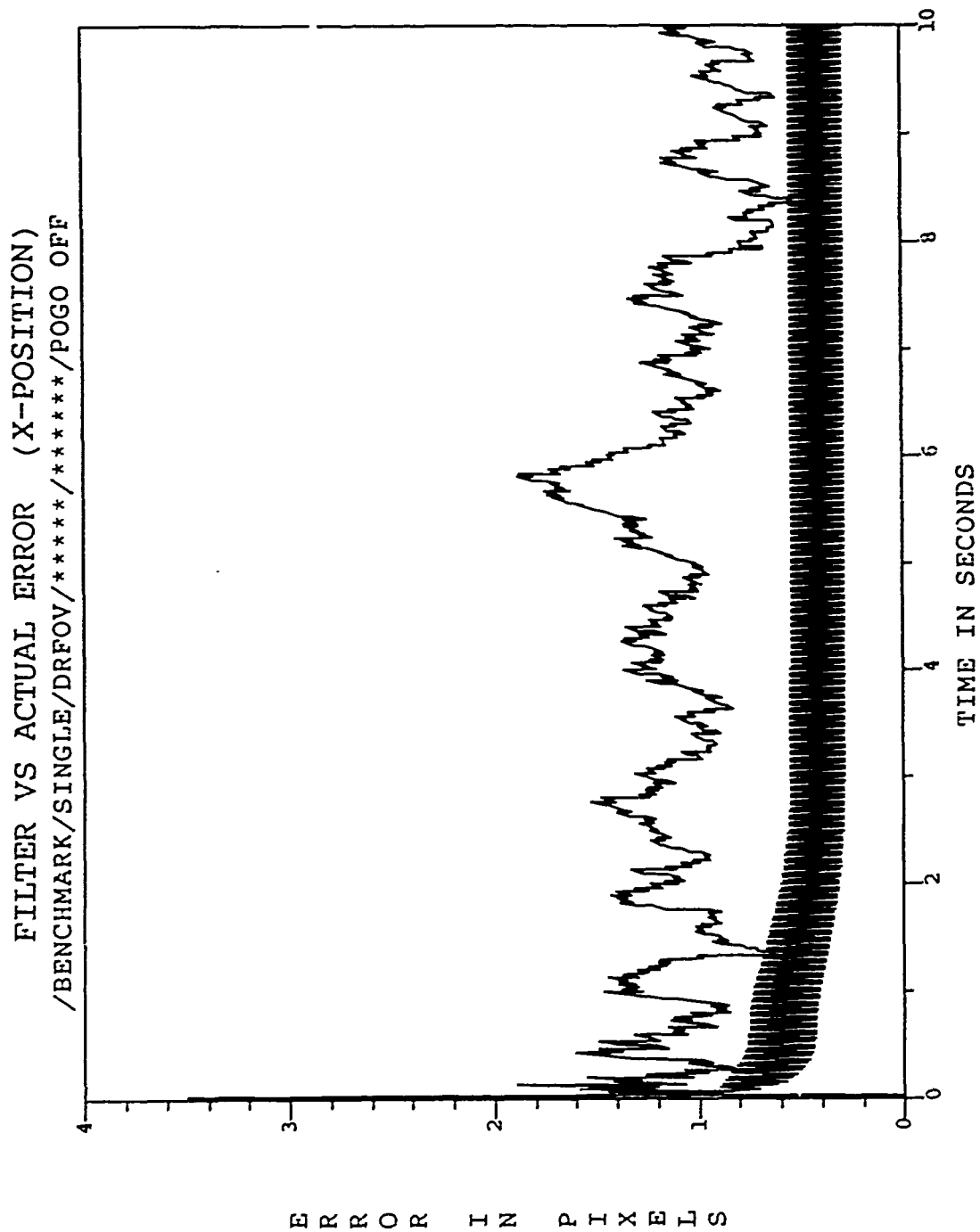


Figure B.31. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

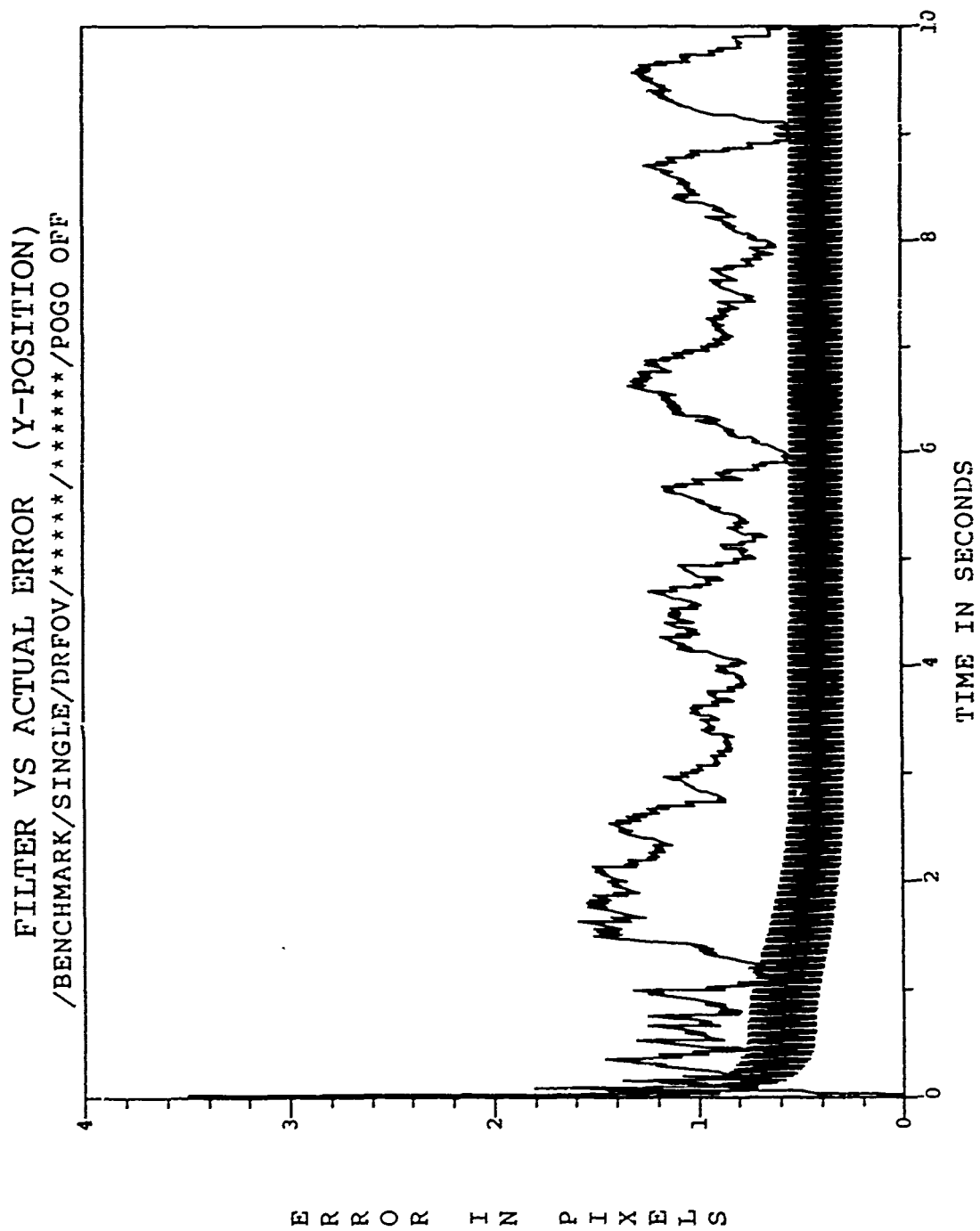


Figure B.32. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

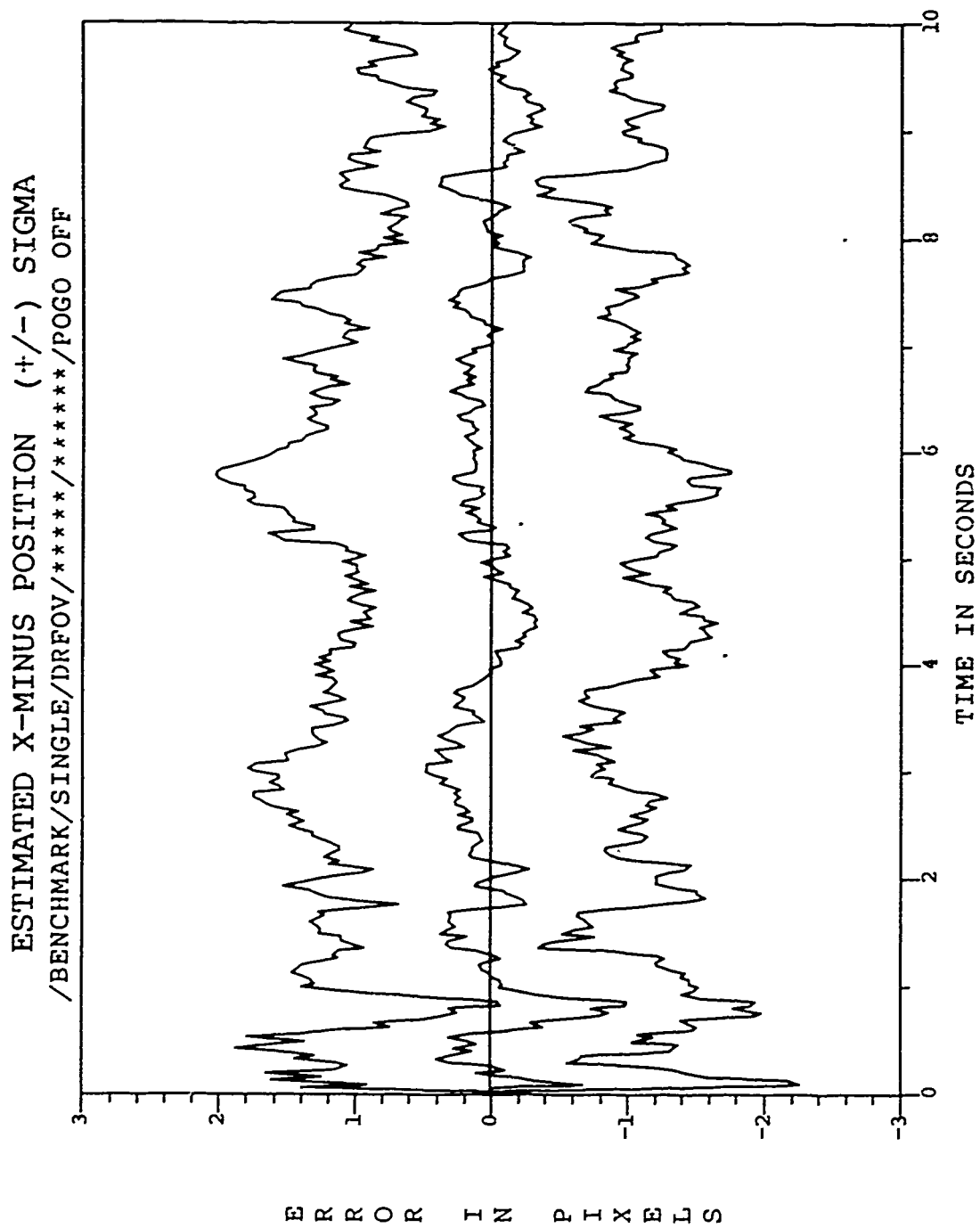


Figure B.33. /BENCHMARK/SINGLE/DRFOV/****/*****/POGO OFF/

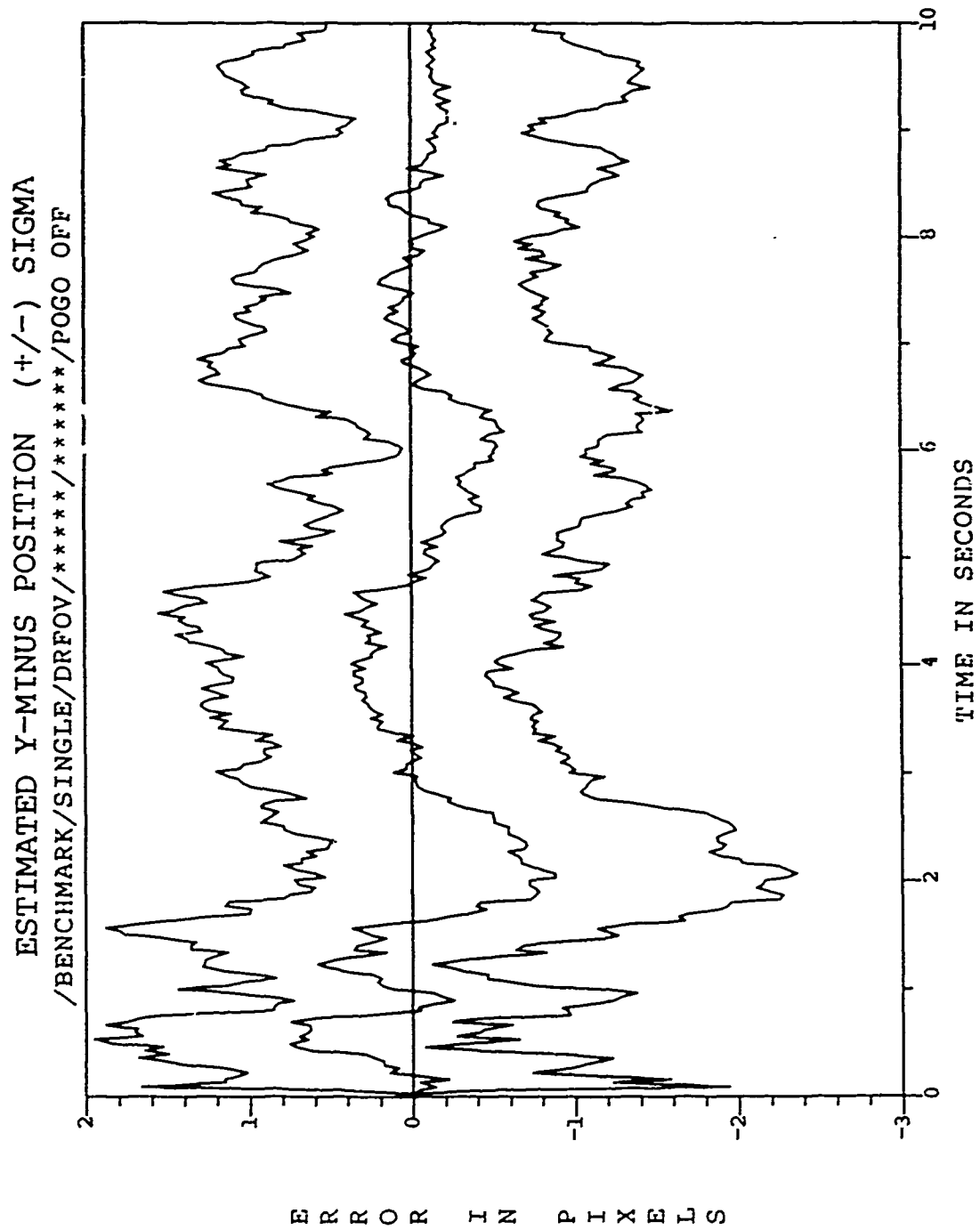


Figure B.34. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

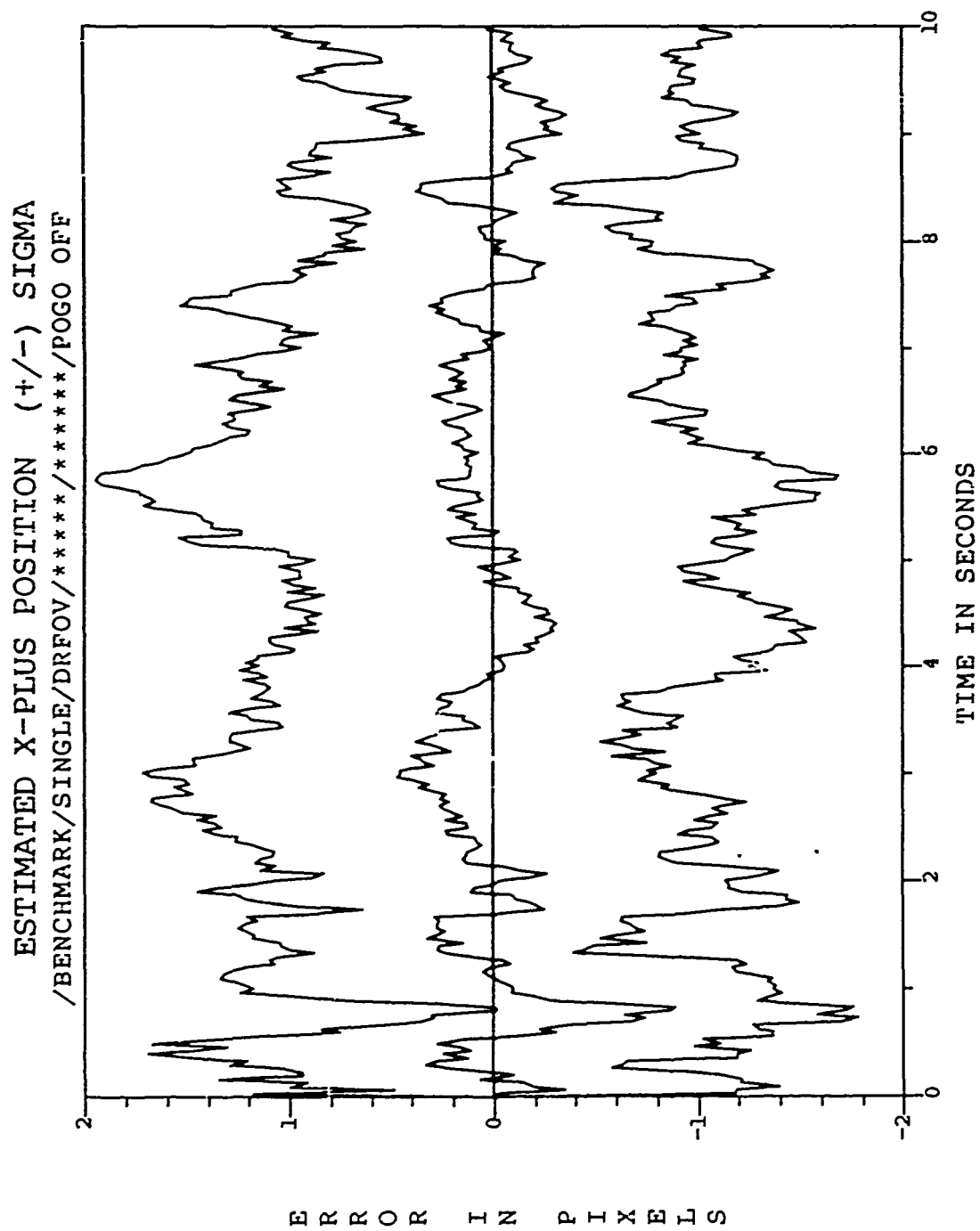


Figure B.35. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

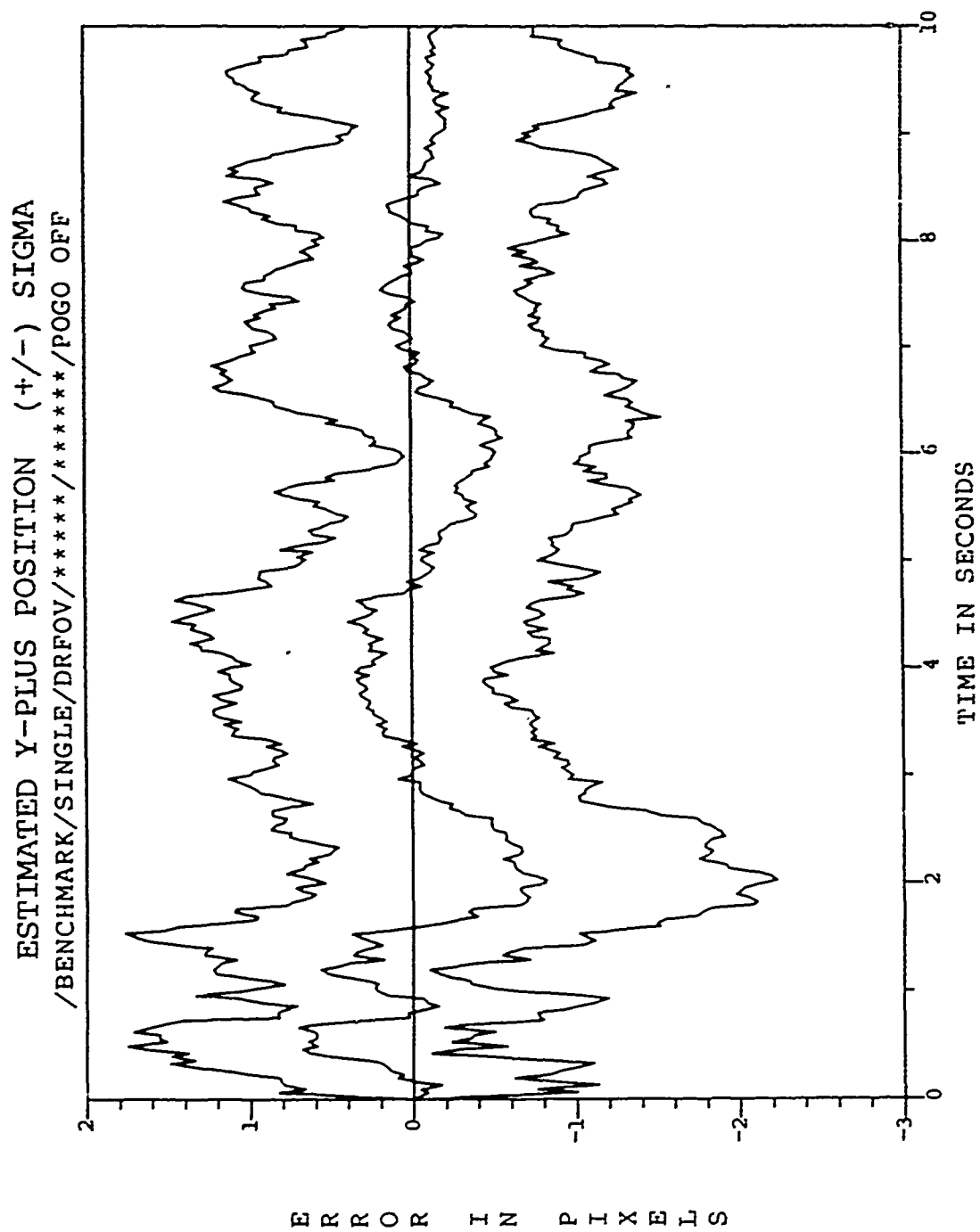


Figure B.36. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

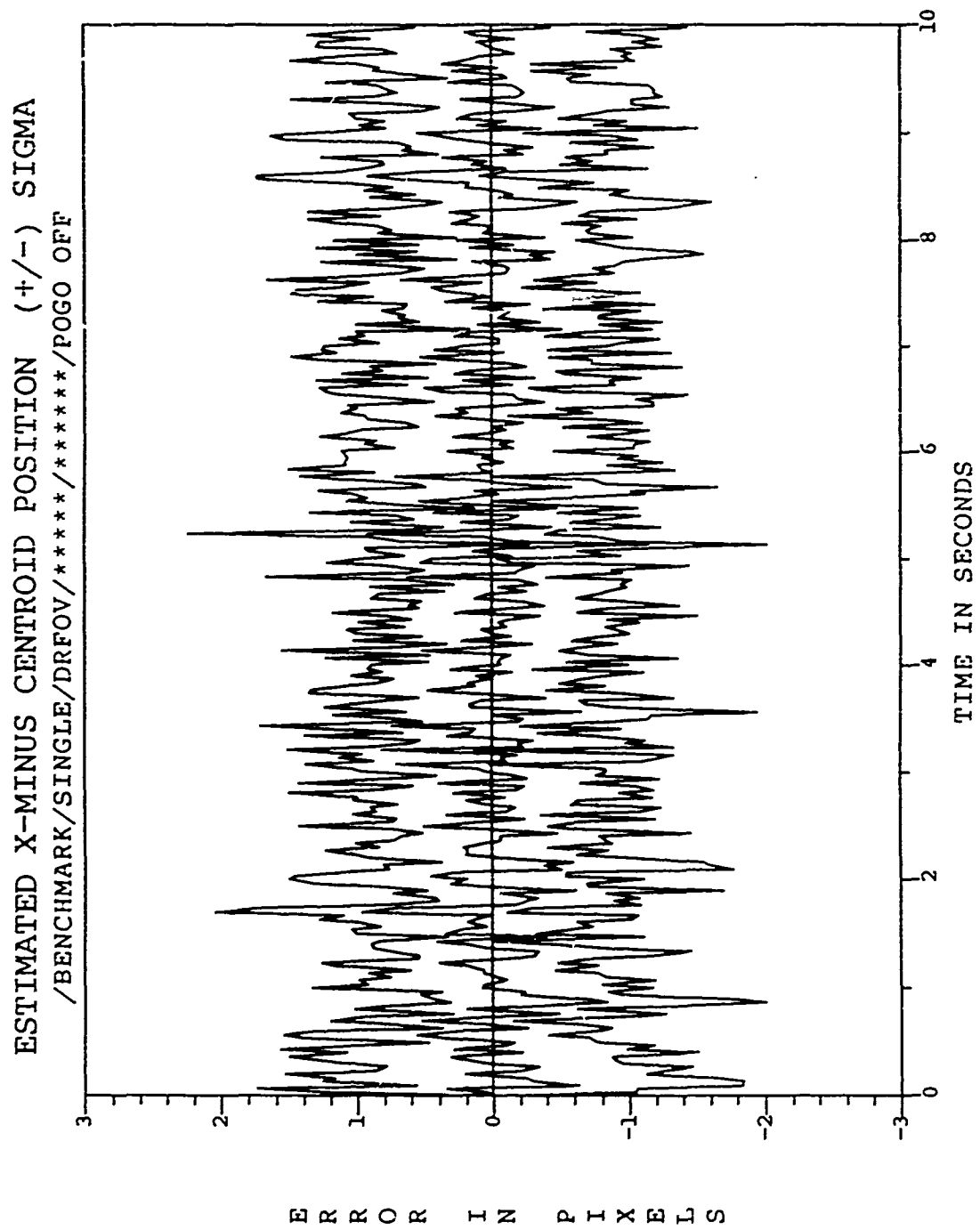


Figure B.37. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

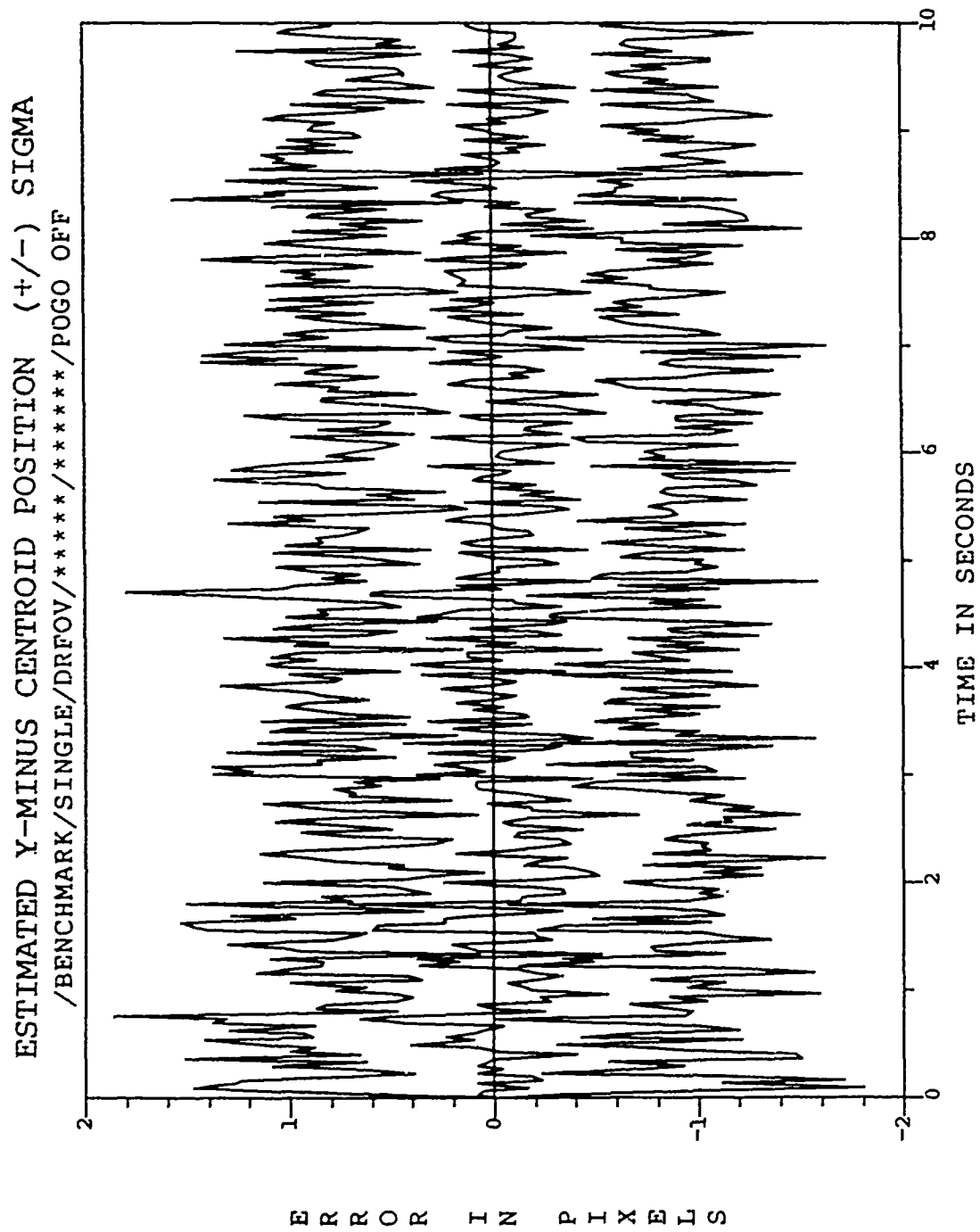


Figure B.38. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

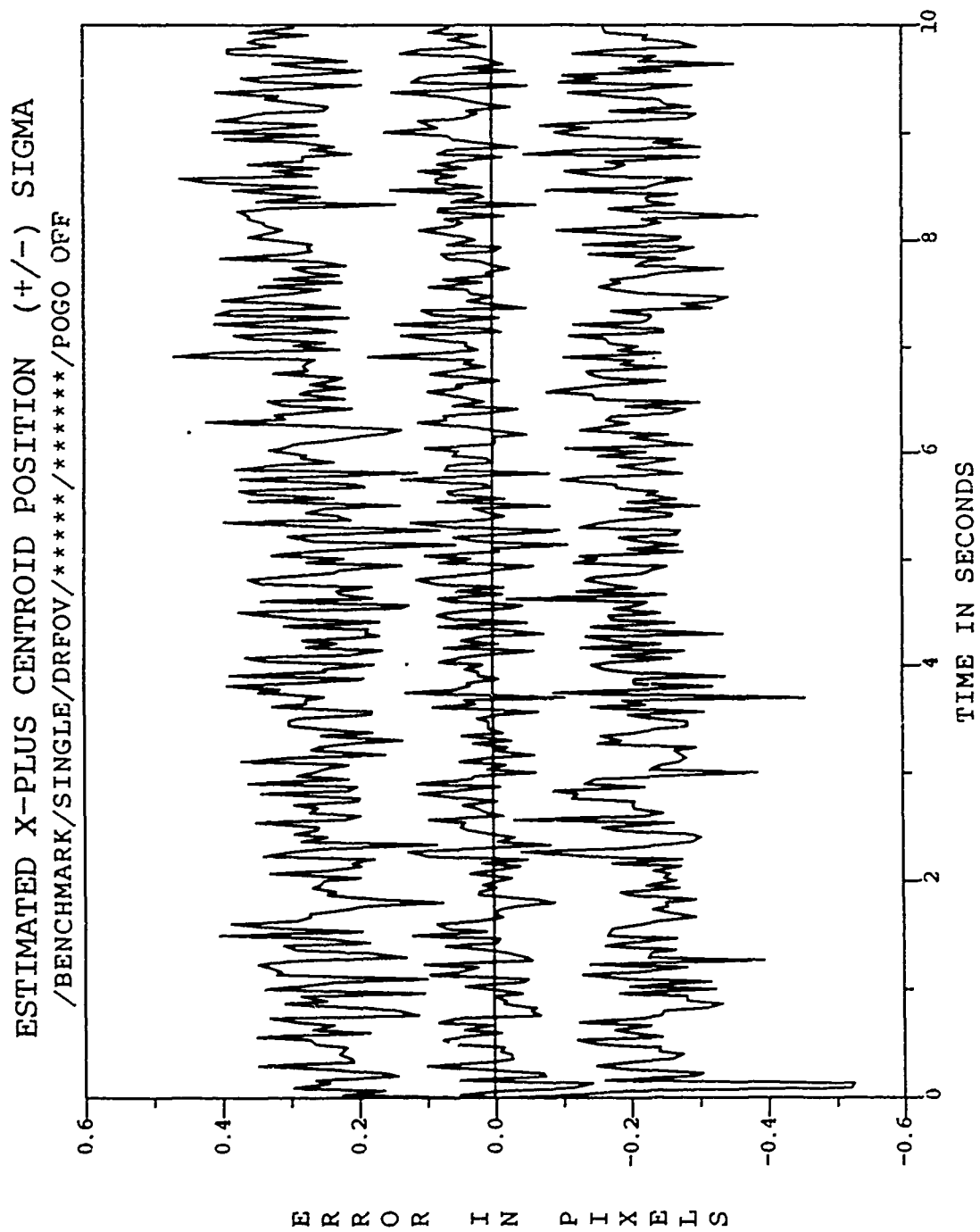


Figure B.39. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

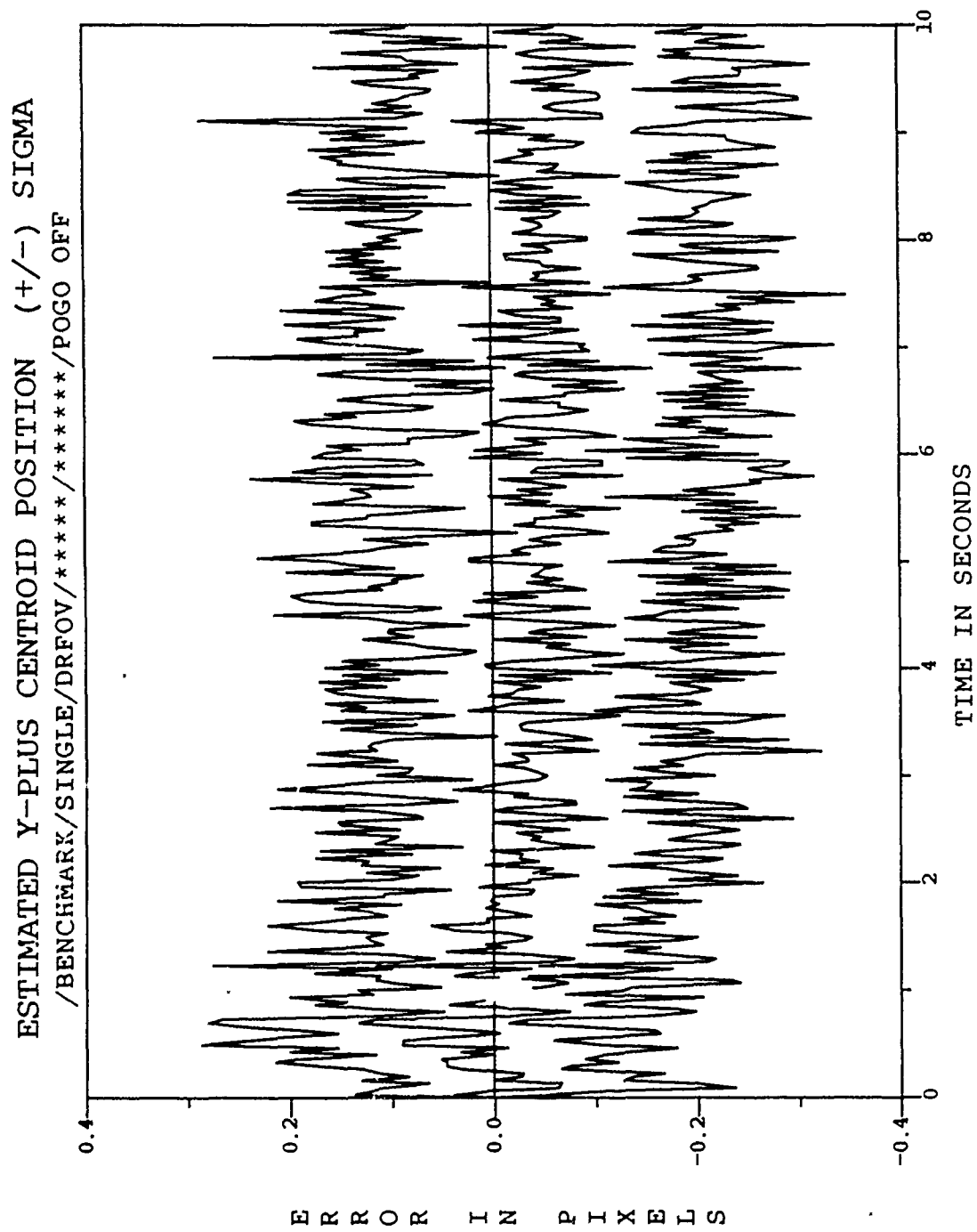


Figure B.10. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

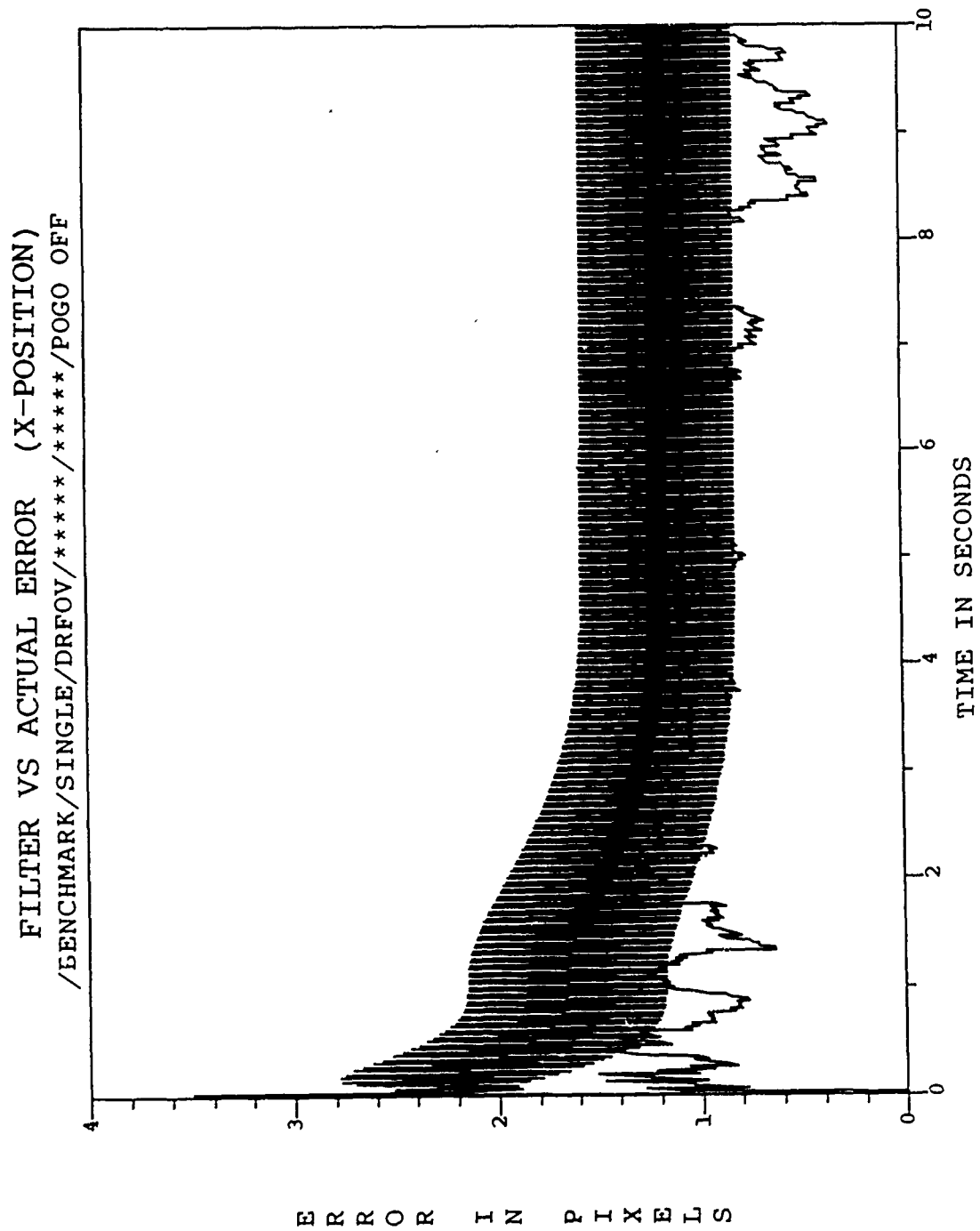


Figure B.41. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

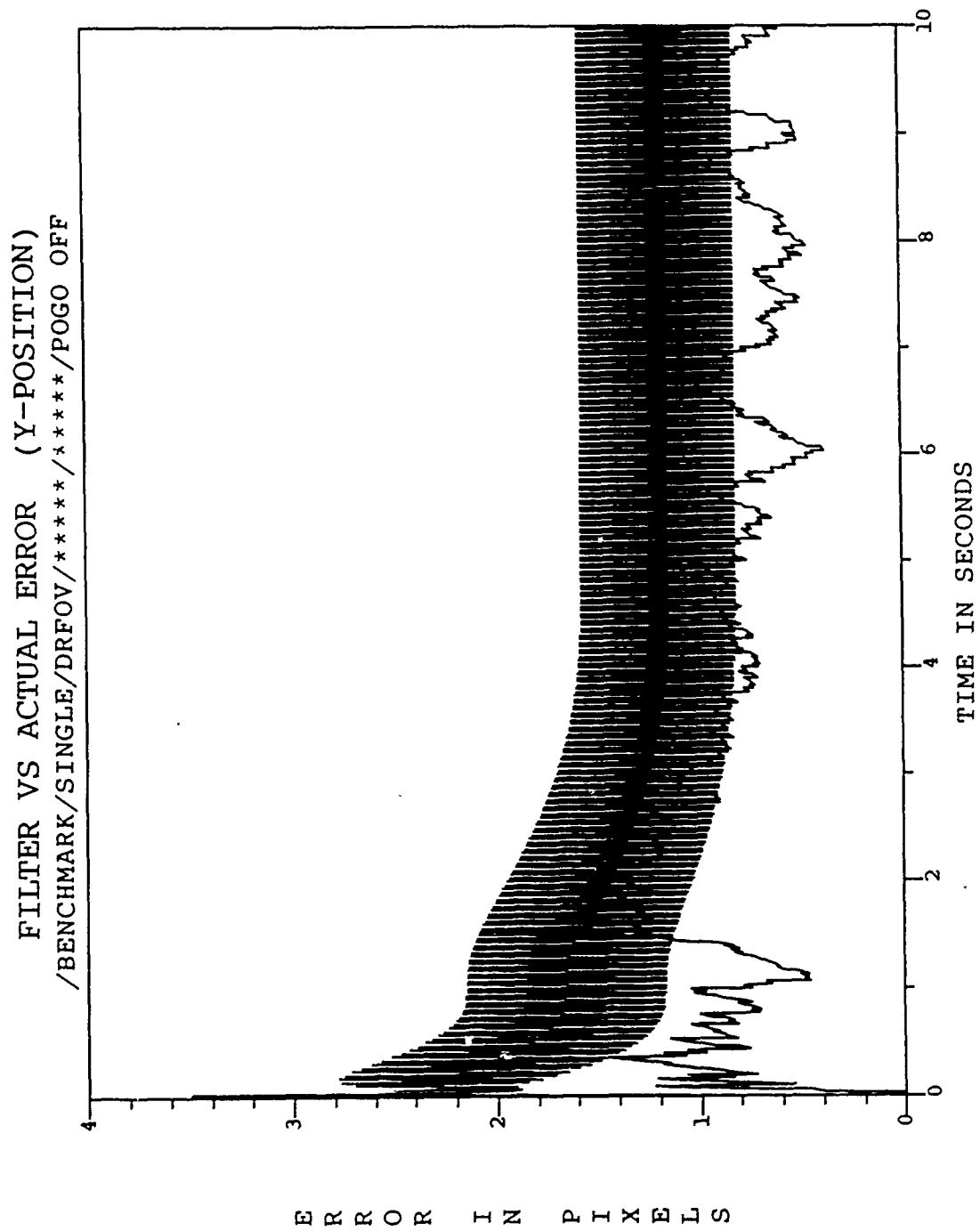


Figure B.42. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

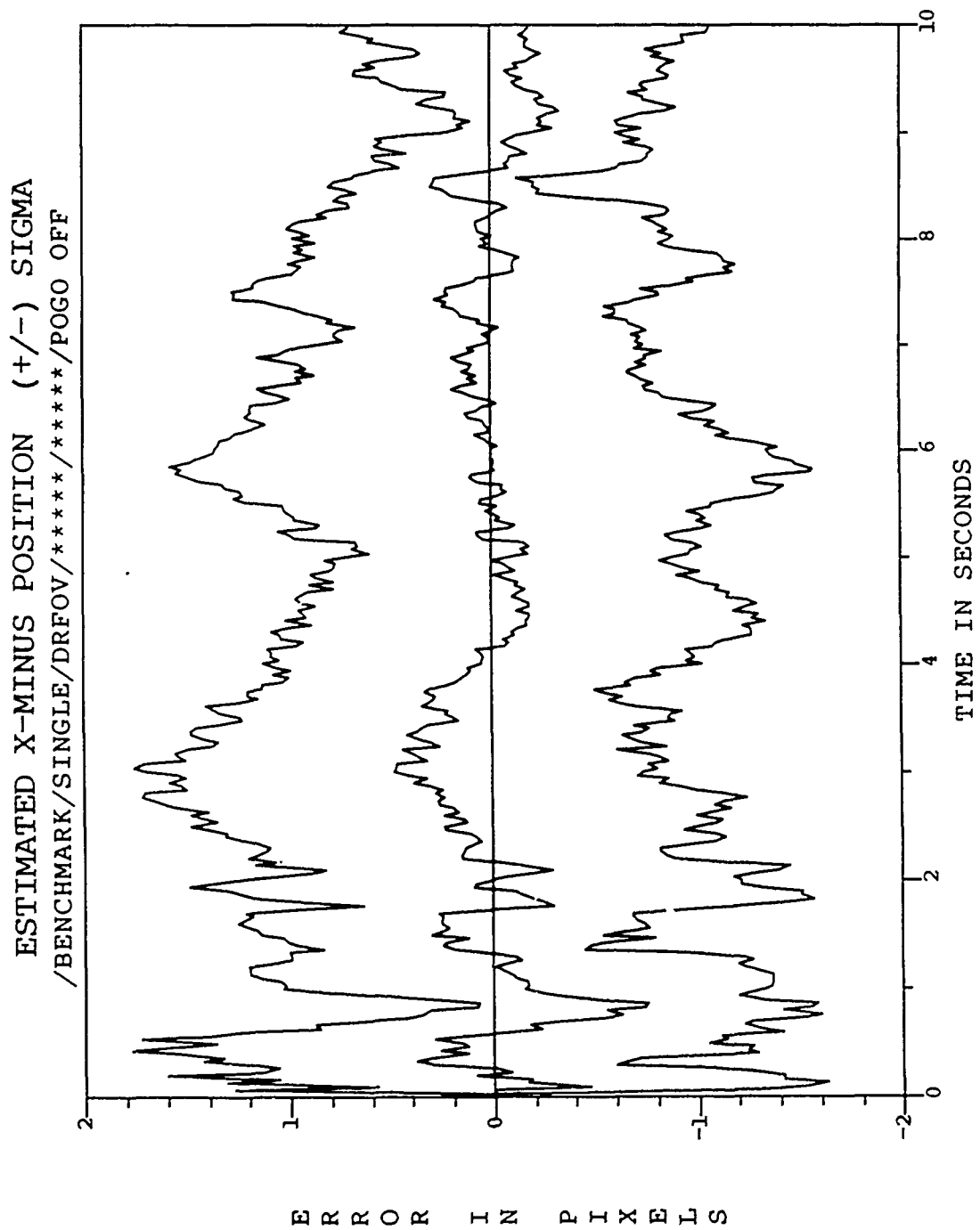


Figure B.43. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

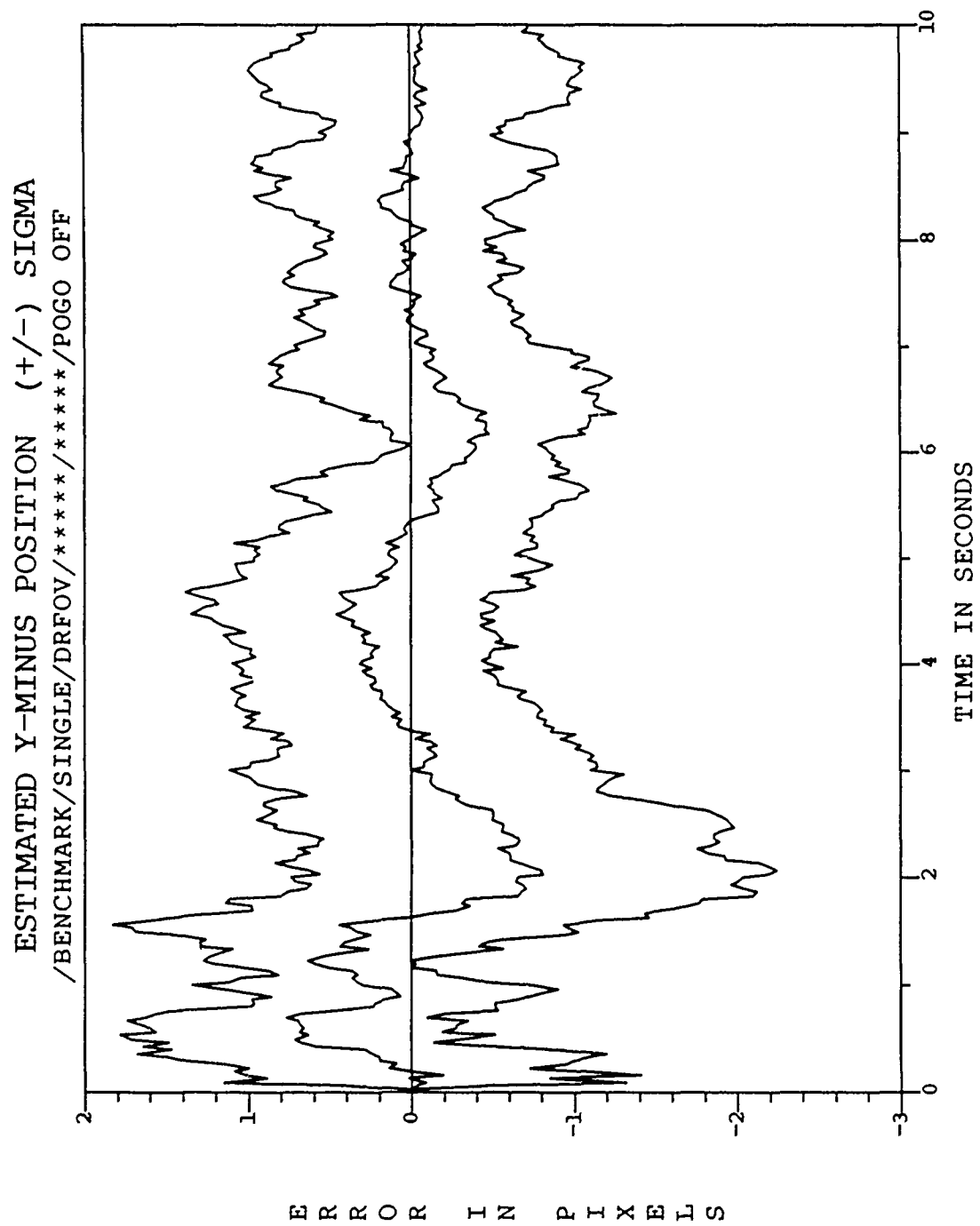


Figure B.44. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

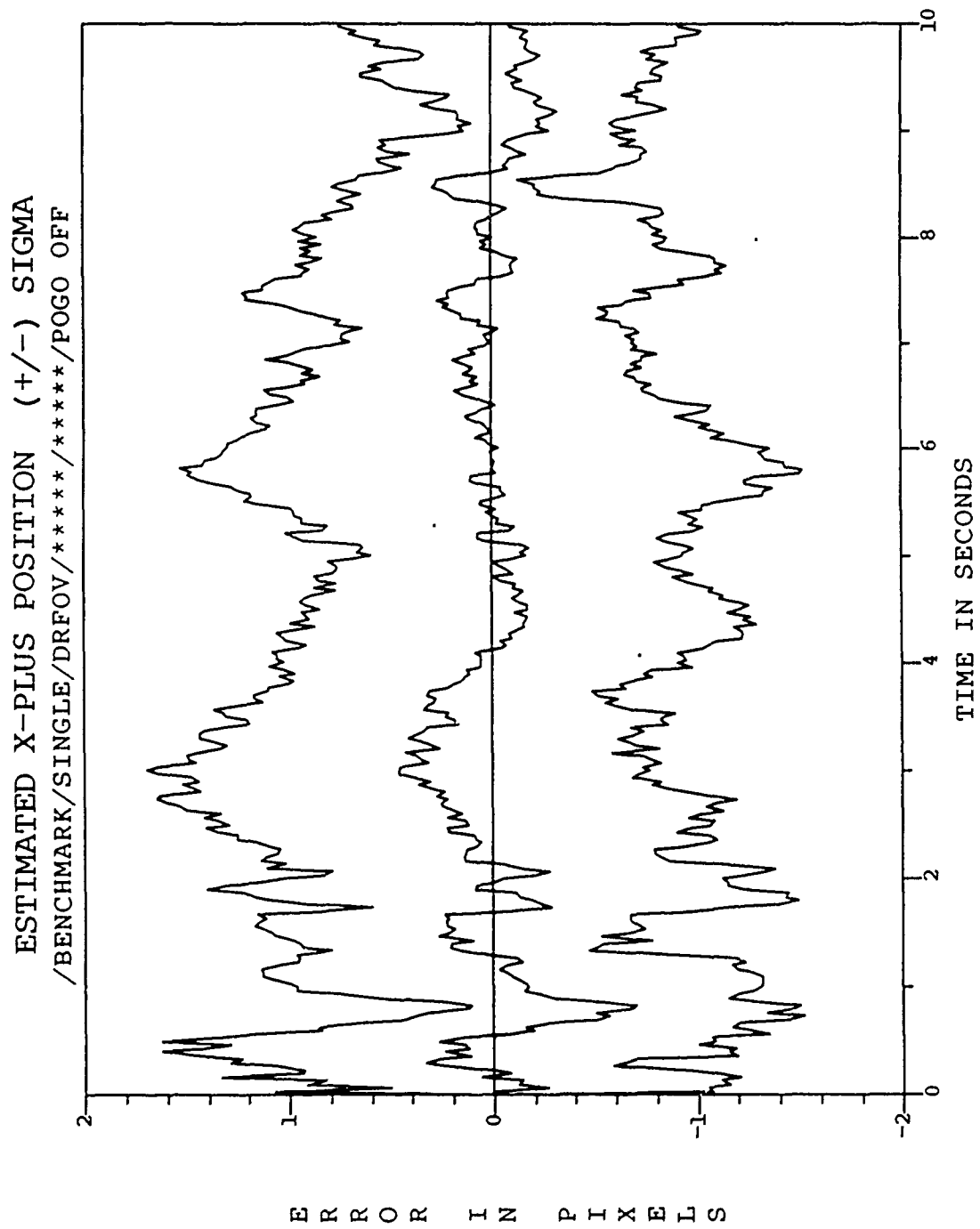


Figure B.45. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

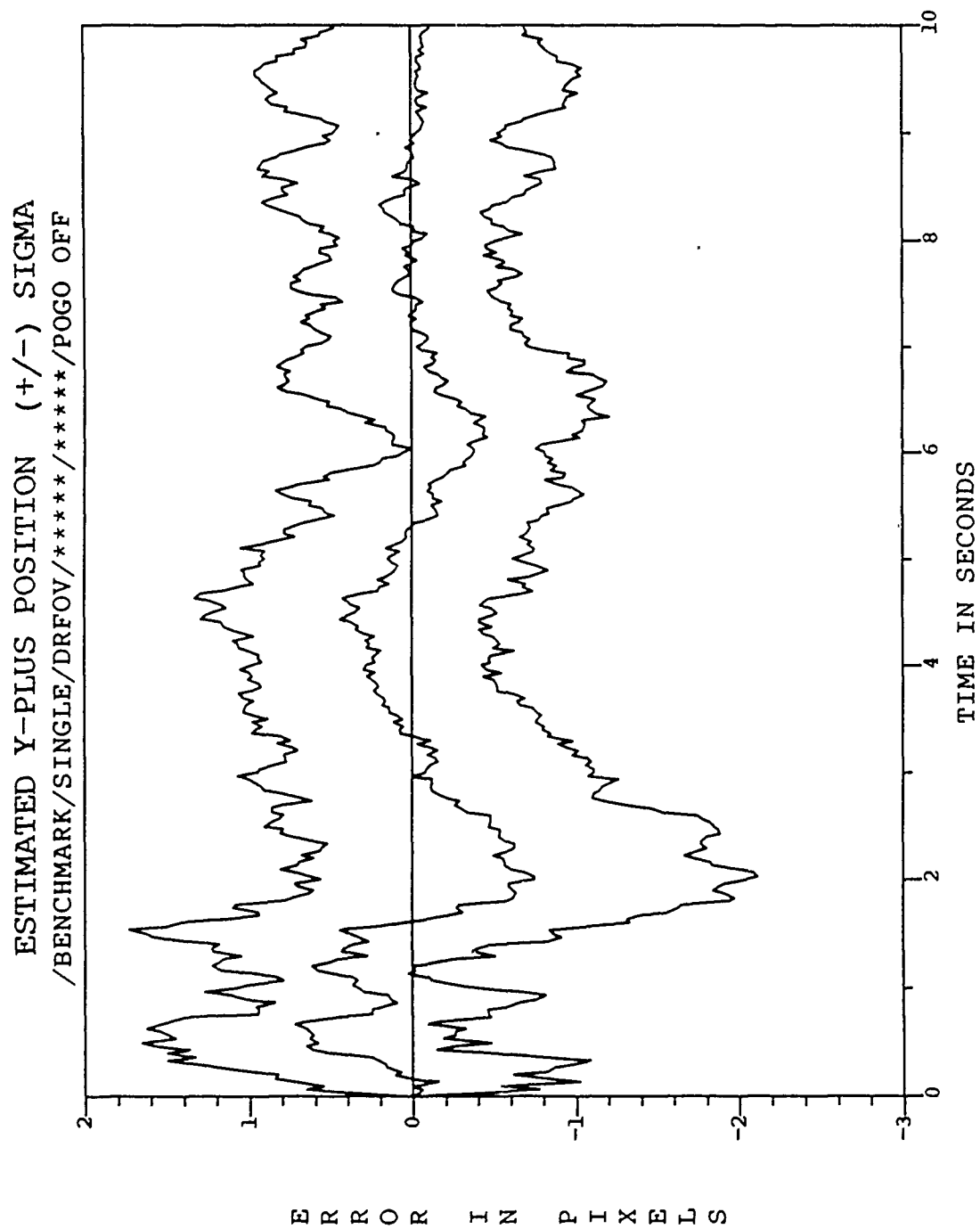


Figure B.46. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

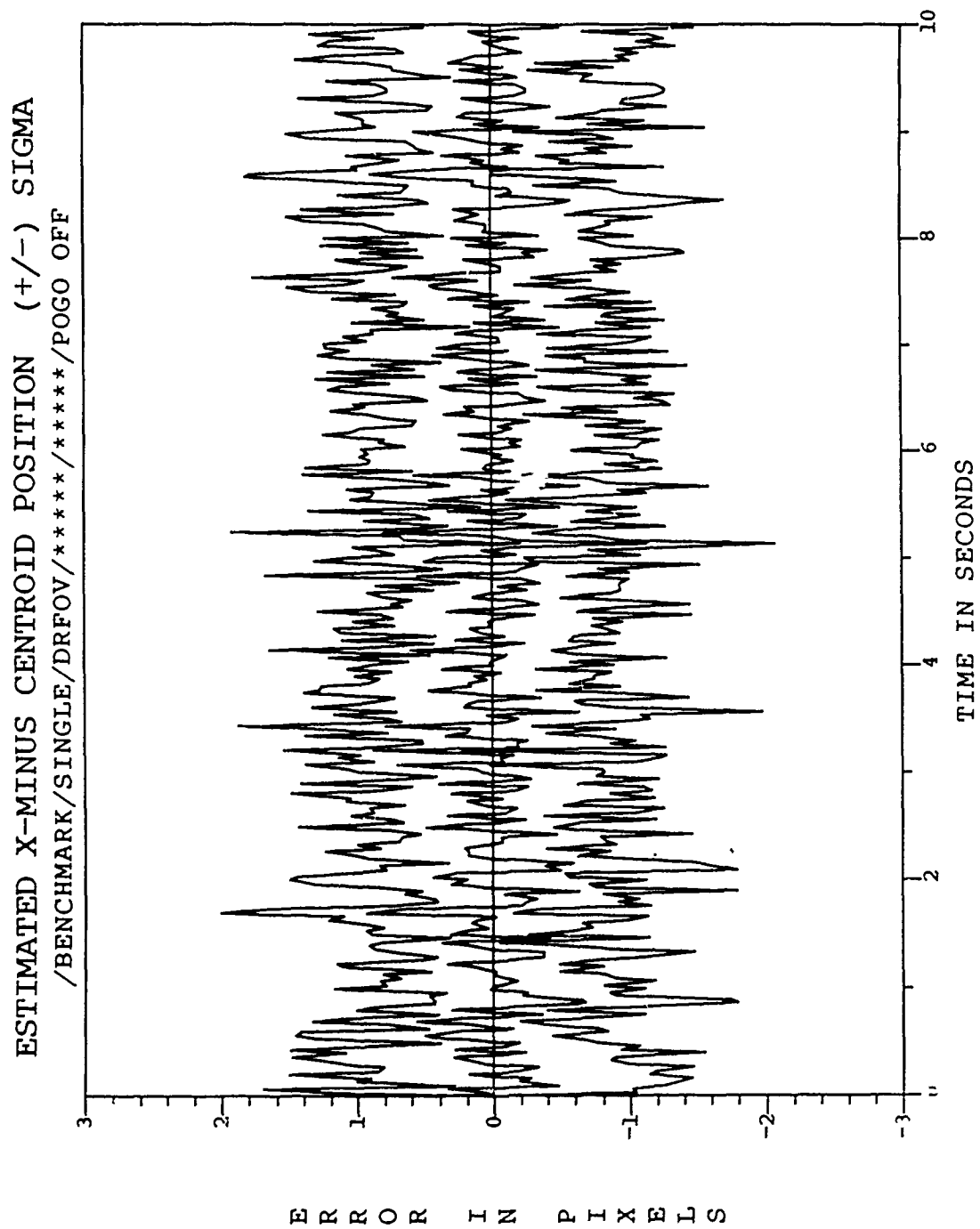


Figure B.47. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

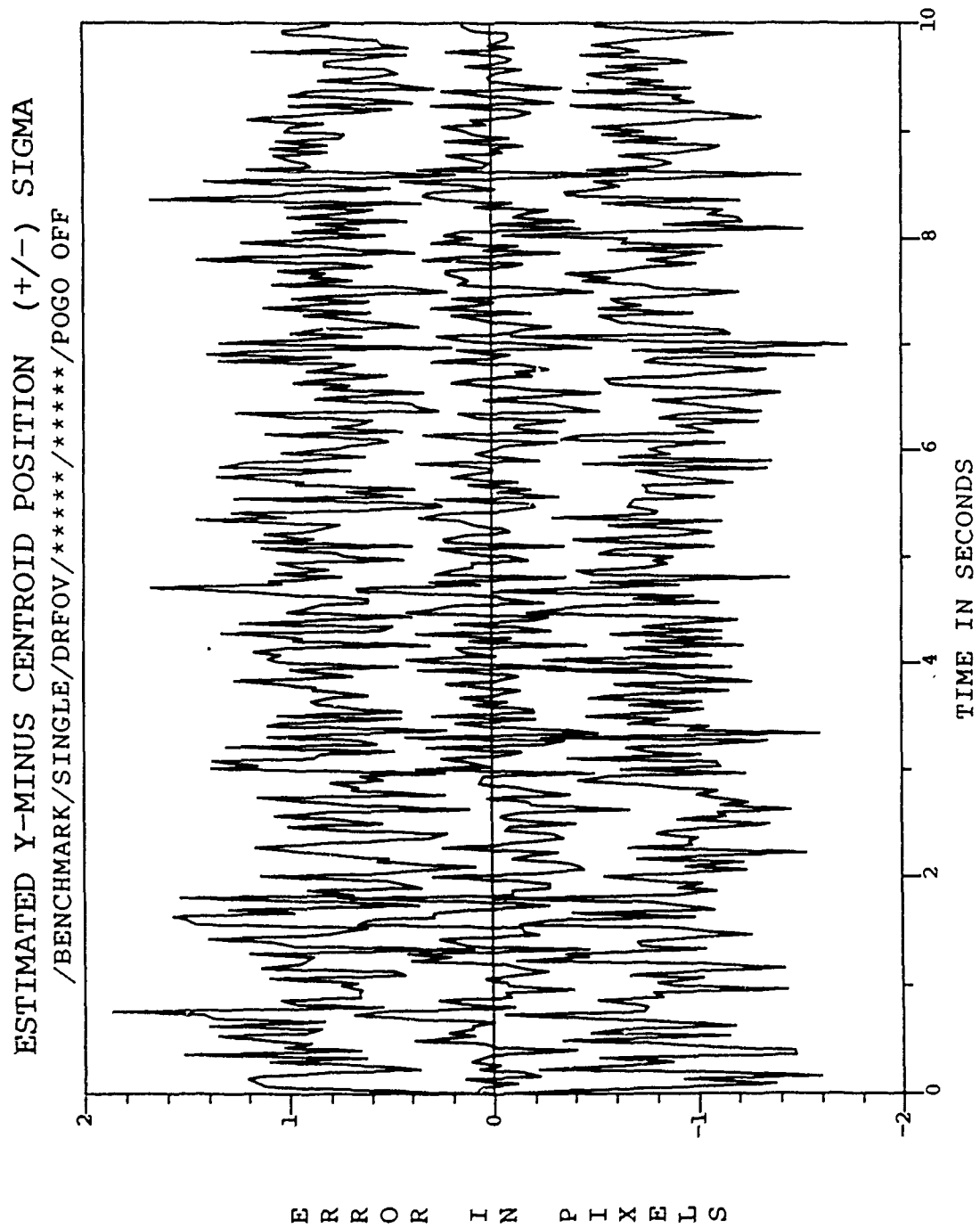


Figure B.18. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

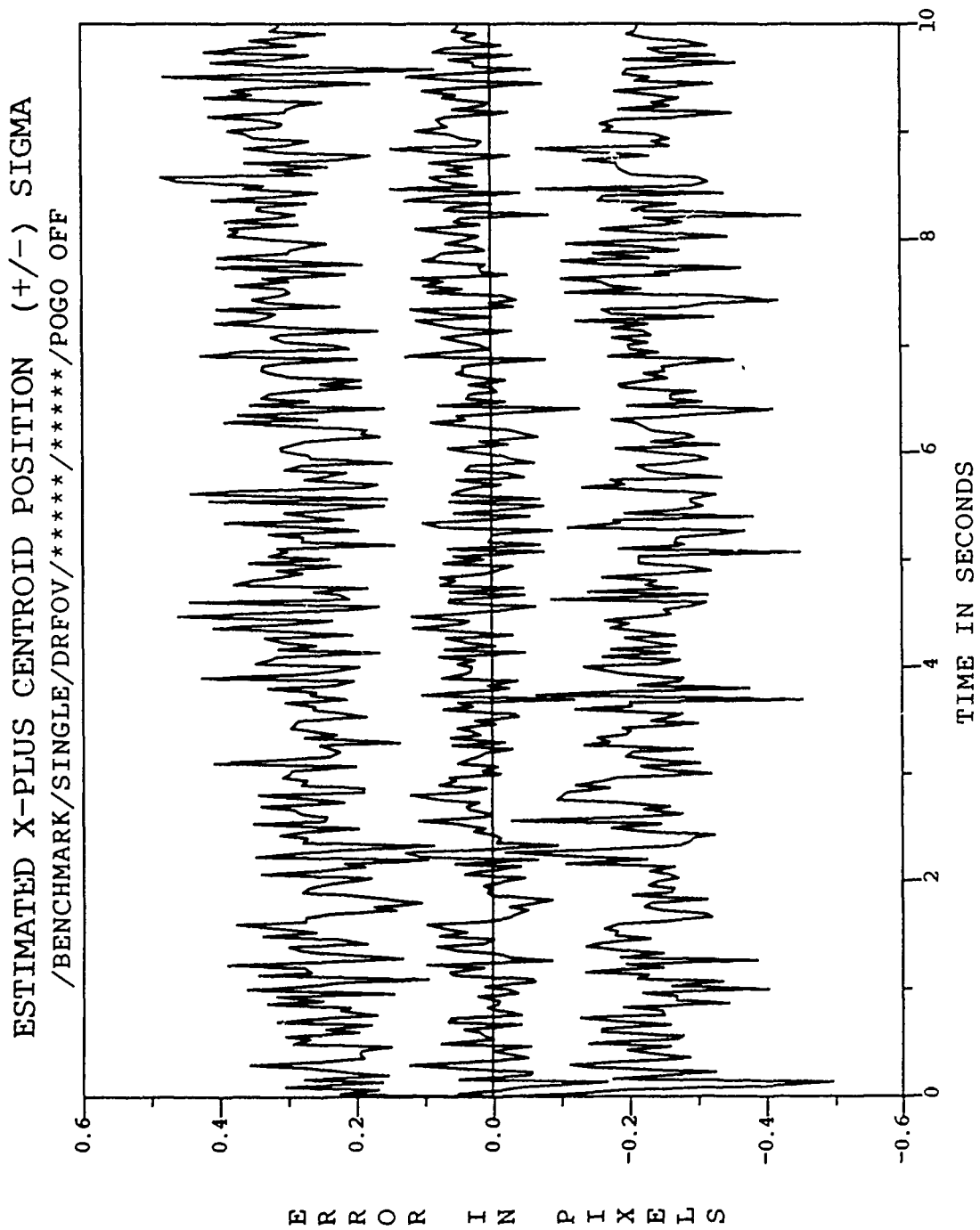


Figure B.49. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

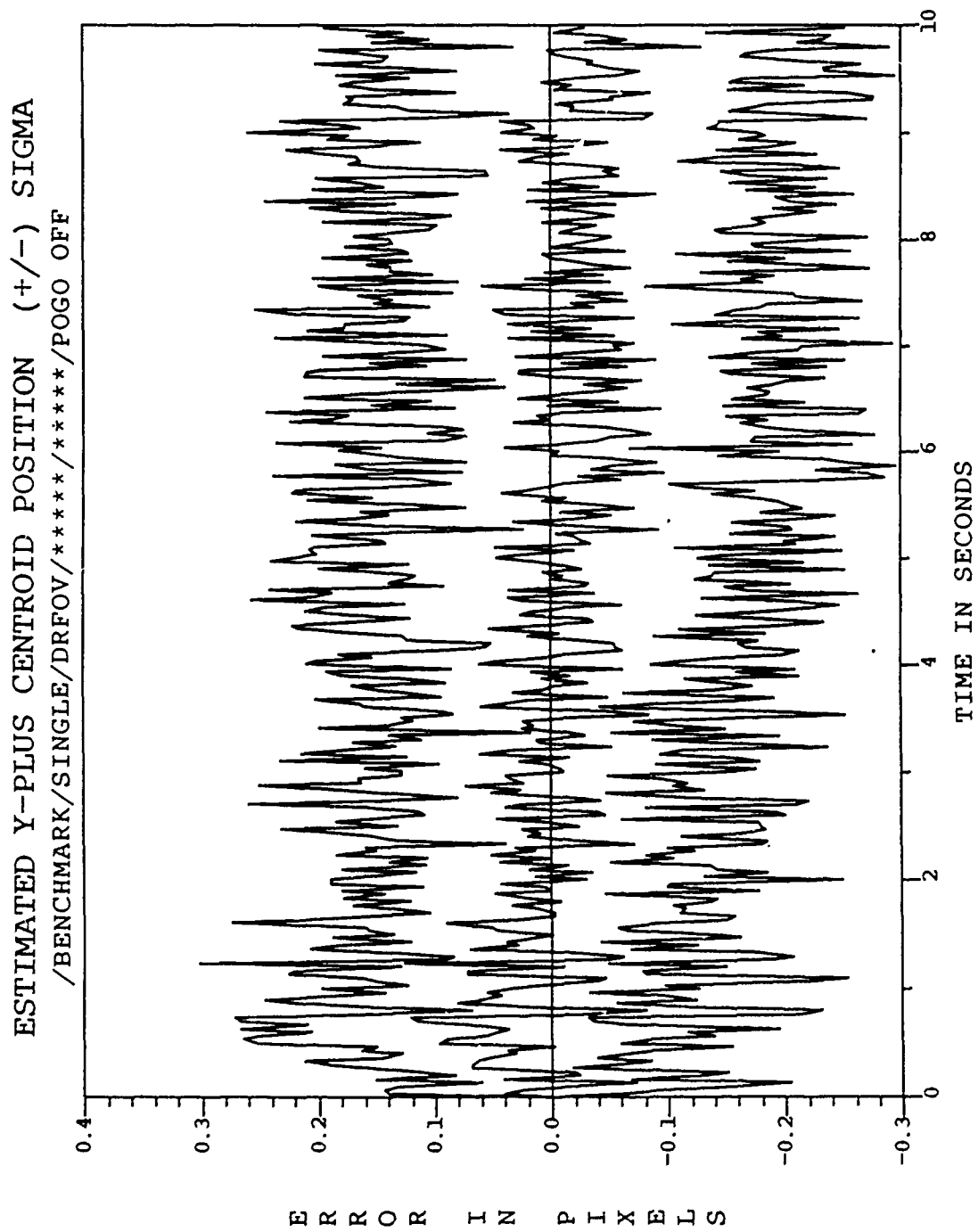


Figure B.50. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO OFF/

Appendix C. *8-State Filter Benchmark Plots: Discussion in*
Section 6.3

C.1 Figures C.1-C.10: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.20$

C.2 Figures C.11-C.20: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.15$

C.3 Figures C.21-C.30: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.10$

C.4 Figures C.31-C.40: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.15$

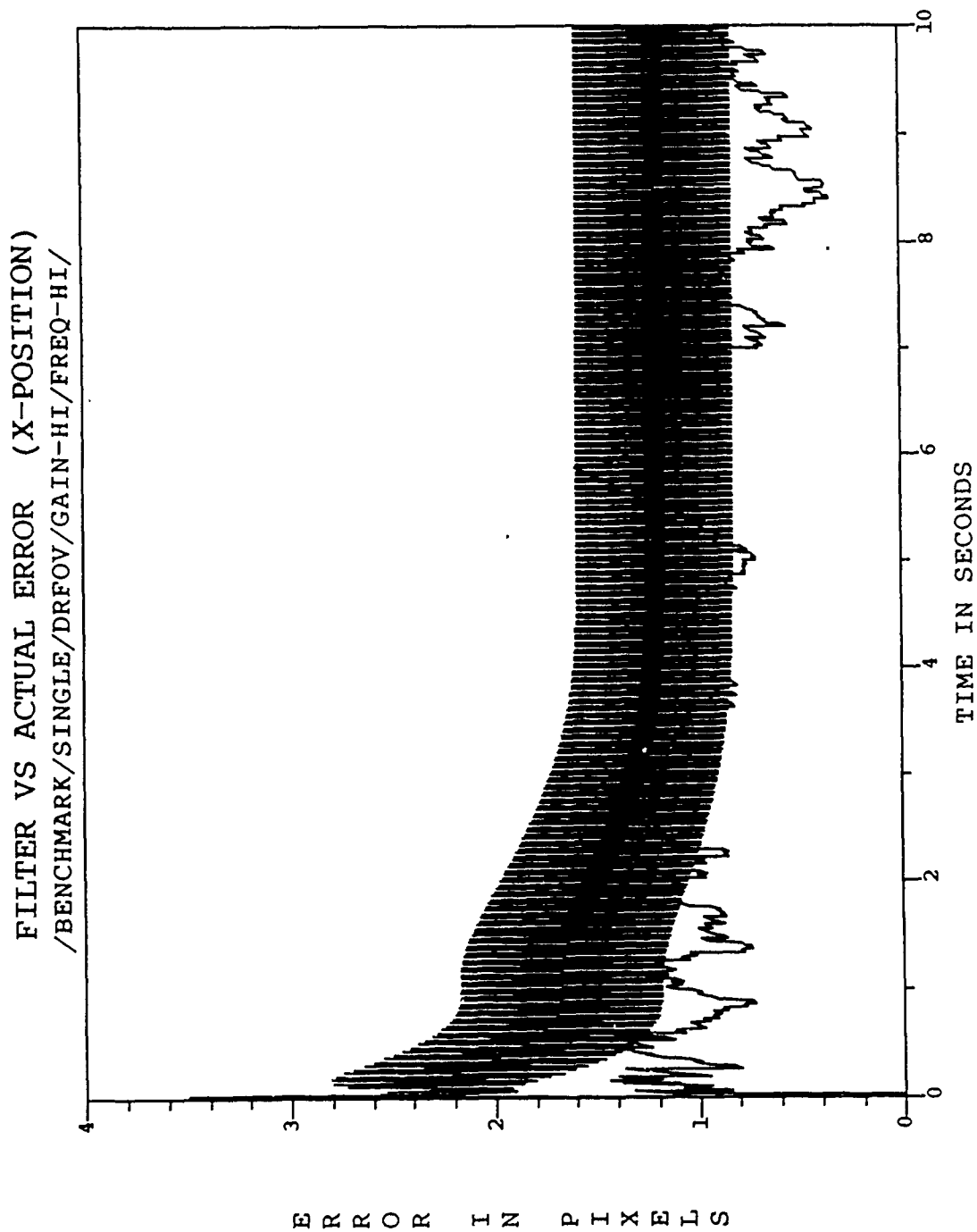


Figure C.1. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

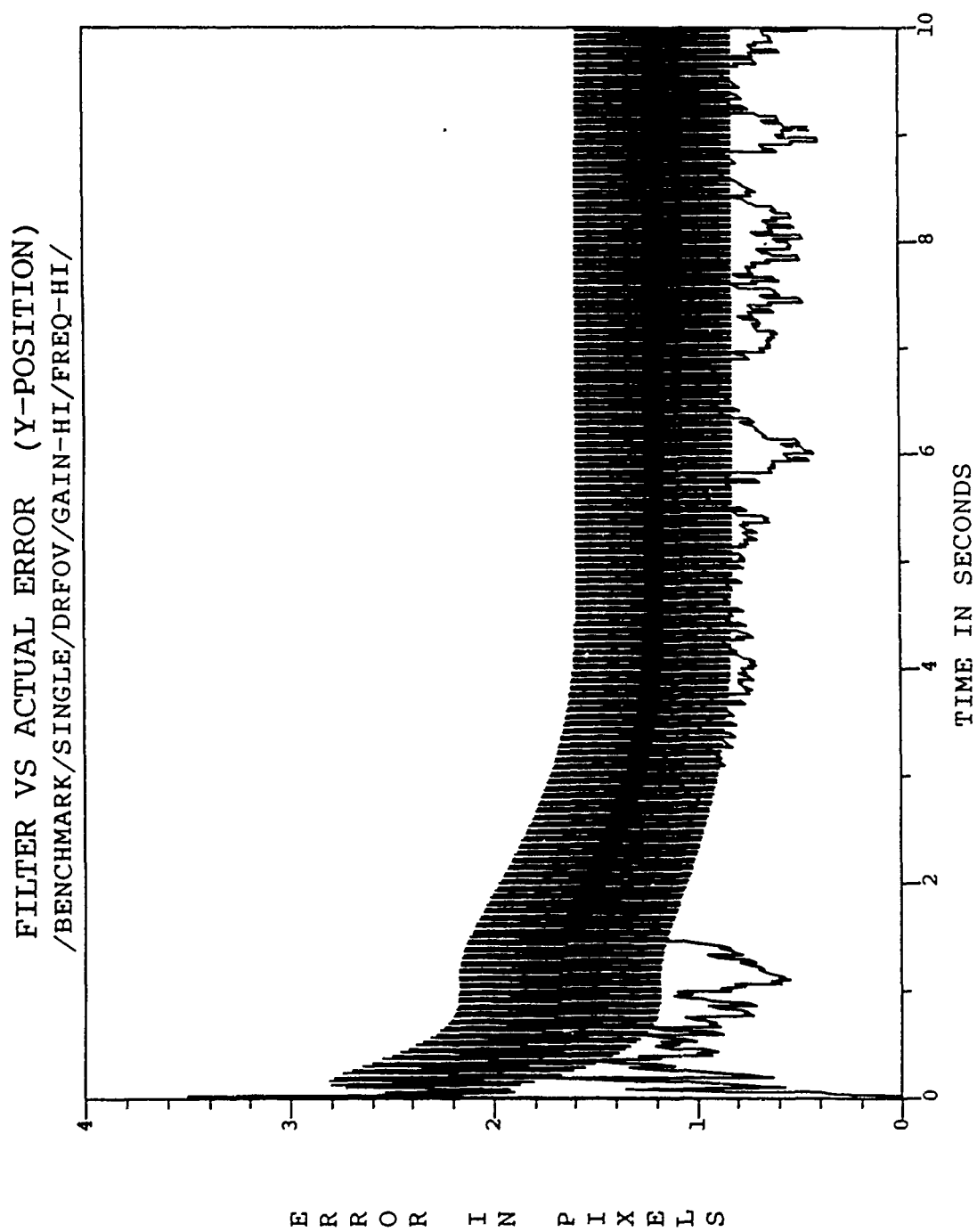


Figure C.2. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

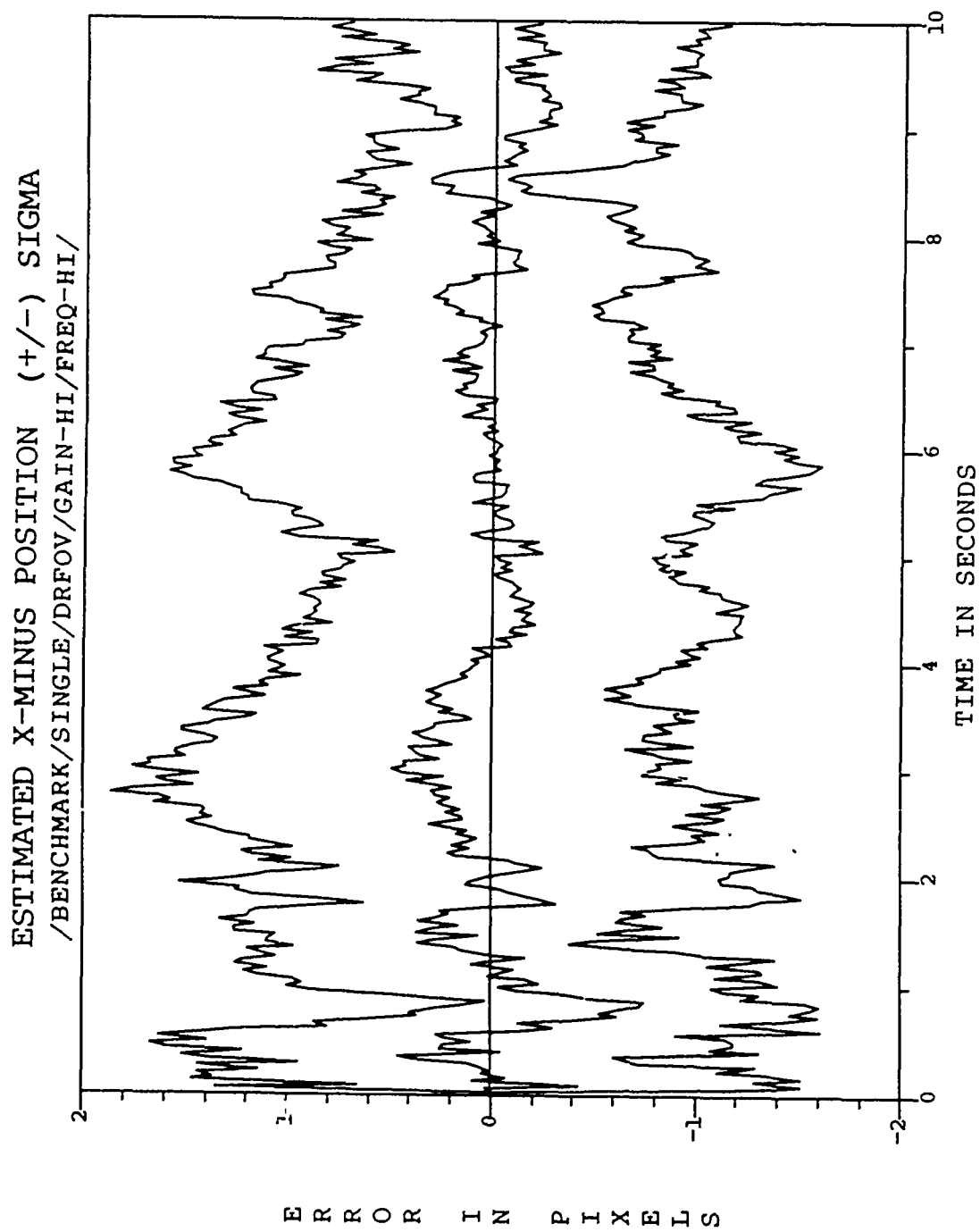


Figure C.3. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

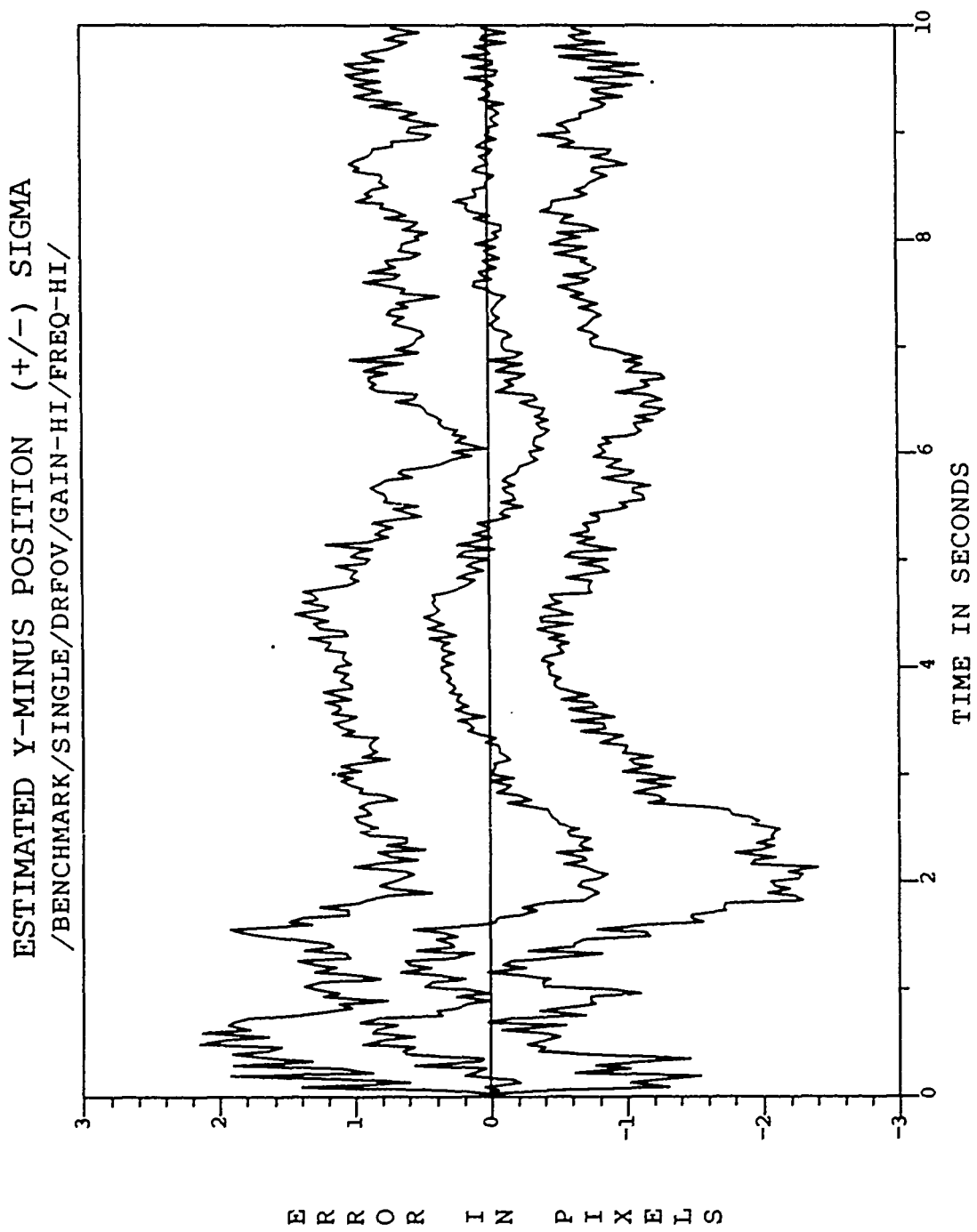


Figure C.1. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

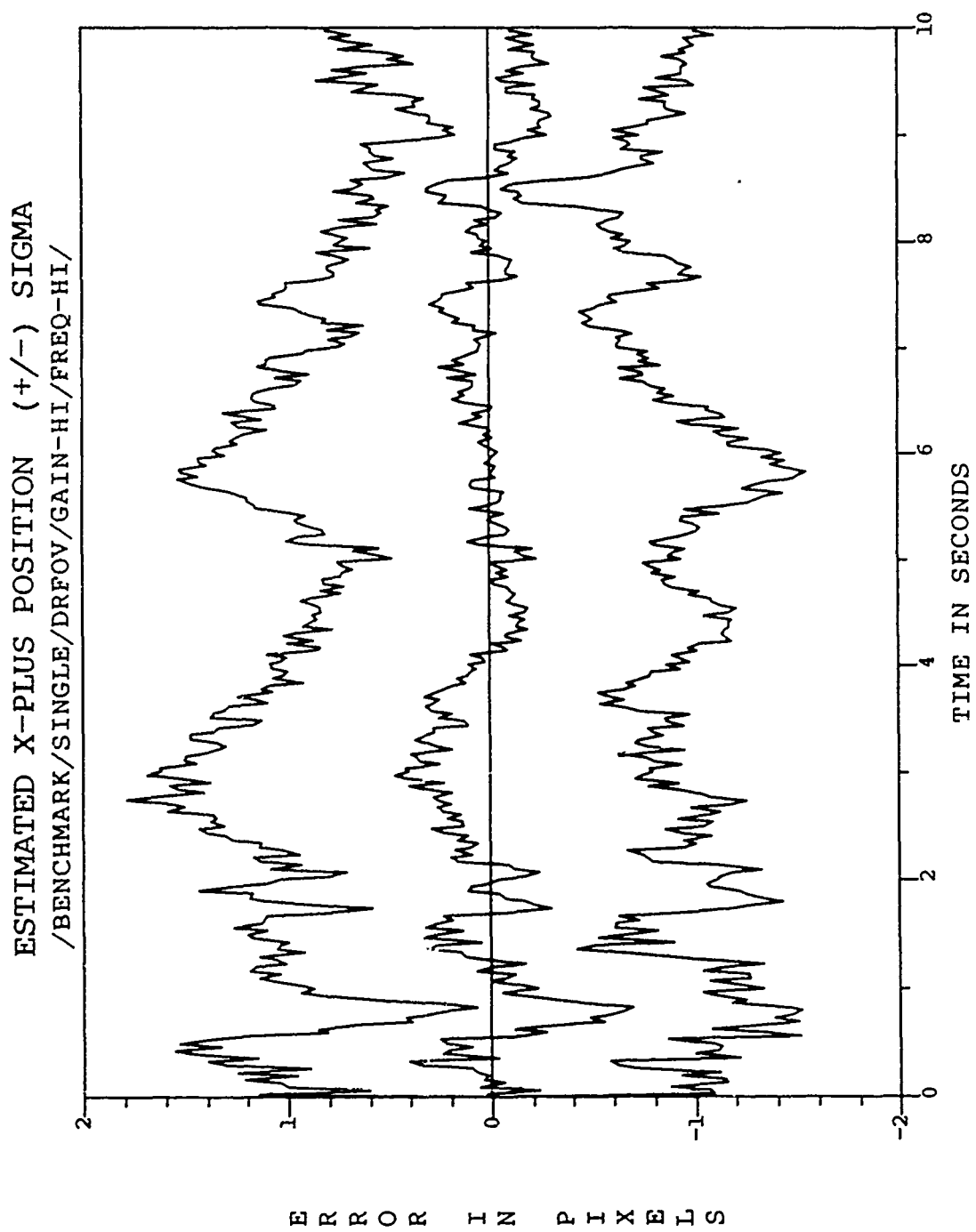


Figure C.5. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

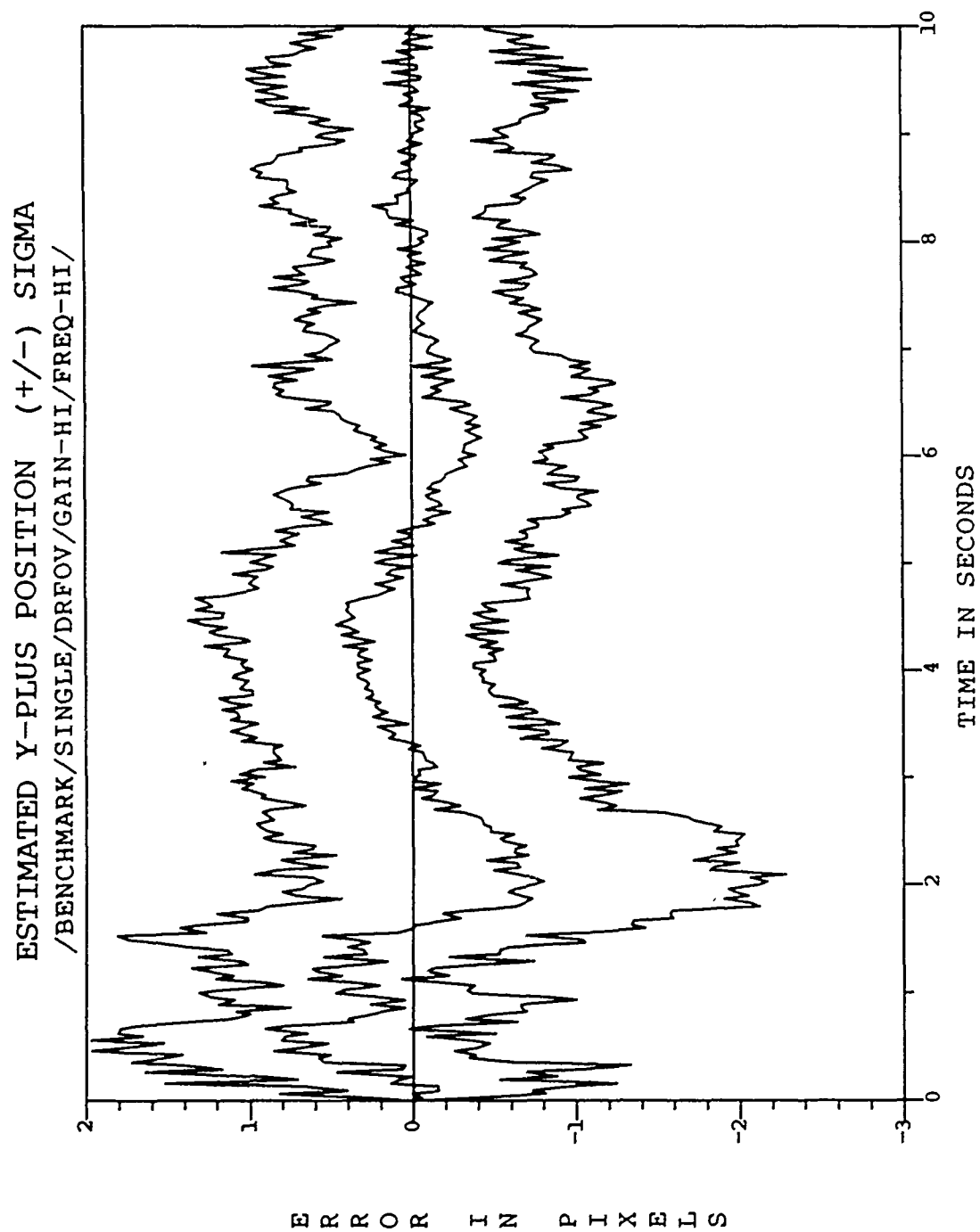


Figure C.6. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

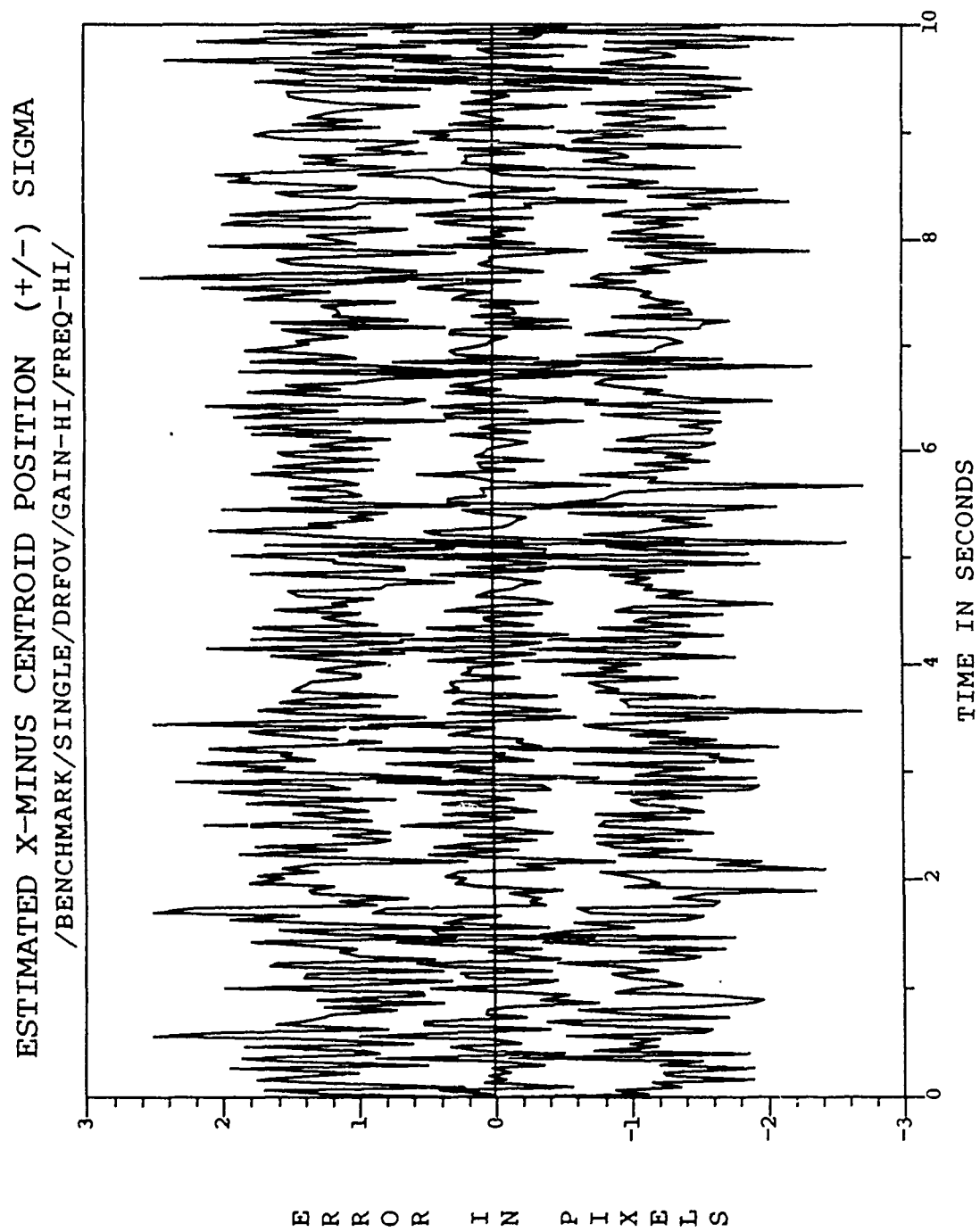


Figure C.7. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

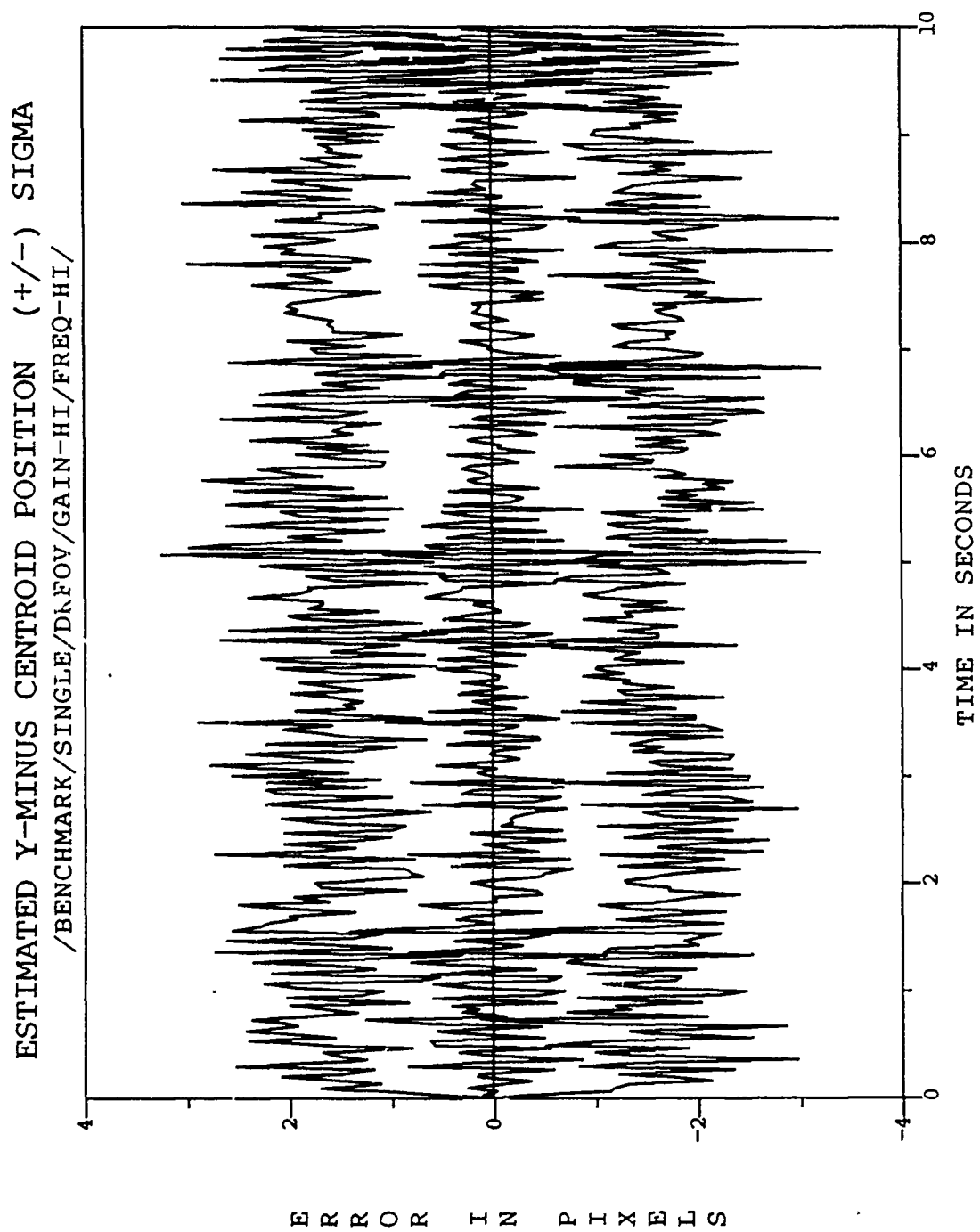


Figure C.8. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

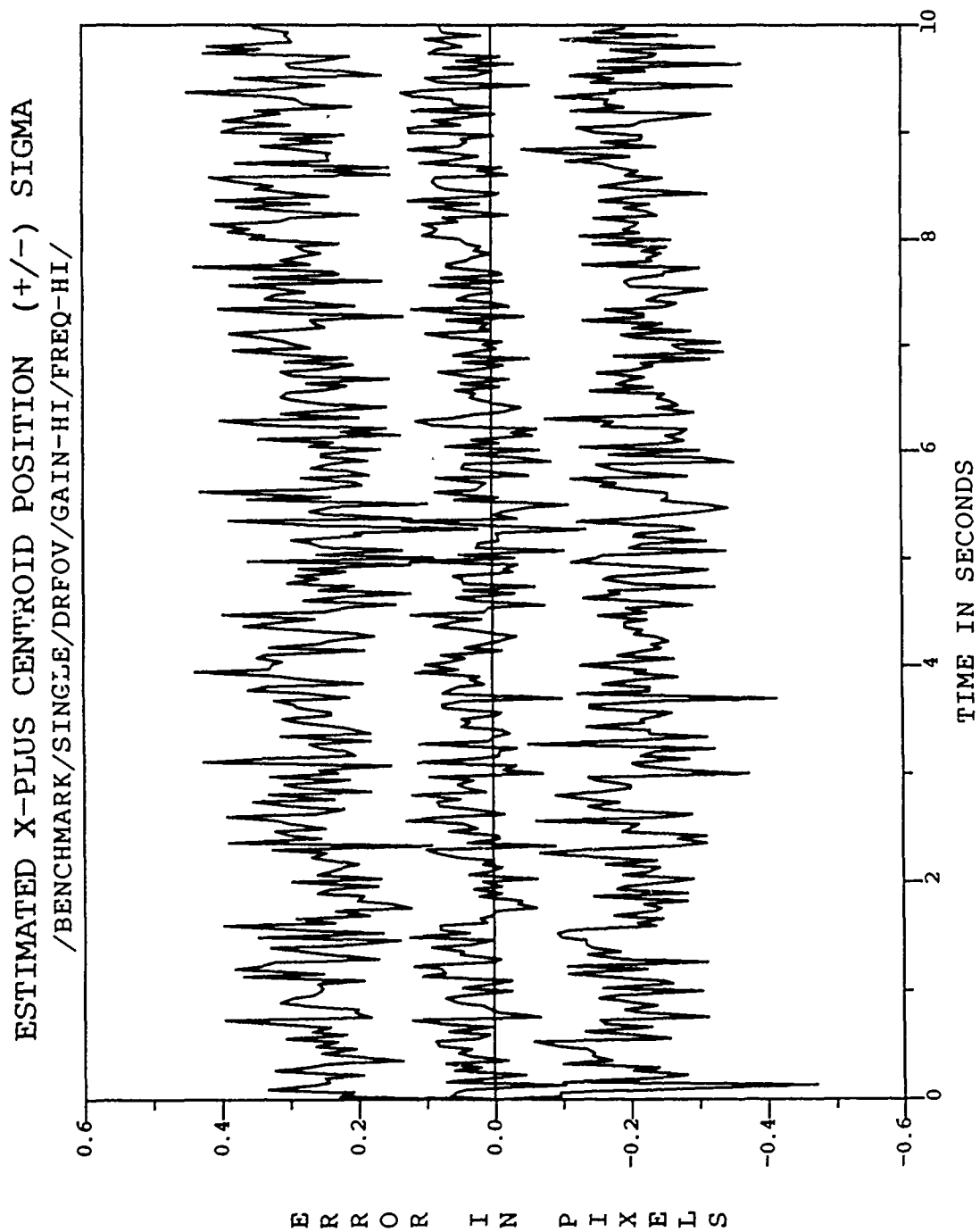


Figure C.9. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

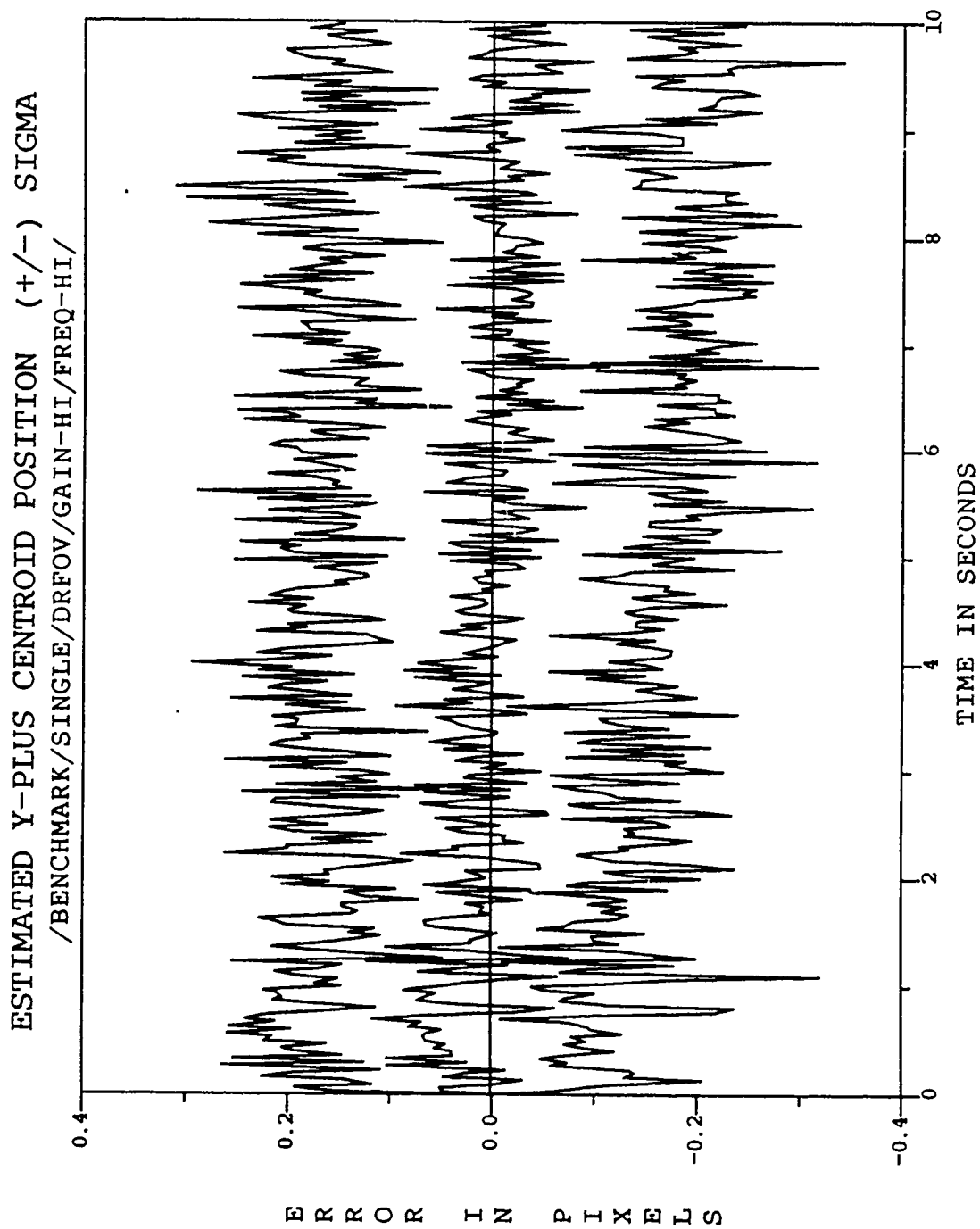


Figure C.10. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

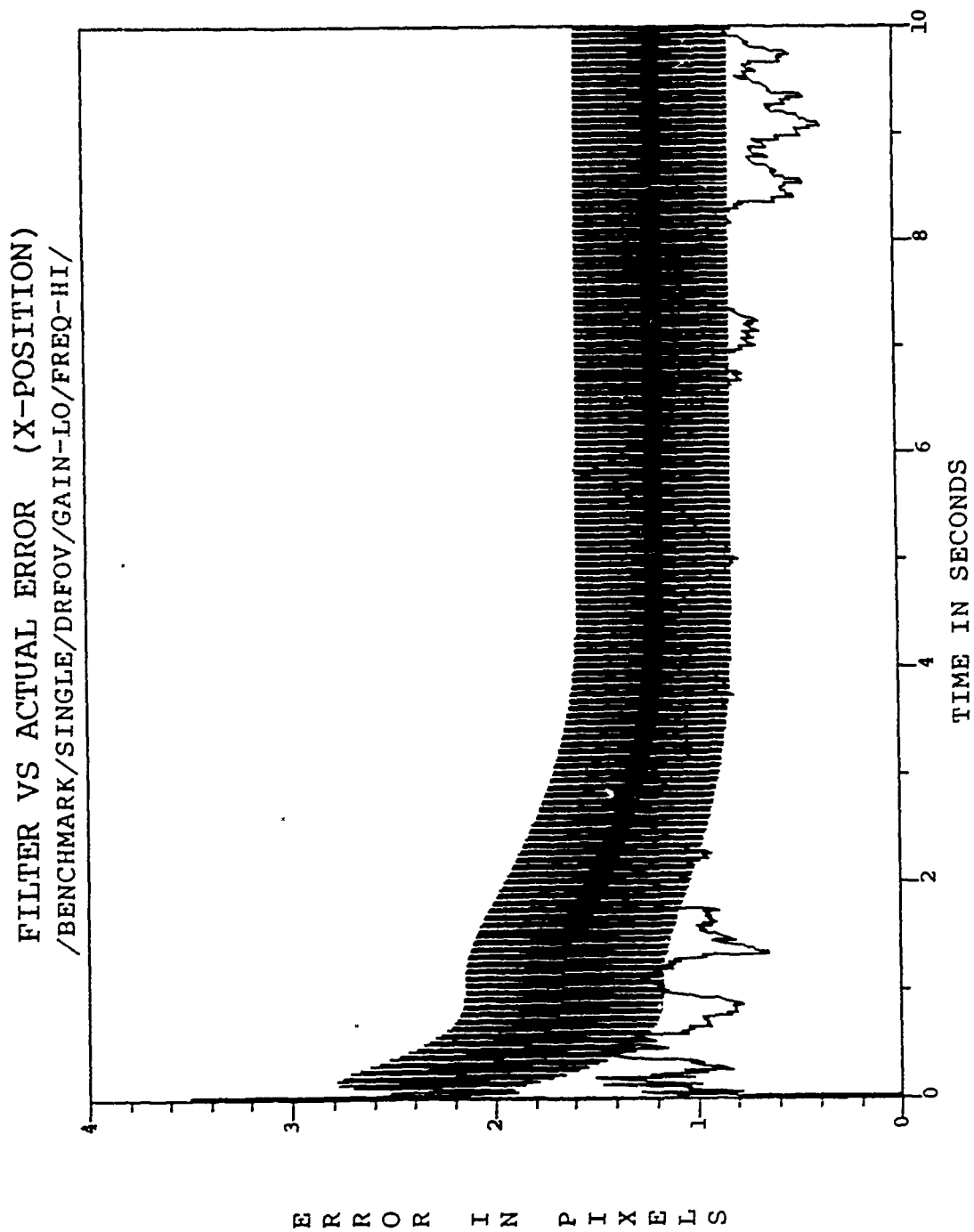


Figure C.11. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

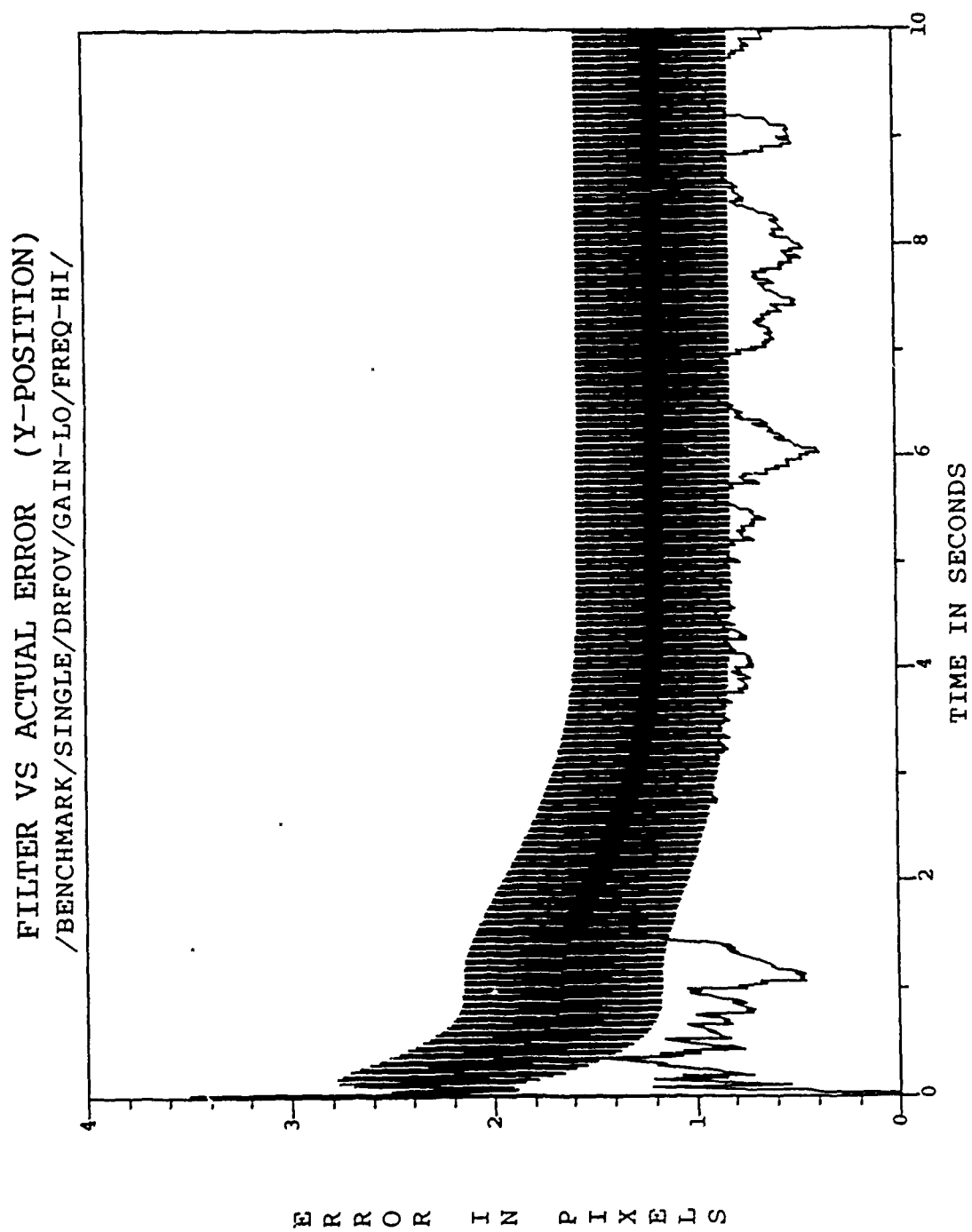


Figure C.12. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

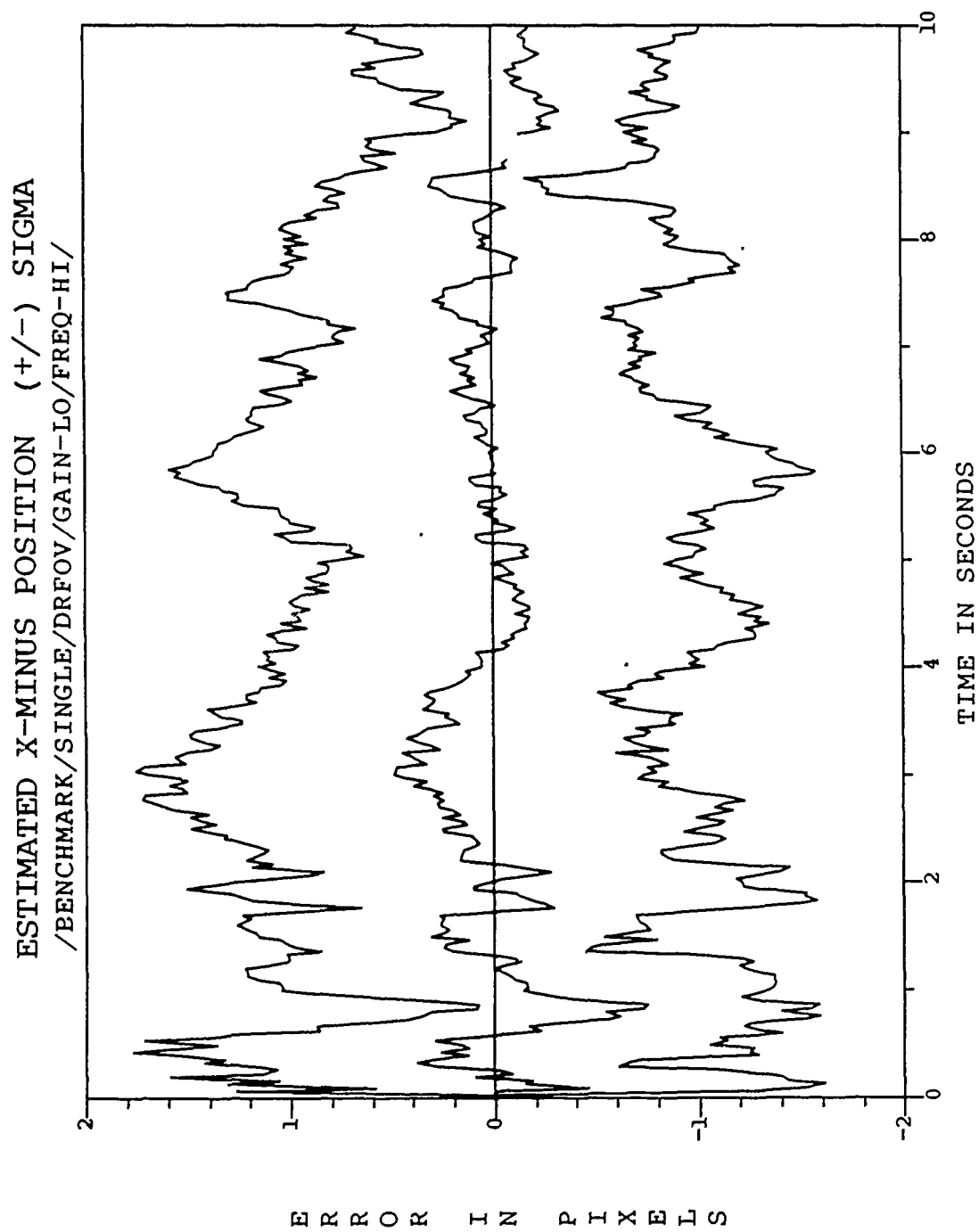


Figure C.13. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

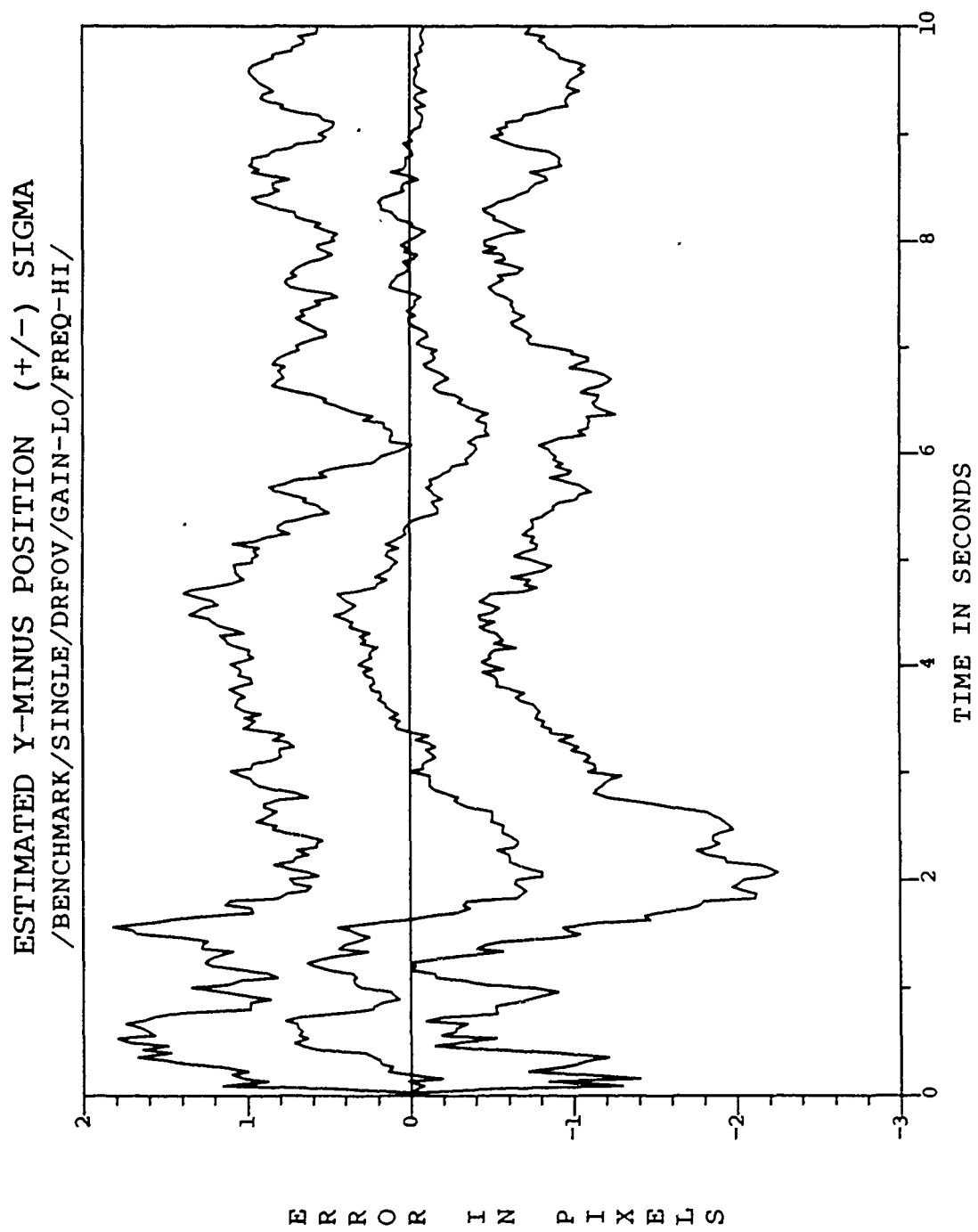


Figure C.14. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

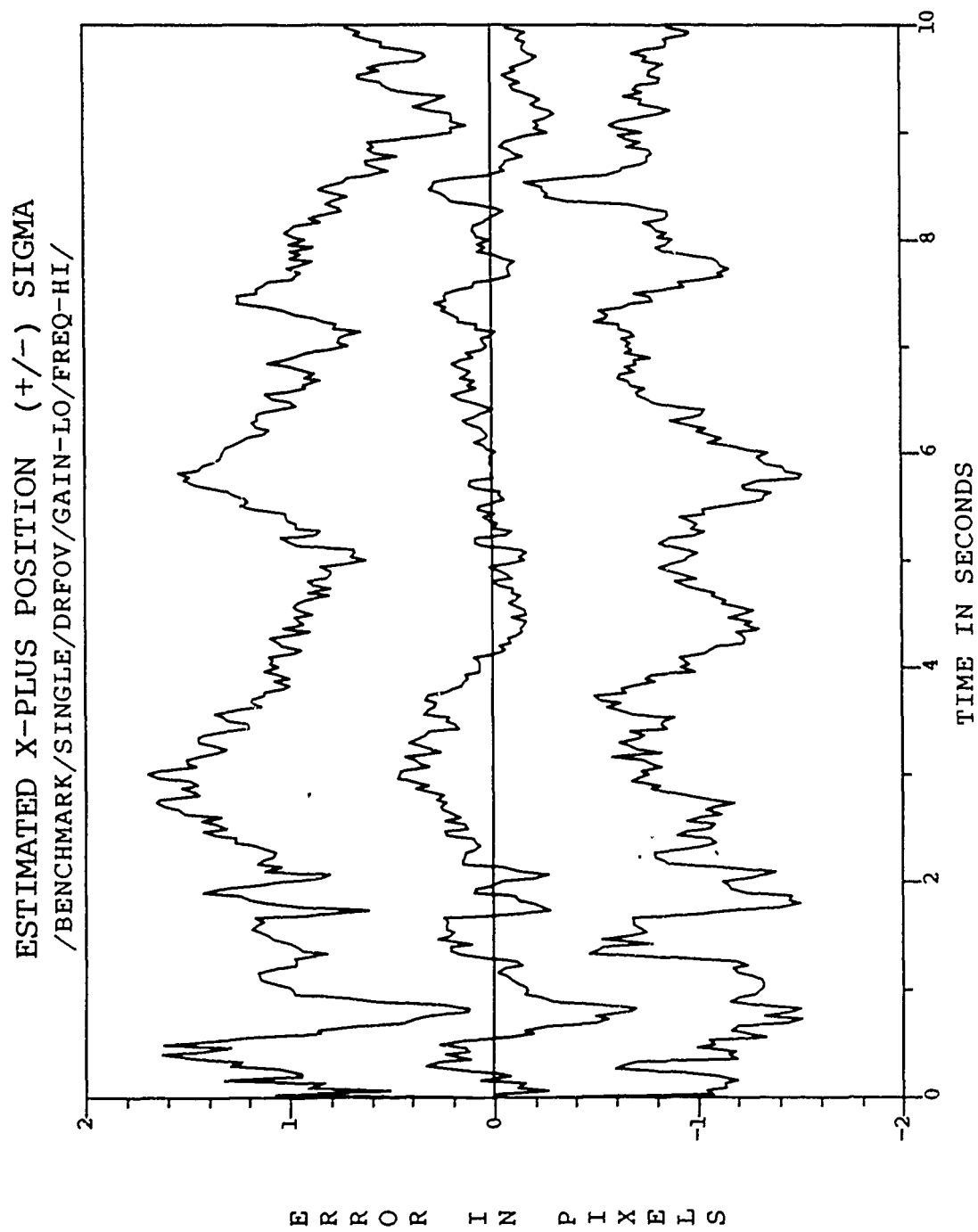


Figure C.15. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

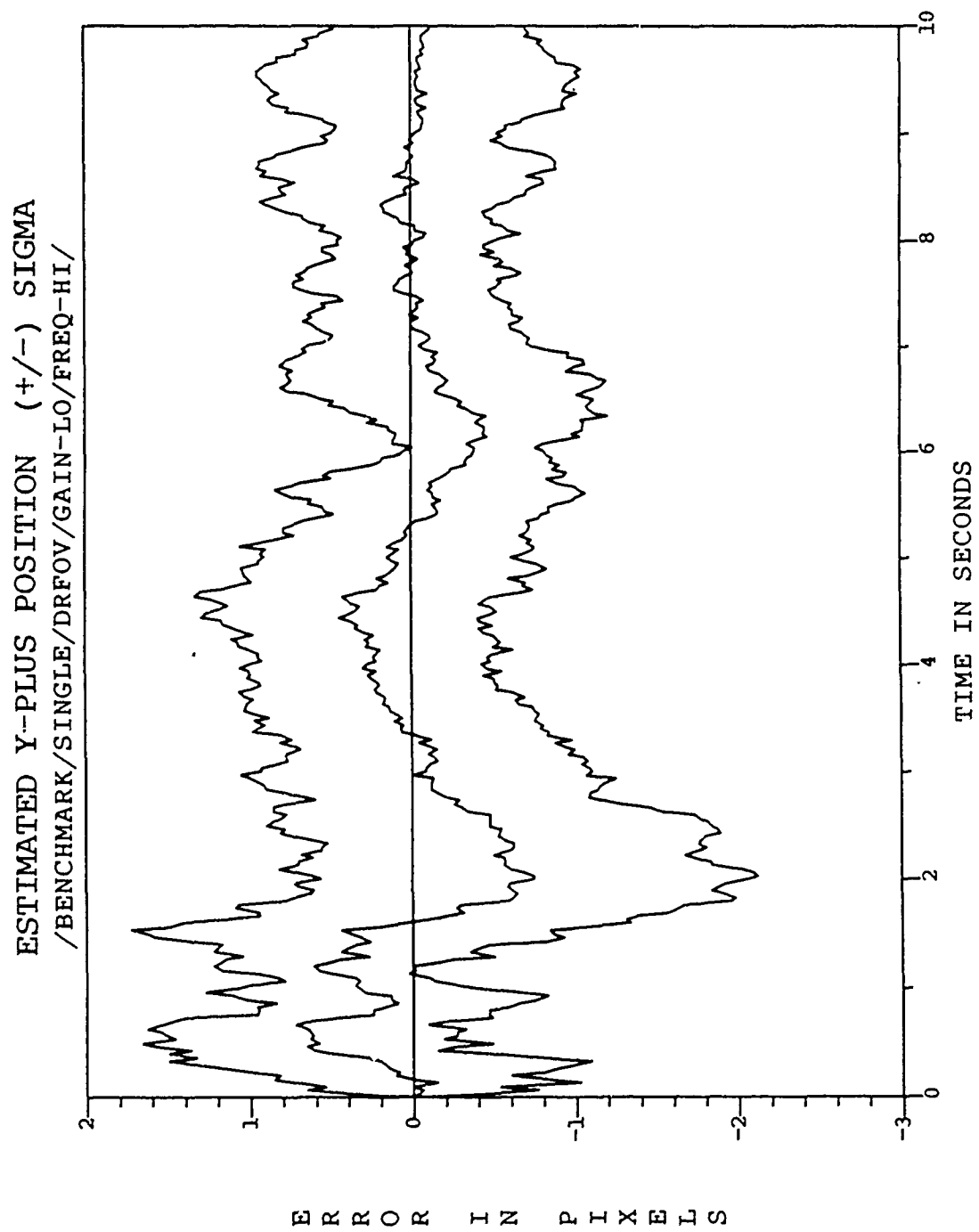


Figure C.16. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

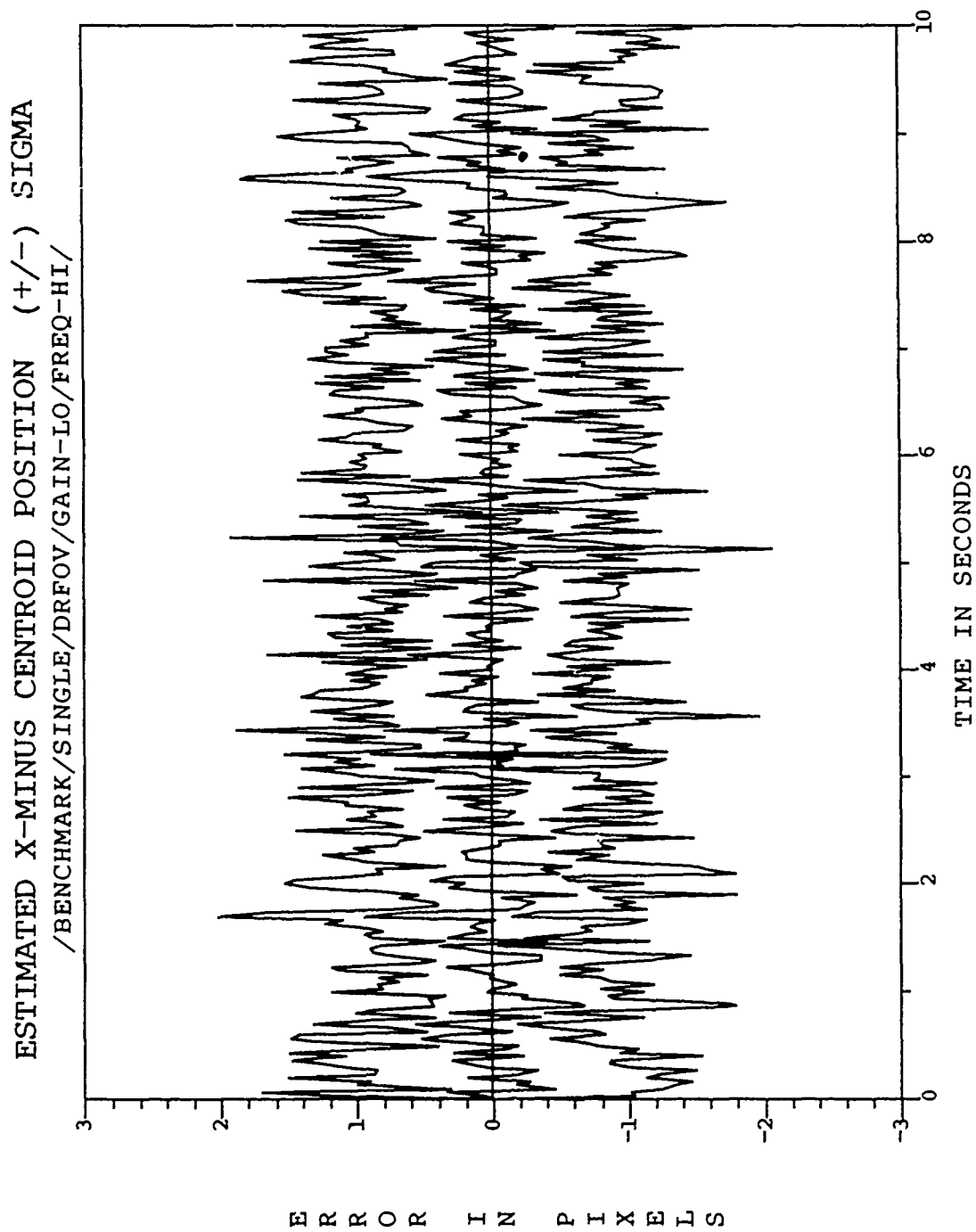


Figure C.17. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

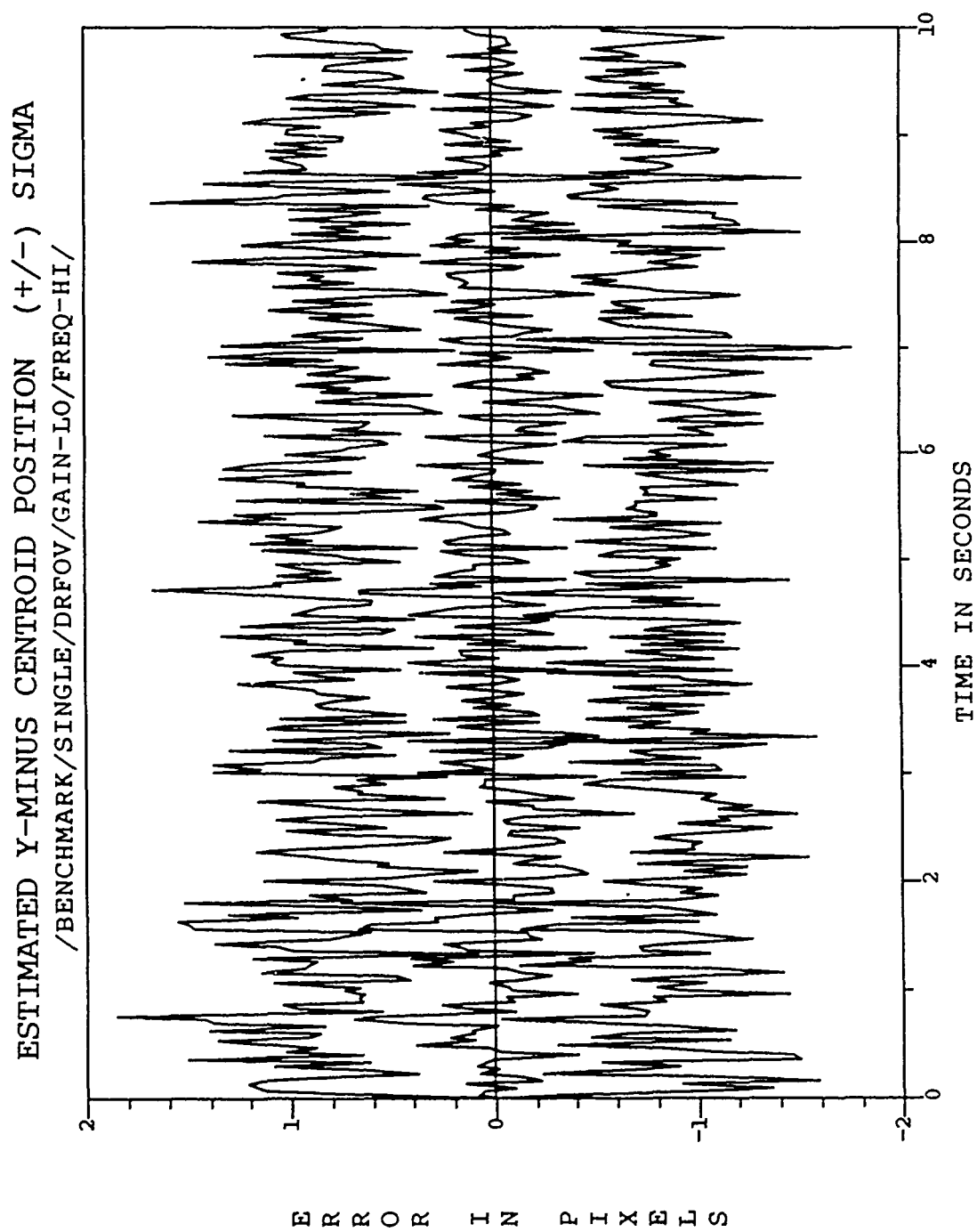


Figure C.18. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

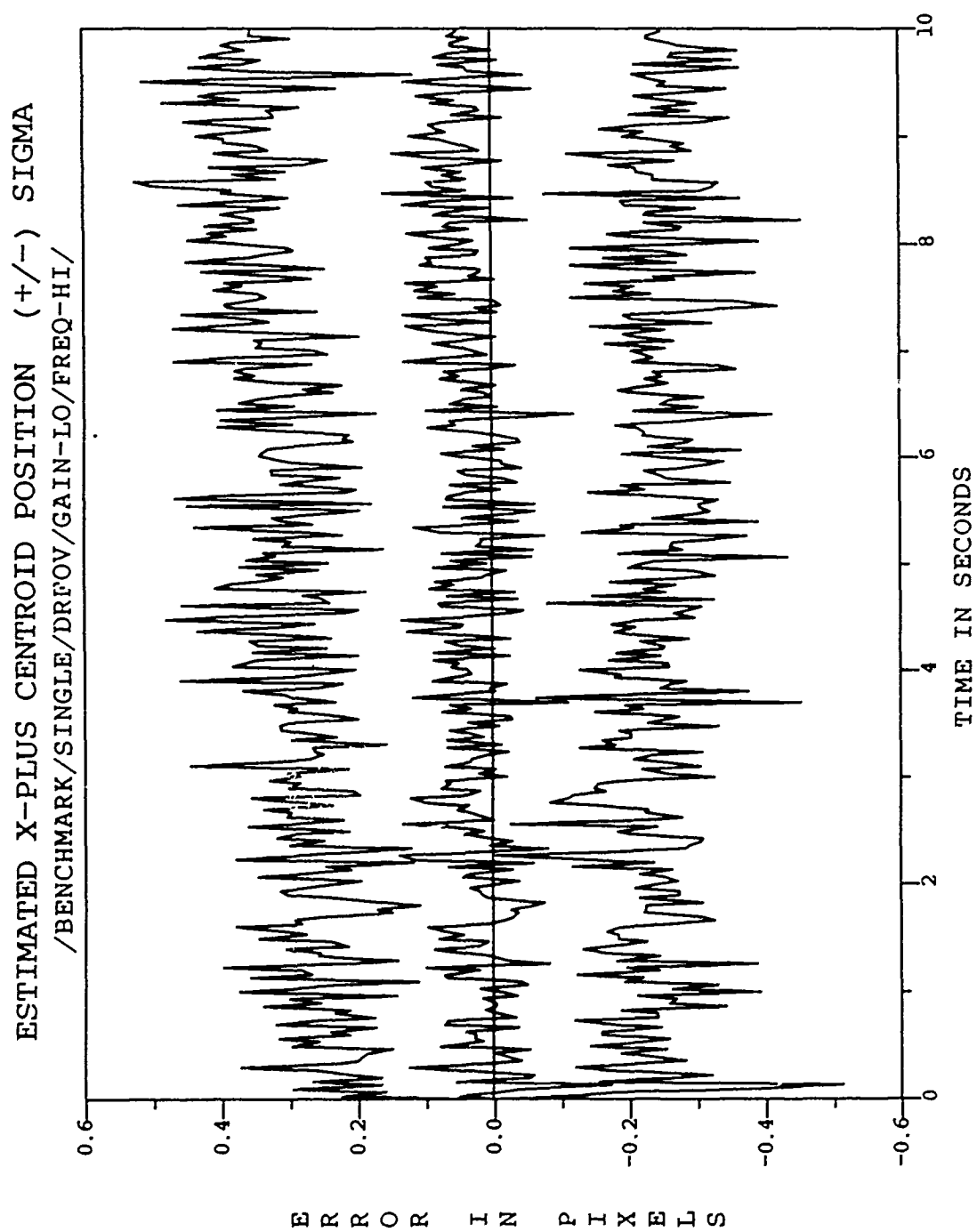


Figure C.19. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

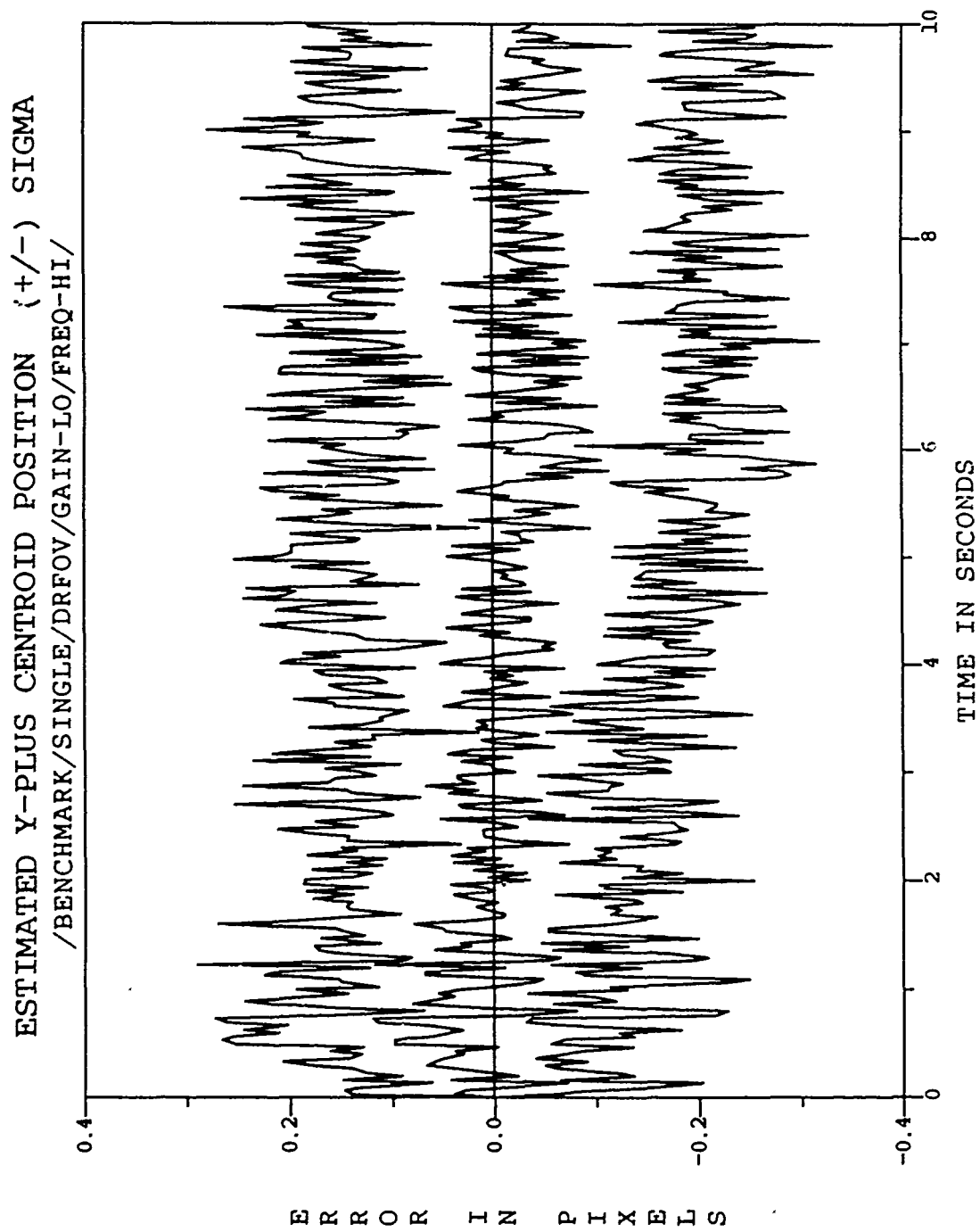


Figure C.20. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

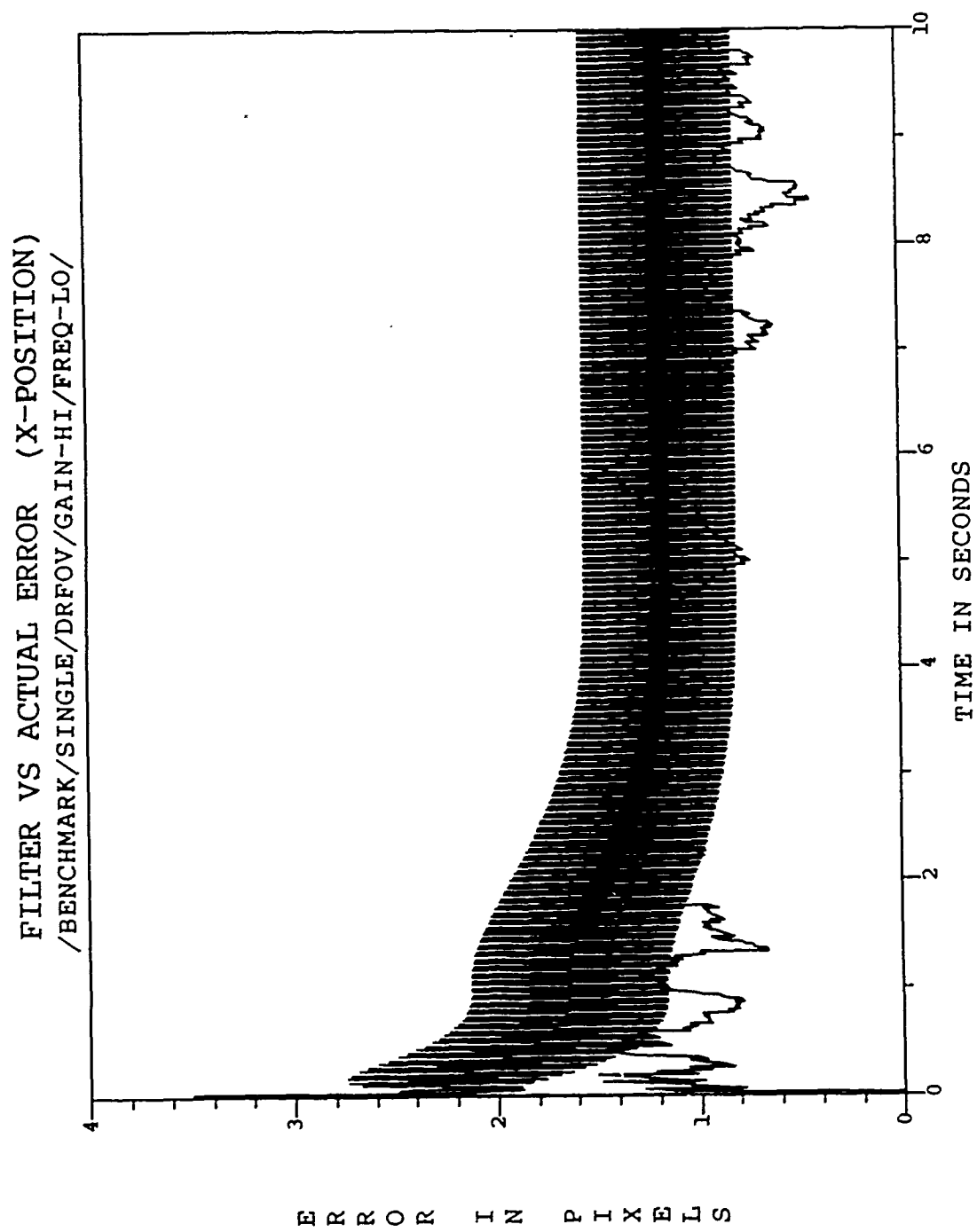


Figure C.21. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

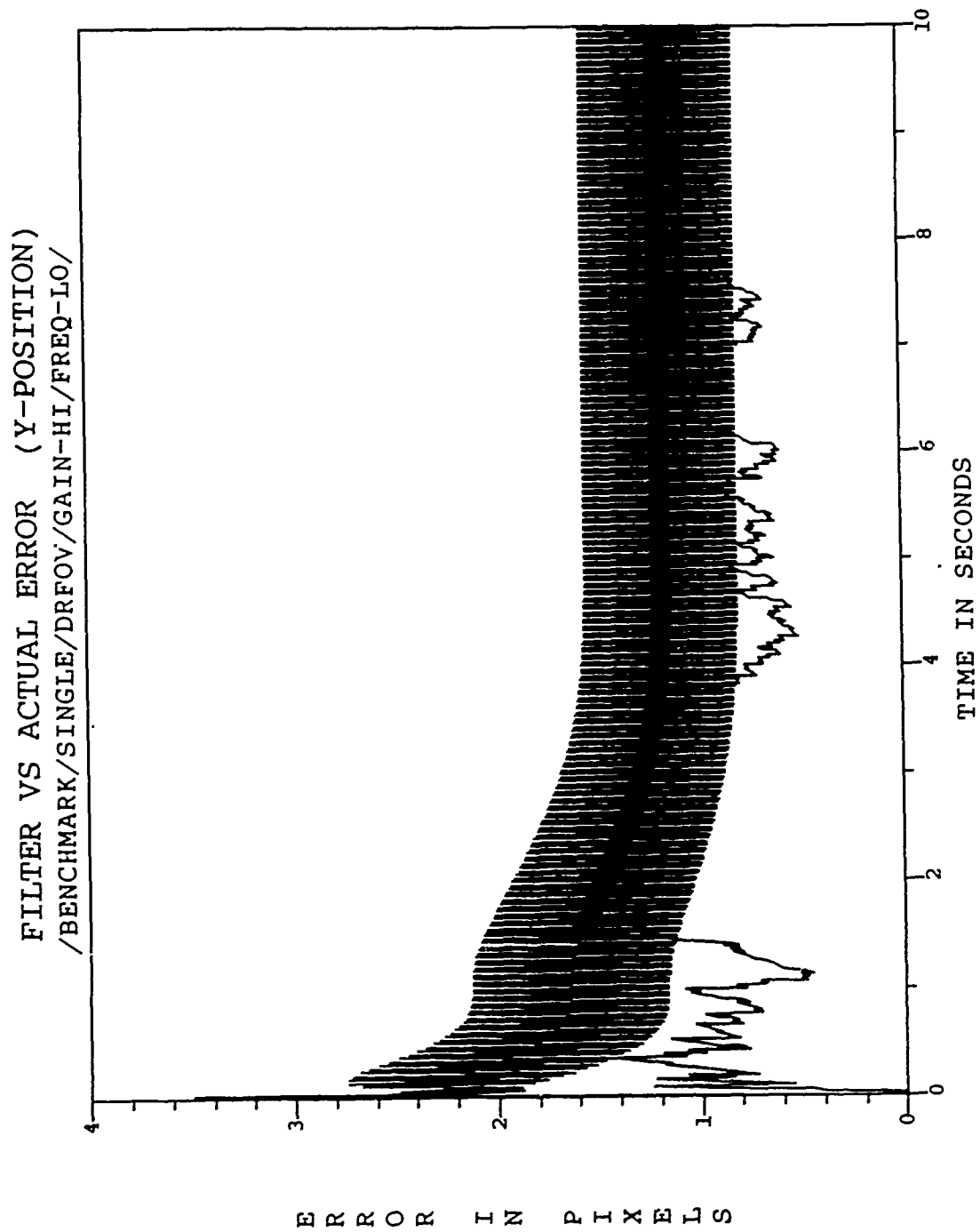


Figure C.22. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

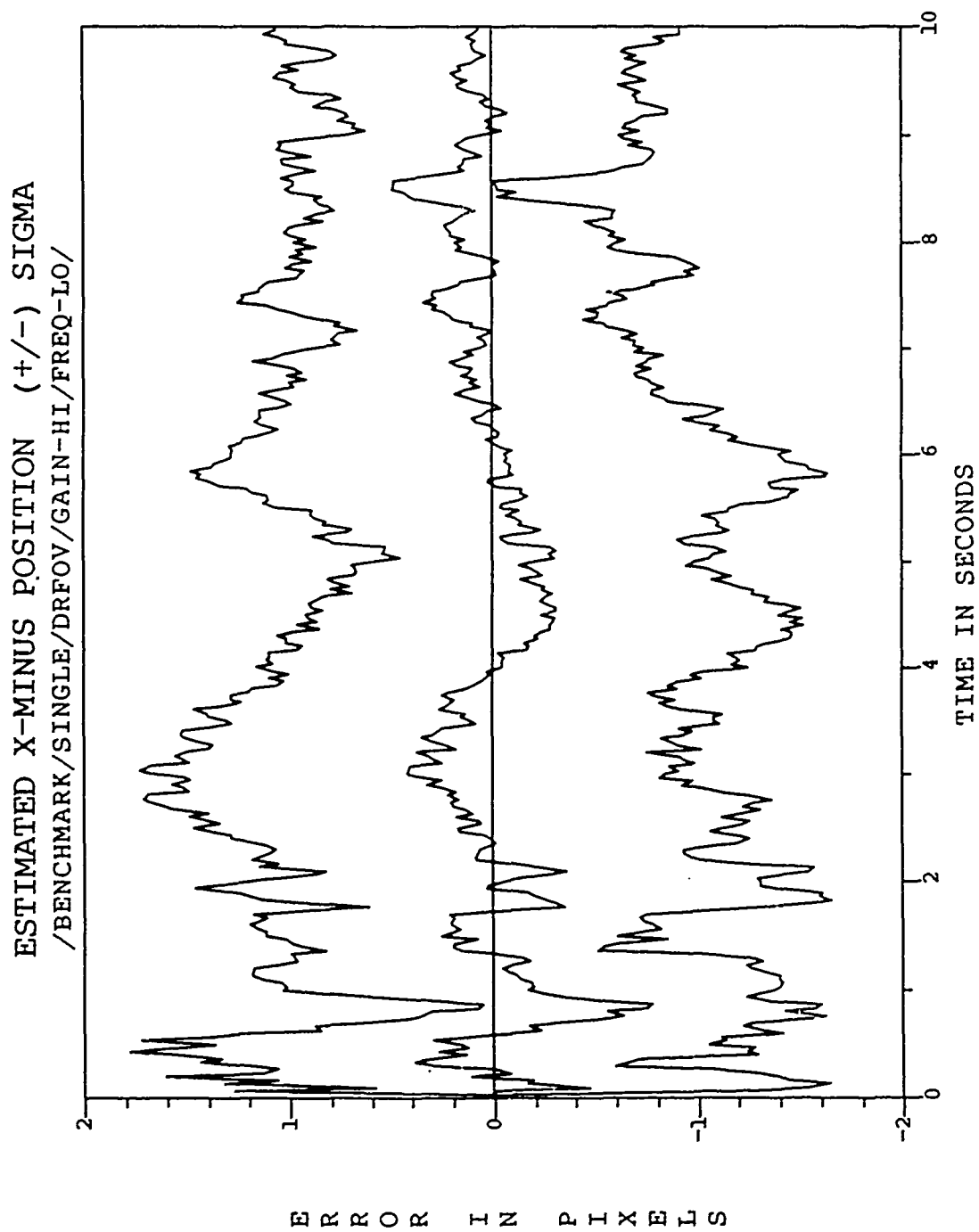


Figure C.23. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

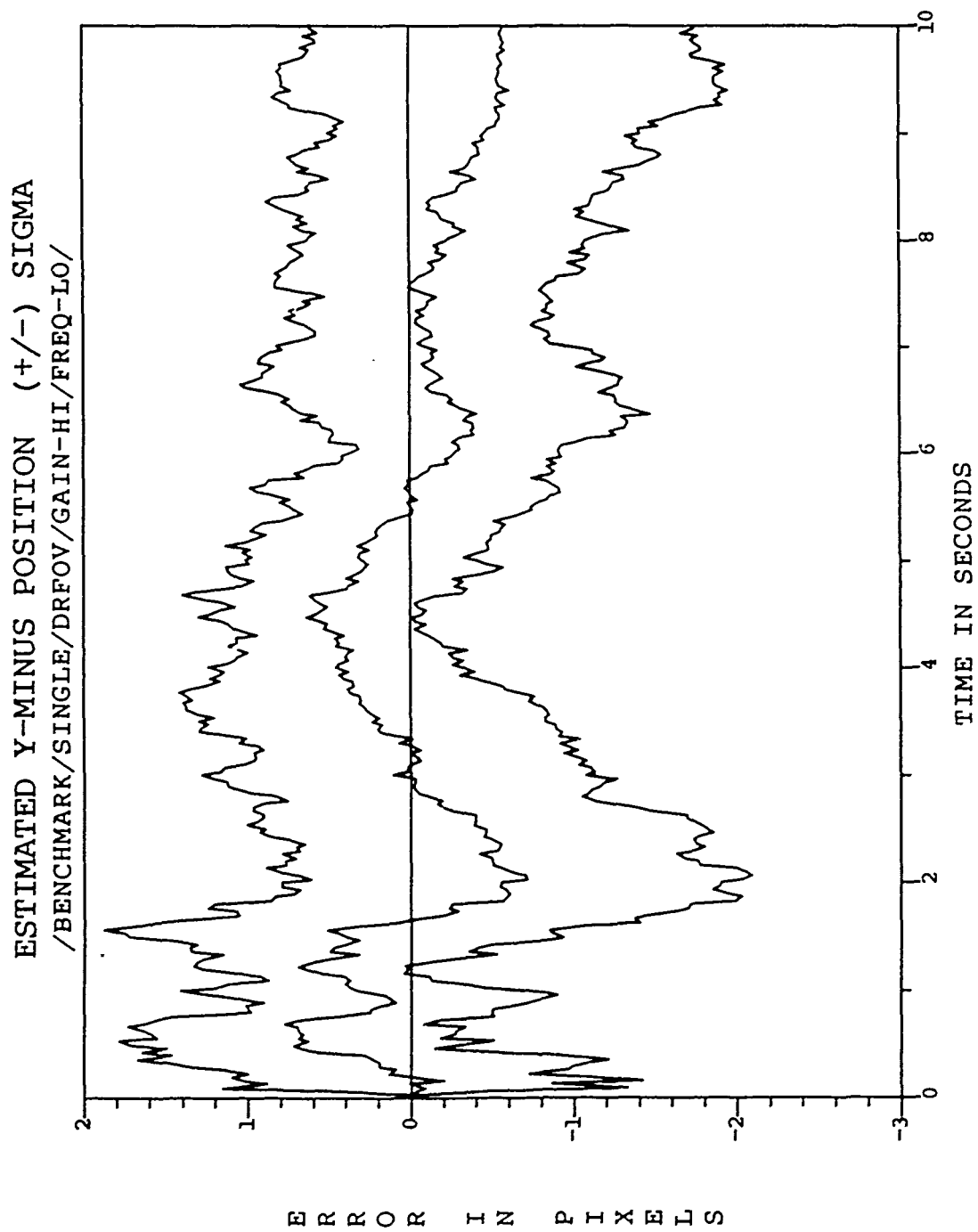


Figure C.24. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

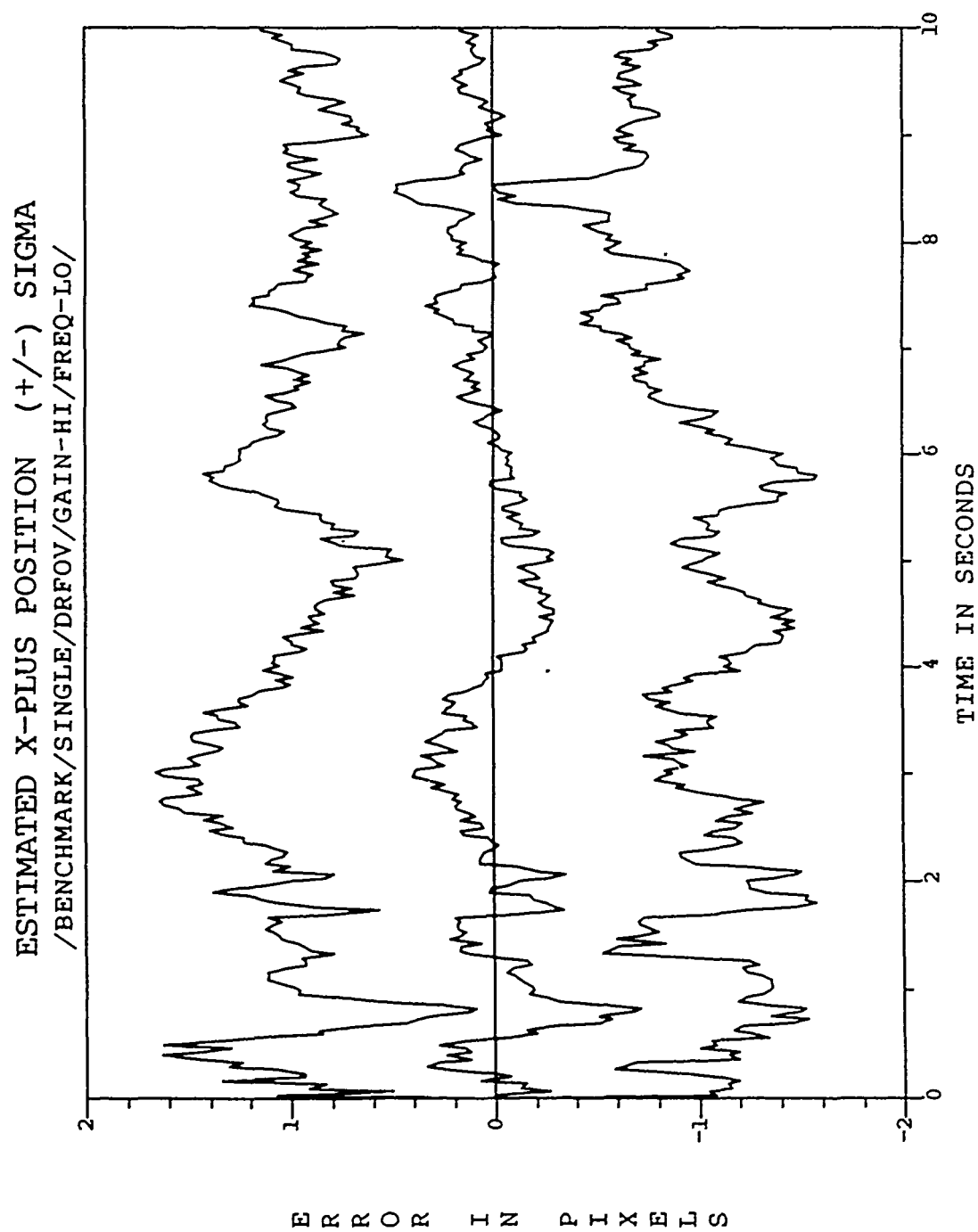


Figure C.25. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

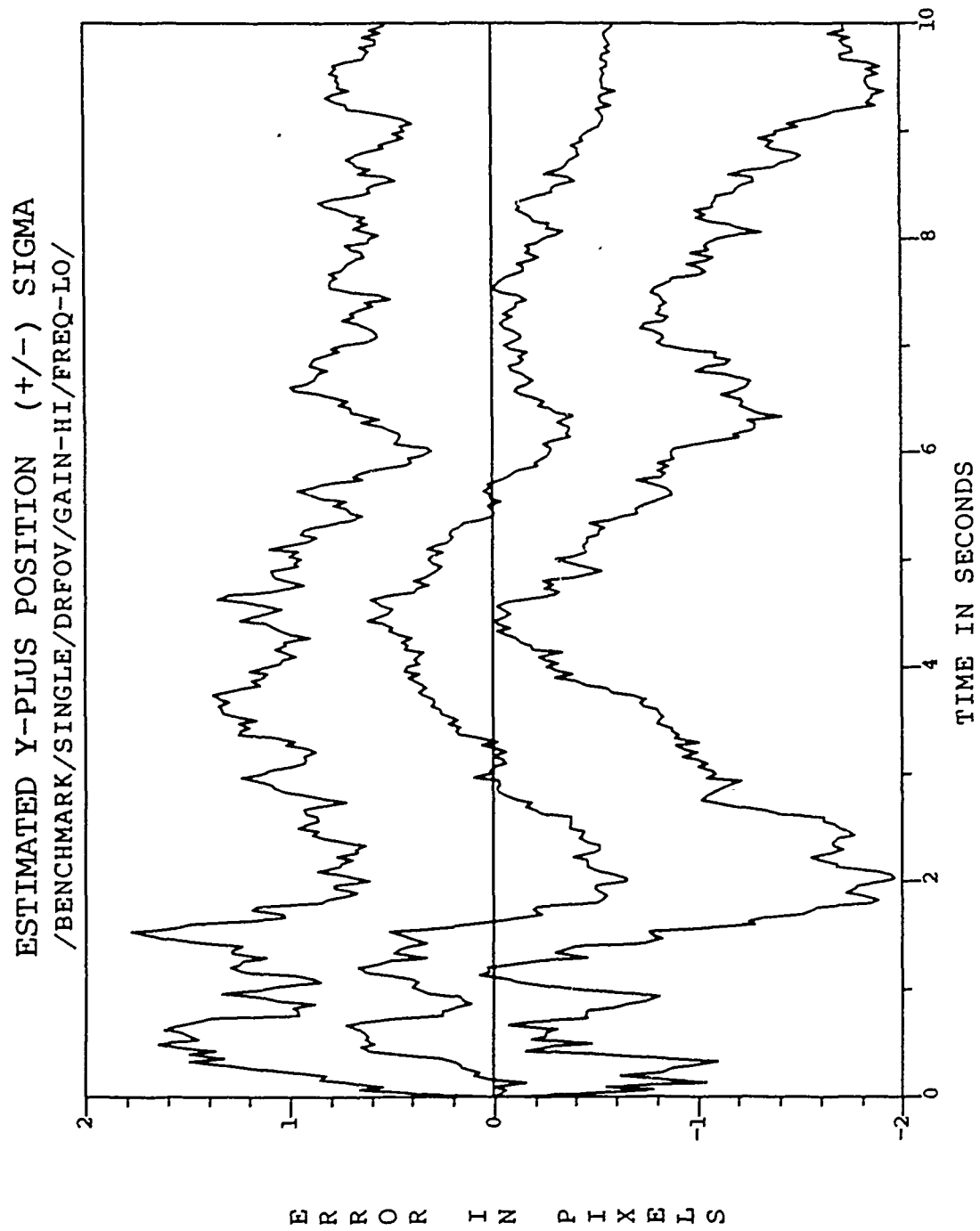


Figure C.26. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

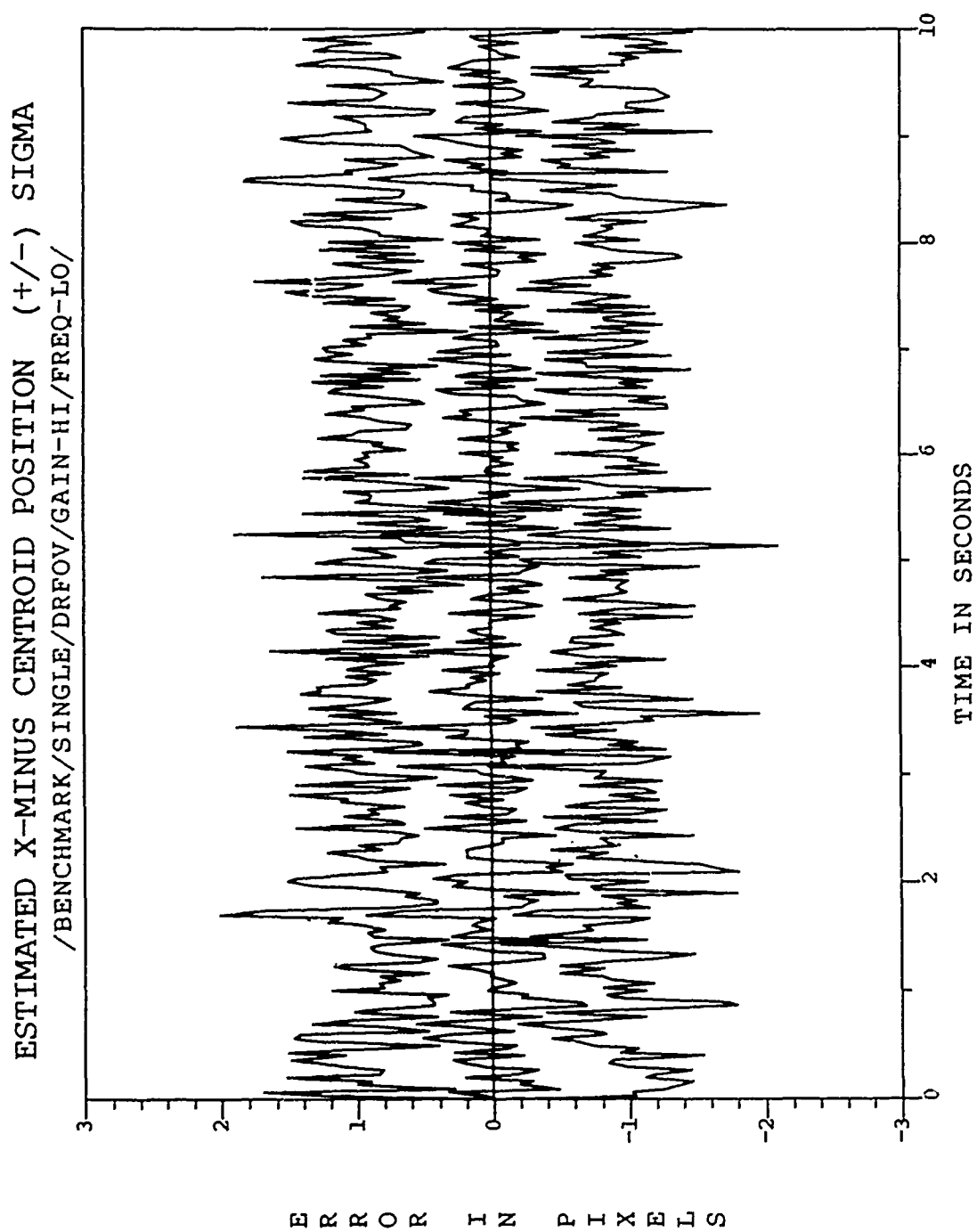


Figure C.27. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

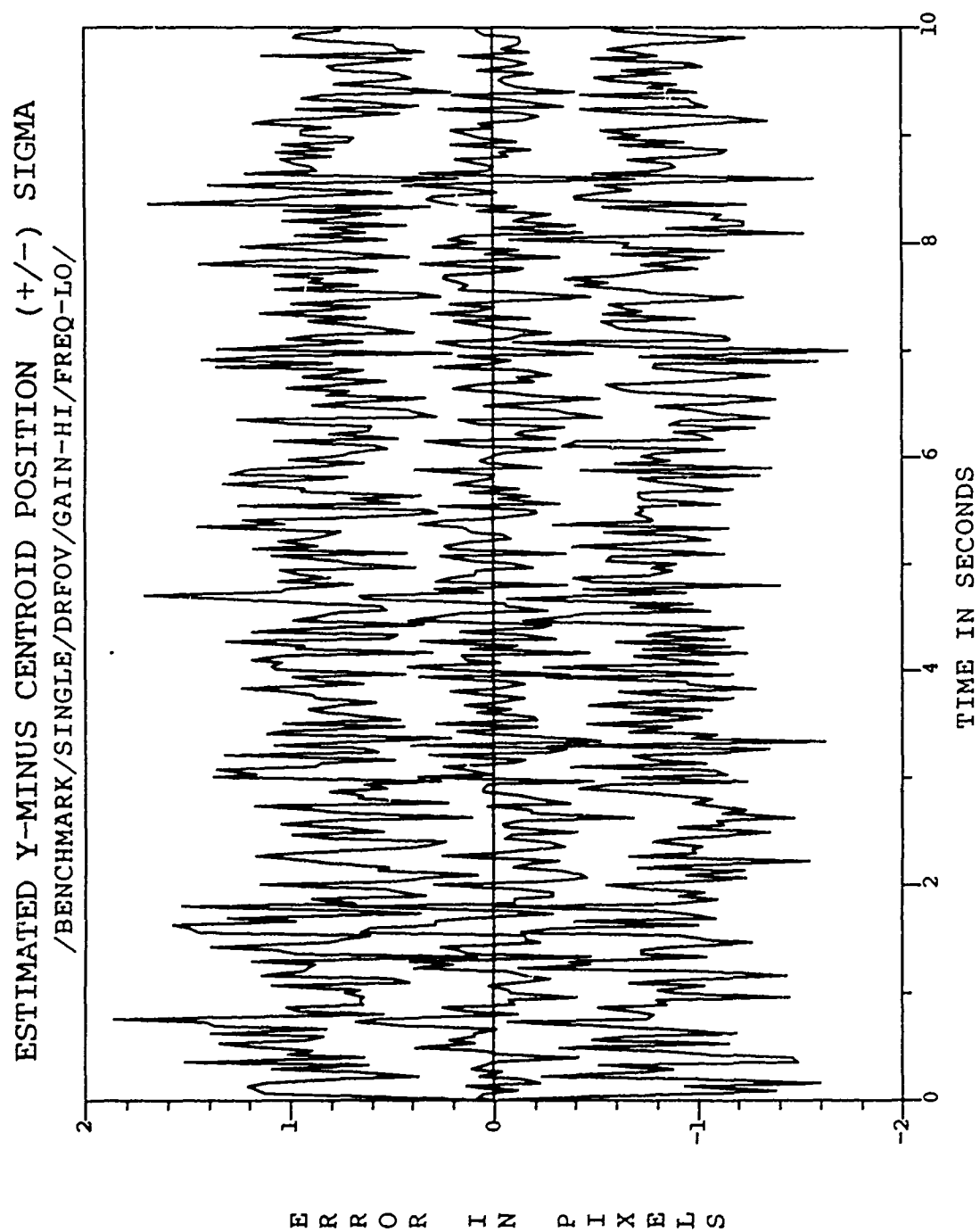


Figure C.28. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

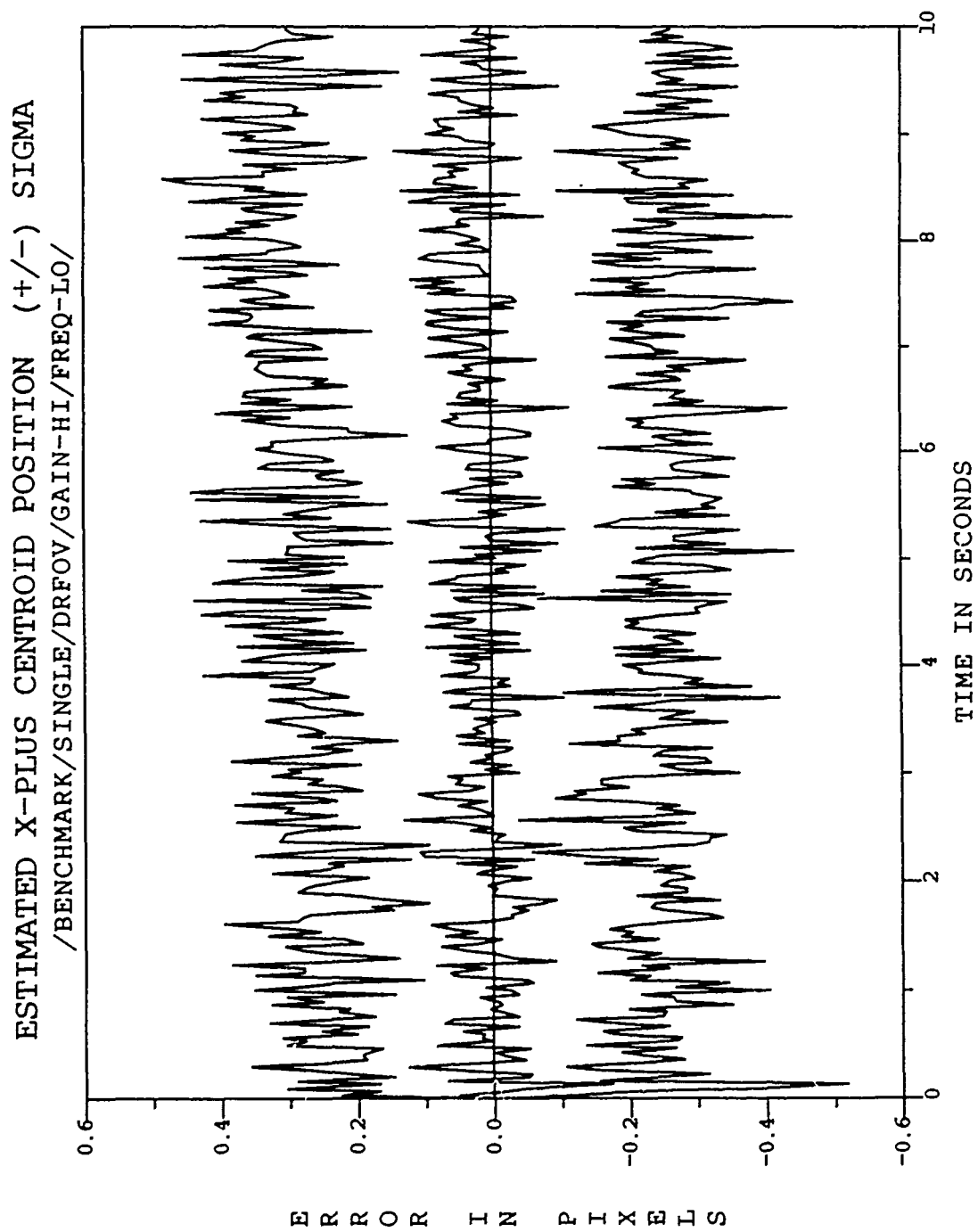


Figure C.29. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

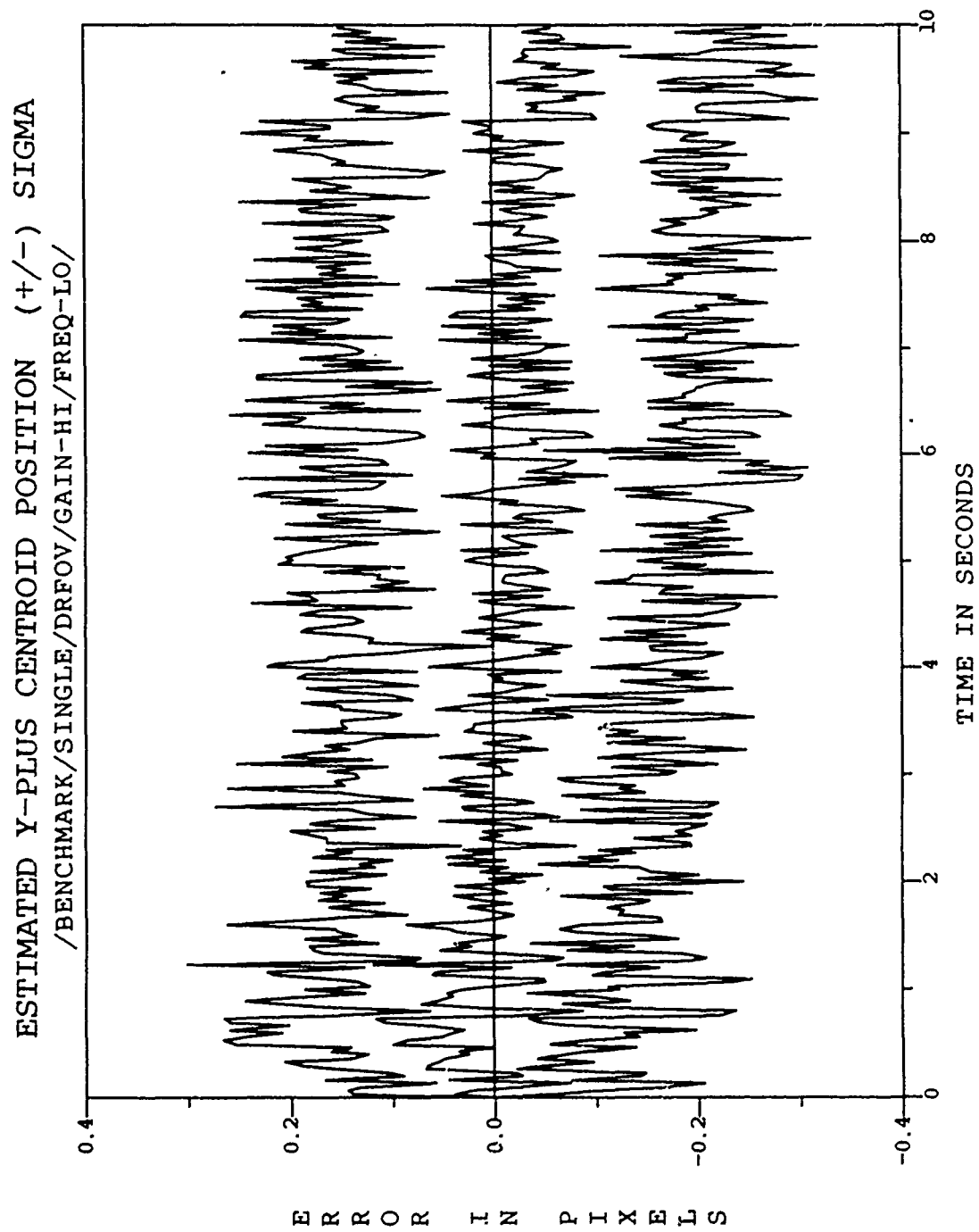


Figure C.30. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

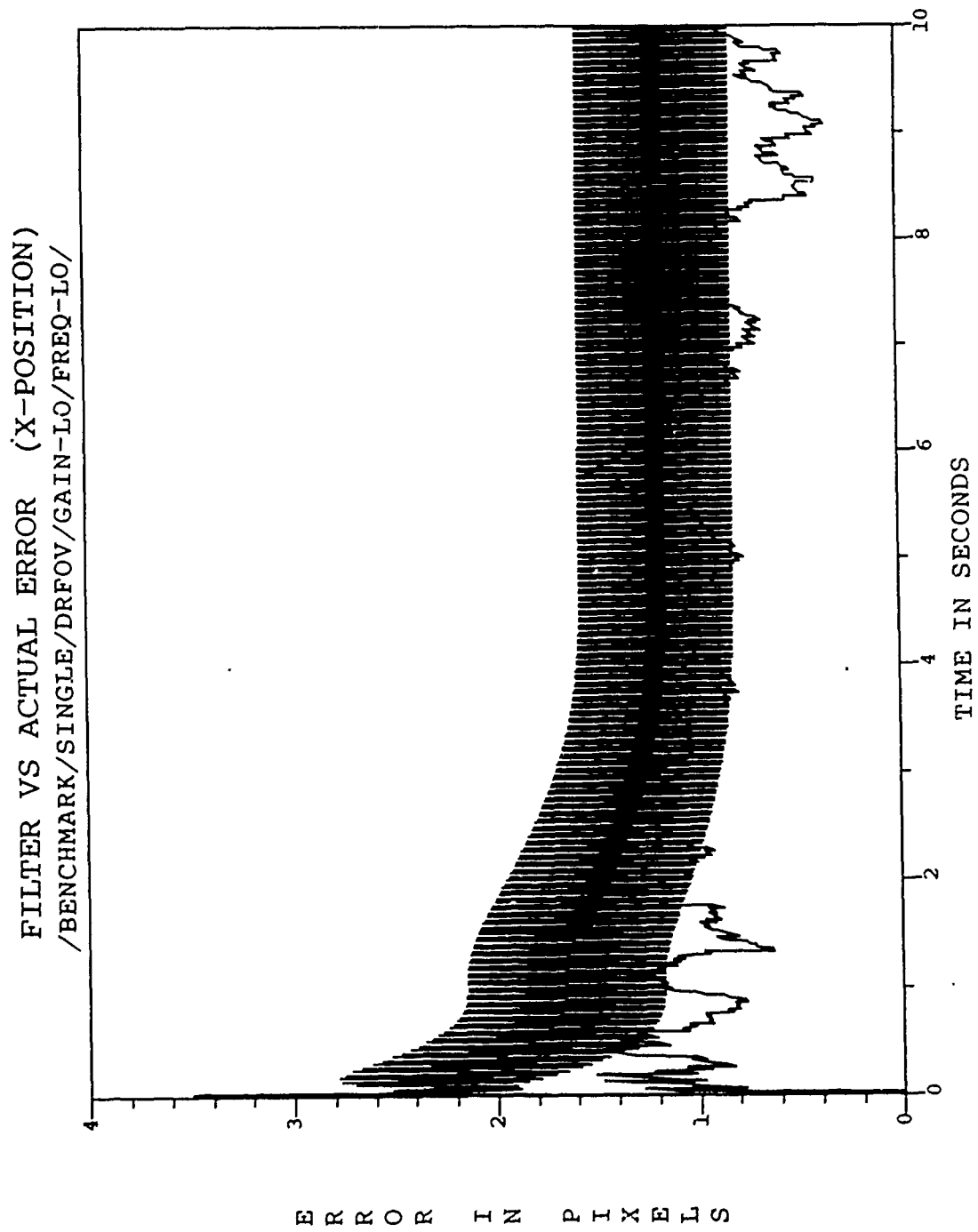


Figure C.31. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

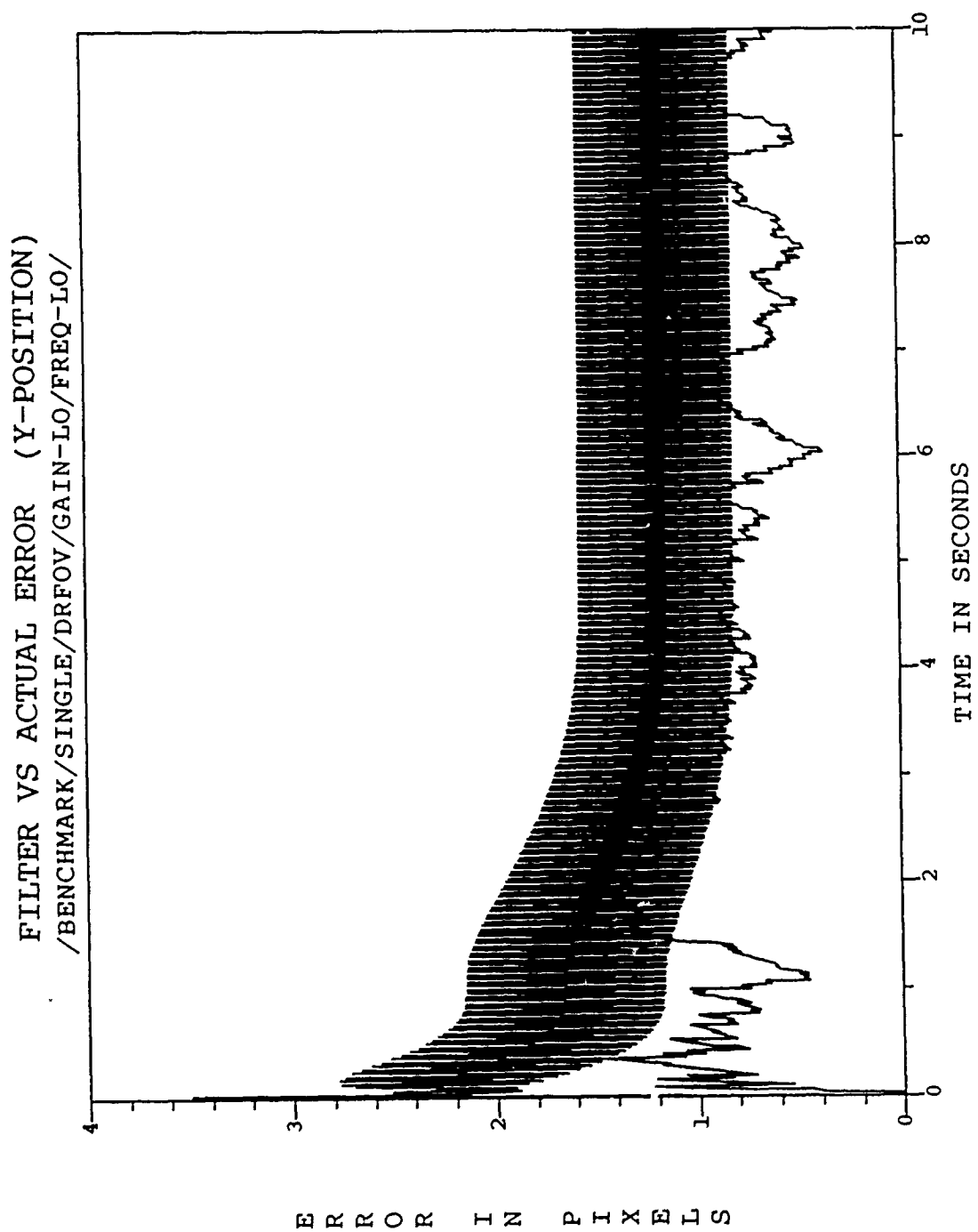


Figure C.32. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

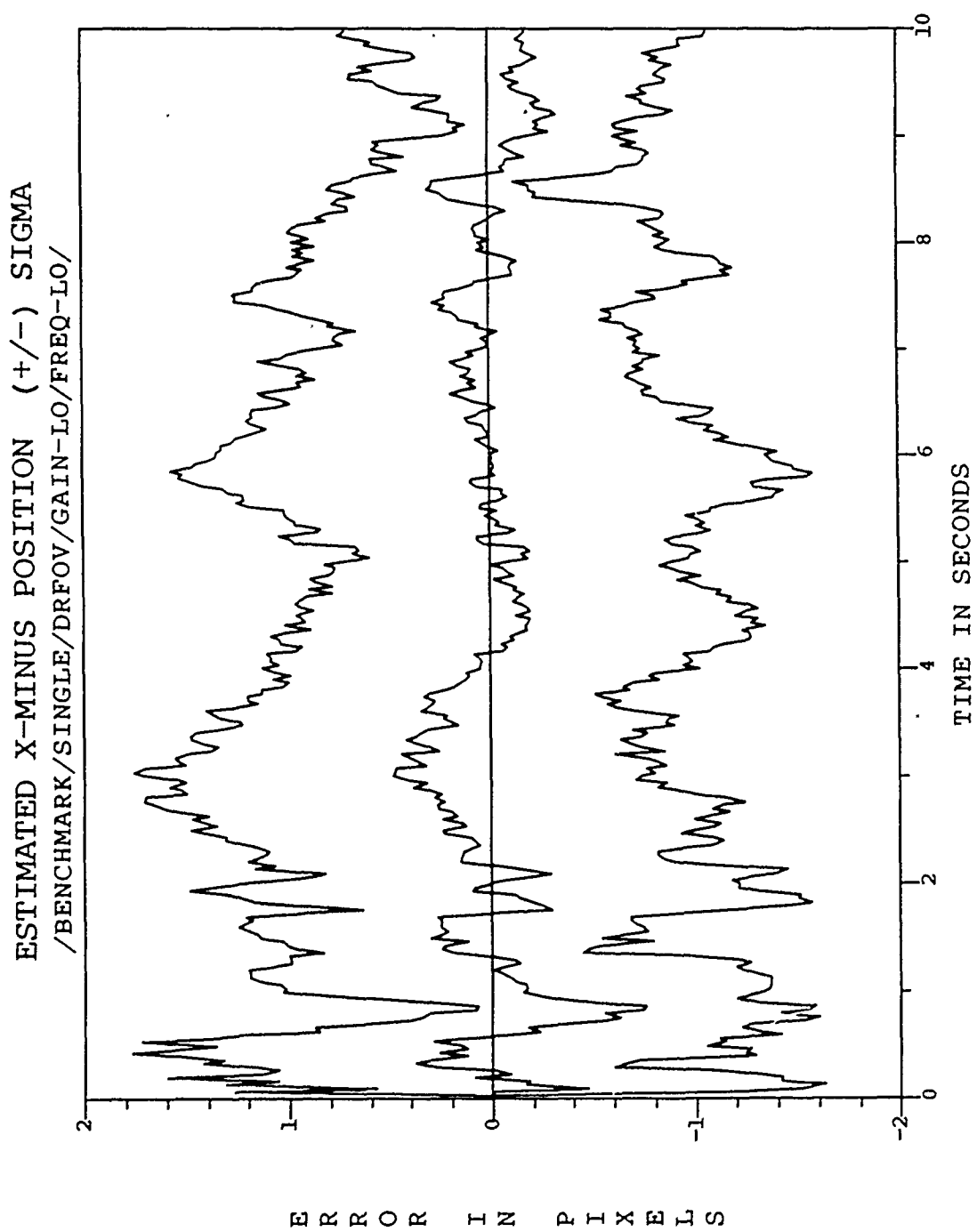


Figure C.33. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

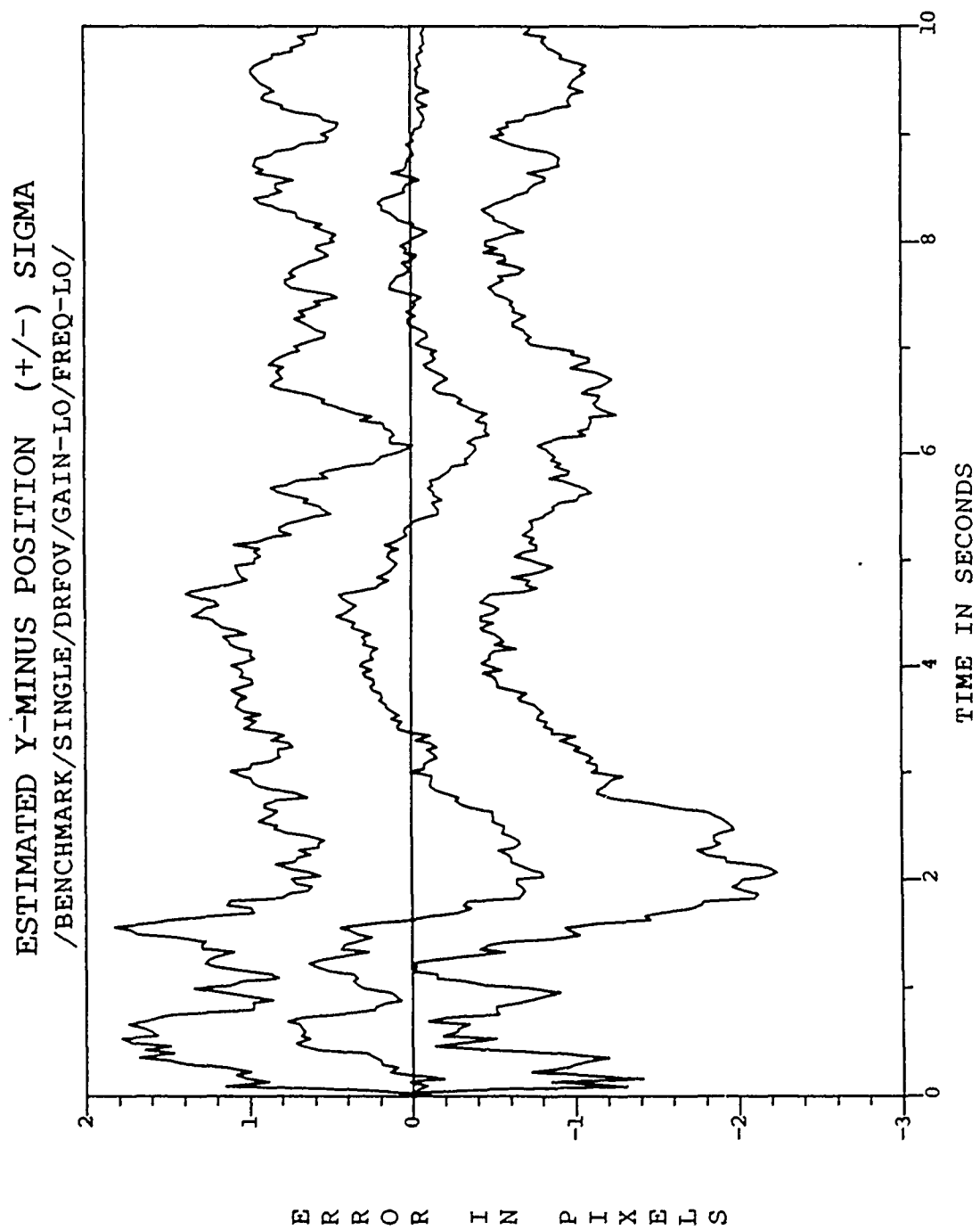


Figure C.34. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

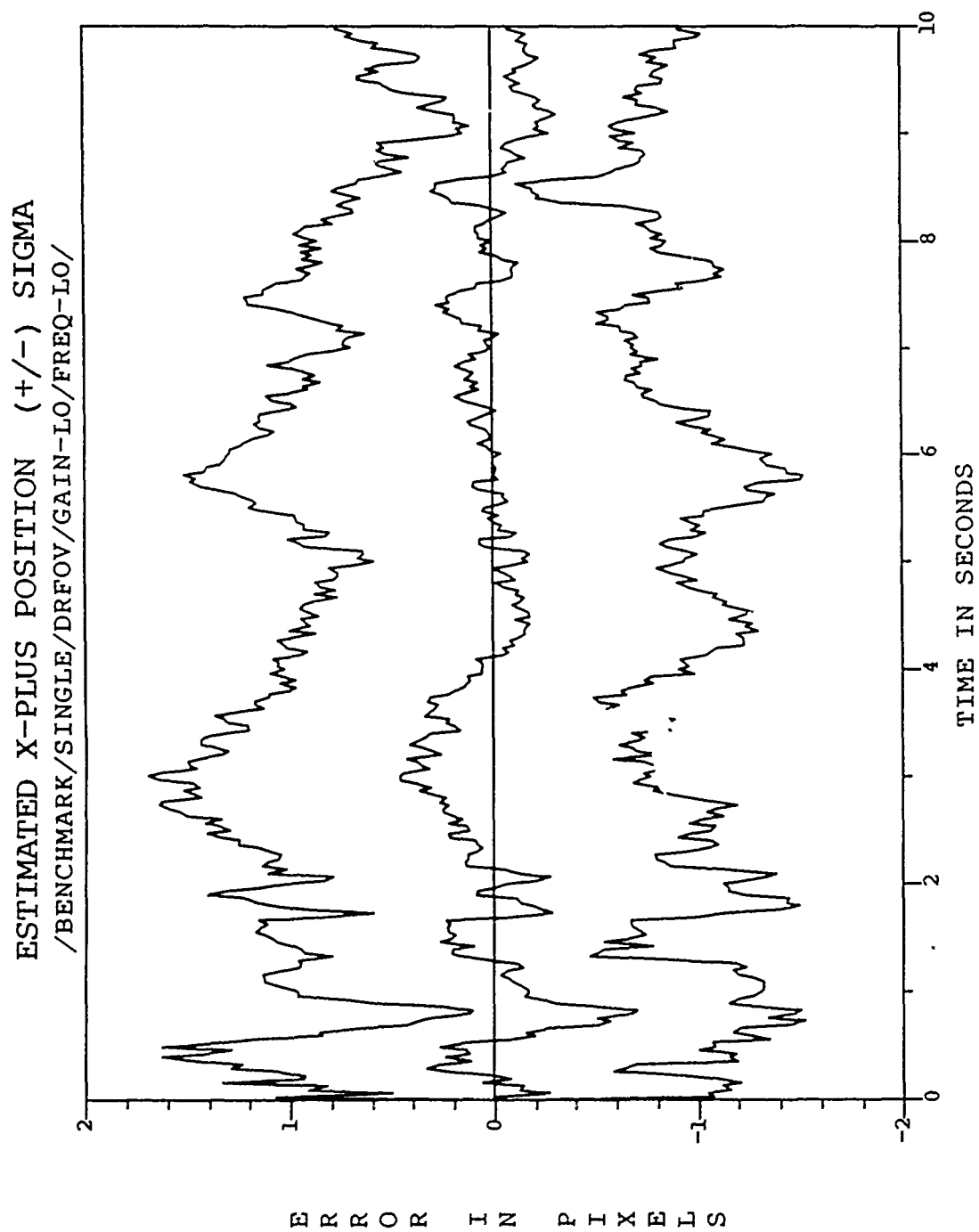


Figure C.35. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

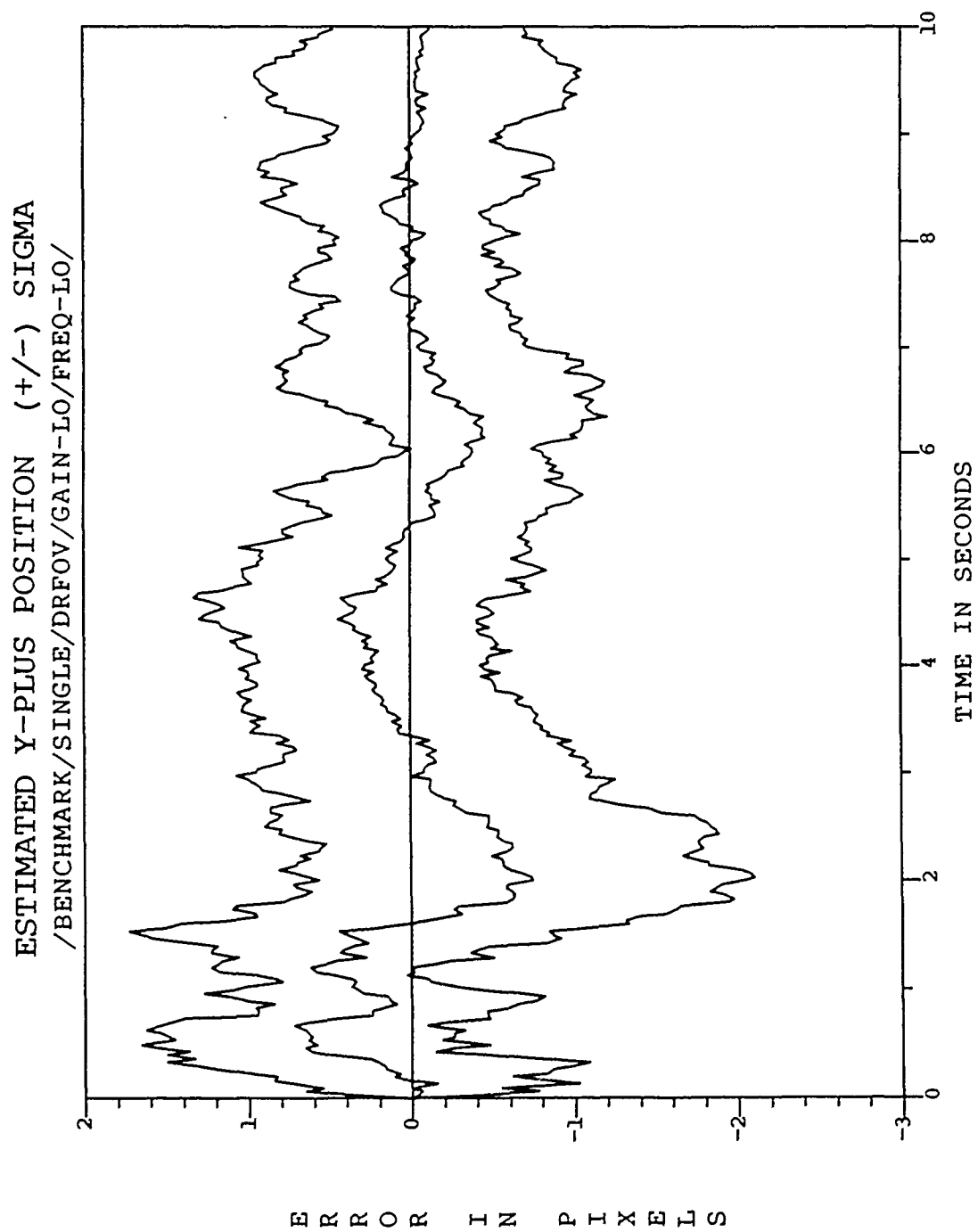


Figure C.36. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

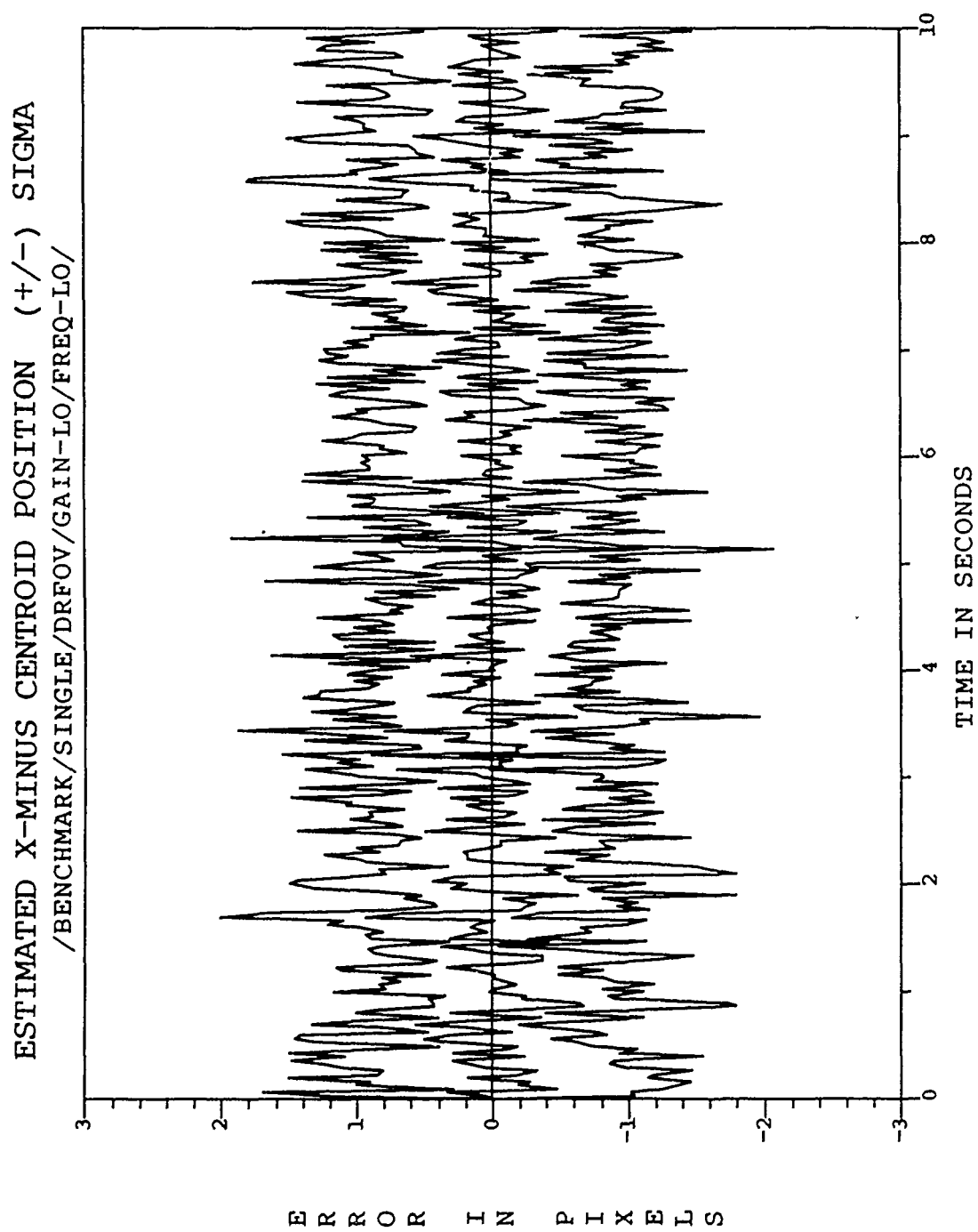


Figure C.37. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

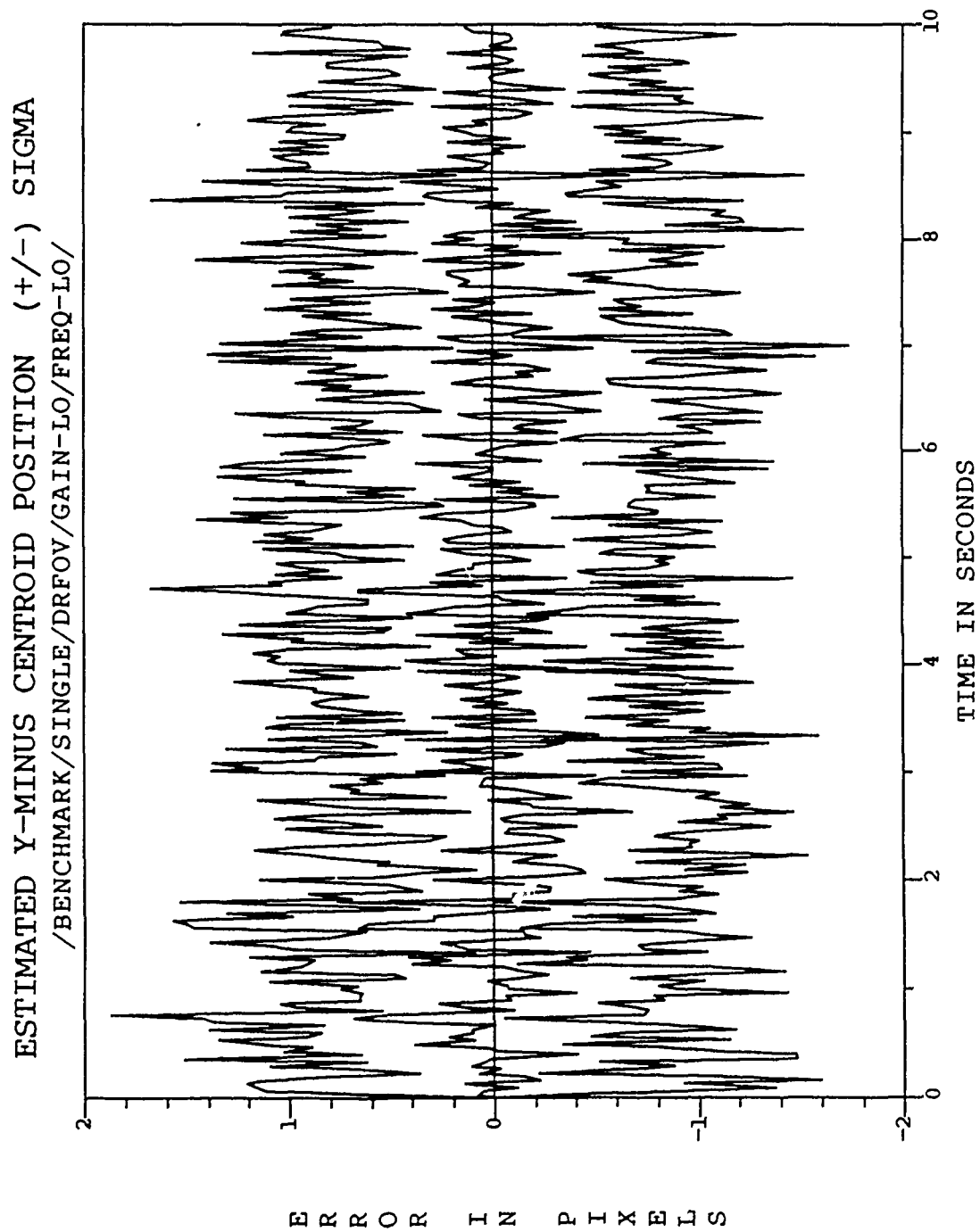


Figure C.38. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

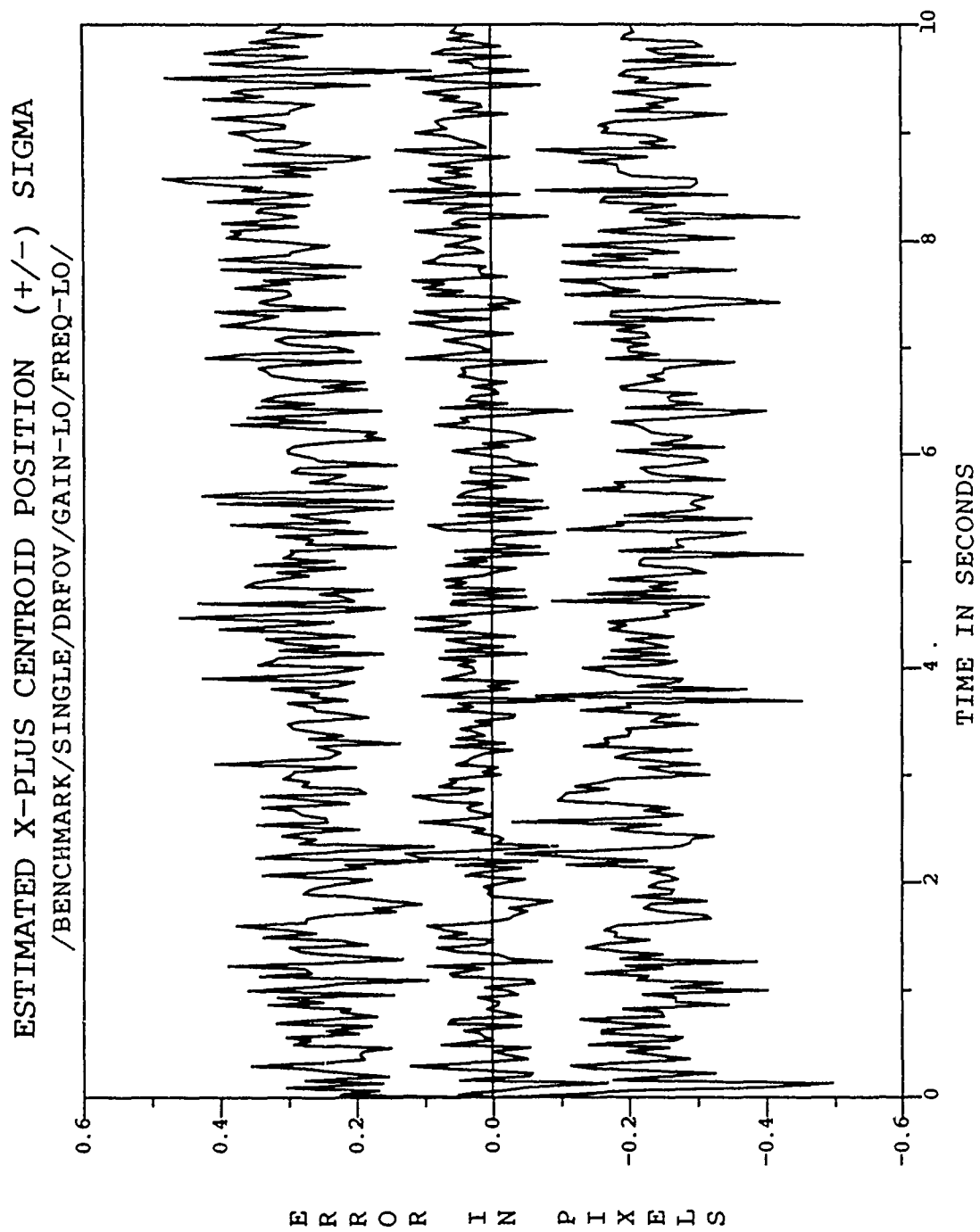


Figure C.39. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

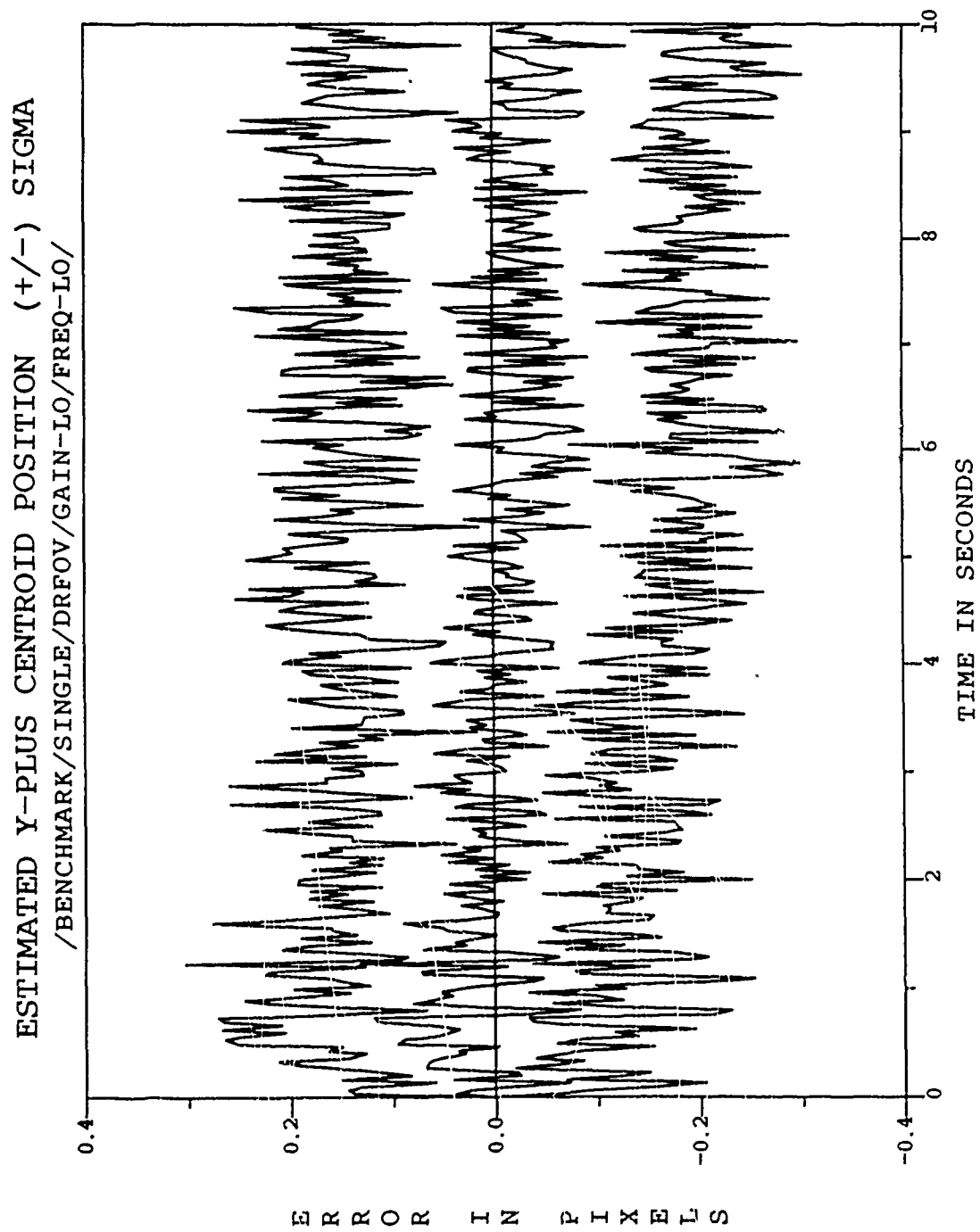


Figure C.40. /BENCHMARK/SINGLE/DRFOV/GAIN-LO/FREQ-LO/

Appendix D. *Plots for FOV Rotation Analyses: Discussion in*
Section 6.4

D.1 Figures D.1-D.10: NRFOV, $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.45$

D.2 Figures D.11-D.20: RFOV, $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 1.95$

D.3 Figures D.21-D.30: DRFOV, $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.10$

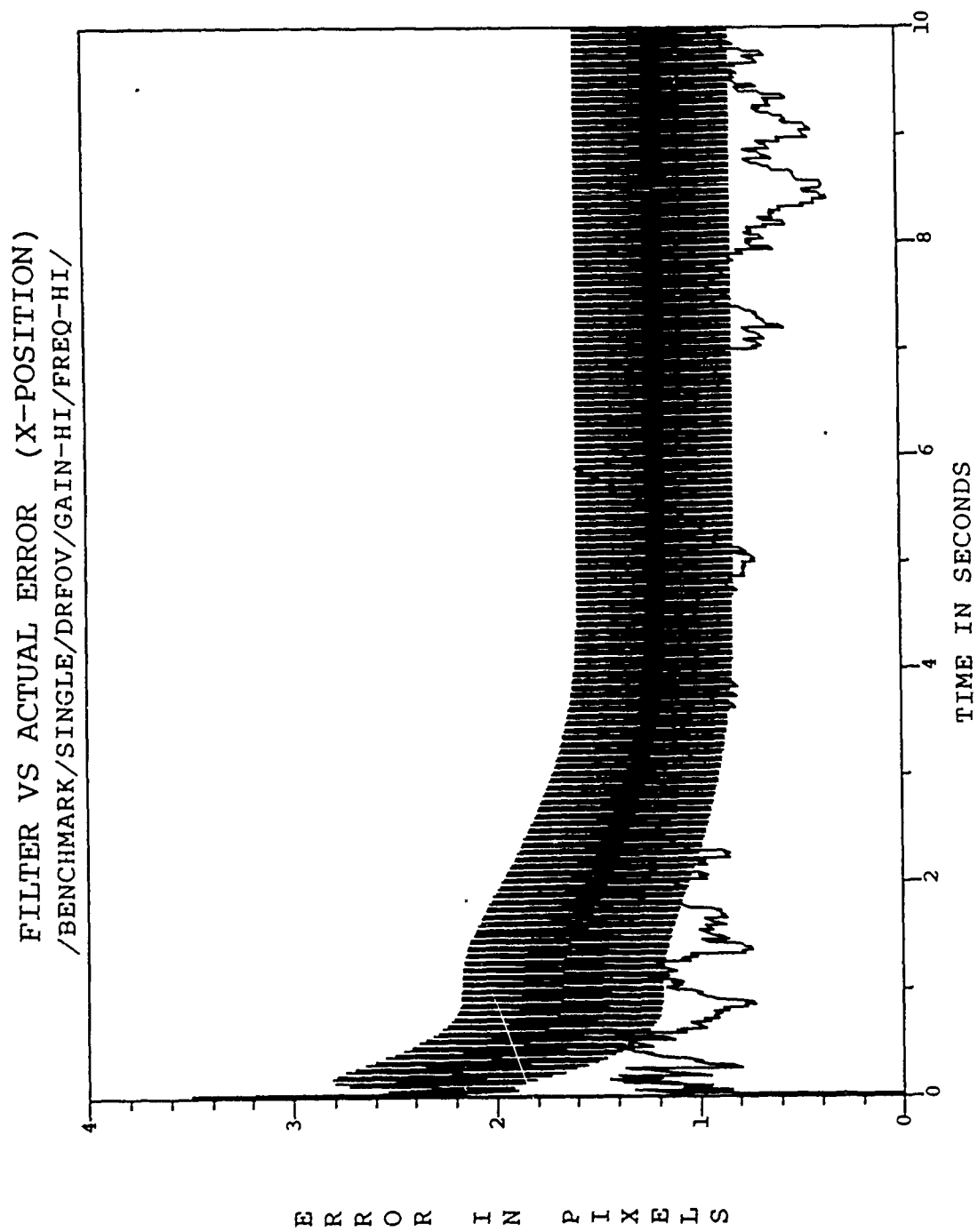


Figure D.1. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

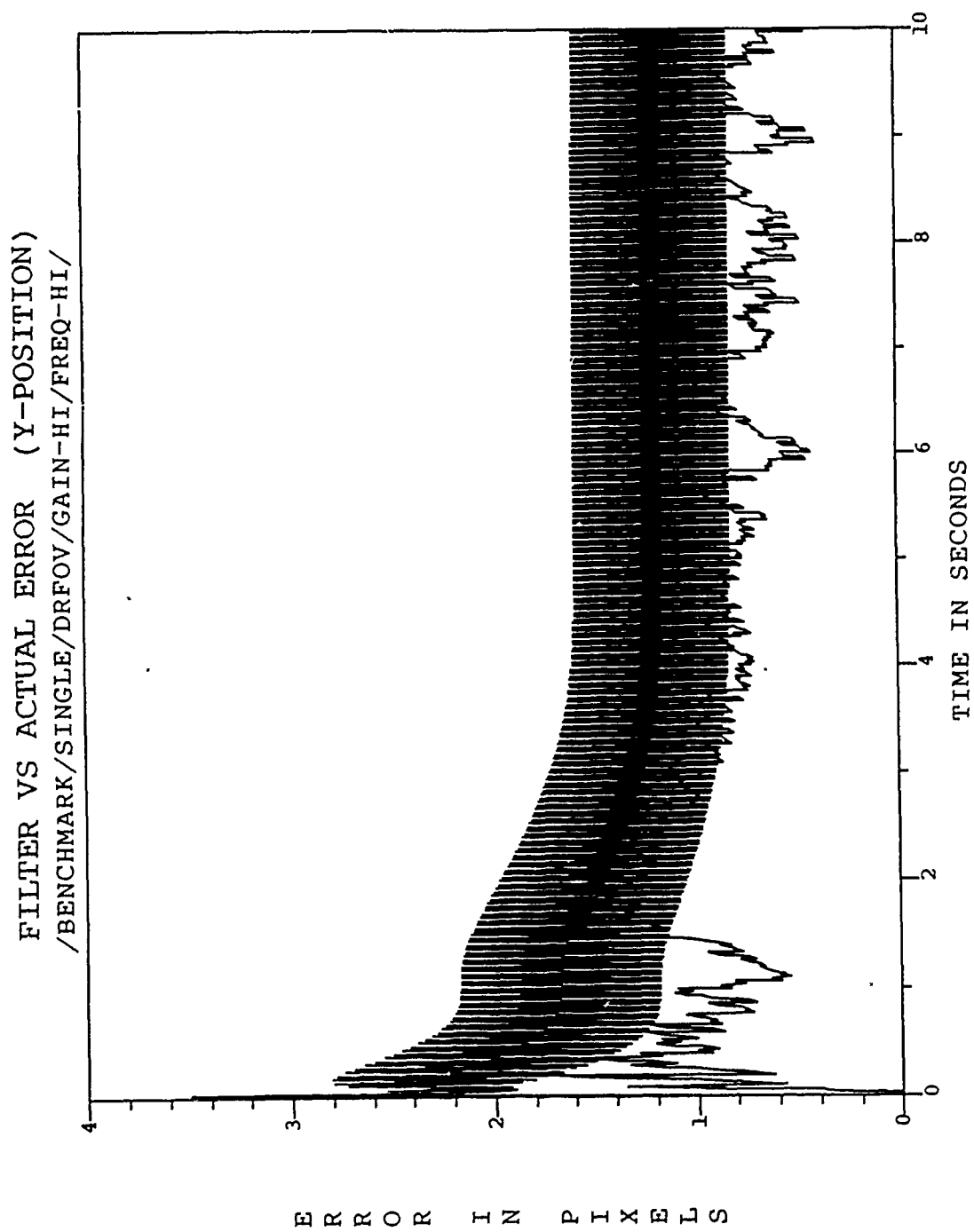


Figure D.2. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

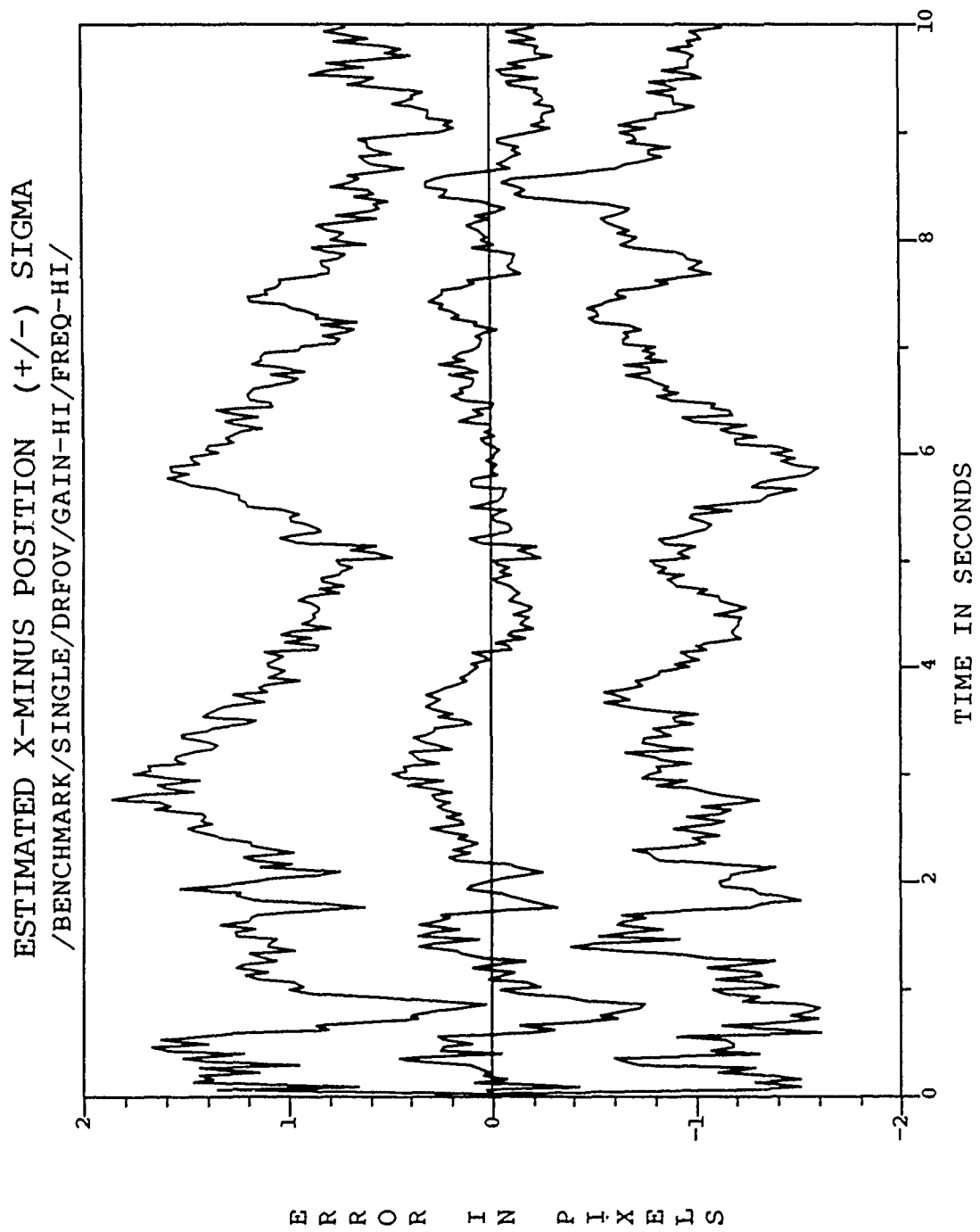


Figure D.3. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

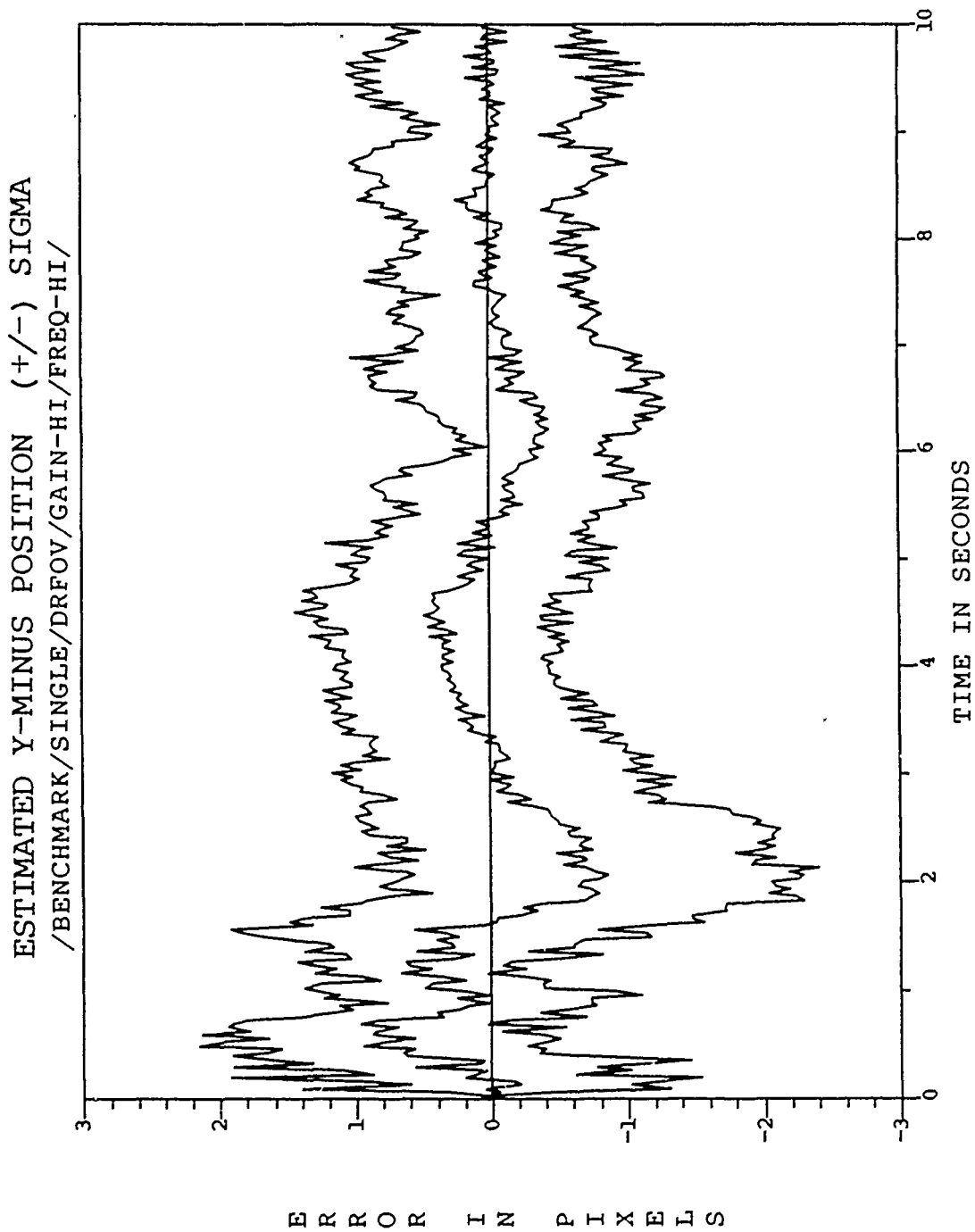


Figure D.1. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

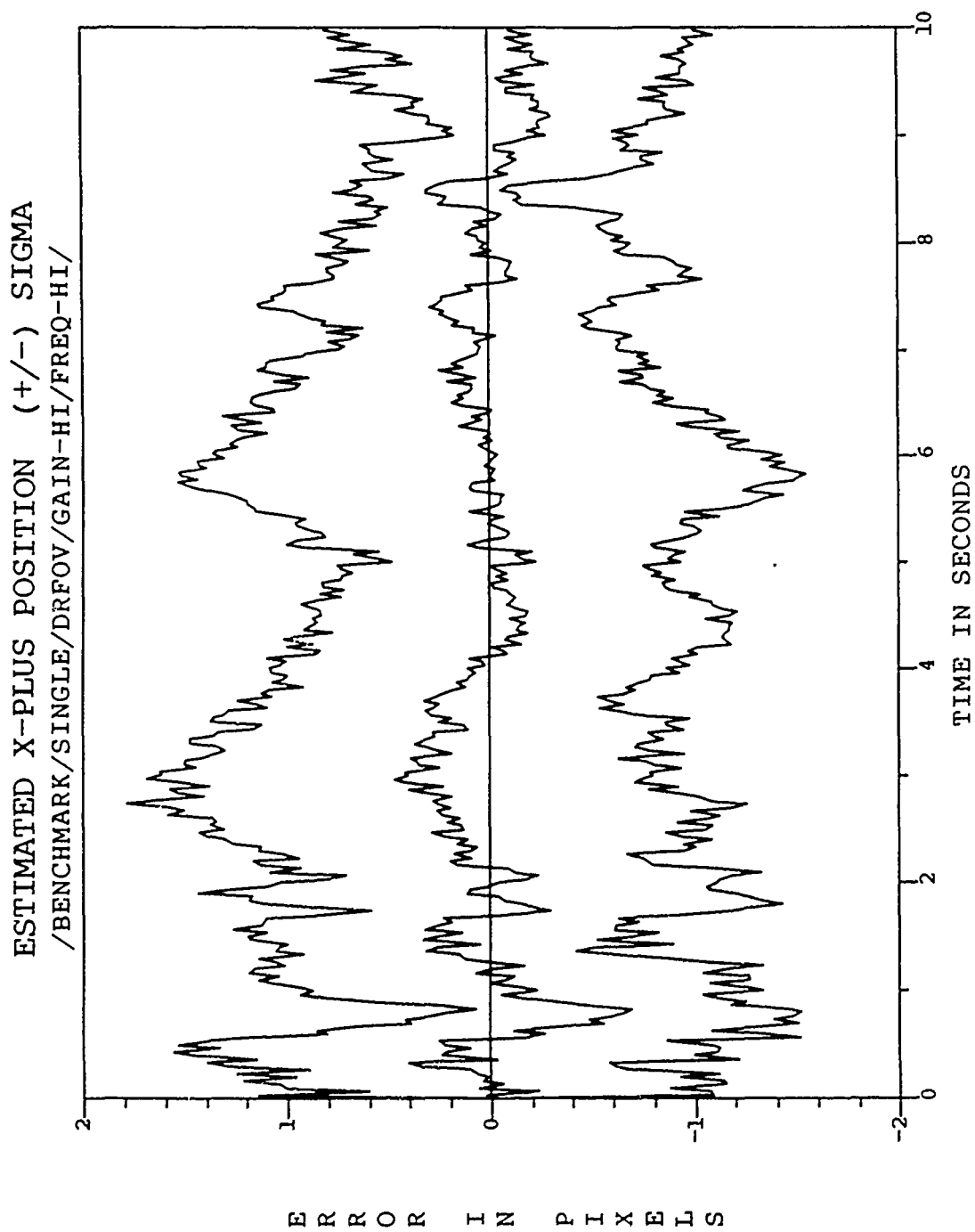


Figure D.5. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

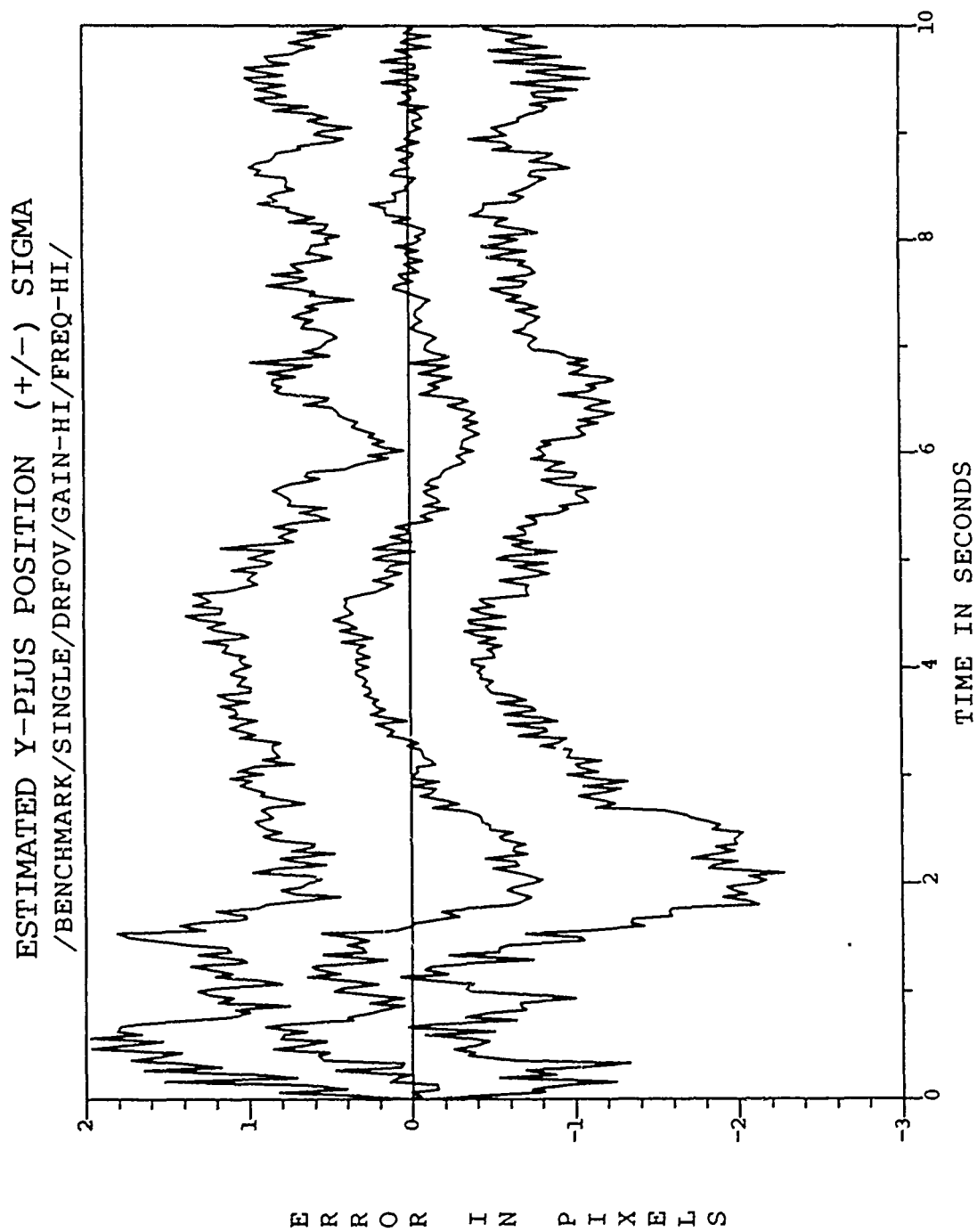


Figure D.6. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-III/

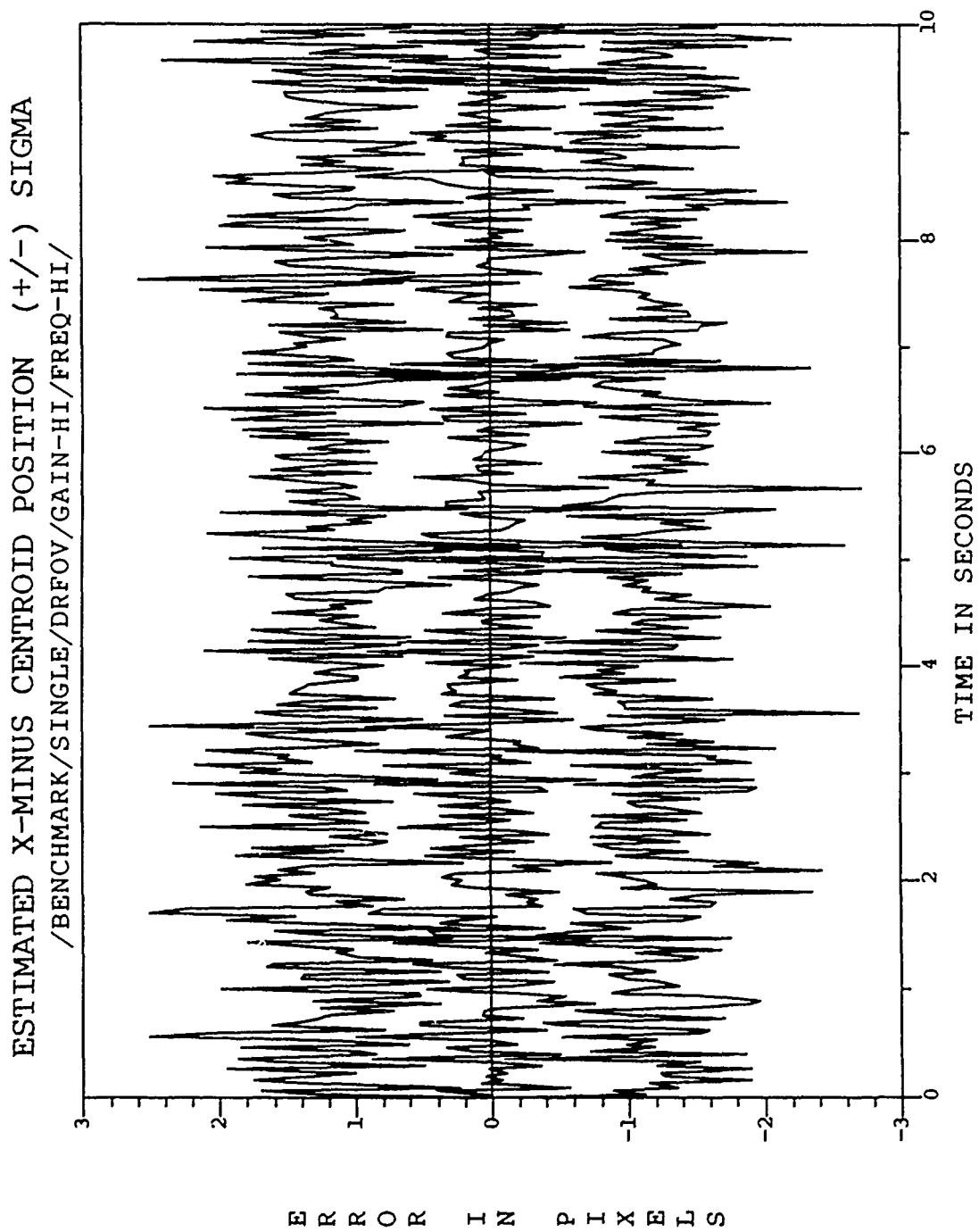


Figure D.7. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

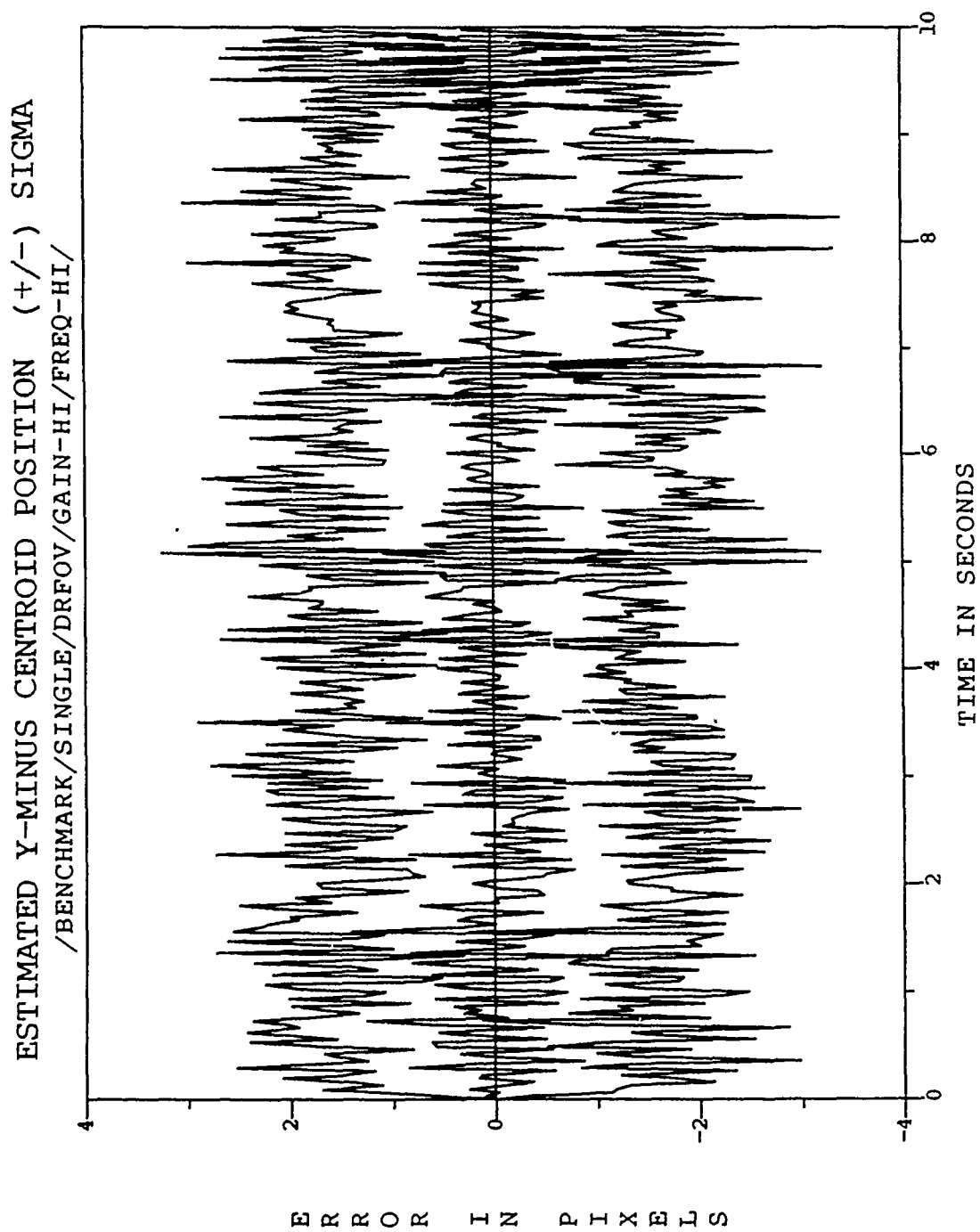


Figure D.8. /BENCHMARK/SINGLE/NRFOV/GAIN-III/FREQ-III/

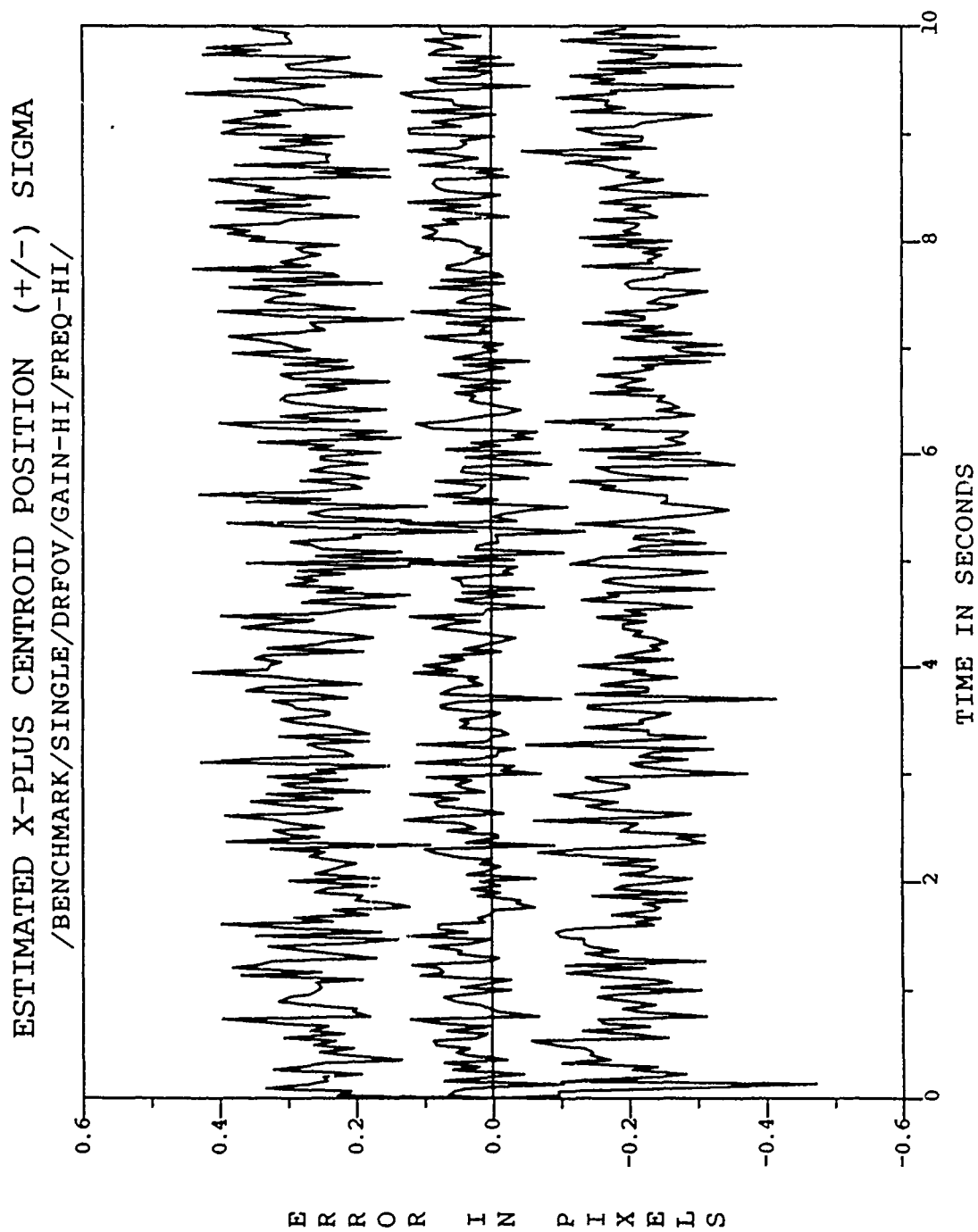


Figure D.9. /BENCHMARK/SINGLE/NRFOV/GAIN-III/FREQ-III/

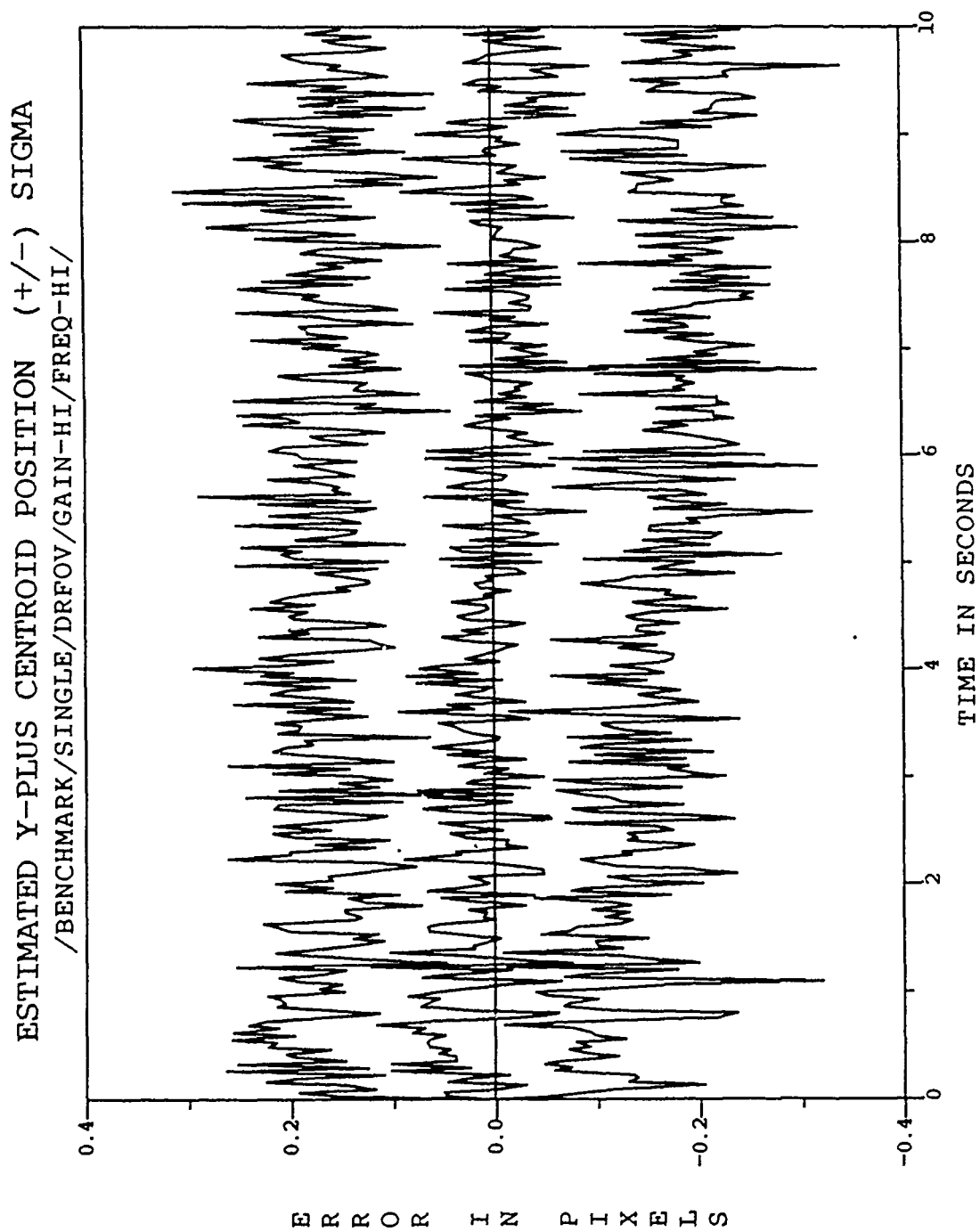


Figure D.10. /BENCHMARK/SINGLE/NRFOV/GAIN-HI/FREQ-HI/

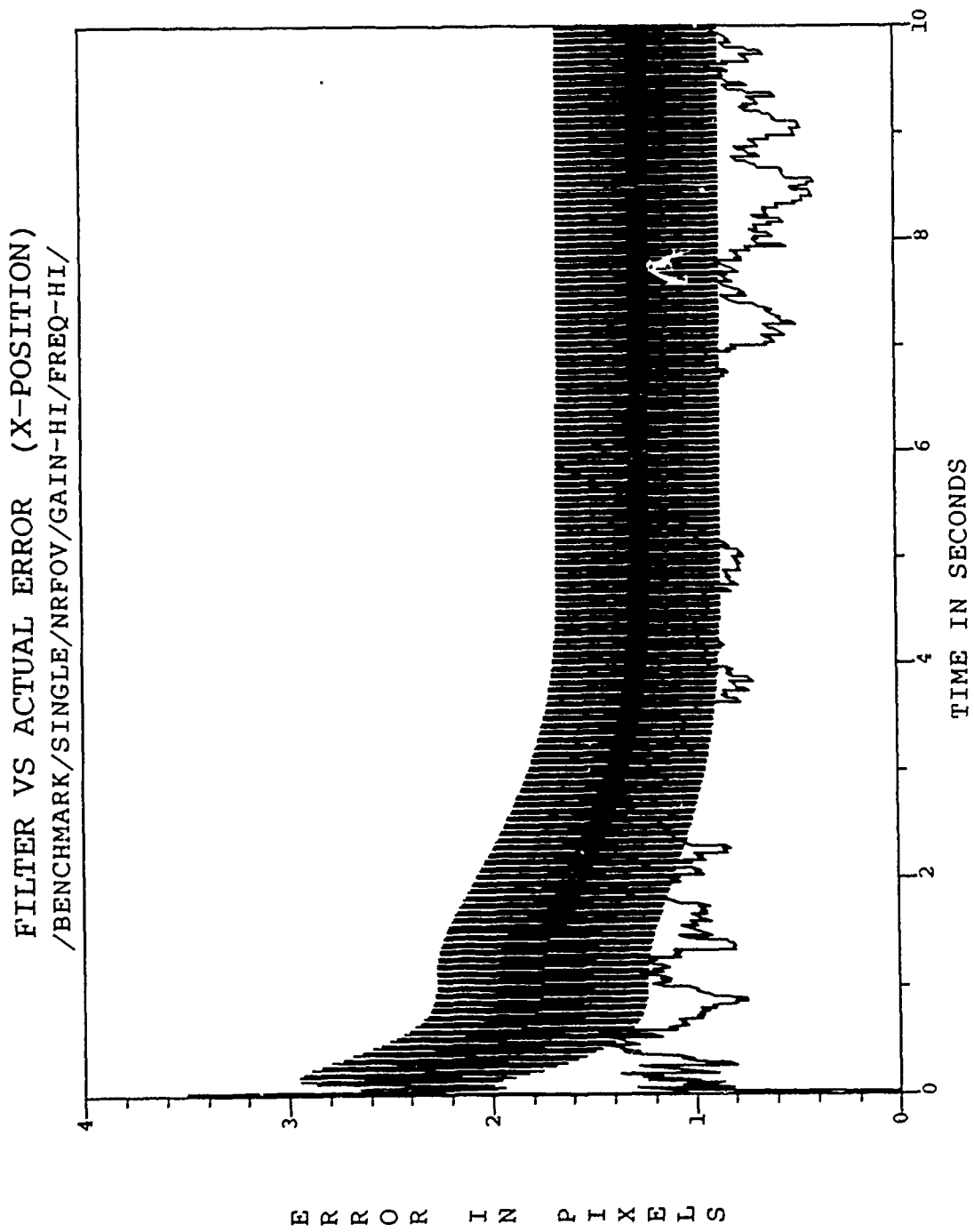


Figure D.11. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

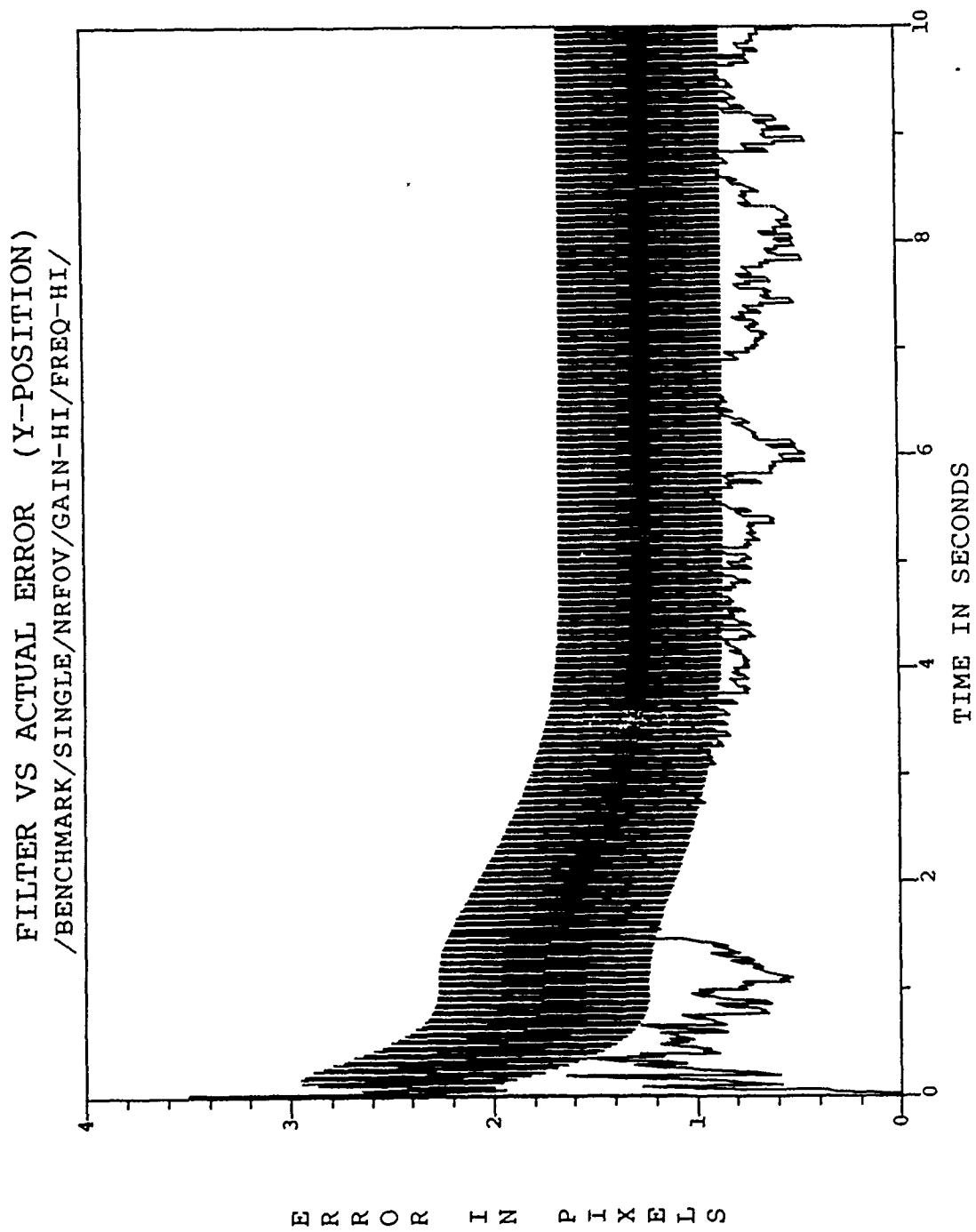


Figure D.12. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

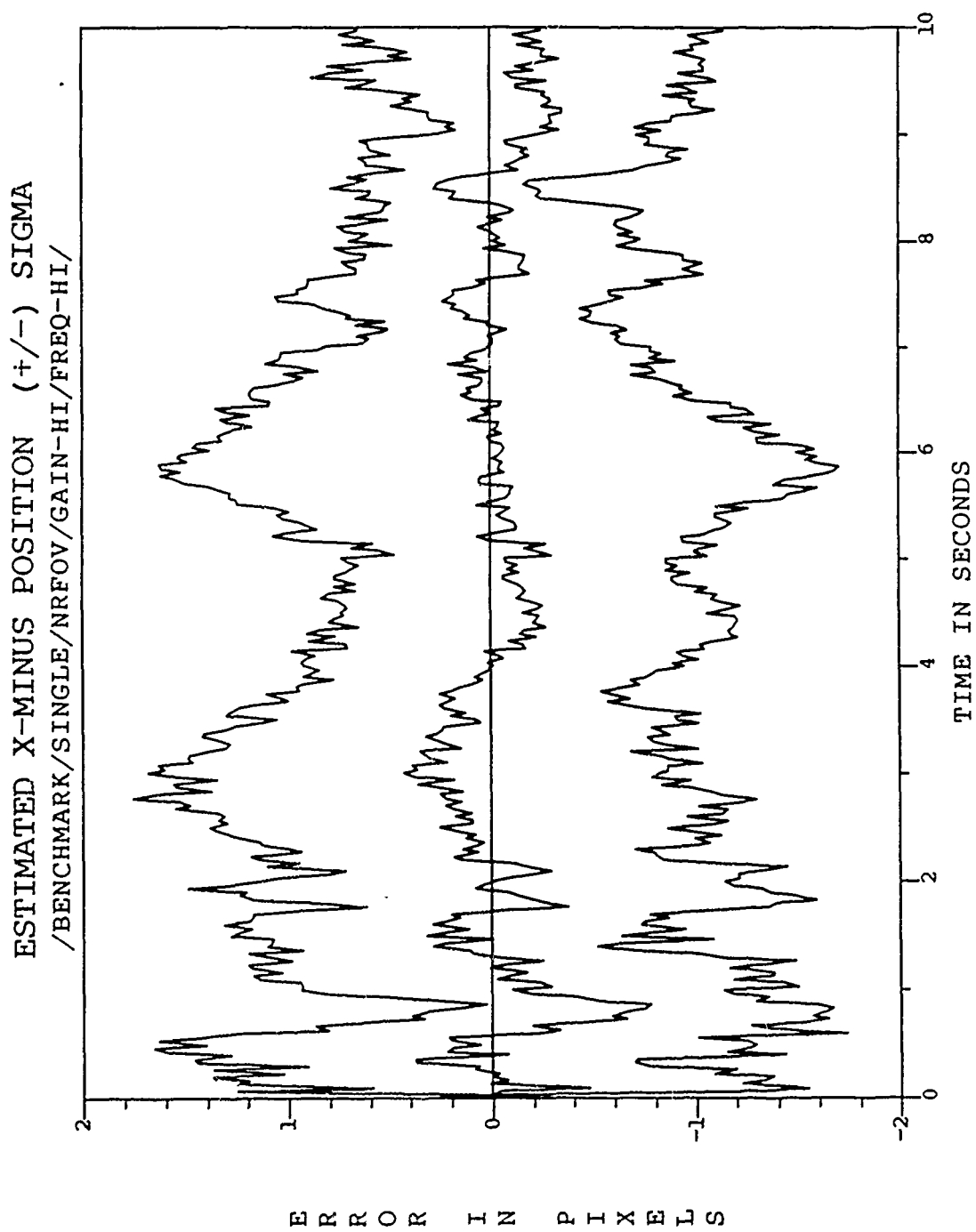


Figure D.13. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

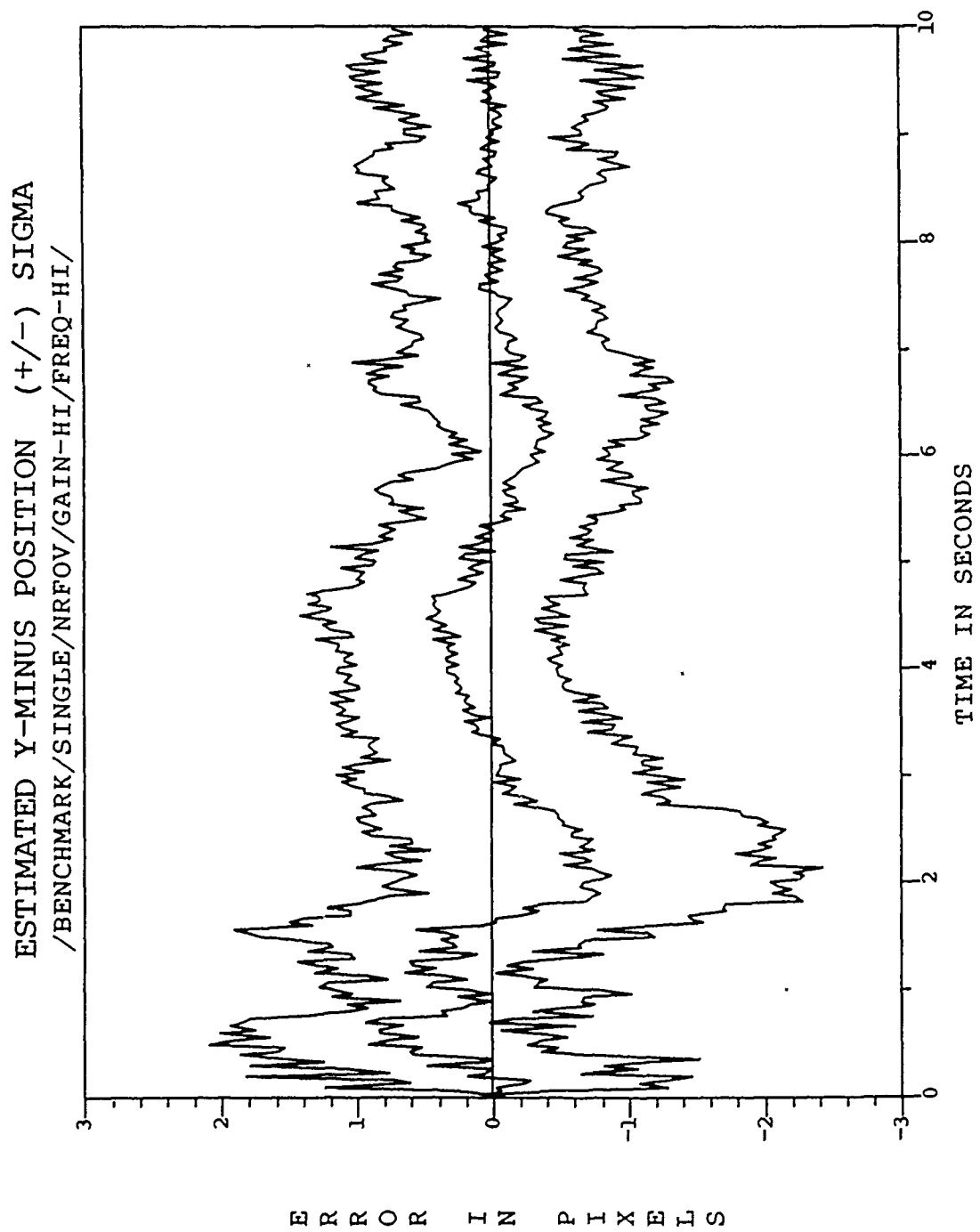


Figure D.14. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

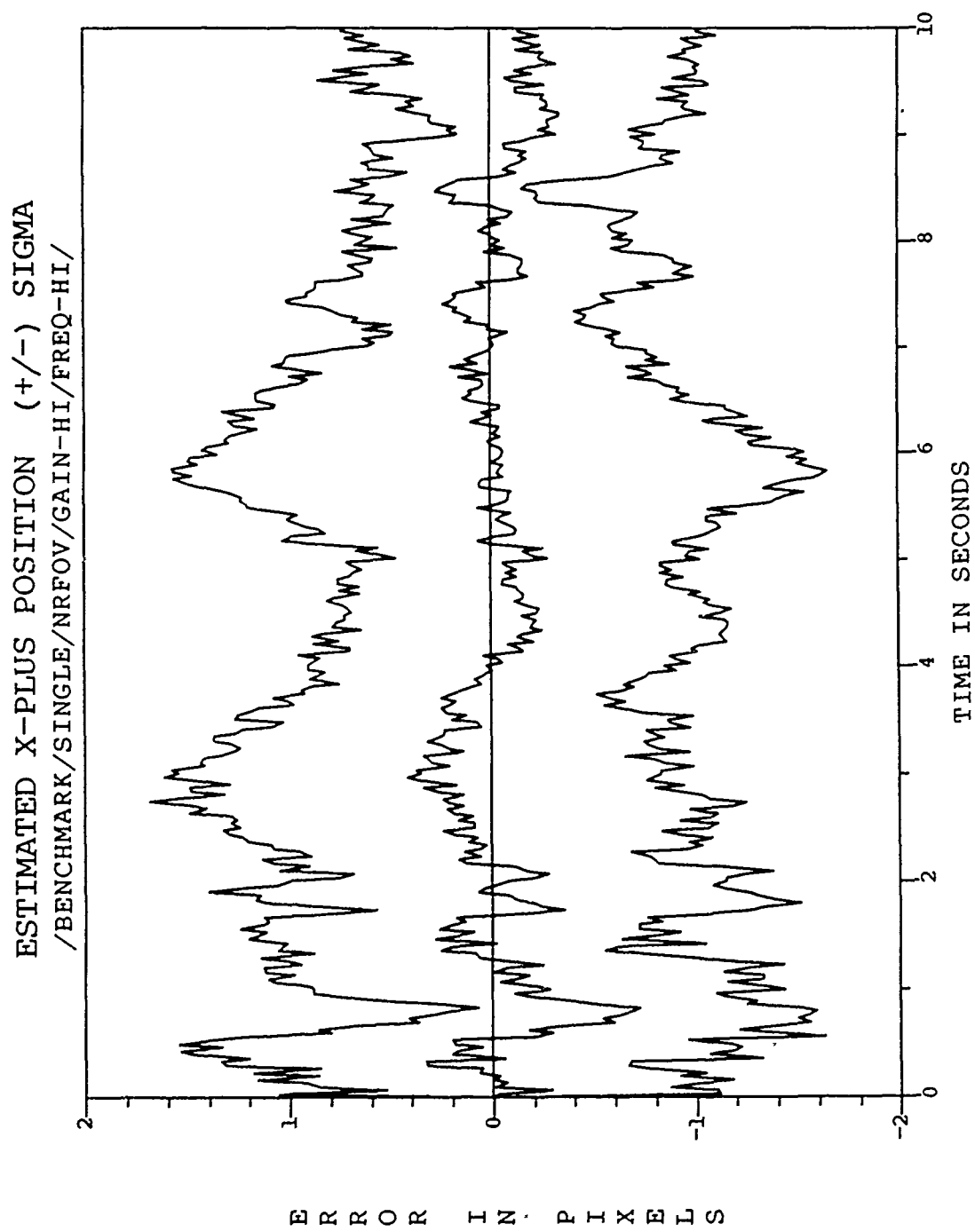


Figure D.15. /BENCHMARK/SINGLE/RFOV/GAIN-HI/FREQ-HI/

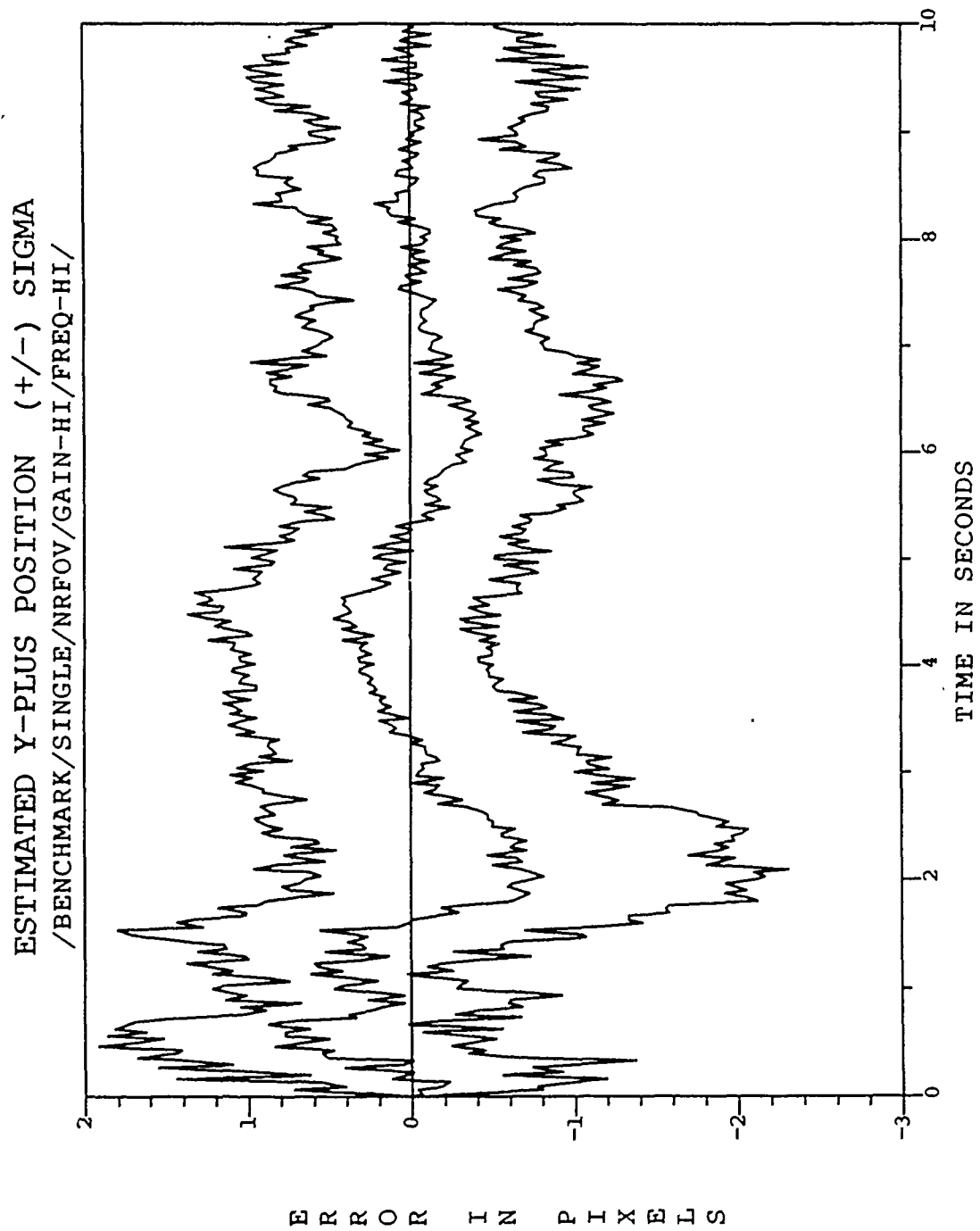


Figure D.16. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

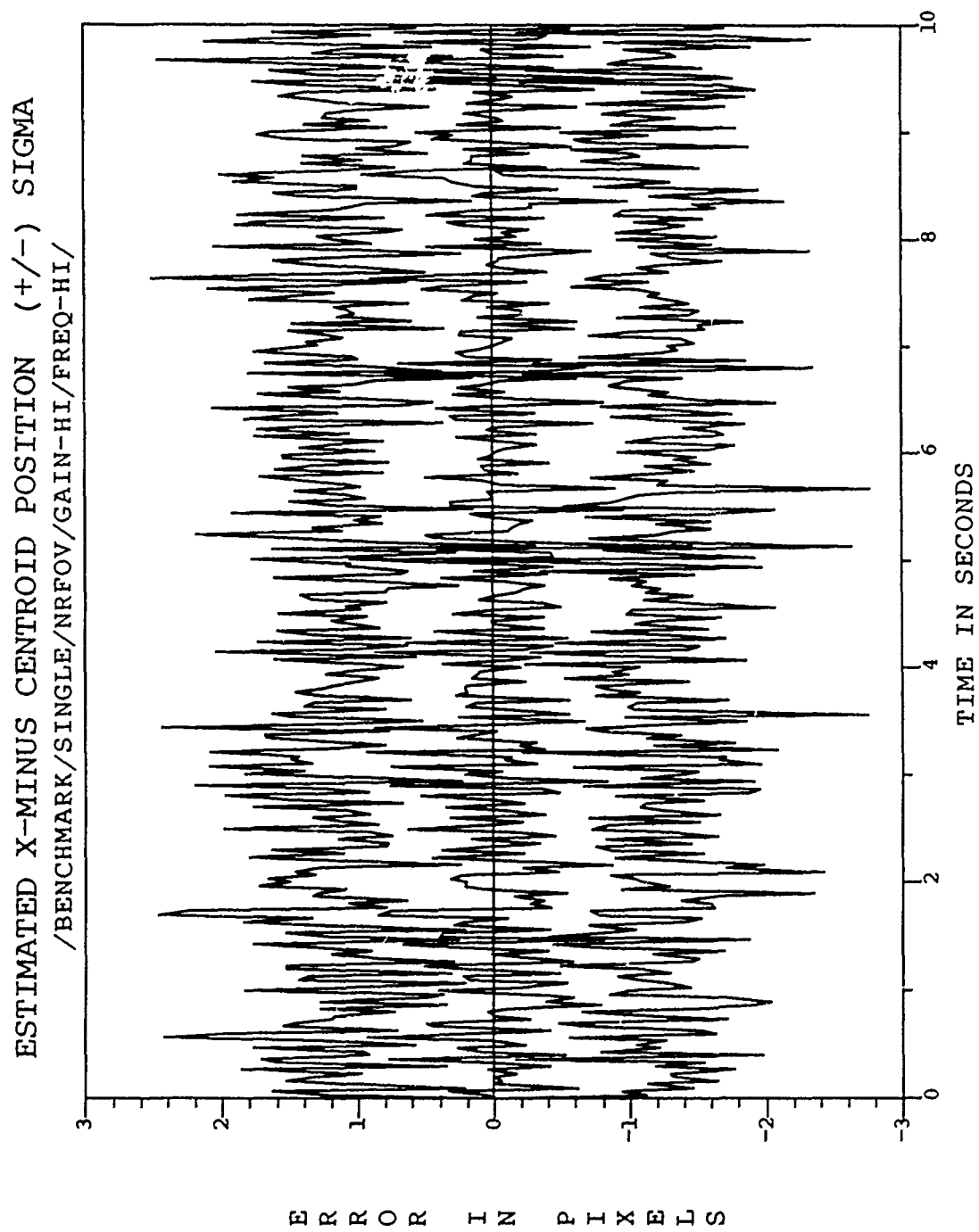


Figure D.17. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

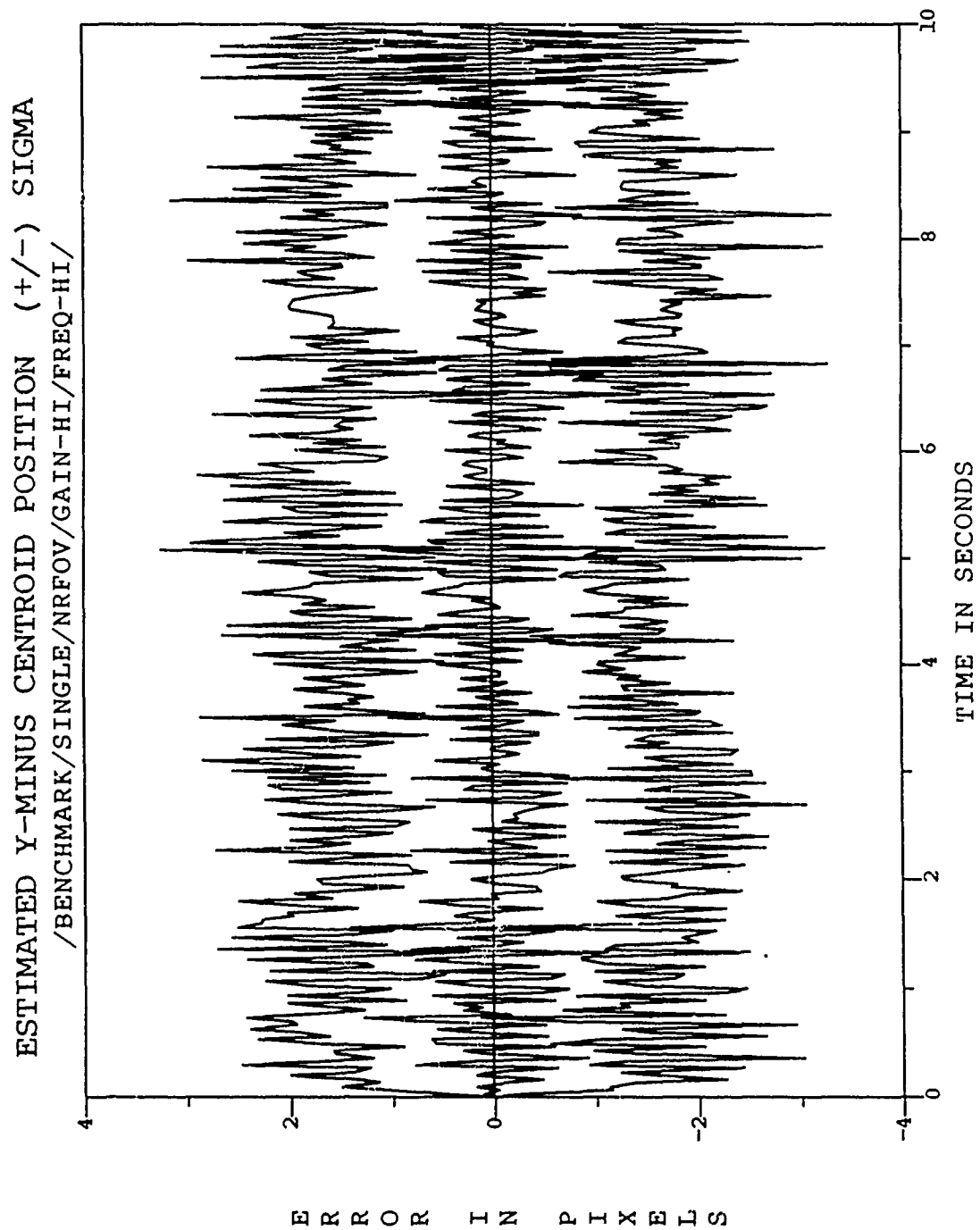


Figure D.18. /BENCHMARK/SINGLE/RFOV/GAIN-HI/FREQ-HI/

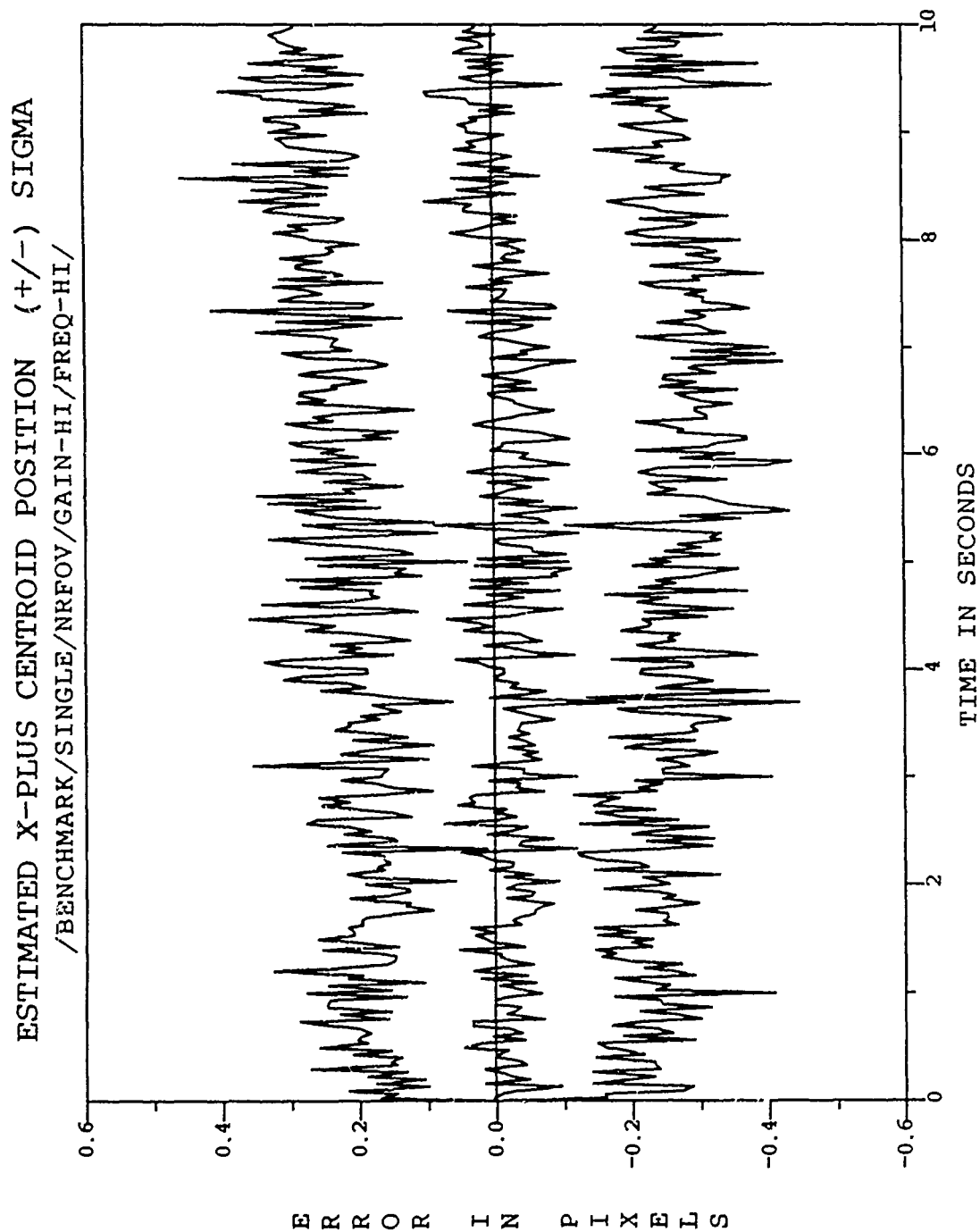


Figure D.19. /BENCHMARK/SINGLE/RFOV/GAIN-III/FREQ-III/

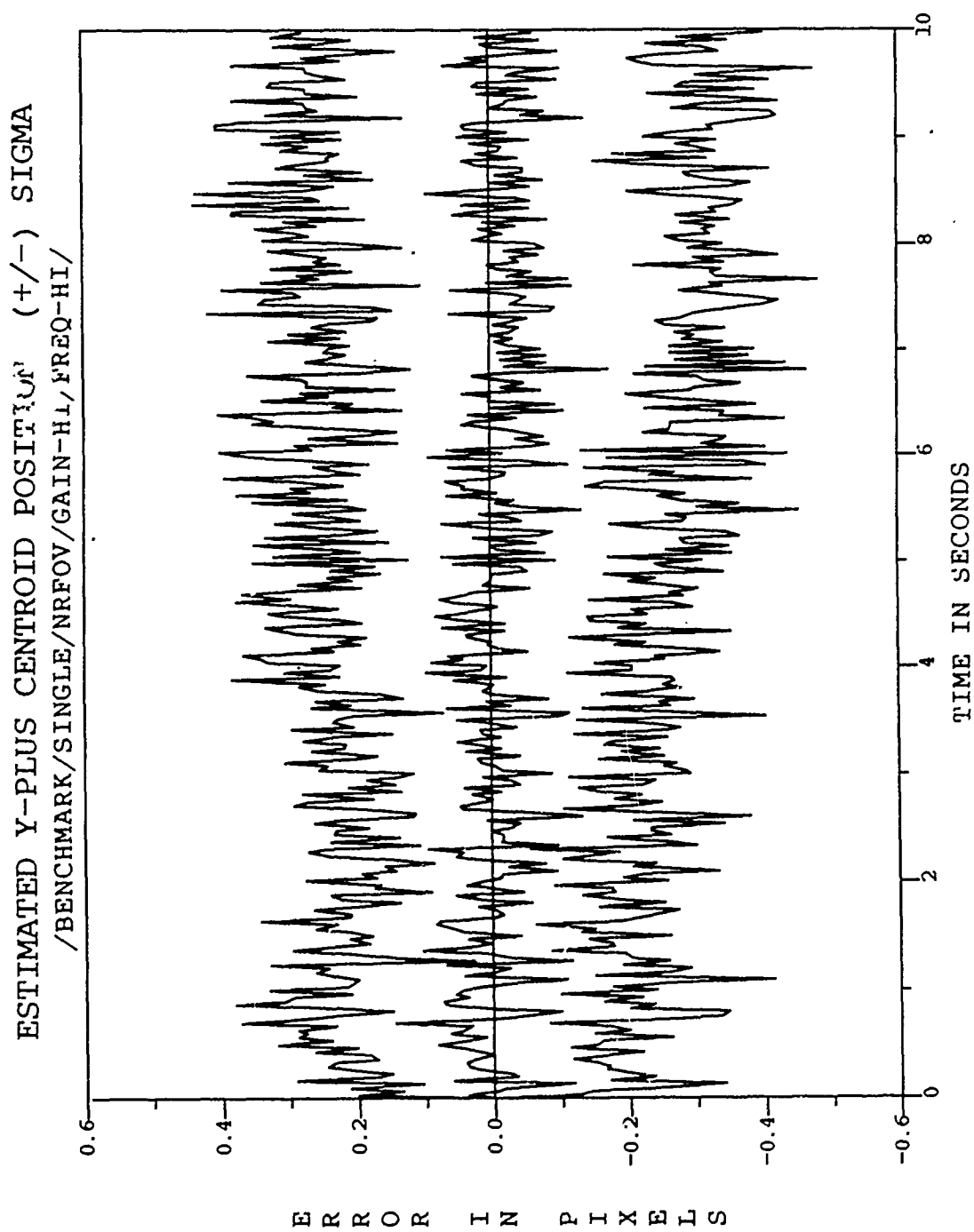


Figure D.20. /BENCHMARK/SINGLE/RFOV/GAIN-HI/FREQ-HI/

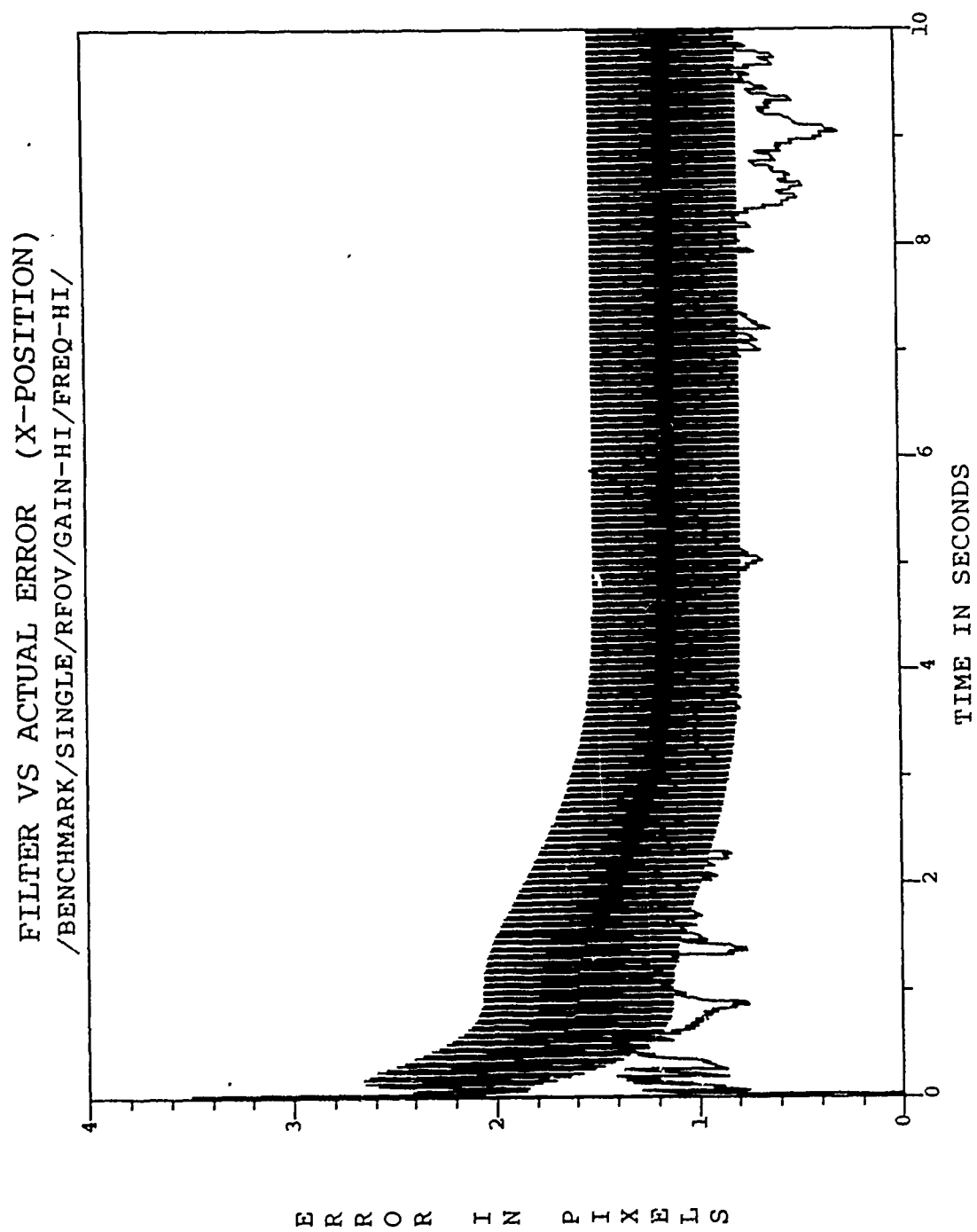


Figure D.21. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

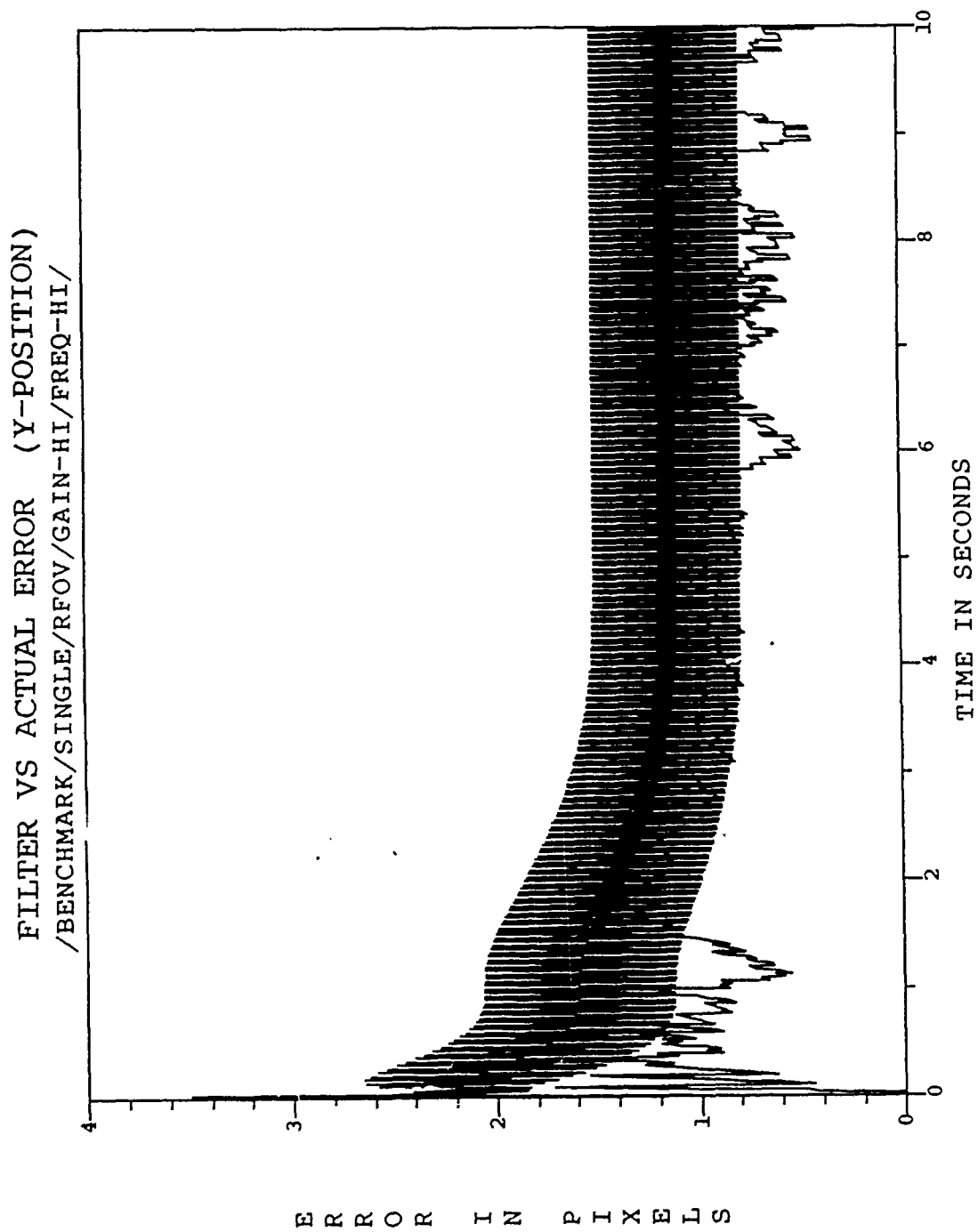


Figure D.22. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

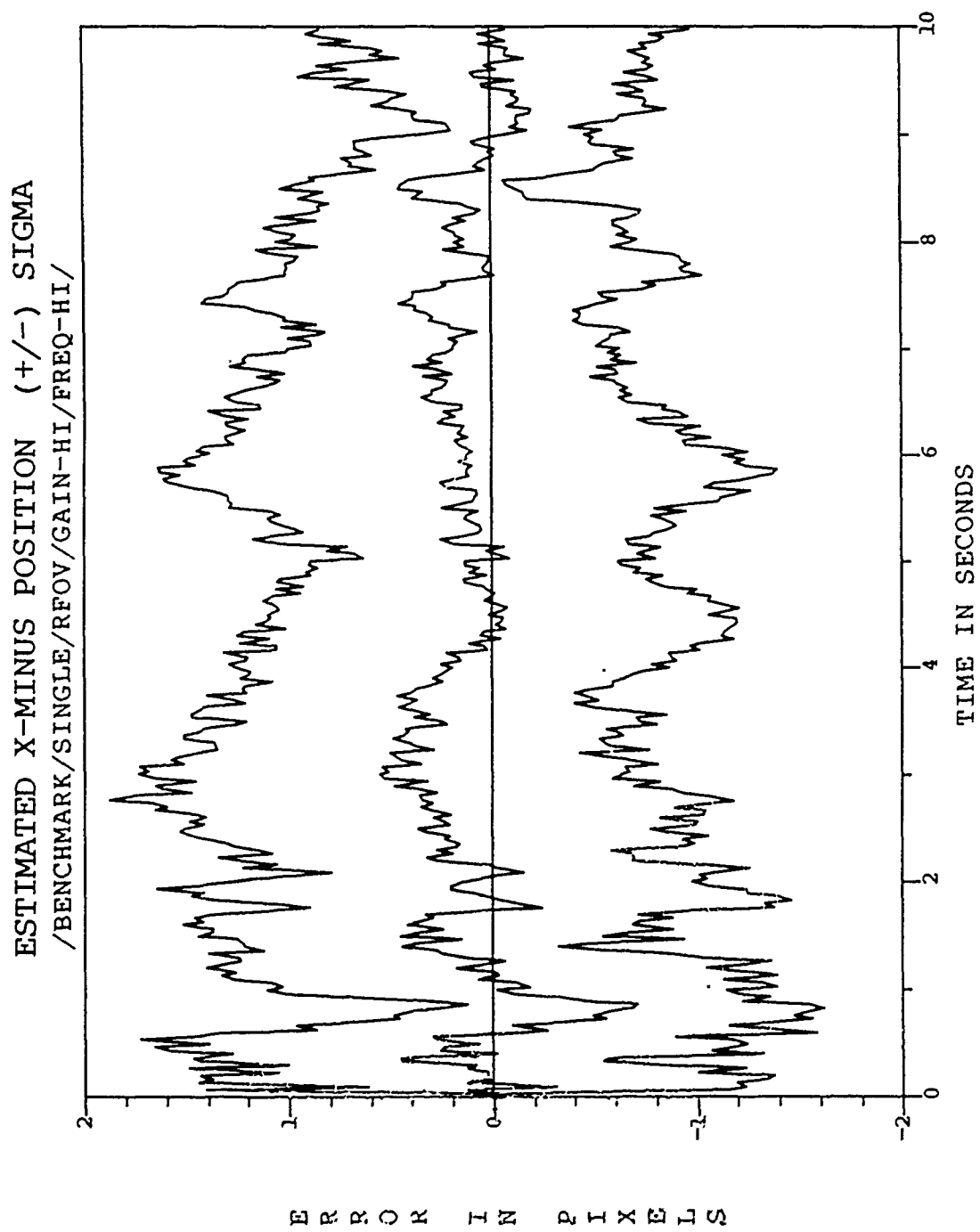


Figure D.23. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

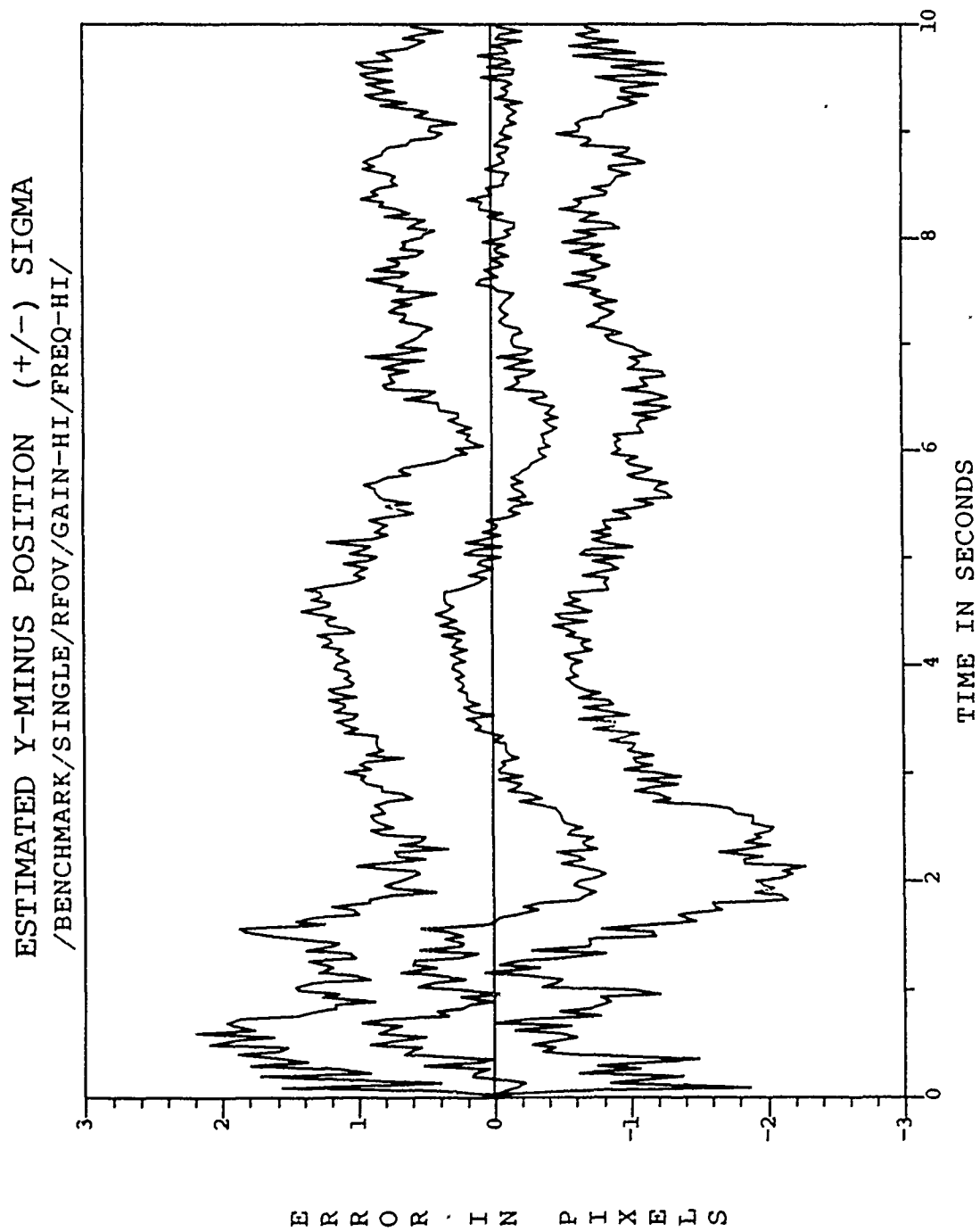


Figure D.21. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

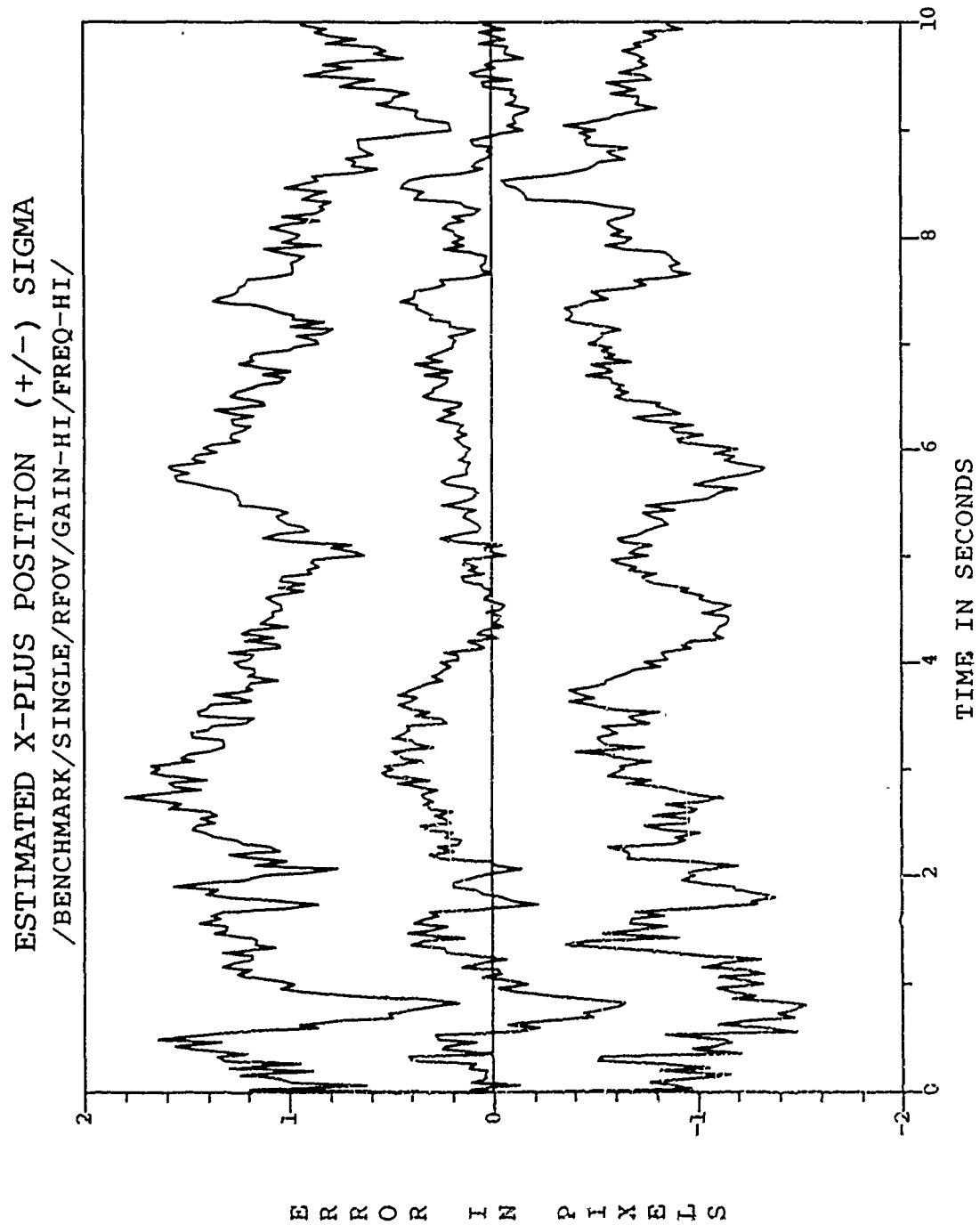


Figure D.25. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

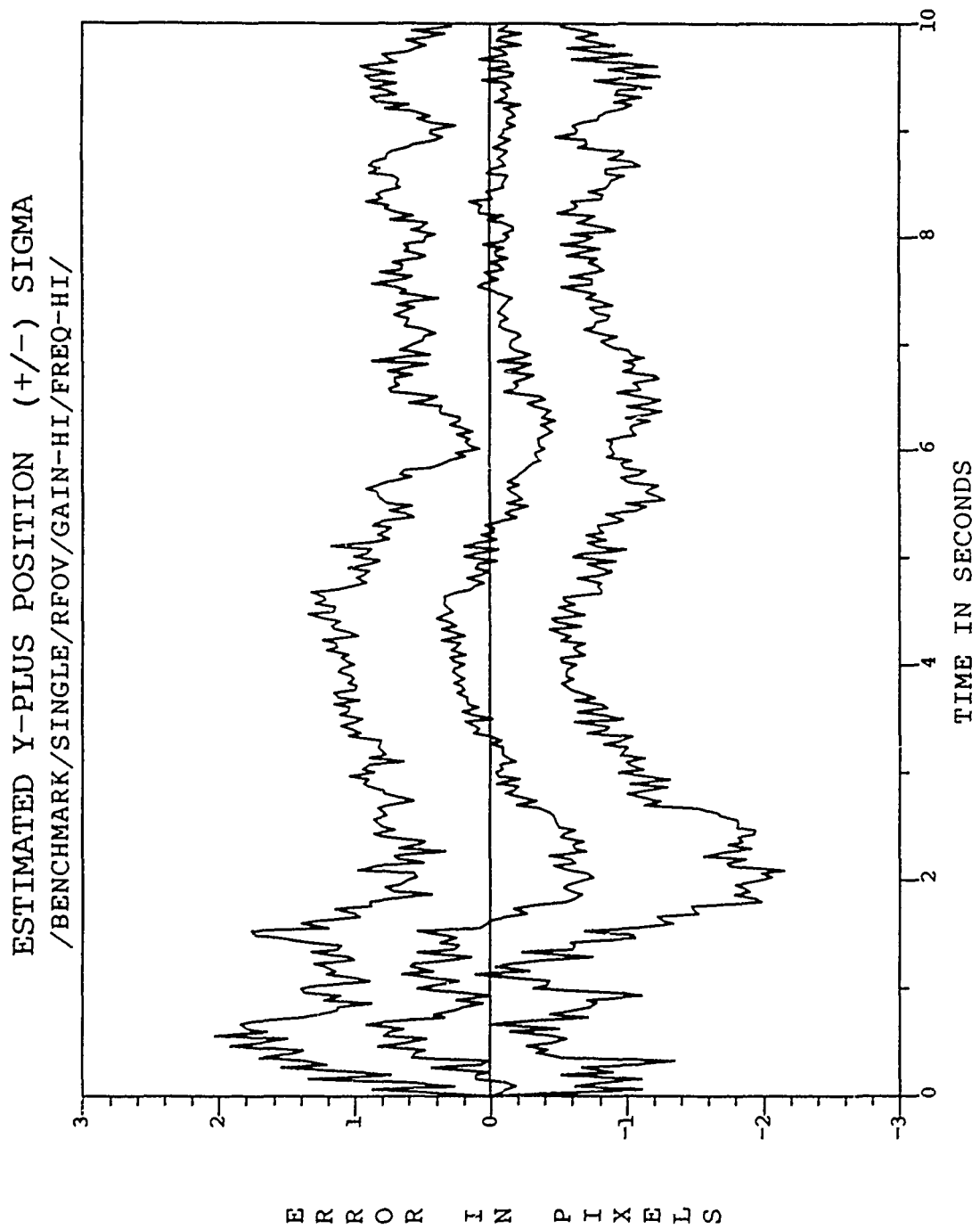


Figure D.26. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-HI/

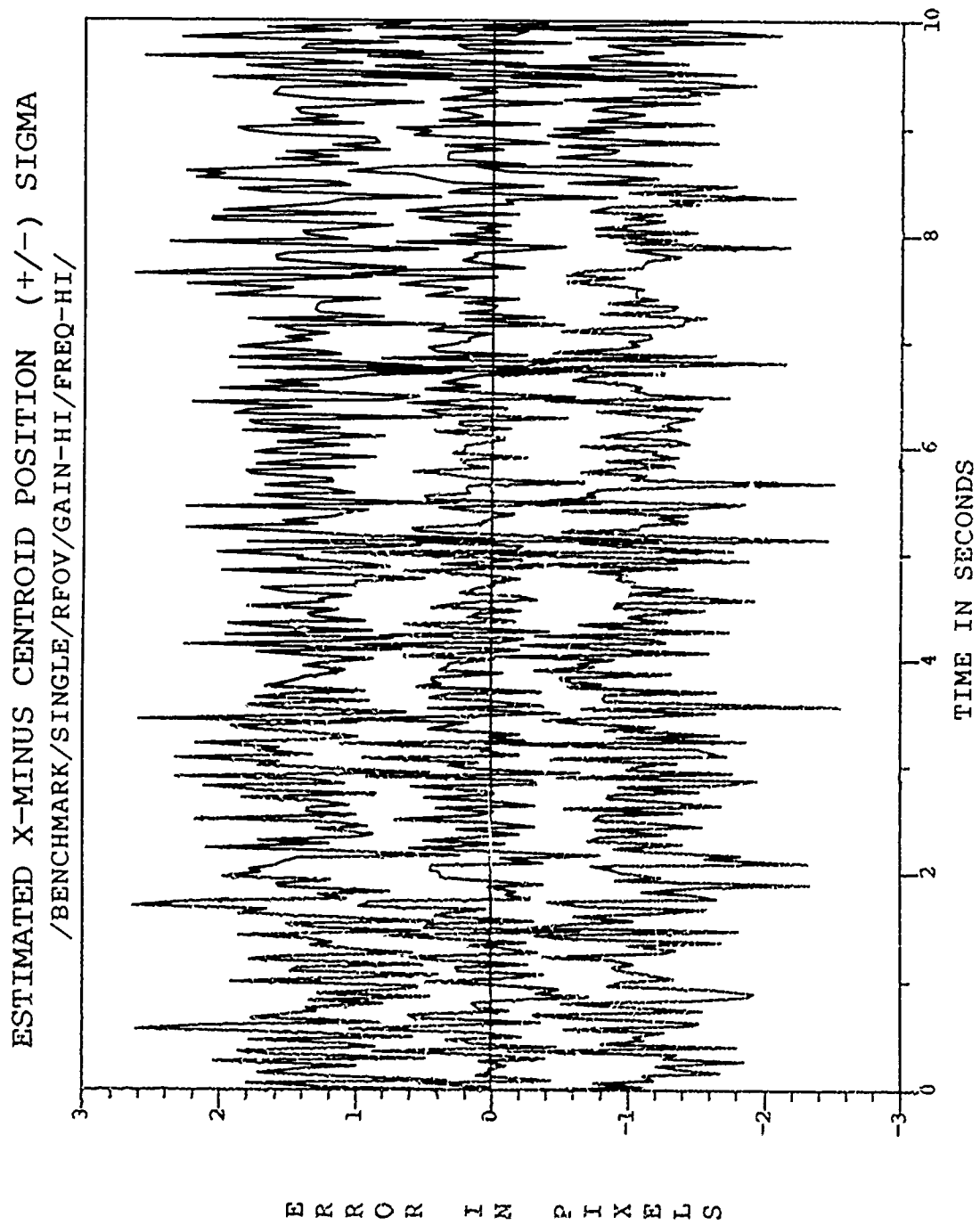


Figure D.27. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

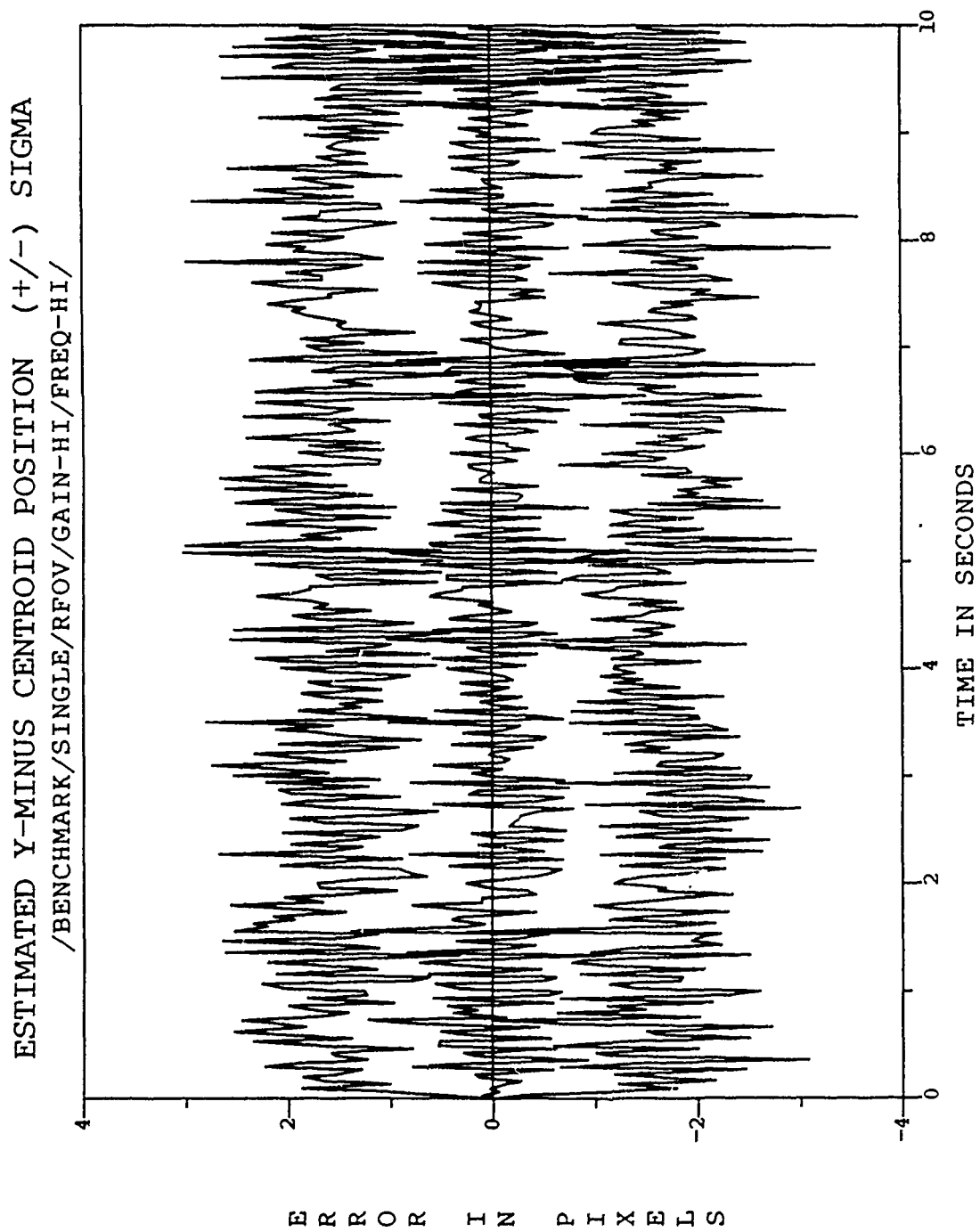


Figure D.28. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

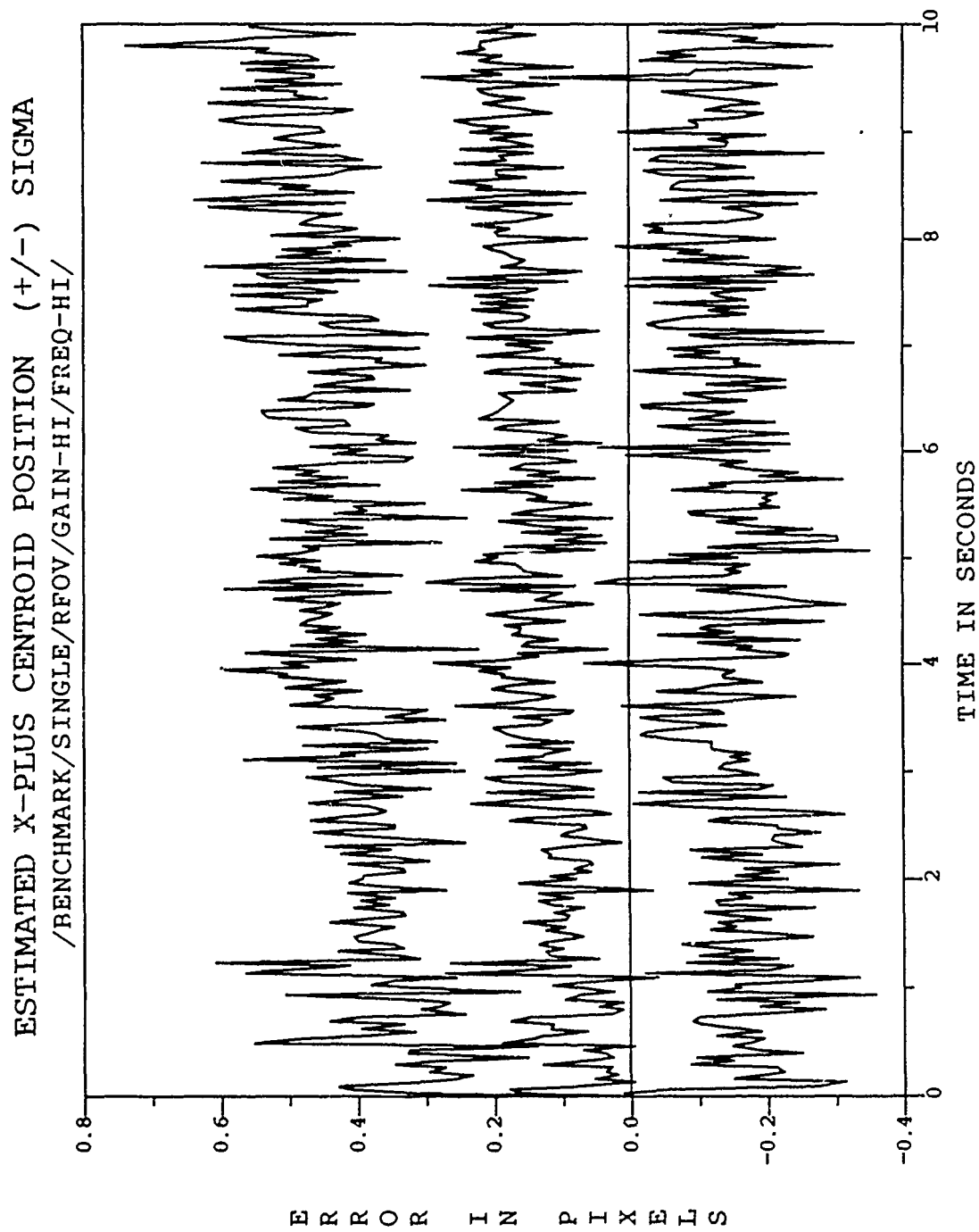


Figure D.29. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

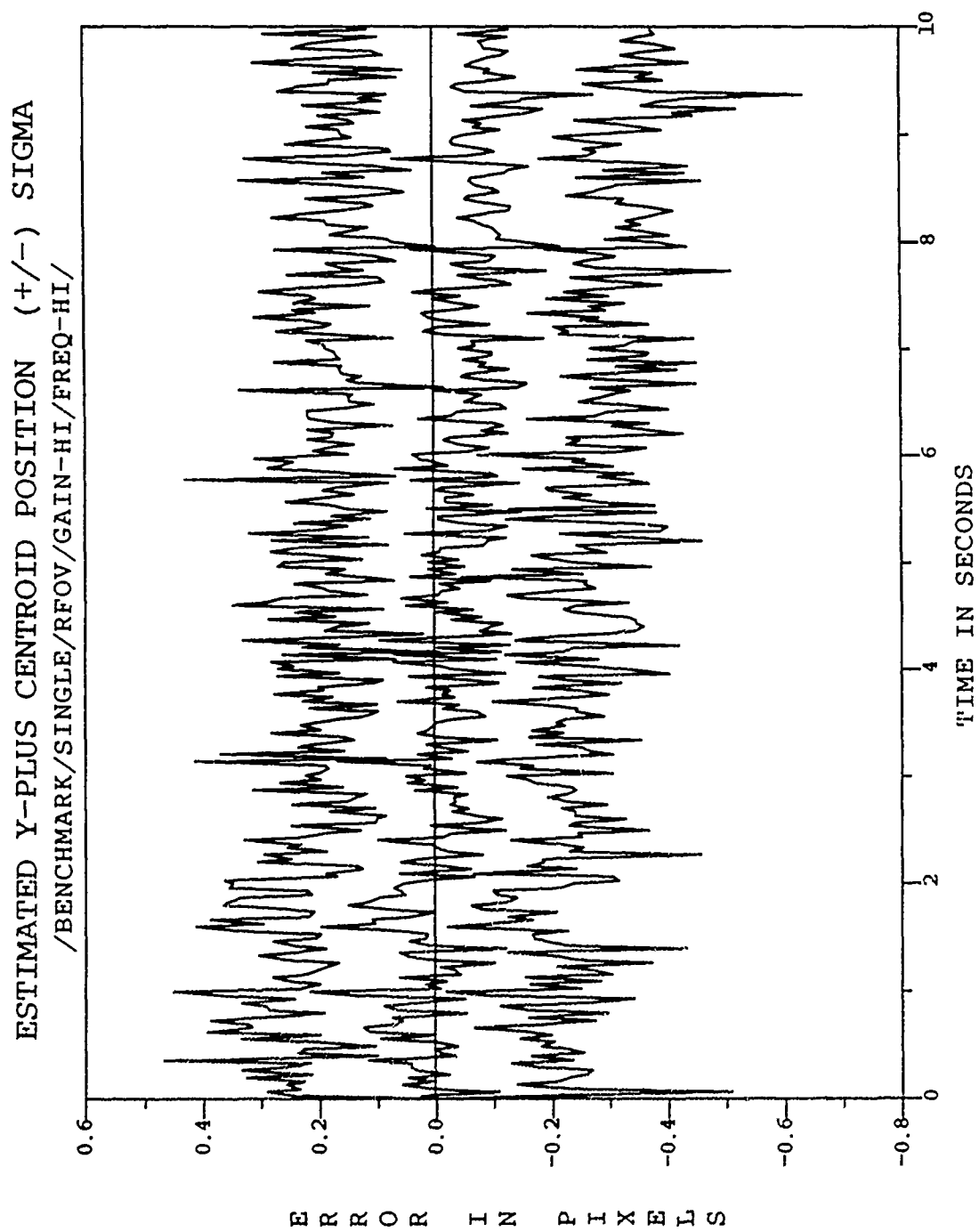


Figure D.30. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

Appendix E. *10-State Filter Pogo Performance Plots: Discussion*
in Section 6.5

E.1 Figures E.1-E.10: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.20$

E.2 Figures E.11-E.20: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.15$

E.3 Figures E.21-E.30: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 1.70$

E.4 Figures E.31-E.40: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = 2.05$

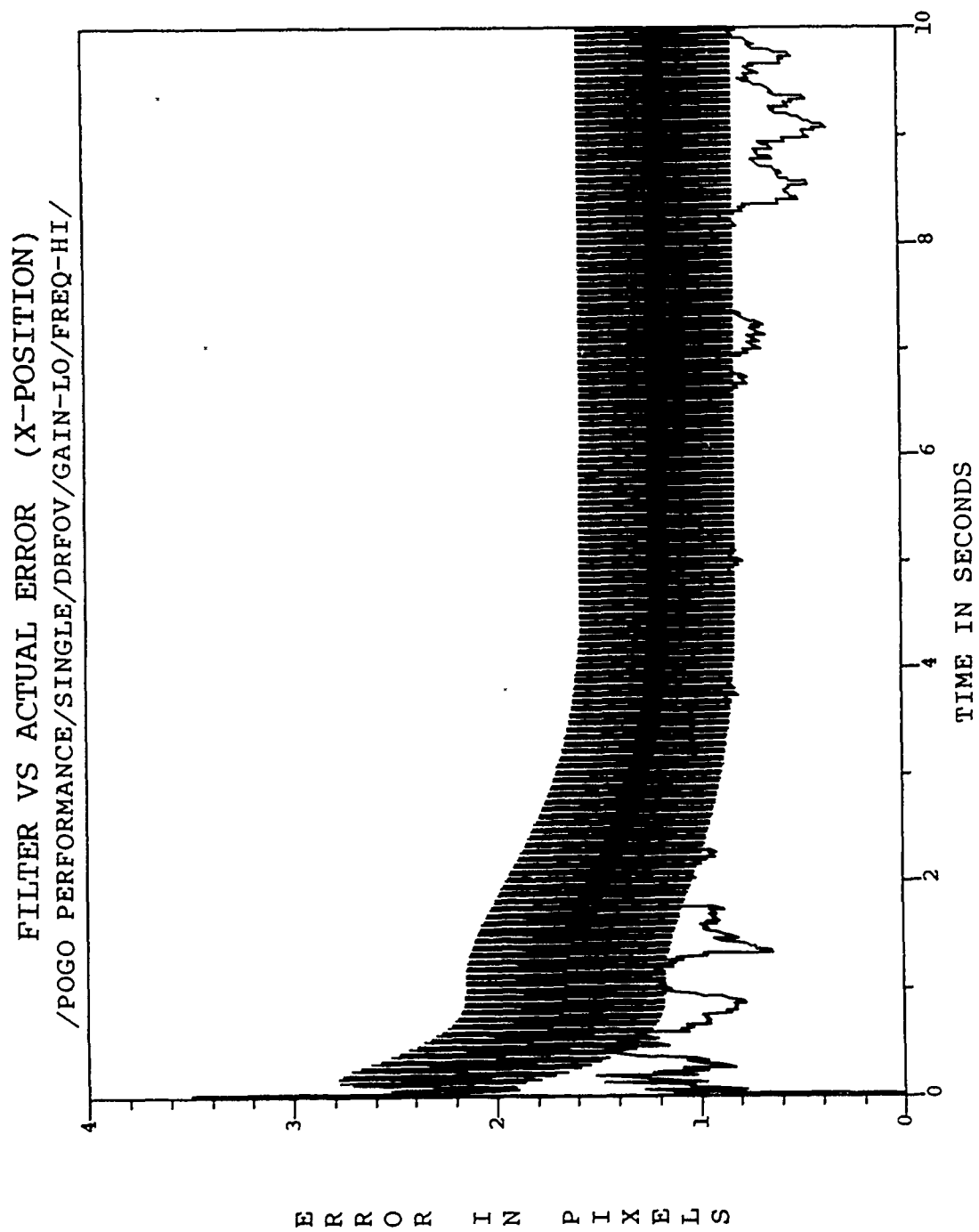


Figure E.1. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

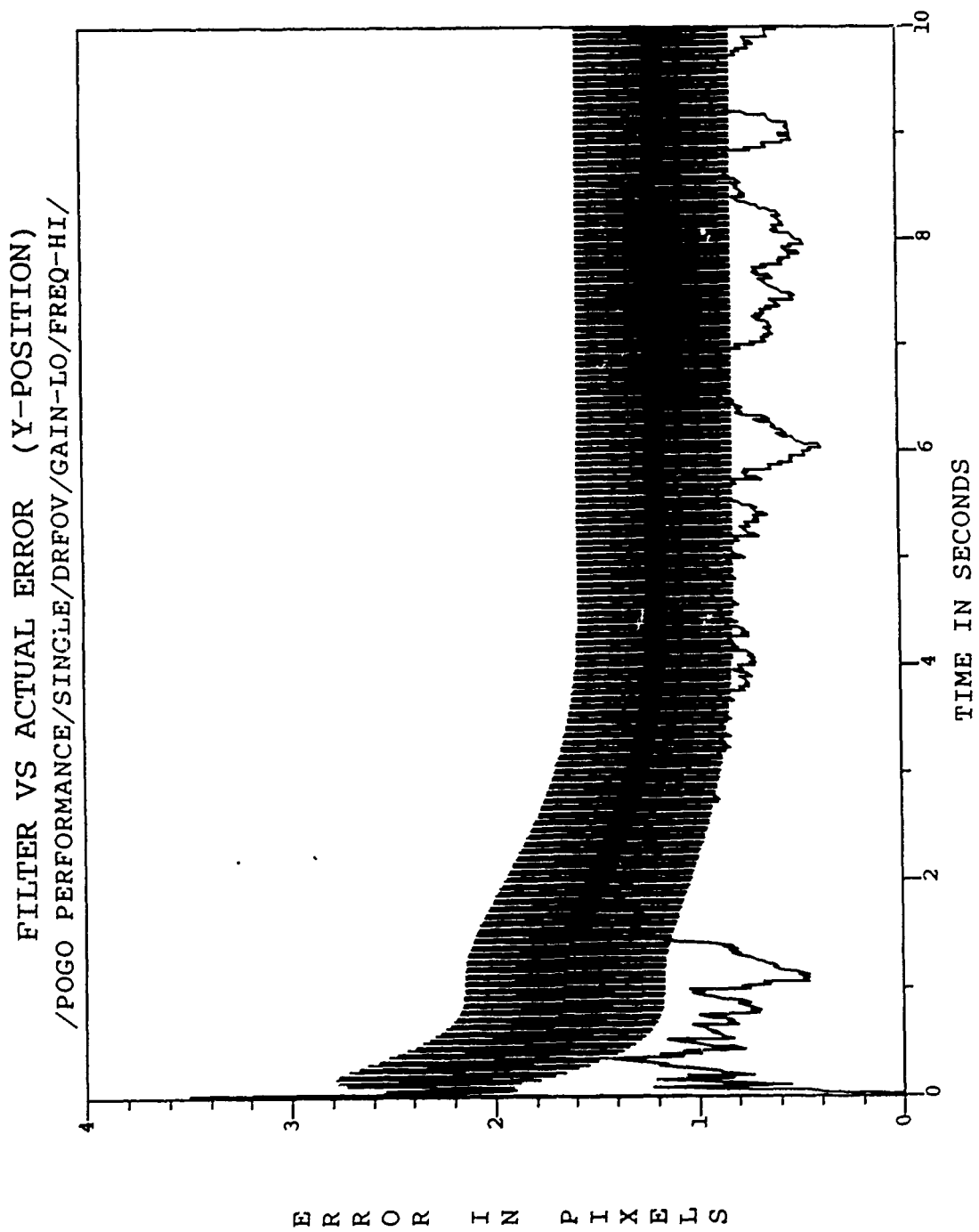


Figure E.2. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

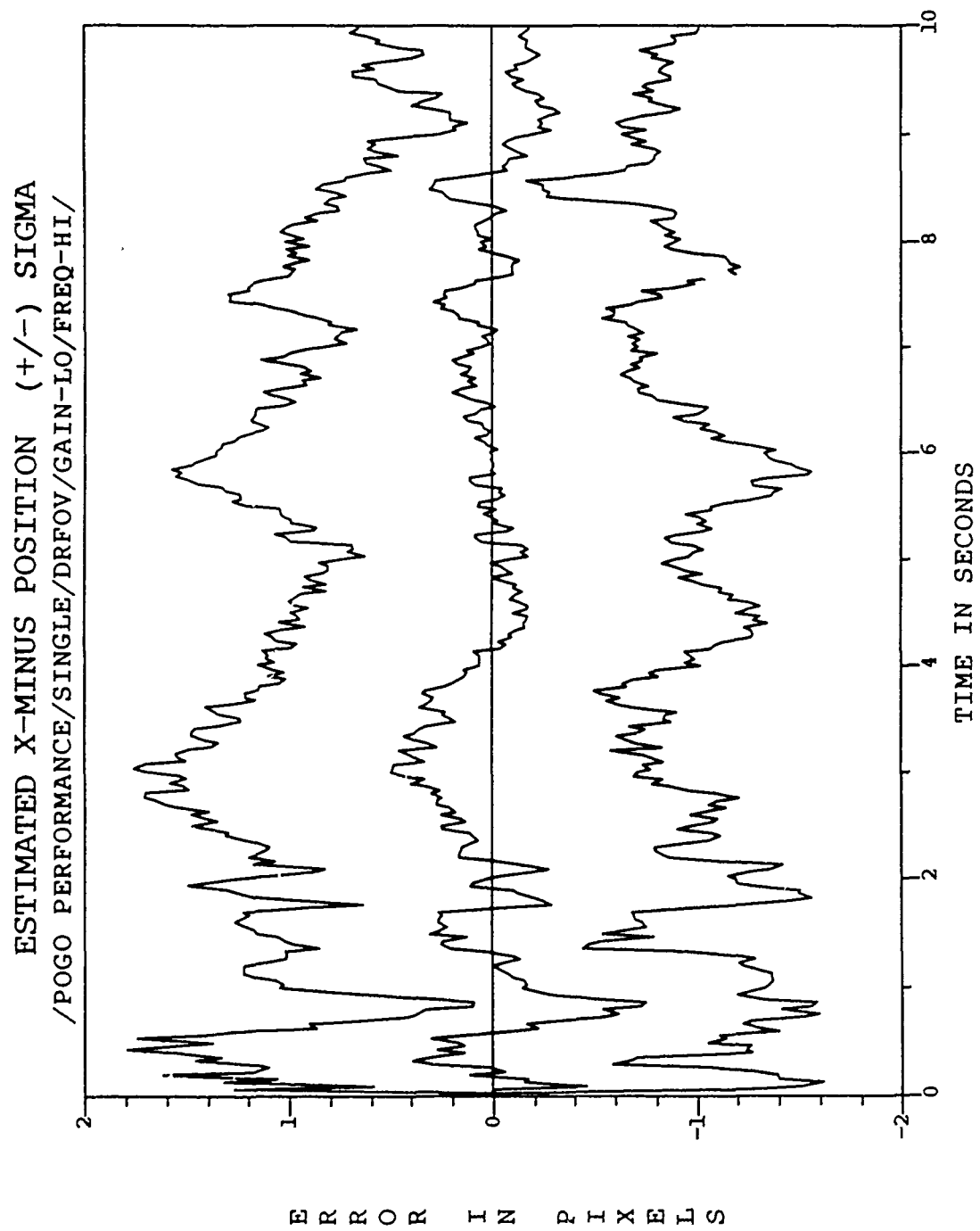


Figure E.3. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-III/

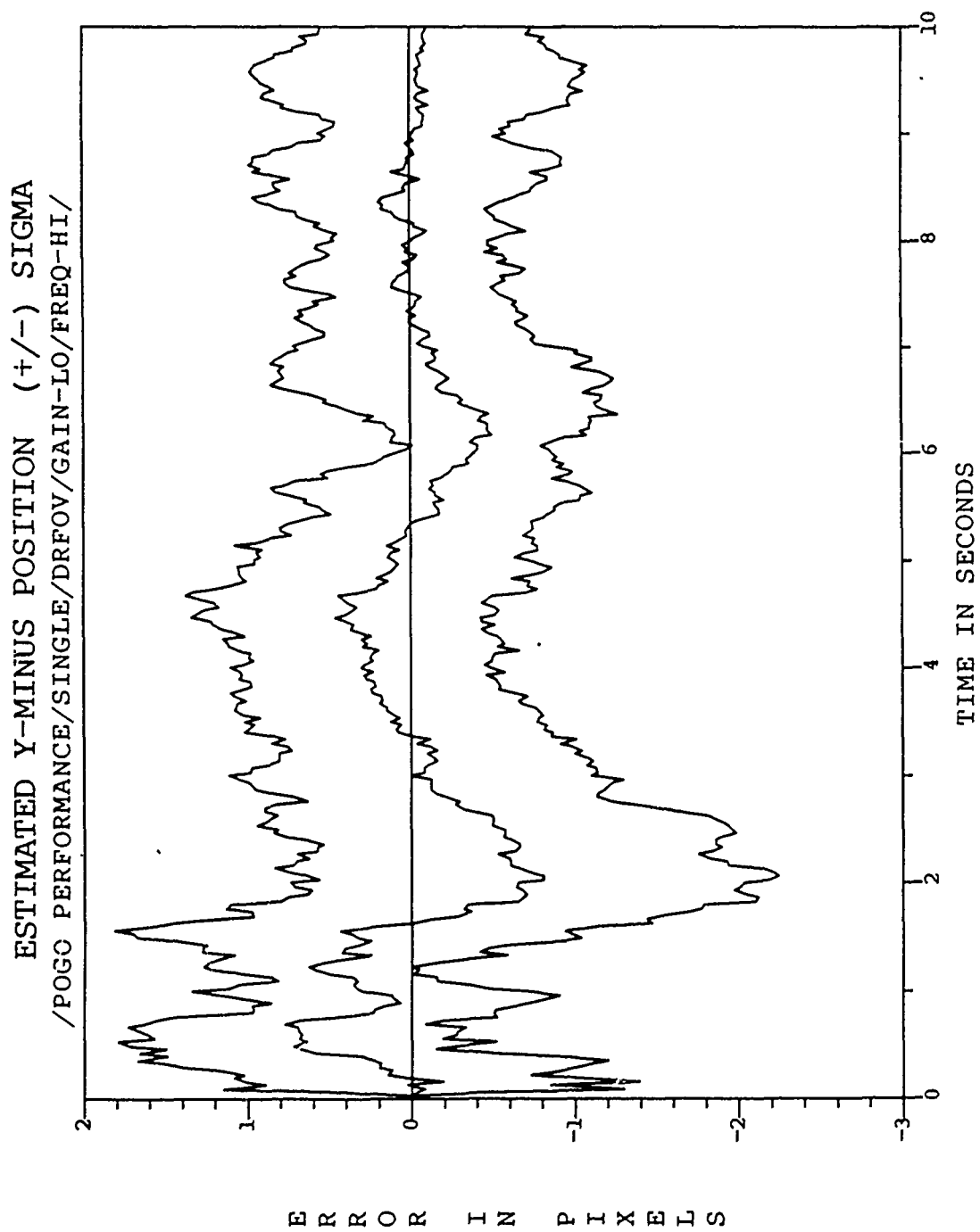


Figure E.4. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

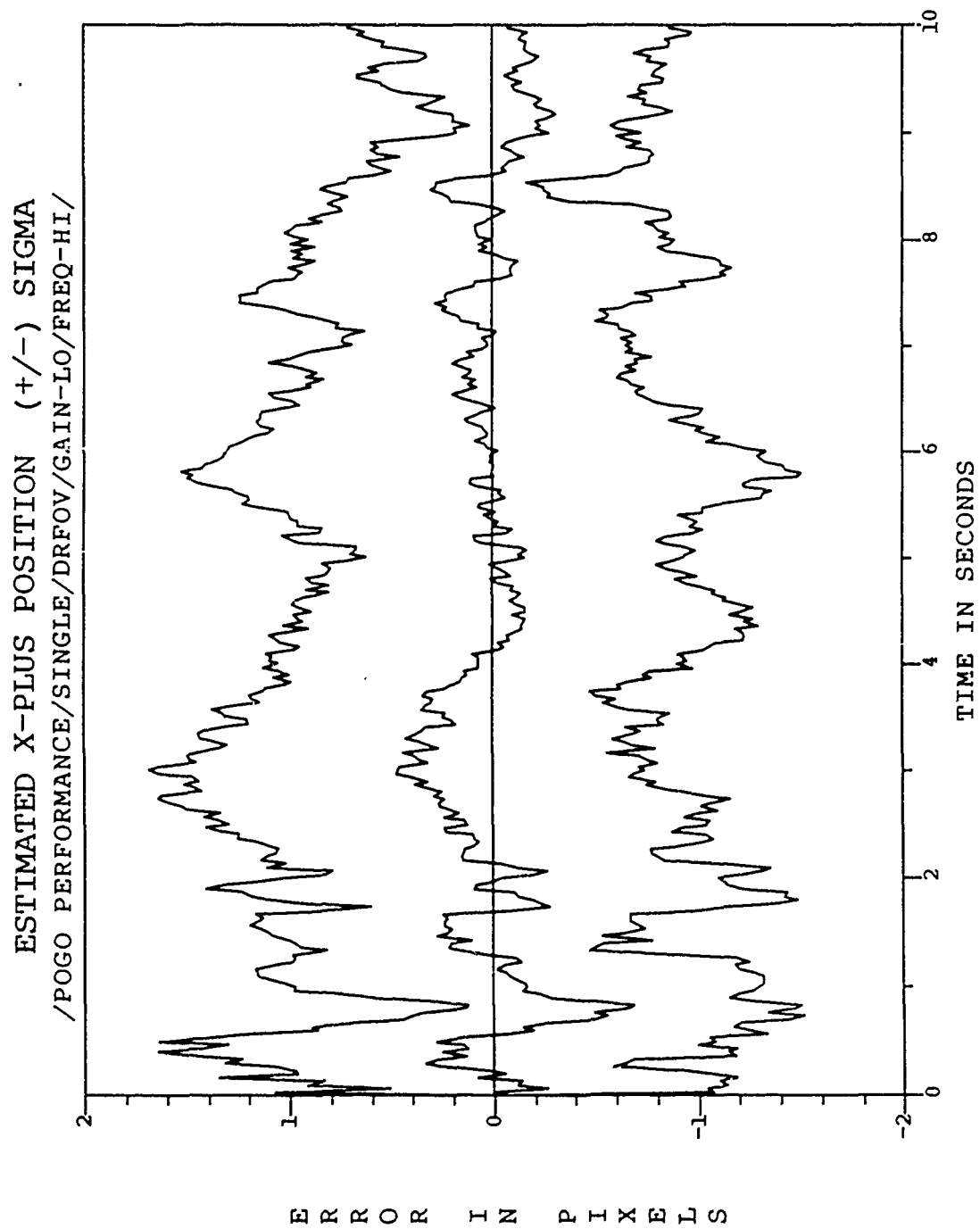


Figure E.5. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-III/

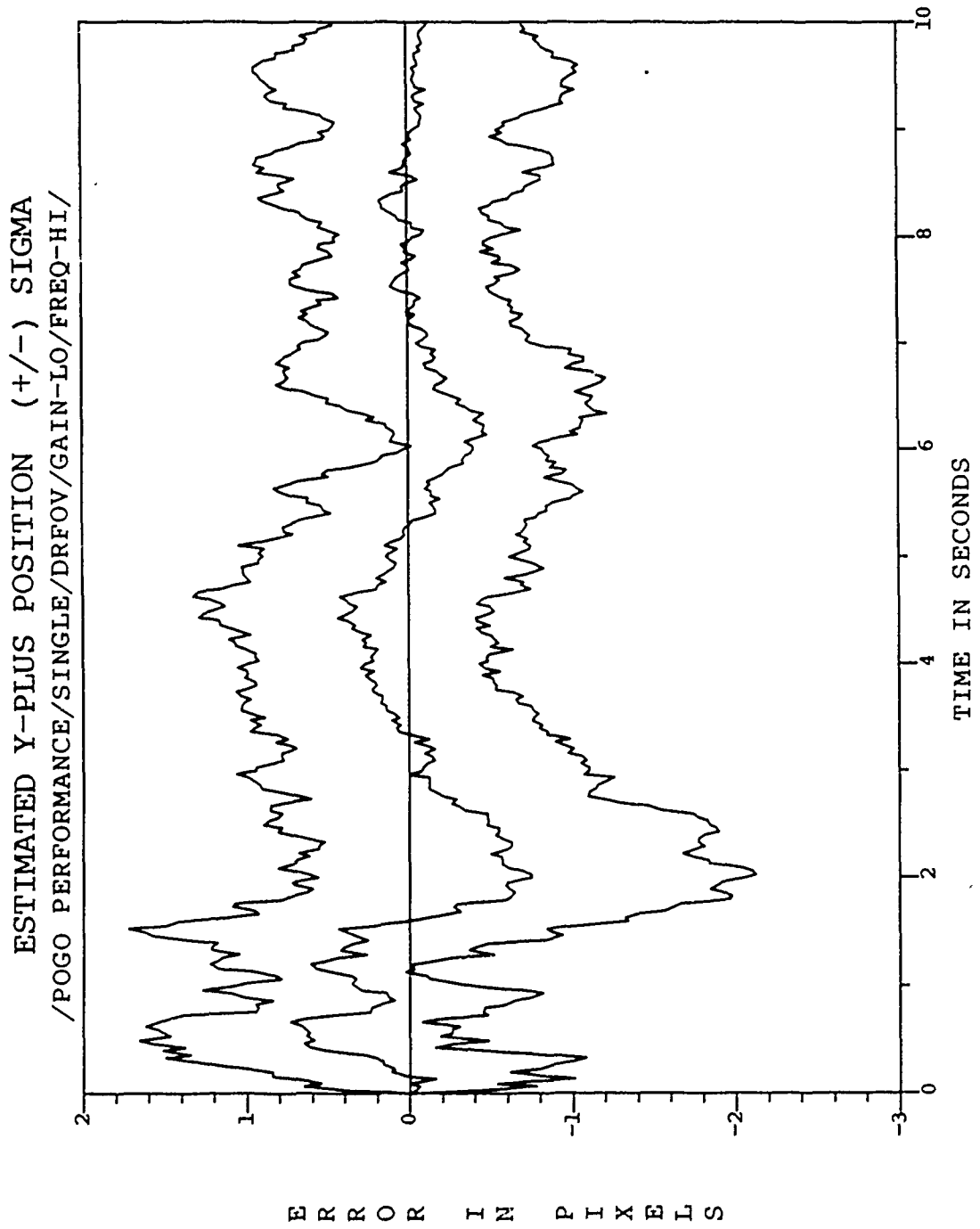


Figure E.6. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

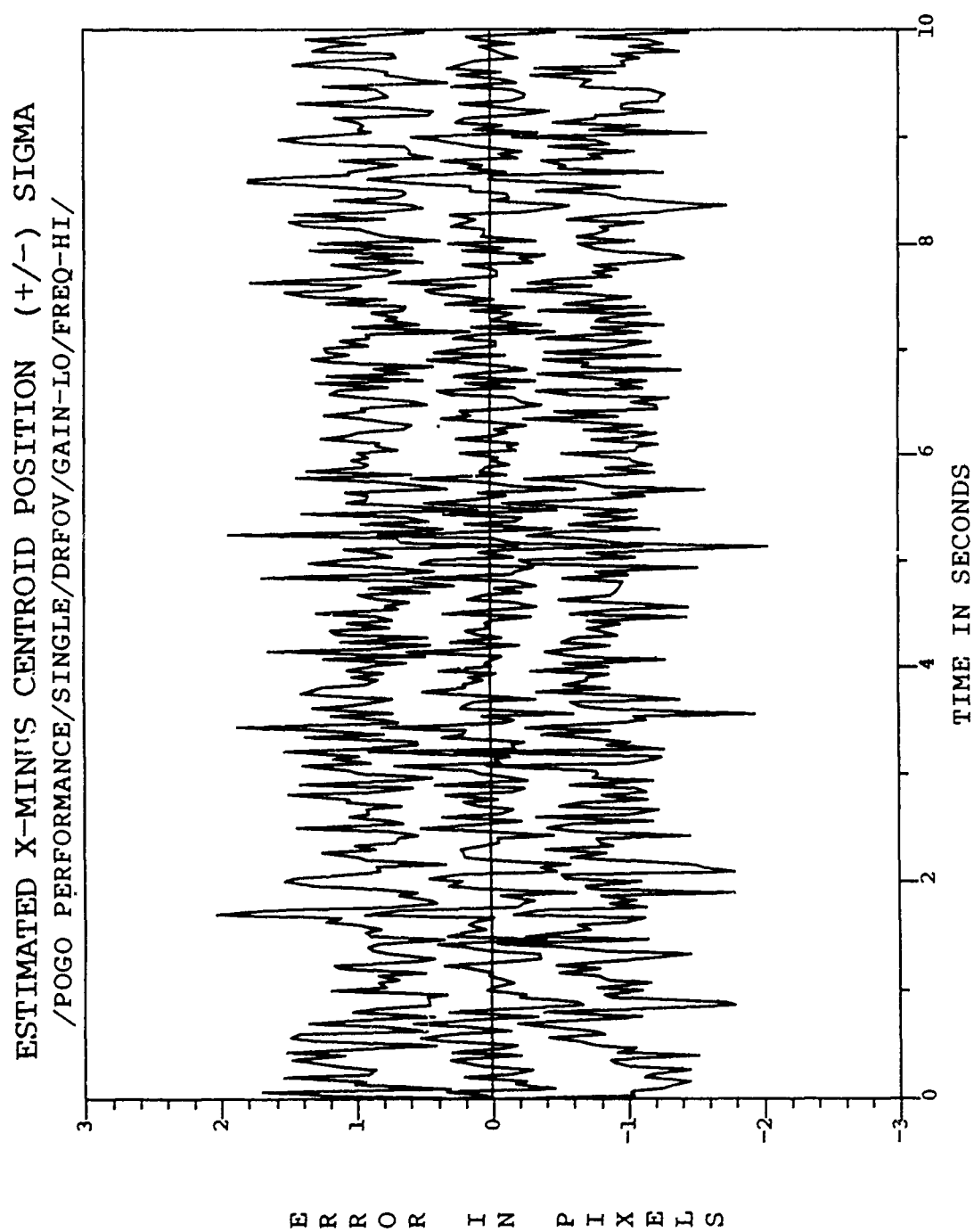


Figure E.7. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-HI/

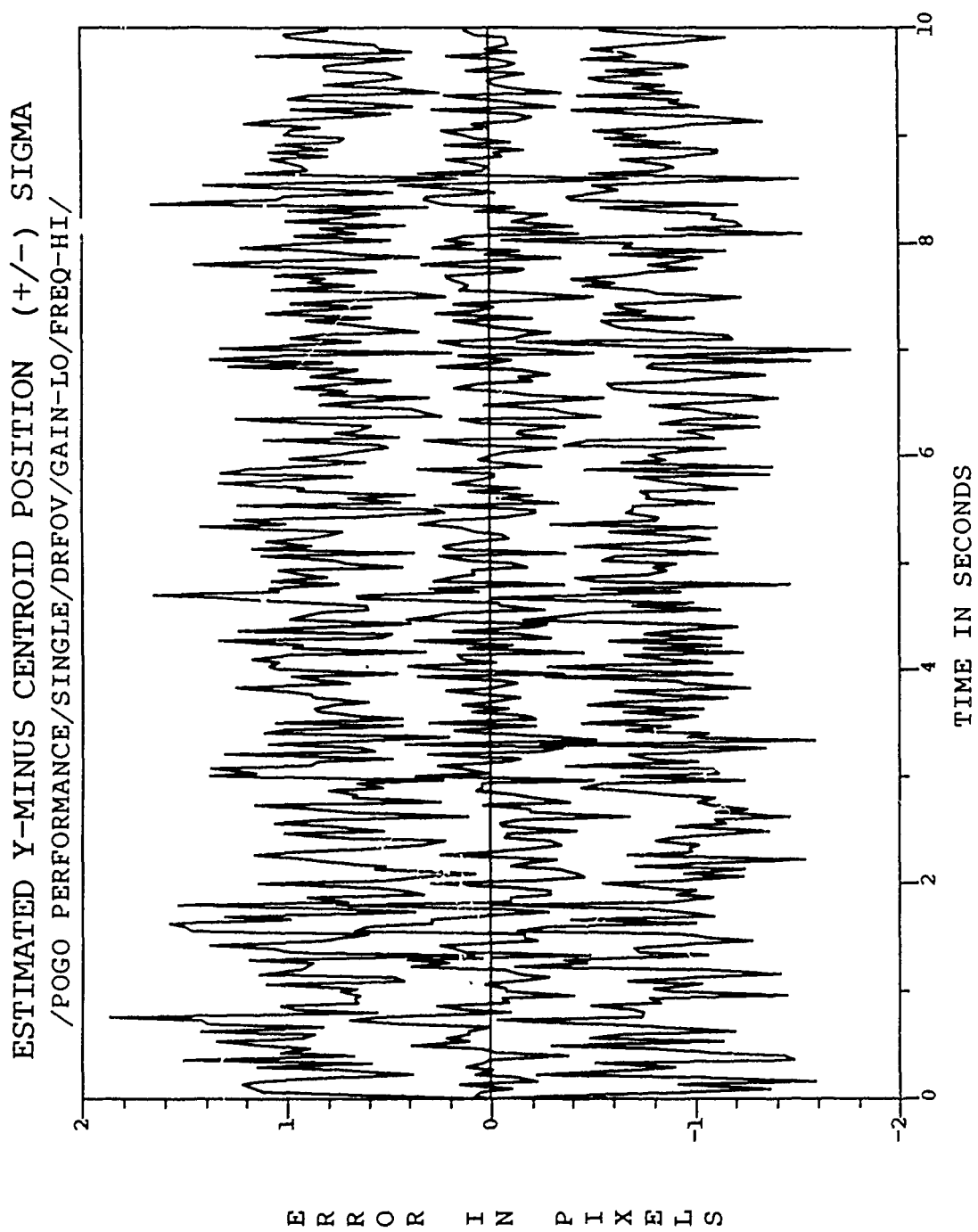


Figure E.8. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

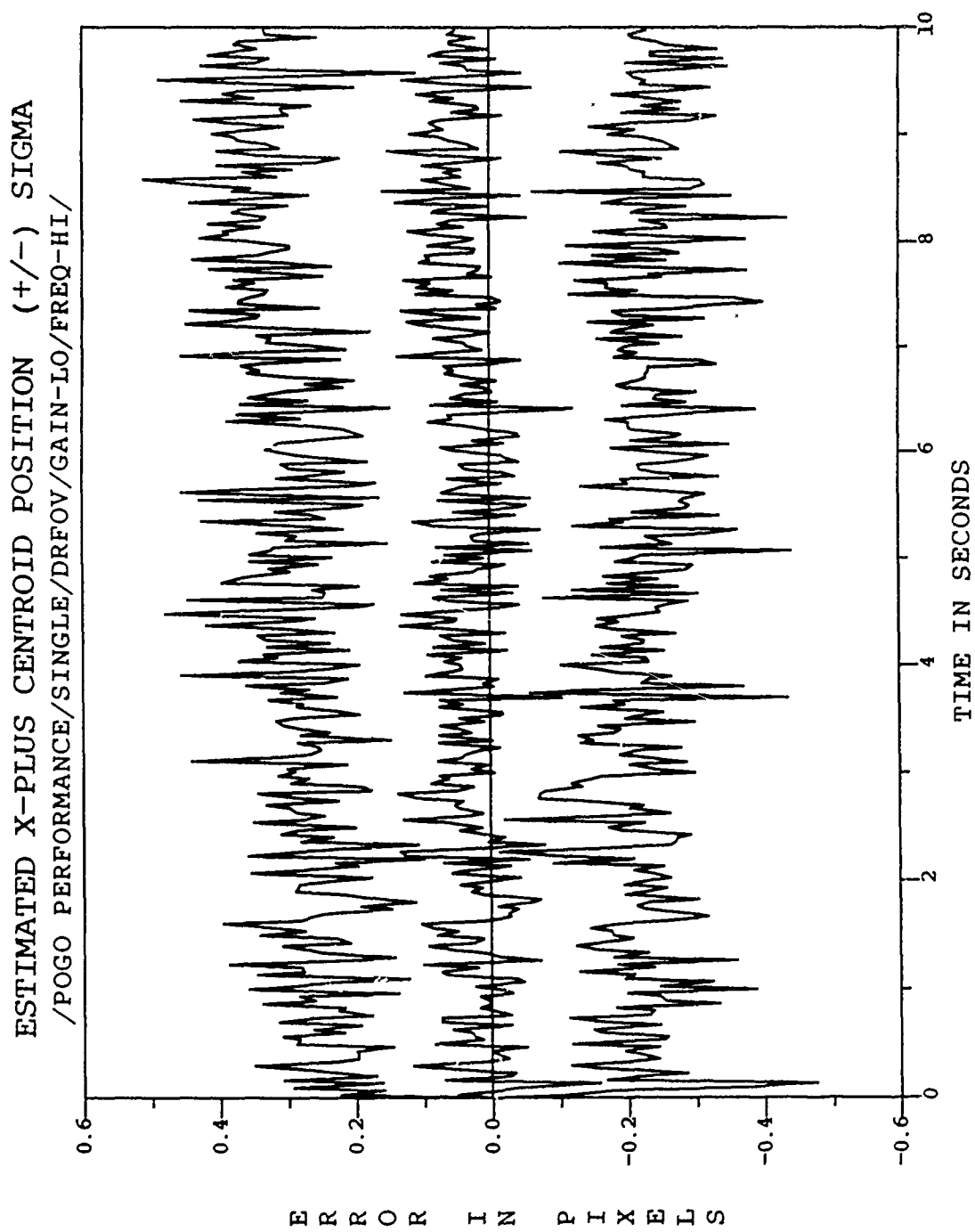


Figure E.9. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-III/

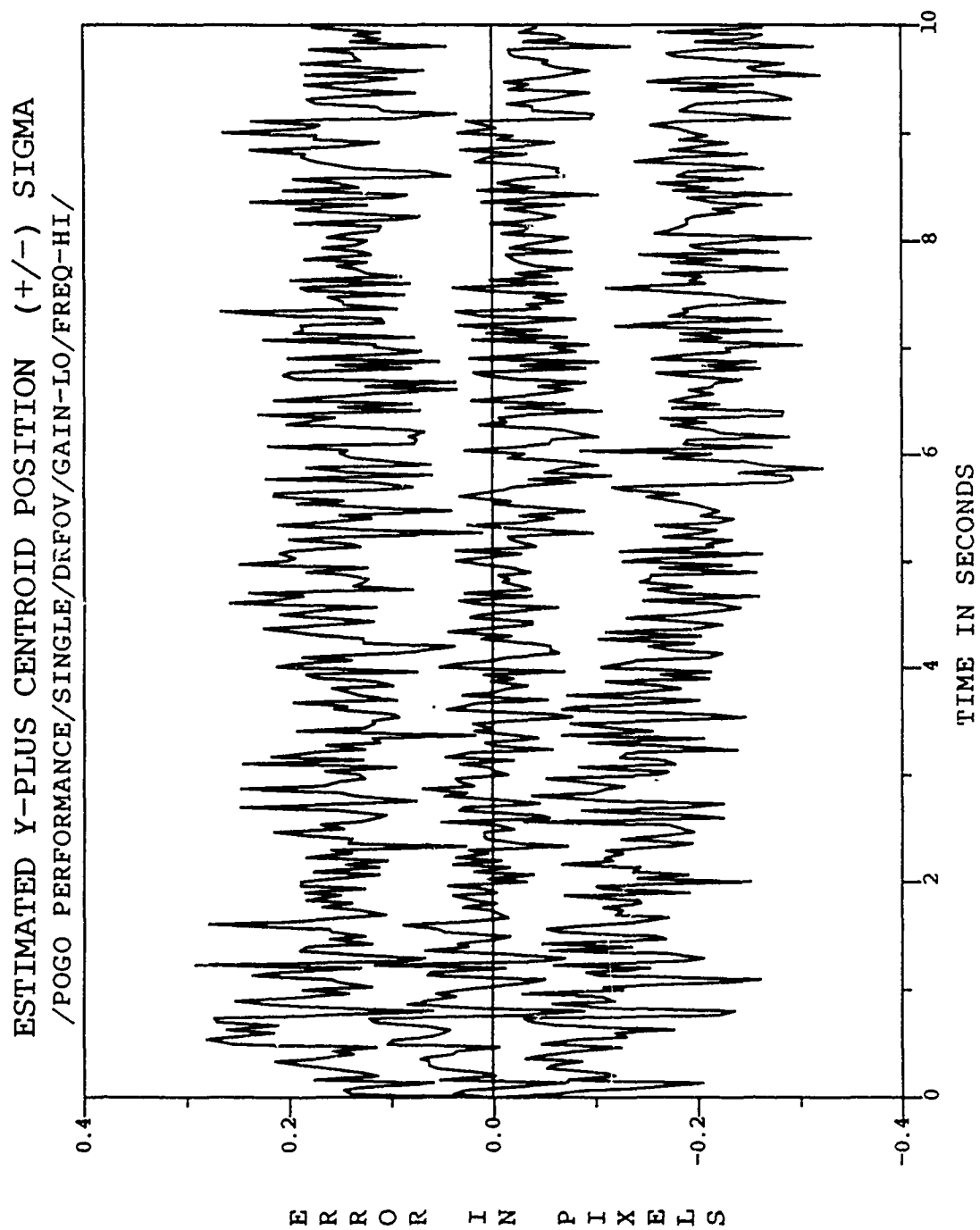


Figure E.10. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

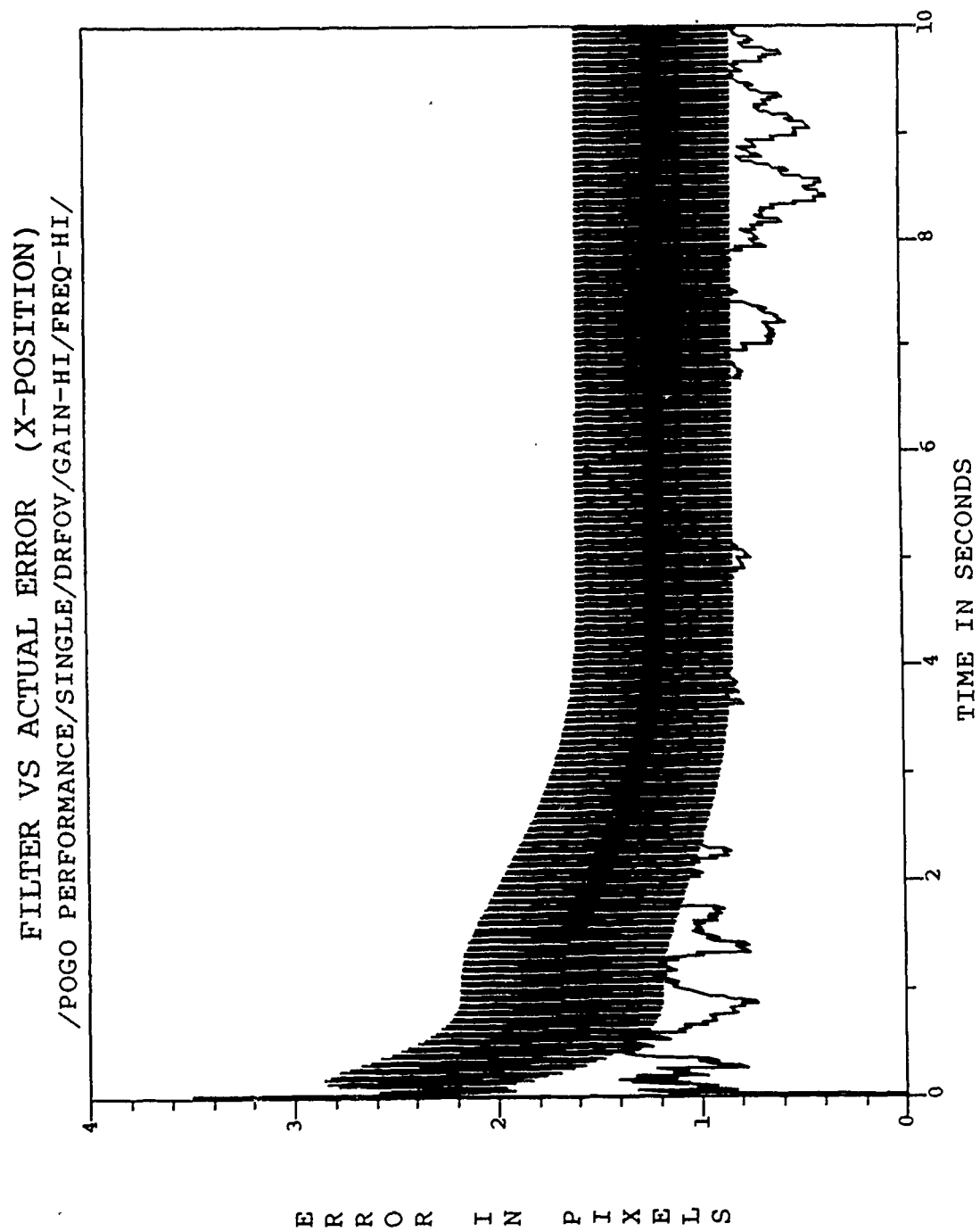


Figure E.11. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

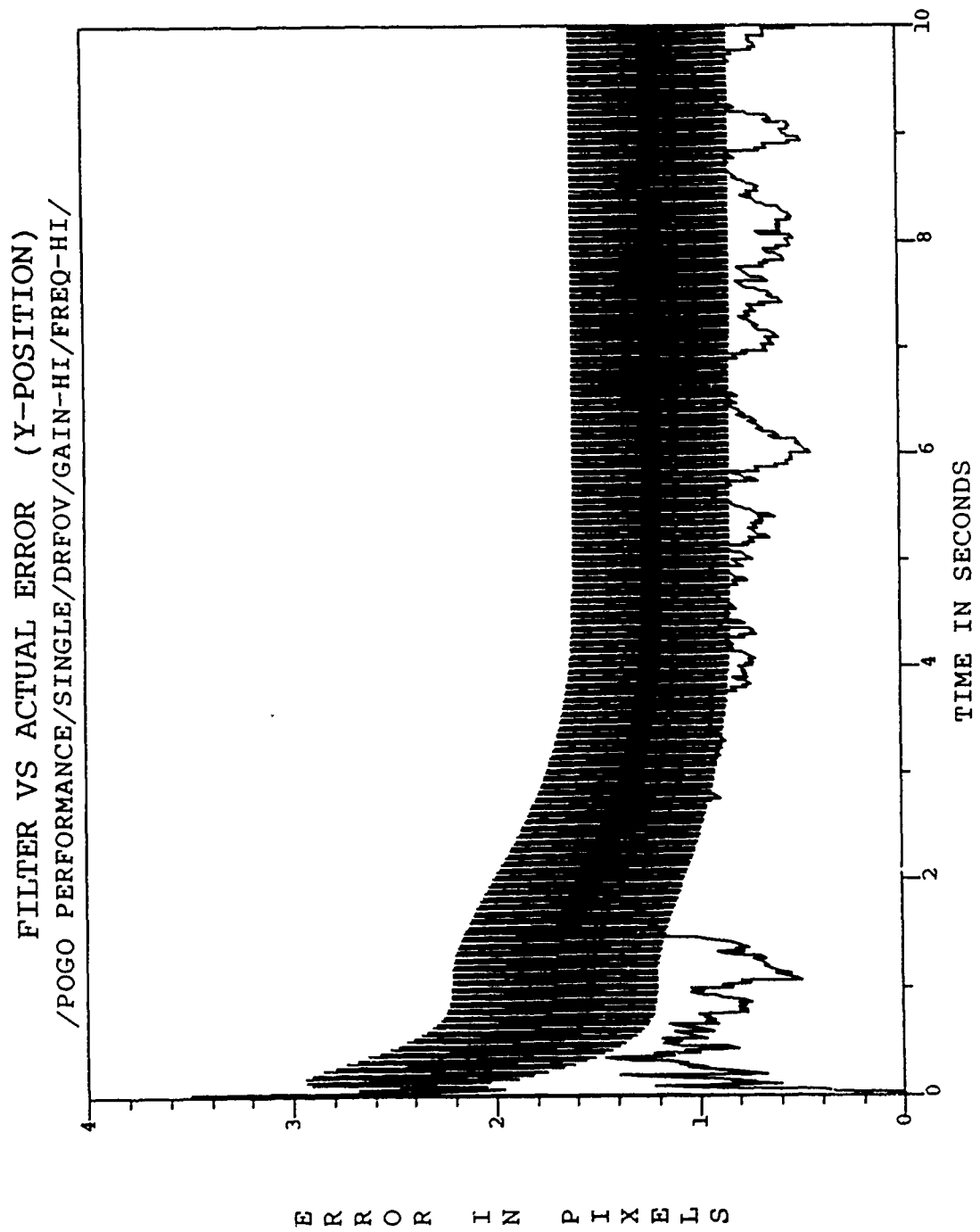


Figure E.12. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

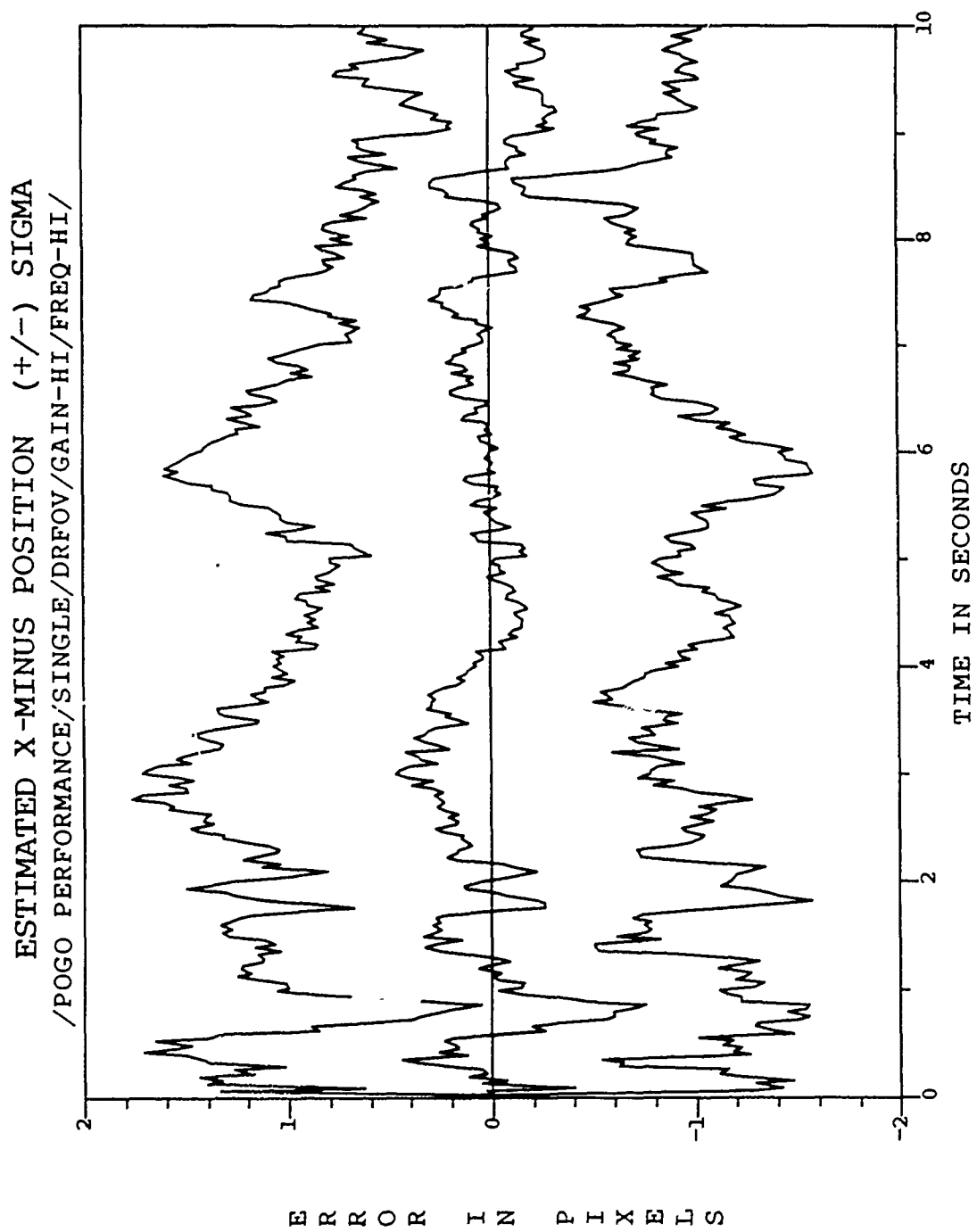


Figure E.13. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

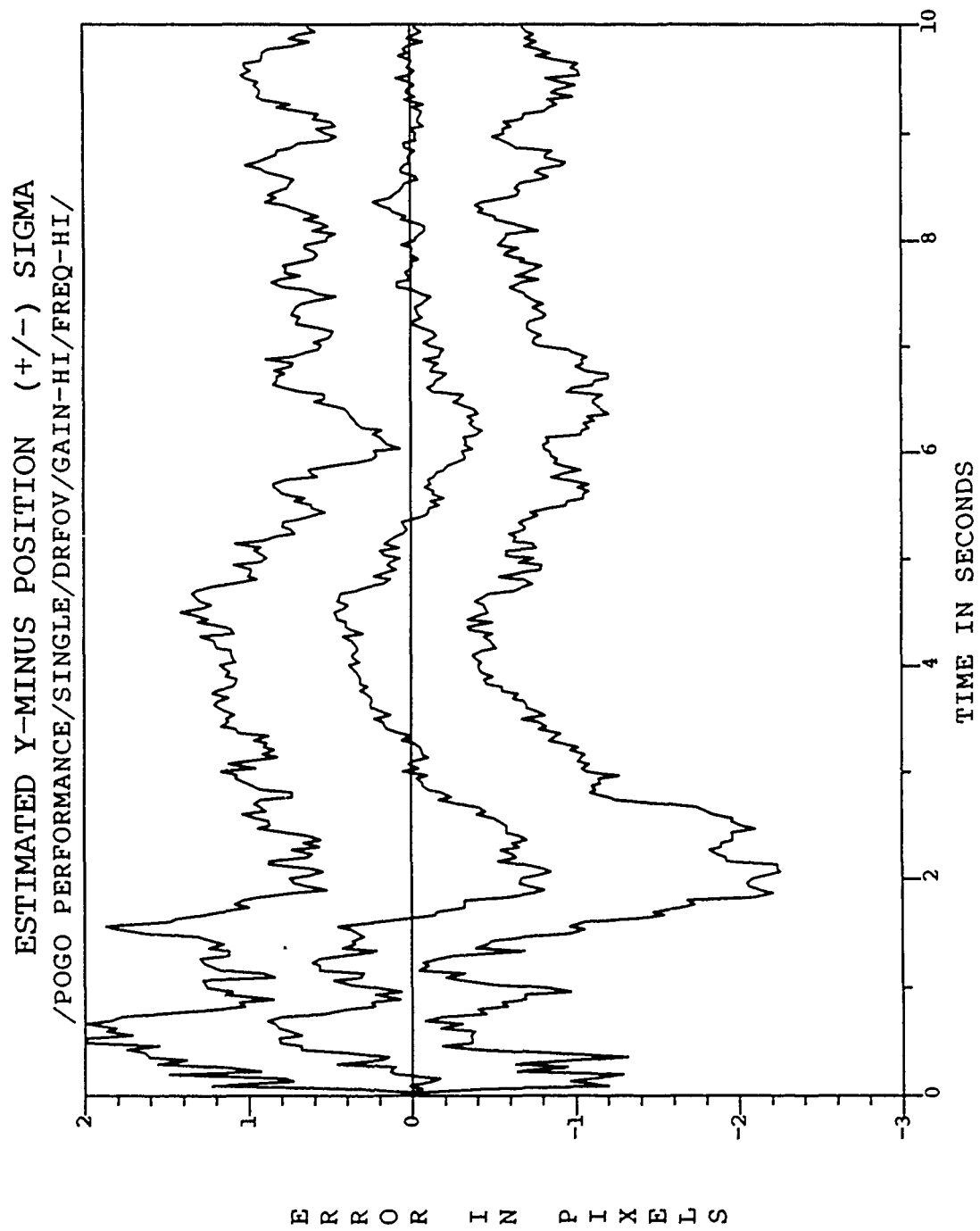


Figure E.14. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

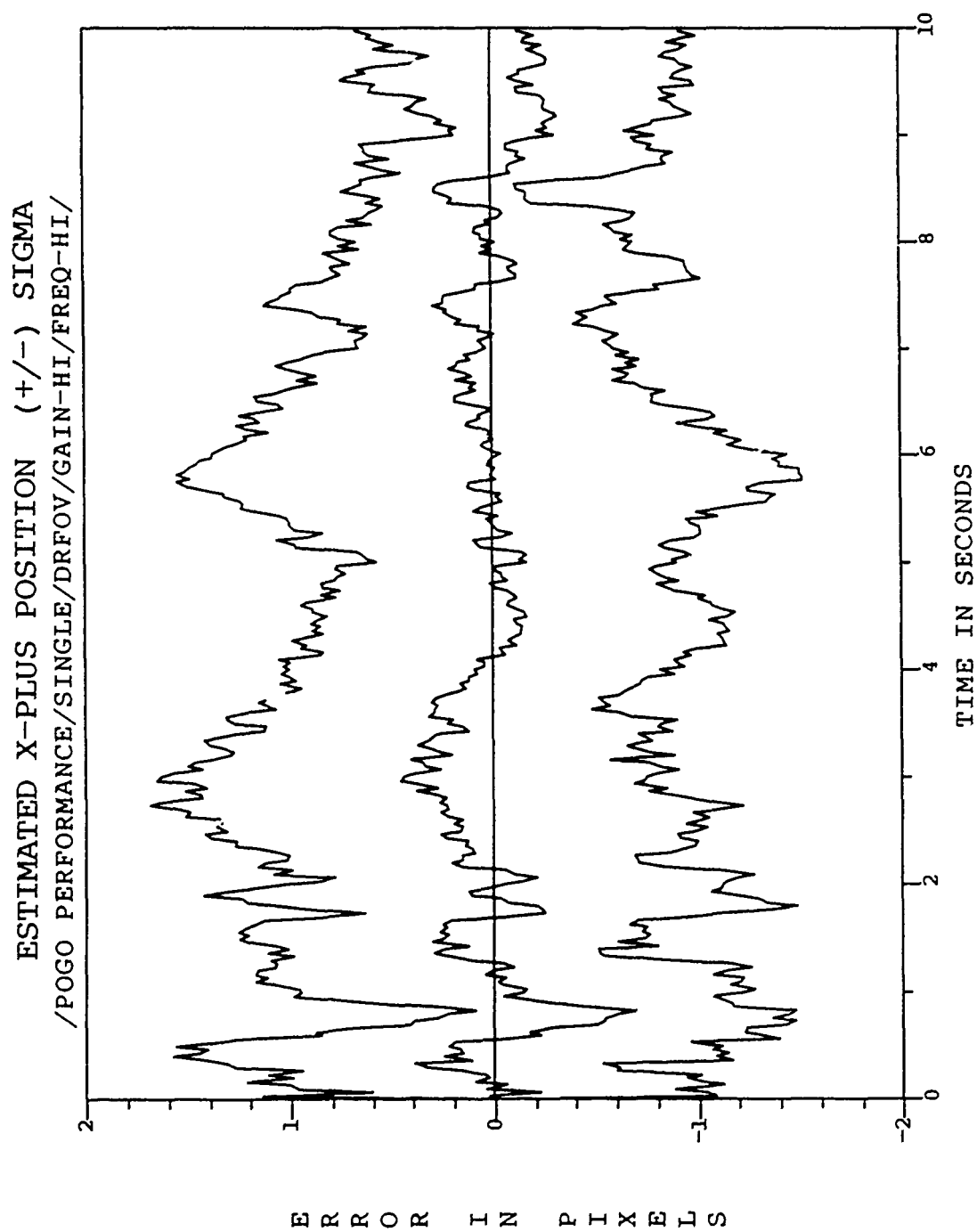


Figure E.15. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

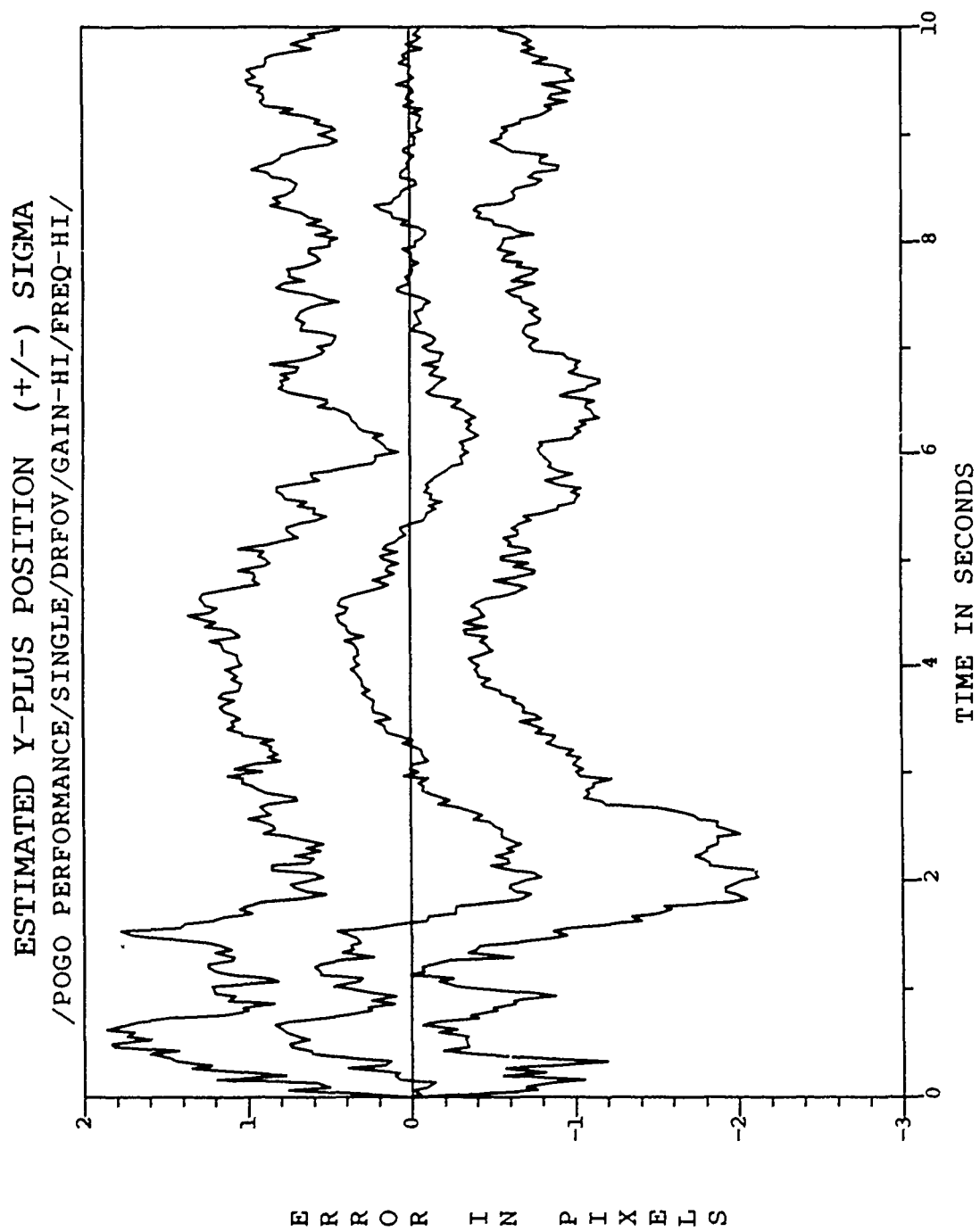


Figure E.16. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

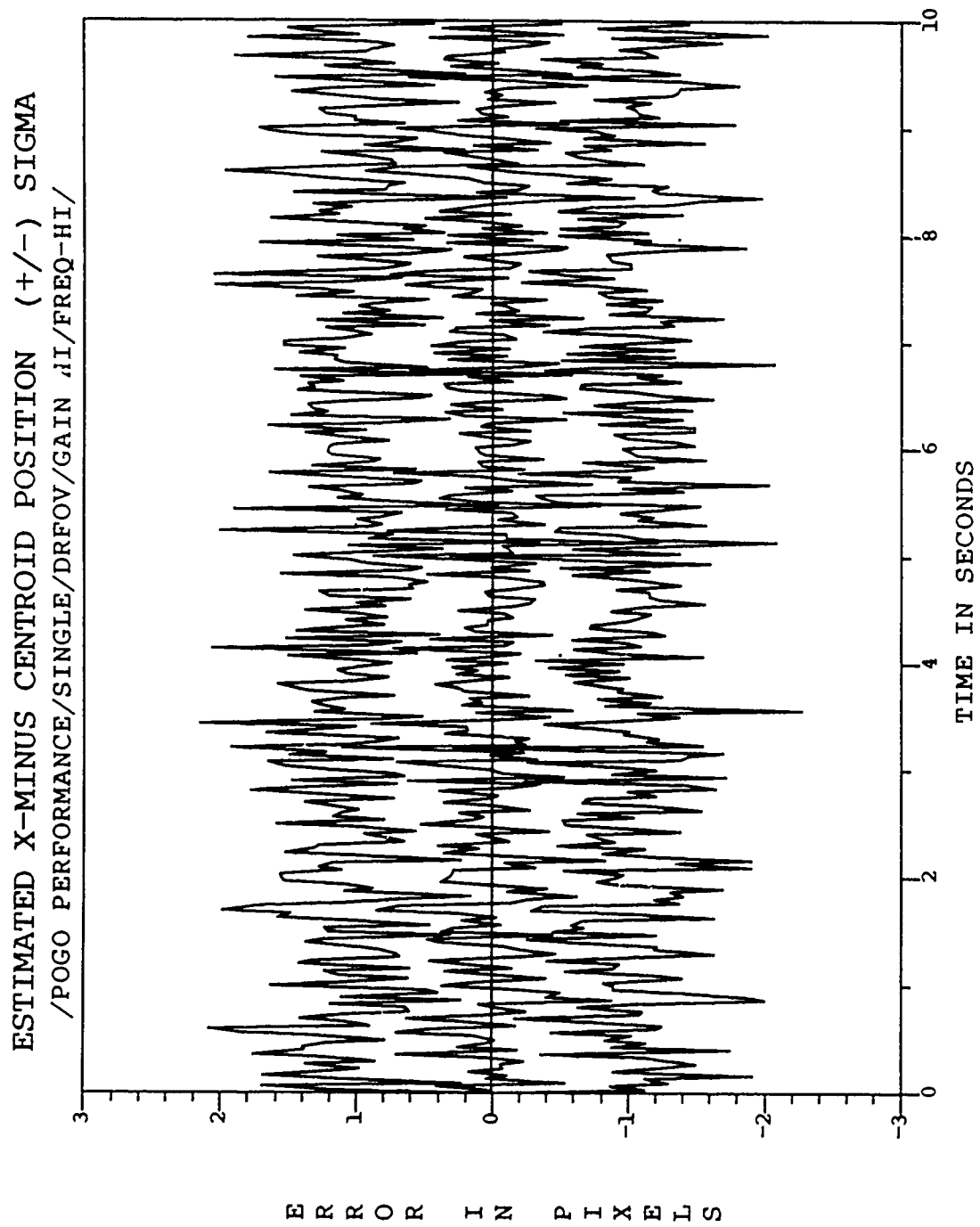


Figure E.17. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

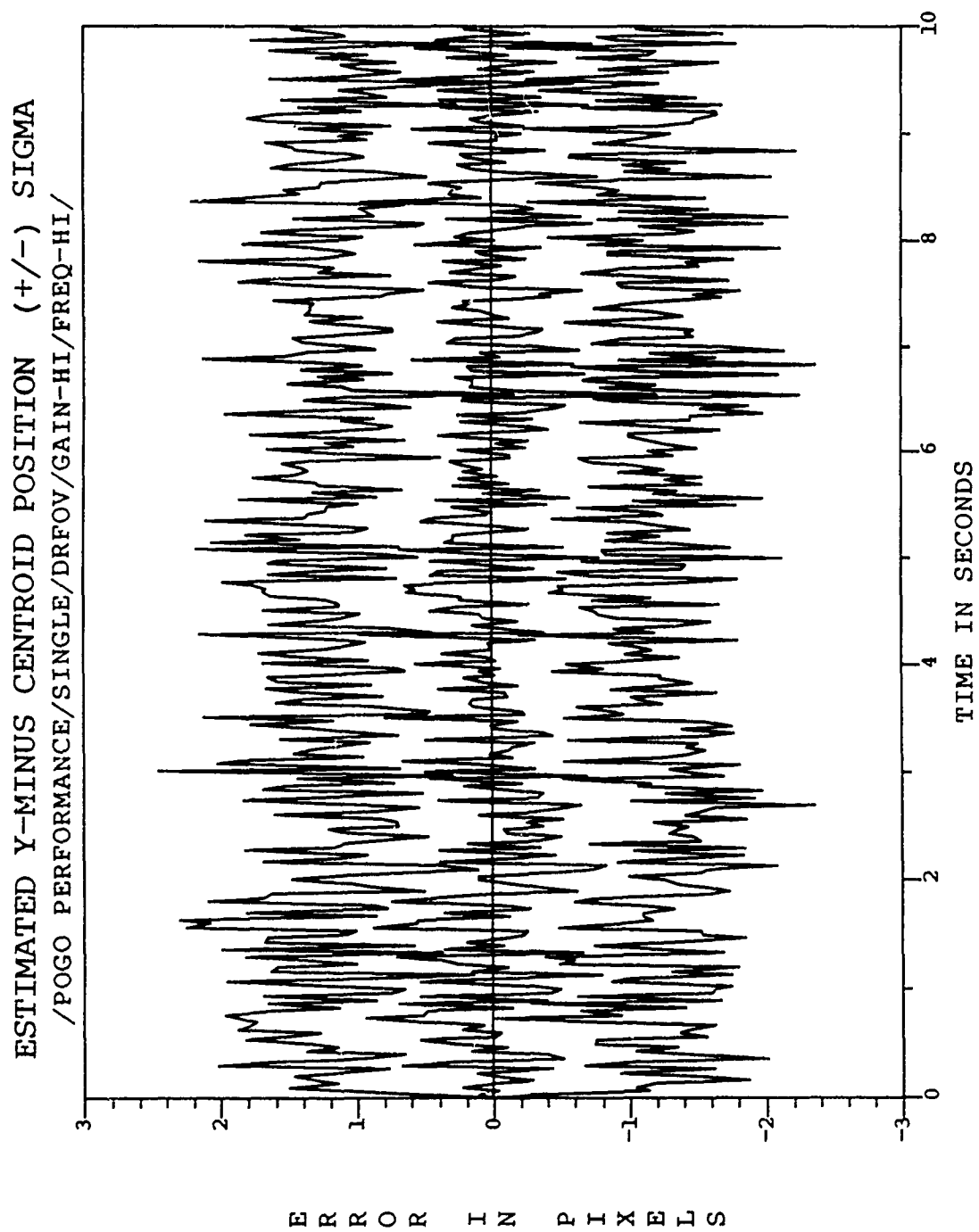


Figure E.18. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

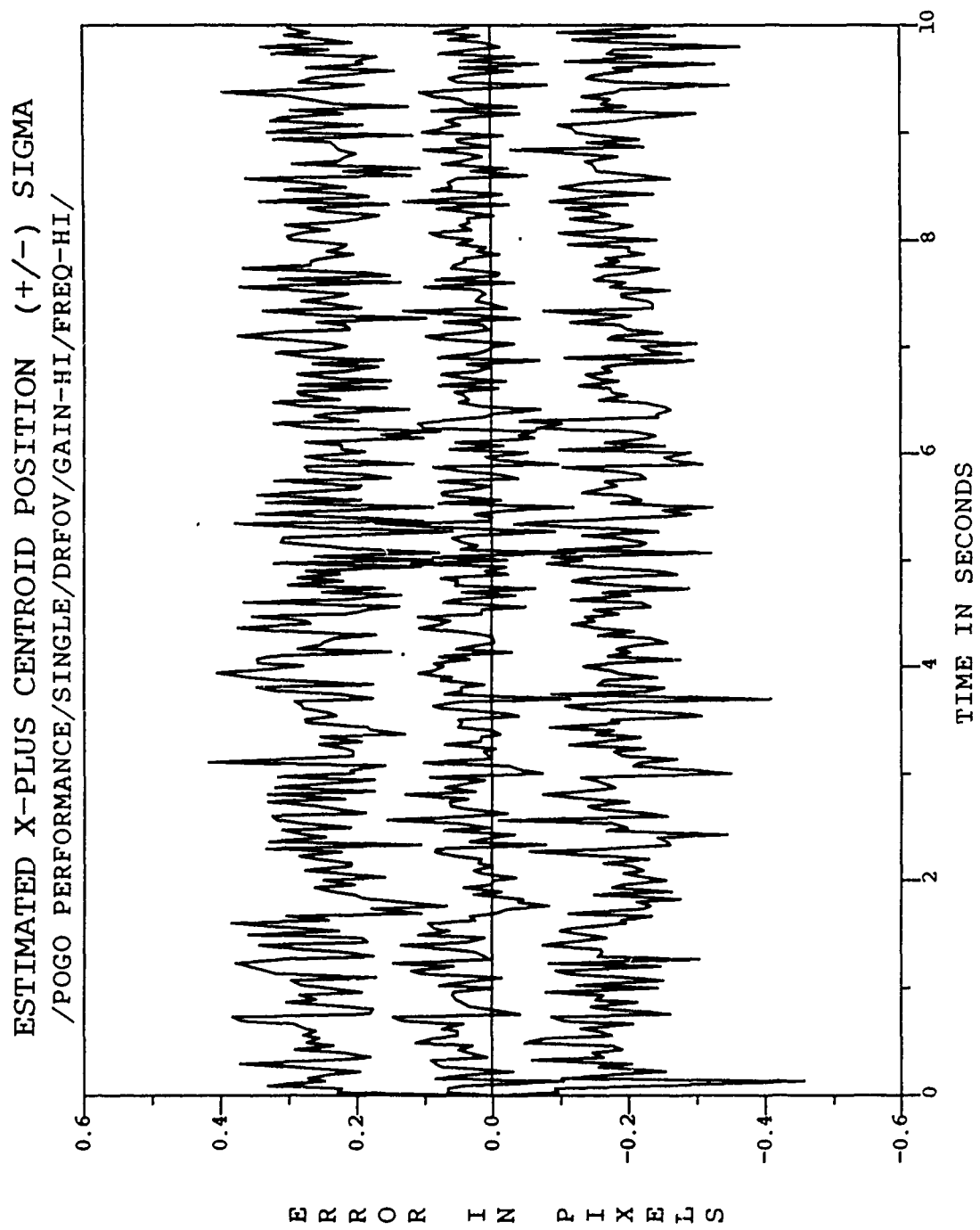


Figure E.19. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-HI/

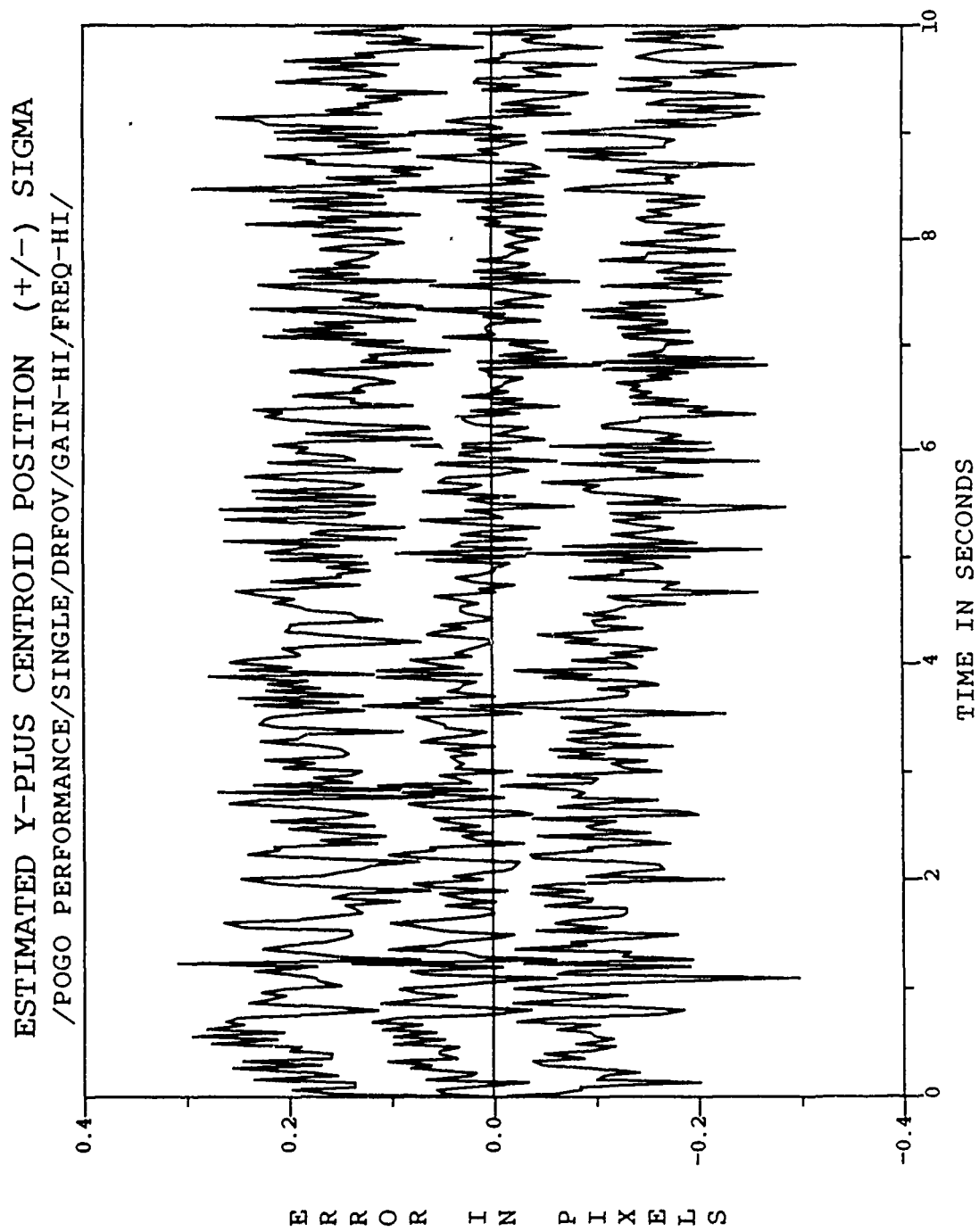


Figure E.20. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-III/

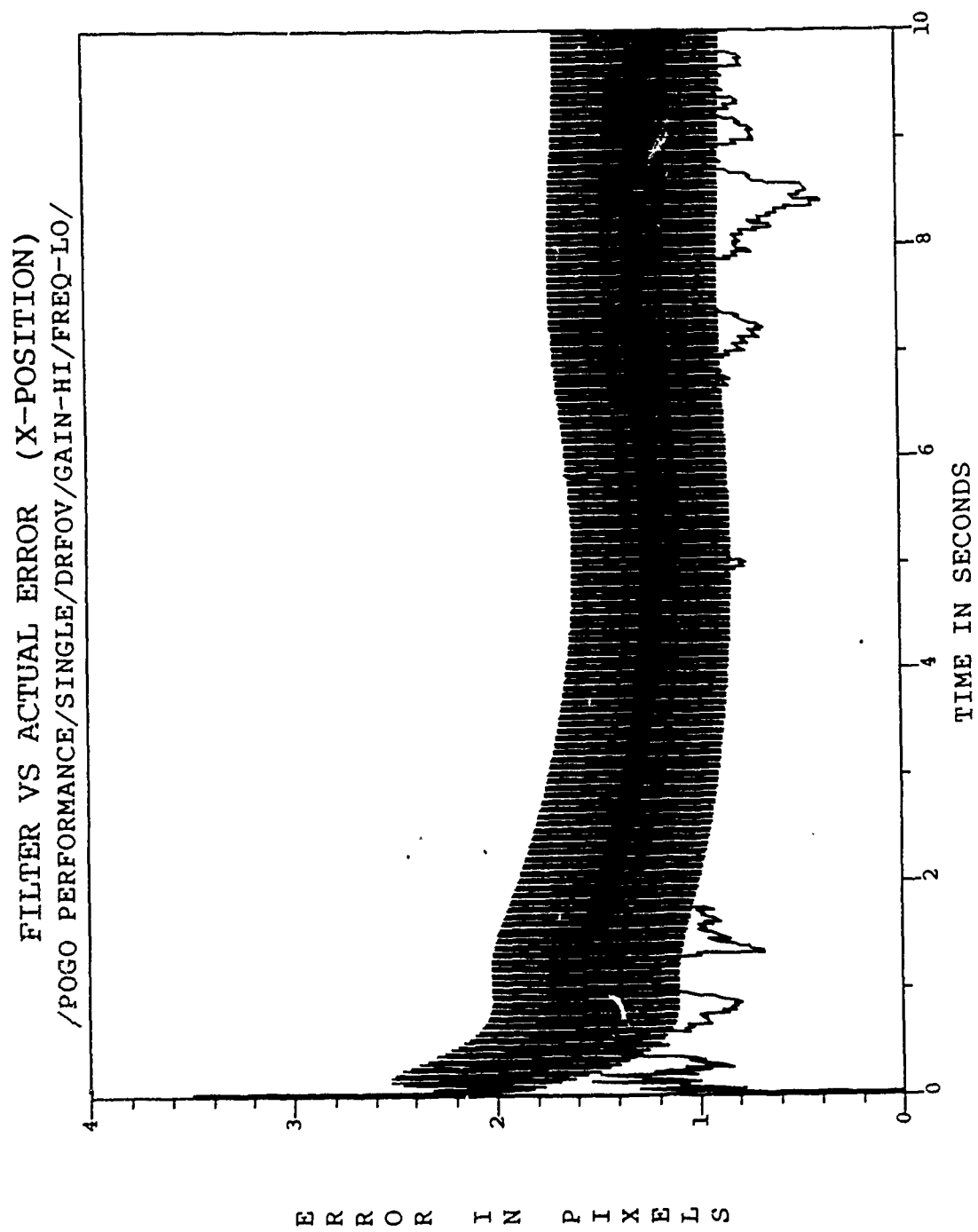


Figure E.21. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

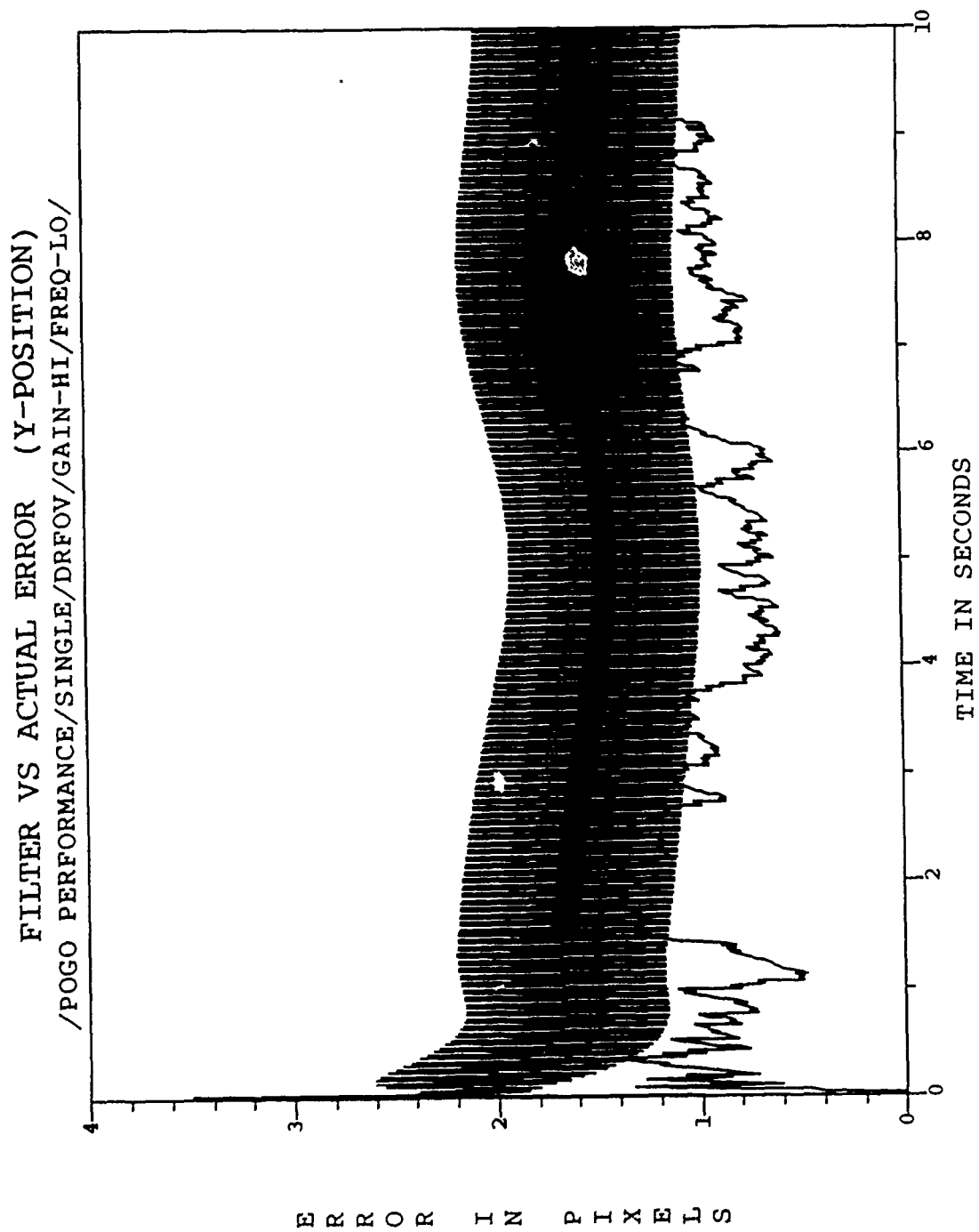


Figure E.22. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

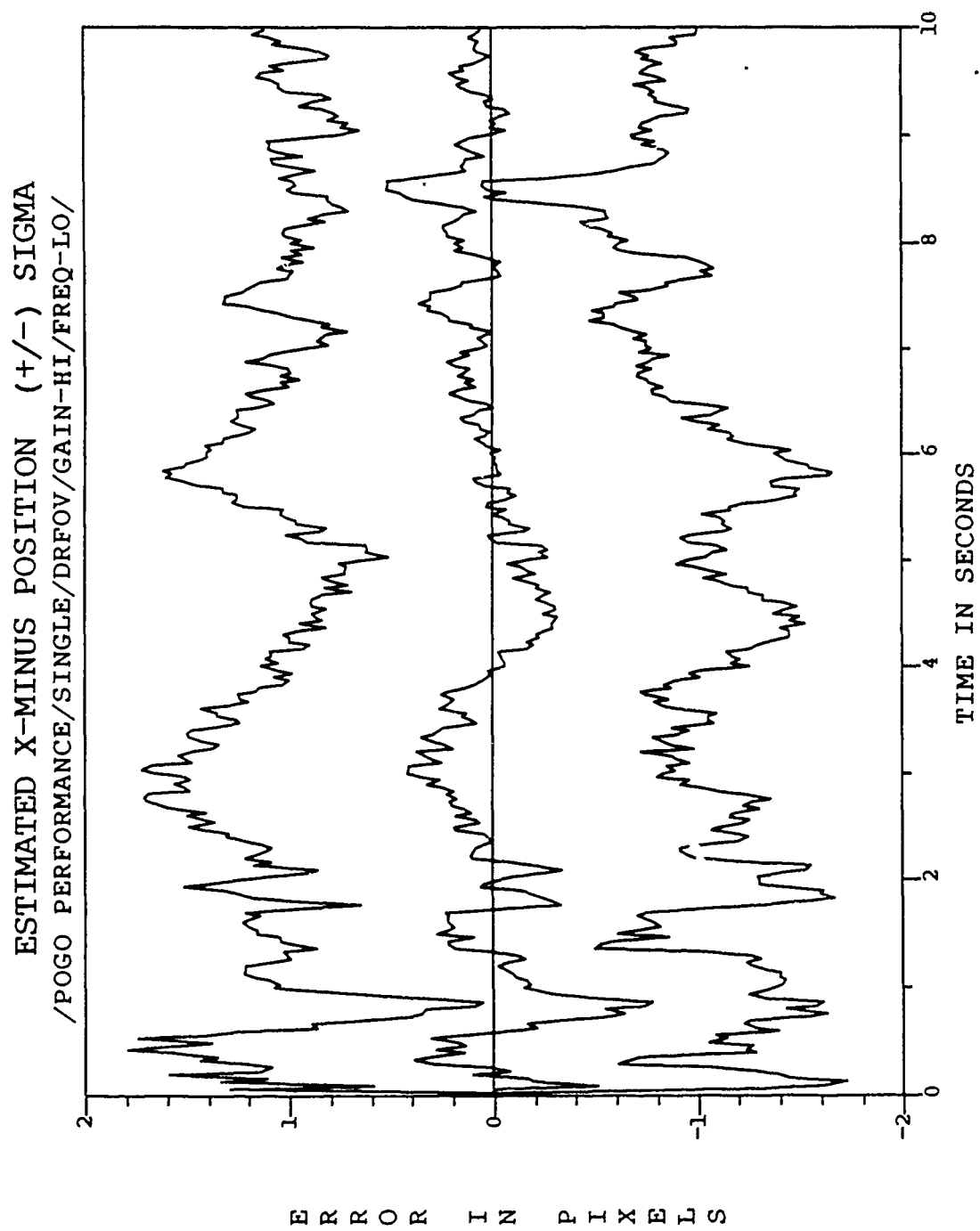


Figure E.23. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

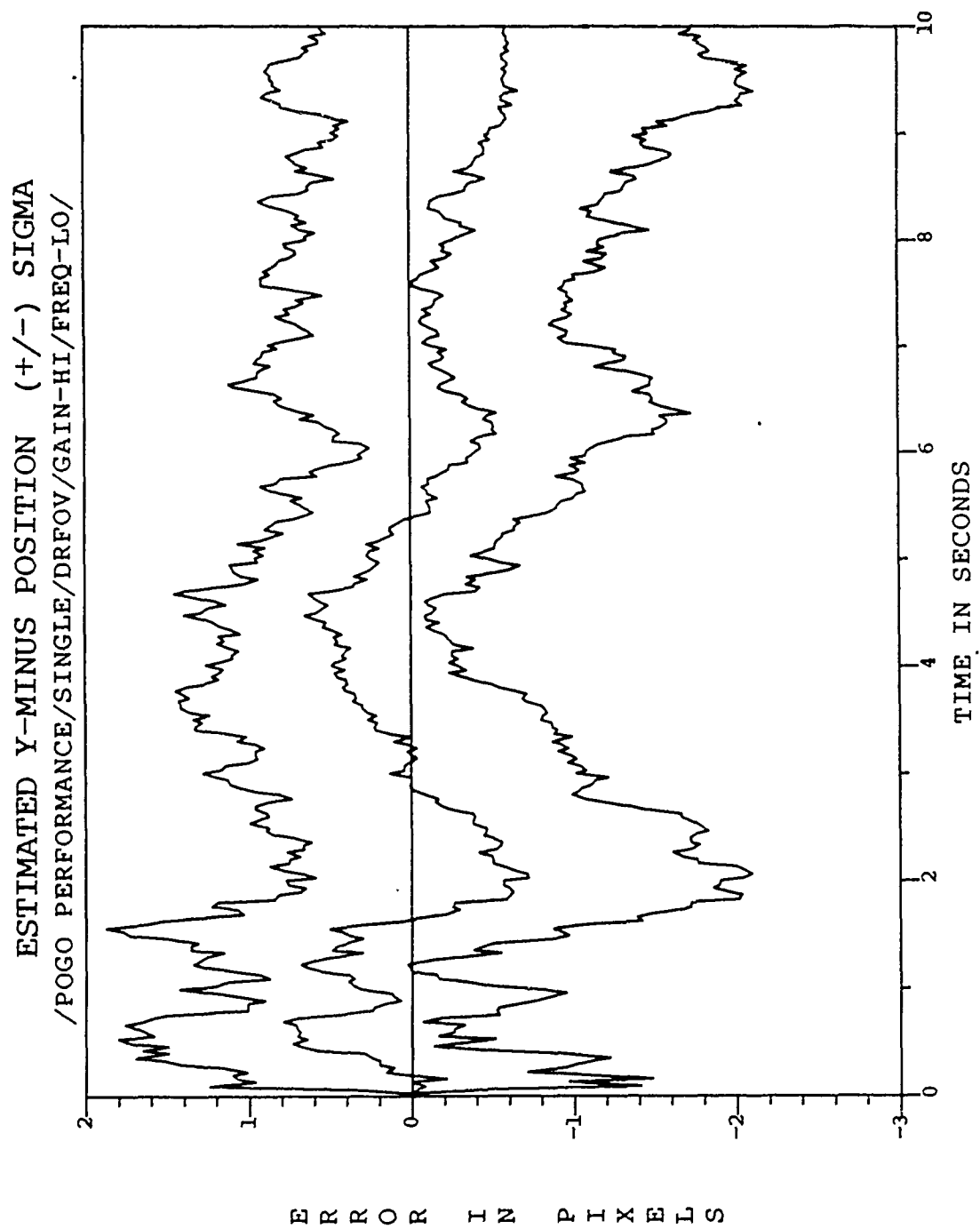


Figure E.24. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

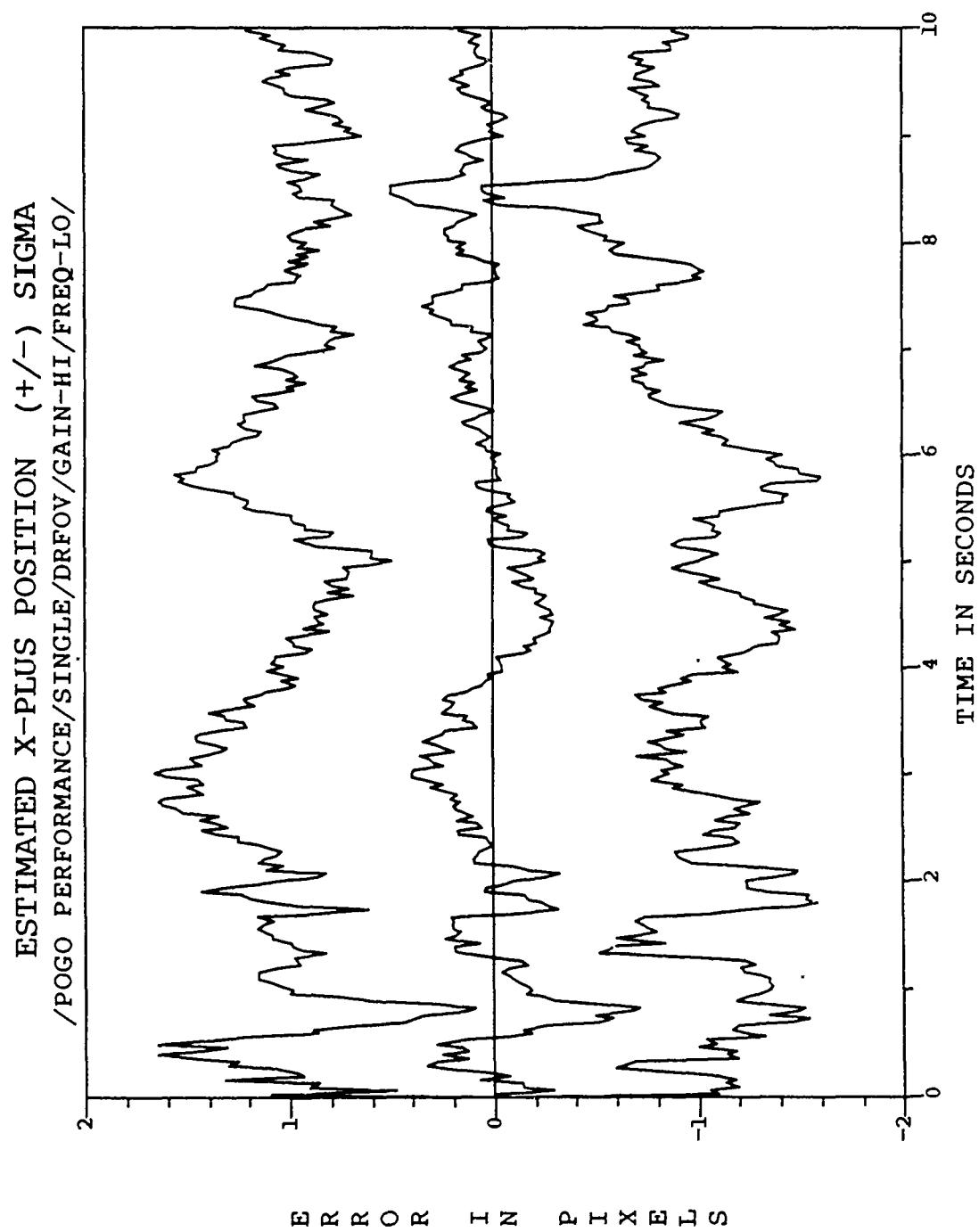


Figure E.25. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

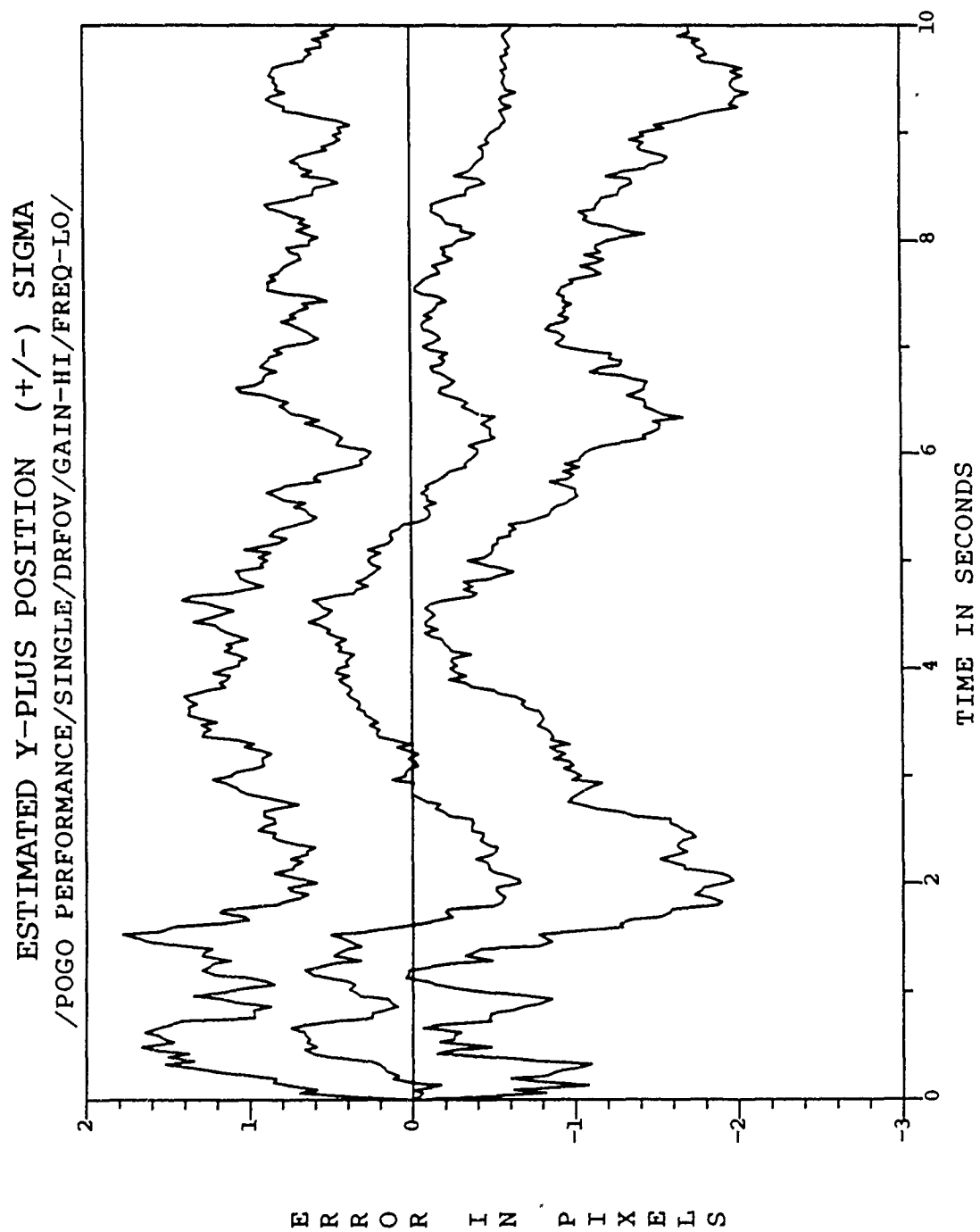


Figure E.26. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-LO/

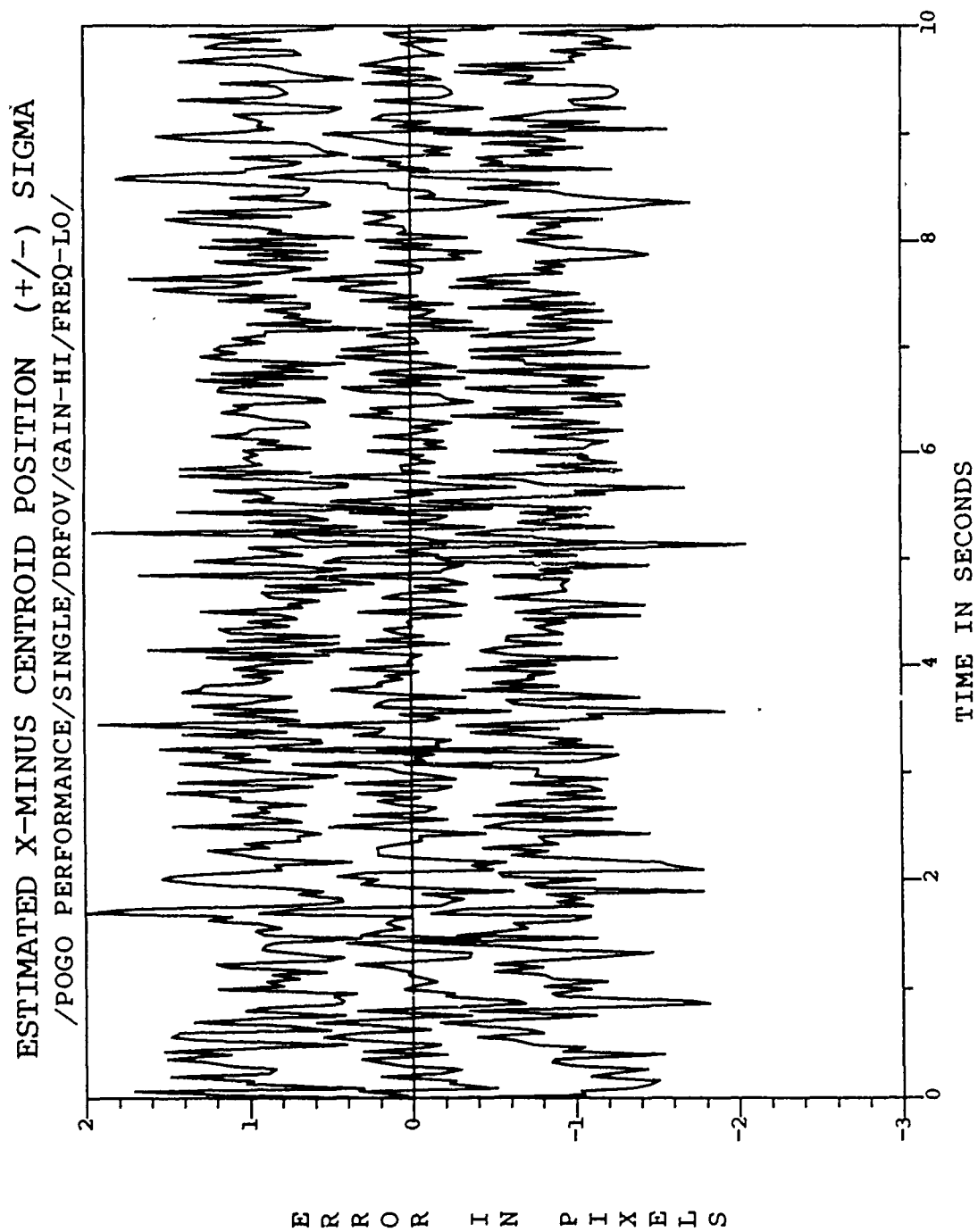


Figure E.27. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

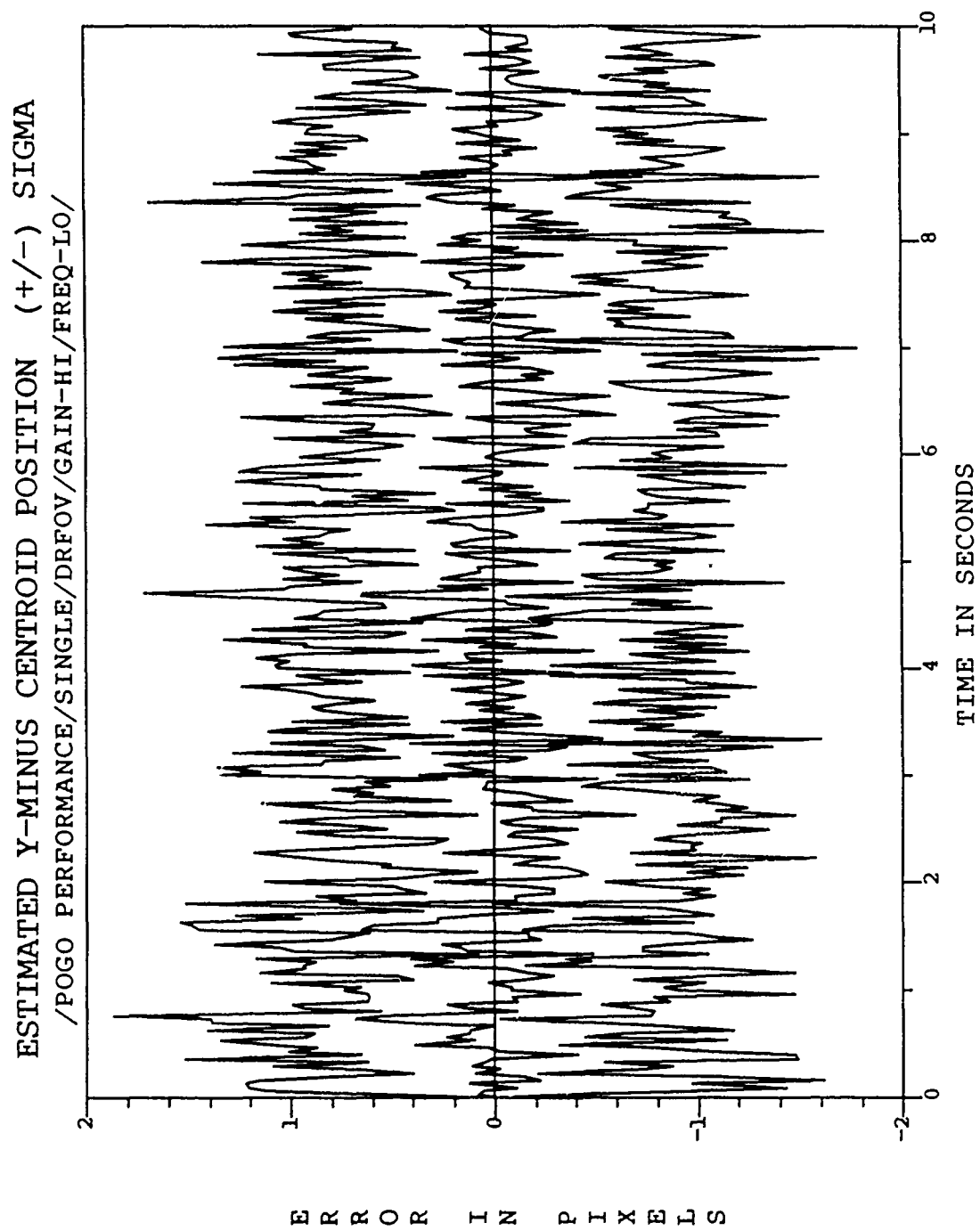


Figure E.28. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-LO/

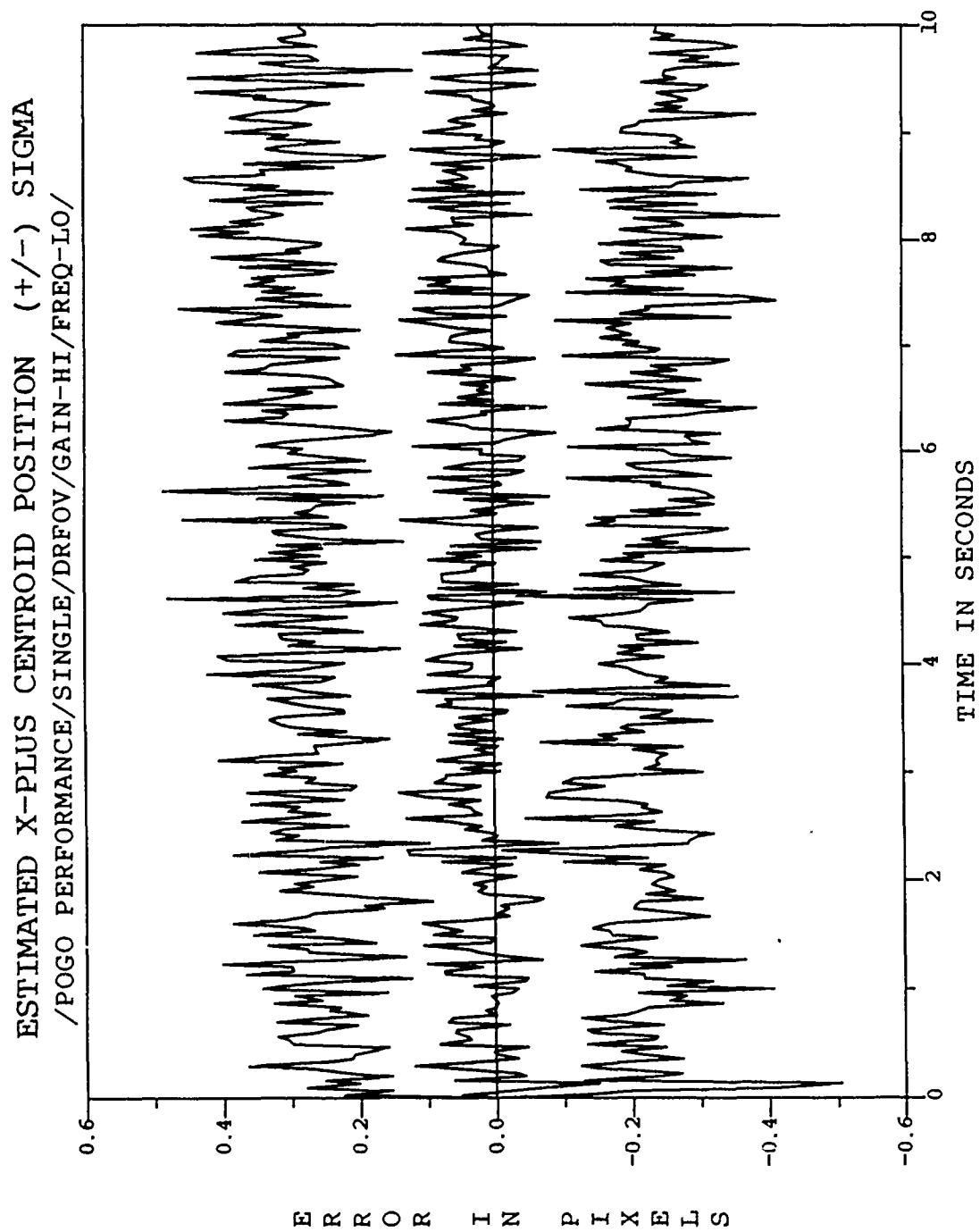


Figure E.29. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

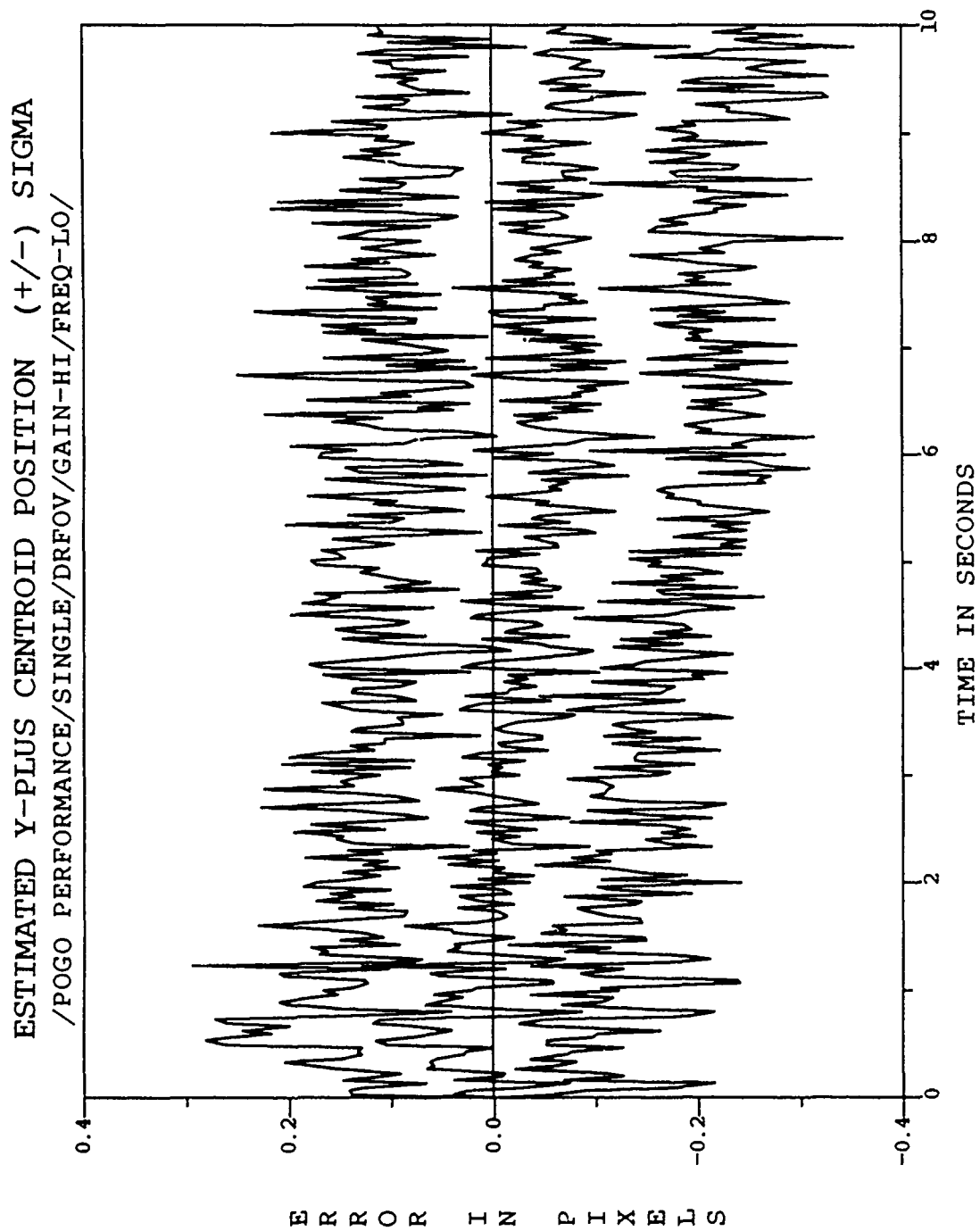


Figure E.30. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-LO/

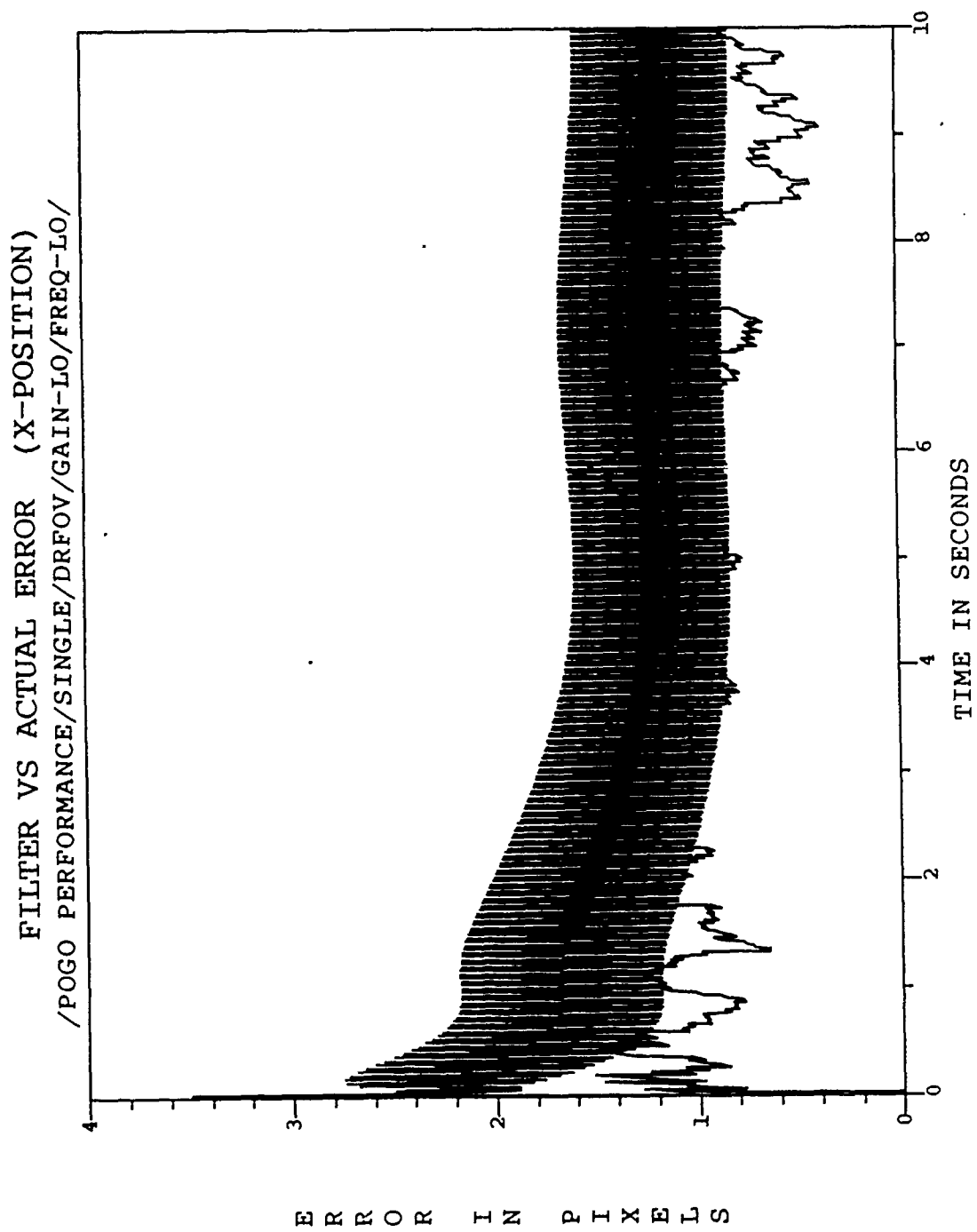


Figure E.31. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

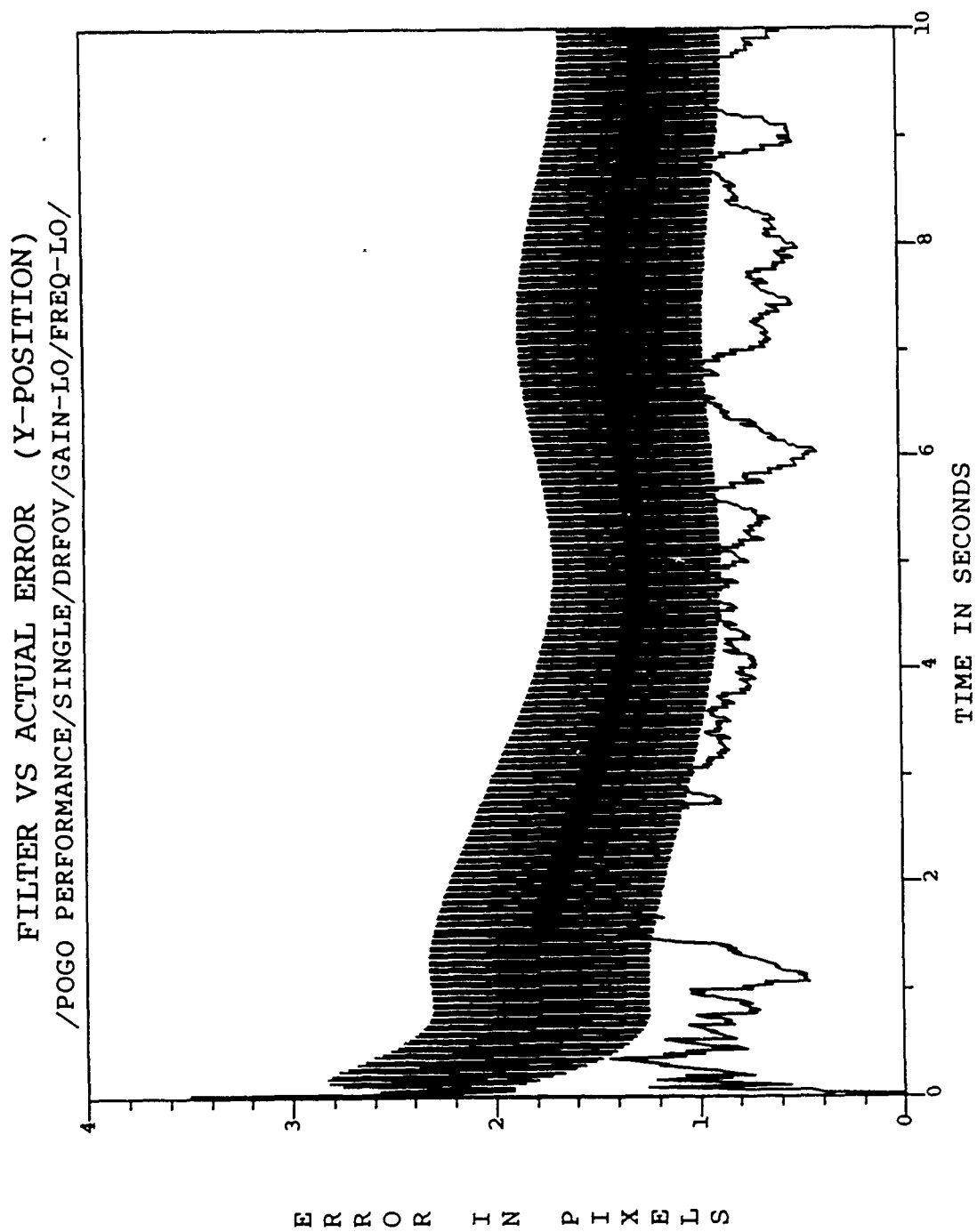


Figure E.32. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

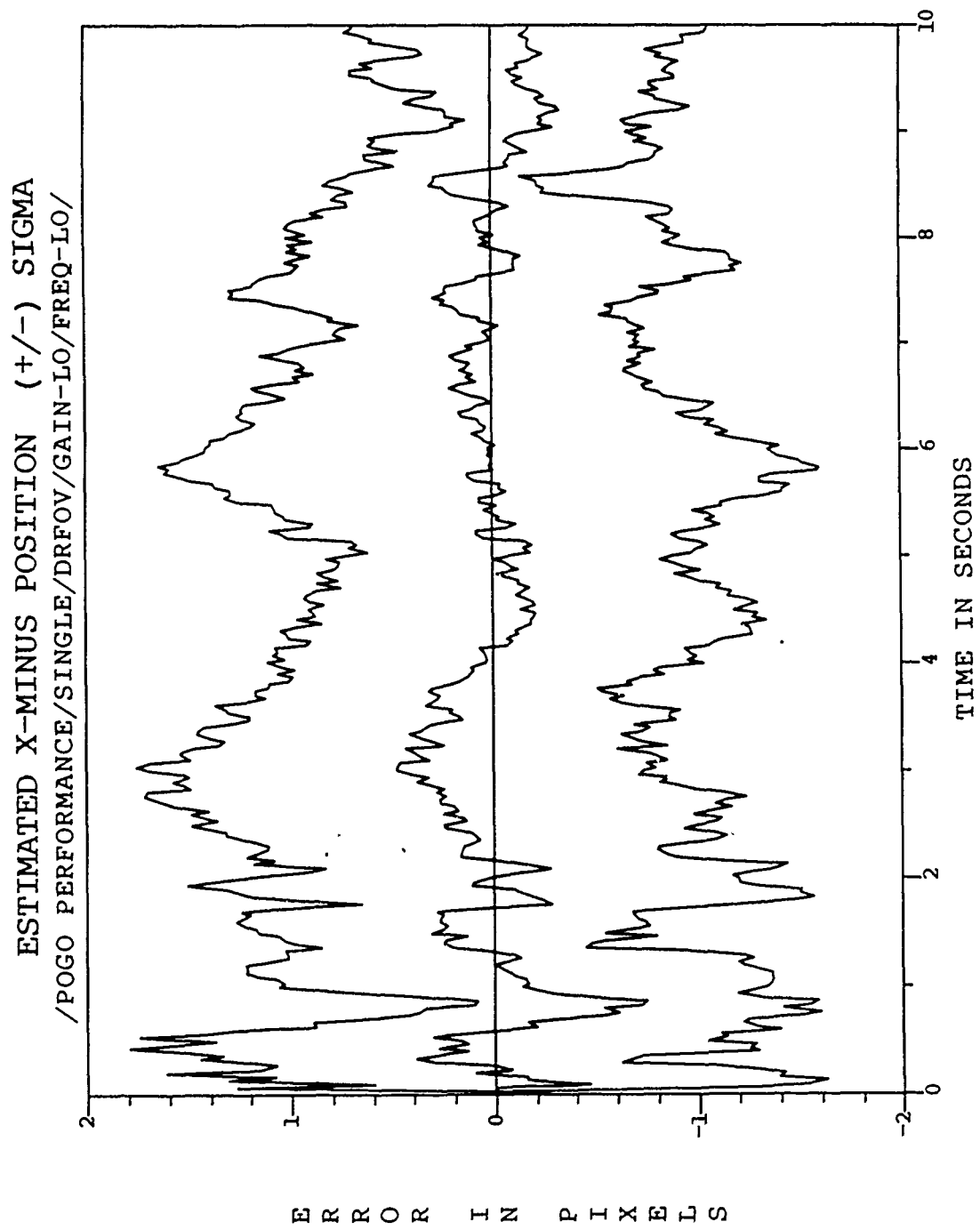


Figure E.33. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

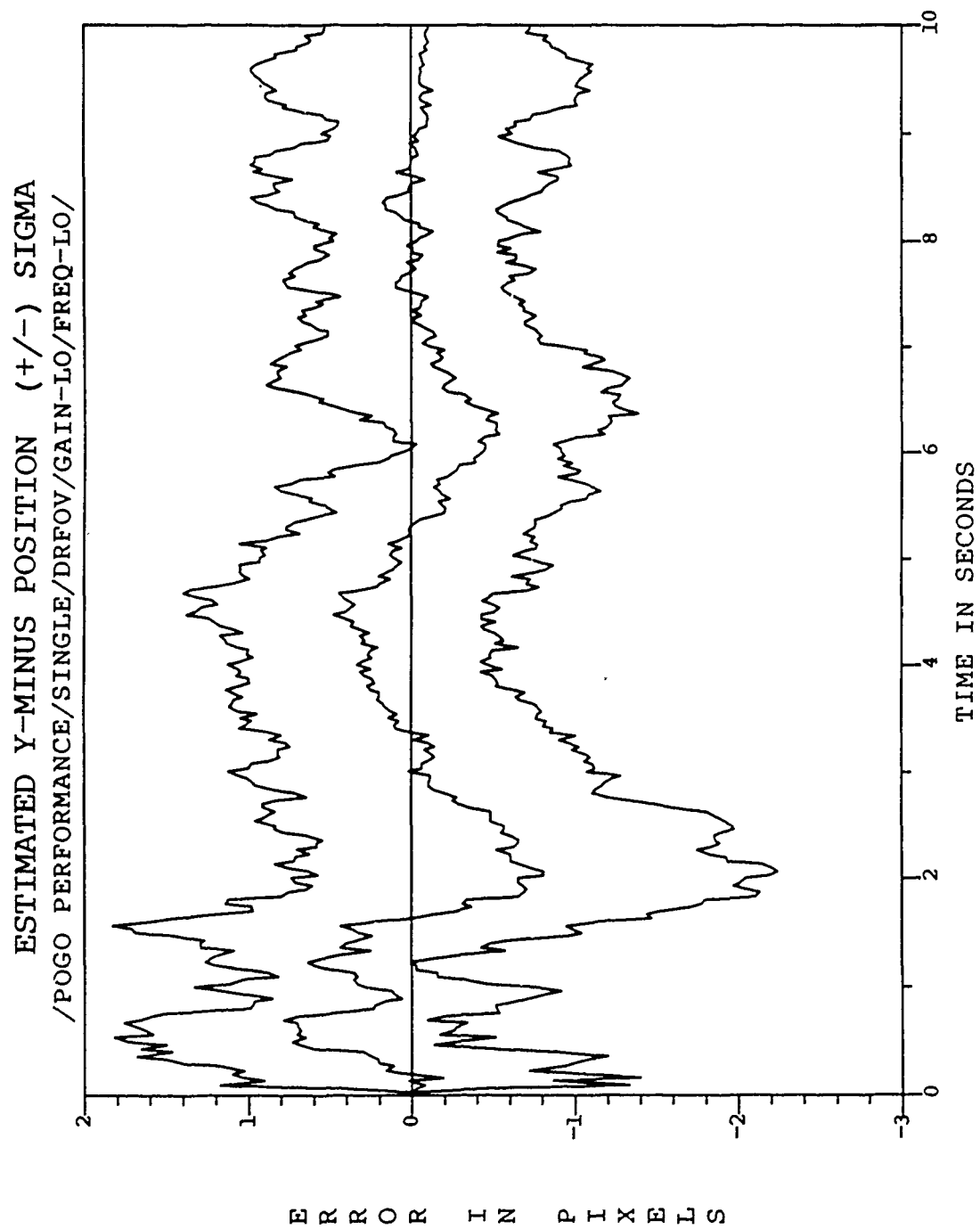


Figure E.34. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

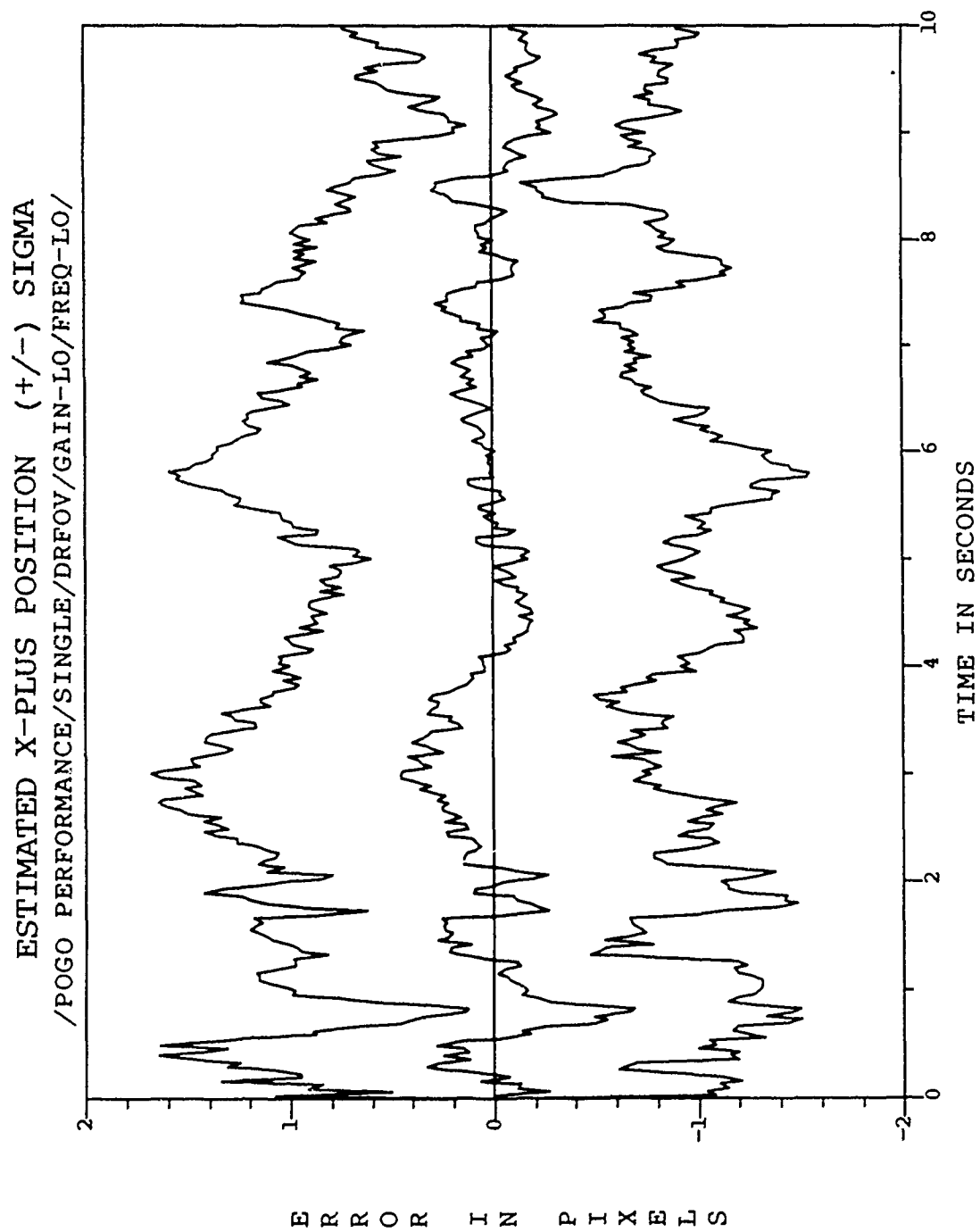


Figure E.35. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

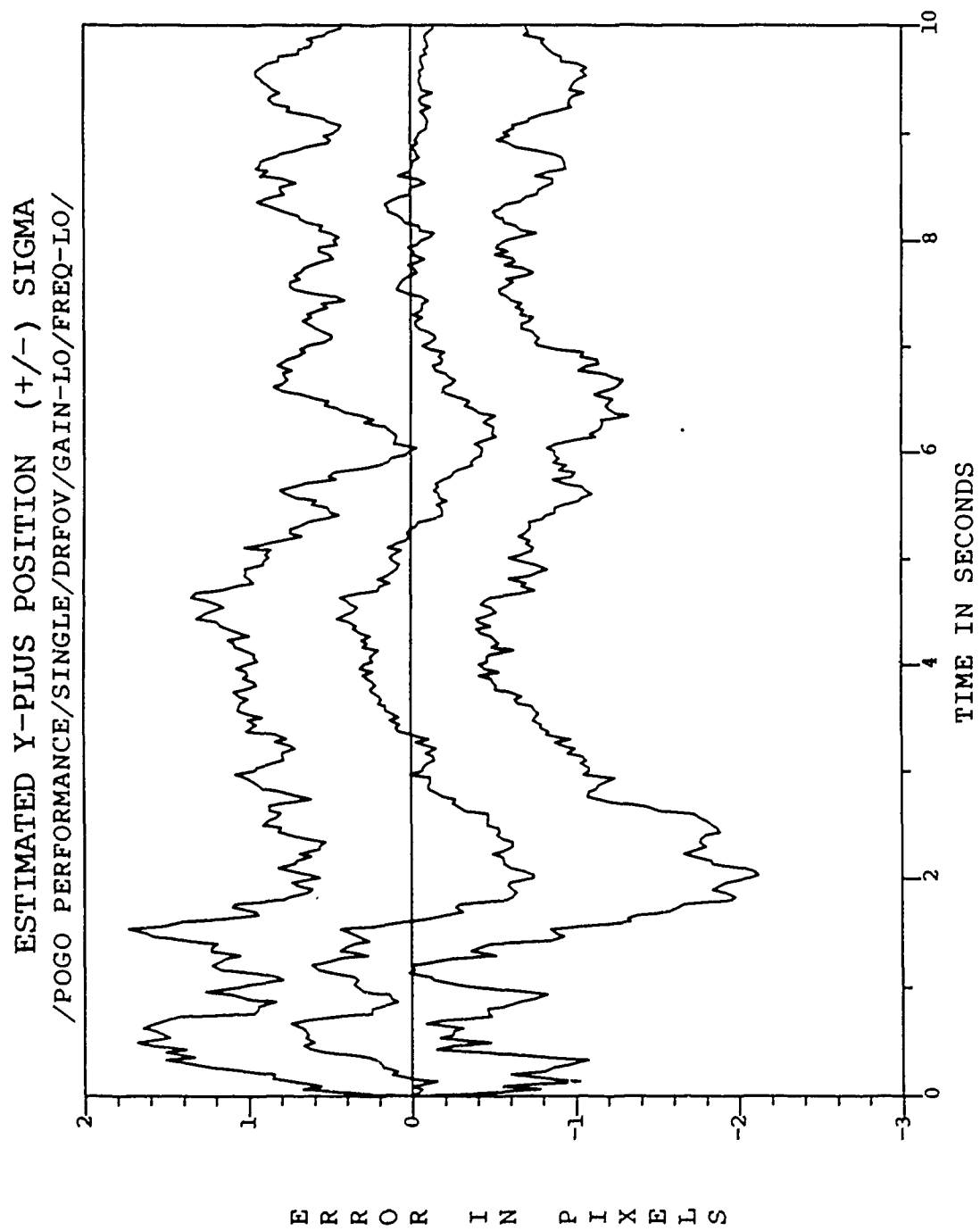


Figure E.36. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

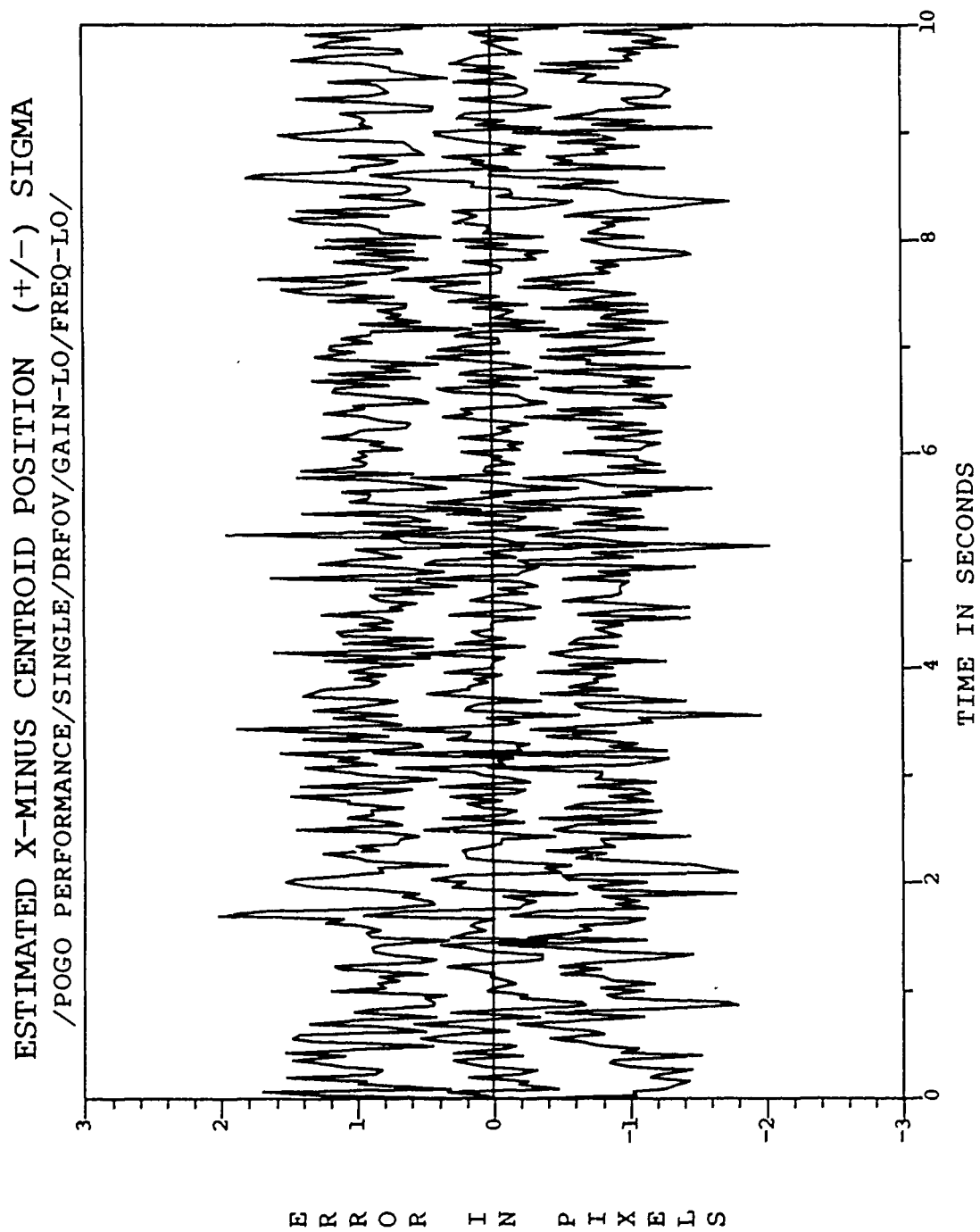


Figure E.37. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

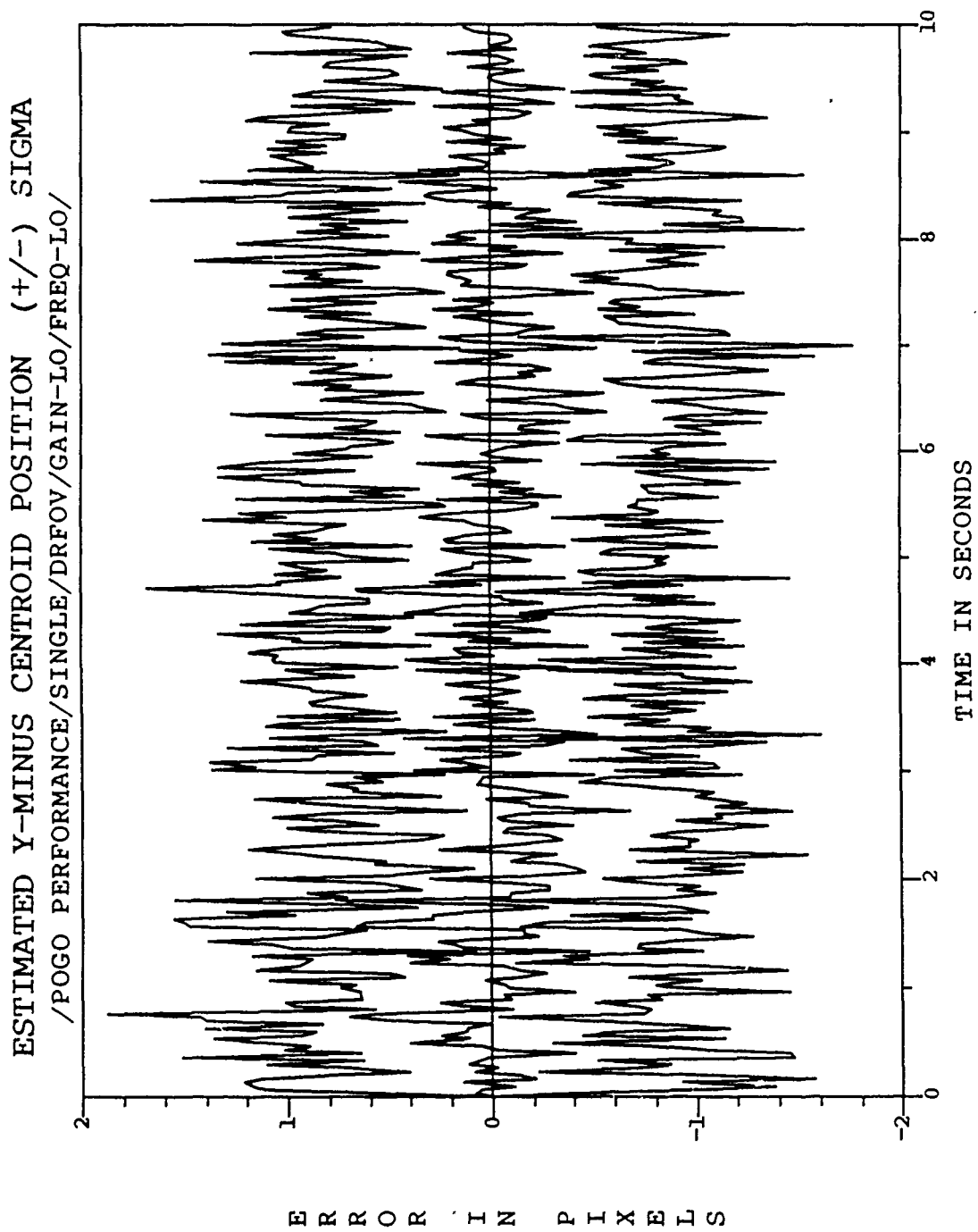


Figure E.38. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

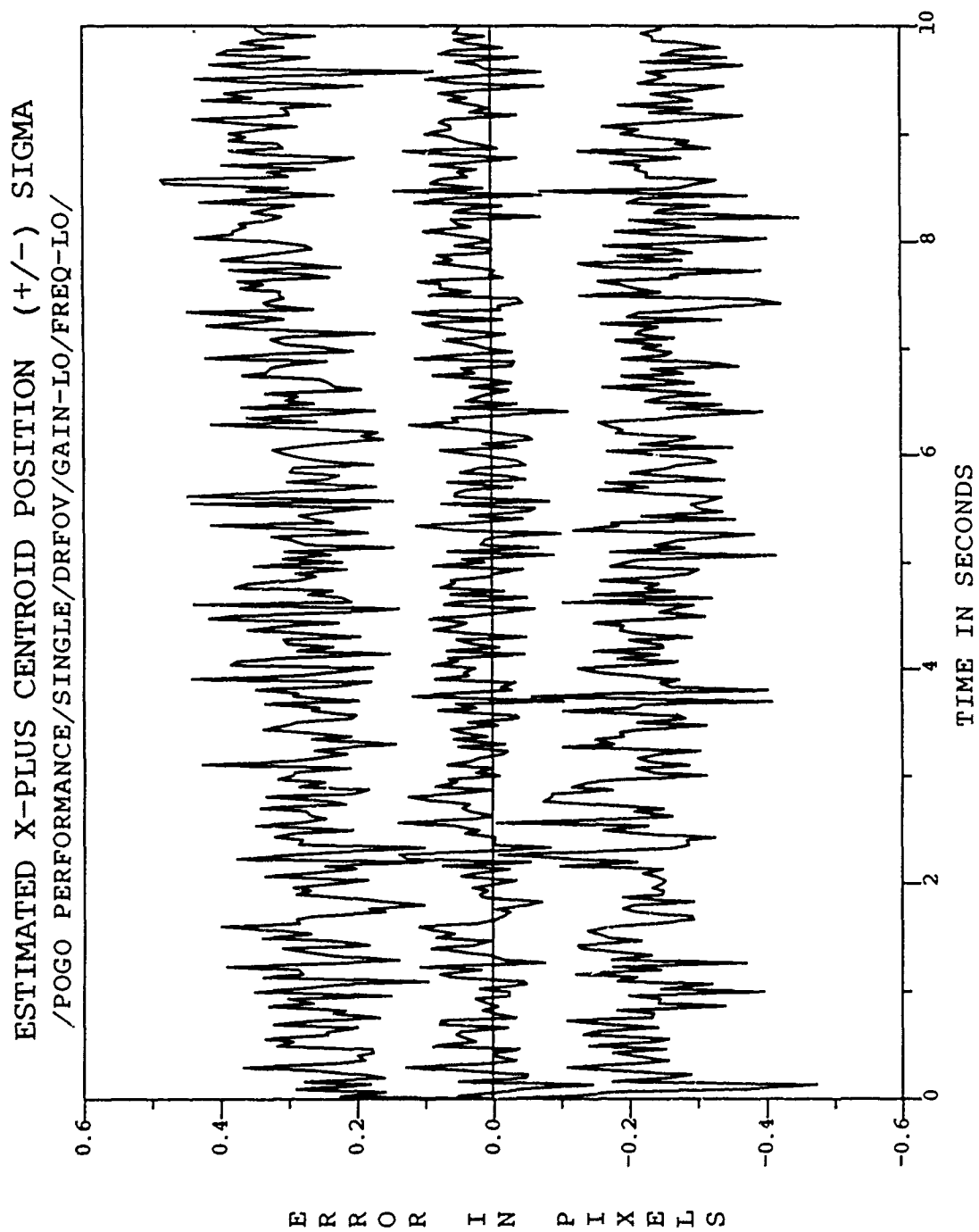


Figure E.39. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

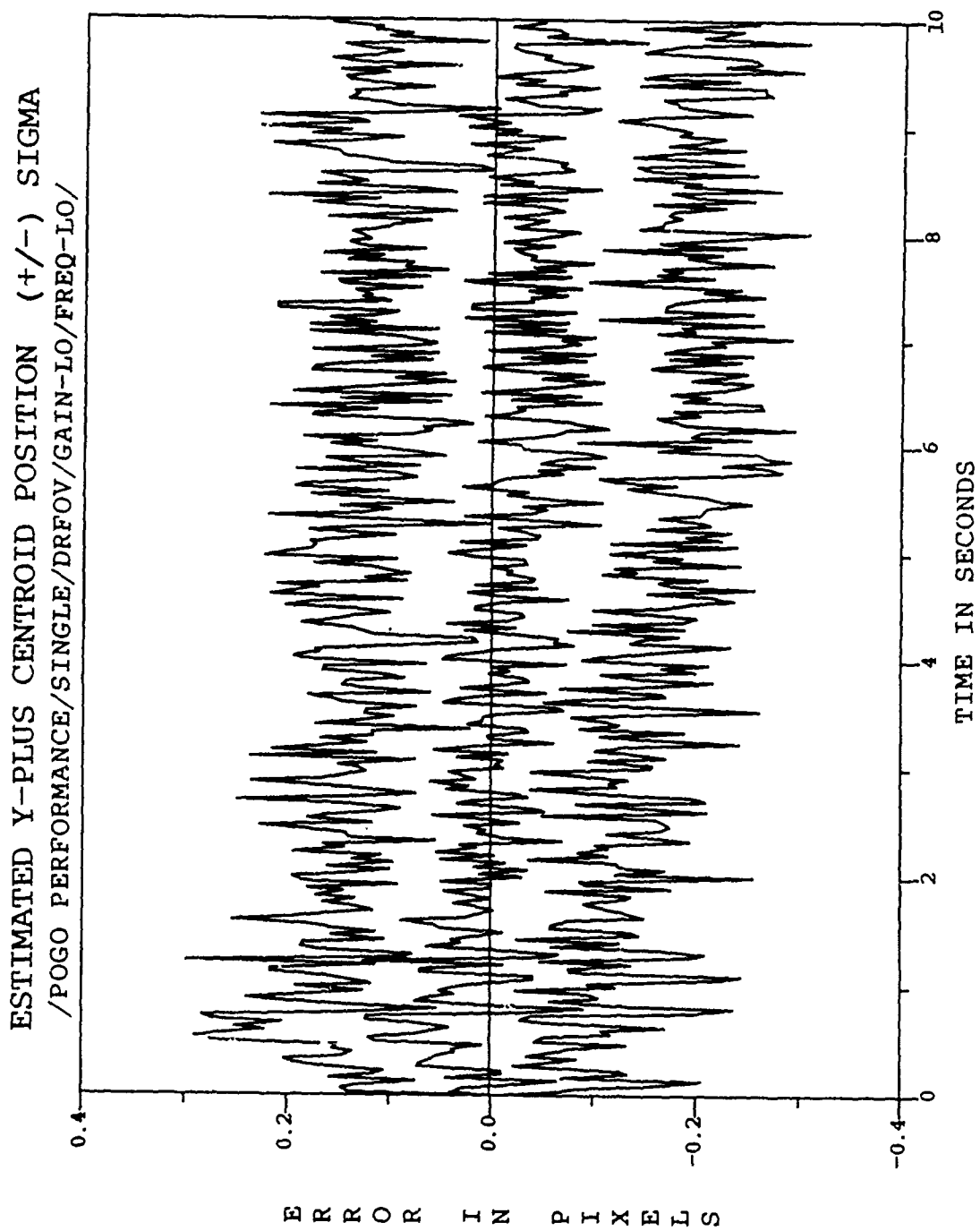


Figure E.40. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-LO/FREQ-LO

Appendix F. *8-State/10-State Filter Performance Plots: Bending
Removed from Truth Model (Discussions in Sections 6.6 and 6.7)*

F.1 Figures F.1-F.10: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = .85$

F.2 Figures F.11-F.20: $\sigma_x^2, \sigma_y^2 = 5; \tau_x, \tau_y = 8.5; \sigma_a^2 = .70$

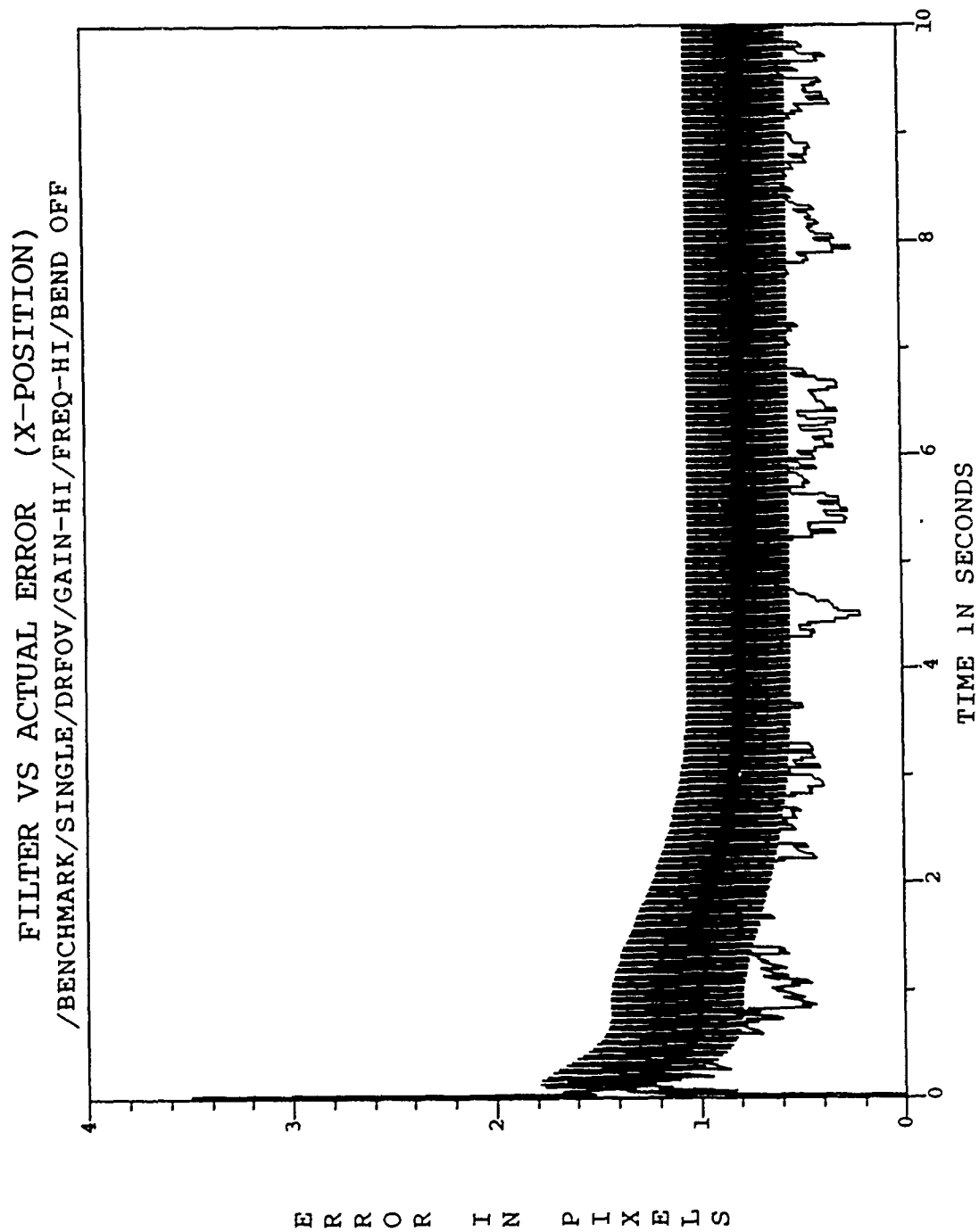


Figure F.1. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

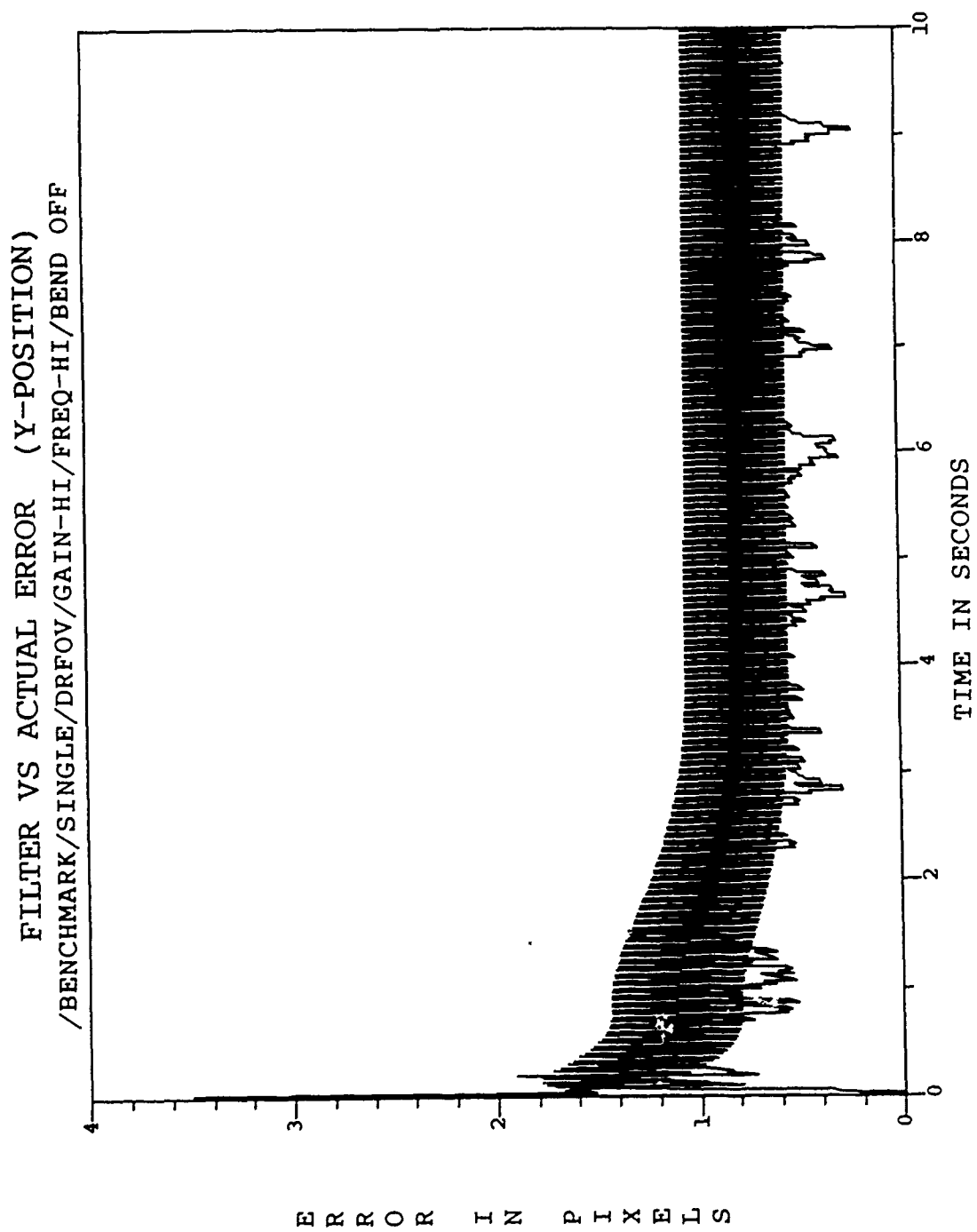


Figure F.2. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

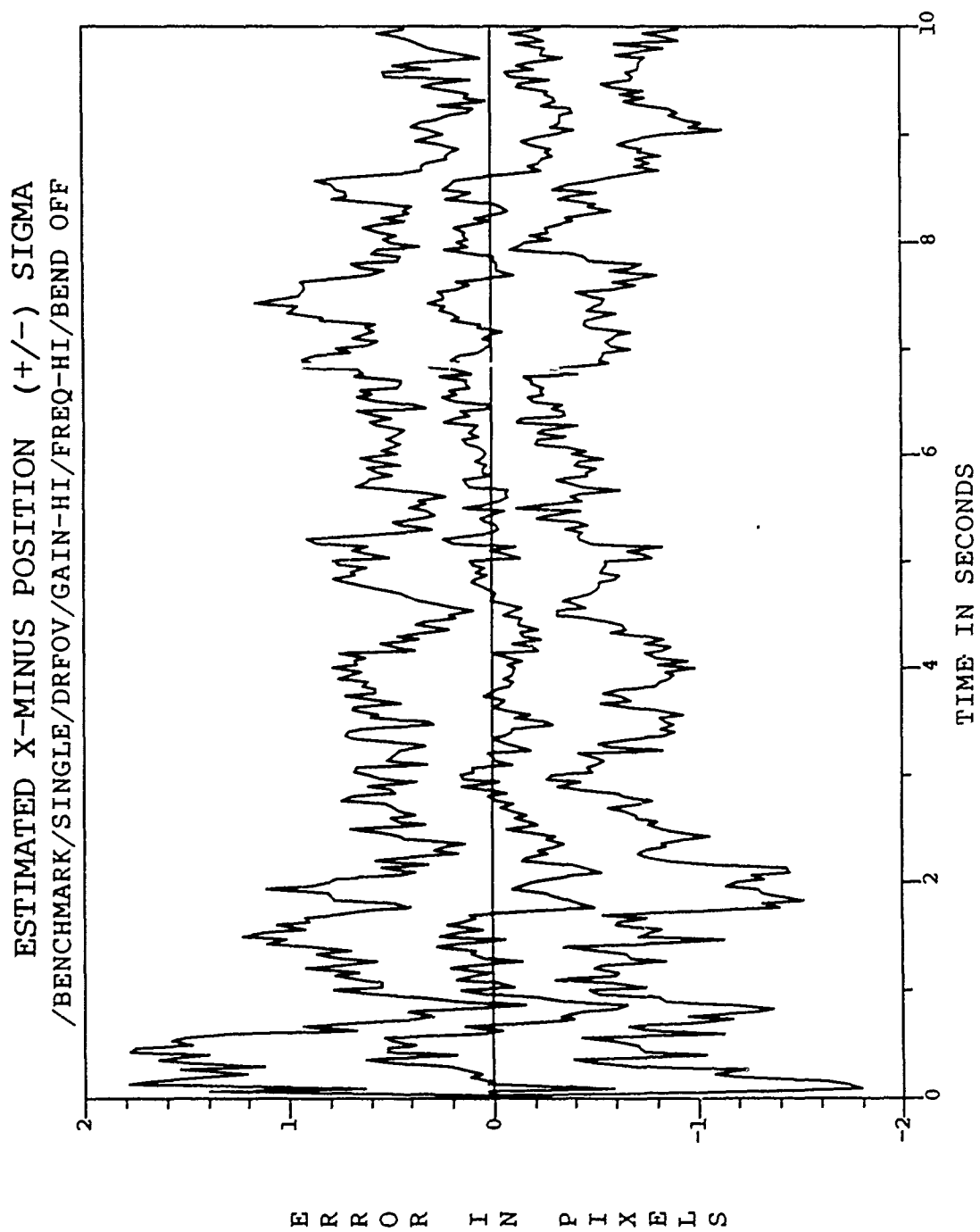


Figure F.3. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-HI/

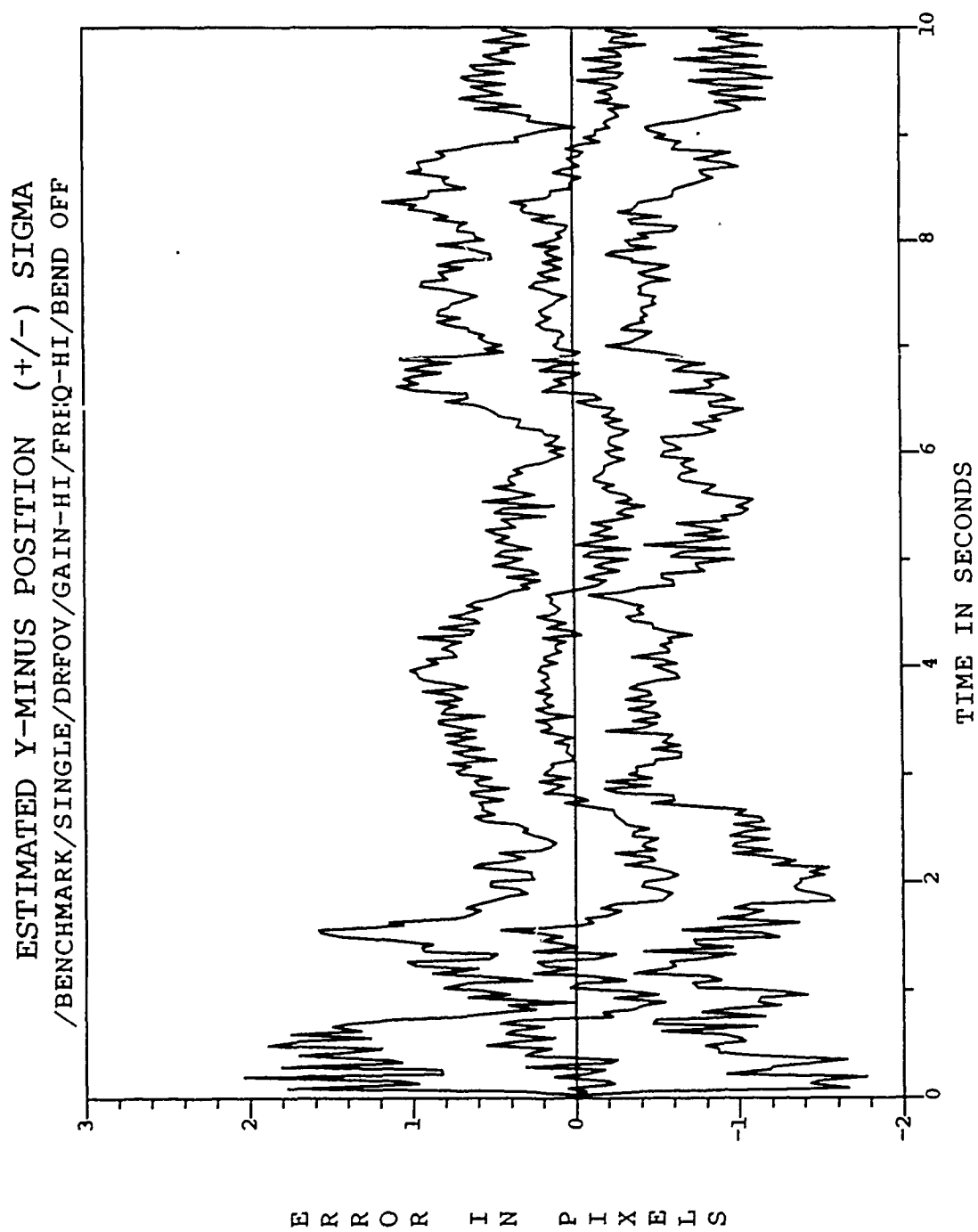


Figure F.4. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

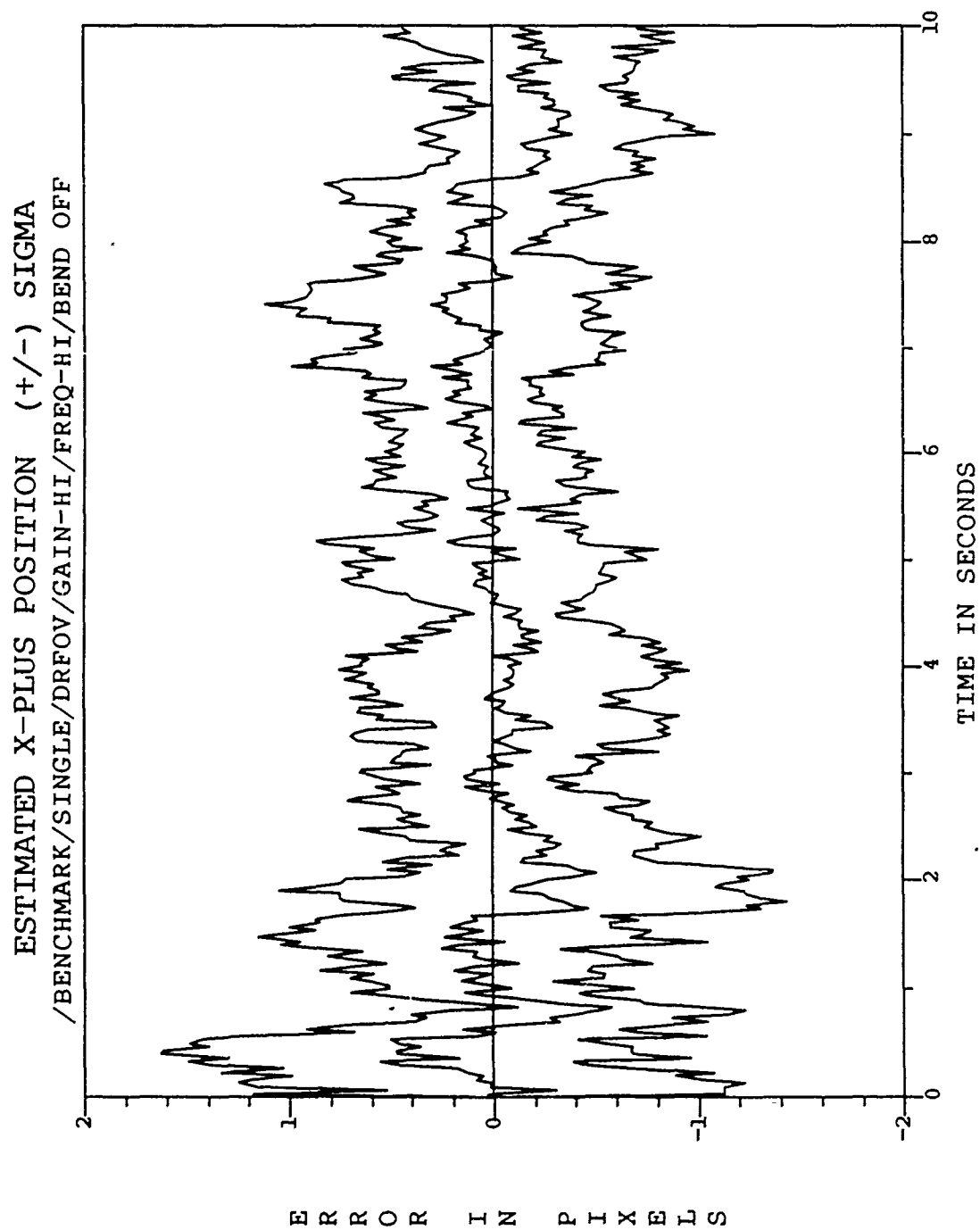


Figure F.5. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

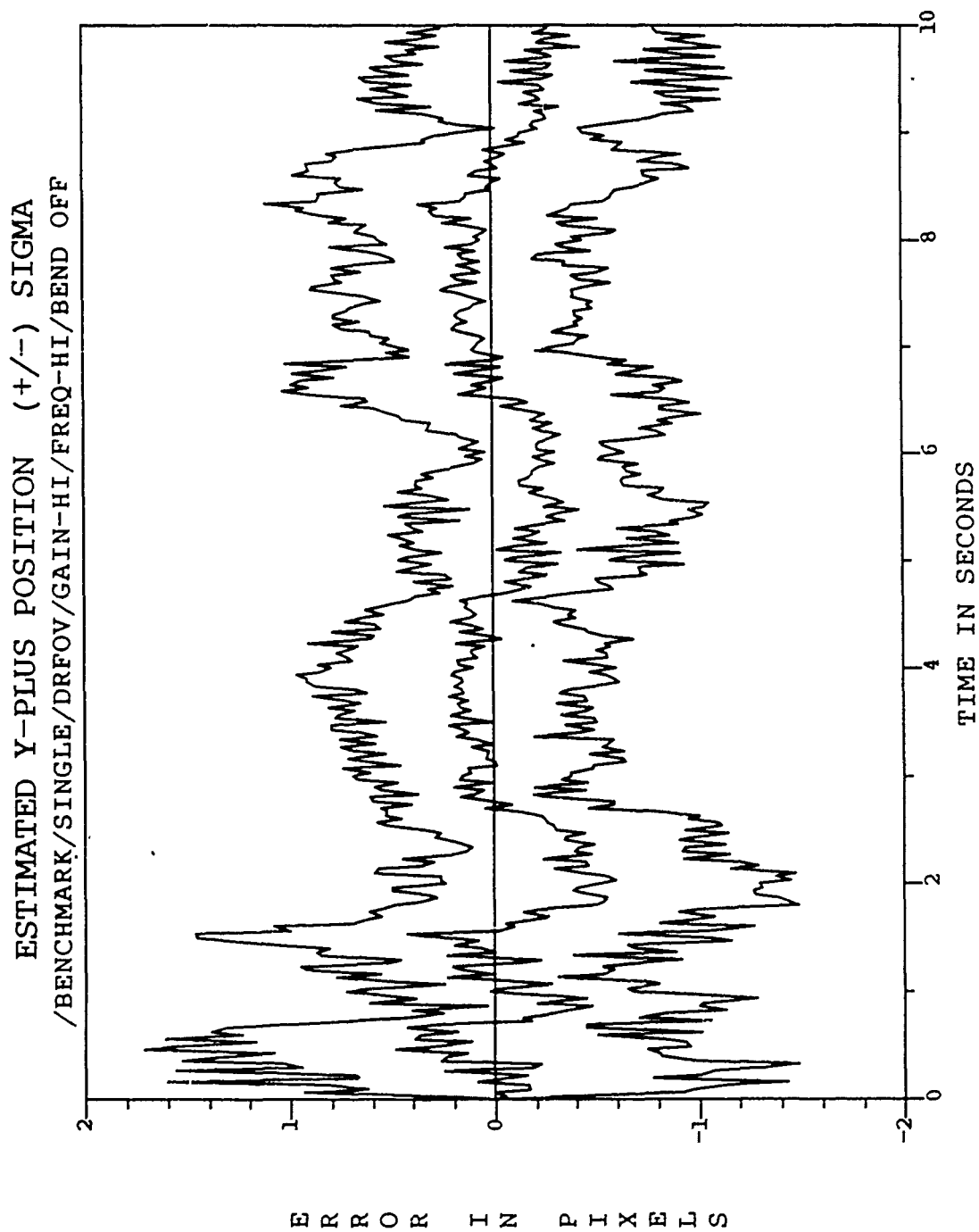


Figure F.6. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

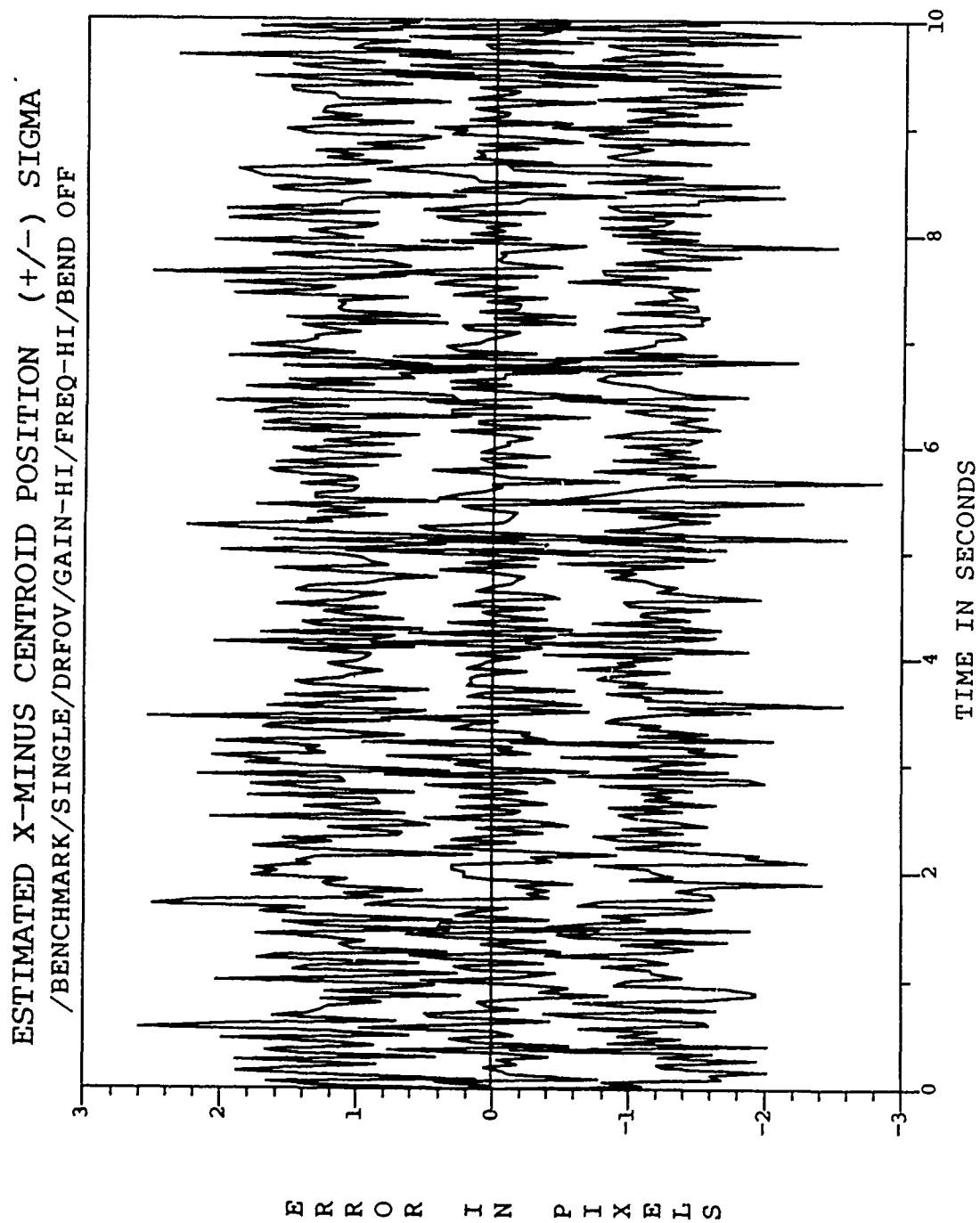


Figure F.7. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

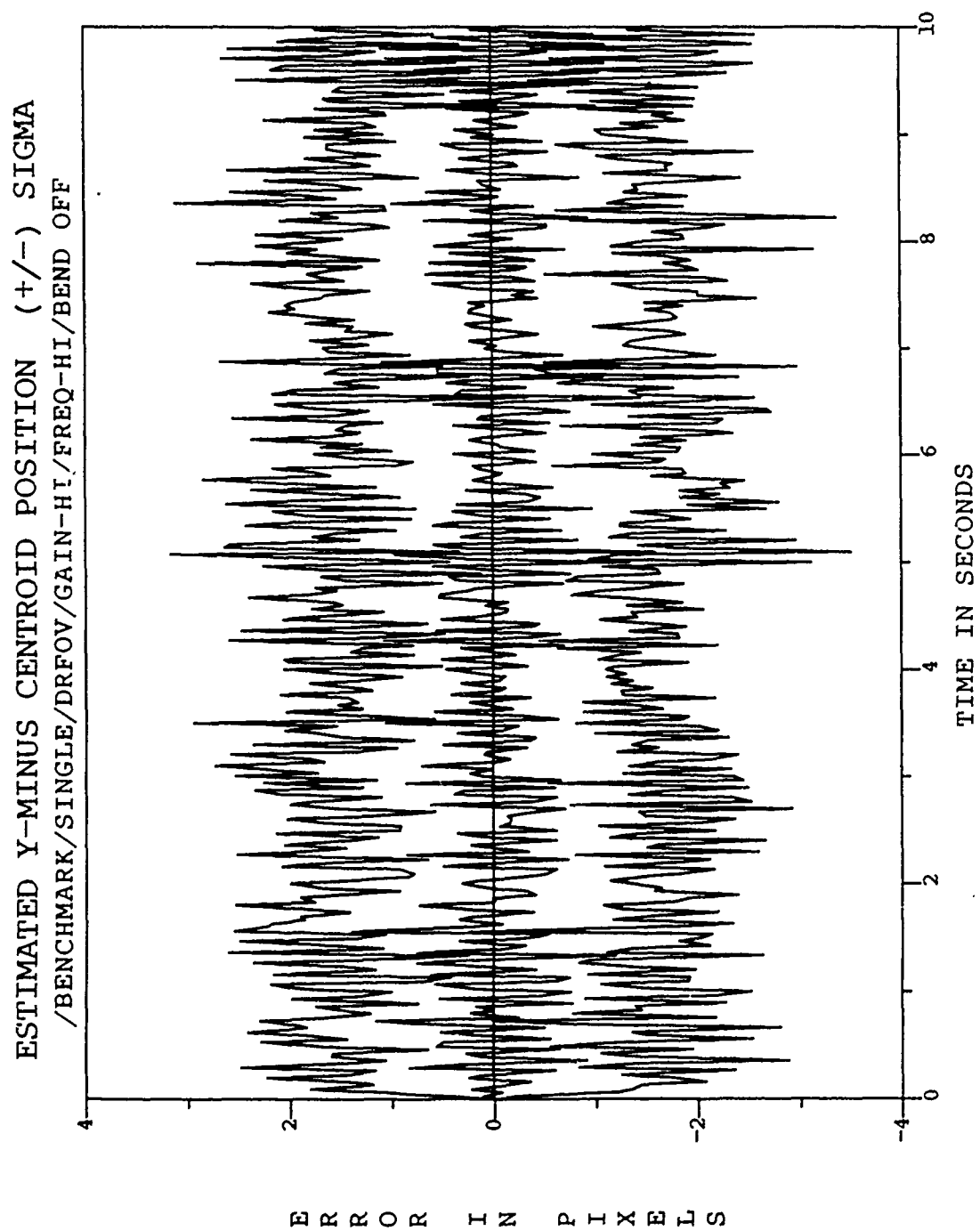


Figure F.8. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

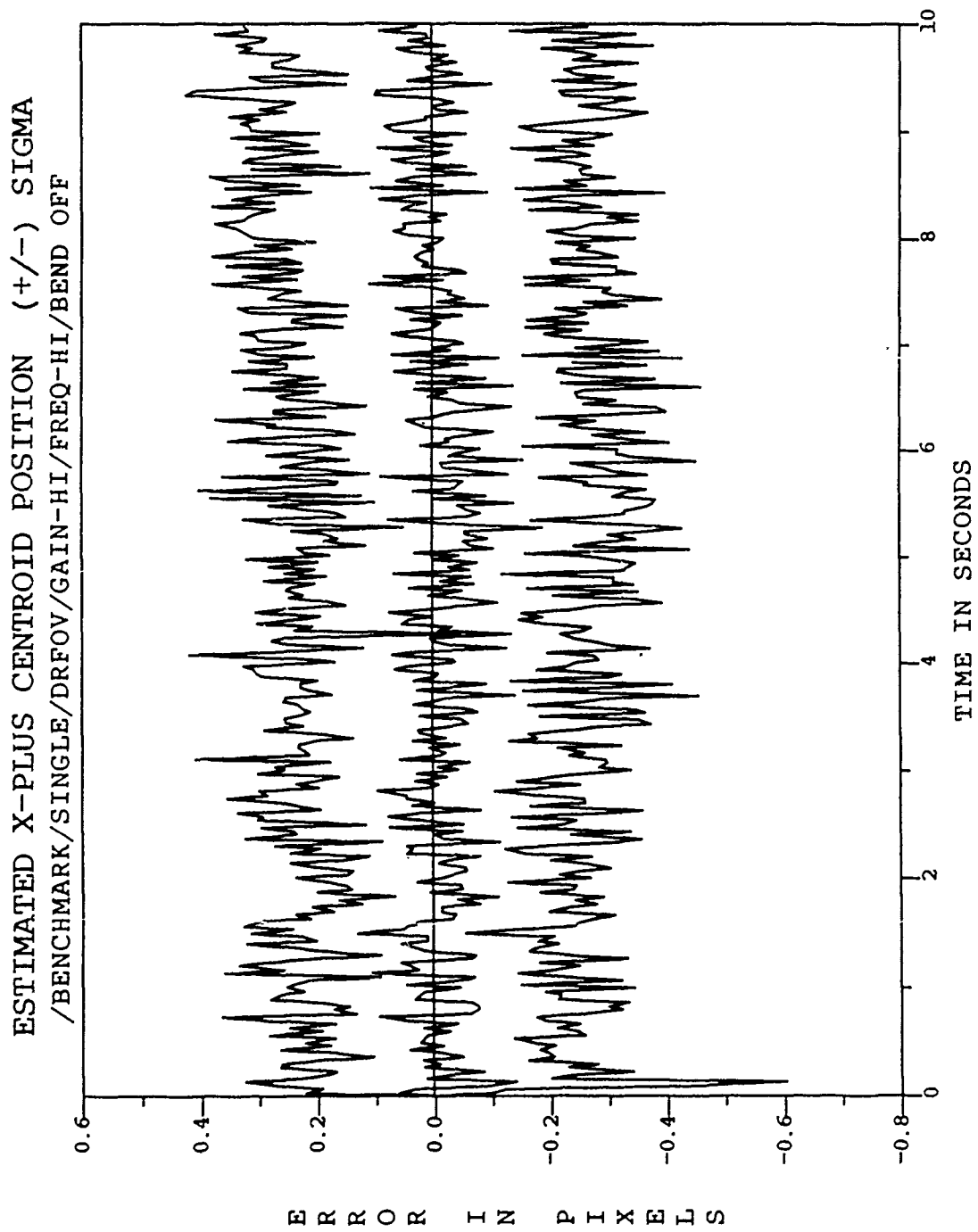


Figure F.9. /BENCHMARK/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

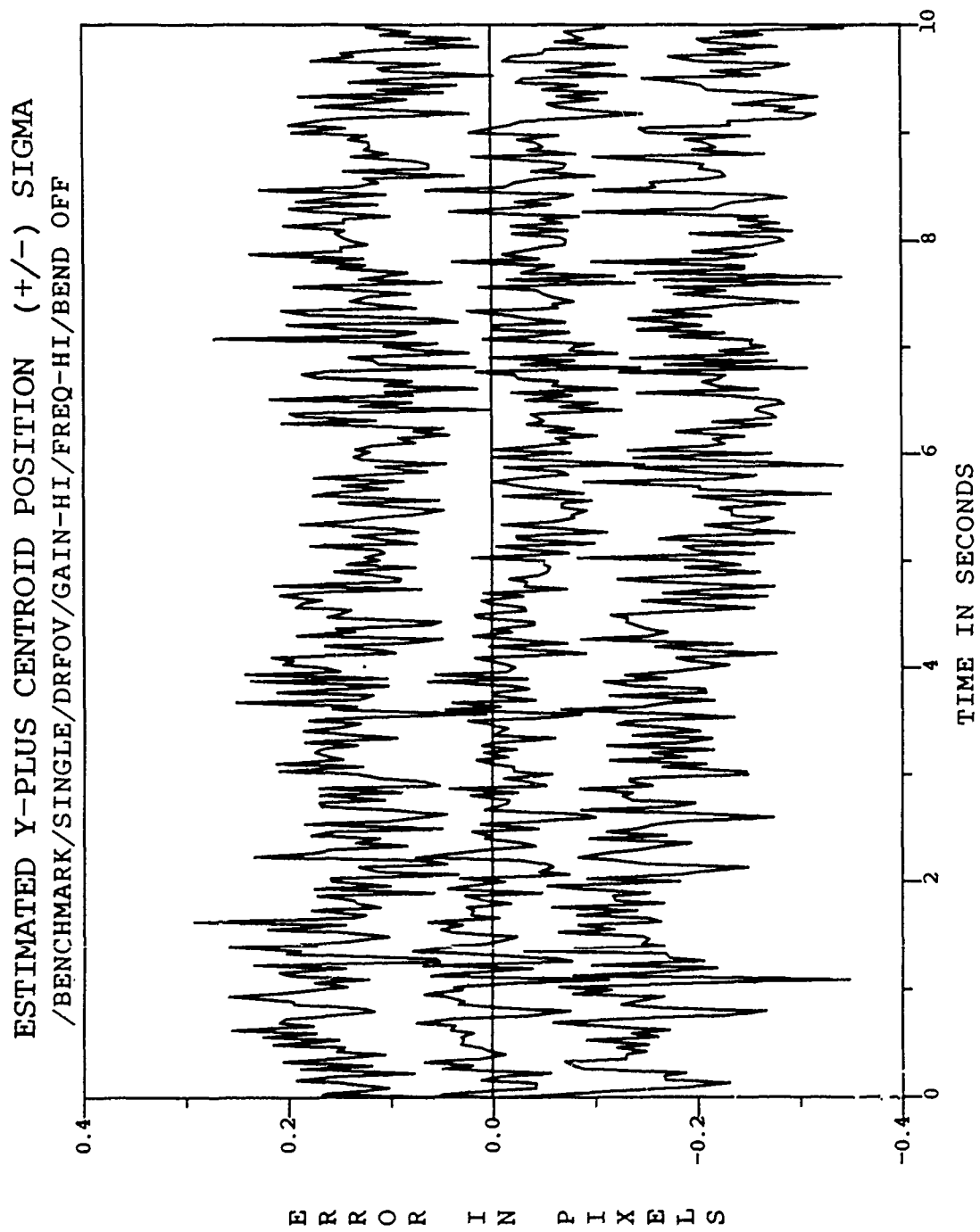


Figure F.10. /BENCHMARK/SINGLE/DRFOV/GAIN-III/FREQ-III/

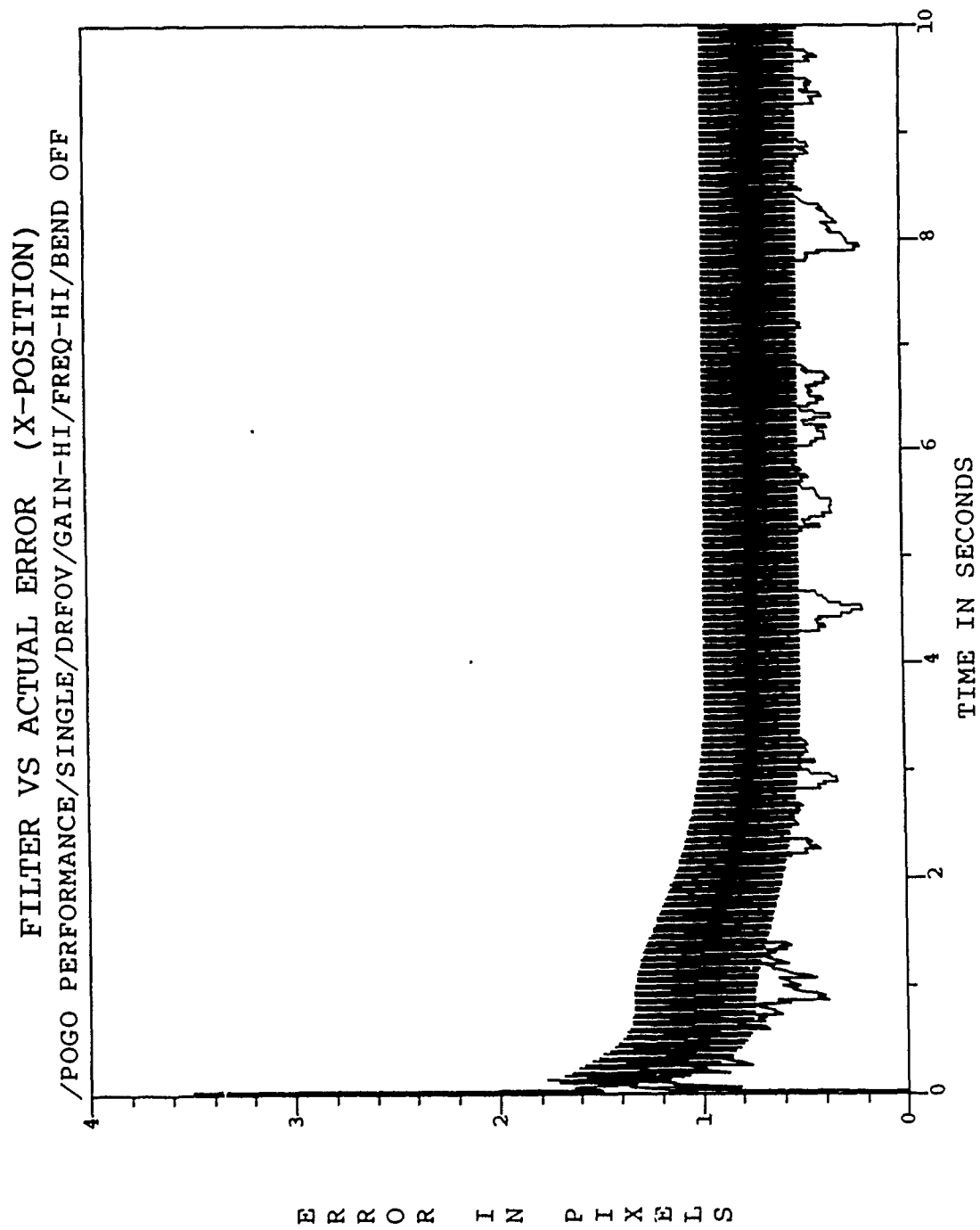


Figure F.11. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

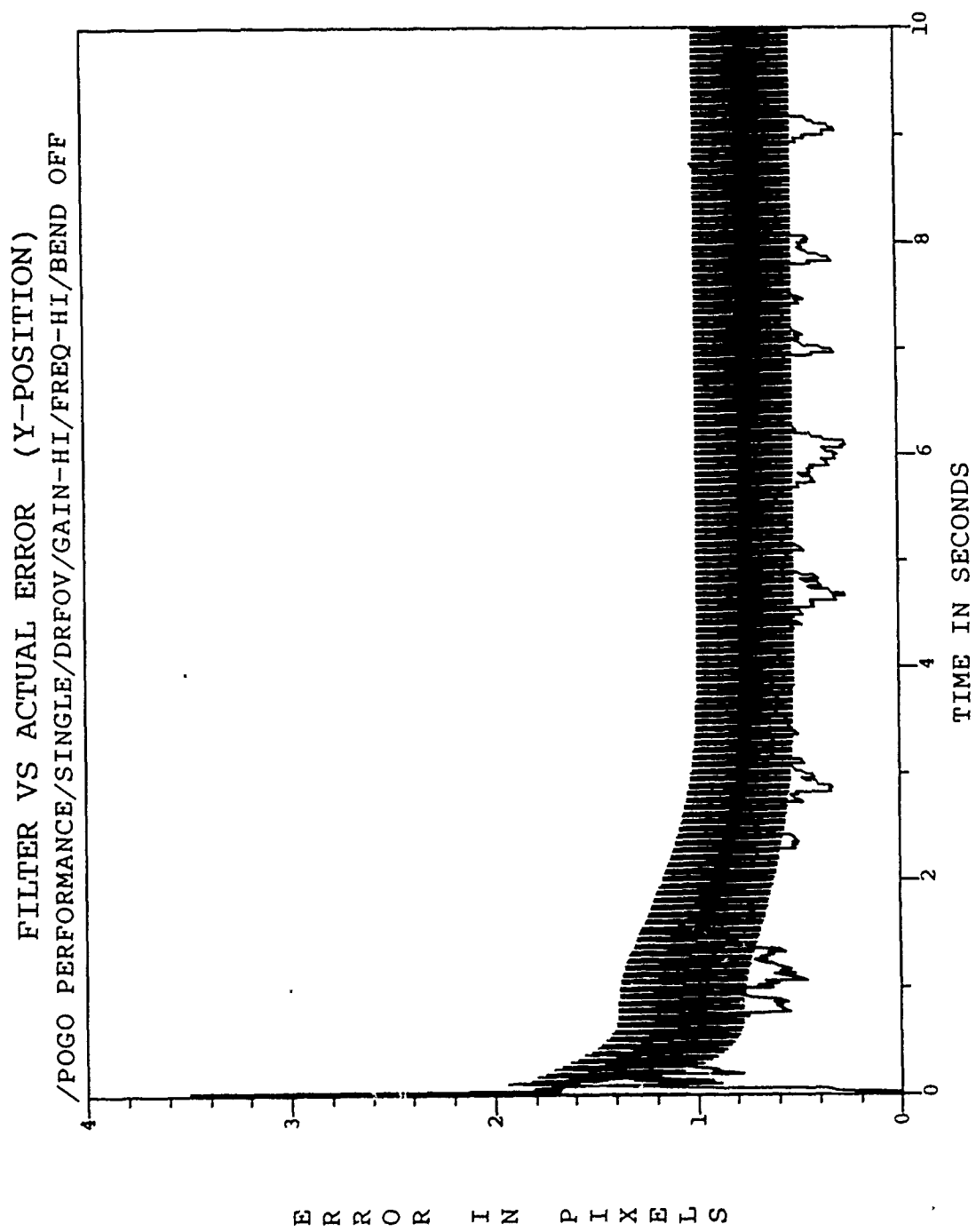


Figure F.12. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

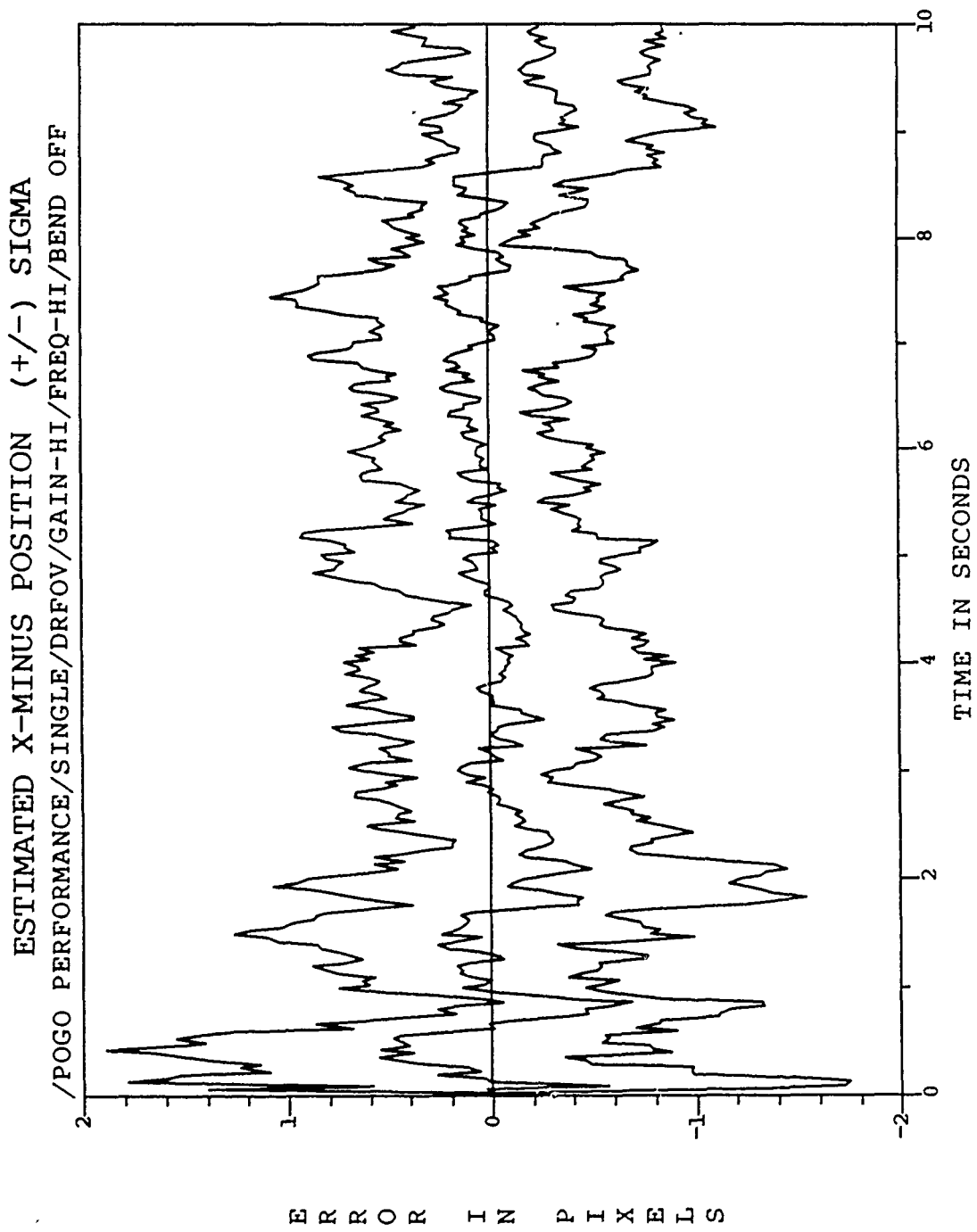


Figure F.13. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-III/

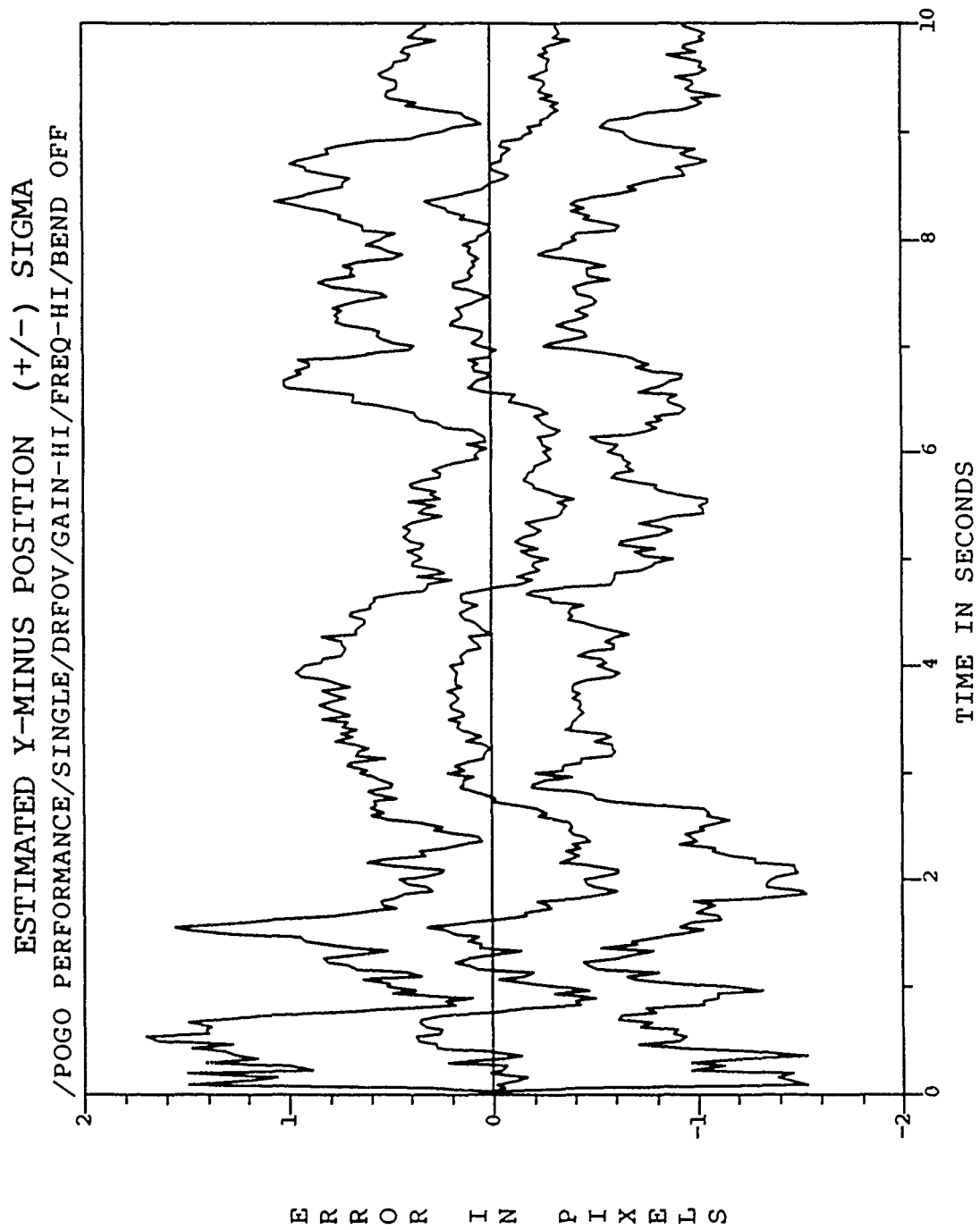


Figure F.14. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/BEND OFF

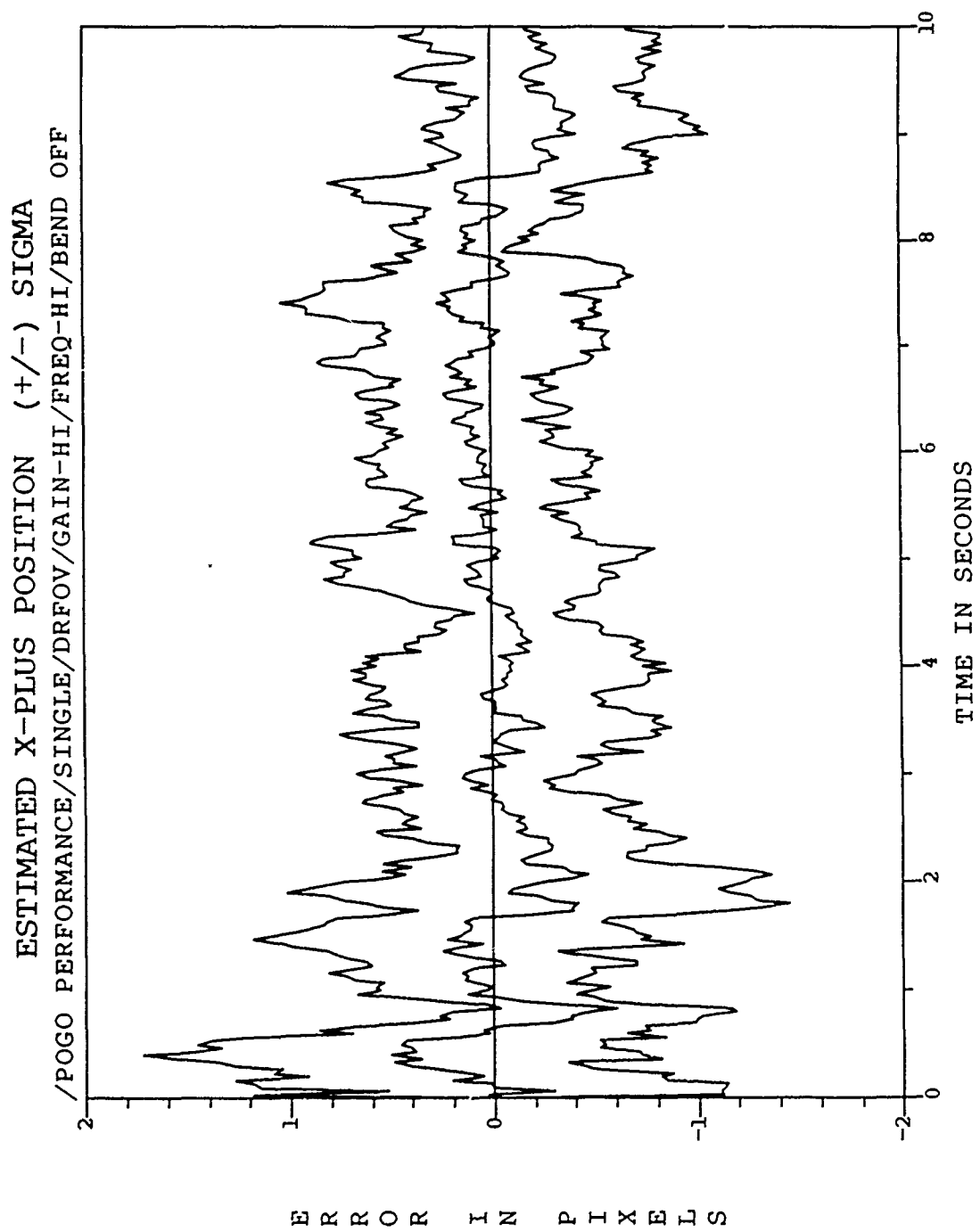


Figure F.15. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

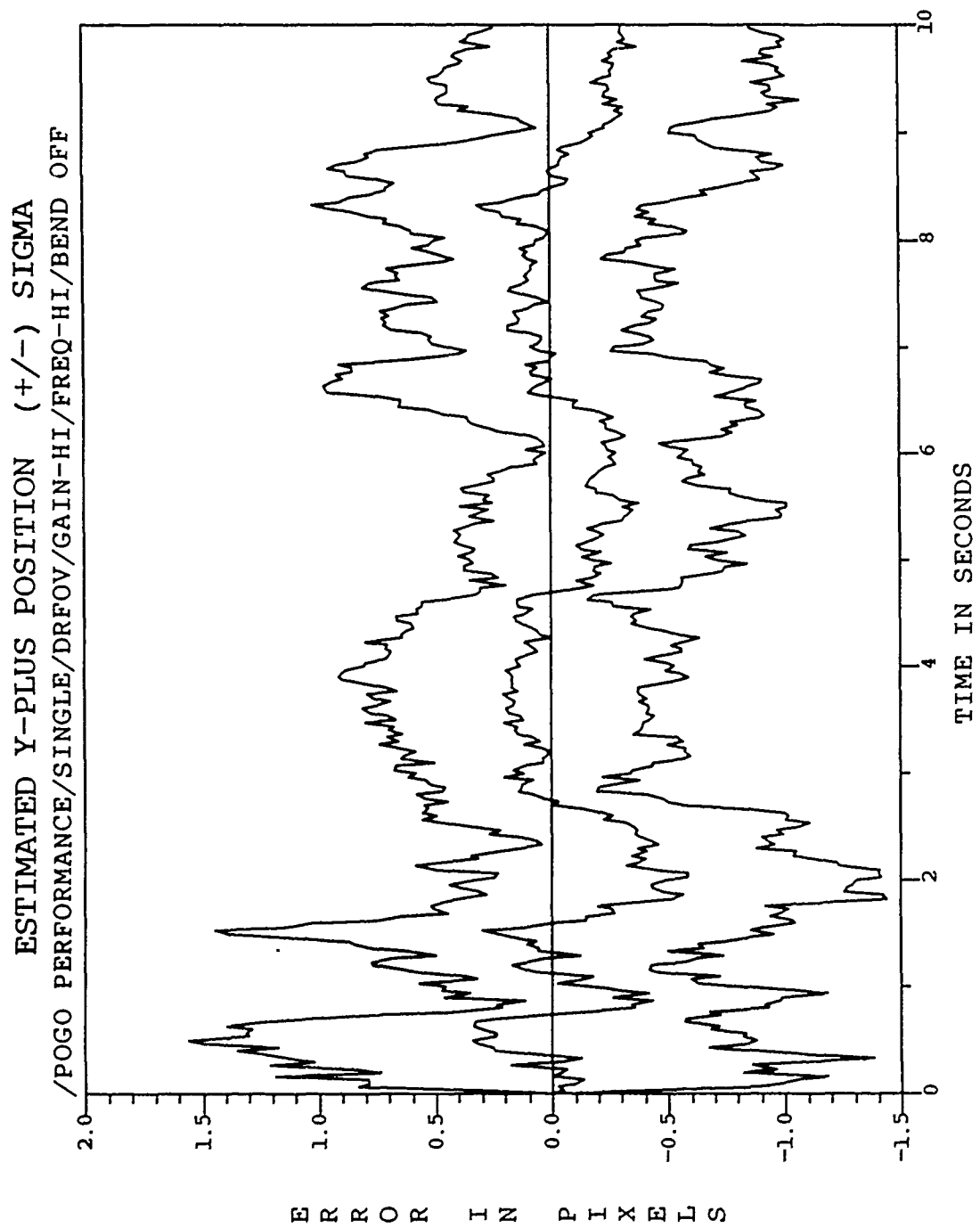


Figure F.16. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

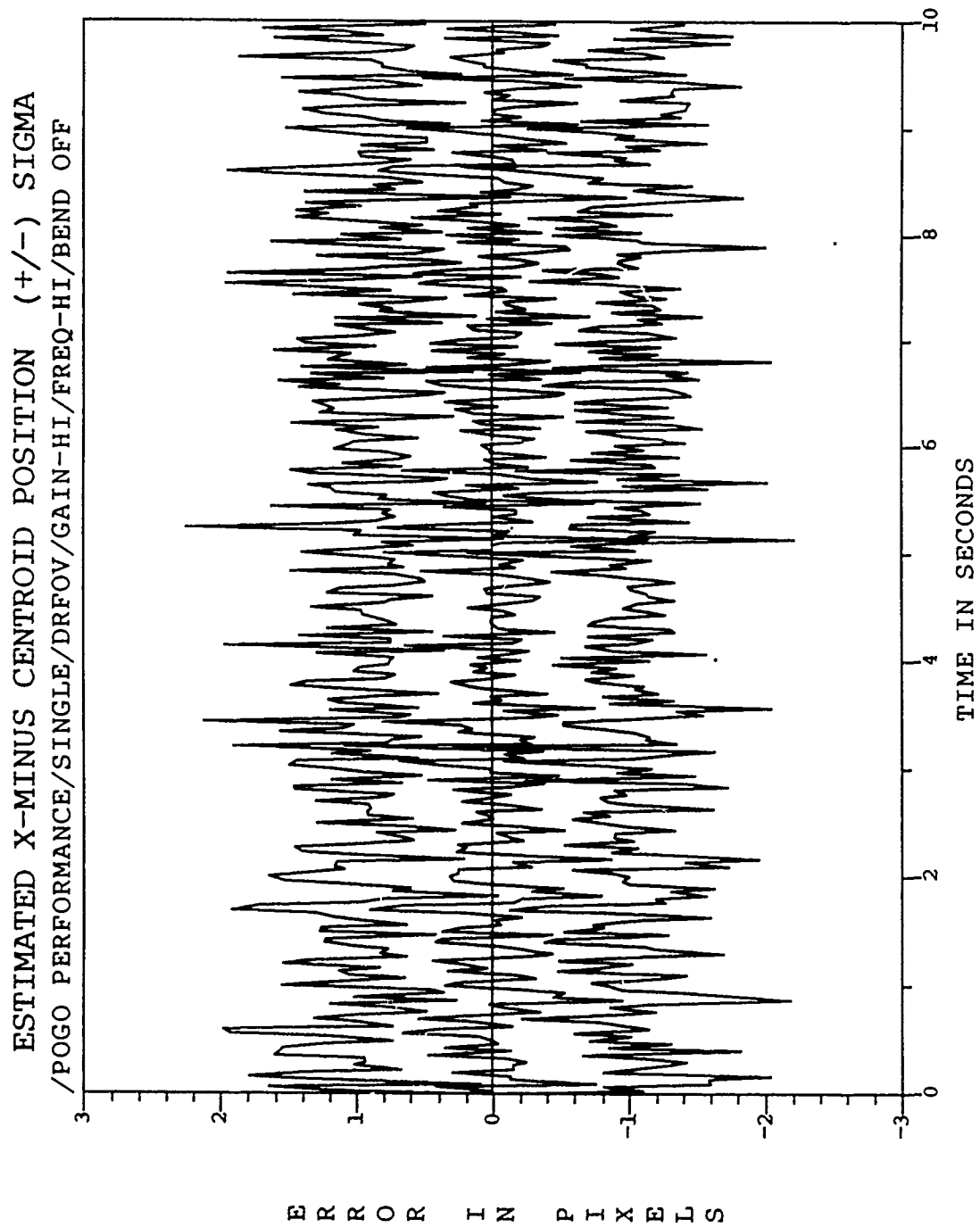


Figure F.17. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

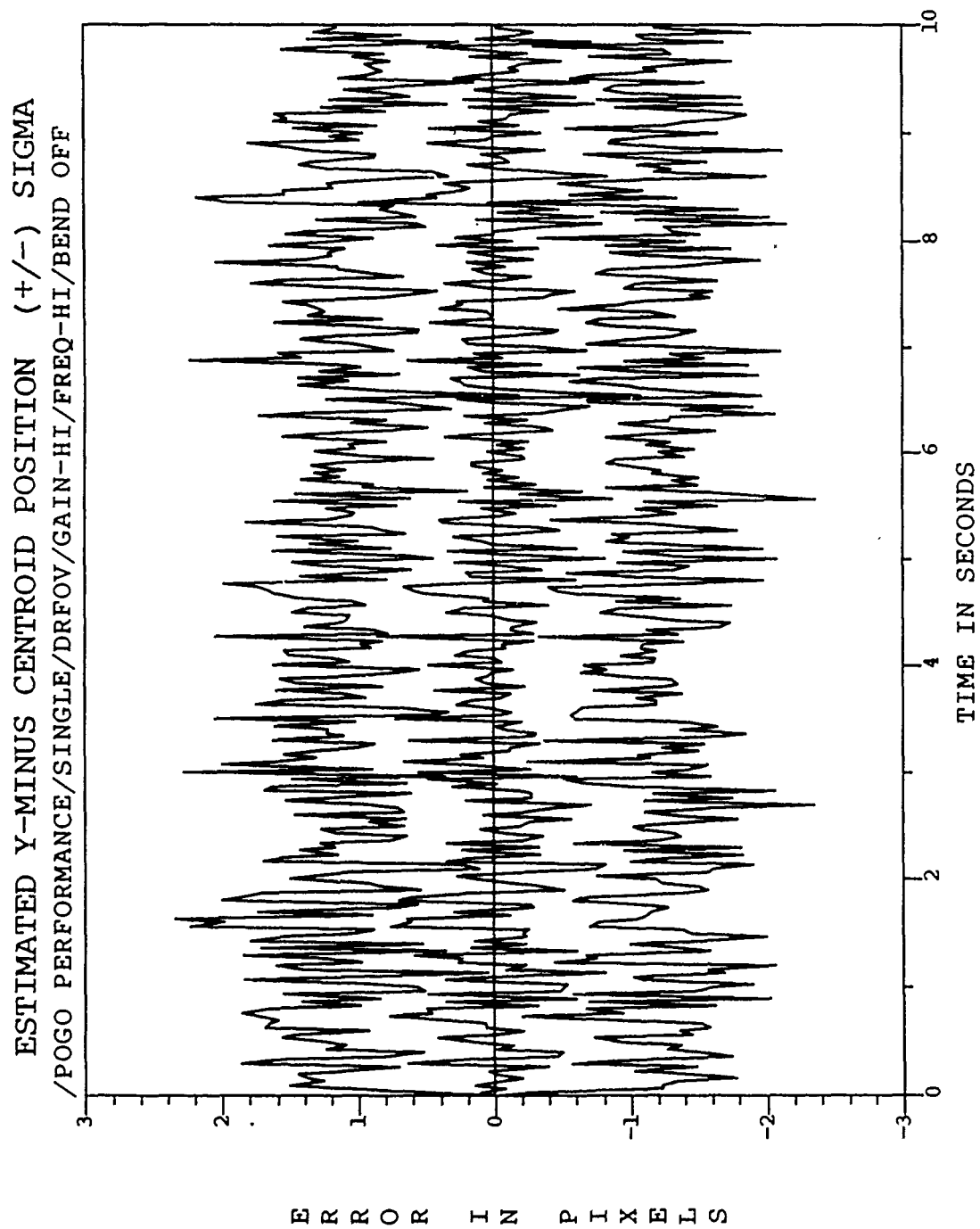


Figure F.18. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-III/FREQ-III/

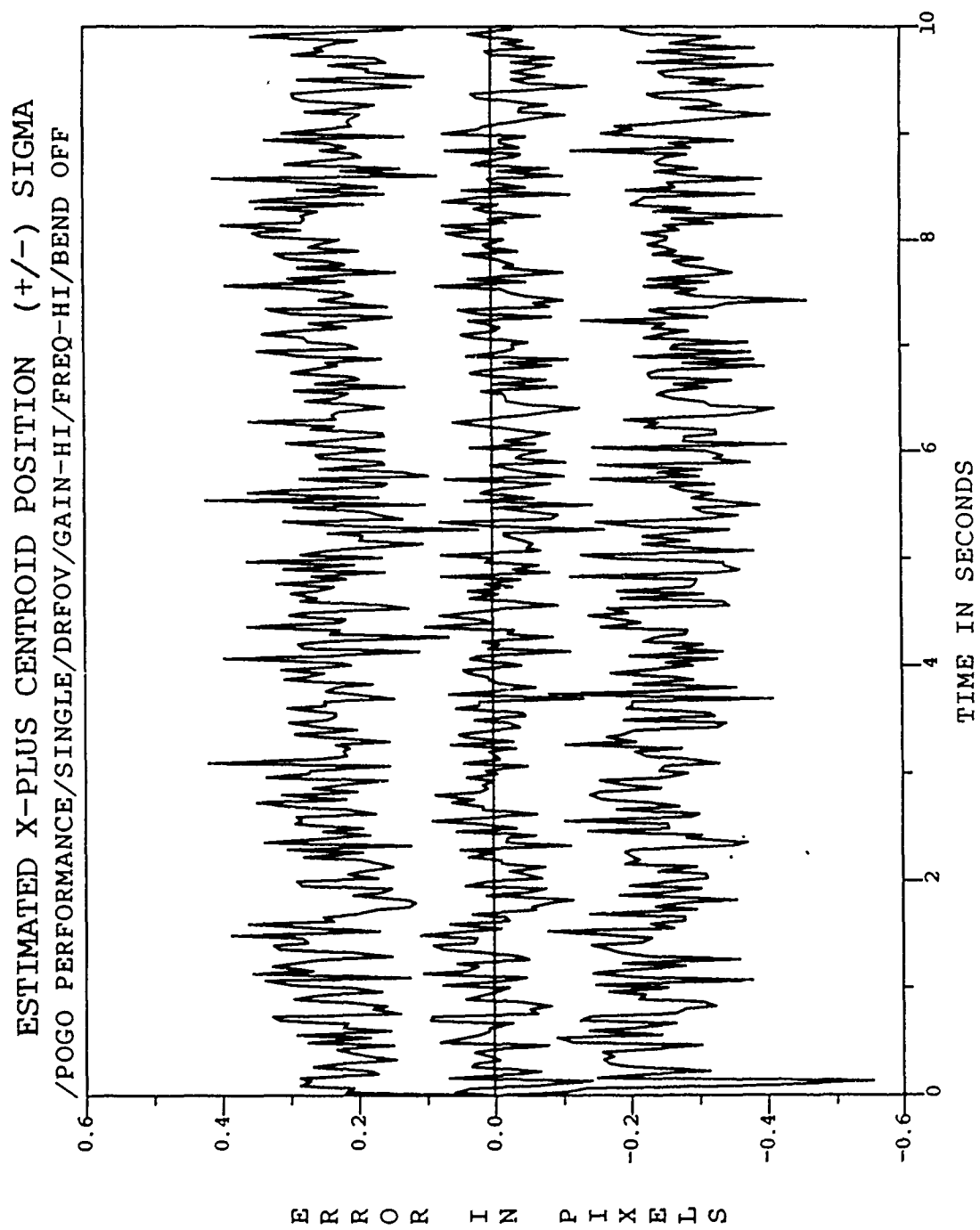


Figure F.19. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

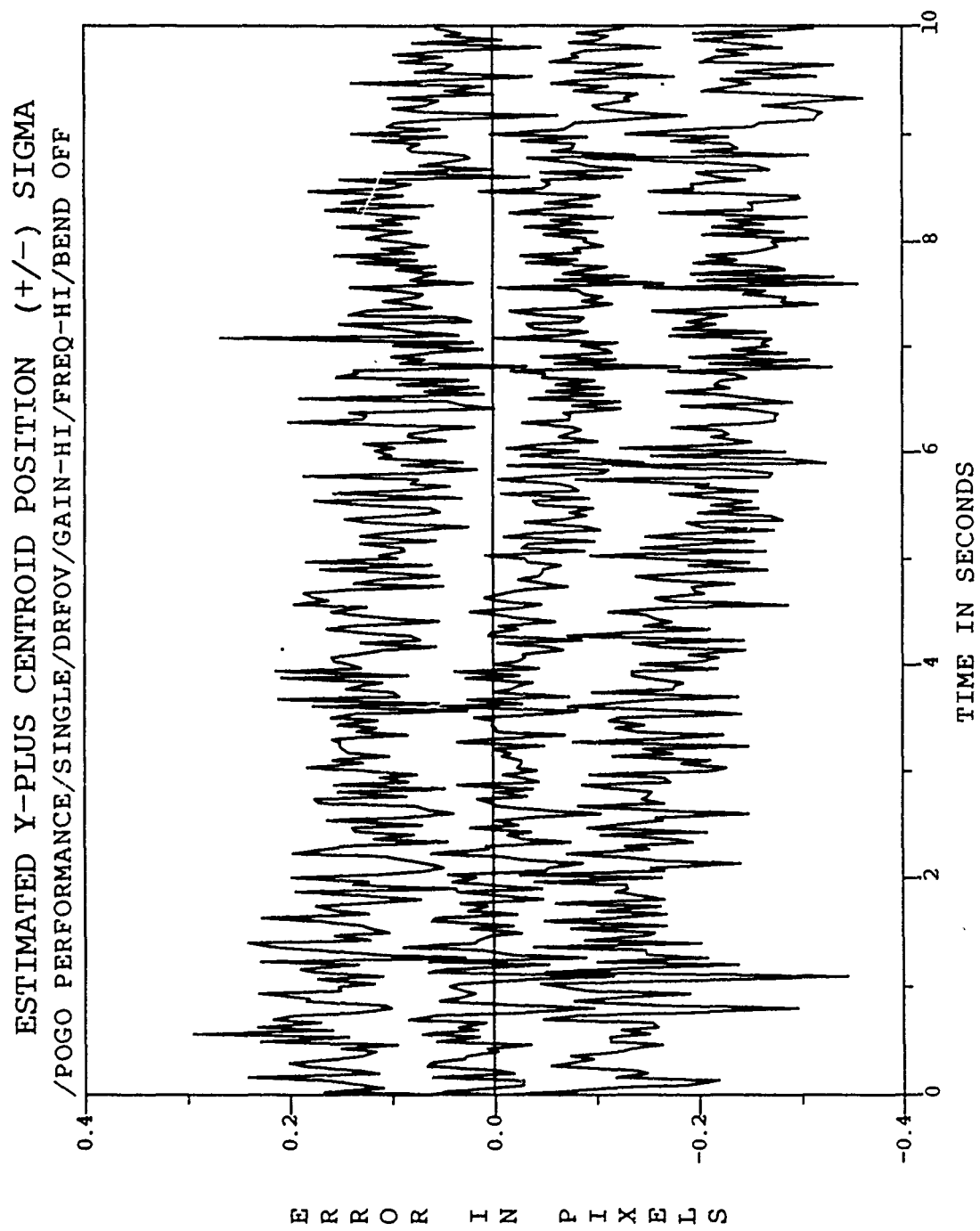


Figure F.20. /POGO PERFORMANCE/SINGLE/DRFOV/GAIN-HI/FREQ-HI/

Appendix G. *Tuning Plots Used in Jitter Sign Test: Discussion in*
Section Section 6.9.2

G.1 *Figures G.1–G.2: Filter Jitter Hardcoded to Equal Truth Model Jitter; $\sigma_a^2 = .7$*

G.2 *Figures G.3–G.4: Filter Jitter Hardcoded to Equal Truth Model Jitter; $\sigma_a^2 = .2$*

G.3 *Figures G.5–G.6: Nominal Filter and Truth Model Jitter; $\sigma_a^2 = .7$*

G.4 *Figures G.7–G.8: Nominal Filter and Truth Model Jitter; $\sigma_a^2 = .2$*

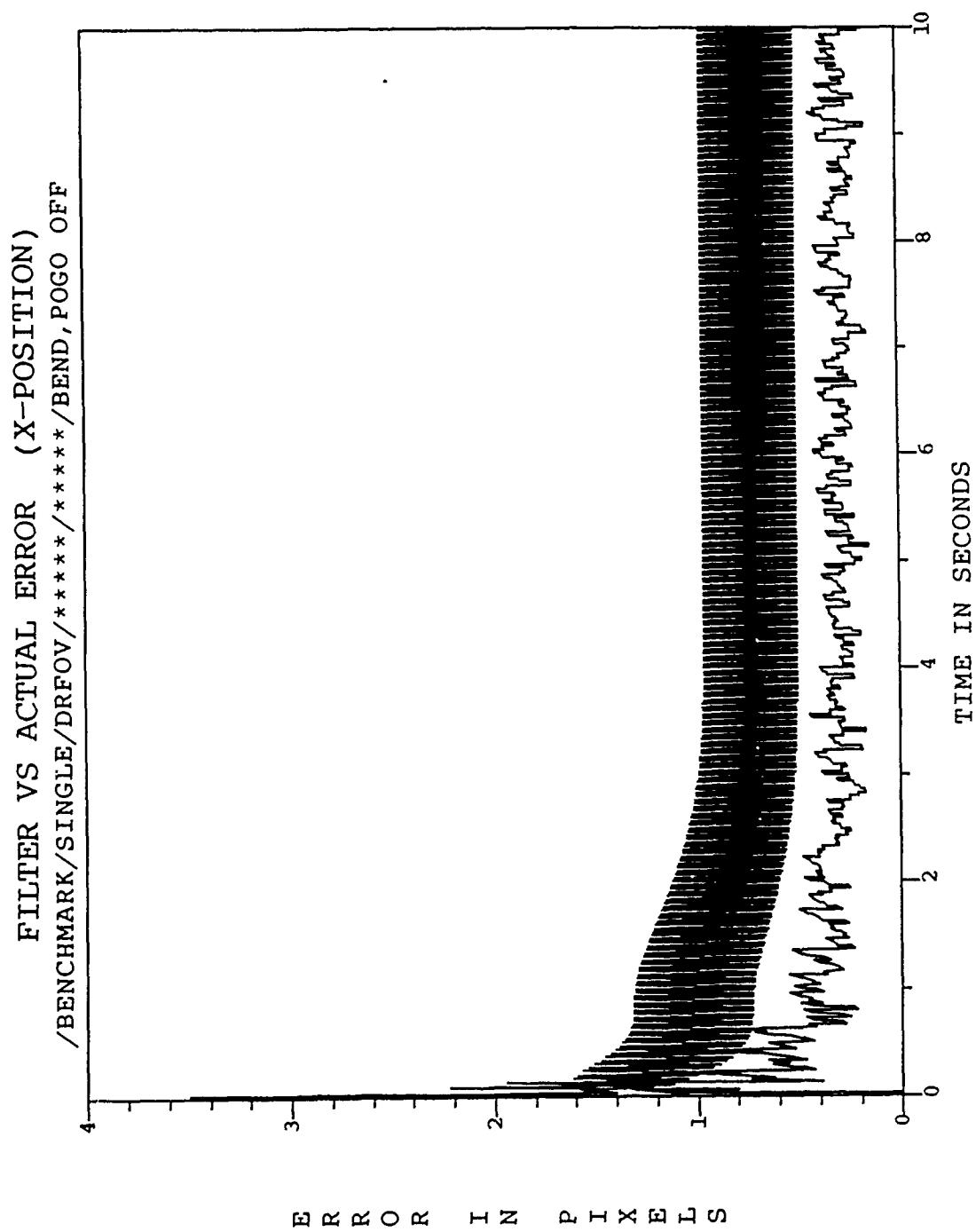


Figure G.1. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO.BEND OFF/

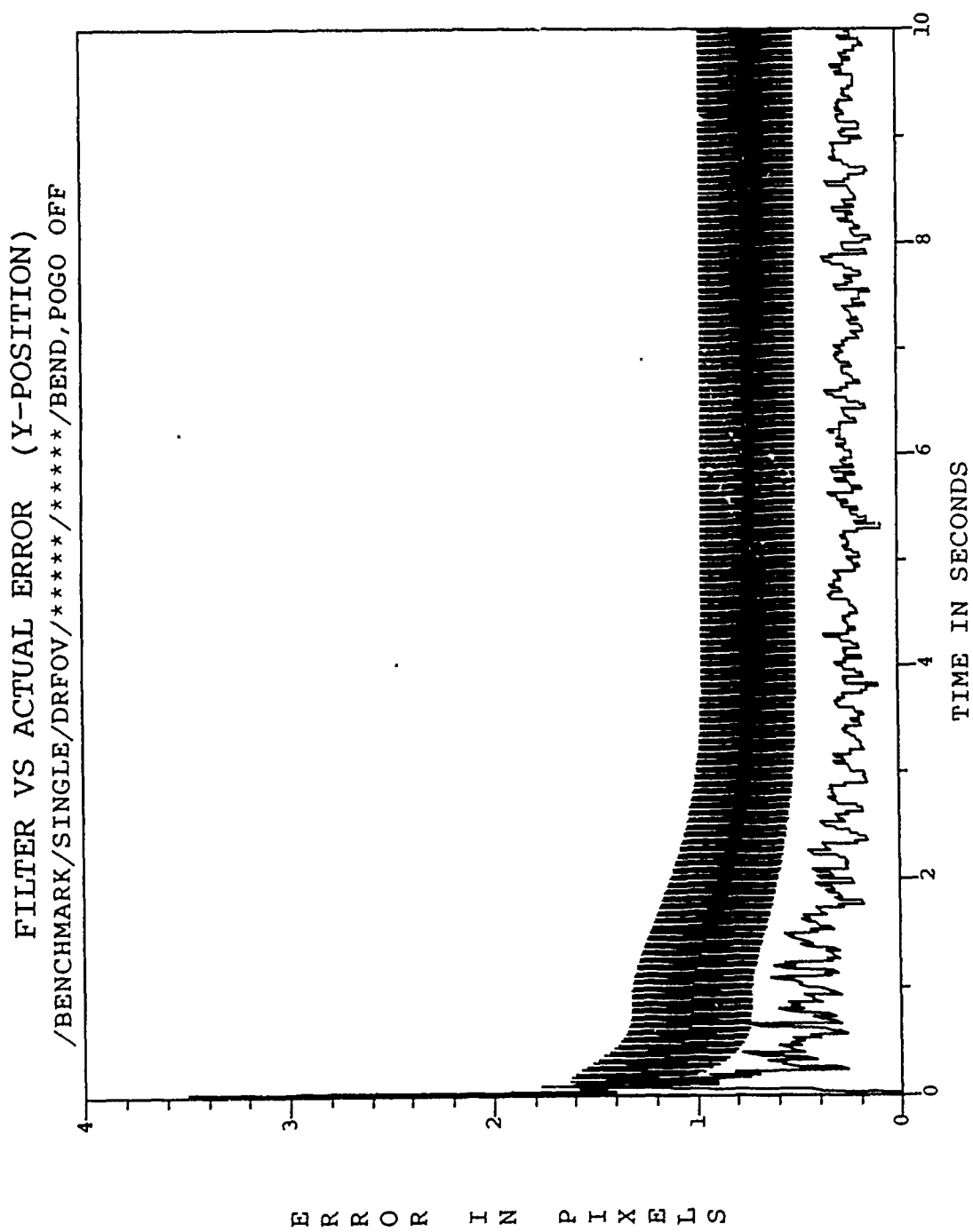


Figure G.2. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO.BEND OFF/

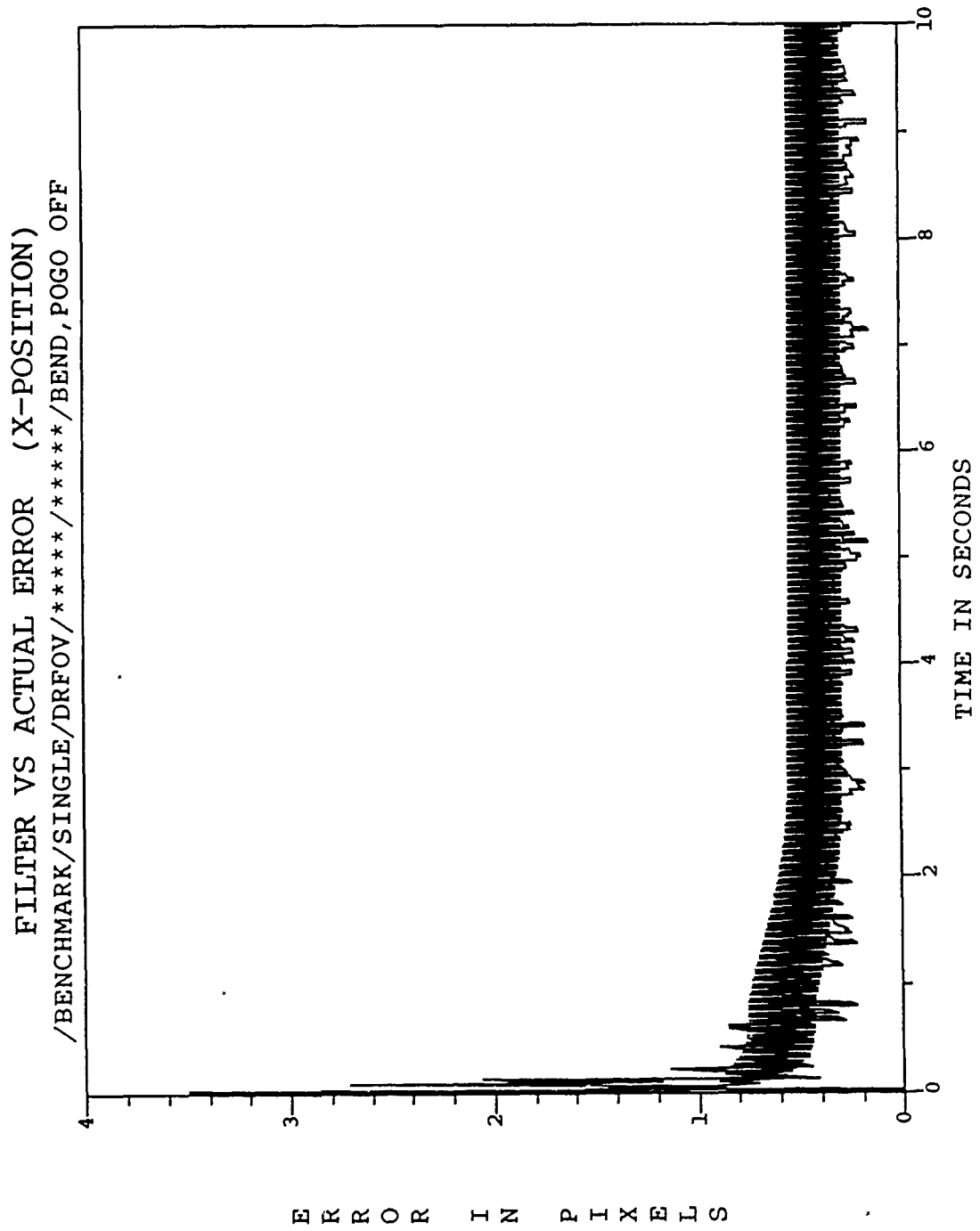


Figure G.3. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO,BEND OFF/

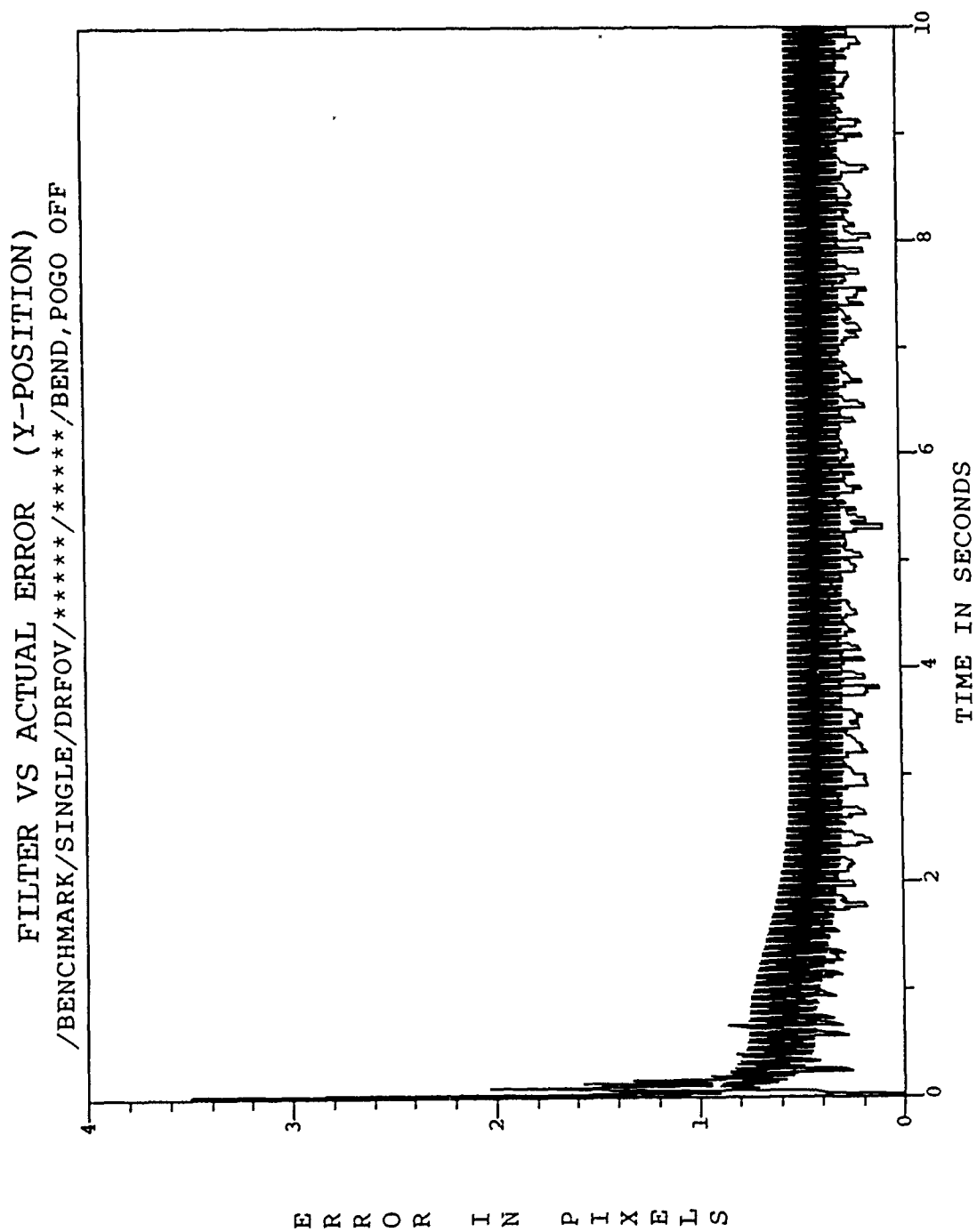


Figure G.4. /BENCHMARK/SINGLE/DRFOV/****/****/POGO.BEND OFF/

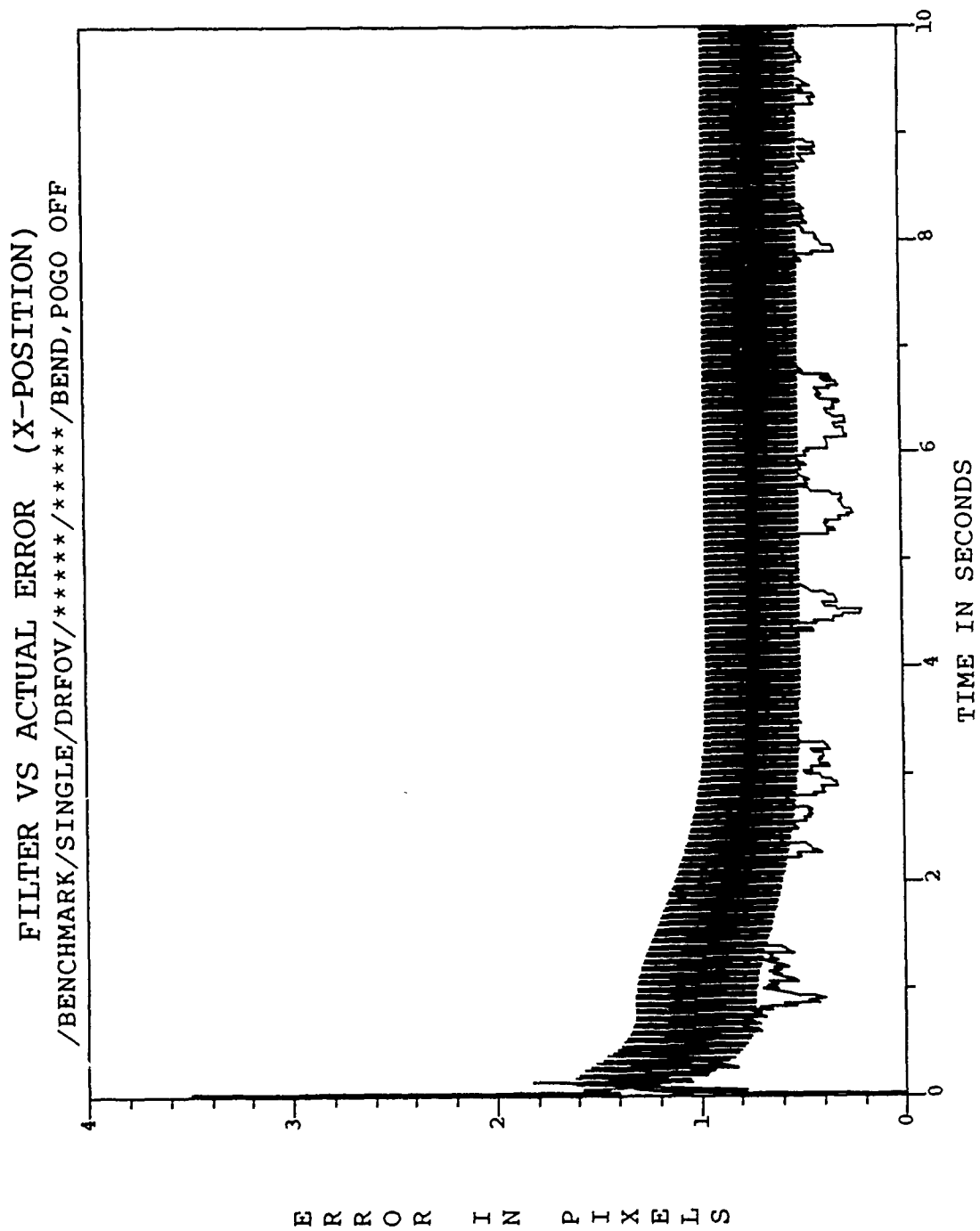


Figure G.5. /BENCHMARK/SINGLE/DRFOV/****/****/POGO,BEND OFF/

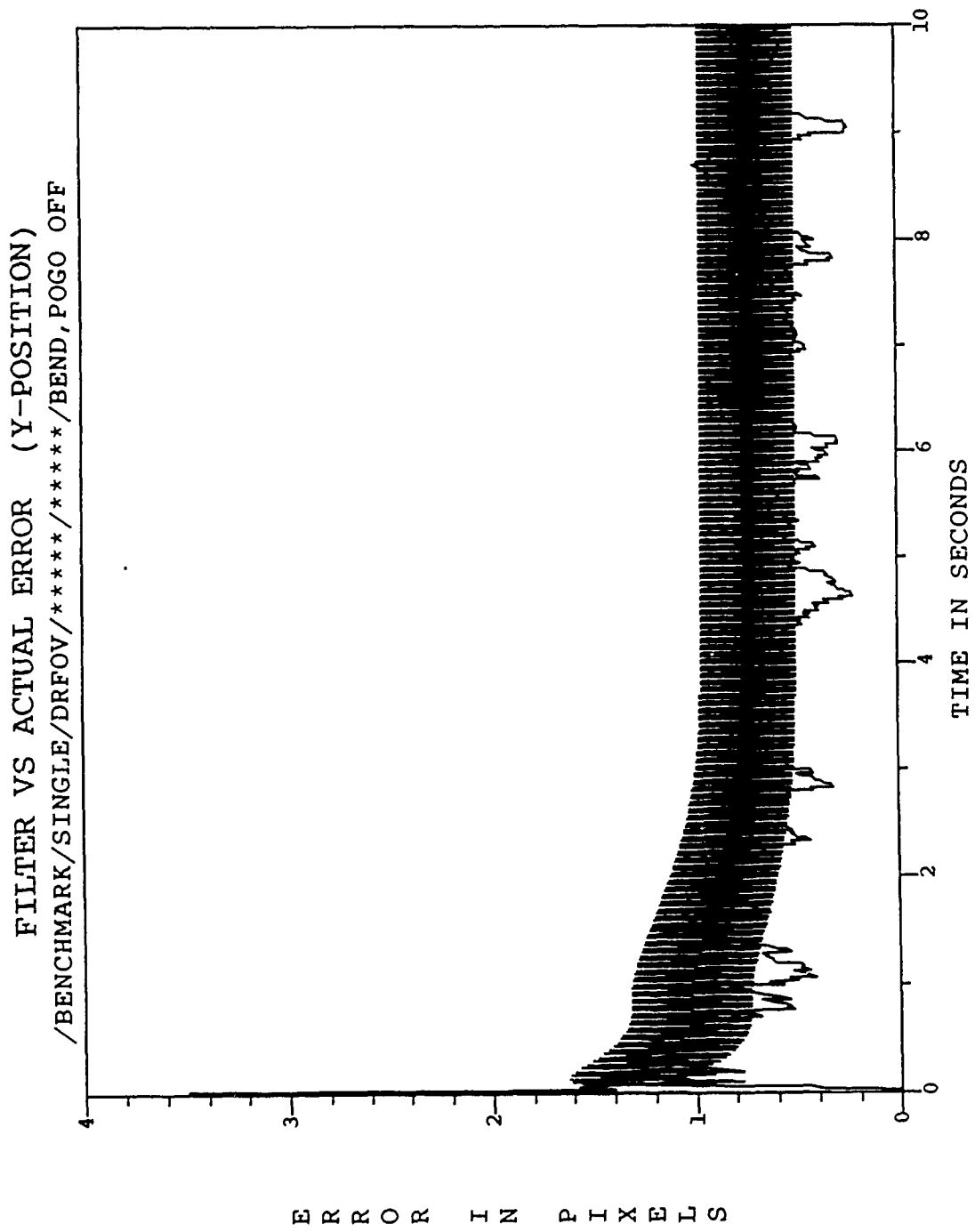


Figure G.6. /BENCHMARK/SINGLE/DRFOV/***/***/POGO,BEND OFF/

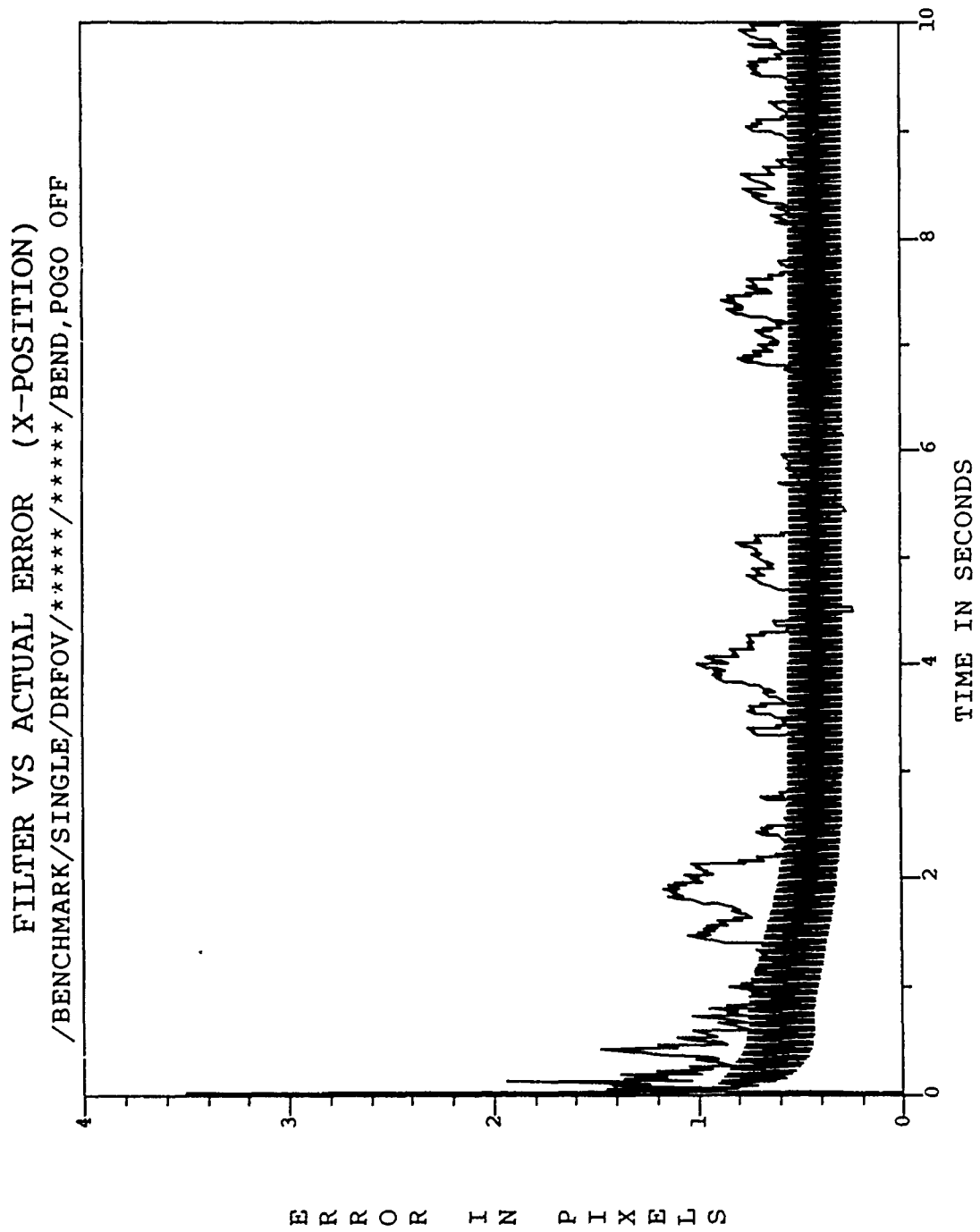


Figure G.7. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO,BEND OFF/

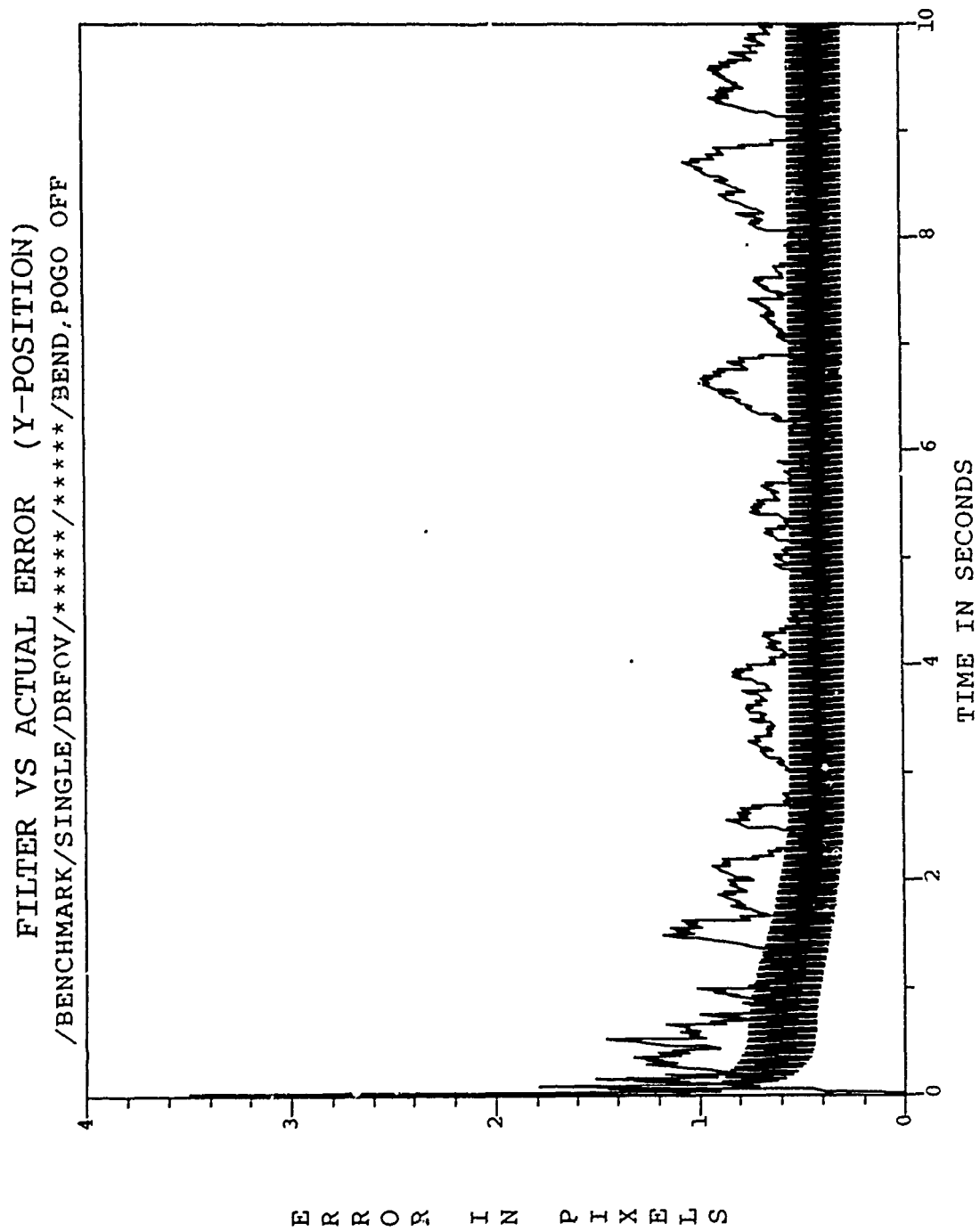


Figure G.8. /BENCHMARK/SINGLE/DRFOV/*****/*****/POGO,BEND OFF/

Bibliography

1. Air University. *Space Handbook*. AU-10. Maxwell Air Force Base, AL: Air University Press, January 1985.
2. Flynn, P. M. *Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker*. MS thesis, AFIT/GE/ENG/81D. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981.
3. Harnly, Douglas A., and Robert L. Jensen. *An Adaptive Distributed-Measurement Extended Kalman Filter for a Short Range Tracker*. MS thesis, AFIT/GA/EE/79-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1979.
4. Kozemchak, Mark R. *Enhanced Image Tracking: Analysis of Two Acceleration Models in Tracking Multiple Hot-Spot Images*. MS thesis, AFIT/GE/ENG/82D. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1982.
5. Leeney, T. A. *A Multiple Model Adaptive Tracking Algorithm Against Airborne Targets*. MS thesis, AFIT/GE/ENG/87D-37. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.
6. Loving, Phyllis A. *Bayesian vs. MAP Multiple Model Adaptive Estimation for Field of View Expansion in Tracking Airborne Targets*. MS thesis, AFIT/GE/ENG/85M. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1985.
7. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume I*. New York: Academic Press, 1979.
8. - - - -. *Stochastic Models, Estimation, and Control, Volume II*. New York: Academic Press, 1982.
9. - - - -. "Adaptive Tracking of Dynamic Airborne Vehicles Based on (FLIR) Image Plane Intensity Data," *Stochastic Differential Systems*, Proceedings of the 3rd Bad Honnef Conference, 284-304, Springer-Verlag, Berlin, 1985.
10. - - - -. Professor of Electrical Engineering. Personal Interviews. Air Force Institute of Technology (AU), School of Engineering, Wright-Patterson AFB OH, June 1988 through March 1989.
11. - - - -, D. A. Harnly and R. L. Jensen. "An Adaptive Extended Kalman Filter for Target Image Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-17, No. 2, 173-180, March 1981.
12. - - - - and D. E. Mercier. "A Target Tracker Using Spatially Distributed Infrared Measurements," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 2, 222-225, April 1980.

13. - - - - and S. K. Rogers. "Adaptive Tracking of Multiple Hot-Spot Target IR Images," *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 10, 937-943, October 1983.
14. - - - - and R. I. Suizu. "Adaptive Tracker Field of View Variation Via Multiple Model Filtering," *IEEE Transactions of Aerospace and Electronic Systems*, Vol. AES-21, No. 4, 529-539, July 1985.
15. - - - - , W. H. Worsley, and P. M. Flynn. "Investigation of Constant Turn-Rate Dynamics Models in Filters for Airborne Vehicle Tracking," *Proceedings of the IEEE National Aerospace and Electronics Conference*, Dayton, OH, 896-903, May 1982.
16. Mercier, D. E. *An Extended Kalman Filter for Use in a Shared Aperture Medium Range Tracker*. MS thesis, AFIT/GA/EE/78D-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1978.
17. Millner, P. P. *Enhanced Tracking of Airborne Targets Using a Correlator/Kalman Filter*. MS thesis, AFIT/GE/ENG/82M. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1982.
18. Netzer, Allan S. *Characteristics of Bayesian Multiple Model Adaptive Estimation for Tracking Airborne Targets*. MS thesis, AFIT/GAE/ENG/85D-2. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.
19. Norton, John E. *Multiple Model Adaptive Tracking of Airborne Targets*. MS thesis, AFIT/GA/ENG/88D-2. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.
20. Richards, C. L. *Correlation Tracking Algorithm*, SAMRT-76-0087, Engineering Data Release, Aeronutronic Ford Corporation, Newport Beach CA, 2 July 1976.
21. Rogers, Steven K. *Advanced Tracking of Airborne Targets Using Forward Looking Infrared Measurements*. MS thesis, AFIT/GE/ENG/81D. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981.
22. Singletery, J. *Adaptive Laser Pointing and Tracking Problem*. MS thesis, AFIT/GEO/EE/80D-12. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980.
23. Suizu, Robert I. *Advanced Tracking of Airborne Targets Using Multiple Model Adaptive Filtering Techniques for Adaptive Field of View Expansion*. MS thesis, AFIT/GE/ENG/83D. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1983.

24. Tobin, David M. *A Multiple Model Adaptive Tracking Algorithm for a High Energy Laser Weapon System*. MS thesis, AFIT/GE/ENG/86D-37. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.
25. - - - - and P. S. Maybeck. "Enhancements to a Multiple Model Adaptive Estimator/Image-Tracker," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, No. 4, 417 - 426, July 1988.

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			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/89M-7			7a. NAME OF MONITORING ORGANIZATION		
6a. NAME OF PERFORMING ORGANIZATION School of Engineering		6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7b. ADDRESS (City, State, and ZIP Code)		
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright Patterson AFB, Ohio 45433-6583			9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Weapons Laboratory		8b. OFFICE SYMBOL (If applicable)	10. SOURCE OF FUNDING NUMBERS		
8c. ADDRESS (City, State, and ZIP Code) Kirtland AFB, NM 87117			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) ENHANCED TRACKING OF BALLISTIC TARGETS USING FORWARD LOOKING INFRARED MEASUREMENTS; UNCLASSIFIED					
12. PERSONAL AUTHOR(S) Capt David R. Rizzo					
13a. TYPE OF REPORT MS Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1989 March	15. PAGE COUNT 362
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
17	5		Kalman filter, Plume, Pogo, Tracking, Ballistic Missile, FLIR, Laser, Multiple Models		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Thesis Chairman: Dr. Peter S. Maybeck Abstract: (on back of this form)					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Peter S. Maybeck			22b. TELEPHONE (Include Area Code) 513-255-3576		22c. OFFICE SYMBOL AFIT/ENG

This thesis is an extension of the work performed over the past ten years at the Air Force Institute of Technology (AFIT) towards tracking of airborne targets using forward looking infrared (FLIR) measurements. The research has aimed at replacing a standard correlation tracker with a hybrid Kalman filter/enhanced correlation tracker for implementation in a high energy laser weapon.

This research deviates somewhat from past research at AFIT in that the target trajectory being tracked is modelled as a benign, non-maneuvering, thrusting ballistic missile trajectory at large sensor-to-target ranges. In addition, to capture the characteristic shape of the exhaust plume, the plume is modelled as the difference between two bivariate Gaussian functions with elliptical equal intensity contours. As the missile ascends on its thrusting trajectory, the exhaust plume tends to oscillate (pogo) along the direction of the velocity vector. In this thesis, a second-order Gauss-Markov process is used to model the plume's "pogo" oscillation properties.

The ultimate goal of this research effort is to design a multiple model adaptive filter (MMAF) algorithm composed of elemental filters tuned for varying plume pogo parameters (frequency and amplitude characteristics). This MMAF accounts for atmospheric disturbance effects of the propagating infrared wave fronts, as well as bending/vibrational effects of the optical hardware associated with the FLIR sensor. The bank of filters provide the accurate estimation capability to guide the pointing mechanism of a shared aperture laser/FLIR sensor.

An 8×8 -pixel tracking field of view (FOV) of the FLIR sensor provides the infrared data to the enhanced correlation tracking algorithm. To enhance performance of the tracking algorithm, a FOV rotation scheme is analyzed in an effort to maintain accurate tracking of a plume undergoing the pogo phenomenon. A FLIR rotation scheme which aligns the diagonal dimension of the 8×8 -pixel tracking window with the missile velocity vector demonstrates a 50% performance improvement over a non-rotating FOV FLIR.

A benchmark of performance involving an eight-state Kalman filter is established in order to compare results from various tracking enhancement techniques. The eight-state filter excludes explicit modelling of the pogo phenomenon, but the pogo effect is compensated by the addition of pseudo-noise in the filter model. To implement the MMAF, a ten-state filter which models the additional two pogo states is analyzed, and results are compared to the eight-state filter benchmark for performance enhancement. The ten-state filter consistently showed an unexpected performance degradation compared to the eight-state filter. Various trouble-shooting techniques are employed to uncover the source(s) of this degradation. Possible problems include: (1) a pogo-atmospheric jitter interaction, (2) poor estimation by the Kalman filter atmospheric jitter model and (3) observability issues of the target dynamics model. Recommendations to overcome these shortcomings are proposed in order to enhance performance of the ten-state filter and eventually implement the MMAF algorithm.