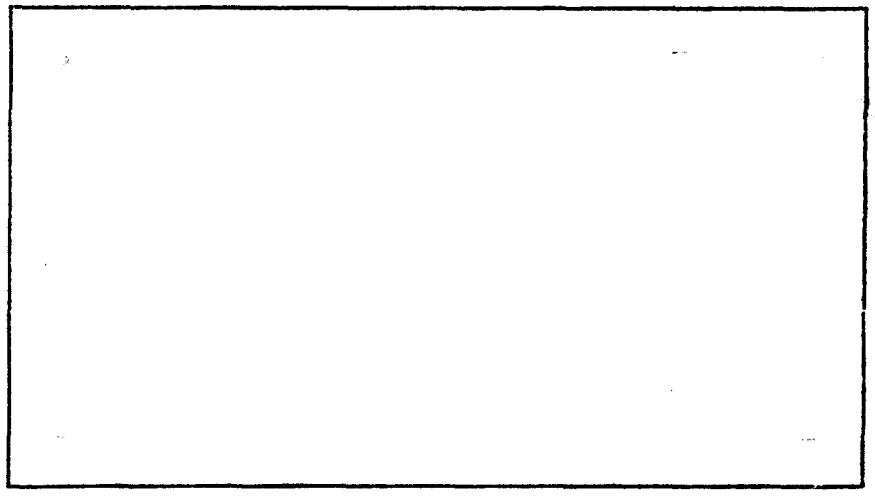
AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE



SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

(1)

(6)

COOPERATIVE ESTIMATION OF TARGETS BY MULTIPLE AIRCRAFT.

(9) [illegible] THESIS

(14) AFIT/GE/EE/80-5

(10) Ralph S. Bryan

(11) Dec 80

(12) 331

A

Ø12225

AFIT/GE/EE/80-5

COOPERATIVE ESTIMATION OF TARGETS BY MULTIPLE AIRCRAFT

THESIS

by

Ralph S. Bryan, B.S. Engineering Physics

Graduate Electrical Engineering

June 1980

Presented to the Faculty of the School of Engineering

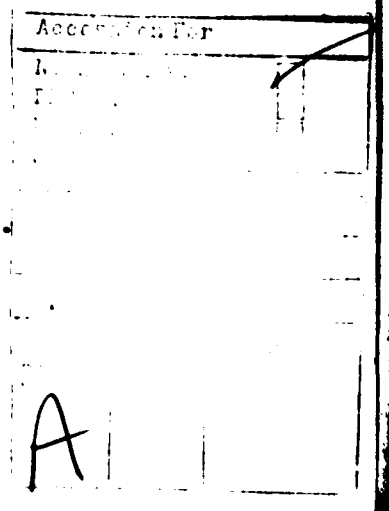
of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science



Acknowledgments

The cooperation and help of many people made this thesis effort possible. Dr. Peter Maybeck provided me invaluable guidance, and his attention to detail never failed. Major Gary Reid and Capt James Silverthorn similarly provided helpful comments on the thesis. Mr. Stan Musick, of the Reference Systems Branch of the Avionics Laboratory helped me numerous times in incorporating my software into his general-purpose program. More importantly, the software he provided was in good working order when delivered. My group chief, Mr. Arthur Duke, provided the most necessary element, sufficient time in which to finish the project, and Debi Walters did an excellent and prompt job of typing. Finally, for putting up with late hours and little attention, my wife Bevie and son Joshua deserve a special thanks.

Contents

Contents (Cont'd)

List of Figures

List of Tables

Abstract

A cooperative estimator of aerial target motion was designed and evaluated using an Extended Kalman Filter and a Monte-Carlo simulation. The estimator was designed to combine tracking information from two cooperating fighter aircraft which are tracking a single aerial target in either range and angle, or in angle-only. The cooperation occurs when information, in the form of raw measurements or state estimates, is transferred between aircraft via a telemetry link. The design and evaluation were both preliminary in the sense that necessary information about attackers' relative position and orientation was assumed error-free, that transmission delays and other details of a data link were not included, tracker errors were modeled as white noise, and timing and word-length constraints of actual real-time filter software was not addressed.

Two methods of data transfer were developed and simulated. These were Direct Measurement Transfer and State Estimate Transfer. An attempt was made to develop an optimal algorithm for combining state estimates directly, but no efficient algorithm was found, so that state estimates were utilized as pseudo-measurements in a heuristic application of the extended Kalman Filter. The results of the Monte-Carlo simulation of Direct Measurement Transfer using two different target trajectories, indicated up to 30% performance improvement for cooperative estimation as compared to single filter estimation when range and angle measurements were present. When only angle measurements were available, a single filter was unable to estimate the complete target state, but the cooperative estimator was able to do so, converging to within 1% of

true range within two seconds on the trajectories simulated. State Estimate Transfer, which was simulated for the angle-only case, proved successful only when the transfer rate was approximately 1.0 Hz.

I. Introduction

Problem

Even though fighter aircraft avionics are becoming increasingly sophisticated, individual elements of a tactical fighter aircraft formation operate on an autonomous basis from the standpoint of real-time data transfer. Once the flight is airborne, coordination is done through verbal communication, so that data transfer is necessarily at a very low rate and of limited content. This imposes several constraints on the operations. First, raw data must be processed onboard the same aircraft as obtained it. Secondly, except for verbal transfer of data (e.g. "you have a target at 3 o'clock"), the processed data can only be used aboard the original aircraft in real time. As a direct result, no raw or processed data from another friendly aircraft is available. The lack of data results in either reduced mission effectiveness, or in increased avionics requirements to supply all the aircraft with the sensors and associated processing needed to perform a difficult task.

This lack of data transfer is especially disadvantageous to a numerically outnumbered force, and is not even necessarily compensated by having sophisticated, yet autonomous aircraft. The ACEVAL flight test program (Ref 1:606) demonstrated that qualitative superiority does not necessarily prevail over a quantitative advantage. The cooperation in tactical maneuvers to defeat an enemy aircraft is an example of the synergistic effect of coordinating resources for maximum effect. An example of the disadvantage of one aircraft having to perform all functions is the defensive disadvantage that an aircraft has while illuminating a target for his semi-active missile, while he is simultaneously

closing within the target's missile launch envelope. If another aircraft could illuminate from beyond missile range, using data transferred from the launching aircraft, the defensive disadvantage could be avoided.

Finally, some operations are not possible without some sort of cooperative data transfer. Present tactical radar cannot do terrain following and terrain avoidance (TF/TA) simultaneously with aerial search. Using a cooperative technique, however, it should be possible for the low aircraft of a formation to perform the TF/TA function, while another aircraft behind him does the aerial search function. The avionics systems on both aircraft would have all the information available. Another example is a pair of aircraft angle-tracking a target at long range, each with an angle-only tracker, such as passive infrared. Either aircraft alone cannot determine the range to the target because it is mathematically unobservable. Yet, if their relative position vectors and attitude are known accurately, the combined measurement set is sufficient to form an estimate of target position, velocity, and possibly acceleration. These state estimates are then useful in the fire control systems of each aircraft.

Thesis Topic

This last example of data transfer directly motivates this thesis study. Specifically, this thesis investigates the target state estimation problem when two aircraft are available to make measurements, and transfer of data between aircraft takes place in the 1-25 Hz range. For the purpose of this study, both aircraft are presumed to be configured identically, that is, both have the same sensors, computing capacity, and software program, as opposed, say, to one aircraft making all the

measurements, while the other does all the processing. Although this may not be ideal under some circumstances, it simplifies the range of the investigation, and it may also be desirable in situations where the availability of the second aircraft is intermittent or uncertain. It also should allow reconfiguring the task assignment of each fighter aircraft if one aircraft is taken out of action.

II. Analytic Development

This chapter develops the necessary equations for implementing an Extended Kalman Filter for cooperative estimation. Since a single, non-cooperative estimator, such as now used in fighter aircraft radars, is the building block for the cooperative estimators, it is developed first in the "Single Tracker Estimation" section. That section has subsections which develop the necessary dynamics, measurements, and filter equations. In the "Dual Tracker Estimation" section, dynamics and measurements are again treated to show what modifications are necessary when exchanging measurements. Finally, four possible structures for dual tracker estimation are discussed.

Single Tracker Estimation

Dynamics.

Since most of the technical literature dealing with target estimation applies to a single aircraft tracking a single target, and since that is the foundation for the developments in this thesis, it will be discussed first. Important points to be discussed in the area of filter dynamics are the estimator order and coordinate frame, the target acceleration model, and the matrices required to define the model. Later sections will deal with the measurement model, filter algorithm, and extensions necessary to perform the dual tracker estimation.

A third order, nine state estimator having inertial components of relative target position, total target velocity, and total target acceleration in each of three axes is chosen as the dynamics model. Third order is chosen because of the greater volume of literature documenting its performance, thus establishing a larger baseline, (Ref 3, 5, 8) and

the fact that position, velocity, and acceleration are needed for many fire control applications. In addition, including the acceleration state produces better velocity estimates. Higher order derivatives of target motion are not considered because no good model for target acceleration rate is known.

Two primary possibilities for the coordinate frame are line-of-sight coordinates and local level inertial coordinates. Both choices have been implemented operationally. A line-of-sight coordinate frame, usually roll-stablized to reduce dynamic coupling between azimuth and elevation axes, is the more traditional choice. This is because rate-integrating gyros on a gimbaled tracker form a stabilized physical frame in which azimuth, elevation, and range equations can be decoupled successfully. Because of the necessity of implementing the tracking loops and target filter in electronic hardware, anything more complicated than a line-of-sight filter has been considered infeasible until recently. The simplicity of the filter lies in the close coorespondence of the measurements to the estimator states, but results in more complicated dynamics equations. More importantly for this thesis, separate line-of-sight coordinate frames for two separated tracking aircraft complicates the relationship between states estimated by one aircraft and the measurements of the other, thus complicating information transfer between the two aircraft.

An inertial coordinate frame is simpler for modeling target dynamics. The angular rate terms, namely Coriolis and centripetal acceleration, which appear in differential equations written in a rotating coordinate frame are not present. From the standpoint of complication, these angular rate terms are more significant in determining the partial

derivatives of the nonlinear dynamics equations, which are required for the covariance matrix integration, than they are for the state equation integration. While in inertial coordinates the measurement equations are nonlinear in the states, the equations for relating the measurements of one tracker to the states being estimated are virtually the same for either aircraft. Thus, the problem of transferring measurements to another filter is much simpler than in line-of-sight coordinates.

There can be some real-world problems with using inertial coordinates. For example, gimbal resolver errors now are inside the estimator loop and enter the update equations directly, rather than corrupting the state outputs when transforming the filter outputs from line-of-sight axes to inertial axes. Although this is a valid issue to consider in choosing a coordinate frame, it is not the subject of this thesis, and the inertial frame was judged to be more suitable overall.

The modeling of target dynamics in inertial coordinates is not very difficult, and for the most part, follows the work of previous authors (Ref 3). Relative position is chosen rather than absolute position to avoid a dependence on ownship absolute position. Total, rather than relative target velocity, was chosen because total velocity is needed in the acceleration dynamics discussed subsequently. For the purpose of this investigation, the choice was made for convenience, though no loss of generality results. Since the differential equations for position and velocity are deterministic, i.e., no additive noise is included, the only basis for any expected difference would be in the manner in which ownship quantities enter the equations. Since the ownship quantities in the simulation are assumed error-free, there should be no difference in the simulated performance.

The target acceleration model is the point at which some statistical uncertainty must be introduced. Since the target future acceleration is not known exactly, some amount of additive noise must be included. However, there is some systematic portion of the acceleration rate, for which the mathematical "random walk" is certainly not very descriptive. An acceleration model widely used (Ref 3, 5, 8) has been the exponentially time-correlated acceleration model, in which acceleration rate has been described as

$$\dot{\underline{a}}(t) = -\lambda \underline{a}(t) + \underline{w}(t)$$

where $\underline{a}(t)$ is the acceleration, λ is a scalar coefficient, and $\underline{w}(t)$ is zero-mean, gaussian "white" noise. The purpose of λ is merely to reduce the bandwidth of the gaussian noise, and to make the covariance of $\underline{a}(t)$ bounded. Unfortunately, there are two serious problems with this model. Examining the statistics of this model, it is found that (Ref 6:184)

$$E\{a(t)a(t+\tau)\} \triangleq \phi_{xx}(\tau) = \sigma_A^2 \, e^{-\lambda|\tau|} \tag{2-1}$$

and

$$\sigma_A^2 = \frac{Q}{2\lambda} \tag{2-2}$$

where $\phi_{xx}(\tau)$ = autocorrelation function for acceleration model

σ_A^2 = variance of acceleration model output

Q = strength of driving noise, $\underline{w}(t)$

λ = inverse of correlation time

The first problem occurs when λ is picked large enough to make a short correlation time. Its effect in the filter state equations is to reduce

the filter's acceleration estimate toward zero. While for an ensemble of targets it may be expected that there will be as many targets accelerating east as west, yet for any given target, its acceleration can stay non-zero in one direction for considerable periods of time. The second problem results from the first. Since the process tends to be zero mean, the variance of the additive noise, $\underline{w}(t)$, must be quite high to allow for the $\pm$6g typical of aircraft maneuvers. Sample calculations using this model are included in Appendix A.

A more appropriate model of target maneuvers is to consider that the target continues to turn at a constant turn rate ($\underline{\omega}_t$) so that its inertial velocity ($\underline{V}_t$) and acceleration ($\underline{A}_t$), if undisturbed, would describe circles in the plane determined by $\underline{V}_t$ and $\underline{A}_t$. Disturbances would add to the acceleration so as to increase or rotate current turn rate, or change the speed. This model should, and does in fact, give higher performance. For the purpose of the simulation, both terms are included for evaluation, even though only one model is evaluated on any given simulation run.

Before proceeding with the definition of the required matrices for this application, the nomenclature and generic equations for the stochastic differential equation, mean and covariance differential equations, and extended Kalman filter will be listed (Ref 6:163-170). The stochastic differential equation is, using white noise notation:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t),t) + B(t)\underline{u}(t) + G(t)\underline{w}(t) \qquad (2\text{-}3)$$

where $\underline{x}(t)$ is the time-varying state

$\underline{f}(\underline{x}(t),t)$ is the homogeneous part of the differential equation

$B(t)$ is the input weighting matrix

$\underline{u}(t)$ is the input deterministic vector

$G(t)$ is the noise weighting matrix

$\underline{w}(t)$ is the zero-mean white noise vector of strength $Q(t)$

The approximate differential equations for propagating the conditional mean and covariance between sampled-data measurement times are given by:

$$\dot{\hat{\underline{x}}}(t) = \underline{f}(\hat{\underline{x}}(t),t) + B(t)\underline{u}(t) \tag{2-4}$$

$$\dot{P}(t) = F(\hat{\underline{x}}(t),t)P(t) + P(t)F^T(\hat{\underline{x}}(t),t) + G(t)Q(t)G^T(t) \tag{2-5}$$

where $\hat{\underline{x}}(t)$ = approximated conditional mean value estimate of $\underline{x}(t)$

$P(t)$ = conditional covariance matrix estimate of $\underline{x}(t)$

$F(\underline{x}(t),t)$ = matrix of partials,

$$Fij = \frac{\partial f_i}{\partial x_j}$$

$Q(t)$ = strength of white gaussian noise, $\underline{w}(t)$

Although the above equations can be re-written in terms of a state-transition matrix, $\Phi(t+\tau,t)$, for a linear differential equation, the above form will be used since $\underline{f}(\underline{x}(t),t)$ is nonlinear in $\underline{x}(t)$.

The state and covariance estimates are updated whenever the next measurement becomes available (Ref 6:217). At the next measurement time t_i, the form of the measurement is assumed to be:

$$\underline{z}(t_i) = \underline{h}(\underline{x}(t_i), t_i) + \underline{v}(t_i) \tag{2-6}$$

where,

$\underline{Z}(t_i)$ = noise-corrupted vector measurement at discrete time t_i

$\underline{h}(\underline{x}(t_i),t_i)$ = nonlinear measurement function

$\underline{v}(t_i)$ = *discrete zero-mean white gaussian noise at time* t_i *with strength* $R(t_i)$

The update itself is performed via the following equations

$$K(t_i) = P(t_i^-)H^T(t_i)[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]^{-1} \quad (2\text{-}7)$$

$$P(t_i^+) = P(t_i^-) - K(t_i)H(t_i)P(t_i^-) \quad (2\text{-}8)$$

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + K(t_i)[\underline{z}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)] \quad (2\text{-}9)$$

where,

$K(t_i)$ = Kalman gain at time t_i

$H(t_i)$ = matrix of partials,

$$Hij = \frac{\partial h_i}{\partial x_j}$$

t_i^- = *time* t_i *immediately preceding update*

t_i^+ = time t_i immediately following update

The filter dynamic equations used in this thesis are shown below.

Position: $\dot{\underline{P}}_{t/a} = \underline{V}_t - \underline{V}_a$ (2-10)

Velocity: $\dot{\underline{V}}_t = \underline{A}_t$ (2-11)

Turning Acceleration: $\dot{\underline{A}}_t = -\omega_t^2 \underline{V}_t + \underline{w}_1$ (2-12a)

or

Correlated Acceleration: $\dot{\underline{A}}_t = -\lambda \underline{A}_t + \underline{w}_2$ (2-12b)

The subscript "a" denotes attacker, "t" denotes target, "t/a" denotes target relative to attacker, and $\underline{w}_1$ and $\underline{w}_2$ are zero-mean white, gaussian noise of differing strengths. A "dot" over a variable indicates the

derivative of a vector as seen by an observer in an inertial coordinate frame. The ω_t^2 term must be calculated from velocity and acceleration as follows. Acceleration of the target is the inertial derivative of the velocity vector, and can be re-expressed in terms of the derivative taken in a coordinate frame rotating with the target's velocity vector as

$$\underline{A}_t = \frac{i_d}{dt} \underline{V}_t \equiv \dot{\underline{V}}_t^i \tag{2-13}$$

$$\underline{A}_t = \frac{v_d}{dt} \underline{V}_t + \underline{\omega}_{vi} \times \underline{V}_t = \dot{\underline{V}}_t^v + \underline{\omega}_{vi} \times \underline{V}_t \tag{2-14}$$

Crossing $\underline{V}_t$ into both sides yields

$$\begin{aligned} \underline{V}_t \times \underline{A}_t &= \underline{V}_t \times \dot{\underline{V}}_t^v + \underline{V}_t \times (\underline{\omega}_{vi} \times \underline{V}_t) \\ &= 0 + (\underline{V}_t \cdot \underline{V}_t)\, \underline{\omega}_{vi} - (\underline{V}_t \cdot \underline{\omega}_{vi})\, \underline{V}_t \end{aligned} \tag{2-15}$$

The first term was zero because the vectors were colinear. Since any component of $\underline{\omega}_{vi}$ along target velocity does not affect the acceleration computation, it can be set to zero. Dividing both sides by $(\underline{V}_t \cdot \underline{V}_t)$ yields

$$\text{Target turn rate:} \quad \underline{\omega}_t \triangleq \underline{\omega}_{vi} = \frac{\underline{V}_t \times \underline{A}_t}{\underline{V}_t \cdot \underline{V}_t} \tag{2-16}$$

$$\omega_t^2 = \underline{\omega}_t \cdot \underline{\omega}_t \tag{2-17}$$

Due to the suitability of the constant turn-rate model, the strength of the additive noise can be based on some reasonable estimate of the acceleration rate away from the constant turn-rate path. For this study, a constant power spectral density, q, of 1/4 g^2/sec was chosen as reasonable. The corresponding autocorrelation function is

$E\{w_1(t)w_1^T(t+\tau)\} = -qI\delta(\tau)$, where q = 1/4 g^2/sec, is also the strength of the white noise. Since it is not possible to define the most accurate value without statistically analyzing a large sample of tactical target maneuvers, the arbitrariness of the choice is actually preferable to finding the most accurate value for the limited sample of maneuvers simulated.

To complete the description of the filter dynamics, the "F" and "Q" matrices used in the covariance integration are developed.

$$F = \frac{\partial \underline{f}(\underline{x}(t))}{\partial \underline{x}} \tag{2-18}$$

where $\underline{x}$ is the state vector formed from the 9 components of (Eqs 2-10 - 2-12), and $\dot{\underline{x}} = \underline{f}(\underline{x}(t)) + \underline{w}(t)$

$$F = \begin{bmatrix} 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & I_3 \\ 0_3 & F_{3,2} & F_{3,3} \end{bmatrix} \tag{2-19a}$$

where, for the turning target acceleration model,

$$F_{3,2} = -\omega_t^2 I_3 - 2\underline{V}_t \left[\frac{\partial \underline{\omega}_t}{\partial \underline{V}_t}\, \underline{\omega}_t\right]^T$$

$$= -\omega_t^2 I_3 - 2 \begin{bmatrix} V_{tx} \\ V_{ty} \\ V_{tz} \end{bmatrix} \left[\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_t}{\partial V_{tx}}\right)\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_t}{\partial V_{ty}}\right)\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_T}{\partial V_{tz}}\right)\right] \tag{2-19b}$$

$$F_{3,3} = -2\underline{V}_t \left[\frac{\partial \underline{\omega}_t}{\partial \underline{A}_t}\, \underline{\omega}_t\right]^T \tag{2-19c}$$

$$= -2 \begin{bmatrix} V_{tx} \\ V_{ty} \\ V_{tz} \end{bmatrix} \left[\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_t}{\partial A_{tx}}\right)\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_t}{\partial A_{ty}}\right)\left(\underline{\omega}_t^T \frac{\partial \underline{\omega}_t}{\partial A_{tz}}\right)\right]$$

and for the time-correlated acceleration model,

$$F_{3,2} = 0_3 \tag{2-19d}$$

$$F_{3,3} = -\lambda I_3 \tag{2-19e}$$

and where I_3 and 0_3 are three-by-three identity and zero matrices, respectively.

$$Q = \begin{bmatrix} 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & qI_3 \end{bmatrix} \tag{2-20}$$

where $q = 256$ ft^2/sec^5 corresponds to 1/4 g^2/sec for the turning model, and $q = 1000.$ ft^2/sec^5 for the correlated model.

Measurements.

For the single tracker case, the target measurements available to the tracking aircraft are assumed to be line-of-sight direction and slant range, which is the magnitude of the relative position vector. To construct an estimator, it is necessary to relate each of the measurements to the estimator states in a measurement equation, model the measurement errors with a random additive noise, and calculate the partial derivative matrices required to update the estimator covariance. At this point, it is worth discussing the basic philosophy used in developing these measurement models. Whereas the dynamics model is fairly straightforward, with the possible exception of the acceleration rate equation, which was discussed in the previous section, the measurement errors occurring in actual target tracking hardware are anything but straightforward. Many error sources are fairly well known, such as

resolver cyclic errors, alignment errors, radar radome refraction errors, and analog-to-digital conversion errors. Others, like radar angular scintillation, or atmospheric optical variations, are harder to describe. It seemed both appropriate and convenient for this study to simplify measurement errors as much as possible. It was considered appropriate because the initial feasibility of cooperative filters should be established before proceeding with more elaborate evaluations. It was convenient because any attempt to locate accepted error models could easily dissipate most of the available research time.

The line-of-sight direction measurement could be in any of several forms. Two of these are Euler angles and direction cosines. Euler angles are formed by rotating one coordinate frame into alignment with a second by rotations, in a prescribed order, about successively orthogonal axes. They are described in any elementary physics text and are the most familiar and easy to visualize. They have the disadvantage of a mathematical singularity when the tracking axis is aligned with the outer rotational axis. To avoid involving the estimator in unnecessary detail concerning how a third Euler angle (physically, a third gimbal) can be used to avoid this, it is easier to assume that the sensor handles all the gimbal angle relationships and only outputs the measured line-of-sight direction in the form of direction cosines.

Direction cosines, which are the components of the line-of-sight unit vector resolved in attacker axes, are mathematically well behaved, having no singular regions. Since there are only two degrees of freedom for a unit vector, three direction cosines contain redundant information. Assuming that the sensor has some processing capability, it can combine the resolver outputs and raw tracker error into a unit vector,

as well as normalize the result. At this point, a question arises concerning modeling of errors in direction cosines. Since the direction cosines are normalized by definition, there can be no error along the length of the unit vector. Does this imply that the error model needs to ensure that the error along the line-of-sight is zero? As will be shown in Appendix B, this is not necessary because the elements of the Kalman gain "K" corresponding to a measurement along the line-of-sight go to zero, such that the filter ignores that component of the measurement.

The direction cosine measurements are resolved into attacker body axes, and it is this vector that has the additive white noise errors. In order to relate this vector to the state vector, which is in inertial coordinates, the transformation matrix from inertial to attacker body axes is used. In equation form, this is expressed as

$$\text{L.O.S. Unit Vector: } \underline{U}_R^A = \frac{1}{R} T_{IA} \underline{P}_{t/a}^I \tag{2-21}$$

$$\text{Unit Vector Measurement: } \underline{U}_{R_m} = \underline{U}_R^A + \underline{v}_{UR} \tag{2-22}$$

where $\underline{U}_R^A$ = true L.O.S. unit vector in attacker body axes

R = true slant range = distance from attacker to target

T_{IA} = 3x3 transformation matrix which transforms a given vector from inertial to attacker body coordinates

$\underline{P}_{t/a}^I$ = relative position vector in inertial coordinates described in Dynamics section

$\underline{U}_{R_m}$ = measured L.O.S. vector

$\underline{v}_{UR}$ = white Gaussian noise vector

The range measurement equation and assumptions are fairly straightforward. Since these are scalar, rather than vector quantities, no

transformation matrix is required. The measurement equations are

$$R = [\underline{P}_{t/a} \cdot \underline{P}_{t/a}]^{1/2} \tag{2-23}$$

$$R_m = R + v_R \tag{2-24}$$

where R_m = measured range

v_R = white Gaussian range noise

For this study, a typical sample rate of 25 per second for all measurements was used. Thus, the white noise error model has finite bandwidth, which is more plausible than the infinite bandwidth of continuous-time white noise. The standard deviations chosen for the discrete white Gaussian errors are given in Table 1.

Table 1

Assumed Measurement Errors

Measurement	Error (1σ)
Range	30 ft
Direction cosines, radar	0.004 (rad)
Direction cosines, IR	0.001 (rad)

The latter two values represent 4 milliradians for radar and 1 milliradian for infrared as typical. Due to a large part of the evaluation concentrating on angle-only tracking, two different values were used to ensure that the filter worked over a reasonable range of errors.

Two matrices must be evaluated for the filter to perform the measurement update. The "H" matrix is the sensitivity of the measurement

vector to the state vector, i.e.,

$$\partial \underline{h}/\partial \underline{x} \Big| \hat{\underline{x}}(t_i^-).$$

The "R" matrix is the positive definite covariance of the measurement errors. The "H" matrix is the partial derivative of equations (2-21) to (2-24) with respect to each of the filter states, evaluated at the filter's estimate of the state vector prior to the measurement. With a measurement vector defined as

$$\underline{z} = \begin{bmatrix} R_m \\ \underline{U}_{R_m} \end{bmatrix} \tag{2-25}$$

the "H" matrix becomes

$$H = \frac{1}{\hat{R}} \begin{bmatrix} \hat{\underline{P}}^T_{t/a} & 0_{1x3} & 0_{1x3} \\ T_{IA} - \hat{\underline{U}}^A_R \frac{\hat{\underline{P}}^T_{t/a}}{\hat{R}} & 0_{3x3} & 0_{3x3} \end{bmatrix} \tag{2-26}$$

where all quantities have been previously defined, and are to be evaluated at the state estimate, superscript "T" means transpose, and equations (2-21). (2-23) are used to compute $\hat{\underline{U}}^A_R$ and $\hat{R}$. $\underline{V}_a$ (Eq. 2-10) and T_{IA} (Eq. 2-26) are ownship quantities supplied external to the filter, and assumed to be error-free for this feasibility study.

The "R" matrix development assumes that the measurement errors are uncorrelated with each other, and that the matrix elements are constant. The full matrix is given below for the case of all measurements being available.

$$R = \begin{bmatrix} R_R & 0 & 0 & 0 \\ 0 & R_{UR} & 0 & 0 \\ 0 & 0 & R_{UR} & 0 \\ 0 & 0 & 0 & R_{UR} \end{bmatrix} \tag{2-27}$$

where each R_i is determined from

$$R_i = E(v_i^2) \tag{2-28}$$

Filter Algorithm.

The filter algorithm, as such, is not a subject of this investigation. That is, while the basic matrices must be defined and the basic measurement equations extended for cooperative information transfer, the efficiency, numerical stability, and computational aspects are not being investigated. Accordingly, it was appropriate to use an existing algorithm for which computer software was already developed. The filter algorithm in SOFE* (Ref 7) uses a Runge-Kutta numerical integration of the filter state and covariance estimates between measurements, and a Carlson Square-Root measurement update.

The numerical integration as orginally incorporated in SOFE was fifth-order variable step, but as explained in Chapter 4, this was reduced to second-order fixed step for computational efficiency. Thus the propagation equations are very general, at the expense of more computation than a more efficient method, which might be developed for a specific application. The Carlson Square-Root measurement update (Ref 6:385) is used in SOFE to avoid numerical problems which frequently occur in the standard Kalman update when the covariance is drastically

*SOFE: Simulation for Optimal Filter Evaluation

reduced by an accurate measurement. Because of the way in which the Square-Root filter equations are developed, the measurements must be processed one at a time, necessitating uncorrelated measurement errors. Thus the "R" matrix must be diagonal, or the measurements must be transformed so that the new "R" is diagonal (Ref 6:375). This fact will become important in the implementation of state estimate transfers that will be discussed in the next section. Also, the use of a scalar update produces somewhat different results than a vector update for the extended Kalman filter, since the measurement sensitivity matrix is linearized about a different $\hat{x}(t_i^-)$ after each scalar measurement update.

Dual Tracker Estimation

The subject of this thesis is an investigation of applying Kalman filtering techniques to the combination of tracker information from different aircraft. The previous section covering single tracker estimation serves as the basis for the application extensions covered in this section. The extensions discussed can be placed into two categories. First, it is possible to develop a dual tracker Kalman filter that is an application extension, but not a theoretical extension of extended Kalman filtering theory. The example illustrated herein is incorporation of the alternate tracker's measurements into the ownship filter through the standard measurement update equations. This in no way violates the basic formulation of an extended Kalman filter. The second category does not satisfy the basic formulation, but might involve using the other tracker's state estimates as though they were measurements, or combining or resetting state estimates in some other way than through the standard measurement update equations. Before discussing these possibilities, though, the dynamics and measurement equations for single tracker estimation must be re-examined to develop additional necessary equations.

Dynamics.

As stated in the introduction, identical structures in each of the cooperating filters are assumed. Thus, equations (2-10) through (2-12), and (2-16) through (2-20) are used in each of the filters for the dynamic model. It is important to note that the actual disturbance noise in equation (2-12) is identical for both filters, since it is uniquely determined by a single target. This fact will be important in the later

section where possible structures for dual tracker estimation are developed.

At this point, it is important to distinguish between what can be denoted as a joint filter and dual filters. A joint filter would be a single 9-state filter which incorporates the fact that the disturbance noise is identical as seen by each tracker. However, such a dynamics formulation does not satisfy the requirement that each tracking aircraft have its own filter capable of autonomous estimation. Dual filters, on the other hand, are separate filters estimating the same basic process, the target motion. Depending upon the structure used, each filter may or may not be optimal, and may or may not duplicate some or all of the other filter's processing.

Measurements.

Direct measurement transfer is a straightforward application of the extended Kalman filter, and will be investigated first. The principal new element involved with this approach is the need for, and use of accurate relative position and a common coordinate frame for the cooperating aircraft. As previously noted, this additional information is assumed error-free in this preliminary investigation, although such would certainly not be the case in an actual situation. Presumably, estimating this information with sufficient accuracy could be an independent problem, and further investigation of the information transfer problem could then be evaluated more extensively.

In order to incorporate a measurement from the other cooperating aircraft (henceforth referred to as an alternate measurement), an equation must be developed which defines the alternate measurement in terms of states being estimated by the primary filter. ("Primary" is not

intended to imply "more important", but the local one which is processing the information. Since both filters are configured identically, they each provide primary measurement to their own filters, and alternate measurements to the other's.)

Since the primary filter must calculate the expected secondary measurement, it requires the inertial-to-attacker transformation matrix for the alternate tracker, and the position vector to the alternate tracker. Note that the use of relative target position as a state is what requires a different "H" matrix and $\underline{h}(\underline{x})$ function. If absolute target position were the state, the identical form for "H" and $\underline{h}(\underline{x})$ would be used, although evaluated at somewhat different nominal values in each filter Despite this apparent advantage of absolute target position, the use of relative target position is retained because it should not have the roundoff errors which would likely occur when the target moved very far away from the absolute origin. The measurement equations can now be developed making use of equation (2-21) through (2-26). As those equations are written, $\hat{\underline{P}}_{t/a}$ is the estimate of target position relative to the primary tracker. To clarify the notation in the following development, let

$\ddot{\underline{P}}_{t/a11}$ = target position estimate relative to primary, estimated by primary

$\hat{\underline{P}}_{t/a21}$ = target position estimate relative to secondary, estimated by primary

$\hat{\underline{P}}_{t/a12}$ = target position estimate relative to primary, estimated by secondary

$\hat{\underline{P}}_{t/a22}$ = target position estimate relative to secondary, estimated by secondary

Then, $\hat{\underline{P}}_{t/aij}$ is the filter position state directly when i=j, and must be calculated when i≠j. This is calculated as

$$\hat{\underline{P}}_{t/aij} = \hat{\underline{P}}_{t/ajj} + \underline{P}_{aj/ai} \tag{2-29}$$

where $\underline{P}_{aj/ai}$ is the (known) position of attacker "j" with respect to attacker "i". Equation (2-29) can be substituted into equation (2-21) through (2-26) for $\hat{\underline{P}}_{t/a}$ when processing alternate measurements, along with T_{IA} for the alternate tracker. After substitution, the equations can be used without further modification when calculating the "H" matrix, due to the fact that $\underline{P}_{aj/ai}$ enters only in an additive form in equation (2-29) such that

$$\frac{\partial \hat{\underline{P}}_{t/aij}}{\partial \hat{\underline{P}}_{t/ajj}} = I \tag{2-30}$$

Thus, when the chain rule is applied to the calculation of H_{ij}, the 3 by 9 partition of H corresponding to the target position states in equation (2-26),

$$H_{ij} = \frac{\partial h(\hat{x})}{\partial \hat{P}_{t/aij}} \frac{\partial \hat{\underline{P}}_{t/aij}}{\partial \hat{\underline{P}}_{t/ajj}} \tag{2-31}$$

H_{ij} reduces to equation (2-26) evaluated at $\hat{\underline{P}}_{t/aij}$.

From the standpoint of data transfer requirements, the following vectors must be transmitted from aircraft "i" to aircraft "j" at each time that measurement transfer is desired:

$\underline{z}_i$, the measurement vector obtained on aircraft "i"

T_{IA_i}, the inertial-to-attacker transformation matrix for attacker "i", or equivalent information

No filter outputs from attacker "i" are used, so that the filters are not structurally coupled; that is, they only process data from the same environment, $\underline{x}_t(t)$. The accuracy of the measurement processing will obviously depend upon the accuracy of $\underline{P}_{aj/ai}$, and upon the relative alignment of the inertial frames in each aircraft. It is not significant whether absolute geographic locations are known, or whether the inertial frame has a particular geographic alignment.

Possible Structures for Dual Estimation.

The preceding section discussed the application of existing extended Kalman filter theory for handling measurements transferred to the filter from another moving observation point. There are a number of other possibilities for transferring information that are not part of extended Kalman filter theory. Several of these possibilities are explored in this section, especially those that were finally evaluated using a Monte-Carlo analysis. The considerations involved in evaluating these possibilities are making maximum use of computations already performed in the alternate filter, mathematical rigor of the algorithm, if it is so developed, or potential pitfalls of a heuristic approach. The following four approaches are tentatively considered, and two of these are carried forward to numerical evaluation.

1. Direct Measurement Transfer
2. Residual Transfer
3. Using state estimates as "measurements" for the other filter

4. Optimal estimates via parallel processing

Items 1. and 3. are evaluated in simulation.

Direct Measurement Transfer.

Direct transfer of unprocessed sensor data from the alternate tracker to the primary is an approach that will be evaluated. The necessary equations for processing the alternate measurements were developed in the previous section. This approach has the following advantages:

1. The transferred data is not a function of how well the alternate estimator is functioning.
2. The dimension of the measurement is less than that of the state estimate, so that fewer variables need be transferred.
3. The measurement covariance matrix, R, is the same as for the primary tracker if both aircraft are identically configured, and hence would not need to be transmitted.
4. If all of the alternate measurements are transmitted to the primary filter and processed every sample period, then the full performance potential of the extended Kalman filter should be realized.

The approach has the following disadvantages:

1. The quality of a single sample measurement is not as high as the alternate filter's estimate of the position states. This can be demonstrated by comparing the standard deviation of the estimate to the measurement standard deviation from a simulation run. (See Chapter 5.) Thus, it does not appear to be passing the best available information.

2. No information about the alternate filter's velocity and acceleration estimates is used.

3. Transmission of all the raw measurement data may require excessive transmission time or bandwidth. Intuitively, it seems that the data could be compressed. If only a small fraction of the data is transmitted to conserve transmission bandwidth, the performance improvement may be too small.

Because this approach requires no new theory beyond the extended Kalman filter, it serves as the performance benchmark (when transfers are made every sample time) for what is possible in cooperative estimation. This baseline is attained when all measurements are transferred. Because of the desireability of minimizing the volume of transmitted data, lower rates of transfer are of interest and will be investigated.

Residual Transfer.

The filter residual is the difference between the actual and expected measurement.

$$\underline{r}(t_i) = \underline{z}(t_i) - \underline{h}(\hat{\underline{x}}^-(t_i), t_i) \tag{2-32}$$

where $\underline{r}(t_i)$ = measurement residual at time t_i

$\underline{z}(t_i)$ = measurement at time t_i

$\underline{h}(\hat{x}(t_i^-), t_i)$ = measurement function evaluated at the filter state propagated to time t_i

It has the potential advantage of not requiring separate computation of the measurement function, $\underline{h}(\ )$ as would be required if $\underline{z}(t_i)$ itself were transferred. However, it has the distinct disadvantage of including the alternate tracker's estimate in the residual. When this residual

is incorporated into the primary filter, it cross-couples the two filters in such a way that each filter processes the other filter's error in $\hat{\underline{x}}$ as though it were its own error. (This effect is developed in Appendix D.) Thus, it would not be expected to perform well because the system response of each filter is changed. In fact, this approach was initially programmed in the simulation, and did produce an unstable response after approximately 1.5 second of data transfer. This approach was then eliminated from further consideration.

Using State Estimates as Measurements.

The list of disadvantages of Direct Measurement Transfer given in the previous section motivates the consideration of transferring the alternate filter's state estimates to the primary filter. Intuitively, this should overcome each of those disadvantages if a proper method of incorporating these estimates exists. One method to be considered is to treat the alternate state estimate as though it were a measurement vector. This would have the effect of cascading the two filters so that the alternate filter's output becomes the primary filter's input.

Treating the alternate estimates as measurements does not satisfy the assumptions of the filter formulation, however. In particular, it is not true that the measurement error, $\underline{v}(t_i)$, and the additive disturbance noise, $\underline{w}(t_i)$ are uncorrelated, i.e.,

$$E[\underline{v}(t_i)\ \underline{w}(t_j)^T] \neq 0 \text{ for } t_i > t_j \qquad (2\text{-}33)$$

since $\underline{v}(t_i)$ in this case is the error in $\hat{\underline{x}}(t_i)$. Using the alternate estimates as measurements in a filter that assumues v and w to be uncorrelated can produce erroneous results. Although there are extensions

that properly account for the correlations, evaluation of the correlations is difficult, the computational burden is high, and that approach is not evaluated here. One problem that will likely result from this correlation is that the difference between the actual and assumed target acceleration will tend to produce correlated errors between the two filters' estimates. Since the filter is unaware of this, it assumes that the errors are independent, and for the case of equal covariance matrices, it will calculate a new covariance matrix that is approximately one-half of the previous value. After this process is repeated several times, both filters will have essentially identical estimates and the covariance estimate will decrease to an unrealistically low value.

One possible ad hoc solution to this problem is to use the alternate state estimate to update the primary state estimate, but not to update the primary filter's covariance matrix. This would avoid the problem mentioned in the previous paragraph, reduce the amount of computation, and leave the filter covariance and gain identical to the single filter problem. This may be reasonable for range and angle measurements, where the filter covariance converges well, but not for angle-only measurements, where it does not converge.

The additional equations required for this approach are just the measurement equations as follows:

$$\underline{z}_s(t_i) = \underline{x}_{alt}(t_i) + \begin{bmatrix} \underline{P}_{alt/pri}(t_i) \\ \hline 0 \\ \hline 0 \end{bmatrix} + \underline{v}_s(t_i) \qquad (2\text{-}34)$$

$$H_s = I \qquad (2\text{-}35)$$

$$E[\underline{v}_s(t_i)\underline{v}_s(t_i)^T] \triangleq R_s(t_i) = \hat{P}_{alt}(t_i) \qquad (2\text{-}36)$$

where,

$\underline{z}_s(t_i)$ = secondary filter state used as measurement = $\hat{\underline{X}}_s(t_i)$

$\underline{x}_{alt}(t_i)$ = true target state relative to alternate aircraft

$\underline{P}_{alt/pri}(t_i)$ = position vector of alternate aircraft relative to primary

$\underline{v}_s(t_i)$ = (assumed) white gaussian errors included in $\hat{\underline{x}}_s(t_i)$

$R_s(t_i)$ = covariance of $\underline{v}_s(t_i)$

$\hat{P}_{alt}(t_i)$ = alternate estimator's estimate of state vector covariance

Due to the fact that the identical disturbance noise, $\underline{w}(t_i)$, enters each filter's dynamics model at the acceleration rate level, while the independent range and angle measurement noise is added to target position, it might be expected that there is a higher degree of correlation between acceleration estimation errors than between position estimation errors. Thus, the simplifying assumption that $\underline{w}$ and $\underline{v}$ are uncorrelated might be less valid for acceleration state estimate transfers than velocity or position estimate transfers. In order to evaluate this possibility, the measurement vector and matrices, $\underline{z}_s$, H_s, and R_s, can be partitioned into position, velocity, and acceleration components. This will produce a measurement with either three, six, or nine components, depending upon the level of transfer selected.

One final consideration in the use of state estimates as measurements is the possibility that direct measurement transfer might need to occur along with state transfer. A specific instance of this is the situation in which two cooperating aircraft are tracking a single target in angle only. In this case, the mathematical observability of all tar-

get states depends upon combining measurements at least initially. It is also important to update the covariance matrix to prevent divergence due to the observability problem. In this instance, it may be necessary to start the estimator with direct measurement transfer and continue frequently enough for good covariance performance.

Optimal Estimates via Parallel Processing.

This section is motivated by the fact that the inherent correlation between the alternate filter's state estimates and the disturbance noise, $\underline{w}(t_i)$, is not properly considered with the previous approach. It would be desireable if an efficient algorithm could be derived which would perform at a reduced transfer rate as well as the extended Kalman filter using direct measurement transfer every measurement sample time. An efficient algorithm ideally would capitalize on the measurement processing already done in each filter separately, combining state and covariance estimates in an optimal or sub-optimal manner, with a minimum of arithmetic computation beyond the single filter computational load. Ideally, the result would be sub-optimal only because the contributing single filters are already sub-optimal due to the nonlinear $\underline{h}(\underline{x}(t),t)$ function which disallows a linear filter. What is desired could thus be described as optimal (or suboptimal) estimation using parallel processing.

The structure of the problem can be illustrated using simplified block diagrams. Existing extended Kalman filter theory, as applied directly to this problem, can be described as centralized estimation. The method of direct measurement transfer, described previously, effectively implements a complete, centralized, extended Kalman filter in each of the cooperating aircraft. Structurally, then, it can be diagrammed as in Fig 1. What is desired, however, is for each tracking

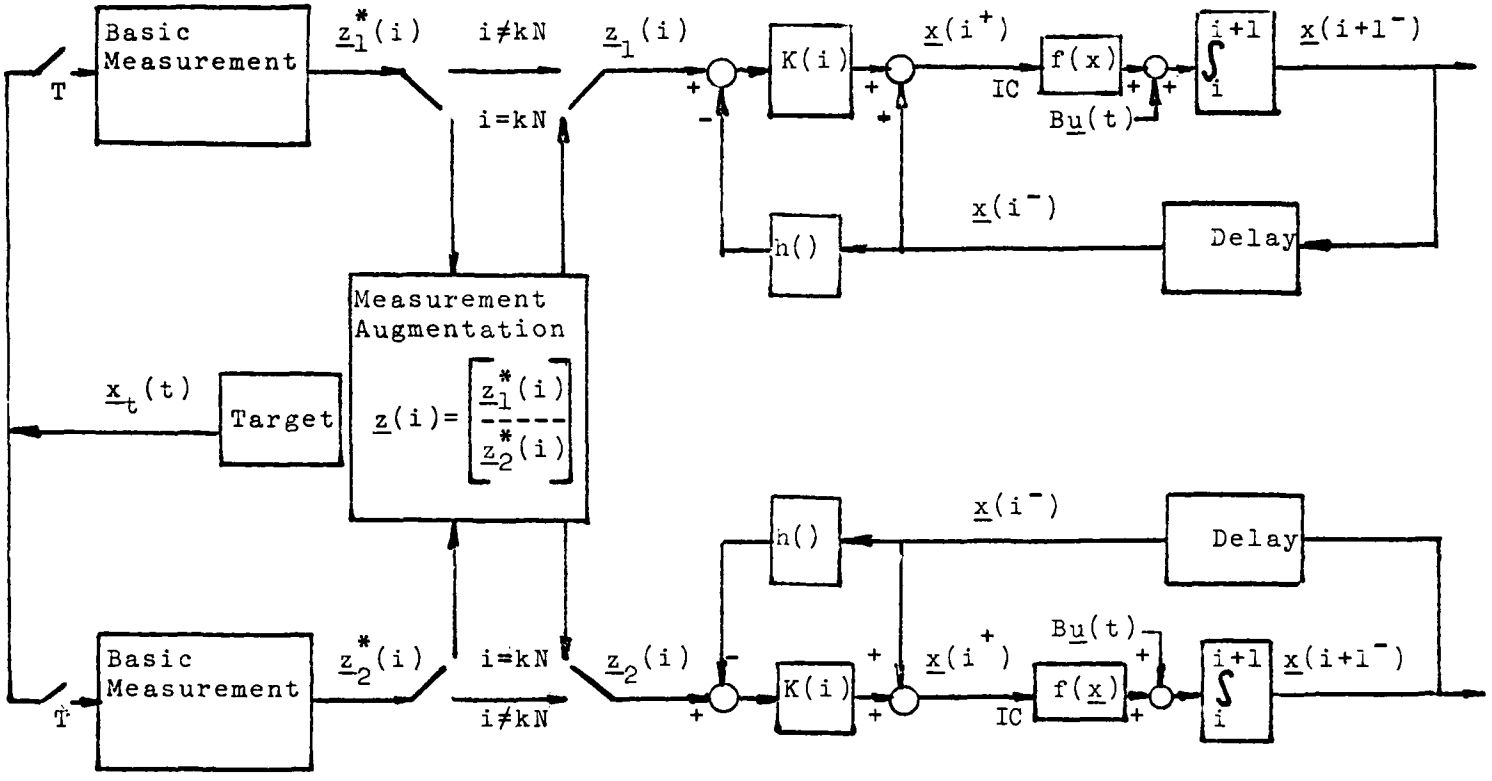


Figure 1. Kalman Filter Using Direct Measurement Transfer

aircraft to estimate the same target's states autonomously for at least several measurement sample periods, and then to combine the individual estimates in what would be an optimal estimate for a linear filter, or in this case, suboptimal due to the measurement nonlinearity. This linear system structure can be diagrammed as in Fig 2. In this structure note that N>1 measurement samples, $\underline{z}_j(t_i)$ are incorporated into each filter "j" before the optimal combination $(\hat{\underline{x}}(t_N), P(t_N))$ is formed. This implies that each filter has to perform "N" propagation and update cycles with only part of the available measurement information, so that $\hat{\underline{x}}_j(t_N)$ and $P_j(t_N)$ for each filter (j = 1 or 2) are definitely suboptimal in the context of the total measurement information, $\underline{z}_1(t_i)$ and $\underline{z}_2(t_i)$. Note also that although the errors in $\underline{z}_1(t_i)$ are presumed uncorrelated with the errors in $\underline{z}_2(t_i)$, i.e.

$$E[(\underline{z}_1(t_i) - \underline{h}_1(\underline{x}(t_i))(\underline{z}_2(t_i) - \underline{h}_2(\underline{x}(t_i))^T] = 0 \qquad (2\text{-}37)$$

the same disturbance noise, $\underline{w}(t_i)$, enters both filters identically since they are both observing the same target manuevers. Thus, the errors in $\hat{\underline{x}}_1(t_i)$ are not uncorrelated with the errors in $\hat{\underline{x}}_2(t_i)$, i.e.,

$$E[(\hat{\underline{x}}_1(t_i) - \underline{x}_t(t_i)(\hat{\underline{x}}_2(t_i) - \underline{x}_t(t_i))^T] \neq 0 \qquad (2\text{-}38)$$

In addition to the same errors being introduced through the dynamics equations, the initial errors, $[\hat{\underline{x}}_j(t_1) - \underline{x}_t(t_1)]$, will be perfectly correlated if $\hat{\underline{x}}_j(t_i)$ is initialized identically. Finally, the global optimal estimate, $\hat{\underline{x}}(t_N)$, is produced only every "N" sample periods.

Note that transferring states as measurements, as shown in Figure 3, does feedback state information every "N" sample periods, but not in an optimal fashion.

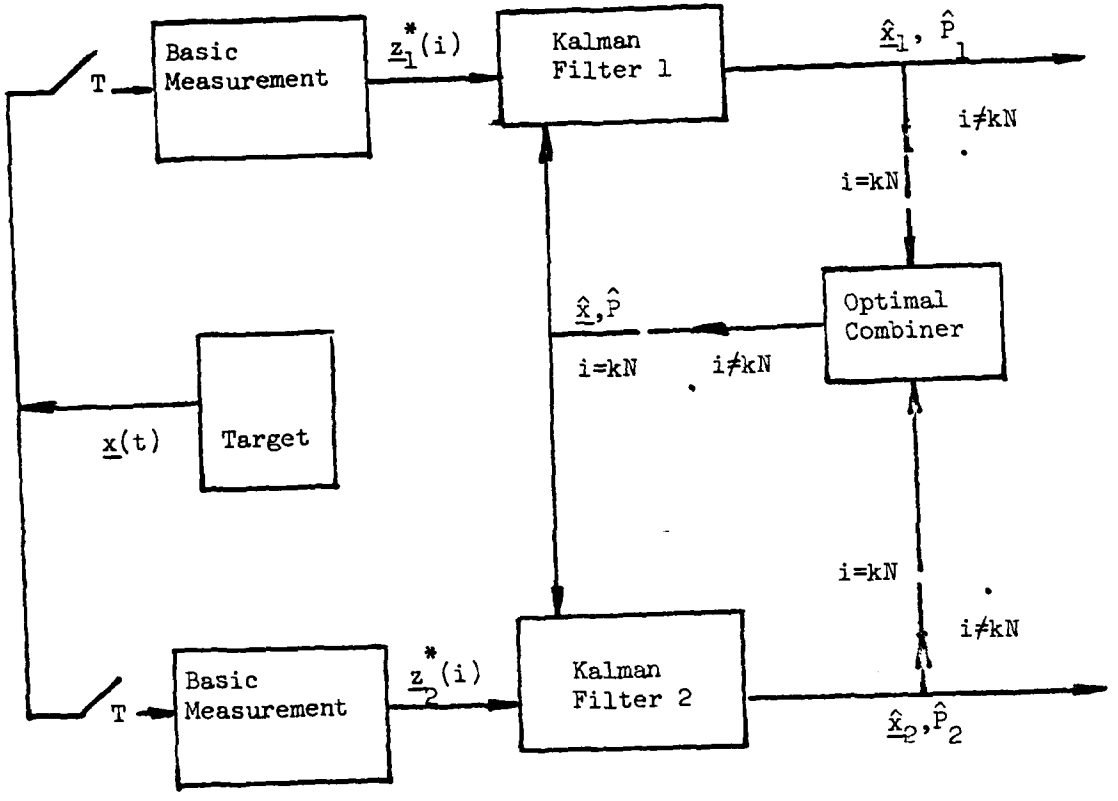


Figure 2. Desired Parallel Filtering Structure

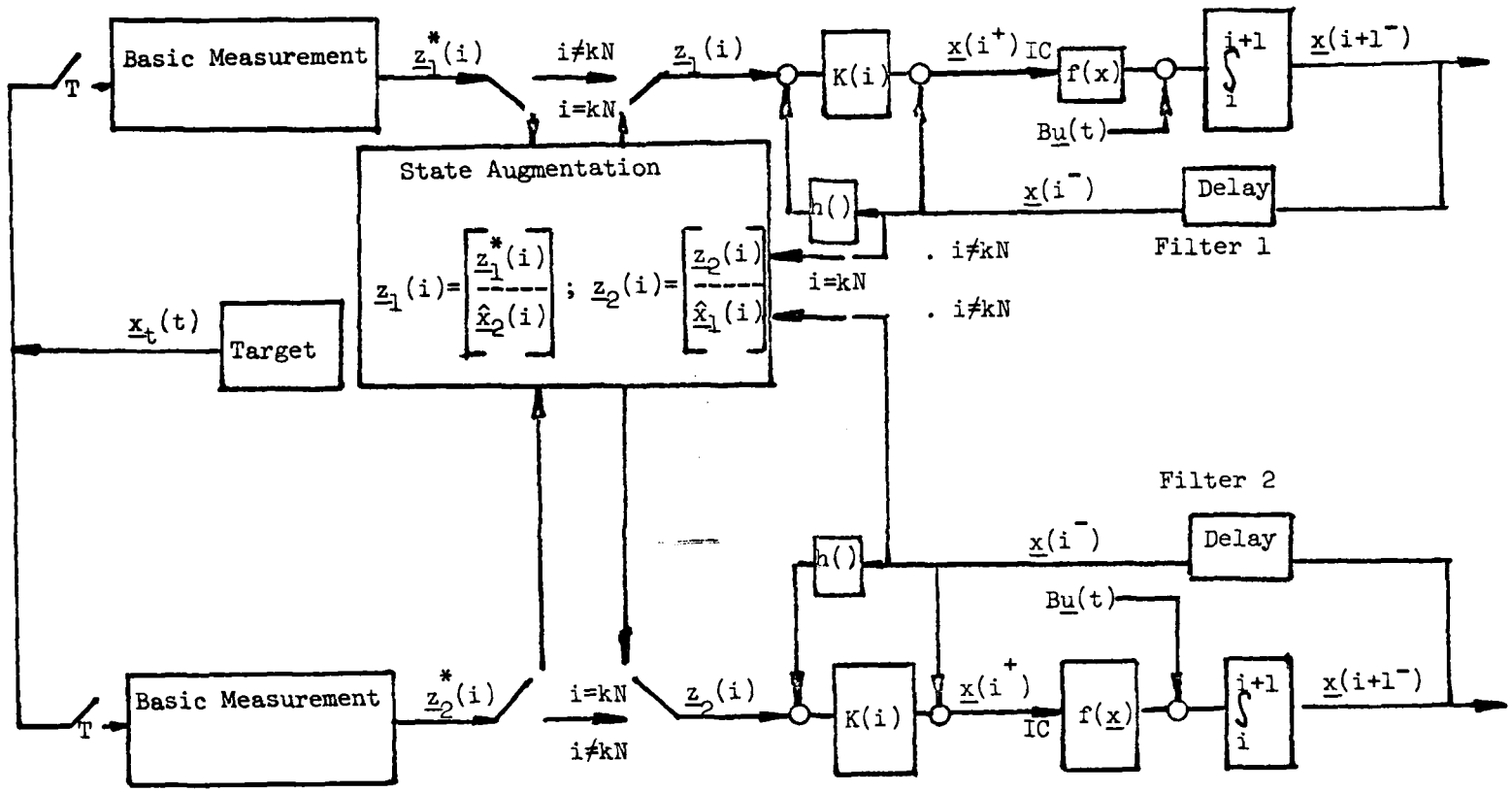


Figure 3. Filter Structure Using State Estimates as Measurements

Several attempts were made to develop such a parallel Kalman filter algorithm that would optimally combine the individual suboptimal results from each filter without involving significantly more computations than the Direct Measurement Transfer method. Since the Direct Measurement Transfer method satisfies all of the required assumptions of an extended Kalman filter, a parallel filter algorithm, if developed, could not be expected to have better estimator performance. Its only advantage would be to reduce the amount of data transfer necessary, and possibly reduce the number of computations.

The results of these attempts failed to develop such an algorithm. The basic impediment in the development is the correlated driving disturbance representing the target maneuvers, which is identical for both filters. The measurement information added to each filter during its own filter update is independent of the other filter's measurement information. However, the addition of the driving disturbance reduces that information level in such a way that the independent and dependent (between the two filters) information is no longer separable. The separation of the independent information is what is required for this algorithm to be successful.

III. Simulation Test Plans

Simulation Test Requirements

The necessity for a Monte-Carlo simulation arises from at least three aspects of the proposed filter structures. Since the filter is basically an extended Kalman filter, rather than a linear Kalman filter, there is no theoretical basis for guaranteeing that the covariance computations of the filter will accurately describe the true errors, as would be the case for a linear, full-order Kalman filter. The nonlinearities in the measurement equations are significant, so that a covariance analysis of an associated linearized Kalman filter, linearized about a given a priori nominal state trajectory, might produce significantly erroneous results. Secondly, the effect of initial filter errors are important, since for an extended filter there is not a theoretical guarantee that the initial transient will be stable. This effect could be particularly important for the case of angle-only measurements, where the first measurement provides essentially no information in range (an observability problem). Even with measurement transfer, the range estimation problem, which is geometrically a triangulation problem, is highly dependent on the relative positions of the two tracking aircraft and the target, so that different geometries should be considered. Thirdly, the proposed technique of transferring filter states as measurements does not satisfy even the assumptions of an extended Kalman filter, in that the assumption of independent errors in the two filters may reduce the covariance too low, so that even a tentative performance evaluation of that method requires a Monte-Carlo simulation.

The Monte-Carlo simulation is required to generate both true and estimated values. The true values drive the measurement equations, and both are needed to evaluate estimator errors. Since the required true values are the kinematic variables (position, velocity, and acceleration vectors) for both attackers and target, a trajectory must be generated which contains these parameters.

Since it was already decided that controlling the attacker's tactical maneuvers to aid the target estimator is outside the scope of the investigation, the trajectories can be generated independently of the Monte-Carlo simulation of the estimator. Due to the difficulty and time that would be consumed in generating random target maneuvers, and to the difficulty of interpreting results of such a simulation, a collection of deterministic trajectories is considered more desireable.

The requirements of the trajectory set are that it presents a significant variety of initial conditions and target maneuvers which will exercise the filter over a broad enough range of conditions to increase confidence that the measured performance is truly representative of its overall performance. It is important to note that tactical realism is not a specific requirement for filter evaluation. As long as the trajectory set chosen is not radically different from realistic maneuvers, the performance results should be very similar to operational performance.

A requirement of the trajectory is that two aircraft be available to track the target. Although in the general case, one might ultimately be interested in "m" attackers versus "n" targets, that should be a straightforward extension, conceptually, of the two versus one problem.

Note also, that although this investigation considers only two versus one, there is no inherent reason why attackers have to out-number targets for the technique to be used. If two interceptors had infrared search and track (IRST) sets capable of measuring angles to "n" multiple targets, the two versus one problem would have to be repeated "n" times. Although conceptually simple, these extensions obviously would involve difficult engineering problems.

Other requirements of the evaluation software are that the user be able to control initial conditions, computation of the $H(\underline{x}(t_i),t_i)$ matrix, target acceleration filter model, and timing of measurement and/or state transfers. In the area of initial conditions, there are three types of initial filter errors possible. There are zero errors (perfect initialization), random errors, and initial errors which are fixed from one Monte-Carlo run to the next. This latter type of error is needed in the evaluation of filter recovery from initial errors. Computation of $H(\underline{x}(t_i),t_i)$, while actually done with the filter state vector estimate as $H(\hat{\underline{x}}_f(t_i^-),t_i)$ in an actual implementation, may need to be investigated with the true state vector as $H(\underline{X}_t(t_i),t_i)$ in order to isolate the effect of the erroneous partial derivative evaluation point on measurement processing. The two target acceleration models to be included are the exponentially time-correlated target acceleration model and the turning target acceleration model discussed in Chapter Two. Starting, stopping, and interval timing on measurement and state transfers are to be user controlled.

The implementation of these requirements in software is discussed in Chapter Four and the appendices. The definition of test cases to be run is given in the next section.

Test Case Definition.

Various test cases are needed to determine the performance of the candidate filter configurations. The objectives of the test cases are to evaluate

1. The effect of noise and bias errors on estimates
2. Recovery from initial errors
3. Filter performance during target maneuvers
4. Filter estimates of its own error variance as compared to the true RMS errors

Due to the volume of output, this analysis will be primarily graphical, with some key results tabulated.

Some general description can be applied to all of the test cases. Unless otherwise noted, each case will have ten Monte-Carlo replications. Since the only random variables are the measurement errors, ten seemed a sufficient number, based upon previous experience. As described previously, the trajectories do not vary from one replication to the next. The initial filter error, if random initial conditions are chosen, may vary from one replication to the next.

What is referred to as "radar" means discrete-time (25 Hz) range and direction cosines measurement available throughout the entire pass with discrete-time, zero-mean white, gaussian errors of 30 feet and 4 milliradians, one sigma, respectively. "IR" (infrared) means direction cosine measurements with 1 milliradian, one-sigma errors, also at 25 Hz. The trajectories numbered 1 through 6 are described more fully in Appendix C, but are described briefly here.

Trajectory No.	Description
1.	Tail chase at 8000 feet initially, target makes 3 g level turn to right
2.	Target offset 5000 feet to right, initially turns to right, then reverses and loses speed
3.	Attackers at 20K alt., target at 15K alt. at 60,000 feet head-on, target straight and level, all three A/C at Mach=1.2
4.	Target at 60,000 feet head-on as in 3., but rolls left and turns, then rolls right and climbs
5.	Target straight ahead at 12000 feet flying straight and level on relative heading of 90° right
6.	Target initially flying straight and level at 12000 feet at relative heading 90° left, makes ascending, turning climb to left

Detailed Test Case Description.

Table 2 gives the exact conditions for the test cases described below. Blanks in a particular column indicate that that value is unchanged from the previous case.

1. Single-filter baseline (no information transfer)

1.1 Evaluate target acceleration model

Test cases 1.1.1 through 1.1.4 will be run using target trajectories 1 and 4 with both the correlated and turning acceleration models, and using radar measurements.

TEST CONDITIONS

SEQ #	SET #	CASE #	STATE TRANS	UPDATE "P"	STATE TRANS RATE	MEAS. TRANS RATE	MEAS. AVAIL	TST. TRAJ	INIT. XF	FILTER H(X(T))	ACCEL. MODEL
1	1.1	1.1.1	NONE	NO	-	0.	R+A	1	T	NO	C
2		1.1.2									T
3		1.1.3						4			C
4		1.1.4									T
5	1.2	1.2.1						2			T
6		1.2.2						3			
7		1.2.3						5			
8		1.2.4						6			
9	1.3	1.3.1					A	1	T		
10		1.3.2						4	T		
11		1.3.3						1	R		
12		1.3.4						4	R		
13		2.1.1.1				25.	R+A	1	T		
14		2.1.1.2						4	T		
15		2.1.2.1					A	1	T		
16		2.1.2.2						4	T		
17		2.1.2.3						1	R		
18		2.1.2.4						4	R		
19		2.1.2.5						1	E		
20		2.1.2.6						4	E		
21		2.1.3.1				5.		1	T		
22		2.1.3.2						4	T		
23		2.1.3.3				1.		1	T		
24		2.1.3.4						4	T		
25		2.1.4.1				5.		1	E		
26		2.1.4.2						4	E		
27		2.1.4.3				1.		1	E		
28		2.1.4.4						4	E		
29		2.1.5.1				25.		1	R	YES	
30		2.1.5.2						4	R	YES	
31		2.1.5.3						1	E	YES	
32		2.1.5.4						4	E	YES	

TEST CONDITIONS

SEQ #	SET #	CASE #	STATE TRANS	UPDATE "P"	STATE TRANS RATE	MEAS. TRANS RATE	MEAS. AVAIL	TGT. TRAJ	INIT. XF	FILTER H(X(T))	ACCEL. MODEL
	2.2										
33		2.2.1.1	P	YES	5.	0.	A	4	T		NO
34		2.2.1.2			1.						
35		2.2.1.3			0.2						
36		2.2.1.4		NO	5.						
37		2.2.1.5			1.						
38		2.2.1.6			0.2						
39		2.2.1.7	P+V	YES	5.						
40		2.2.1.8			1.						
41		2.2.1.9			0.2						
42		2.2.1.10		NO	5.						
43		2.2.1.11			1.						
44		2.2.1.12			0.2						
45		2.2.1.13	P+V+A	YES	5.						
46		2.2.1.14			1.						
47		2.2.1.15			0.2						
48		2.2.1.16		NO	5.						
49		2.2.1.17			1.						
50		2.2.1.18			0.2						
	2.2.2	(NO CASES WERE RUN WITH IMPERFECT INITIAL CONDITIONS)									
	2.2.3										
51		2.2.3.1	P	NO	1.	1.	A	4	E		NO

NOTES:
- R+A = RANGE + ANGLE
- A = ANGLE ONLY
- T = TURNING TARGET ACCELERATION MODEL
- C = TIME-CORRELATED ACCELERATION MODEL
- T = INITIALIZE XF = X TRUE
- R = INITIALIZE XF = X TRUE + RANDOM ERROR
- E = INITIALIZE XF = X ERROR
- P = POSITION TRANSFER AS MEASUREMENT
- P+V = POSITION AND VELOCITY TRANSFER AS MEASUREMENT
- P+V+A = POSITION, VELOCITY, AND ACCELERATION TRANSFER AS MEASUREMENT

TABLE 5-1
TEST OF TIME-CORRELATED AND TURNING ACCELERATION MODEL
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
1.1.1	-3.01	-10.90	-16.20	1.35	6.70	12.50	-.49	-2.10	-1.89	M
1	10.70	21.30	13.80	9.35	19.20	11.90	10.15	21.20	13.40	S
1.1.2	-.52	-2.24	-1.24	.91	4.61	8.30	-.48	-2.13	-2.06	M
2	10.20	19.40	12.90	8.84	16.90	9.84	9.56	18.60	11.00	S
1.1.3	.56	4.15	3.68	12.60	7.84	13.10	-8.60	-19.80	-18.00	M
3	14.80	25.00	14.00	47.20	56.20	14.20	42.80	53.80	23.60	S
1.1.4	1.36	4.05	.14	3.41	4.35	-.20	-5.05	-19.40	-13.30	M
4	14.20	22.60	11.80	47.80	56.60	15.10	47.10	59.30	17.20	S

1.2 Evaluate performance on each trajectory

Test Cases 1.2.1 through 1.2.4 will be run using targets 2, 3, 5, 6 with turning acceleration model.

1.3 Evaluate angle-only (IR) performance of single filter

Test cases 1.3.1 through 1.3.4 will be run with angle-only on target trajectories 1, 4 with both perfect and random initial errors.

2. Cooperative filter comparison

2.1 Direct measurement transfer

2.1.1 Radar tracking benchmark

Test Cases 2.1.1.1 and 2.1.1.2 will be run on targets 1, 4 with perfect initial conditions and 25 Hz measurement transfer rate.

2.1.2 Angle-only (IR) benchmark

Test Cases 2.1.2.1 through 2.1.2.6 will be run on targets 1, 4 with true, random and constant error initial conditions.

2.1.3 Angle-only (IR) tracking with lower transfer rates and no initial errors

Test cases 2.1.3.1 through 2.1.3.4 will be run on targets 1, 4 at transfer rates of 5 Hz and 1 Hz.

2.1.4 Angle-only (IR) tracking with lower transfer rates and imperfect initial conditions

Test cases 2.1.4.1 through 2.1.4.4 will be run on targets 1, 4 at transfer rates of 5 Hz and 1 Hz.

Results of 2.1.3 and 2.1.4 will be compared with 2.1.1 and 2.1.2 respectively.

2.1.5 Effects of imperfect H on angle-only (IR) estimation

Test cases 2.1.5.1 through 2.1.5.4 will be run on targets 1, 4 at 25 Hz transfer rate with perfect $H = H(\underline{x}_t)$ with random and error initial conditions and compared with 2.1.2.

2.2 State Transfer Investigation

2.2.1 State transfer with no measurement transfer and perfect initial conditions

Test cases 2.2.1.1 through 2.2.1.18 will be run on target 4 with state transfer occurring at 5.0, 1.0, and 0.2 Hz. Target 4 was chosen because of the longer range, less advantageous geometry for triangulation, and significant maneuvers. These cases will be run with both state and covariance updates, and with state updates only. Normal ownship measurement updates continue as in previous test cases. In addition, combinations of position only, position and velocity, and position, velocity, and acceleration transfers will be simulated.

2.2.2 State transfer with no measurement transfer and imperfect initial conditions

Test cases from set 2.2.1 that perform satisfactorally will be rerun with imperfect initial state estimates.

2.?.3 State transfer with simultaneous measurement transfer

Test cases similar to set 2.2.1 will be run with state and measurement transfer at appropriate transfer rates.

IV. Software Development

In order to meet the evaluation requirements which were described in Chapter Three, existing software had to be modified and new software written. The modifications to existing software were minor, and the new software was developed only for this particular application of the extended Kalman filter to cooperative estimation. This chapter will describe the basic capabilities of the Monte-Carlo Kalman filter simulation program, "SOFE" (Simulation for Optimal Filter Evaluation), the application routines which were written as part of the thesis effort and which supply problem-dependent data needed by "SOFE", and several pre- and post-processing programs associated with "SOFE".

Monte-Carlo Evaluation.

The reason for requiring a Monte-Carlo evaluation was described in Chapter Three. The basic requirements of such a simulation are described here as a prelude to the software discussion. In order to evaluate the results, or outputs, of a process which has stochastic process inputs, the evaluation must take into account the effect of the process on those stochastic inputs. One way that this can be done is to characterize the inputs with some probability distribution function,

$$F_{\underline{X}}(\underline{\xi}) = P(\underline{X} \leq \underline{\xi}) \tag{4-1}$$

where F = probability distribution function

$\underline{X}$ = random input variable (vector)

$\underline{\xi}$ = particular value of $\underline{X}$

$P(\underline{X} \leq \underline{\xi})$ = "probability that $\underline{X}$ is less than or equal to "ξ"

If one could somehow use the input probability distribution, along with the process description, to produce the joint probability distribution of the output at each time t_i (e.g. $F_{xt_1, xt_2, \cdots xt_n}$) the evaluation would be complete. However, this is not practially possible for this problem. Another one possibility might be to let the process operate on a certain parameter of $F_{\underline{x}}(\underline{\xi})$, such as the covariance. For certain restricted conditions, i.e., a linear process with gaussian-distributed random inputs, this is possible and produces an exact result, but for a nonlinear process, such as the one simulated, the output covariance cannot be computed exactly. In addition, the mean and covariance are no longer sufficient to characterize completely the output probability distribution.

The remaining possibility is to sample the input stochastic process to obtain an actual realization, and then to process it to obtain an actual realization of the output random variables. This process is duplicated many times so that enough samples from the output random variable exist to characterize its probability distribution. This is called a Monte-Carlo simulation.

An important concern of this problem is to measure estimated quantities against the corresponding true values and to describe the errors which result from the differences. For this problem, the important quantities are

$\underline{x}_t(t_i)$ = vector of true target quantities at discrete intervals

$\hat{\underline{x}}_f(t_i)$ = vector of estimated target quantities at discrete intervals (with measurement updates, there will be values at both t_i^- and t_i^+)

$$\underline{\varepsilon}(t_i) \triangleq \underline{x}_t(t_i) - \hat{\underline{x}}_f(t_i) \qquad (4\text{-}2)$$

= error vector at time t_i

When the Monte-Carlo simulation is run, the entire time history of $\hat{\underline{X}}_f(t_i)$ is generated a number of times, each time with a different random time history of measurement noise. The values generated for a set of Monte-Carlo runs are

$$\underline{x}_t(t_i,m) = \underline{x}_t(t_i) \text{ for run \#m}$$

$$\hat{\underline{x}}_f(t_i,m) = \hat{\underline{x}}_f(t_i) \text{ for run \#m}$$

$$\underline{\varepsilon}(t_i,m) = \underline{\varepsilon}(t_i) \text{ for run \#m}$$

Since for this simulation, the true target quantities do not vary from one run to the next, $\underline{x}_t(t_i,m)$ does not vary with "m".

The outputs from the simulation are reduced to form the mean and standard deviation from equations (4-3) and (4-4) (Ref 10)).

$$\text{Av}[\underline{\varepsilon}(t_i)] = \frac{1}{M} \sum_{m=1}^{M} \underline{\varepsilon}(t_i,m) \qquad (4\text{-}3)$$

$$\text{Sig}[\underline{\varepsilon}(t_i)] = \frac{(M-1.0)^{\frac{1}{2}}}{M-1.25} \left[\sum_{m=1}^{M} (\underline{\varepsilon}(t_i,m) - \text{Av}[\underline{\varepsilon}(t_i)])^2\right]^{1/2} \qquad (4\text{-}4)$$

"SOFE" Usage.

The "SOFE" program (Ref 7) was developed by Stanton H. Musick of the Reference Systems Branch of the Avionics Laboratory, primarily for application to inertial navigation problems, although the program itself is general purpose. The total "SOFE" program includes basic "SOFE", which is invariant from one problem to another and provides the basic Monte Carlo simulation structure and efficient general purpose routines

as for integration, matrix manipulation, and input/output functions, and user-written "SOFE", which defines the functions and matrices peculiar to a particular problem. The basic filter computations performed by basic "SOFE" are described in Chapter Two in the Filter Algorithm section. Basic "SOFE" also performs problem initialization, Monte-Carlo run initialization, and input/output including data printout, printer plots, trajectory input, and Monte-Carlo data output. The Monte-Carlo data is analyzed and plotted using a post-processing program, "SOFEPL", described later.

Basic "SOFE" has provisions for both a truth model, i.e., a set of nonlinear stochastic differential equations describing the true values, and for an external trajectory. For this problem, the truth model states are defined to be equal to the external trajectory values.

"SOFE" Application Routines.

There are a minimum of nine user-written routines which are necessary to satisfy the requirements of basic "SOFE". In addition to these nine, there are ten additional routines which are called from within the nine required routines, and which were designed for this investigation. Detailed discussion of these routines is given in Appendix E.

Pre- and Post-Processing Software

Trajectory Generation and Plots.

Trajectories were generated using an interactive aerial gunnery simulation. Initial conditions and target maneuvers were controlled from an interactive computer terminal, while the attacker flew a gunnery lead pursuit course against the target. Trajectories were generated on

a one versus one basis, and then two such trajectories were merged to form a two-versus-two geometry. While one target remained almost directly in front of each attacker due to the lead pursuit course (in which the lead angle is generally less than 15 degrees off the nose), the other target, which was not being tracked by that attacker in lead pursuit, maneuvered independently of it. This independent target remains generally in front of the attacker for target trajectories one through five, simply due to the initial conditions and the duration of the encounter. For trajectory six, the target eventually moves behind attacker number one. Although two attackers exist and are estimating target states throughout the simulation, only the output from attacker number one is shown in the results. Early testing showed that both filters converged to essentially the same estimates after the initialization transient settled. The purpose of pointing this out here is merely to establish the following relationship between attacker and targets, that is, that the odd-numbered target trajectories correspond to targets being chased in lead pursuit, while even-numbered targets are maneuvering independently of the attacker.

The method of controlling the target maneuver was through piecewise constant commands of roll rate, load factor, and thrust to a five degree-of-freedom aerodynamic model of an F-4 aircraft. Moment equations were used in the pitch axis to prevent discontinuities in pitch rate, where tail surface deflection was generated in a feedback loop to achieve the commanded load factor. Thrust and roll rate were assumed to be instantly achieved. In the case of instant thrust, this produced somewhat unrealistic step changes in target acceleration that sometimes made the

target estimator's input more difficult to follow. This is not considered to invalidate the data; however, this fact must be considered when examining the results.

In addition to the trajectory generation and merging programs, a three-dimensional plotting program was developed to graph the trajectories. These graphs are shown in Appendix C for each of the three merged data files, each of which contains two target and attacker trajectories.

"SOFEPL".

"SOFEPL" is a statistical analysis and plotting post-processing program developed by University of Dayton Research Institute. Using a "SOFE" generated input file, it calculates time histories of averages of $\underline{x}_t$ and $\hat{\underline{x}}_f$, and time histories of averages and standard deviations of errors. Errors are defined as the differences between user-specified pairs of components of $\underline{x}_t$ and $\hat{\underline{x}}_f$. Of the many plot options available, the two which were used were the plot of X_{t_j} and $\hat{X}_{f_{j(aver)}}$, and the plot of average error, plus or minus one standard deviation, Avg $[X_{t_j} - \hat{X}_{f_j}] \pm \sigma_{\varepsilon_j}$. In addition, an option to plot average error, average error plus σ_{ε_j}, and $-(P_{f_j})^{1/2}$ was developed for test set 2.2.1.

Program Output

Printer Output.

An example of printer output is given in Figures 4 and 5. Figure 4 is an example of time history output of truth, (or "system", as called in "SOFE") states, filter estimated states, and filter estimated standard deviation. Since "SOFE" itself performs no ensemble averaging, all the printed output represents results from an individual run. Ensemble

THESIS FILTER 05/03/80 12.09.58. RUN NUMBER - 2

STATE VECTOR XSO AT T = .80000E-01

1.	59903.9	2.	-1199.70	3.	7.43709	4.	3200.00	5.	3.183518E-03
6.	-1.973498E-05	7.	9999.90	8.	.720688	9.	18.0172	10.	59903.9
11.	-1199.70	12.	7.43709	13.	-2800.00	14.	3.183518E-03	15.	-1.973498E-05
16.	9999.90	17.	.720688	18.	18.0172				

STATE VECTOR XFO AT T = .80000E-01

1.	59903.9	2.	-1199.70	3.	7.43709	4.	3200.00	5.	3.183518E-03
6.	-1.973498E-05	7.	9999.90	8.	.720688	9.	18.0172	10.	59903.9
11.	-1199.70	12.	7.43709	13.	-2800.00	14.	3.183518E-03	15.	-1.973498E-05
16.	9999.90	17.	.720688	18.	18.0172				

SIGMAS FOR XFO AT T = .80000E-01

1.	10000.0	2.	1000.00	3.	100.000	4.	10000.0	5.	1000.00
6.	100.000	7.	10000.0	8.	1000.00	9.	100.000	10.	10000.0
11.	1000.00	12.	100.000	13.	10000.0	14.	1000.00	15.	100.000
16.	10000.0	17.	1000.00	18.	100.000				

STATE VECTOR XS AT T = 6.0000

1.	45439.9	2.	-1224.52	3.	6.08204	4.	3320.79	5.	238.272
6.	63.7421	7.	8631.49	8.	-5.61979	9.	-8.60841	10.	45433.6
11.	-1224.52	12.	6.08204	13.	-2122.11	14.	238.272	15.	63.7421
16.	8608.86	17.	-5.61979	18.	-8.60841				

STATE VECTOR XF AT T = 6.0000

1.	45410.7	2.	-1204.89	3.	15.1313	4.	3305.79	5.	223.653
6.	59.4747	7.	8647.41	8.	33.3100	9.	15.7335	10.	45543.2
11.	-1157.51	12.	26.0234	13.	-2142.52	14.	221.480	15.	59.0117
16.	8643.32	17.	33.6799	18.	11.4087				

SIGMAS FOR XF AT T = 6.0000

1.	129.052	2.	59.5187	3.	25.3032	4.	15.6532	5.	18.3587
6.	18.5801	7.	27.6308	8.	20.3826	9.	18.8159	10.	129.774
11.	59.4272	12.	25.0851	13.	14.3013	14.	17.8739	15.	18.5220
16.	27.7503	17.	20.4024	18.	18.8138				

STATE VECTOR XS AT T = 12.000

1.	30233.2	2.	-1125.05	3.	17.6488	4.	5220.33	5.	570.591
6.	23.8813	7.	6085.23	8.	-156.558	9.	-50.1590	10.	30199.5
11.	-1125.05	12.	17.6488	13.	-19.1450	14.	570.591	15.	23.8813
16.	6232.65	17.	-156.558	18.	-50.1580				

STATE VECTOR XF AT T = 12.000

1.	30310.3	2.	-1085.42	3.	29.6395	4.	5232.45	5.	590.514
6.	48.9350	7.	6112.40	8.	-125.995	9.	-21.9553	10.	30246.1
11.	-1093.91	12.	29.6732	13.	.167892	14.	598.838	15.	53.9019
16.	6234.97	17.	-144.241	18.	-29.6230				

SIGMAS FOR XF AT T = 12.000

1.	68.4941	2.	39.1436	3.	23.0373	4.	14.7137	5.	16.6439
6.	17.7272	7.	16.9987	8.	16.4472	9.	18.0030	10.	71.3533
11.	40.0432	12.	23.0407	13.	9.72817	14.	14.6853	15.	17.4957
16.	17.4611	17.	16.5713	18.	17.9833				

RUN NUMBER 2 COMPLETE AT T = 15.00000

Figure 4. Sample "SOFE" Printed Output

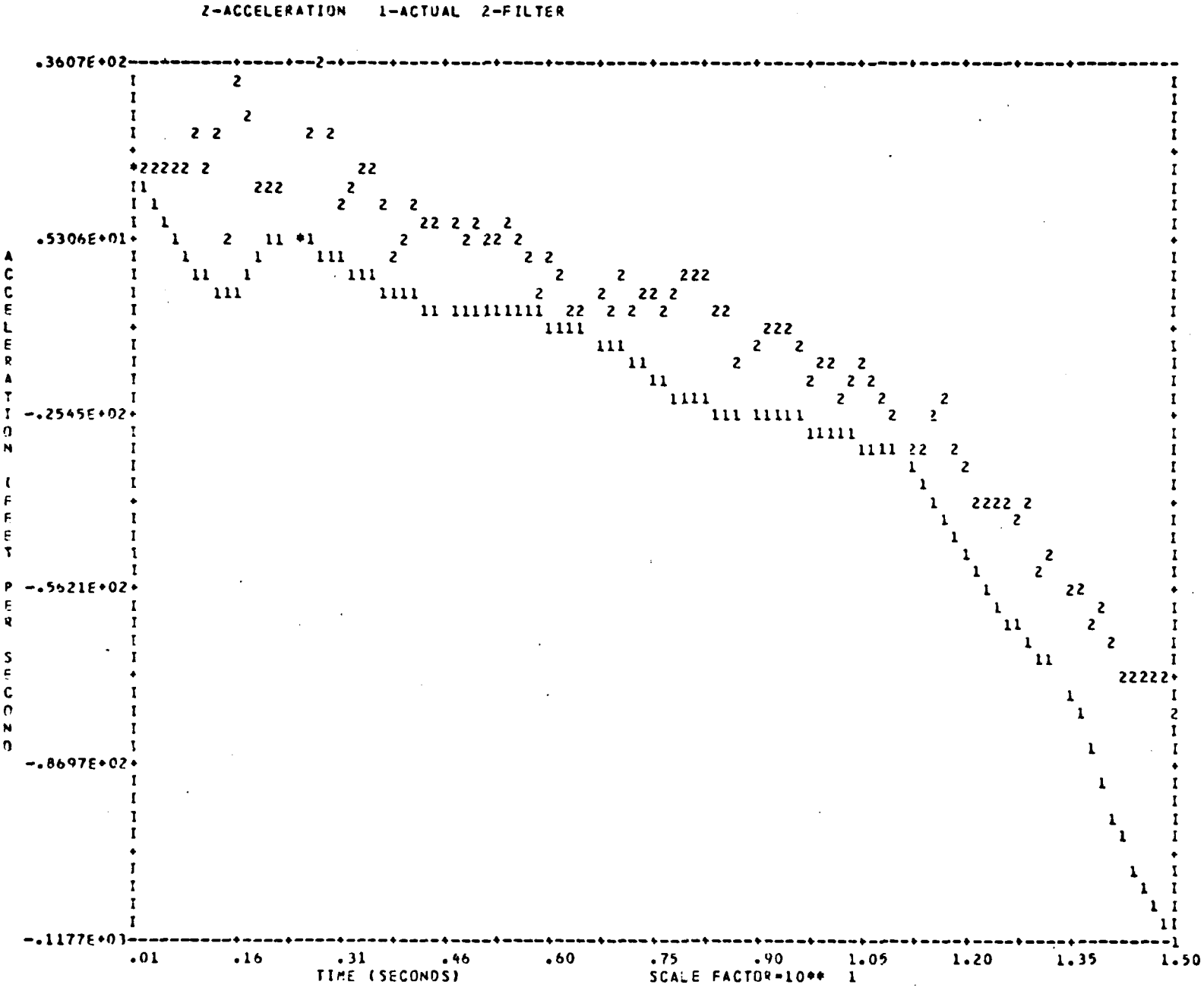


Figure 5. Sample "SOFE" Printer Plot

averaging is performed by "SOFEPL". All of the eighteen filter states, nine for each of two filters, are printed out here, even though plots were generated only for filter one, which corresponds to the first nine states. Figure 5 shows one of the printer plots generated by "SOFE". The data used in these plots is only from the first Monte-Carlo run. The example given is of the filter error and filter-estimated standard deviation, as distinguished from the actual Monte-Carlo standard deviation.

Plotted Outputs.

Numerous examples of plotted output are referenced in Chapter Five on test results. For the full state plots, titled "TRUE AND EST. VALUES", the average estimates can be distinguished from the true state by the estimation noise appearing on the estimate curve. By contrast, the true value time history, which is identical from one Monte-Carlo replication to the next, appears as a smooth curve. The error plots, titled "ERROR, ERROR +- ONE SIGMA", contain three curves showing the average error, and the average error, plus and minus the computed standard deviation. Beginning with the plots that are labeled "CASE36" and thereafter and also on "CASE15" and "CASE16" plots, the bottom curve is the negative square root of the average filter-estimated covariance, i.e., $-Av[P_f(j, j)^{1/2}]$. This allows better insight into accuracy of the filter in its own estimate of its errors, by allowing the reader to compare the filter's estimate of standard deviation with the actual Monte-Carlo standard deviation.

V. Test Results

The test conditions described in Chapter Three were simulated utilizing the software described in Chapter Four. This chapter describes the results of those tests, making specific and general observations and inferences. The paragraph numbering follows that of Chapter Three to aid the reader in cross-referencing test objectives and results. There are a number of tables of statistical data referenced throughout this chapter. They contain means and standard deviations of the errors committed by the filter in each of the nine filter states. They are computed in the post-processing program, "SOFEPL" by first generating time histories of average error and standard deviation, and then averaging each of those curves over the entire time interval. Since this interval does include the filter initialization, the time averages generally represent worse errors than the steady-state errors. Nonetheless, for a given target trajectory, the statistics of any two given test cases can be compared to indicate relative performance. To understand the exact conditions of a particular test case, it is necessary to refer to Table 2.

The plots referred to in this chapter are contained in Appendix F, and are labeled with a set of sequential test case numbers, as well as decimal test case numbers from Chapter Three. Since the plotting program scales each graph independently, the vertical scales are not necessarily the same for the same plot on different test cases.

1. Single Filter Baseline

1.1 Target Acceleration Model (Ref Pages F-(3-44))

Target trajectories one (a short-range, turning engagement) and four (a long-range head-on pass) were used to evaluate the performance of the turning target acceleration model relative to the time-correlated acceleration model. These trajectories were chosen because they were significantly different, and would present the filter with a broad range of conditions. On trajectory one, the turning acceleration model produced standard deviations of all the state errors that were 5% to 15% lower than the time-correlated model. See Table 5-1, cases 1.1.1 and 1.1.2. More significantly, the bias errors were considerably less for those axes in which a significant acceleration occurred. The bias errors for both models were worse in acceleration than velocity, and worse in velocity than position, but the acceleration biases for the turning model was approxmately 8% in X, 67% in Y, and 110% in Z, of the time-correlated bias. The Z-axis had very little acceleration resulting in small biases for both filters.

On trajectory four (cases 1.1.3 and 1.1.4), the target is much further away than on trajectory one, (60,000 feet vs. 8000 feet) resulting in much larger azimuth and elevation measurement errors (in feet) since the errors are a constant 4 milliradians, RMS. This results in larger estimation errors in the Y and Z axes than on trajectory one. In addition to the longer range, the target performs a significant out-of-plane maneuver, i.e., by rolling the aircraft, it rolls the acceleration vector from the inertial Y-axis to the inertial Z-axis. This is a significant test for the turning acceleration model, because its propagation equations can only account for the in-plane turn. The test results show

TABLE 5-1
TEST OF TIME-CORRELATED AND TURNING ACCELERATION MODEL
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
1.1.1	-3.01	-10.90	-16.20	1.35	6.70	12.50	-.49	-2.10	-1.89	M
1	10.70	21.30	13.80	9.35	19.20	11.90	10.15	21.20	13.40	S
1.1.2	-.52	-2.24	-1.24	.91	4.61	8.30	-.48	-2.13	-2.06	M
2	10.20	19.40	12.90	8.84	16.90	9.84	9.56	18.60	11.00	S
1.1.3	.56	4.15	3.68	12.60	7.84	13.10	-8.60	-19.80	-18.00	M
3	14.40	25.00	14.00	47.20	56.20	14.20	42.80	53.80	23.60	S
1.1.4	1.36	4.05	.14	3.41	4.35	-.20	-5.05	-19.40	-13.30	M
4	14.20	22.60	11.80	47.80	56.60	15.10	47.10	59.30	17.20	S

significantly lower mean errors in Y-axis position and acceleration, and nearly equal or slightly lower mean errors in other estimated states for the turning acceleration model, as compared to the time-correlated model. There were no significant differences in standard deviations. Based on the tests run in this set, it was concluded that the turning acceleration model out-performed the time-correlated acceleration model, especially in the reduction of bias errors on acceleration states.

1.2 Performance of Turning Acceleration Filter on Different Trajectories (Ref. Pages F-(45-53))

On trajectory two (test case 1.2.1), the target performs very dynamic maneuvers by reversing his turning plane. The acceleration estimates appear to oscillate or initially diverge before convergence begins at about two or three seconds. (Page F-47) A sharp roll reversal by the target at four seconds creates a large dynamic lag in estimated acceleration from five seconds to nine seconds. At 14 seconds, the target maneuver becomes unrealistic because excessive load factor is causing a rapid loss of energy and airspeed. This trajectory is not used hereafter for that reason. (Table 5-2)

On trajectory three (test case 1.2.2) the target flies south, straight and level at 15000 feet altitude, slowly increasing speed from 1200 to 1320 feet per second. Except for some unusually large initialization transients in X and Y velocity estimates, the filter performed well, having essentially equal standard deviation of errors as for trajectory four (test case 1.1.4), and lower mean error, due to the lack of target maneuver.

Trajectory five (case test 1.2.3) is another straight and level target, but flying east at 90 degrees to the attacker's initial velocity.

TABLE 5-2
TEST OF TURNING ACCELERATION MODEL ON OTHER TRAJECTORIES
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
1.2.1			3.00			-12.90			-3.82	M
5			13.20			10.10			11.40	S
1.2.2	-.45	.72	-.94	4.49	-7.93	.56	-.73	-2.34	-.56	M
6	13.40	22.40	12.50	47.20	55.90	16.20	42.10	53.20	14.10	S
1.2.3	-.15	-1.21	-.15	-.34	1.04	.93	-.23	-3.65	-1.19	M
7	9.94	18.80	12.00	9.95	19.80	10.30	10.30	19.80	11.60	S
1.2.4	.96	-1.45	-2.43	-2.22	-5.18	-7.49	-.81	-6.45	-4.68	M
8	16.40	22.20	11.70	16.20	25.40	12.10	14.50	23.30	12.10	S

Performance is similar in that the mean errors are small, but the standard deviations are lower due to the shorter range.

Test Case 1.2.4 was conducted on trajectory six, in which the target initially is flying west in a climb, rolls left and pulls a three "g" turn and then rolls level and continues to climb. A three "g" transient in X-acceleration during the first three seconds, while the filter is still settling, generates a large bias error in that state until 4.5 seconds into the simulation. Another target maneuver at 8 seconds induces a bias that lasts until 12 seconds in both X and Y axis acceleration estimates. After examining the Y and Z axis true accelerations, the target maneuvers appear unrealistically too dynamic. The maneuvers in Z axis acceleration appear to be high enough frequency that the filter estimate seems to be 180 degrees out of phase. The overall observation from test cases 1.2.1 through 1.2.4 is that the filter's performance was consistent with runs 1.1.1 through 1.1.4 on trajectories one and four. The similarity of statistics for these cases, as compared to trajectories one and four, indicate that further investigation can reasonably be carried out using only trajectories one and four. It is important in order to control the overall number of runs that this reduction in number of trajectories be made if a performance indication is to be generated with a reaonsable total number of runs.

1.3 Performance of Single Fiter with Angle-only Track (Test Cases 1.3.1 - 1.3.4) (Ref. Pages F-(56-67))

Cases 1.3.1 through 1.3.4 were run only for comparison with the cooperative filter runs described in the next section. A single aircraft that is tracking a target with an angle-only tracker cannot observe

TABLE 5-3
TEST OF ANGLE-ONLY MEASUREMENTS WITH SINGLE FILTER
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
1.3.1	-2934.00	307.00	-52.90	-215.00	-46.70	4.34	-.28	-.78	-3.54	M
9	2229.00	194.00	32.90	197.00	65.30	18.10	5.67	11.50	10.70	S
1.3.2	-6500.00	24.90	-9.14	-729.00	-95.70	-12.90	-1241.00	-13.00	-8.64	M
10	2692.00	195.00	19.20	292.00	51.90	11.00	512.00	55.60	12.00	S
1.3.3										M
11										S
1.3.4	-3492.00	-264.00	-19.50	-504.00	-102.00	-15.60	-681.00	-58.50	-7.81	M
12	11666.00	498.00	82.10	1290.00	186.00	37.30	2220.00	127.00	29.20	S

range to the target. This affects all of the other estimates, since an estimate of range is necessary to relate angular measurements to linear states. Equation (2-26) shows how estimated range enters the H matrix, which is required to process the angle measurements.

For both case 1.3.1 and 1.3.2, the filter was initialized to the true state, although the initial filter covariance was high so that an initialization transient still occurred. Not surprisingly, the performance of X axis estimates was considerably worse than Y or Z, since the slant range was primarily in the X axis direction. (See Table 5-3). For the short period of time simulated, there was some tendency for X-axis position and velocity to follow the true values, but whether this was convergent or actually a divergent oscillation could not be determined without a run of longer duration. The Y and Z axis estimates tended to diverge slowly but steadily from the true values. Numerical problems resulting from the lack of range information forced the simulations to be stopped short of the full trajectory length before the covariance matrix $P(t_i)$ became indefinite.

Test Cases 1.3.3 and 1.3.4 were like 1.3.1 and 1.3.2 except that random initial errors were used. Case 1.3.3 was never successfully run because the P matrix became indefinite shortly after starting the run. This may be largely due to the fact that the initial P matrix induced numerical errors large enough to produce such large variations in $\hat{\underline{x}}(t)$ during the initial transient, that the target was estimated as behind the attacker, so that fluctuations in the H matrix were too large. Case 1.3.4 was successfully run and, as expected, showed worse errors than case 1.3.2.

The result from test set 1.3 was that, as expected, a single tracker making angle-only measurements available to a Kalman filter results in filter divergence. This was expected due to the mathematical observability problem.

The overall result from test set 1. was that the turning acceleration model was selected as superior to the time-correlated acceleration model, that this filter performed satisfactorally on a variety of target trajectories, and that angle-only information from a single tracker is insufficient to estimate the complete target state.

2. Cooperative Filter Comparison

2.1 Direct Measurement Transfer (DMT)

2.1.1 DMT Benchmark, Complete Data Transfer, Radar Parameters (Ref. Pages F-(68-79))

Test Cases 2.1.1.1 and 2.1.1.2 were run using perfect initialization, trajectories 1 and 4, and a data transfer rate of 25 time per second. Since this is also the basic measurement rate, this means that all of the measured data was transmitted to the other filter, as well as being used on the original filter, every sample time. The outputs from these two runs should be compared with cases 1.1.2 and 1.1.4 to determine the degree of improvement over single filter performance. The outputs will also be used later for comparison with DMT runs made at a lower transfer rate, and for comparison with the state transfer technique evaluated in subsection 2.2 of this section.

Using Table 5-4, it can be seen that the standard deviations improvement over the single filter performance ranged from approximately a 30% improvement in position to no improvement in acceleration esti-

TABLE 5-4
COMPARISON OF SINGLE AND DUAL FILTERS WITH RANGE AND ANGLE
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
1.1.2	-.52	-2.24	-1.24	.91	4.61	8.30	-.48	-2.13	-2.06	M
2	10.20	19.40	12.90	8.84	15.90	9.84	9.56	18.60	11.00	S
1.1.4	1.36	4.05	.14	3.41	4.35	-.20	-5.05	-19.40	-13.30	M
4	14.20	22.60	11.80	47.80	56.60	15.10	47.10	59.30	17.20	S
2.1.1.1	-.28	-1.39	-1.01	1.04	4.84	7.02	-.13	-1.37	-2.12	M
13	7.22	15.70	12.10	7.37	15.60	10.40	7.43	15.80	11.00	S
2.1.1.2	.65	2.35	.68	1.15	-2.00	-1.02	-5.87	-12.90	-15.30	M
14	9.49	17.60	12.10	34.70	47.60	15.40	33.70	47.00	14.50	S

mates. This was true for both trajectories one and four. This also indicates that while increasing the number of samples at a given sample rate of a position measurement (still maintaining a white noise sequence corruptive noise assumption) will improve the position estimates, the acceleration estimation accuracy is limited for other reasons. Note also that the 30% improvement in standard deviation (i.e., reduction to approximately 70%) correlates well with the 50% reduction in variance expected when doubling the number of equal accuracy, independent measurements. (The variance equals the standard deviation squared.) The result of this set of runs was that data transfer significantly improved position estimates, somewhat improved velocity estimates, but made no improvement in acceleration estimates.

2.1.2 DMT Benchmark, Complete Data Transfer, Angle-only Measurements (Ref. Pages D-(80-112))

While for range and angle tracking, the improvement in estimation accuracy as shown by the cooperative filters was primarily a matter of degree, the potential improvement in angle-only target estimation should be considerable. Section 1.3 established the performance of a single filter with angle-only measurements, showing the divergence of estimates when no range information is available. Test case set 2.1.2 establishes the benchmark for angle-only measurement transfer in that all runs in this set interchanged all measurements taken at each sample time. Tests were run with no initial errors, random initial errors, and constant initial errors, but no significant differences were noted as a function of initial errors (See Table 5-5). The convergence of X-axis estimates was always noticeably slower than Y or Z, since the X-axis

TABLE 5-5
PERFORMANCE OF ANGLE-ONLY DUAL FILTER AT 25 HZ.
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.2.1	.02	-.80	-1.94	.37	4.07	5.92	-.04	5.76	-1.17	M
15	6.74	13.40	10.70	2.48	6.64	7.90	2.21	6.29	3.22	S
2.1.2.2	-25.90	4.43	-10.90	-.31	-.73	.01	-6.70	-5.97	-13.00	M
16	125.00	104.00	23.40	12.00	22.70	12.20	25.40	29.00	12.40	S
2.1.2.3	-.45	-2.00	-4.98	.46	4.34	9.82	-.01	-.28	.61	M
17	7.89	25.90	23.50	3.14	13.30	20.20	2.75	11.90	20.90	S
2.1.2.4	-32.30	-11.10	-13.60	.09	1.68	2.16	-7.24	-6.53	-10.40	M
18	131.00	96.60	36.20	11.90	20.50	17.10	26.20	26.20	17.10	S
2.1.2.5	.36	-1.14	-.68	.33	4.11	7.93	-.02	-.28	.17	M
19	6.69	13.40	10.70	2.48	6.69	7.98	2.21	6.31	3.23	S
2.1.2.6	-14.90	-41.60	-2.67	1.48	15.80	-5.38	-3.07	-9.93	-11.30	M
20										S

was initially aligned (trajectory 1) or nearly aligned (trajectory 4) with the initial range vector. Nevertheless, the X position error converged to under 0.5% of range within the first second of data transfer due to the triangulation effectively being performed. It should be noted that both filters were starting data transfer from the initialization point; that is, each filter started to receive the alternate tracker's measurements at the same time as its own. Since the steady-state errors are always less than the errors during initialization, the table values (averaged over the entire run) are higher than for steady-state. This is especially true for these angle-only cases, which require a longer initialization time, so that the table values should not be thought of as the steady-state performance.

Table 5-6 presents a comparison of "radar" and "IR" type measurement transfer at 25 Hz. The larger errors in X-axis estimates for case 2.1.2.2 result from initialization errors, and do not represent steady-state results. By comparing the corresponding graphs, it can be seen that even without a range measurement, the filter using "IR"-type measurements, with information transfer, outperformed the radar filter on trajectory one, even in the X-axis. Comparing results from 2.1.1.2 with 2.1.2.2 ("radar" vs. "IR" on trajectory four), the "IR" configuration performs better than the "radar" in "Y" and "Z" axes, though not in "X". In this instance, the "radar" range measurement is unrealistically accurate ($\pm$30 feet, 1 sigma) at a range of 60,000 feet.

2.1.3 DMT at Lower Transfer Rates: Angle-only with no Initial Errors (Ref. Pages D-(113-136))

As Table (5-7) shows, the estimator had very low errors on

TABLE 5-6
COMPARISON OF DUAL "RADAR" AND "IRST" FILTERS AT 25 HZ.
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.1.1	-.28	-1.39	-1.01	1.04	4.84	7.02	-.13	-1.37	-2.12	M
13	7.22	15.70	12.10	7.37	15.60	10.40	7.43	15.80	11.00	S
2.1.1.2	.65	2.35	.68	1.15	-2.00	-1.02	-5.87	-12.90	-15.30	M
14	9.49	17.60	12.10	34.70	47.60	15.40	33.70	47.00	14.50	S
2.1.2.1	.02	-.80	-1.94	.37	4.07	5.92	-.04	5.76	-1.17	M
15	6.74	13.40	10.70	2.48	6.64	7.90	2.21	6.29	3.22	S
2.1.2.2	-25.90	4.43	-10.90	-.31	-.73	.01	-6.70	-5.97	-13.00	M
16	125.00	104.00	23.40	12.00	22.70	12.20	25.40	29.00	12.40	S

TABLE 5-7
COMPARISON OF ANGLE-ONLY DUAL FILTERS AT 25, 5, & 1 HZ. & TRUE KO
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.2.1	.02	-.80	-1.94	.37	4.07	5.92	-.04	5.76	-1.17	M
15	6.74	13.40	10.70	2.48	6.64	7.90	2.21	6.29	8.22	S
2.1.2.2	-25.90	4.43	-10.90	-.31	-.73	.01	-6.70	-5.97	-13.00	M
16	125.00	104.00	23.40	12.00	22.70	12.20	25.40	29.00	12.40	S
2.1.3.1	-.04	-1.04	-.45	.34	4.07	5.31	-.17	-.69	-1.60	M
21	12.20	22.00	13.90	3.00	7.41	7.91	2.74	7.43	8.44	S
2.1.3.2	4.41	-5.64	-12.50	.95	1.08	.96	-1.01	-9.72	-9.48	M
22	223.00	145.00	26.80	19.90	25.80	12.80	44.50	40.30	12.70	S
2.1.3.3	-2.32	-2.35	-1.19	.15	4.32	5.68	0.00	-.82	-.56	M
23	27.60	28.90	13.90	5.21	9.58	8.97	2.82	7.31	8.05	S
2.1.3.4	-32.60	-23.90	-12.10	-4.06	-1.47	.58	-8.40	-13.70	-9.25	M
24	398.00	203.00	28.90	34.40	31.70	13.80	75.70	49.90	13.20	S

trajectory 1 (case 2.1.3.1) with 0.2 second transfers. The plot of X-position error illustrates an interesting aspect of the filter performance. For approximately the first eight to ten seconds, the standard deviation can be seen to be increasing rapidly for a 0.2 second interval, only to decrease abruptly when the transferred data is received. This is to be expected for the angle-only case, since the initially incorrect range estimate affects all other estimates. This abrupt up-and-down behavior gradually diminishes until it is no longer perceptible at 12 seconds. In fact, at 12 seconds the standard deviation ($\tilde{}5'$) is the same as on the 25 Hz transfer rate figure. This indicates that while the higher transfer rate aids the filter more during initialization, the lower transfer rate is adequate thereafter.

On trajectory 4 (case 2.1.3.2), errors were considerably higher than for trajectory 1, which was closer in range and had a more favorable geometry for triangulation, and were also higher than for case 2.1.2.2. The steady-state standard deviations were roughly twice the values of case 2.1.2.2. While of relatively poorer quality, the information would still have been good enough to be tactically useful, having range errors under 1% of range.

These two cases were repeated with the transfer rate reduced to 1.0 Hz on cases 2.1.3.3 and 2.1.3.4. A similar behavior involving rapidly increasing and abruptly decreasing errors occurred on trajectory 1 (case 2.1.3.3), but the 14 second trajectory was not long enough to be sure that initialization was complete, i.e., that the standard deviation was not changing appreciably in time. Initialization appeared complete in the X-axis, but the Y-axis position errors (Page F-128) were increasing

to higher values between 1 second transfers during the final 3 seconds, though this was possibly due to the target maneuver. This test case would have to be continued longer to determine the steady-state behavior. The same phenomenon occurred on trajectory 4 (case 2.1.3.4).

2.1.4 DMT at Lower Transfer Rates: Angle-only with Constant Initial Errors (Ref. Pages D-(137-145))

Trajectories 1 and 4 were again simulated, this time with fixed initial errors in position that corresponded to 1000 feet too far in range for trajectory 1 (actual range = 8000 feet), and 10,000 feet too close in range for trajectory 4 (actual range = 60,000 feet). These significant, but not extreme, errors exercise the filter's capability to recover from initial errors. Initial velocity estimate was set nearly to zero and acceleration was set to zero. Velocity was not set exactly to zero because of the ω_t calculation (Eq. 2-16). This represents the more realistic conditions in which such an estimator would be used. On both trajectories, the estimator recovered well, when data was transferred at 5 Hz, converging to under 1% error in range within 1 second. However, when data was transferred at only 1 Hz, the same type of behavior was evident as in Test Set 2.1.3. Due to the initial error and slow transfer rate, convergence was very slow (approximately 8 seconds) with large errors occurring during the initialization transient phase. Overall statistics are contained in Table 5-8.

Tests 2.1.3 and 2.1.4 showed that good performance of angle-only direct measurement transfer occurred at significantly lower rates than the 25 Hz basic measurement rate. Although the initialization transient phase lasted somewhat longer, range errors of approximately 0.5% were

TABLE 5-8
COMPARISON OF ANGLE-ONLY DUAL FILTERS AT 25, 5, & 1 HZ. & IMPERFECT XO
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.2.5	.36	-1.14	-.68	.33	4.11	7.93	-.02	-.28	.17	M
19	6.69	13.40	10.70	2.48	6.69	7.98	2.21	6.31	3.23	S
2.1.2.6	-14.90	-41.60	-2.67	1.48	15.80	-5.38	-3.07	-9.93	-11.30	M
20										S
2.1.4.1	1.37	-.21	2.71	.32	4.25	3.02	-.14	-.45	-.17	M
25	12.20	21.90	13.80	3.00	7.23	7.98	2.74	7.20	3.43	S
2.1.4.2										M
26										S
2.1.4.3										M
27										S
2.1.4.4	-63.00	-284.00	58.90	.67	-8.13	8.75	-14.40	-57.20	5.04	M
28	485.00	328.00	53.50	39.60	36.20	15.00	92.10	71.40	18.20	S

achieved on trajectory four, and approximately 0.1% on trajectory one, with a 5.0 Hz transfer rate. For the trajectories and time durations used, a transfer rate as low as 1.0 Hz was used successfully, but the general success of a rate this low should be verified with more complete test conditions.

2.1.5 DMT - Effect of Imperfect H Matrix (Ref Pages F-(146-149))

All of the previous cases in section 2.1 had the measurement sensitivity, $H(\underline{x}(t),t)$ evaluated at $\underline{x}(t) = \hat{\underline{x}}(t)$, so that estimation errors indirectly affected the measurement processing by introducing errors in H. Since the errors can be quite large, at least initially, in $\hat{\underline{X}}(t_o)$, this effect was separately investigated. This was done by making a series of runs with $H(\underline{x}(t),t)$ evaluated at $\underline{x}(t) = \underline{x}_t(t)$, the true state, for comparison with the previous runs. All runs were made at the full 25 Hz transfer rate, with both random and constant errors in initial conditions. Comparing values in Table 5-9 for case 2.1.5.1 with 2.1.2.3 and for 2.1.5.2 with 2.1.2.4 indicates no significant differences, but that the standard deviations computed with a "perfect" H matrix were sometimes slightly larger. For constant initial errors (cases 2.1.5.3 and 2.1.5.4), there was also no improvement. See Table 5-10.

The conclusion of this test is that errors in the H matrix caused by the erroneous initial $\hat{\underline{X}}(t_o)$ either did not cause any detectable amount of estimation error, or possibly, that the "error" introduced into H by using $\hat{\underline{x}}(t)$ reduces, rather than compounds the estimation error.

2.2.1 State Transfer - Perfect $\hat{\underline{X}}_o$ (Ref. Pages F-(152-192))

A large number of conditions were simulated to determine the performance of a filter which transferred state estimates as pseudo-

TABLE 5-9
ANGLE-ONLY DUAL FILTERS AT 25 HZ. WITH RANDOM XO, H(XF) & H(XTRJE)
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.2.3	-.45	-2.00	-4.98	.46	4.34	9.82	-.01	-.28	.61	M
17	7.89	25.90	23.50	3.14	13.30	20.20	2.75	11.90	20.90	S
2.1.2.4	-32.30	-11.10	-13.60	.09	1.68	2.16	-7.24	-6.53	-10.40	M
18	131.00	96.60	36.20	11.90	20.50	17.10	26.20	26.20	17.10	S
2.1.5.1	-.79	-2.34	-7.15	.51	4.59	5.95	.08	.21	.70	M
29	8.61	25.90	24.90	3.52	14.80	25.80	3.33	15.50	23.10	S
2.1.5.2	-31.50	-10.60	-13.60	.10	1.79	2.01	-7.12	-6.32	-10.60	M
30	130.00	85.40	36.00	11.90	20.70	17.20	26.10	26.20	17.00	S

TABLE 5-10
ANGLE-ONLY DUAL FILTERS AT 25 HZ. WITH IMPERFECT XO, H(XF) & H(XTRUE)
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.2.5	.36	-1.14	-.68	.33	4.11	7.93	-.02	-.28	.17	M
19	6.69	13.40	10.70	2.48	6.69	7.98	2.21	6.31	3.23	S
2.1.2.6	-14.90	-41.60	-2.67	1.48	15.80	-5.38	-3.07	-9.93	-11.30	M
20										S
2.1.5.3	.17	2.05	-8.80	.33	3.62	5.52	-.02	-.28	.19	M
31	6.74	13.40	10.70	2.48	6.64	7.89	2.20	6.29	3.23	S
2.1.5.4	-44.90	-8.56	-10.70	-.11	4.97	-5.12	-8.87	-6.10	-12.60	M
32	128.00	108.00	24.10	12.10	25.30	12.20	25.60	28.40	12.40	S

measurements (Tables 5-11 through 5-13). Three different levels of transfer (position only, position and velocity, or position, velocity and acceleration) were simulated and three different transfer rates (5., 1., and 0.2 Hz). Also simulated were state updates only, and both state and covariance update. These runs should be compared with cases 2.1.2.2, 2.1.3.2 and 2.1.3.4. Examining the results of case 2.2.1.1, which was position only transfer at 5.0 Hz, it is evident that the filter estimates are divergent and unsatisfactory. In this case, the position estimates were transferred as though the errors in position estimates were uncorrelated from sample to sample, which was not true. This caused the updated filter covariance to become unrealistically low, which resulted in low gains for the original angle measurements and the subsequent divergence. Test Case 2.2.1.2 was run at a lower transfer rate of 1 per second, with improved results, since the uncorrelated assumption was more nearly valid. The plots (pages F-(157-162)) show a convergent trend, although the Y-axis position estimate exhibits a "see-saw" behavior, i.e., an abrupt decrease in error after a rapid increase, from 12. to 15. seconds, similar to the 1 per second direct measurement transfer test case 2.1.3. By comparing the values for case 2.2.1.2 with 2.1.3.4 (Table 5-14) which corresponds to the case of 1 second measurement transfers, it can be seen that the state transfer case had somewhat lower standard deviations, especially for position estimates. No significant difference in acceleration estimation errors was noted. By comparing case 2.2.1.2 with 2.1.3.2, which was direct measurement transfer at 5 per second, the statistics are closer than the previous case, although the state transfer at 1 per second tended to have higher

TABLE 5-11

POSITION STATE TRANSFER AT 5., 1., 0.2 HZ., WITH & WITHOUT "P" JPDATE

STATISTICAL DATA

TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.2.1.1	-519.00	-180.00	-55.70	-60.00	-35.50	-2.61	-113.00	-44.10	-19.10	M
33	1191.00	409.00	75.70	125.00	97.10	56.30	245.00	112.00	51.70	S
2.2.1.2	-32.00	-10.20	-11.80	-1.99	-3.84	.04	-8.57	-9.72	-14.40	M
34	209.00	136.00	25.20	19.20	26.30	12.20	40.40	35.70	12.00	S
2.2.1.3	-253.00	-80.10	-11.80	-24.60	-13.60	-1.44	-49.40	-23.40	-15.00	M
35	547.00	137.00	19.40	56.60	33.30	11.40	105.00	38.40	11.30	S
2.2.1.4	-77.10	-104.00	-21.40	-1.32	-20.30	-4.11	-15.10	-29.00	-15.40	M
36	737.00	145.00	17.10	74.60	29.00	10.00	139.00	39.50	9.72	S
2.2.1.5	-551.00	-83.10	-22.00	-34.30	-21.10	-4.18	-102.00	-27.60	-15.50	M
37	1358.00	168.00	17.80	134.00	37.30	10.00	258.00	45.80	10.60	S
2.2.1.6										M
38										S

TABLE 5-12
POSITION AND VELOCITY TRANSFER AT 3 RATES, WITH AND WITHOUT "P" UPDATE
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.2.1.7										M
39										S
2.2.1.8	-45.90	-3.97	-11.70	-2.74	-3.48	-.22	-12.20	-9.11	-13.70	M
40	219.00	113.00	25.10	21.40	24.90	12.00	43.70	32.00	11.40	S
2.2.1.9	-98.40	-19.50	-5.77	-11.80	-7.64	-.46	-20.00	-12.00	-13.80	M
41	507.00	106.00	27.20	52.40	31.10	12.10	97.00	33.60	12.50	S
2.2.1.10	-76.60	-12.20	-18.70	-2.13	-2.00	-.68	-15.90	-10.80	-16.00	M
42	723.00	175.00	14.60	73.90	28.60	10.50	137.00	40.80	9.52	S
2.2.1.11	-537.00	-53.70	-20.00	-33.00	-10.90	-2.05	-99.40	-21.30	-16.10	M
43	1309.00	212.00	18.20	130.00	39.40	10.70	249.00	50.30	10.40	S
2.2.1.12	-3301.00	-118.00	-22.60	279.00	-46.40	-5.62	-613.00	-40.90	-15.30	M
44	2141.00	181.00	19.10	223.00	47.50	11.10	407.00	49.20	13.90	S

TABLE 5-13

POSITION, VELOCITY, ACCELERATION TRANSFER AT 3 RATES, W/ & W/O "P" JPDATE

STATISTICAL DATA

TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #		X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.2.1.13											M
45											S
2.2.1.14		-59.90	-6.53	-11.40	-4.12	-4.00	-.32	-15.60	-10.50	-14.60	M
46		209.00	91.30	14.60	21.30	23.30	9.07	41.60	27.90	8.65	S
2.2.1.15											M
47											S
2.2.1.16											M
48											S
2.2.1.17		-517.00	-37.00	-26.90	-32.90	-10.70	-3.50	-95.60	-18.30	-17.90	M
49		1286.00	192.00	21.30	127.00	38.20	10.20	244.00	47.50	10.50	S
2.2.1.18											M
50											S

TABLE 5-14
COMPARISON OF MEASUREMENT AND STATE TRANSFER AT 5. & 1. HZ.
STATISTICAL DATA
TIME AVERAGE MEAN(M) & STD. DEV(S)

CASE #	X-POS.	X-VEL.	X-ACC.	Y-POS.	Y-VEL.	Y-ACC.	Z-POS.	Z-VEL.	Z-ACC.	
2.1.3.2	4.41	-5.64	-12.50	.95	1.08	.96	-1.01	-9.72	-9.48	M
22	223.00	145.00	26.80	19.90	26.80	12.80	44.50	40.30	12.70	S
2.1.3.4	-32.60	-23.90	-12.10	-4.06	-1.47	.58	-8.40	-13.70	-9.25	M
24	398.00	203.00	28.90	34.40	31.70	13.80	76.70	49.90	13.20	S
2.2.1.2	-32.00	-10.20	-11.80	-1.99	-3.84	.04	-8.57	-9.72	-14.40	M
34	209.00	136.00	25.20	19.20	26.30	12.20	40.40	35.70	12.00	S

lower standard deviations. When state transfer was simulated at a low transfer rate of 1 transfer every 5 seconds (case 2.2.1.3), position error converged very slowly, while Y-position error appeared to be slowly divergent. It appears that position state transfers as measurements could be useful, if care is taken in the choice of transfer rate.

Test cases 2.2.1.4 through 2.2.1.6 were run to evaluate the effect of not updating the covariance matrix, P, when state estimates were transferred. The normal covariance updates still occurred when angle measurements were incorporated. The results of these runs showed that the covariance converged so slowly that the position estimates were noisy due to the high Kalman gain, and significant errors developed in X-velocity (≈500 ft/sec at 14. seconds) (See Page F-168). Basically, although the filter performed better than the single-filter angle-only case (test set 1.3), its covariance was calculated the same as in test set 1.3, so that the filter treated its own angle measurements in the same way. Not knowing enough information about the target state (according to its covariance estimate), it was not able to use the angle measurements effectively. Based on these test cases, then, it appears that this approach (not updating P(t) when states are transferred) is not a good one when a basic observability problem exists with the single-tracker measurements, as it does with angle-only measurements.

Test cases 2.2.1.8 through 2.2.1.12 were run to determine the performance of state transfer of position and velocity, both with and without covariance updates corresponding to the state "measurements". Performance appeared satisfactory only for case 2.2.1.8, which had states and covariance updated once per second when the states were

transferred. The covariance converged fairly rapidly, reaching an X-position estimated deviation of 200 feet within 4. seconds. By the 14-second point, the average error was sometimes exceeding the estimated deviation, but this is probably due to the target maneuver at this point, rather than filter incipient divergence. (See pages F-(176-181)). Case 2.2.1.9 developed much larger errors between its 5 second updates, but appeared to remain stable. Test cases 2.2.1.10 through 2.2.1.12 (position and velocity transfer without covariance updating) did not perform well, having the same problem as 2.2.1.4 through 2.2.16, that is, estimated covariance values that were too large.

Finally, transfer of position, velocity and acceleration was tested only at the 1.0 Hz transfer rate (Table 5-13) in cases 2.2.1.14 and 2.2.1.17. When the covariance matrix was updated, the results were similar to 2.2.1.8, except that the acceleration estimates were more accurate (See pages F-(187-191)). When the covariance was not updated, per-performance was unsatisfactory due to divergence, as illustrated by the Y-position estimate (page F-193).

2.2.2 State Transfer with Errors in Initial Estimates

No cases were run in this category, due to the overall poorer performance of state transfers, as compared to measurement transfers. If state transfers are to be seriously considered, however, these cases will need to be run.

2.2.3 Simultaneous State and Measurement Transfer

One test case 2.2.3.1 was run with both states and measurements transferred at 1.0 Hz, and not updating "P" when states were transferred. As seen in the figures on pages F-(194-197), the

performance was poor. At one point, the X-axis velocity estimate was over 3000 feet per second in error. This case should be rerun with "P" updates.

VI. Conclusions and Recommendations

Numerous detailed conclusions concerning specific test results were drawn in Chapter Five. In this chapter, more general conclusions are given, which summarize the detailed conclusions, and which relate them more directly to the application for which this technology was originally considered. That was the tactical fighter aircraft operating in an air superiority role in cooperation with other aircraft in his immediate vicinity. In addition to the conclusions, recommendations for further study are made, both for further verification of the validity of this study, and for possible modifications worthy of investigation.

Conclusions

Basic Single Filter Structure.

There were two areas in which the basic single filter differed from the traditional development of a target estimator. First, the turning target acceleration model was used in lieu of the time-correlated target acceleration model. This proved to lower the bias errors in acceleration estimates, and was therefore used for all succeeding test cases. Secondly, direction cosines were assumed for the raw measurement, rather than Euler angles, for the reasons outlined in Chapter Two. Basically, these reasons were to avoid mathematical singularity and trigonometric computations. Although a direct comparison with Euler angles was never simulated, no problems were known to have occurred in the simulation runs because of the use of direction cosines.

Direct Measurement Transfer.

There were a number of general conclusions drawn from the test of direct measurement transfer. First, when range and angle measurements

were available and being transferred, there was a significant improvement (30%) in position estimates, a lesser improvement in velocity estimates (15%-20%), and virtually unchanged acceleration estimation errors. This degree of improvement might be worth the cost of implementation depending upon the mission and task in which the aircraft is involved, and its present ability to meet the requirements of that mission.

Secondly, good performance was achieved in the case of two aircraft making and transferring angle-only measurements, such as would be the case with an infrared search and track (IRST) set, or other passive electro-optical sensor. This is a vast improvement over the single filter performance, for which the filter is unable to estimate the complete target state, due to the absence of any range information. The filter was able, when transferring the data at 25. Hz, to estimate target range within 0.25% for trajectory number four, in which the target started at 60,000 feet and closed to 22,000 feet. In this geometry, the attackers were initially 6,000 feet apart laterally, giving a 10 to 1 ratio of target range to attacker separation. It was able to obtain this level of accuracy within the first two seconds on this trajectory.

Thirdly, reducing the transfer rate of angle measurements increased the estimation errors and settling time, although the performance could still be considered tactically acceptable. With a transfer rate of 5 Hz, position estimation errors were roughly twice those obtained with a 25 Hz transfer rate, velocity estimation errors increased by roughly 50%, and acceleration estimation errors increased by roughtly 10% to 30%. These values may still be acceptable in many applications, where a reduced data transmission bandwidth is necessary or desired. When the

data transfer rate was further reduced from 5.0 to 1.0 Hz, the errors and settling time again increased, both roughly by a factor of two. These performance levels are dependent upon both the "Q" and "R" filter matrices.

Fourthly, no separate, ad-hoc start-up procedure was used, nor did one appear necessary or desireable to compensate for the lack of initial range. This was true even when extremely large errors in initial range estimate (e.g. error = 100% of true range) were simulated. Although these tests were run informally with only one simulation run, they showed that the filter would converge to the true values after a very large transient error had subsided. Although this did not cause a problem on the scientific computer used (floating point, 60 bit words), it would probably cause problems on computers with shorter wordlength, and certainly would cause problems on a fixed-point computer. Although this aspect was not thoroughly investigated in this thesis, it is mentioned here because such large errors might be expected in operational use.

State Estimate Transfer.

The test of transferring state estimates as pseudo-measurements produced a number of positive and negative results. There were a large number of configurations that had less than satisfactory performance. First, it was found that transferring states at 5 Hz as pseudo-measurements was too fast, because it grossly violated the assumption of uncorrelated errors between state estimates and "measurements", which were actually state estimates from the other filter. The estimated covariance dropped to unrealistically low levels, resulting in eventual filter divergence. Attempts at evaluating or approximating the cross correlations did not yield fruitful, useful results. If evaluated, such cross

correlations can be incorporated into the filter.

Secondly, transferring states without making the corresponding P matrix updates was unsatisfactory, at least for the angle-only measurement case that was simulated. In the angle-only case, the target state is unobservable from a single tracker, and the filter covariance does not converge if this is the only information of which it is aware. Thus, this conclusion might not necessarily be true for a range and angle measurement tracker.

Thirdly, when state transfer was performed at a low 0.2 Hz rate, the filter error experienced 5 seconds of significant error increase, followed by a large reset, which reduced the error. This type of performance may be unacceptable in many applications. However, the fact that the filter operated in an otherwise successful manner at 0.2 Hz, tends to ensure that the performance occurring at 1 Hz transfer rate was not overly sensitive to that rate.

Fourthly, state transfer at 1 Hz produced the best results of any configuration tested within the overall test on state transfer. It produced better performance than direct measurement transfer at 1 Hz, and performance comparable to direct measurement transfer at 5 Hz. Even though the transfer rate was lower than measurement transfer at 5 Hz, the number of data items transferred each time was higher, because a full-rank, symmetric covariance matrix (e.g., 3 by 3 yielding 6 unique elements for position-only transfer) had to be transferred. For measurement transfer, no covariance matrix would have to be transferred, since each aircraft would know the measurement error covariance of the other.

Fifthly, the effect of transferring velocity and/or acceleration estimates, in addition to position estimates, was to improve velocity

and/or acceleration estimates accordingly, without further improving position estimates. Since this increased both the amount of data to transfer and number of computations to perform, it seemed to provide a marginal improvement in return for the substantial increase in cost. This is especially true because both the amount of data to transfer and number of computations to perform increase quadratically as velocity and acceleration are added (e.g., 6 by 6 yielding 21 unique elements in P for position and velocity; 45 elements in P for adding acceleration, and the associated matrix inverses). Thus, transferring anything beyond position estimates seemed marginally efficient, except possibly where unusual requirements for velocity and acceleration exist.

Recommendations

Further Verification and Realism.

In order to verify further the simulation already performed through more realistic conditions, future investigation of this subject should include the following considerations.

1. The effects of errors in the relative positions and orientations of cooperating aircraft should be simulated. There are at least three different manners in which this could be done. The first approach would be simply to assume errors of a specified distribution. For example, the specified accuracy of the Global Positioning System (GPS) could be assumed for each aircraft independently, and inertial navigation system (INS) accuracy for platform tilt and wander angle could be used. The second approach would be more complicated and would entail the development of an estimator that would improve the independent estimates of ownship parameters through the use of tracking data provided by

the data transfer telemetry antenna. The third approach would combine the relative position estimation problem with the target estimation problem. There might be some advantage that would occur in this formulation, such as estimation of target tracker alignment errors, using the redundancy of the information obtained.

2. More elaborate measurement error modeling should be simulated. While this might not initially involve actual hardware-verified models, it should at least investigate the effects of bias errors, time-correlated errors, and state-dependent errors, such as radar glint, or atmospheric propagation anomalies.

3. The effects of time delays should be more thoroughly investigated, once some estimate of the delays in queing, transmission, decoding, etc., are known. The filter is not now structured to incorporate "old" data.

4. Other target trajectories and geometries should be evaluated, especially a much longer range geometry with comparatively close spacing on cooperating aircraft. The "long range" geometry simulated in this investigation had a range of 10 nautical miles, with a 10 to 1 ratio of target range to attacker separation. Ranges on the order of 50 to 100 miles should be simulated with different attacker separations, to obtain an upper bound on the range-to-separation ratio. This should be done in conjunction with number 1., above.

Possible Modifications.

In the case of simplification, several possibilities could be investigated. First, the exact partials of $\underline{f}(\underline{x})$, used in forming F(t), could be simplified by ignoring the partial derivatives of $\underline{\omega}_t$ with

respect to $\underline{V}_t$ and $\underline{A}_t$. This should have the effect of assuming that $\underline{\omega}_t$ is constant between measurement updates. $\underline{\omega}_t$ itself would still be recomputed after each measurement update, and would thus vary in time. Secondly, a faster, lower order numerical integration could probably be used between updates. Since the target driving disturbance is fairly large by comparison with the numerical integration error, there is little to be gained by a highly accurate integrator for this application. Thirdly, since the raw measurements are not extremely accurate, it is not likely that numerical problems would occur if a standard Kalman filter update were performed in lieu of the Carlson Square Root update now being simulated. A variety of update methods, including standard, Carlson, and U-D factorization, should be explored.

Extensions to the Form and Use of the Algorithm.

Two extensions to the present algorithm are presented conceptually here. The first is an extension to the form of the algorithm, in the target acceleration model. While the turning-target acceleration model capitalizes on the fact that a target tends to turn at a constant rate, at least for short periods of time, it does not capitalize on the fact that a target can change its acceleration less drastically along the velocity vector, than it can across the velocity vector. Mathematically, this would imply that the component of $\underline{w}(t)$ (the target driving disturbance) along $\underline{V}_t(t)$ should have a lower variance than the two components of $\underline{w}(t)$ perpendicular to $\underline{V}_t(t)$. Incorporation of this fact into the filter structure has the potential to decrease the estimation error in states that lie along the target's velocity. Note also that this would be an additional nonlinearity in the filter by explicitly making Q(t) a function of $\hat{\underline{V}}_t(t)$.

The second extension would be in the application of the algorithm, by applying it to air-to-ground target-tracking situations. For moving targets, it could be used to determine target motion without any range measurement, in the same way as for aerial targets. For stationary targets, its utility might lie more in the area of simultaneously updating target and friendlies' relative positions, and also the friendlies' absolute inertial coordinates. For this effect to occur, it would be necessary first to modify the filter according to the third approach given in recommendation number one.

Bibliography

1. Aden, Timmy C. "Inter-Laboratory Air-to-Air Missile Technology - An Innovative Approach", IEEE Proceedings of the National Aerospace Electronics Conference, NAECON '79, Dayton, Ohio, May 15-17, 1979, 606-614. Published by IEEE, New York, N.Y.

2. Aidala, V.J. "Kalman Filter Behavior in Bearings-Only Tracking Application", IEEE Transactions on Aerospace and Electronics Systems, January 1979.

3. Farrell, James et al. "Dynamic Scaling for Air-to-Air Tracking", IEEE Proceedings of the National Aerospace Electronics Conference, NAECON '75, Dayton, Ohio, May 1975, 157-162. Published by IEEE, New York, N.Y.

4. Fried, W. R. "Principles and Simulation of JTIDS Relative Navigation", IEEE Transactions on Aerospace and Electronics Systems, January 1978.

5. Kendrick, Jerry D. "Estimation of Aircraft Target Motion Using Pattern Recognition Orientation Measurements", Ph.D. Dissertation, Air Force Institute of Technology, December 1978.

6. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 1. New York: Academic Press, 1979.

7. Musick, Stanton H. SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation User's Manual. AFAL-TM-78-19, Revision A, Nov. 78. Wright-Patterson AFB, Ohio; Air Force Avionics Laboratory.

8. Pearson, John B. III, and Edwin B. Stear. "Kalman Filter Applications in Airborne Radar Tracking", IEEE Transactions on Aerospace and Electronics Systems, May 1974.

9. Womble, M. Edward and James E. Potter. "A Prefiltering Version of the Kalman Filter with New Numerical Integration Formulas for Ricatti Equations", IEEE Transactions on Automatic Control, June 1975.

10. Maybeck, Peter S. Course Notes for EE 6.80, Systems Modeling. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1975.

Appendix A

Statistics of Target Acceleration Models

The two target acceleration models considered in Chapter Two, the exponentially time-correlated Gauss-Markov, and the constant turning rate models, are described by the mean and covariance propagation models described below. Single-axis equations are used to illustrate the potential advantage of the constant turning rate model. When measurements are incorporated into a filter, they act as a feedback to these models from the actual environment. What is described in this appendix can be termed the open-loop acceleration estimator. As much accuracy as possible is desireable in these open-loop equations to allow the reduction of the Kalman gain through which noisy measurements are introduced. The use of the constant turning rate model did, in fact, allow such a reduction.

Time-Correlated Acceleration Model.

The propagation of a stochastic process variable, $\underline{x}(t)$, and its mean value, $\hat{\underline{x}}(t)$, can be developed from the following general equations:

$$\dot{\underline{x}} = F(t)\,\underline{x}(t) + w(t)$$

$$\dot{\bar{\underline{x}}}(t) = F(t)\,\bar{\underline{x}}(t)$$

where,

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

$F(t)$ = plant matrix

$w(t)$ = white noise

$\bar{\underline{x}}(t)$ = mean value of $\underline{x}(t)$

For the time-correlated model, this reduces to

$$\dot{A}(t) = -\lambda A(t)$$

where,

$\dot{A}(t)$ = acceleration mean value derivative

$A(t)$ = acceleration mean value

λ = reciprocal of correlation time

The propagation of the covariance can be developed from the following general matrix equation:

$$\dot{P}(t) = F(t)\ P(t) + P(t)\ F^{T}(t) + Q(t)$$

where,

$P(t)$ = covariance matrix

$F(t)$ = plant matrix

$Q(t)$ = strength of driving noise

For the time-correlated model (subscript "c"), this reduces to the scalar equation,

$$\dot{P}_c(t) = -2\lambda\ P_c(t) + Q_c$$

which is simply a damped first-order response to the white noise strength, Q_c. The accompanying figures (A-1, A-2) give the time response of the mean and covariance using the values of λ and Q_c used in the simulation. The parameters for the plots are

$\lambda = 0.2/\text{sec}$

$Q_c = 1000.\ \text{ft}^2/\text{sec}^5$

$A(t_o) = 100.\ \text{ft}/\text{sec}^2$

$P_c(t_o) = 0.\ \text{ft}^2/\text{sec}^4$

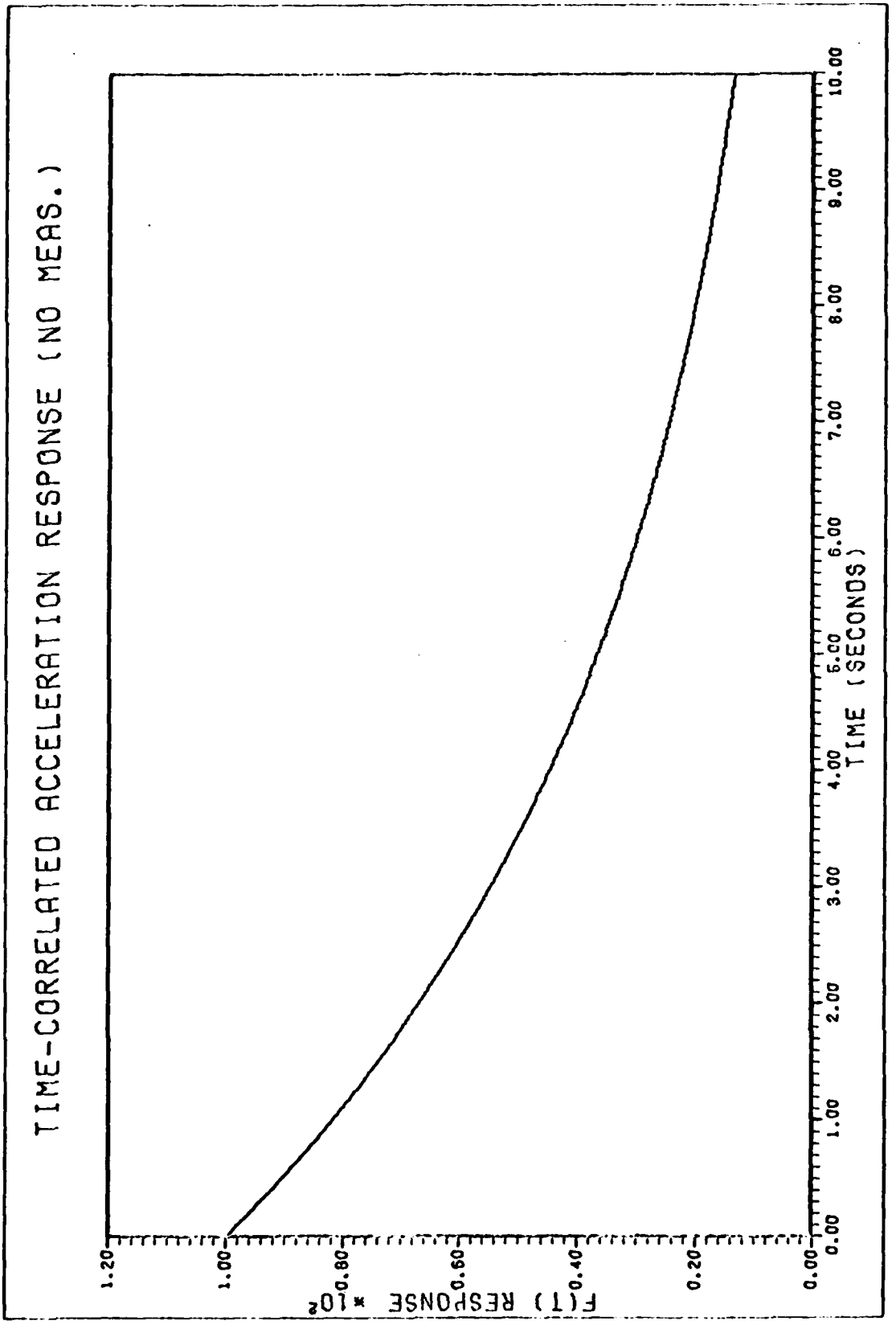


Figure A-1. Time-Correlated Acceleration Response

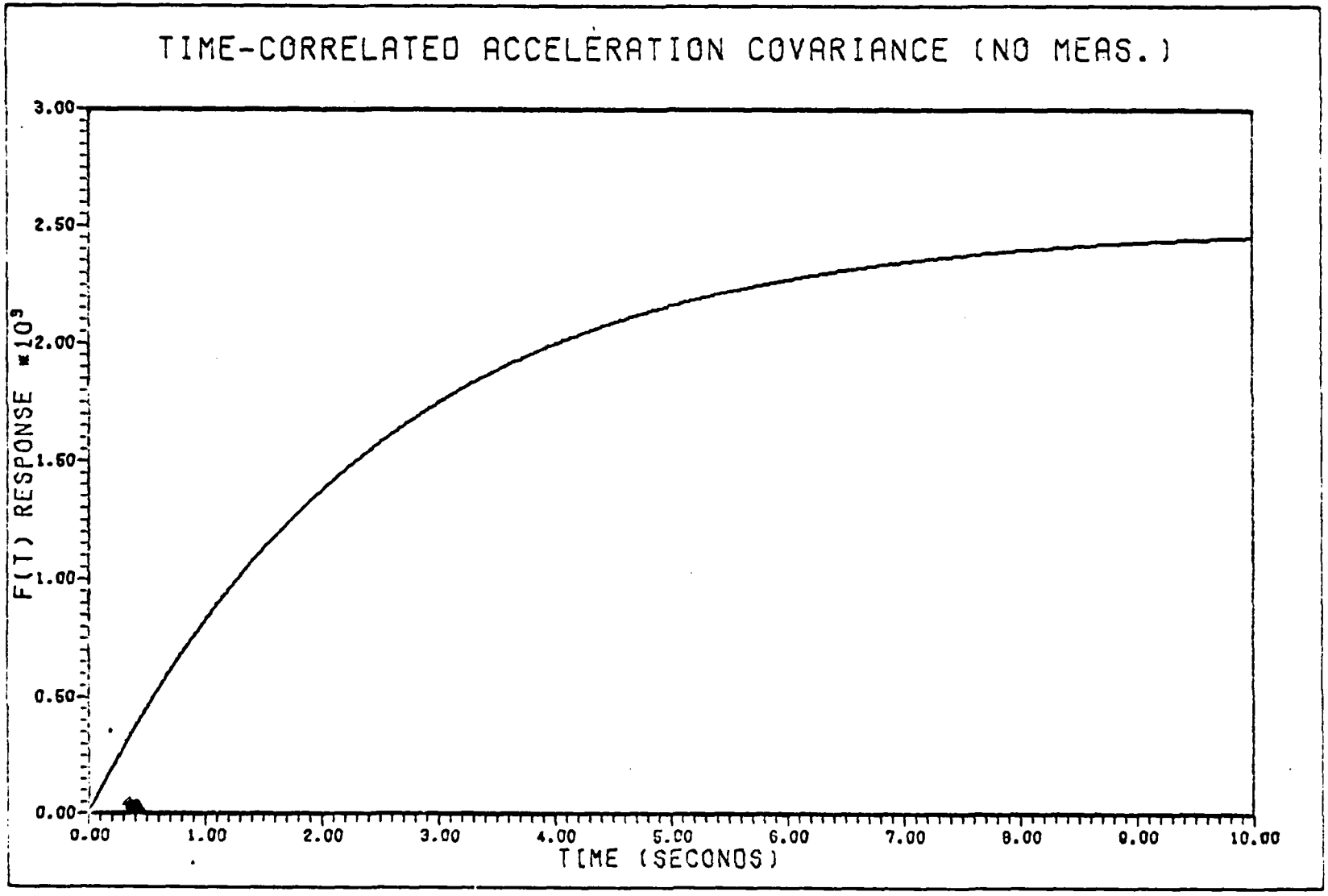


Figure A-2. Time-Correlated Acceleration Covariance Response

The solutions for the mean and covariance are

$$A(t) = A_o\, e^{-\lambda t}$$

$$P_c(t) = P_{c_o}\, e^{-2\lambda t} + \frac{Q_c}{2\lambda}\,(1 - e^{-2\lambda t})$$

Important features of this model are that the acceleration is assumed to be independent of velocity magnitude and direction, that the variance reaches a final value, $Q_c/2\lambda$, and that the covariance and mean value change most rapidly at $t = o$.

Constant Turning Rate Acceleration Model.

Propagation of Mean (Velocity and Acceleration):

$$\dot{V}(t) = A(t)$$

$$\dot{A}(t) = -\omega^2 V(t)$$

where

$V(t)$ = target total velocity component

$A(t)$ = target total acceleration component

ω = target turn rate, assumed constant and known (in this Appendix)

For this model, the plant matrix, $F(t)$, is given by

$$F(t) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$$

and the covariance, $P_T(t)$, propagation by

$$\dot{P}_T(t) = \begin{bmatrix} 2P_{T12} & P_{T22} - \omega^2 P_{T11} \\ P_{T22} - \omega^2 P_{T11} & Q_T - 2\omega^2 P_{T12} \end{bmatrix}$$

This can be rearranged as a linear differential equation for a vector

composed of the three distinct elements of P_T, which are P_{T11}, P_{T12}, and P_{T22}, since $P_{T21} = P_{T12}$. The result is

$$\begin{bmatrix} \dot{P}_{T11} \\ \dot{P}_{T12} \\ \dot{P}_{T22} \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ -\omega^2 & 0 & 1 \\ 0 & -2\omega^2 & 0 \end{bmatrix} \begin{bmatrix} P_{T11} \\ P_{T12} \\ P_{T22} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ Q_T \end{bmatrix}$$

for which the zero-state response is (setting $t_o = 0$)

$$P_{T11}(t) = \frac{Q_T}{2\omega^2}\left[t - \frac{\sin 2\omega t}{2\omega}\right]$$

$$P_{T12}(t) = \frac{Q_T}{4\omega^2}\left[1 - \cos 2\omega t\right]$$

$$P_{T22}(t) = \frac{Q_T}{2}\left[t + \frac{\sin 2\omega t}{2\omega}\right]$$

The response for velocity and acceleration with zero initial velocity is

$$V(t) = \frac{A_o}{\omega} \sin \omega t$$

$$A(t) = A_o \cos \omega t$$

The accompanying figures (A-3 through A-7) give the time response, using the following parameters:

$Q_T = 256.\ ft^2/sec^5$

$\omega = .1$ rad/sec

$A(t_o) = 100.\ ft/sec^2$

$V(t_o) = 0.\ ft/sec$

$P_T(t_o) = 0._{2x2}$

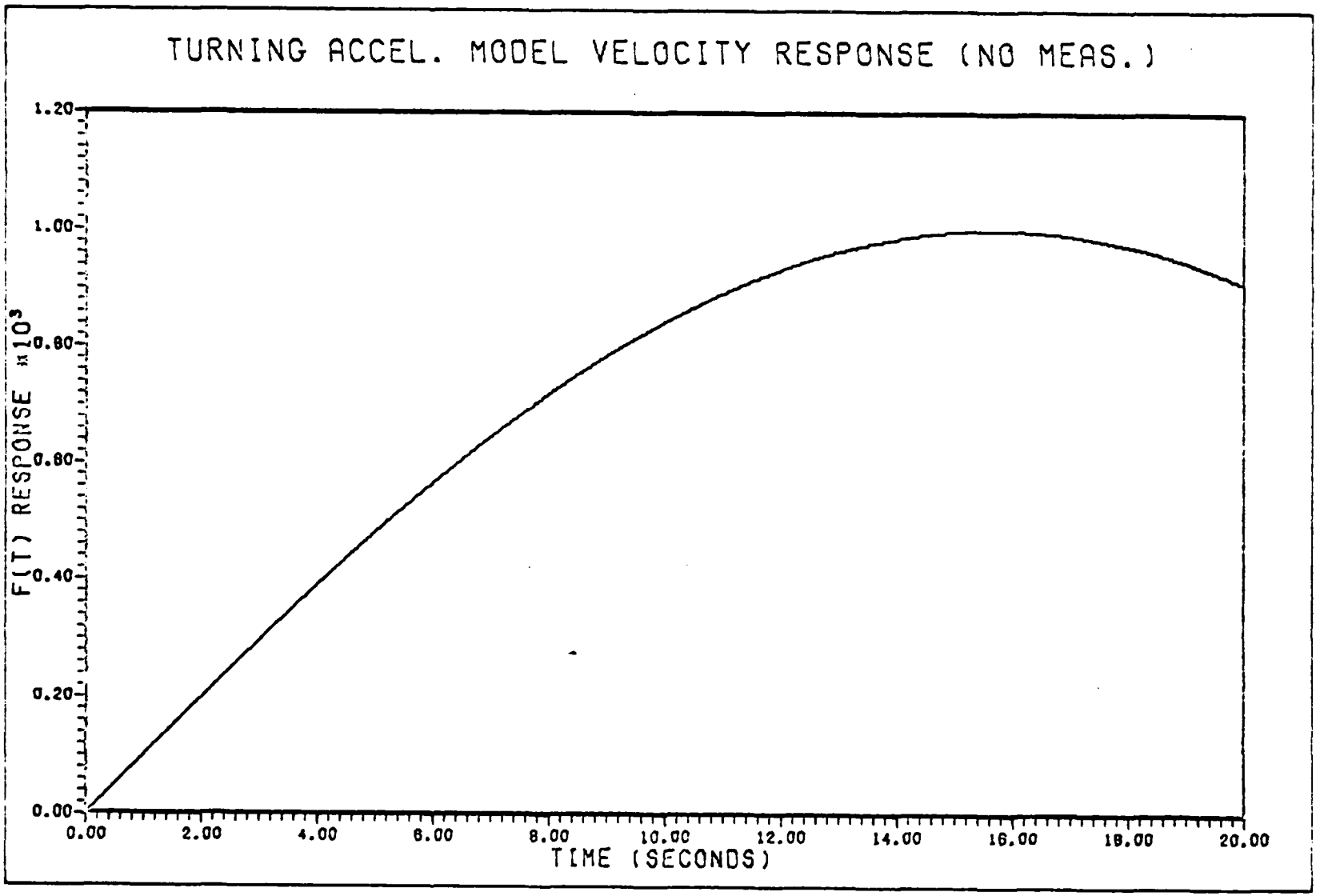


Figure A-3. Turning Model Velocity Response

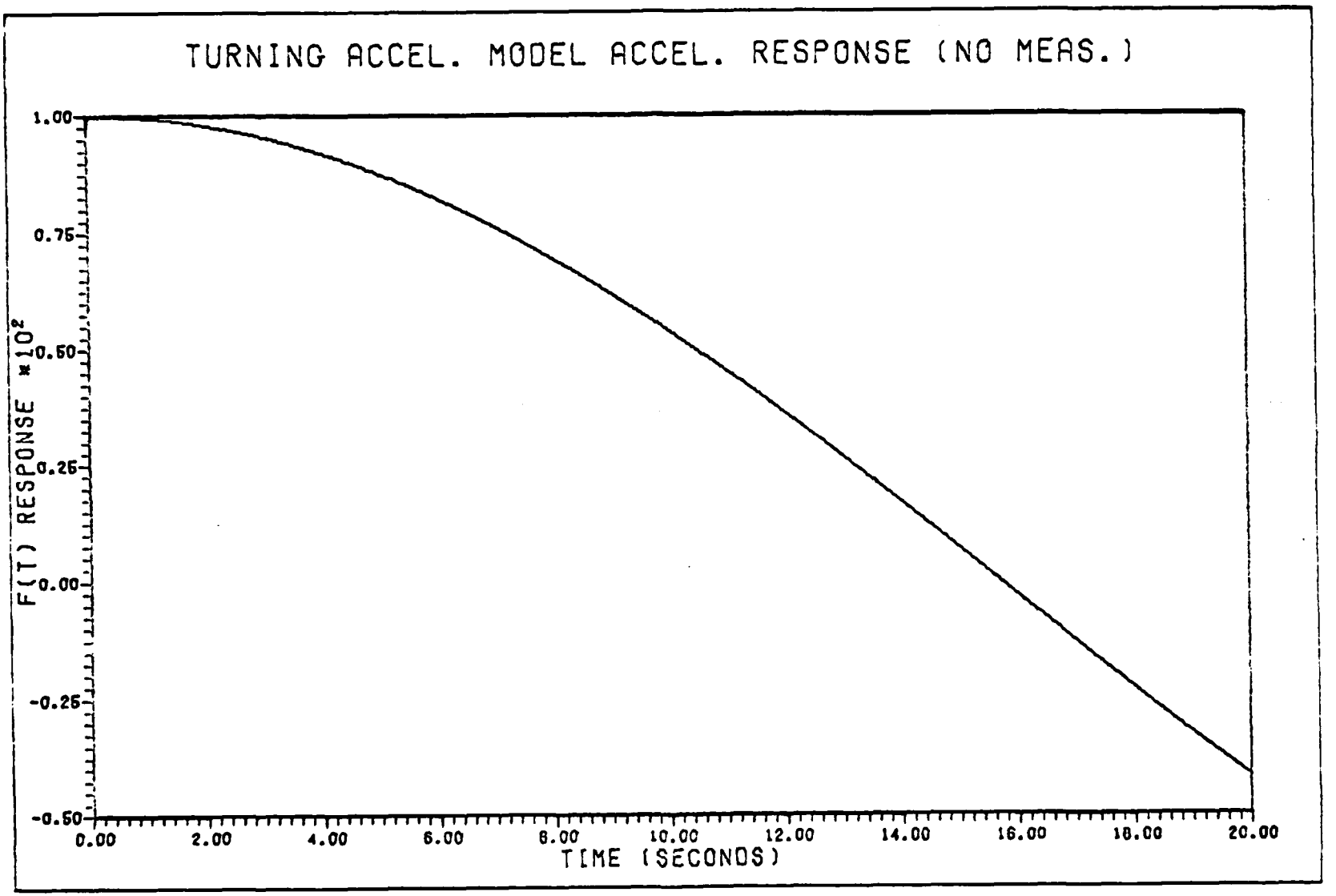


Figure A-4. Turning Model Acceleration Response

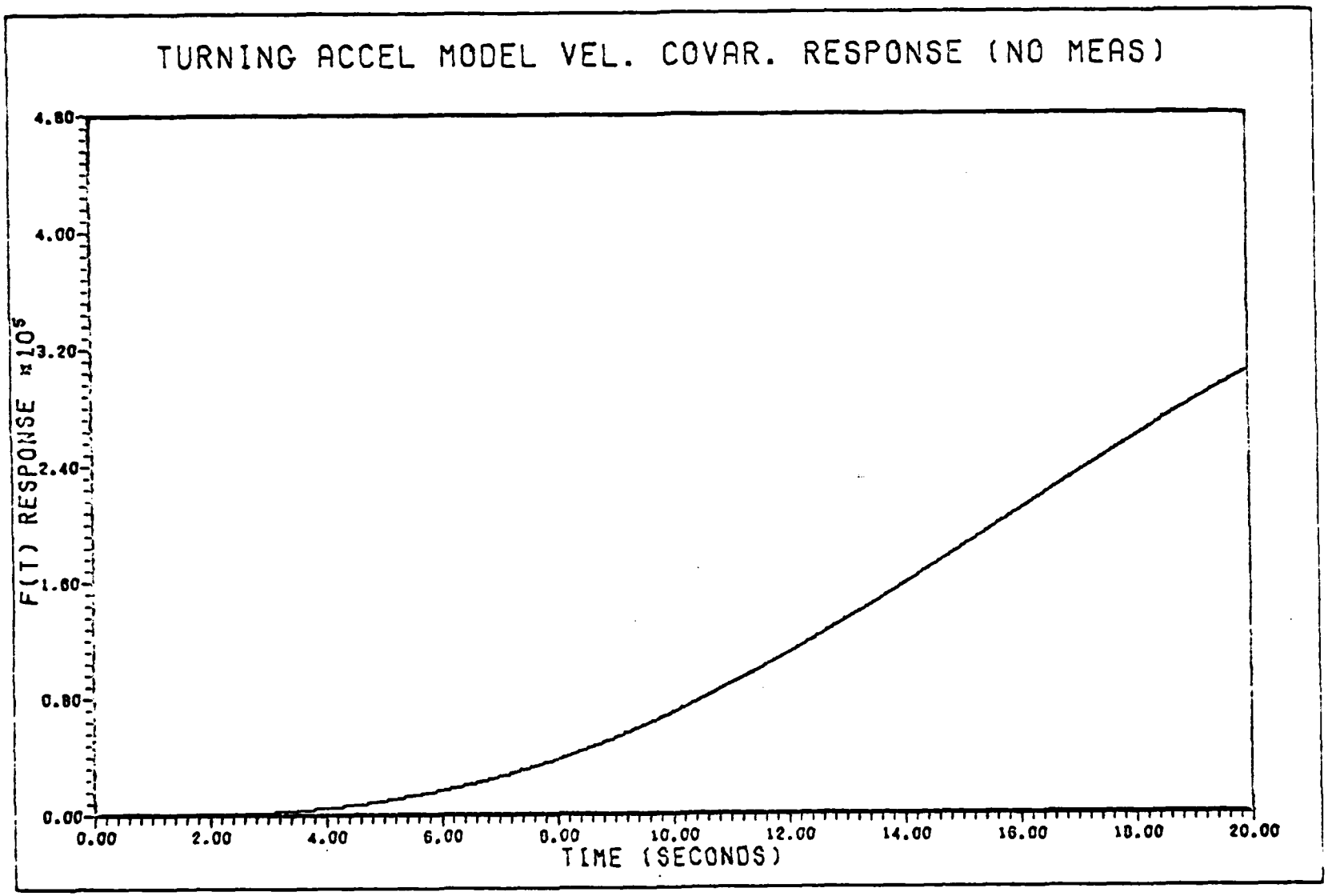


Figure A-5. Turning Model Velocity Variance

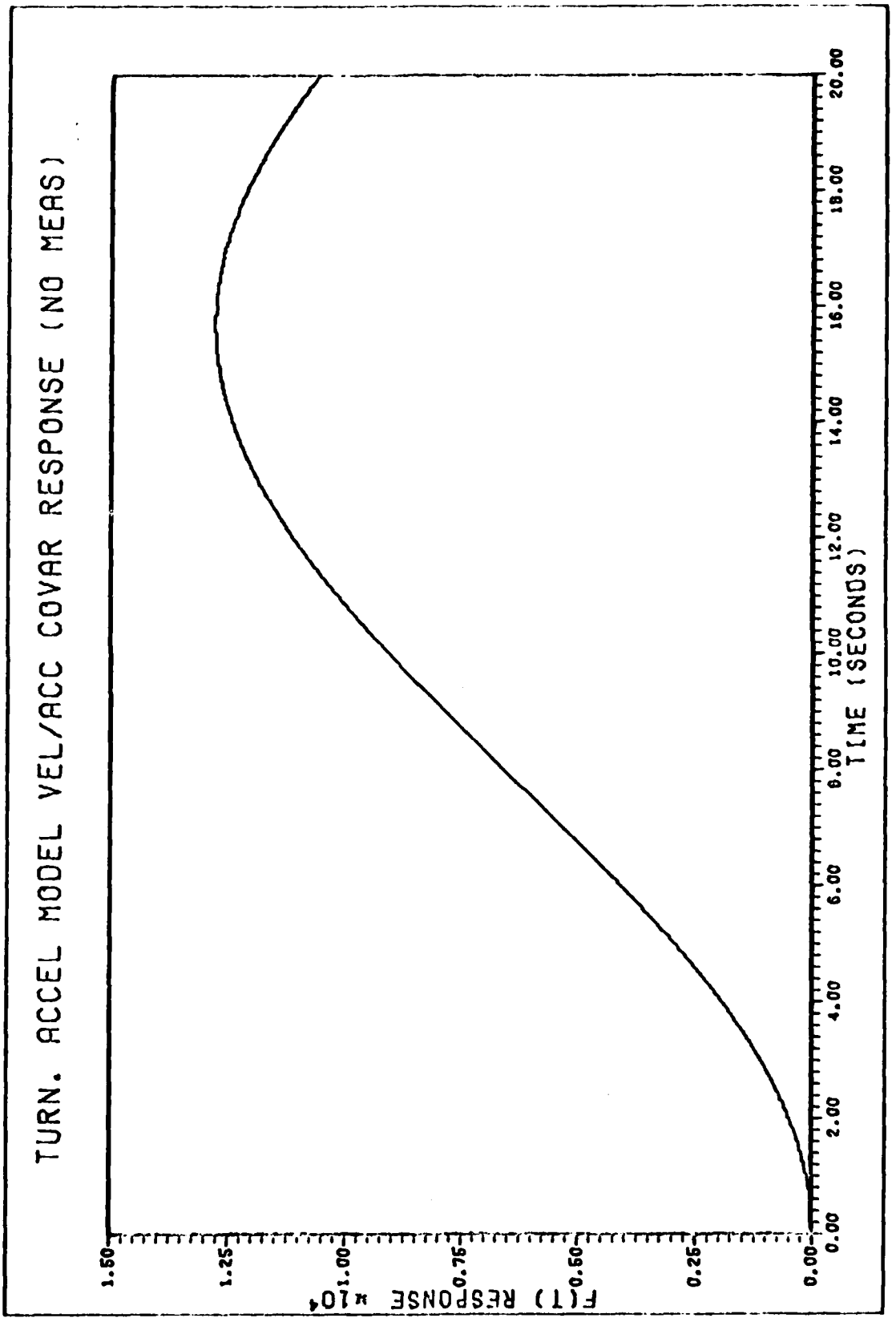


Figure A-6. Turning Model Velocity/Acceleration Covariance

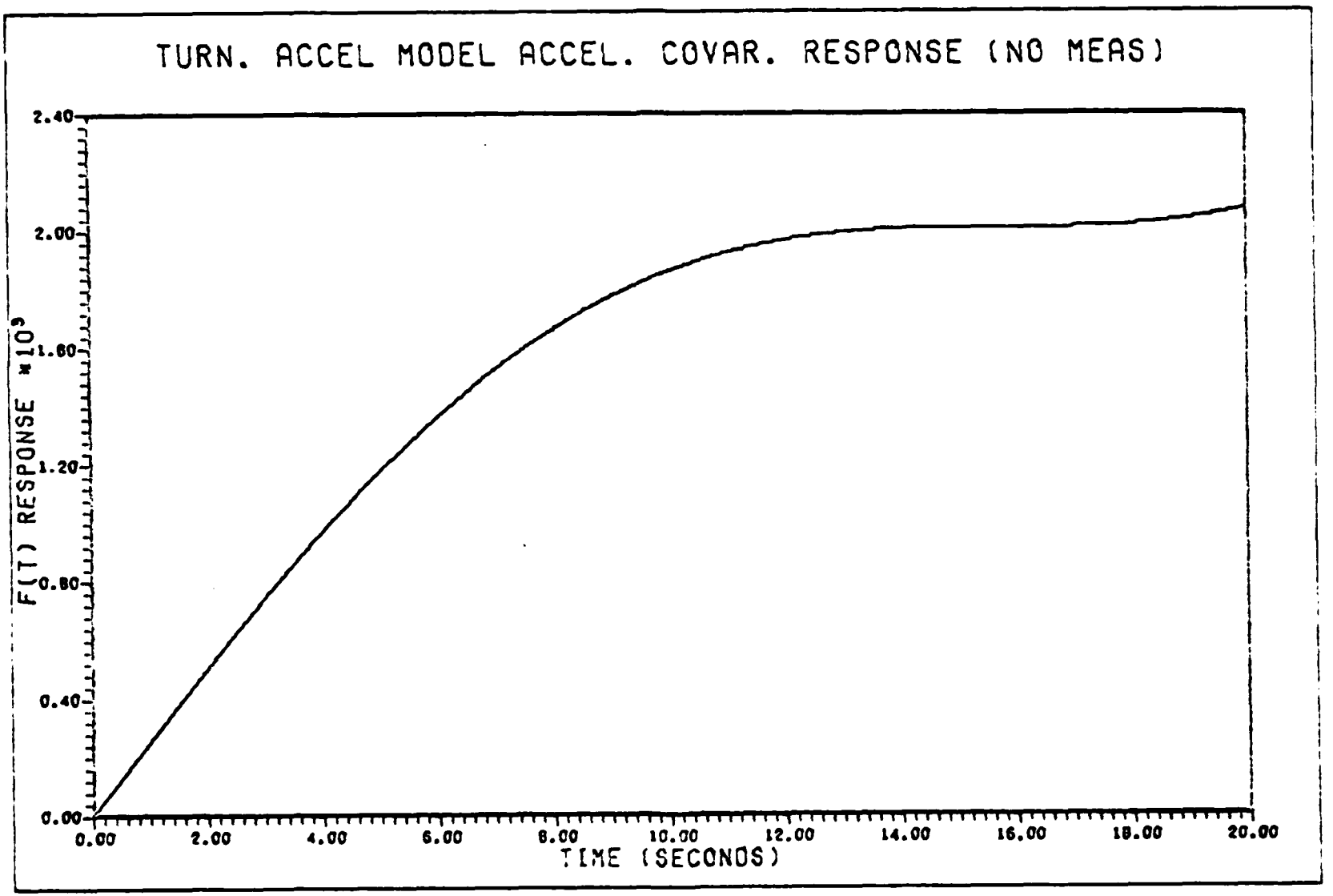


Figure A-7. Turning Model Acceleration Variance

Important differences between this model and the time-correlated model are 1) a strong relationship between acceleration and velocity, 2) unbounded covariance elements P_{T11} and P_{T22} (if no measurements are taken), and 3) the significantly lower value of Q_T that is possible if you assume this is a better model of actual target motion. The lower value of Q_T is possible because the turning rate model does not assume a zero-mean process, as does the time-correlated model. Note that the value of Q_C used, 1000. ft^2/sec^5, produces a one sigma value for acceleration of

$$\sigma_{AC} = [Q_c/2\lambda]^{1/2} = 50.\ ft/sec^2$$

which corresponds to about 1.5 g, which is not a very dynamic target. If a higher value of Q_c is used to account for more dynamic targets, more measurement noise will pass through the filter.

When the models are incorporated into an estimator with measurements, the Kalman gain effects a feedback path which will correct the long term errors of the open-loop models. Due to the measurement noise, the feedback will not affect the short-term (high-bandwidth) response. The degree to which real targets tend to turn at constant rate, rather than let acceleration decay toward zero, at least for short periods of time, determines the advantages of the turning target model. In the actual filter, ω is not given, but must be estimated from velocity and acceleration, thus making the filter nonlinear.

Appendix B

Sensitivity of Filter to Line-of-Sight Unit Vector

In Chapter Two, a question was raised concerning the senstivity of the measurement update equations to an error along the line-of-sight. Assuming that the three components of the unit vector are input to the filter as measurements, the filter should ignore any component of that unit vector measurement, $\underline{U}^A_{R_m}$, that changes its length from unity. For the three direction cosines, the Kalman gain, (Eq. (2-7)) $K(t_i)$, can be rewritten as

$$K(t_i) = P(t_i^+)H(t_i)^T R(t_i)^{-1} \tag{B-1}$$

Since the components of $\underline{U}^A_{R_m}$ are resolved into aircraft coordinates, the fact that the filter ignores errors along the line-of-sight can be established by proving that the Kalman gain times a residual in the line-of-sight direction is zero. This is expressed in equation (B-2):

$$K(a\,\underline{U}^A_R) = 0 \tag{B-2}$$

where,

K = Kalman gain, from eq. (B-1)

a = arbitrary scalar length of error along L.O.S.

$\underline{U}^A_R$ = unit vector along L.O.S. in aircraft coordinates

and the time, t_i, is dropped as unnecessary for this proof.

Since the errors in $\underline{U}^A_{R_m}$ are assumed homogeneous, the covariance matrix, R, can be written as

$$R = r\, I_{3x3} \tag{B-3}$$

so that,

$$R^{-1} = \frac{1}{r} I_{3x3} \tag{B-4}$$

Substituting (B-4), (2-26), and (B-1) into (B-2),

$$K(a\ \underline{U}_R^A) = a\ P^+\ H^T\ R^{-1}\ \underline{U}_R^A = \frac{a}{\hat{r}R}\ P^+\ [T_{IA}^T - \underline{U}_R^I\ \underline{U}_R^{A^T}]\ \underline{U}_R^A$$

$$= \frac{a}{\hat{r}R}\ P^+\ [\underline{U}_R^I - \underline{U}_R^I\ (\underline{U}_R^{A^T}\ \underline{U}_R^A)] = \frac{a}{\hat{r}R}\ P^+\ [\underline{U}_R^I - \underline{U}_R^I(1)]$$

$$K(a\ \underline{U}_R^A) = 0 \tag{B-5}$$

since $\frac{\underline{P}_{T/A}}{\hat{R}} = \underline{U}_R^I$, the L.O.S. unit vector in inertial coordinates

and $\underline{U}_R^{A^T}\ \underline{U}_R^A = 1$, the length of a unit vector

Therefore, any component of the measured unit vector along the line-of-sight produces no change in the updated state vector.

Appendix C

Trajectory Descriptions

The six trajectories which were used in the simulation are illustrated in this Appendix with three-dimensional ribbon plots of each attacker and target time history. Figures C-1 through C-4 shows target trajectories 1 through 6, along with their corresponding attackers, who are flying gunnery lead pursuit, i.e. pointing the gun axis of the attacker sufficiently far ahead of the target to provide near-zero miss distance. The accuracy of the lead pursuit is not important for this investigation. For each aircraft, the closely-spaced pair of lines represent traces of the wing tips throughout the pass. The aircraft dimensions were deliberately scaled-up by a factor of four to improve resolution. There is also a single-line ground trace to enable some degree of separation of horizontal and vertical components of the trajectory.

The aircraft trajectory which is marked at the starting point with a circle is attacker number 1. Attacker number 2 is marked with a "+" sign. Target number 1 is marked with a triangle, and target number 2 is marked with an "X". The quadrilateral is the ground plane which contains all the trajectories, and the (0., 0., 0.) point (inertial origin) is the intersection of the ground plane and the attacker one starting point. A vertical axis extends upward from one corner to the maximum vertical point for that test case. In all cases, attacker one is initially flying north. To provide a general idea of scale, the minimum and maximum X, Y, and Z values for the "box" defined by the ground plane and vertical axis are defined in Table C-1. In addition, the X, Y, and Z locations of the viewing point are given. The graphs are drawn in true perspective.

Since a north, east, down coordinate frame is used, all Z-values are negative for points above ground.

Table C-1

Trajectory Dimensions (ft)

Fig. No.	C-1	C-2	C-3	C-4
Min. X	-1000.	-1000.	-1000.	-4000.
Max. X	21000.	-60000.	-60000.	13000.
Min. Y	-1000.	-1000.	-1000.	-14000.
Max. Y	12000.	9000.	9000.	28000.
Min. Z	-10000.	-25000.	-25000.	-21000.
Max. Z	0.	0.	0.	0.
View X	10000.	30000.	75000.	22000.
View Y	21000.	-50000.	10000.	-10000.
View Z	-23000.	-60000.	-25000.	-30000.
Traj. Nos.	1,2	3,4	3,4	5,6

Two figures were included for trajectories 3 and 4 because of the long range in X.

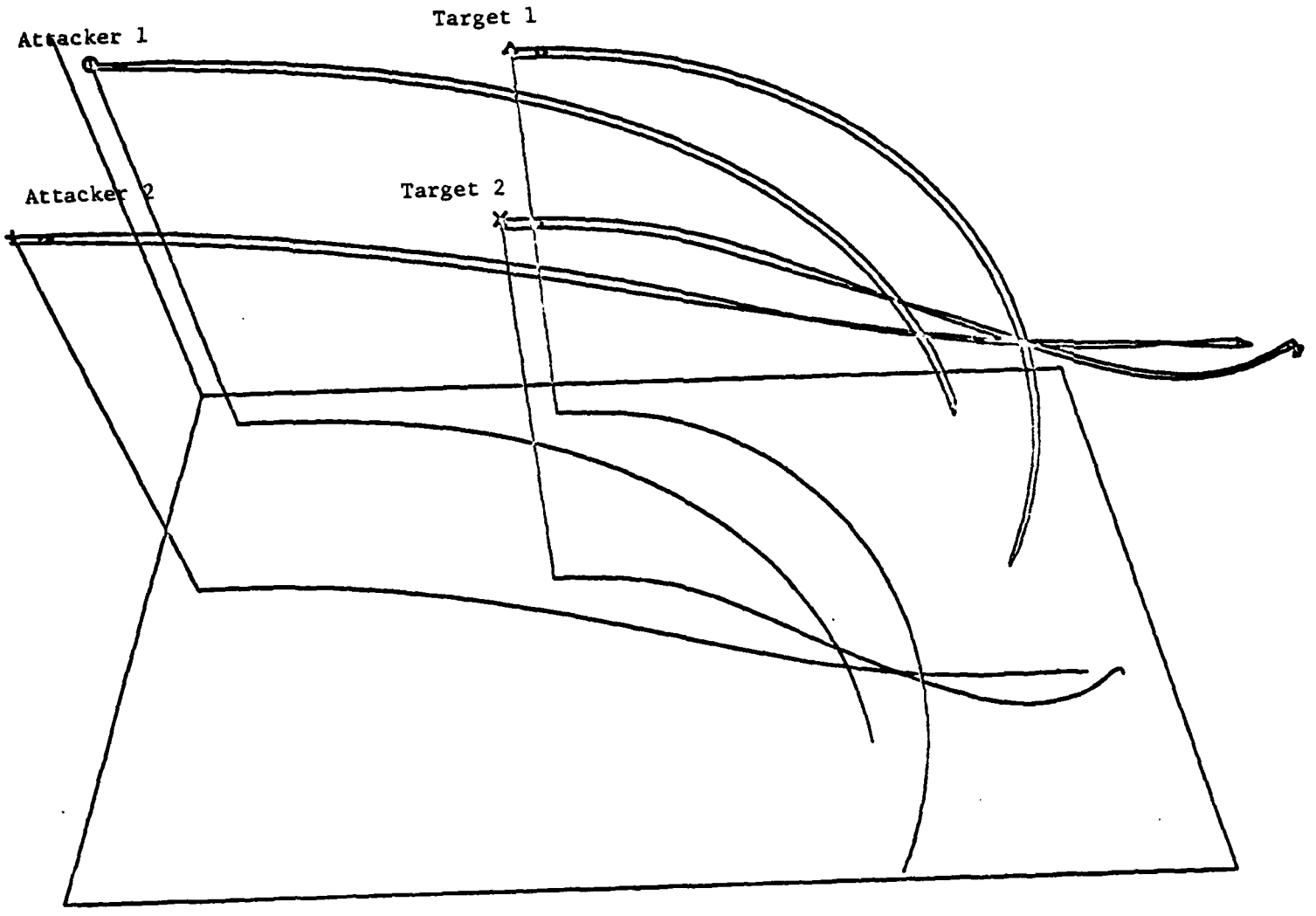


Figure C-1. Trajectories 1 and 2

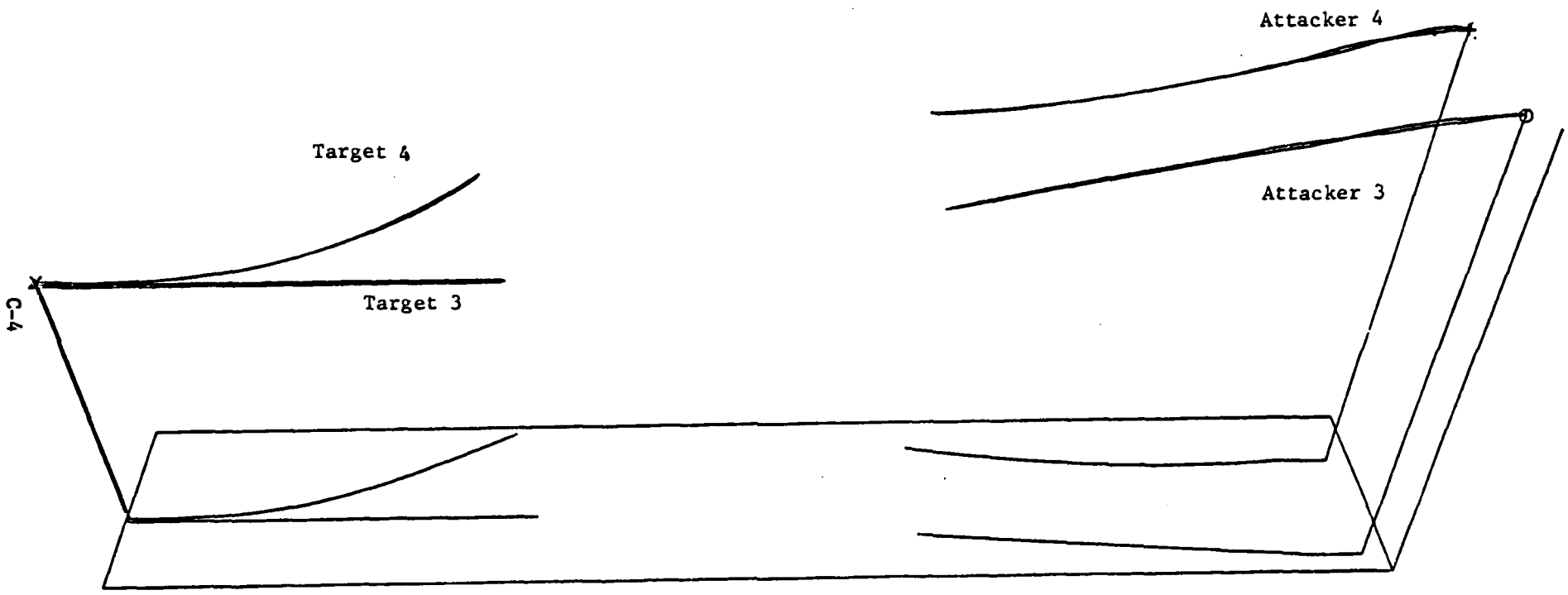


Figure C-2. Trajectories 3 and 4, View 1

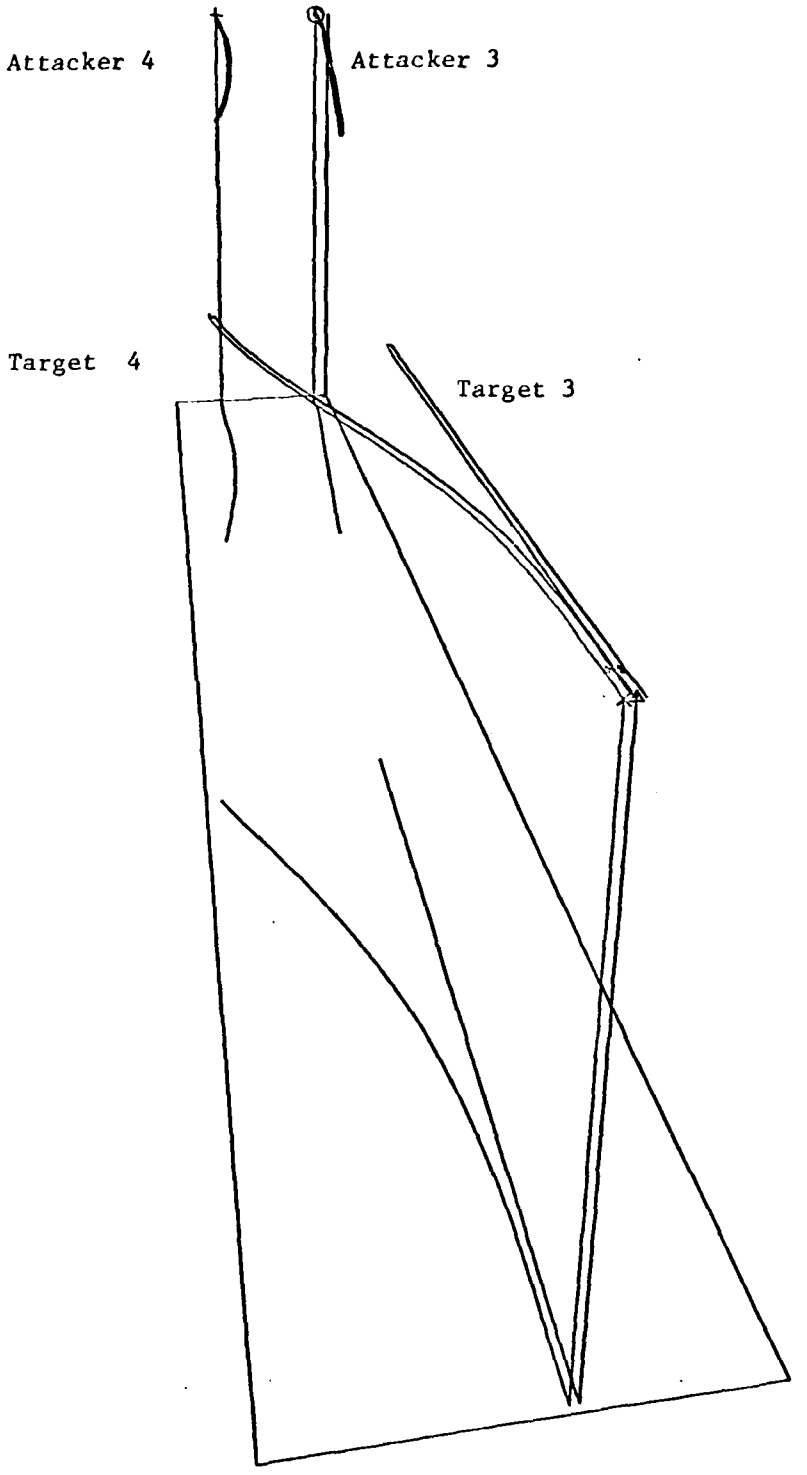


Figure C-3. Trajectories 3 and 4, View 2

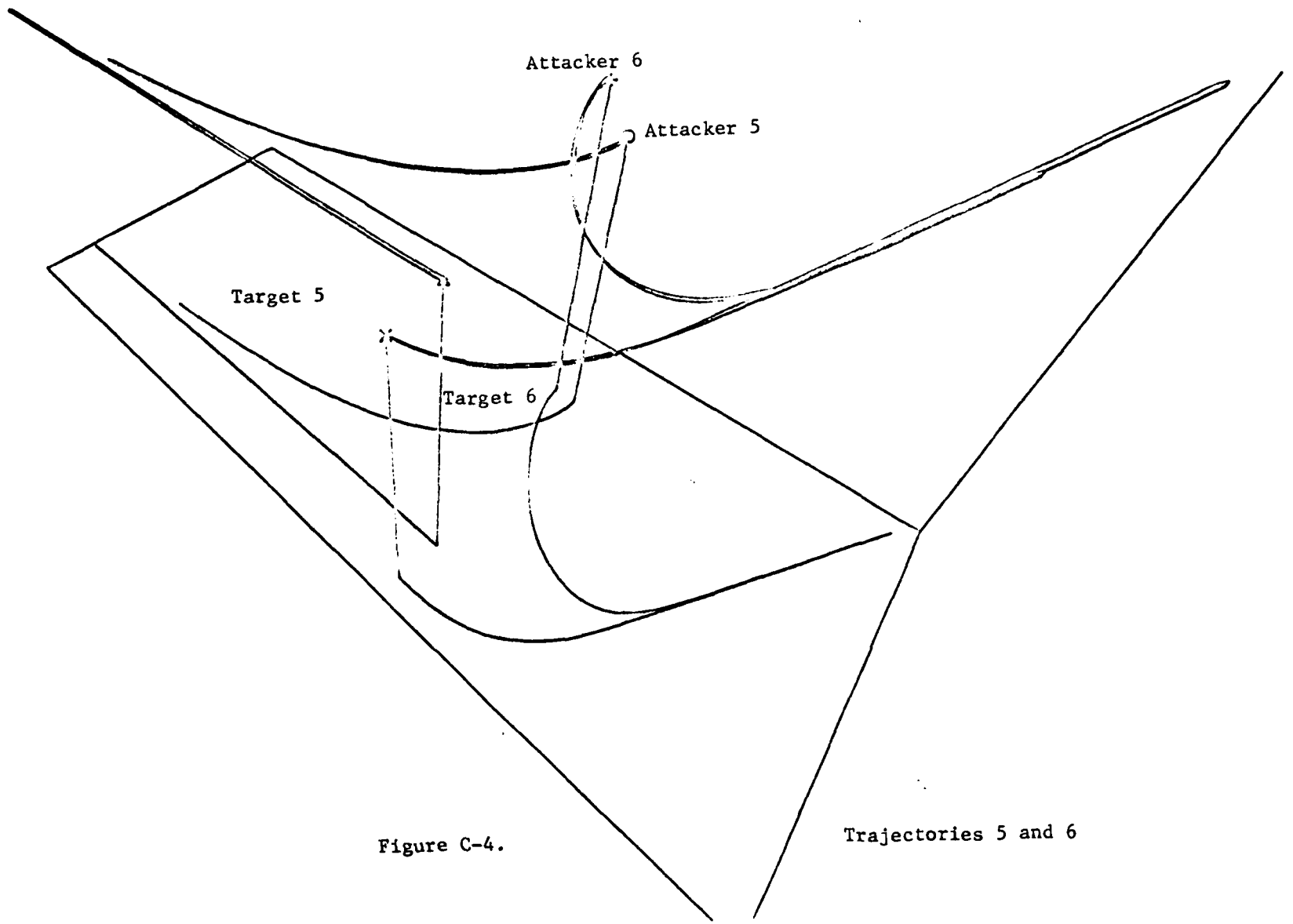


Figure C-4. Trajectories 5 and 6

Appendix D

Effect of Residual Transfer on Filter Response

The purpose of this appendix is to show how transferring residuals between filters significantly alters the response of the filter to inputs from either aircraft. This can be illustrated with a linear, time-invariant filter. Assuming a single filter with an open-loop transfer function F(s), the unity feedback, closed-loop response can be written as:

$$C(s) = F(s)[r(s) - C(s)]$$

where

C(s) = Laplace domain closed-loop response

F(s) = Laplace domain open-loop response

r(s) = Laplace domain input signal

Equation (D-1) can be rearranged as

$$\frac{C(s)}{r(s)} = \frac{F(s)}{1 + F(s)}$$

In this analog to the filter problem, C(s) represents the state estimate, and r(s) represents the measurement. Unity feedback represents a system with H=I. If two nearly identical filters are now coupled through residual transfer, and one-half of the residual, [r(s) - C(s)], is incorporated from each filter (i.e., the residuals are averaged) the simultaneous equations become:

$$C_1(s) = \frac{1}{2} F_1(s)[r_1(s) - C_1(s) + r_2(s) - C_2(s)]$$

$$C_2 = \frac{1}{2} F_2(s)[r_2(s) - C_2(s) + r_1(s) - C_1(s)]$$

The solution to these simultaneous equations is

$$C_1(s) = \frac{F_1(s)}{\frac{F_1(s) + F_2(s)}{2} + 1} \cdot \frac{r_1(s) + r_2(s)}{2}$$

$$C_2(s) = \frac{F_2(s)}{\frac{F_1(s) + F_2(s)}{2} + 1} \cdot \frac{r_1(s) + r_2(s)}{2}$$

It can now be seen that the denominator of the transfer function has been altered to include both $F_1(s)$ and $F_2(s)$, so that the two filters are structurally coupled. In the extended Kalman filter implemented in this investigation, not only are the filter responses different for each aircraft due to the geometry, but they are to some degree measurement data dependent. This fact, together with the coupling effect just demonstrated, provides a mechanism for instability in the coupled filters.

Appendix E

User-Written Routines for SOFE

Subroutine "FQGEN" generates the F and Q matrices for use in propagating the filter P matrix (Eq. 2-5). The eighteen filter states, in order, are "x" position, velocity, and acceleration, "y" position, velocity, and acceleration, and "z" position, velocity, and acceleration for the filter in the aircraft number one, followed by the same nine states, in the same order, for the filter in aircraft number two. Since there are two tracking aircraft in the simulation, the computations to generate the F and Q matrices are performed once for each aircraft, each time using the appropriate filter states.

Subroutine "HRZ" is a fairly large subroutine which calculates the 1-by-18 H matrix (which is the linearized measurement sensitivity $\frac{\partial h}{\partial x}$), R, the scalar measurement noise variance, and $[z - h(\hat{x}(t^-))]$, the scalar measurement residual. Since "SOFE" performs iterative scalar measurement updates using the Carlson Square Root method, subroutine "HRZ" is called three times to incorporate a range, azimuth, and elevation measurement into a single filter. With two single filters running independently, six scalar updates are performed. With the two filters coupled together, there are twelve scalar updates, although there are still only six measurements, each measurement having been incorporated once into each of two filters. "HRZ" saves each measurement until it has been incorporated into both filters. Besides controlling the times at which measurement transfers are allowed, there is another subtle, but important aspect of the way in which random noise is obtained to generate the noisy measurement. Function "GAUSS", which generates the gaussian-distributed

measurement noise, is always called the same number of times each time "HRZ" is called, to ensure repeatability of noise samples from one test case to the next. That is, although each Monte-Carlo replication encounters a different noise sequence, the "m"th run of test case "n" has exactly the same sequence of radar range errors as test case "ℓ". There two possible ways of computing $H = \frac{\partial h}{\partial x}$ in "HRZ". Normally, H is computed as a function of the filter state, $\underline{\hat{x}}_f$. For a special investigation of the effects of errors in $\underline{\hat{x}}_f$ on the evaluation of H, it is possible to select H computed with $\underline{x}_t$. This capability is included to support test case set 2.1.5, as described in Chapter 3.

Subroutine "XSDOT" calculates the derivative of the true state, $\underline{x}_t$ i.e., it specified $f[x(t),t] + B(t)u(t)$. Although this is done the same way as in subroutine "XFDOT", described below, it is not really important since $\underline{x}_t$ will be reset to the external trajectory values immediately after integration. More importantly, the initial values of $\underline{x}_t$ are always set equal to the trajectory values during the initialization portion of subroutine "XSDOT", so that true values are used in computing the first measurement.

Subroutine "XFDOT" calculates the derivative of the filter state $\underline{\hat{x}}_f$. During its initialization, it can optionally reset the initial $\underline{\hat{x}}_{f_o}$, which was set in the basic "SOFE" program, to perfect, error-free values, or to random initial values with zero mean error and variance determined by the initial filter covariance matrix, PF.

Subroutine "SNOYS" simply resets the true values, $\underline{x}_t$, to the trajectory values after the integration, but before the measurement update.

Subroutine "ESTIX" is provided for specialized user output. In this instance, it was only used to output the Kalman gain and residual

used in the option of using state transfers as measurements.

Subroutine "AMEND" is called after the usual measurement update and prior to another propagation cycle of the filter. It is used to manage the update of the filter when the alternate filter's states are used as measurements. This was implemented via a standard Kalman filter vector measurement update, rather than the Carlson Square Root method incorporated in "SOFE". There are two reasons for this choice. First, the alternate filter's estimated covariance matrix, $P(\hat{\underline{x}}(t_i),t_i)$, is not diagonal, which would necessitate a transformation of the state before incorporation using the Carlson Square-Root update (which assumes a diagonal measurement noise covariance matrix.) Secondly, it would not be convenient to update the states without updating the covariance simultaneously (as described in Chapter Two, "Using State Estimates as Measurements") unless further changes were made to basic "SOFE". "AMEND" controls the timing of state transfers, determines the order of state transfer, i.e., which combinations of position, velocity, and acceleration states will be transferred, calls subroutine "RSDUL" to form the "measurement" residual $(\hat{\underline{x}}_2(t_i) - \hat{\underline{x}}_1(t_i))$, and calls subroutine "PPLGAN", which calculates the 9 by 9 Kalman gain matrix, optionally updates the covariance matrix, and finally updates the states themselves by calling subroutine "UPDSTS".

Subroutine "TRAJ" reads trajectory values from an external file as they are required by the simulation.

Subroutine "USRIN" reads the input which controls the operation of, and data for, the user-written subroutines. These inputs are specifically detailed in the next section.

Input to "USRIN".

Namelist input is used to control the data and options used within the application routines. Table E-1 gives the name, default value, and description of each input variable.

Table E-1

User Input to SOFE

	Fortran Name	Chap. 2 Symbol	Default Value	Description
1.	QF(9)	Q	$1.\times10^{7}$ pos., vel. 256. acc.	Disturbance noise strength assumed in filter, see 2., below
2.	TQ*	-	0. sec	Time at which to reset pos., vel. QF to zero.
3.	RR	R_R	900.	Range measurement variance. (30 ft, 1σ)
4.	RE	R_{UR}	$16.\times10^{-6}$	Elevation measurement variance. (4 mr, 1σ)
5.	RT	R_{UR}	$16.\times10^{-6}$	Traverse measurement variance. (4mr, 1σ)
6.	RRDOT*		-100.	Range rate measurement variance. Negative value causes SOFE to ignore measurement.
7.	RLAMBDA	λ	0.	Correlation constant for correlated target acceleration model.
8.	CTURN	-	1.	Coefficient to multiply $\omega_{t-t}^2 \underset{\sim}{V}_t$ term in turning target model.
9.	NTGT(2)	-	1,1	Target number for each of the two trackers to track.
10.	MAXNUM	-	2	Maximum number of filters.
11.	PRFCIN	-	FALSE	Initialize $\hat{\underset{\sim}{X}}_o$ perfectly if true.
12.	PRFCH	-	FALSE	Compute perfect $H = \frac{\partial h}{\partial x}\Big\|_{x = x_t}$ if true.
13.	TXFERO	-	$1. \times 10^{9}$	Time for initial measurement transfer.

14.	DTXFER	-	$1. \times 10^9$	Delta time between measurement transfer.
15.	TXFERF	-	$1. \times 10^9$	Final time for measurement tranfer.
16.	LOGORD(1)	-	True	Transfer position states if true
	(2)	-	False	Transfer velocity states if true
	(3)	-	False	Transfer acceleration states if true
17.	TSFERO	-	$1. \times 10^9$	Initial time for state transfer.
18.	DTSFER	-	$1. \times 10^9$	Delta time between state transfers.
19.	TSFERF	-	$1. \times 10^9$	Final time for state transfer.

*Not used in the investigation

Appendix F

Simulation Data Plots

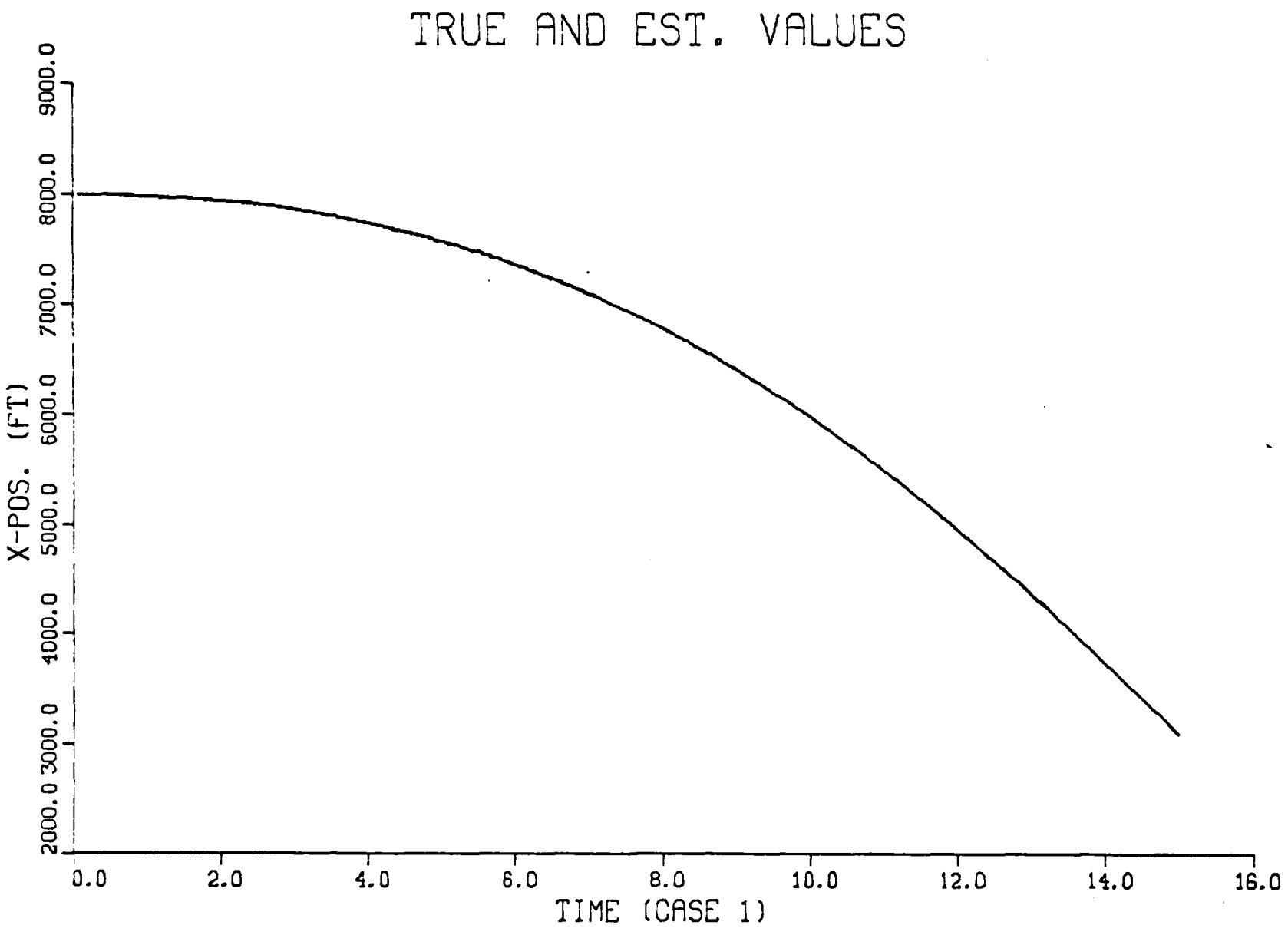
The primary output from the simulation is the set of graphs contained in this Appendix. Since "SOFE" itself does not calculate ensemble statistics, its printed output is not used to determine averages and variances. Rather, it provides data to "SOFEPL", which calculates and plots these data. Printed output would have been too voluminous to compare data from various runs.

There are three different types of plots contained in this Appendix. The first type of plot, labeled "TRUE AND EST. VALUES", shows true and average estimated values for the indicated state, and is used only on test cases 1.1.1, 1.1.3, and 1.2.4. The purpose for showing this type of plot is to indicate the dynamics of the target motion, especially for trajectory 1 (case 1.1.1) and trajectory 4 (case 1.1.3), which are used extensively throughout the investigation. The second type of plot, labeled "ERROR, ERROR +- ONE SIGMA", shows the average error and standard deviation of the error for the indicated state. This type of plot is used on all plots, except for those indicated as type 1, above, and type 3, below. The error, as defined in "SOFEPL" is Average $[x_t(t) - x_f(t)]_j$, where the average is taken over the ten Monte-Carlo runs on state number "j". The top and bottom curves are the average error, plus and minus the standard deviation, respectively.

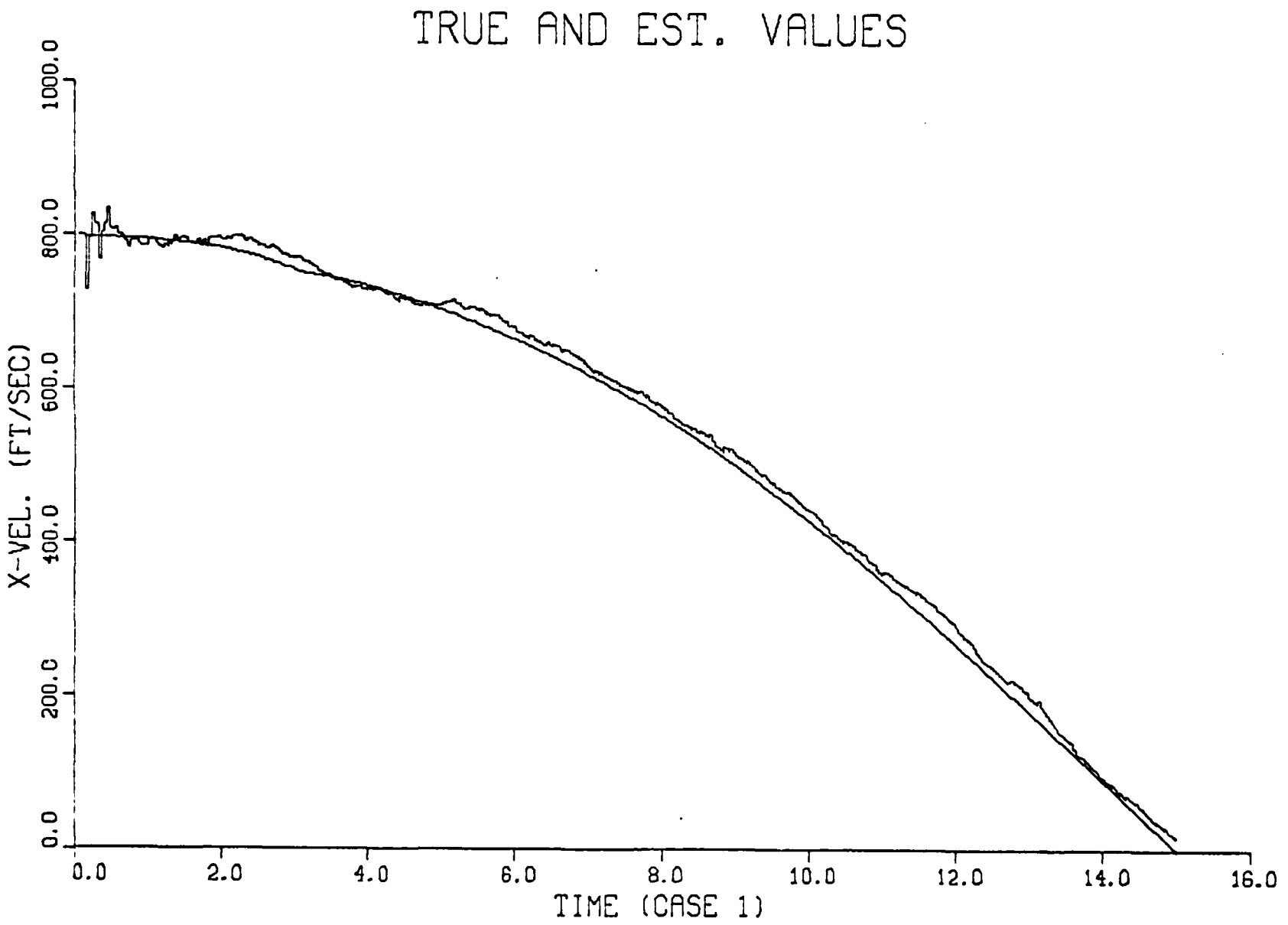
The third type of plot was developed while the test runs were in progress, in order to provide additional useful information. Since the second type of plot shows only the standard deviation of the actual error, no information is present on the filter-estimated standard deviation.

In the third type of plot, this information is shown as $-\text{Aver}[P_{f_{jj}}(t)^{1/2}]$. Since the filter-estimated average error is zero, the average of the square-root of the covariance is subtracted from zero, rather than from th actual average error. This type of plot is shown on cases 2.1.2.1, 2.1.2.2, and 2.2.1.4 through 2.2.3.1, and is labeled, "ERROR, ERROR+ONE SIGMA, -FILTER SIGMA".

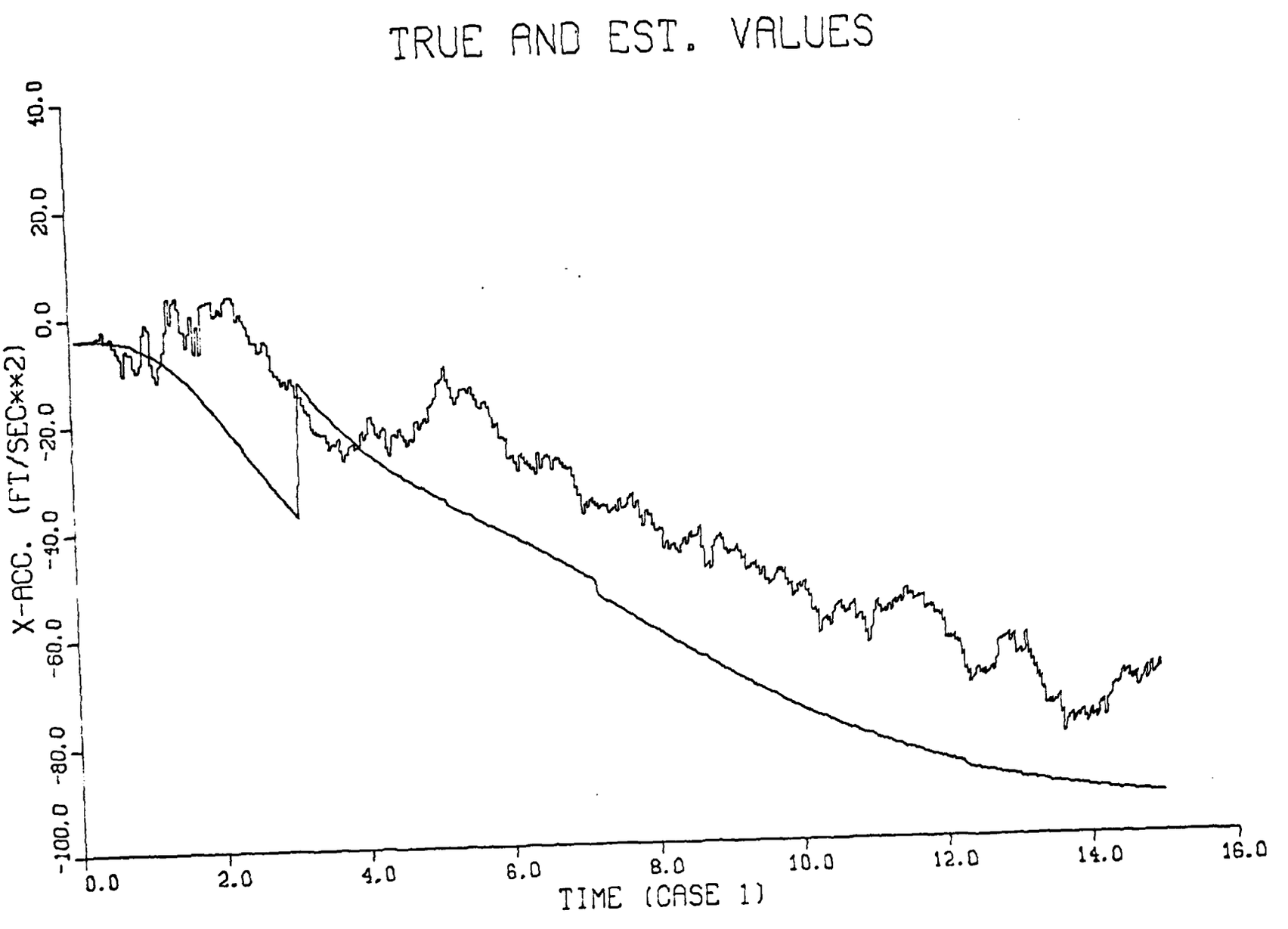
Since all plots are arranged in ascending order by case number, it was not necessary to include a separate figure number and caption on each plot, and so the plots are referred to by page number in Chapter Five.

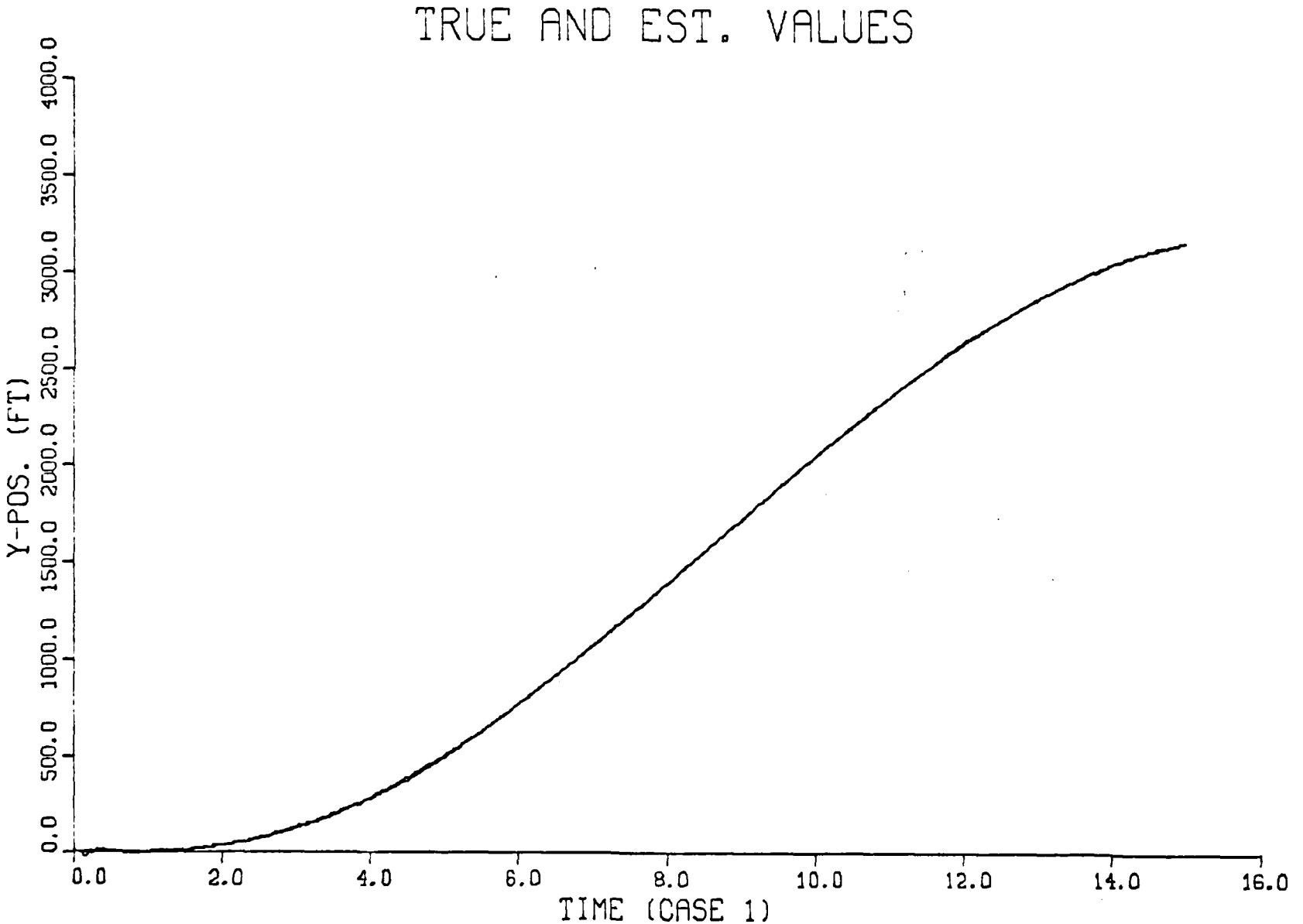

TRUE AND EST. VALUES
X-POS. (FT)
2000.0
3000.0
4000.0
5000.0
6000.0
7000.0
8000.0
9000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

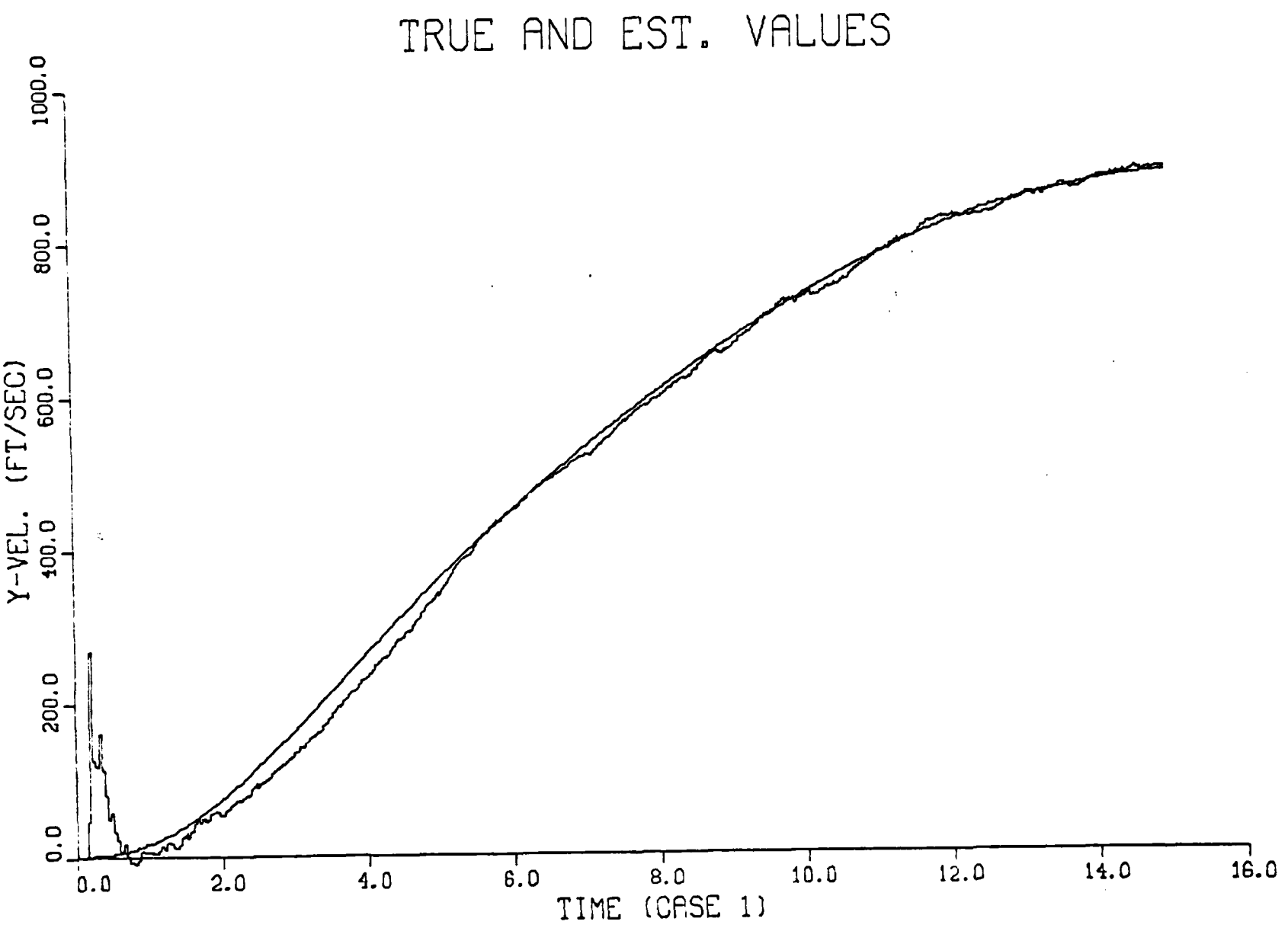
1.1.1


TRUE AND EST. VALUES
X-VEL. (FT/SEC)
1000.0
800.0
600.0
400.0
200.0
0.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

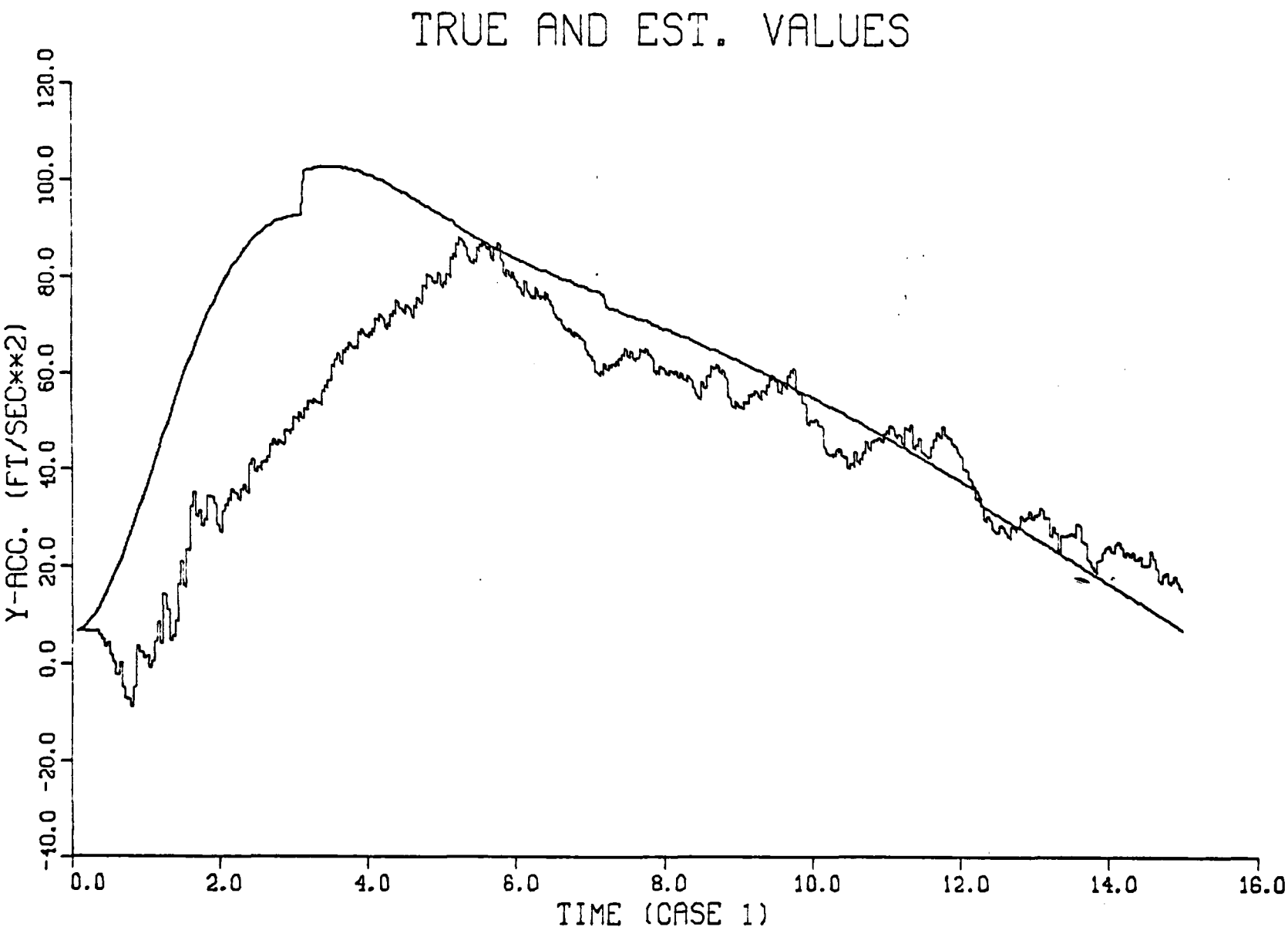
1.1.1


TRUE AND EST. VALUES
X-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

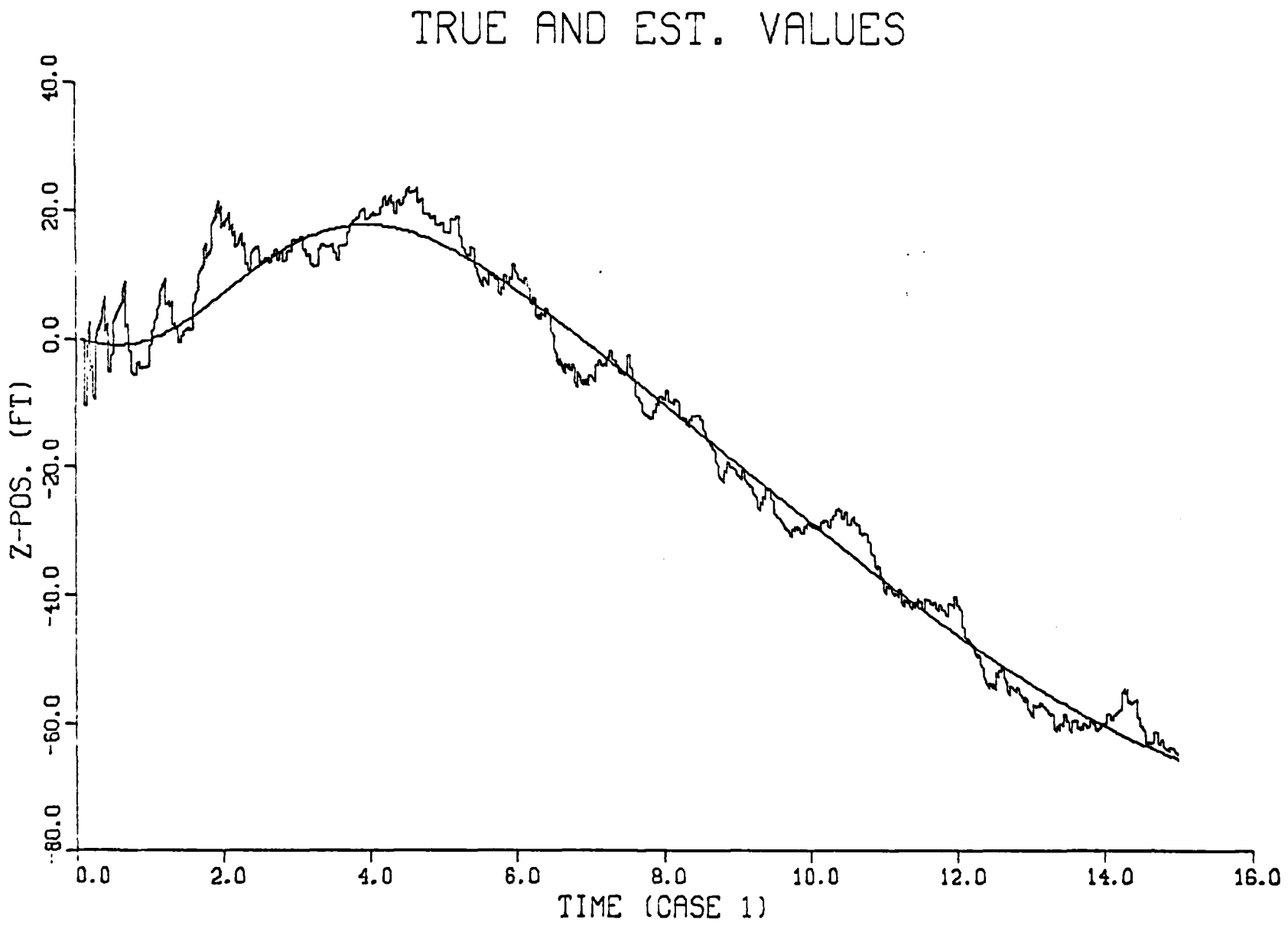

TRUE AND EST. VALUES
Y-POS. (FT)
0.0
500.0
1000.0
1500.0
2000.0
2500.0
3000.0
3500.0
4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)


TRUE AND EST. VALUES
Y-VEL. (FT/SEC)
0.0
200.0
400.0
600.0
800.0
1000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

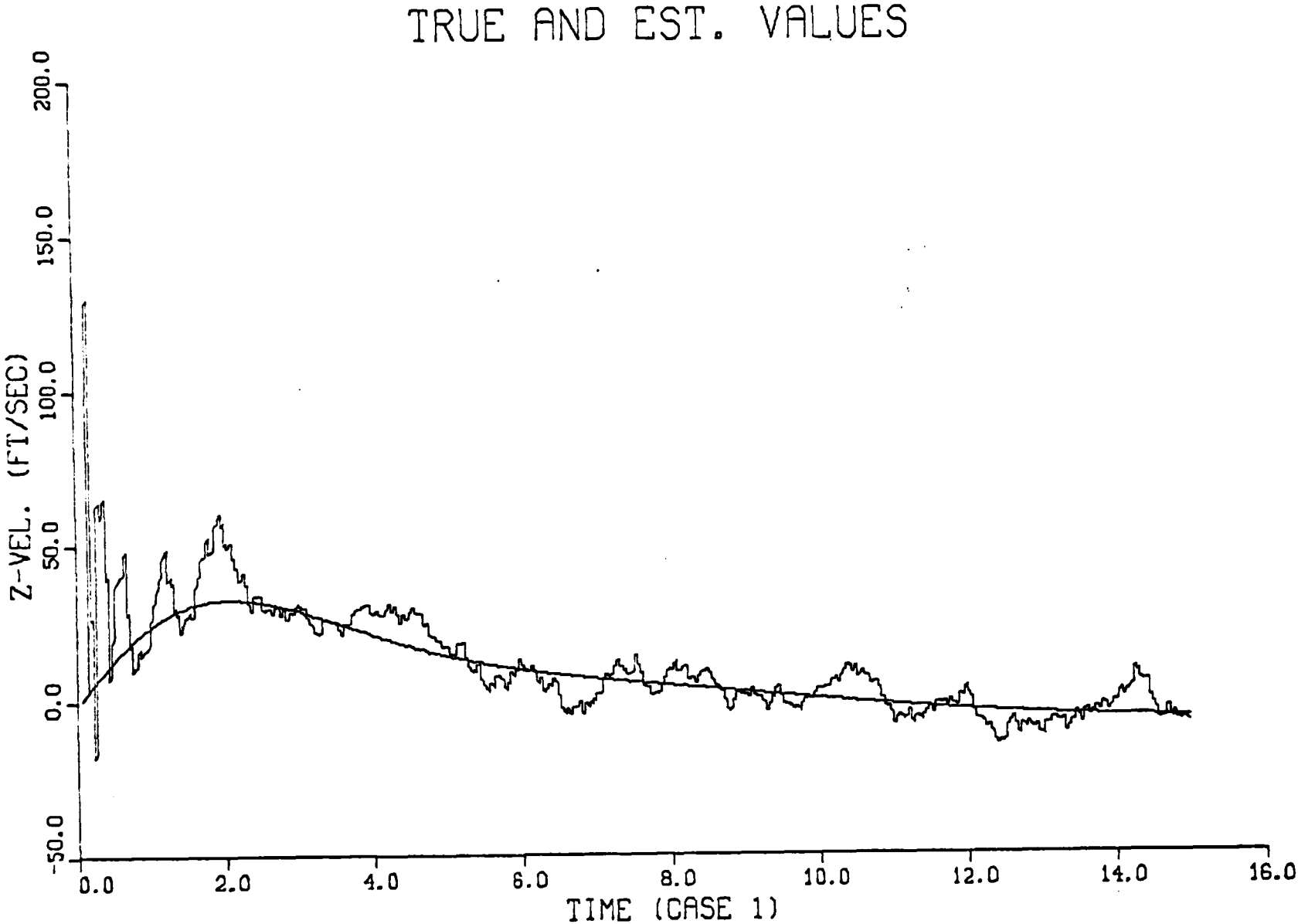
1.1.1


TRUE AND EST. VALUES
Y-ACC. (FT/SEC**2)
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

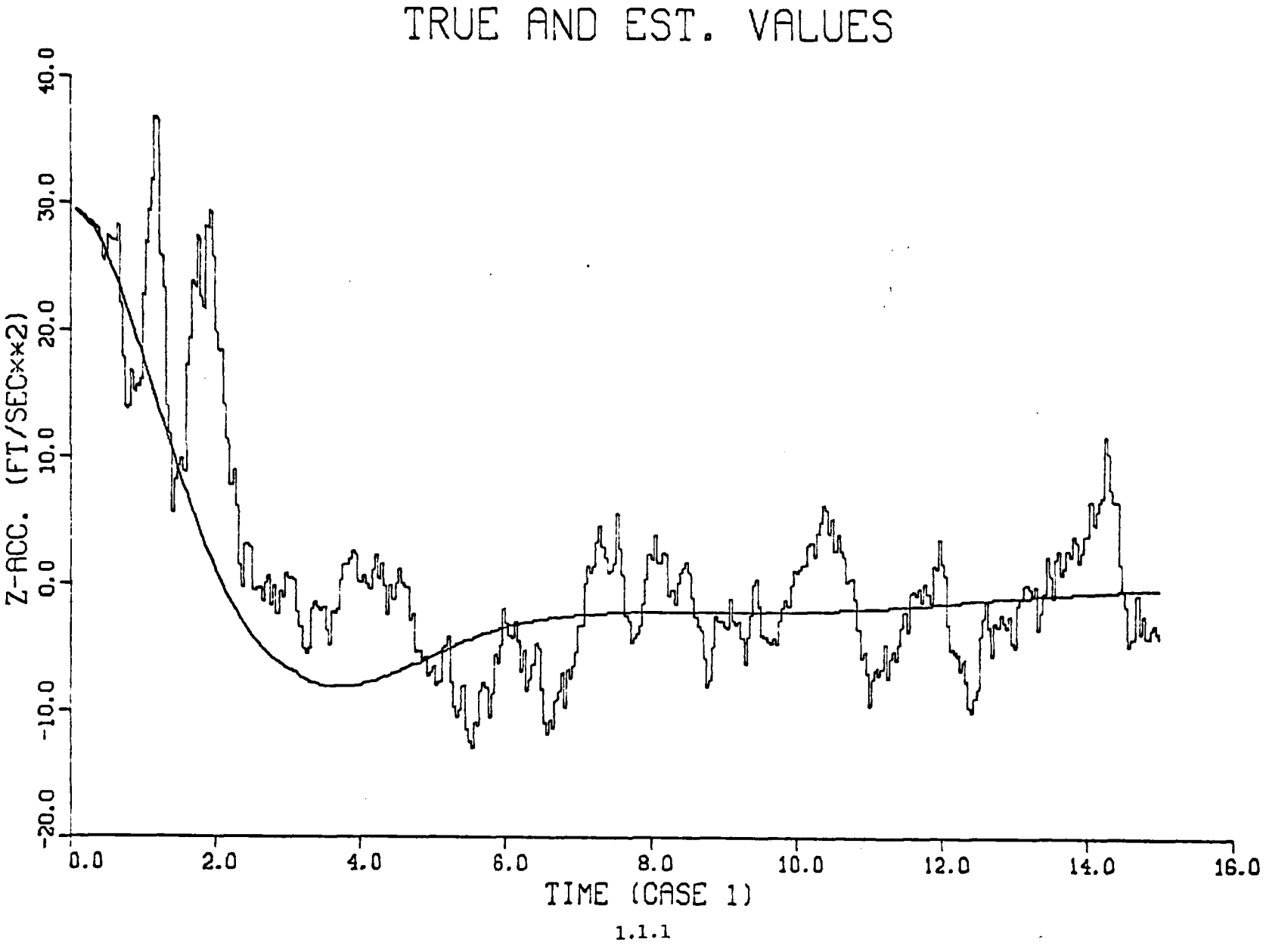
1.1.1

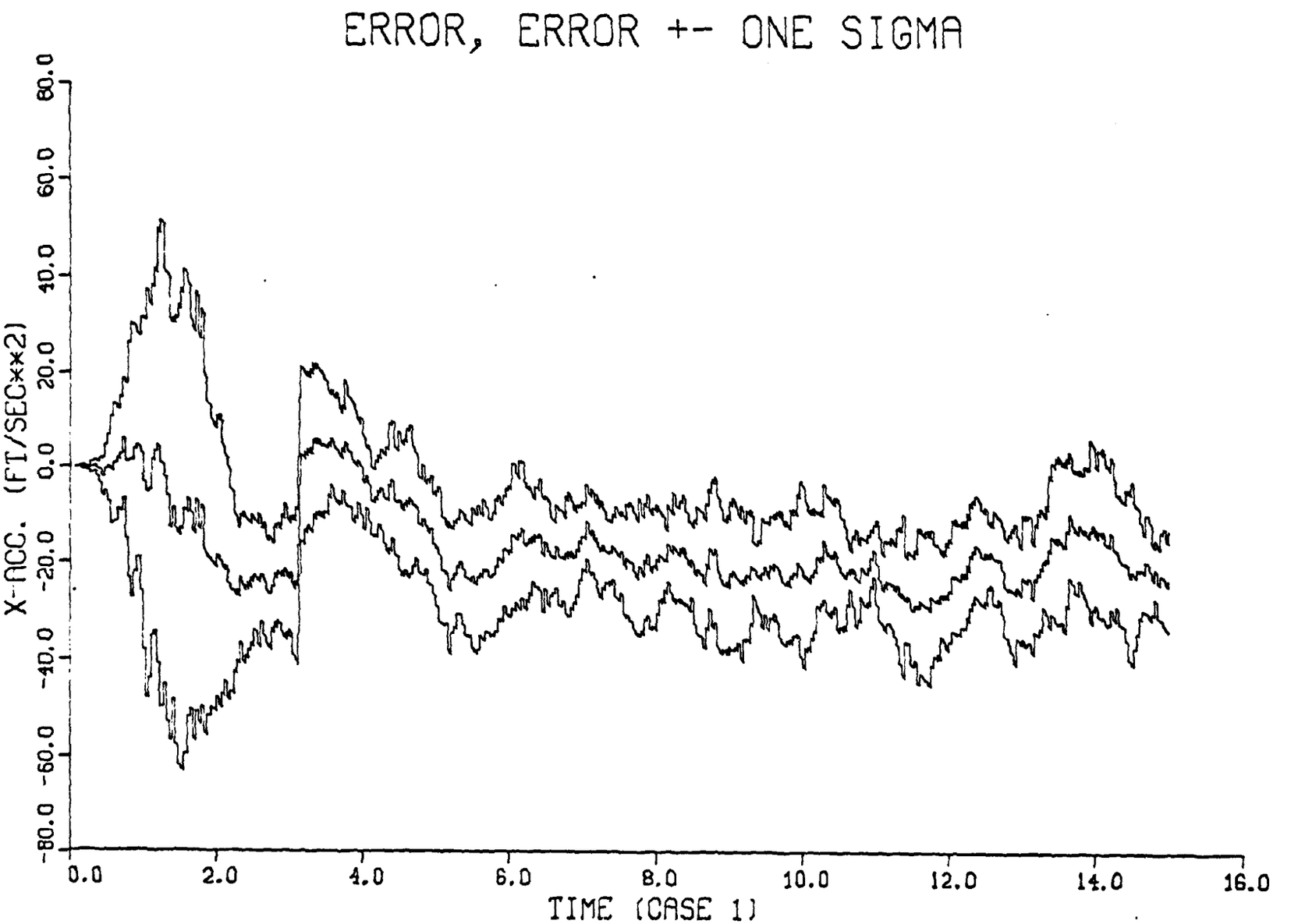

TRUE AND EST. VALUES
Z-POS. (FT)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

1.1.1

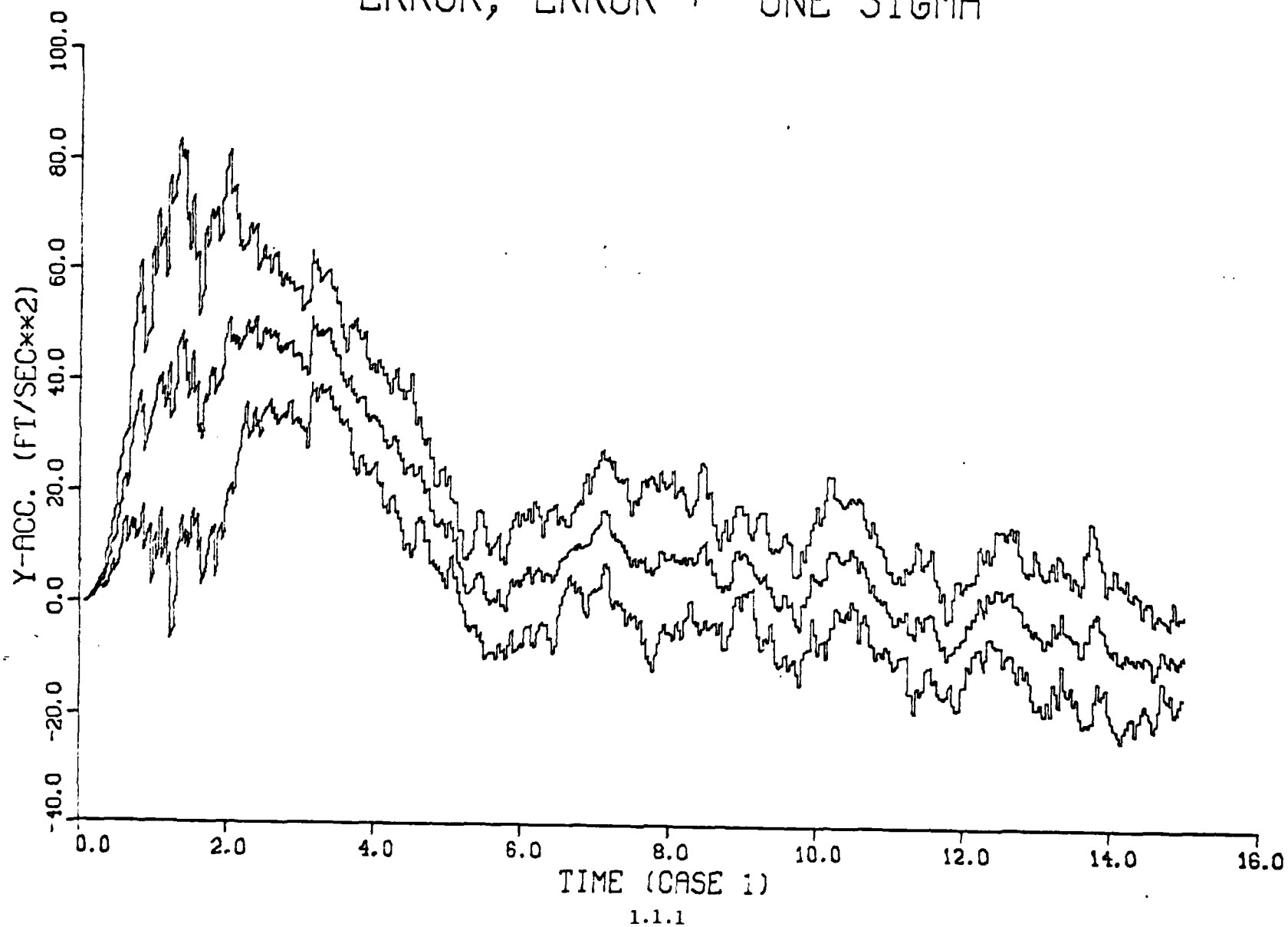

TRUE AND EST. VALUES
Z-VEL. (FT/SEC)
200.0
150.0
100.0
50.0
0.0
-50.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

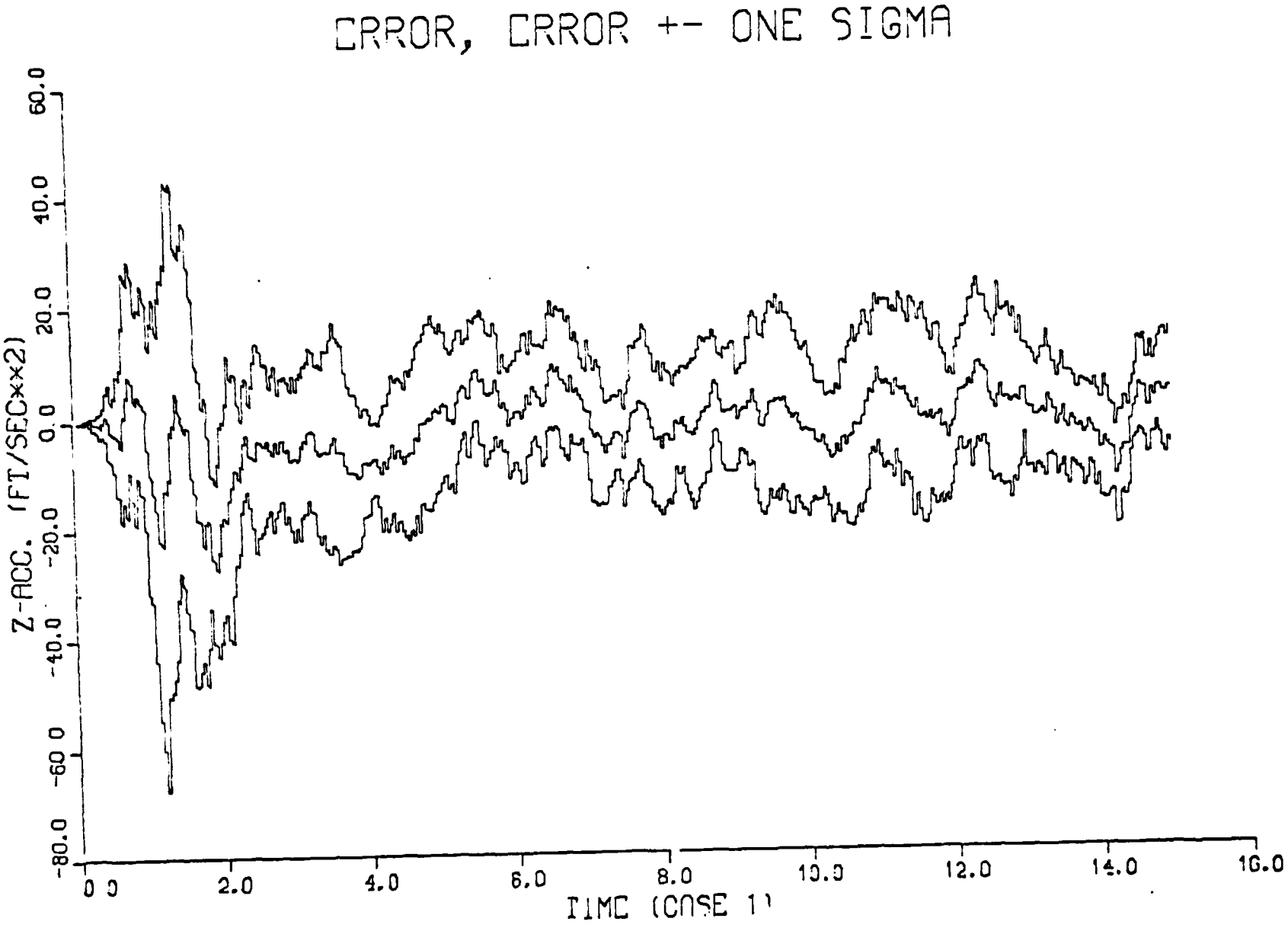
1.1.1


TRUE AND EST. VALUES
Z-ACC. (FT/SEC**2)
40.0
30.0
20.0
10.0
0.0
-10.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)
1.1.1

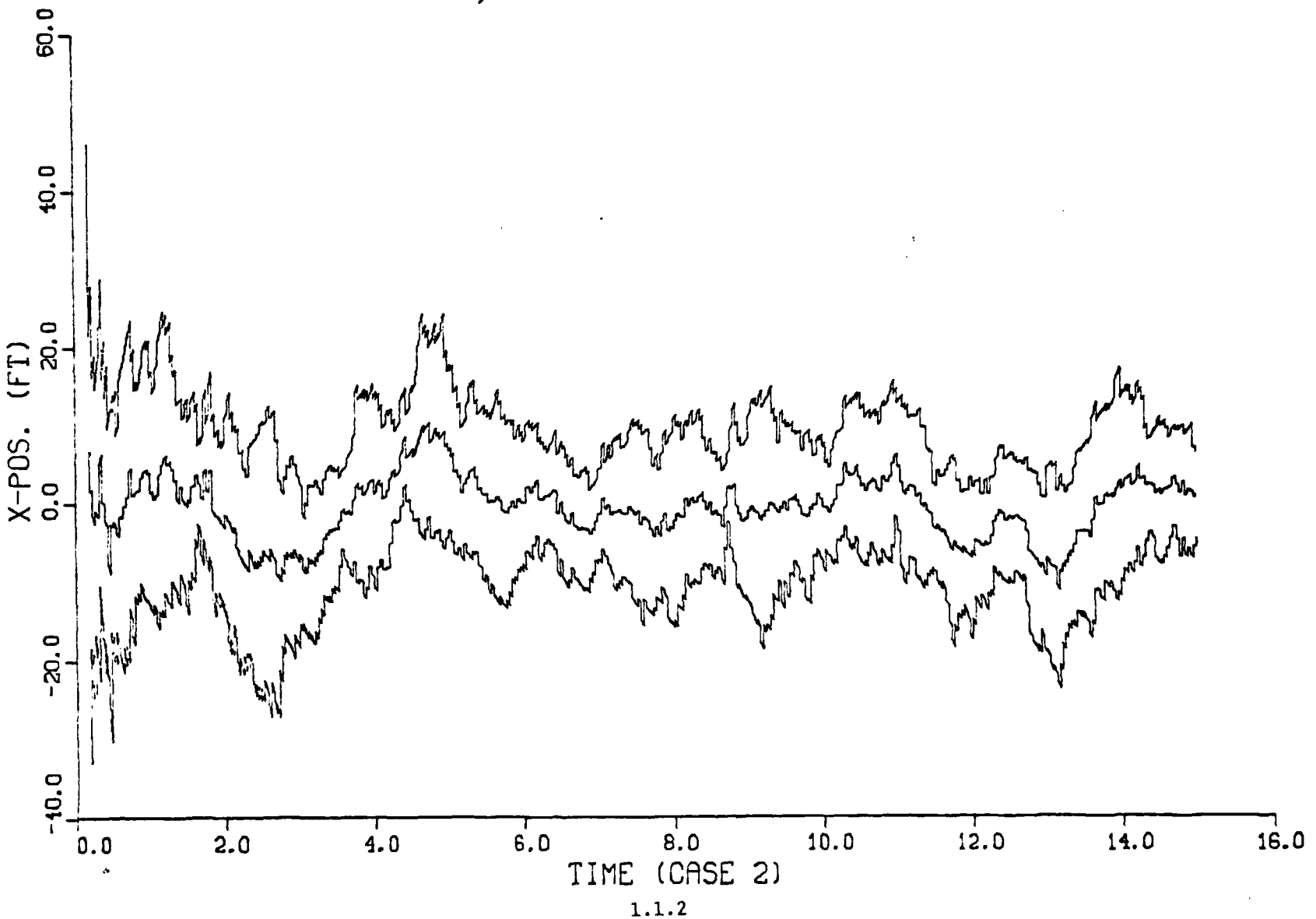

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

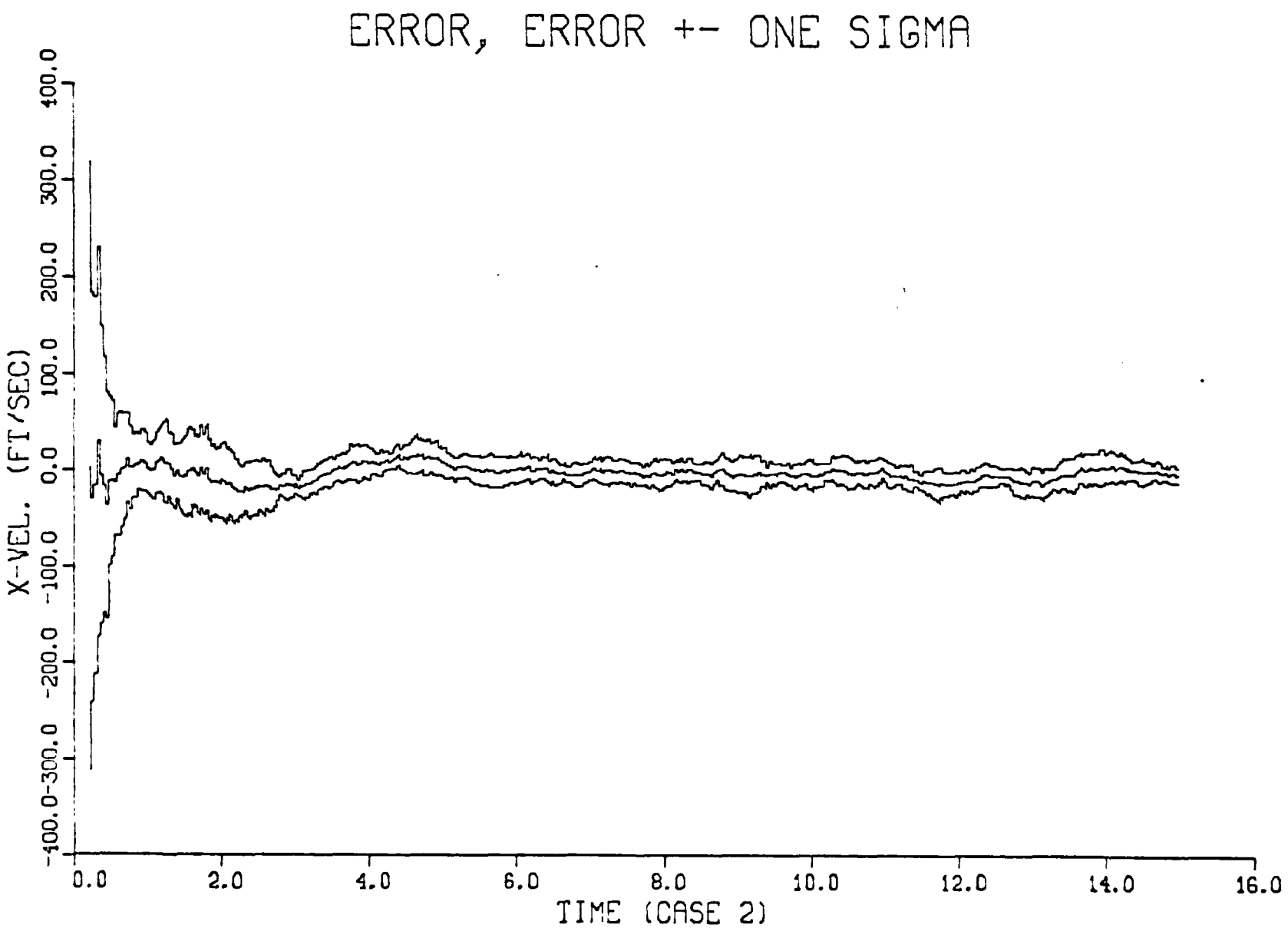
1.1.1

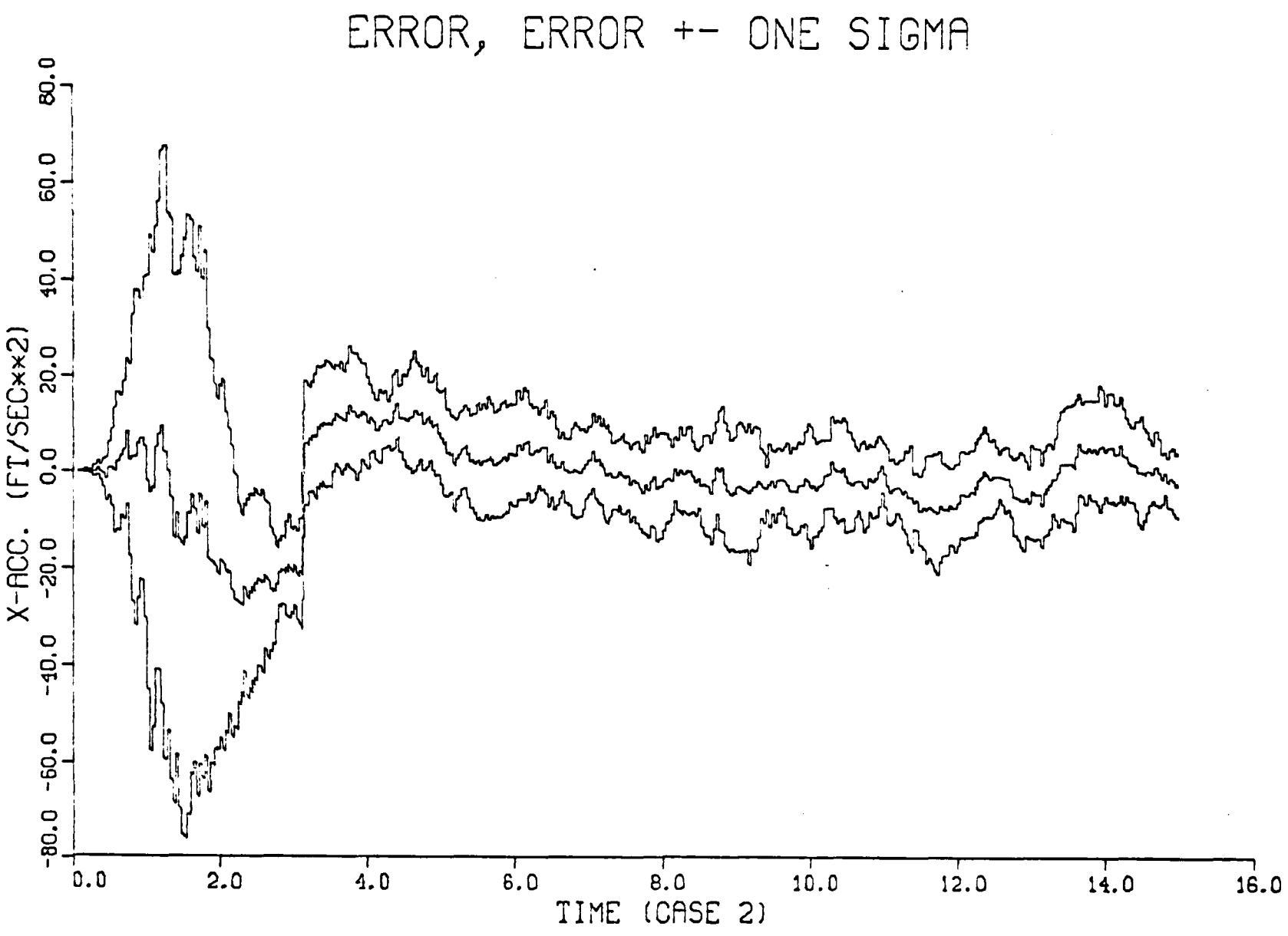

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)
1.1.1

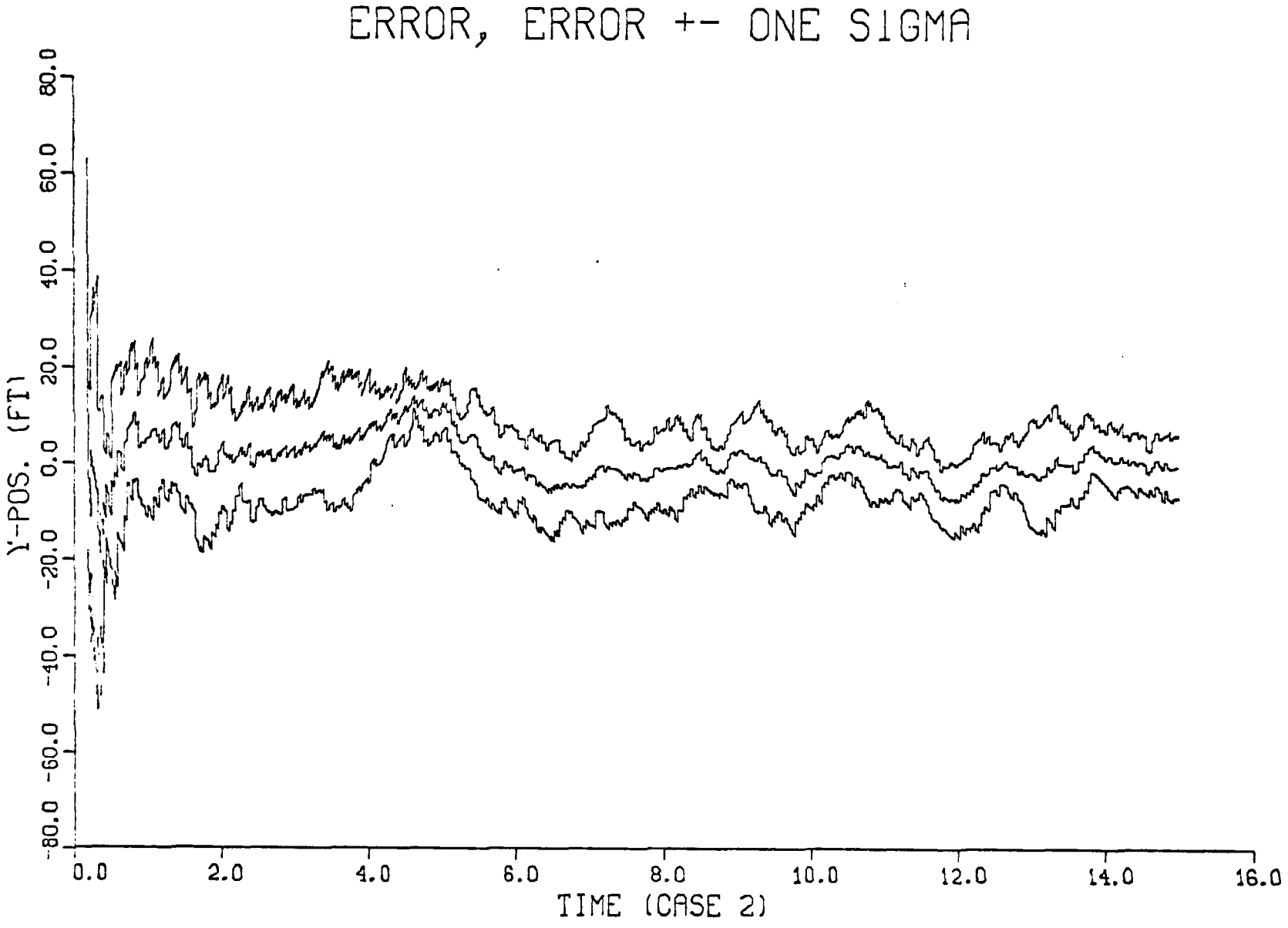

ERROR, ERROR +- ONE SIGMA
Z-ACC. (FT/SEC**2)
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 1)

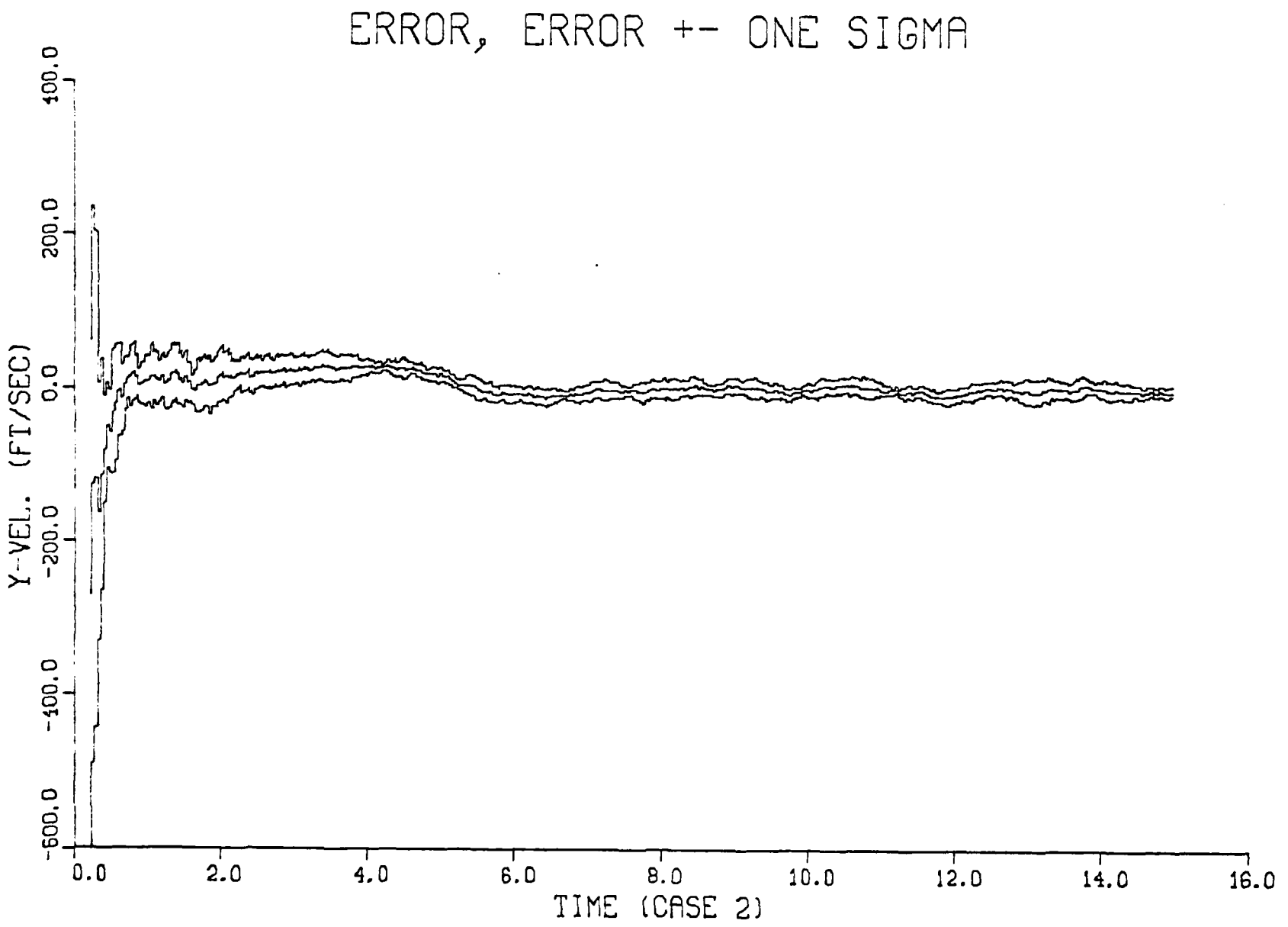
1.1.1


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)
1.1.2

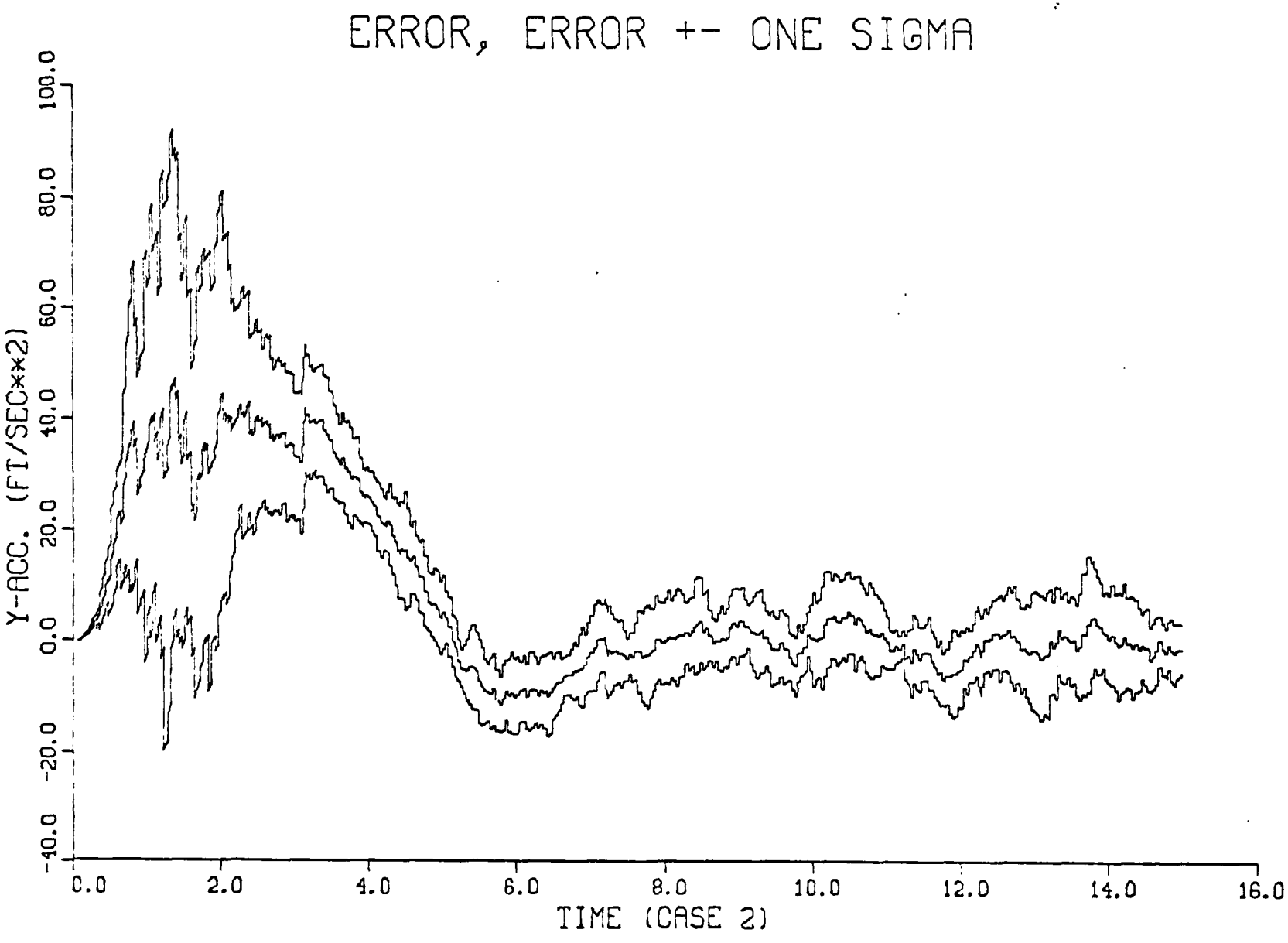

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

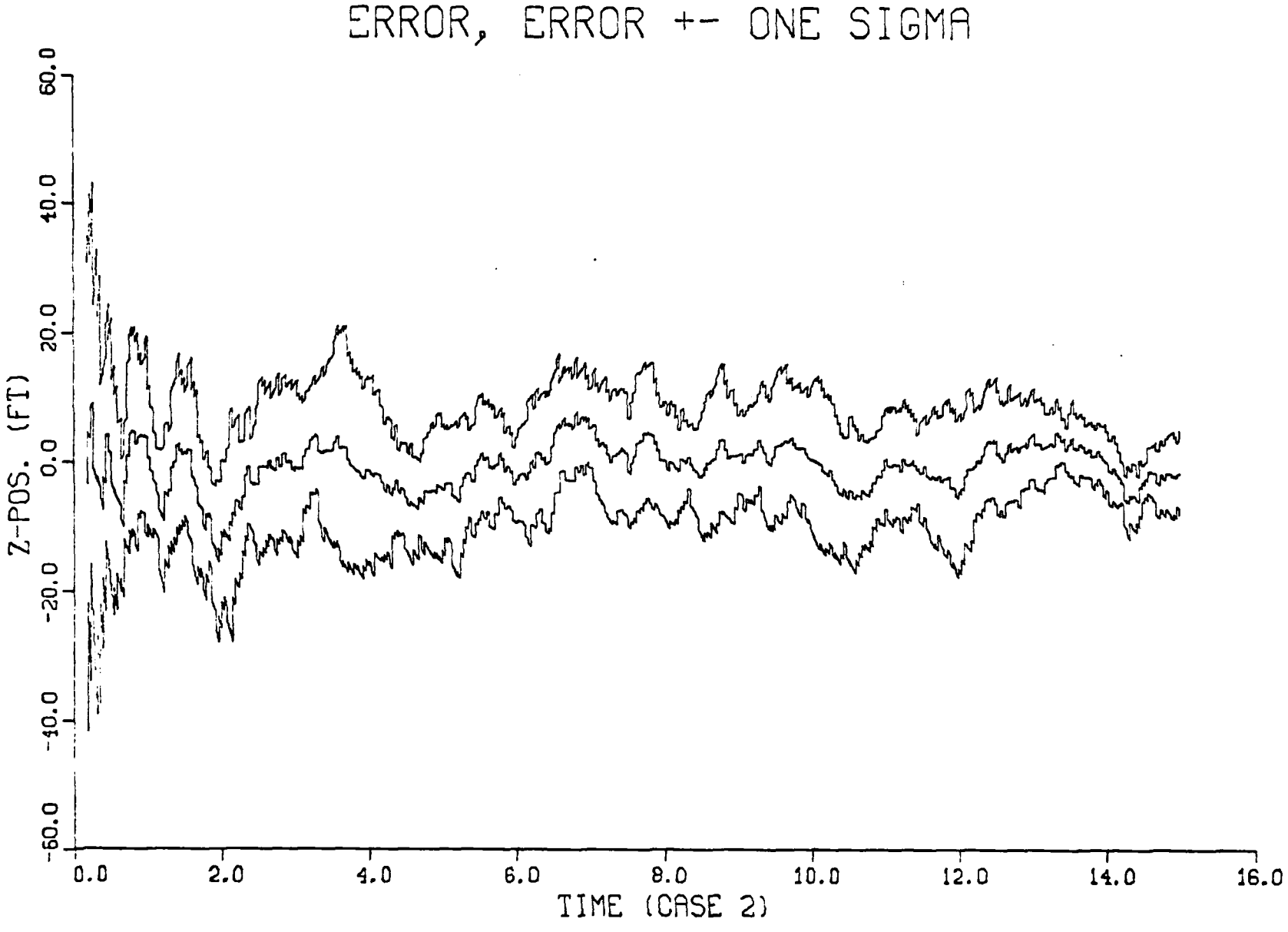

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
100.0
200.0
0.0
-200.0
-100.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

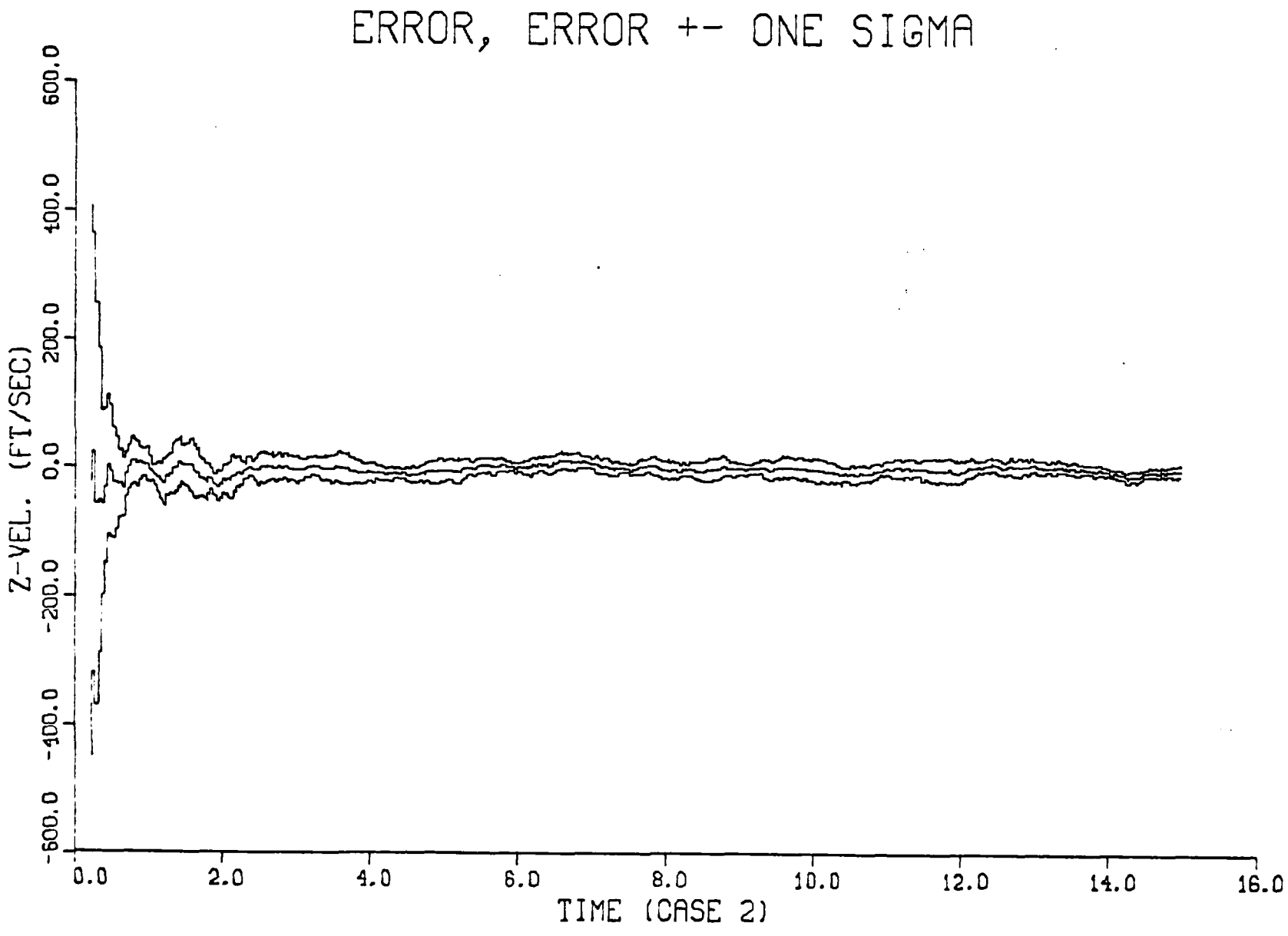
1.1.2


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

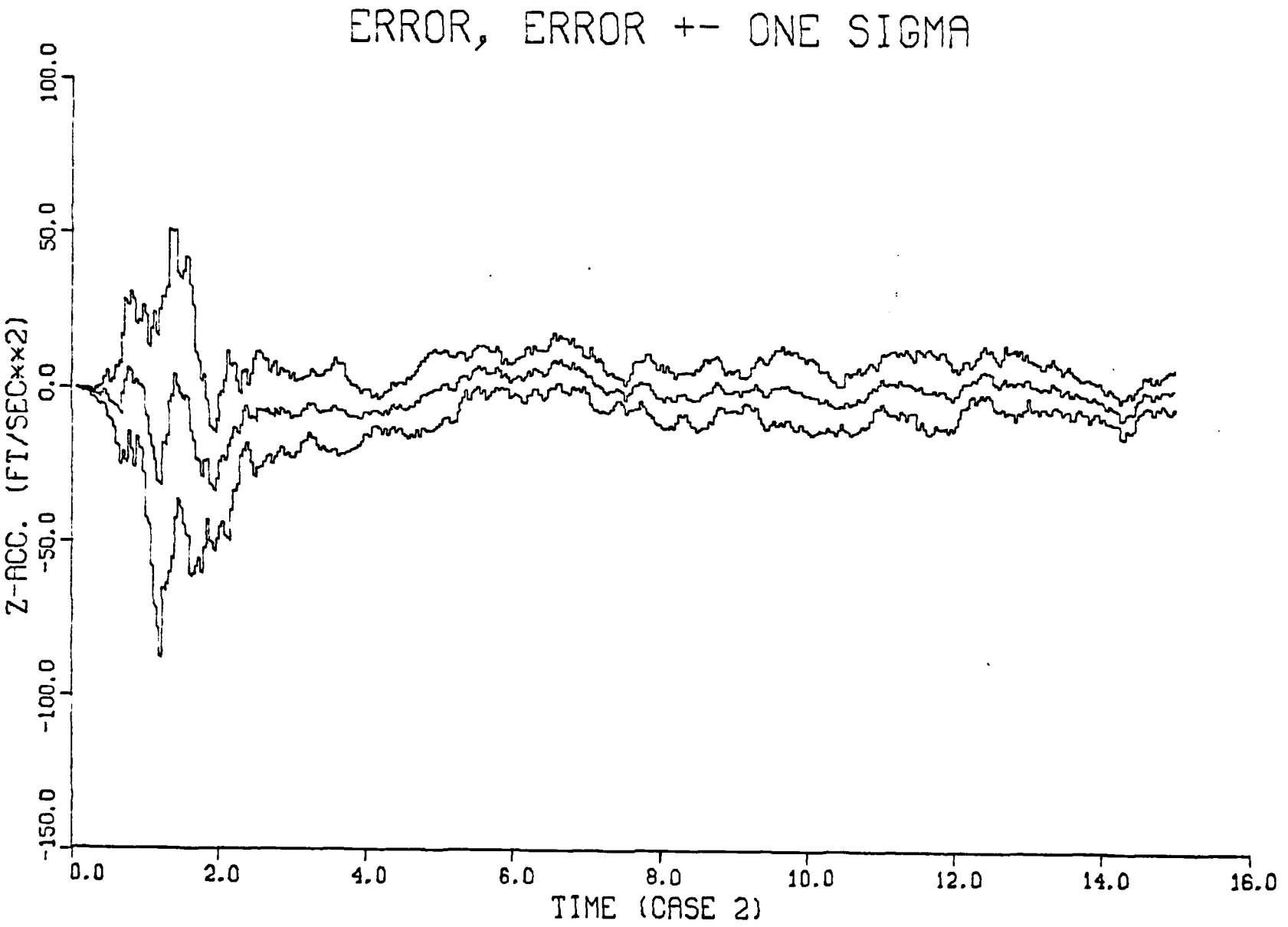
1.1.2


ERROR, ERROR +- ONE SIGMA
Z-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

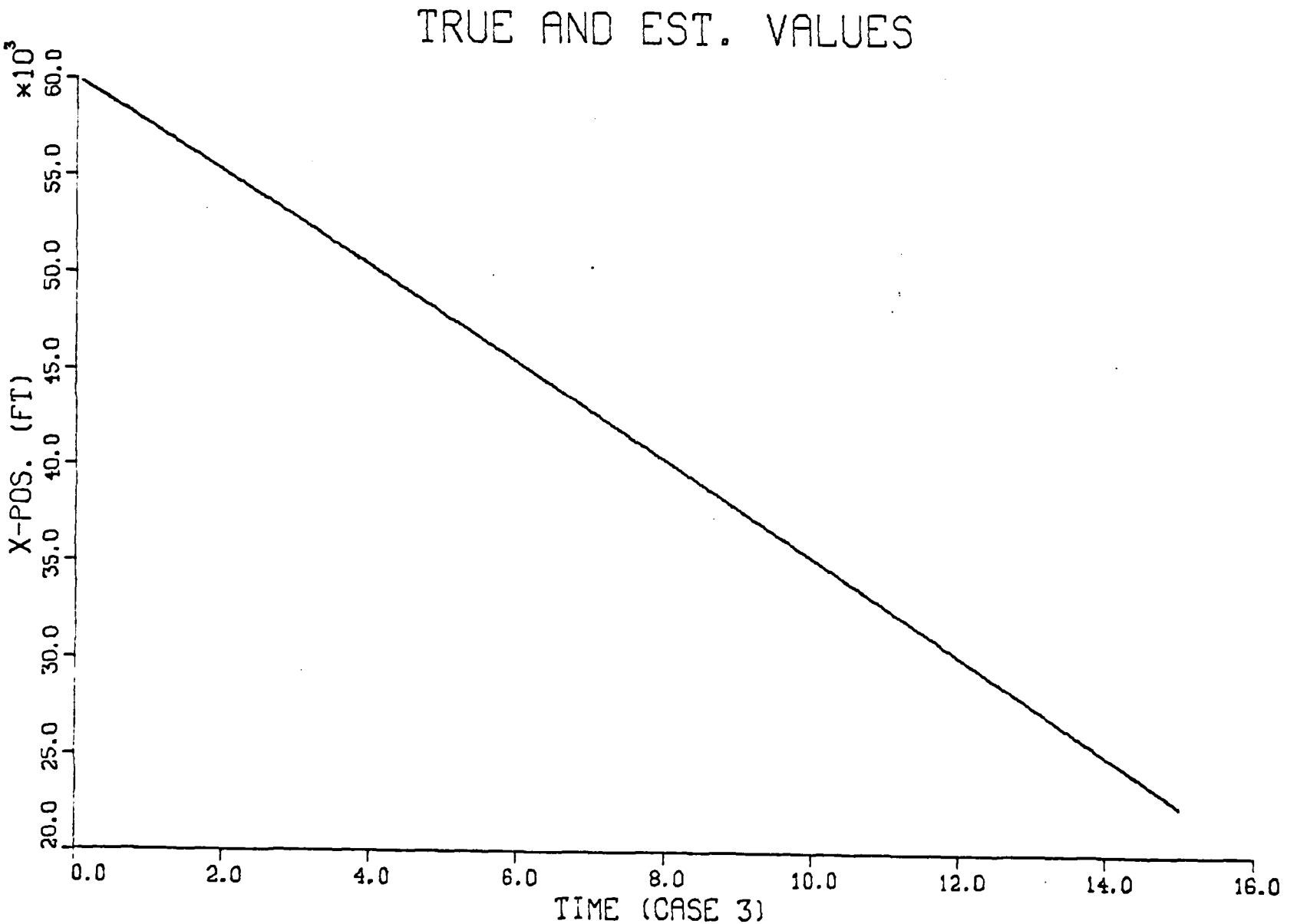
1.1.2

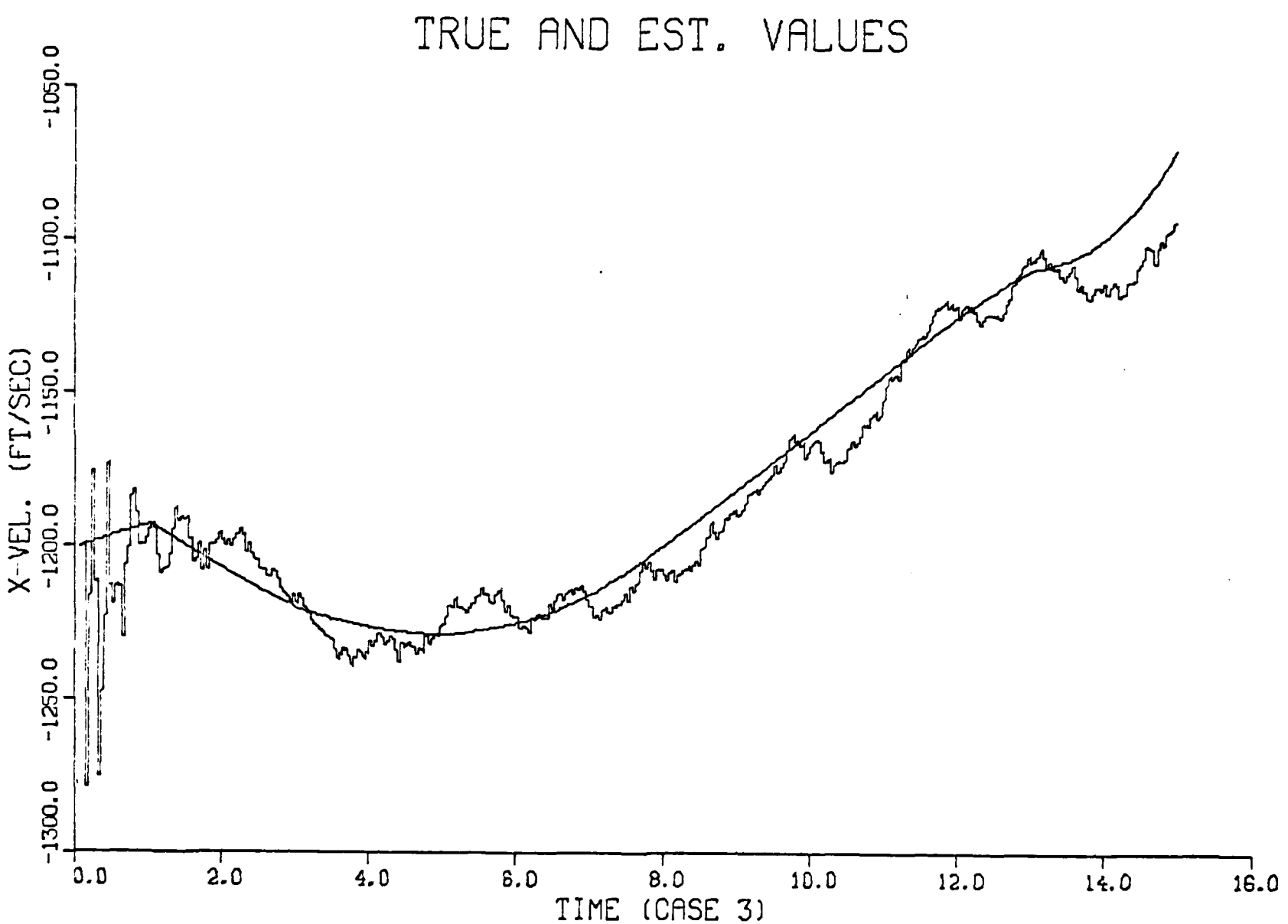

ERROR, ERROR +- ONE SIGMA
Z-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

1.1.2

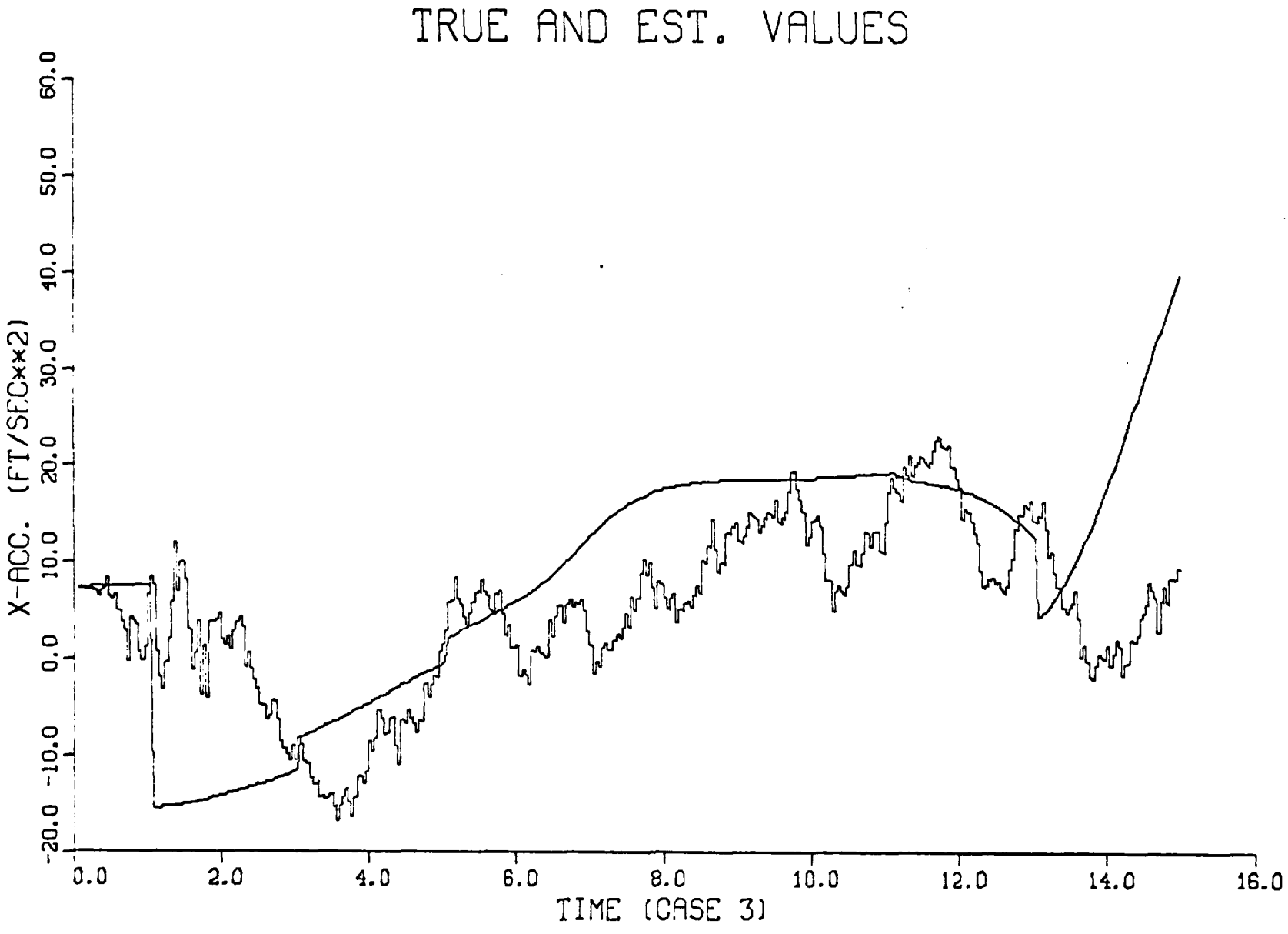

ERROR, ERROR +- ONE SIGMA
Z-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 2)

1.1.2

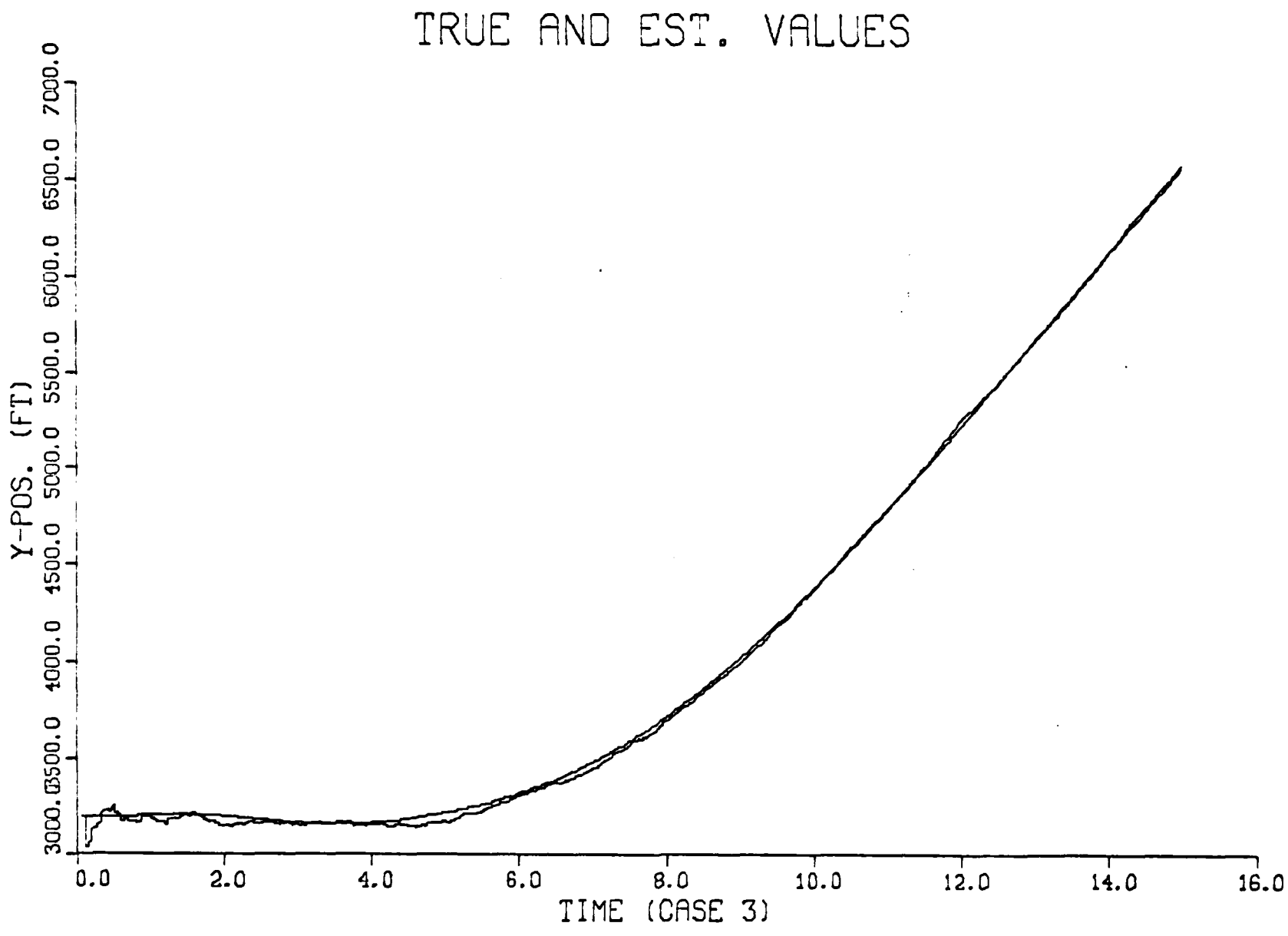

TRUE AND EST. VALUES
X-POS. (FT)
*10^3
60.0
55.0
50.0
45.0
40.0
35.0
30.0
25.0
20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)


TRUE AND EST. VALUES
X-VEL. (FT/SEC)
-1050.0
-1100.0
-1150.0
-1200.0
-1250.0
-1300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

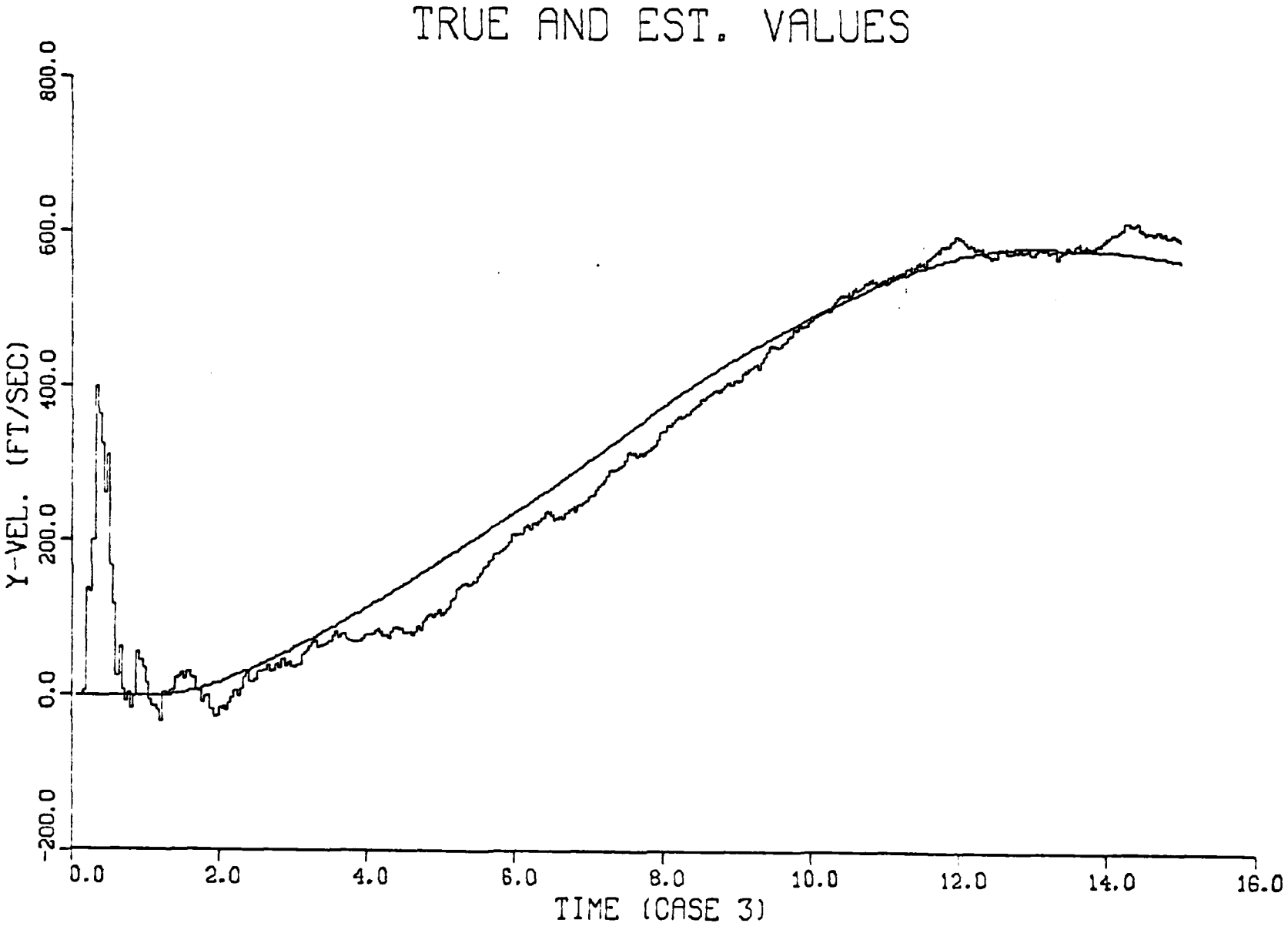
1.1.3


TRUE AND EST. VALUES
X-ACC. (FT/SEC**2)
60.0
50.0
40.0
30.0
20.0
10.0
0.0
-10.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

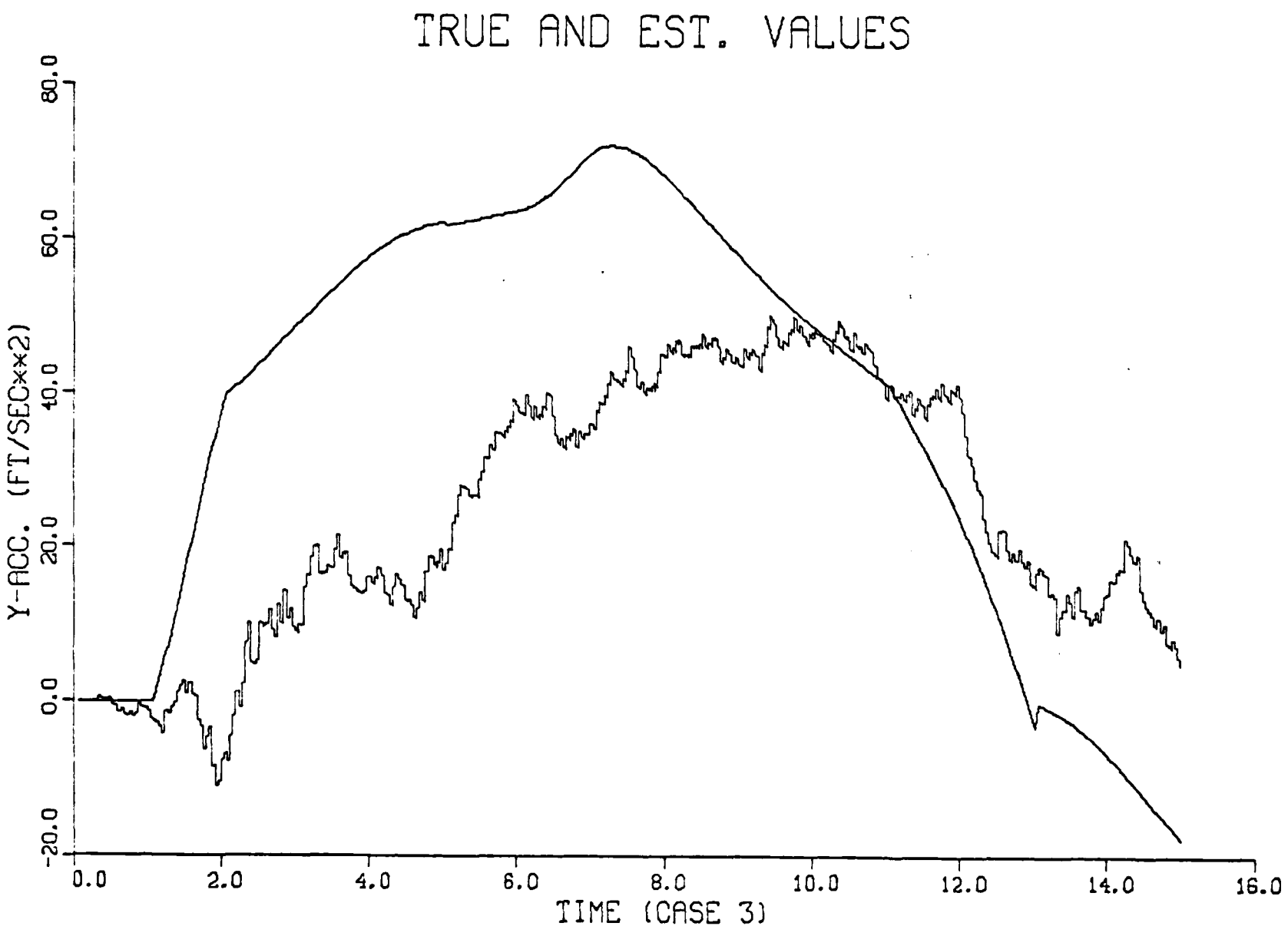
1.1.3


TRUE AND EST. VALUES
Y-POS. (FT)
3000.0
3500.0
4000.0
4500.0
5000.0
5500.0
6000.0
6500.0
7000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

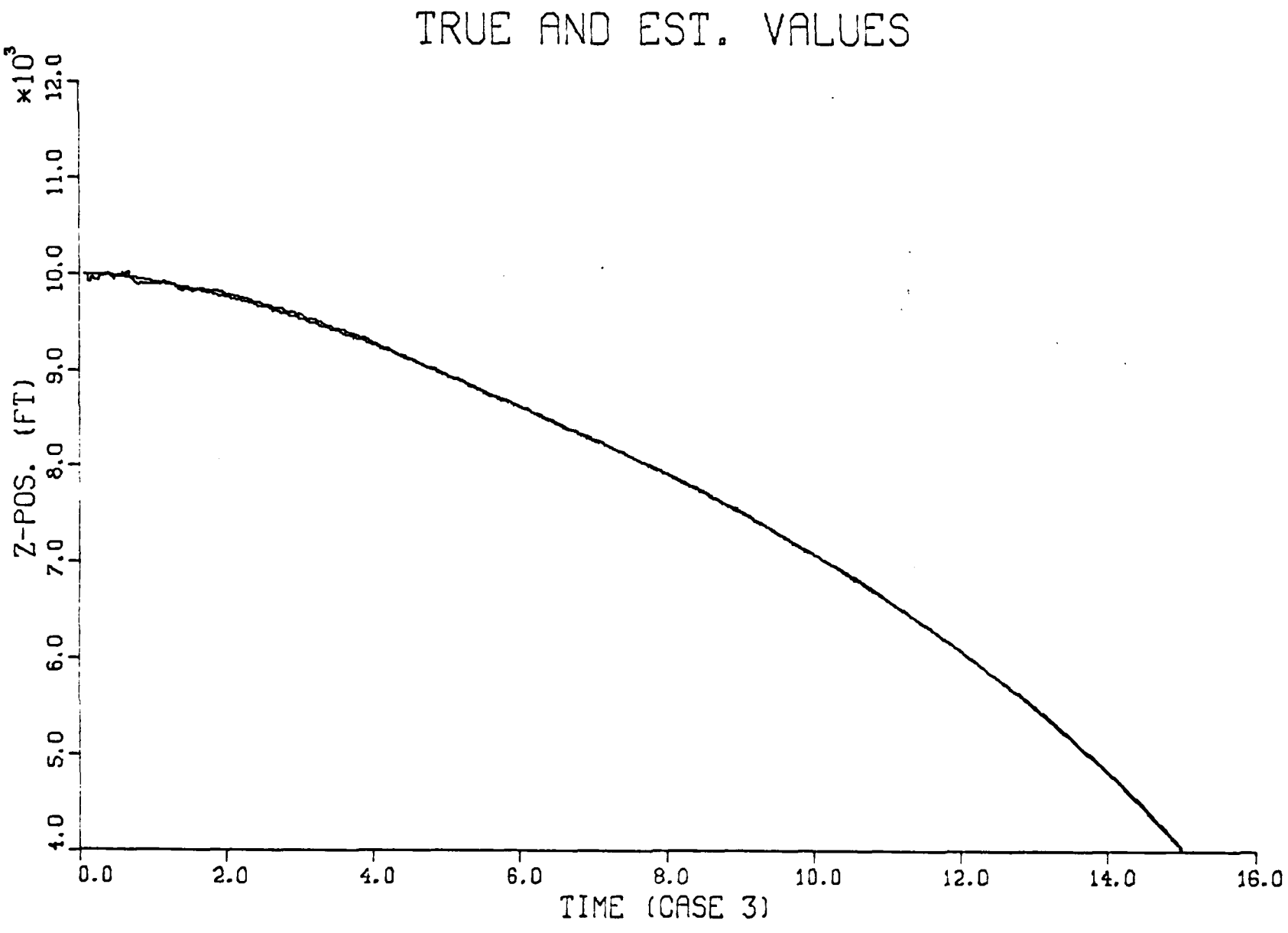
1.1.3

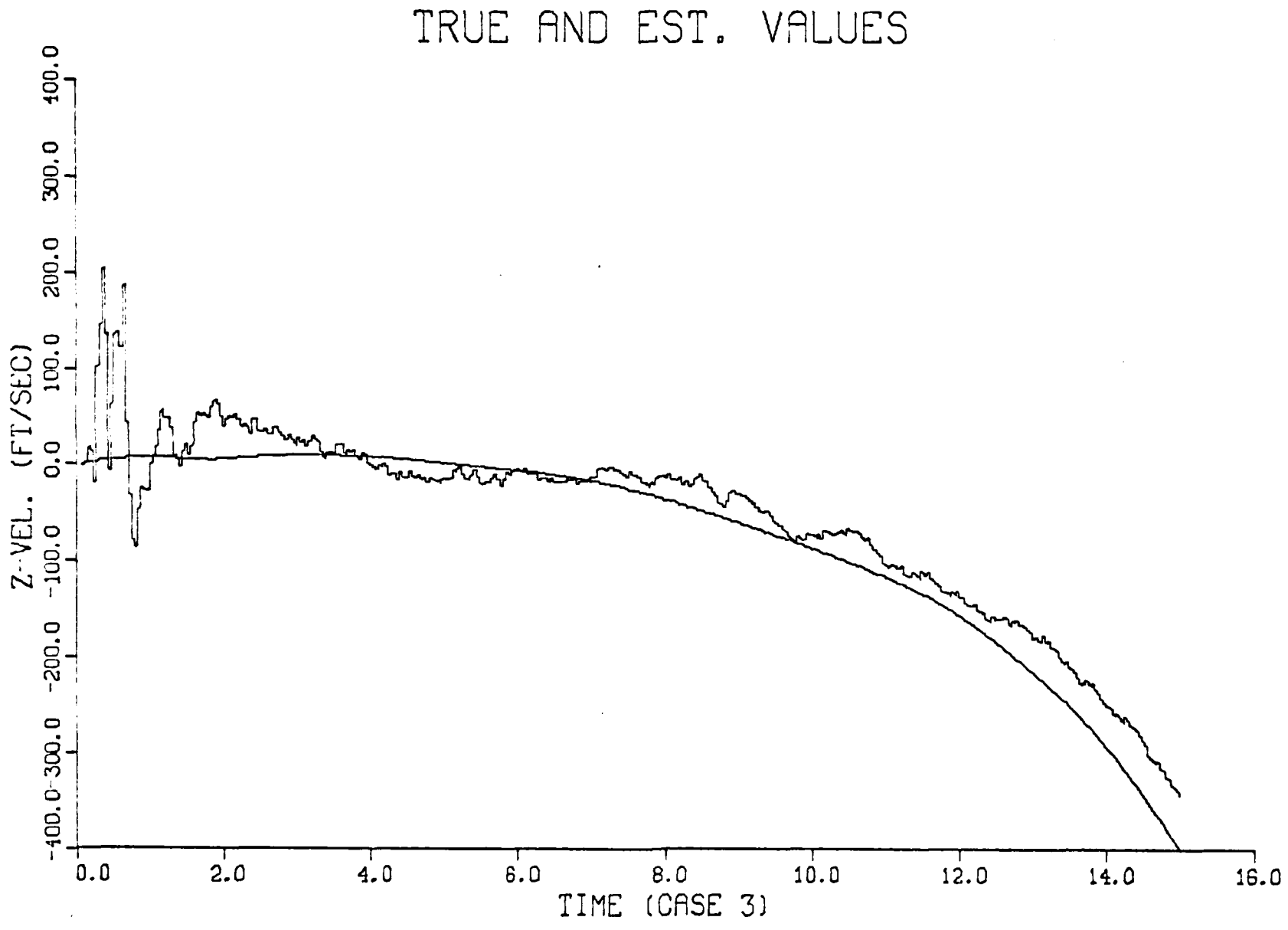

TRUE AND EST. VALUES
Y-VEL. (FT/SEC)
800.0
600.0
400.0
200.0
0.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

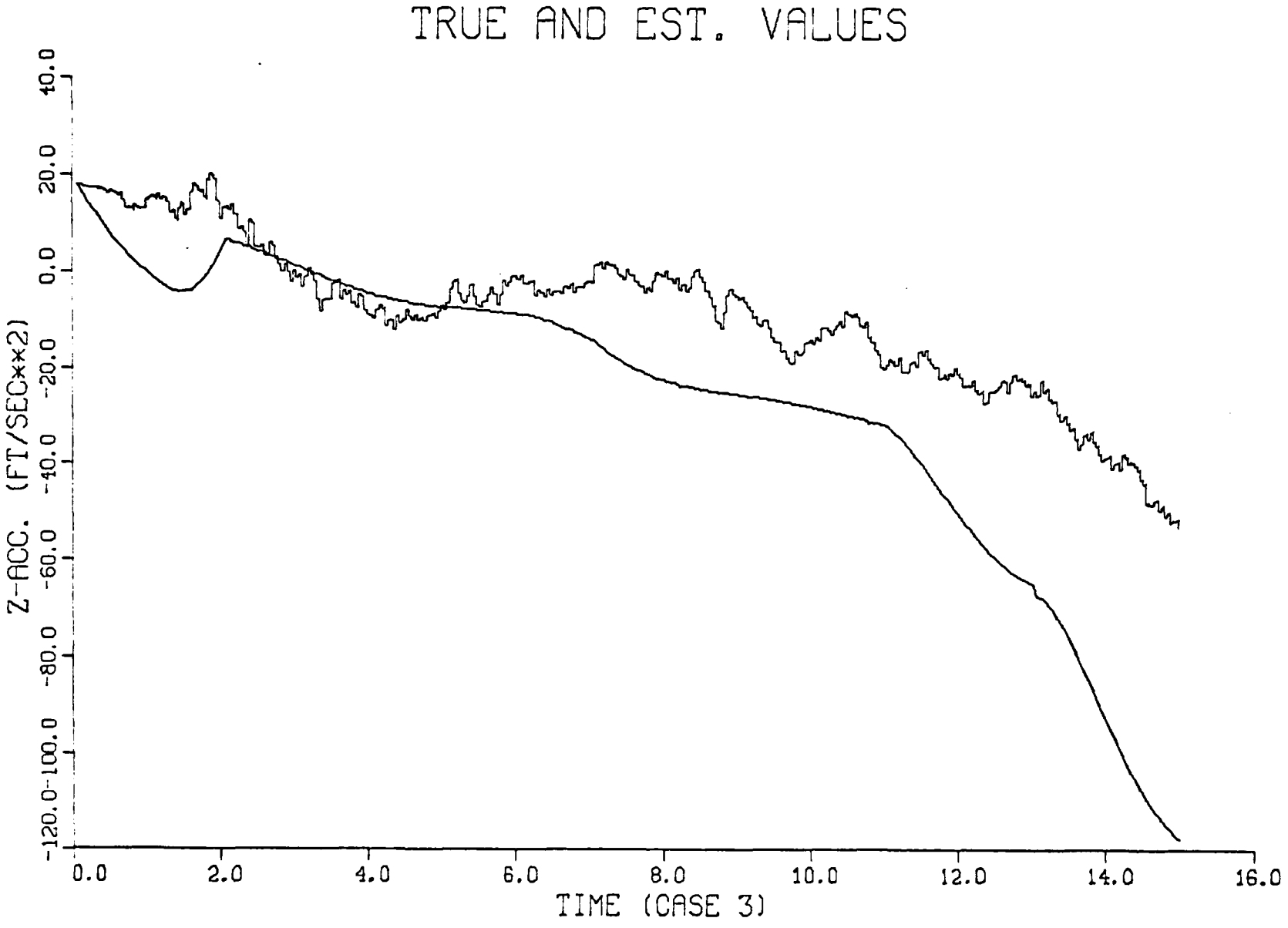
1.1.3

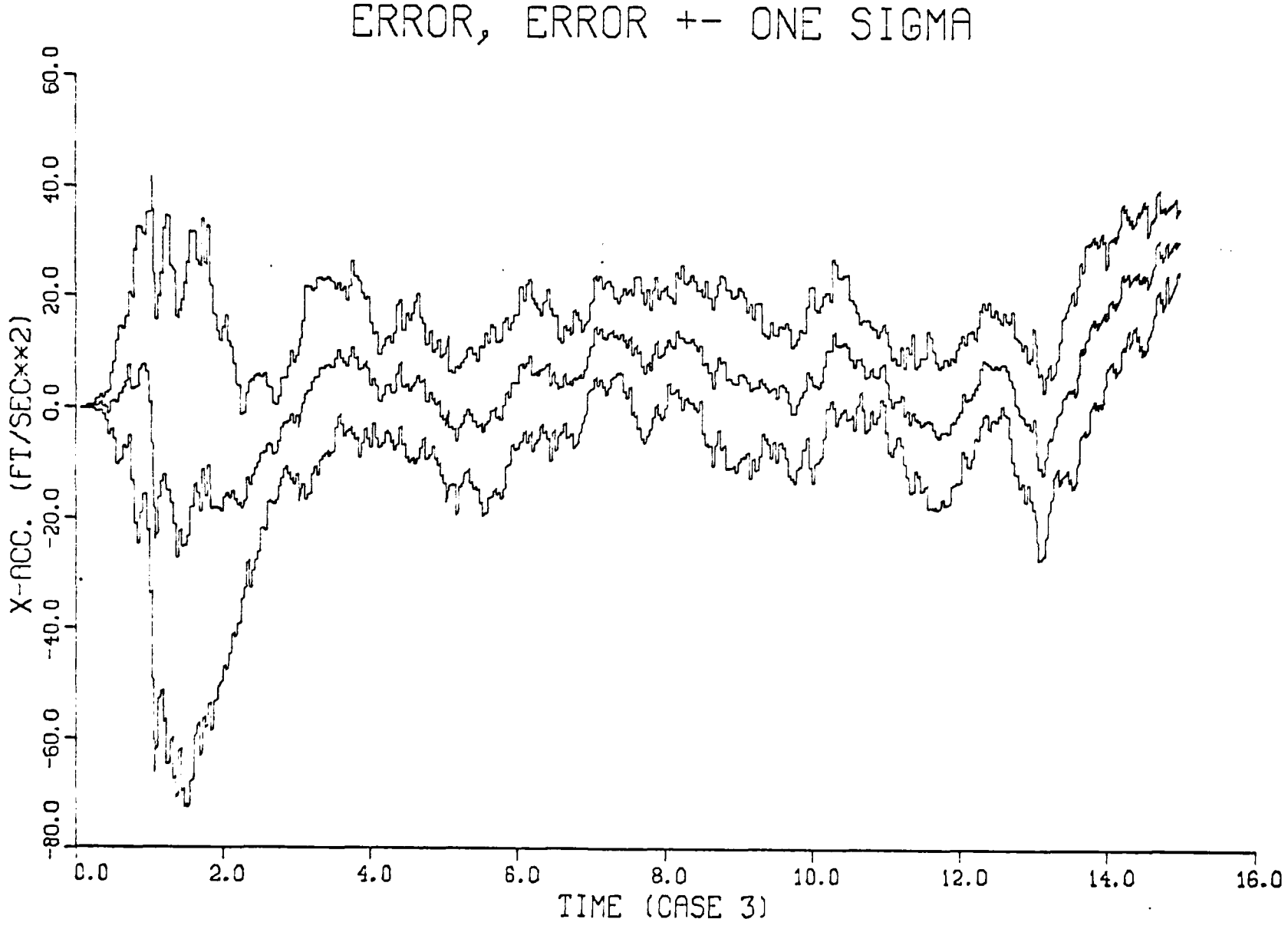

TRUE AND EST. VALUES
Y-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

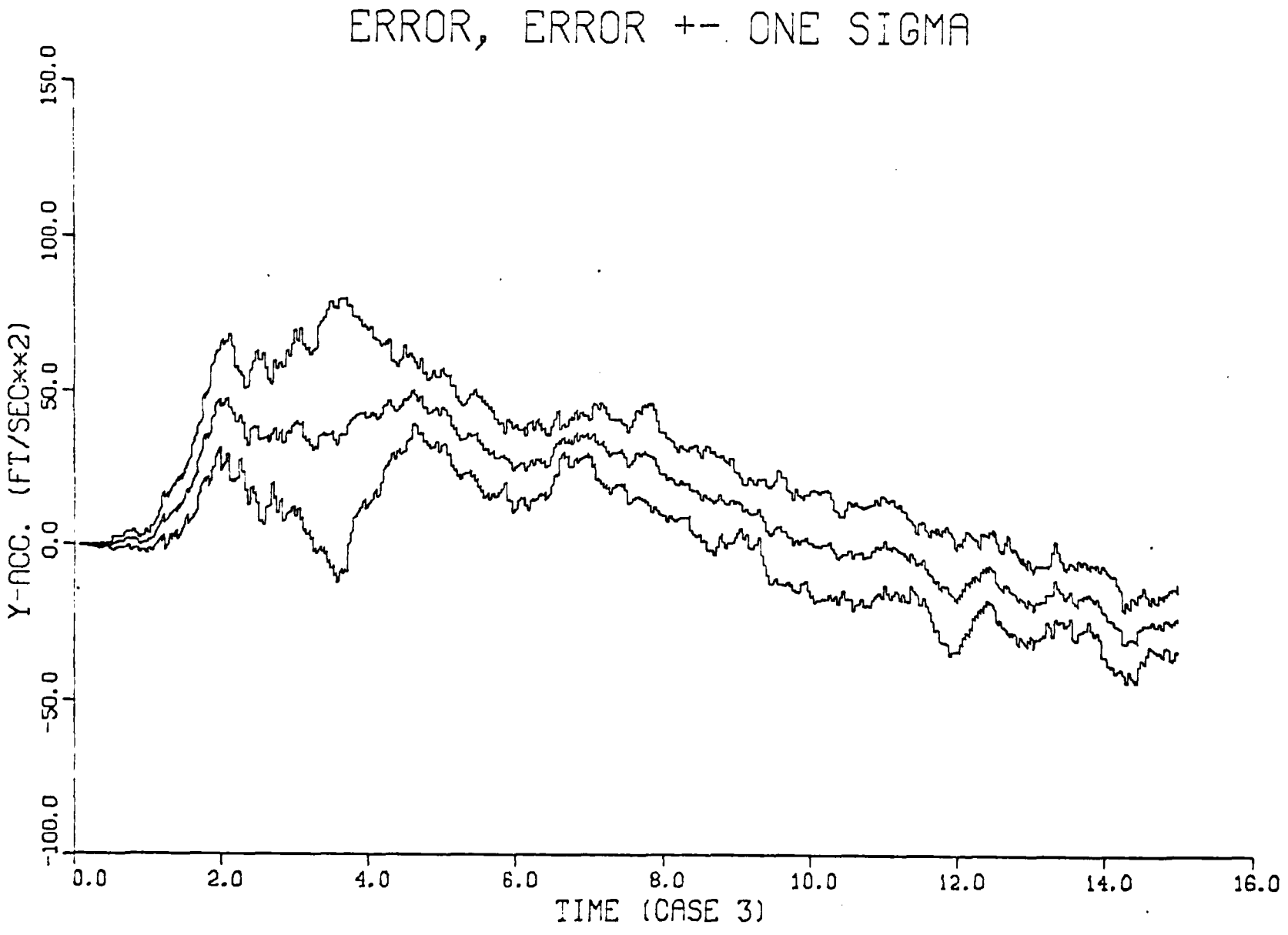
1.1.3


TRUE AND EST. VALUES
Z-POS. (FT)
×10³
12.0
11.0
10.0
9.0
8.0
7.0
6.0
5.0
4.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

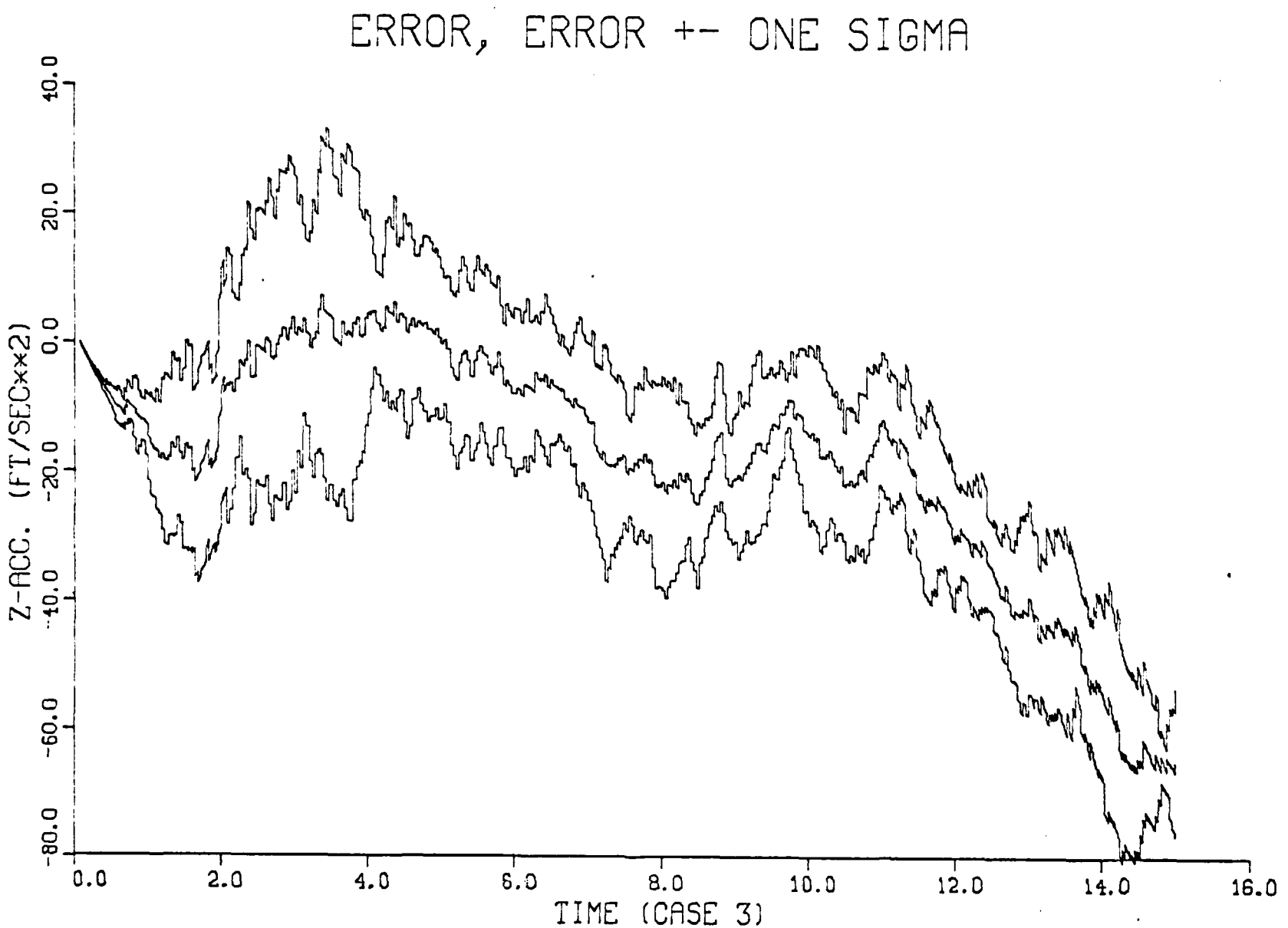

TRUE AND EST. VALUES
Z-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)


TRUE AND EST. VALUES
Z-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

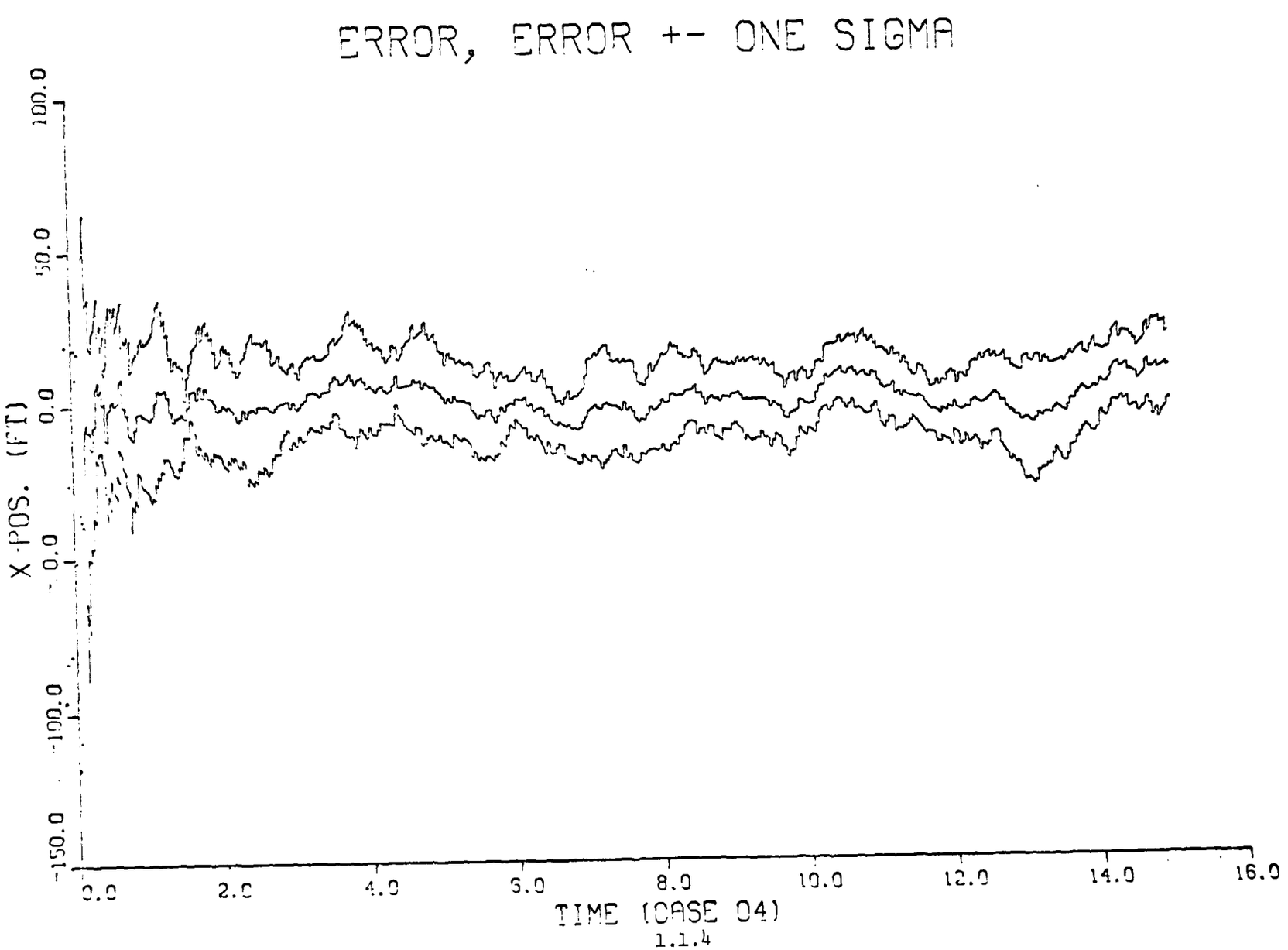

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

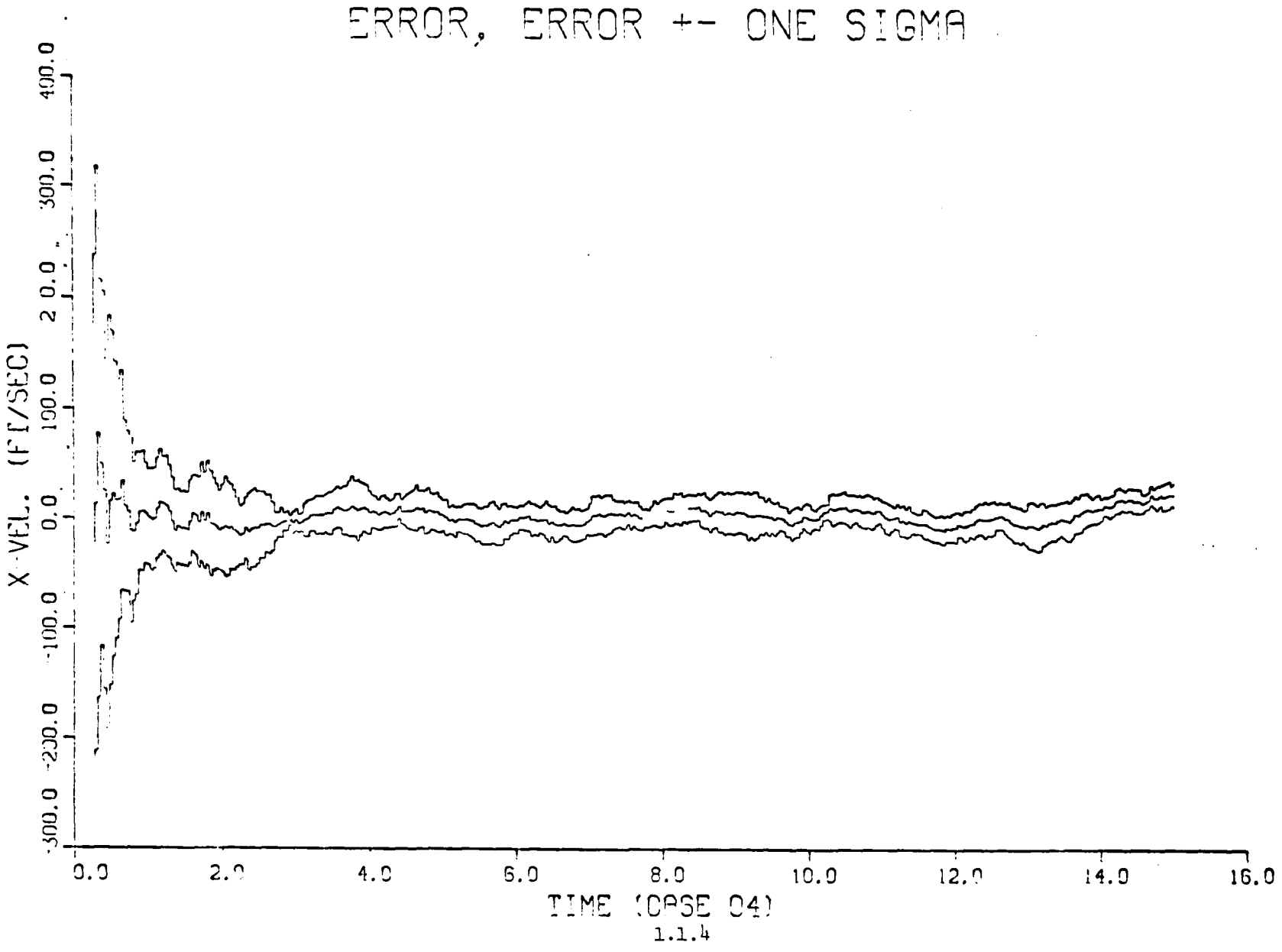

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

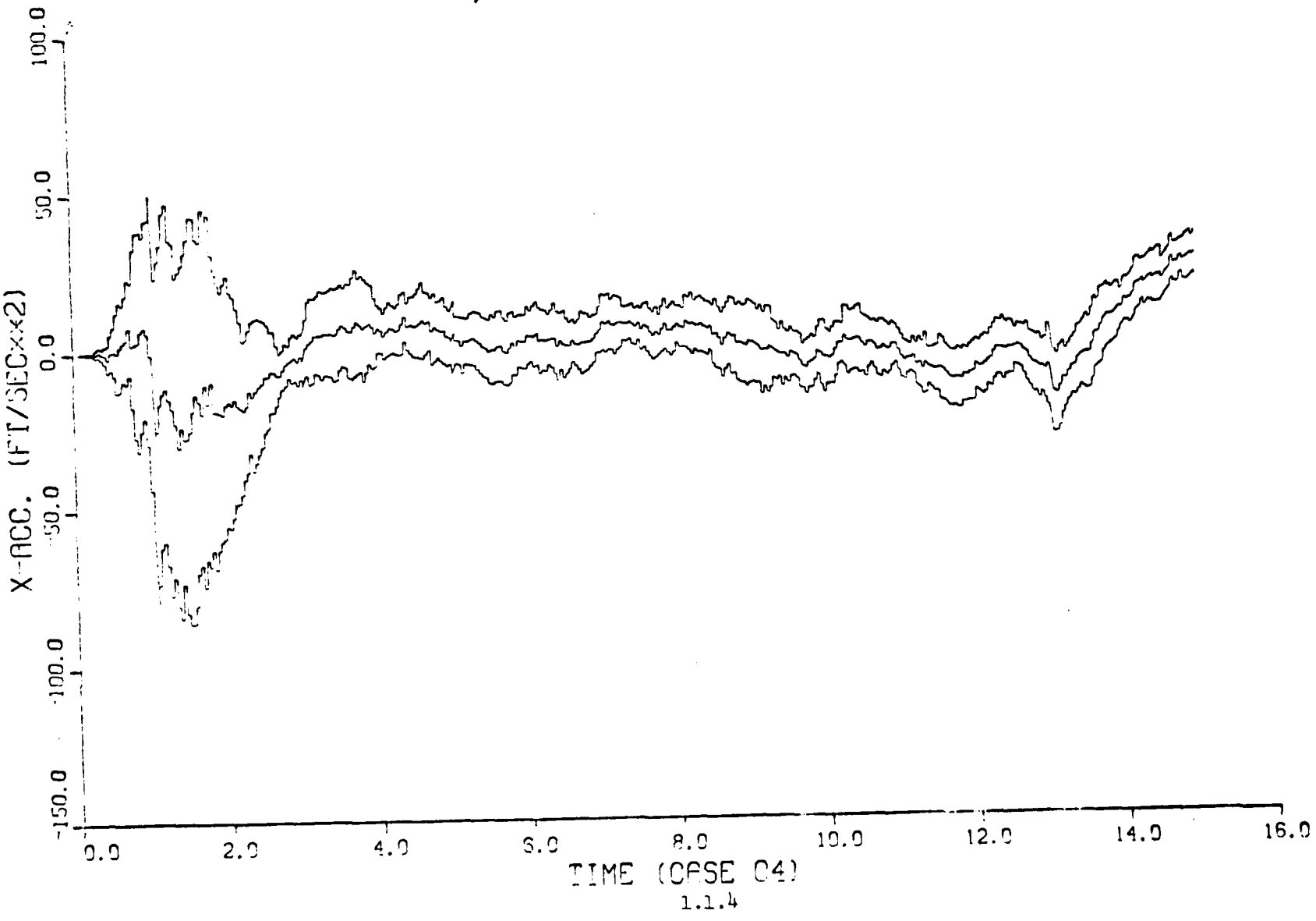
1.1.3


ERROR, ERROR +- ONE SIGMA
Z-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 3)

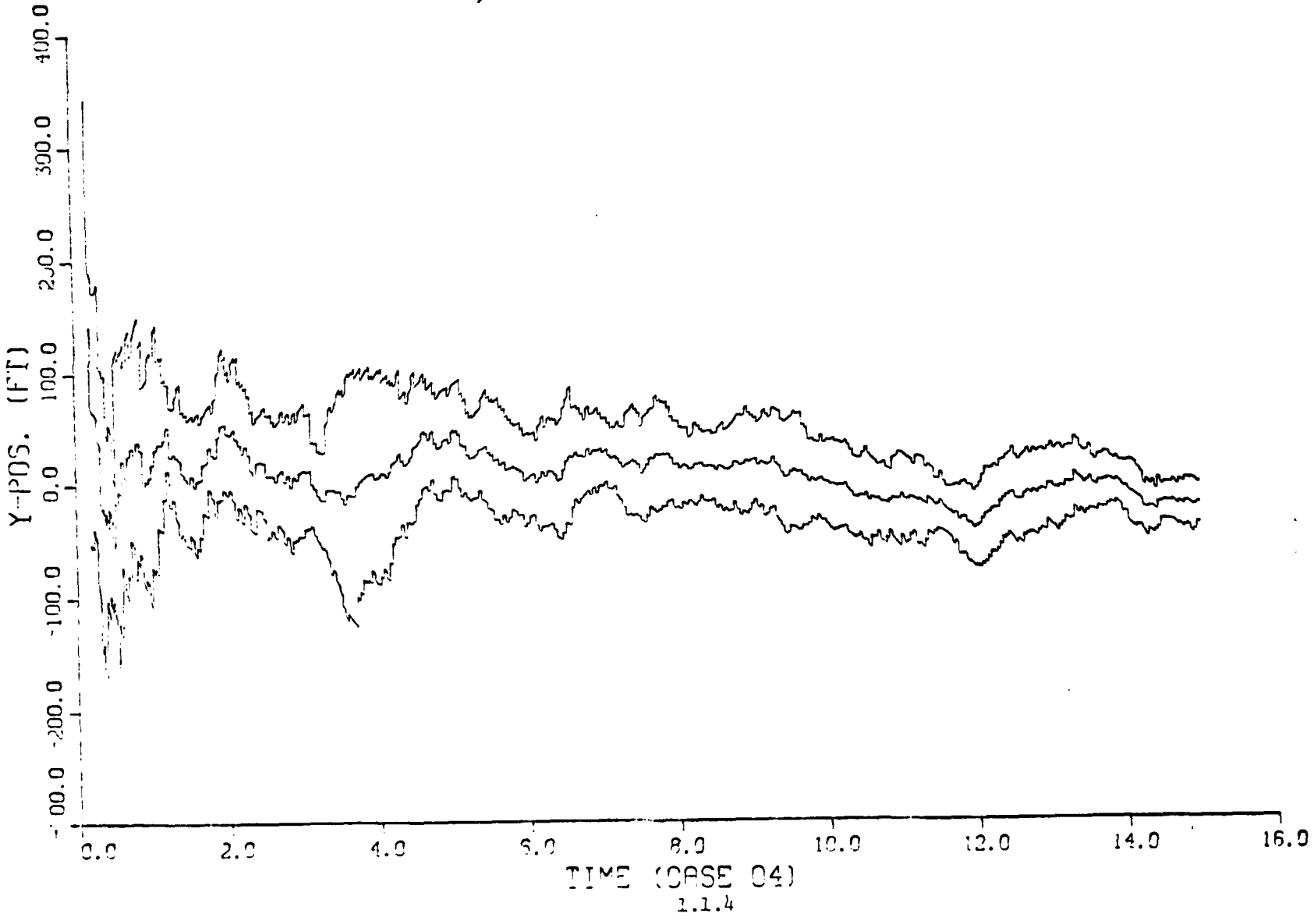
1.1.3


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

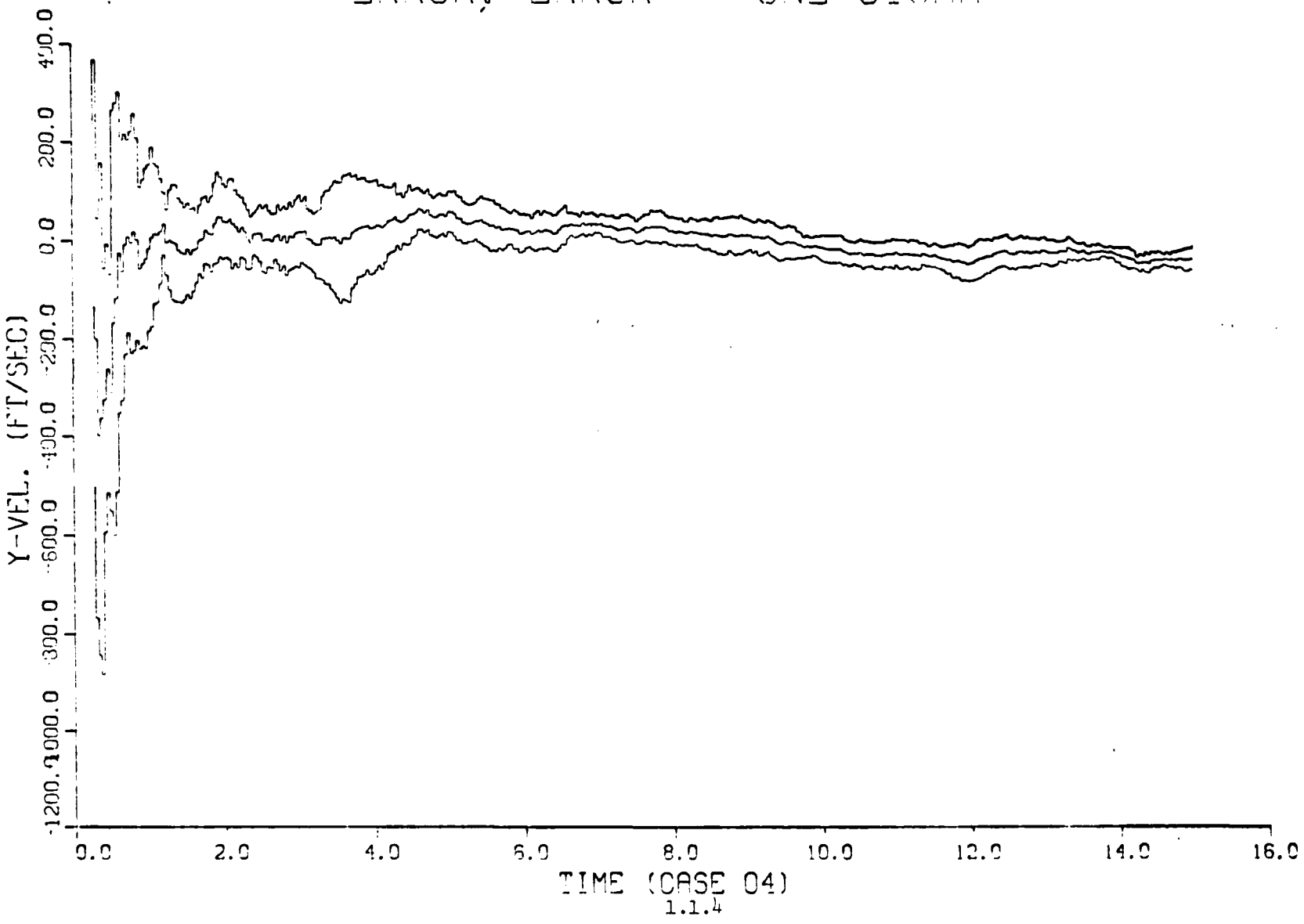

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

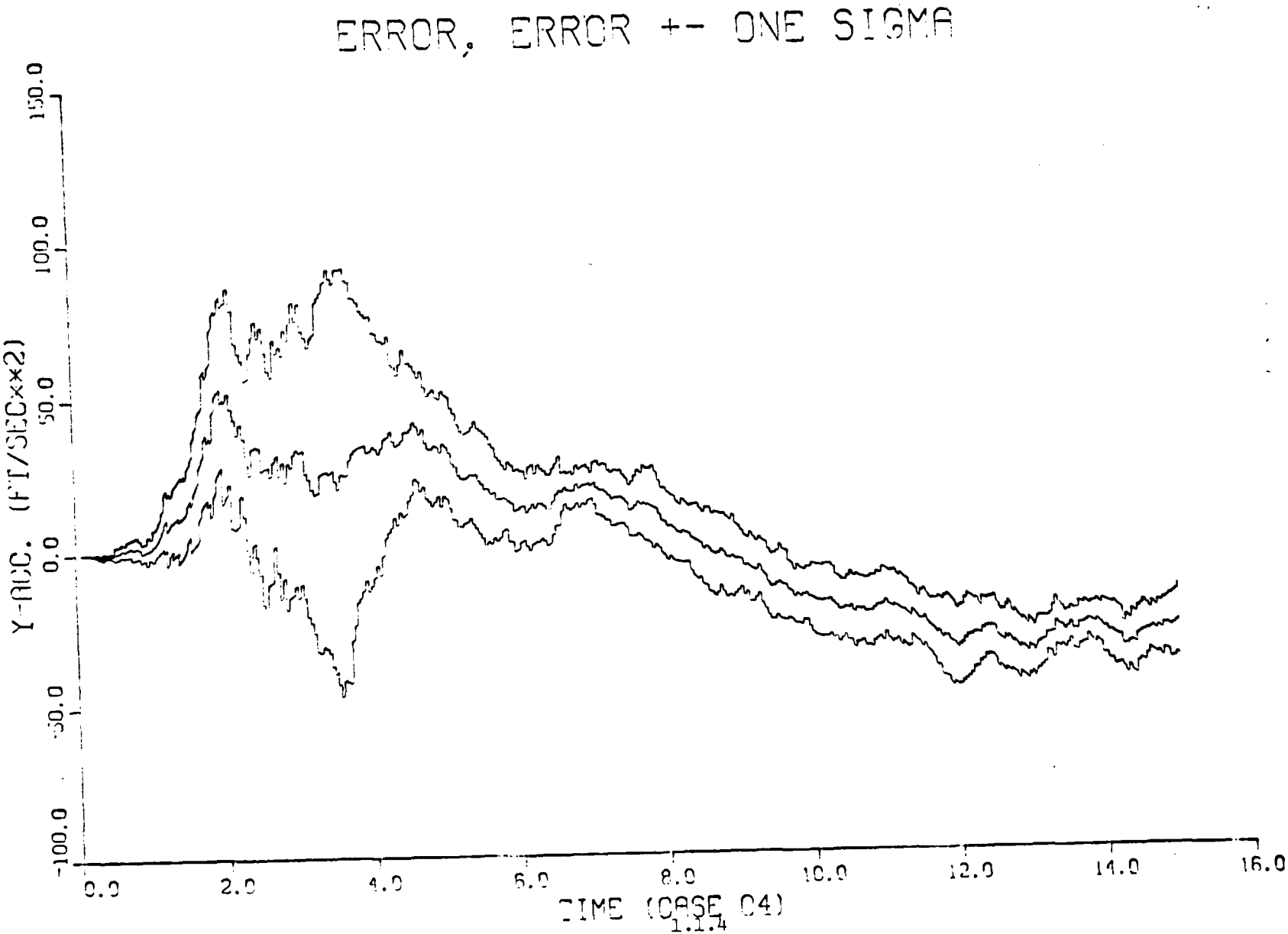

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

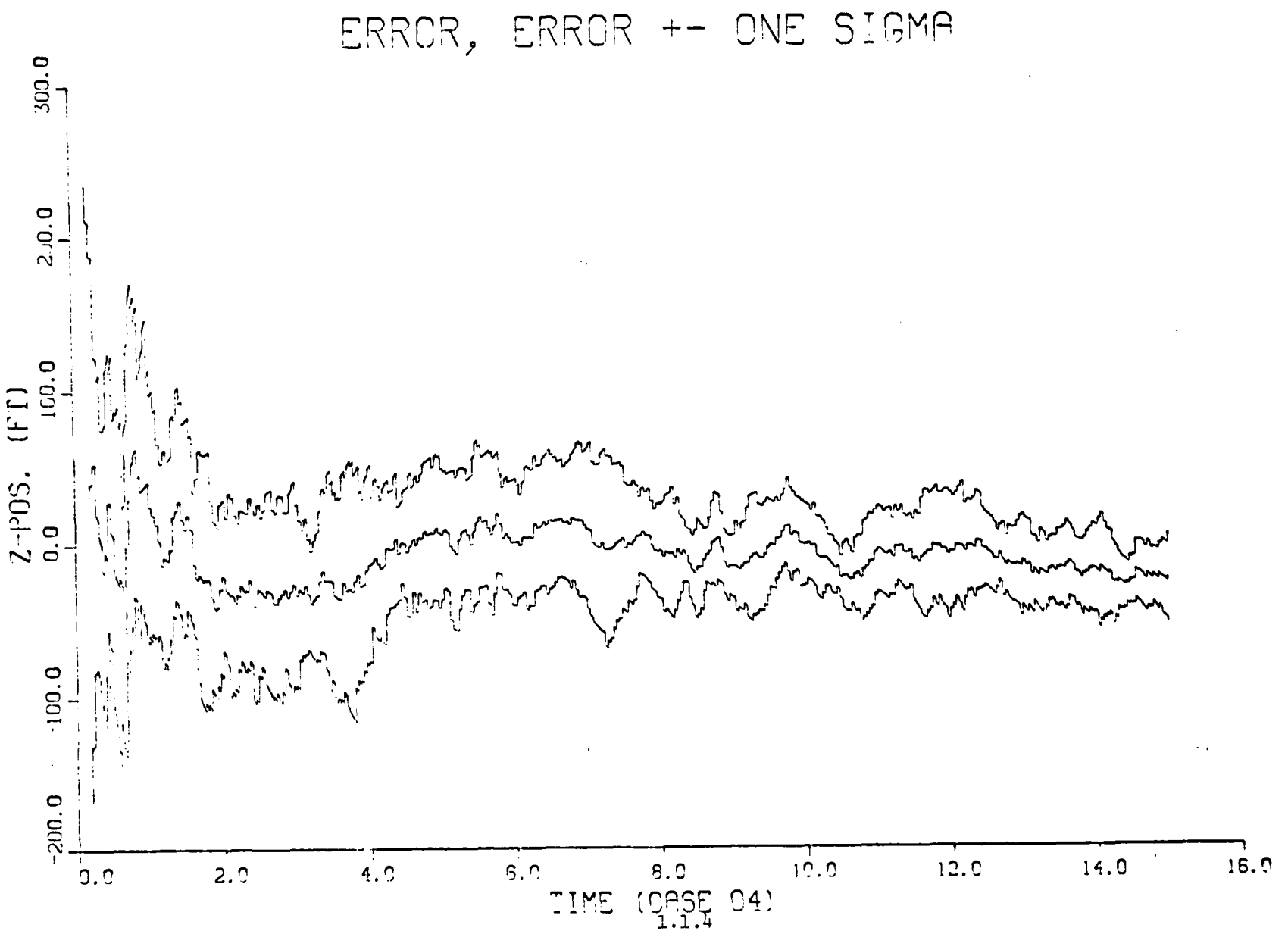
ERROR, ERROR +- ONE SIGMA



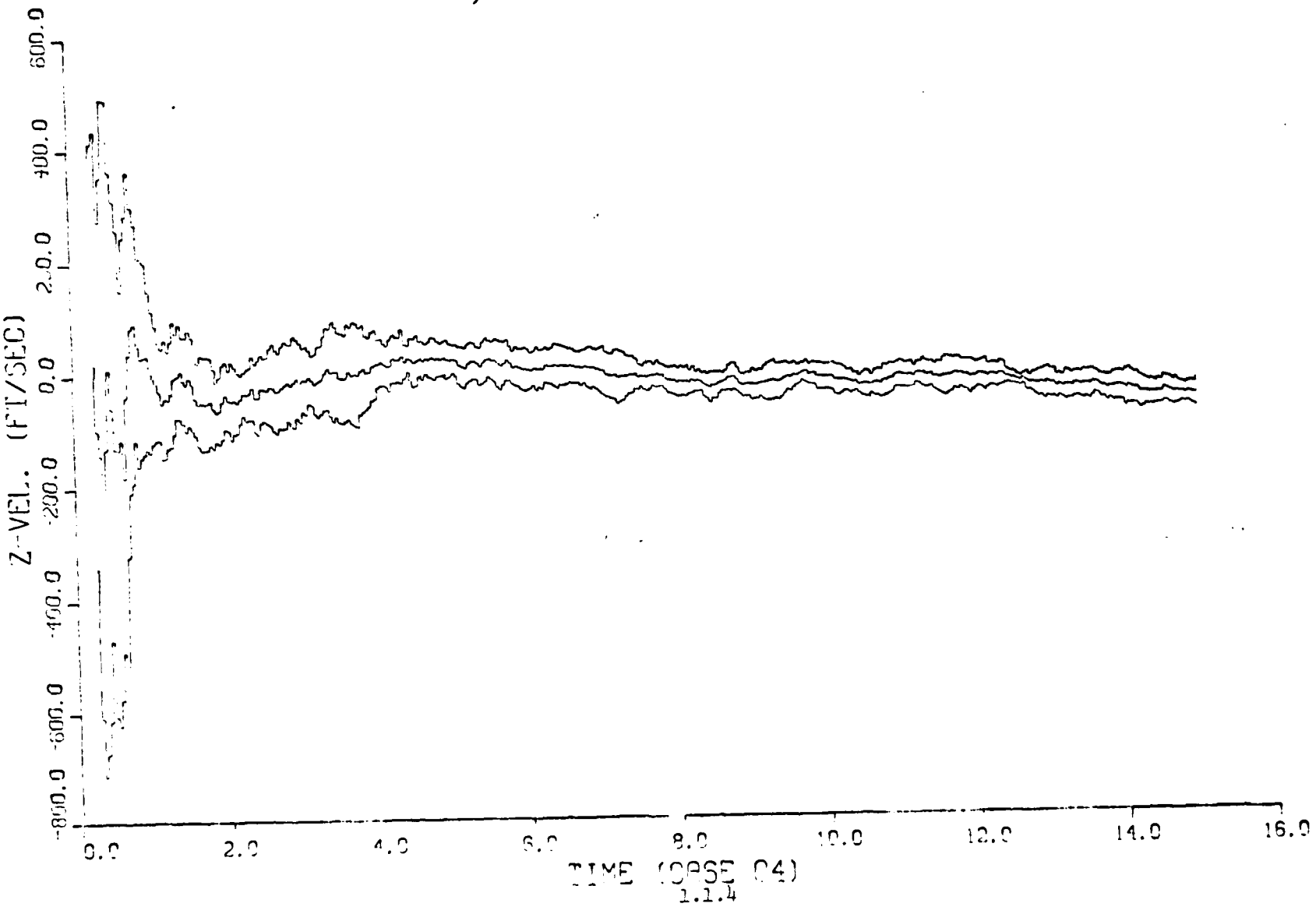
1.1.4

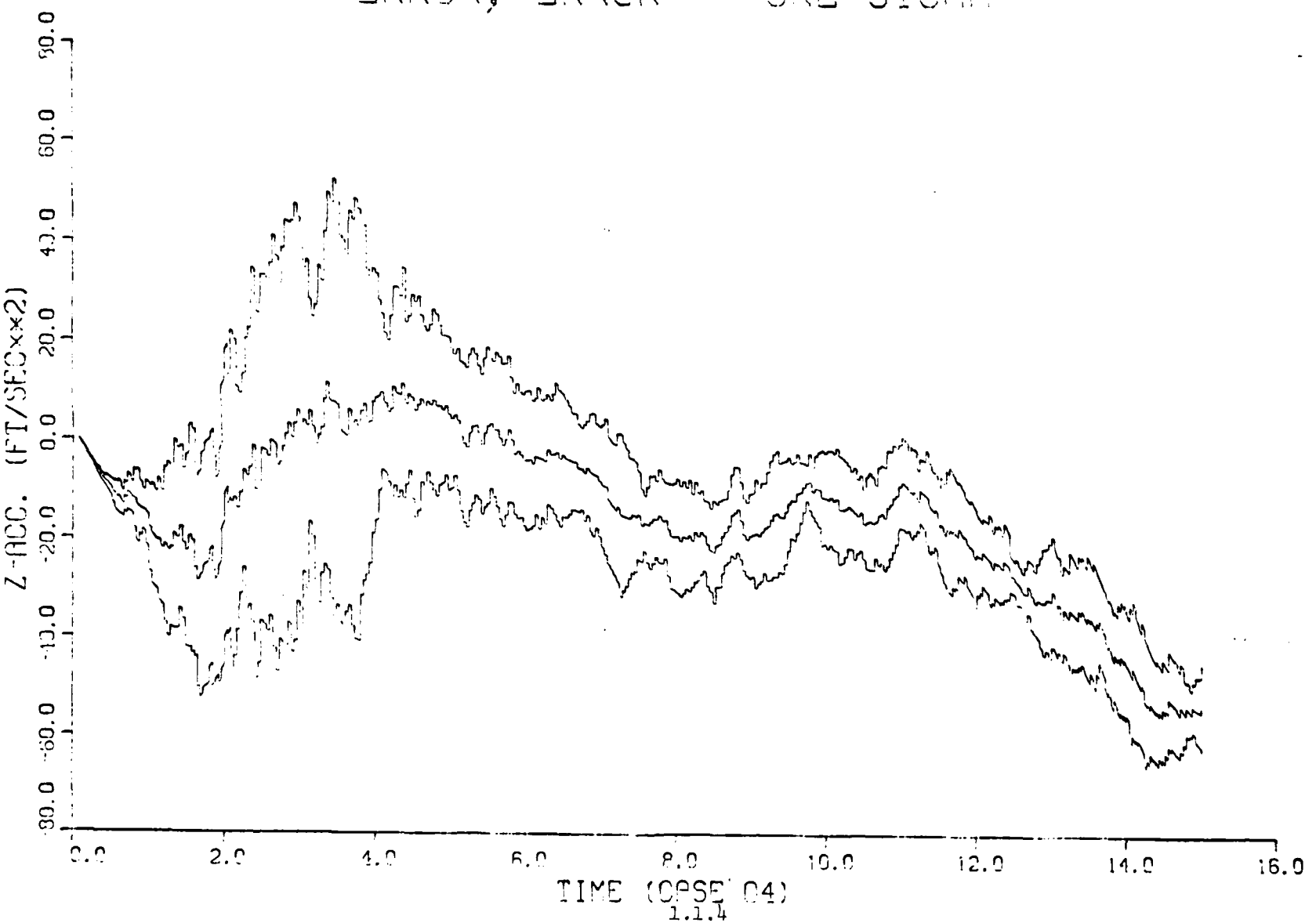

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
-1000.0
-1200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

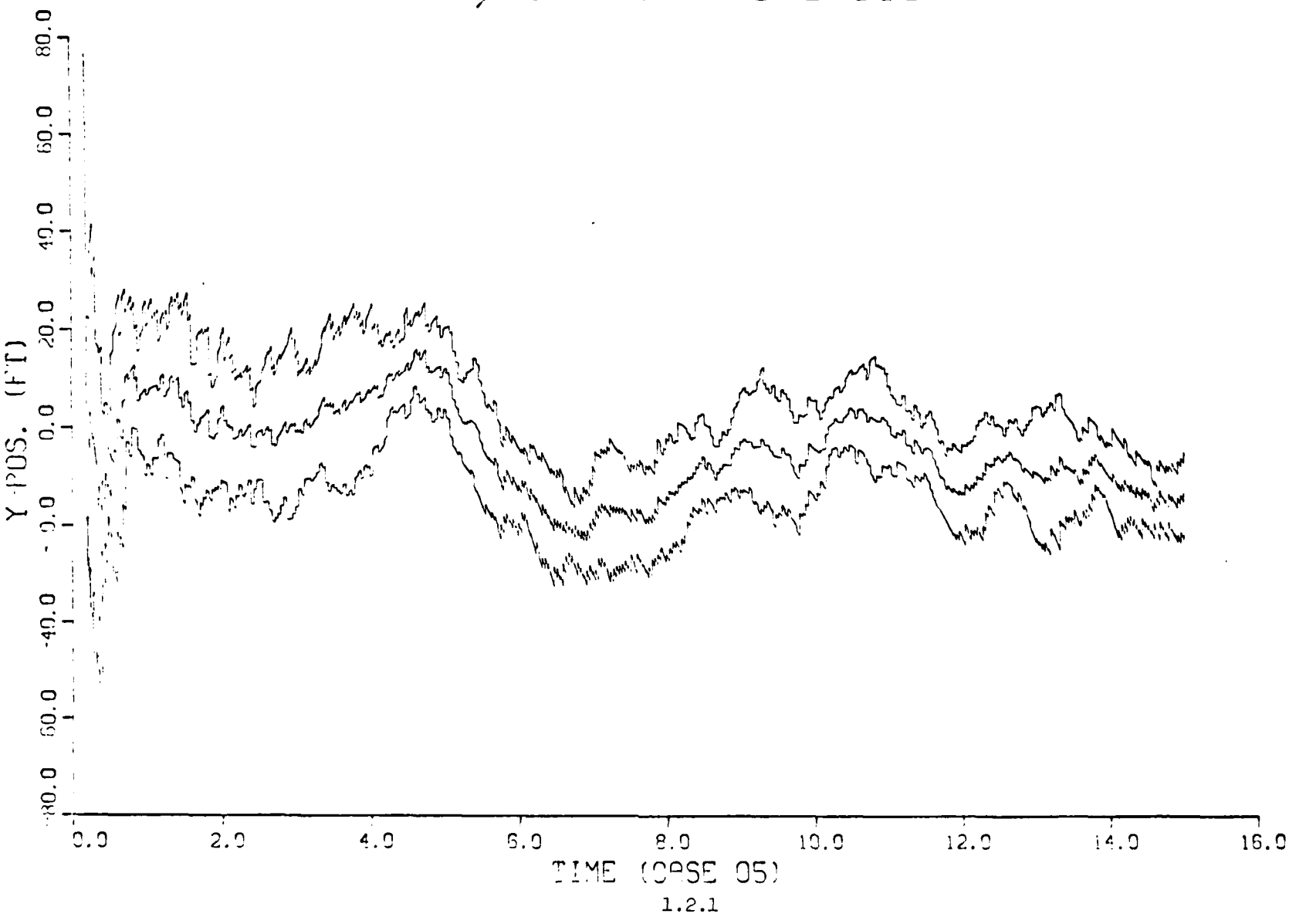

ERROR, ERROR +- ONE SIGMA
Y-REC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)

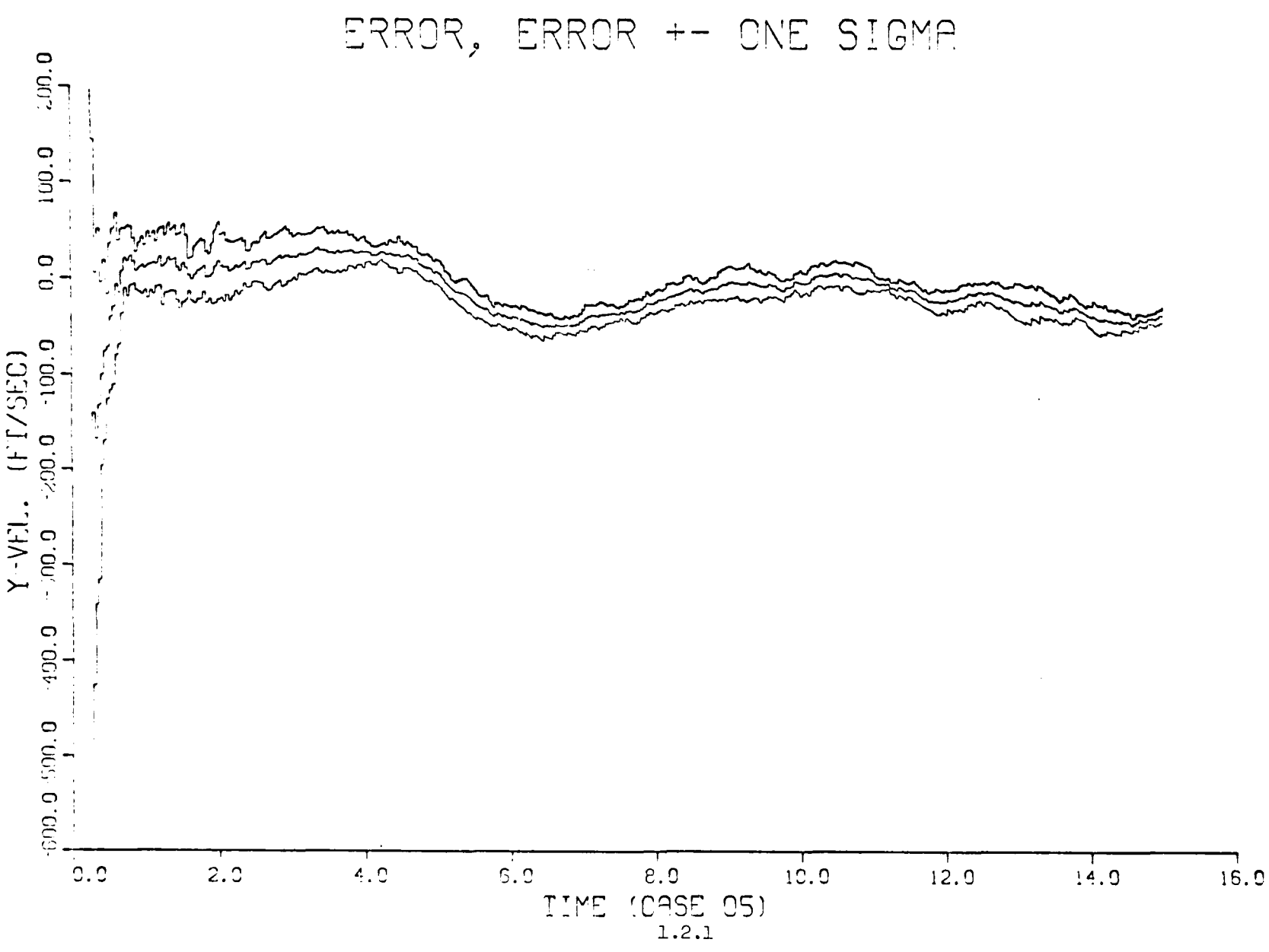

ERROR, ERROR +- ONE SIGMA
Z-POS. (FT)
300.0
200.0
100.0
0.0
-100.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

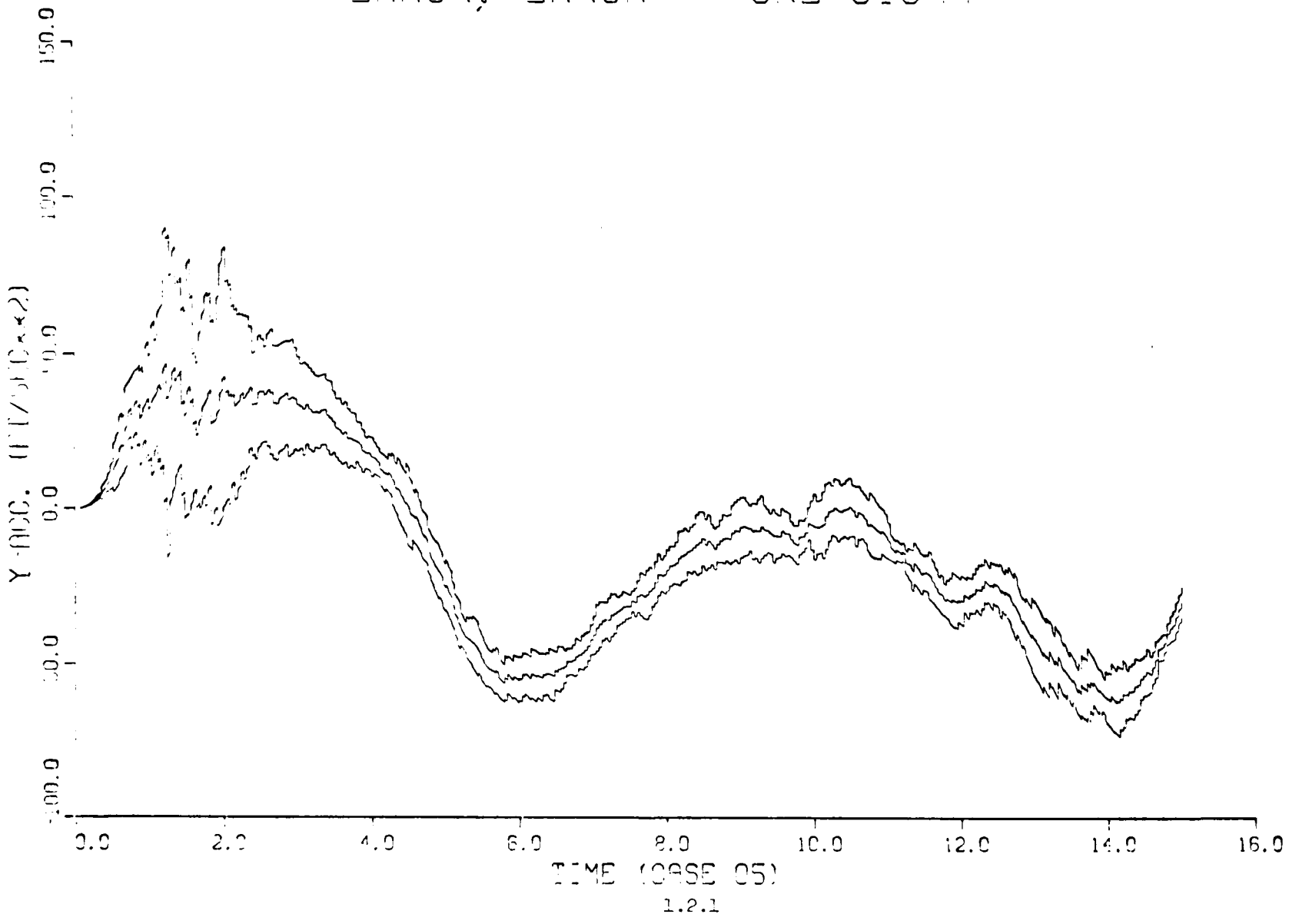
ERROR, ERROR +- ONE SIGMA

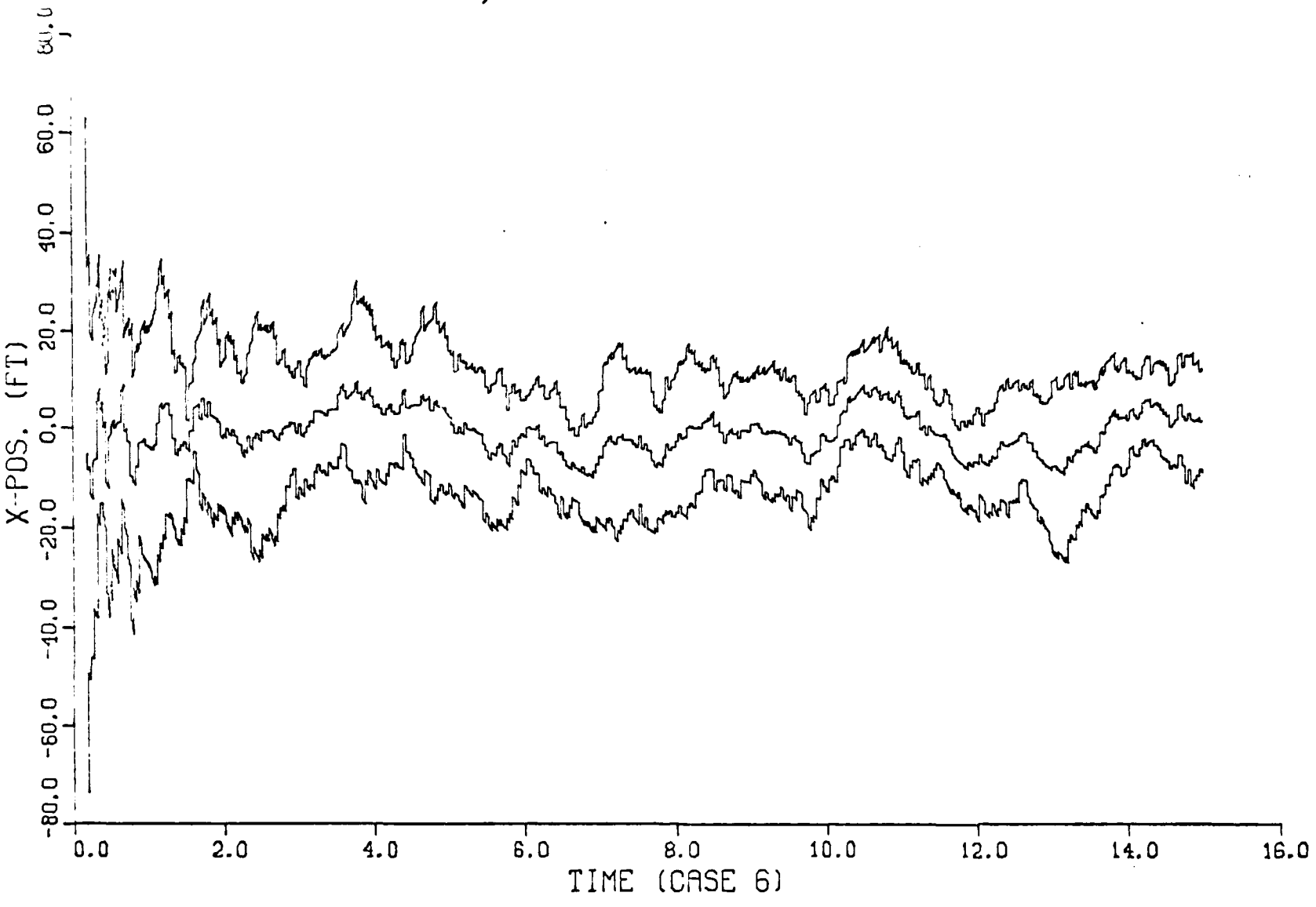



ERROR, ERROR +- ONE SIGMA
Z-ACC. (FT/SEC**2)
90.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 04)
1.1.4

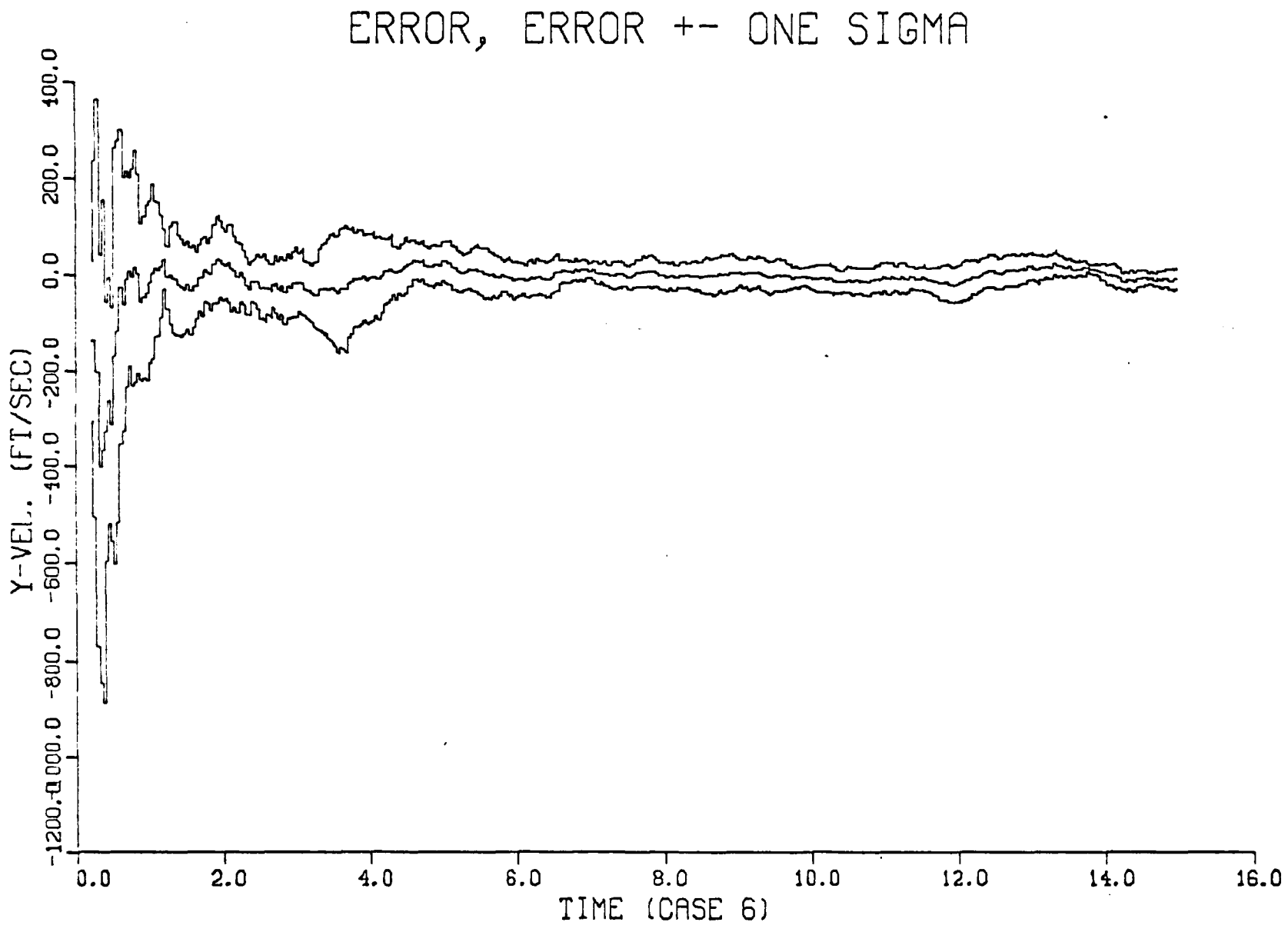

ERROR, ERROR +- ONE SIGMA
Y POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 05)
1.2.1


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
500.0
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 05)
1.2.1

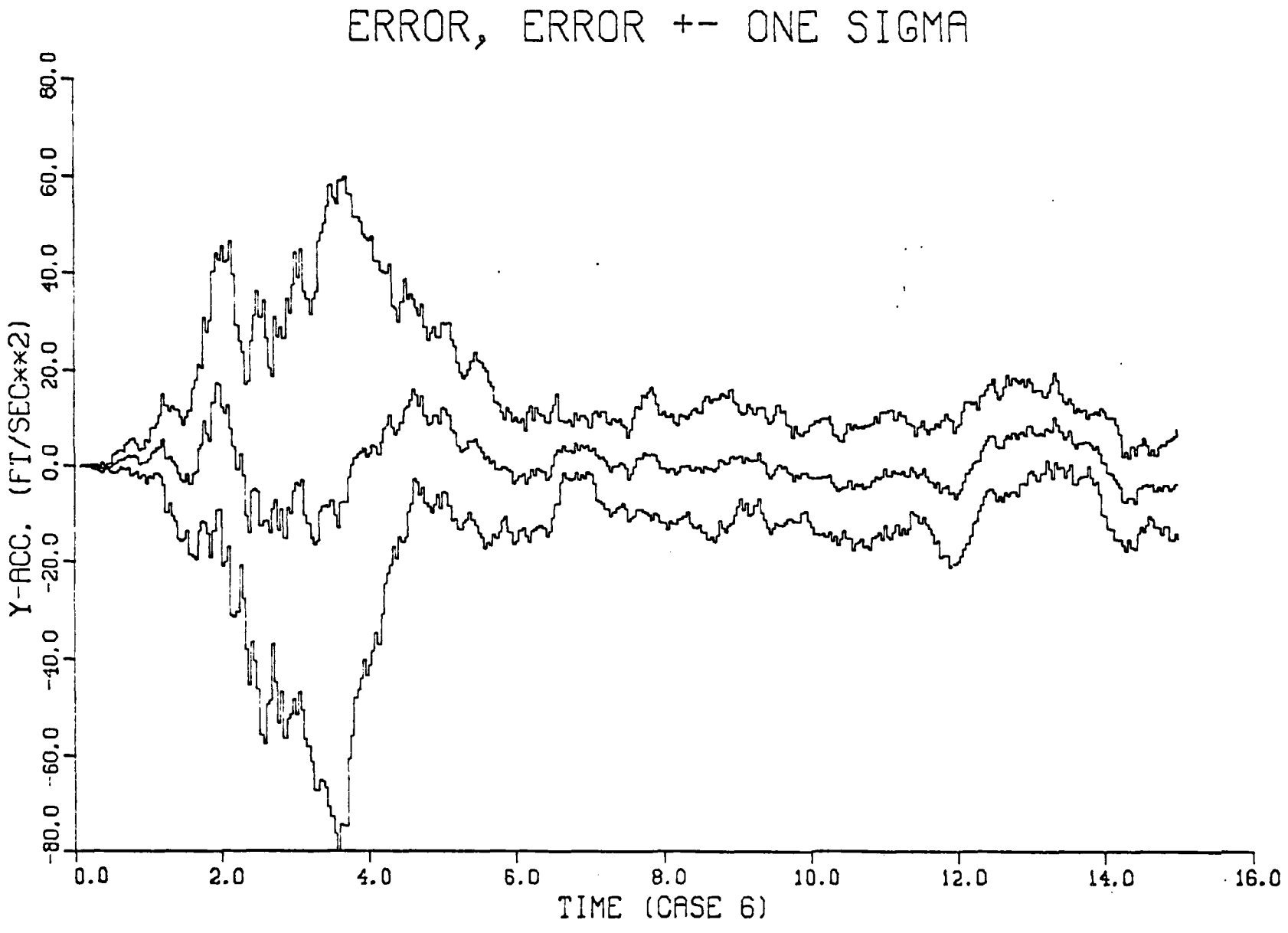

ERROR, ERROR +- ONE SIGMA
150.0
100.0
50.0
0.0
-50.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (BASE 05)
1.2.1


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 6)

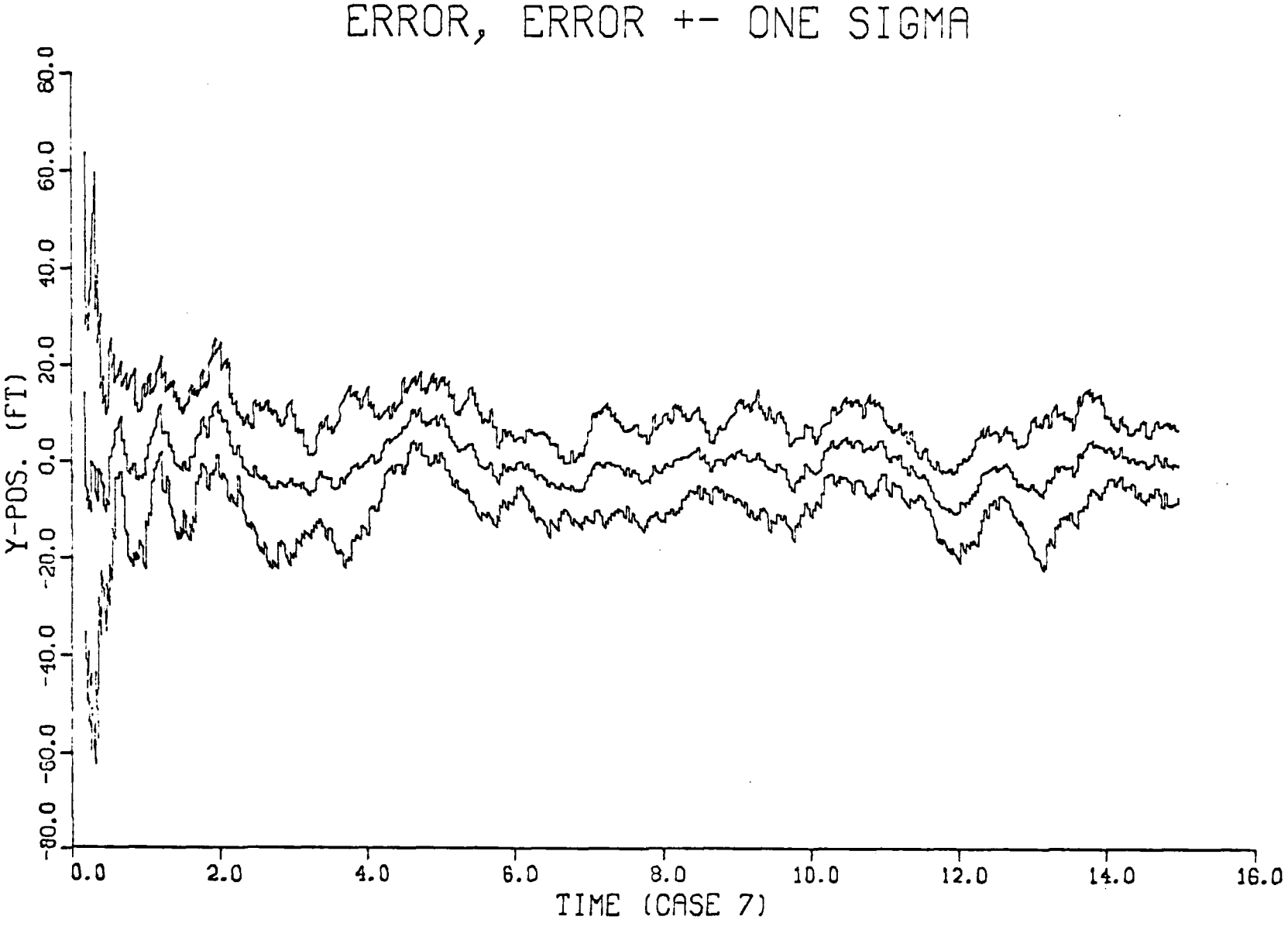
1.2.2

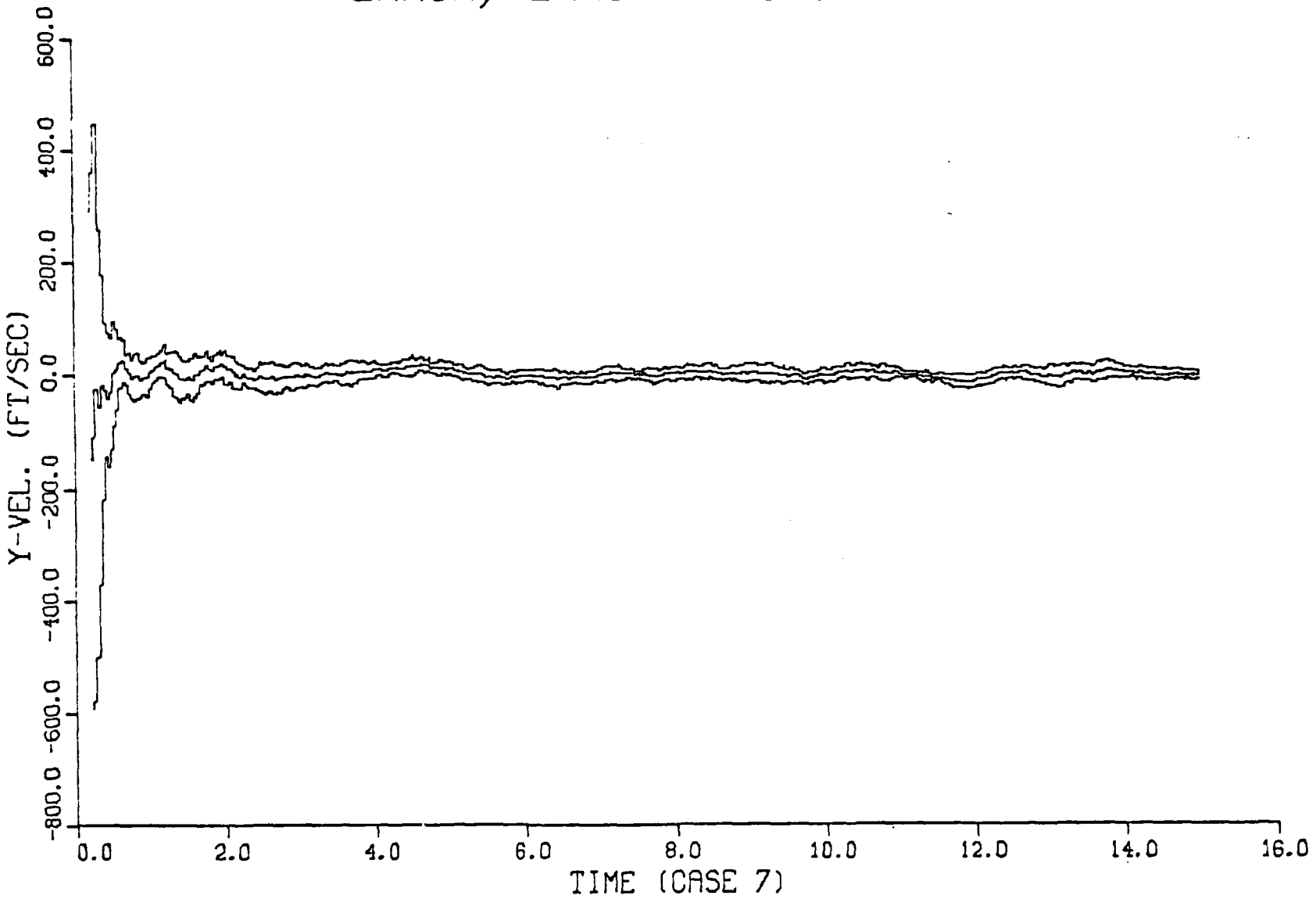

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
100.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
-1000.0
-1200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 6)

1.2.2

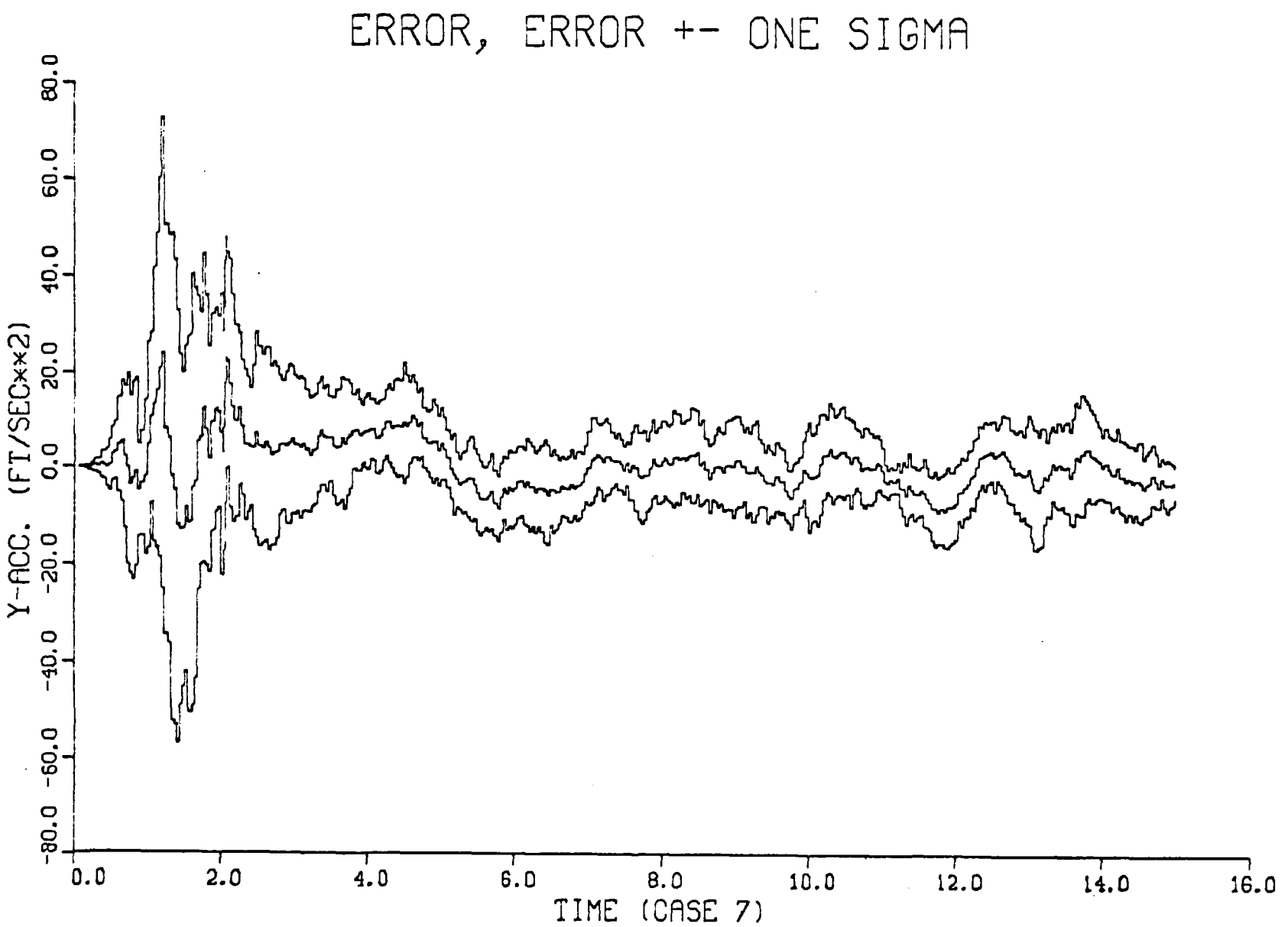

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 6)

1.2.2

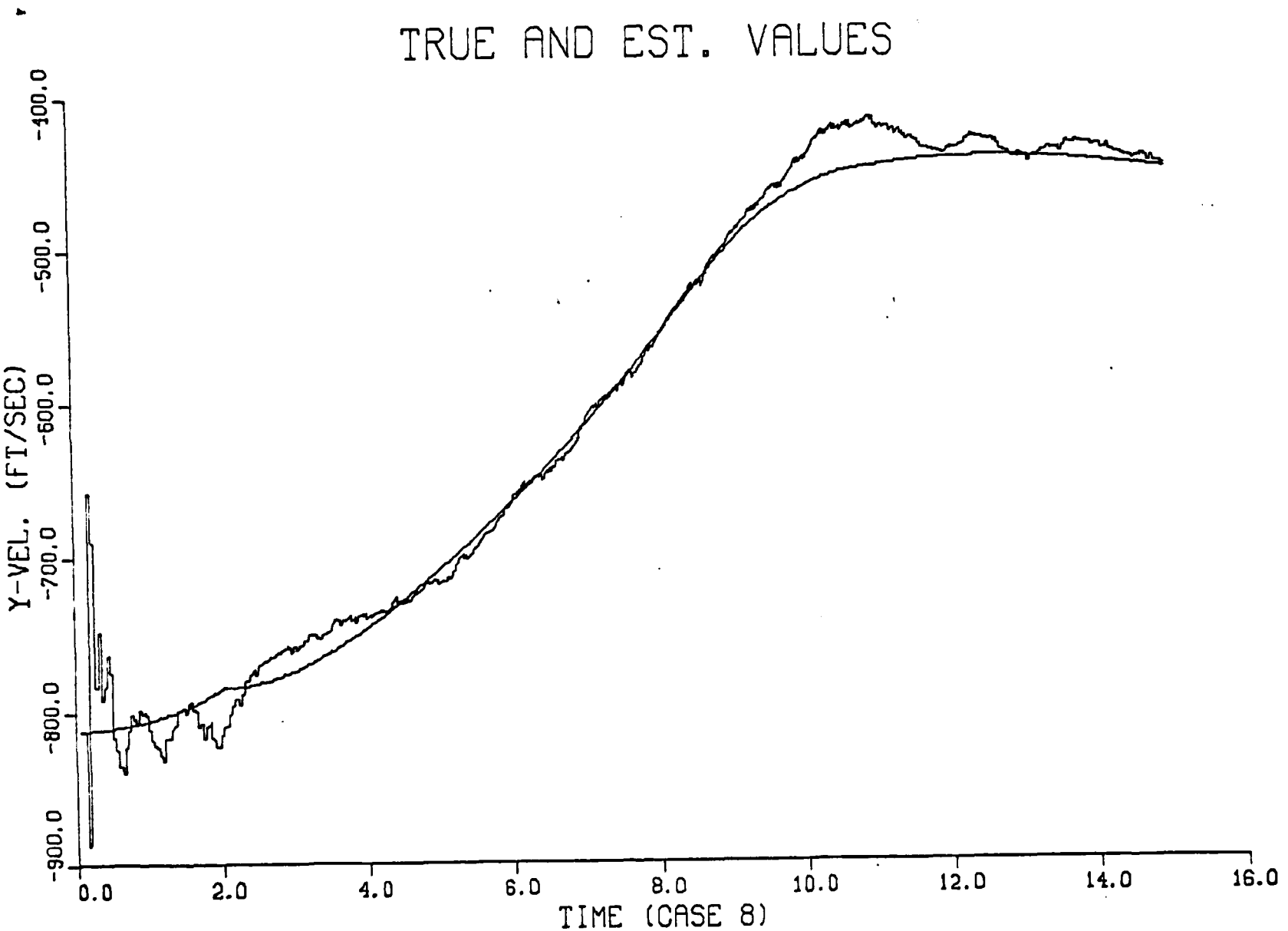

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 7)


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 7)

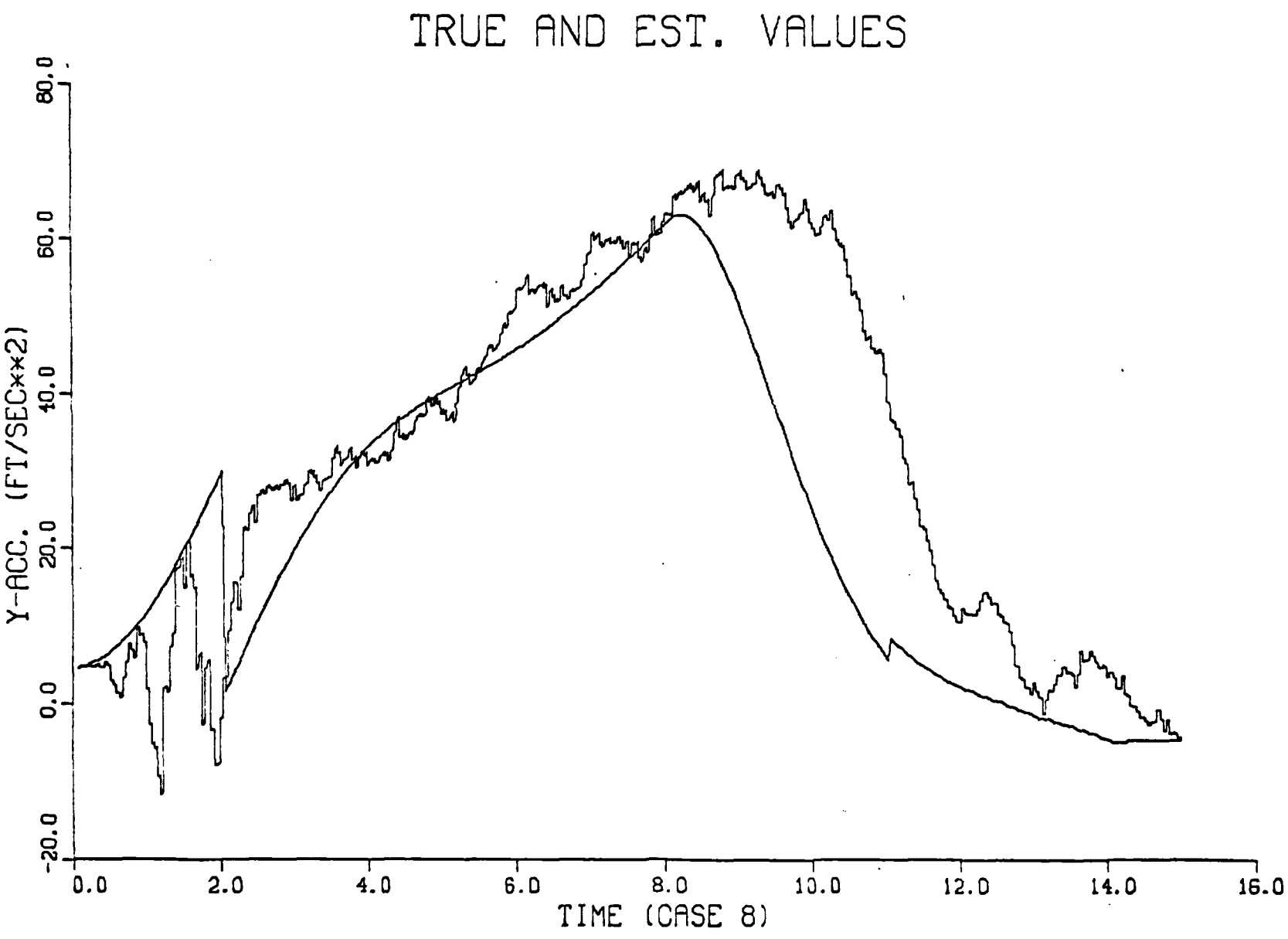
1.2.3


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 7)

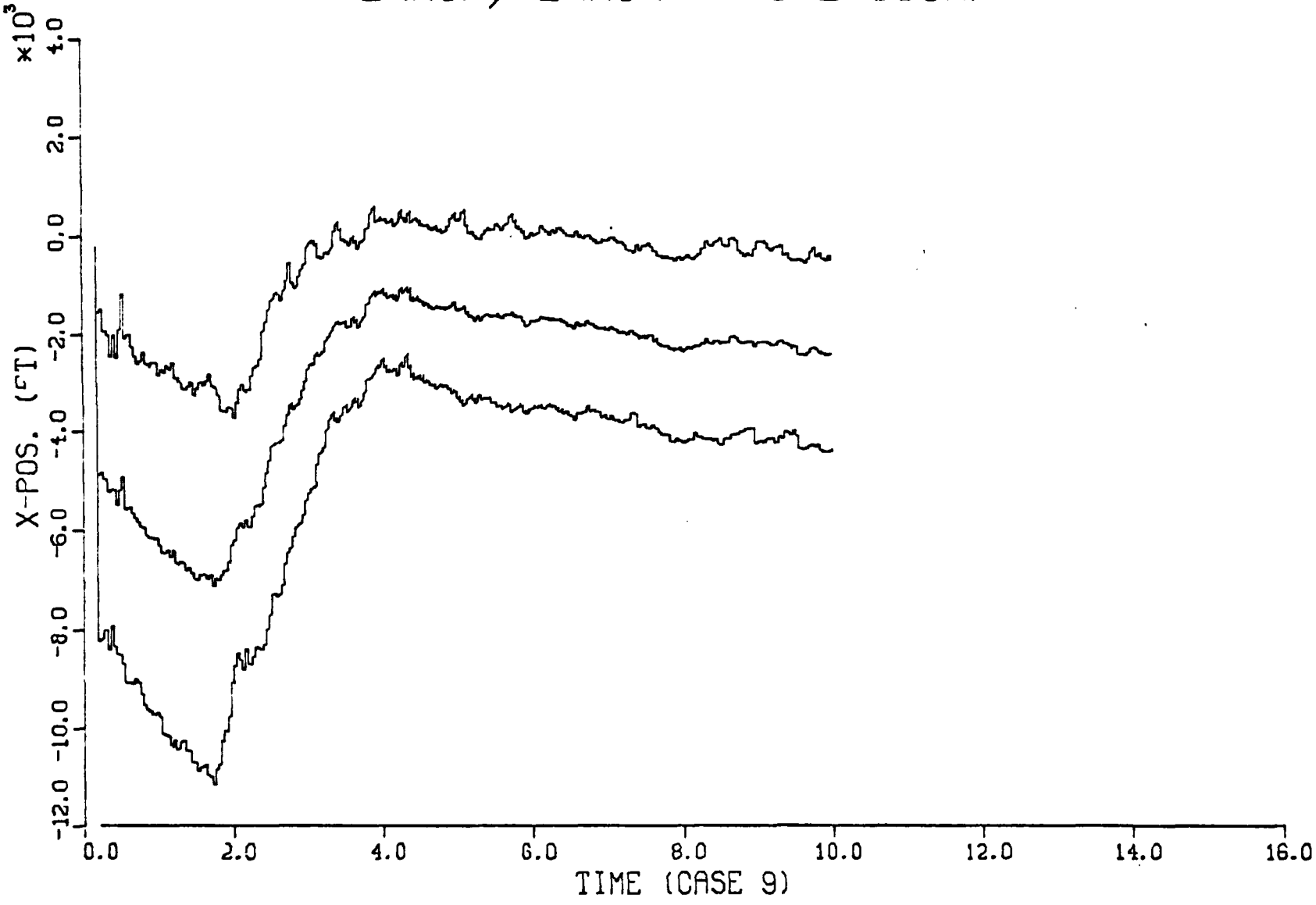
1.2.3


TRUE AND EST. VALUES
Y-VEL. (FT/SEC)
-400.0
-500.0
-600.0
-700.0
-800.0
-900.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 8)

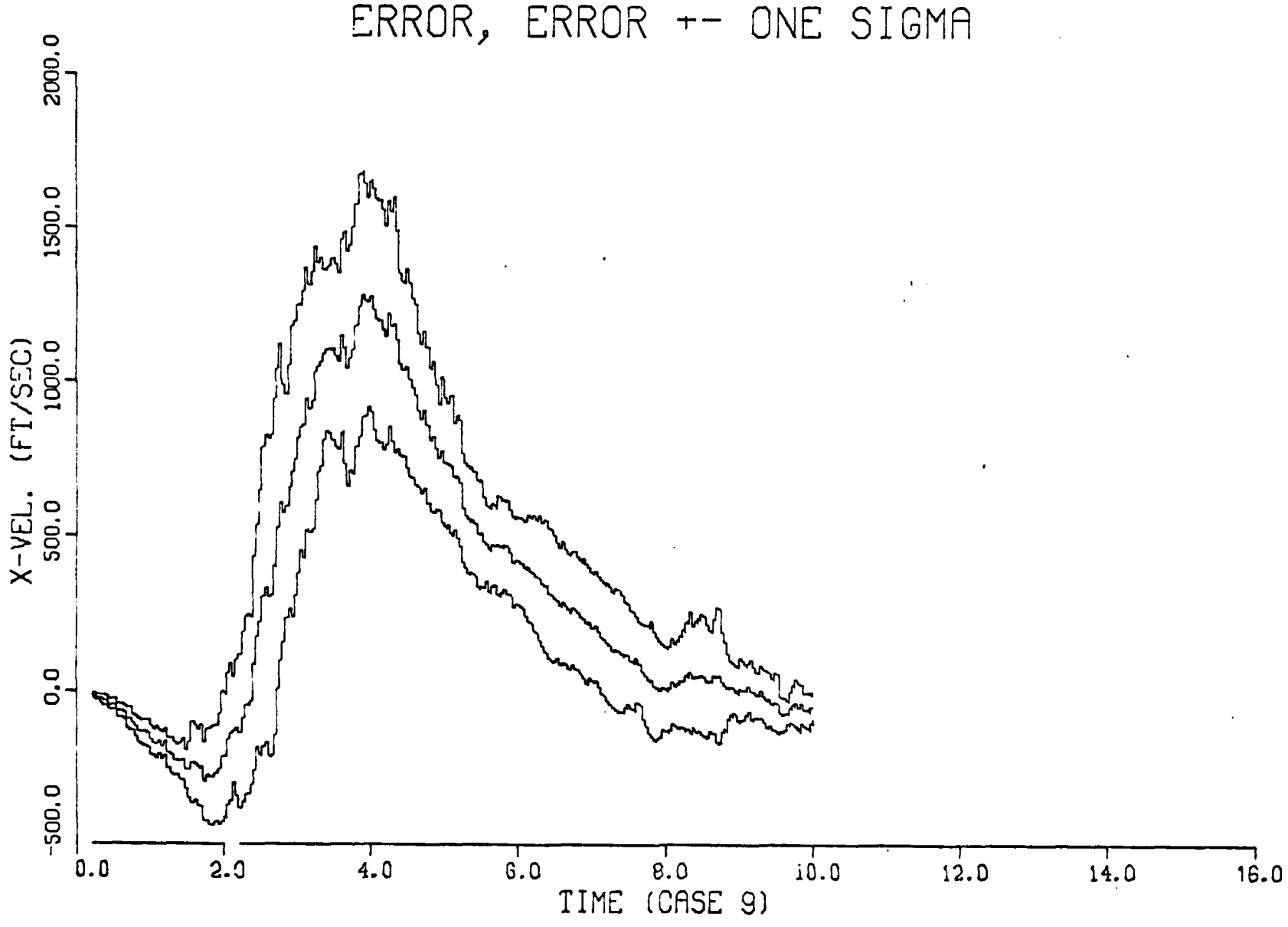
1.2.4


TRUE AND EST. VALUES
Y-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 8)

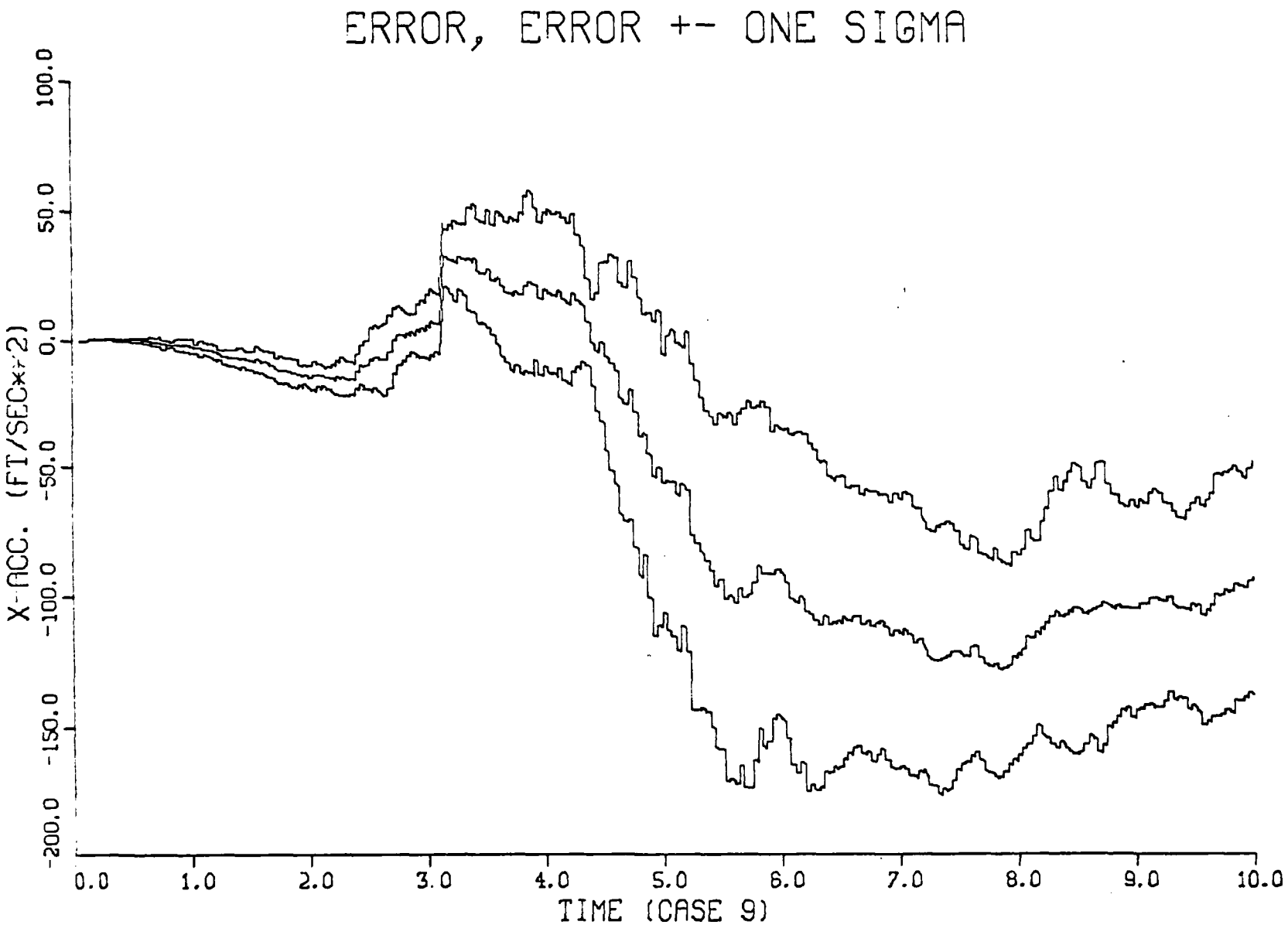
1.2.4

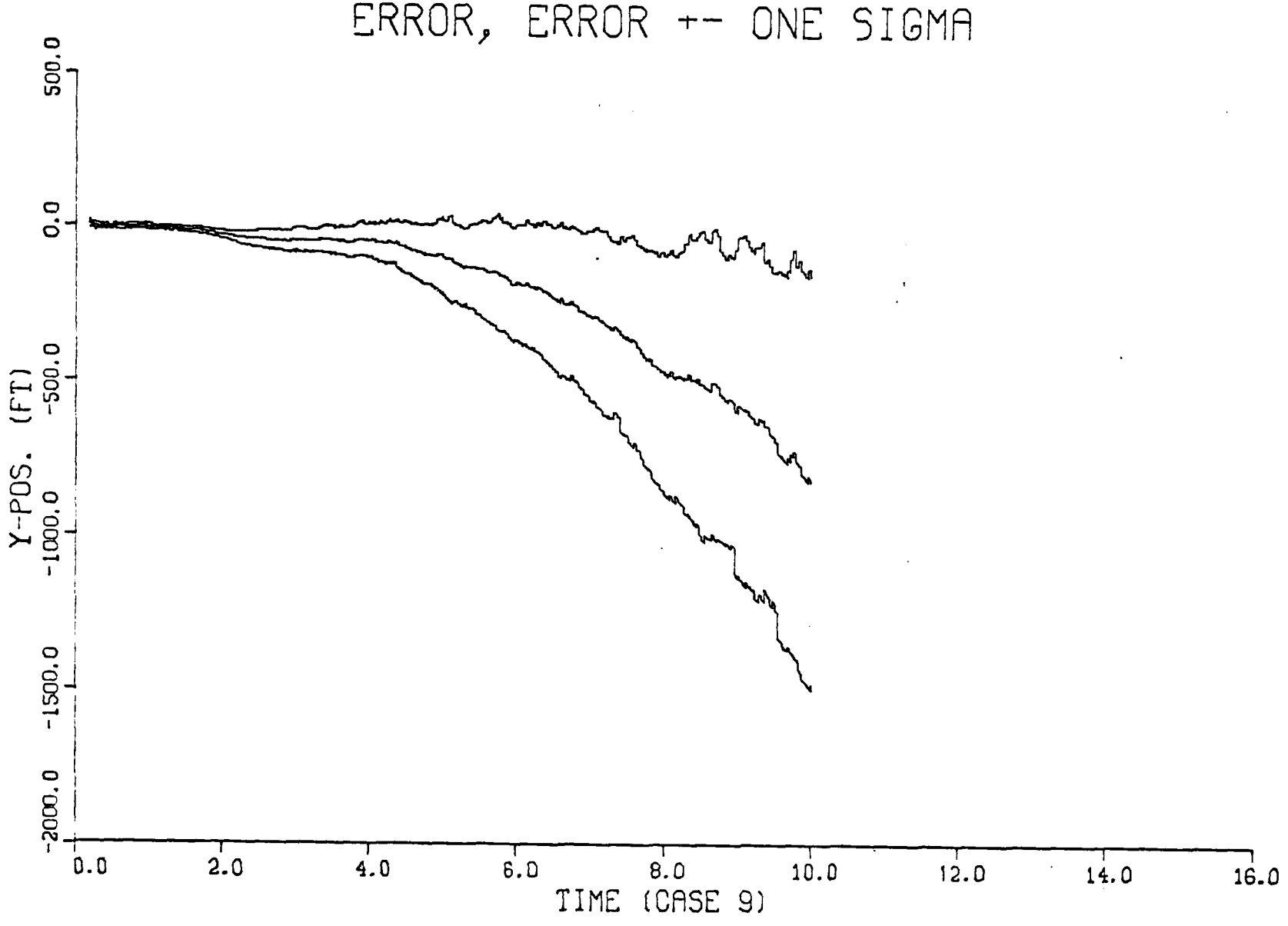

ERROR, ERROR +- ONE SIGMA
X-POS. (CT)
×10³
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
-12.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 9)

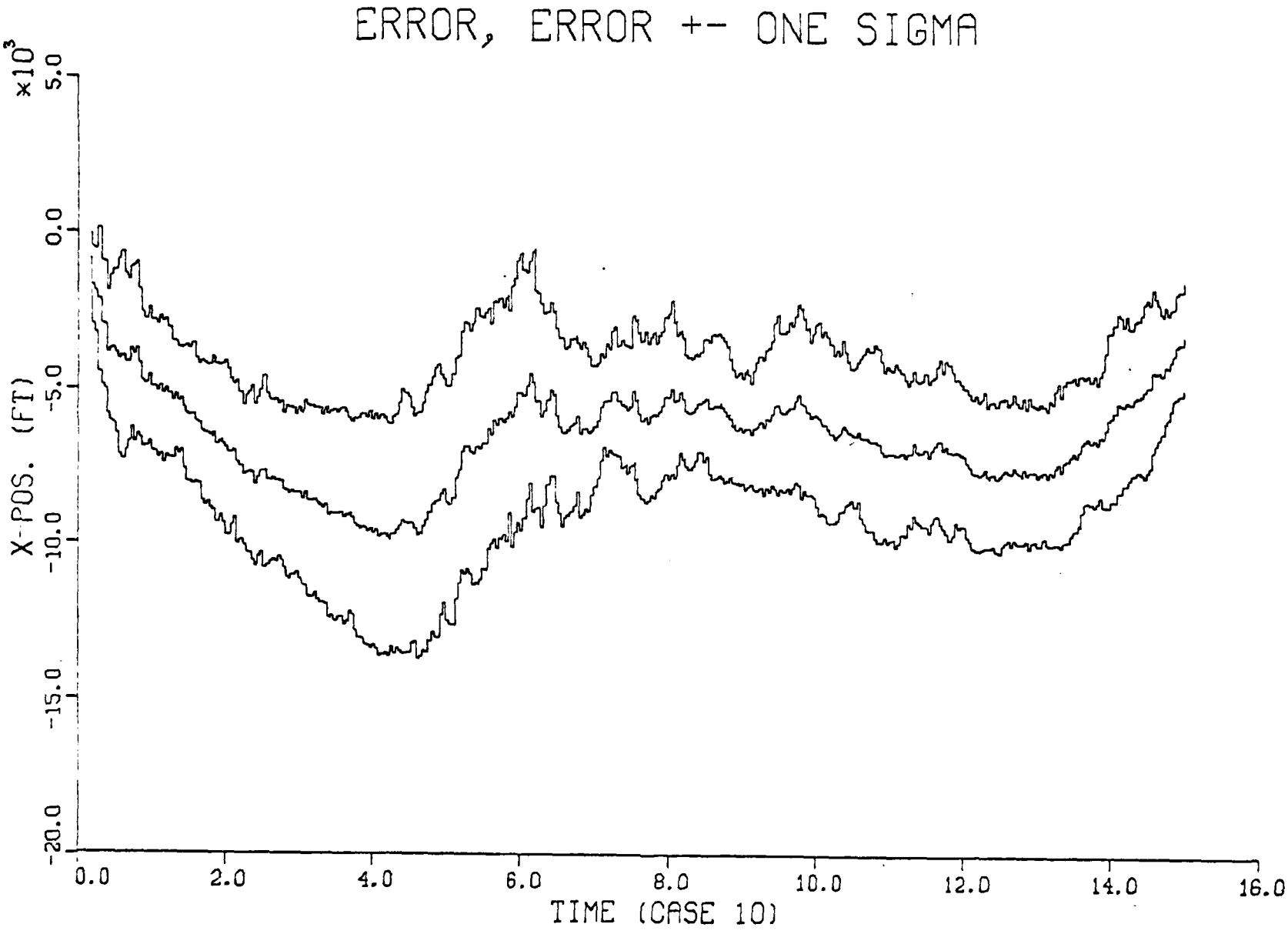
1.3.1


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
2000.0
1500.0
1000.0
500.0
0.0
-500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 9)

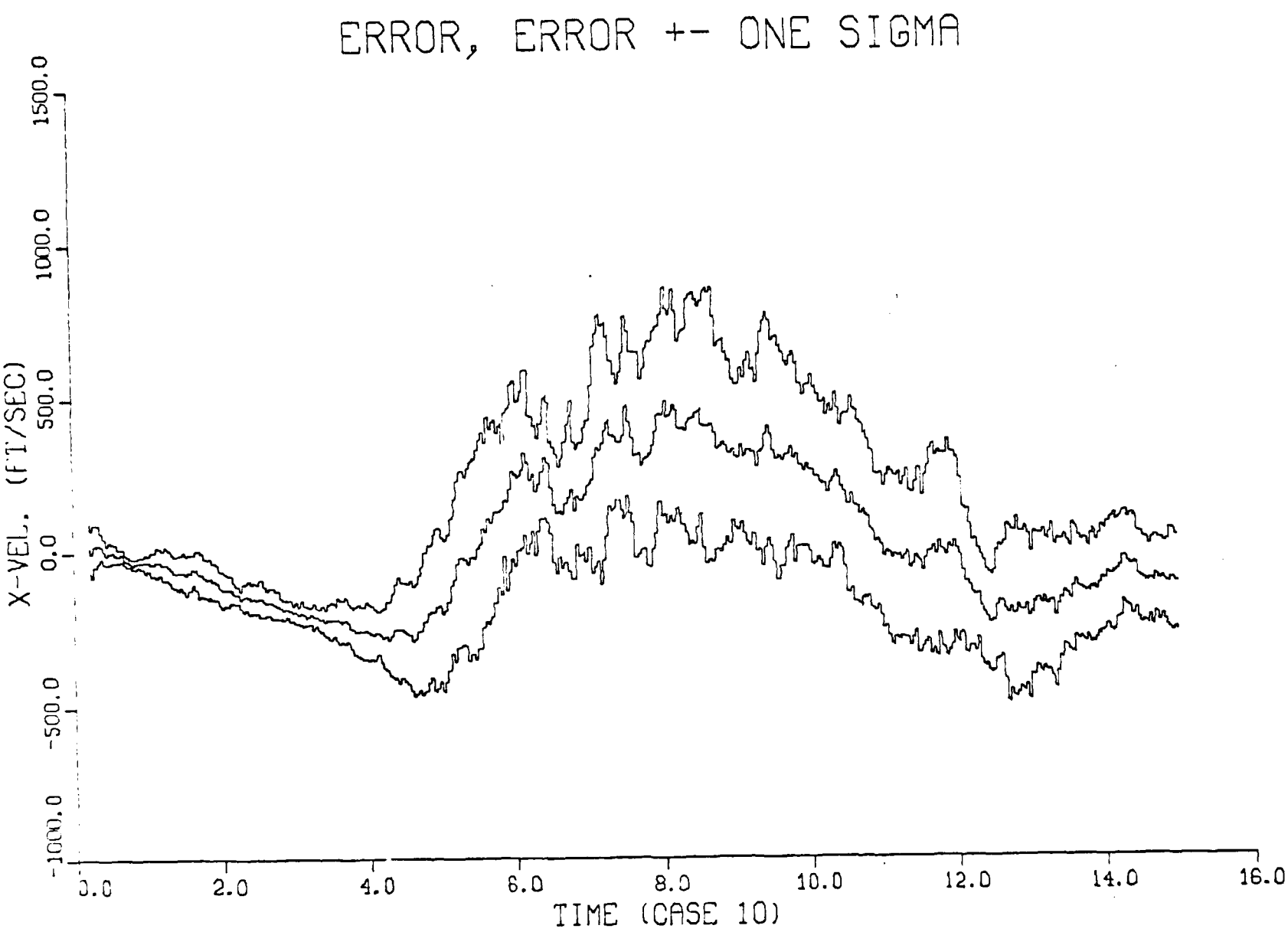
1.3.1


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
TIME (CASE 9)


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
500.0
0.0
-500.0
-1000.0
-1500.0
-2000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 9)

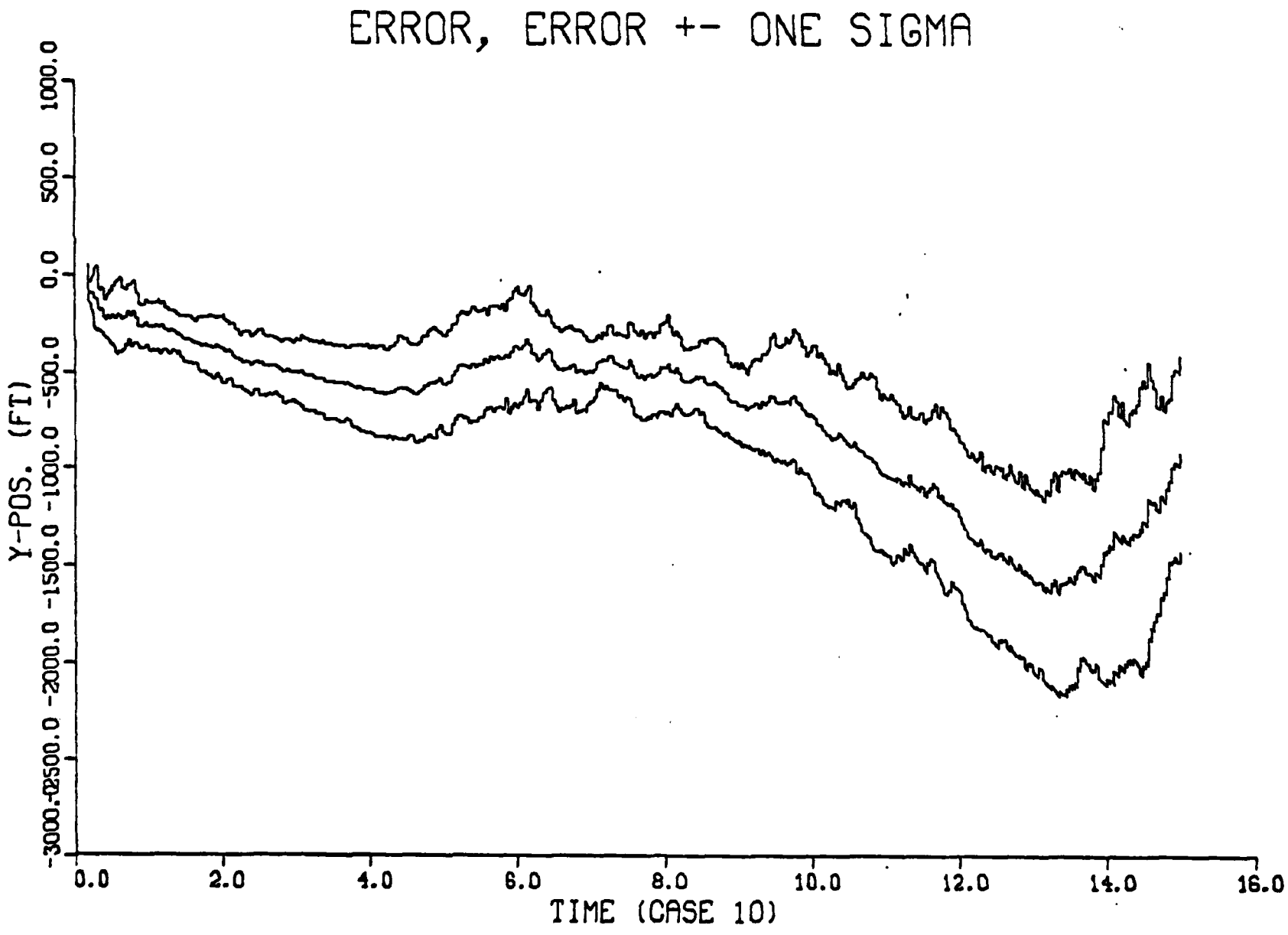

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
×10³
5.0
0.0
-5.0
-10.0
-15.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 10)

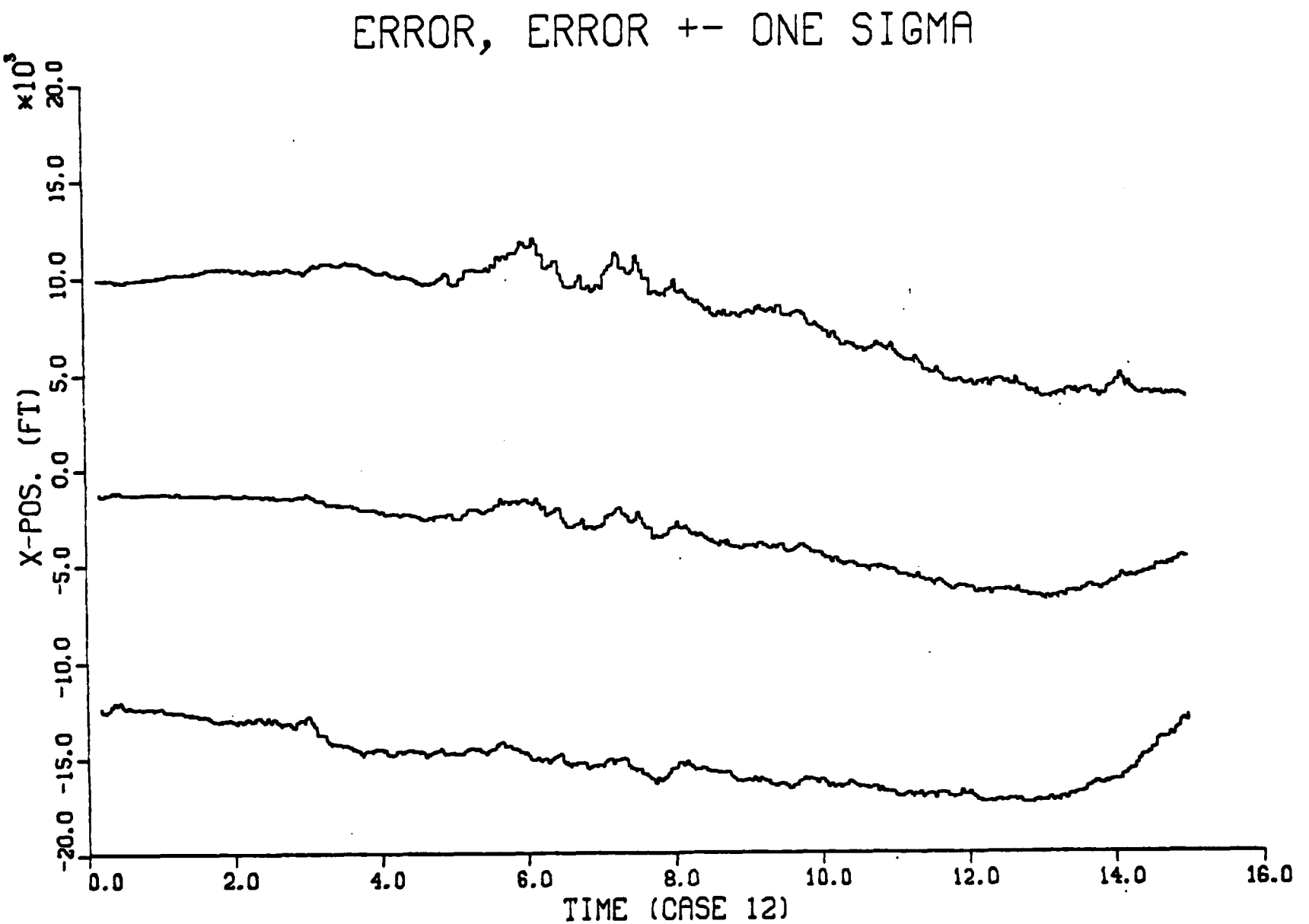
1.3.2


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
1500.0
1000.0
500.0
0.0
-500.0
-1000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 10)

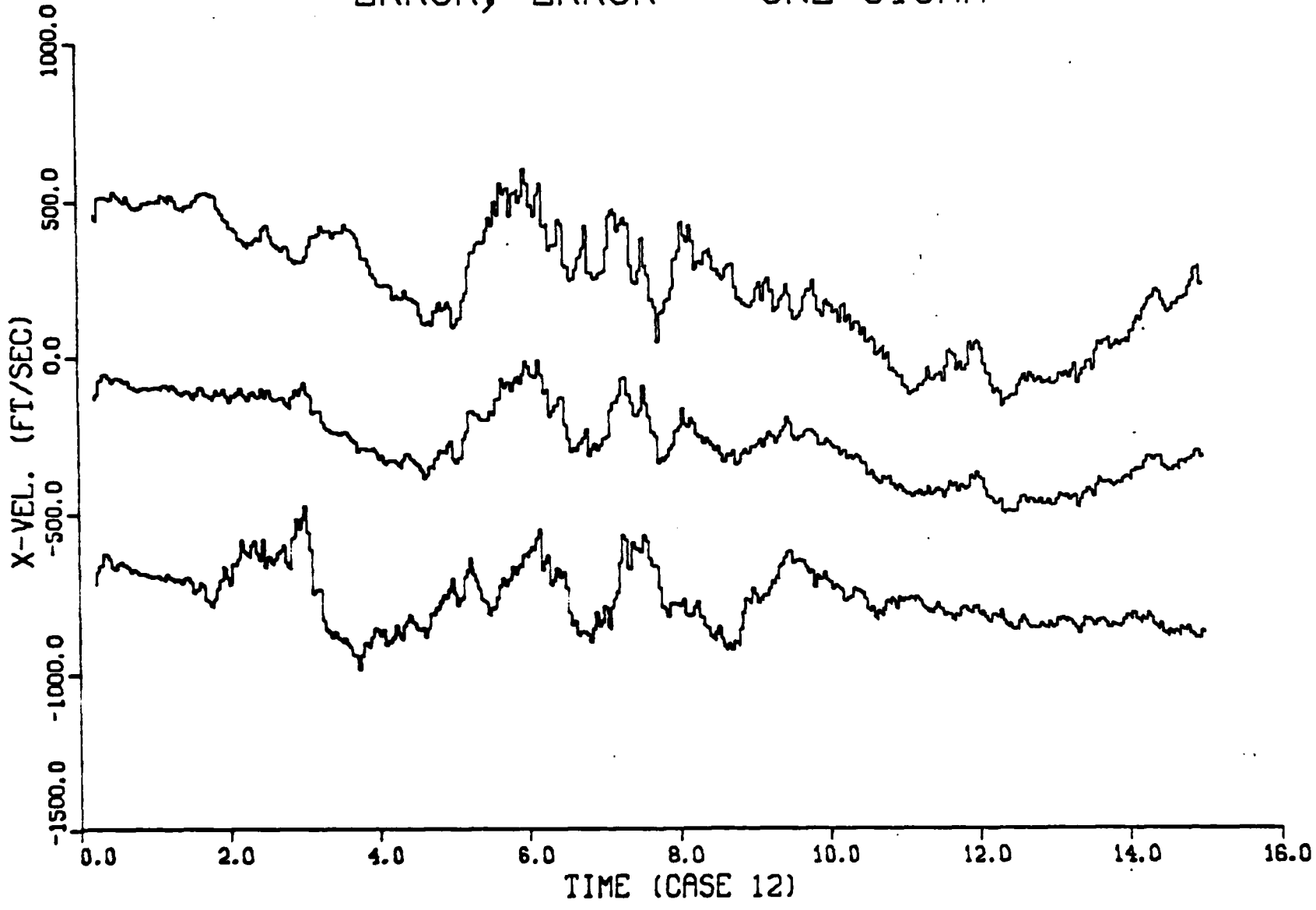
1.3.2

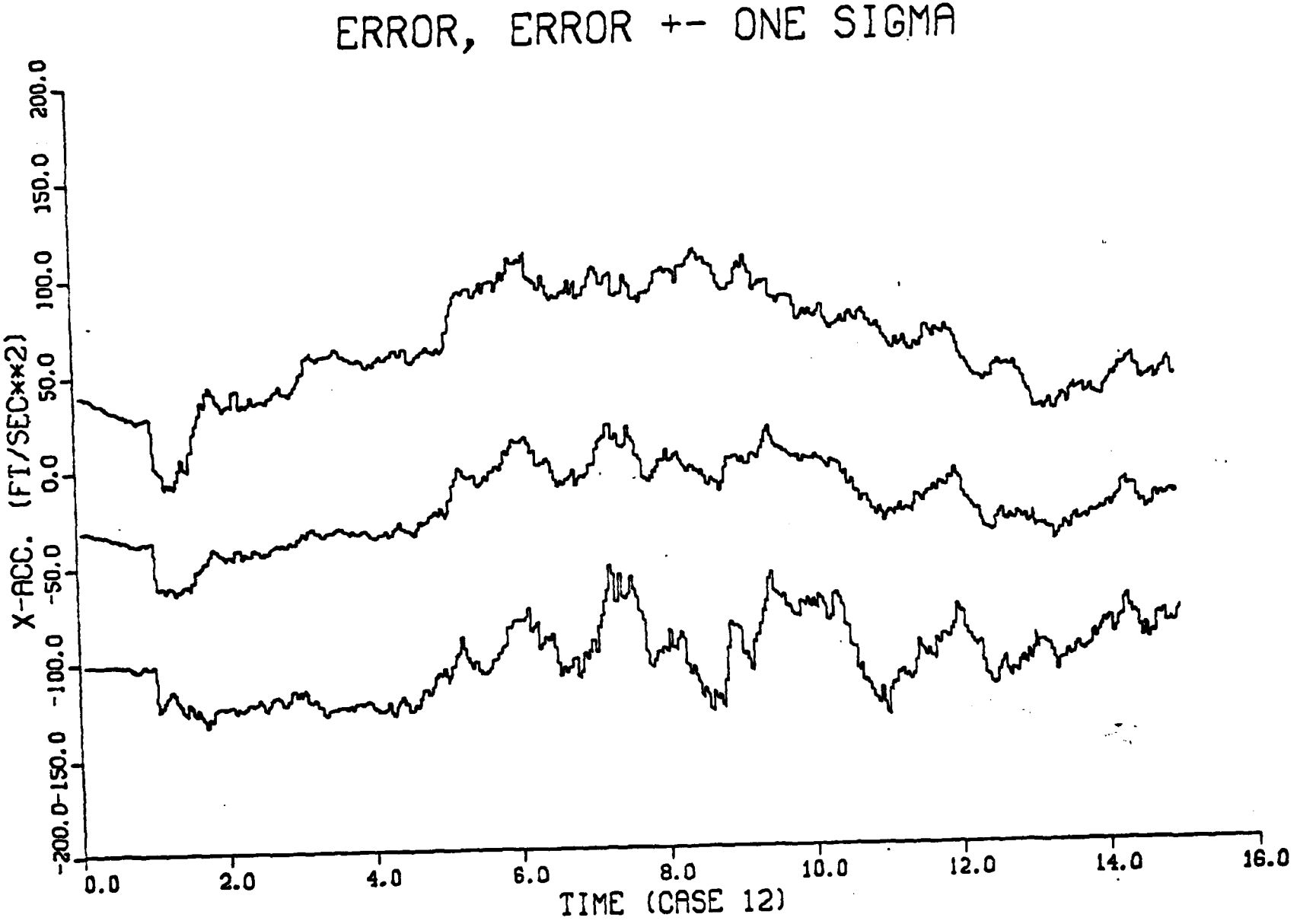
ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
0

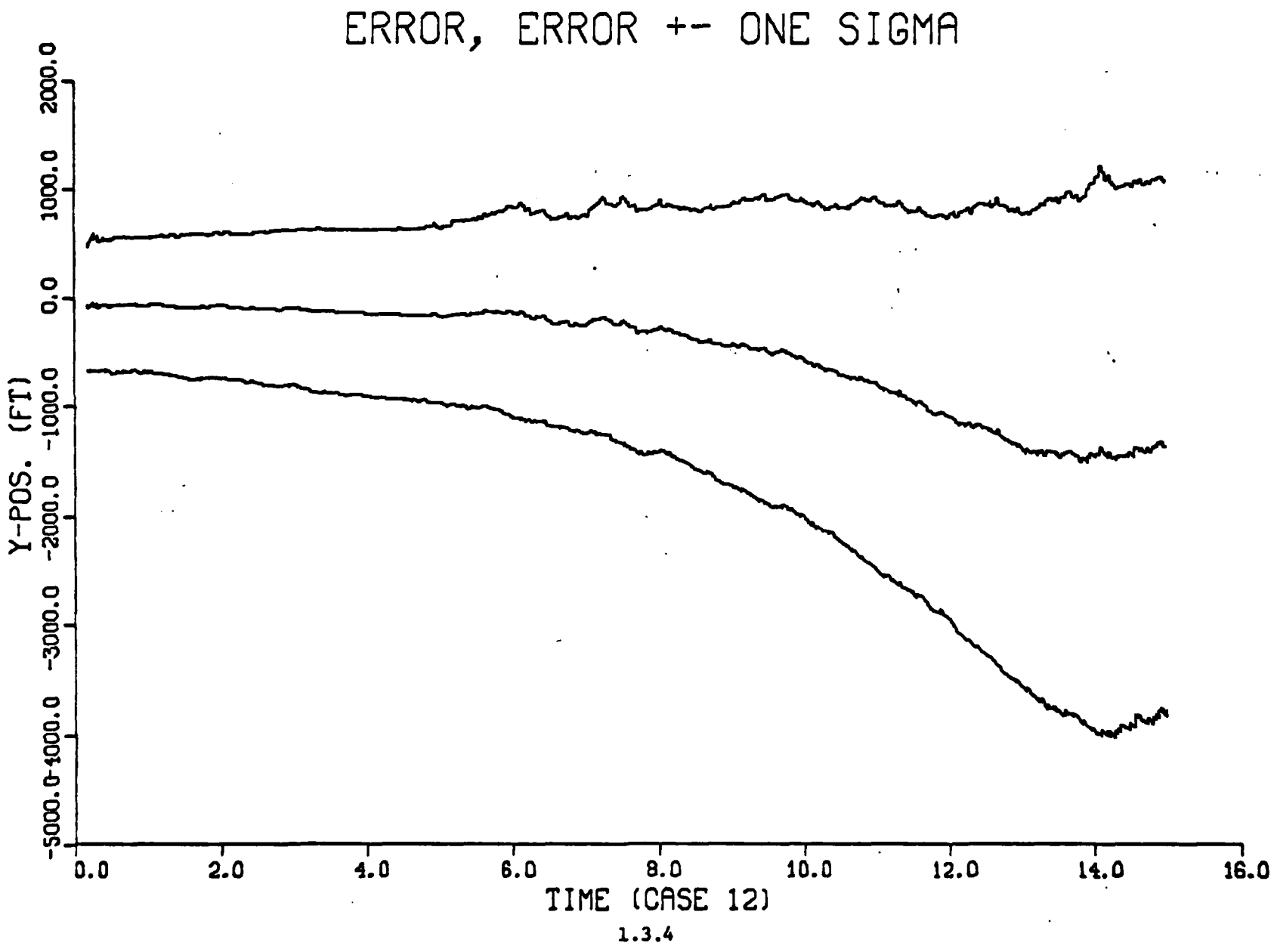

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
-2000.0
-2500.0
-3000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 10)

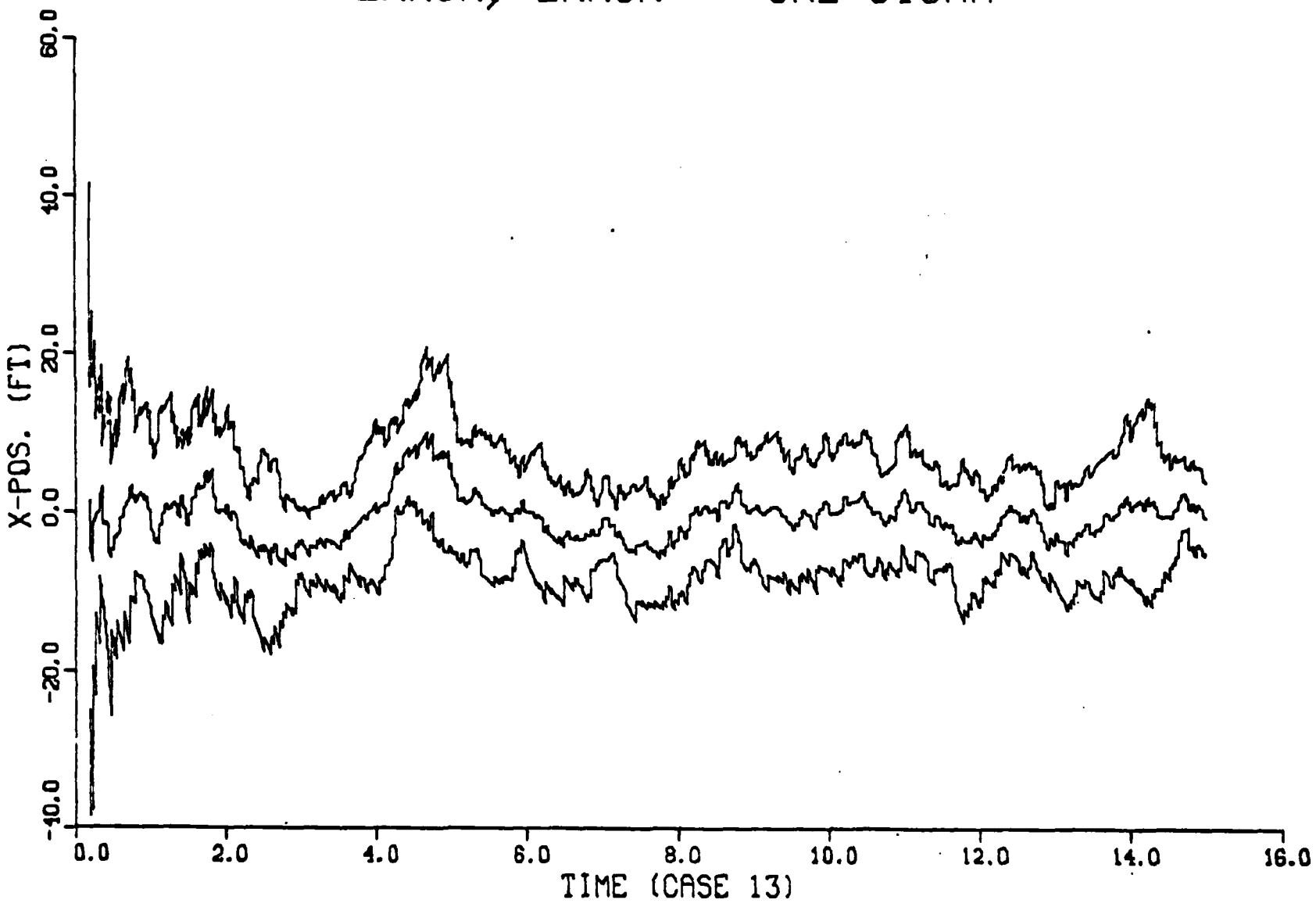

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
×10³
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
-15.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 12)

1.3.4

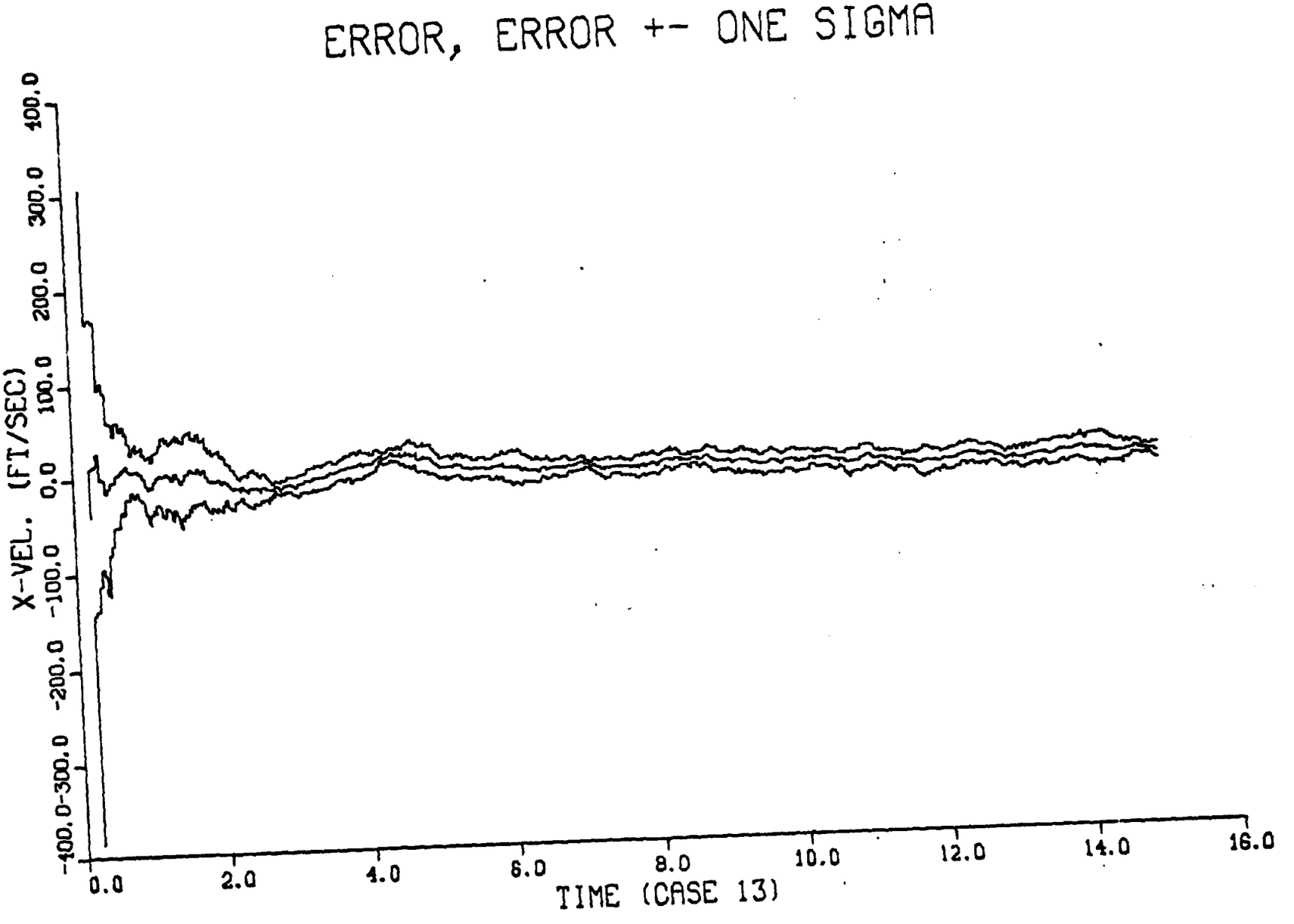

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 12)


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 12)

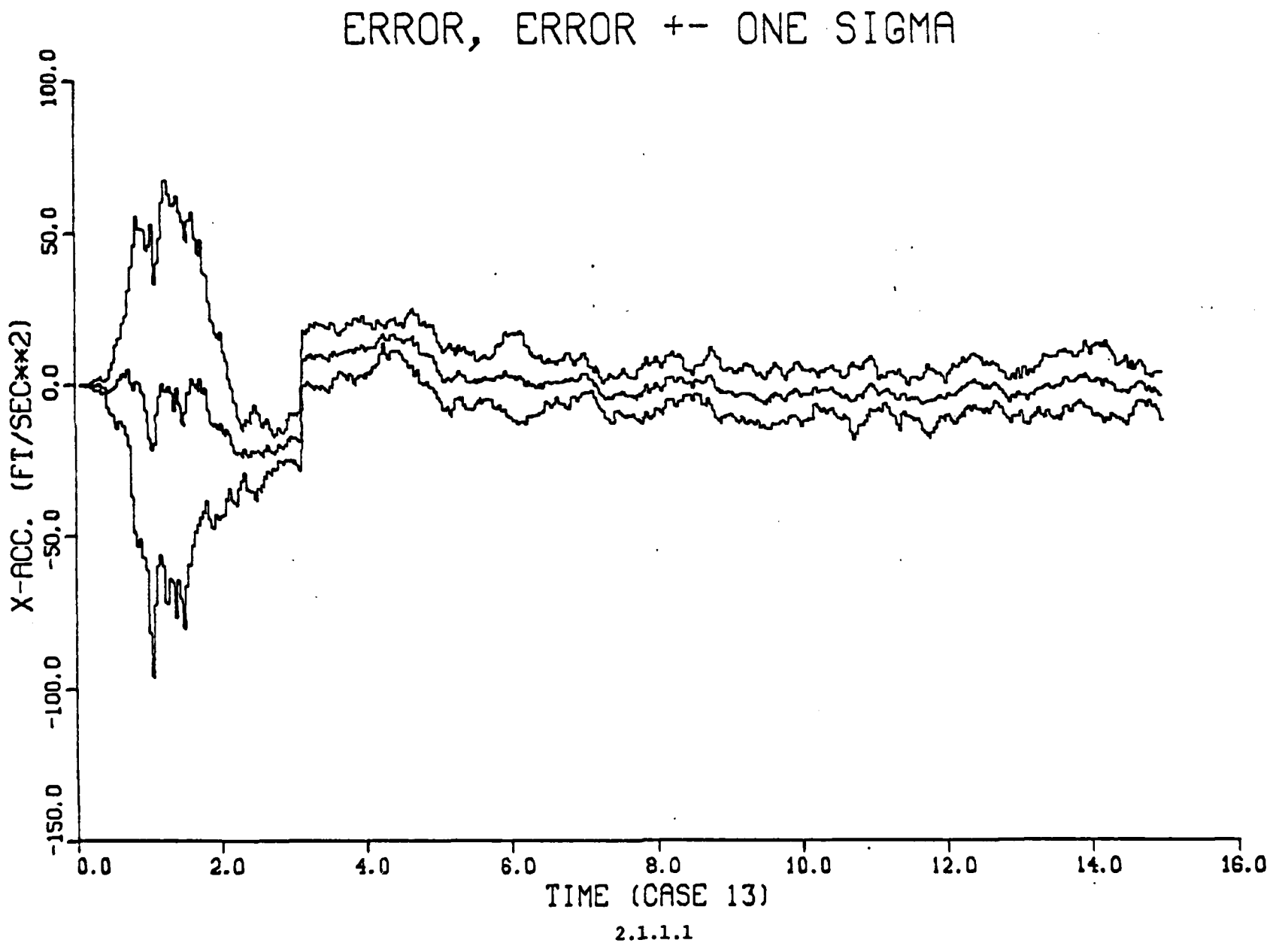

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
2000.0
1000.0
0.0
-1000.0
-2000.0
-3000.0
-4000.0
-5000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 12)
1.3.4

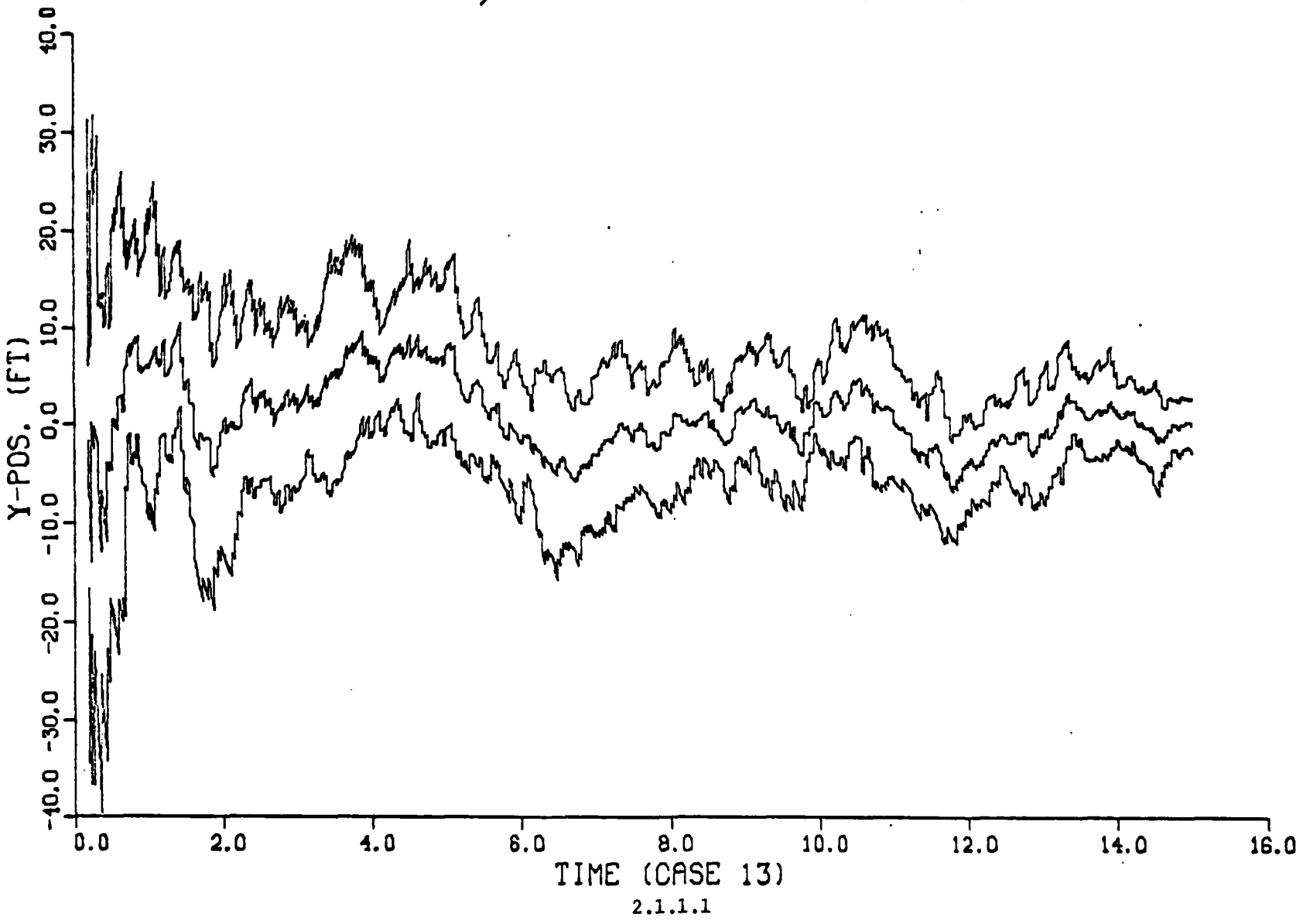

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)

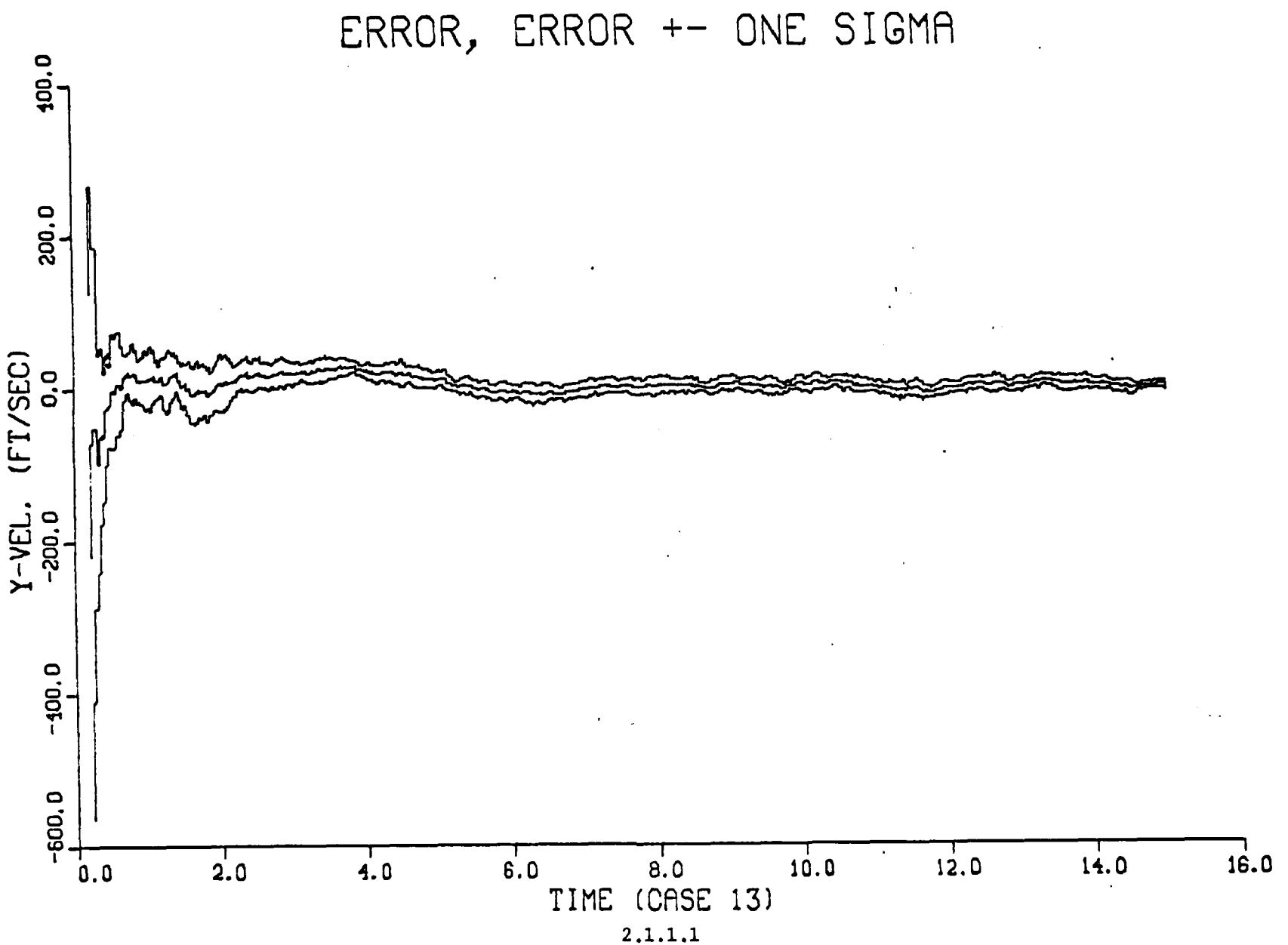
2.1.1.1

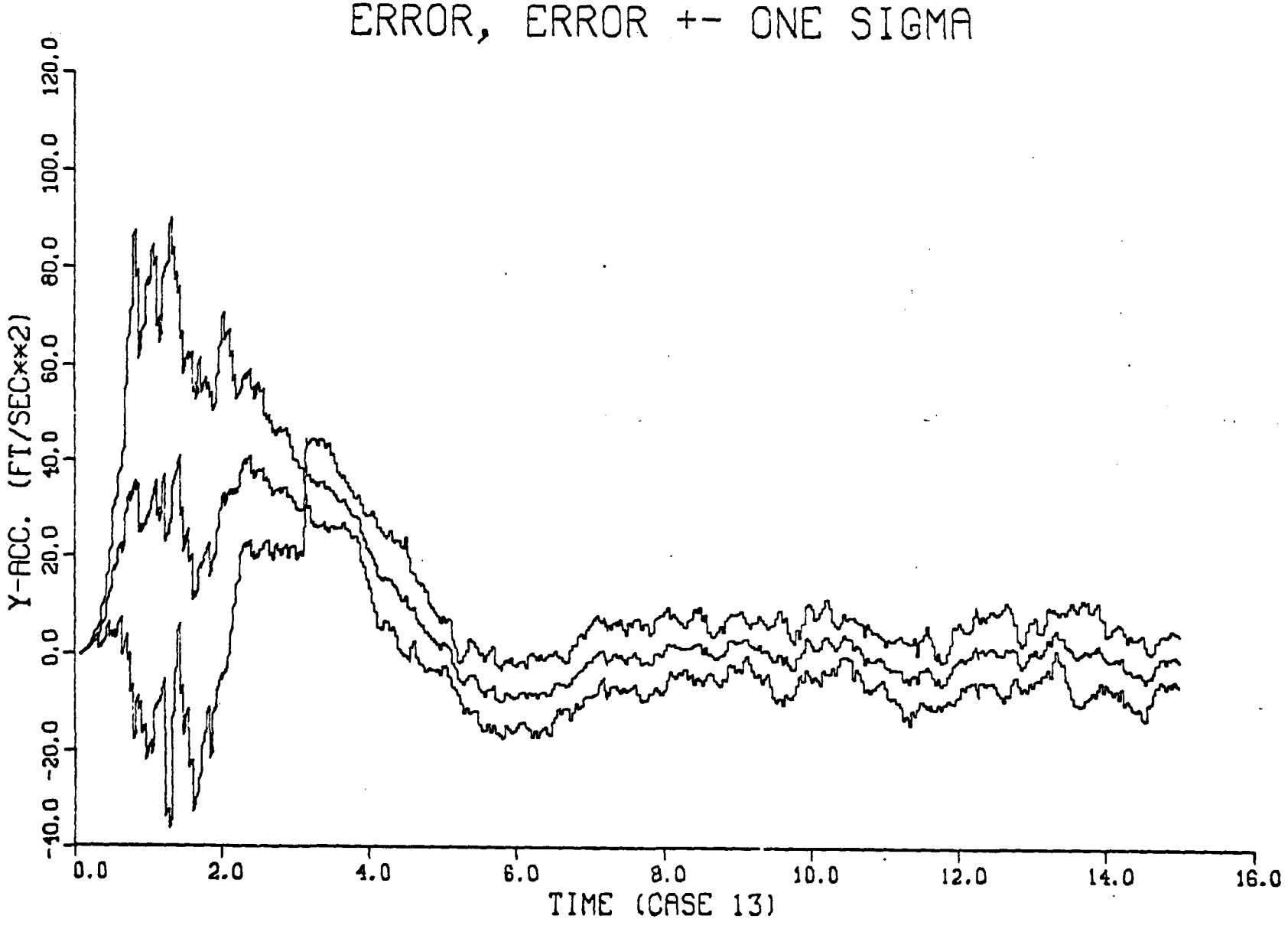

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)

2.1.1.1

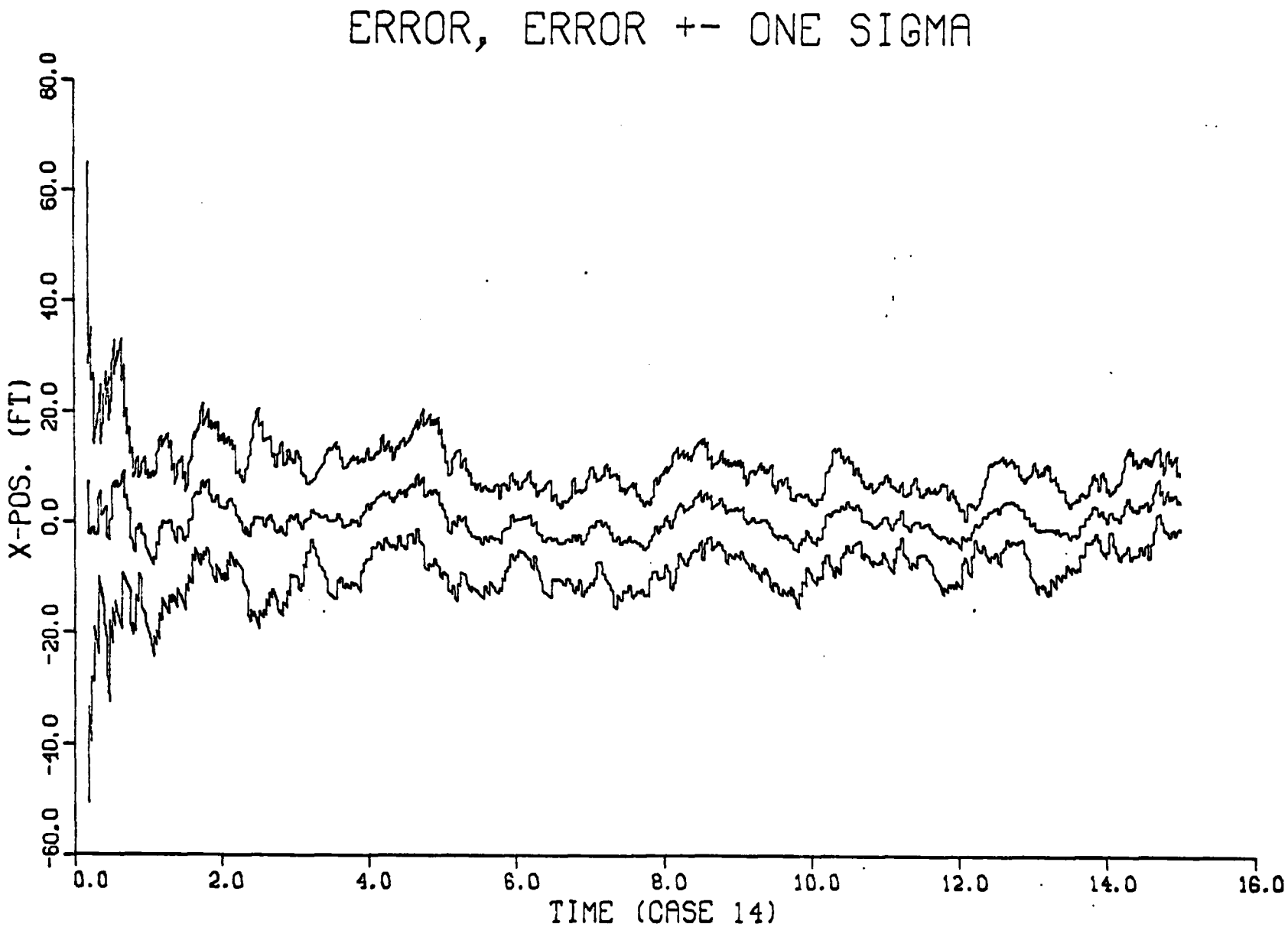

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)
2.1.1.1


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
40.0
30.0
20.0
10.0
0.0
-10.0
-20.0
-30.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)
2.1.1.1

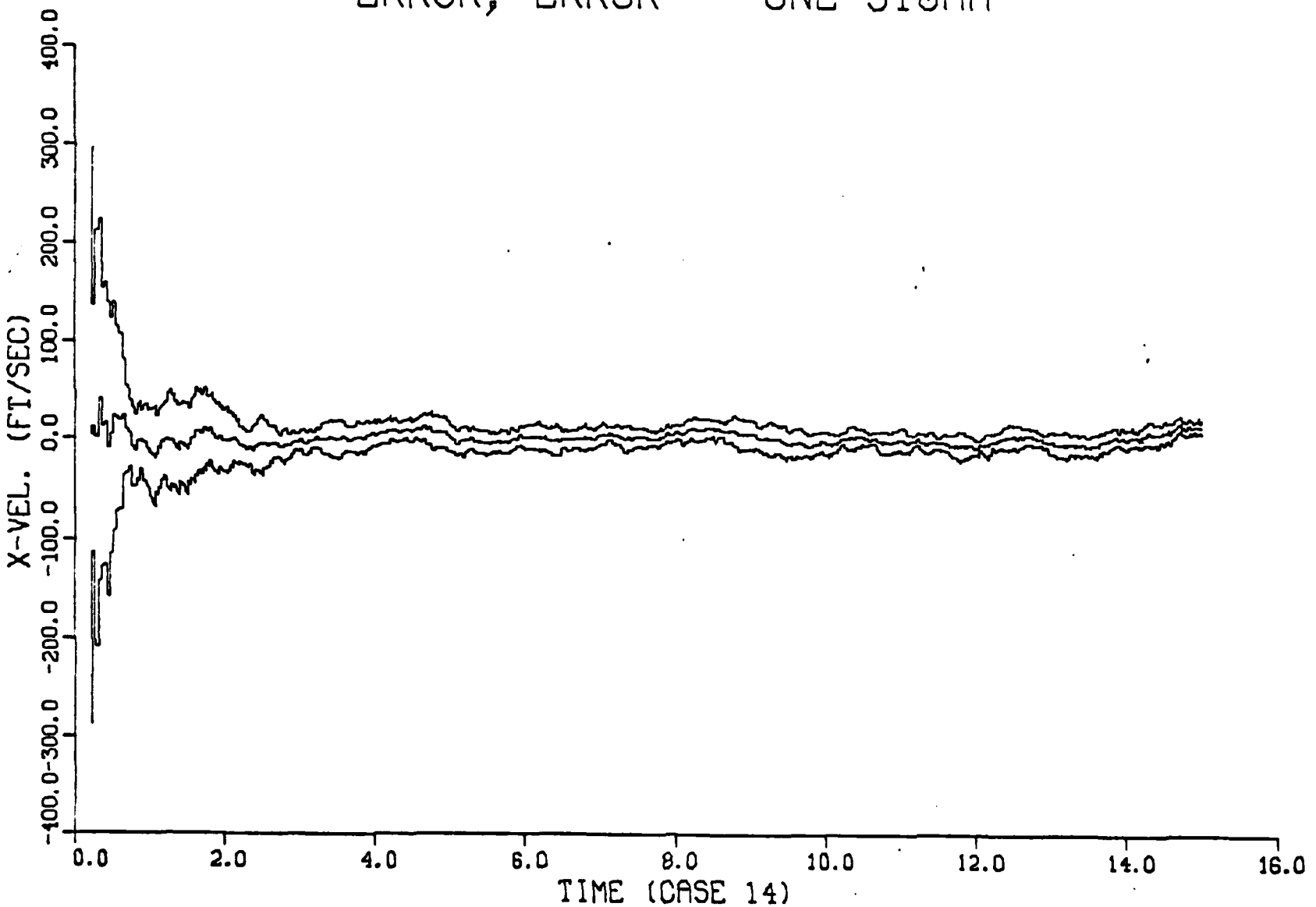

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
100.0
200.0
0.0
-200.0
-100.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)
2.1.1.1


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 13)

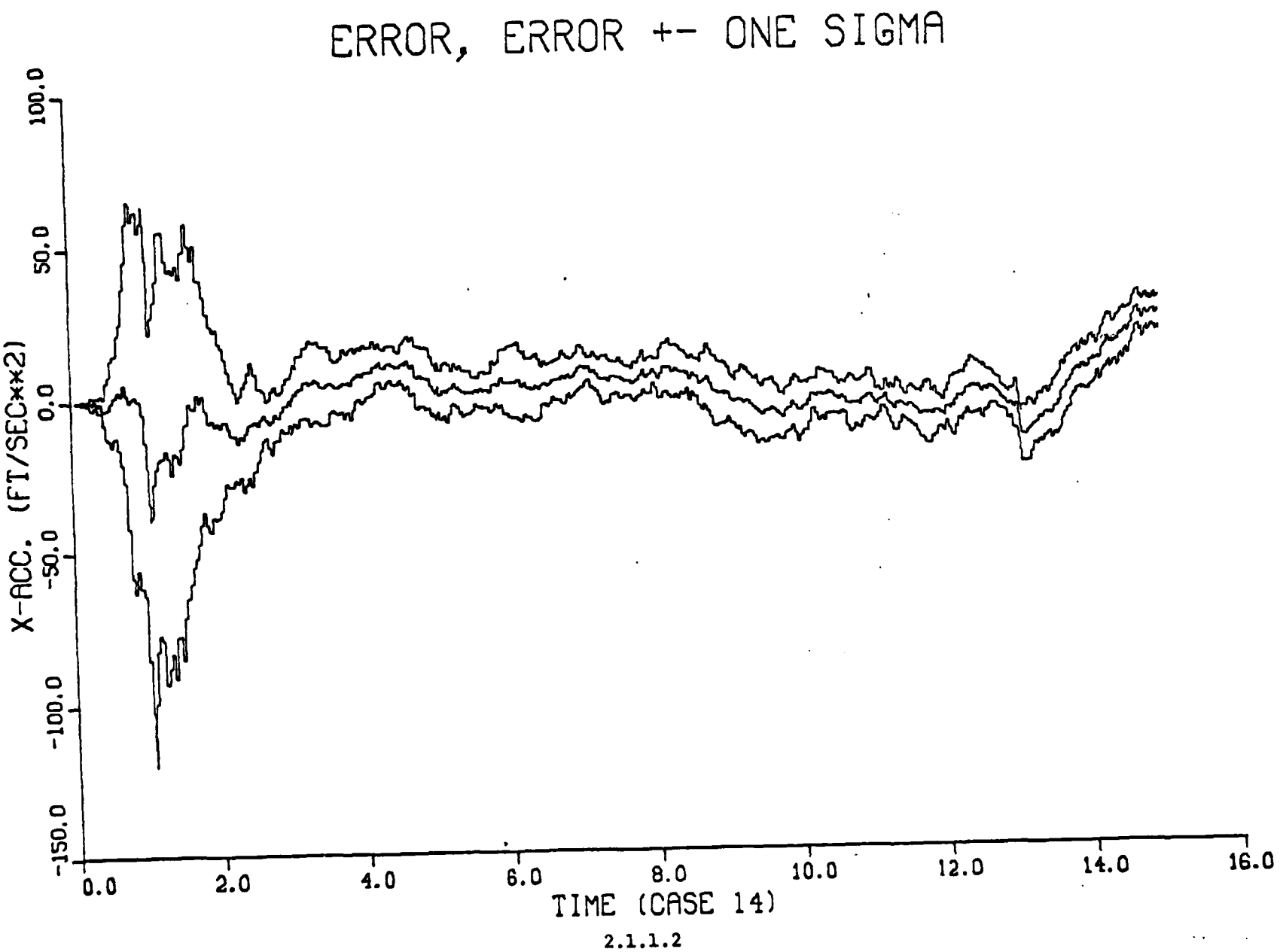
2.1.1.1

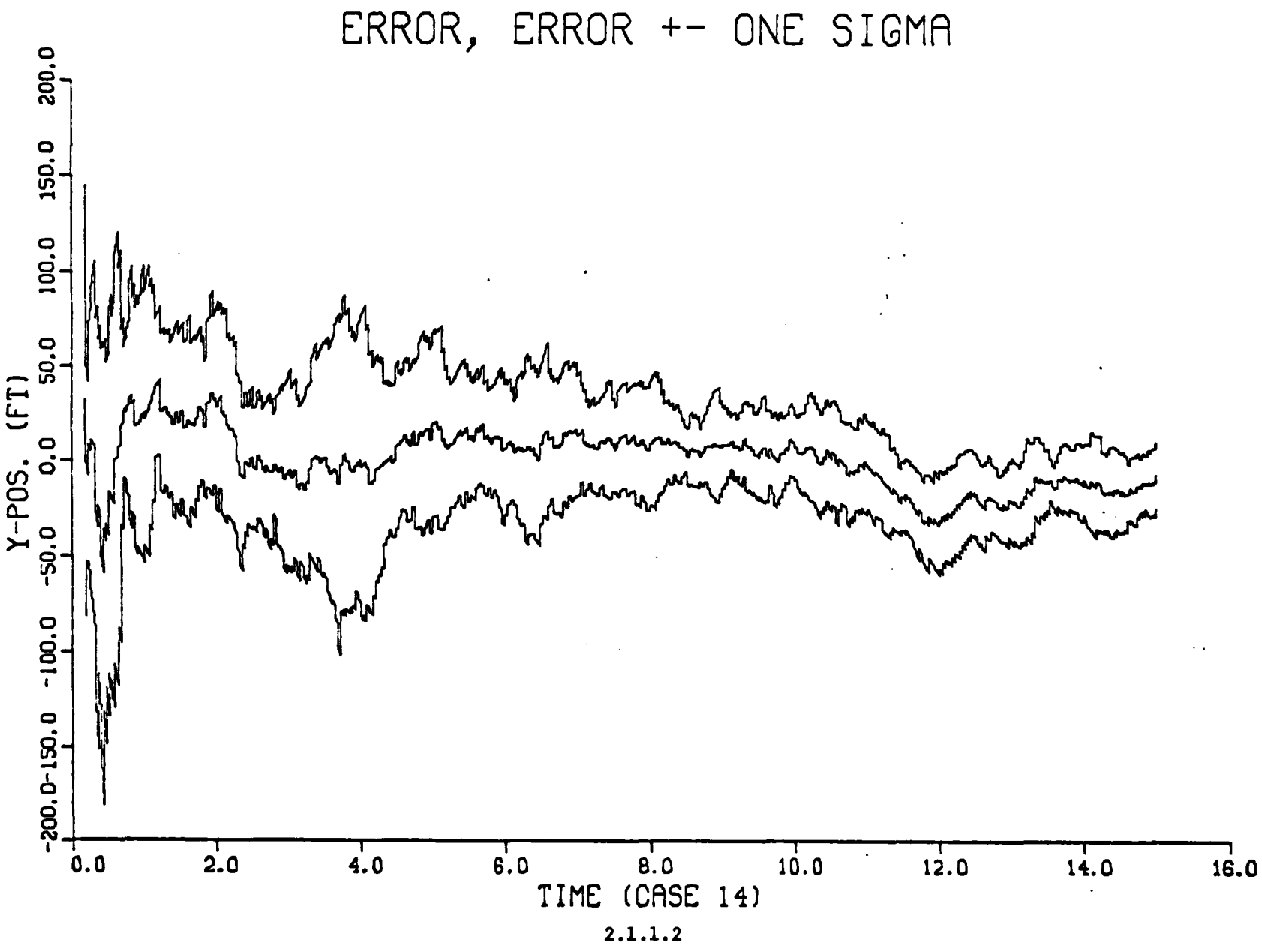

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)

2.1.1.2

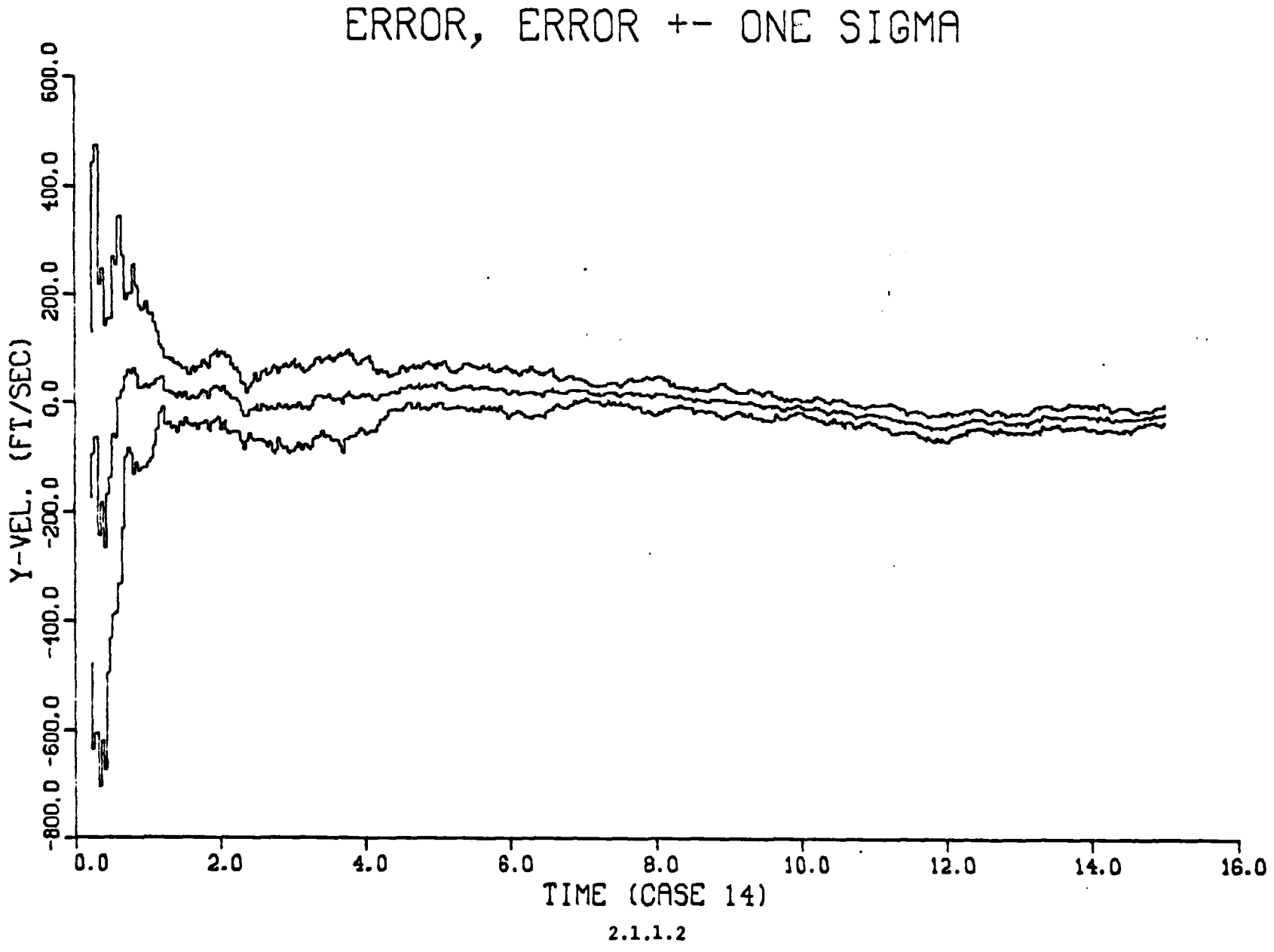

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)

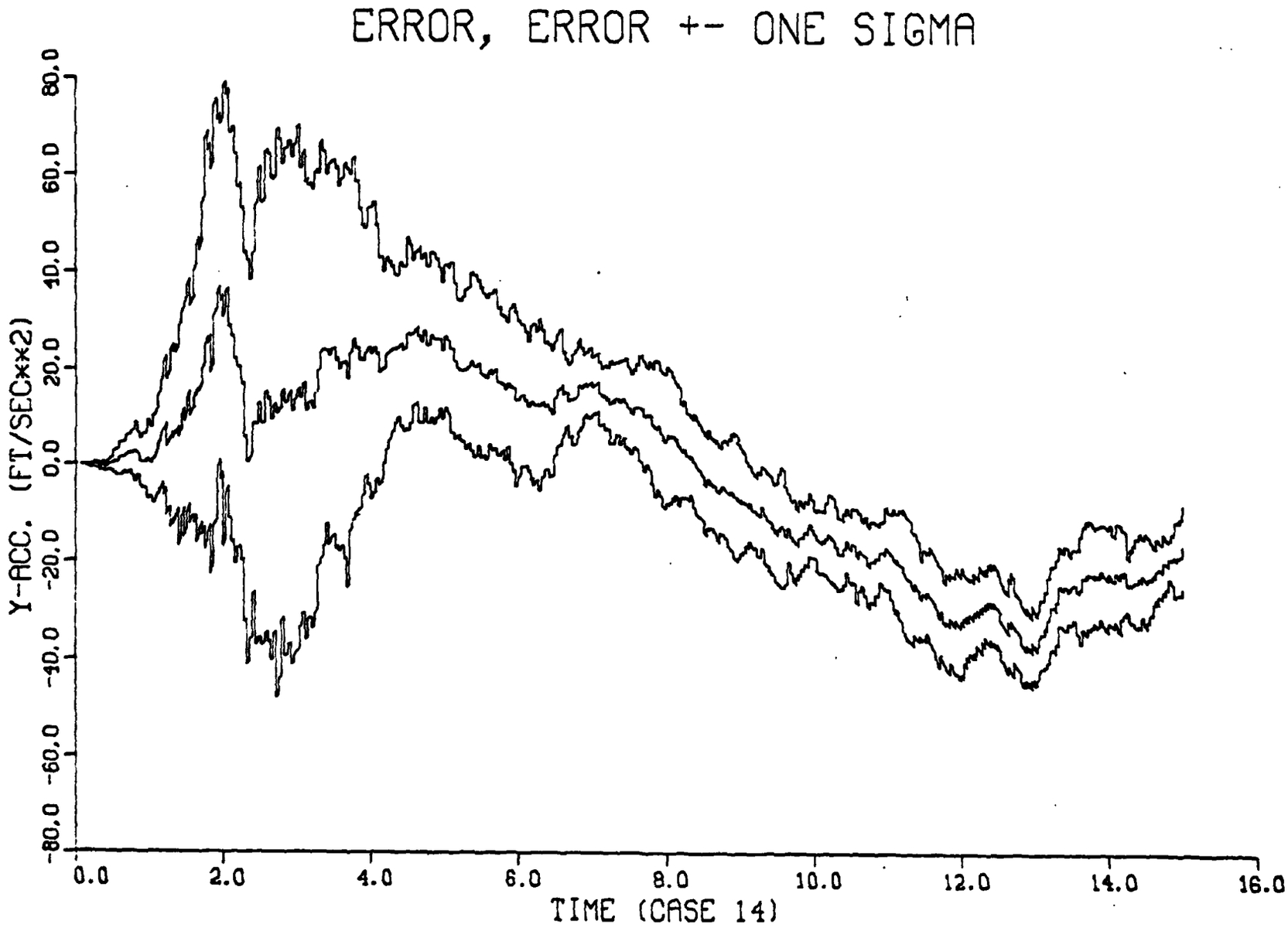
2.1.1.2


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)
2.1.1.2

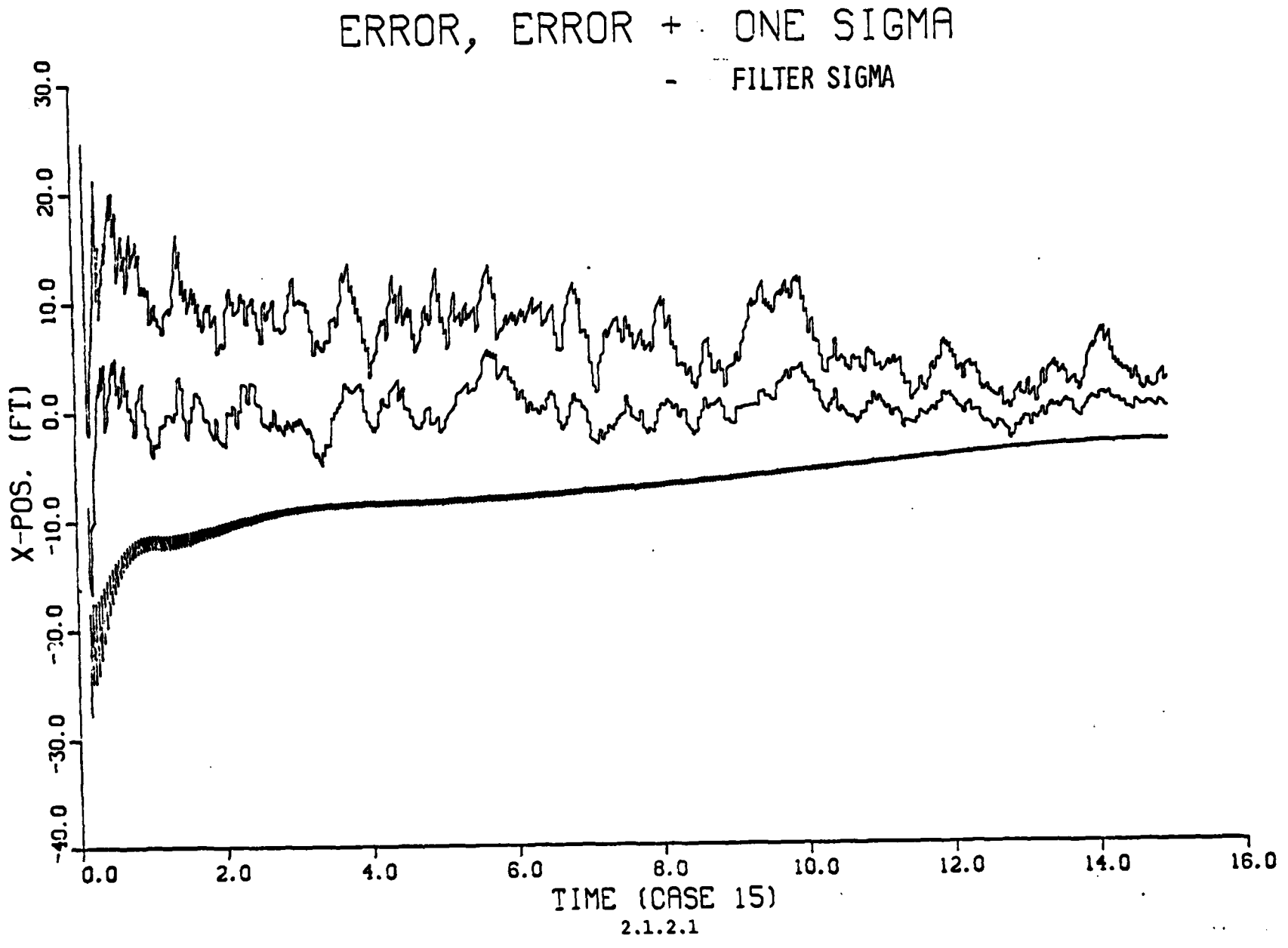

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)

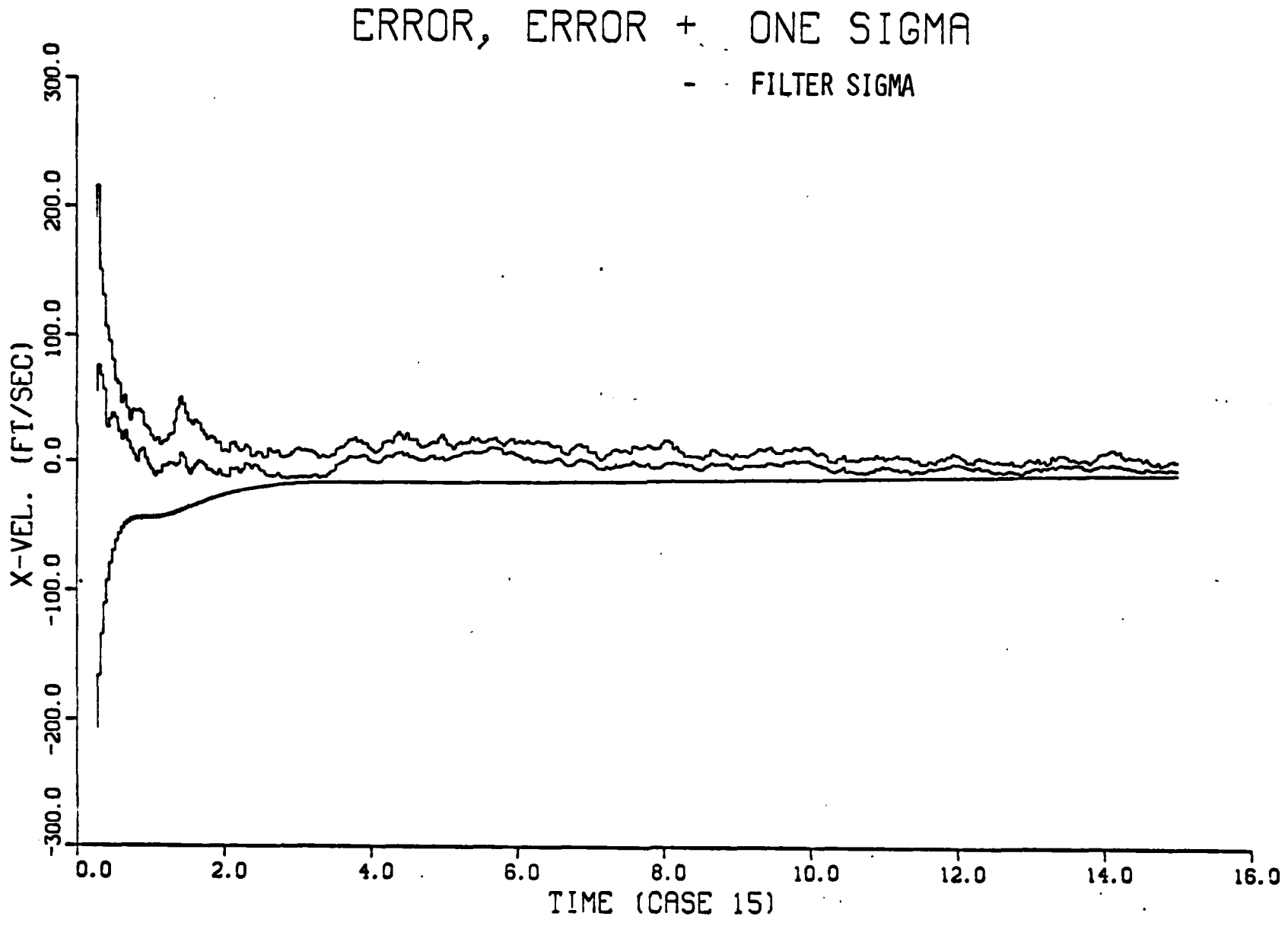
2.1.1.2

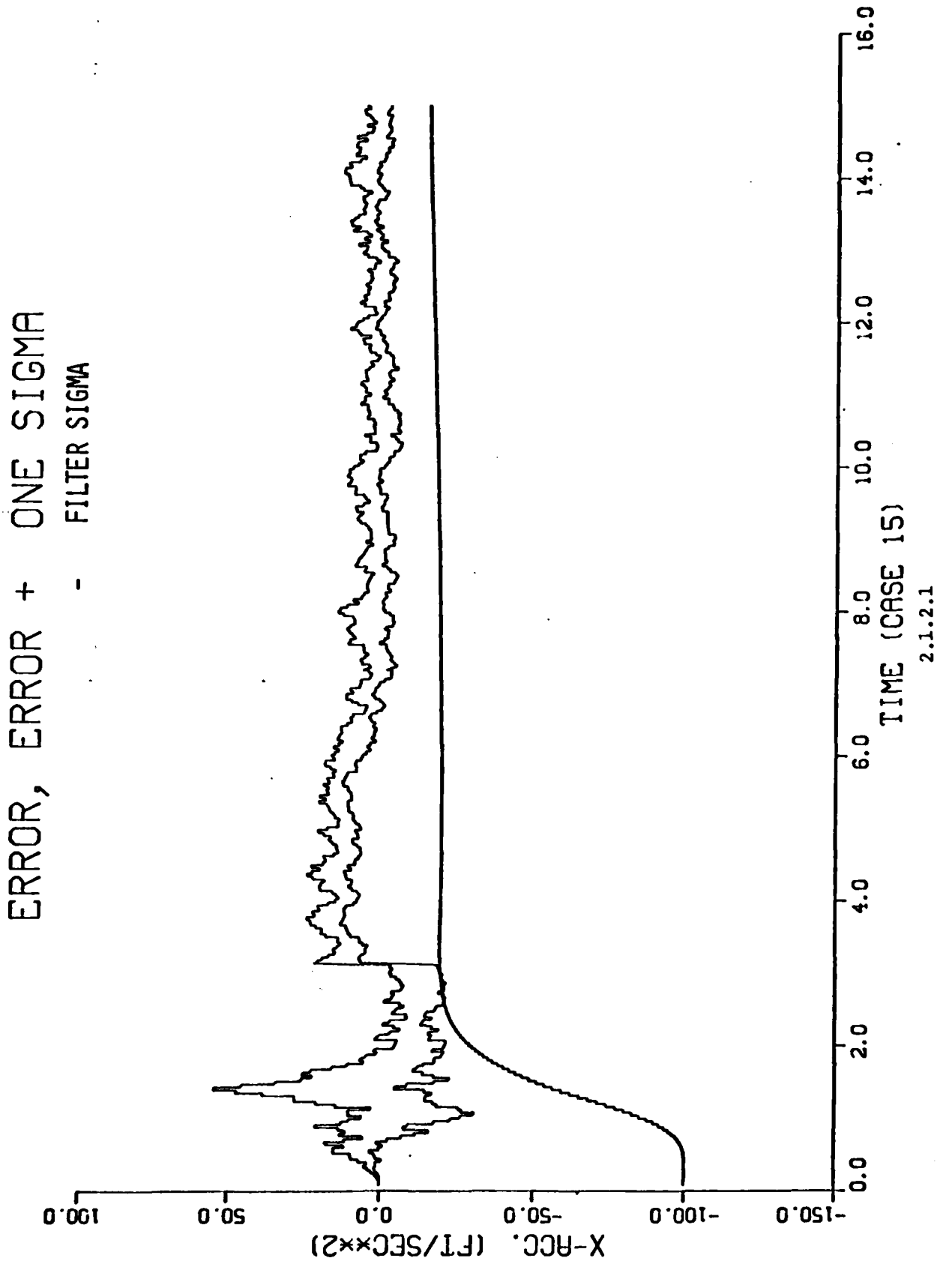

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-100.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)
2.1.1.2

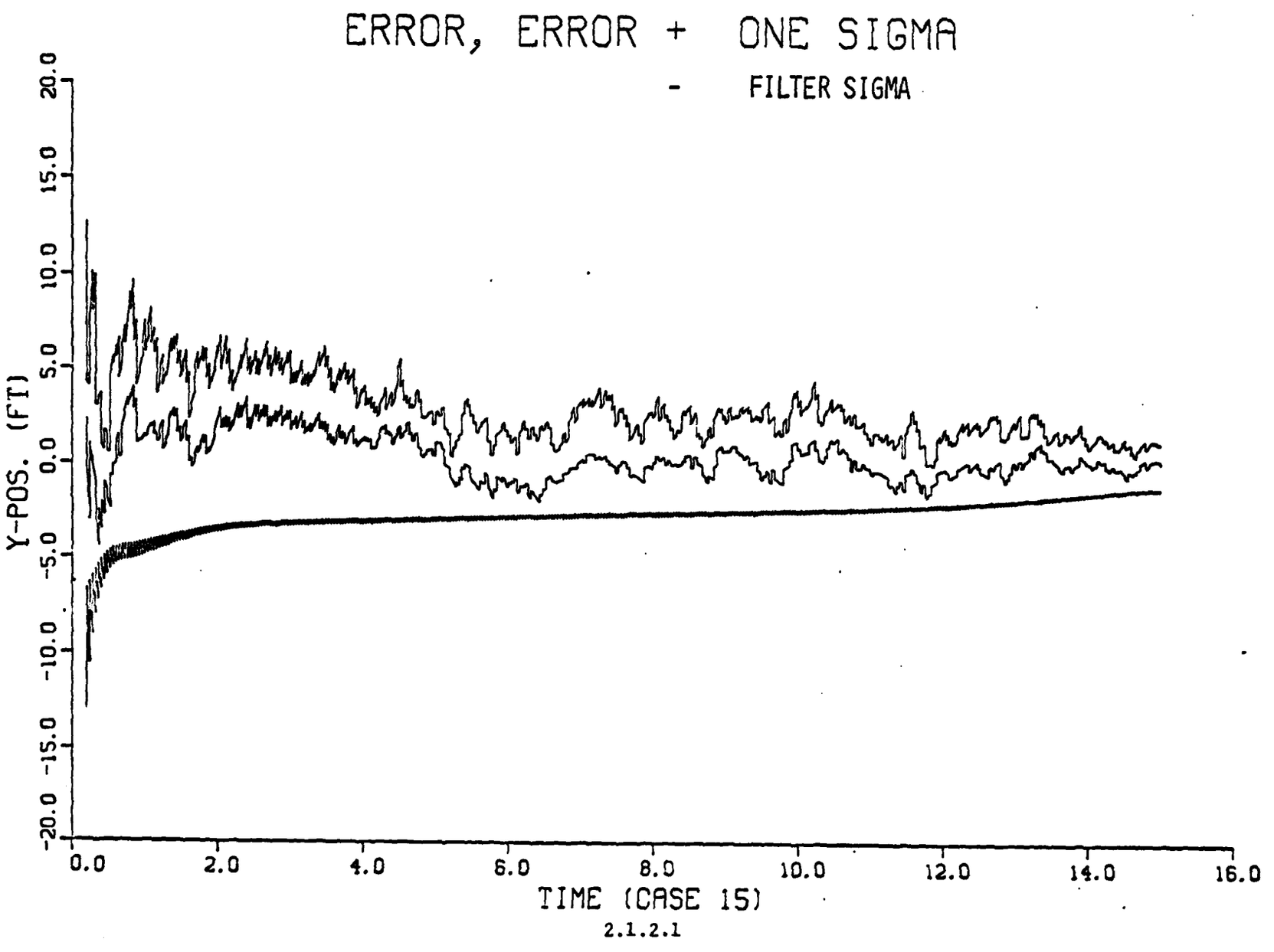

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 14)

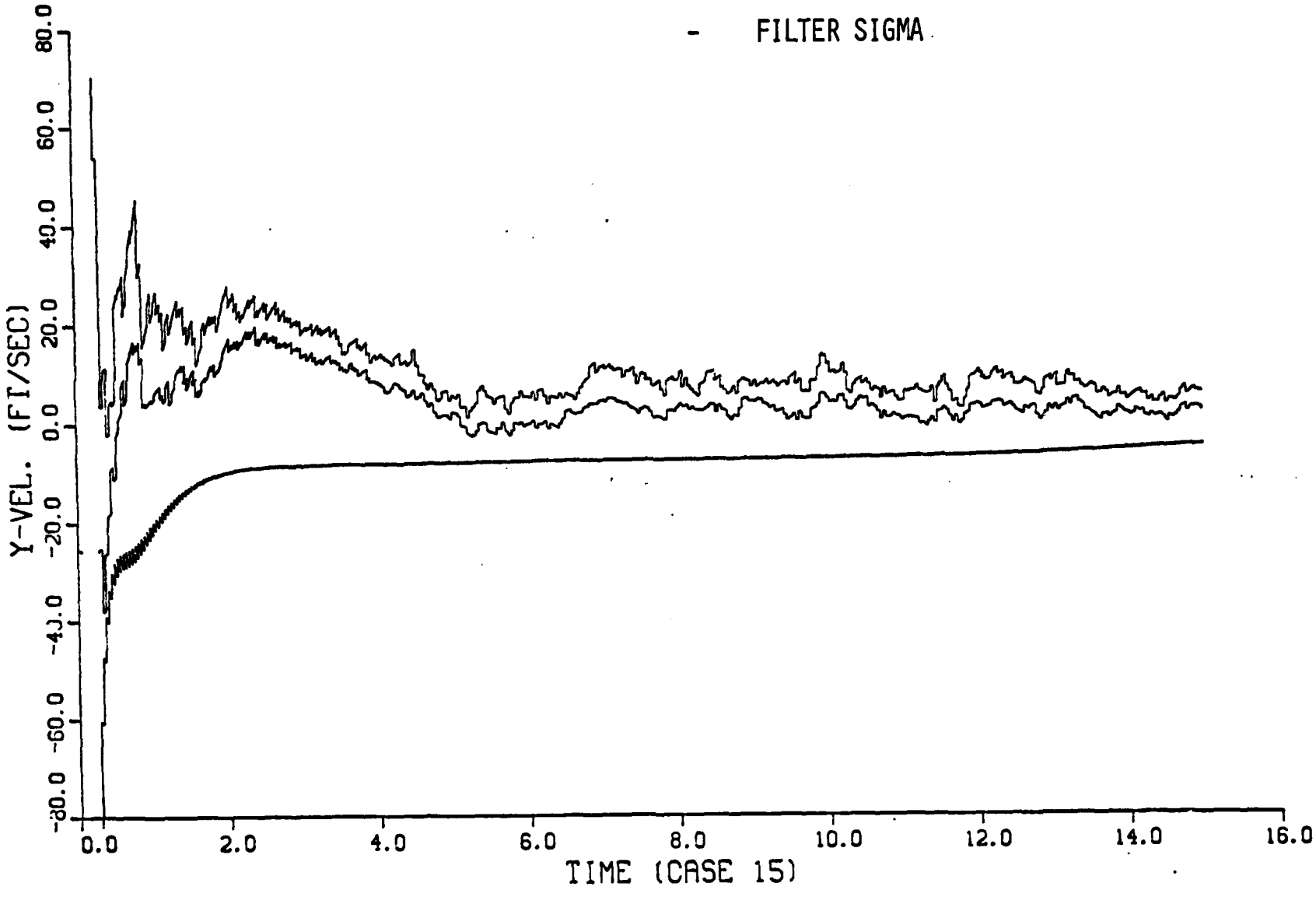
2.1.1.2


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
30.0
20.0
10.0
0.0
-10.0
-20.0
-30.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)
2.1.2.1

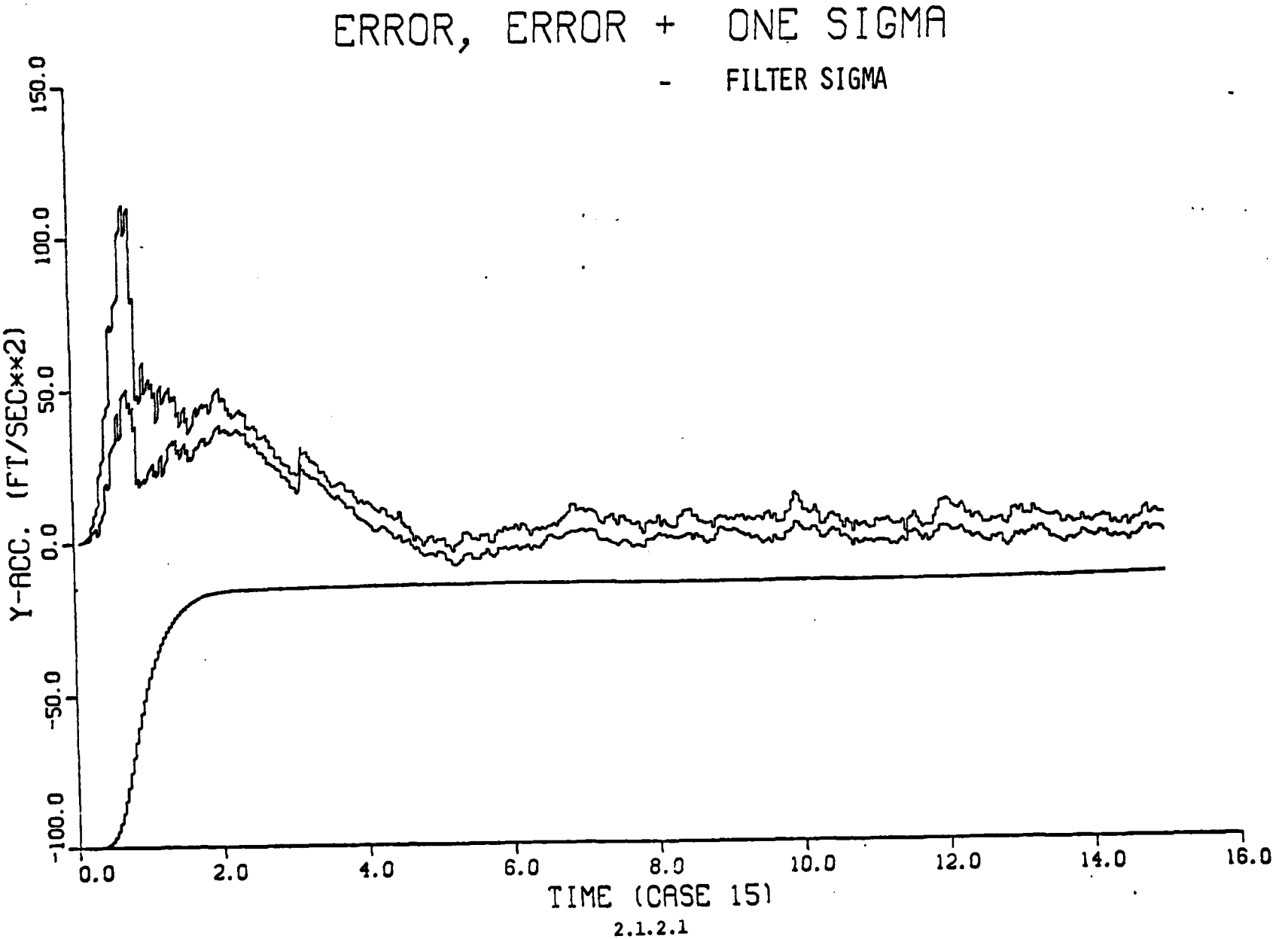

ERROR, ERROR + ONE SIGMA
- - FILTER SIGMA
X-VEL. (FT/SEC)
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)

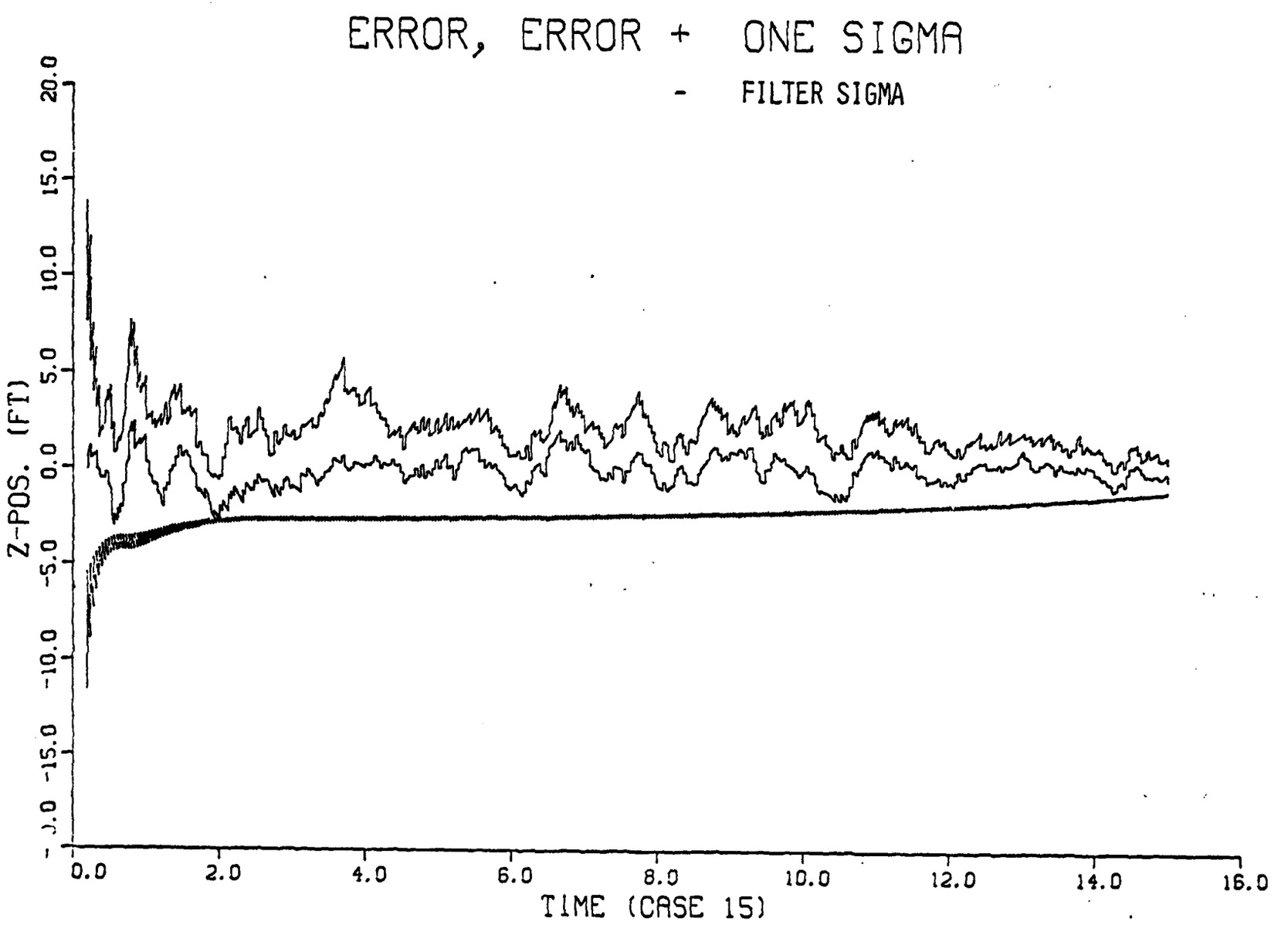

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)
2.1.2.1


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
-15.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)
2.1.2.1

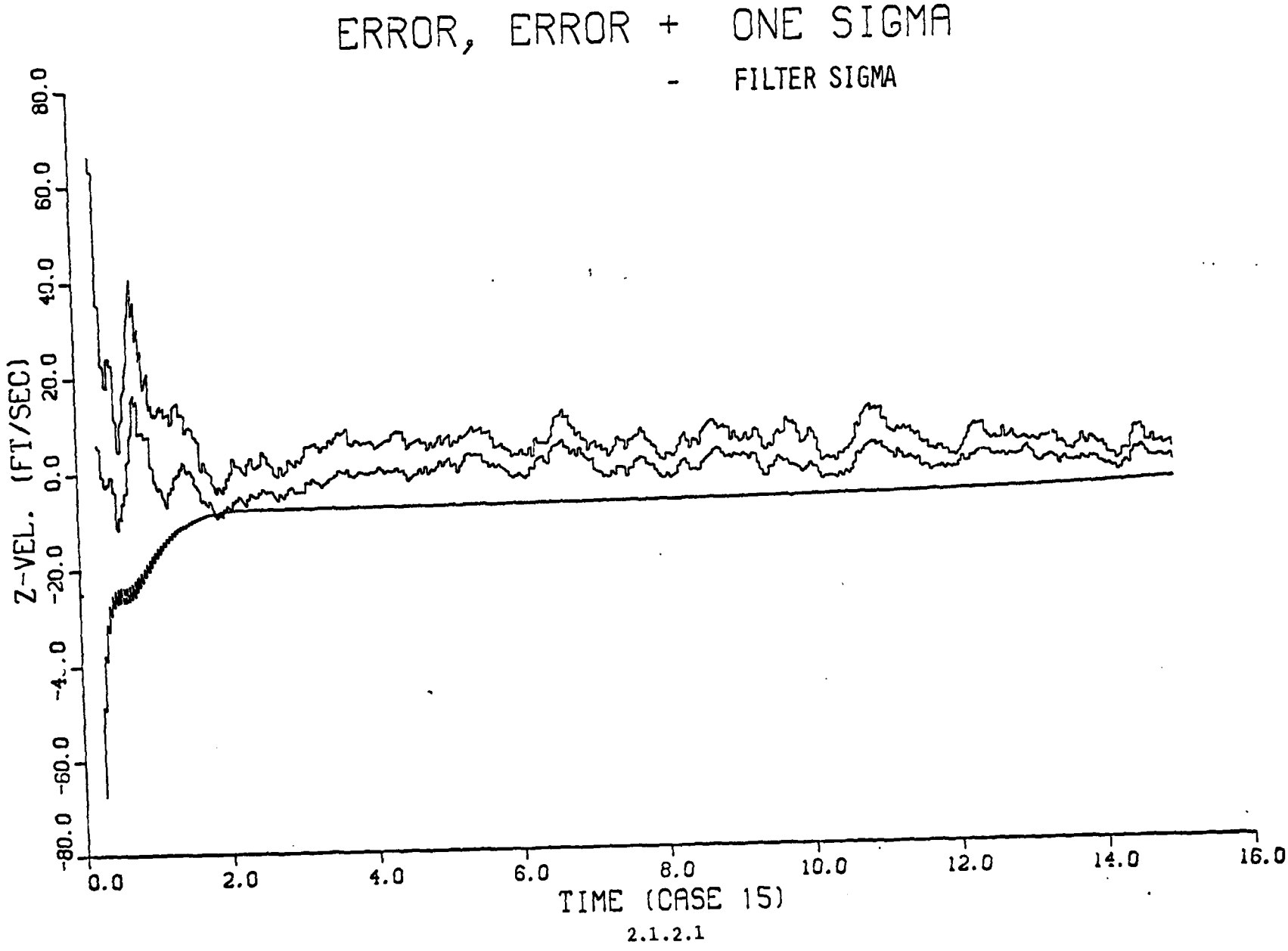

ERROR, ERROR + : ONE SIGMA
- FILTER SIGMA
Y-VEL. (FT/SEC)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)

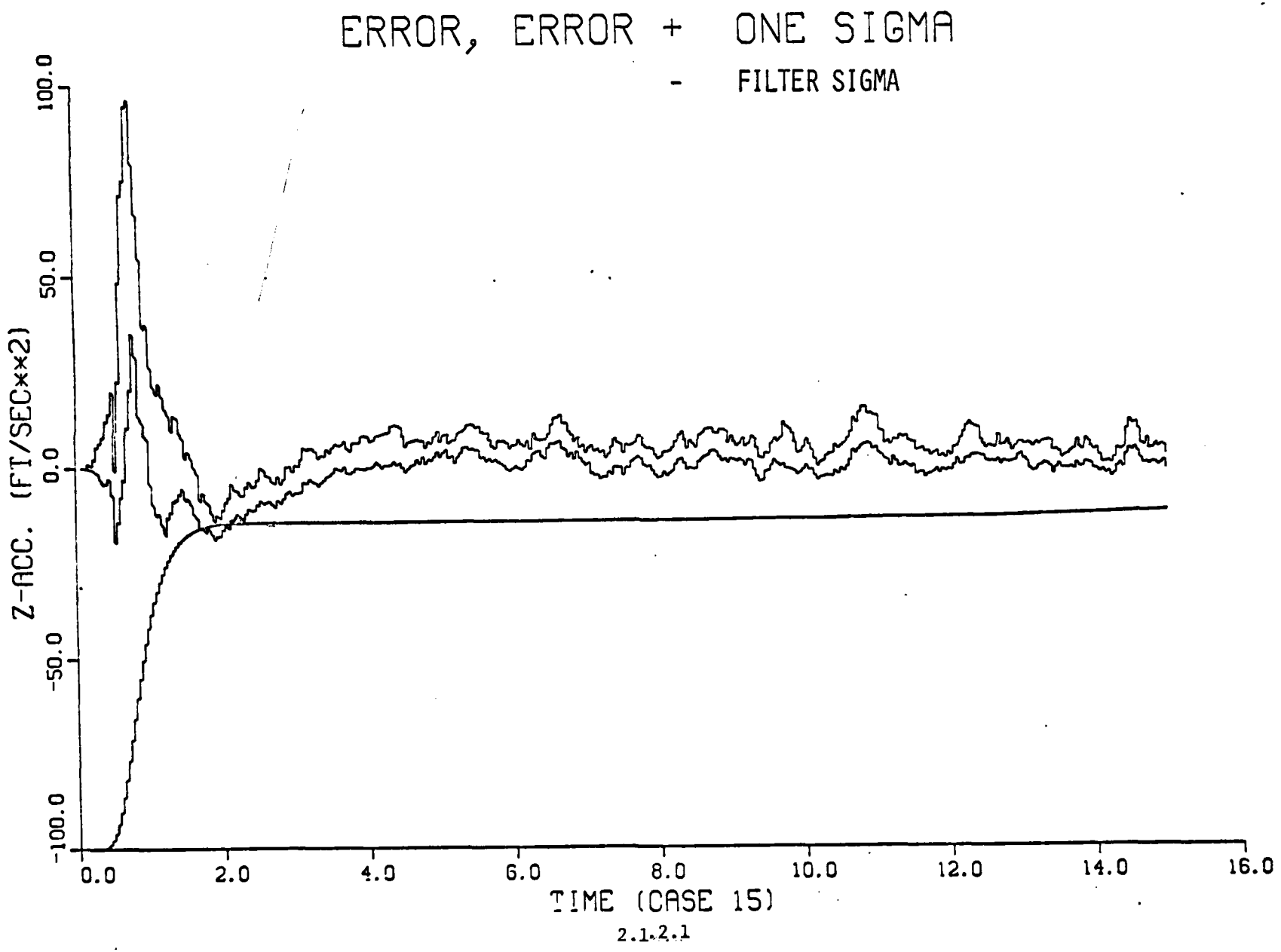
2.1.2.1


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)
2.1.2.1

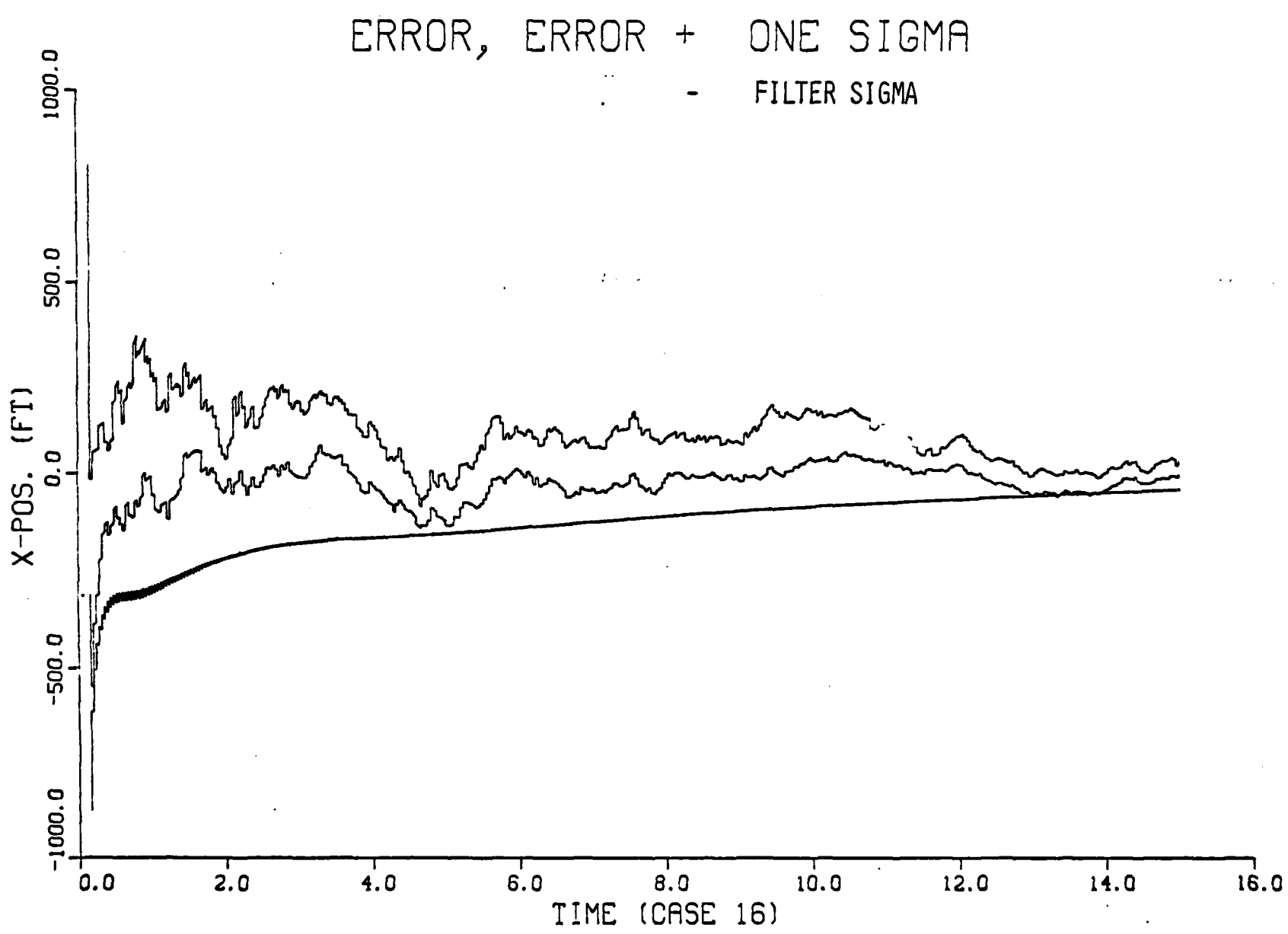

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Z-POS. (FT)
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
-15.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)

2.1:2.1

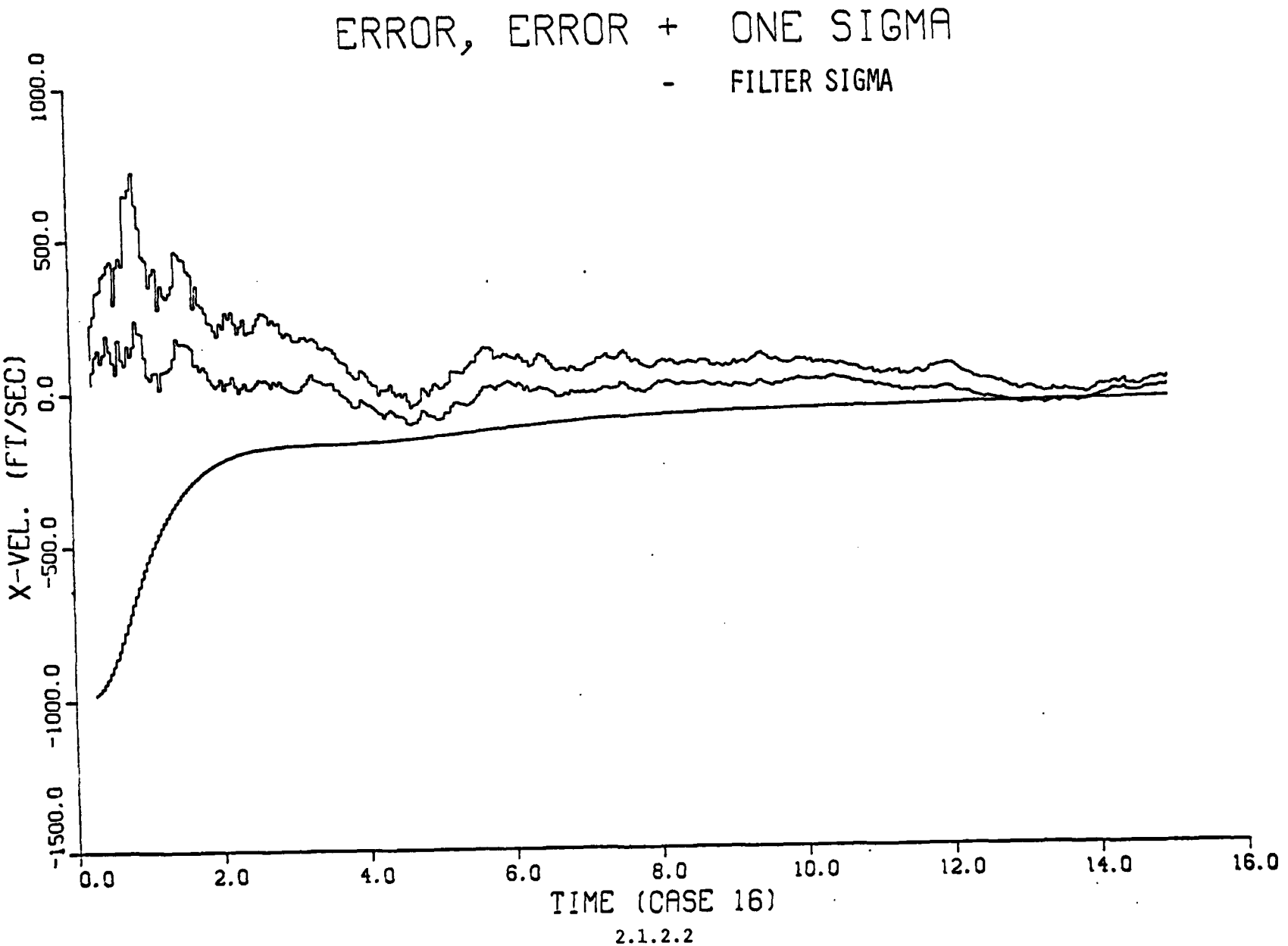

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Z-VEL. (FT/SEC)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)
2.1.2.1


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Z-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 15)

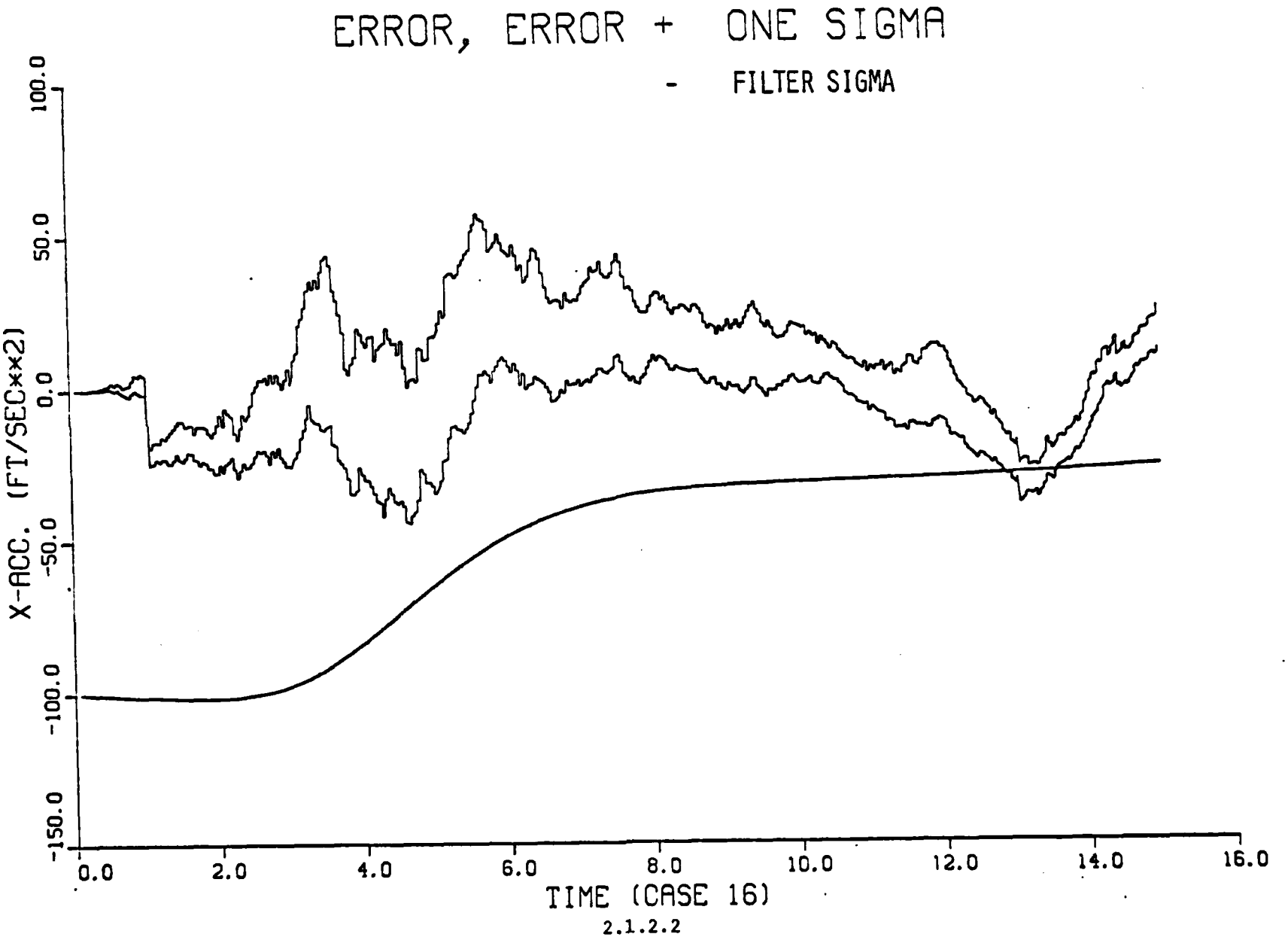
2.1.2.1

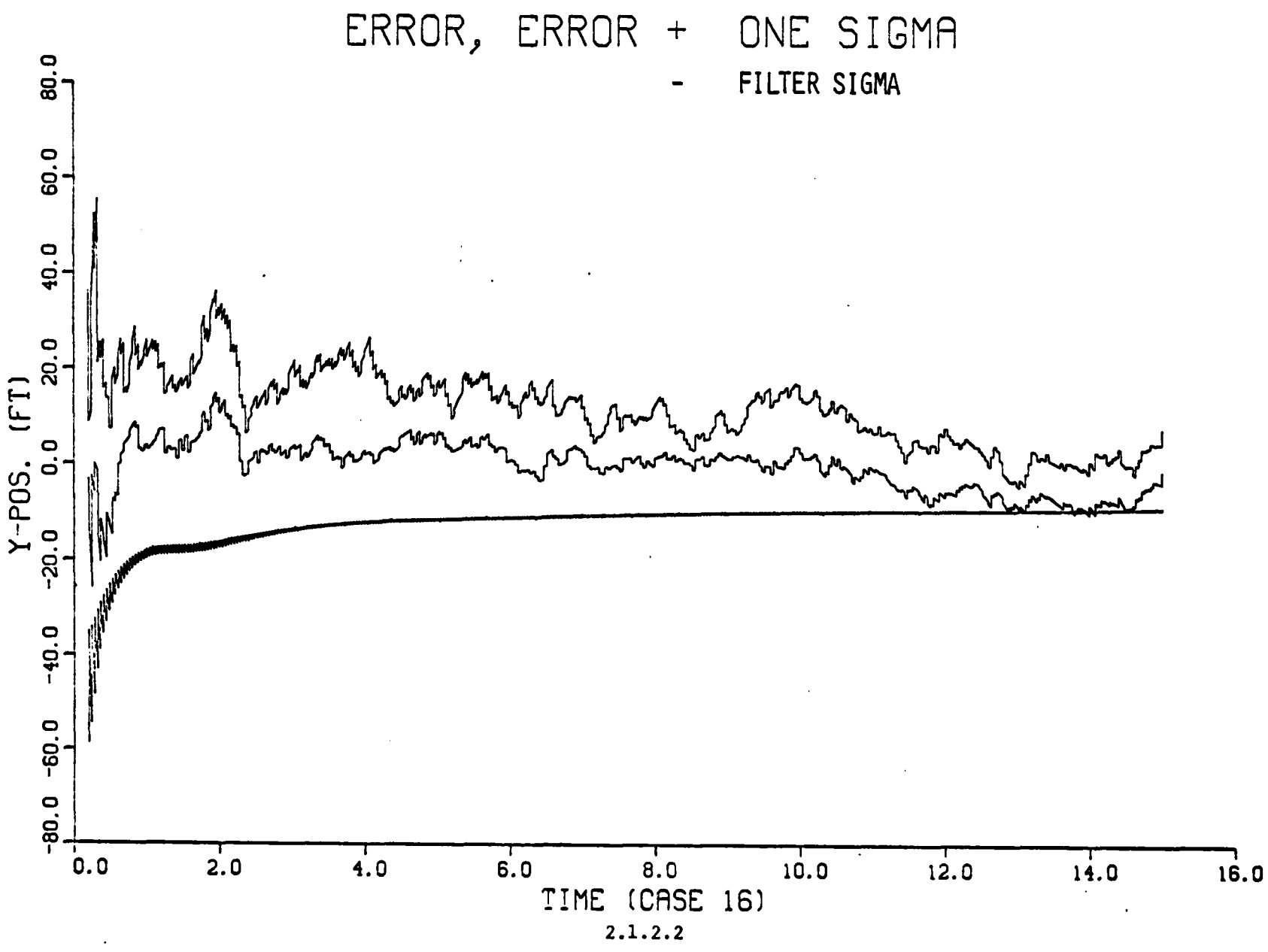

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

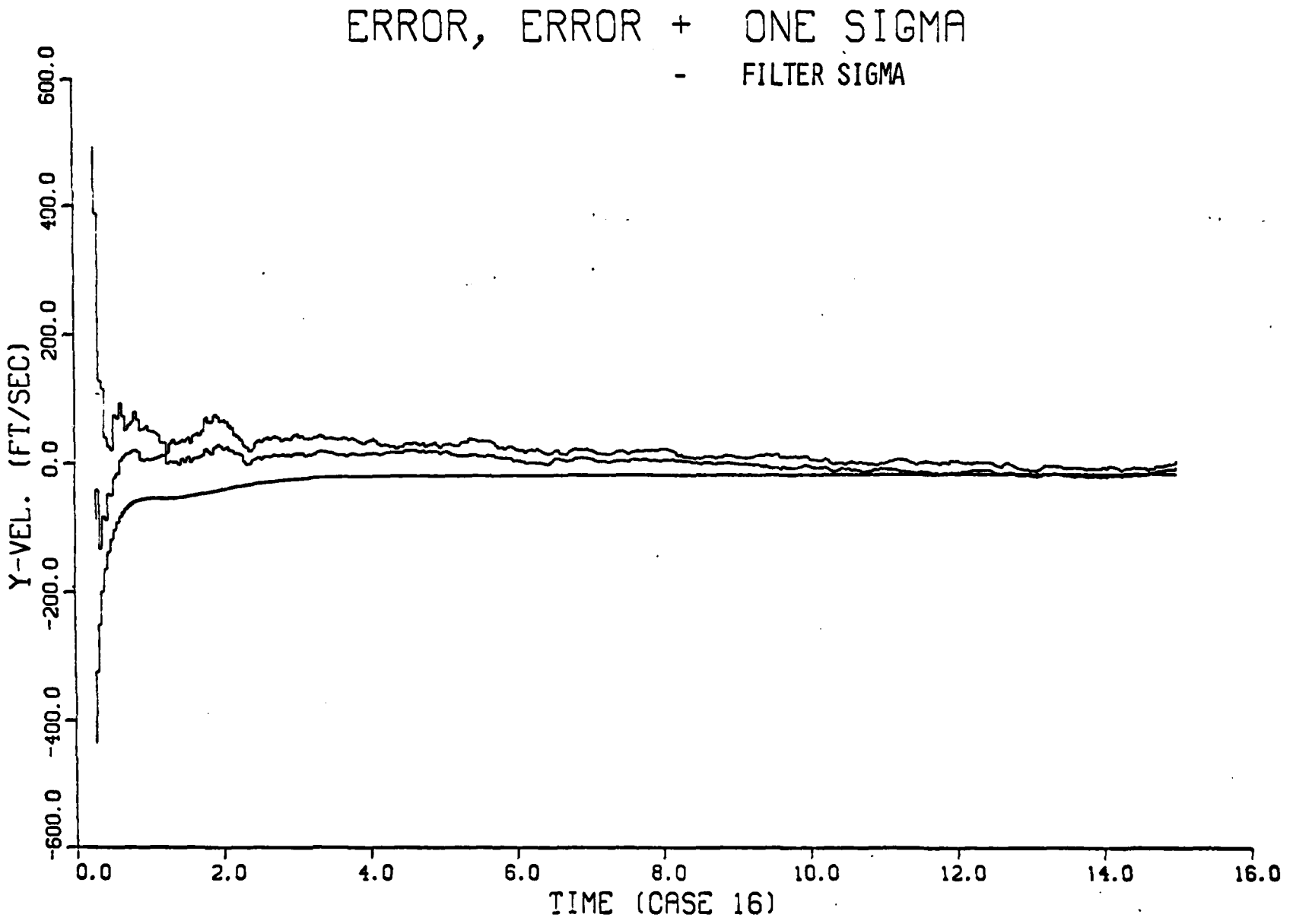
2.1.2.2


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

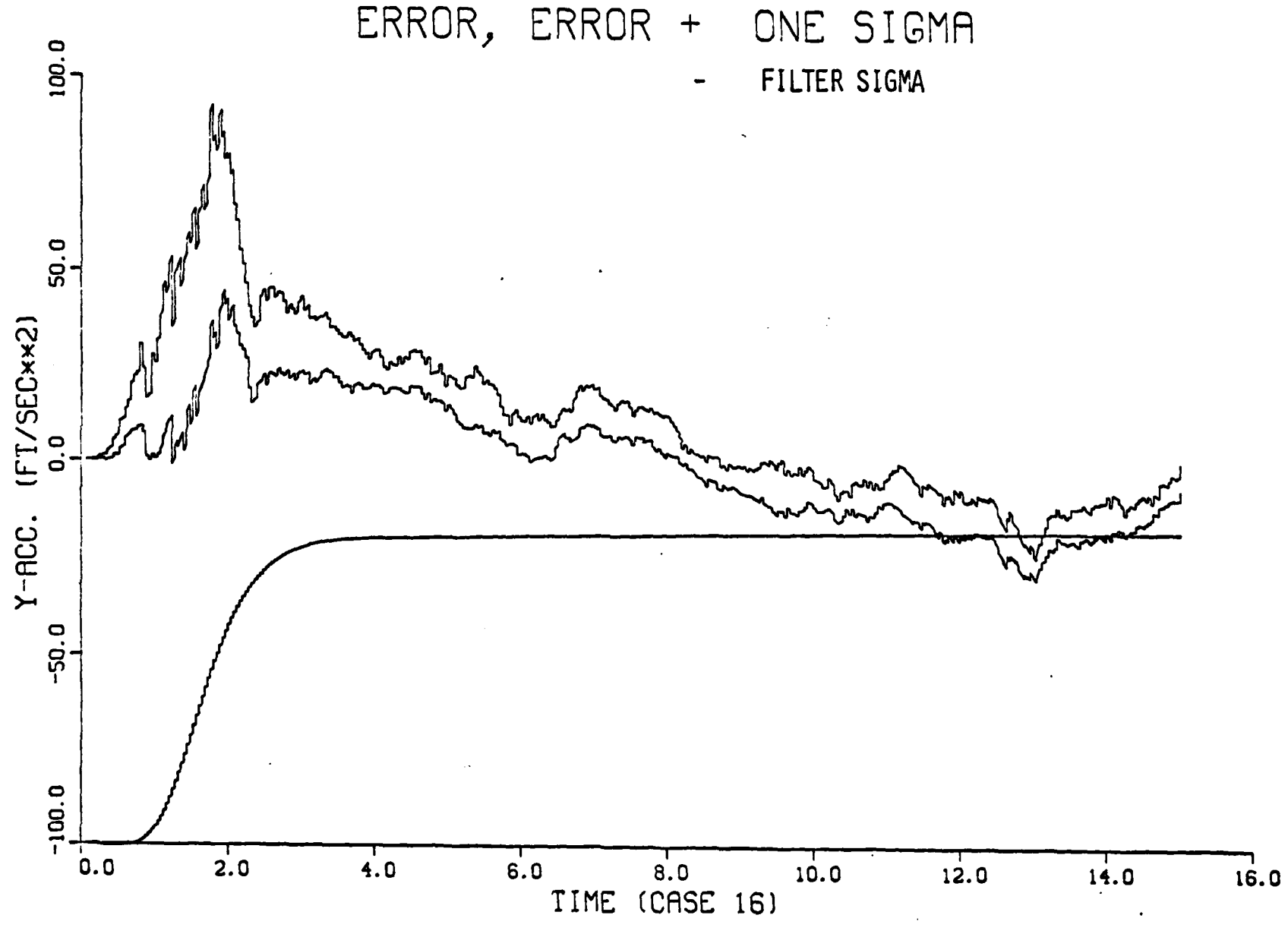
2.1.2.2


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)
2.1.2.2

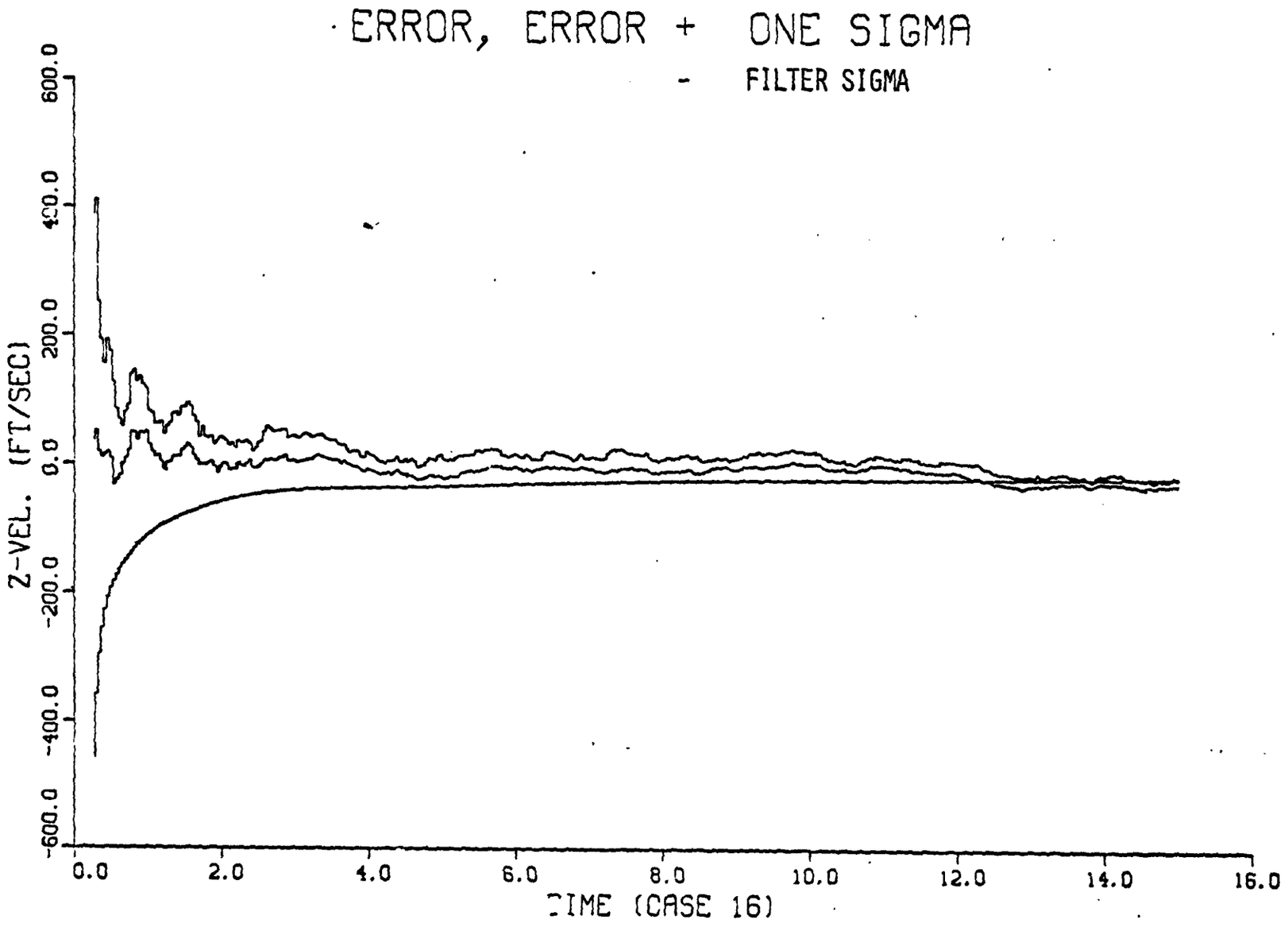

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)
2.1.2.2

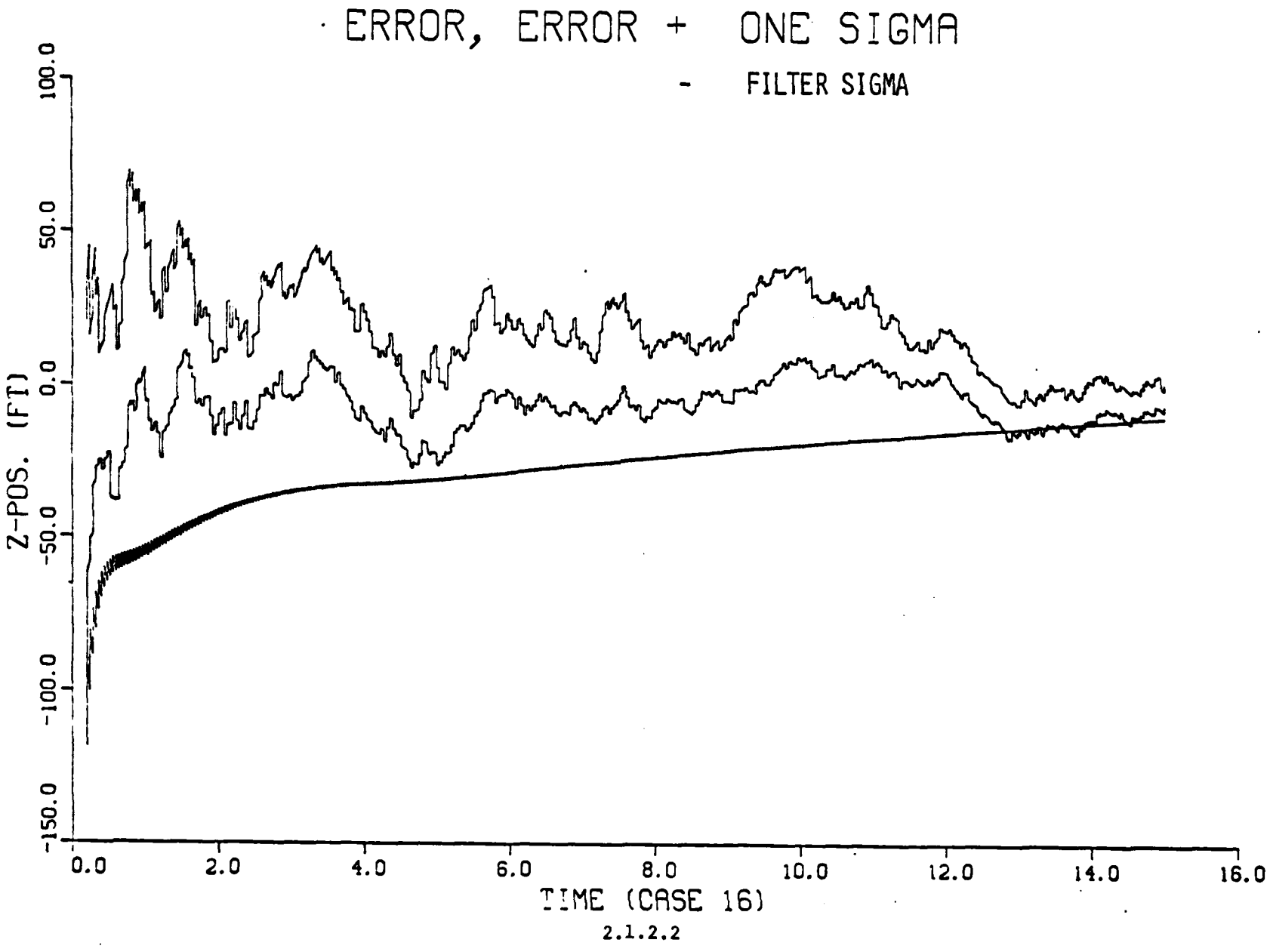

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

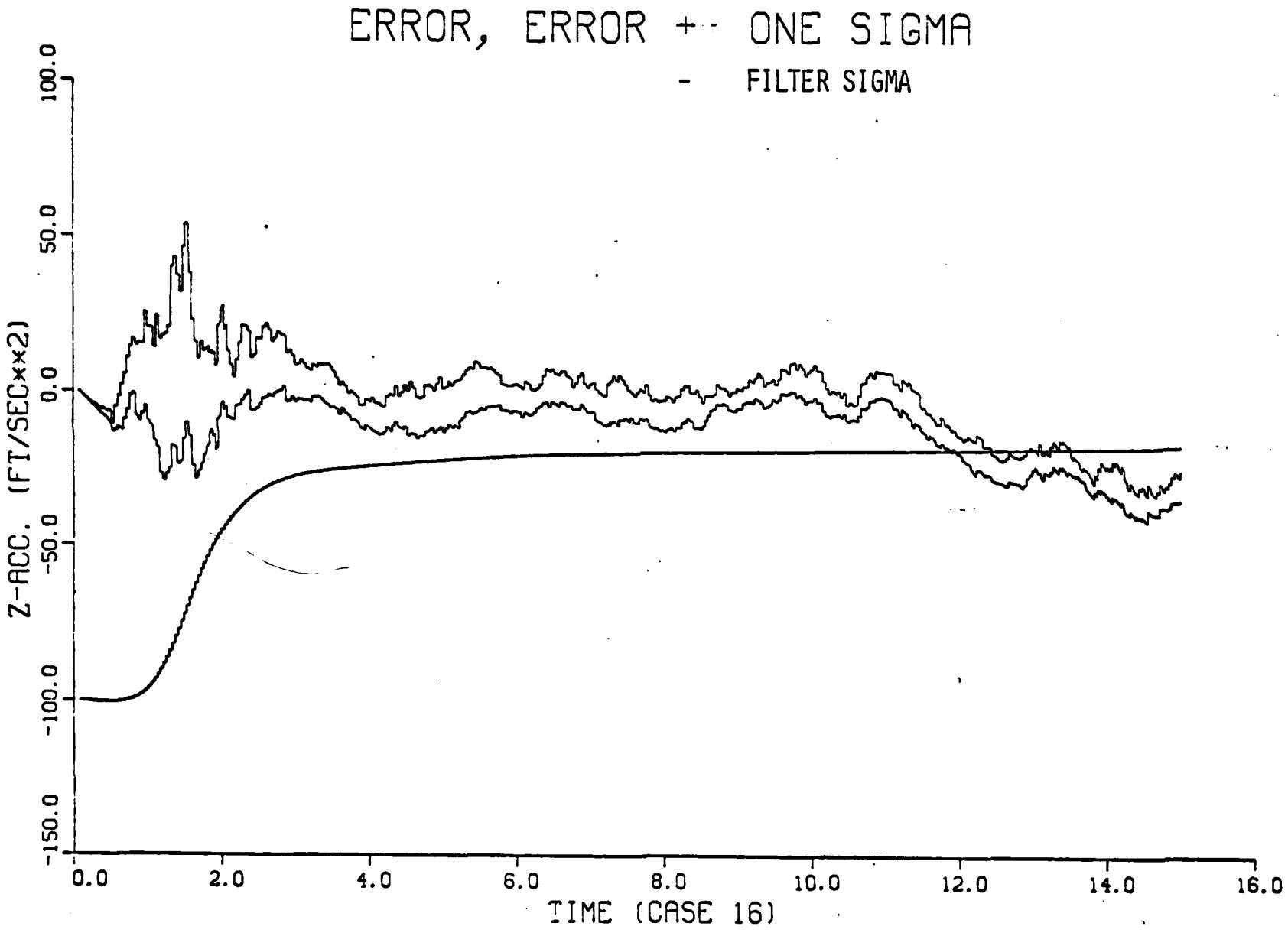
2.1.2.2

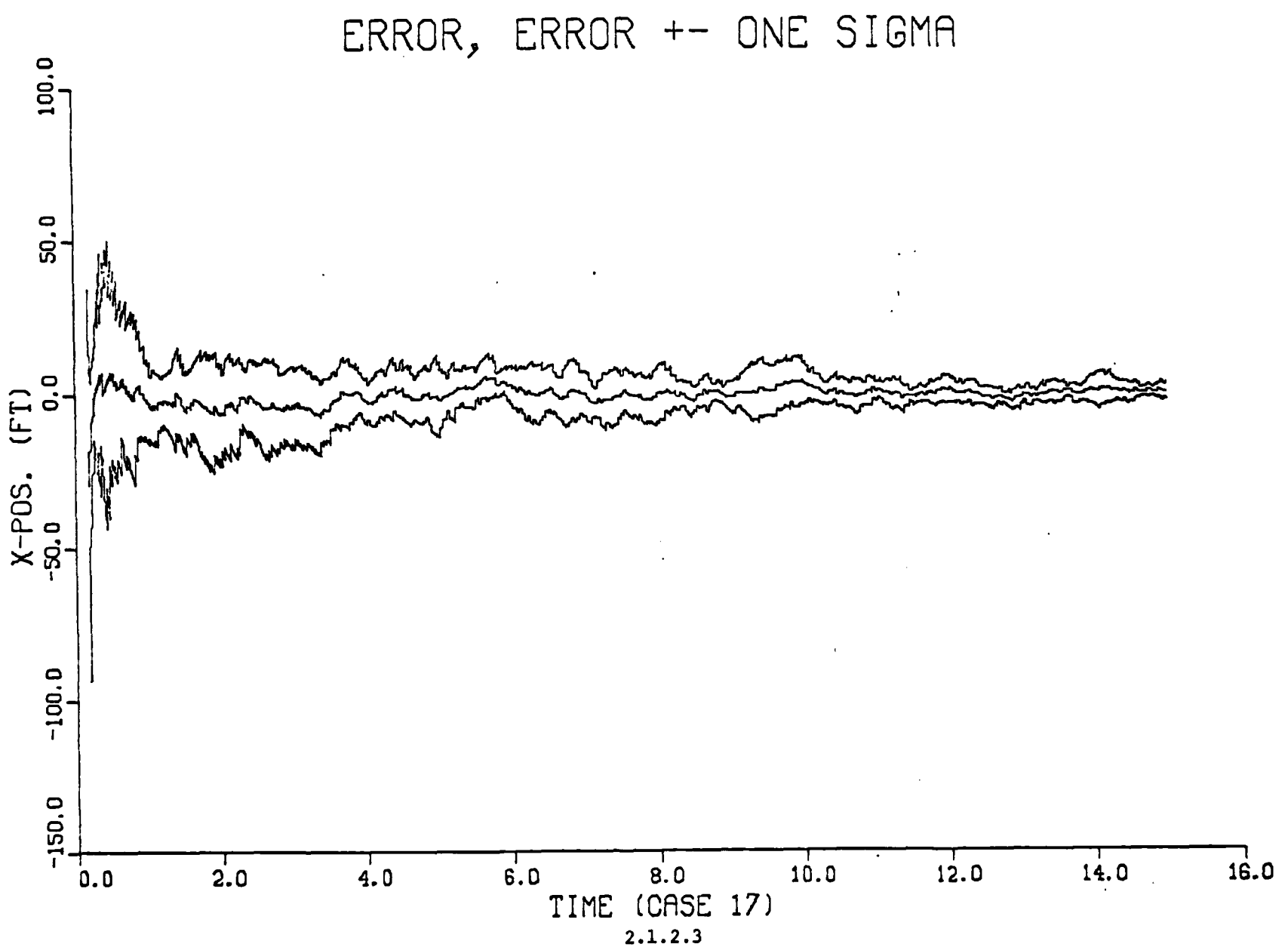

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

2.1.2.2


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Z-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

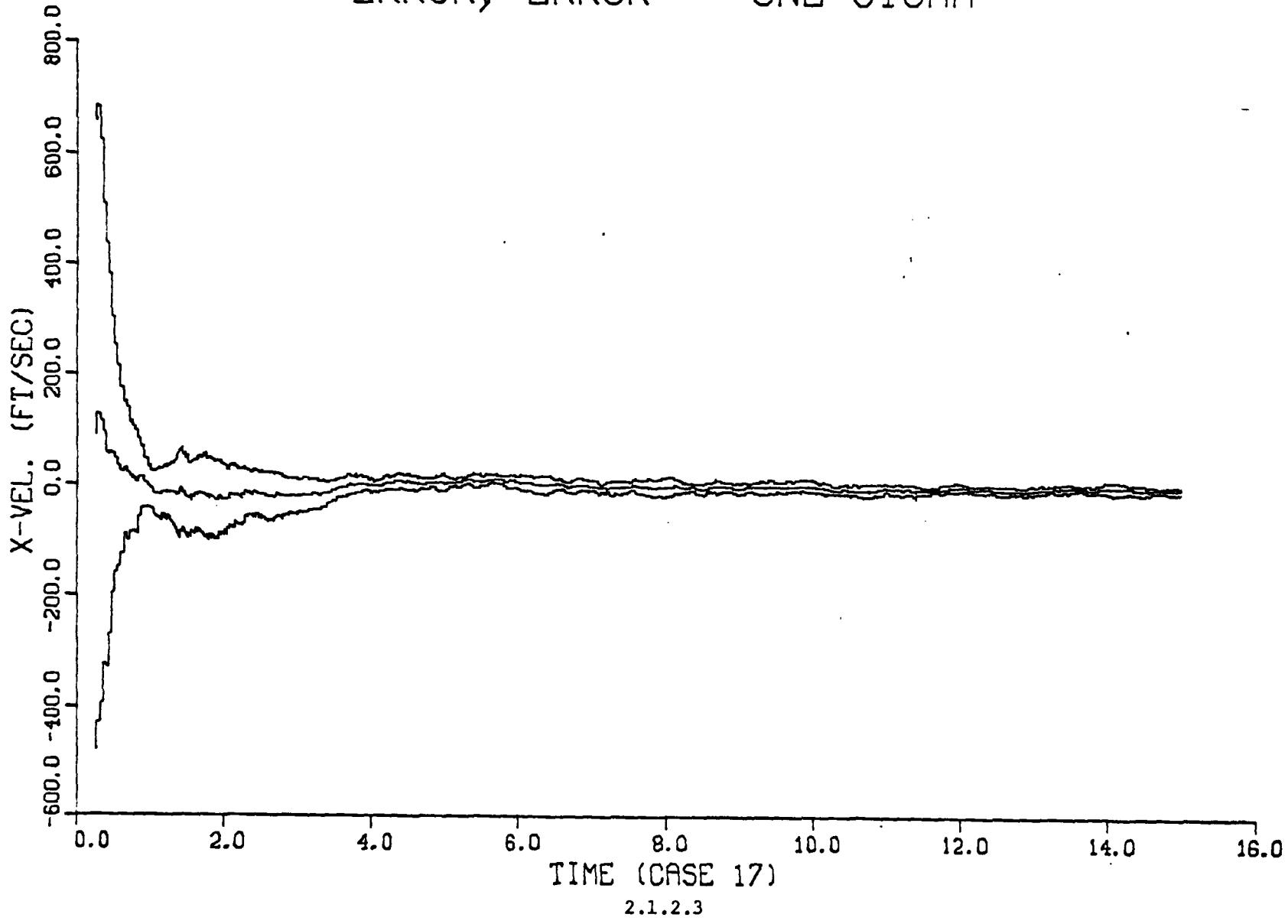

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Z-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)
2.1.2.2


ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
Z-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 16)

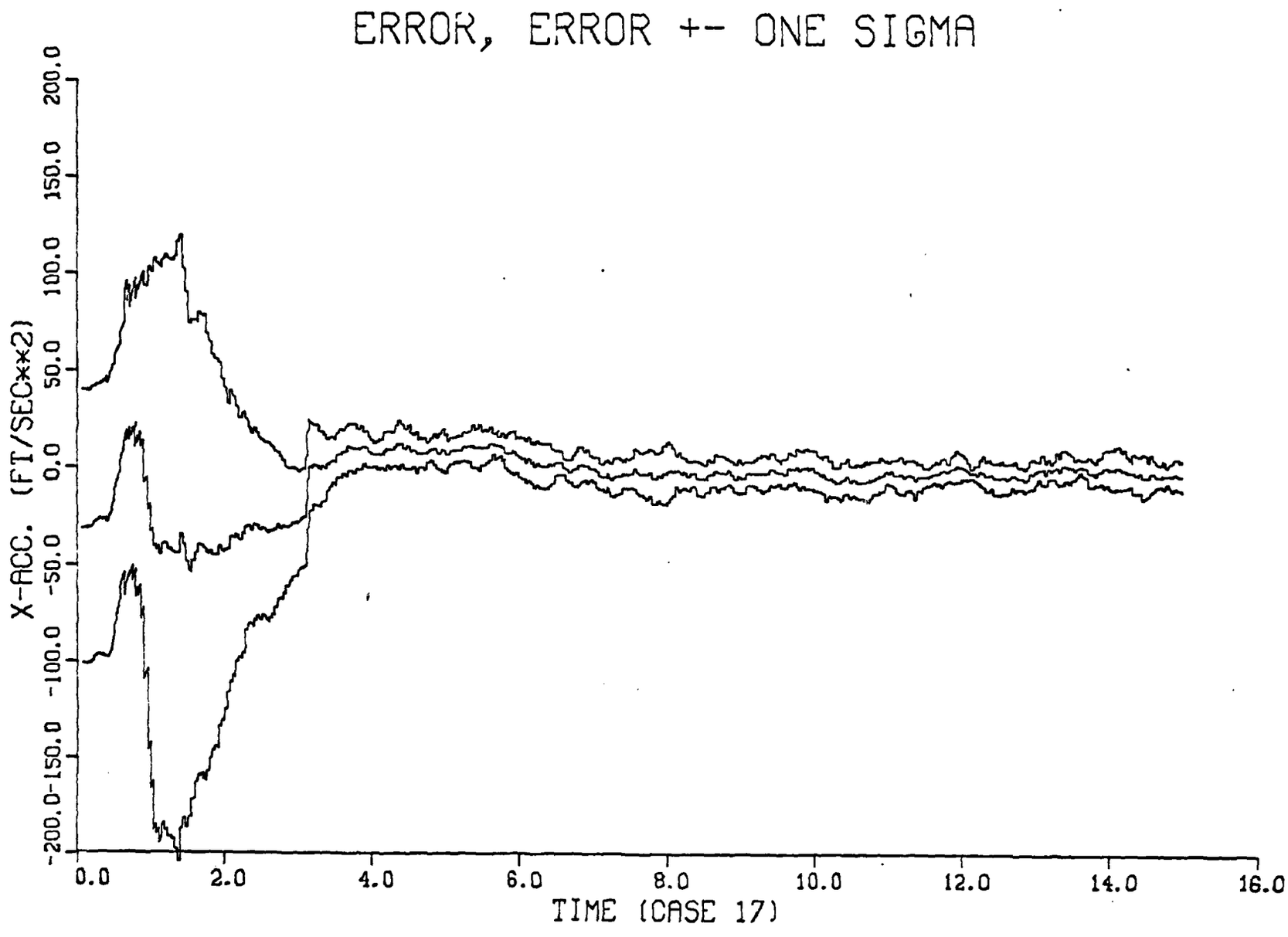

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 17)

2.1.2.3

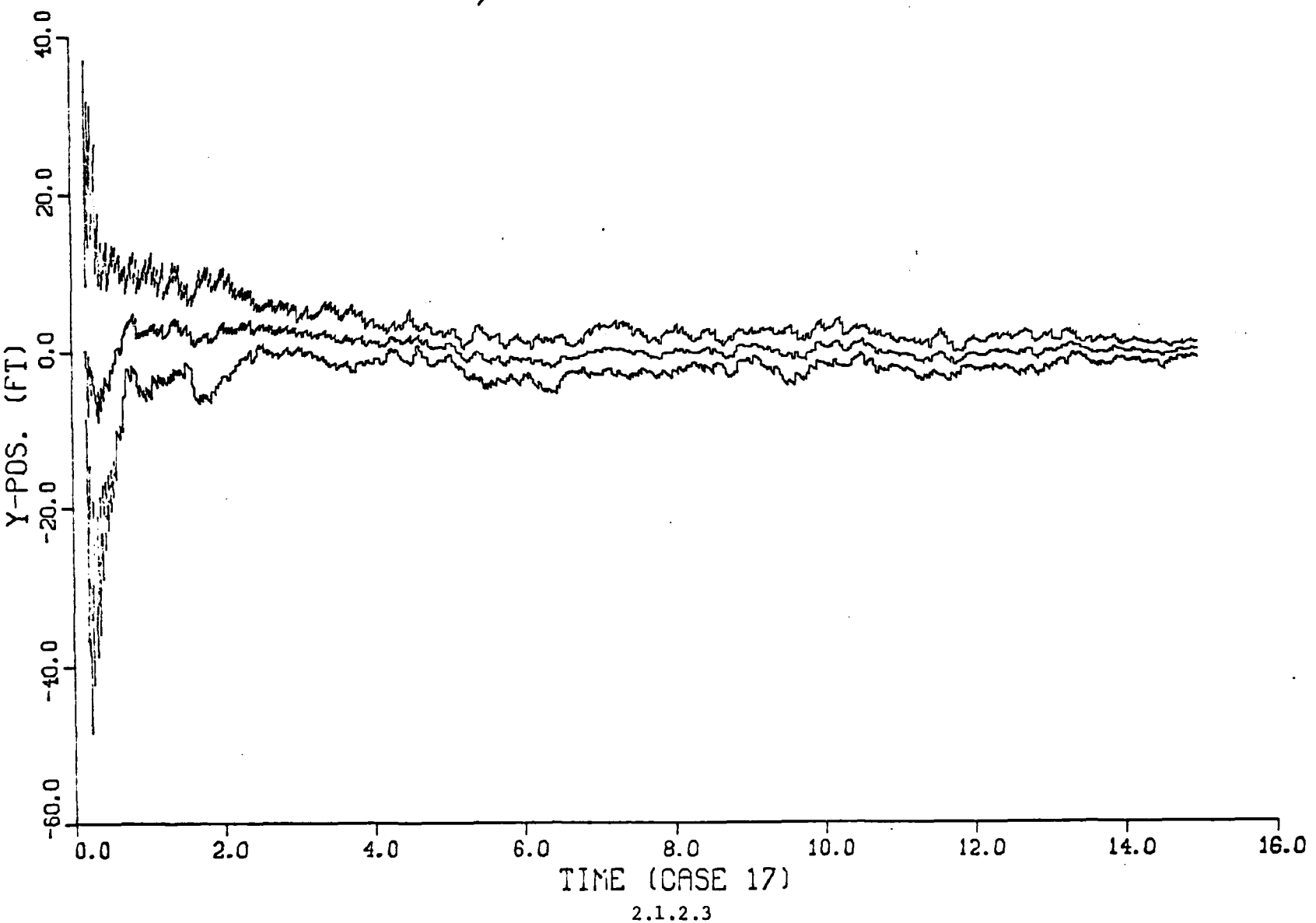
ERROR, ERROR +- ONE SIGMA

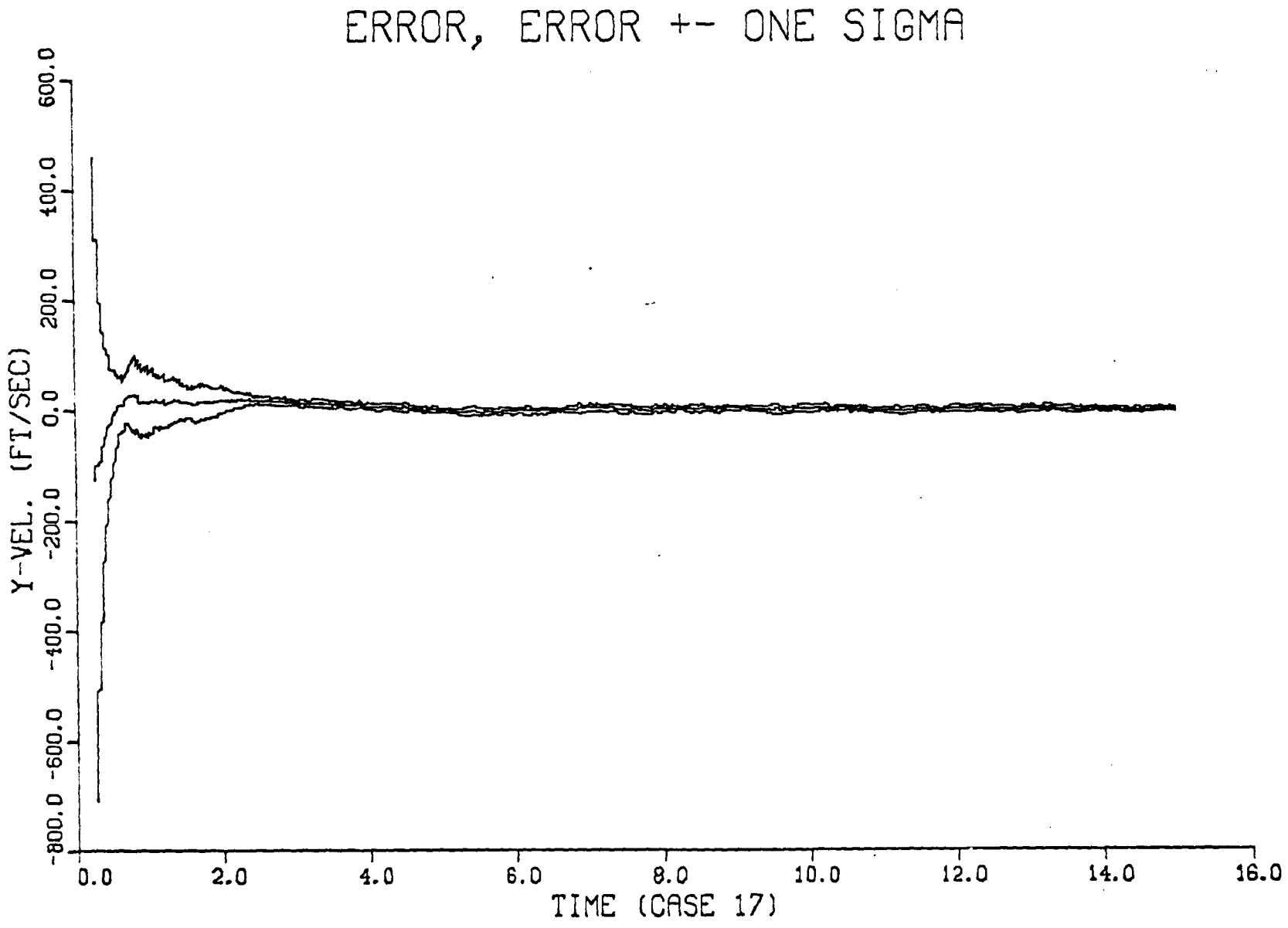


2.1.2.3

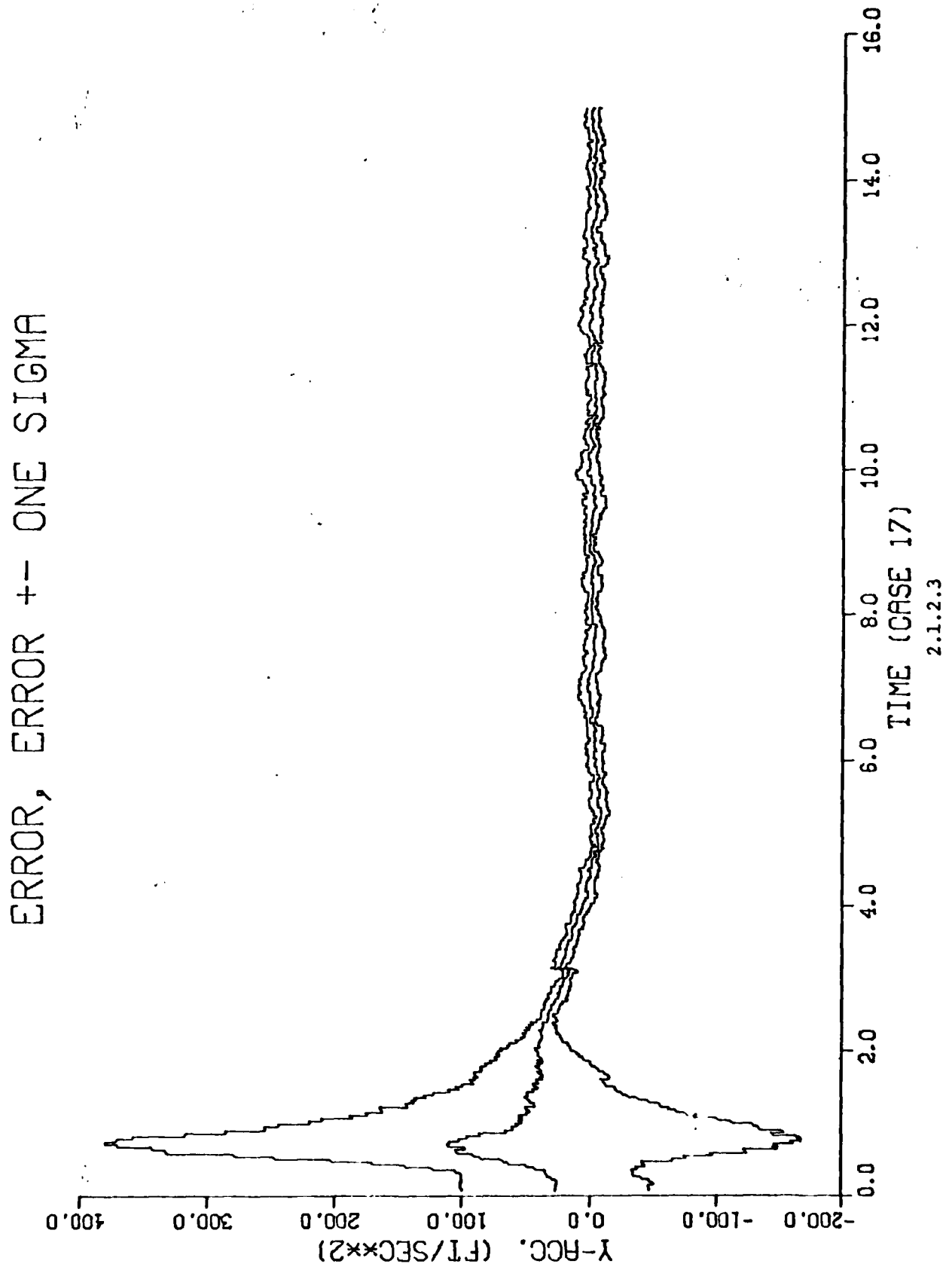

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 17)

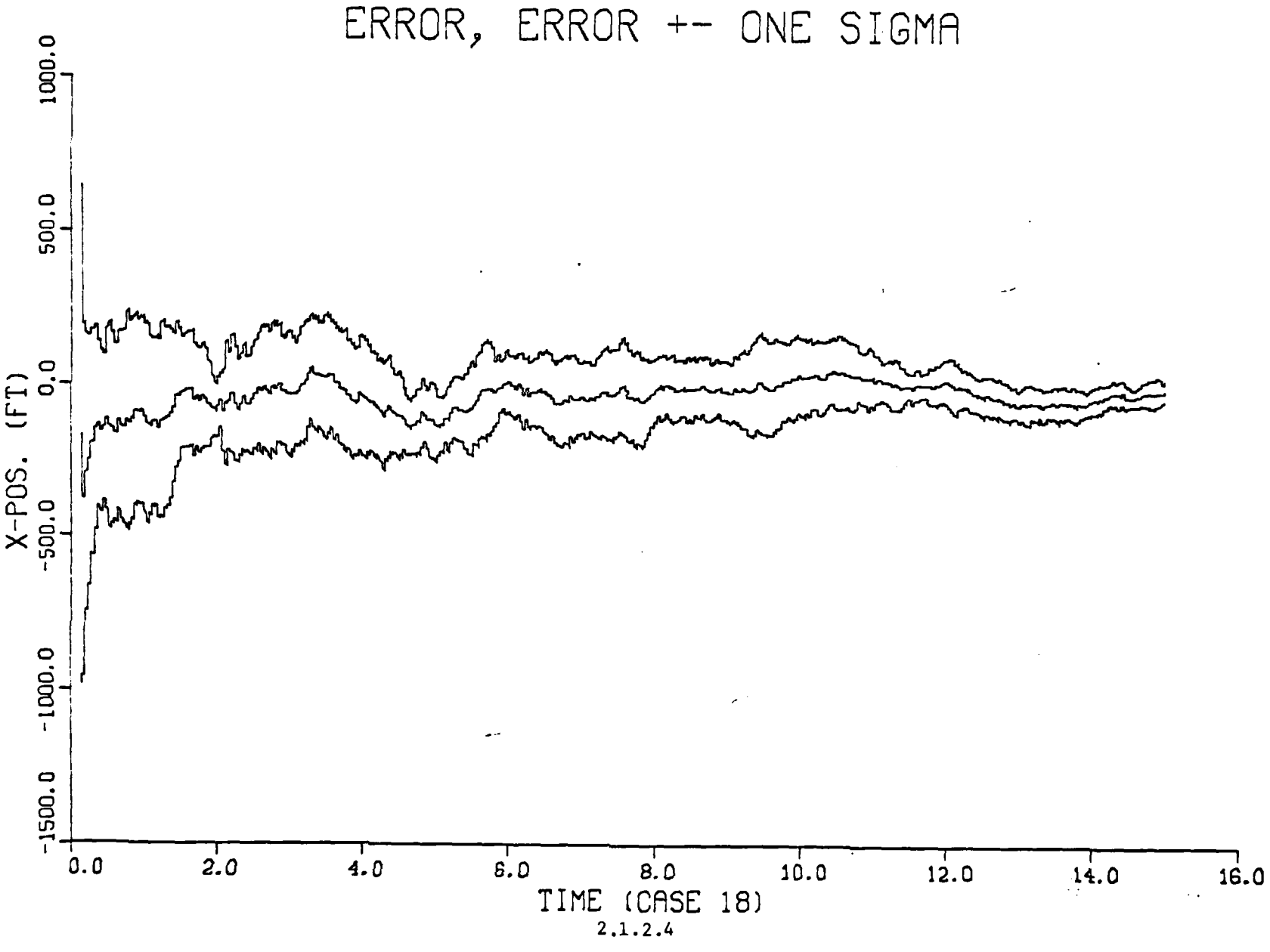
2.1.2.3

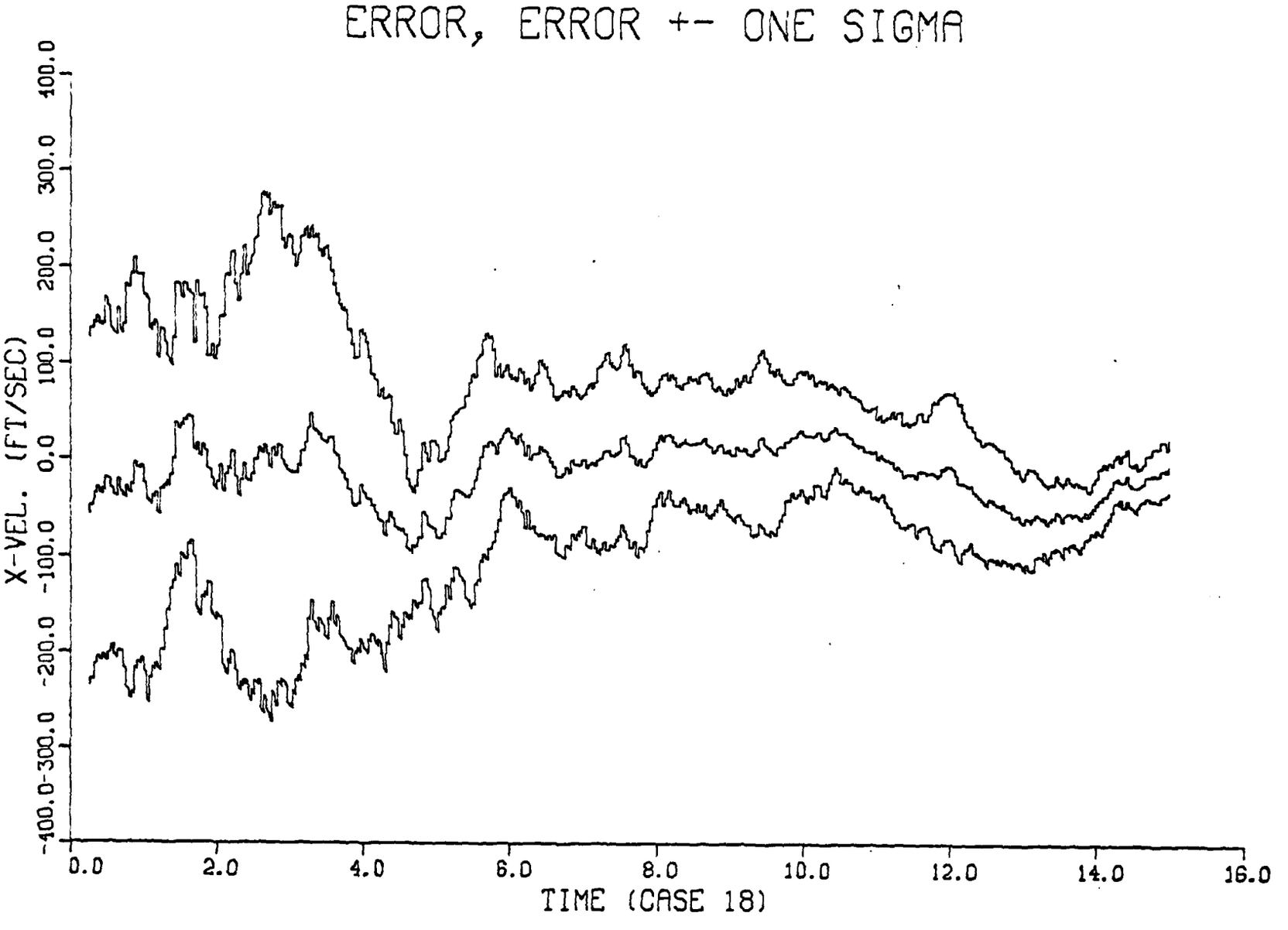

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
40.0
20.0
0.0
-20.0
-40.0
-60.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 17)
2.1.2.3


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 17)

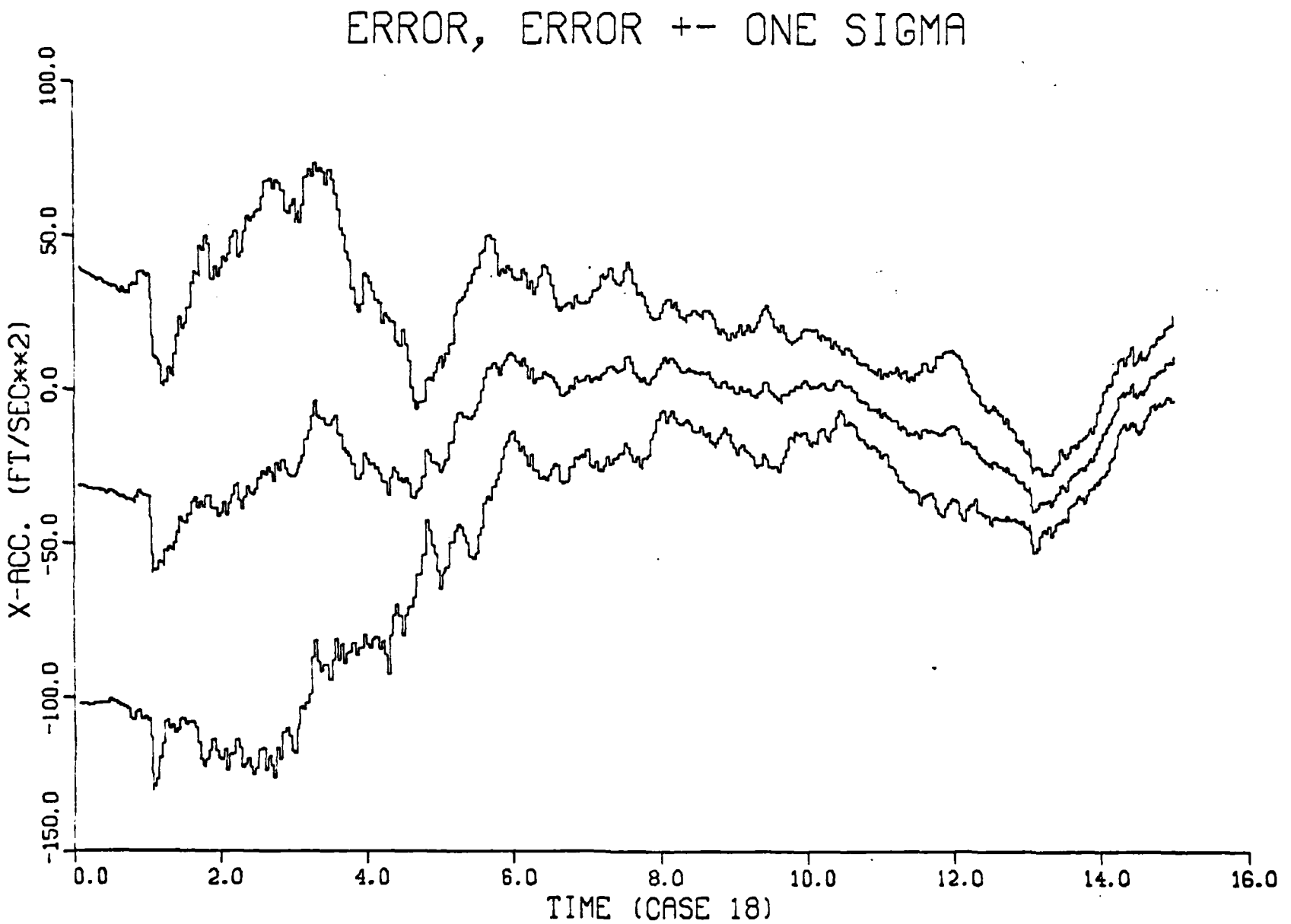
2.1.2.3


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
-200.0
-100.0
0.0
100.0
200.0
300.0
400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 17)
2.1.2.3

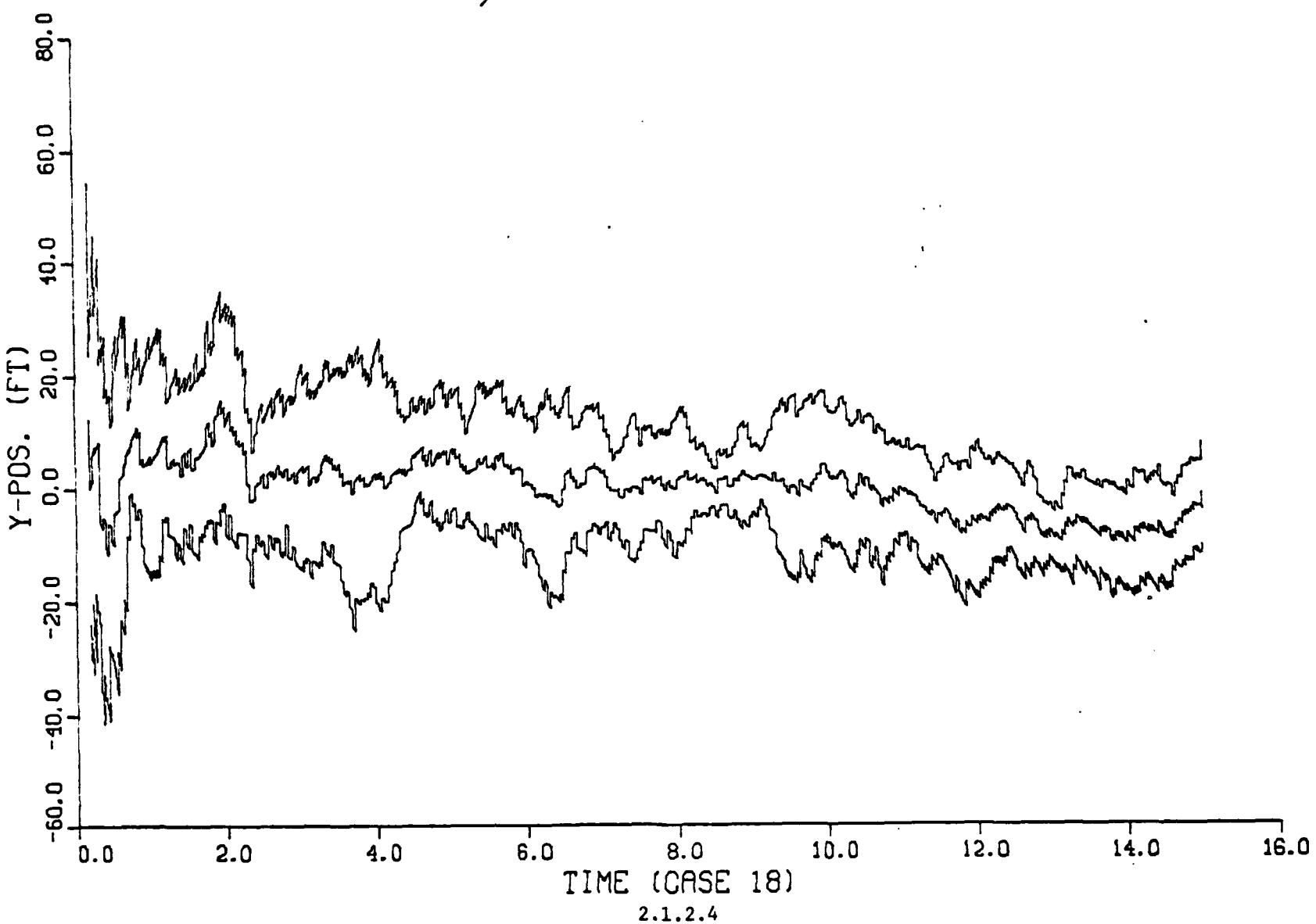

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)
2.1.2.4

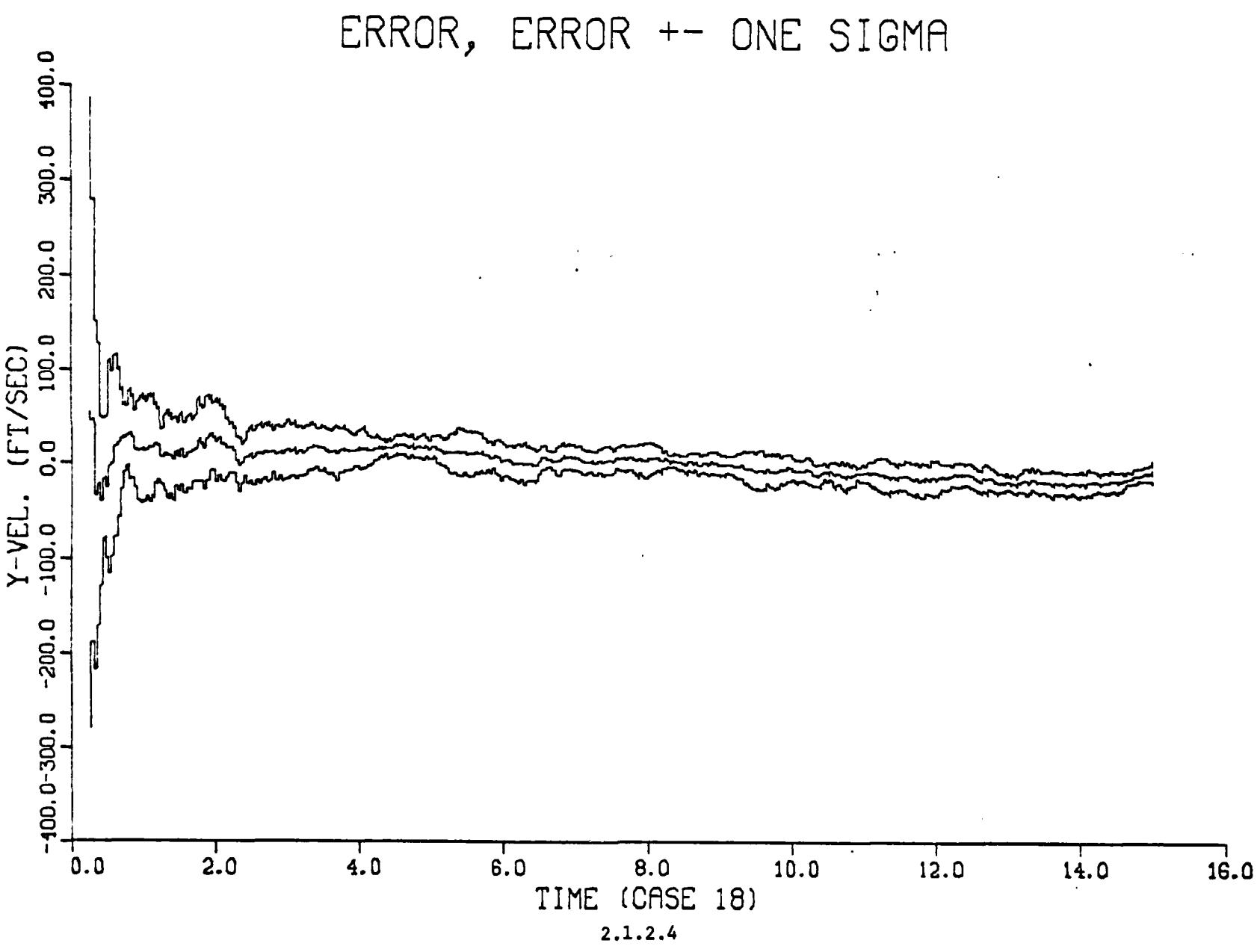

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
100.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)

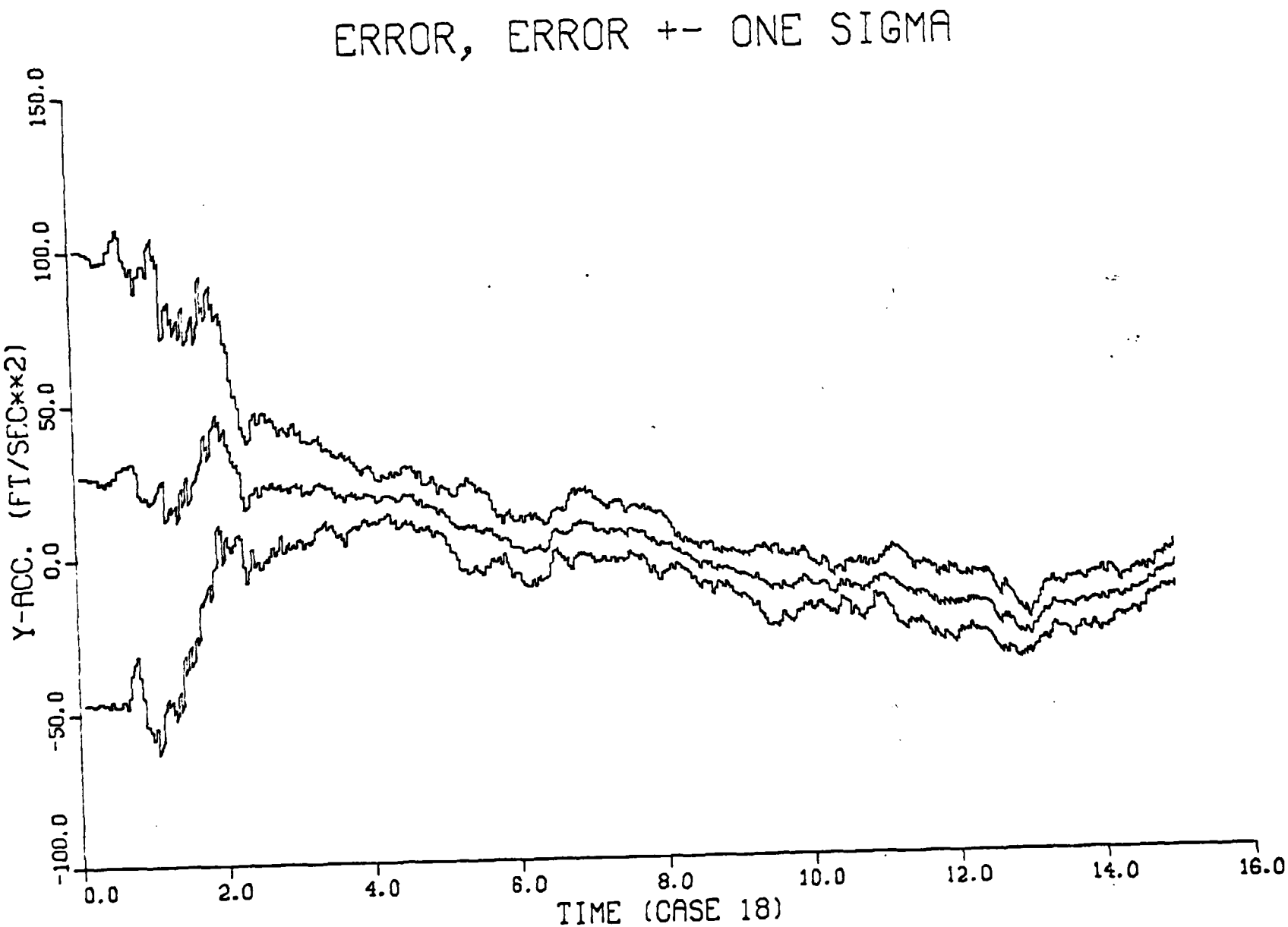
2.1.2.4


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)

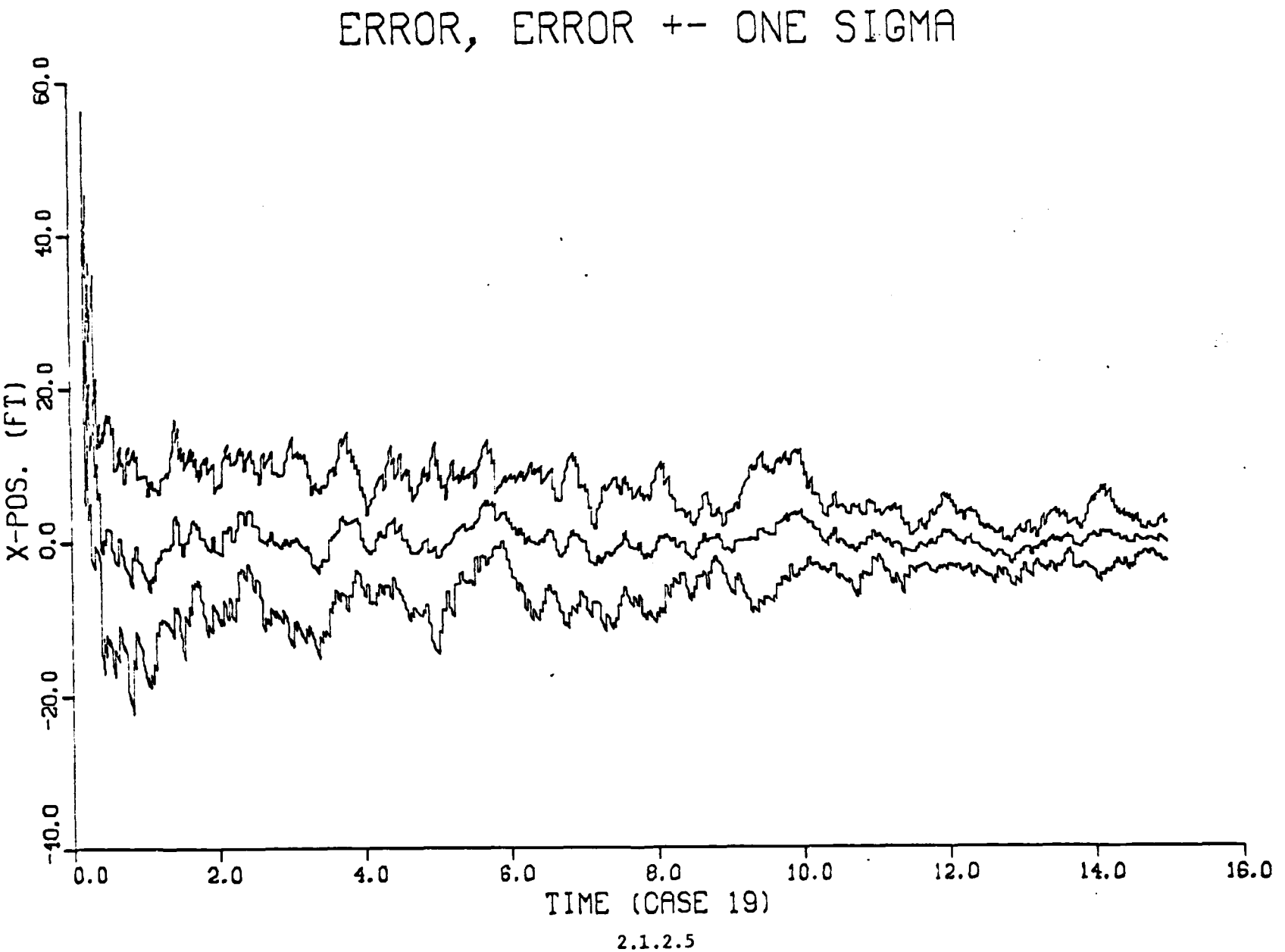
2.1.2.4


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)
2.1.2.4

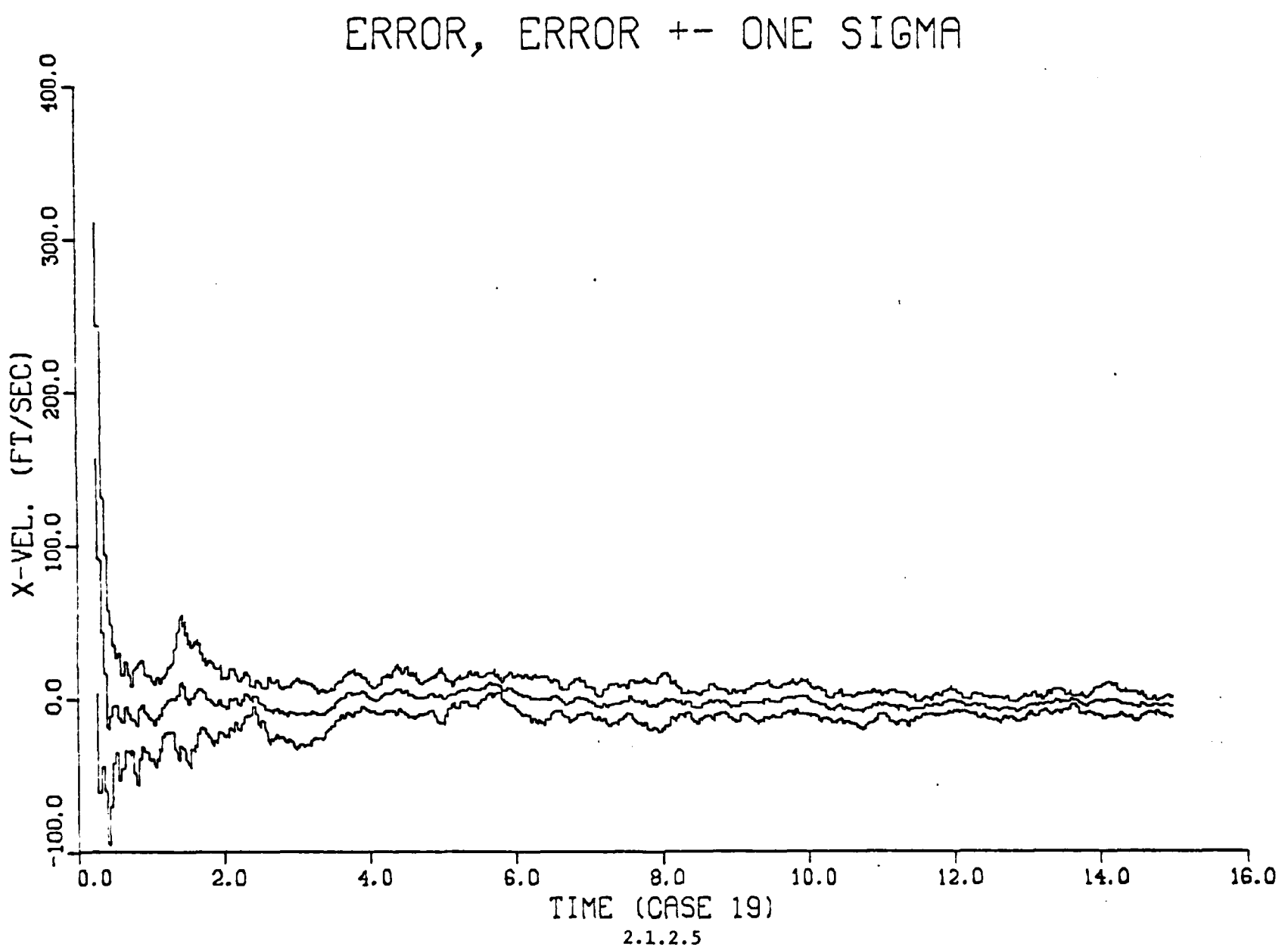

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)
2.1.2.4

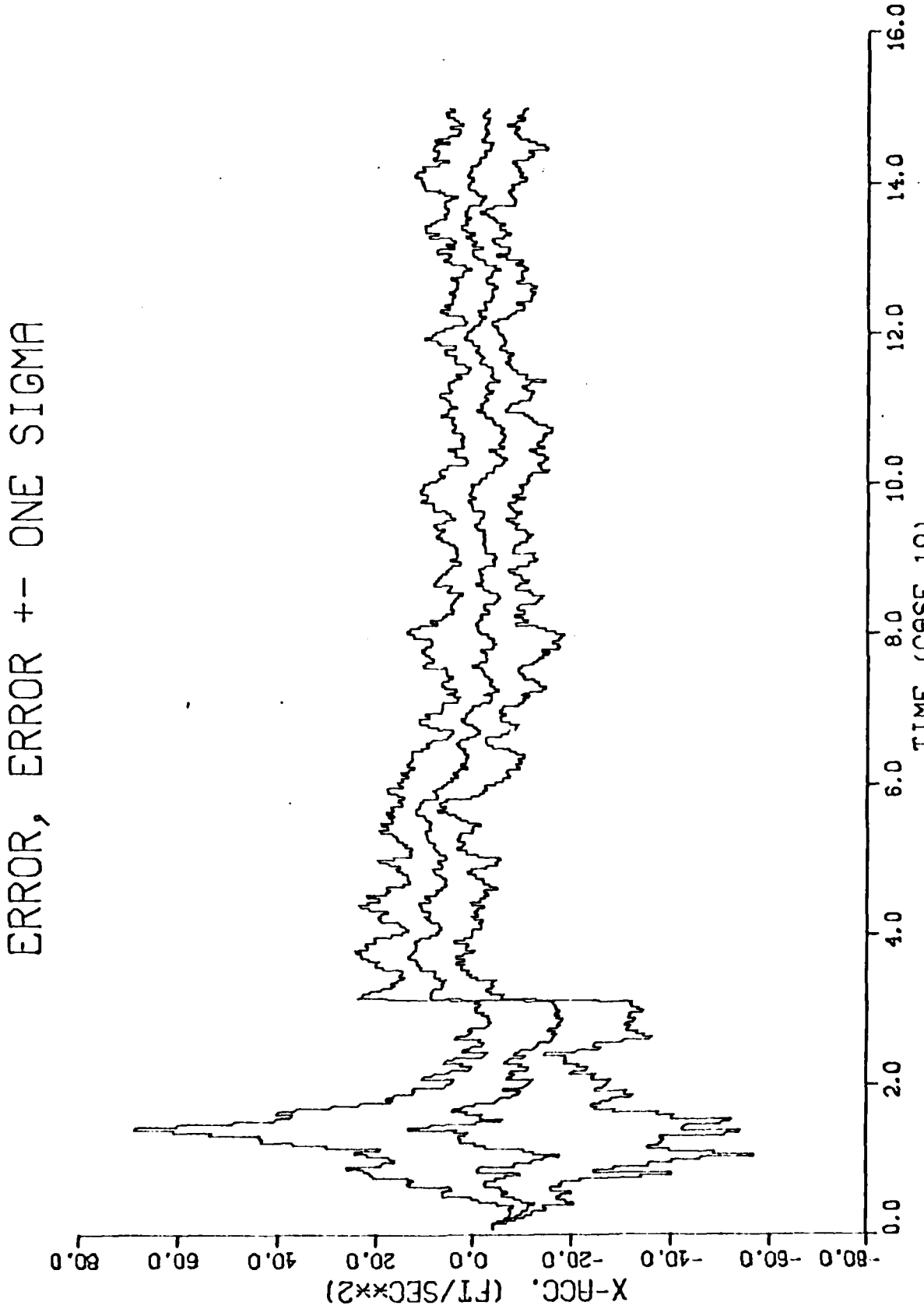

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 18)

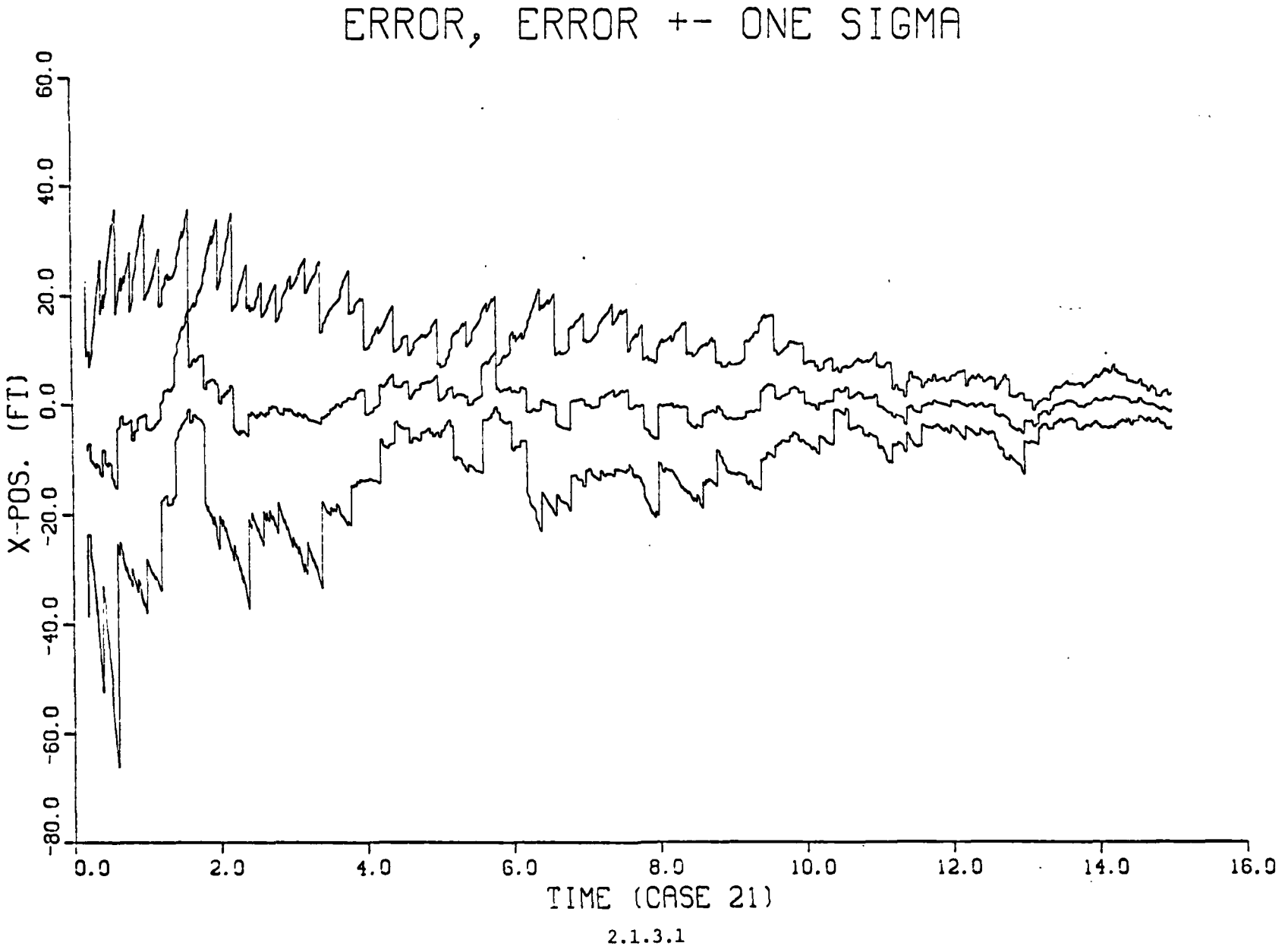
2.1.2.4


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 19)

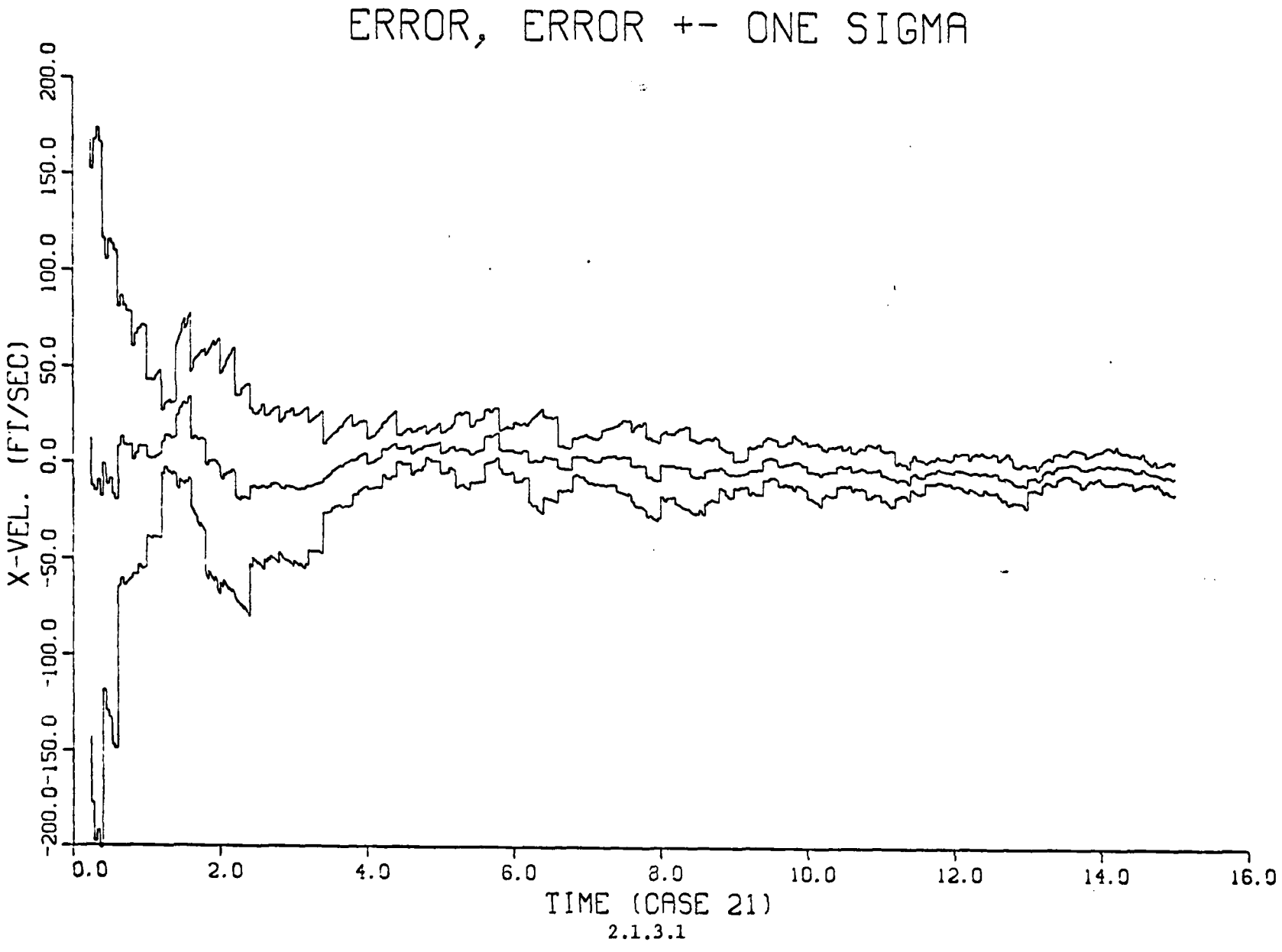
2.1.2.5

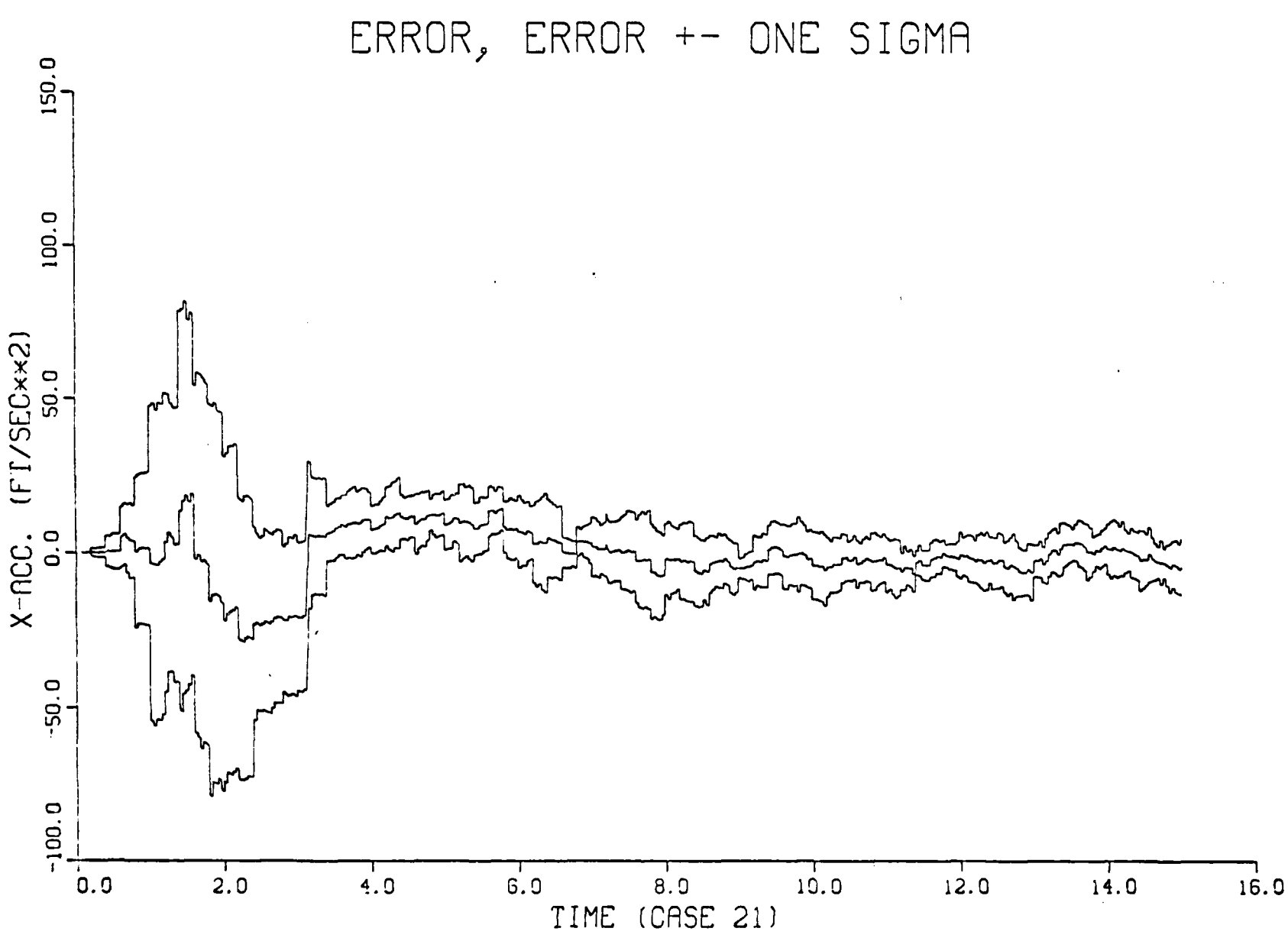

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
100.0
300.0
200.0
100.0
0.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 19)
2.1.2.5

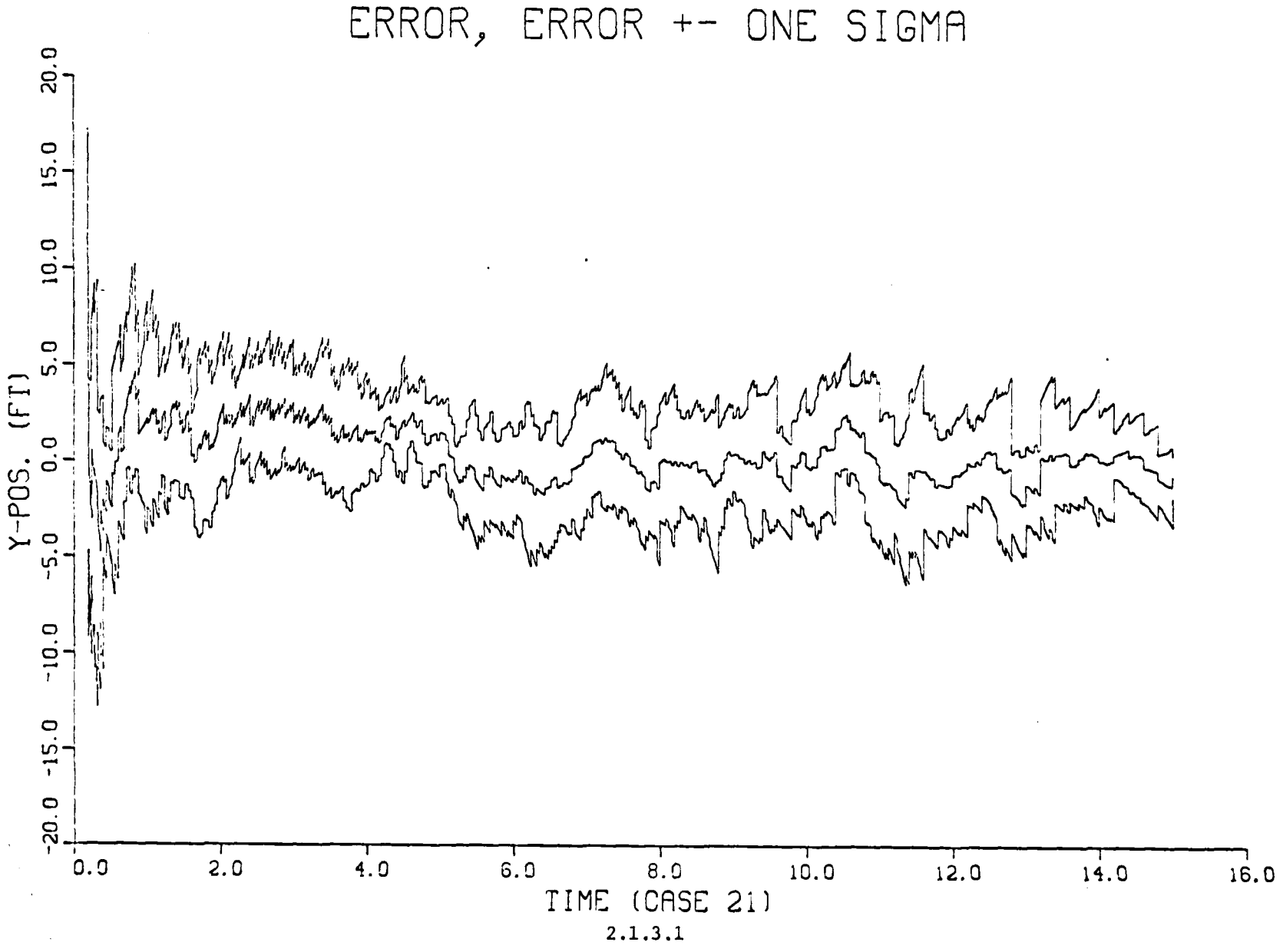

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 19)
2.1.2.5

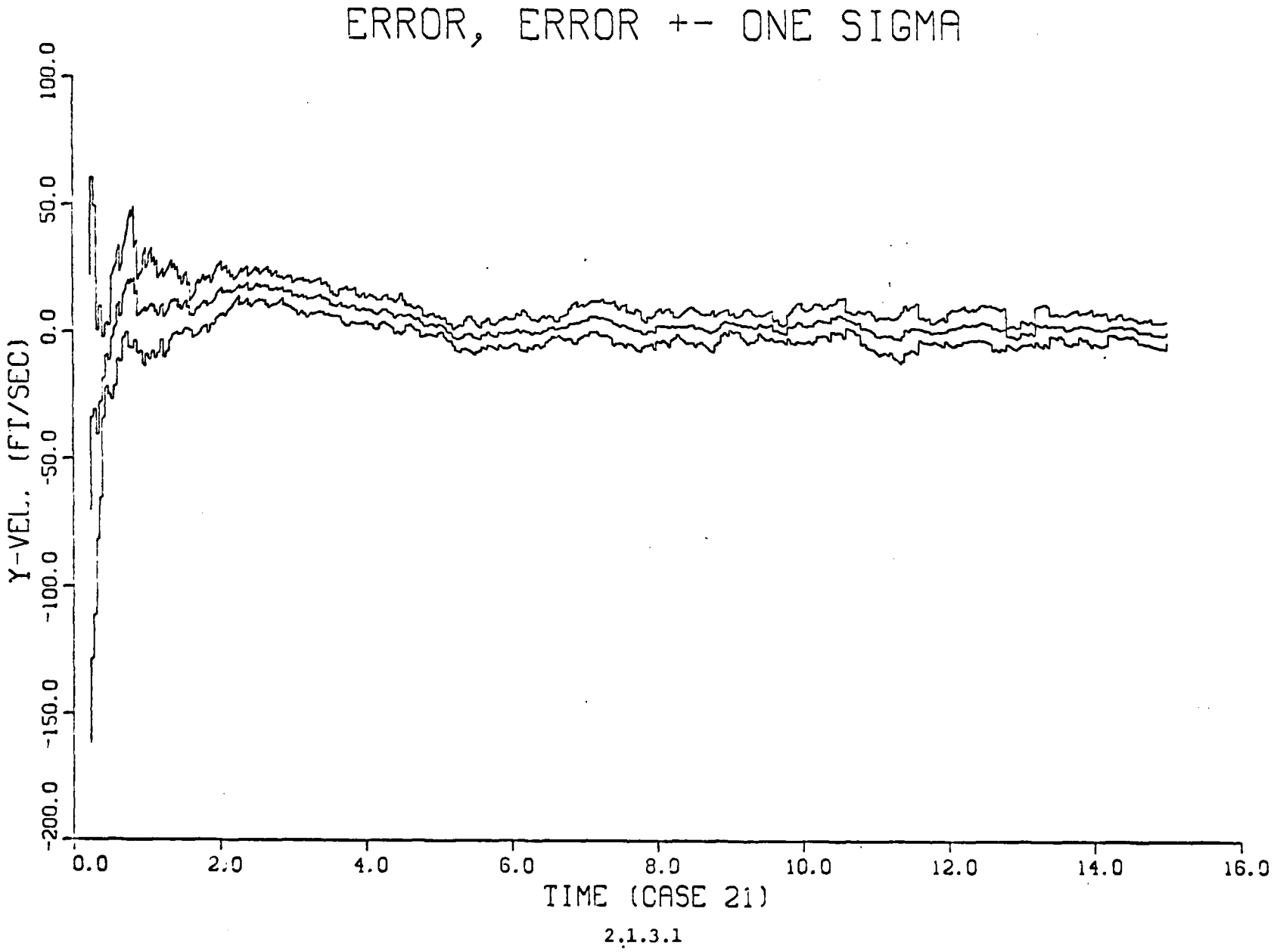

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)

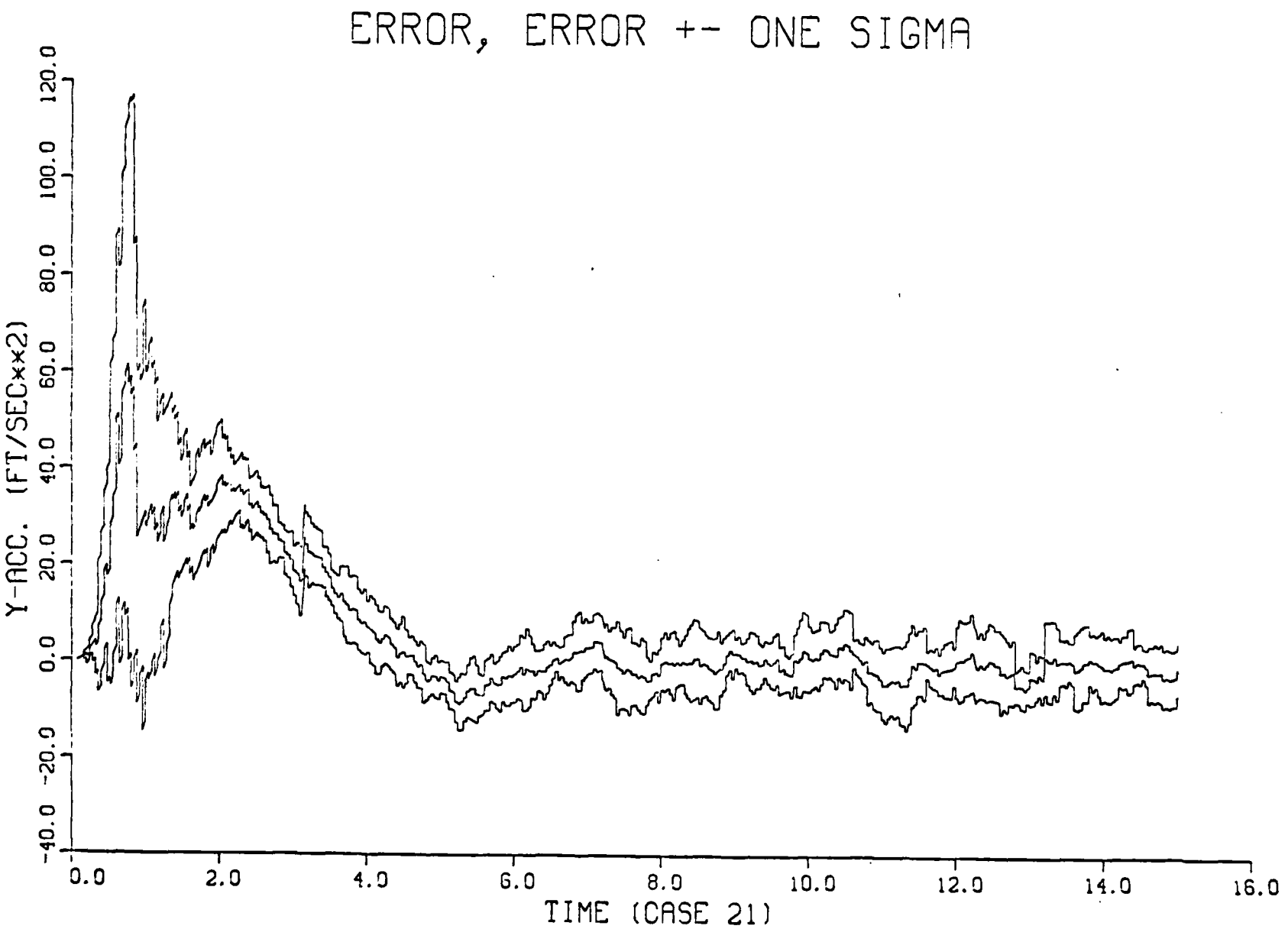
2.1.3.1


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)
2.1.3.1

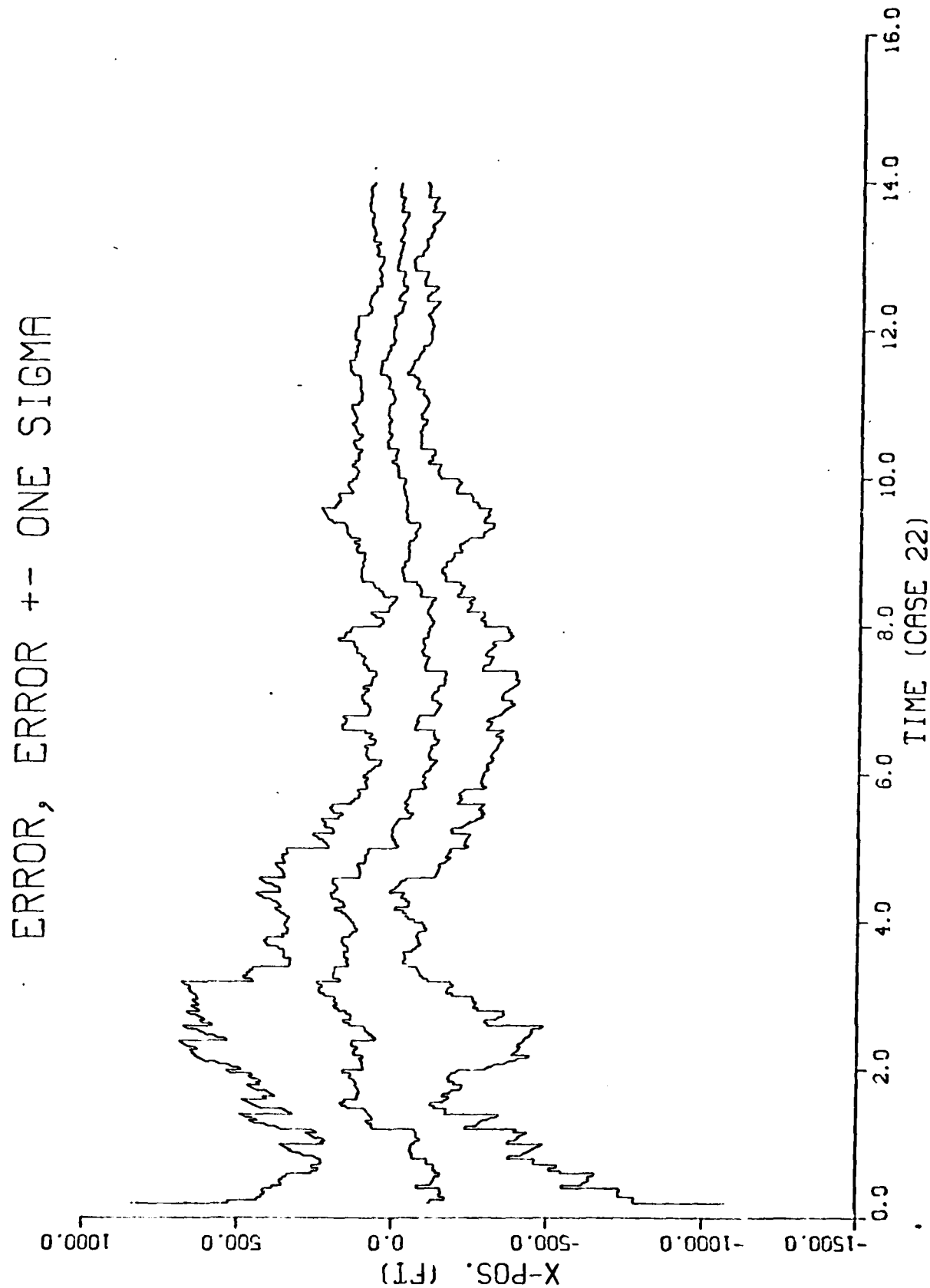

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
-15.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)
2.1.3.1

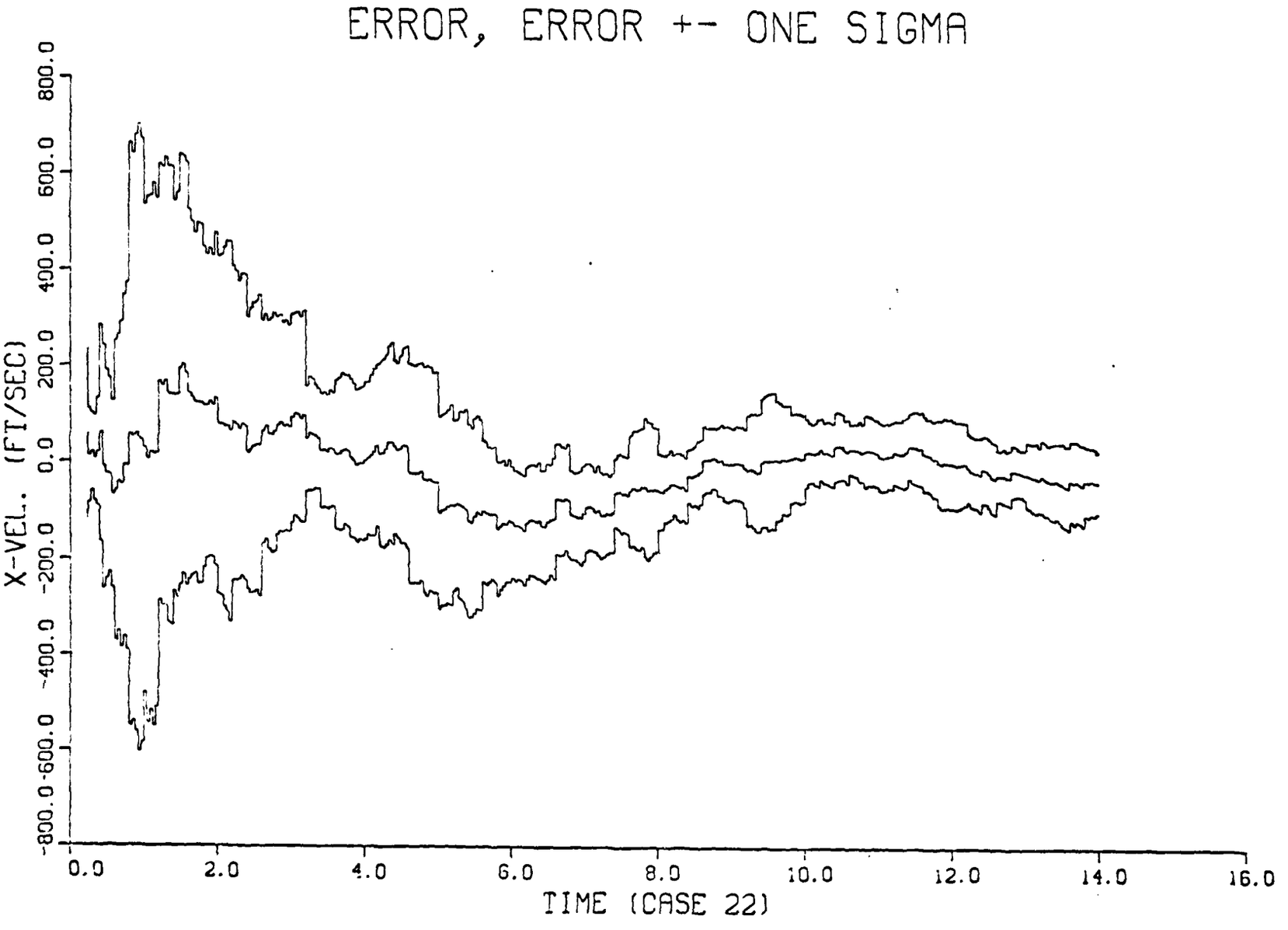

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)
2.1.3.1


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 21)

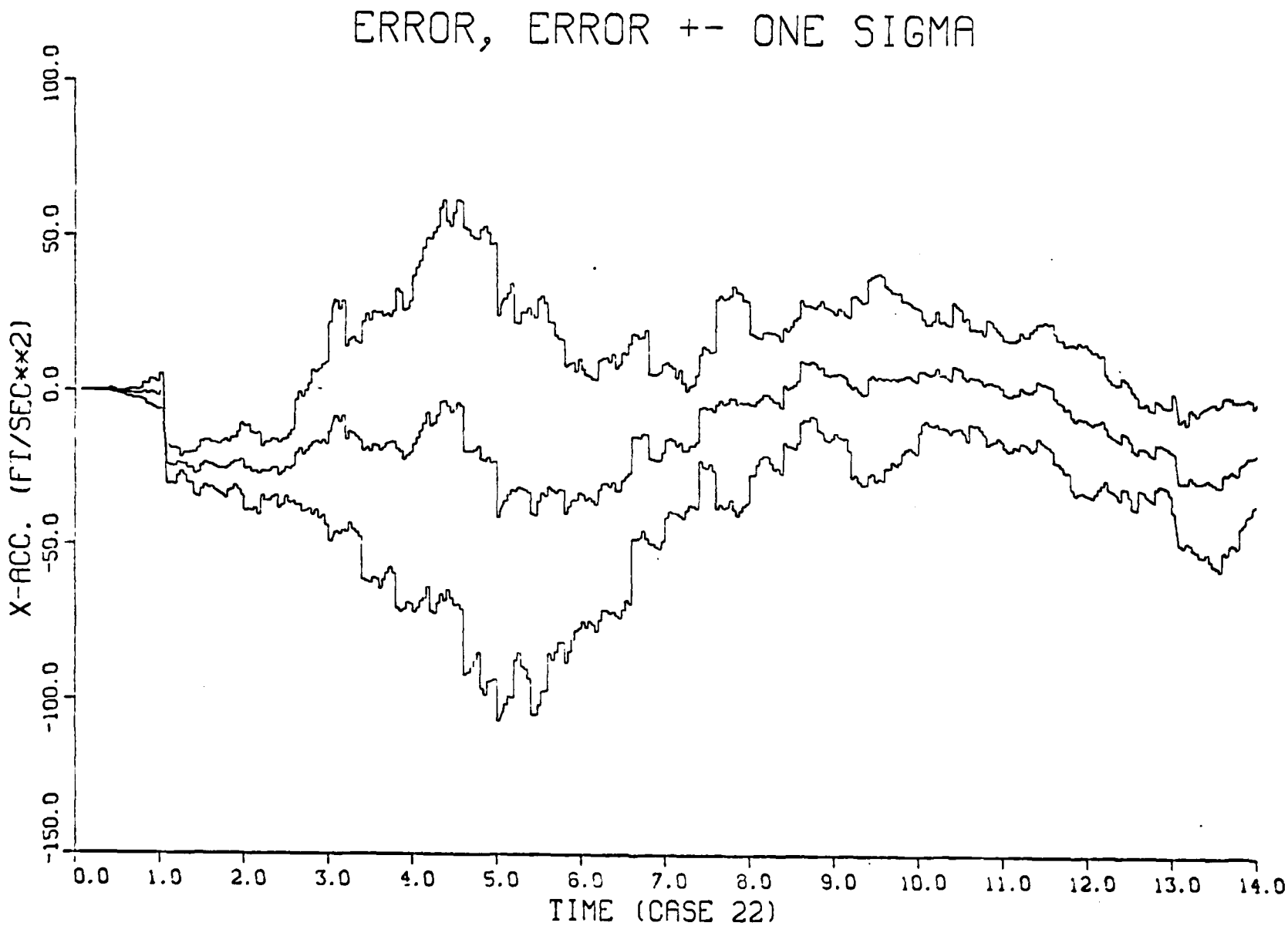
2.1.3.1


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 22)

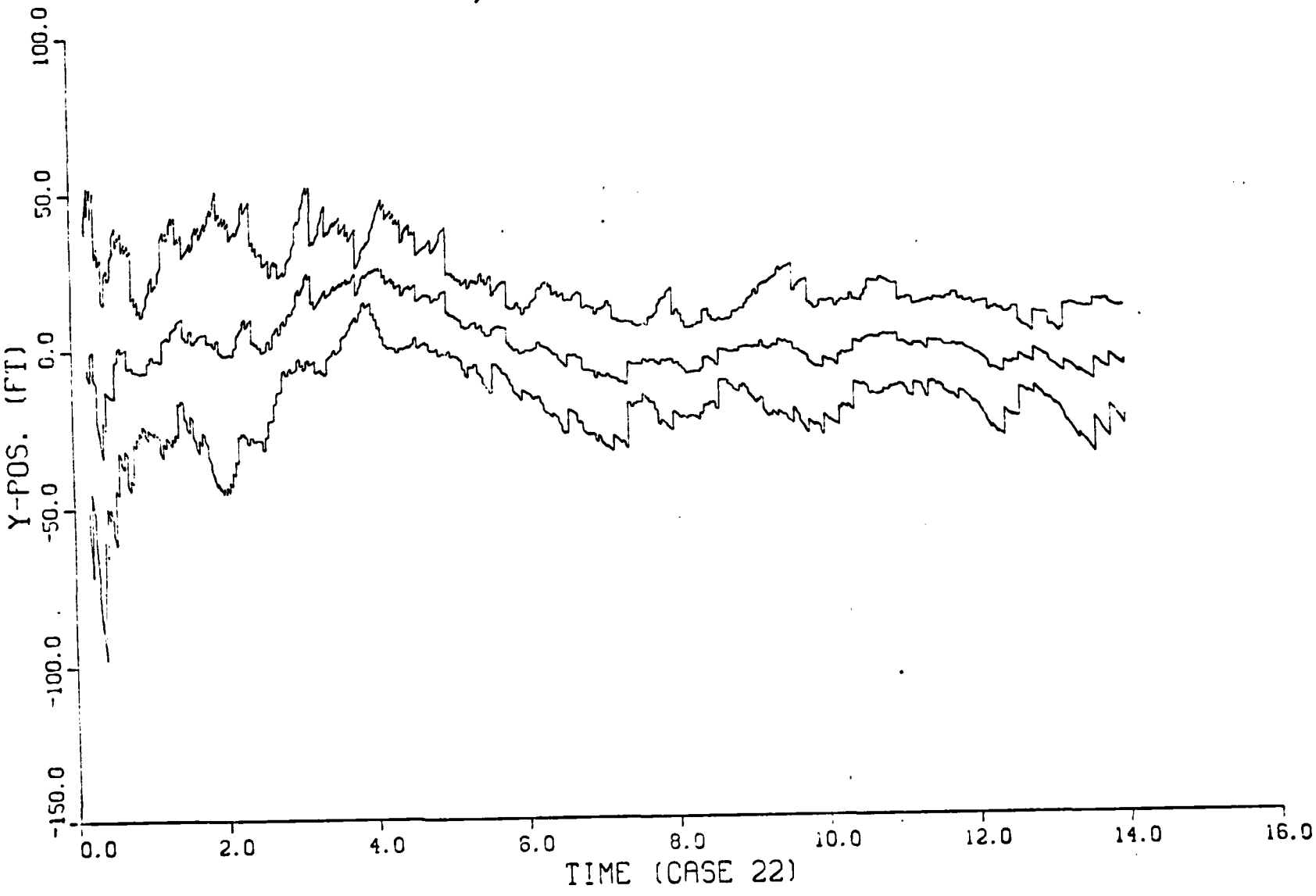
2.1.3.2


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
800.0
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 22)

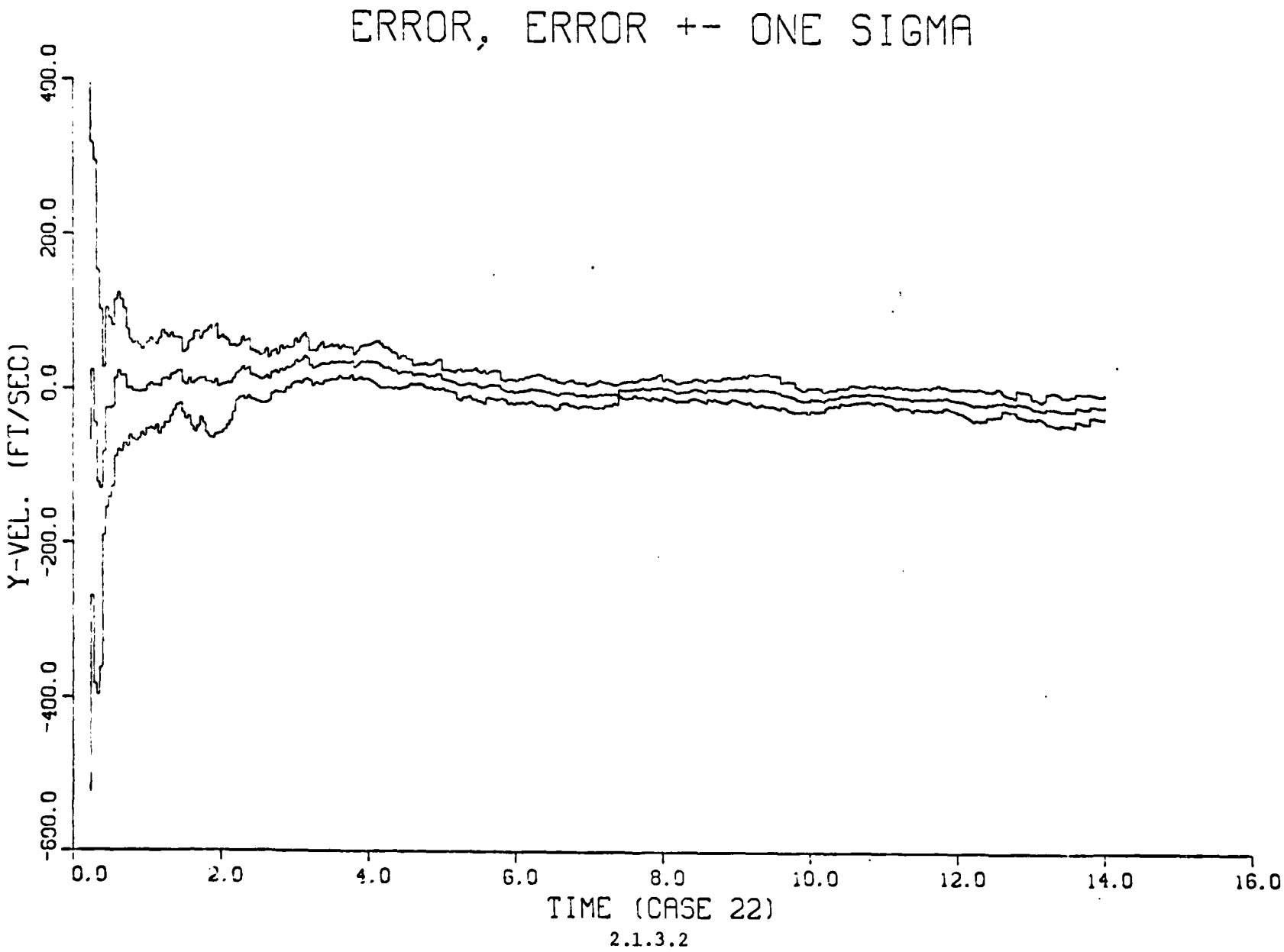
2.1.3.2

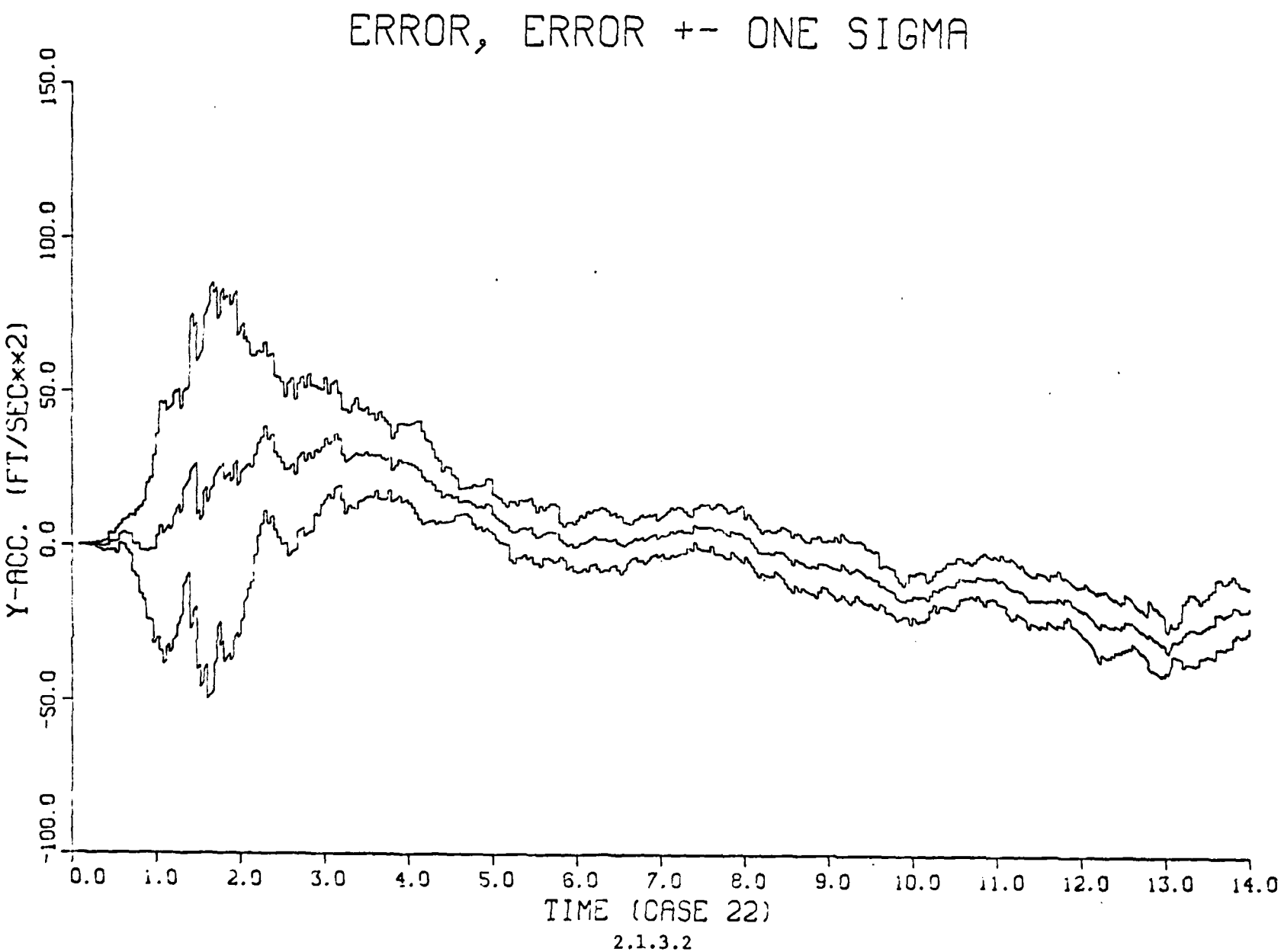

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 22)

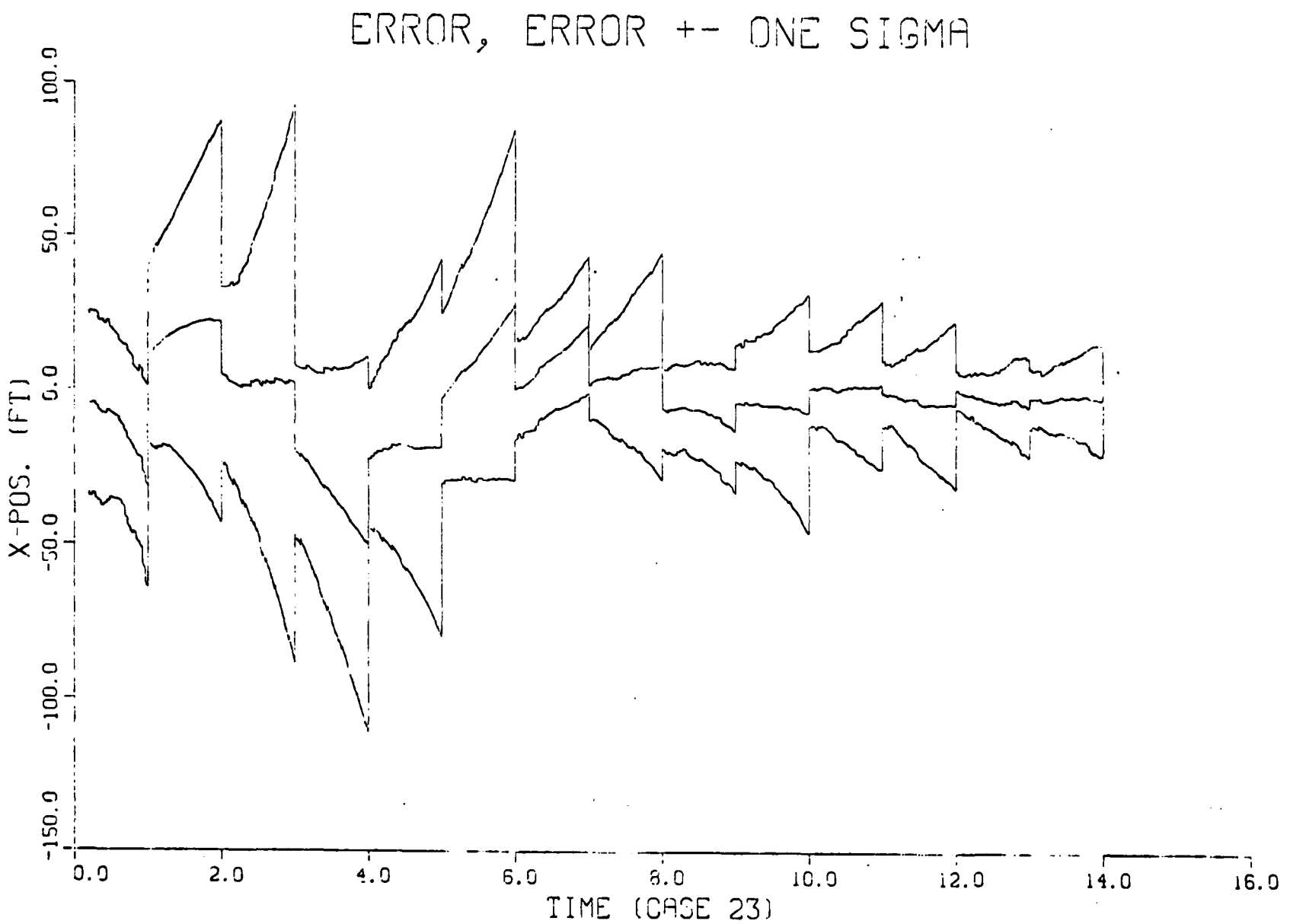
2.1.3.2


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 22)

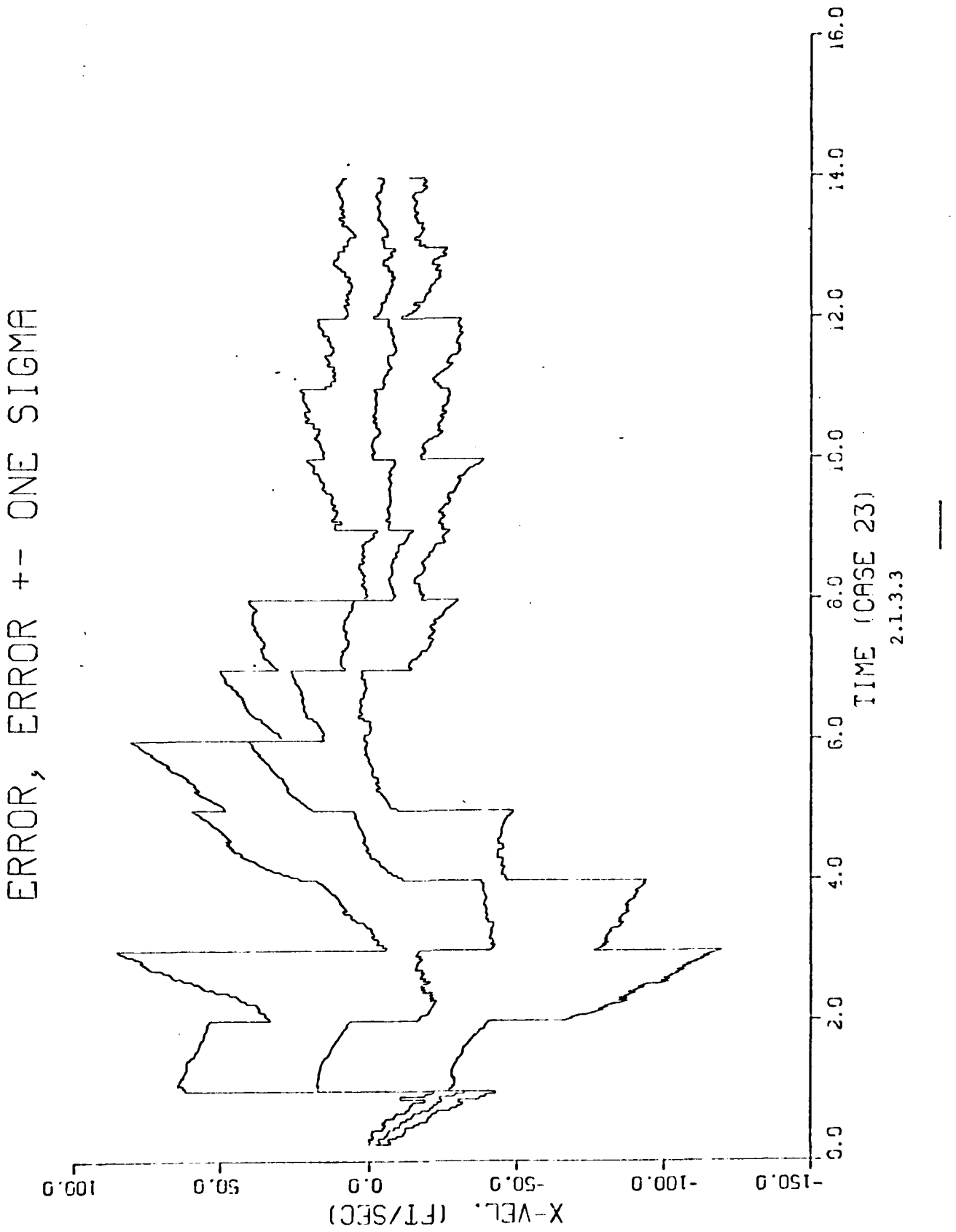
2.1.3.2

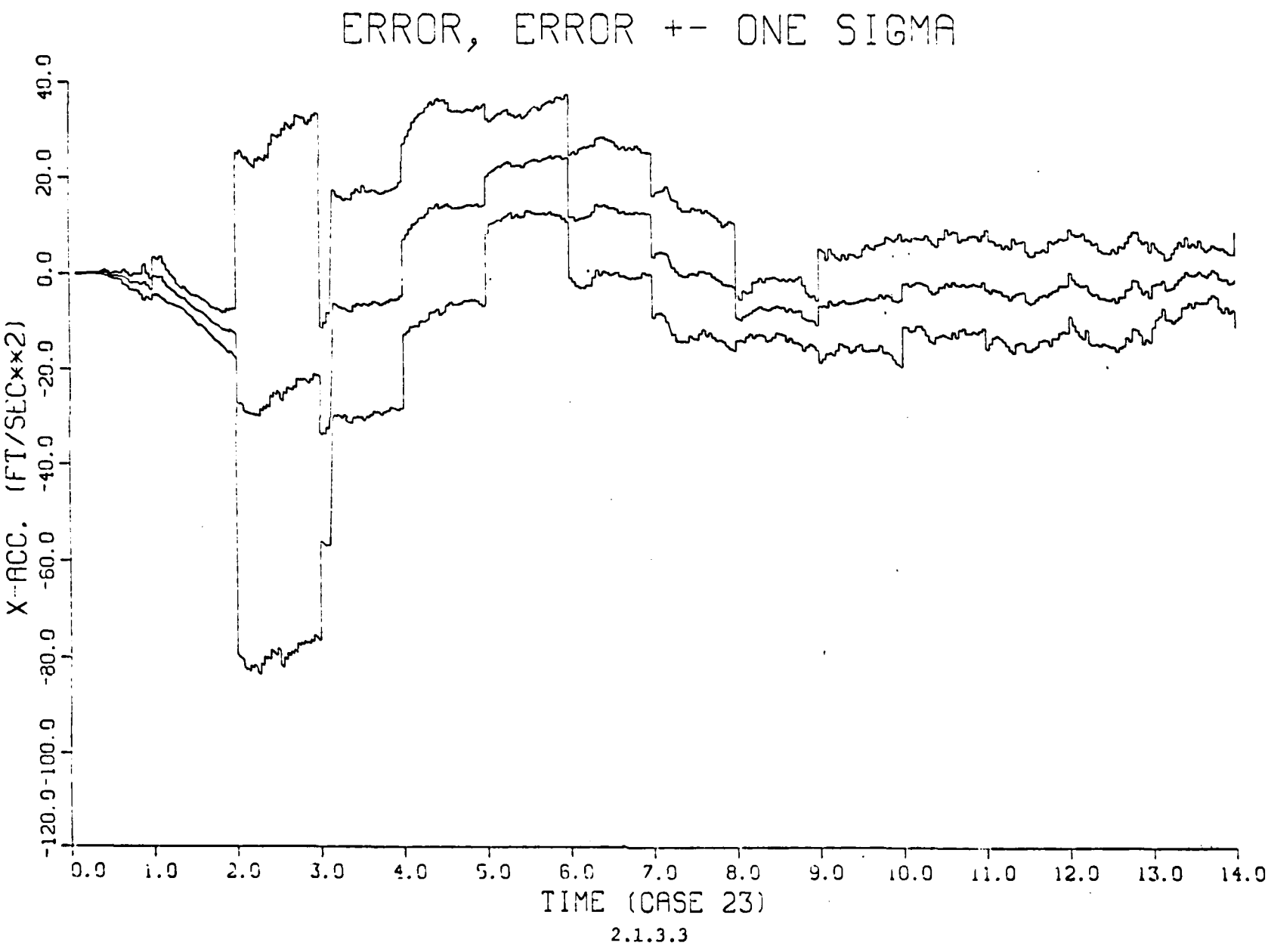

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 22)
2.1.3.2


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 22)
2.1.3.2

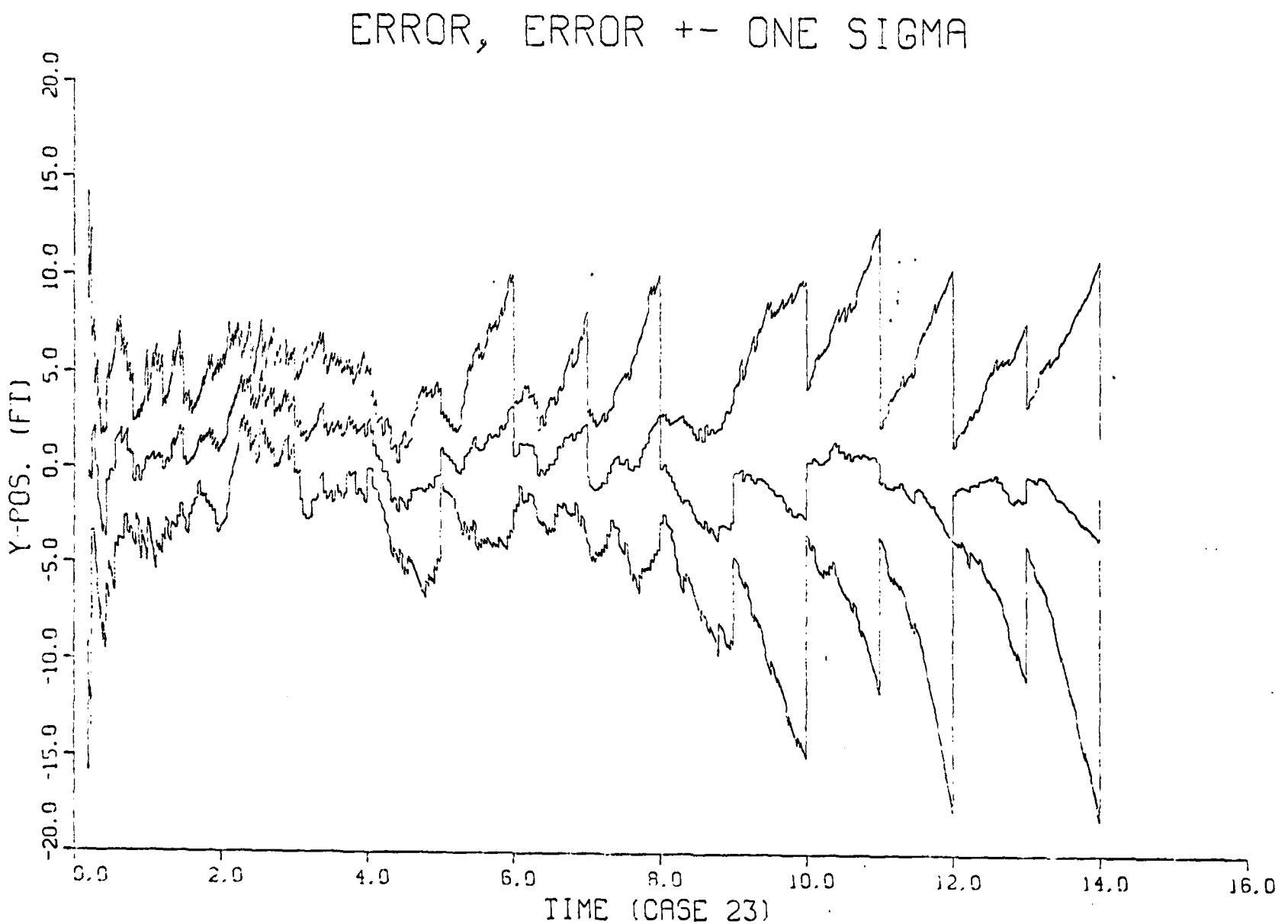

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 23)

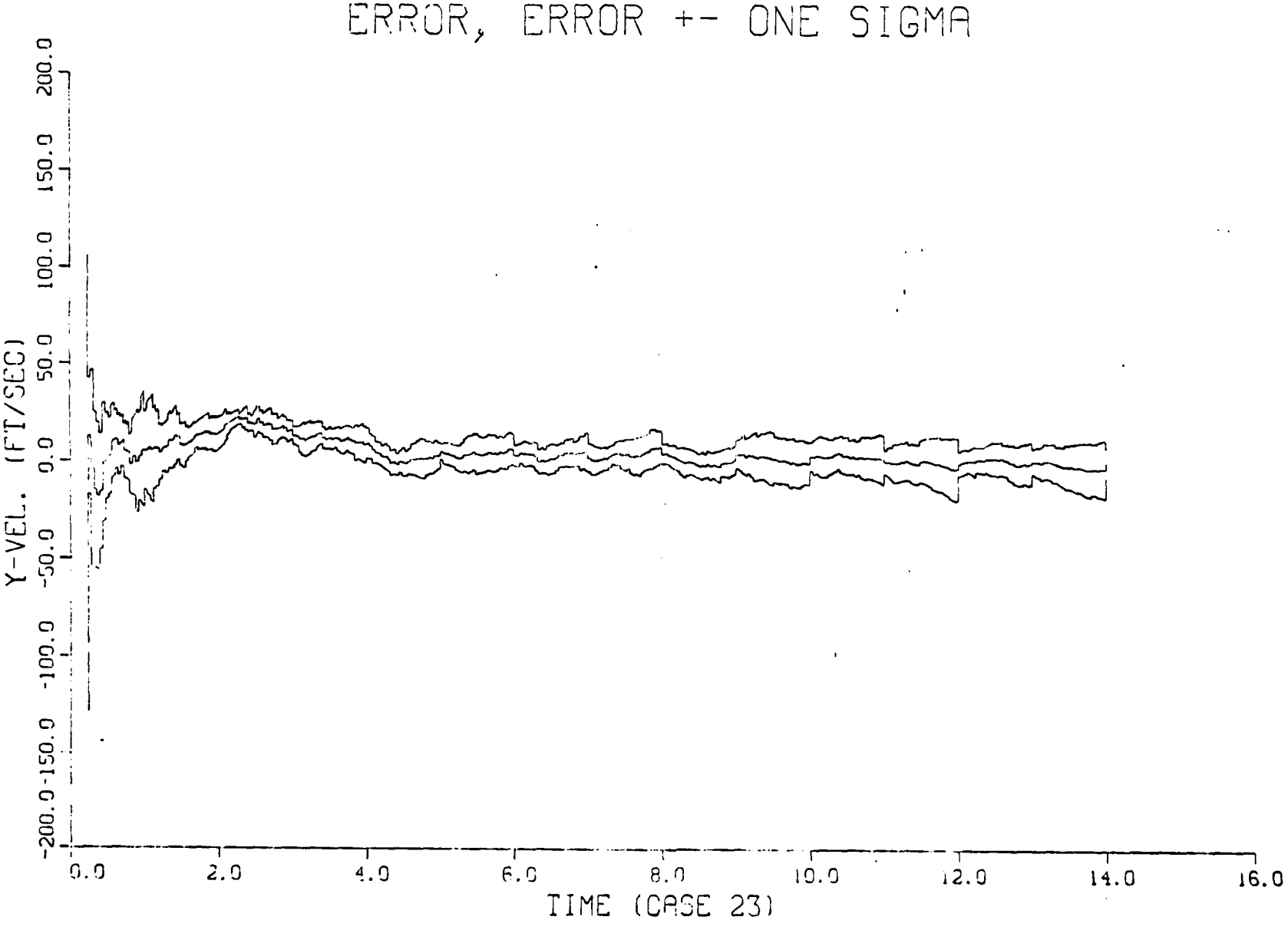
2.1.3.3


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 23)
2.1.3.3

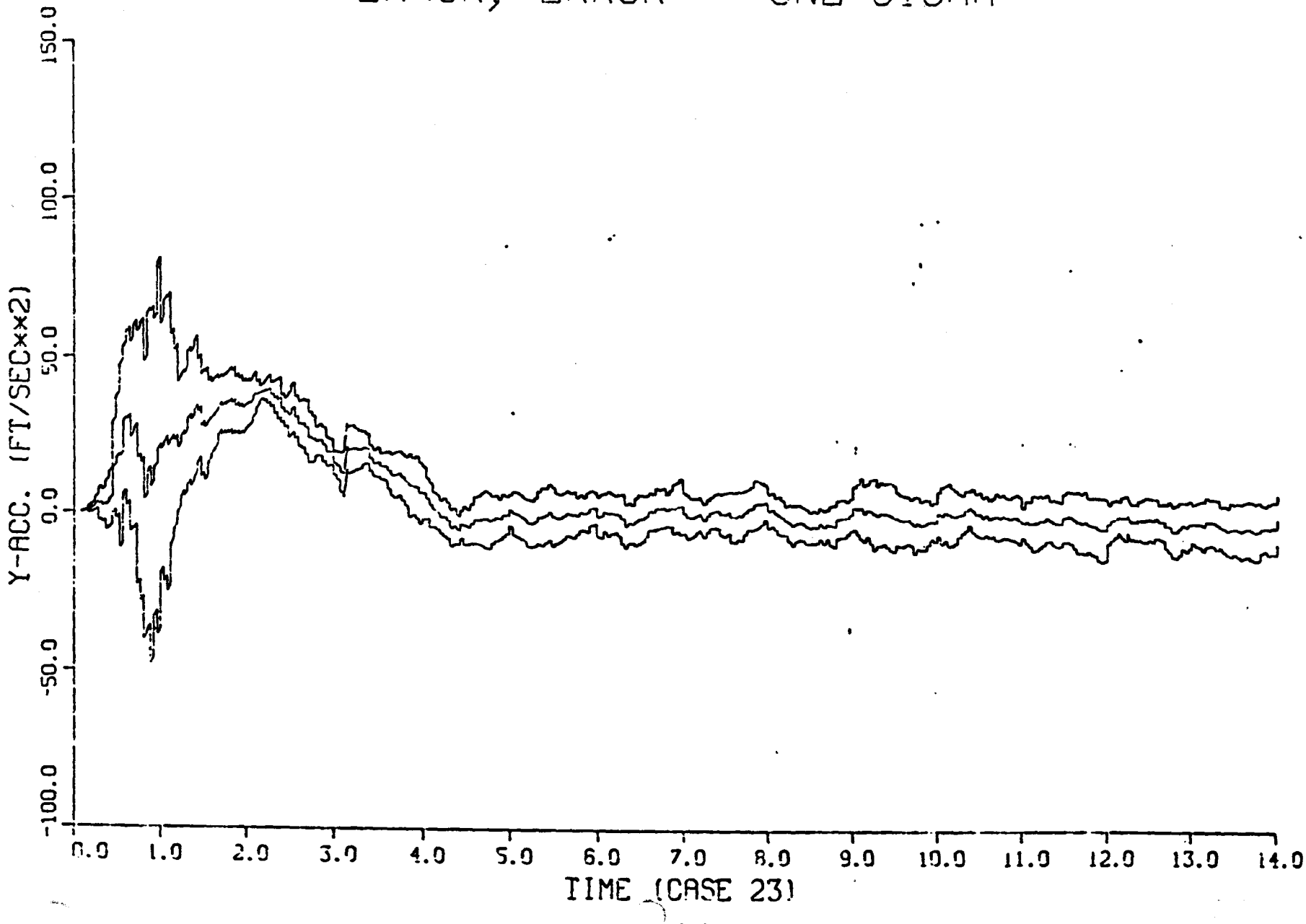

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 23)

2.1.3.3

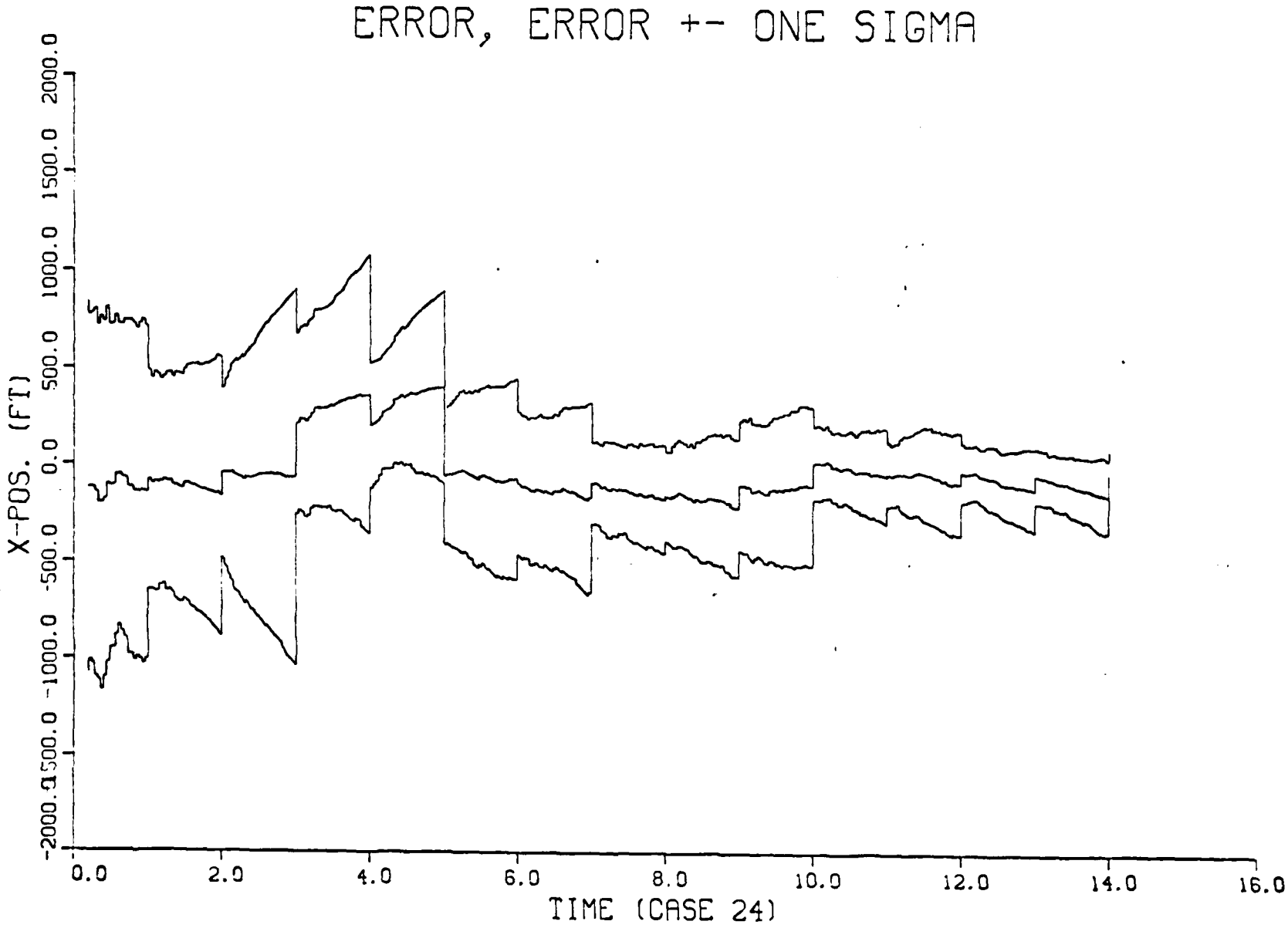

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
-15.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 23)


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 23)

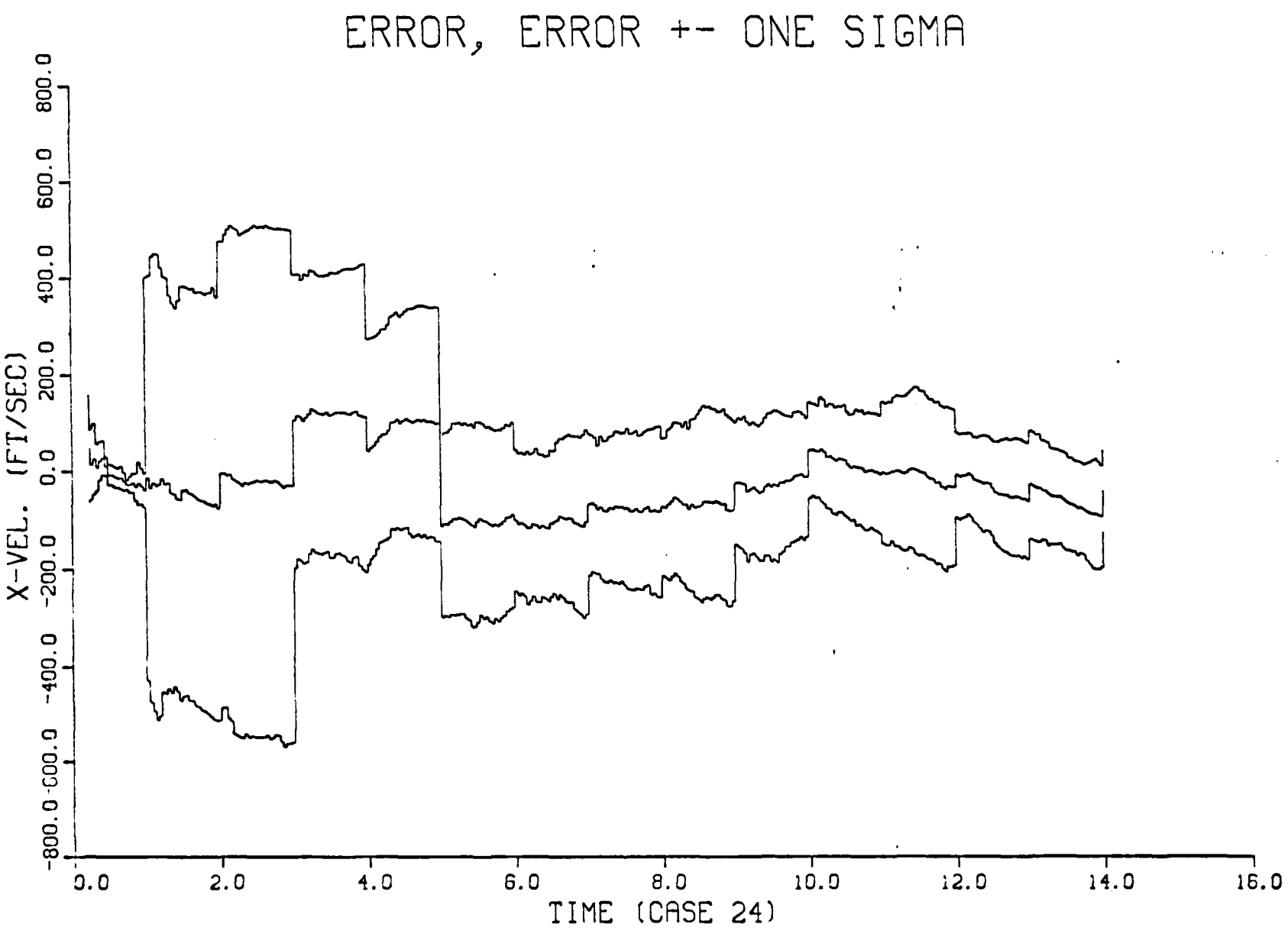
2.1.3.3


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 23)

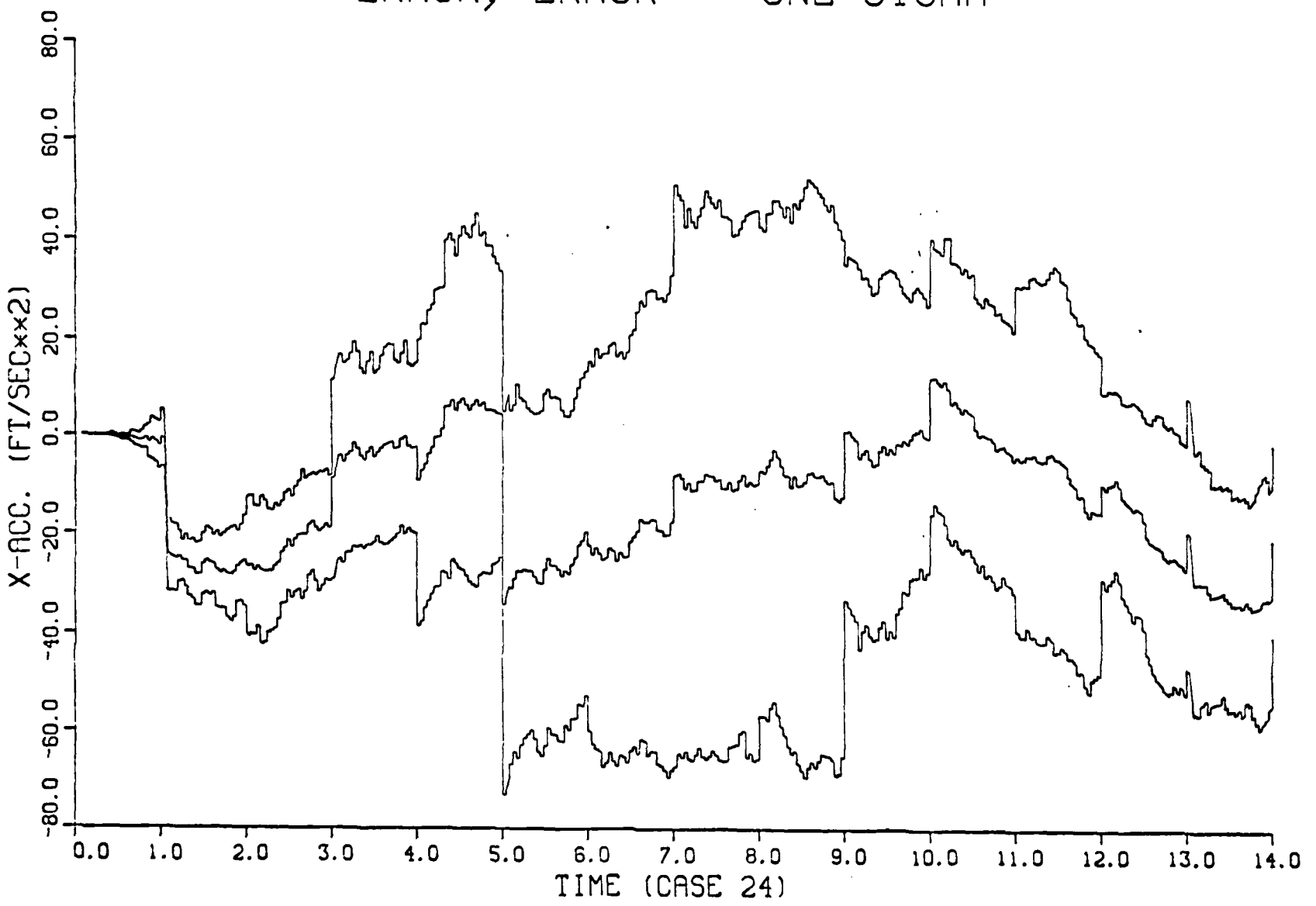
2.1.3.3


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
2000.0
1500.0
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
-2000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 24)

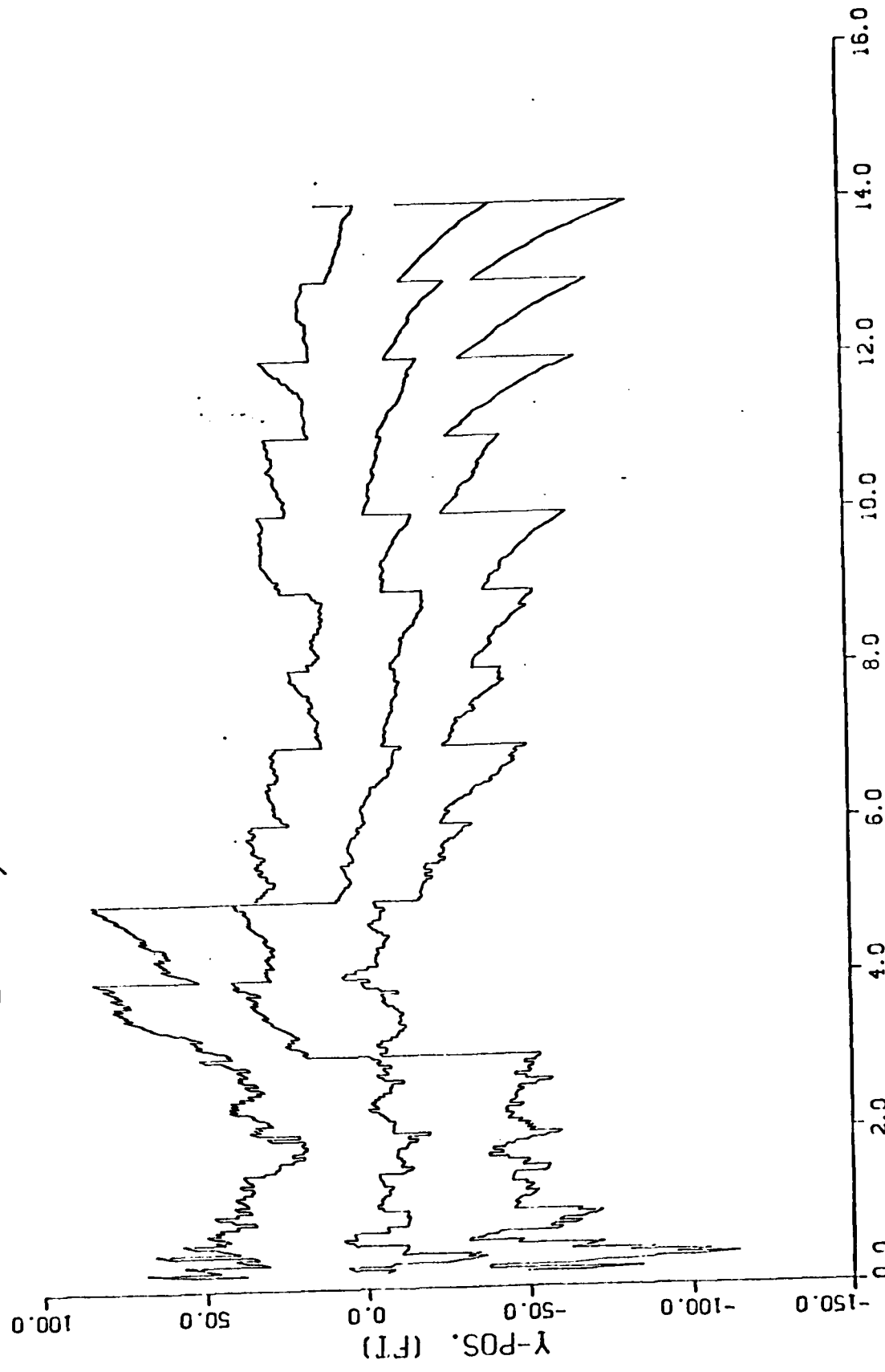
2.1.3.4


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
800.0
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 24)

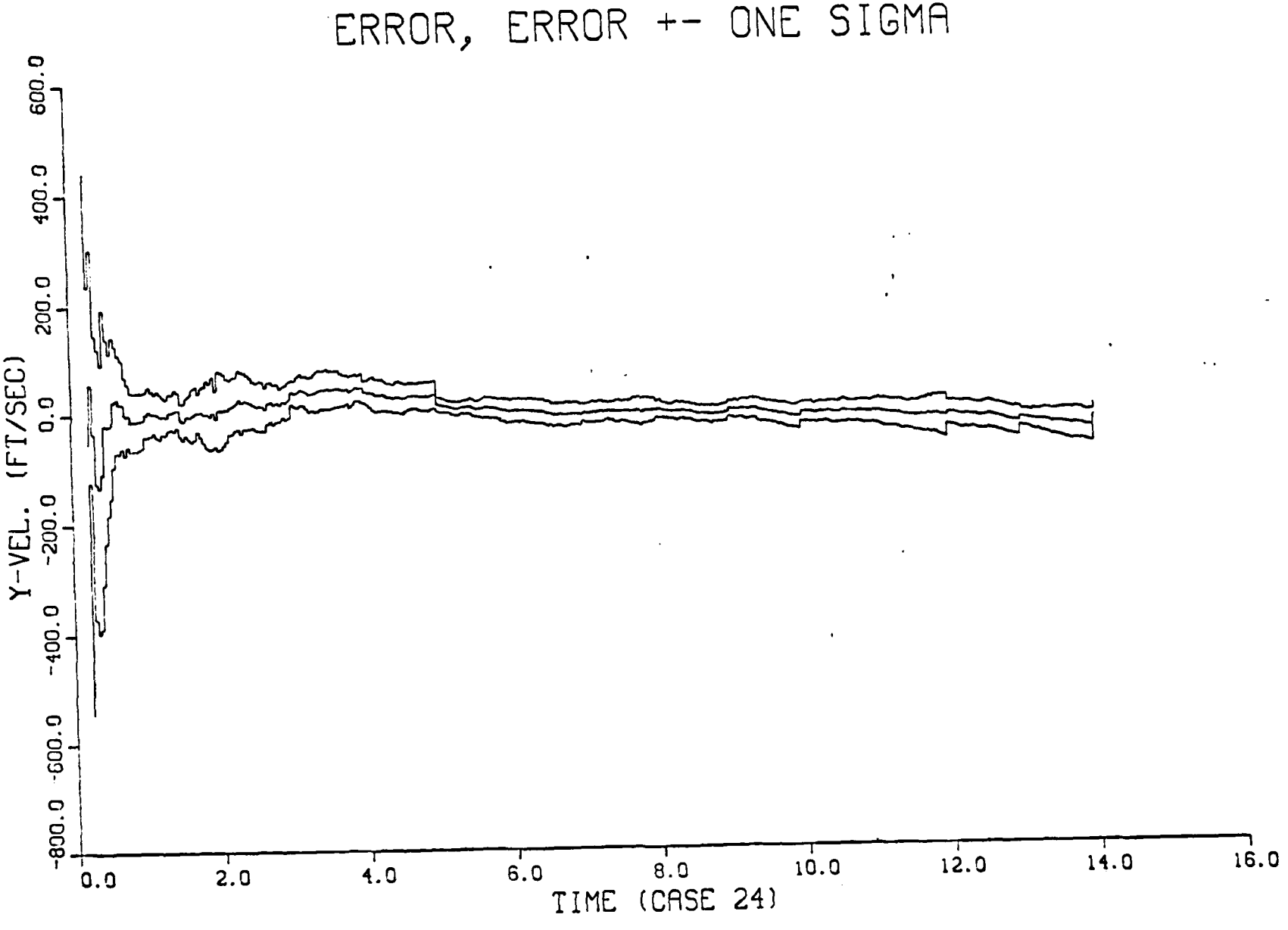
2.1.3.4


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 24)

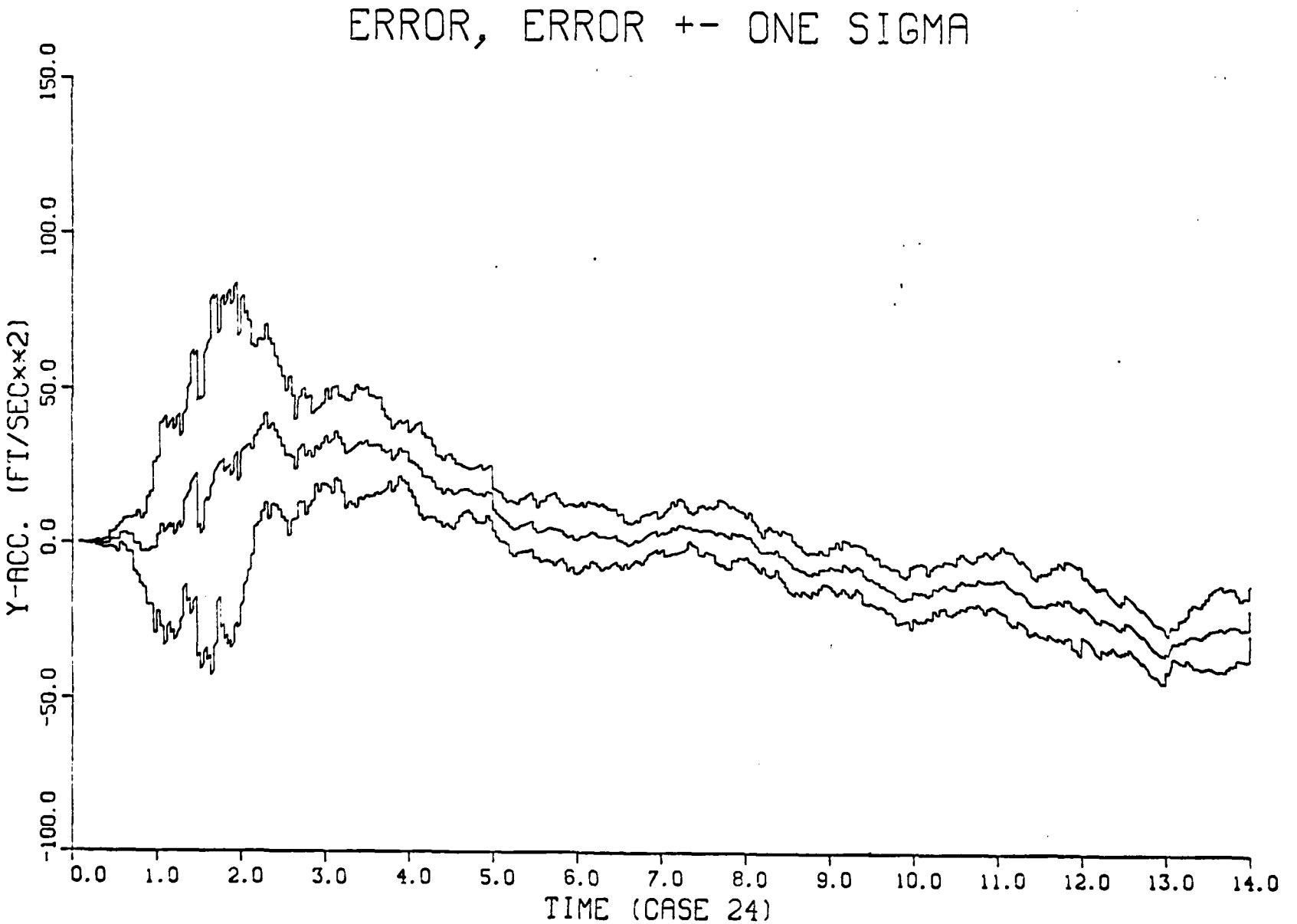
2.1.3.4


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 24)

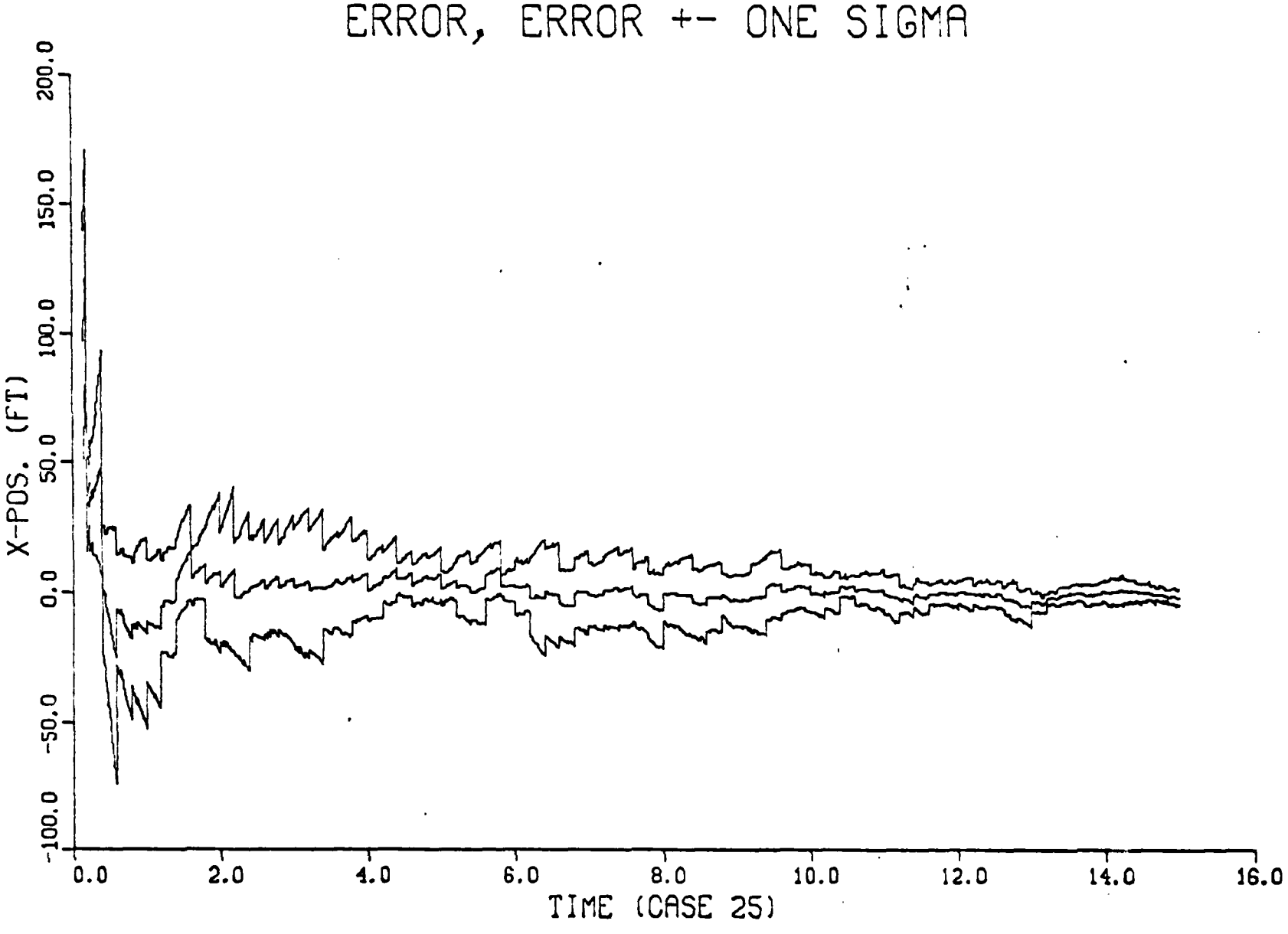
2.1.3.4


ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 24)

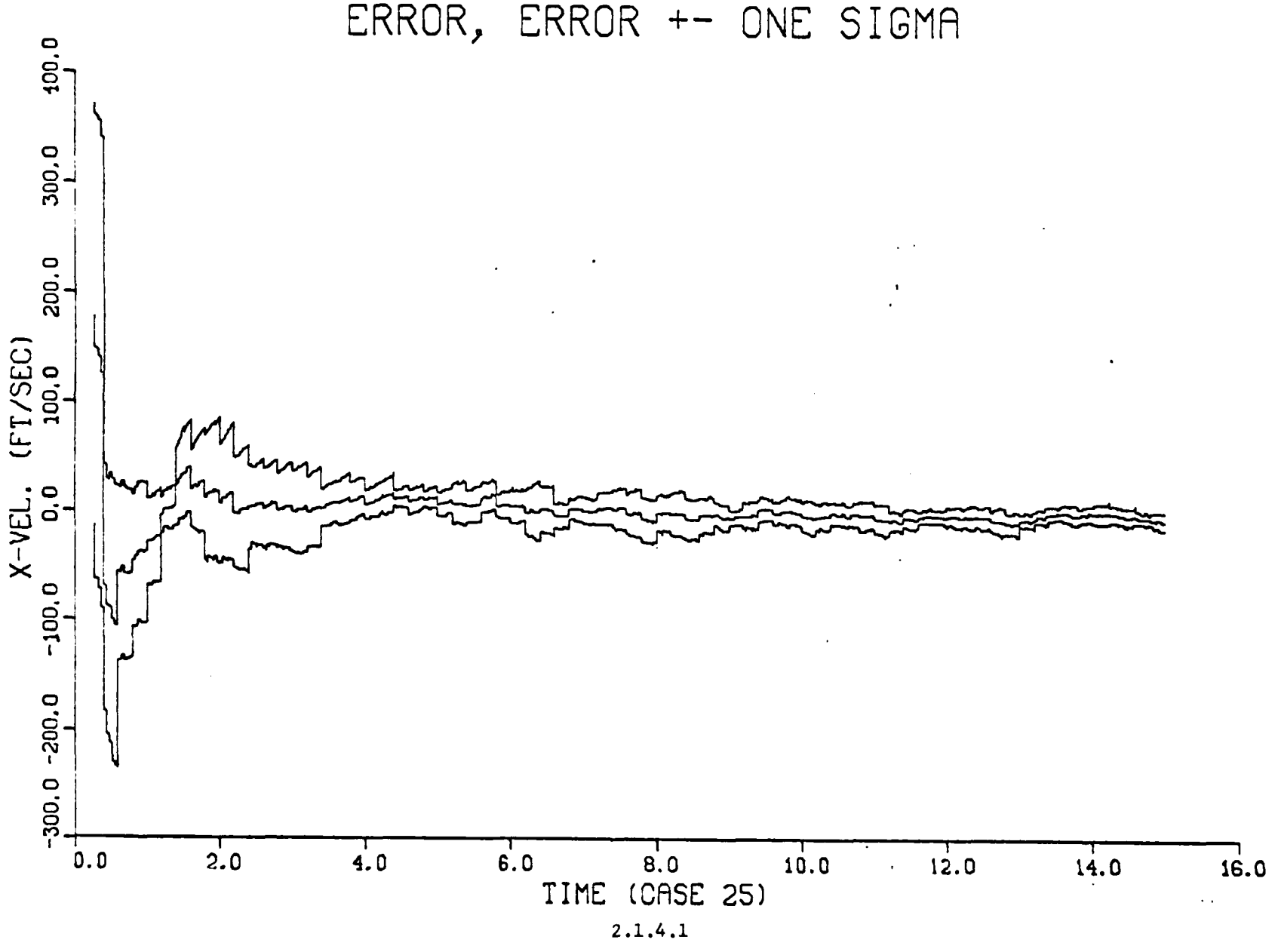
2.1.3.4

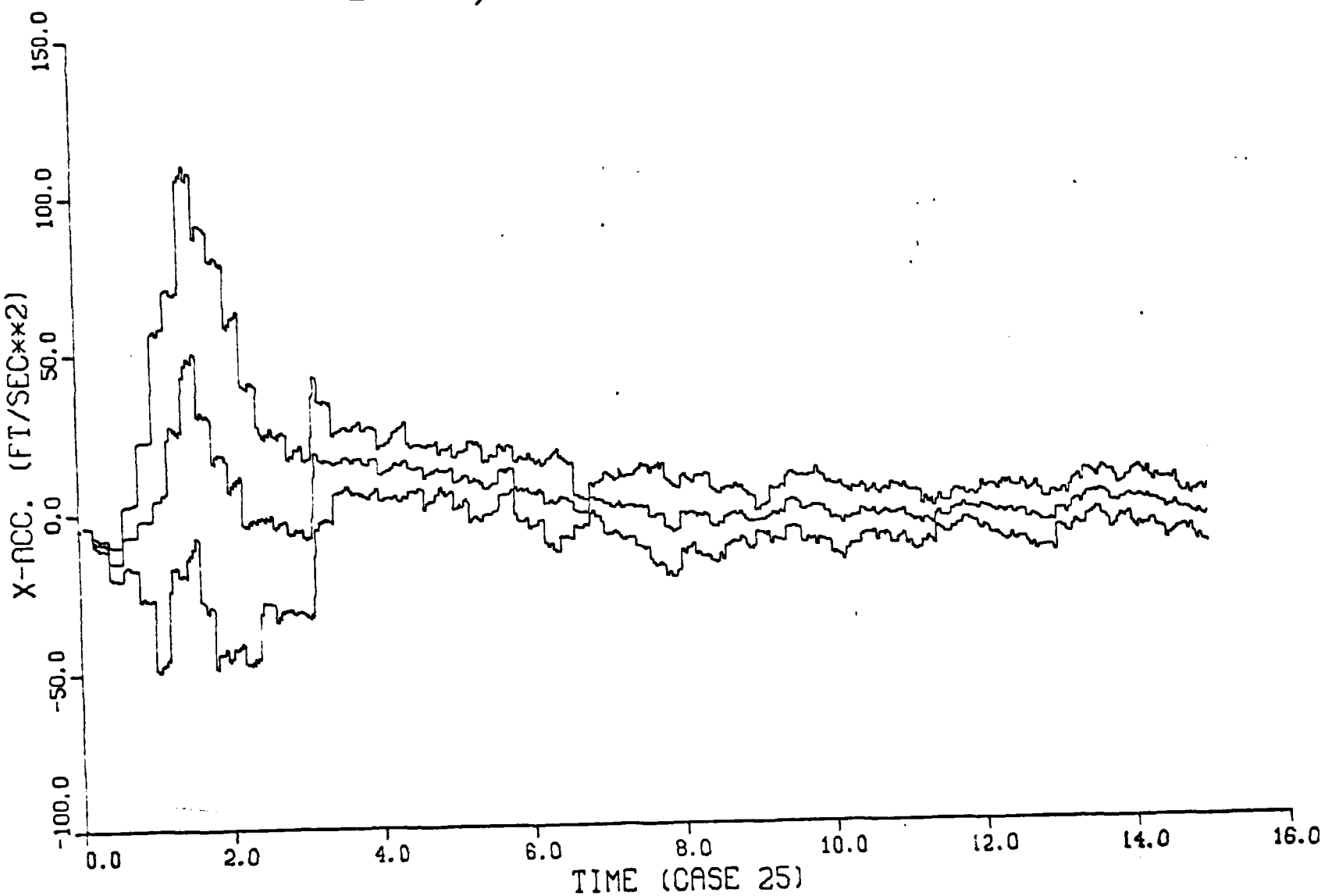

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 24)

2.1.3.4

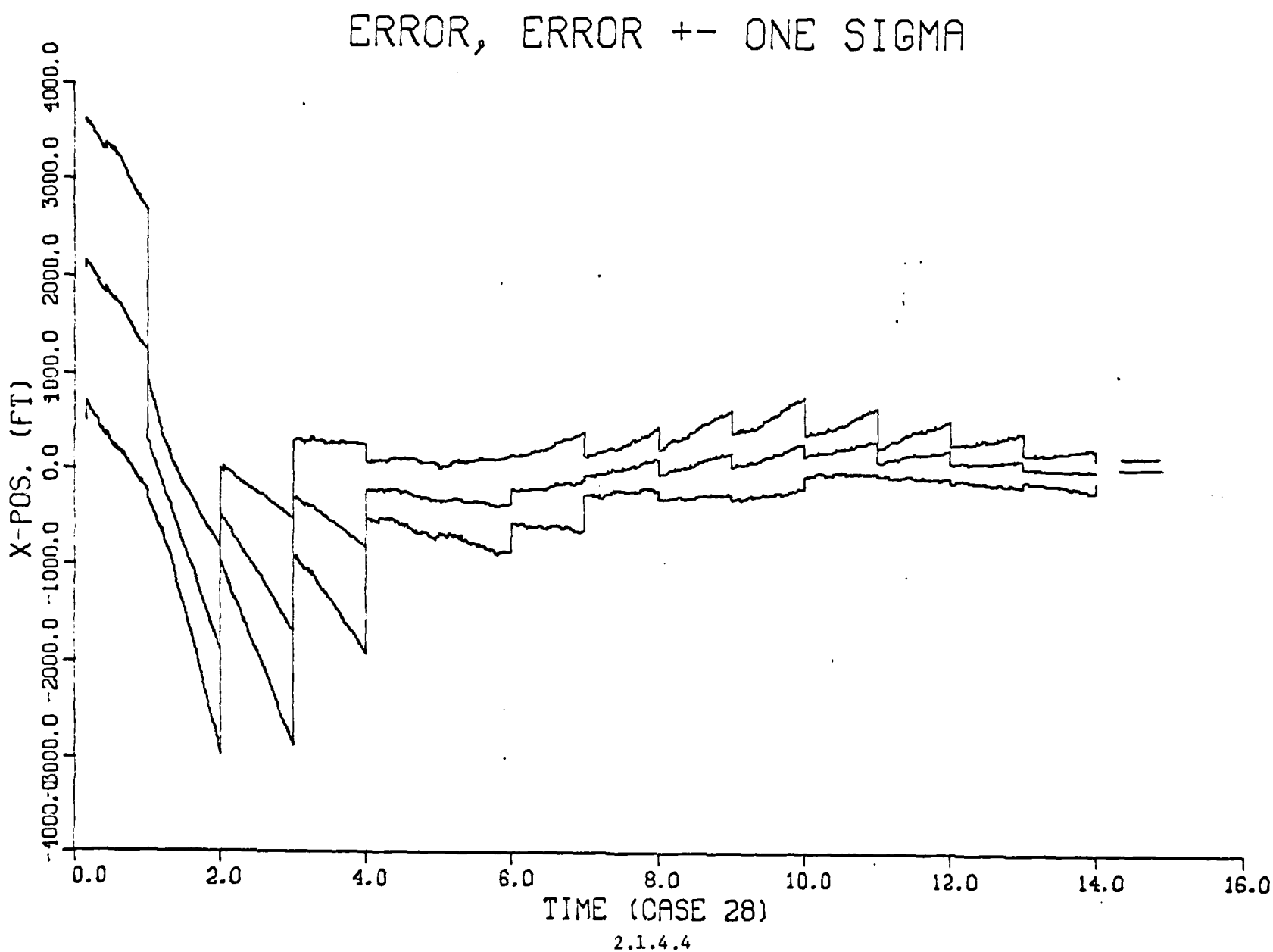

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 25)

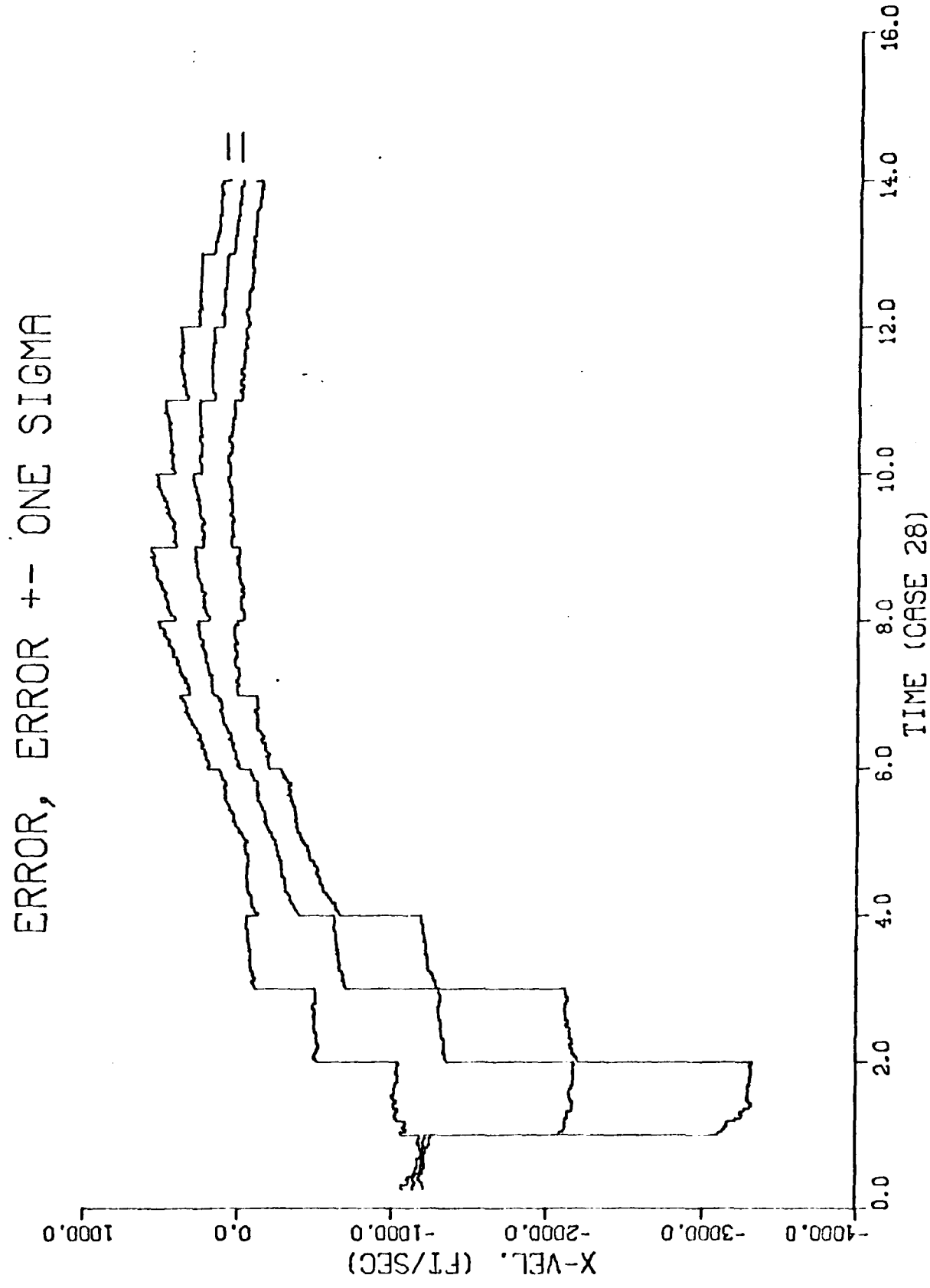
2.1.4.1


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
100.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 25)
2.1.4.1

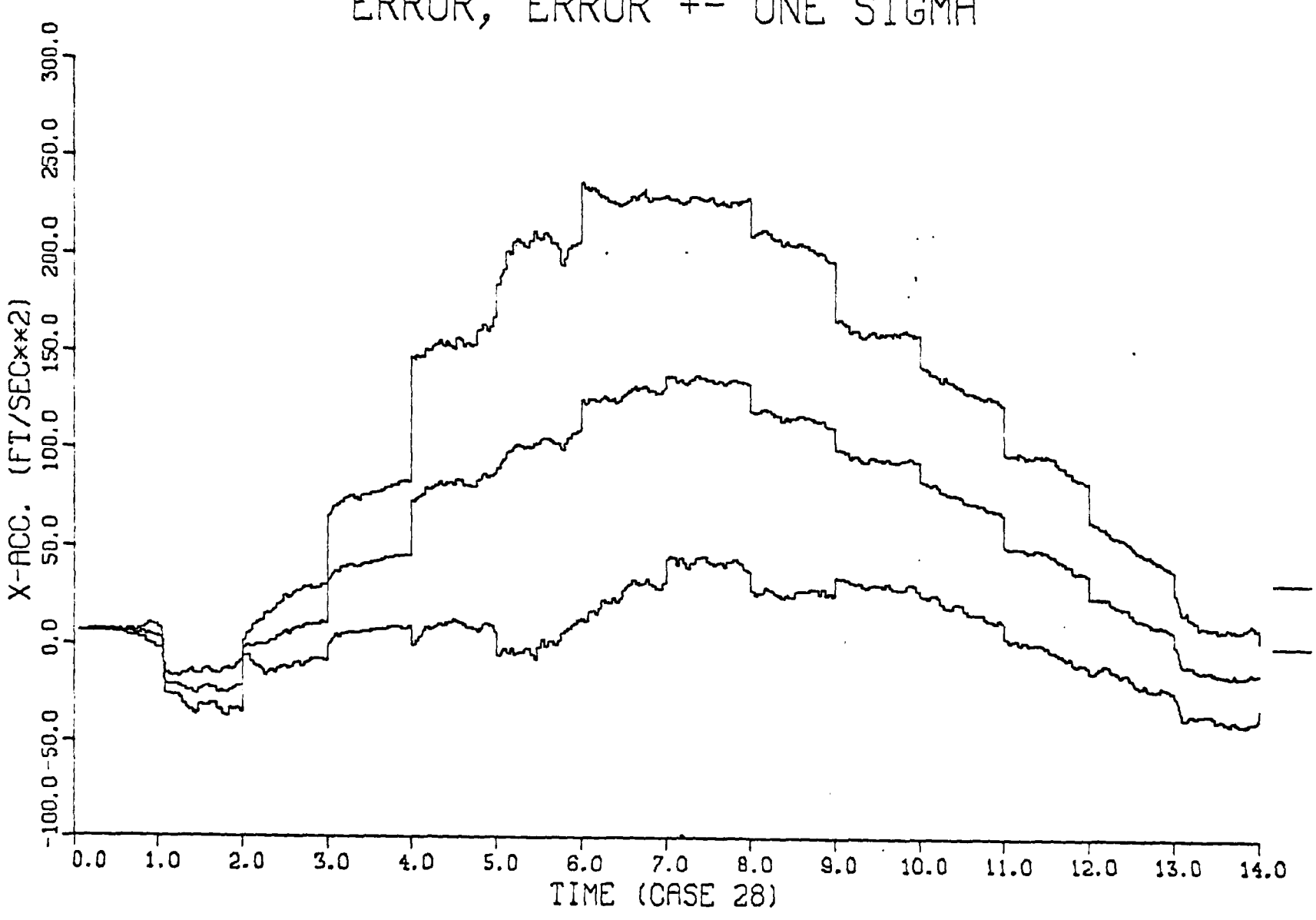

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 25)

2.1.4.1

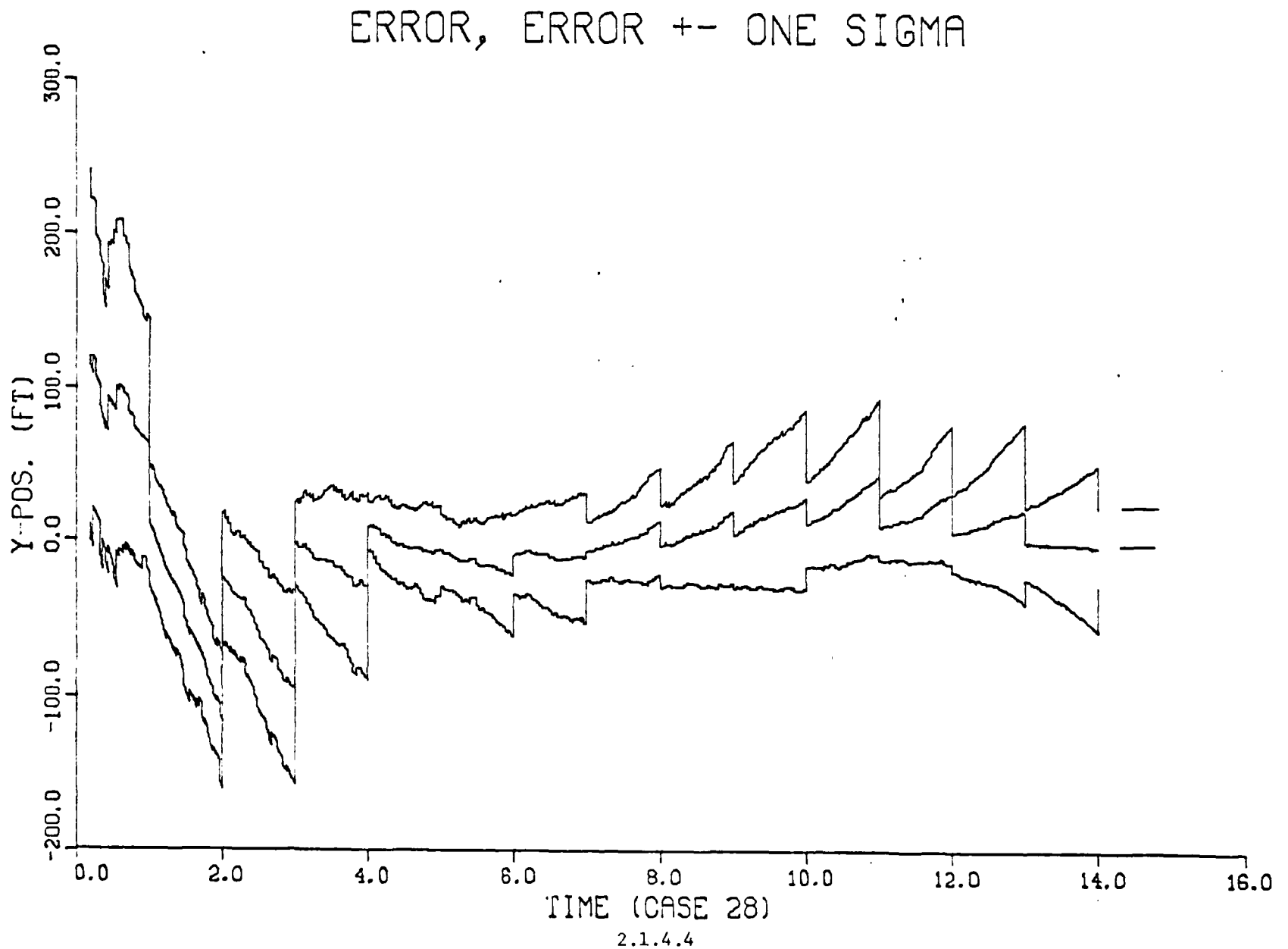

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
-4000.0
-3000.0
-2000.0
-1000.0
0.0
1000.0
2000.0
3000.0
4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 28)
2.1.4.4

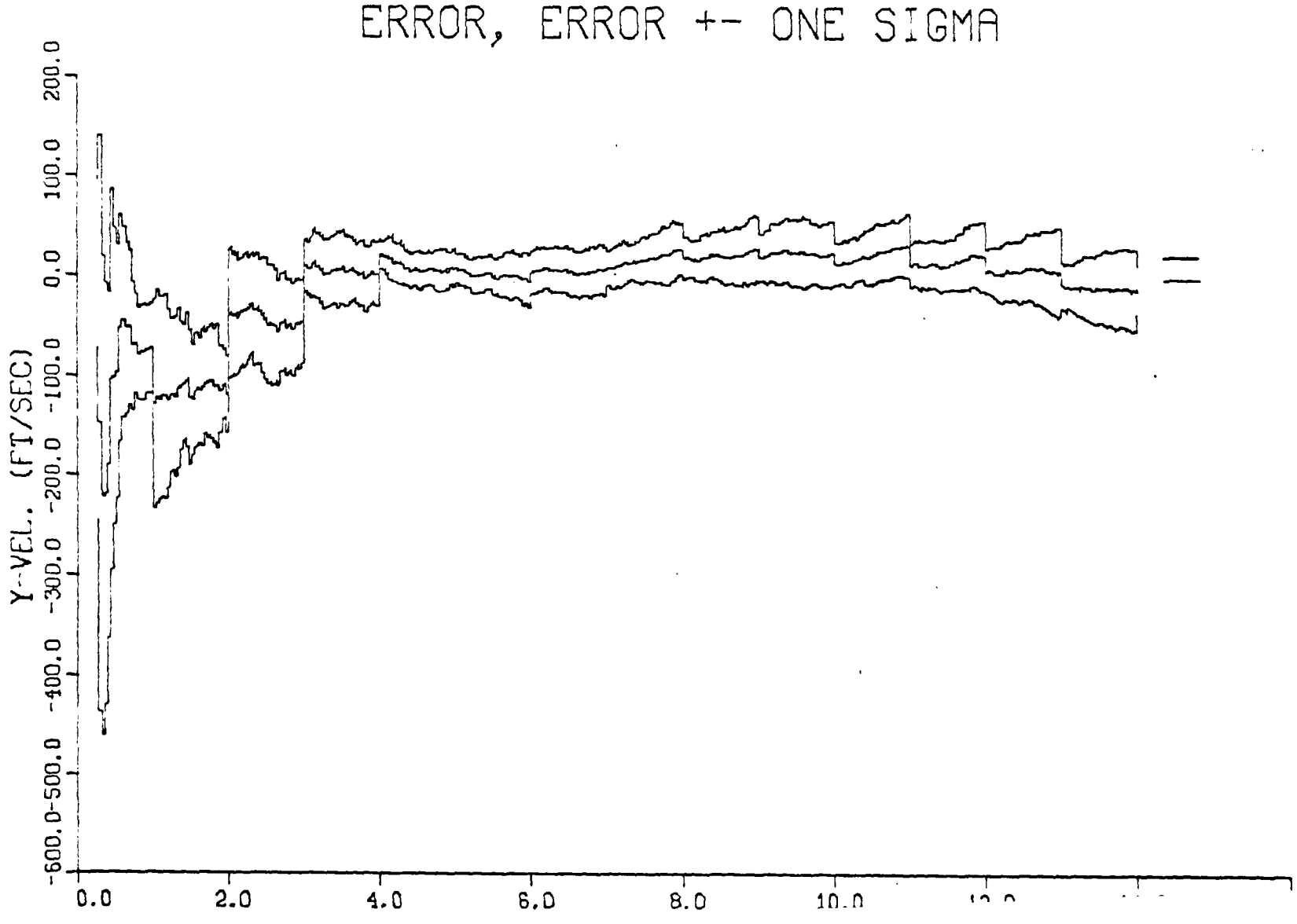

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
1000.0
0.0
-1000.0
-2000.0
-3000.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 28)

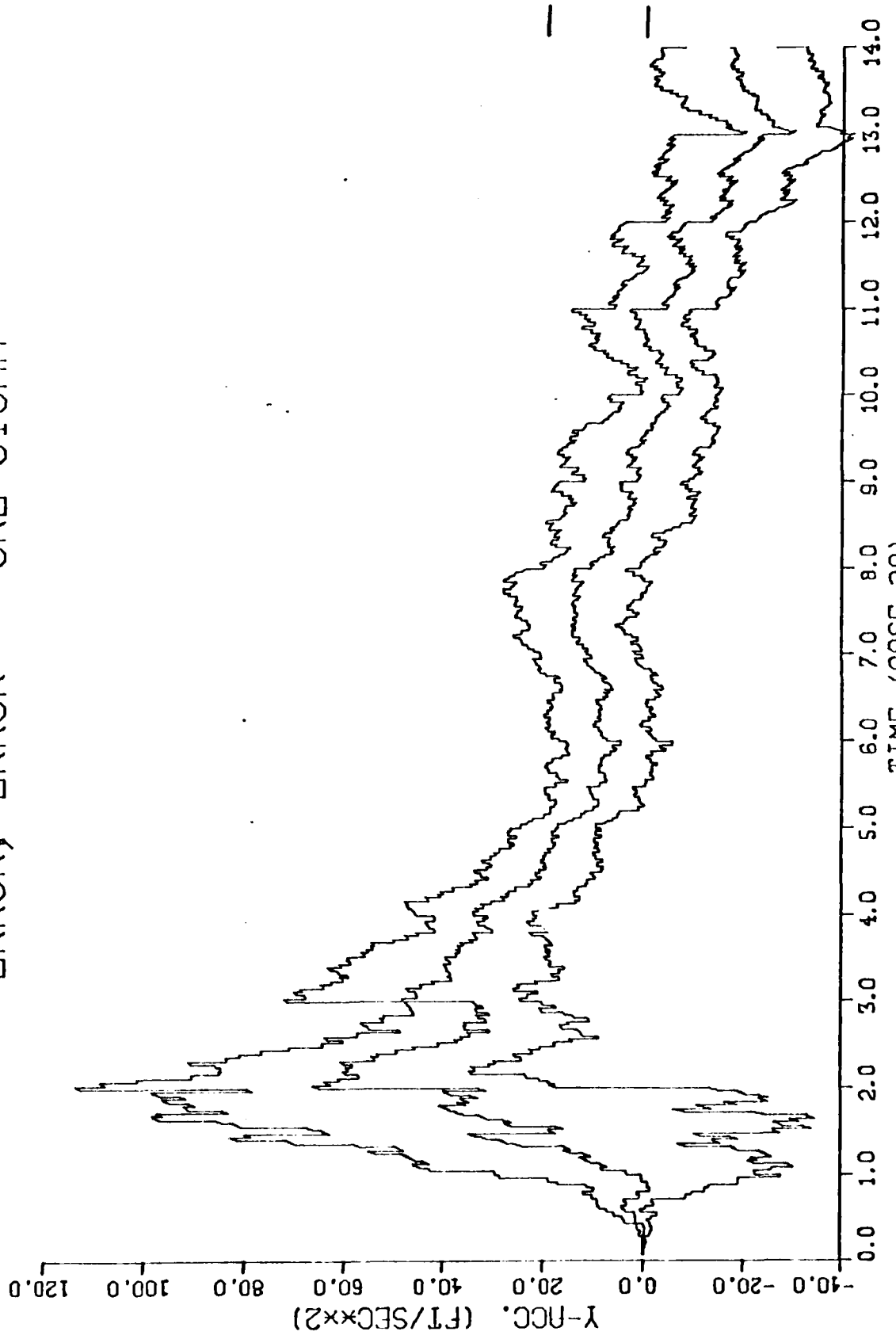
2.1.4.4


ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 28)

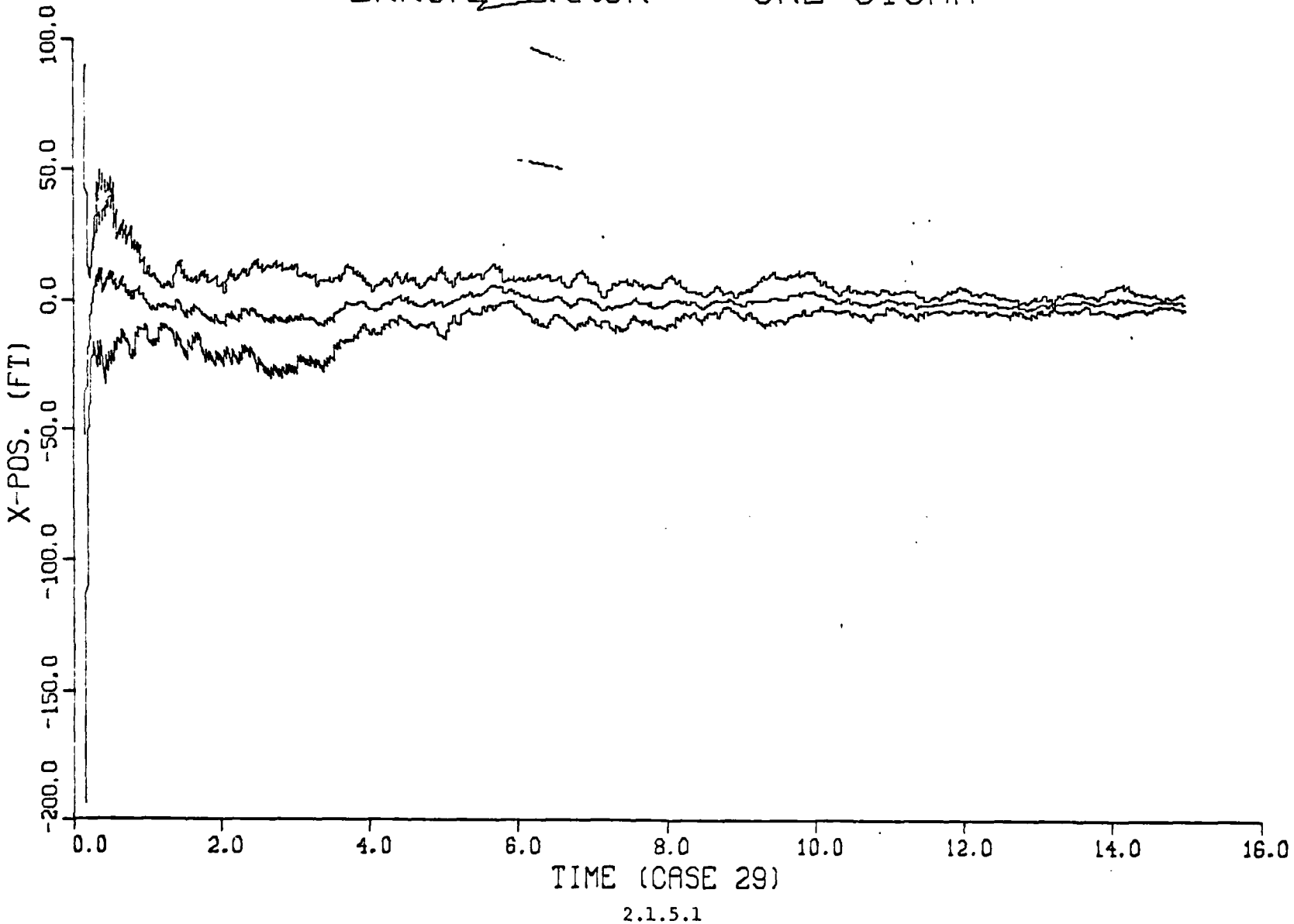
2.1.4.4


ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
300.0
200.0
100.0
0.0
-100.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 28)
2.1.4.4

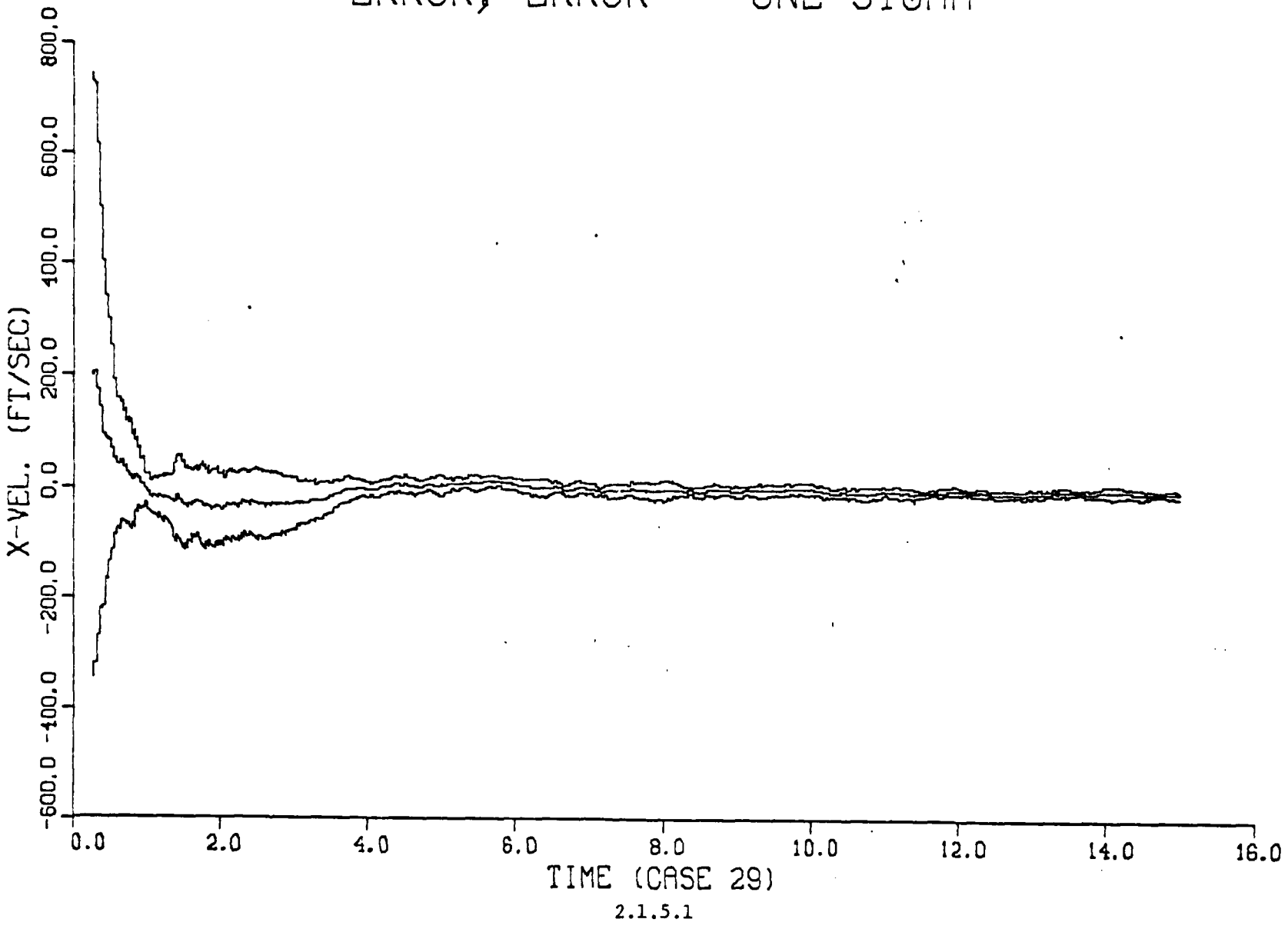

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
-500.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0


ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
-40.0
-20.0
0.0
20.0
40.0
60.0
80.0
100.0
120.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
TIME (CASE 28)

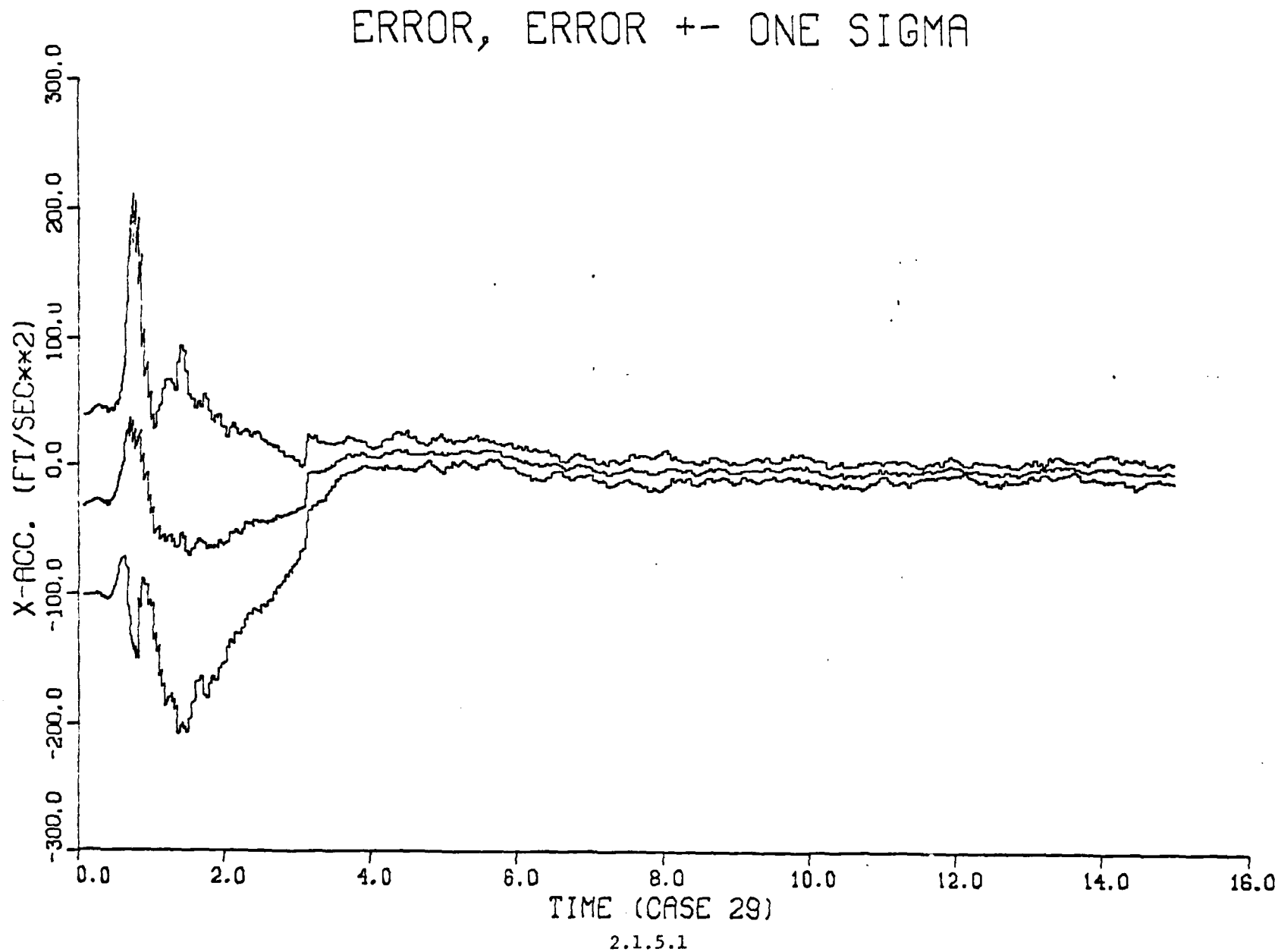
2.1.4.4

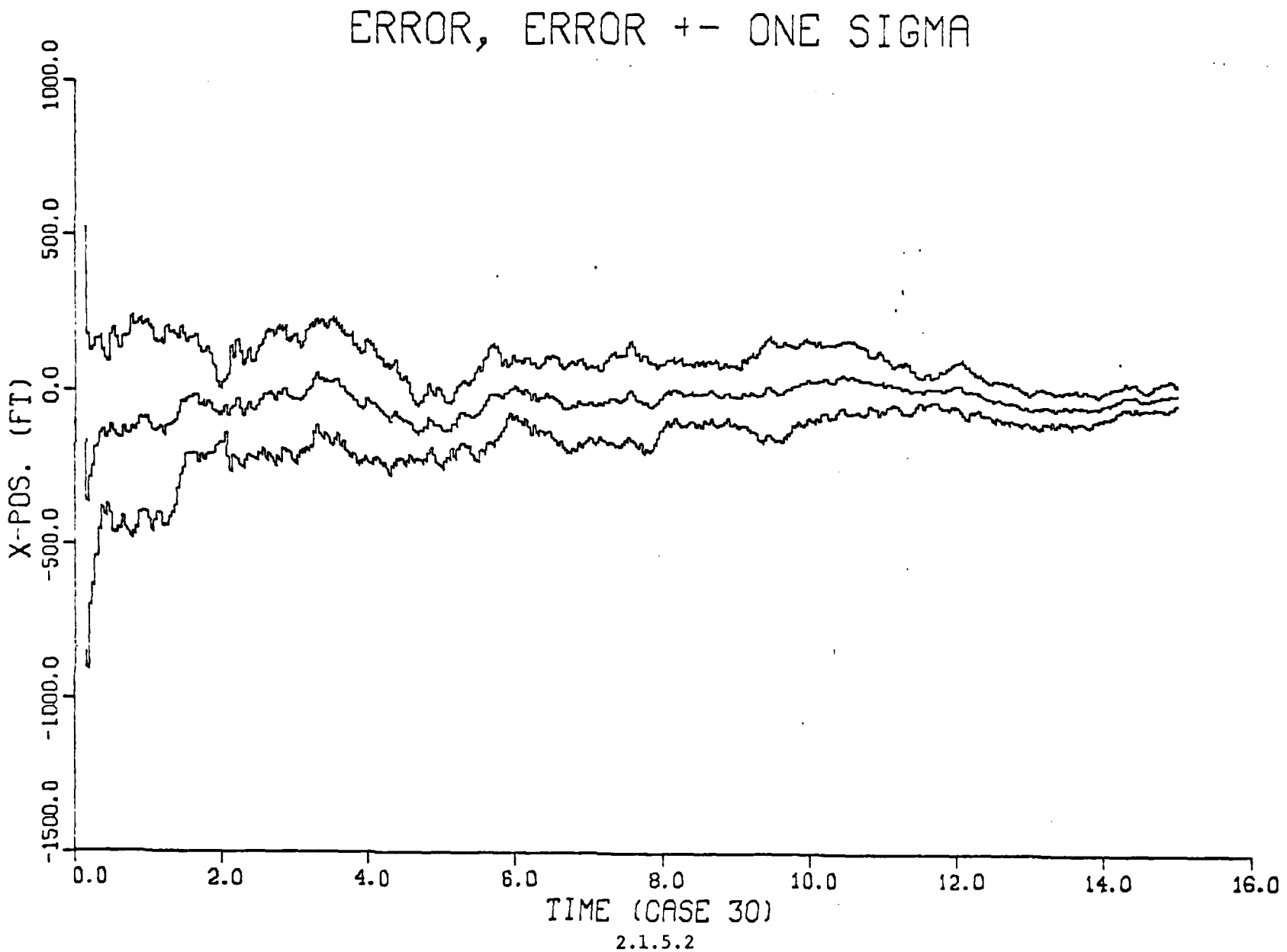

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 29)

2.1.5.1

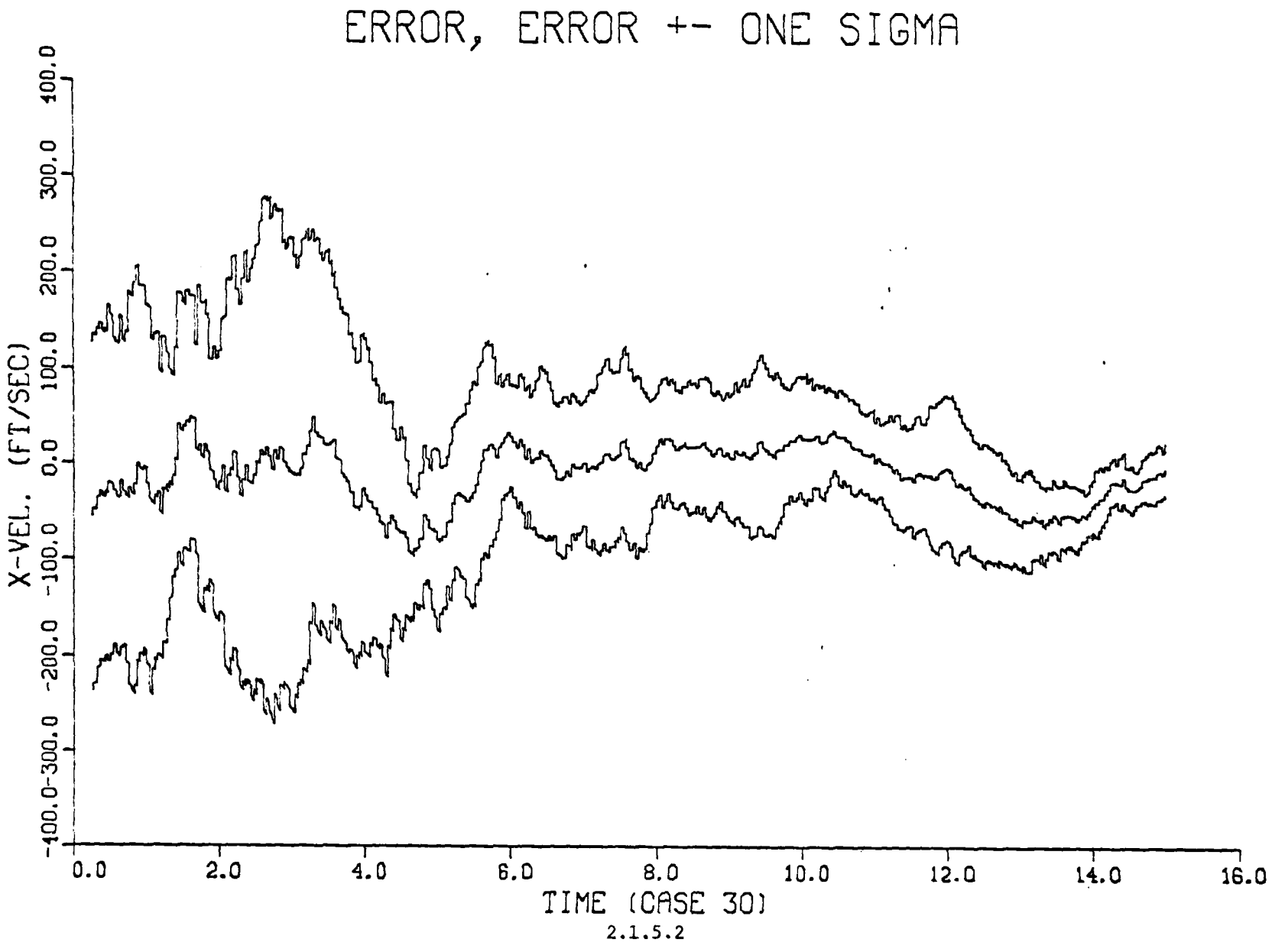

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
800.0
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 29)

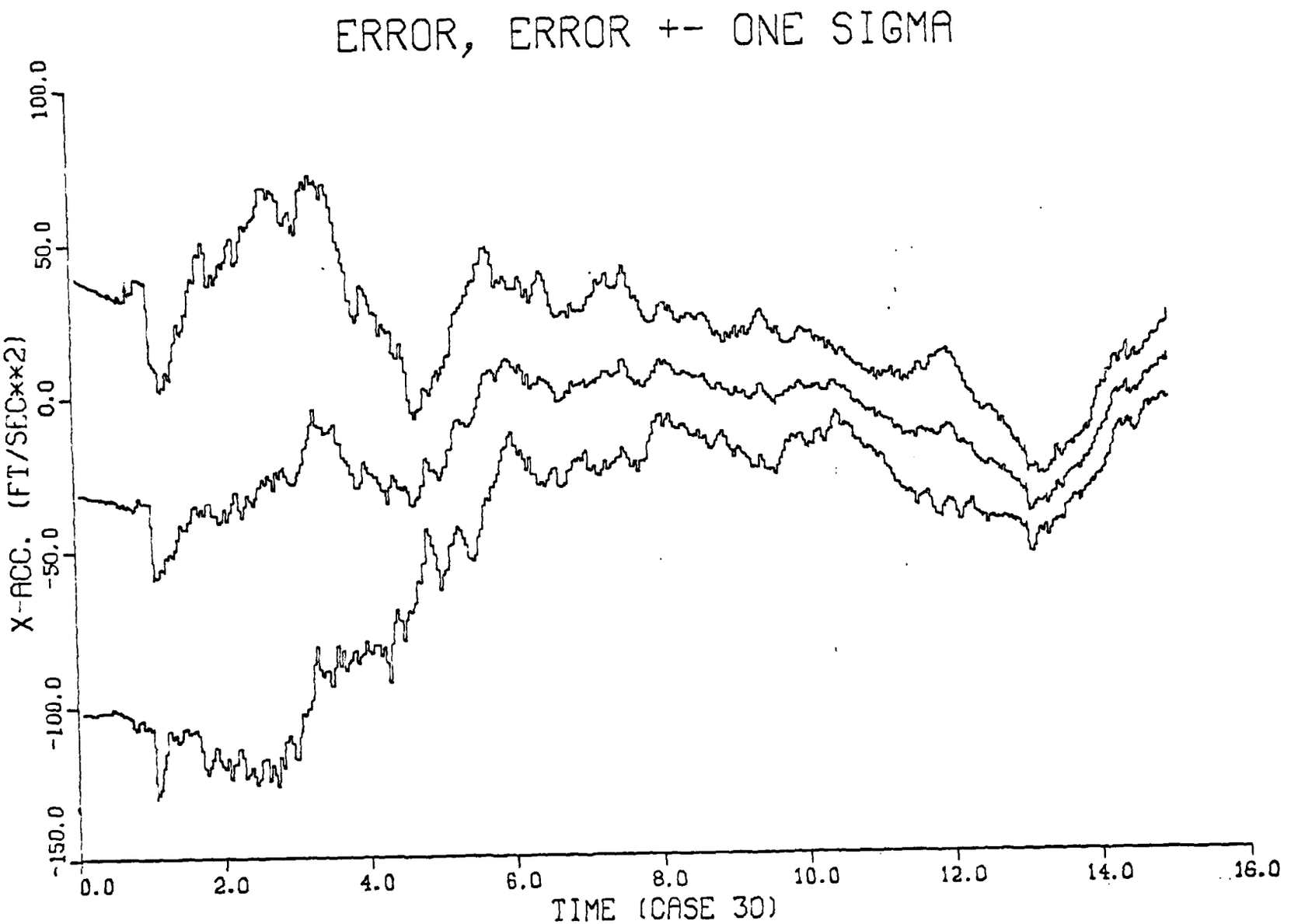
2.1.5.1

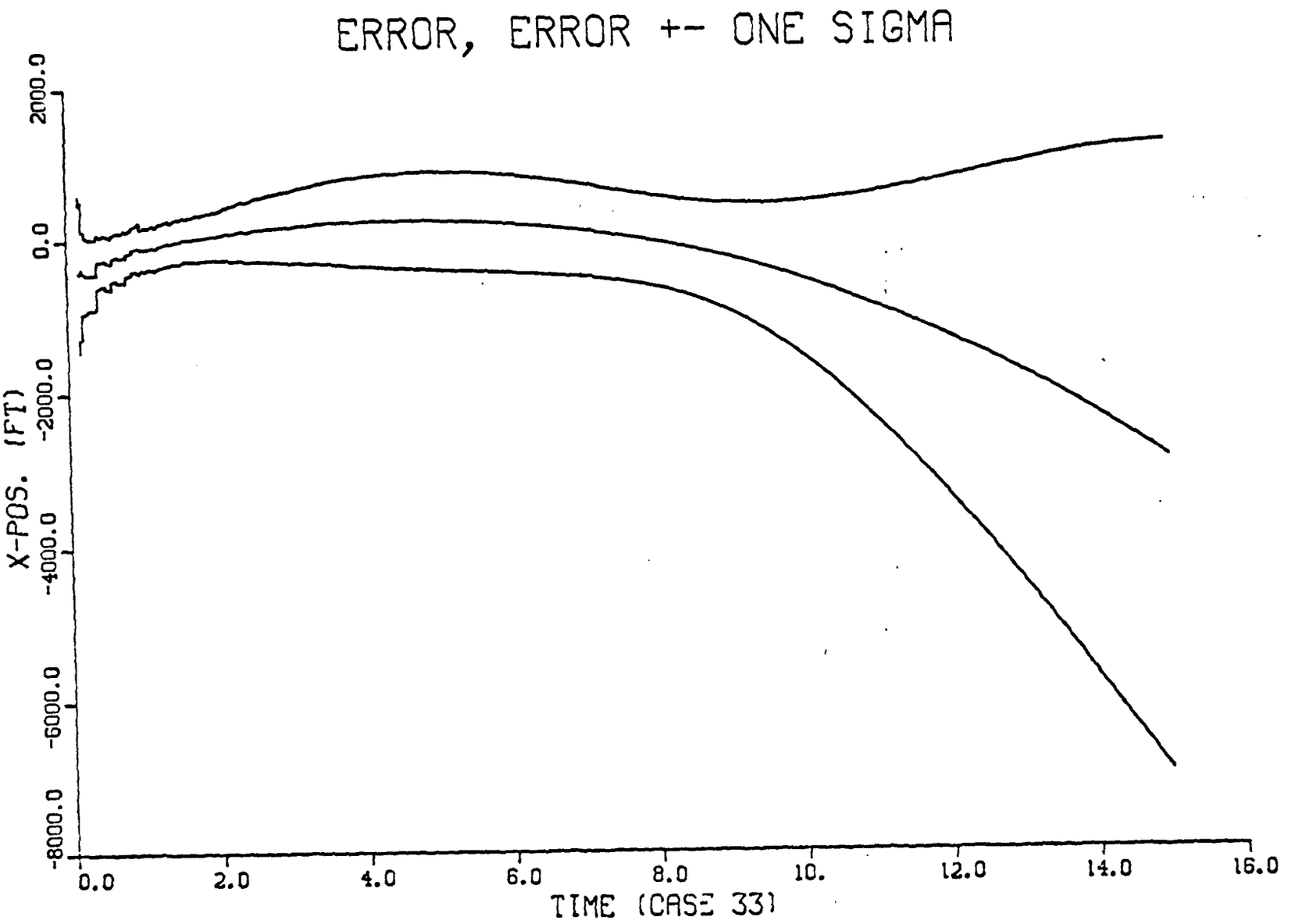

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 29)
2.1.5.1


ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 30)

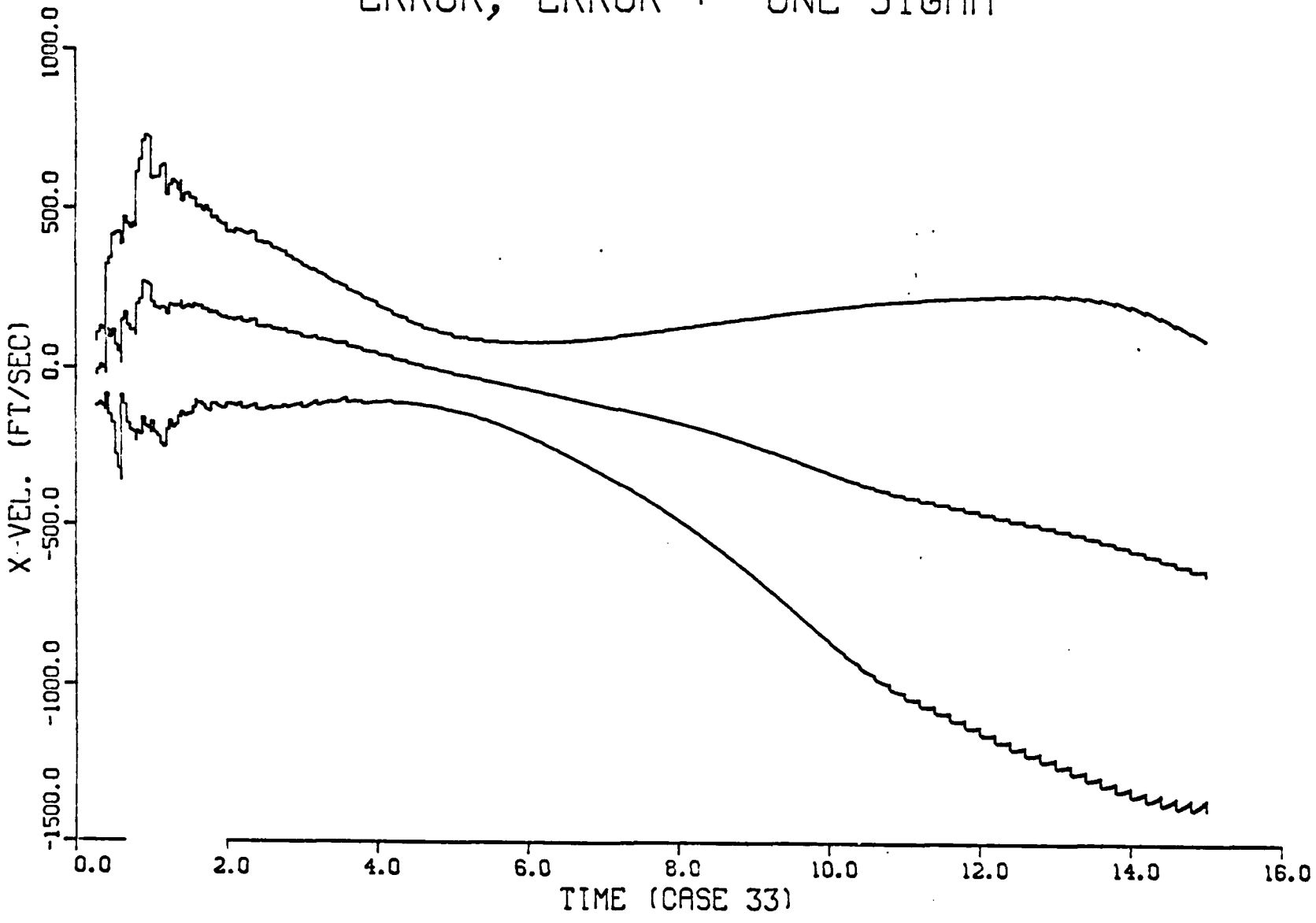
2.1.5.2


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
-400.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 30)
2.1.5.2

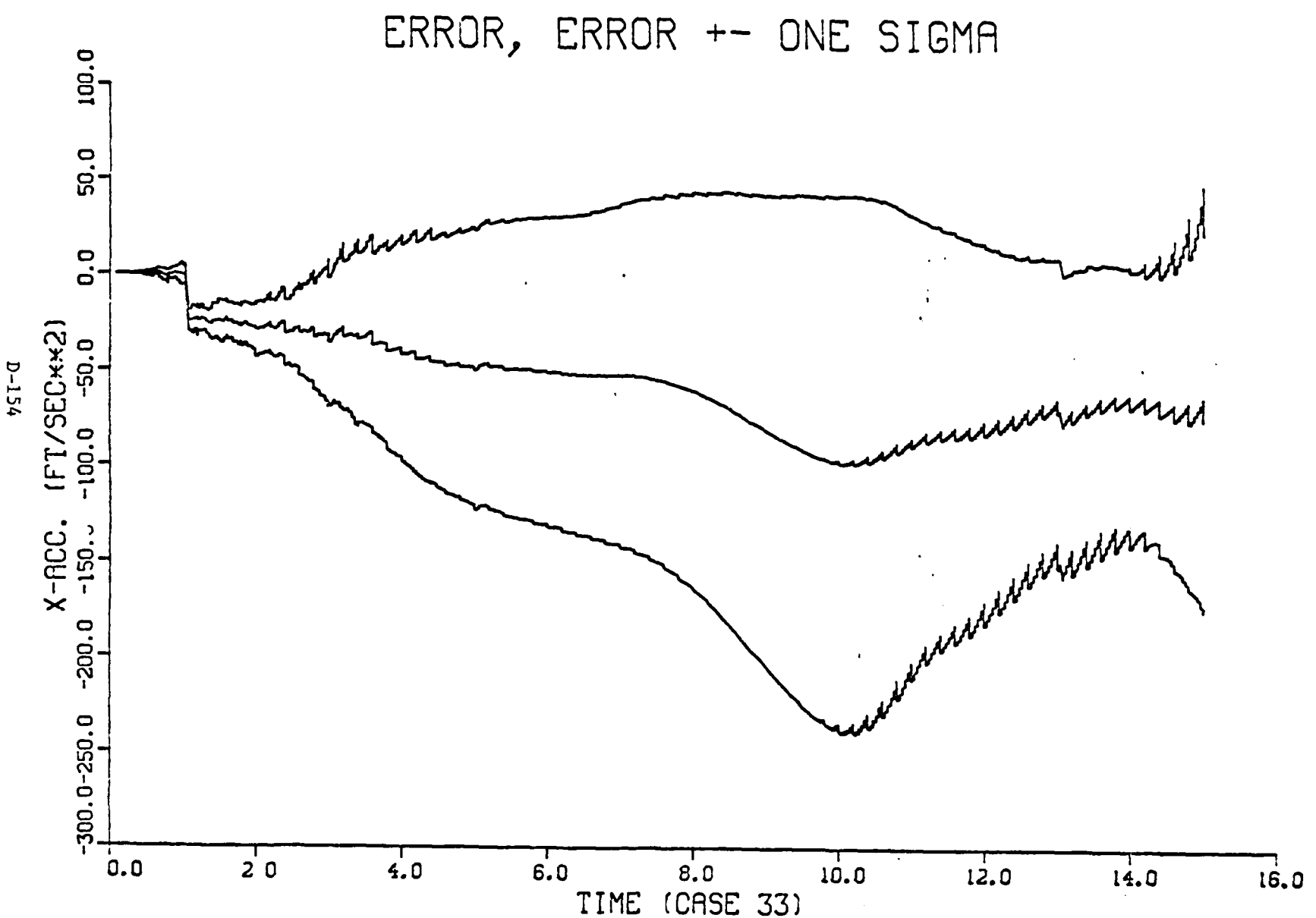

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 30)

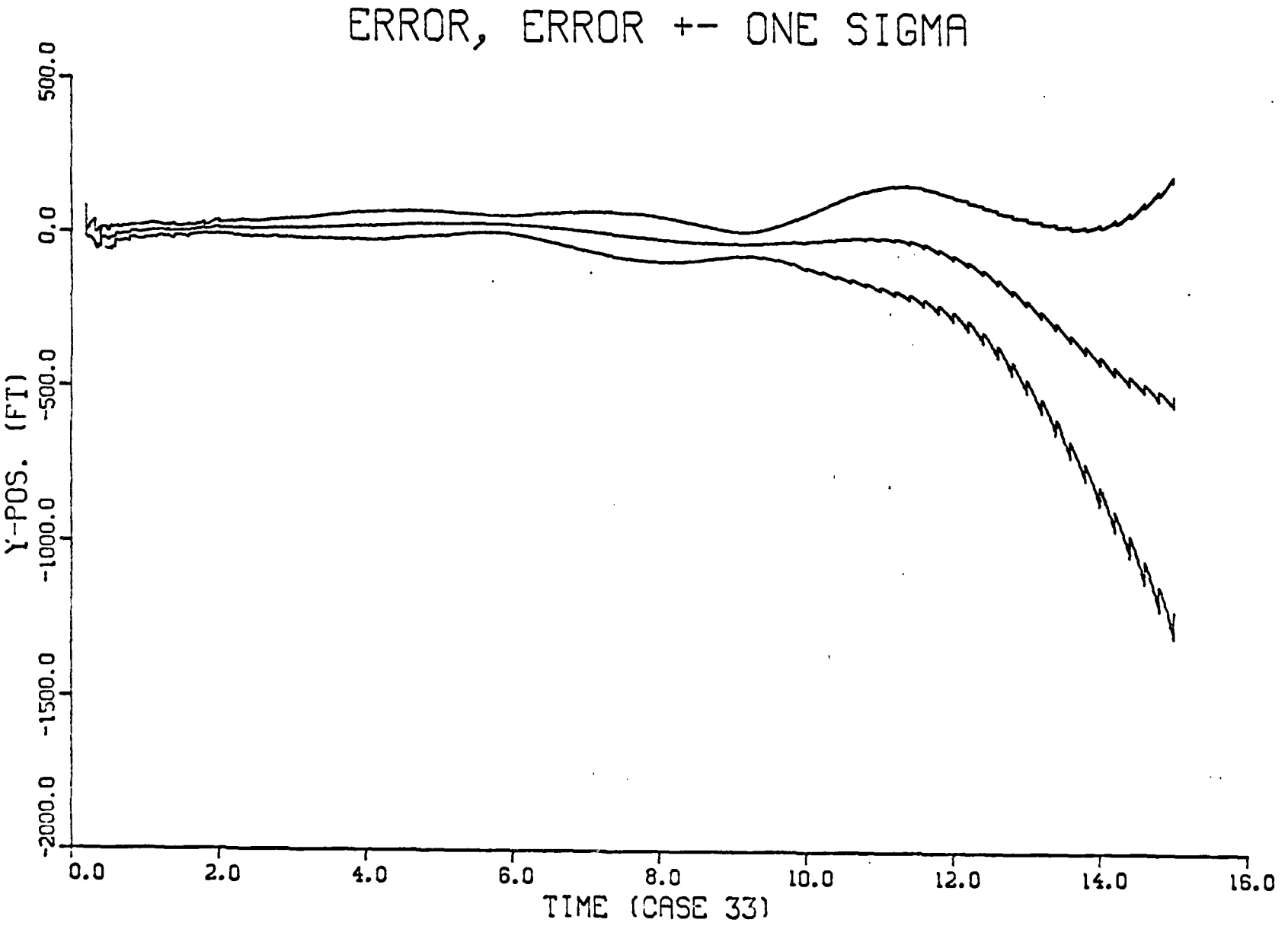

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
2000.0
0.0
-2000.0
-4000.0
-6000.0
-8000.0
0.0
2.0
4.0
6.0
8.0
10.
12.0
14.0
16.0
TIME (CASE 33)

2.2.1.1

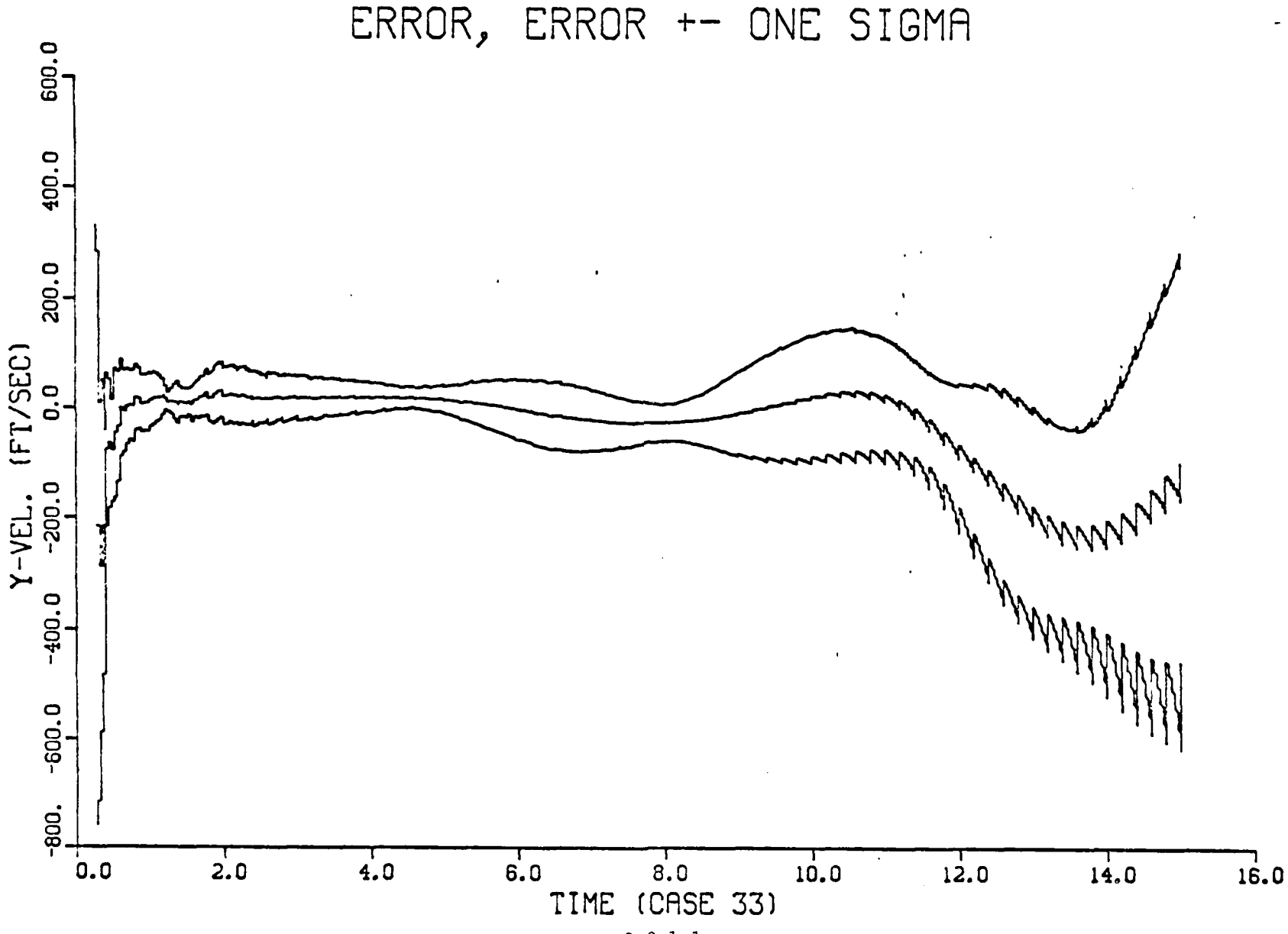

ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 33)

2.2.1.1

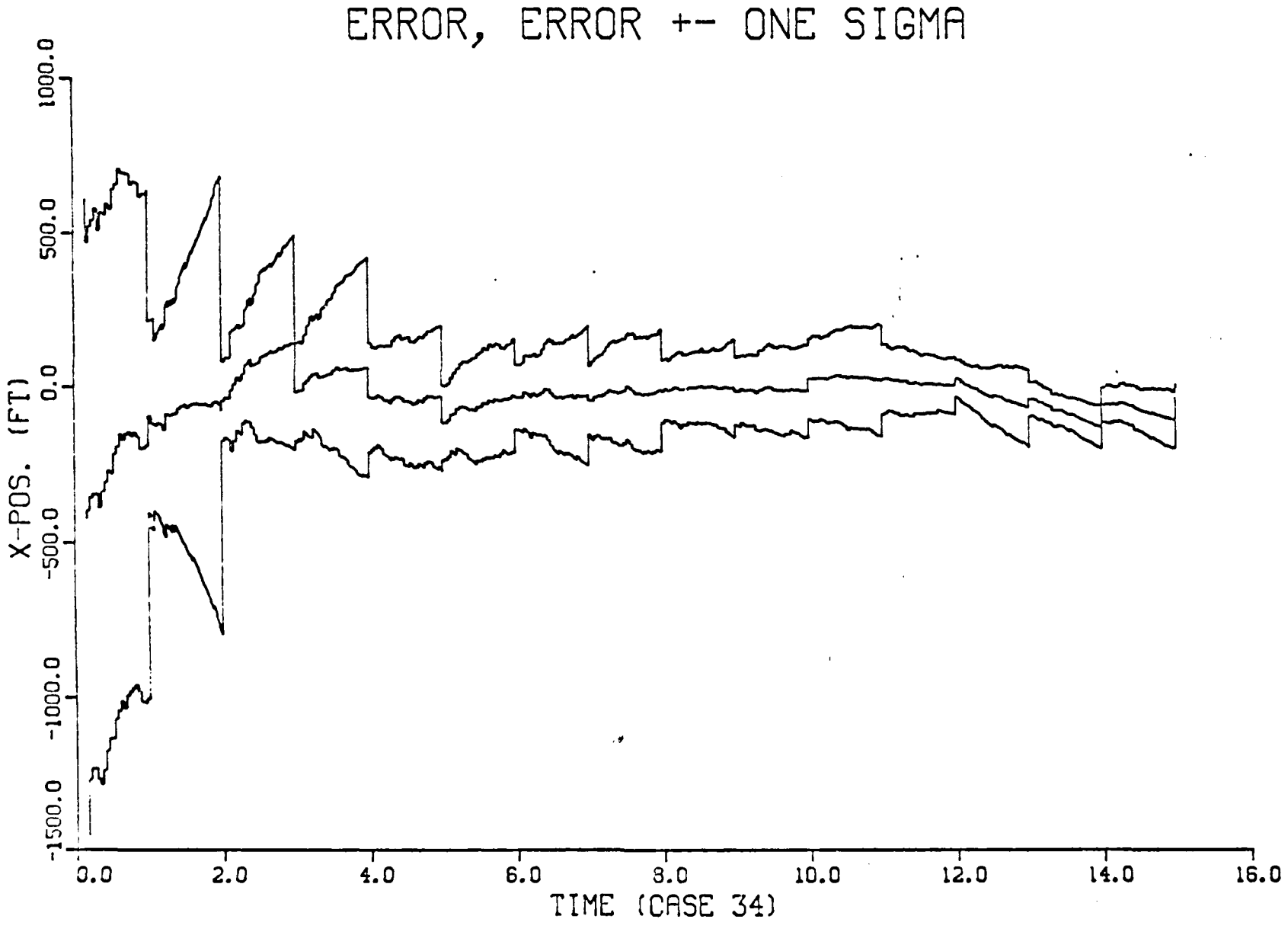

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
-200.0
-250.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 33)

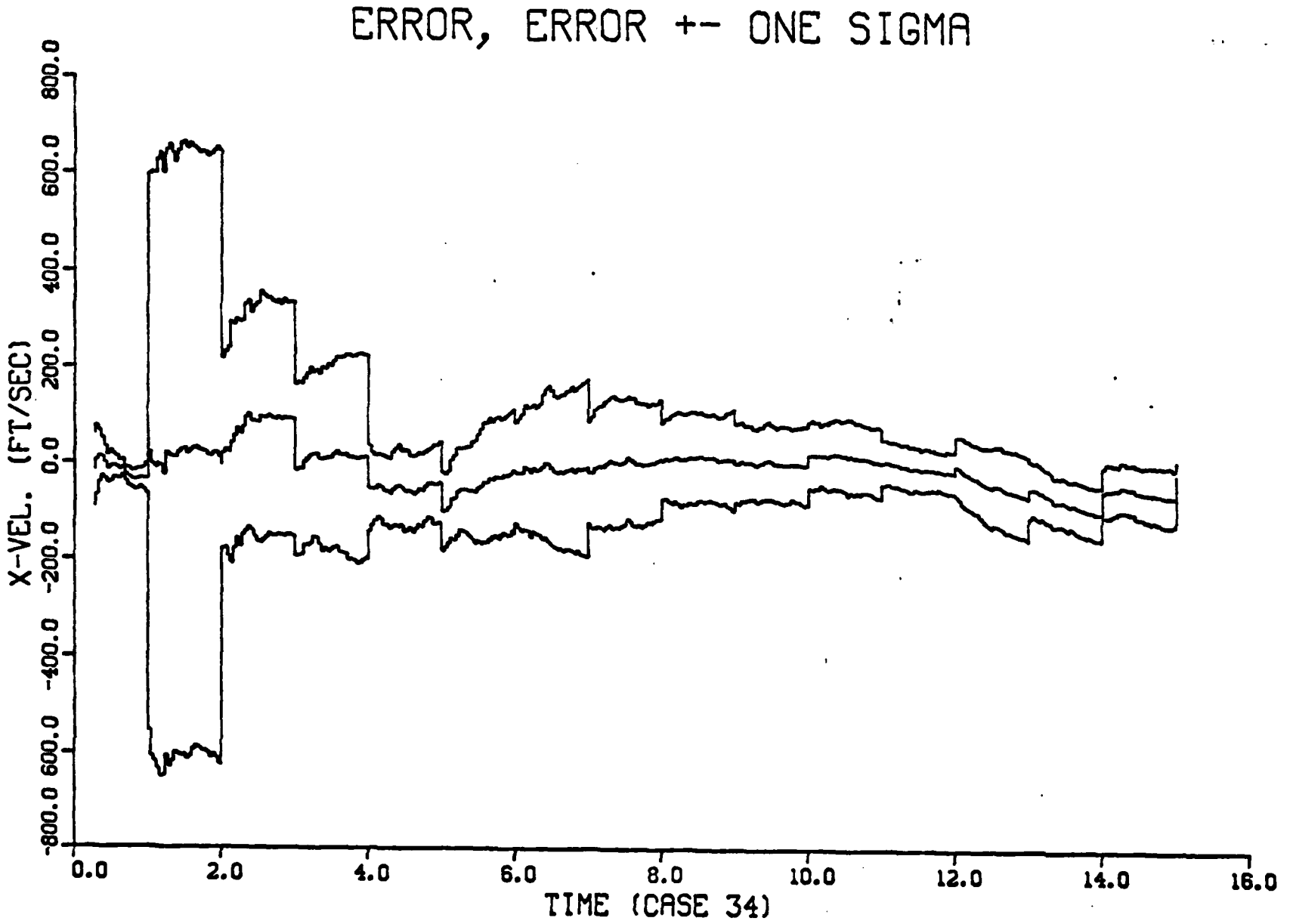

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
500.0
0.0
-500.0
-1000.0
-1500.0
-2000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 33)

2.2.1.1

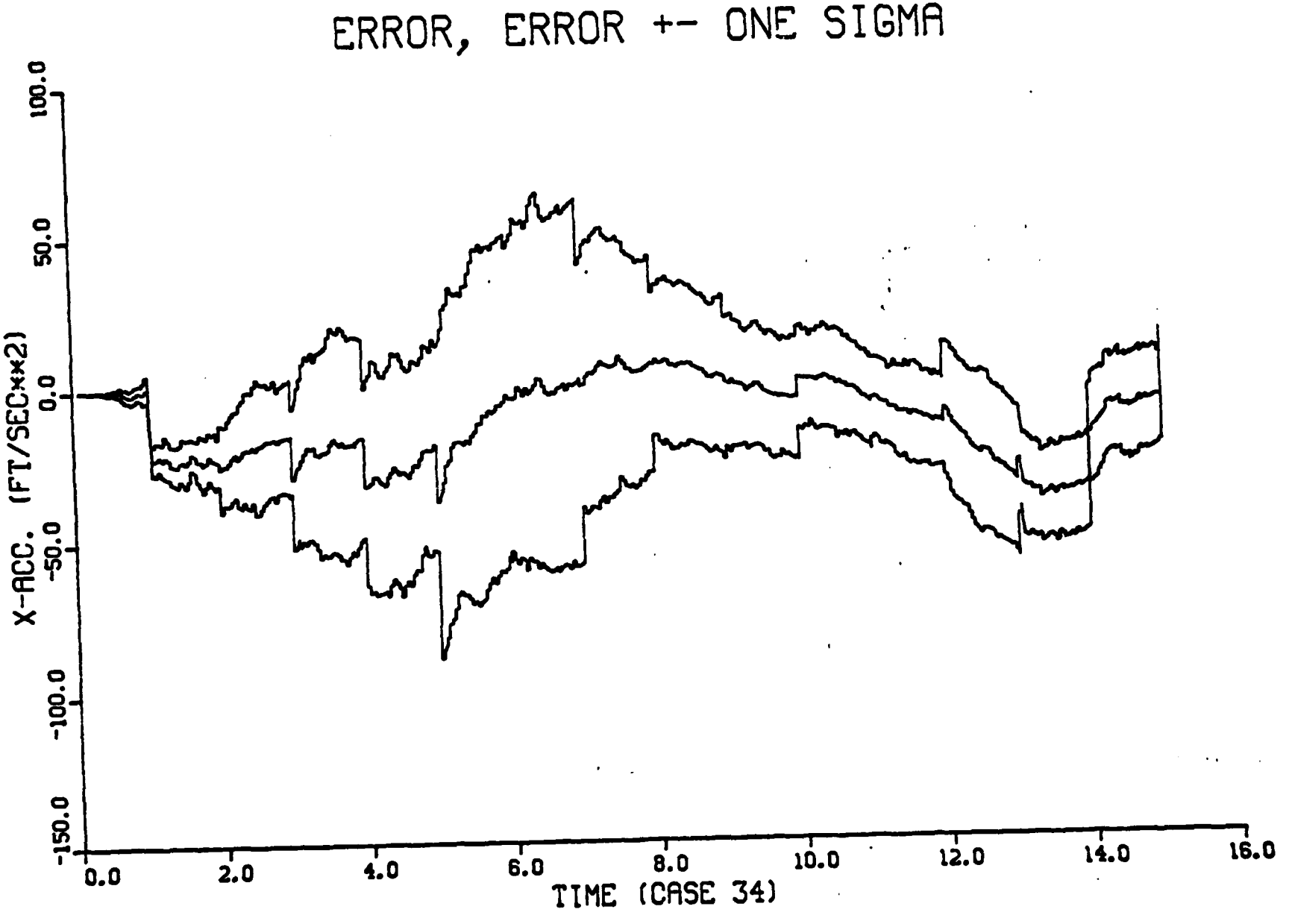

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 33)

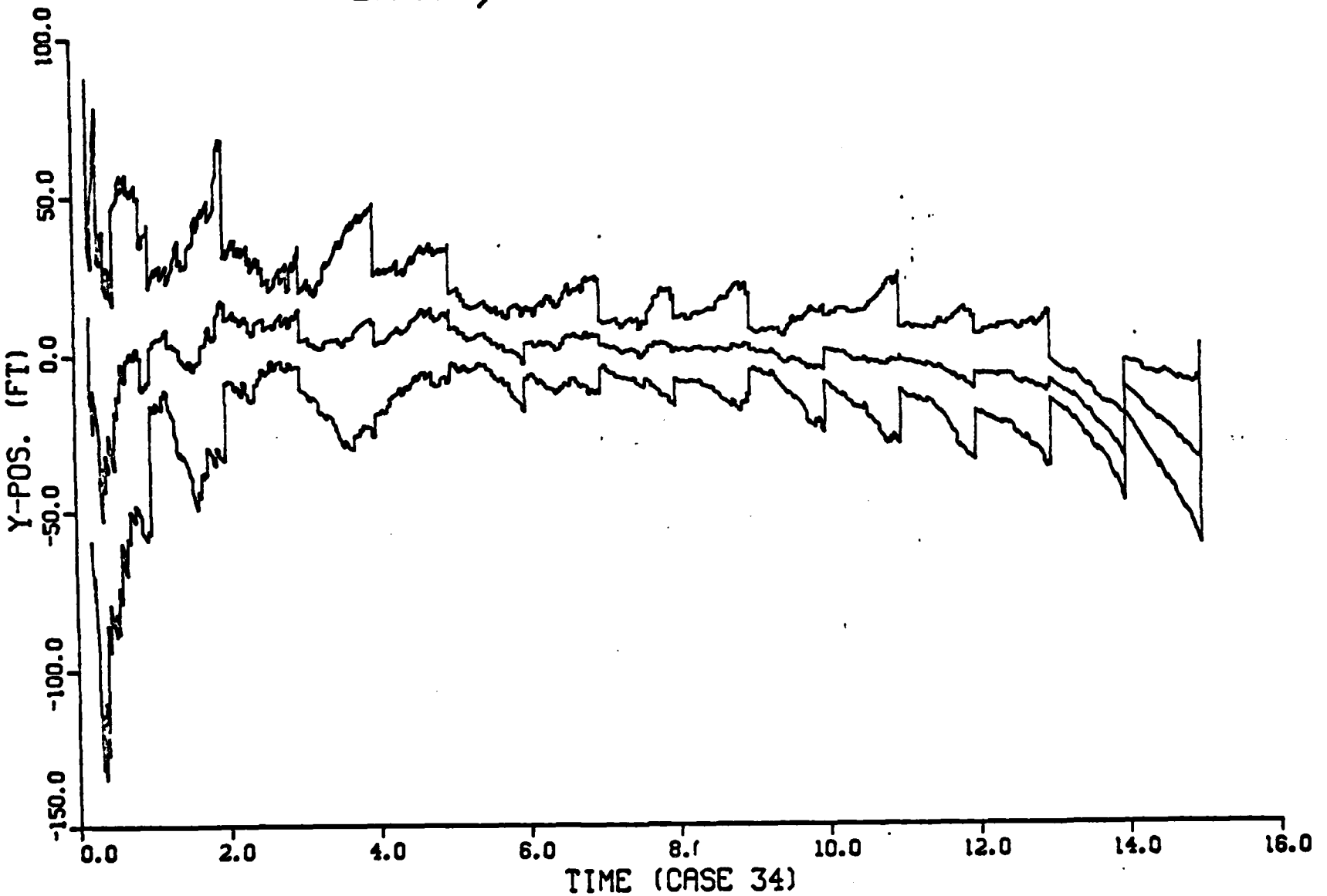
2.2.1.1

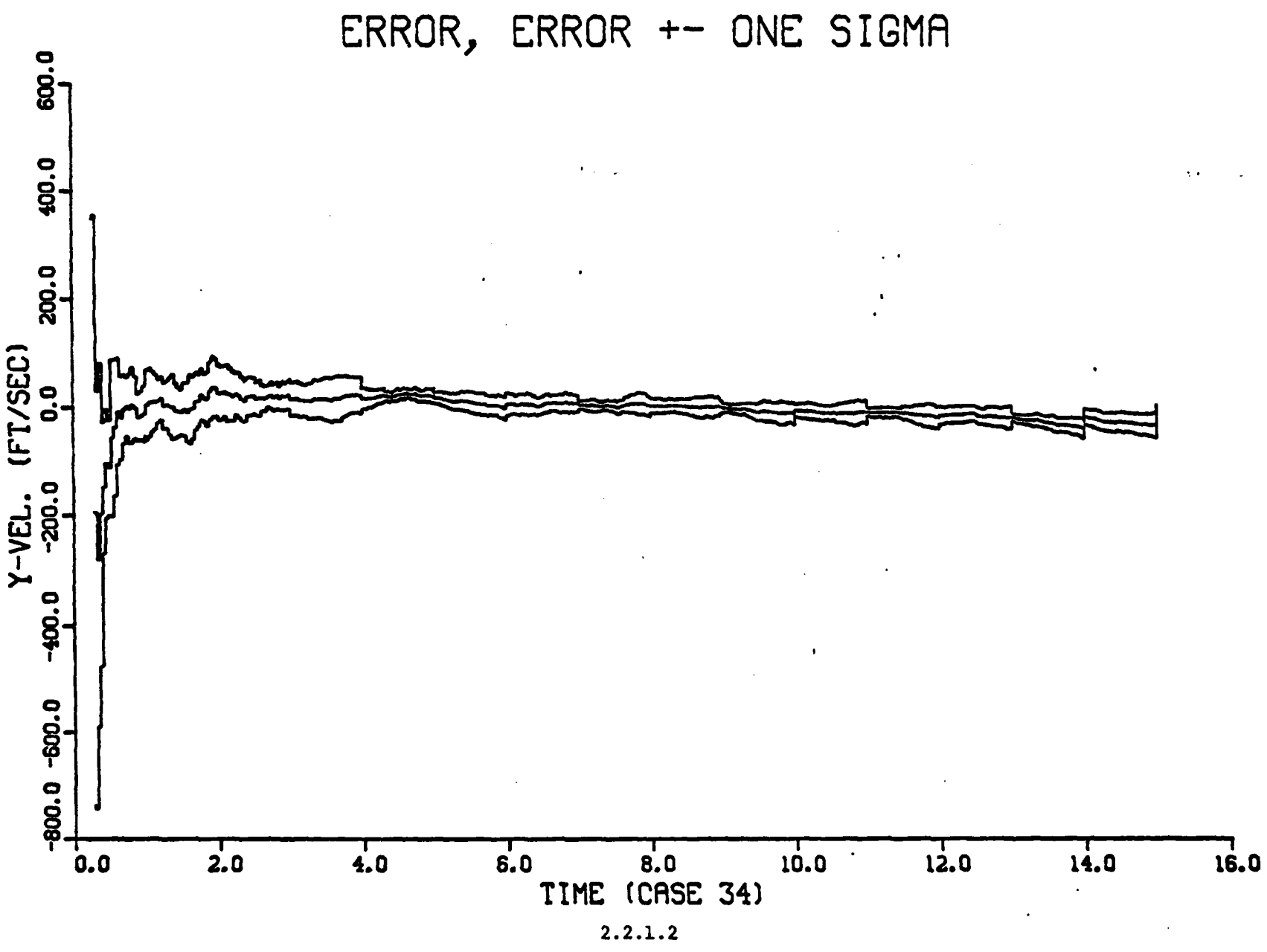

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
800.0
600.0
400.0
200.0
0.0
-200.0
-400.0
600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)

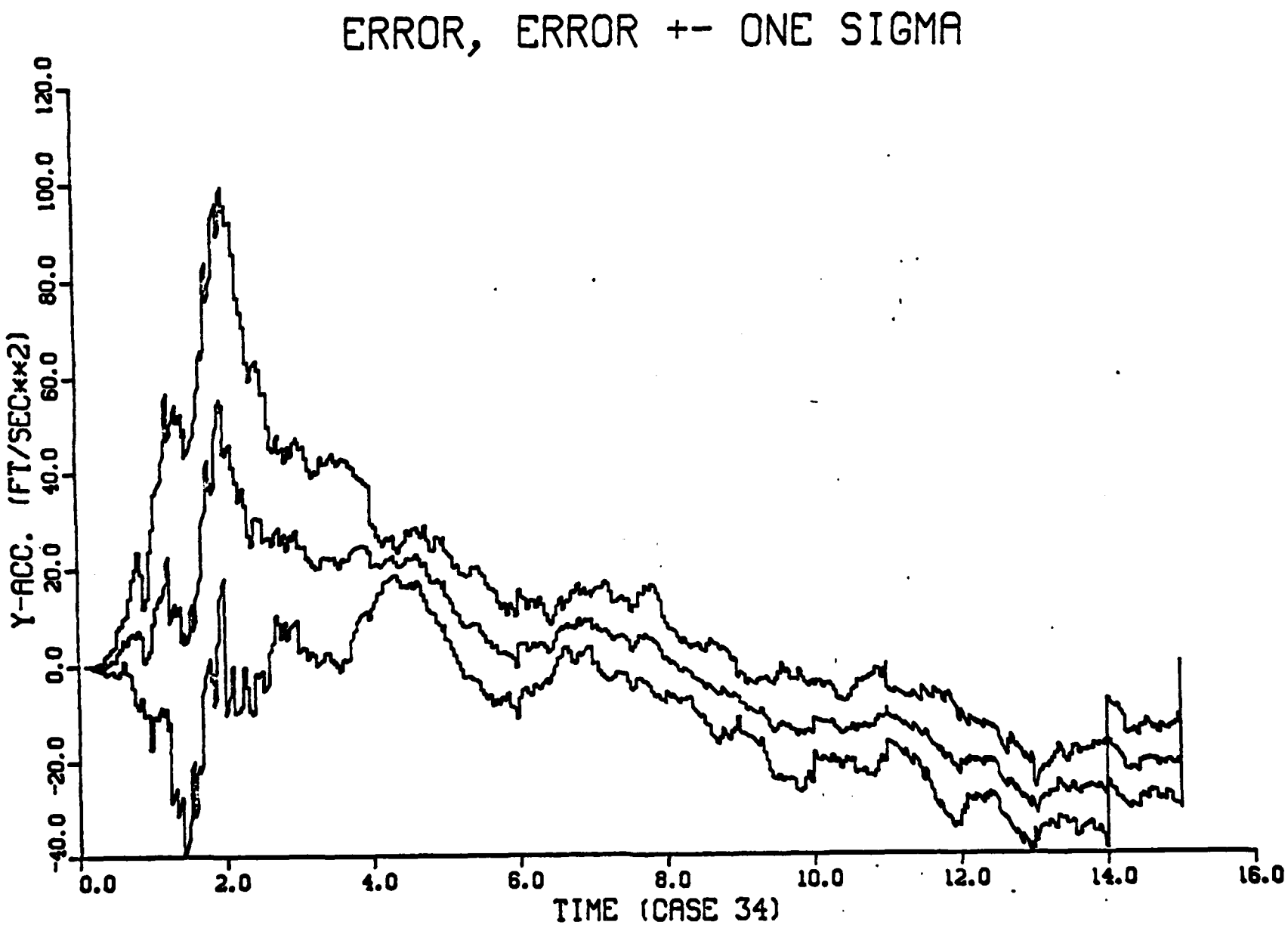
2.2.1.2

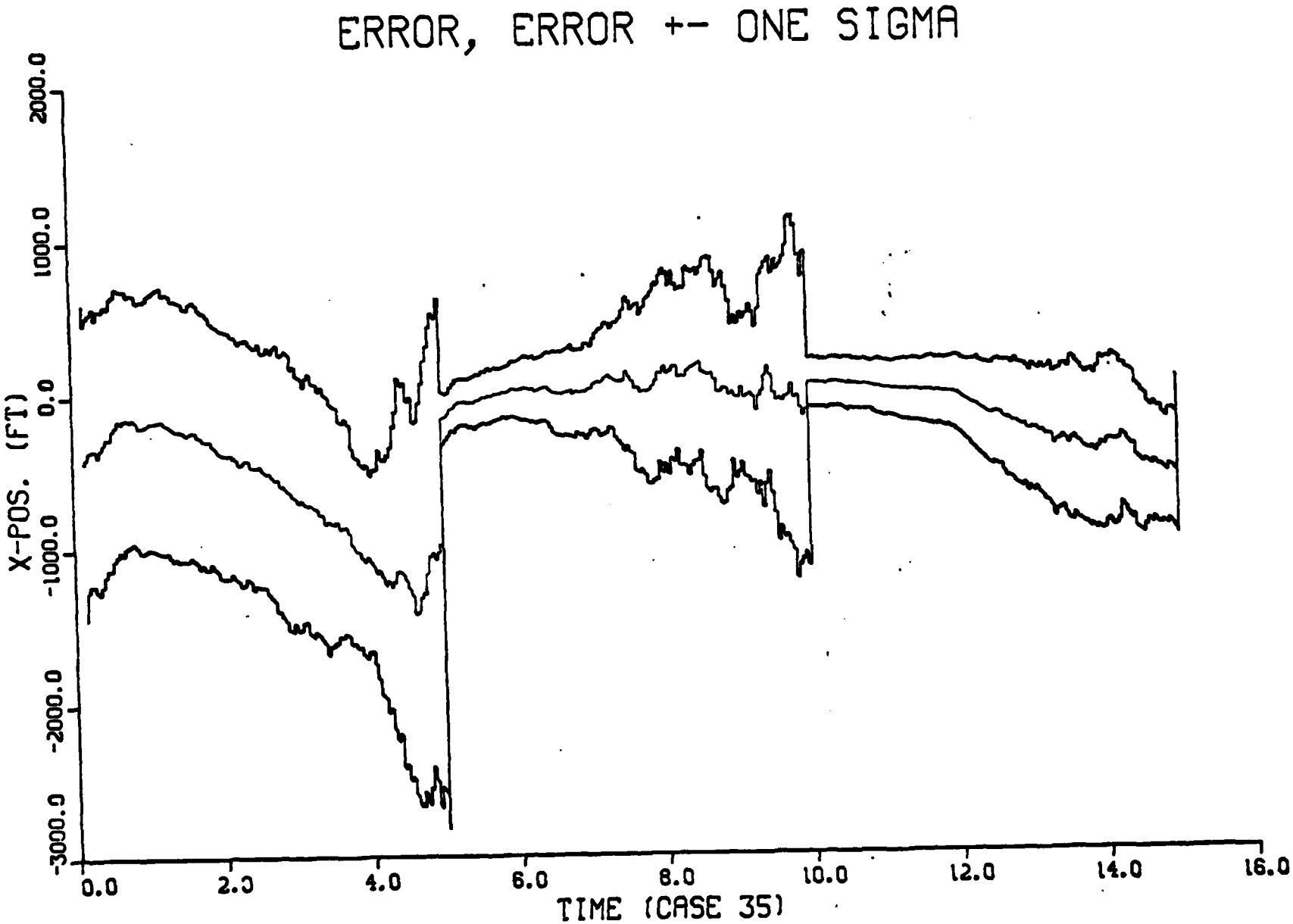

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)

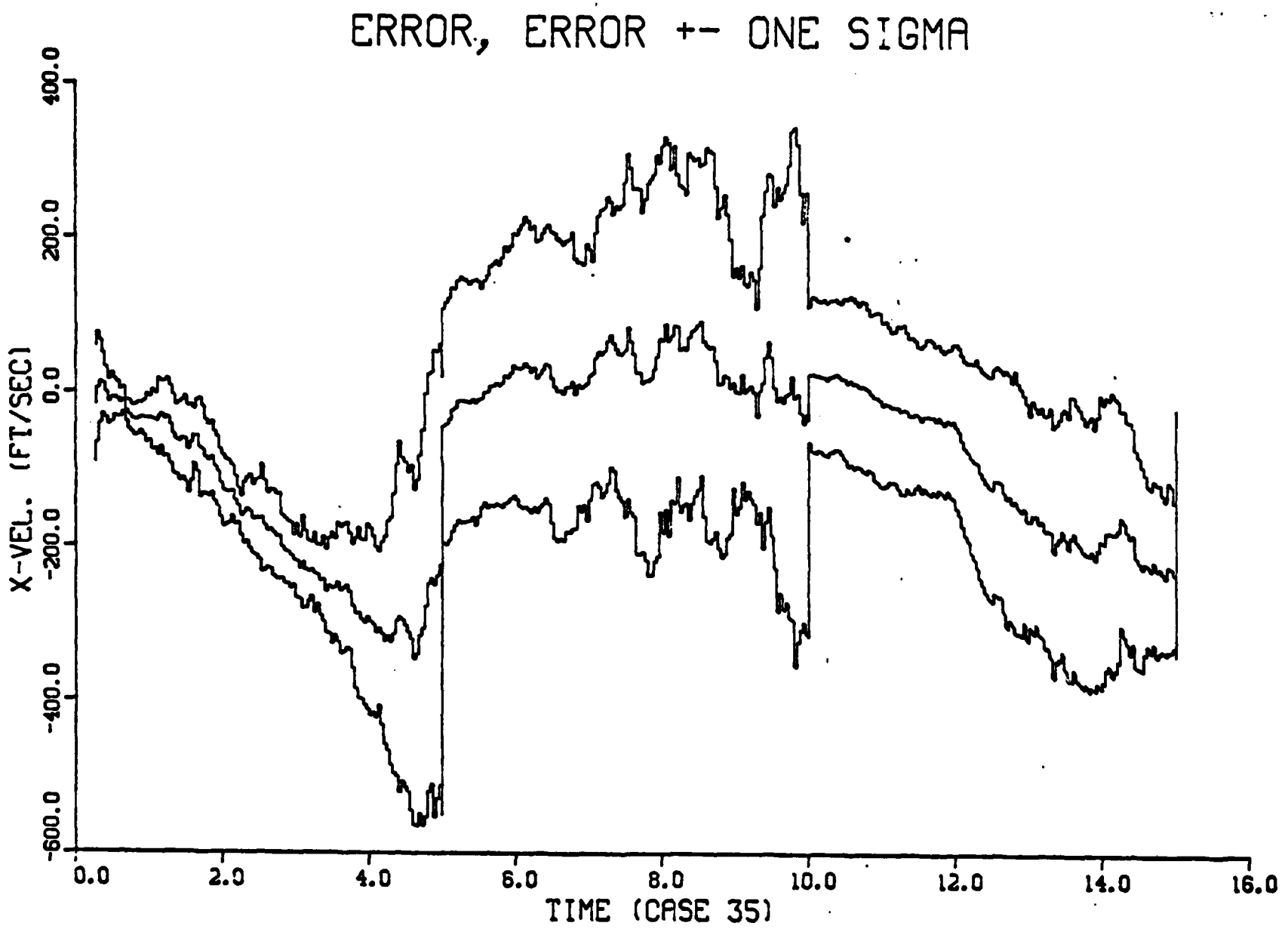

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)

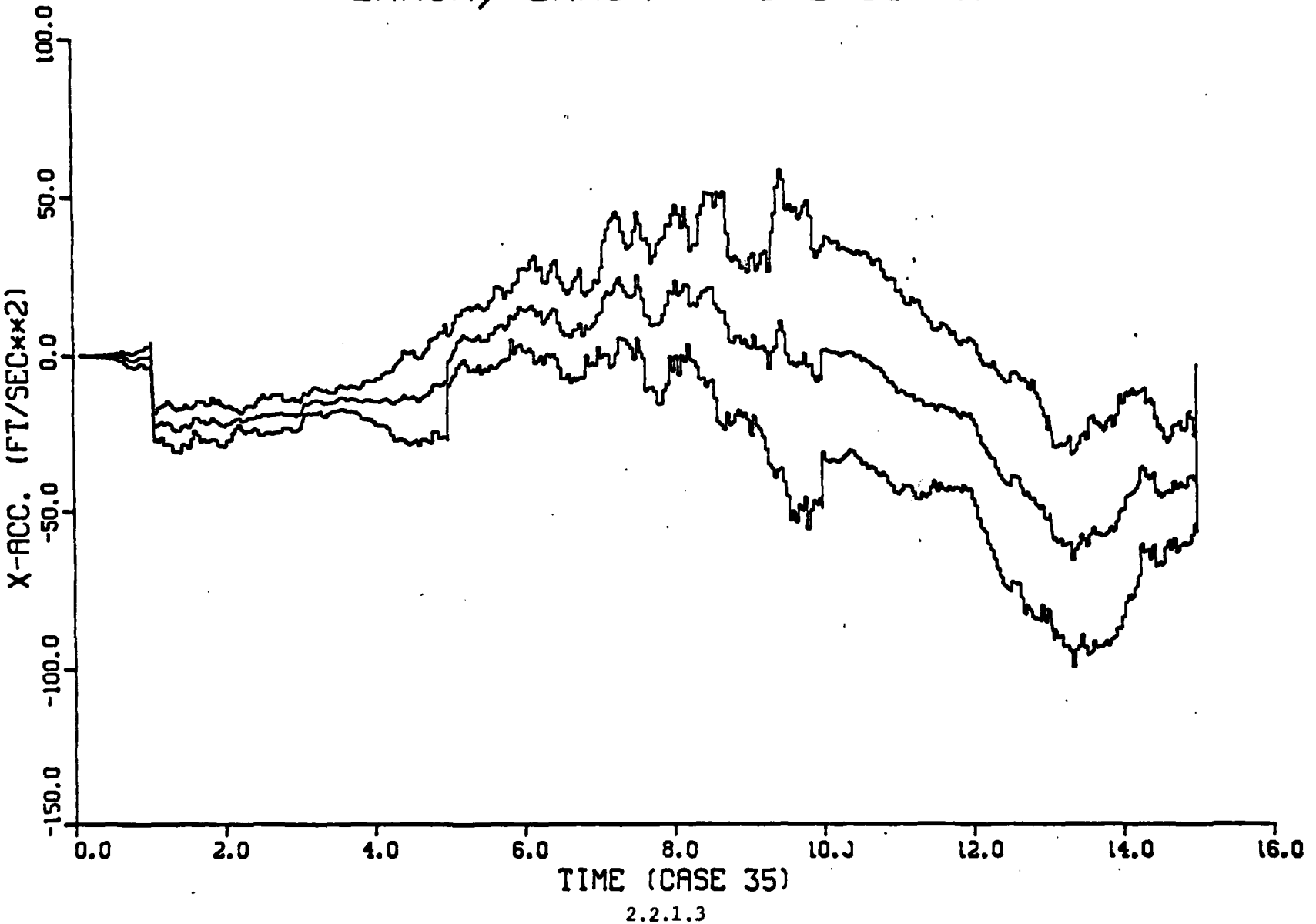

ERROR, ERROR +- ONE SIGMA
Y-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)

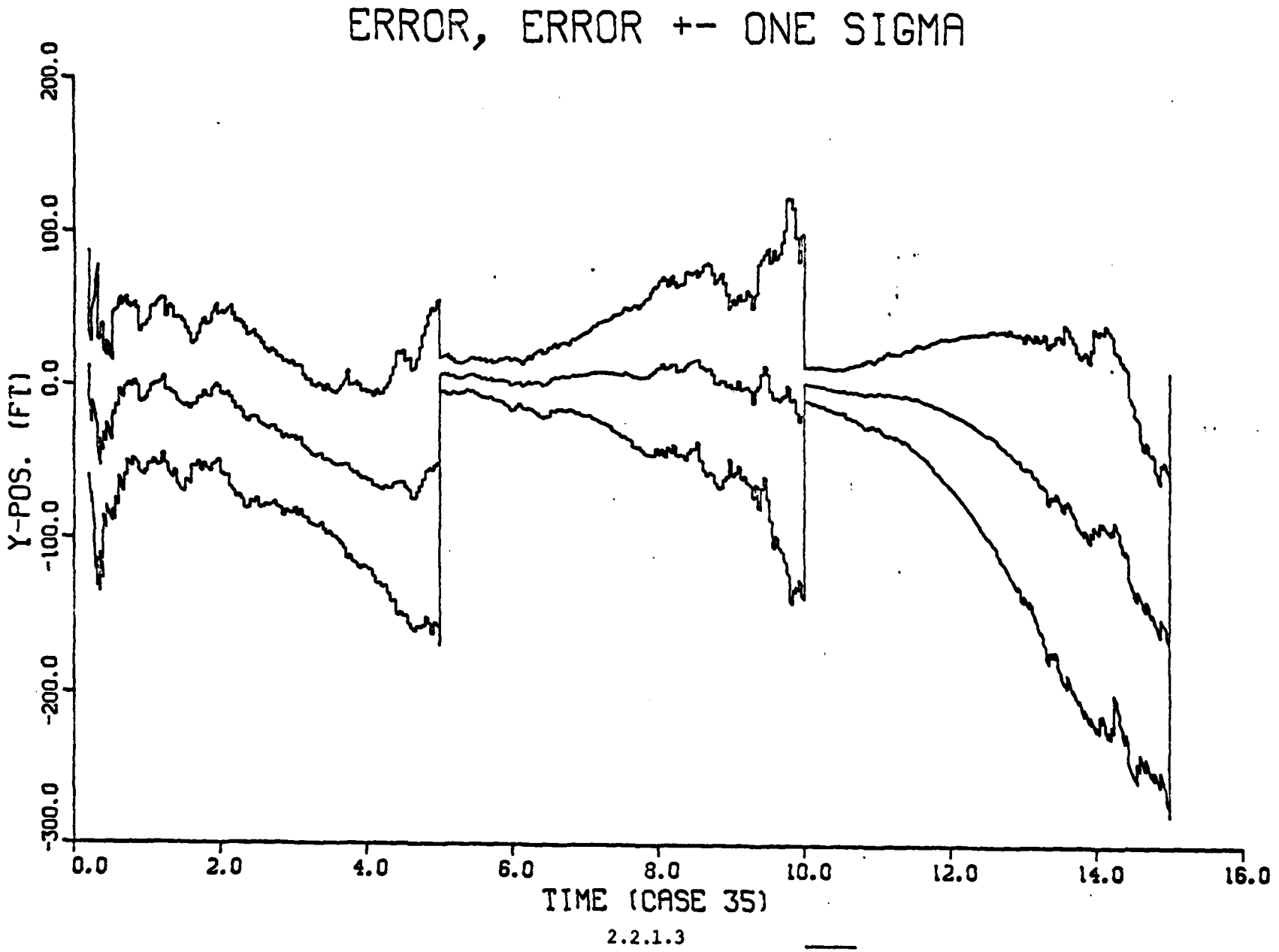
2.2.1.2

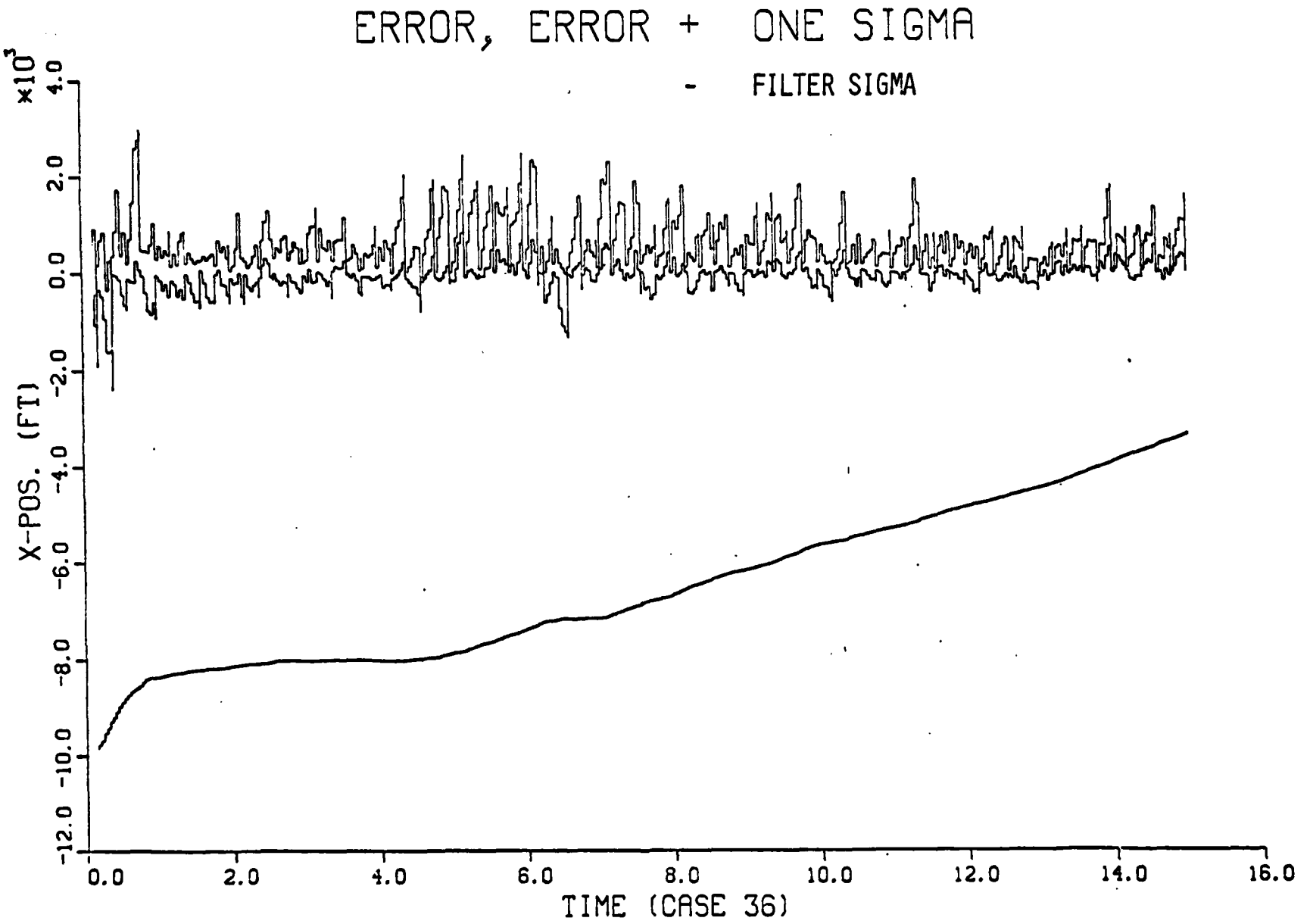

ERROR, ERROR +- ONE SIGMA
Y-ACC. (FT/SEC**2)
120.0
100.0
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 34)

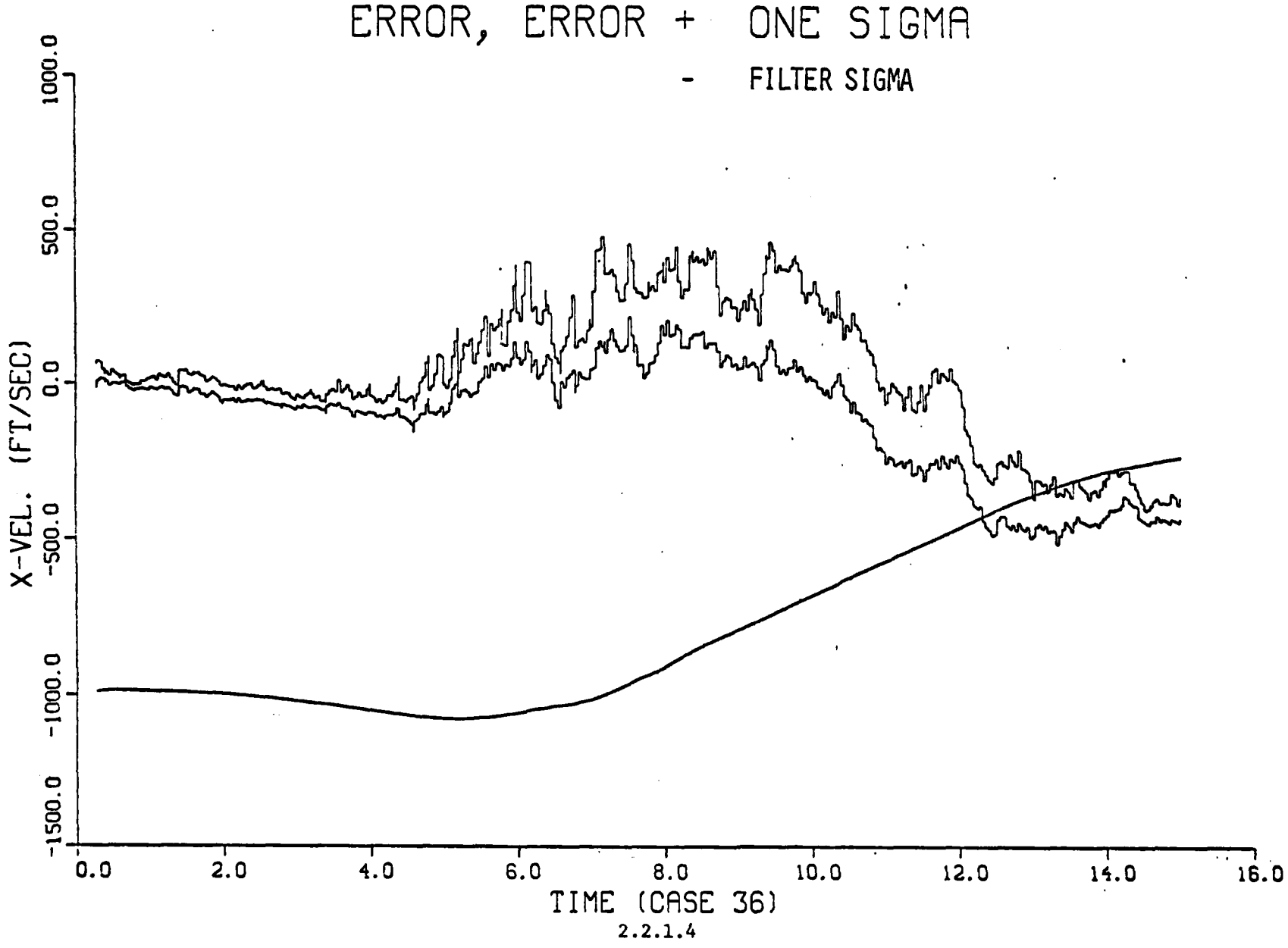

ERROR, ERROR +- ONE SIGMA
X-POS. (FT)
2000.0
1000.0
0.0
-1000.0
-2000.0
-3000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 35)


ERROR, ERROR +- ONE SIGMA
X-VEL. (FT/SEC)
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 35)

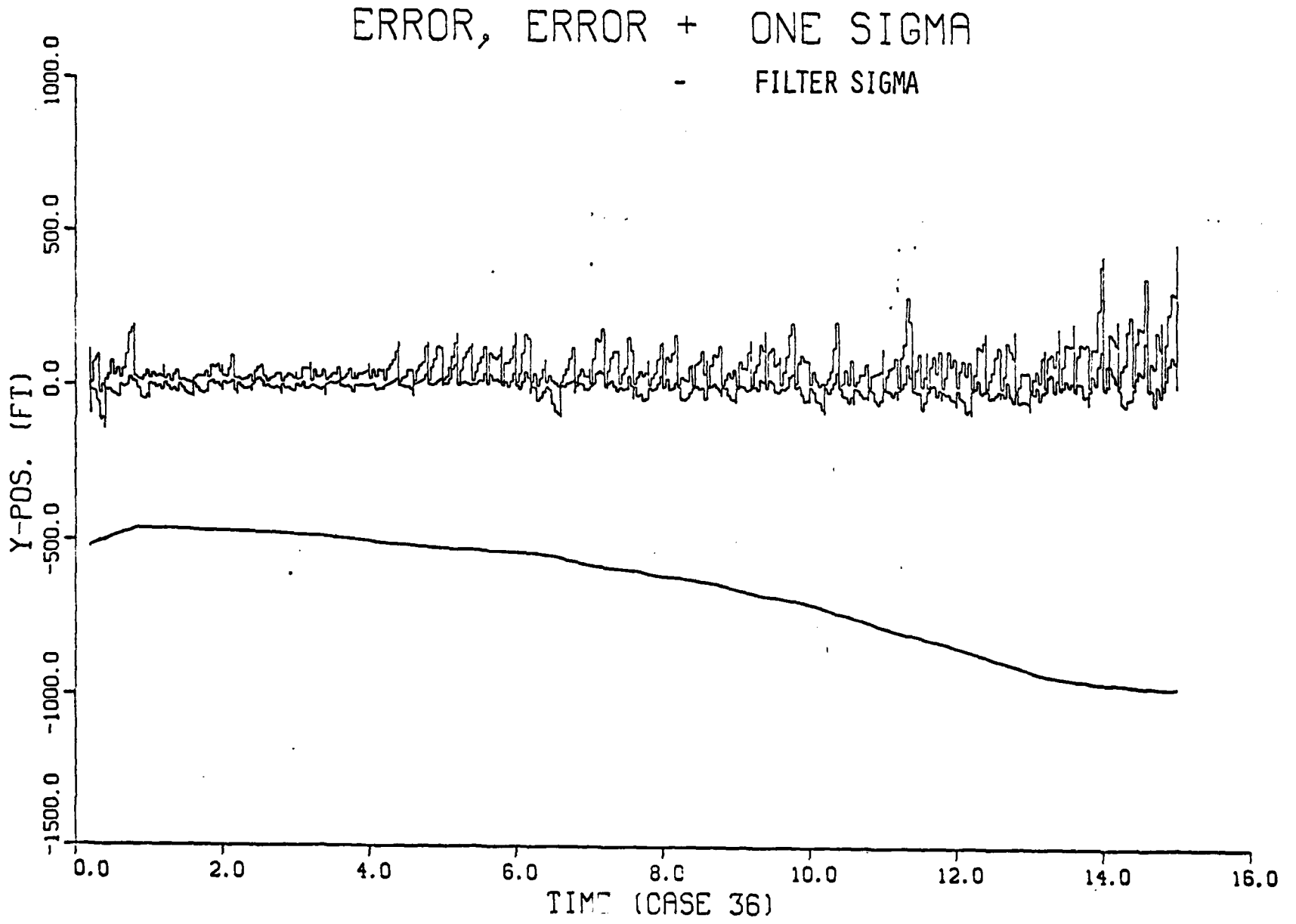

ERROR, ERROR +- ONE SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 35)
2.2.1.3

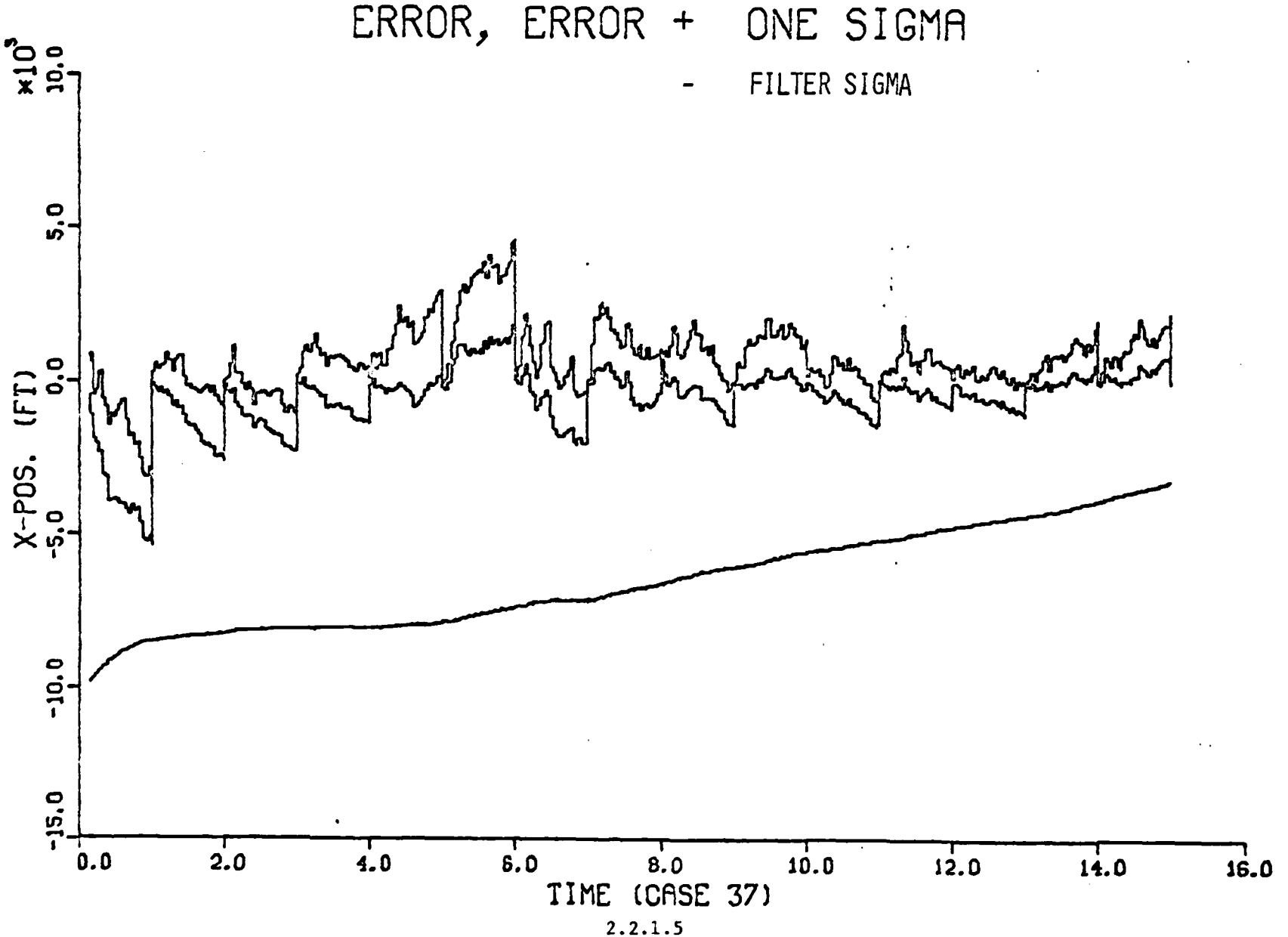

ERROR, ERROR +- ONE SIGMA
Y-POS. (FT)
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 35)
2.2.1.3


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
×10³
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
-12.0
X-POS. (FT)
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 36)

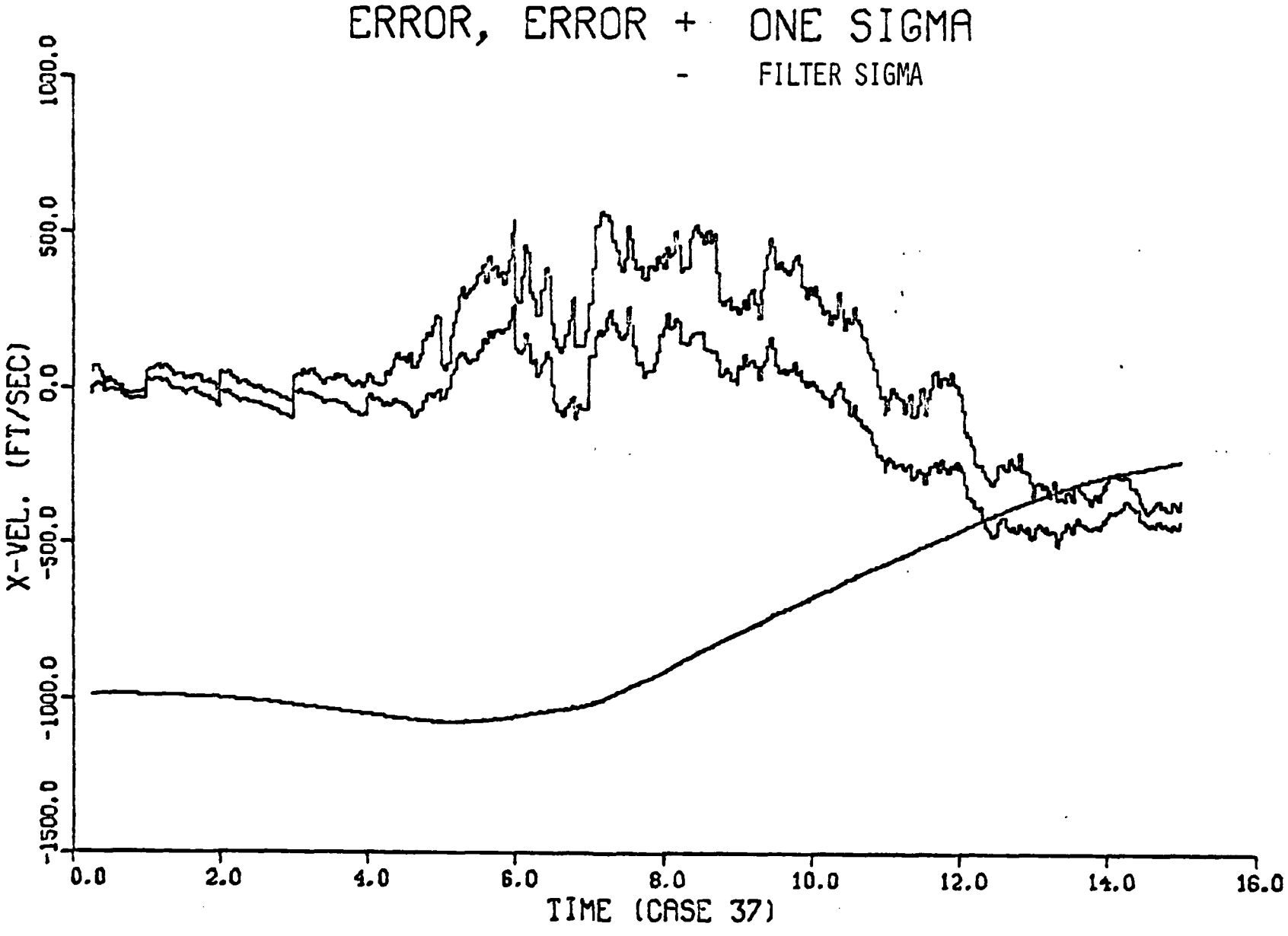

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 36)

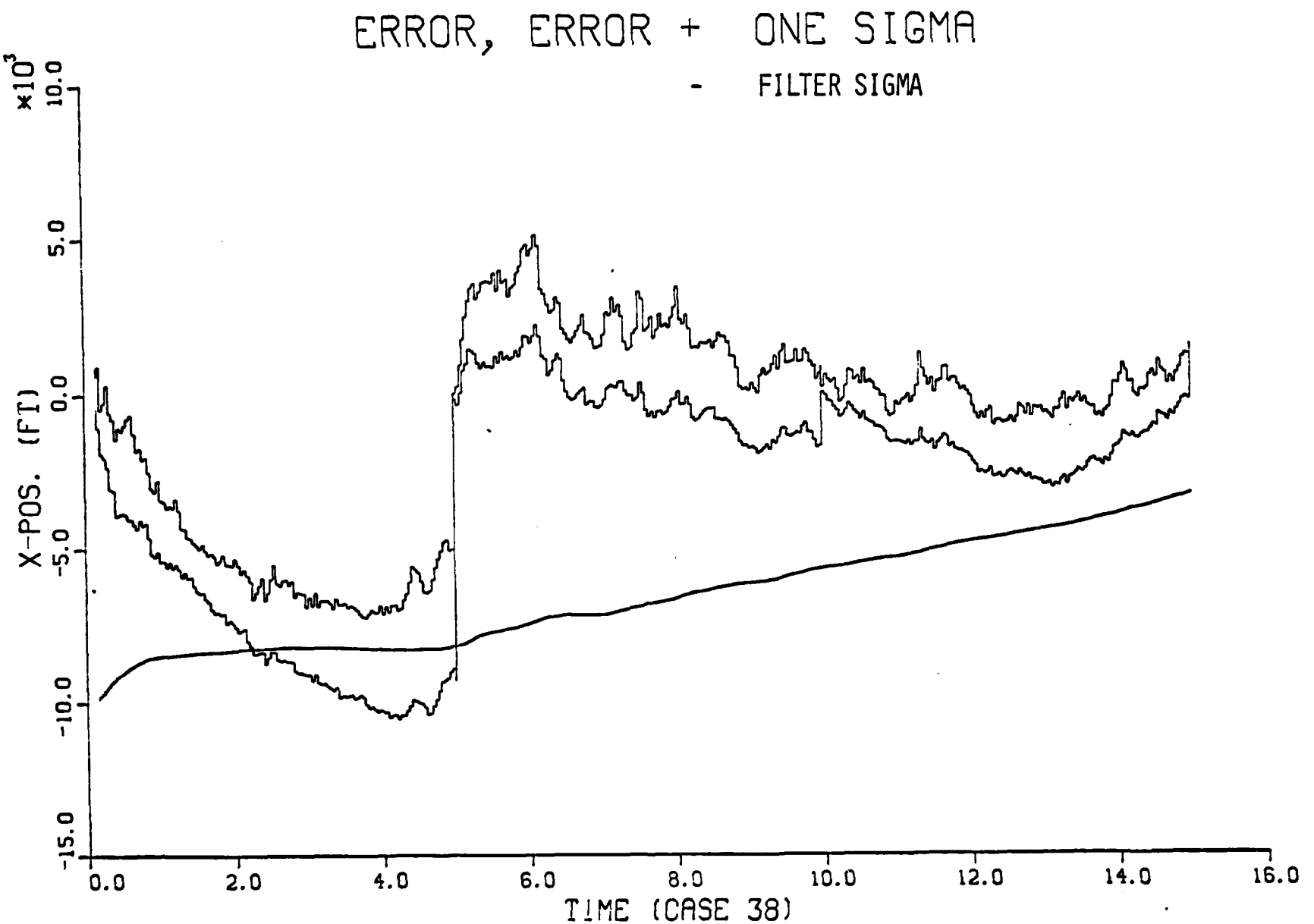
2.2.1.4


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 36)

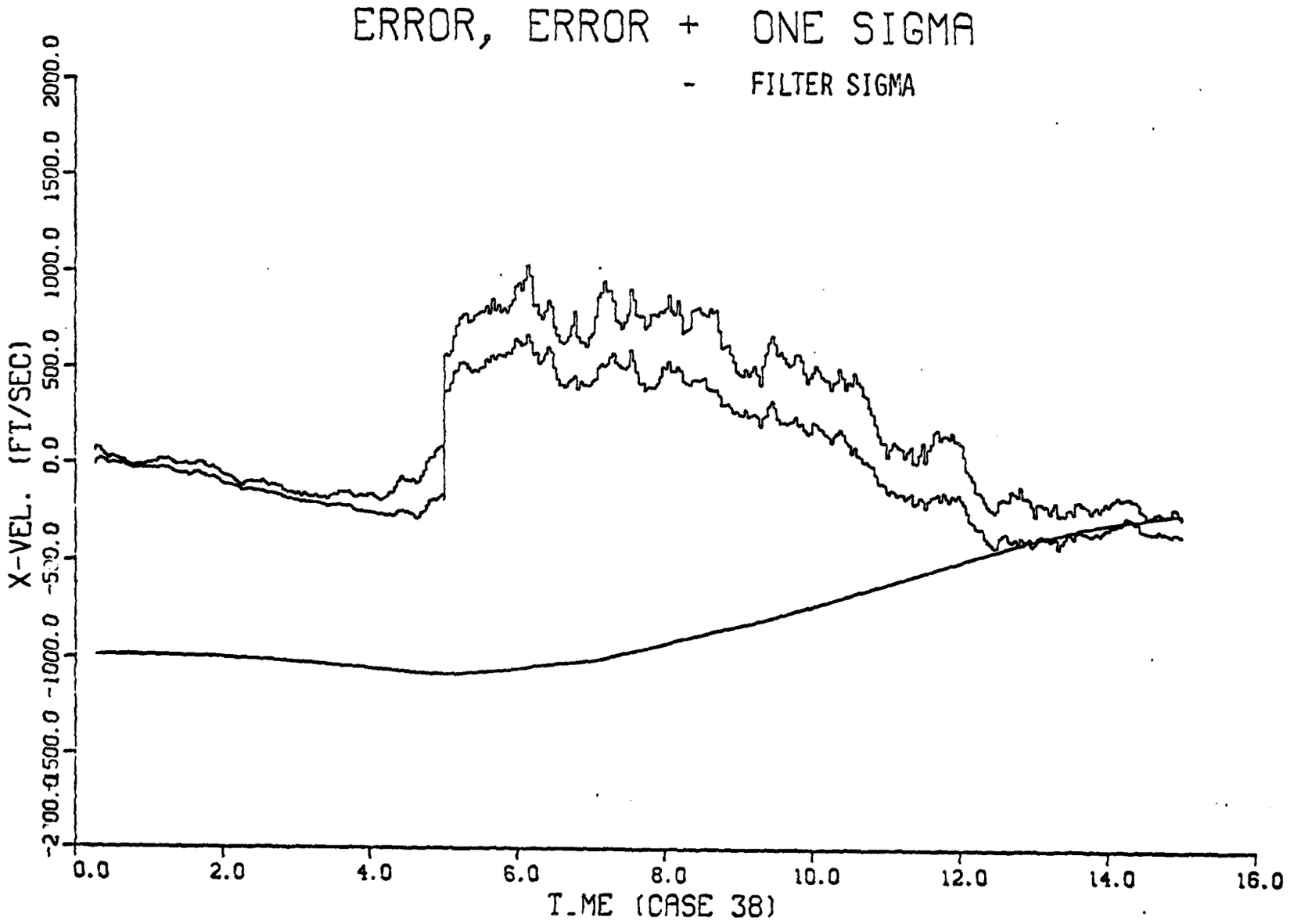

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
×10³
10.0
5.0
0.0
-5.0
-10.0
-15.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 37)

2.2.1.5

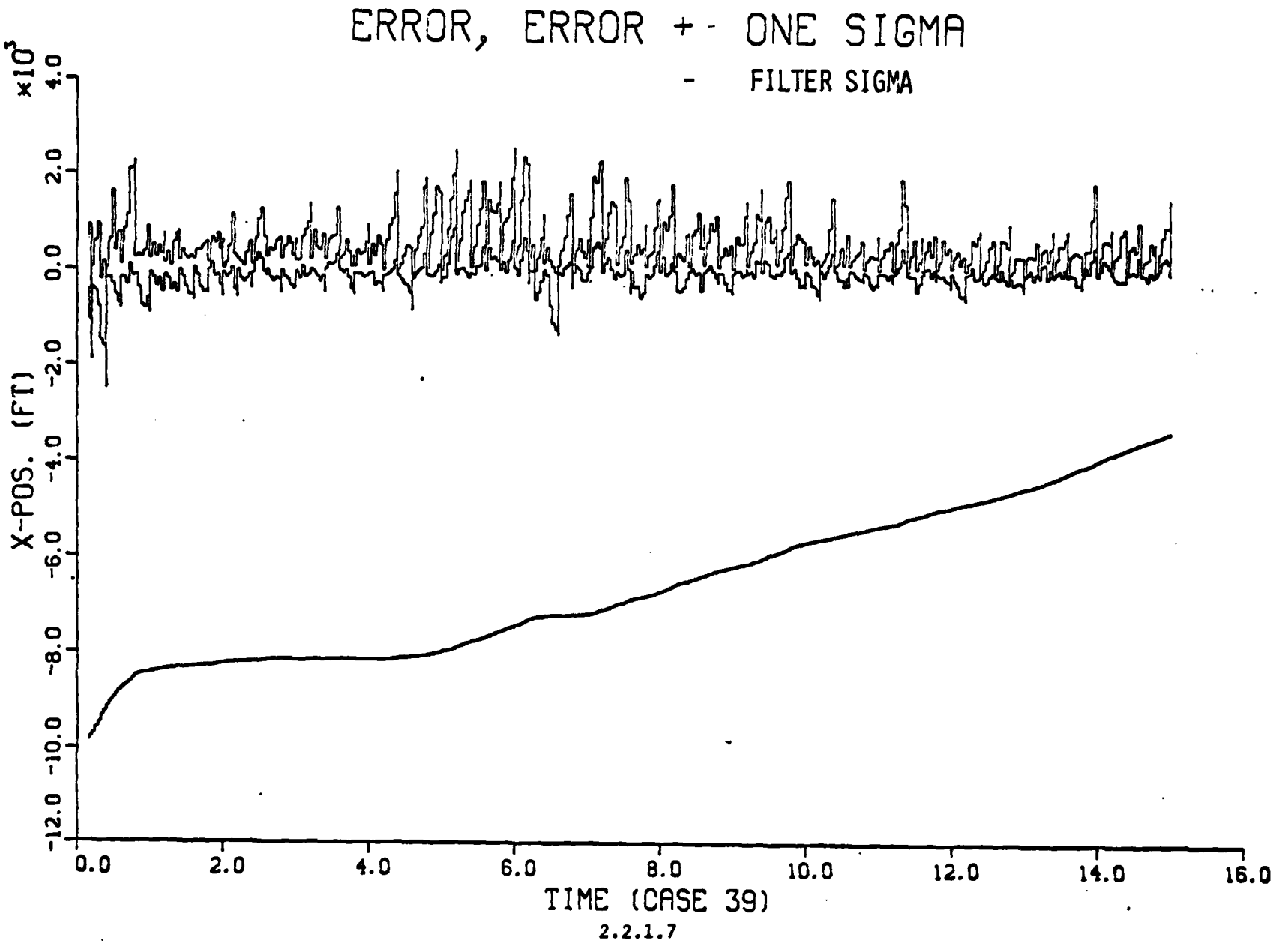

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 37)


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
*10^3
10.0
5.0
0.0
-5.0
-10.0
-15.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 38)

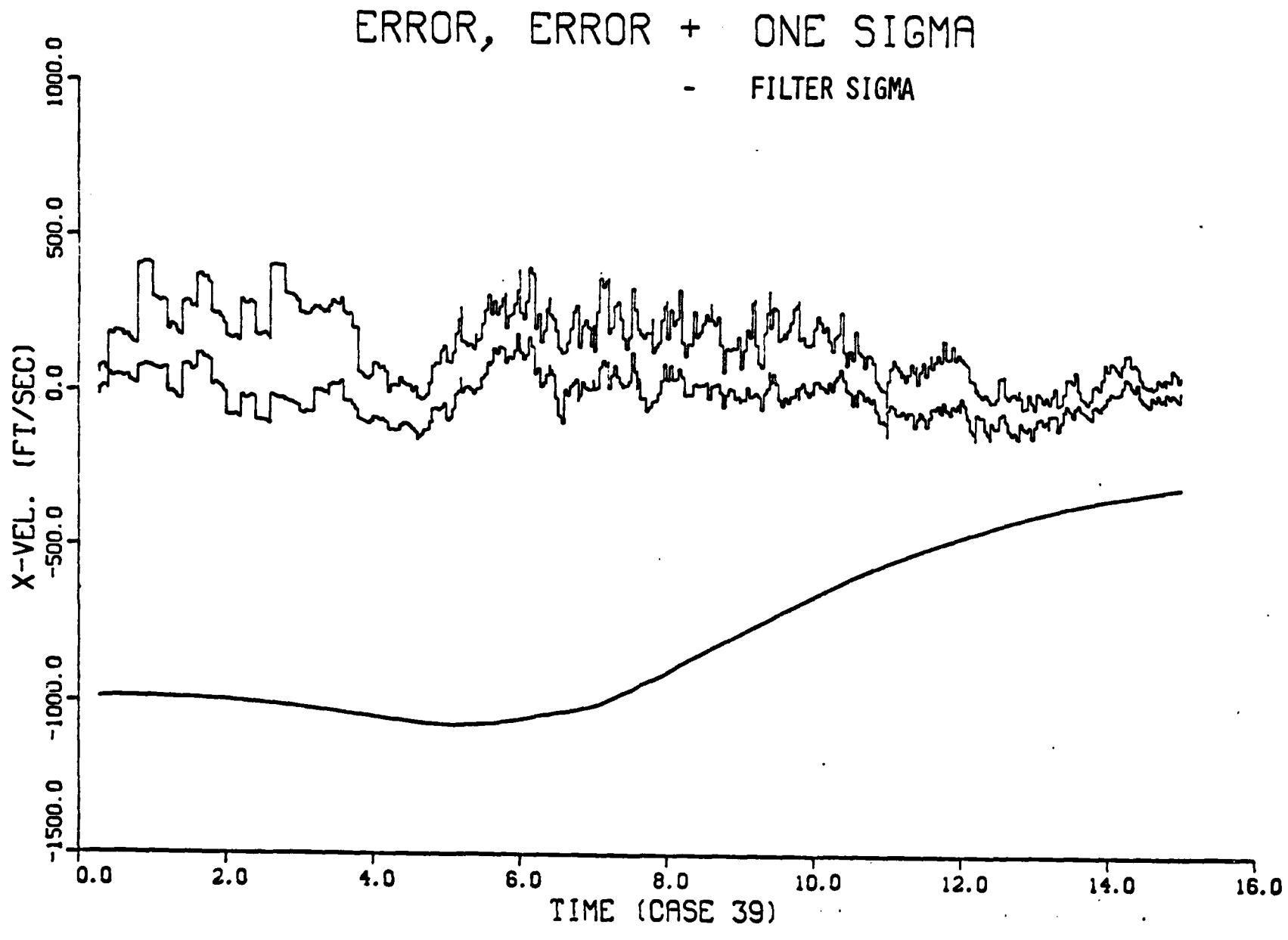
2.2.1.6

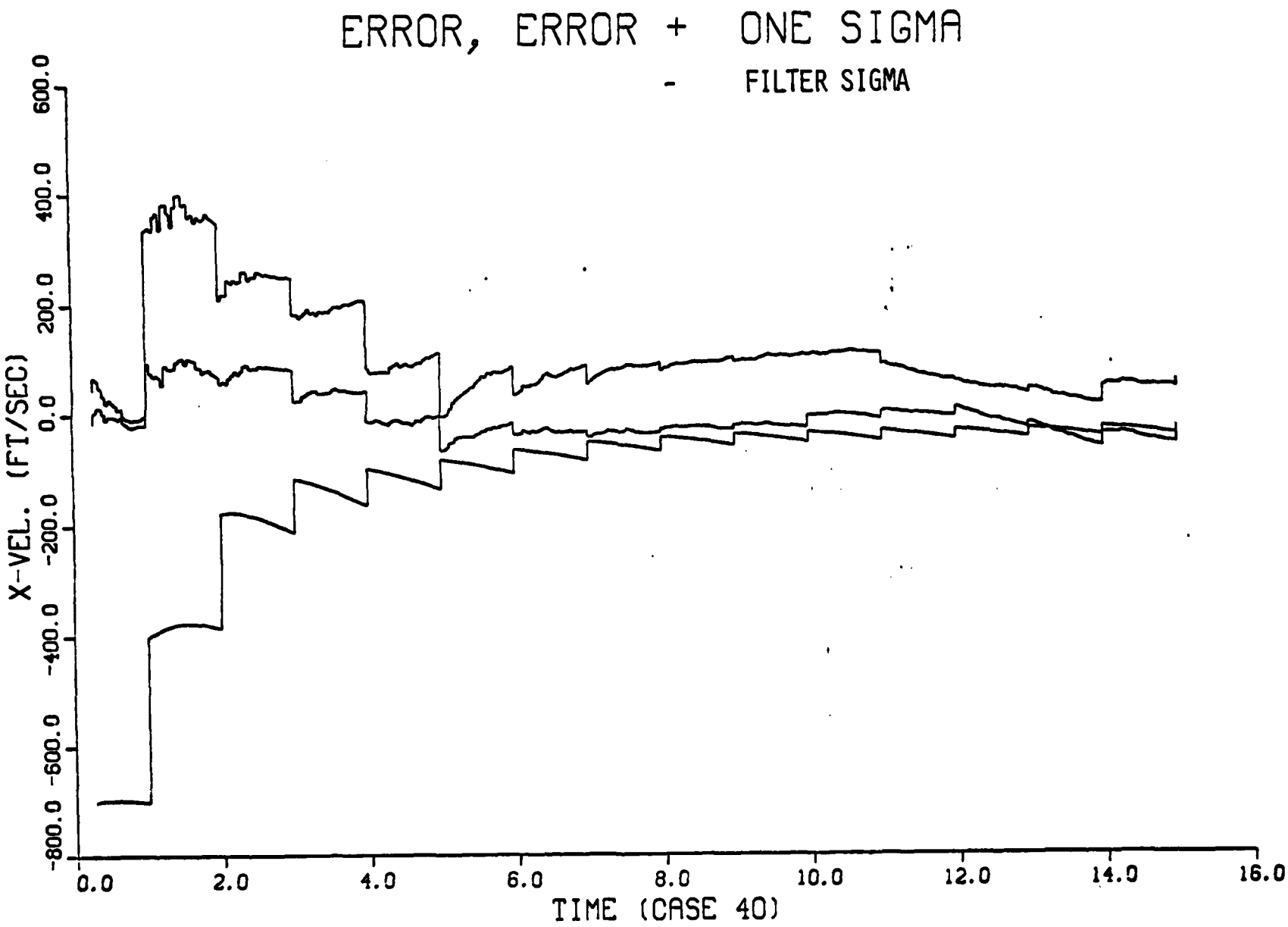

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
2000.0
1500.0
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
-2000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 38)

2.2.1.6

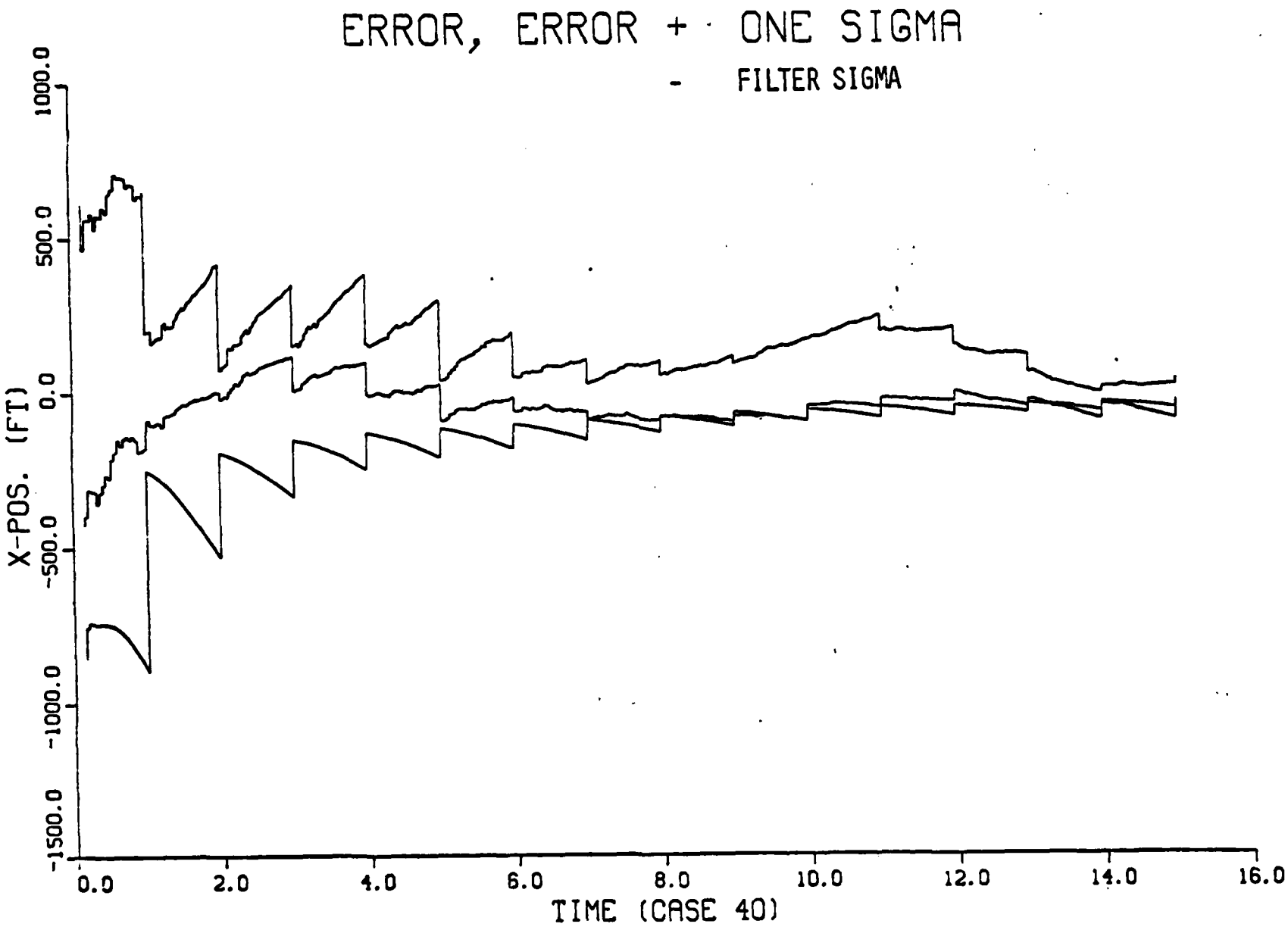

ERROR, ERROR + - ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
×10³
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
-12.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 39)

2.2.1.7

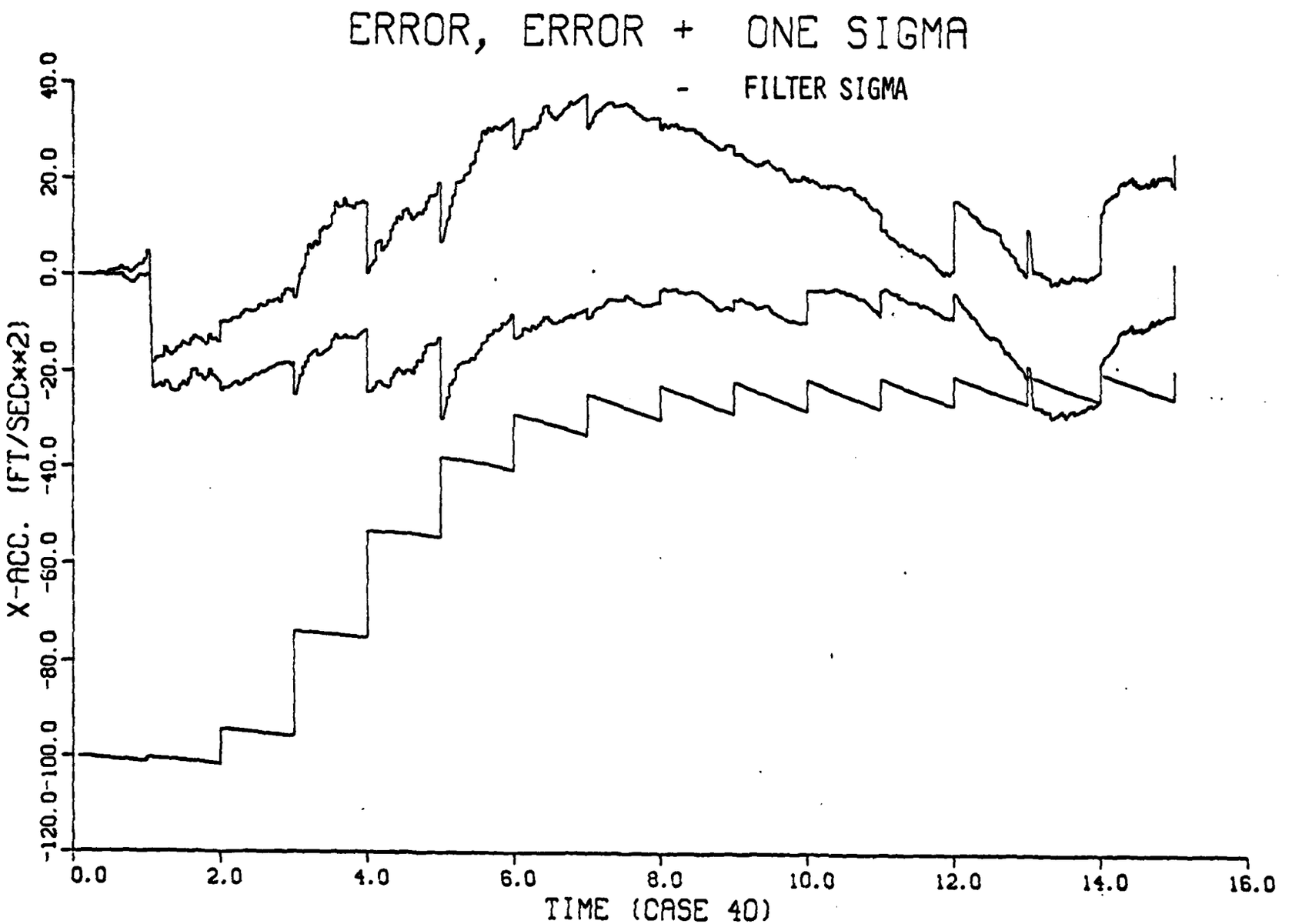

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 39)


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)

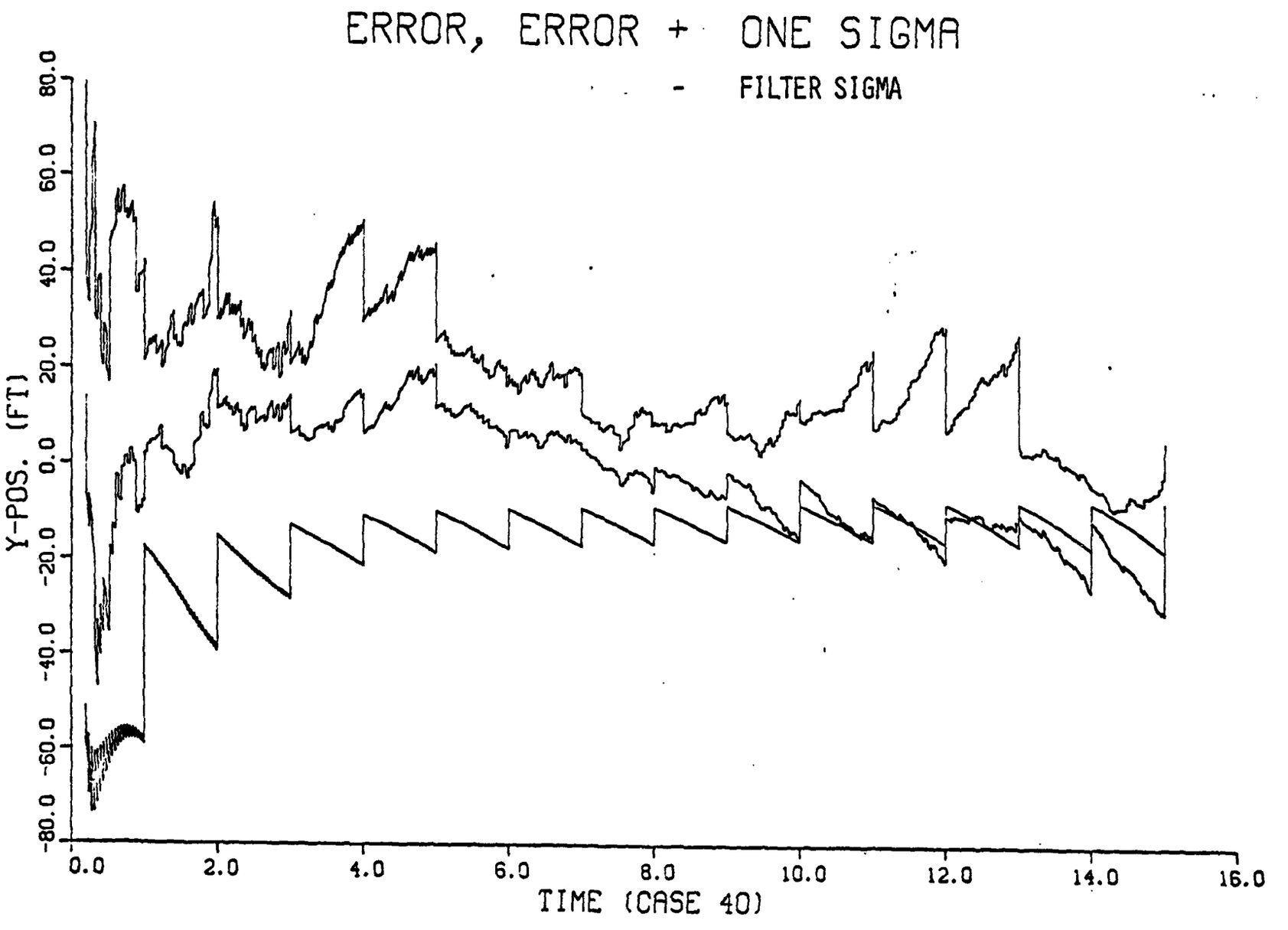
2.2.1.8


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)

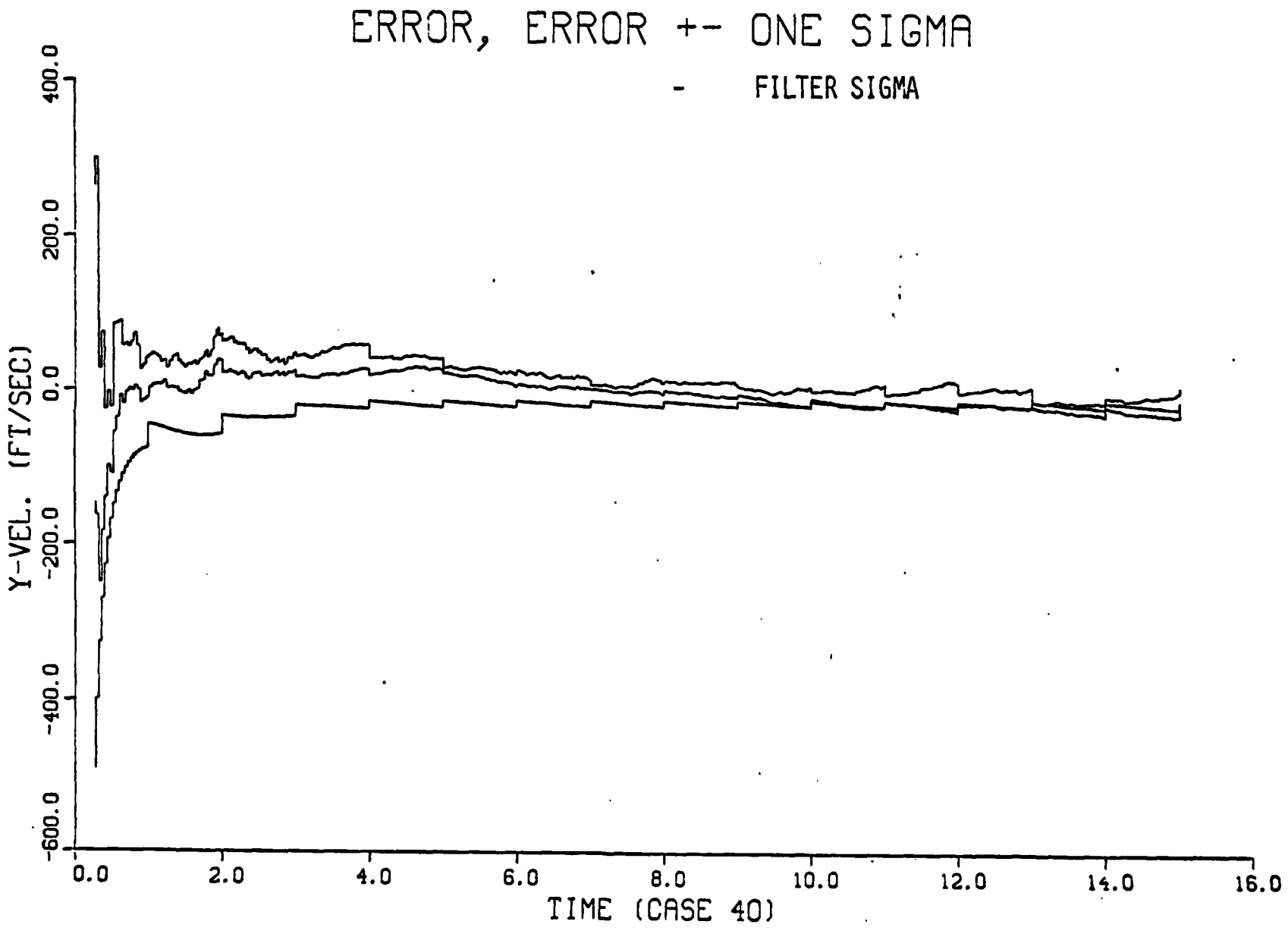
2.2.1.8


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)

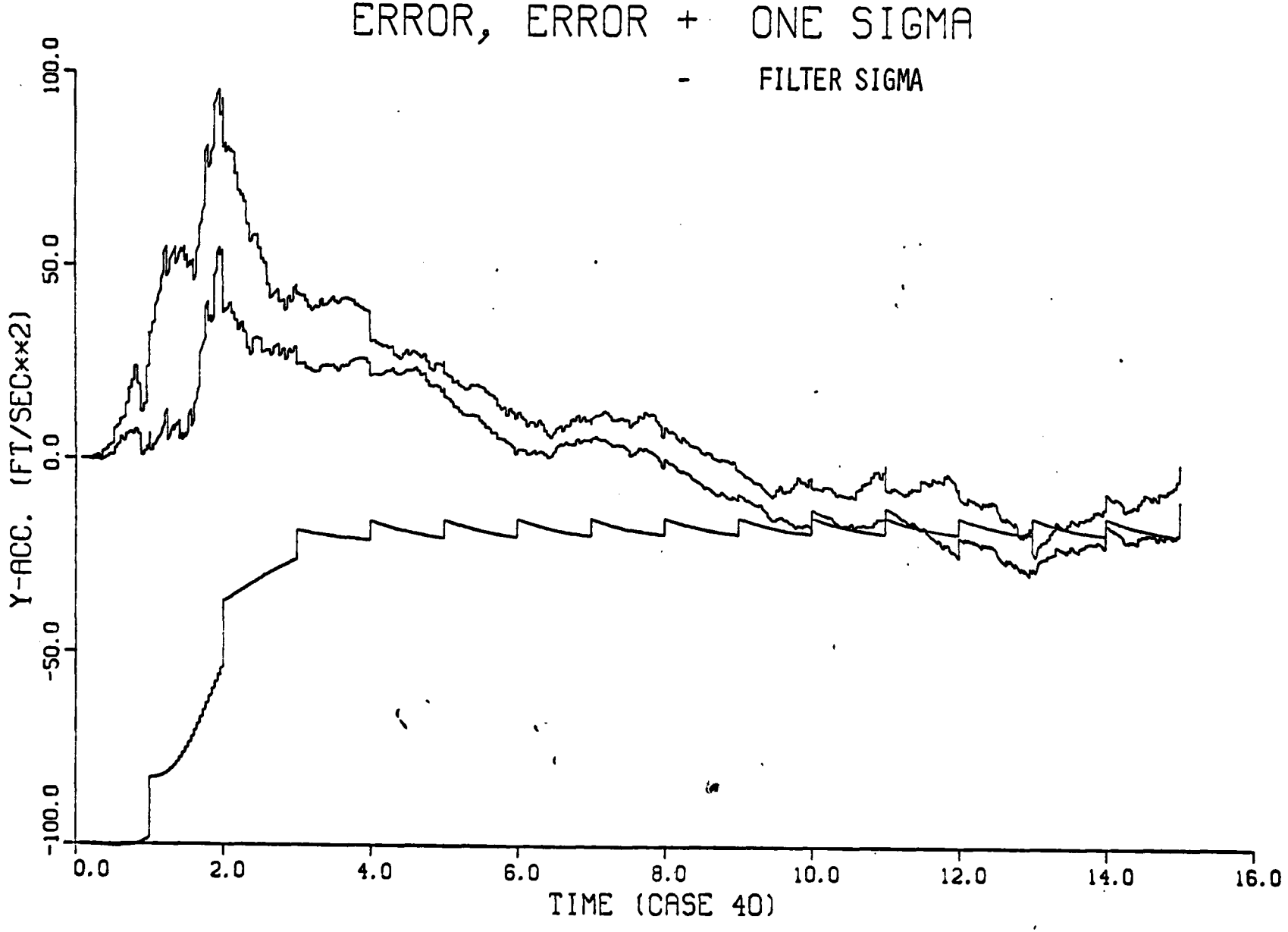
2.2.1.8

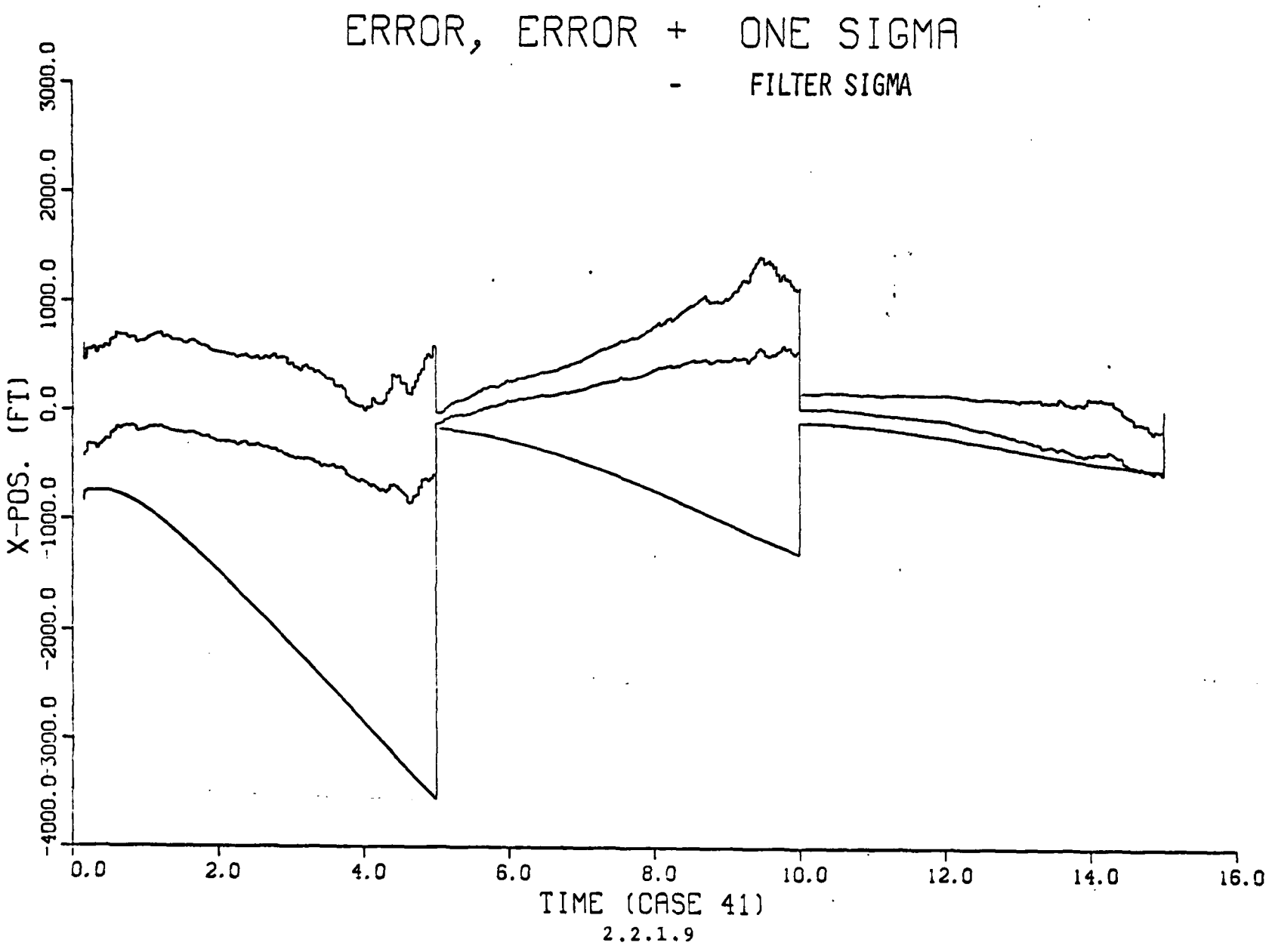

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)

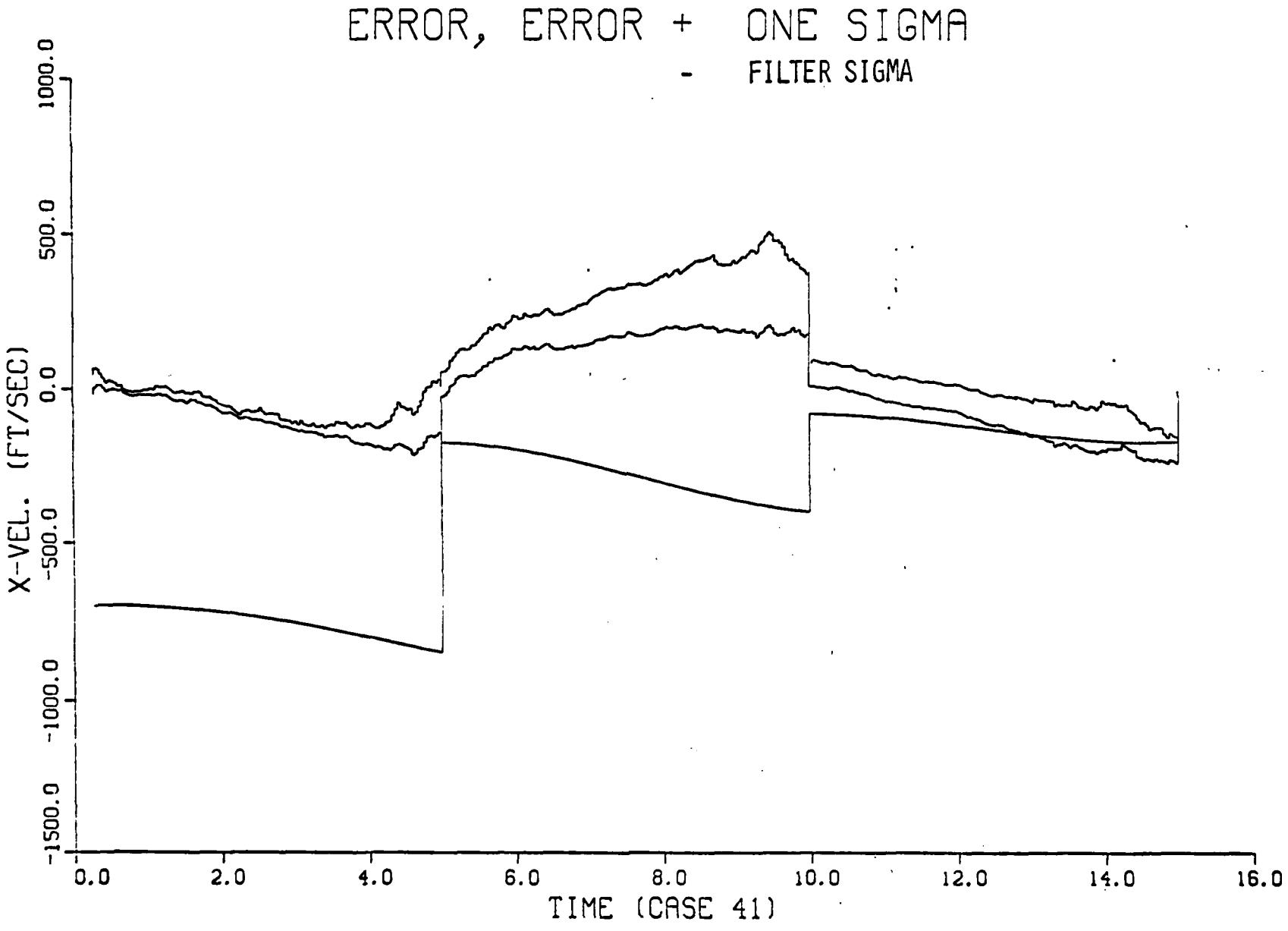
2.2.1.8

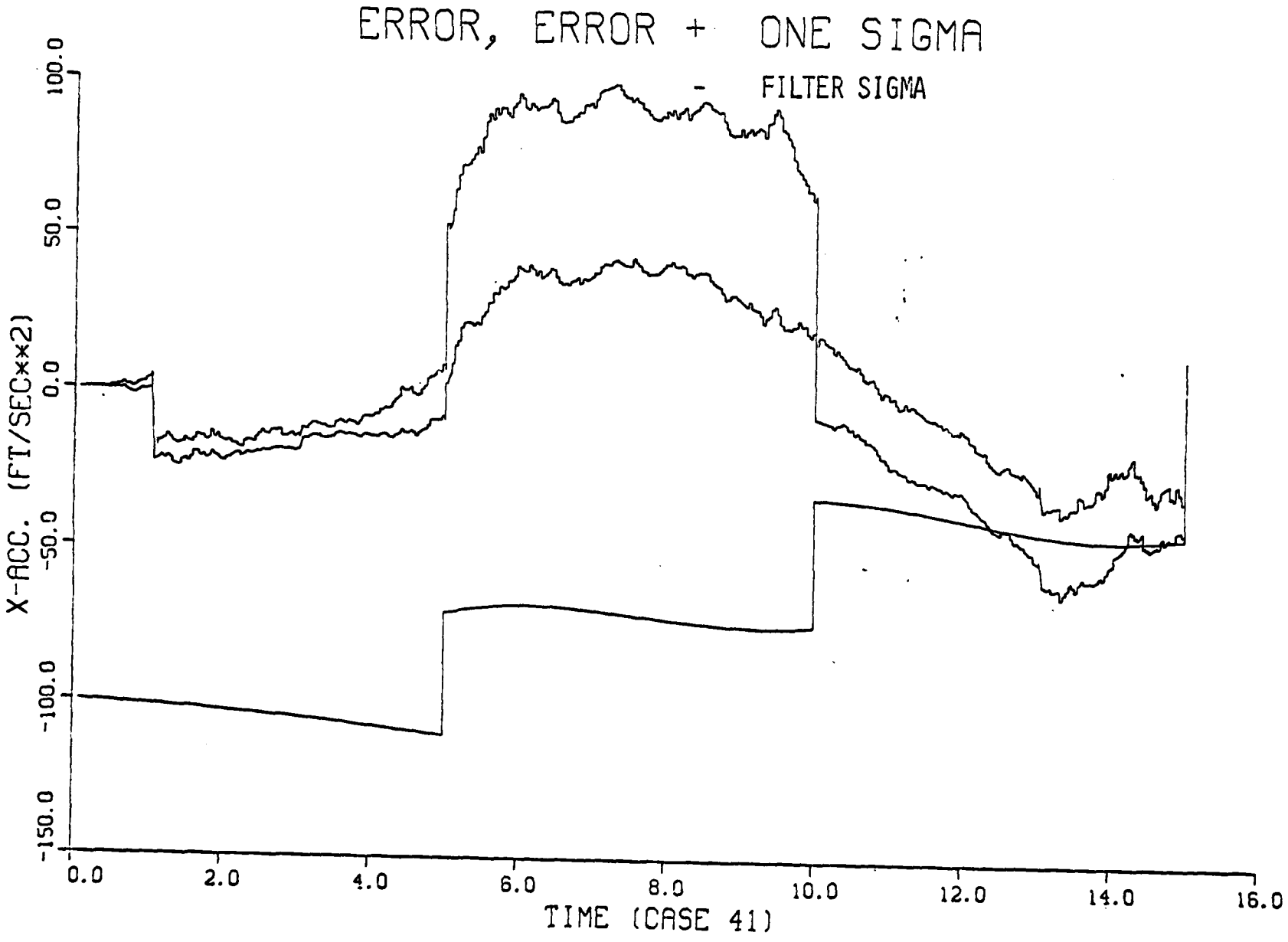

ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
Y-VEL. (FT/SEC)
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)

2.2.1.8

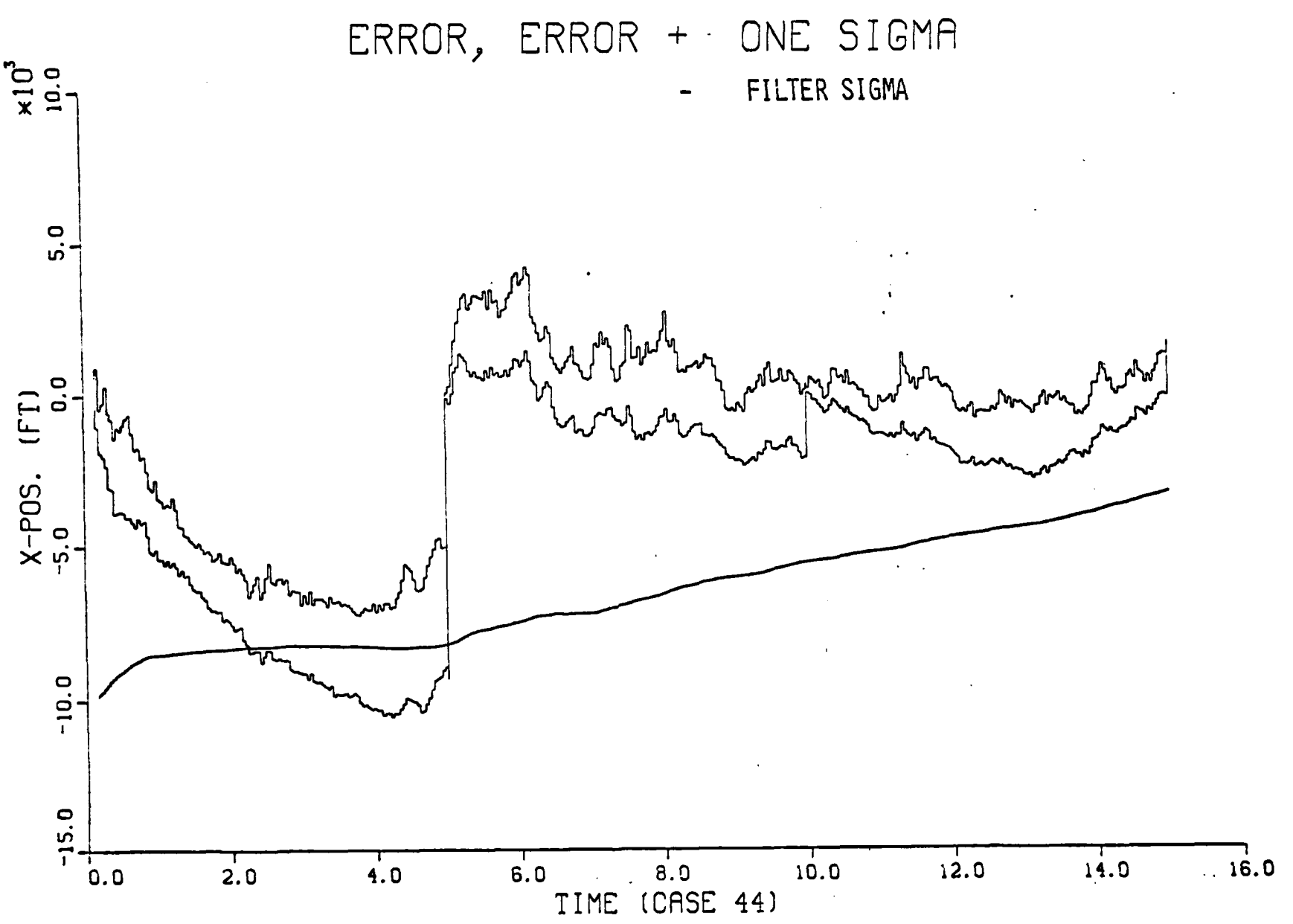

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 40)


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
3000.0
2000.0
1000.0
0.0
-1000.0
-2000.0
-3000.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 41)
2.2.1.9

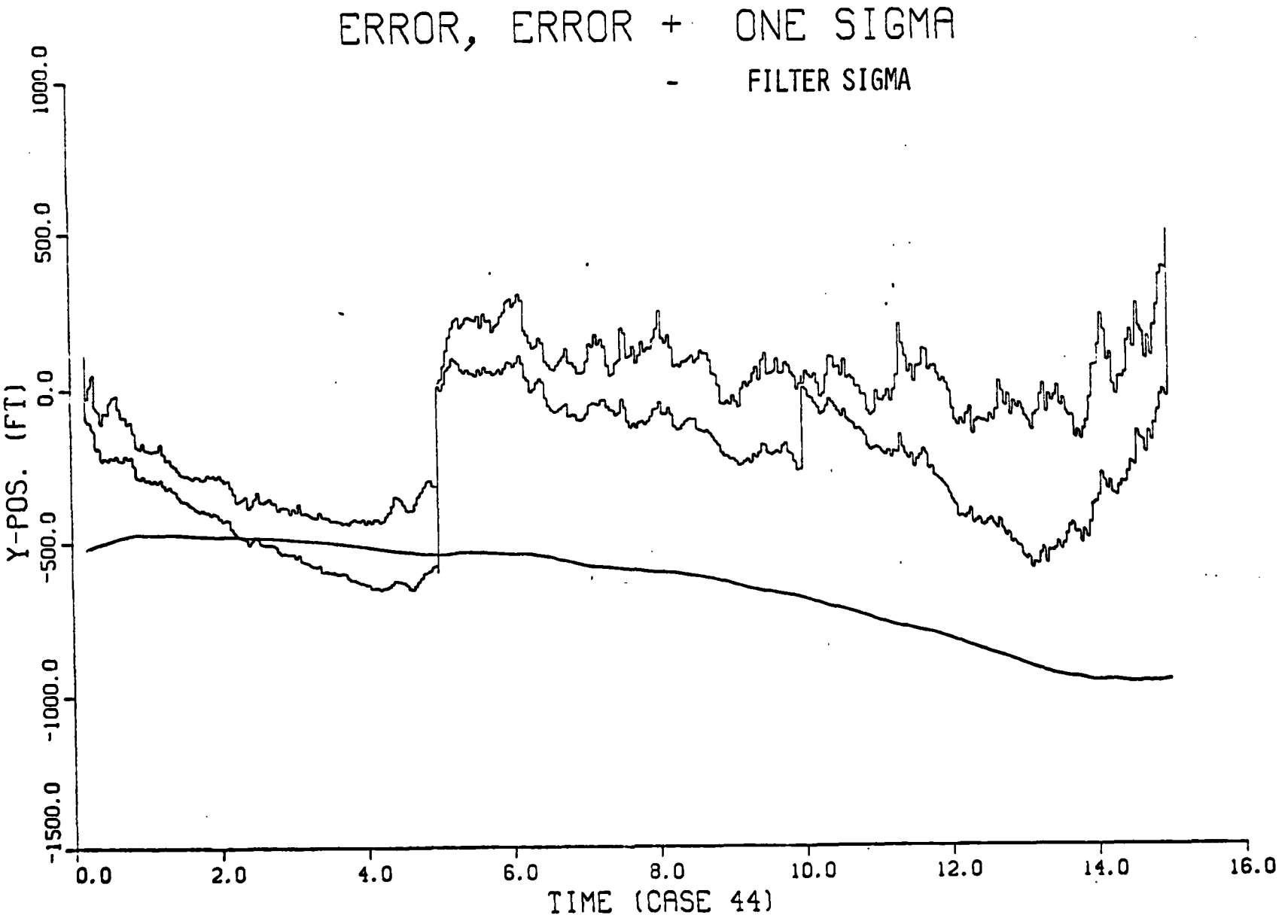

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 41)


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
100.0
50.0
0.0
-50.0
-100.0
-150.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 41)

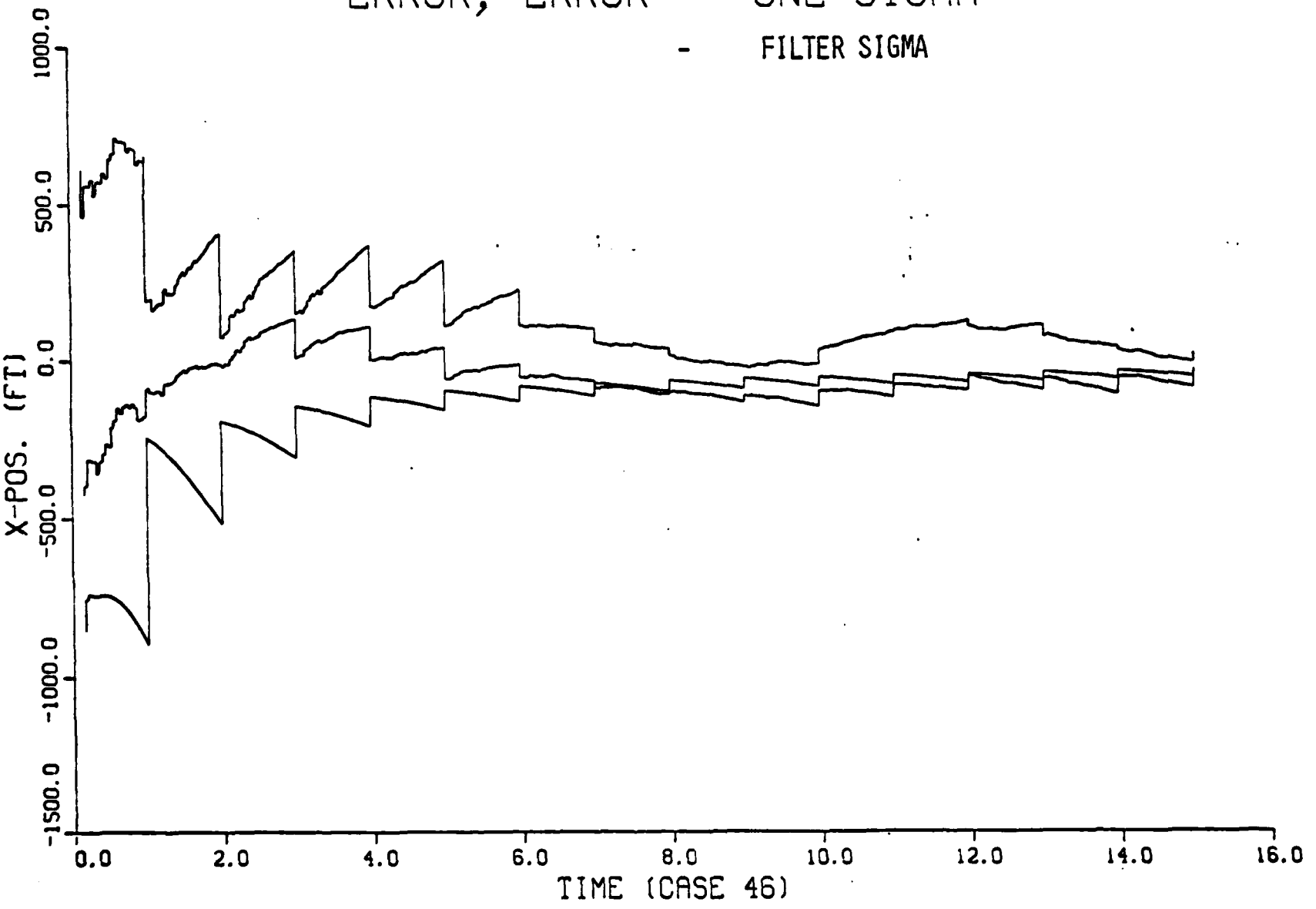
2.2.1.9


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
×10³
10.0
5.0
0.0
-5.0
-10.0
-15.0
X-POS. (FT)
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 44)

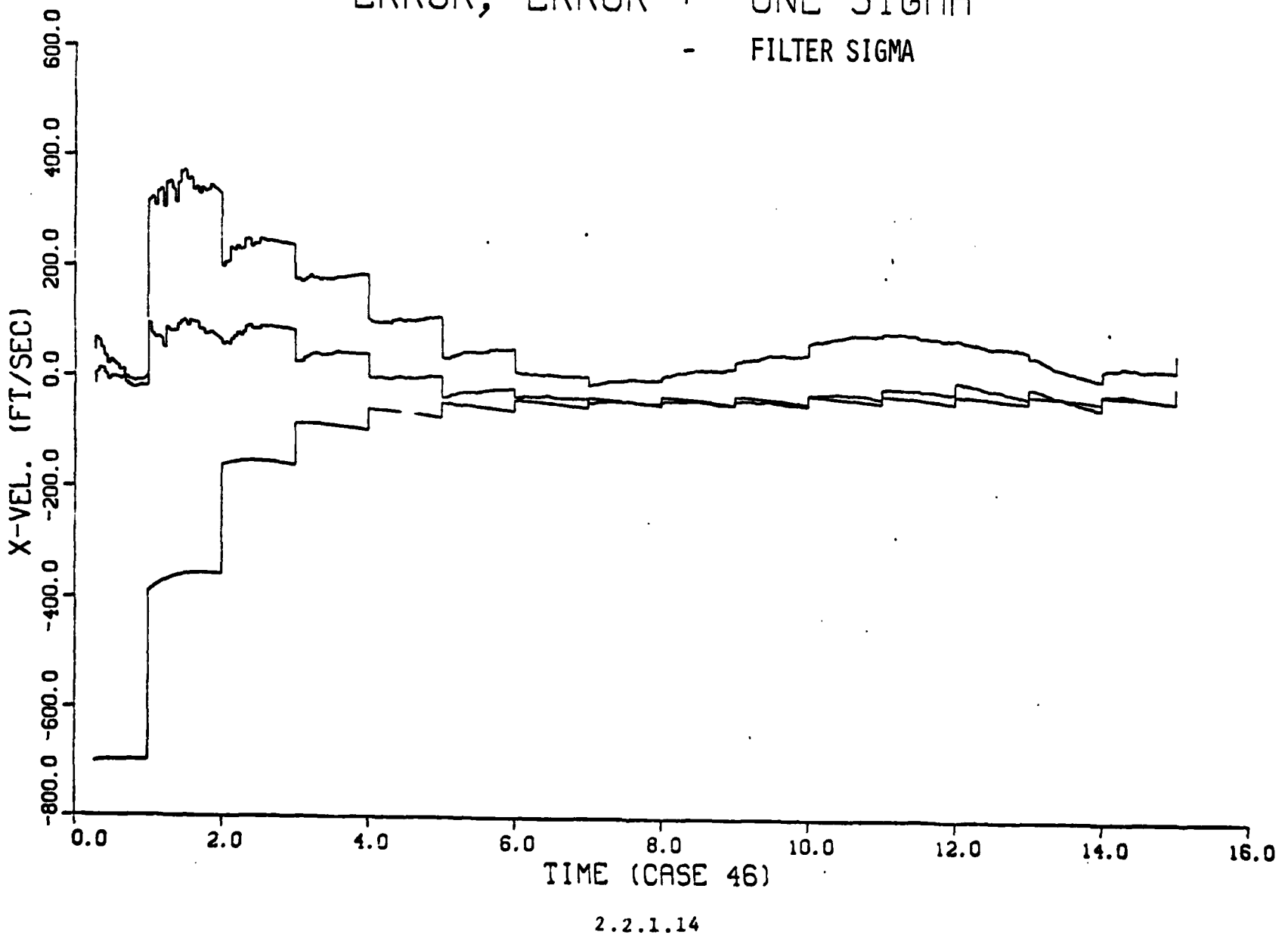
2.2.1.12


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 44)

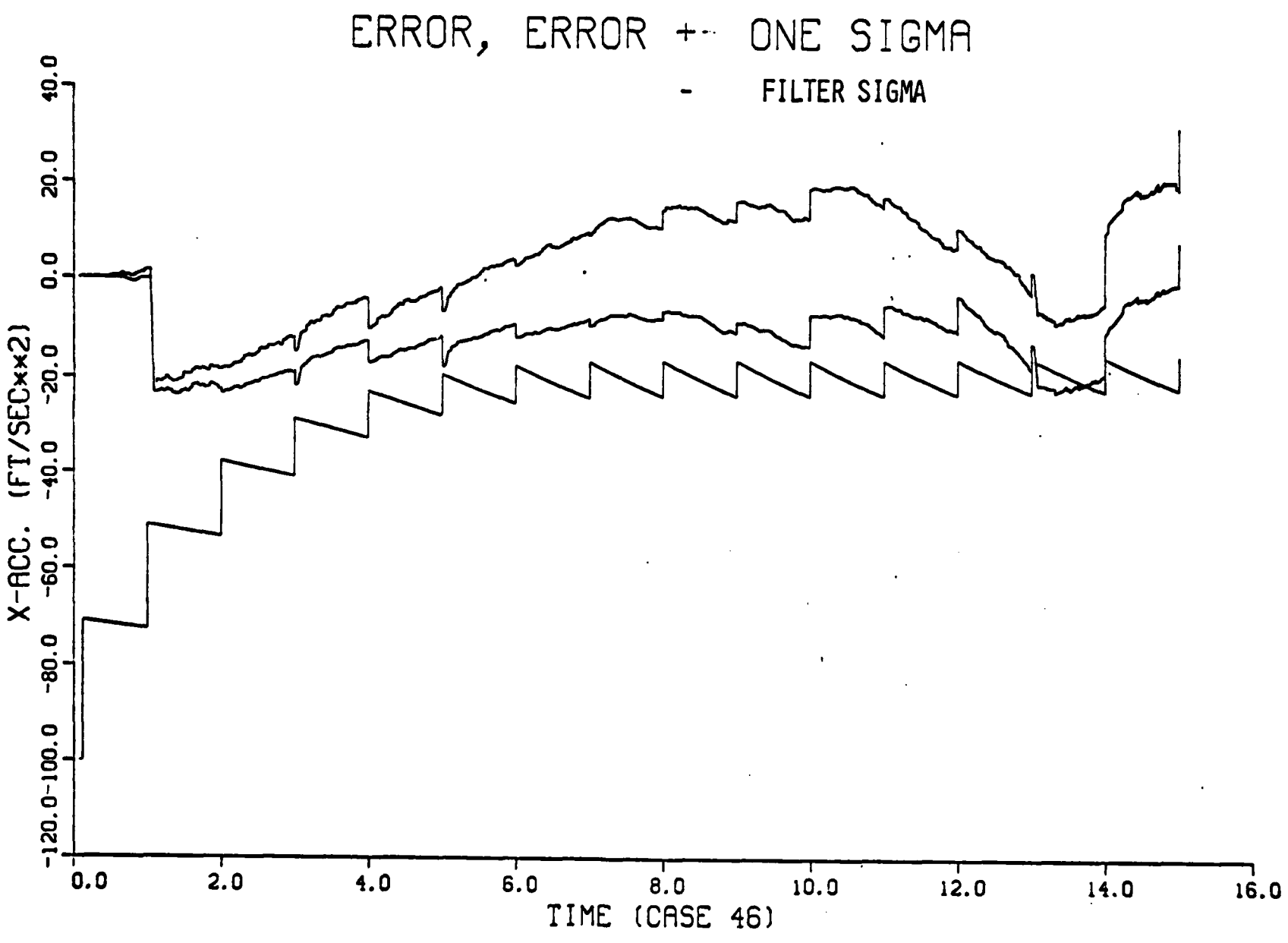
2.2.1.12


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
X-POS. (FT)
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 46)

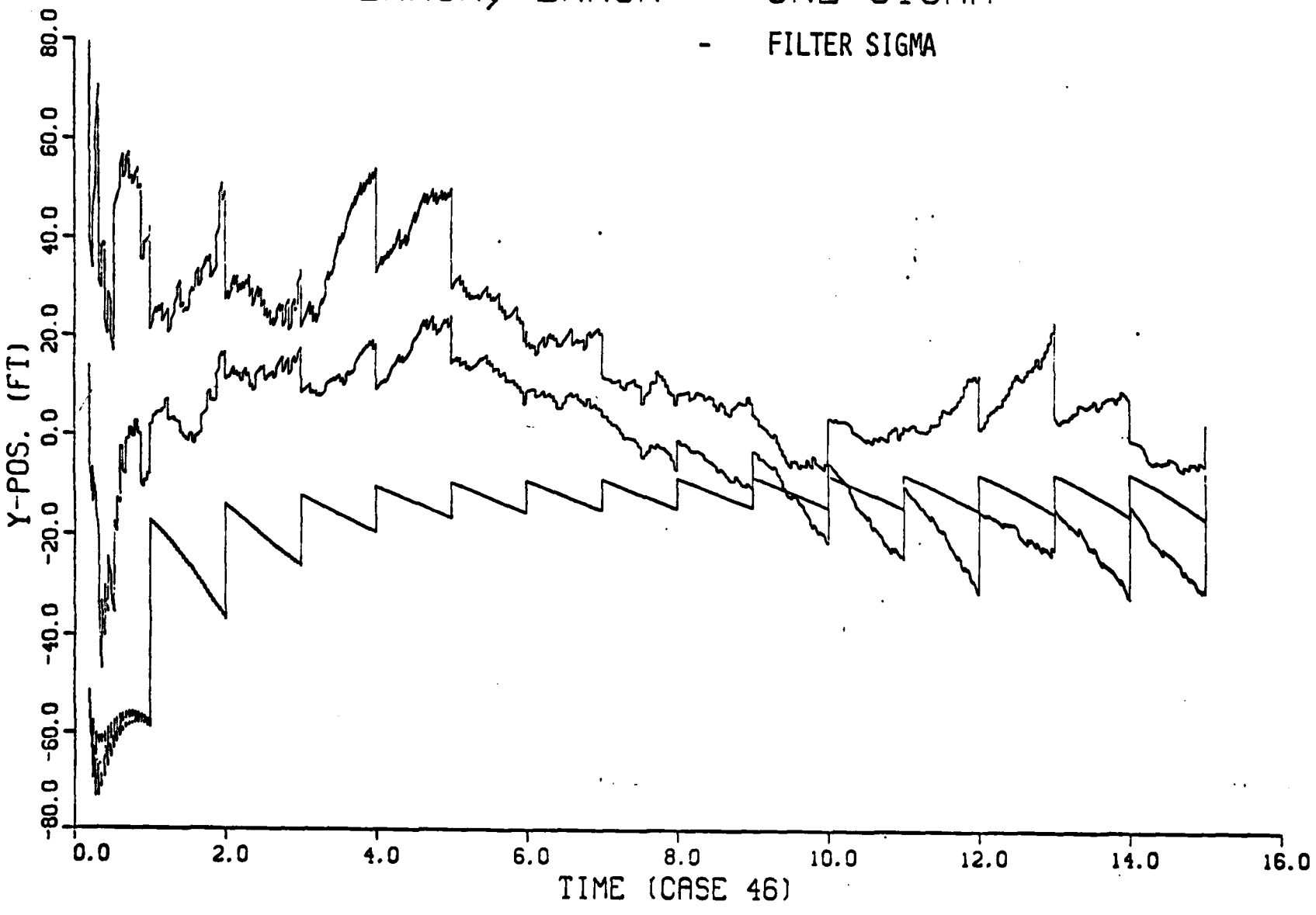
2.2.1.14


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
600.0
400.0
200.0
0.0
-200.0
-400.0
-600.0
-800.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 46)

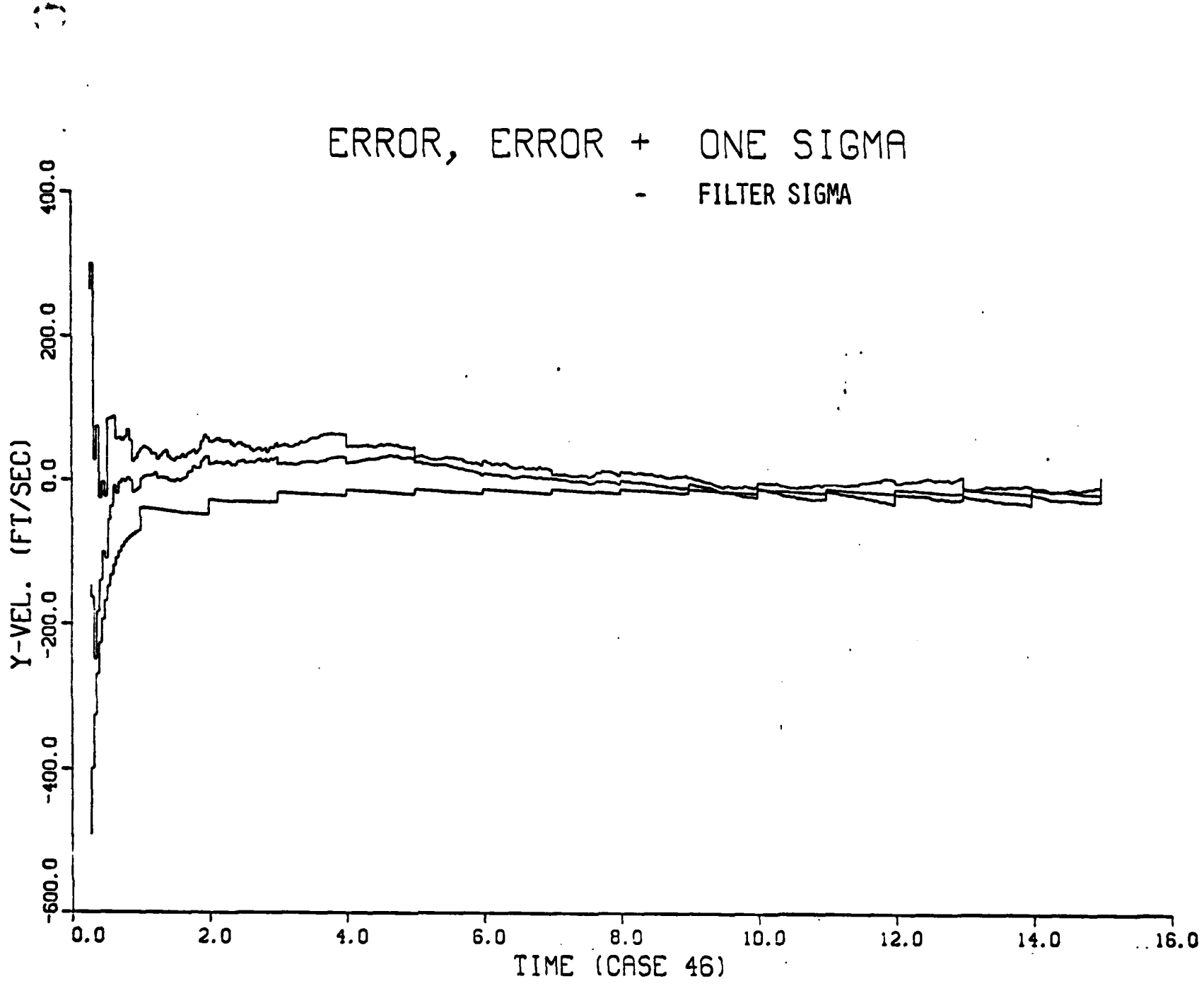
2.2.1.14


ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 46)

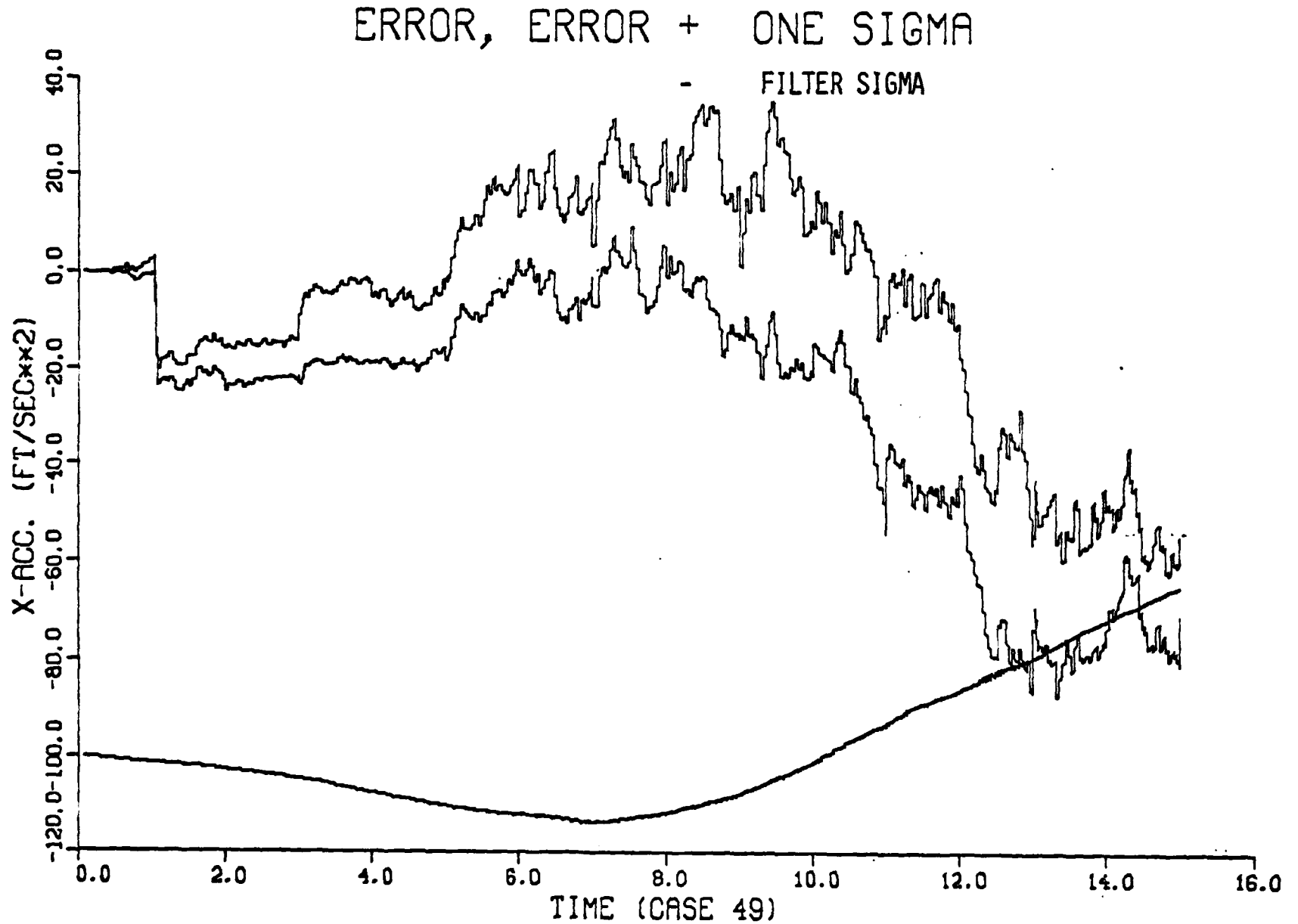
2.2.1.14


ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 46)

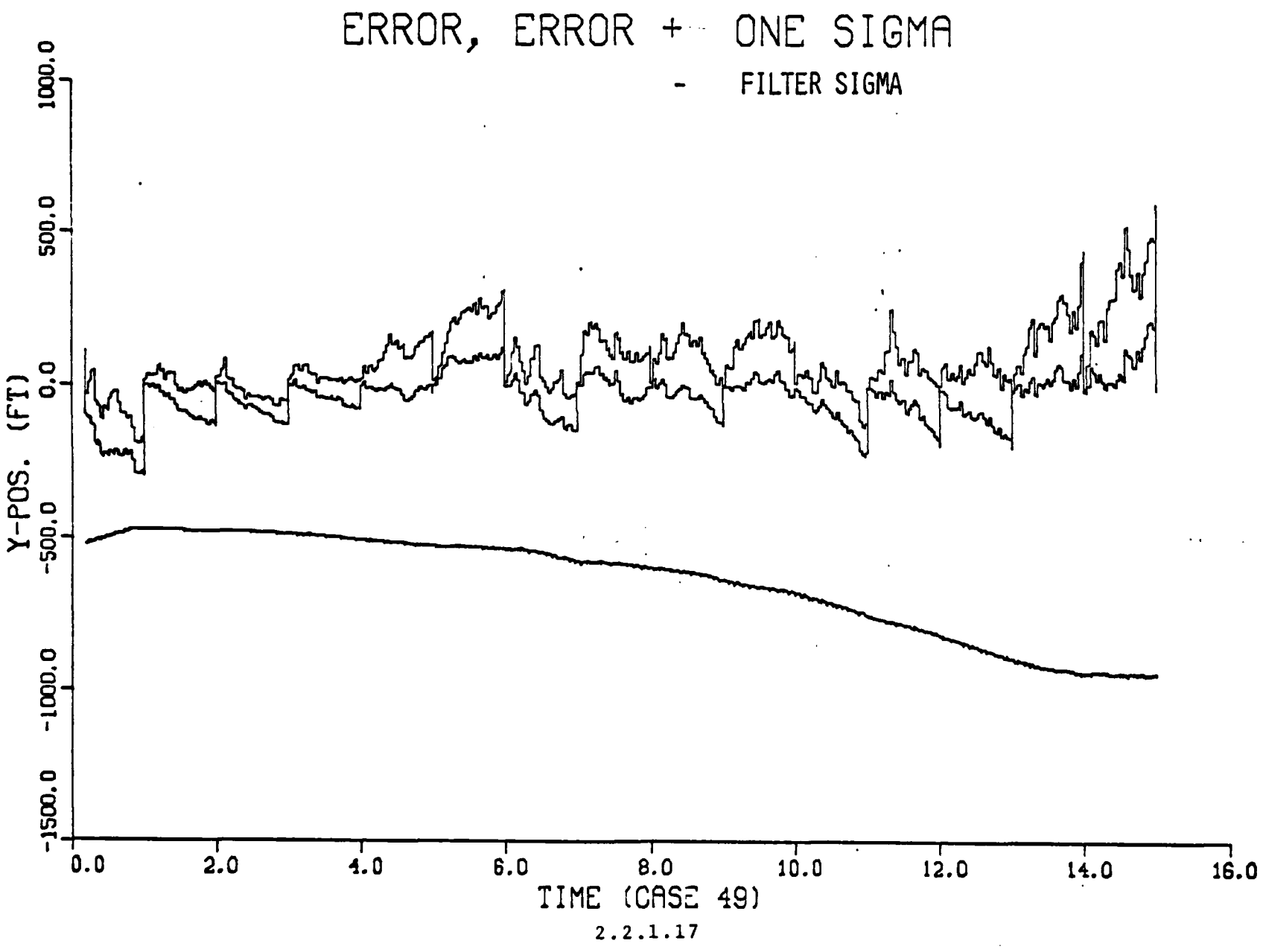
2.2.1.14

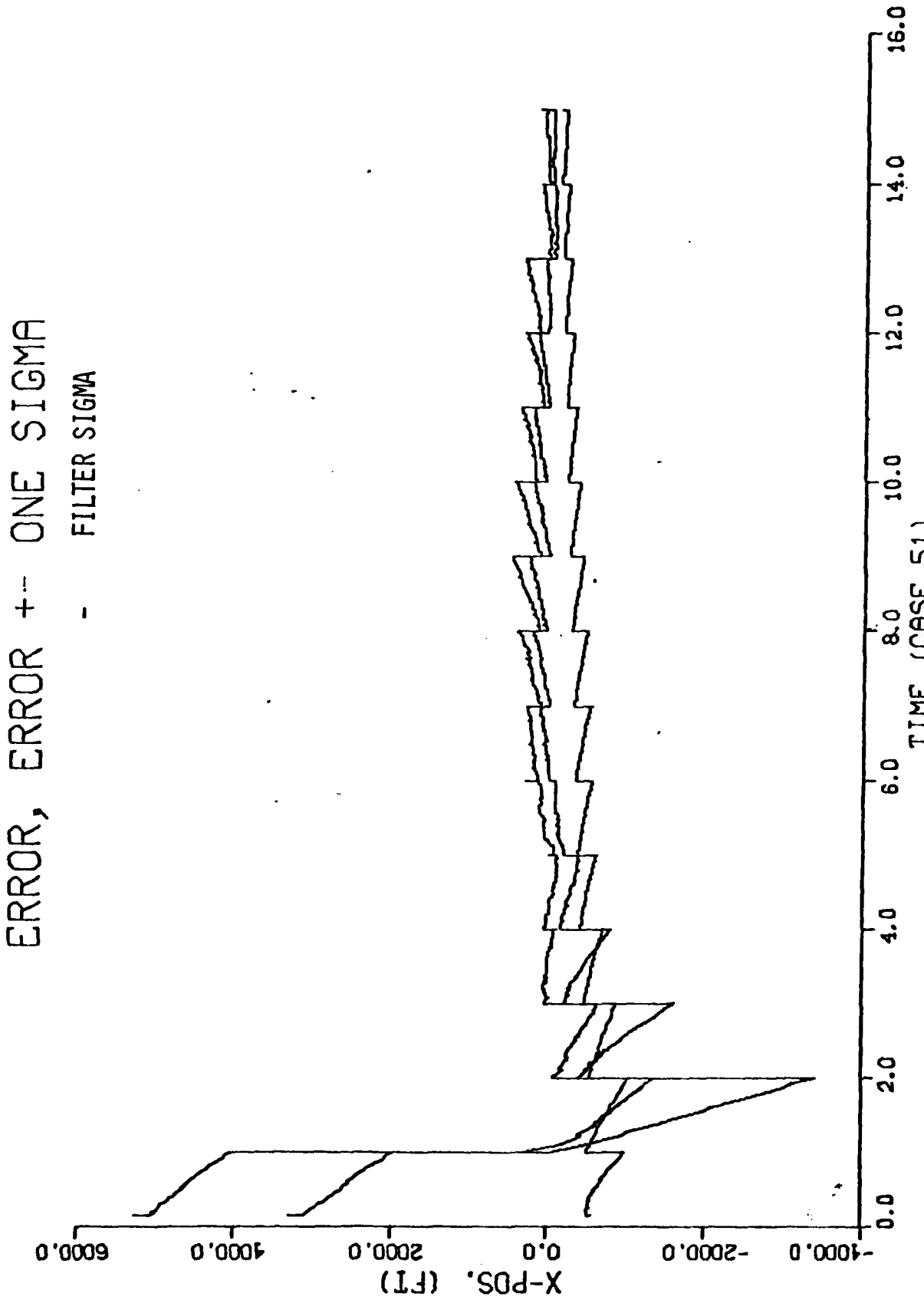

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-VEL. (FT/SEC)
400.0
200.0
0.0
-200.0
-400.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 46)

2.2.1.14

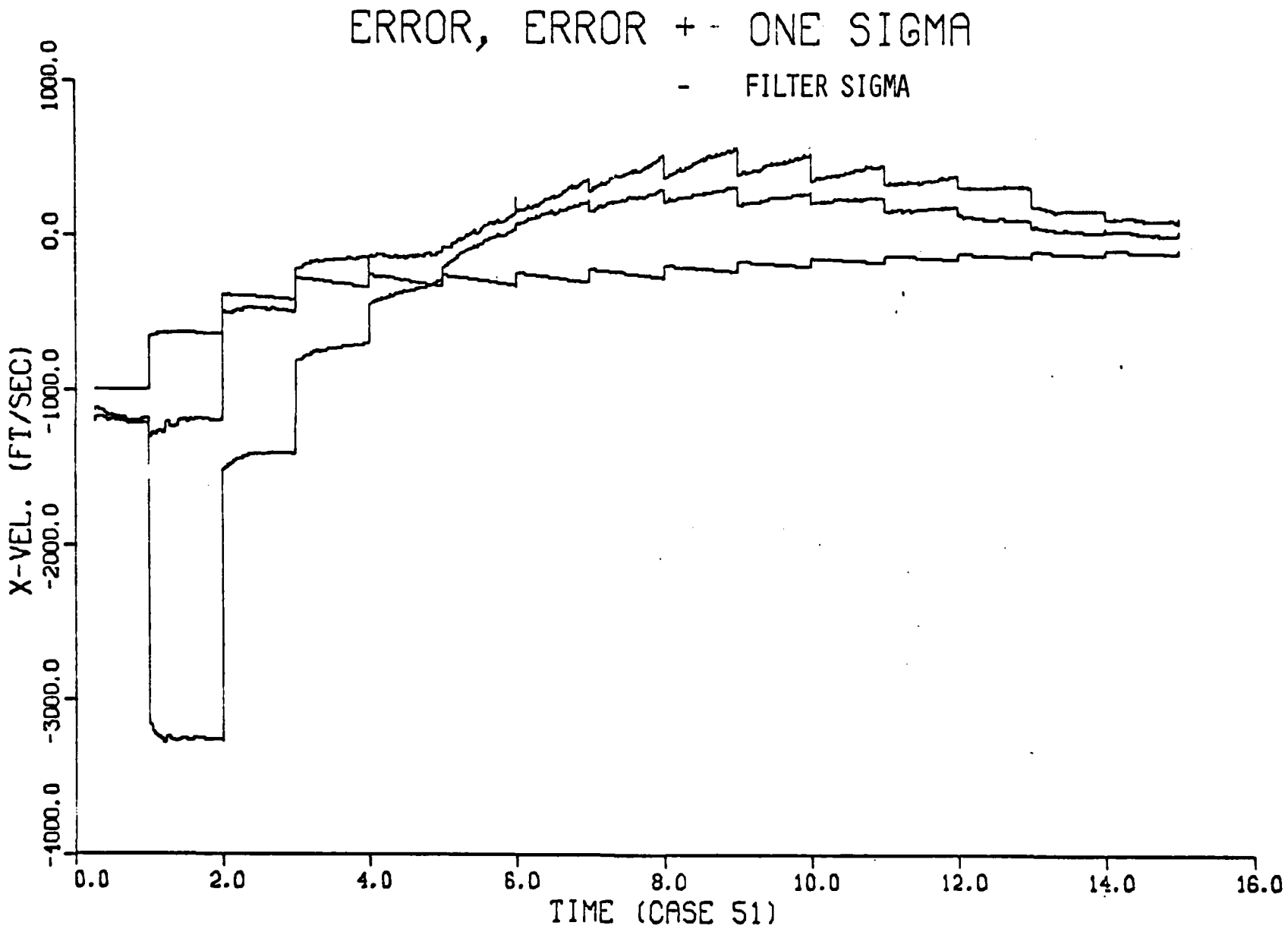

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
X-ACC. (FT/SEC**2)
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 49)

2.2.1.17

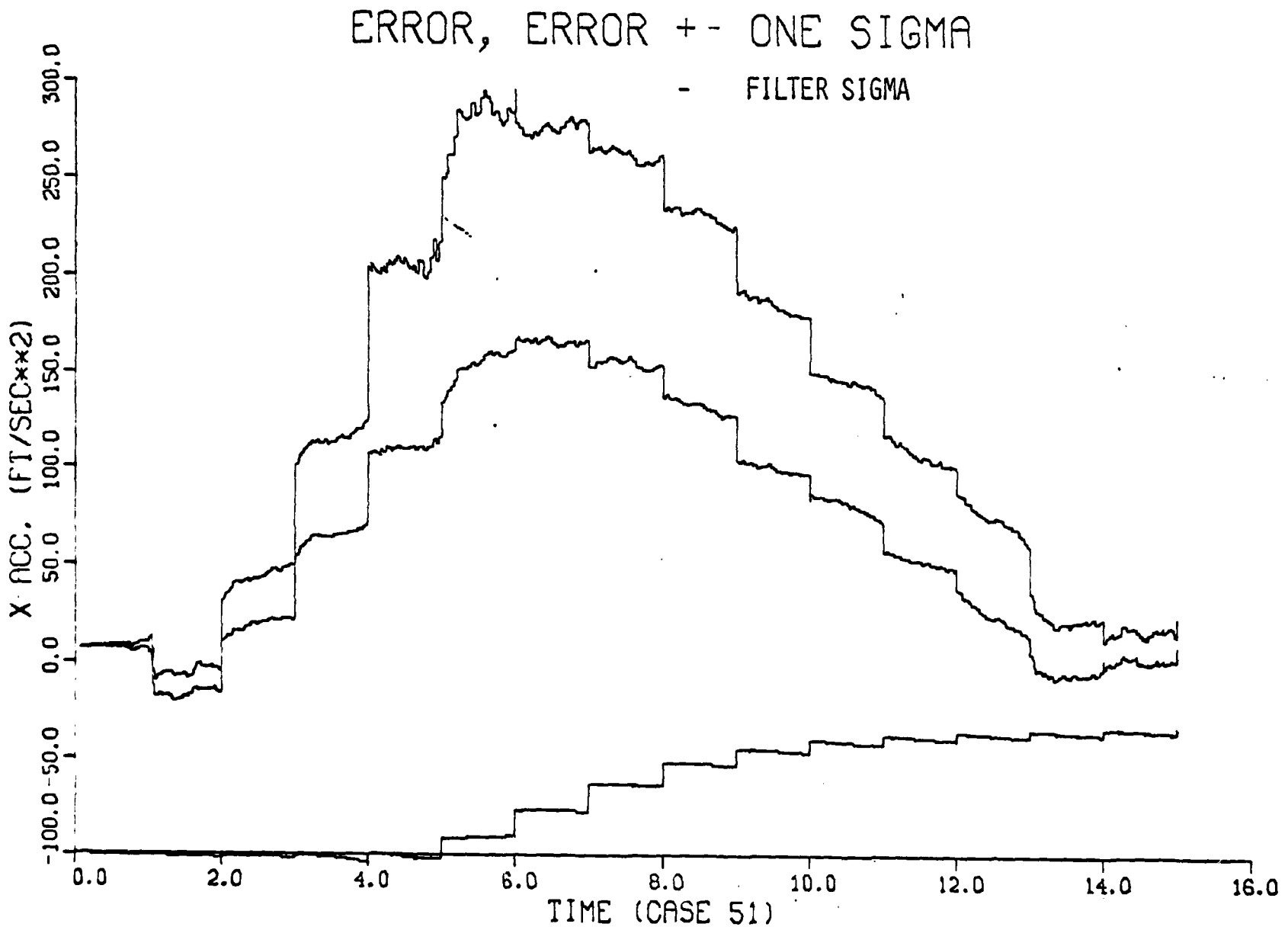

ERROR, ERROR + ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
1000.0
500.0
0.0
-500.0
-1000.0
-1500.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 49)
2.2.1.17

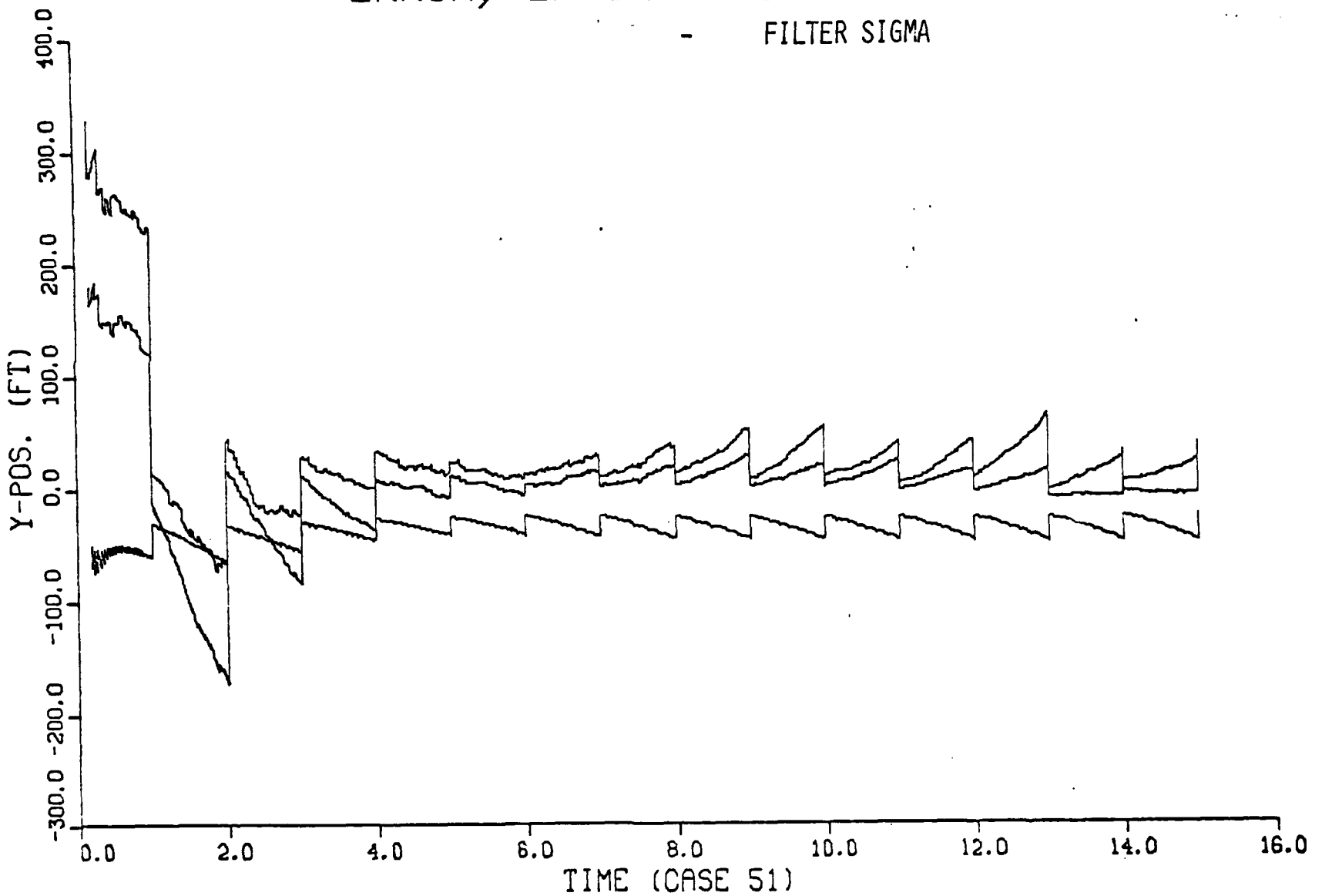

ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
X-POS. (FT)
6000.0
4000.0
2000.0
0.0
-2000.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 51)

2.2.3.1


ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
X-VEL. (FT/SEC)
1000.0
0.0
-1000.0
-2000.0
-3000.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 51)

2.2.3.1


ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
X ACC. (FT/SEC**2)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
-50.0
-100.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 51)


ERROR, ERROR +- ONE SIGMA
- FILTER SIGMA
Y-POS. (FT)
400.0
300.0
200.0
100.0
0.0
-100.0
-200.0
-300.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
TIME (CASE 51)

2.2.3.1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/80-5	2. GOVT ACCESSION NO. AD-A085 799	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Cooperative Estimation of Targets by Multiple Aircraft		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ralph S. Bryan		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Wright Aeronautical Laboratories AFWAL/AART Wright-Patterson AFB, Ohio 45433		12. REPORT DATE June, 1980 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17 JOSEPH P. HIPPS, Maj, USAF
Director of Public Affairs

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman Filter	Multiple Targets	Cooperative Estimation
Target Tracking	JTIDS	
Estimation	Telemetry	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A cooperative estimator of aerial target motion was designed and evaluated using an Extended Kalman Filter and a Monte-Carlo simulation. The estimator was designed to combine tracking information from two cooperating fighter aircraft which are tracking a single aerial target in either range and angle, or in angle-only. The cooperation occurs when raw measurements or state estimates are transferred between aircraft via a telemetry link. The design and evaluation were both preliminary in the sense that necessary information about attacker's relative position and orientation was assumed error-free and tracker

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. errors were modeled as white noise.

Two methods of data transfer were simulated. These were Direct Measurement Transfer and State Estimate Transfer. The Monte-Carlo simulation of Direct Measurement Transfer, using two different target trajectories, indicated up to 30% performance improvement for cooperative estimation compared to single filter estimation with range and angle measurements. When only angle measurements were available, the cooperative estimator was able to estimate the complete target state, converging to within 1% of true range within two seconds on the trajectories simulated. State Estimate Transfer, which was simulated for the angle-only case, proved successful only when the transfer rate was approximately 1.0 Hz.