Statistically Robust Pseudo Linear Identification

by

Harald Alnor

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of science

in

Electrical Engineering

APPROVED:

H.F. VanLandingham, chairman

D.K. Lindner

L. Mili

March 31, 1989

Blacksburg, Virginia

Statistically Robust Pseudo Linear Identification

by

Harald Alnor

H.F. VanLandingham, chairman

Electrical Engineering

(ABSTRACT)

It is common to assume that the noise disturbing measuring devices is of a Gaussian nature. But this assumption is not always fulfilled. A few examples are the cases where the measurement device fails periodically, the data transmission from device to microprocessor fails or the A/D conversion fails. In these cases the noise will no longer be Gaussian distributed, but rather the noise will be a mixture of Gaussian noise and data not related to the physical process. This posses a problem for estimators derived under the Gaussian assumption, in the sense that these estimators are likely to produce highly biased estimates in a non Gaussian environment.

This thesis devises a way to robustify the *Pseudo Linear Identification* algorithm (PLID) which is a joint parameter and state estimator of a Kalman filter type. The PLID algorithm is originally derived under a Gaussian noise assumption. The PLID algorithm is made robust by filtering the measurements through a nonlinear odd symmetric function, called the ψ function, and let the covariance updating depend on how far away the measurement is from the prediction. In the original PLID the measurements are used unfiltered in the covariance calculation.

This is not the case for the robust version. Here the prediction plus a correction term is used instead.

Simulations show that this is a very promising way to extend the PLID Algorithm to the non Gaussian noise case.

Acknowledgements

I wish to thank my family for giving me the opportunity to get a higher education and for their support throughout all my school years.

I am thankful to my advisor Professor H. F. VanLandingham, for providing me with the initial idea to this thesis, and for the patience and guidance he has given throughout my time at Virginia Tech and in this project.

I also wish to thank Professor L. Mili, with whom I had many fruitful discussions about robustness, and whose lectures on robustness in power systems were especially helpful.

Also thank you to Professor D.K. Lindner for his willingness to serve on my graduate committee.

A special thank you goes to my supervisor at Danfoss A/S, Leif Sturlason who encouraged me to go on with graduate studies and who helped me get a "Andreas Jepsens Mindelegat" scholarship.

Thanks also to my roommate for proofreading and improving my writing style.

Table of Contents

List of Illustrations

1.0 Introduction

This study investigates statistically robust pseudolinear identification. It combines concepts from robust statistics and from system identification.

System identification is the branch of control engineering that deals with the problem of extracting information from measurements of input and output signals about a dynamic systems structure. This information can be represented as Bode plots, parameters in differential equations and difference equations. There are two ways to acquire knowledge on the dynamic system. The first method consists in modeling the system by the physical or socio economic laws that govern the system behavior. This approach results in equations with parameters. The equations can then be interpreted and have a meaning, like inertia, heat transfer coefficients, interest rates etc. The second approach is to view the system as a black box with only the inputs and outputs visible. By visible inputs and outputs it is meant that they are measurable. One assumes that the system inside the black box has a certain structure or the equations have a certain order,

but the parameters of these equations are unknown. Hence, part of the control designer's work is to find the numerical value of the parameters. The black box approach does not provide any insight to certain physical parameters , like resistance, inductance, inertia, friction, etc., influence the performance of the system. Therefore, one usually models the system from the governing physical laws to obtain information on the structure of the system. The numerical values of the parameters are then determined by system identification.

This study considers the Pseudo Linear Identification (PLID) method. The PLID is a joint state and parameter estimator, which is derived as an optimal state observer. The states, which the PLID estimates include both the parameters of the dynamic equation and the states of the dynamic system. The estimated states are called extended states.

The PLID assumes that measurement and process noise is Gaussian distributed, and the optimality criterion used to derive the PLID is to minimize the trace of the estimation error covariance matrix. This results in a Kalman filter like algorithm. Like the Kalman filter is the optimal observer in the case of Gaussian noise, the PLID is the optimal joint parameter and state estimator in the Gaussian noise case. But these least squares estimators are only optimal as long as the noise is Gaussian.

To see what happens when the Gaussian noise assumption is violated, the recursive least squares identification method is applied to the dynamic equation :

$$y_{k+1} = 0.8y_k + 0.6u_k + v_k \tag{1}$$

Figures 1 and 2 show the effect on the recursive least squares identification algorithm between Gaussian noise and contaminated Gaussian noise case. Figure 1 shows the identification of the two parameters from equation (1). The simulation is done with Gaussian distributed noise. The identification of the parameters converges to the values in equation (1). In figure 2 the noise is Gaussian added with outliers. The outliers are constants with magnitude 1000. The noise is therefore no longer purely Gaussian distributed. The simulations show that the identification algorithm looses track of the parameters and fails to converge. This example shows that the Recursive Least Squares algorithm is very sensitive to violations of the Gaussian noise distribution.

In order to devise a statistically robust system identification method this study combines concepts from the theory of robust estimation and the PLID identification algorithm. The basic idea behind the statistically robust PLID is to apply "data cleaning" to the measurement data. Data cleaning means that one compares measurement data with predicted measurements, and decides if the the measurement is "good" or "bad". Martin and Thomson [17], succesfully applied this idea to robust spectrum estimation. Data cleaning removes or reduces the influence of the non-Gaussian distributed data.

In this work, the term robustness is from the context of robust statistics. One defines a *robust estimator* as an estimator that only gives small differences in the estimate when the data distribution changes slightly from the assumed distribution. One usually assumes that data is Gaussian distributed, but experience proves that 5 to 10% of errors occurring in engineering systems cannot be ex-

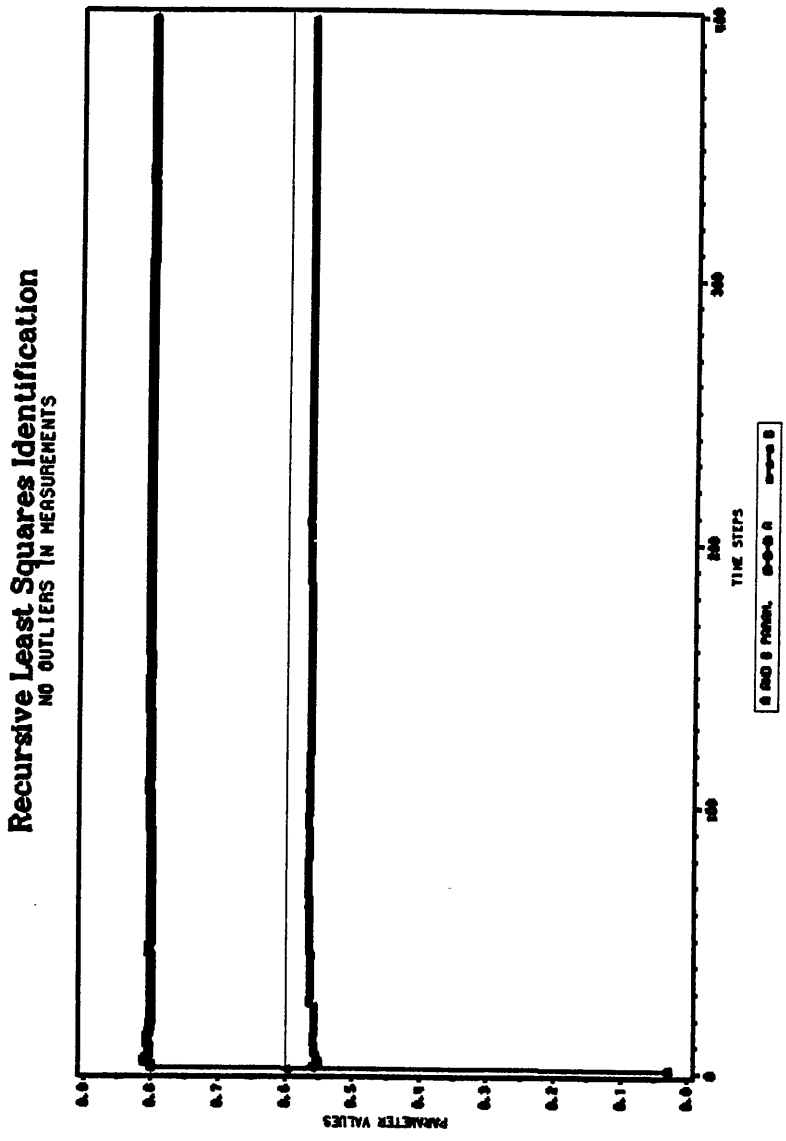

Figure 1. Recursive Least Squares.: This figure shows the parameter estimation without outliers.

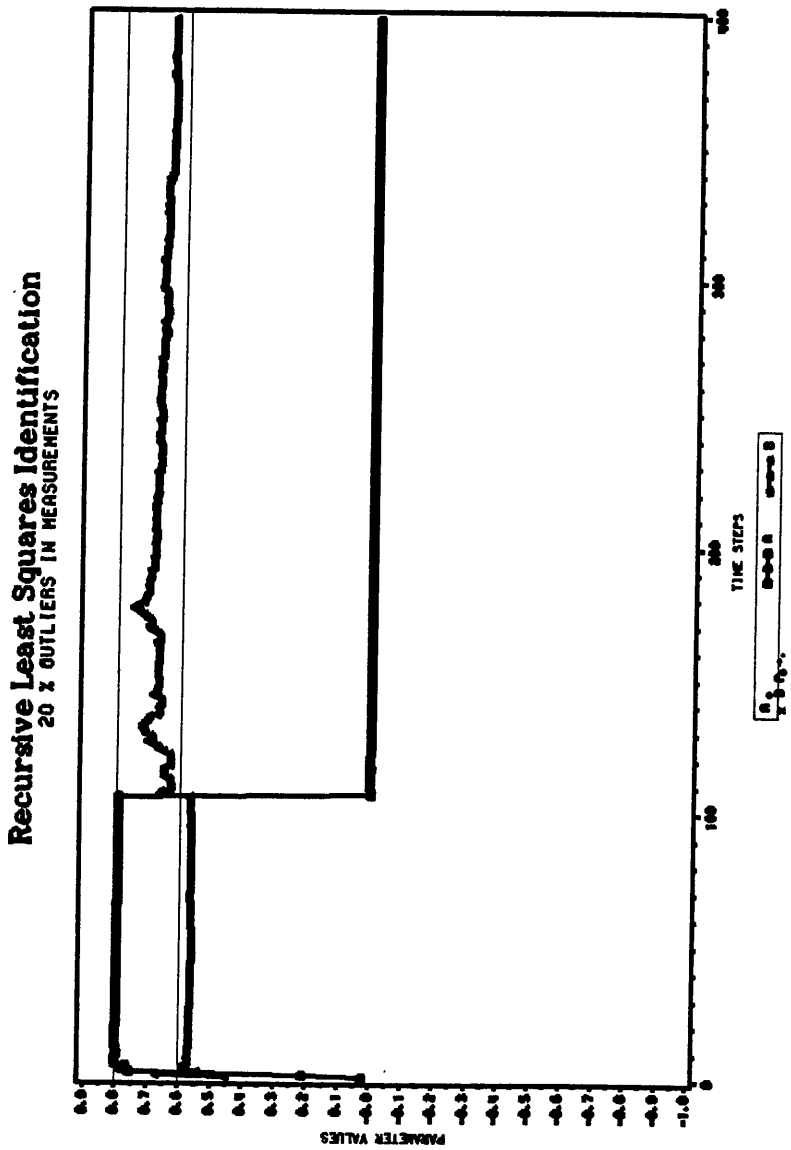

Figure 2. Recursive Least Squares.: This plot shows the parameter estimation in the presence of 20% outliers in the data.

plained by small fluctuations, (see [11], [12]). These data distributions can be viewed as consisting of a Gaussian part and a contaminating part. Data are thus coming from two (or more) different statistical populations. The classical least squares estimators are optimal when the data distribution is purely Gaussian. These estimators, however, tend to give a large bias when the assumptions concerning the Gaussian distribution are violated. Most estimators in engineering are based on the assumption that data is Gaussian distributed. If the data is non-Gaussian distributed, one can expect largely biased estimates. It is therefore useful to apply the theory of robust statistics to estimators used frequently in engineering. This study will apply the theory of robust statistics to the PLID to make it resistant to deviations from the Gaussian distribution.

In the past twenty five years statisticians have shown increasing interest in "robust statistics". Statisticians, engineers, and scientists who work with information extracted from data sets and measurements have been aware for a long time that data sets can contain erroneous data. In this case the data sets are said to be *contaminated.* Before statisticians can extract useful information from contaminated data sets or data distributions, they must discard the erroneous data by some ad hoc rejection methods. In the early 1960s researchers formalized the idea that data sets can be "contaminated". They found ways to describe mathematically the contaminated data distributions. This mathematical approach allowed them to identify the conditions for estimators to extract information from this contaminated body of data.

Two researchers F. Hampel and P. Huber (see [4] and [5]) made very important contributions to this new area of statistics. Their approaches to robust statistics differ slightly. Hampel investigates the robustness of an estimator by studying an estimators behavior in an infinitesimal neighborhood of the assumed distribution, and extrapolating the properties to a finite neighborhood. In contrast, Huber investigates the robustness of an estimator in a finite neighborhood around the assumed distribution. Statisticians refer to Huber's method as *Hubers Min-Max Method* and Hampels method as *The Influence Function Approach.*

The major contribution of this thesis is combining ideas from robust kalman filtering with the PLID algorithm. The equations that constitute the PLID are modified with the techniques encountered in technical papers. This study investigates different methods to reduce the influence of contamination in the identification process. Hence, a robust estimator has the ability to handle non- Gaussian noise cases.

The major result of this study is that the PLID by a few simple and intuitively appealing modifications, is able to function reliably in an environment with non-Gaussian measurement noise. The robust PLID is also able to function with only bad measurements for a longer time period. The best results are obtained when the robust PLID is started with good initial guesses, because the PLID is most sensitive to outliers in the transient phase of the identification.

Before discussing more fully the different methods to robustify the Kalman filter one must understand the PLID Algorithm and robustness concepts. Thus, the first chapter discusses the PLID Algorithm. It examines the PLID algorithm

in relation to single input and single output systems. It also analyses the stochastic PLID for single input and single output systems.

As a consequence, an analysis of robustness concepts proves worthwhile to devise a method to robustify the PLID algorithm. The third chapter introduces the concepts of robust statistics, and introduces the termonoligy used in this branch of statistics.

The fourth chapter reviews the Kalman filter and presents some methods to robustify the Kalman filter. The ideas used to robustify the Kalman filter are then applied to the PLID algorithm. The idea of the robust PLID algorithm is briefly described as follows. The residual of the prediction and measurement is limited by a nonlinear function, and the updating of the covariance is weighted by an also nonlinear function. Finally the measurements are cleaned by using prediction and the bounded residual.

The fifth chapter presents the results of simulations on a single input single output system.

2.0 The PLID Algorithm

2.1 The Extended State Representation (ESR)

The *Pseudo Linear Identification Algorithm* (PLID) is a joint parameter and state estimator. The development of the PLID algorithm is discussed in [6]. The PLID algorithm includes the system parameters in the state representation. The term *'Pseudo linear'* is used because generally if the state vector includes the system parameters, then the state equations become nonlinear, whereas the PLID algorithm uses an extended state representation (ESR), which makes the nonlinearities implicit.

This chapter will summarize some of the developments made in [6].

An initial assumption for the PLID algorithm to be able to identify the parameters of a dynamic system, is that the system be completely observable and controllable.[1]

The Extended State Representation includes the system parameters in the state representation. This usually causes explicit nonlinearities to appear in the state equations. But the use of the observable canonical form of the state equation, will make the nonlinearities implicit, as shown in the following development.

Let a single input single output (SISO) discrete time system be given by the state equation :

$$x_{k+1} = \begin{bmatrix} a_{11} \ldots a_{1n} \\ \ldots \ldots \ldots \\ a_{n1} \ldots a_{nn} \end{bmatrix} \times x_k + \begin{bmatrix} b_1 \\ \ldots \\ b_n \end{bmatrix} u_k \tag{1}$$

$$y_k = [c_1 \ldots c_n] x_k \tag{2}$$

or in a shorthand notation

$$x_{k+1} = \underline{A} x_k + \underline{b} u_k \tag{3}$$

$$y_k = \underline{C} x_k \tag{4}$$

---

[1] A dynamic system is observable and controllable if there are no pole zero cancellations in the transfer function.

If the equation is observable, there exists a similarity transformation matrix T such that the transformation

$$x = T\underline{x} \tag{5}$$

brings the equation into the observable canonical form. This is discussed in [2].

$$x_{k+1} = \begin{bmatrix} 0\,0 \ldots 0 & -a_n \\ 1\,0 \ldots 0 & -a_{n-1} \\ 0\,1 \ldots 0 & -a_{n-2} \\ \ldots\ldots\ldots\ldots & \\ 0\,0 \ldots 1 & -a_1 \end{bmatrix} x_k + \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \ldots \\ b_1 \end{bmatrix} u_k \quad , \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \tag{6}$$

$$y_k = [0 \ldots 01] x_k \tag{7}$$

The transfer function is written as

$$H(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_{n-1} z + b_n}{z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n} \tag{8}$$

It is now possible to reformulate the observable state representation as

$$x_{k+1} = \begin{bmatrix} 0\,0 \ldots 0\,0 \\ 1\,0 \ldots 0\,0 \\ 0\,1 \ldots 0\,0 \\ \ldots\ldots\ldots\ldots \\ 0\,0 \ldots 1\,0 \end{bmatrix} x_k + \begin{bmatrix} -a_n \\ -a_{n-1} \\ -a_{n-2} \\ \ldots \\ -a_1 \end{bmatrix} x_n(k) + \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \ldots \\ b_1 \end{bmatrix} u_k \tag{9}$$

$$y_k = x_n(k) \tag{10}$$

Now using the last equation gives the following

$$x_{k+1} = \begin{bmatrix} 0\,0 \ldots 0\,0 \\ 1\,0 \ldots 0\,0 \\ 0\,1 \ldots 0\,0 \\ \ldots\ldots\ldots\ldots \\ 0\,0 \ldots 1\,0 \end{bmatrix} x_k + \begin{bmatrix} -a_n \\ -a_{n-1} \\ -a_{n-2} \\ \ldots \\ -a_1 \end{bmatrix} y_k + \begin{bmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \ldots \\ b_1 \end{bmatrix} \times u_k \tag{11}$$

This equation can be written in a more compact form as

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \dots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0\,0 \dots 0\,0\,y\,0 \dots 0\,u\,0 \dots 0 \\ 1\,0 \dots 0\,0\,0\,y \dots 0\,0\,u \dots 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0\,0 \dots 1\,0\,0\,0 \dots y\,0\,0 \dots u \end{bmatrix} \begin{bmatrix} x_1(k) \\ \dots \\ x_n(k) \\ -a_n \\ \dots \\ -a_1 \\ b_n \\ \dots \\ b_1 \end{bmatrix} \tag{12}$$

If one assumes that the parameters $(a_n \dots b_1)$ do not change with time, they can be included in the state representation as

$$\theta^T = [\,-a_n \dots -a_1 b_n \dots b_1]^T \tag{13}$$

$$\begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} J_{n\times n} & yI_n & uI_n \\ 0_{2n\times n} & I_{2n} & \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} \tag{14}$$

This is the extended state representation. The extended state vector is defined by

$$s_k = \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} \tag{15}$$

and the transition matrix is defined by

$$F_k = \begin{bmatrix} J_{n\times n} & yI_n & uI_n \\ 0_{2n\times n} & I_{2n} & \end{bmatrix} \tag{16}$$

where

$$J = \begin{bmatrix} 0\,0 \ldots 0\,0 \\ 1\,0 \ldots 0\,0 \\ 0\,1 \ldots 0\,0 \\ \ldots\ldots\ldots\ldots \\ 0\,0 \ldots 1\,0 \end{bmatrix} \tag{17}$$

The extended state equation becomes

$$s_{k+1} = F_k s_k \tag{18}$$

$$y_k = H s_k \tag{19}$$

The equation is clearly time-varying, because F_k includes measurements of the input u_k and output y_k. There are no explicit nonlinearities in this state representation, so it is tempting to identify the system parameters by building a linear observer for the extended state vector s_k. In fact, this is the idea that led to the development of the PLID algorithm, which is an optimal observer (Kalman filter) for the extended state representation.

2.2 *The PLID algorithm for Single Input Single Output.*

As mentioned earlier, incentive to develop the PLID algorithm is to observe the state vector of the extended state representation of a dynamic system.

At this stage it is necessary to introduce the work in [7], which describes the PLID algorithm for a Single Input Single Output system.

until now, the extended state representation assumed a system with no measurement noise and no process noise. The following extended state representation includes both process and measurement noise.

The state equation with process noise w_k (n dimensional vector) and measurement noise v_k is

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{20}$$

$$y_k = Cx_k + v_k \tag{21}$$

In the ESR the process and measurement noise add in the following way

$$\begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = F_k \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} w_k \\ 0 \end{bmatrix} \tag{22}$$

$$y_k = H \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + v_k \tag{23}$$

In the literature the process and measurement noise are assumed to be white gaussian and independent. Let the covariance matrices of w_k and v_k be given by

$$E[w_k w_k^T] = Q_1(k) \quad (n \times n) \tag{24}$$

$$E[v_k v_k] = r(k) \quad (1 \times 1) \tag{25}$$

The two noise representations $Q_1(k)$ and $r(k)$ give the following noise representation for the ESR case

$$Q(k) = E\left\{\begin{bmatrix} w_k \\ 0 \end{bmatrix}[w_k^T \quad 0]\right\} = \begin{bmatrix} Q_1(k) & 0 \\ 0 & 0 \end{bmatrix} \tag{26}$$

The measurement noise variance is still given as

$$E[v_k v_k] = r(k) \quad (11) \tag{27}$$

Having so defined the process and measurement noise, a Kalman filter type of observer can be formulated for the dynamic system

$$s_k = F_{k-1} s_{k-1} + n_k, \quad n_k = \begin{bmatrix} w_{k-1} \\ 0 \end{bmatrix} \tag{28}$$

$$y_k = H s_k + V_k \tag{29}$$

The Kalman filter is then given by the equations

$$
\begin{aligned}
M_k &= F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1} \\
K_k &= M_k \frac{H^T}{[H M_k H^T + r_k]} \\
P_k &= [I - K_k H] M_k \\
\hat{s}_k &= F_{k-1} \hat{s}_{k-1} + K_k [y_k - H F_{k-1} \hat{s}_{k-1}]
\end{aligned}
\tag{30}
$$

The algorithm is usually initialized with

$$
\hat{s}(0) = 0 \quad \text{and} \quad P(0) = cI \quad , \quad c >> 1 \tag{31}
$$

The large covariance matrix $P(0)$ symbolizes a high degree of uncertainty in the initial estimate of $s(0)$.

Kemp extensively investigated the deterministic PLID algorithm in [7]. There are four major conclusions :

- Deadbeat convergence in the deterministic case.

- unbiased parameter estimates in the case of white process noise and zero measurement noise, but a slower convergence.

- Measurement noise leads to bias in the parameter estimates.

- Introducing blockfading memory enables the PLID to track time varying parameters. The blockfading is introduced in the parameter part of the covariance matrix.

In the model

$$s_k = F_{k-1}s_{k-1} + n_k, \quad n_k = \begin{bmatrix} w_{k-1} \\ 0 \end{bmatrix} \tag{32}$$

$$y_k = Hs_k + v_k \tag{33}$$

F_{k-1} is a function of the measurement y_{k-1} and the input u_{k-1}. If noise is added to the input and output signals, the F matrix is different from an F matrix with noise free input and output signals, and therefore no longer reflects the true transition matrix.

Hopkins, in [6], takes into account the noise added to the F-matrix, and develops a biasfree algorithm. Since this thesis investigates the case of outliers in the measurements, it is justified to begin with the bias free algorithm called the Stochastic PLID algorithm.

2.3 The Stochastic PLID for Single Input Single Output Systems.

The short form extended state representation can represent a system with noise on the input, output and states :

$$s_{k+1} = F_k(y_k, \tilde{u}_k)s_k + w_k \tag{34}$$

$$z_k = y_k + v_k = Hs_k + v_k \tag{35}$$

In this case, F_k represents the true but unknown transition matrix, because y_k is the true noisefree system output, and $\tilde{u}_k$ is the true input signal. The measured input signal $\tilde{u}_k = u_k + n_k$ consists of the actual system input u and the measurement noise part n . The deterministic part is a computer generated control signal. The system is single input single output with the n-th order transfer function

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_{n-1}z + b_n}{z^n + a_1 z^{n-1} + \cdots + a_{n-1}z + a_n} \tag{36}$$

The following vectors group the parameters of the A and B polynomials

$$\theta_a = [a_1 \ldots a_n]^T, \theta_b = [b_1 \ldots b_n]^T \tag{37}$$

$$\theta = [a_1 \ldots a_n b_1 \ldots b_n]^T \tag{38}$$

Writing the extended state equations in matrix notation gives

$$\begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} J_{nn} & yI_n & \tilde{u}I_n \\ 0_{2nn} & I_{2n} & \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} w_x(k) \\ w_\theta(k) \end{bmatrix} \tag{39}$$

$$y_k = [0 \ldots 1\ 0 \ldots 0] \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + v_k \tag{40}$$

Inserting the expressions for y_k and $\tilde{u}_k$ in the F-matrix leads to the expressions

$$\begin{aligned} x_{k+1} &= Jx_k + z_k\theta_a(k) + u_k\theta_b(k) - v_k\theta_a(k) + n_k\theta_b(k) + w_x(k) \\ \theta_a(k+1) &= 0x_k + I_n\theta_a(k) + 0\theta_b(k) - 0v_k + 0n_k + w_a(k) \\ \theta_b(k+1) &= 0x_k + 0_n\theta_a(k) + I_n\theta_b(k) - 0v_k + 0n_k + w_b(k) \end{aligned} \tag{41}$$

and in matrix form this looks like

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \theta_a(k) \\ \theta_b(k) \end{bmatrix} &= \begin{bmatrix} J_n & z_k I_n & u_k I_n \\ 0_{nn} & I_{nn} & 0_{nn} \\ 0_{nn} & 0_{nn} & I_{nn} \end{bmatrix} \begin{bmatrix} x(k) \\ \theta_a(k) \\ \theta_b(k) \end{bmatrix} \\ &+ \begin{bmatrix} -\theta_a(k) & \theta_b(k) \\ 0_{n1} & 0_{n1} \\ 0_{n1} & 0_{n1} \end{bmatrix} \begin{bmatrix} v_k \\ n_k \end{bmatrix} + \begin{bmatrix} w_x(k) \\ w_a(k) \\ w_b(k) \end{bmatrix} \end{aligned} \tag{42}$$

In the above expression the measured outputs z_k and the measured inputs u_k are used in F_k. One can augment the noise terms into one term, like

$$\begin{bmatrix} x(k+1) \\ \theta_a(k) \\ \theta_b(k) \end{bmatrix} = \begin{bmatrix} J_n\, z_k I_n\, uI_n \\ 0_{nn}\; I_{nn}\; 0_{nn} \\ 0_{nn}\; 0_{nn}\; I_{nn} \end{bmatrix} \begin{bmatrix} x(k) \\ \theta_a(k) \\ \theta_b(k) \end{bmatrix}$$

$$+ \begin{bmatrix} -\theta_a(k) & \theta_b(k) & I_n\; 0_n\; 0_n \\ 0_{n1} & 0_{n1} & 0_n\; I_n\; 0_n \\ 0_{n1} & 0_{n1} & 0_n\; 0_n\; I_n \end{bmatrix} \begin{bmatrix} v_k \\ n_k \\ w_x(k) \\ w_a(k) \\ w_b(k) \end{bmatrix} \tag{43}$$

The compact version of this equation is

$$s_{k+1} = F_k(z_k, u_k)s_k + G_k\gamma_k \tag{44}$$

where

$$G_k = \begin{bmatrix} -\theta_a(k)\theta_b(k) I_n 0_n 0_n \\ 0_{n1} \quad 0_{n1}\; 0_n I_n 0_n \\ 0_{n1} \quad 0_{n1}\; 0_n 0_n I_n \end{bmatrix}, \gamma = \begin{bmatrix} v_k \\ n_k \\ w_x(k) \\ w_a(k) \\ w_b(k) \end{bmatrix} \tag{45}$$

The stochastic extended state representation is then

$$\begin{aligned} s_{k+1} &= F_k(z_k, u_k)s_k + G\gamma_k \\ z_k &= Hs_k + v_k \end{aligned} \tag{46}$$

Since the extended noise vector γ_k represents the measurement noise v_k, the measurement noise and the state noise will be correlated.

The idea of the stochastic PLID algorithm is to design a minimum variance observer for the previously developed extended state representation. Before this development takes place, one must make certain assumptions about the noise structures. These assumptions are that the noises are white gaussian with covariances :

$$
\begin{aligned}
Q_k^0 &= E\{\gamma_k \gamma_k^T\} \\
&= E\left\{ \begin{bmatrix} v_k \\ n_k \\ w_x(k) \\ w_a(k) \\ w_b(k) \end{bmatrix} [v_k\, n_k\, w_x^T(k)\, w_a^T(k)\, w_b^T(k)] \right\} \\
Q_k^0 &= \begin{bmatrix} R_k & S_{nv}^T(k) & S_{vw_x}^T(k) & & \\ S_{nv}(k) & Q_k & S_{nw_x}^T(k) & 0_{(2+n)2n} & \\ S_{nw_x}(k) & S_{nw_x}(k) & \Sigma_{xx}(k) & & \\ 0_{2n1} & 0_{2n1} & 0_{2nn} & \Sigma_{\theta\theta}(k)_{(2n2n)} & \end{bmatrix}
\end{aligned}
\tag{47}
$$

The dimension of $Q^0(k)$ is $(3n + 2) \times (3n + 2)$.

R, Q, S_{nv} are scalars. $S_{vw_x}^T(k)$ and $S_{nw_x}^T(k)$ are 1 by n dimensional. $\Sigma_{xx}(k)$ is nn. $\Sigma_{\theta\theta}(k)$ is $2n \times 2n$.

It is assumed that the noise disturbing the states x is independent of the noise disturbing the parameters. The covariance between measurement noise and extended state noise is.

$$E[v_k\gamma_k] = \begin{bmatrix} R_k \\ S_{nv}(k) \\ S_{nwx}(k) \\ 0_{2n1} \end{bmatrix} = S \tag{48}$$

The well known observer equations are now given for the extended state representation as

$$\hat{s}_{k+1|k} = F_k\hat{s}_{k|k-1} + K_k(z_k - H\hat{s}_{k|k-1}) \tag{49}$$

The state prediction error $e_{k+1|k}$ is defined as

$$\begin{aligned} e_{k+1|k} &= \hat{s}_{k+1|k} - s_{k+1} \\ &= F_k\hat{s}_{k+1|k} + K_k(z_k - H\hat{s}_{k|k-1}) - F_k s_k - G\gamma_k \\ &= (F_k - K_kH)e_{k|k-1} + K_k v_k - G\gamma_k \end{aligned} \tag{50}$$

The covariance of the prediction error is

$$P_{k+1|k} = E\{e_{k+1|k}e_{k+1|k}^T\} \tag{51}$$

After a lot of tedious algebra, the covariance of the prediction error is found to be

$$P_{k+1|k} = (F_k - K_k H)P_{k|k-1}(F_k - K_k H)^T + K_k R_k K_k^T + E_\psi\{GQ_k^0 G^T\} - K_k S^T E_\psi\{G^T\} - E_\psi\{G\}SK_k^T \tag{52}$$

Where E_ψ is the conditional expectation operator for the measurements up to time k.

To derive the Stochastic PLID Algorithm, the trace of $P_{k+1|k}$ is minimized with respect to K_k, which results in the optimal gain $K_{k|k}$ given as

$$K_{k|k} = [F_k P_{k|k-1} H^T + E_\psi\{G\}S](HP_{k|k-1}H^T + R_k)^{-1} \tag{53}$$

This expression for $K_{k|k}$ is then inserted into the covariance expression yielding

$$P_{k+1|k} = F_k P_{k|k-1} F_k^T + E_\psi\{GQ_k^0 G^T\} - K_{k|k}(HP_{k|k-1}H^T + R_k)K_{k|k}^T \tag{54}$$

The last two equations, together with the observer equation

$$\hat{s}_{k+1|k} = F_k \hat{s}_{k|k-1} + K_{k|k}(z_k - H\hat{s}_{k|k-1}) \tag{55}$$

define the stochastic PLID algorithm. It turns out that there are some computational problems related to this formulation, because the G matrix is a function of the parameter estimates $\hat{\theta}_{k+1|k}$

$$E_\psi\{G\} = E(G \mid \psi_k) = G[E(\theta \mid \psi_k)] = G[\hat{\theta}_{k+1|k}] \tag{56}$$

This result is now used in the computation of

$$E_{\psi}\{G\}S = \begin{bmatrix} -\hat{\theta}_a & \hat{\theta}_b & I_n & 0_{n2n} \\ 0_{2n1} & 0_{2n1} & 0_{2nn} & I_{2n2n} \end{bmatrix} \begin{bmatrix} R_k \\ S_{nv}(k) \\ S_{nw_x}(k) \\ 0_{2n1} \end{bmatrix} = \begin{bmatrix} -\hat{\theta}_a R_k + \hat{\theta}_b S_{nv}(k) + I_n S_{nwx}(k) \\ 0 \end{bmatrix} \tag{57}$$

This result is very useful because the θ part of the gain

$$K_{k|k} = \begin{bmatrix} K^x_{k|k} \\ K^{\theta}_{k|k} \end{bmatrix} \tag{58}$$

is now independent of $\hat{\theta}_{k+1|k}$ and can be computed directly from the following intermediate gain equation.

$$K_k = \begin{bmatrix} K^x_k \\ K^{\theta}_k \end{bmatrix} = [F_k P_{k|k-1} H^T](H P_{k|k-1} H^T + R_k)^{-1} \tag{59}$$

Because $K^{\theta}_{k|k} = K^{\theta}_k$, the θ part of the extended state vector can now be calculated by the also intermediate prediction equation

$$\hat{s}_{k+1} = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} = F_k \hat{s}_{k|k-1} + K_k(z_k - H\hat{s}_{k|k-1}) \tag{60}$$

with $\hat{\theta}_{k+1|k} = \hat{\theta}_{k+1}$. Now the x-part of the gain vector can be calculated by noticing that

$$\begin{aligned} K_{k|k} &= [F_k P_{k|k-1} H^T + E_\psi\{G\}S](HP_{k|k-1}H^T + R_k)^{-1} \\ &= [F_k P_{k|k-1} H^T](HP_{k|k-1}H^T + R_k)^{-1} + \\ &E_\psi\{G\}S(HP_{k|k-1}H^T + R_k)^{-1} \\ &= K_k + E_\psi\{G\}S(HP_{k|k-1}H^T + R_k)^{-1} \\ &= \begin{bmatrix} K_k^x \\ K_k^\theta \end{bmatrix} + \begin{bmatrix} -\theta_a R_k + \theta_b S_{nv}(k) + I_n S_{nw_x}(k) \\ 0 \end{bmatrix} \\ &(HP_{k|k-1}H^T + R_k)^{-1} \end{aligned} \tag{61}$$

From this last equation it is given that the x part of the gain is given by.

$$K_{k|k}^x = K_k^x + [\,-\theta_a R_k + \theta_b S_{nv}(k) + I_n S_{nw_x}(k)](HP_{k|k-1}H^T + R_k)^{-1} \tag{62}$$

Both the state and the parameter part of the gain vector are now determined. The remaining state prediction is finished with

$$\begin{aligned} \hat{x}_{k+1|k} = \hat{x}_{k+1} + [\,&-\theta_a R_k + \theta_b S_{nv}(k) + I_n S_{nw_x}(k)] \\ &(HP_{k|k-1}H^T + R_k)^{-1}(z_k - H\hat{s}_{k|k-1}) \end{aligned} \tag{63}$$

The prediction error covariance still remains to be determined.

$$P_{k+1|k} = F_k P_{k|k-1} F_k^T + E_\psi\{GQ_k^0 G^T\} - (K_{k|k-1}H^T + R_k)K_{k|k}^T \tag{64}$$

The only unknown in this equation is $E_\psi\{GQ_k^0 G^T\}$. This turns out to be a very complex expression. The result for a single input single output system is

$$E_\psi\{GQ_k^0G^T\} = \begin{bmatrix} \#_{nn} & 0_{n2n} \\ 0_{2nn} & 0_{2n2n} \end{bmatrix} \tag{65}$$

where

$$\begin{aligned} \# = {} & P_{1,1}^{\theta\theta}(k+1 \mid k)R_k + P_{2,2}^{\theta\theta}(k+1 \mid k)Q_k - P_{1,2}^{\theta\theta}(k+1 \mid k)S_{nv}(k) \\ & - P_{2,1}^{\theta\theta}(k+1 \mid k)S_{nv}(k) + E_\psi(G^x)Q_k^0E_\psi(G^x)^T \end{aligned} \tag{66}$$

The $P^{\theta\theta}$ terms come from the decomposition of $P_{k+1|k}$ into

$$P_{k+1|k} = \begin{bmatrix} P_{k+1|k}^{xx} & P_{k+1|k}^{x\theta} \\ P_{k+1|k}^{\theta x} & P_{k+1|k}^{\theta\theta} \end{bmatrix} \tag{67}$$

$P_{k+1|k}^{\theta\theta}$ is further divided into 4 n by n matrices

$$P_{k+1|k}^{\theta\theta} = \begin{bmatrix} P_{1,1}^{\theta\theta}(k+1 \mid k) & P_{1,2}^{\theta\theta}(k+1 \mid k) \\ P_{2,1}^{\theta\theta}(k+1 \mid k) & P_{2,2}^{\theta\theta}(k+1 \mid k) \end{bmatrix} \tag{68}$$

Evaluating from the last equation $E_\psi(G^x)Q_k^0E_\psi(G^x)^T$ gives

$$
\begin{aligned}
&E_\psi(G^x)Q_k^0E_\psi(G^x)^T = \\
&[-\theta_a\theta_bI_n0_{n2n}]\begin{bmatrix} R_k & S_{nv}^T(k) & S_{vw_x}^T(k) & \\ S_{nv}(k) & Q_k & S_{nw_x}^T(k) & 0_{(2+n)2n} \\ S_{nw_x}(k) & S_{nw_x}(k) & \Sigma_{xx}(k) & \\ 0_{2n1} & 0_{2n1} & 0_{2nn} & \Sigma_{\theta\theta}(k)_{(2n2n)} \end{bmatrix}\begin{bmatrix} -\theta_a^T \\ \theta_b^T \\ I_n \\ 0 \end{bmatrix} \\
&= \theta_aR_k\theta_a^T - \theta_bS_{nv}\theta_a^T - S_{nw_x}\theta_a^T - \\
&\theta_aS_{nv}\theta_b^T + \theta_bQ_k\theta_b^T + S_{nw_x}\theta_b^T + \\
&\theta_aS_{vw_x}^T + \theta_bS_{nw_x}^T + \Sigma_{xx}(k) = \Delta
\end{aligned}
\tag{69}
$$

The notation $(k+1 \mid k)$ is implied in the above equation ($\theta = \hat{\theta}_{k+1|k}$). As in the gain computation, the prediction error covariance is calculated via an intermediate step.

$$
\begin{aligned}
P_{k+1} = F_kP_{k|k-1}F_k^T + E(G \mid \psi)Q_k^0E(G \mid \psi) \\
- K_{k|k}(HP_{k|k-1}H^T + R_k)K_{k|k}^T
\end{aligned}
\tag{70}
$$

$$
\begin{bmatrix} P_{k+1}^{xx} & P_{k+1}^{x\theta} \\ P_{k+1}^{\theta x} & P_{k+1}^{\theta\theta} \end{bmatrix} = F_kP_{k|k-1}F_k^T + \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} - K_{k|k}(HP_{k|k-1}H^T + R_k)K_{k|k}^T \tag{71}
$$

From the structure of $E_\psi(GQ_k^0G^T)$ and the above equation, it is clear that

$$
\begin{aligned}
P_{k+1|k}^{x\theta} &= P_{k+1}^{x\theta} \\
P_{k+1|k}^{\theta x} &= P_{k+1}^{\theta x} \\
P_{k+1|k}^{\theta\theta} &= P_{k+1}^{\theta\theta}
\end{aligned}
\tag{72}
$$

The last part of the prediction error covariance is now computed as

$$P^{xx}_{k+1|k} = P^{xx}_{k+1} + P^{\theta\theta}_{1,1}(k+1 \mid k)R_k + P^{\theta\theta}_{2,2}(k+1 \mid k)Q_k \\ - P^{\theta\theta}_{1,2}(k+1 \mid k)S_{nv}(k) - P^{\theta\theta}_{2,1}(k+1 \mid k)S_{nv}(k) \tag{73}$$

In his work Hopkins specified the PLID algorithm with the above equations. These are the equations that will be modified to robustify the PLID algorithm. In order to do so, the next chapter will present the concepts of robustness theory.

3.0 Robustness Concepts

In the field of estimation theory, the concept of robustness has gained increased attention because researchers have come to the conclusion that data is not always distributed according to the Gaussian assumption. This chapter describes the most important concepts of robustness in estimation in relation to this particular study.

Engineers want an estimator to be robust because the assumptions about the physical world and the underlying distribution of data do not always hold. Therefore if a minor error exists between the mathematical model of the data distribution and the real data distribution exists, the estimator should only produce a small error in the output.

Robustness in estimation means insensitivity to small deviations from the underlying statistical assumptions. *"Distributional robustness"* means that a small deviation in the shape of the true underlying distribution from the assumed model (usually the Gaussian distribution) should only cause a small deviation from the

assumed output. According to [5] p.4 "Any reasonable, formal or informal, procedure for rejecting outliers will prevent the worst." By preventing the worst, the author means limiting the bias in the case of outlier contaminated data. *Outliers* are measurements that do not fit the underlying assumption about the distribution. A reasonable approach to robustness would therefore include

- cleaning of the data by rejection and
- the use of classical estimation methods (Gaussian, least squares)

Huber argues that this approach will not work because of the following reasons :

- It is difficult to separate the two steps clearly.
- Cleaned data will consist of false rejections and false retention.
- It is an empirical fact that the best robust procedures are better than the best rejection rules.

Therefore one must aim at developing estimators based on robustness rather than estimators that will use rejection rules.

For notational reasons it is convenient to introduce an estimator as a functional of the data distribution. An estimator transforms measurements into an estimate, such as

$$\hat{\theta}_m = T_m(z_1, \ldots, z_m) \tag{74}$$

The measurements are distributed empirically according to $F_m(z, \theta)$, for which

$$\lim_{m \to \infty} F_m(z, \theta) = F(z, \theta) \tag{75}$$

is the true distribution. For example the sample mean is

$$\begin{aligned} \hat{\theta} &= \frac{1}{m} \sum_{i=1}^{m} z_i \\ &= \frac{1}{m} \sum_{i=1}^{m} \int_{-\infty}^{+\infty} u \delta(u - z_i) du \\ &= \int_{-\infty}^{+\infty} u \frac{1}{m} \sum_{i=1}^{m} \delta(u - z_i) du \end{aligned} \tag{76}$$

In this case the empirical probability density function is given as

$$f_m(z) = \frac{1}{m} \sum_{i=1}^{m} \delta(u - z_i) \tag{77}$$

Where δ is the Dirac impulse function. If m approaches infinity, then $f_m(z)$ approaches the continuous density function and the well known expression for the mean emerges

$$\hat{\theta} = \int_{-\infty}^{+\infty} uf(u)du = \int_{-\infty}^{+\infty} udF(u) \tag{78}$$

In general the functional is written as

$$T(F_m) = \int \psi(z_i)dF_m \tag{79}$$

Huber states in [5] p.10 that a linear functional T is robust everywhere if and only if the corresponding ψ is bounded and continuous. That ψ be continuous is important in the case of rounding errors. If for example ψ is discontinuous and F_m puts probability weight on to discontinuity points of ψ then many small rounding errors in z_i may produce large changes in the estimate $T(F_m)$. When dealing with outliers in the measurements, the underlying distribution will no longer be purely Gaussian (if that is the model one is assuming), but rather a mix of Gaussian in the center and heavy tails to the sides. In this case the least squares estimator will no longer be optimal and can produce biased estimates. It is essential to consider other estimators than the Gaussian based least squares. The branch of statistics that deals with building estimators for these contaminated distributions is called *Robust Estimation.*

3.1 *Robustness Measures*

It is convenient and necessary to have some measure to tell how small changes in the underlying distribution affects the estimate. Let the measurements be distributed according to

$$F = (1 - \varepsilon)F_0 + \varepsilon H \tag{80}$$

Where ε is the degree of contamination with outliers, F_0 is the assumed model which ussually is the Gaussian distribution, H is the outlier distribution and F is the resulting real distribution. F is called the *"gross error model"*. Hampels research in the late sixties and early seventies is credited for the following robustness measures.([4], [14] and [15]).

An important measure for the robustness of an estimator is the maximum bias. This measure depends on the degree ε of contamination.

$$b_1(\varepsilon) = \sup_F | T(F) - T(F_0) | \tag{81}$$

With this measure one can compare the robustness of two estimators at the degree of contamination ε. The maximum bias is given for the worst case distribution F.

An estimator is said to be *"qualitatively robust"* if a small deviation between the assumed underlying distribution and the real distribution F only results in a minor difference between the two estimates. In this case the estimator produces

only a small bias. Or in other words, the estimator T is continuous at the distribution F_0. Saying that the functional T is continuous at F_0 is equivalent to stating that the maximum bias is continuous at $\varepsilon = 0$. Having defined what qualitative robustness means to an estimator, it is also important to have a measure for how much contamination an estimator can handle and still produce decent estimates. One such measure is the breakdown point ε, which is the maximum fraction of outliers in the data the estimator can accept such that the maximum bias is bounded.

$$\varepsilon^* = \max\{\varepsilon; b_1(\varepsilon) = \sup_F | T(F) - T(F_0) | < \infty\} \tag{82}$$

The maximum breakdown point any estimator can achieve is 50%, because the estimator relies on the majority of the data.

Another important tool in the robustness field is the influence function, which is especially useful in predicting the robustness of M-estimators (to be covered later). In the late sixties and early seventies, F. Hampel laid the foundation of the influence function approach to robustness. The influence function expresses how much an additional measurement will change the outcome of an estimate. The influence function is expressed as

$$IF(x,F_0,T) = \lim_{\varepsilon \to 0} \frac{T((1-\varepsilon)F_0 + \varepsilon\delta_x) - T(F_0)}{\varepsilon} \tag{83}$$

Where δ_x denotes the probability mass 1 at the value x. Hampel [3] shows that the asymptotic variance of the estimator given the assumed/model distribution F_0 is

$$V(T,F_0) = \int_{-\infty}^{+\infty} (IF(x))^2 dF_0(x) \tag{84}$$

The maximum absolute value of the Influence Function is called the *gross error sensitivity* γ^*.

$$\gamma = \max\{IF(x)\} \tag{85}$$

An important property of the gross error sensitivity is that it is related to the maximum bias of the estimator. The influence function can be seen as the derivative of $b_1(\varepsilon)$ at $\varepsilon = 0$, because the functional $T((1-\varepsilon)F_0 + \varepsilon\delta_x)$ gets arbitrarily close to $T(F_0)$ as ε goes to 0, and dividing by ε then gives the differential quotient. One must remember that the following equation defines the bias

$$b_1(\varepsilon) = \sup_F | T(F) - T(F_0) | \tag{86}$$

Therefore the Influence Function approximates the bias that an outlier causes, by

$$\text{asymptotic bias } b = \varepsilon\gamma^* \tag{87}$$

Boundedness of the Influence Function is therefore a useful property, with regard to robustness, for an estimator to possess. If the gross error sensitivity γ is finite or bounded, then the estimator is said to be B-robust at the distribution F_0.

Another measure derived from the influence function is the *local-shift sensitivity* λ, which one defines as the largest absolute value of the derivative of the influence function. The local shift sensitivity is a measure for the worst effect that small changes in the data can have on the estimate. These small changes can for example come from rounding errors. If the influence function has any discontinuities, then the local shift sensitivity will go to infinity at the points of discontinuity.

A third measure derived from the influence function is the *rejection point* ρ which tells when the influence function reaches zero, and that data beyond this point have no influence on the estimate.

The influence function deals with the robustness of the estimators' bias. This necessarily led to consideration of the role played by the robustness of the efficiency or the asymptotic variance of the estimator. This study will briefly address that problem by giving the definition of the change of variance function (CVF) and some derived properties of The CVF.

The *change of variance function* is defined as the derivative of the asymptotic variance with respect to the degree of contamination at $\varepsilon = 0$. The CVF is a measure for the local robustness of the asymptotic variance.

$$CVF = \frac{\delta V(\hat{\theta}, F_0)}{\delta \varepsilon} \tag{88}$$

The *change of variance sensitivity* of the M-estimators is defined as

$$\kappa^* = \sup\left[\frac{CVF}{V(\psi, \hat{\theta}, F_0)}\right] \tag{89}$$

If κ^* is finite the estimator is said to be *V-robust* and a small amount of contamination will have a limited effect on the asymptotic variance.

3.2 Maximum Likelihood and M-Estimators

The ordinary maximum likelihood estimators are based on the knowledge or assumed knowledge of an underlying distribution function f, such that the product

$$L = \prod_{i=1}^{m} f(z_i, \theta) \tag{90}$$

is maximized, or equivalently the sum

$$J(\theta) = \sum_{i=1}^{m} -\ln(f(z_i, \theta)) \tag{91}$$

is minimized. Here one assumes that the measurements are identical and independently distributed.

Instead of minimizing the sum of logarithms, one can specify other functions to minimize. These functions do not have to be related to the assumed distribution of the data, but can be chosen to make the estimator robust against deviations from the assumed distribution. These estimators are called M-estimators or generalized maximum likelihood estimators.

The *M-estimators* are defined by minimizing the criterion

$$\hat{\theta} = \min_{\theta} J(\theta) = \min_{\theta} \sum_{i=1}^{m} \rho(z_i, \theta) \tag{92}$$

In the case of estimating the location parameter the criterion is

$$\hat{\theta} = \min_{\theta} J(\theta) = \min_{\theta} \sum_{i=1}^{m} \rho(z_i - \theta) \tag{93}$$

Differentiating with respect to θ and setting equal to 0 yields the equation

$$\sum_{i=1}^{m} \psi(z_i - \theta) = 0 \quad , \quad \psi = \frac{d\rho}{d\theta} \tag{94}$$

In general ψ is a nonlinear function, therefore the argument of ψ has to be normalized for the estimator to give consistent estimates. The residual

$$r_i = z_i - \theta \tag{95}$$

is normalized by dividing by a scale parameter s. The scale parameter gives the variation of the residual. So the normalized M-estimator is the solution to the equation

$$\sum_{i=1}^{m} \psi\left(\frac{r_i}{s}\right) = 0 \tag{96}$$

One must also estimate the scale parameter. This estimate of scale has to be robust. How to obtain a robust estimate of scale will be described later. In the case of *ON*-line computation the scale estimate is somewhat difficult to obtain. For now one must assume that s has been estimated. Huber proves that the M-estimator of location is continuous at the assumed distribution, which is the same as saying that the M-estimator is qualitatively robust if and only if ψ is a bounded function.[2] The breakdown point of the M-estimator is $\varepsilon = 0.5$ if $\psi(-\infty) = \psi(+\infty)$. The Influence Function of the M-estimator of location is given as

$$IF(z) = \frac{\psi(z-\theta)}{-E\{\dfrac{d\psi(z-\theta)}{d\theta}\}} = k\psi(z-\theta) \tag{97}$$

Hence the ψ function directly tells how much influence one outlier will have on the M-estimator. This makes the M-estimator intuitively very appealing from a design point of view.

---

[2] Huber, 1981. p54.

The cost of robustness is the loss of efficiency at the Gaussian distribution, which appears as an increase in the asymptotic variance. Hampel derives a redescending M-estimator that is optimal in the sense that it minimizes the asymptotic variance, given the constraint that the gross error sensitivity is less than a chosen value.[3] The optimal estimator is defined by the ψ function

$$\psi = \begin{array}{c} x \; if 0 < |x| \\ \sqrt{A(k-1)}\,\tanh[0.5\sqrt{(k-1)B^2 \div A}\,(r-|x|)]sign(x)\; if p \le x \le r \\ 0 \; if \; |x| > r \end{array} \quad (98)$$

3.3 *Robust Estimates of Scale*

To use the M-estimators it is necessary to have a robust estimate of the scale. The scale shows how much the measurements differ from the location parameter. One such scale estimator is the *Median Absolute Deviation* from the median.

$$MAD = \underset{j}{med}\,|z_j - \underset{i}{med}(z_i)| \quad (99)$$

For MAD to be consistent with the standard deviation in the Gaussian case, MAD is multiplied by 1.483. Since the median requires that all the data be available at the time when the computing begins, it is impossible to use the MAD estimator when performing on-line calculations, as is the case with Kalman Fil-

[3] Hampel,1986, p.160

tering. The next chapter concentrates on methods to robustify the Kalman filter that have been described in the filtering literature. These ideas will then be extended to the case of the PLID joint parameter and state estimator.

4.0 Robust Kalman Filters

In this chapter the basic Kalman Filter is reviewed, and from the literature examples of robust Kalman filters are given.

The Kalman Filter is an optimal observer for a linear dynamic system given by the equations

$$\begin{aligned} x_k &= \phi x_{k-1} + \Gamma u_{k-1} + w_{k-1} \\ z_k &= H x_k + v_k \end{aligned} \tag{100}$$

where the process and measurement noise have the covariances

$$\begin{aligned} E\{w_k w_k^T\} &= Q_k \\ E\{v_k v_k^T\} &= R_k \end{aligned} \tag{101}$$

The Kalman filter can be described by the five equations

$$\hat{x}_k^- = \phi \hat{x}_{k-1}^+ + \Gamma u_{k-1} \tag{102}$$

$$M_k = \phi P_{k-1} \phi^T + Q_k \tag{103}$$

$$K_k = M_k H^T (H M_k H^T + R_k) \tag{104}$$

$$P_k = M_k - K_k H M_k \tag{105}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \tag{106}$$

The gain K_k minimizes the covariance of the state estimate error

$$P = E\{(x - \hat{x})(x - \hat{x})^T\} \tag{107}$$

The equations can be divided into two groups. Pre- and post measurement equations. Equation (103) is a pre-measurement state prediction. Equation (104) is the prediction error covariance. Equation (105) is the gain calculation. Equation (106) is the filter error covariance after the measurement and (107) is the updated state estimate. Although the covariances in the basic Kalman filter are independent of the actual measurement, it is convenient and later intuitively appealing to divide them into pre and post measurement prediction error covariances.

A basic feature of the Kalman filter is that the gain K_k reflects the confidence that is placed in the measurement z_k. That is if R_k is large because of a high noise level, then K_k will be small indicating that one has more confidence in the pre-measurement state propagation, i.e. the measurement is down weighted. For the

post-measurement prediction error covariance this means less decrease in equation (106), since M_k and P_k are positive definite matrices.

In a sense one could say that the Kalman Filter is already robust, when it down weights measurements at high noise levels. This is however a truth with some conditions. Because one has to know the covariance R_k as a function of time, which is in practice very unlikely. It can be shown that the Kalman filter is a weighted least squares estimator which fits both the measurements z_k and the predicted state $\hat{x}_{k-1}^{+}$. As a result the noise still has to be Gaussian. This is shown in appendix A. So the need to robustify the Kalman filter still exists.

4.1 Robust Kalman Filters from The Literature

This paragraph describes some techniques found in the literature for robustifying the Kalman Filter. The ideas found can be grouped into three categories.

1. use a nonlinear influence function on the prediction error to give a nonlinear effective gain. The covariance is also weighted by some nonlinear function of the scaled innovation (prediction error).

2. Switch between two to n noise covariances (different noise models). This influences both the gain and covariance updating. The switching approach appears in the litterature with more or less elegance and practicality. Both variants have been encountered.

3. Choose an estimator that minimizes the assymptotic variance given that the data distribution is the worst case or least favorable distribution. (The Huber Min-Max approach).

Three papers were found that uses influence function and weighting functions to robustify the Kalman Filter. In the work [19], C.J Masreliez suggests a robust filter using the equations

$$\begin{aligned} \hat{x}_k &= \bar{x}_k + M_k H^T g_k(z_k) \\ P_k &= M_k - M_k H^T G(z_k) H_k M_k \\ M_{k+1} &= \phi P_k \phi^T + Q_k \end{aligned} \tag{108}$$

The influence function g and the weighting function G are given by the equations

$$\{g_k(z_k)\} = -\left[\frac{\delta p(z_k \mid z^{k-1})}{\delta(z_k)_i}\right]\left[p(z_k \mid z^{k-1})\right]^{-1} \tag{109}$$

Where $p(z_k \mid z^{k-1})$is the conditional probability density function of the measurement z, which is computed by convolution of the density functions for Hx and the measurement noise, since these two variables are independent and added in the time domain. usually one assumes that Hx is Gaussian distributed and the measurement noise is a sum of Gaussian distributions. $g_k(z_k)$ is called the score function for the density $p(z_k \mid z^{k-1})$. The weighting matrix G is given by

$$\{G_k(z_k)\}_{ij} = \frac{\delta\{(g_k(z_k))\}_i}{\delta(z_k)_j} \tag{110}$$

This filter requires knowledge of the involved noise statistics, but Masreliez suggests that the g function be approximated directly, without calculating the probability density function for z_k. The filter by Masreliez is derived for the case were the process noise is Gaussian and the measurement noise is non-Gaussian.

In the work [17], R. Douglas Martin and David J. Thomson take a similar approach to the problem of robust Kalman filtering as Masreliez. Although the paper by Martin and Thomson [17] deals with spectrum estimation, they have a section on what they call data cleaning, which is a method of down weighting outliers. They down weight outliers by a nonlinear transformation of the scaled innovation term :

$$\psi\left[\frac{y_t - \hat{y}_t^{t-1}}{s_t}\right] \tag{111}$$

Their state representation is given as

$$x_t = \phi x_{t-1} + u_t + Q_t \tag{112}$$

The paper deals with a p-th order autoregressive process, which is extended to a state space dynamical system with measurement equation

$$z_t = Hx_t + V_t \tag{113}$$

The transition matrix and observation matrix H have the form :

$$\phi = \begin{bmatrix} \phi_1 \phi_2 \cdots \phi_{1-p} \phi_p \\ 1 \ 0 \ldots \ 0 \ \ 0 \\ 0 \ 1 \ldots \ 0 \ \ 0 \\ \ldots \ 0 \ldots \ \ldots \ \ \ldots \\ 0 \ \ldots \ldots \ 1 \ \ 0 \end{bmatrix} \tag{114}$$

$$H = [1 \ 0 \ \ldots \ 0 \ 0]$$

The filter cleaner is given by the equations

$$\begin{aligned} \hat{x}_t &= \phi \hat{x}_{t-1} + \tilde{m}_t s_t \psi \left[\frac{z_t - \hat{y}_t^{t-1}}{s_t} \right] \\ \tilde{m}_t &= M_t H^T / s_t^2 \\ M_t &= \phi P \phi^T + Q \\ P_t &= M_t - w \left[\frac{z_t - \hat{y}_t^{t-1}}{s_t} \right] M_t H^T H M_t^T / s_t^2 , \quad s_t^2 = m_{11,t} \text{ (in this case)} \end{aligned} \tag{115}$$

Where $\hat{y}_t^{t-1}$ is the estimated output. If ψ is the identity function and w is identically 1, and s_t^2 is replaced with R , then the usual Kalman filter emerges.

The function ψ is chosen to be bounded according to the discussion on influence functions in the previous chapter, and w is a positive even function with maximum $w(0) = 1$. The parameter s_t^2 is the robust estimate of scale of the innovation or prediction error. It should not be a major problem to extend this filter to other state representations than the above.

A filter very similar to the above is found in the work [18] by C.J Masreliez and R.D Martin. In this paper the robust Kalman filter is derived as the

filter/estimator that minimizes the maximum asymptotic variance over a class of noise distributions. The equations that constitute this filter are very similar to the previous filter and shall not be repeated here.

A final remark on the influence and weighting function approach to robustifying the Kalman filter is that ψ and w should be chosen such that in the case of no outliers in the measurement noise, the filters should approach the conventional Kalman filter. This is to achieve effieciency at the Gaussian distribution.

Another approach to Robust Kalman filtering is to model the measurement noise as a sum of several Gaussian distributions, and then somehow estimate or guess from which distribution the measurements are corrupted, and then use the specific covariance in the traditional Kalman filter. The actual filter then consists of several filters in parallel and the signal is then switched to the filter with the 'correct' covariance.

An example of switching between different distributions and covariances is given in [23] by Tanaka and Katayama. This paper treats the case of Gaussian sum distributions in both the observation and process noise. The idea presented in this paper is to define selection parameters which select the appropriate noise distribution at the current time. The chosen distributions are then used in the Kalman filter equations. According to the authors, the selection parameters are estimated by examining a number of future observations.

Two papers that deal with switching between distributions and which deserve a closer look are [11] and [12]. They assume a dynamic system of the form

$$x_{n+1} = x_n + \eta_{n+1}\,, \quad \text{var}(\eta) = b^2$$
$$y_{n+1} = Cx_n + \zeta_{n+1}\,, \quad \text{var}(\zeta) = B^2 \tag{116}$$

the Kalman filter is given by

$$\hat{x}_{n+1} = A\hat{x}_n + \frac{AP_nC}{B^2 + C^2P_n}(y_{n+1} - C\hat{x}_n)$$
$$P_{n+1} = A^2P_n + b^2 - \frac{(AP_nC)^2}{B^2 + C^2P_n} \tag{117}$$

It is assumed that the measurement noise can be represented by the contaminated distribution

$$\zeta_n = (1 - \theta_n)\zeta_n^{(1)} + \theta_n\zeta_n^{(2)}\,, \quad var(\zeta^{(1)}) = B^2\,, \quad var(\zeta^{(2)}) = D^2 \tag{118}$$

where θ is a random variable that takes the value 1 with probability ε.

The superscripts 1 and 2 indicate the two different distributions, hence when $\theta = 1$ the noise is coming from distribution 2, and vice versa if $\theta = 0$. For θ and D known the optimal filter becomes

$$\hat{x}_{n+1} = A\hat{x}_n + \frac{AP_nC}{(1 - \varepsilon)B^2 + \varepsilon D^2 + C^2P_n}(y_{n+1} - C\hat{x}_n)$$
$$P_{n+1} = A^2P_n + b^2 - \frac{(AP_nC)^2}{(1 - \varepsilon)B^2 + \varepsilon D^2 + C^2P_n} \tag{119}$$

This filter takes into account the larger covariance from D due to outliers which then reduces the gain and keeps the estimator covariance at a higher level. But

if one could estimate when the outliers occur, then it is only necessary to reduce the gain at those instants. To do so introduce the estimate $\hat{\theta}$ of the noise switch into the optimal filter.

$$\begin{aligned}\hat{x}_{n+1} &= A\hat{x}_n + \frac{AP_nC}{(1-\hat{\theta})B^2 + \hat{\theta}D^2 + C^2P_n}(y_{n+1} - C\hat{x}_n) \\ P_{n+1} &= A^2P_n + b^2 - \frac{(AP_nC)^2}{(1-\hat{\theta})B^2 + \hat{\theta}D^2 + C^2P_n}\end{aligned} \tag{120}$$

$\hat{\theta}$ is set to 1 when the prediction error exceeds some preset threshold value and is set to 0 for smaller prediction errors. The threshold value depends on the robust estimate of scale which is given by P. The idea of switching between two distributions can be seen as a special case of the influence and weighting function approach.

The work in [22] proposes to use the Huber min max approach to robust system identification. The idea of the min max approach is to find an estimator that minimizes the maximum asymptotic variance of the estimator when the disturbances are distributed according to the least favorable distribution. The authors consider the discrete time system

$$y_{k+1} = \phi_k^T\theta + v_{k+1} \tag{121}$$

They propose a robust algorithm of the form

$$
\begin{aligned}
\hat{\theta}_{k+1} &= \hat{\theta}_k + \Gamma_k \phi_k^T T_k \psi(v_k) \\
v_k &= y_{k+1} - \phi_k^T \hat{\theta}_k \\
T_k &= [\phi_k^T P_k \phi_k + r]^{\frac{-1}{2}} \\
P_{k+1} &= P_k - P_k \phi_k^T [\phi_k P_k \phi_k^T + r]^{-1} \phi_k^T P_k \\
\Gamma_{k+1} &= \Gamma_k - \Gamma_k \phi_k^T [S_k \phi_k \Gamma_k \phi_k^T + r]^{-1} \phi_k^T \Gamma_k s_k
\end{aligned}
\tag{122}
$$

Where S_k linearizes the nonlinear ψ function as

$$
\psi(v_k) \simeq S_k v_k \tag{123}
$$

The results for this robustification scheme show that the convergence of the parameter estimates depends on the initial conditions, especially the weighting matrix $\Gamma(0)$. The best results are obtained when one chooses the linearization factor s_k to be equal to the Fisher information

$$
S = E_p\{\psi'(v)\} = I(p) \tag{124}
$$

4.2 *Robustifying the PLID algorithm*

This section describes a method to make the PLID algorithm resistant to the occurrence of outliers in the measurements. The idea is to extend the robust filters in [17], [18] and [19] to fit the PLID algorithm. Recall that the PLID algorithm is given by the equations

$$\tilde{K}_k = \begin{bmatrix} \tilde{K}_k^x \\ \tilde{K}_k^\theta \end{bmatrix} = F_k P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} \tag{125}$$

$$\tilde{S}_{k+1} = \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{\theta}_{k+1} \end{bmatrix} = F_k \hat{S}_{k|k-1} + \tilde{K}_k (z_k - H\hat{S}_{k|k-1}) \tag{126}$$

$$K_k^x = \tilde{K}_k^x + E(G^x) s_k (H P_{k|k-1} H^T + R_k)^{-1} \tag{127}$$

$$\hat{x}_{k+1|k} = \tilde{x}_{k+1} + E(G^x) s_k (H P_{k|k-1} H^T + R_k)^{-1} (z_k - H\hat{S}_{k|k-1}) \tag{128}$$

$$\begin{aligned} P_{k+1|k} = {} & F_k P_{k|k-1} F_k^T + E(G^x) Q_k^0 E(G^x)^T \\ & - K_k (H P_{k|k-1} H^T + R_k) K_k^T \end{aligned} \tag{129}$$

$$\begin{aligned} P_{k+1|k}^{xx} = {} & P_{k+1|k}^{\theta\theta} + P_{1,1}^{\theta\theta}(k+1 \mid k) R_k + P_{k+1|k}^{\theta\theta} + P_{2,2}^{\theta\theta}(k+1 \mid k) Q_k \\ & - P_{k+1|k}^{\theta\theta} + P_{1,2}^{\theta\theta}(k+1 \mid k) s_{vw} - P_{k+1|k}^{\theta\theta} + P_{2,1}^{\theta\theta}(k+1 \mid k) s_{vw} \end{aligned} \tag{130}$$

In [17], [18] and [19] the approach is to use a nonlinear transformation of the scaled residual and weighting the updating of the covariance matrix. This means that equations (127), (129) and (130) are changed to

$$\tilde{S}_{k+1} = \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{\theta}_{k+1} \end{bmatrix} = F_k \hat{S}_{k|k-1} + \text{scale}\, \tilde{\bullet}_k \psi \left[\frac{z_k - H\hat{S}_{k|k-1}}{\text{scale}} \right] \tag{131}$$

$$x_{k+1} = \tilde{x}_{k+1} + E(G^x)S(HP_{k|k-1}H^T + R_k)^{-1}$$
$$\text{scale} \bullet \psi\left[\frac{z_k - H\hat{S}_{k|k-1}}{\text{scale}}\right] \tag{132}$$

$$P_{k+1|k} = F_k P_{k|k-1} F_k^T + E(G^x)Q_k^0 E(G^x)^T$$
$$- weight K_k(HP_{k|k-1}H^T + R_k)K_k^T \tag{133}$$

The weight function is chosen as an even function with maximum 1 for a residual of 0. The weight function decreases to 0 as the residual grows larger, thus slowing the decrease of the covariance down. The purpose of the weighting function is to robustify the covariance equation and for this thesis is chosen to be ψ divided by the scaled residual. The scale factor is chosen as

$$\text{scale} = \sqrt{HP_{k|k-1}H^T + R_k} \tag{134}$$

which is equal to the square root of the robustified covariance of the residual. A third change must be introduced in the PLID algorithm to make it robust because the transition matrix is a function of the input and output measurements. Instead of using the unfiltered measurements z_k, the measurements are prefiltered through the ψ function and added to the prediction. The equation is

$$z'_k = H\hat{S}_{k|k-1} + \text{scale} \bullet \psi\left[\frac{z_k - H\hat{S}_{k|k-1}}{\text{scale}}\right] \tag{135}$$

This arrangement prevents dangerous outliers from entering the transition matrix and causing havoc. As will be seen later through simulations, this is a very promising way to make the PLID algorithm robust. In the simulations three different functions are considered. These are

1. The Huber type

$$\psi(x) = \begin{bmatrix} b & \text{if} \quad x > b \\ x & \text{if} \quad |x| \le b \\ -b & \text{if} \quad x < -b \end{bmatrix} \tag{136}$$

2. The Hampel optimal hyperbolic tangent type (soft rejection)

$$\psi(x) = \begin{bmatrix} x & \text{if} \quad |x| < p \\ \sqrt{A}\,(k-1)\tanh\left[0.5\sqrt{((k-1)B^2/A)}\,(r-|x|)\right]sign(x), & \\ \quad \text{if} \quad p \le |x| \le r & \\ 0 & \text{if} \quad |x| > r \end{bmatrix} \tag{137}$$

3. Huber type skipped mean. (hard rejection).

$$\psi(x) = \begin{Bmatrix} 0 & \text{if} \quad |x| > b \\ x & \text{if} \quad |x| \le b \end{Bmatrix} \tag{138}$$

The corresponding weighting functions w(x) are found by dividing the ψ functions by x.

4.3 *Robust off line identification.*

The previous paragraphs proposed how to robustify the on-line PLID Algorithm with properly chosen nonlinear influence functions. The nonlinearity of the influence functions pose a problem in the choice of the initial values for the PLID, because it is not guaranteed that the PLID will converge to the true parameter values in the case where measurements are contaminated with outliers. Simulations in the next chapter illustrate this fact. It was mentioned earlier that the *OFF*-line M-estimators require that the starting point of the iteration is robust, one assumes therefore that the robust PLID requires robust initial conditions. It is therefore necessary to find a robust *OFF*-line method of system identification that will provide the PLID with a robust initial guess.

There exists high breakdown estimators for the regression problem with breakdown points of 50%. Two examples are the *"Least Median of Squares Estimator"* (LMS) and the *"Least Trimmed Squares Estimator"* (LTS). (See [20]).

The following minimization problem defines the LMS :

$$\hat{\theta} = \min_{\theta} \operatorname{med}_{i} (y_i - x_i^T \theta)^2 \tag{139}$$

One can extend the LMS to the system identification of the n-th order discrete time system

$$\begin{aligned} y_{k+1} &= [\, -y_k \ \ldots \ -y_{k-n} \ \ldots \ u_k \ \ldots u_{k-n}]\theta \\ y_{k+1} &= \phi_k^T \theta \end{aligned} \tag{140}$$

Inserting this expression in the definition of the LMS, and the following expression defines the LMS system identification algorithm.

$$\hat{\theta} = \min_{\theta} \operatorname{med}_{i} (y_{i+1} - \phi_i^T \theta)^2 \tag{141}$$

Similarly, the LTS estimator is defined as the minimum of the trimmed sum of the squared residuals where the residuals are given by

$$r = y_{i+1} - \phi_i^T \theta \tag{142}$$

The residuals are ordered by increasing value and a certain percentage of the data is trimmed and the sum of the remaining residuals is minimized.

5.0 Simulation Results

5.1 using the non robust PLID, Huber, Hampel and Reject robust PLID

Simulations have been performed for the non robust stochastic PLID algorithm. The plant simulated and used for identification is given by the transfer function

$$H(z) = \frac{-1.8 \times z^2 + 1.3 \times z - 0.4}{z^3 - 1.8 \times z^2 + 1.3 \times z - 0.4} \tag{143}$$

which is a stable system. One set of simulations was done with measurement noise distributed according to the symmetric distribution

$$v = (1 - \varepsilon) \times N(0.0, 0.02) + \varepsilon \times N(0.0, 0.02 \times 10^6) \tag{144}$$

with values of ε in the set {0.0, 0.01, 0.05, 0.10, 0.15, 0.40}. In the simulations ε is replaced by a random variable t that takes on the discrete values 0 and 1 with probability ε. This is done by sampling a random number generator and comparing the sampled number to the value of ε; if the random number is greater than ε then t is set to 0, and if the random number is smaller than ε then t is set to 1, which simulates the outlier. Another set was done with a nonsymmetric distributed measurement noise of the kind

$$v = (1 - \varepsilon) \times N(0.0,0.02) + \varepsilon \times 1000.0 \tag{145}$$

The values for ε were the same as in the first set.

In both cases the input to the plant was taken to be a Gaussian sequence with zero mean and variance 2.0. The DC gain H(1) of the plant is 10, signal to noise ratio is therefore approximately

$$S/N = 10 \times \log\left(\frac{10^2 \times 2.0}{0.02} \right) = 40\ dB \tag{146}$$

In addition to the measurement noise, the plant is also corrupted by a noise directly affecting the states. This noise is Gaussian, N(0.0,0.02). The initial states are stochastic variables with distribution N(0.0,0.10).

5.1.1 Generel Results for the non-robust PLID

Plots have been prepared showing the convergence of parameter estimates, the log of the state and parameter norms, the log of the trace of state and parameter covariance, the magnitude of the state and parameter gains in the PLID algorithm. The definitions are as follows :

$$
\begin{aligned}
\textit{state error norm} &= \frac{\|x - \hat{x}\|}{\|x\|} \\
\textit{parameter error norm} &= \frac{\|\theta - \hat{\theta}\|}{\|\theta\|} \\
P &= \begin{bmatrix} P^{xx} P^{x\theta} \\ P^{\theta x} P^{\theta\theta} \end{bmatrix} \\
\textit{trace of state cov.} &= \textit{trace}\,(P^{xx}) \\
\textit{trace of parameter cov.} &= \textit{trace}\,(P^{\theta\theta}) \\
K &= \begin{bmatrix} K^{x} \\ K^{\theta} \end{bmatrix} \\
\textit{magnitude of state gain} &= \|K^{x}\| \\
\textit{magnitude of parameter gain} &= \|K^{\theta}\|
\end{aligned}
\tag{147}
$$

The plots shown in Figure 3 on page 81, Figure 4 on page 82, Figure 5 on page 83, Figure 6 on page 84, Figure 7 on page 85 are for the case of no outliers in the measurements and using the standard nonrobust PLID algorithm. The plots in Figure 8 on page 86, Figure 9 on page 87, Figure 10 on page 88, Figure 11 on page 89, Figure 12 on page 90 are for the symmetric noise case and the standard nonrobust PLID Algorithm. The plots in Figure 13 on page 91,

Figure 14 on page 92, Figure 15 on page 93, Figure 16 on page 94, Figure 17 on page 95 are for the nonsymmetric noise case and using the standard nonrobust PLID algorithm.

5.1.1.1 LOG of Norm of State and Parameter Estimates

The plots in Figure 3 on page 81 and Figure 4 on page 82 show that the state estimates are biased, even in the case where there are no outliers ($\varepsilon = 0$). The plots in Figure 8 on page 86, Figure 9 on page 87, Figure 13 on page 91, Figure 14 on page 92

show that the bias increases when outliers are added to the measurement noise. The bias does not depend heavily on the outlier percentage. 1% is enough to introduce a large bias. Initially the parameter values settle to approximately correct values (except for a small bias) and the log of the state norms approaches -1, which means that the estimates are decent. (See figures 5,10,15). At time step 100 an outlier occurs in the measurements, and the estimates immediately get a large bias, and have trouble converging to the true values again. This happens every time an outlier occurs. The log of the state norms is approximately positive 1, which means that the estimates are off by a factor of 10. In the cases of 5%, 10%, 15% and 40% outliers in the data the same phenomenon is observed. The log of the state norm approaches positive 1. The bias therefore does not depend on the

amount of outliers present. Just a small fraction of outliers is enough to cause havoc in the estimates. It must therefore be concluded that the breakdown point for the nonrobust PLID is 0%.

Rather than depending on the amount of outliers present in the measurements, the bias would depend on the magnitude of the outlier. (A large outlier giving large bias and vice versa). That this is the case will be shown later. That the bias depends on the outlier size is a direct consequence of the least squares linear unbounded influence function. That 0% outlier gives a small bias is somewhat surprising, since the Stochastic PLID is promised to be biasfree in the case of Gaussian noise sequences. A possible explanation could be that the PLID algorithm converges slowly. In the work by Hopkins it is not stated how fast the PLID algorithm would converge.

5.1.1.2 LOG of Trace of State and Parameter Covariance

The plots of the traces in figures Figure 6 on page 84, Figure 11 on page 89, Figure 16 on page 94 show something surprising, the trace for $\varepsilon = 0$ (no outliers) is larger than the trace in the cases where $\varepsilon > 0$. One would might the trace to be smaller for the case of no outliers because the measurement signal contains more useful information. A possible explanation is that the PLID algorithm is 'fooled' to believe that measurements with outliers contain more useful information coming from a more persistent input signal. The plots show that the log of

the trace of the parameter covariance is about a factor of 10 smaller in the cases of outliers than in the case of no outliers. Again the parameter trace does not depend on the outlier percentage, if just outliers are present. This also indicates that the breakdown point for the nonrobust PLID is 0%. It is also noted that the logarithmic trace tends to go to $-\infty$ which means that the covariance goes to 0 along with the uncertainty in the state and parameter estimates. This might be correct but the algorithm does not know that the estimates are biased.

5.1.1.3 Magnitude of State and Parameter Gains

The plots in Figure 7 on page 85, Figure 12 on page 90, Figure 17 on page 95, show that the gains for $\varepsilon = 0$ are larger than for $\varepsilon > 0$, which makes sense, because one wants to extract as much information as possible from the measurement signal when the covariance of the estimate is large and the algorithm 'thinks' it has to extract as much information as possible from the measurement signal. The results for the nonsymmetric noise case are shown in Figure 13 on page 91, Figure 14 on page 92, Figure 15 on page 93, Figure 16 on page 94Figure 17 on page 95. The conclusions drawn from these plots are the same as for the symmetric case. If one compares the norms of the parameters for the plots from the two noise cases, it is seen that the bias is larger in the nonsymmetric case. In the nonsymmetric case all the outlier probability mass is placed in a single point, while in the symmetric case the outlier probability mass is distributed over a range covering both negative and positive numbers. The conclusion

to be drawn from these simulations is clearly that some kind of robustness scheme is needed to bound the bias from large outliers. The plots show that a contamination as small as 1% results in biased parameter and state estimates. The bias depends on the size and distribution of the outliers.

5.1.2 General Results using the Huber M-estimator

This section gives a summary of the simulation results for the PLID algorithm robustified with the Huber M-estimator as ψ function. The value of the treshold is chosen as 1.40.

The simulations were performed under the same conditions as with the non Robust PLID. Although only the non symmetric noise case is considered because this is a worst case situation. The plots for the 1% contamination case are shown in Figure 18 on page 96, Figure 19 on page 97, Figure 20 on page 98, Figure 21 on page 99, Figure 22 on page 100. The cases for 0%, 5%, 10%, 15% contamination are not shown because it would take too many pages. Plots of the parameter values in Figure 18 on page 96 and Figure 19 on page 97 clearly show that the Huber type M-estimator improves the robustness of the PLID algorithm. The parameter values converge to good estimates for outlier percentages of 0, 1, 5, 10 and 15 %. it is noticed that the parameter convergence is slower for the contaminated cases than for the case of no outliers. For the case of 0% outliers . it is hard to tell any difference in performance between the Huber robustified

PLID and the non robust PLID. For the cases of 0, 1, 5, 10, 15% contamination, the log of the parameter norms is close to -1, which is an improvement of a factor 100 over the nonrobust PLID. The norms of the states are more scattered than the parameter norms; that is so because the plant is simulated with noise disturbing the states directly. The log of the trace of the parameter covariance is very close to the case of the nonrobust PLID with 0% outliers. This indicates that the covariance has been robustified. One thing that is not shown on any of the plots is the fact that the Huber robustified PLID algorithm sometimes would diverge when outliers were present. The divergence could be caused by many outliers occurring in the transient phase of the identification. In this phase the algorithm is very sensitive and tries to extract as much information from the measurements as possible. And the outliers still have an influence on the estimates, since the influence function for the Huber M-estimator never goes to zero.

5.1.3 General Results using the Hampel Optimal M-estimator

The Hampel Optimal M-estimator is a redescending M-estimator. Simulations with this ψ function show that the parameter estimates converge for the cases of 0, 1, 5, 10 and 15 % outliers. The case of 1% contamination is shown in Figure 23 on page 101, Figure 24 on page 102, Figure 25 on page 103, Figure 26 on page 104, Figure 27 on page 105. Again it would require too many pages to include the other cases. Higher percentages were also tried and the pa-

rameter estimates also converged. So again the robustness of the PLID is improved when using a function of the Hampel optimal type. Also in this case it is difficult to tell if there is a difference in performance between the nonrobust PLID at 0% outliers and the Hampel robustified PLID at 0% outliers. The log of the parameter norm is again approximately -1 for the cases of 0, 1, 5, 10, 15% contamination. Again this is an improvement of a factor 100 over the nonrobust PLID. Also the log of the trace of the parameter covariance is almost the same as for the nonrobust PLID 0% contamination. This indicates that the covariance has been robustified. None of the simulations so far with the Hampel function showed any sign of diverging estimates as did the Huber function. Instead there was a tendency for the algorithm to move to a local minima. This tendency increases with increasing contamination.

5.1.4 General Results using the Hard Rejection M-estimator

The plots for the hard rejection robustified PLID are shown in Figure 28 on page 106, Figure 29 on page 107, Figure 30 on page 108, Figure 31 on page 109, Figure 32 on page 110 and show the same phenomena as the Hampel robustified PLID.

5.2 *Local Minima and the M-estimators*

There are problems with local minima in the use of the redescending M-estimators. Recall that the M-estimators come from the solution of the equation

$$\sum_{i=1}^{m} \psi(z_i - \theta) = 0 \tag{148}$$

In the robustified PLID this equation is 'solved' *ON*-line and needs a starting point. In the *OFF* line use of M-estimators and the unidimensional case, the starting point is usually taken as the median to give a robust initial value for the iterative solution by a Newton-Raphson method. In the multidimensional case the starting point is found by the *"least absolute value"* (LAV) estimator. But it is time consuming to use the median as a starting point in the *ON* line case. The starting value is therefore chosen arbitrarily as 0, which is not necessarily a robust first guess ! In the transient phase of the identification there will therefore be two sources for the outliers, namely wrong predictions from bad parameter estimates and outliers coming from bad measurements. For the redescending M-estimator this can have the effect that the estimator moves into a local minimum, because ψ will be zero for either far away predictions or far away measurements. And when ψ is zero, the states and parameters in the PLID will not get updated, and the identification will slowly stop because the gains decrease. The Huber M-estimator can also stop updating the parameter estimates. When the outlier frac-

tion is high in the initial phase because of both bad predictions and bad measurements, the estimator will converge to the majority of the data, which then can be outliers. And because the gains decrease, the estimator will come to a stop.

5.3 *Results for Different Initial Conditions*

As mentioned earlier, there is a problem with choosing good initial conditions because the robust algorithms are nonlinear. Therefore simulations have been performed to investigate the influence of the initial conditions of the extended state vector S(0) and the initial covariance matrix P(0). The simulations have been performed on the same plant as before

$$H(z) = \frac{-1.8 \times z^2 + 1.3 \times z - 0.4}{z^3 - 1.8 \times z^2 + 1.3 \times z - 0.4} \tag{149}$$

The simulations were performed with the same input signal, a Gaussian white noise sequence with variance 2.0 to give a rich input, and the same measurement noise signal with variance 0.02 and a nonsymmetric outlier distribution. The outlier size is 1000.0, and the state noise has been removed. The measure of estimator quality of interest is the log of the normalized parameter error norm :

$$\log\left(\frac{\|\theta - \hat{\theta}\|}{\|\theta\|} \right) \tag{150}$$

eqno (151) Three different kinds of initial conditions have been tested :

- $s(0) = 10$, $P(0) = 10^3$. The initial guess is far away from the actual parameters and the initial sensitivity is high.

- $s(0)$ is off by 0.1 from the actual parameters, and $P(0) = 10^3$ gives a high initial sensitivity.

- $s(0)$ is off by 0.1 from the actual parameters, and $P(0) = 10$ gives a low initial sensitivity.

In the first case with an initial estimate far away from the true parameter values and a high initial sensitivity to incoming measurements, simulations with the Huber, Hampel and Reject ψ functions are shown in Figure 33 on page 111, Figure 34 on page 112 and Figure 35 on page 113 As expected, these plots show that the robust algorithms have trouble converging. Especially the Huber robustified PLID is very slow, which can be explained by the fact that every measurement has some finite influence on the updating, outliers will therefore try to pull the estimates away from the actual parameters, whereas in the case of redescending ψ functions the influence of the outlier is reduced to 0, which means that the updating stops, instead of pulling away. In Figure 36 on page 114, Figure 37 on page 115 and Figure 38 on page 116

the initial sensitivity has been lowered by setting $P(0) = 10$ and $S(0) = 0$, but still convergence is slow and not guaranteed. Although the Huber robustified PLID in Figure 36 on page 114

has improved over the one in Figure 33 on page 111.

The plots in Figure 39 on page 117, Figure 40 on page 118 and Figure 41 on page 119 reflect an initial guess close to the true parameters and a high initial sensitivity. Also in this case convergence is slow and not guaranteed. In simulations with the Hampel ψ function where the initial parameter estimate is close to the true parameters and the initial covariance matrix is small, the identification showed good convergence for contaminations as high as 40 %. See Figure 43 on page 121, Figure 44 on page 122, Figure 45 on page 123, Figure 46 on page 124.

This observation is consistent with the use of the *OFF*-line M-estimators where the initial guess has to be robust. By choosing S(0) close to the actual parameters and P(0) small, the requirement of a robust first guess is met.

5.4 Recovery from a Burst of Outliers.

The nonrobust PLID as well as the Huber and Hampel robustified PLID algorithms have been tested for their ability to recover from a burst of outliers in the middle of a good measurement sequence. These simulations were performed by letting no outliers occur the first 100 time steps, so that the algorithms would

settle down to good estimates before the outliers began to arrive. The burst of outliers lasts 100 time steps after which no outliers would occur. The results are seen in Figure 47 on page 125, Figure 48 on page 126 and Figure 49 on page 127.

The non robust PLID is clearly not able to recover from this burst, the error norm stays at a high level. The Huber and Hampel robustified PLID algorithms are able to recover from the bursts. The error norm of the Huber PLID rises when the burst occurs, but falls back again when the measurements are back to normal. The Hampel PLID keeps the error norm constant when the burst occurs, but starts improving the estimates as soon as the outliers stop occurring.

5.5 Outlier Magnitude and Bias.

The outlier magnitudes' influence on the bias of the non robust PLID, the Huber and Hampel robustified PLID algorithms has been investigated by simulating outliers of magnitude 1000 and 10. The outliers occur after the first 100 time steps. The plots in Figure 49 on page 127, Figure 50 on page 128, Figure 51 on page 129, Figure 52 on page 130, Figure 53 on page 131, Figure 54 on page 132 show the logarithm of the normalized parameter error. For the non robust PLID it is clear that the large outlier gives a large bias, and a small outlier gives a smaller bias. This is consistent with the fact that the non robust PLID has a linear unbounded influence function.

The Hampel and Huber robustified PLID algorithms show a less significant change in bias when the size of the outlier changes. This is so because the influence functions are limited, and an outlier will have a limited influence on the parameter estimate and its bias.

5.6 Breakdown Points.

As has been mentioned earlier, the breakdown point for the nonrobust PLID is 0%. Simulations with the Huber PLID suggest that the breakdown point is 50%, because with outlier percentages greater than 50%, the estimates would diverge. For the Hampel and Huber redescending influence functions the estimates would not diverge, but rather stay at a local minimum. See Figure 55 on page 133, Figure 56 on page 134, Figure 57 on page 135, Figure 58 on page 136.

6.0 Conclusion and Further Research

This study shows that Huber, Hampel or a hard rejection filtering function (ψ functions, M-estimators) can improve the robustness of the PLID algorithm substantially. The robustification is achieved by filtering the measurements through the mentioned nonlinear ψ functions, down weighting the covariance updating when outliers occur and the use of corrected predictions of the measurements instead of the raw measurements.

The robust algorithms are able to identify system parameters reliably in an environment with bad measurements, when some sort of good or robust initial estimate has been reached. In that case the robust algorithms can withstand a burst of only bad measurements and improve the estimates when the measurements no longer contain outliers.

Simulations have shown that the robust algorithms are sensitive to the initial values of the extended state vector and the covariance matrix. The best results are obtained when one has an initial estimate of the parameters that is close to

the true parameters and the covariance matrix has a low value. This observation is also consistent with what is known from the robust statistics literature, where the *OFF*-line M-estimators require a robust initial guess to converge to the true value.

It is therefore suggested that the starting point for the robust PLID algorithms is calculated by robust *OFF*-line methods such as the Least Trimmed Squares method or the Least Median of Squares method. Another approach is to apply the available information one has from the physical laws that govern the system and use the theoretical parameter values as initial values in the identification.

But even if one has no access to robust initial values, any attempt of robustification is better than nothing.

This thesis does not give a theoretical proof of robustness for the robustified PLID algorithm. This is a point where further research has to be done. There is also a need to find a robustification scheme that is less sensitive to the initial conditions and to find the theoretical breakdown point for the different robustifications. Intuition says that the breakdown point for the M-robustified (Huber, Hampel, Reject) algorithms will be 50%.

Bibliography

6.1 Text Books and Dissertations

[1] Ackermann, Jürgen, "Sampled-Data Control Systems. Analysis and Synthesis, Robust System Design.", Springer Verlag, Berlin, Heidelberg 1985.

[2] Chen, Chi-Tsong, "Linear System Theory and Design.", Holt, Rinehart and Winston, 1984.

[3] Hampel, Frank R., Ronchetti, Elvezio M., Rousseeuw, Peter J. and Stahel, Werner A. "Robust Statistics. The Approach based on Influence functions.", John Wiley & Sons, 1986.

[4] Hampel, Frank R., 1968, "Cotributions to the theory of robust estimation.", Ph.D. Thesis, University of California, Berkeley.

[5] Huber, Peter J., "Robust Statistics", John Wiley & Sons, 1981.

[6] Hopkins, Mark A. "Pseudo-Linear Identification: Optimal Joint Parameter and State Estimation of Linear Stochastic MIMO Systems.", Ph.D dissertation, Virginia Polytechnic Institute and State University, March 1988.

[7] Kemp, Russell Stephen, "Pseudo-Linear Identification And Its Application to Adaptive Control.", M.S Thesis, Viriginia Polytechnic Institute and State University, 1987.

[8] Papoulis, Athanasios, "Probability, Random Variables and Stochastic Processes.", Mcgraww-Hill Book Company, 1984.

[9] Sinha, N.K. and Kuszta B., "Modelling and Identification of Dynamic Systems." Van Nostrand Reinhold Company, 1983.

[10] Stengel, Robert F., 1986, "Stochastic Optimal Control. Theory and Application." John Wiley & Sons.

6.2 Articles

[11] Ershov, A. A., 1978, "Robust Filtering Algorithms", Automatika, p.992.

[12] Ershov, A. A., Liptser, R. Sh., "Robust Kalman Filter in Discrete Time.", Automatika, p. 359, 1978.

[13] Groutage, F. D., Jacquot, R. G. and Smith, D. E., "Adaptive State Variable Estimation Using Robust Smoothing.", Journal of Dynamic Systems, Measurements and Control., Vol. 106, p. 335, December 1984.

[14] Hampel, Frank R., 1971, "A general qualitative definition of robustness." Ann. Math. Statist. 42, 1887-1896.

[15] Hampel, Frank R., 1974, "The Influence Curve and its role in robust estimation.", J. AM. Statist. Assoc. 69, 383-393.

[16] Martin, Douglas R. and Masreliez, C. J., "Robust Estmation via Stochastic Approximation.", IEEE trans. inf. theory, Vol. IT-21, No. 3, May 1975.

[17] Martin, Douglas R. and Thomson, David J., "Robust Resistant Spectrum Estimation.", Proceedings of the IEEE, Vol. 70, No. 9, September 1982.

[18] Masreliez, C. Johan and Martin, R. Douglas, "Robust Bayesian Estimation for the linear Model and Robustifying the Kalman Filter.", IEEE Transactions on Automatic Control, Vol. AC-22, No. 3, p. 361, June 1977.

[19] Masreliez, C. J., "Approximate Non-Gaussian Filtering with Linear State and Obsevation Relations.", IEEE Trans. Autom. Contr., Vol. AC-20, Feb. 1975.

[20] Mili, L. and Rousseeuw P.J., "High Brekdown Point Estimation in Power Systems : Combinatorial Search Methods.", Submitted for presentation to IEEE summer meeting, Long Beach, CA, 1989.

[21] Poljak, B. T. and Tsypkin Ja. Z., "Robust Identification.", Automatica, Vol. 16 p. 53, 1980.

[22] Stankovic, Srdjan S., and Kovacevic, Branko D., "Analysis of Robust Stochastic Approximation Algorithms for Process Identification.", Automatica, Vol.22, No. 4, pp. 483-488, 1986.

[23] Tanaka, Masahiro and Katayama, Tohru, "Robust Kalman filter for linear discrete-time system with gaussian sum noises.", Int. J. Systems Sci., 1987, vol. 18, No. 9, pp. 1721-1731.

[24] Tukey, John W. "A Survey of Sampling from Contaminated Distributions", in (I. Olkin a.o. Ed.) "Contributions to Problems and Statistics.", Stanford University Press, Stanford, 1960..

Appendix A. The Kalman Filter and The Weighted Least Squares Estimator

One derives the weighted least squares (WLS) state estimator by minimizing the normalized cost function :

$$J = E\{(z - H \times \hat{x})^T R^{-1} (z - H \times \hat{x})\}$$

Where R is the covariance of the measurement error v_k The measurements are given by

$$z_k = H \times x_k + v_k$$

The OFF-line solution to the minimization problem is

$$\hat{x}_k = (H^T R^{-1} H)^{-1} H^T R^{-1} z_k$$

This equation can also be formulated recursively.

In th Kalman Filter one takes the the optimal state estimate as the weighted least squares state estimate, given the measurements and predictions up to time k.

One can formulate the Kalman filter as an OFF-line WLS estimator as follows :

Augment the measurements z and the optimal state predictions $\hat{x}_{k|k-1}$ in the auxiliary measurement vector $\tilde{z}$

$$\tilde{z}_k = \begin{bmatrix} z_k \\ \hat{x}_{k|k-1} \end{bmatrix} = \begin{bmatrix} H \\ I \end{bmatrix} \times x_k + \begin{bmatrix} v_k \\ -\delta_x \end{bmatrix}, \quad \delta_x = x - \hat{x}$$

The WLS estimator for this measurement equation depends on the measurements and predictions up to time k.

$$\hat{x}_{k|k} = (\tilde{H}^T \tilde{R}^{-1} \tilde{H})^{-1} \tilde{H}^T \tilde{R}^{-1} \tilde{z}_k$$

The recursive form of the last equation will yield the Kalman Filter. In terms of robustness, the relation between the WLS and the Kalman Filter means that the Kalman Filter is nonrobust, since the WLS is nonrobust.

Appendix B. Simulation Plots

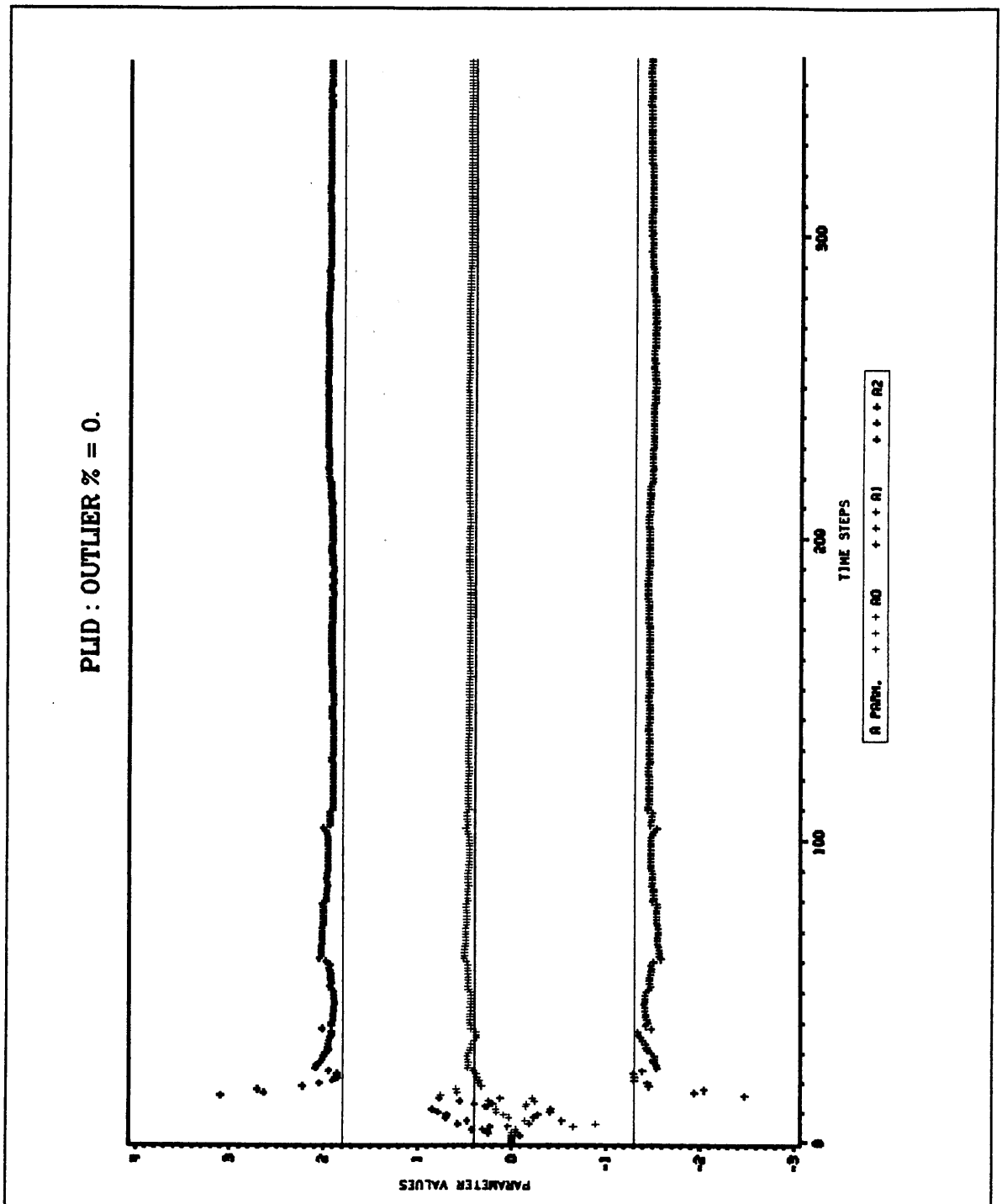

Figure 3. Non robust PLID.: Convergence of plant parameters. No outliers.

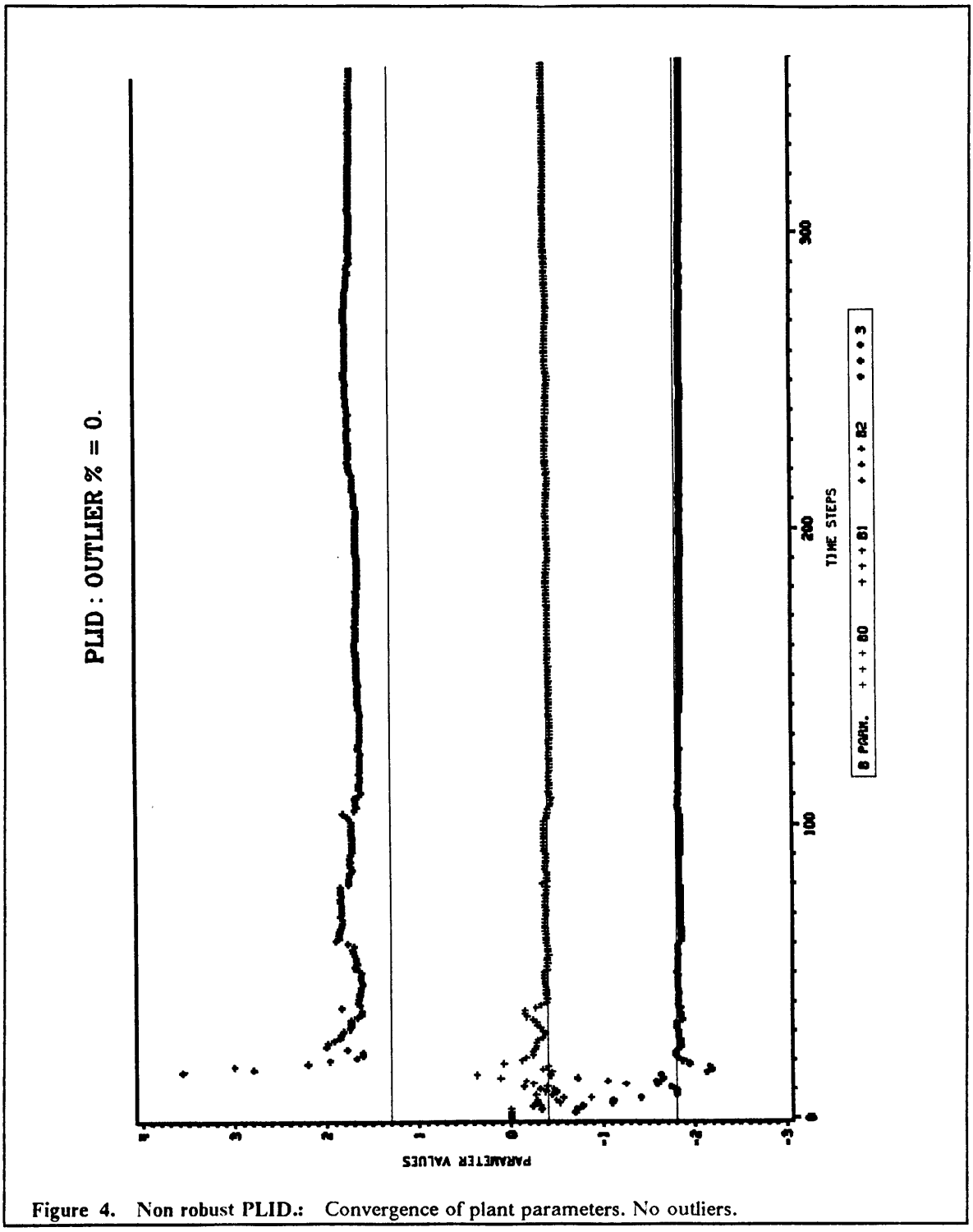

Figure 4. Non robust PLID.: Convergence of plant parameters. No outliers.

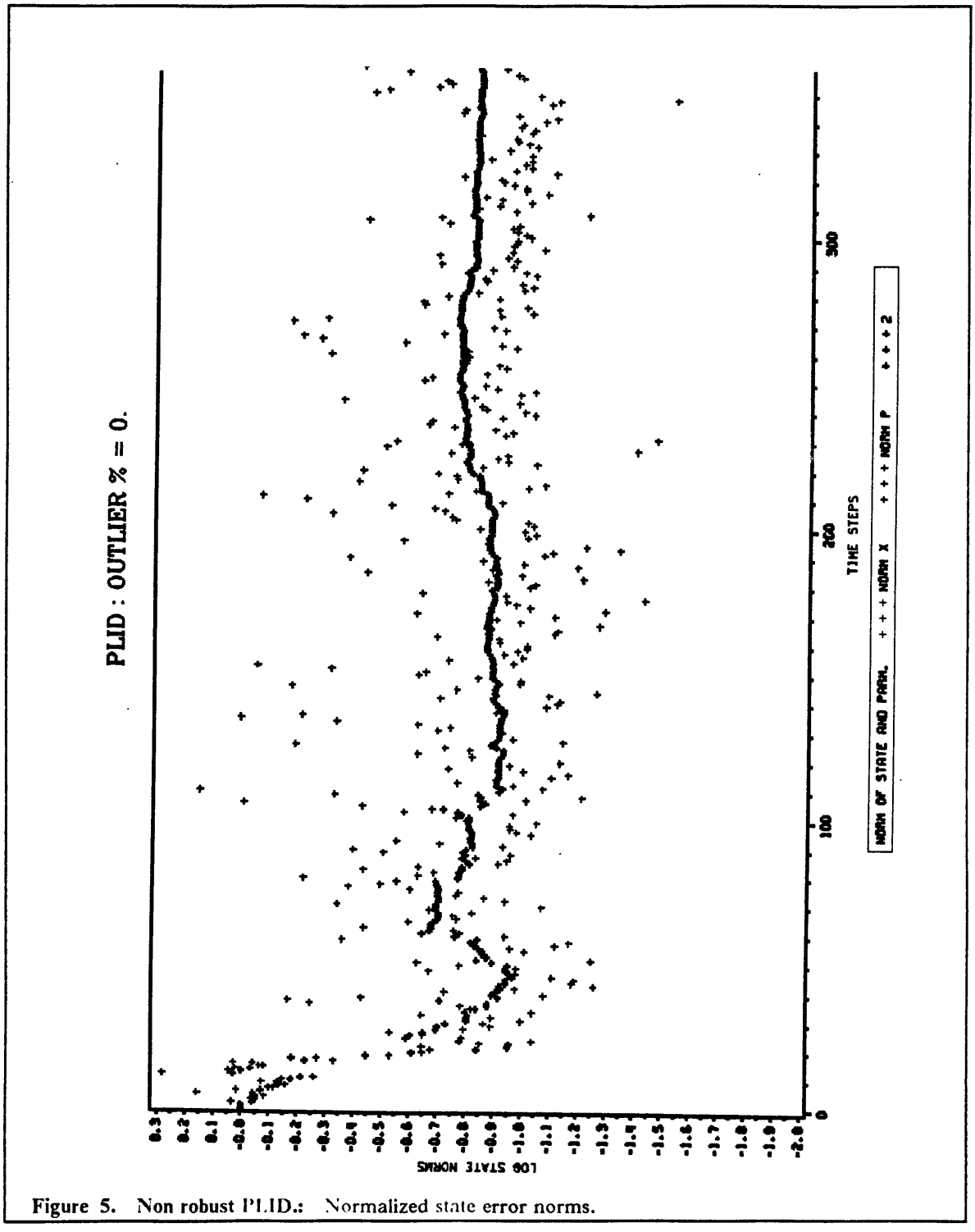

Figure 5. Non robust PLID.: Normalized state error norms.

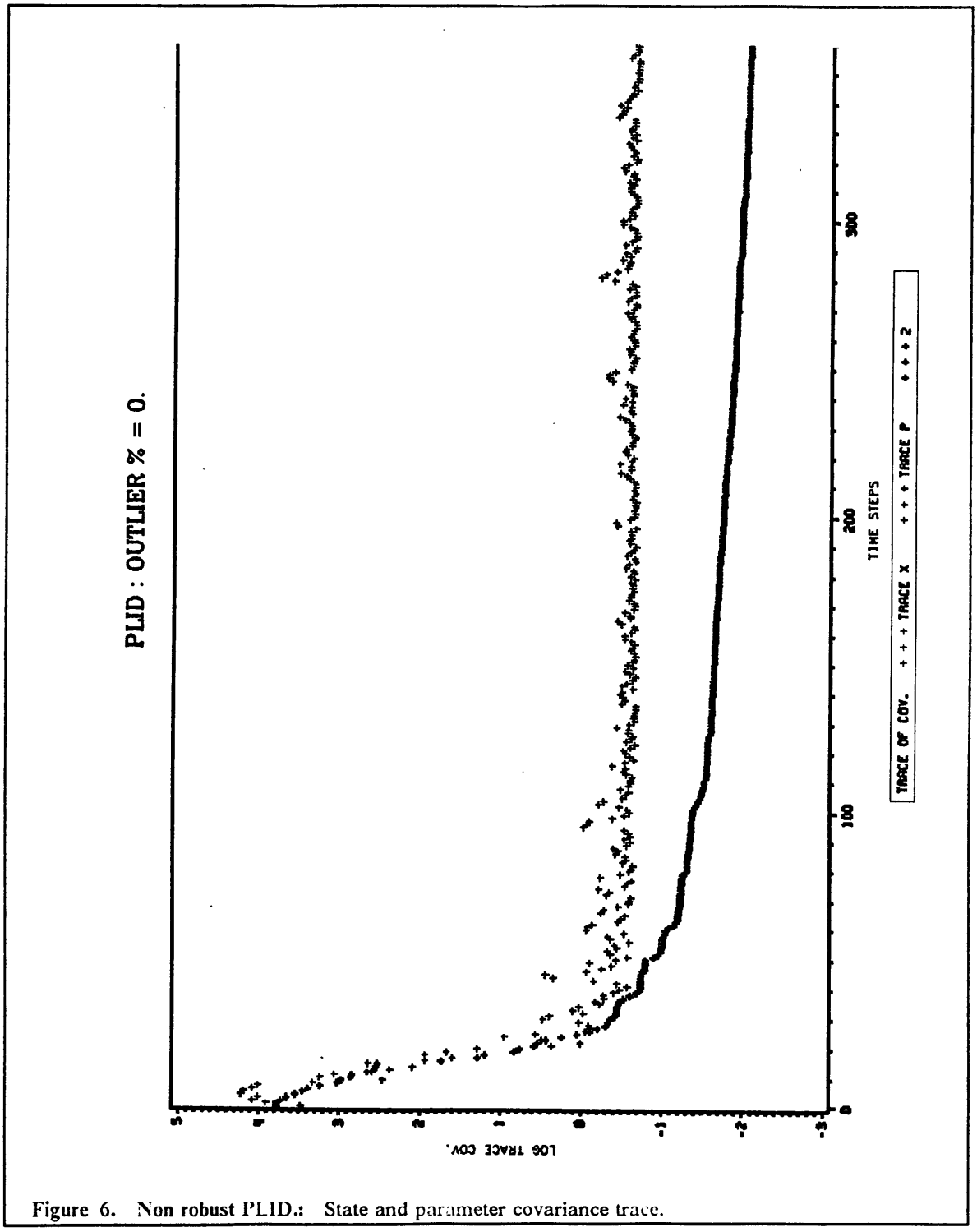

Figure 6. Non robust PLID.: State and parameter covariance trace.

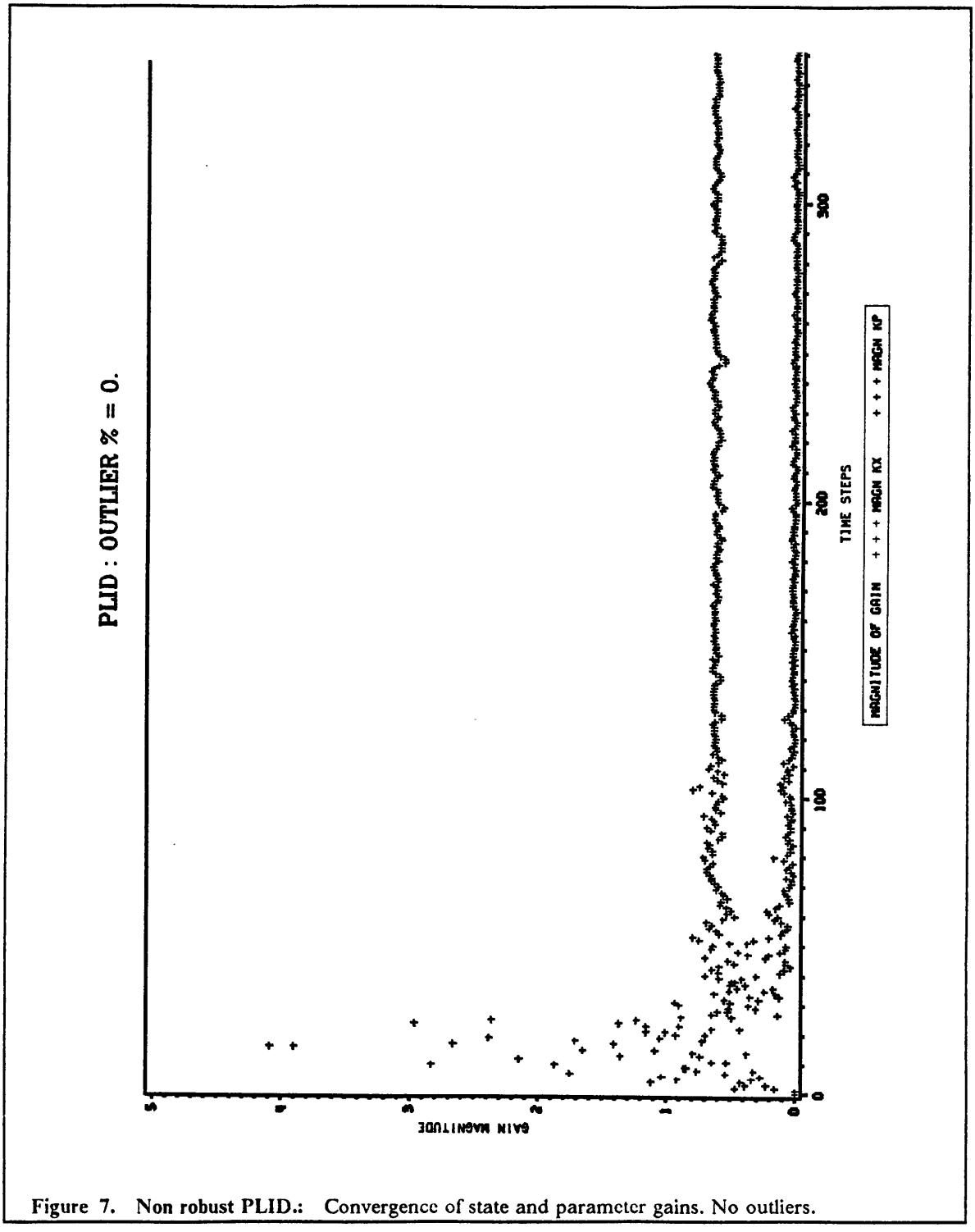

Figure 7. **Non robust PLID.:** Convergence of state and parameter gains. No outliers.

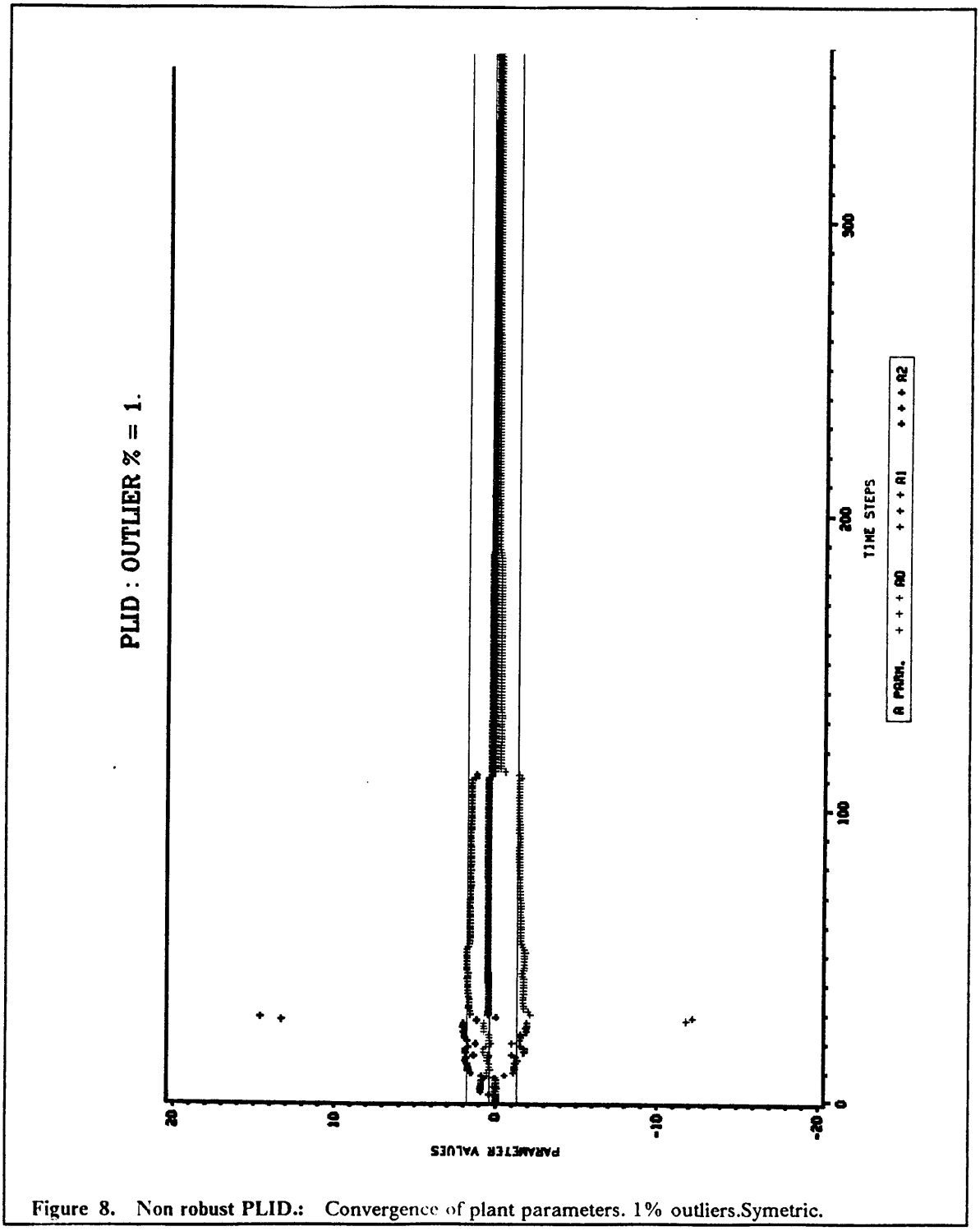

Figure 8. Non robust PLID.: Convergence of plant parameters. 1% outliers.Symetric.

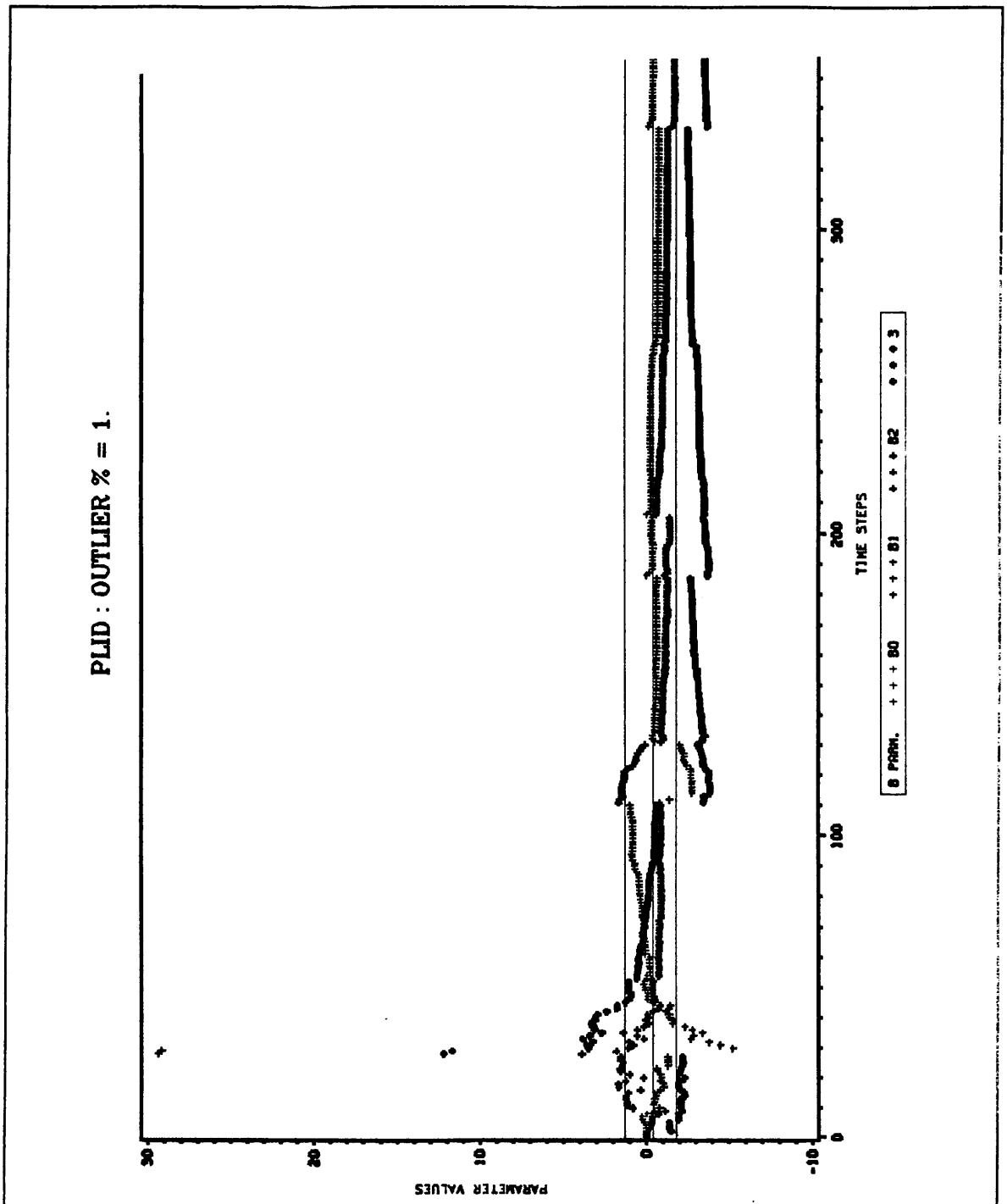

Figure 9. Non robust PLID.: Convergence of plant parameters. 1% outliers.Symetric.

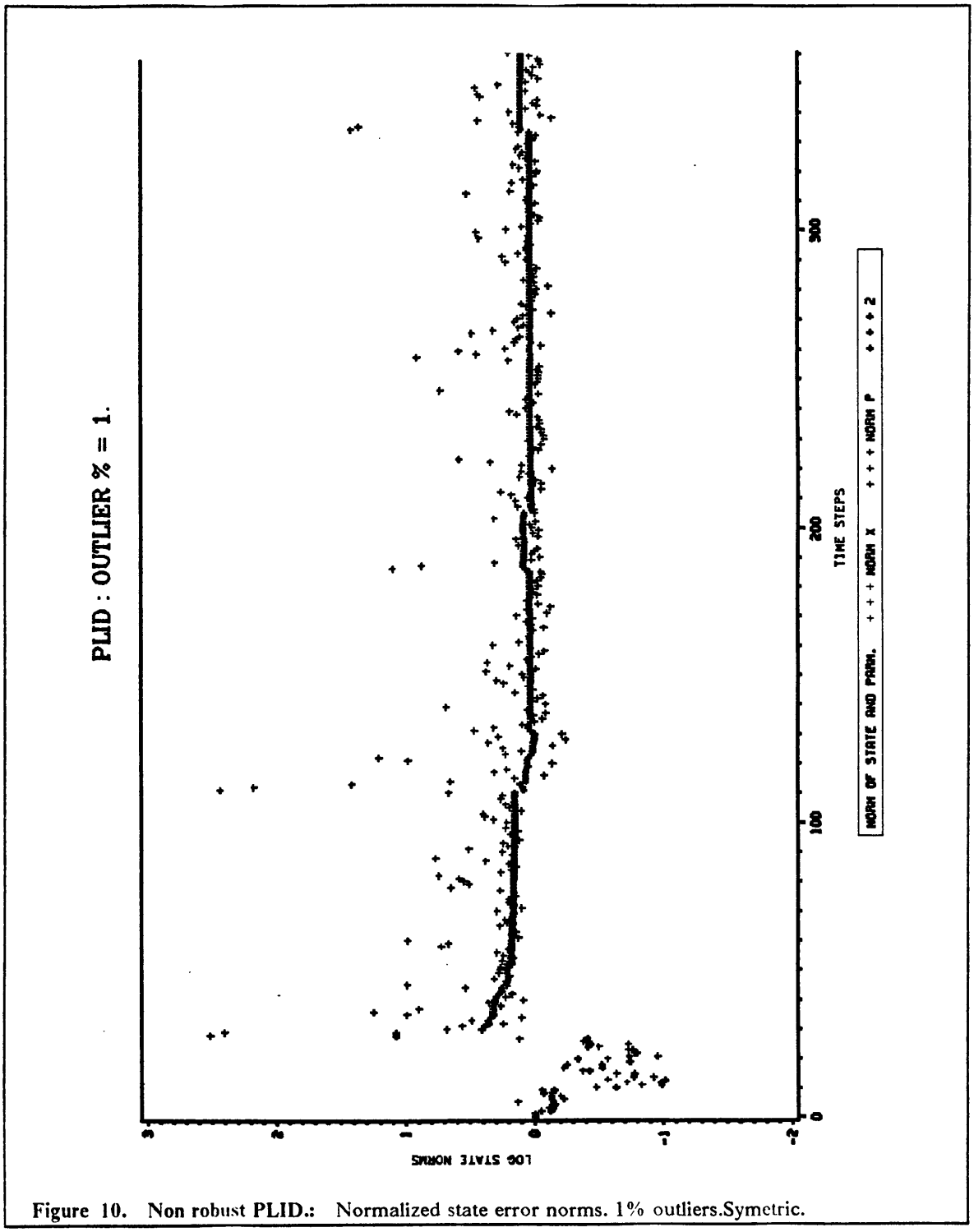

Figure 10. Non robust PLID.: Normalized state error norms. 1% outliers.Symetric.

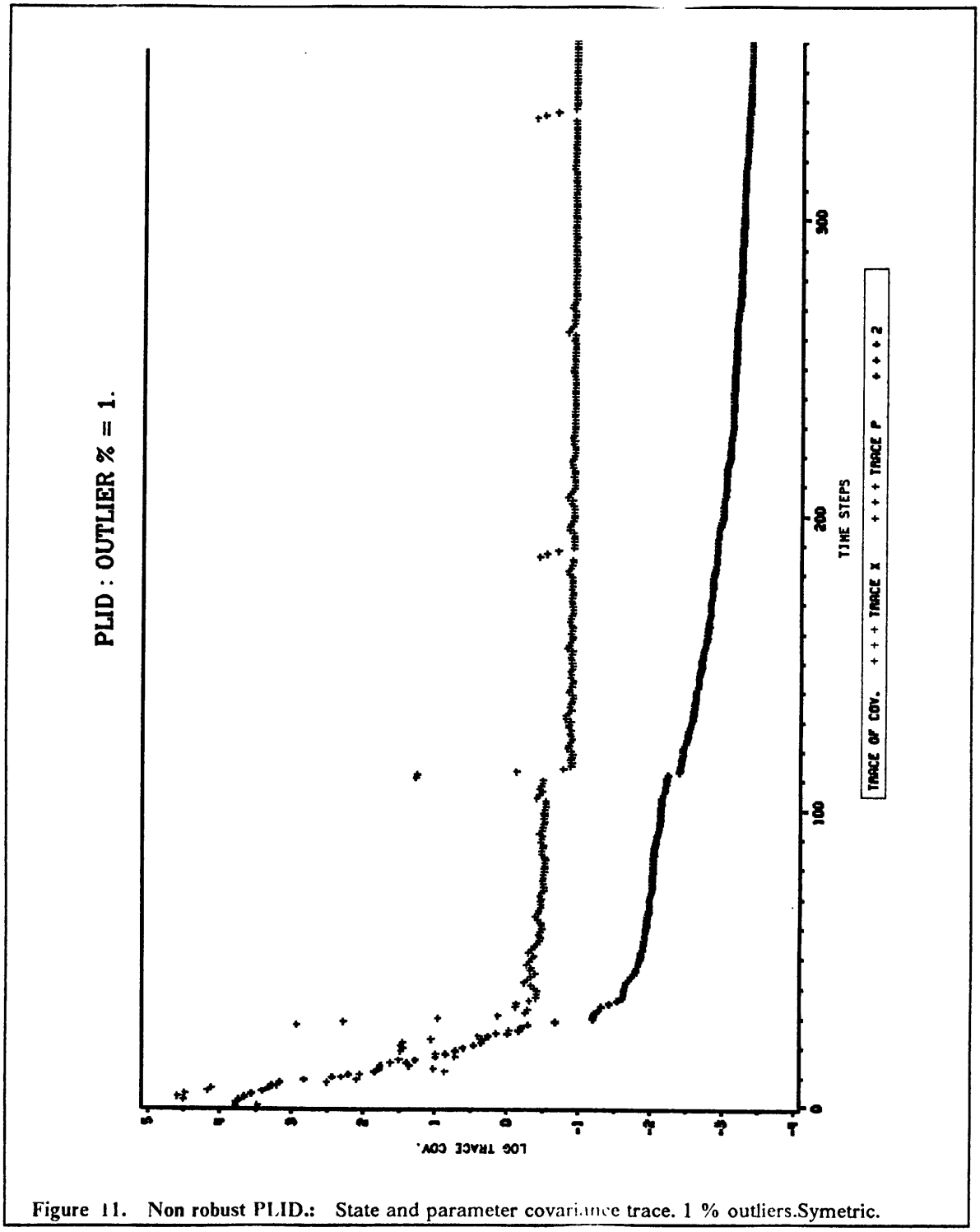

Figure 11. Non robust PLID.: State and parameter covariance trace. 1 % outliers.Symetric.

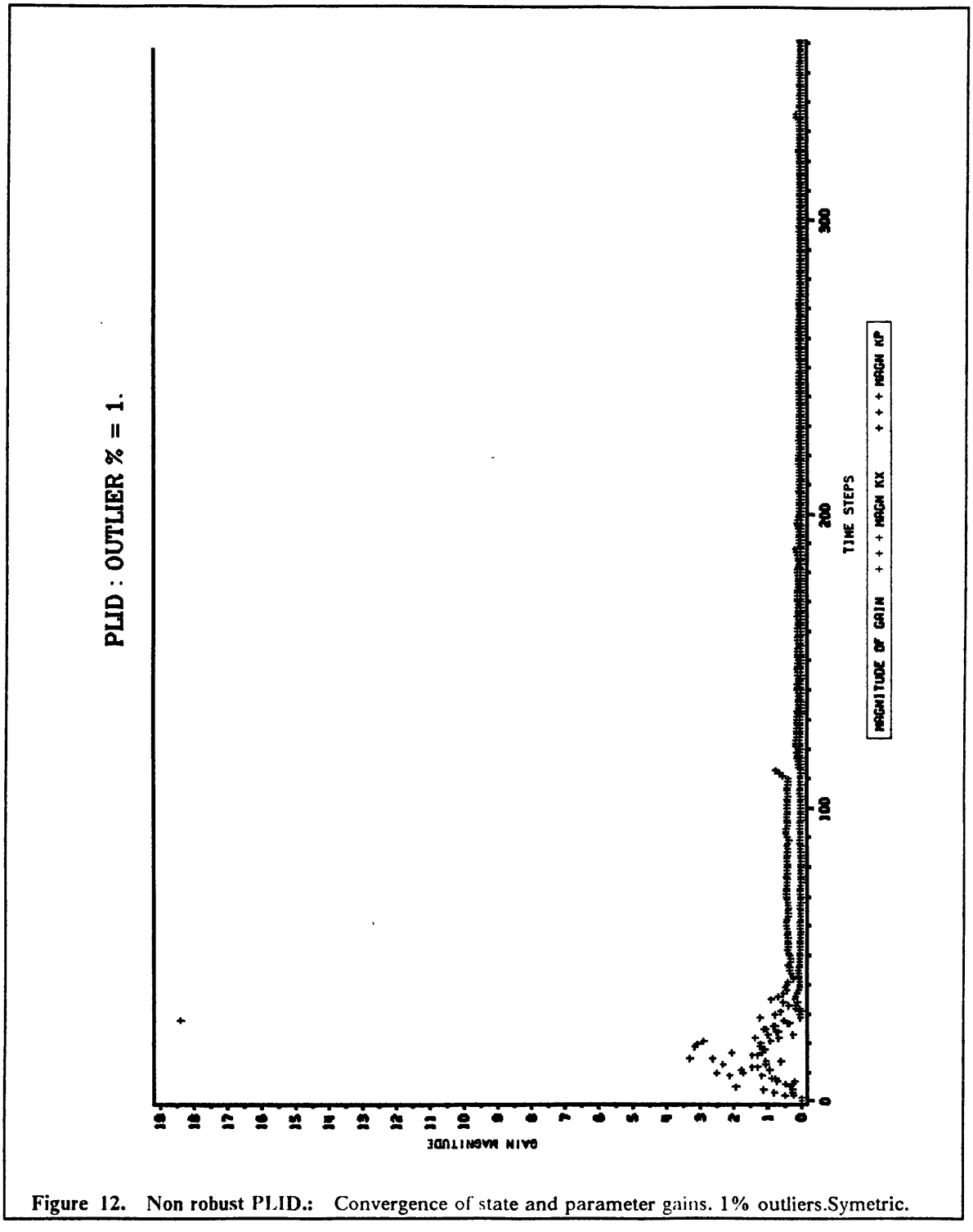

Figure 12. **Non robust PLID.:** Convergence of state and parameter gains. 1% outliers.Symetric.

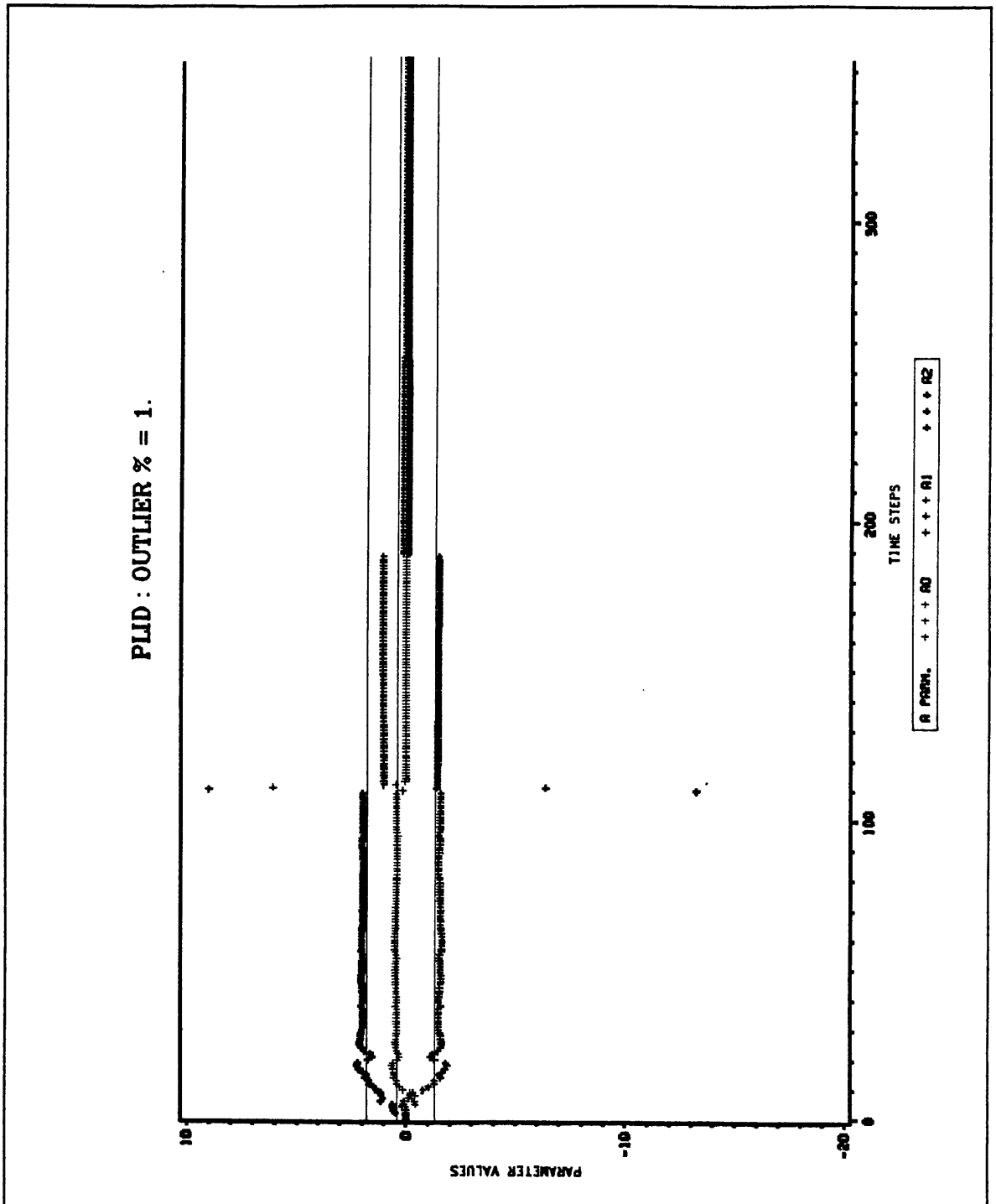

Figure 13. Non robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

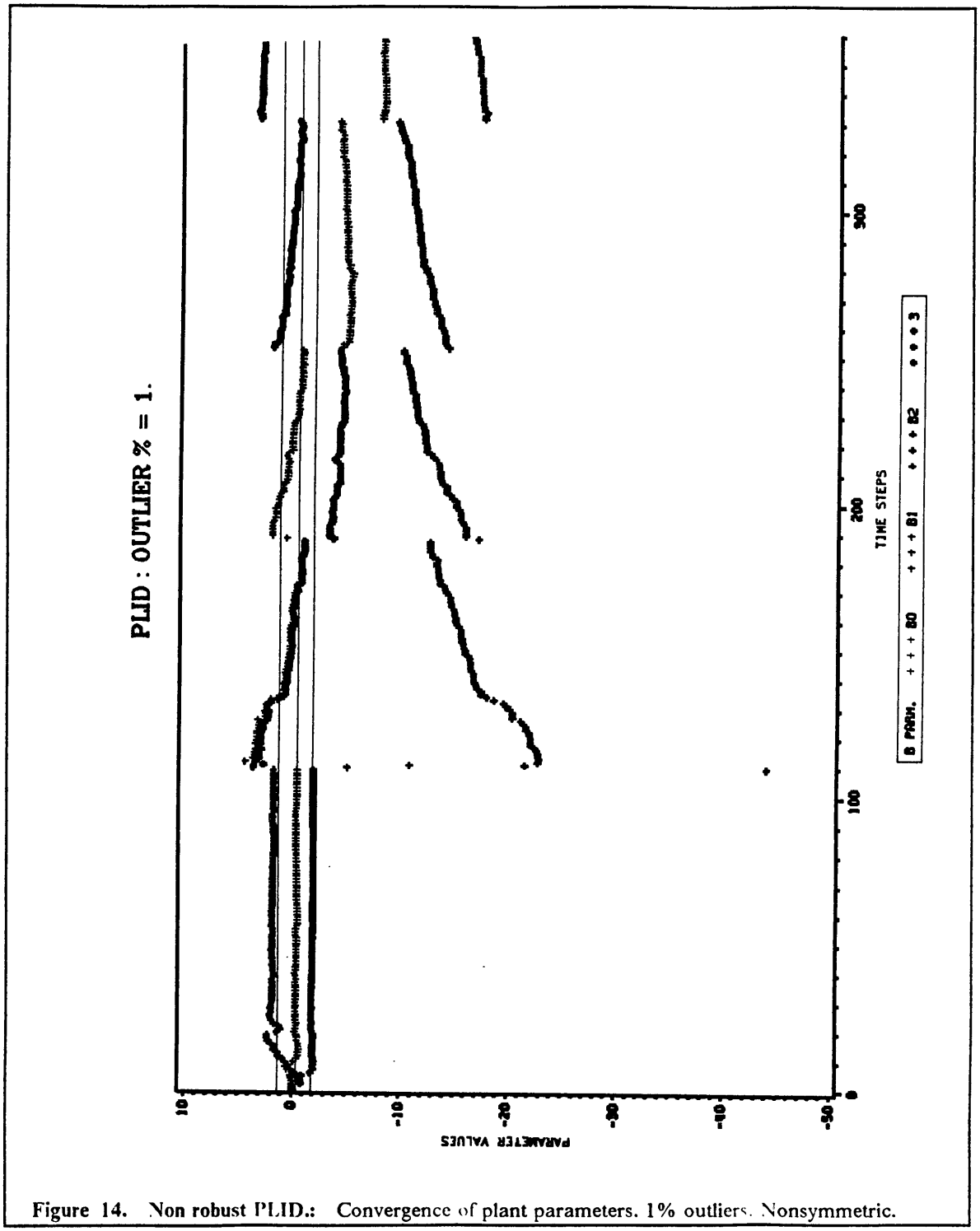

Figure 14. Non robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

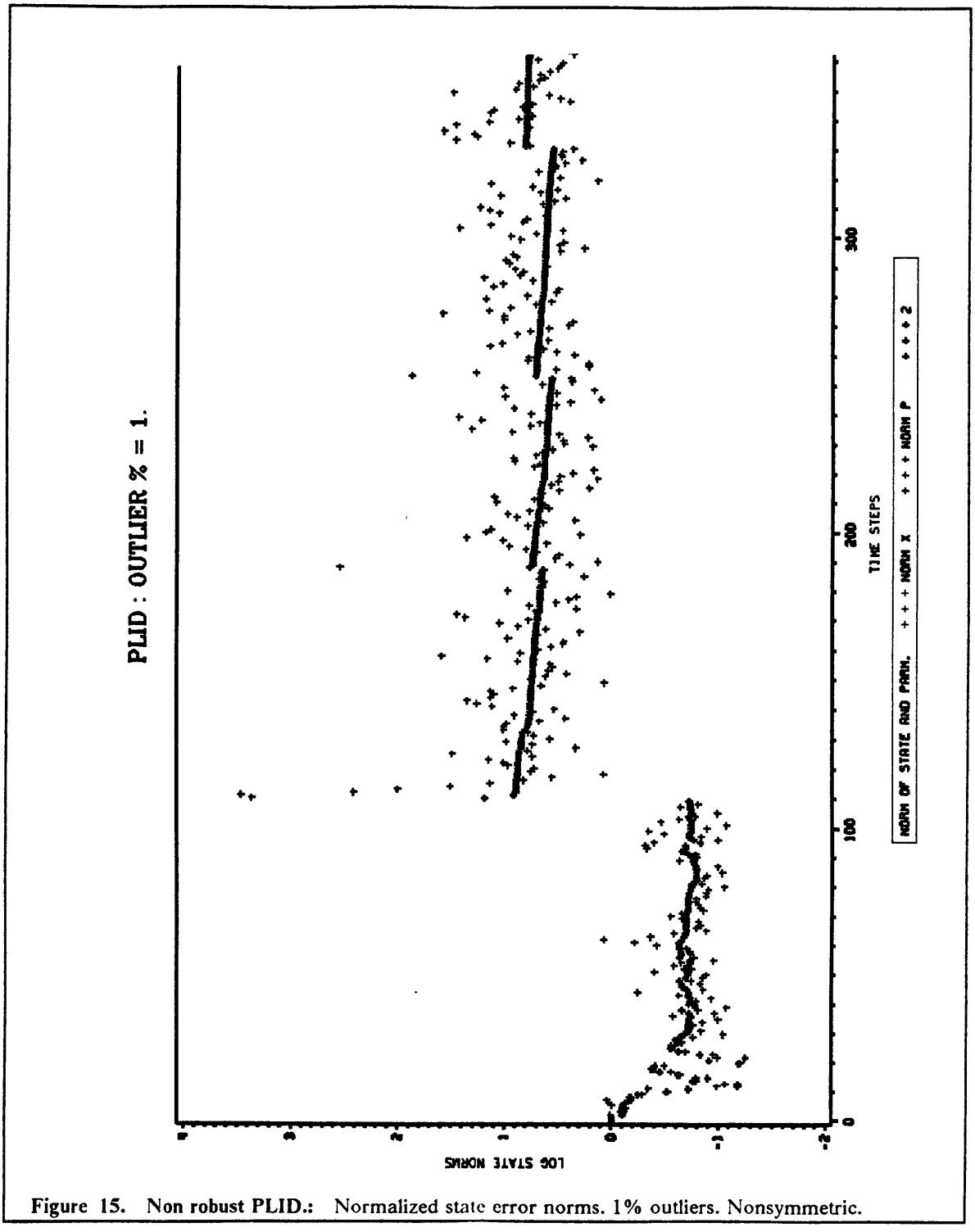

Figure 15. Non robust PLID.: Normalized state error norms. 1% outliers. Nonsymmetric.

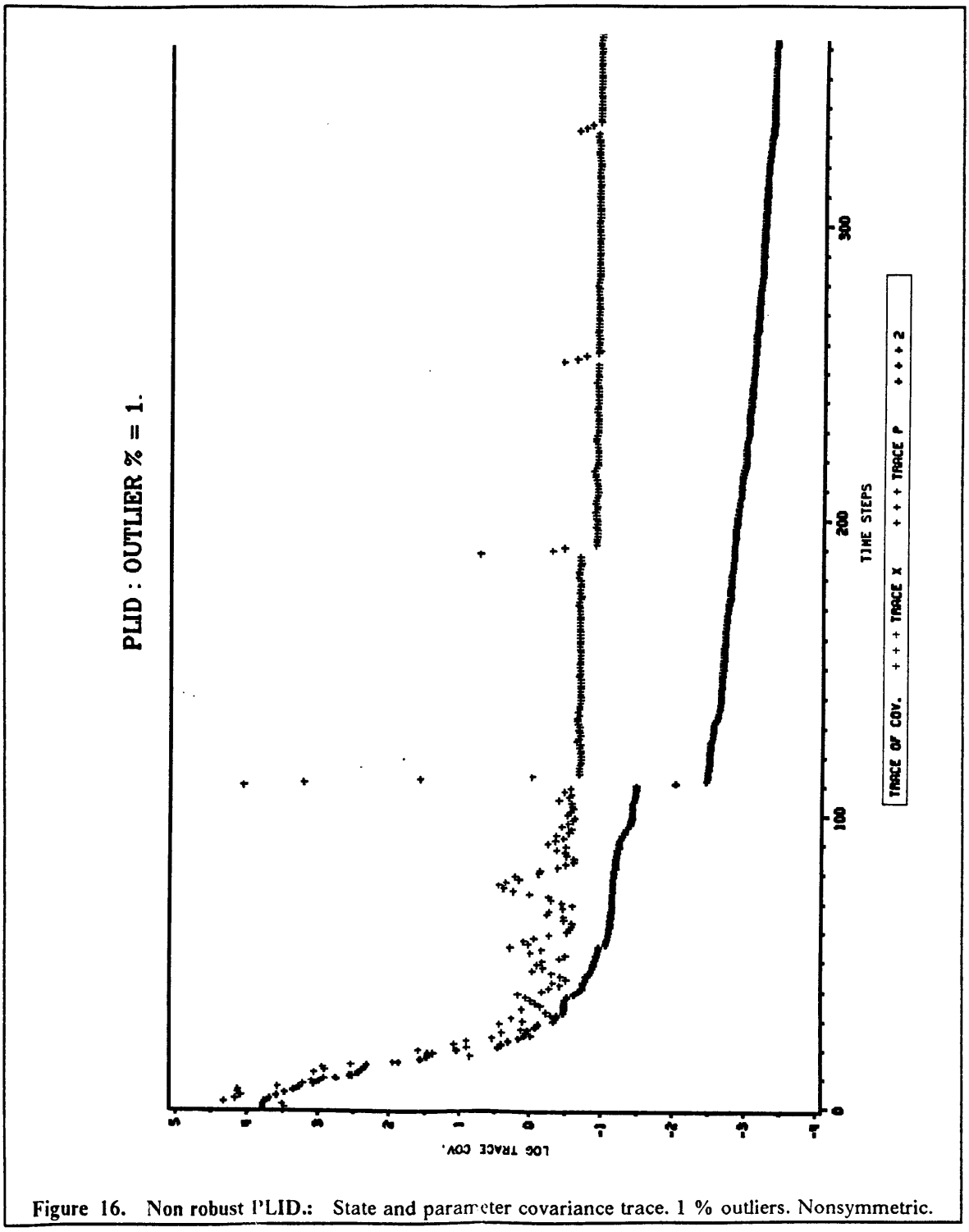

Figure 16. Non robust PLID.: State and parameter covariance trace. 1 % outliers. Nonsymmetric.

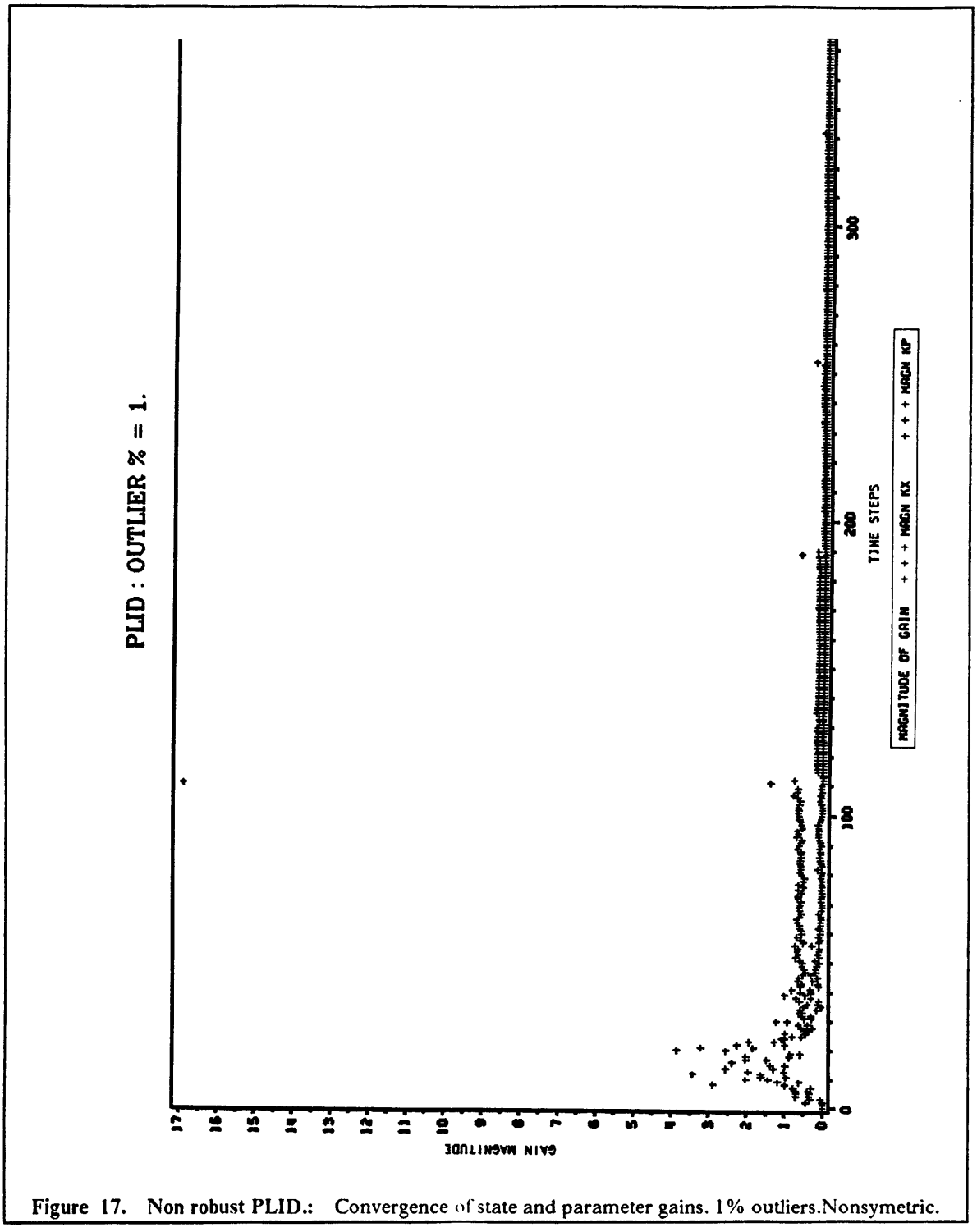

Figure 17. Non robust PLID.: Convergence of state and parameter gains. 1% outliers.Nonsymetric.

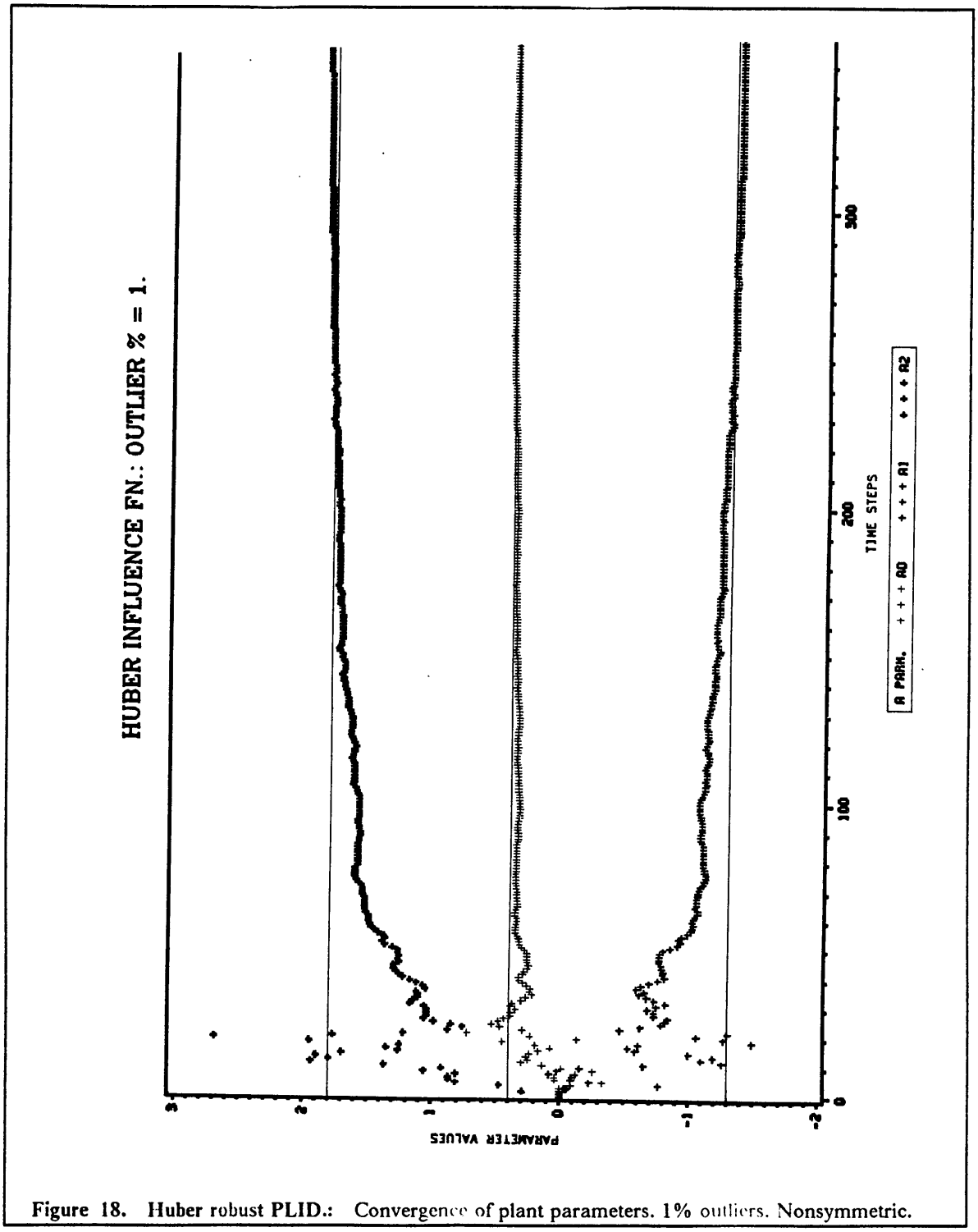

Figure 18. Huber robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

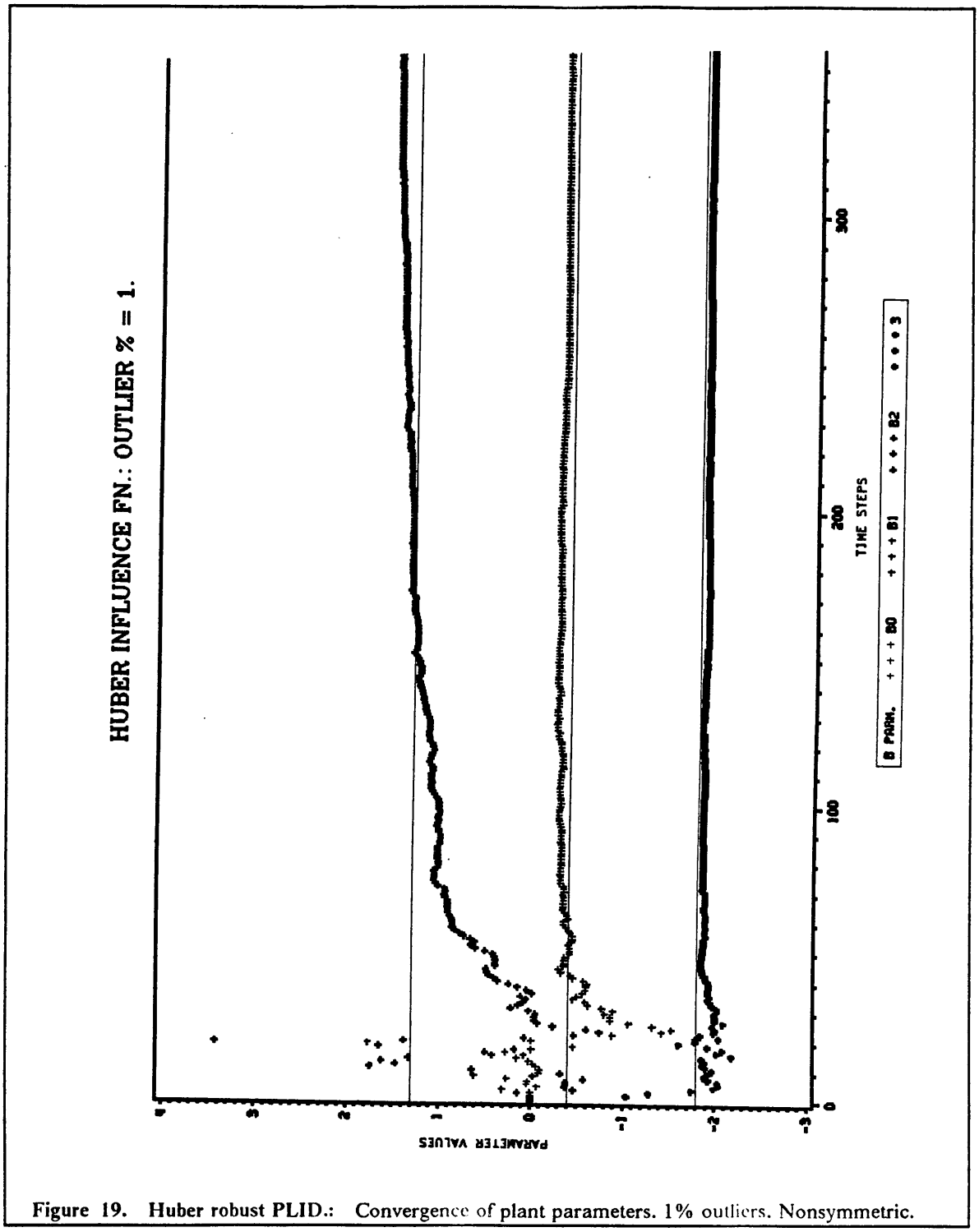

Figure 19. Huber robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

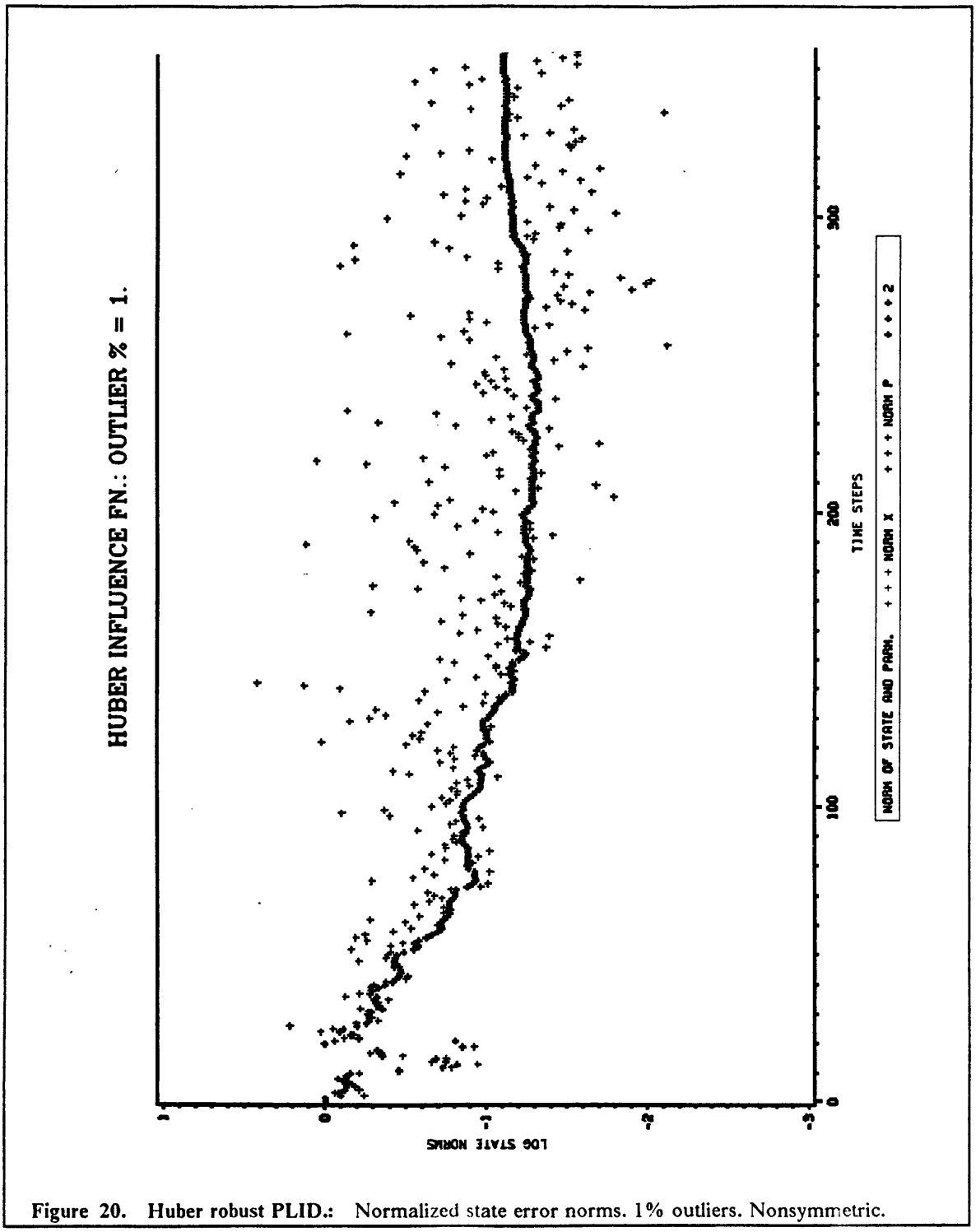

Figure 20. Huber robust PLID.: Normalized state error norms. 1% outliers. Nonsymmetric.

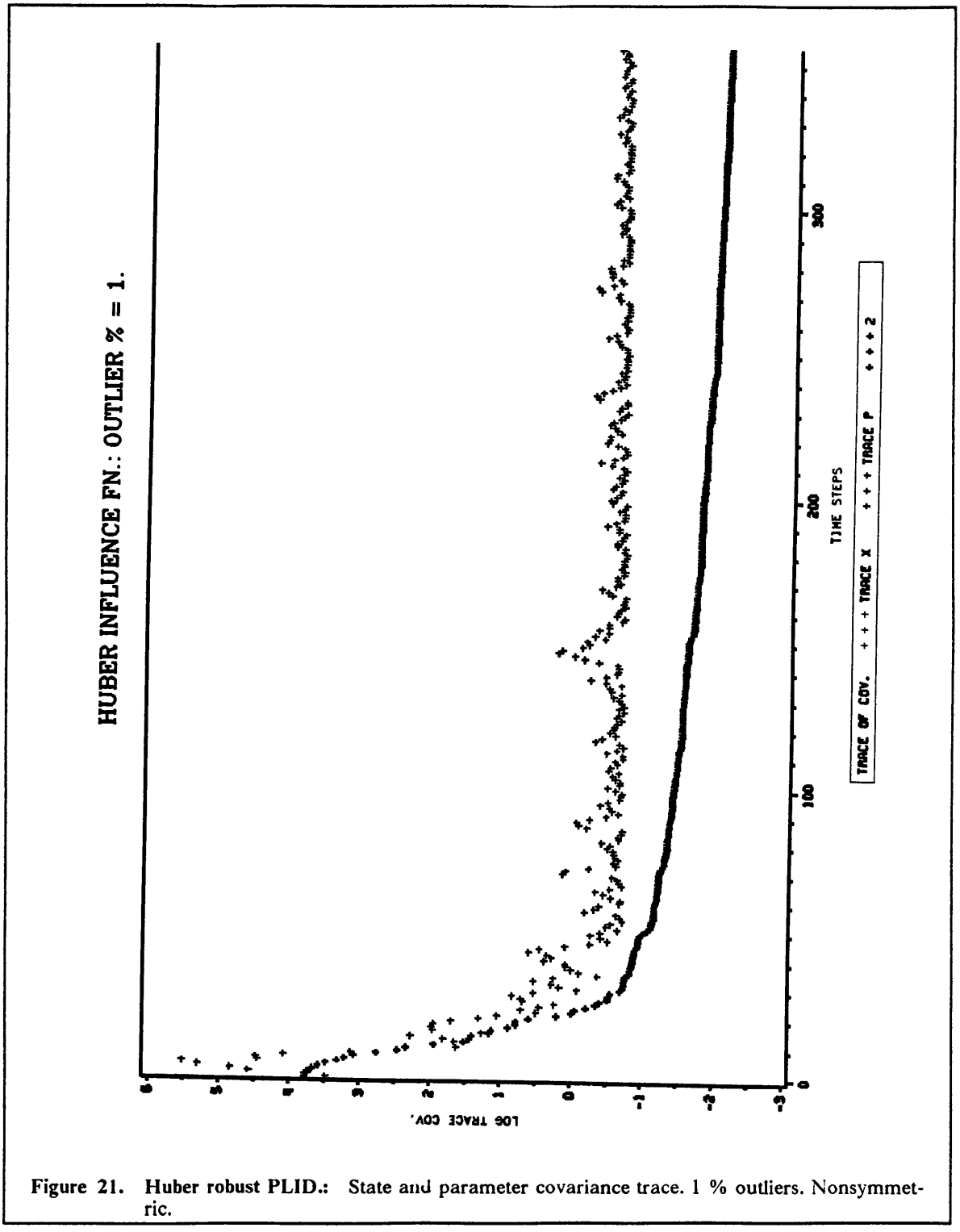

Figure 21. Huber robust PLID.: State and parameter covariance trace. 1 % outliers. Nonsymmetric.

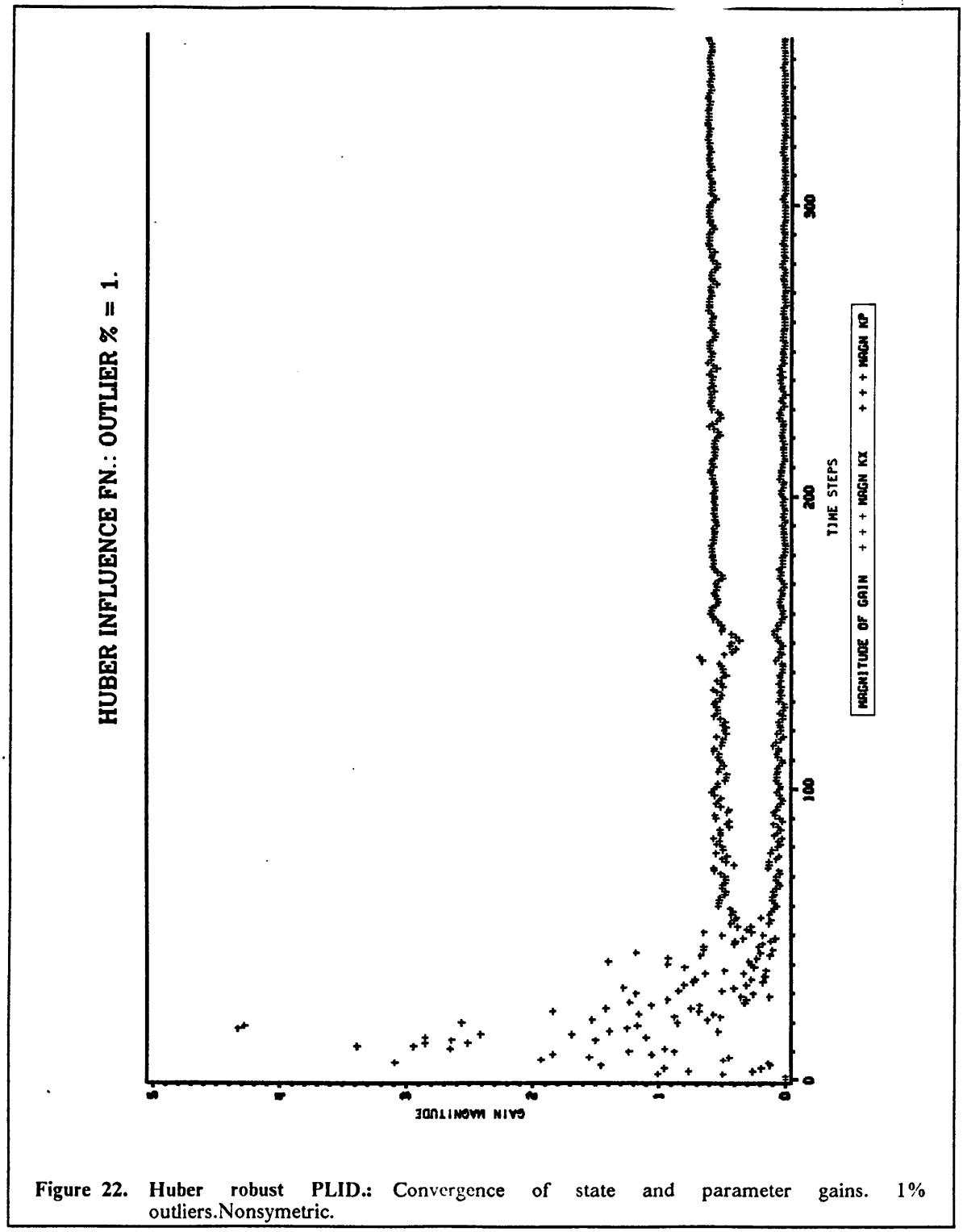

Figure 22. **Huber robust PLID.:** Convergence of state and parameter gains. 1% outliers.Nonsymetric.

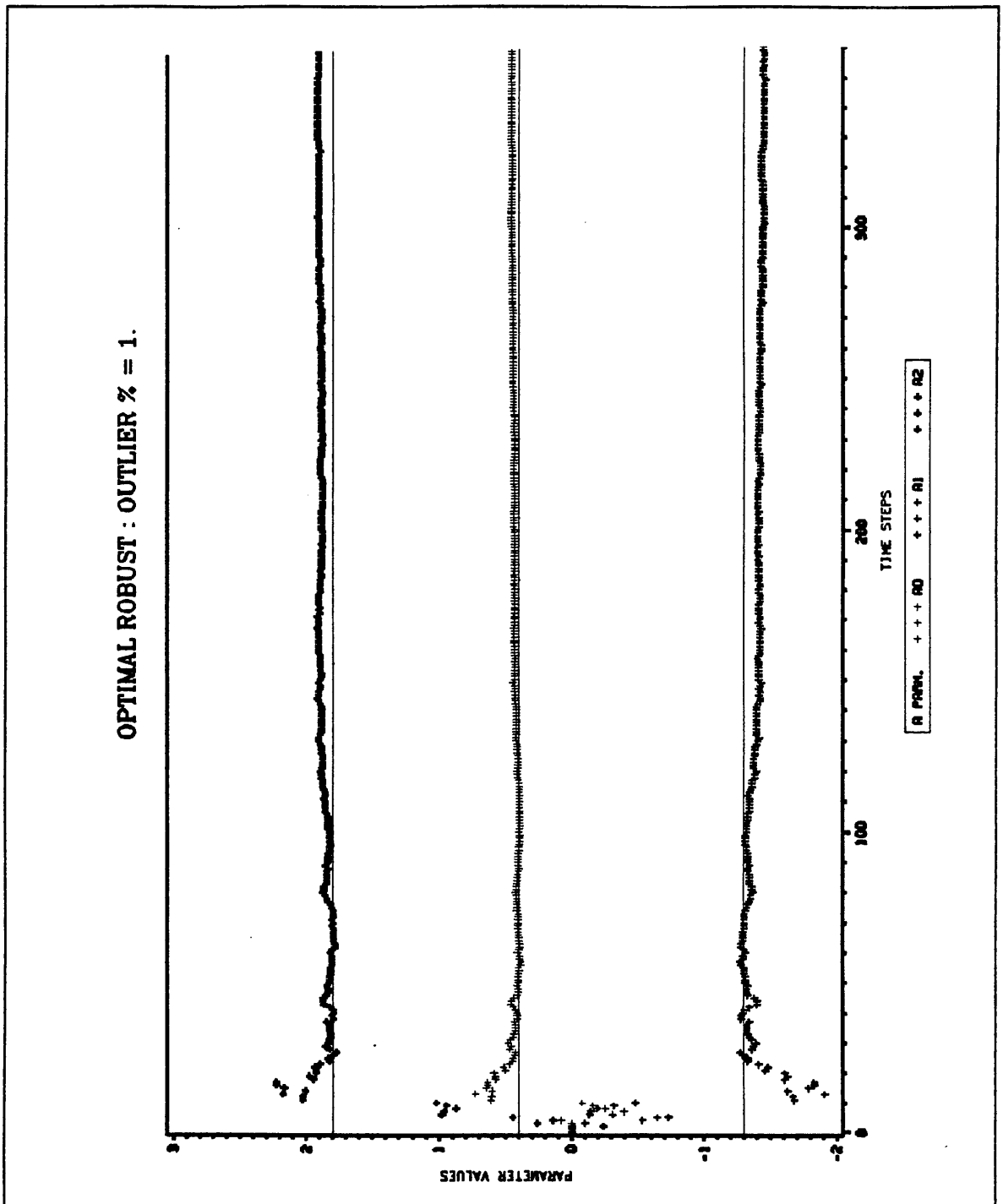

Figure 23. Hampel robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

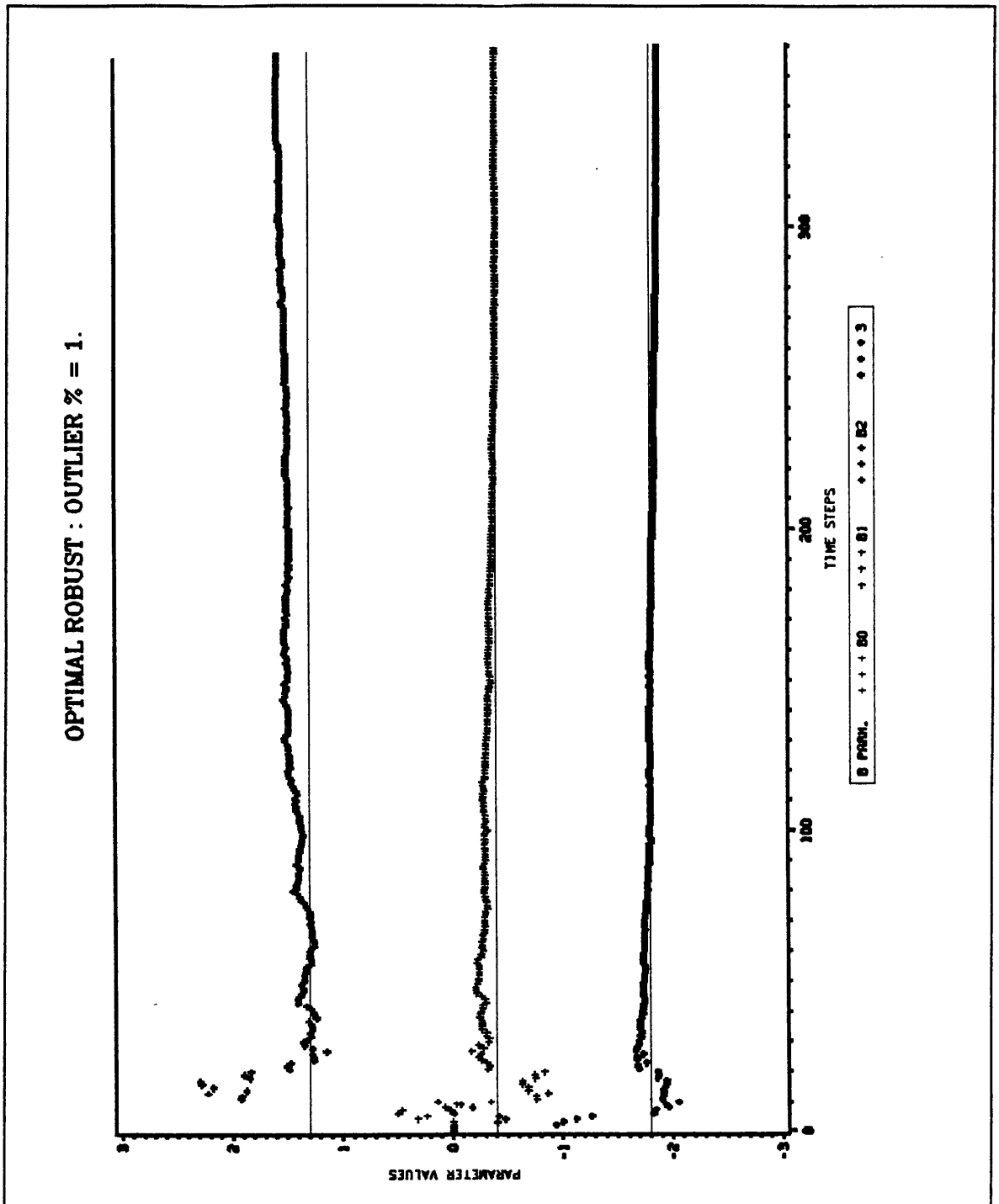

Figure 24. Hampel robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

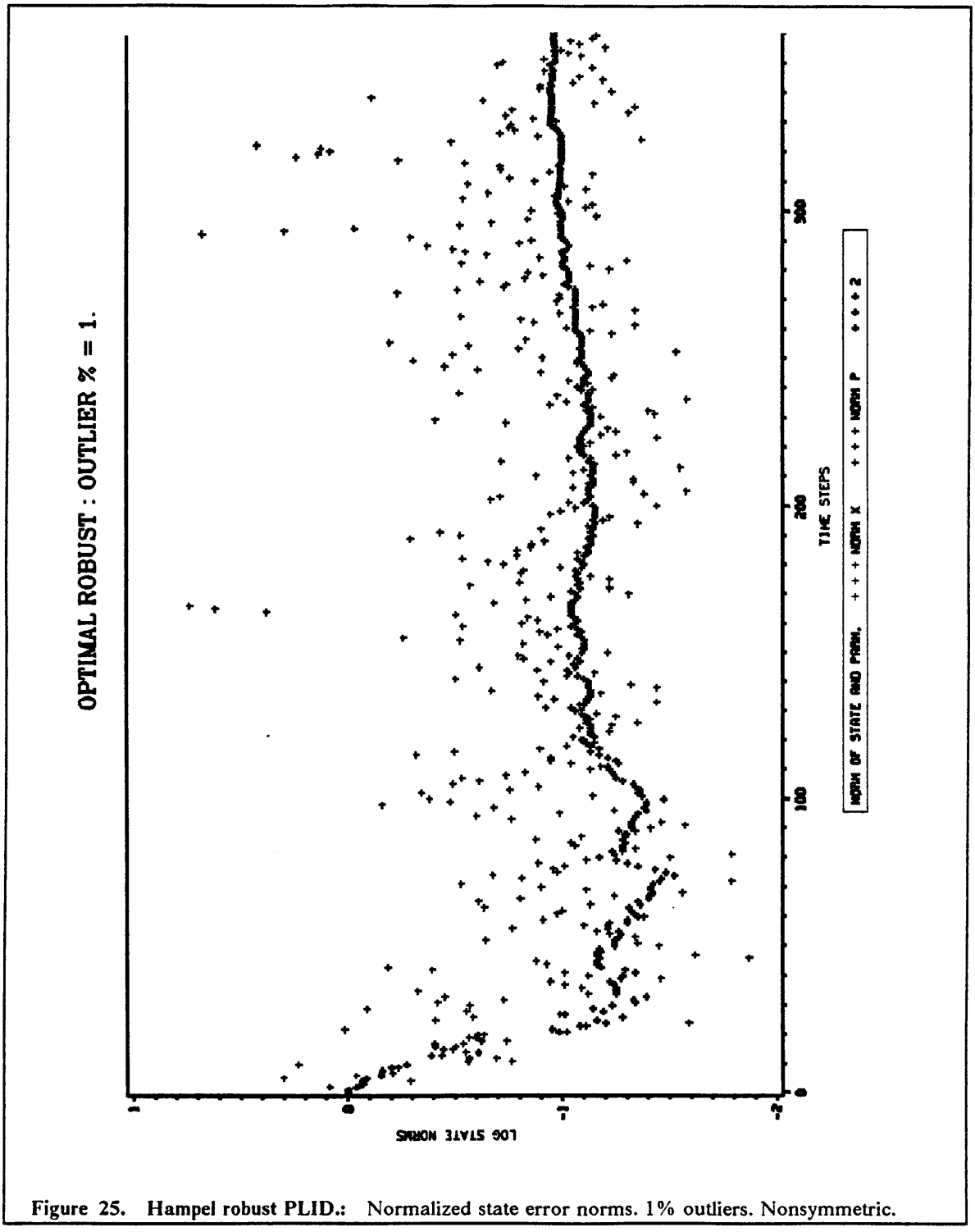

Figure 25. **Hampel robust PLID.:** Normalized state error norms. 1% outliers. Nonsymmetric.

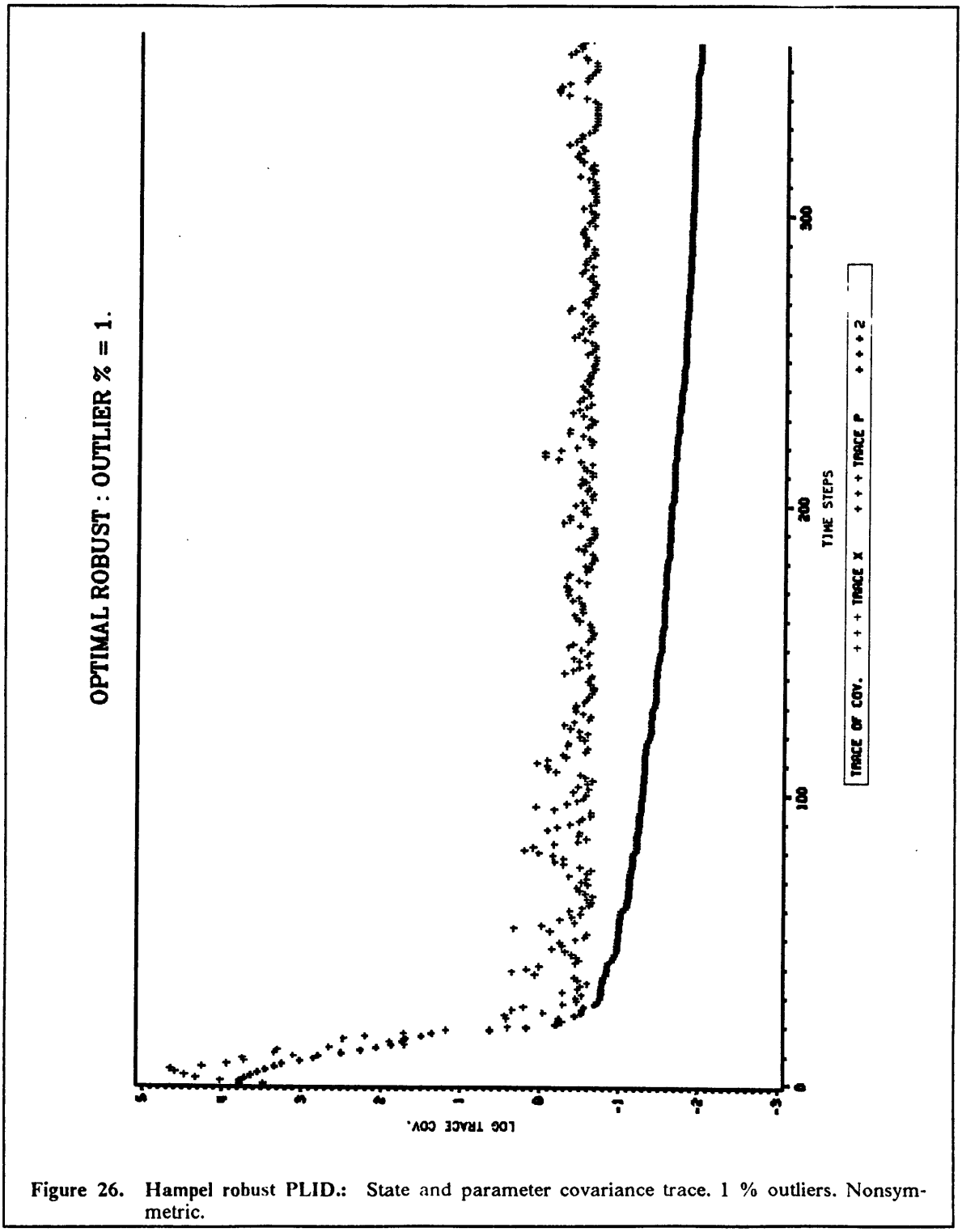

Figure 26. **Hampel robust PLID.:** State and parameter covariance trace. 1 % outliers. Nonsymmetric.

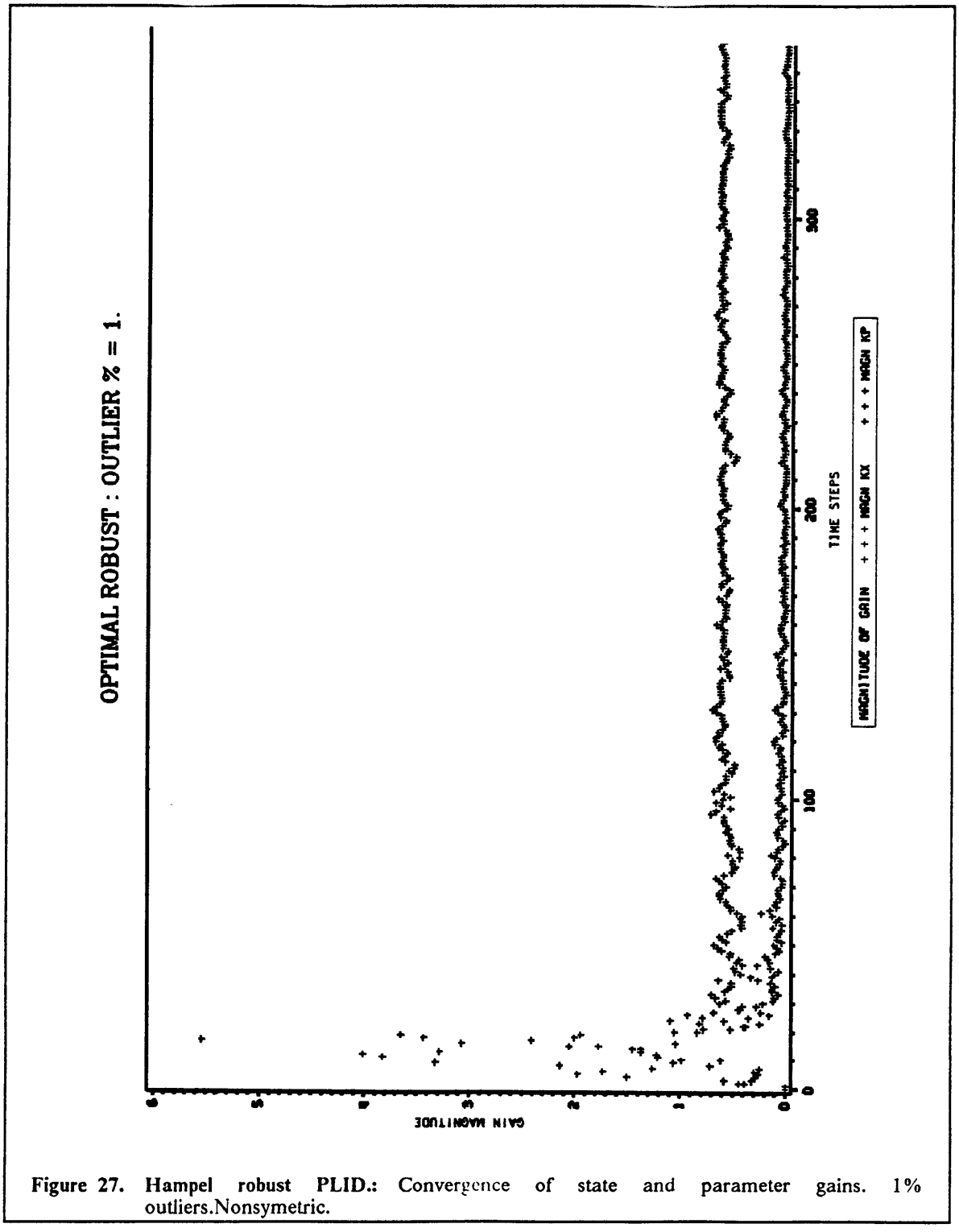

Figure 27. **Hampel robust PLID.:** Convergence of state and parameter gains. 1% outliers.Nonsymetric.

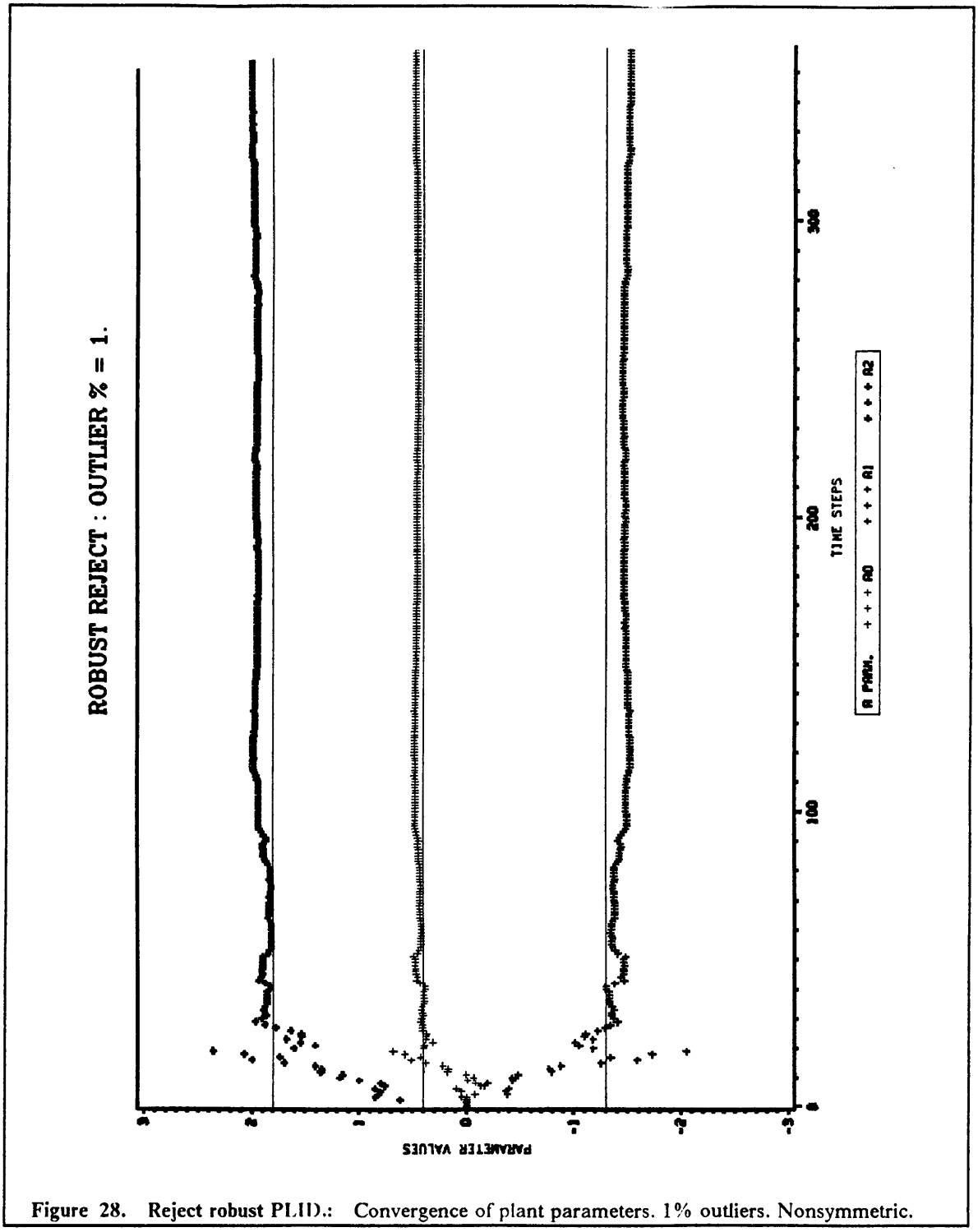

Figure 28. Reject robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

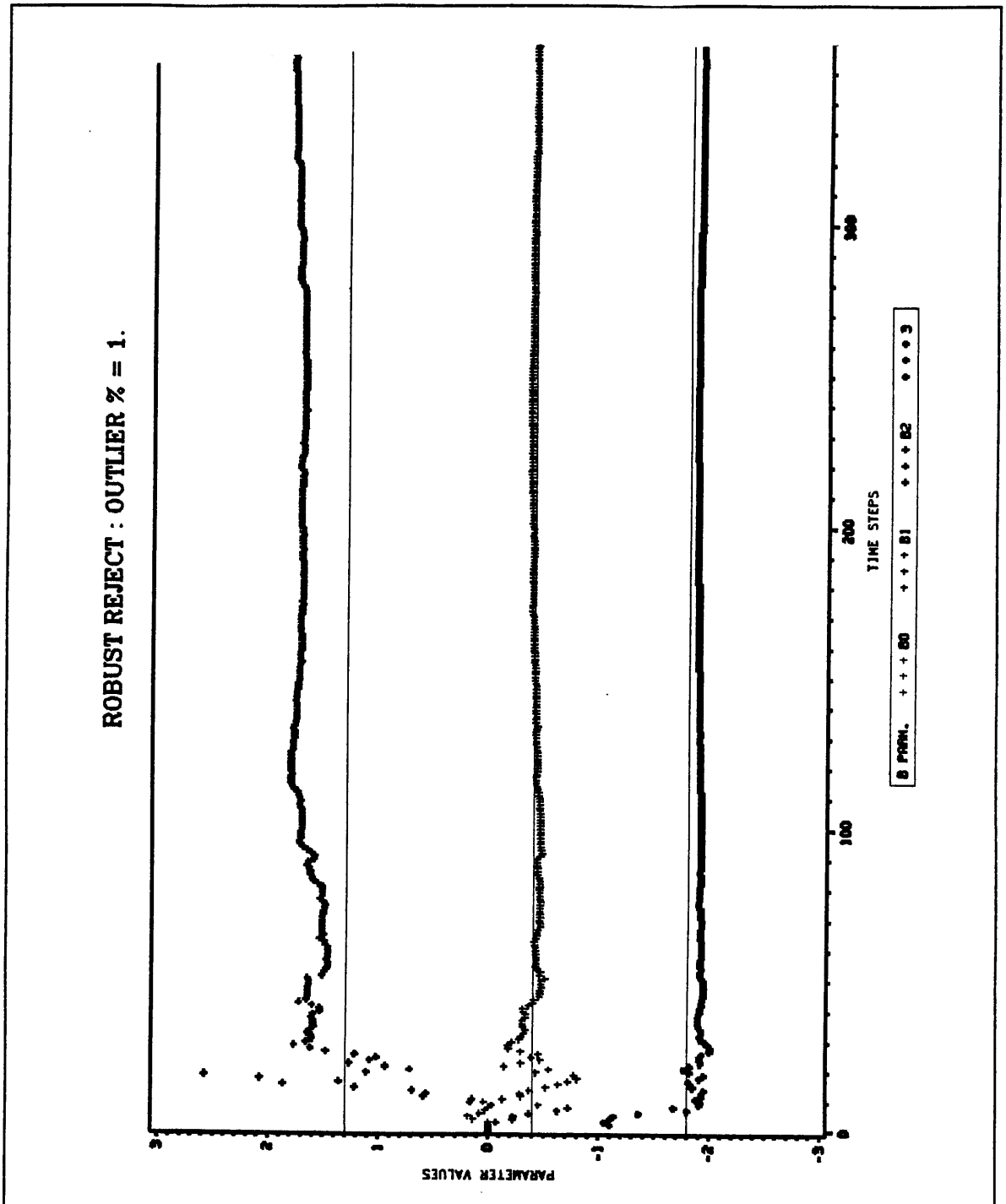

Figure 29. Reject robust PLID.: Convergence of plant parameters. 1% outliers. Nonsymmetric.

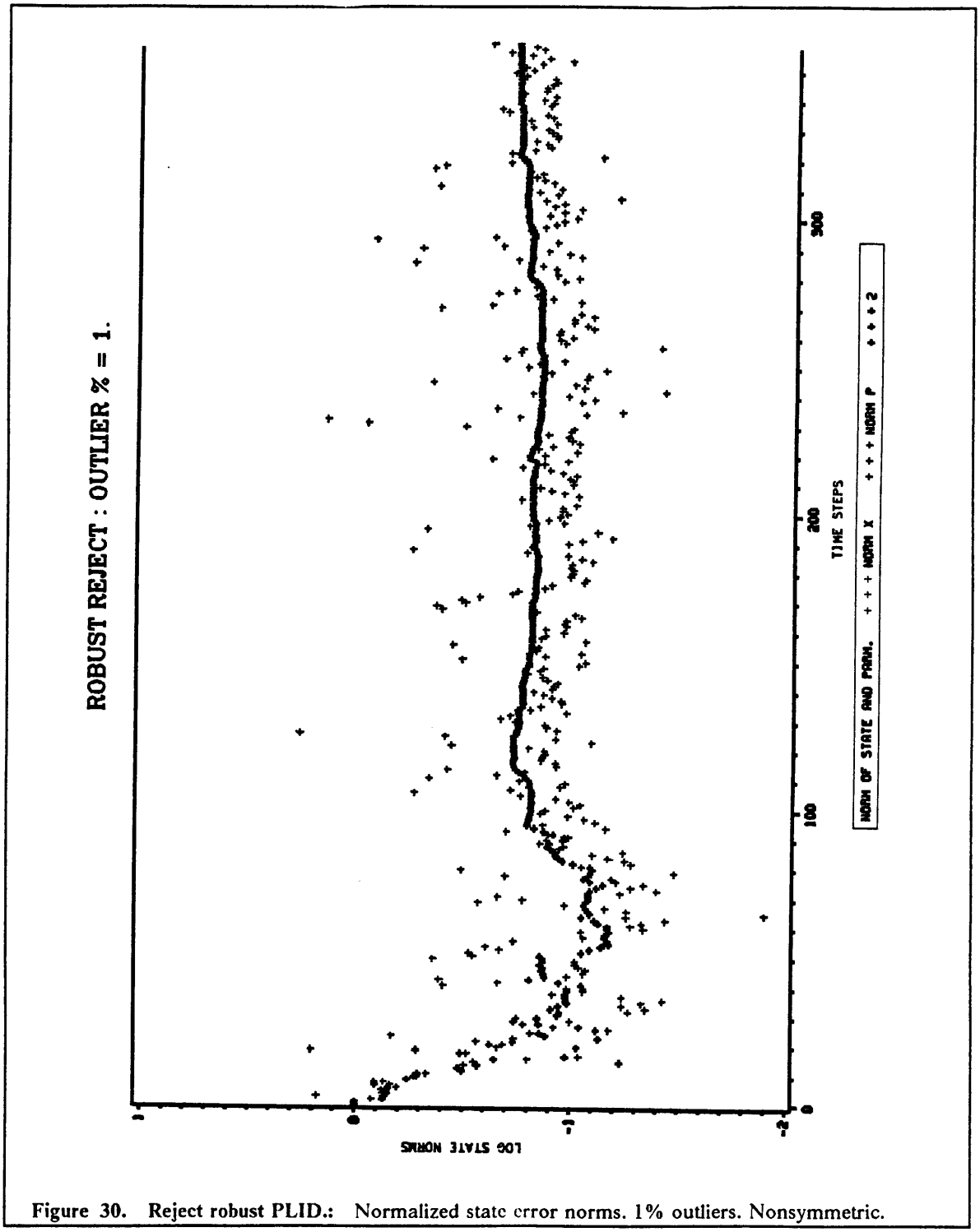

Figure 30. Reject robust PLID.: Normalized state error norms. 1% outliers. Nonsymmetric.

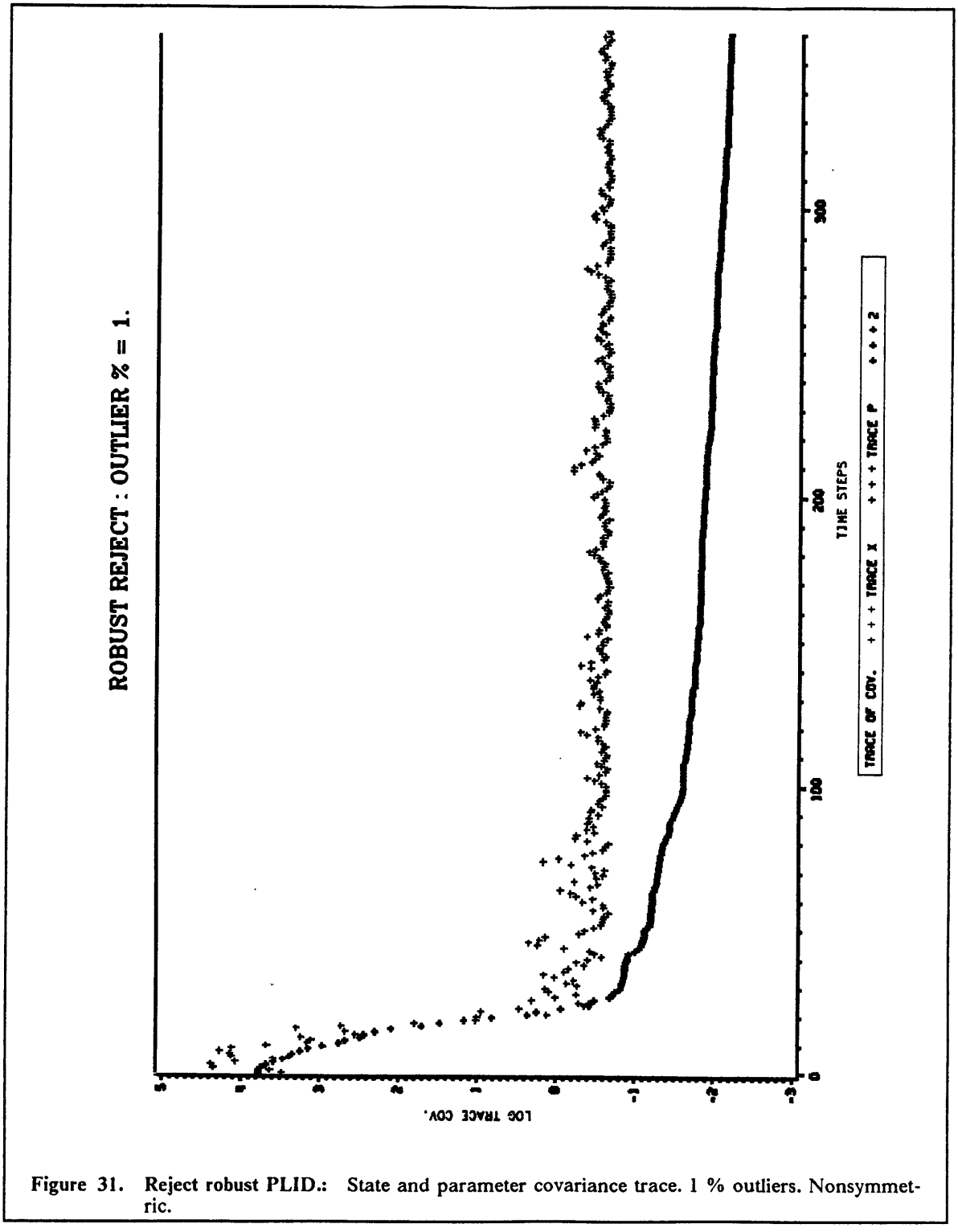

Figure 31. Reject robust PLID.: State and parameter covariance trace. 1 % outliers. Nonsymmetric.

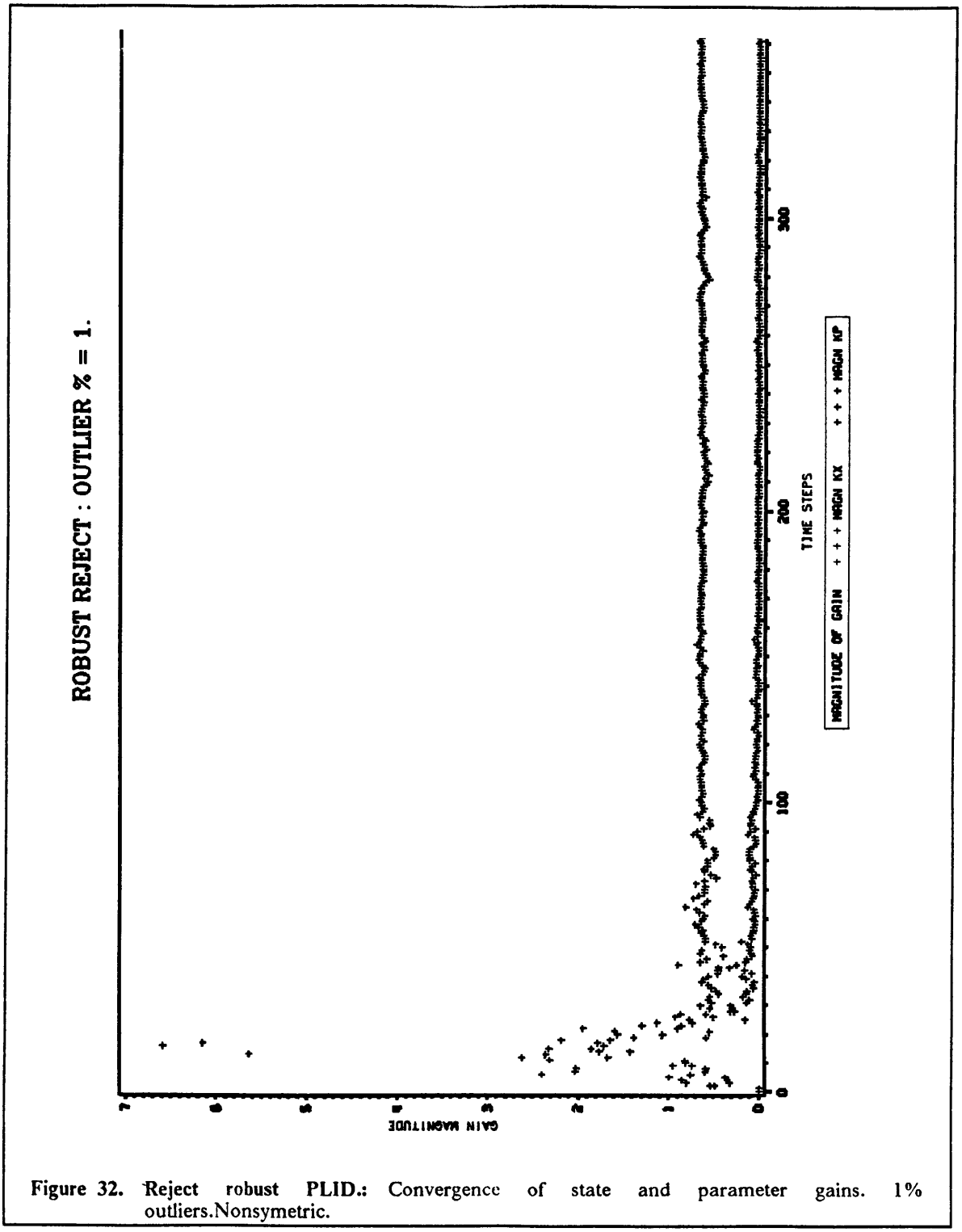

Figure 32. Reject robust PLID.: Convergence of state and parameter gains. 1% outliers.Nonsymetric.

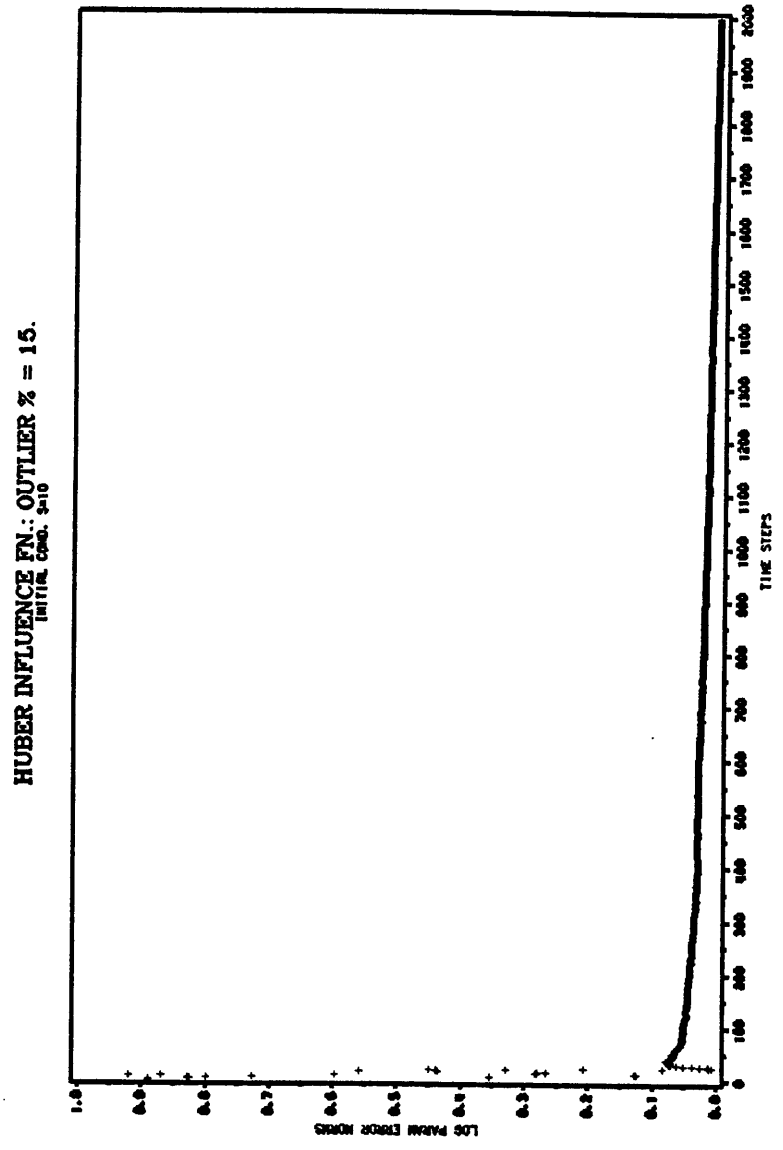

Figure 33. Huber robust PLID.: Logarithmic error norm. 15% outl., S(0)=10

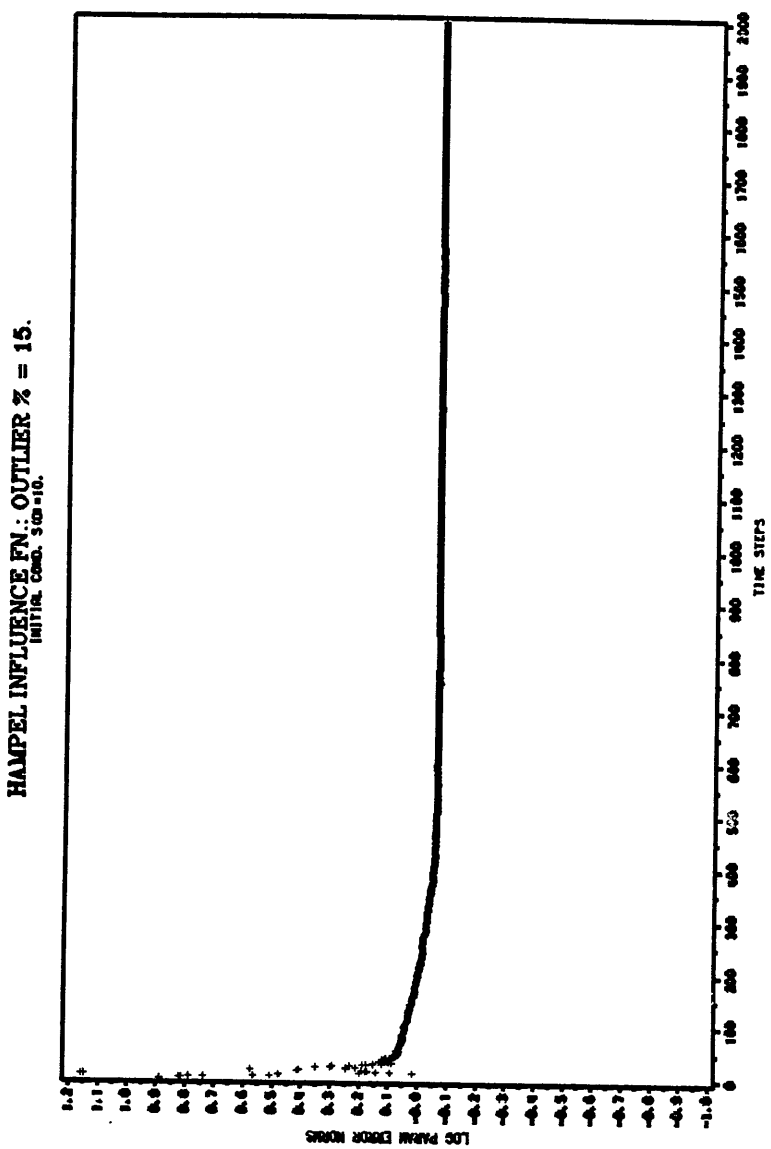

Figure 34. Hampel robust PLID.: Logarithmic error norm. 15% outl., S(0) = 10

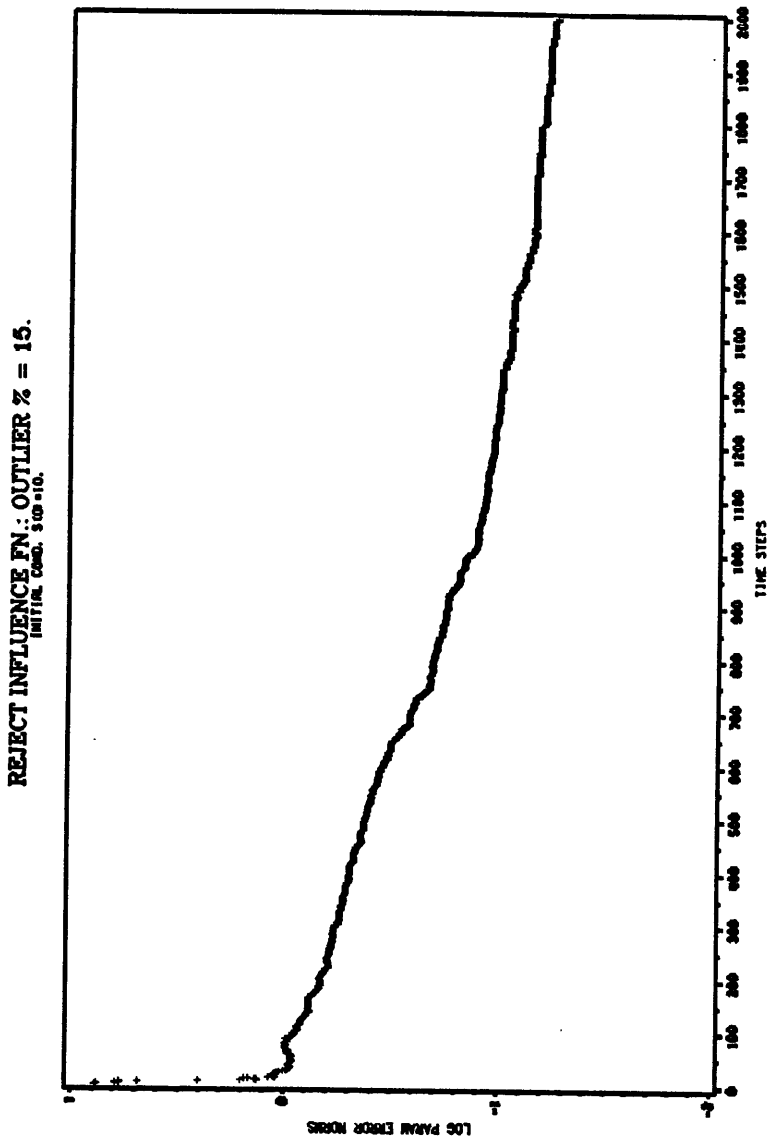

Figure 35. Reject robust PLID.: Logarithmic error norm. 15% outl., S(0) = 10

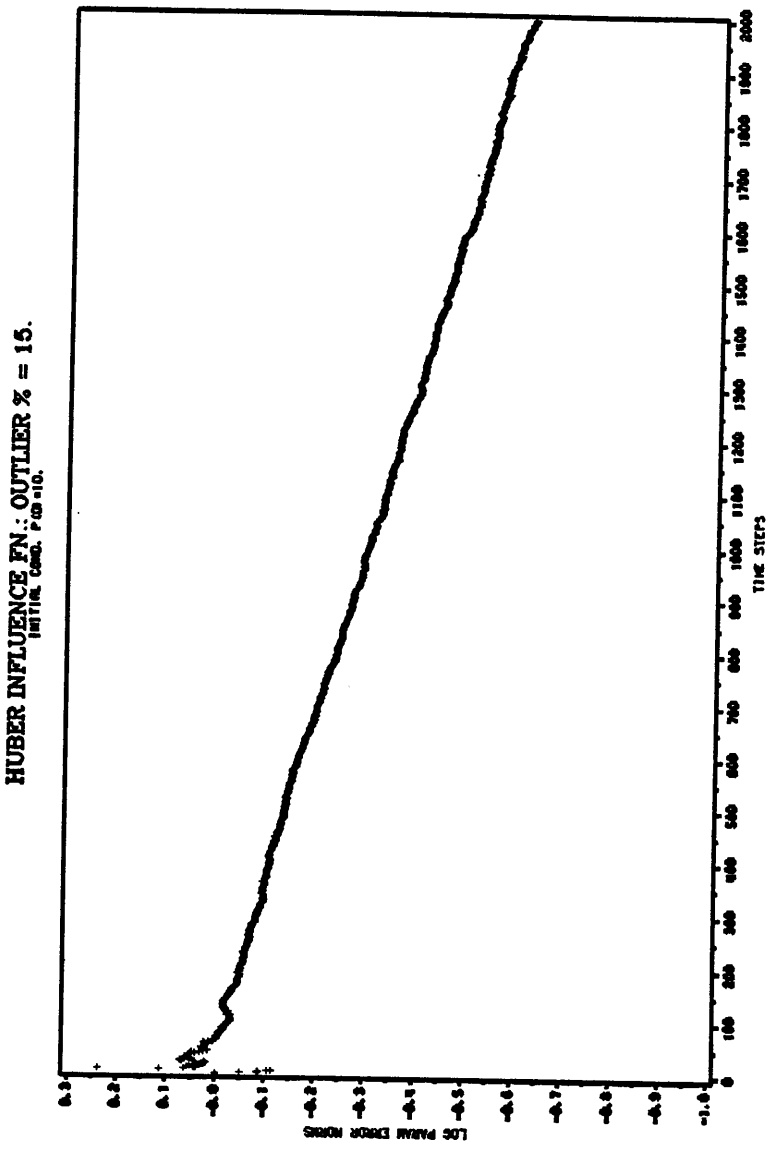

Figure 36. Huber robust PLID.: Logarithmic error norm. 15% outl., P(0)=10

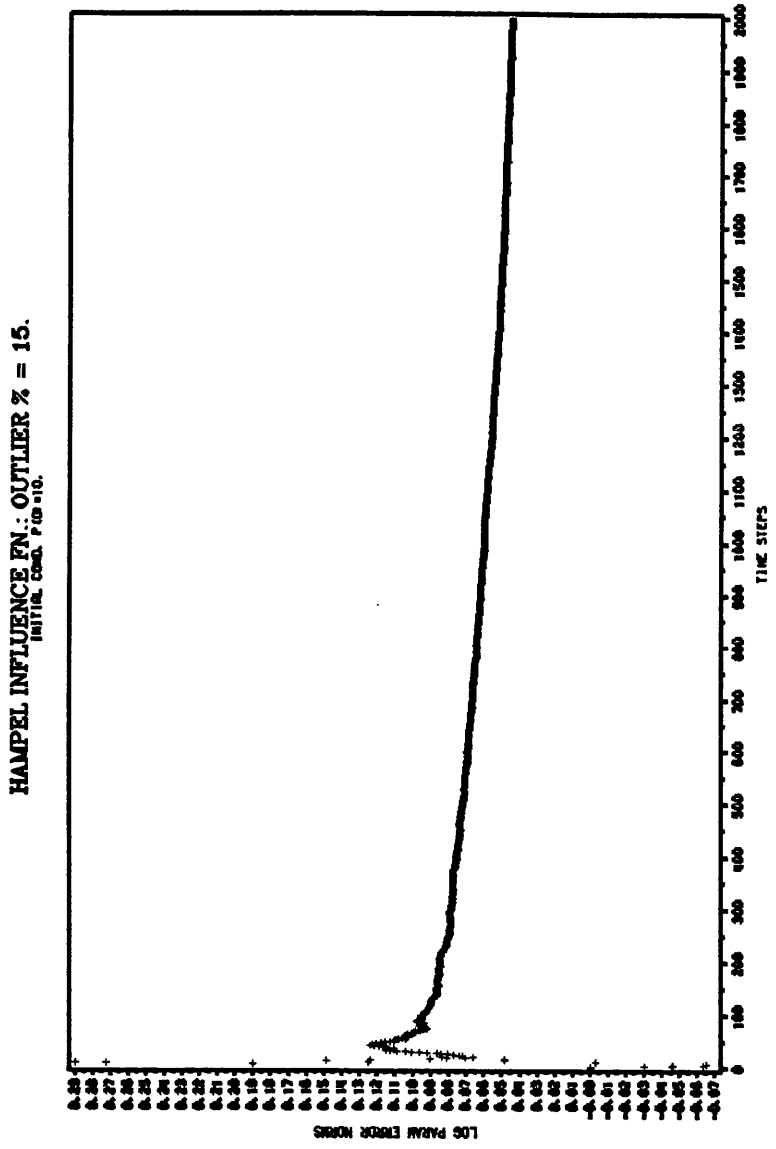

Figure 37. Hampel robust PLID.: Logarithmic error norm. 15% outl., P(0)=10

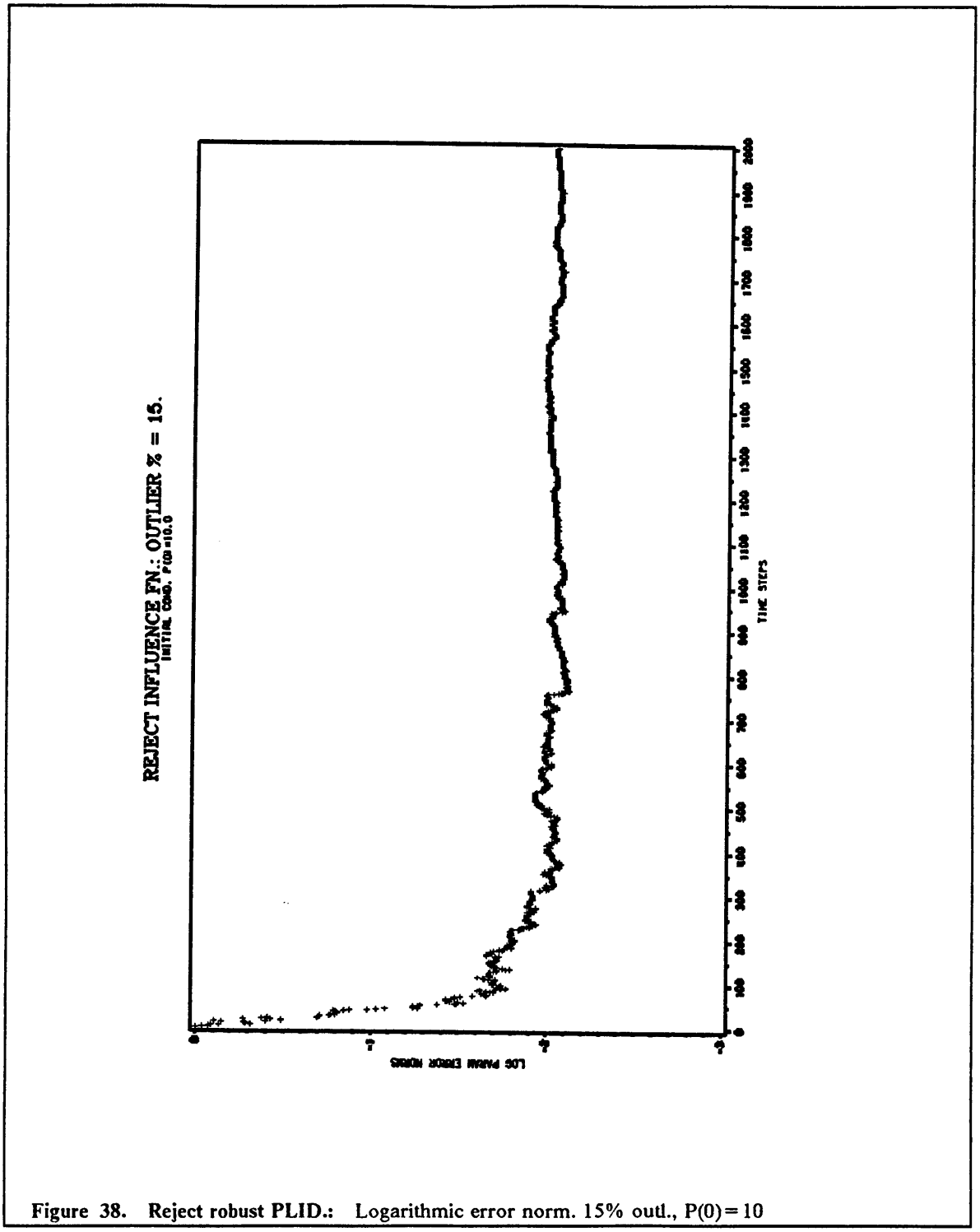

Figure 38. Reject robust PLID.: Logarithmic error norm. 15% outl., P(0)=10

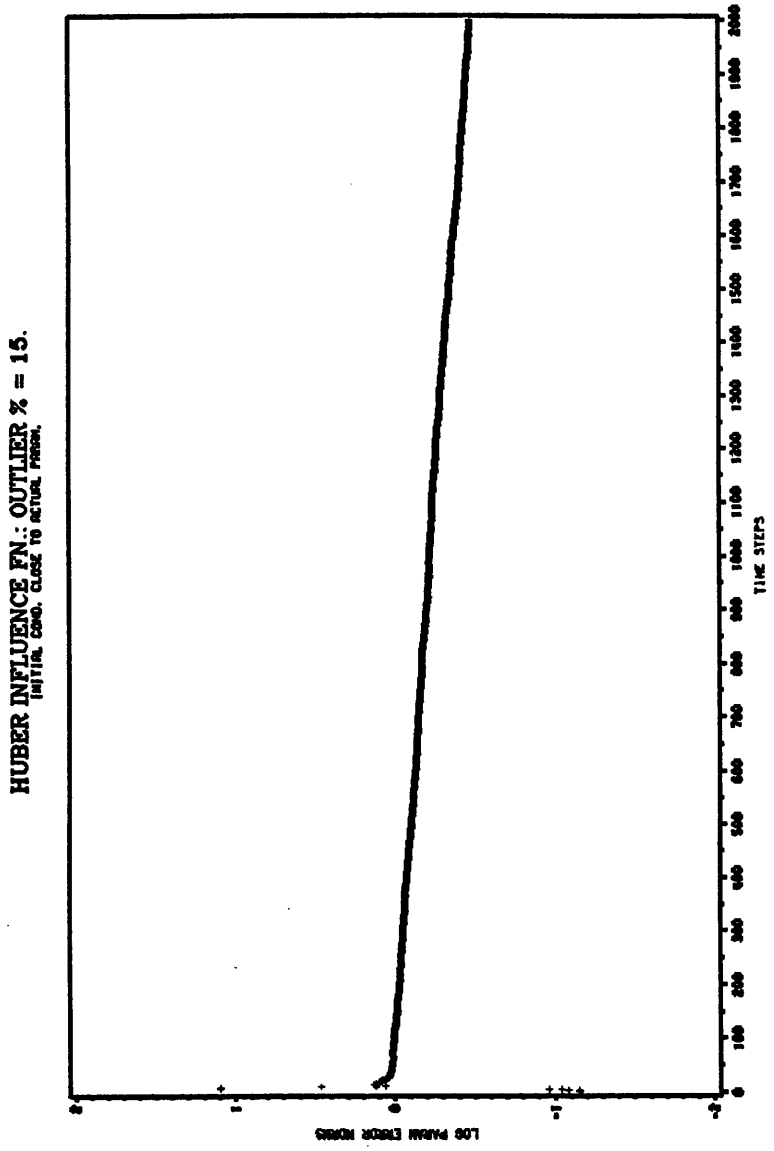

Figure 39. Huber robust PLID.: Logarithmic error norm. 15% outl., S(0) close to true param.

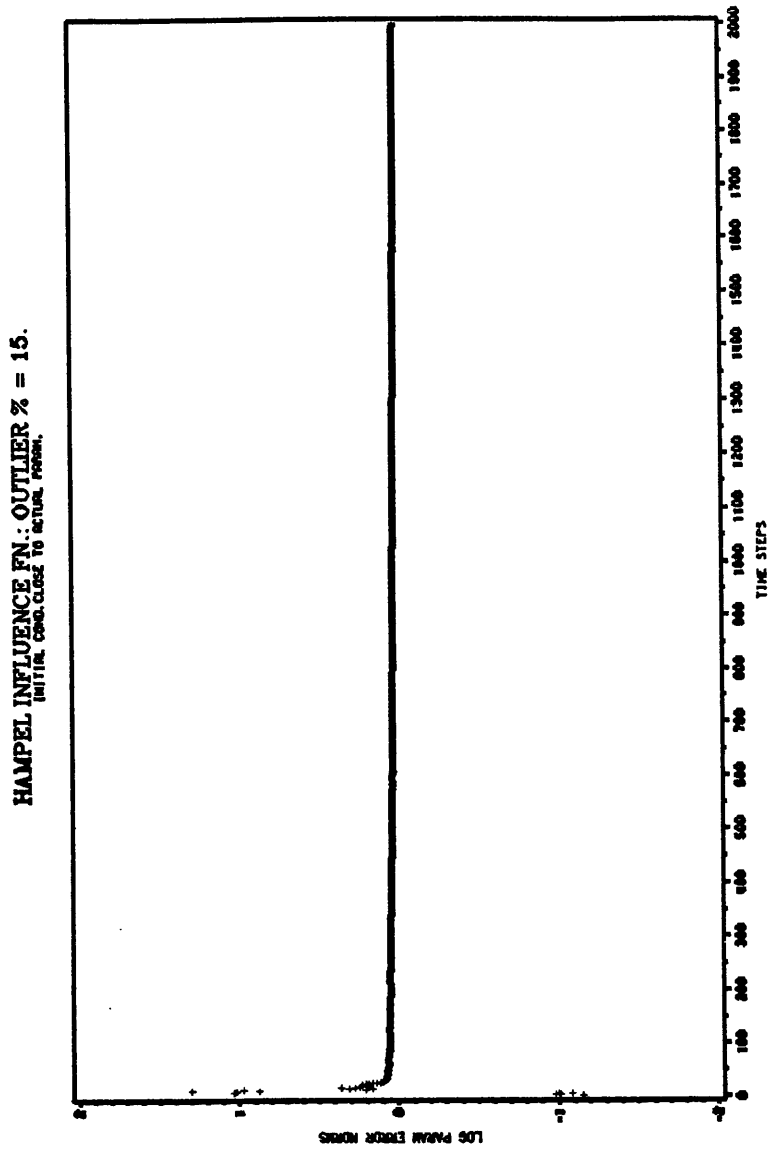

Figure 40. Hampel robust PLID.: Logarithmic error norm. 15% outl., S(0) close to true param.

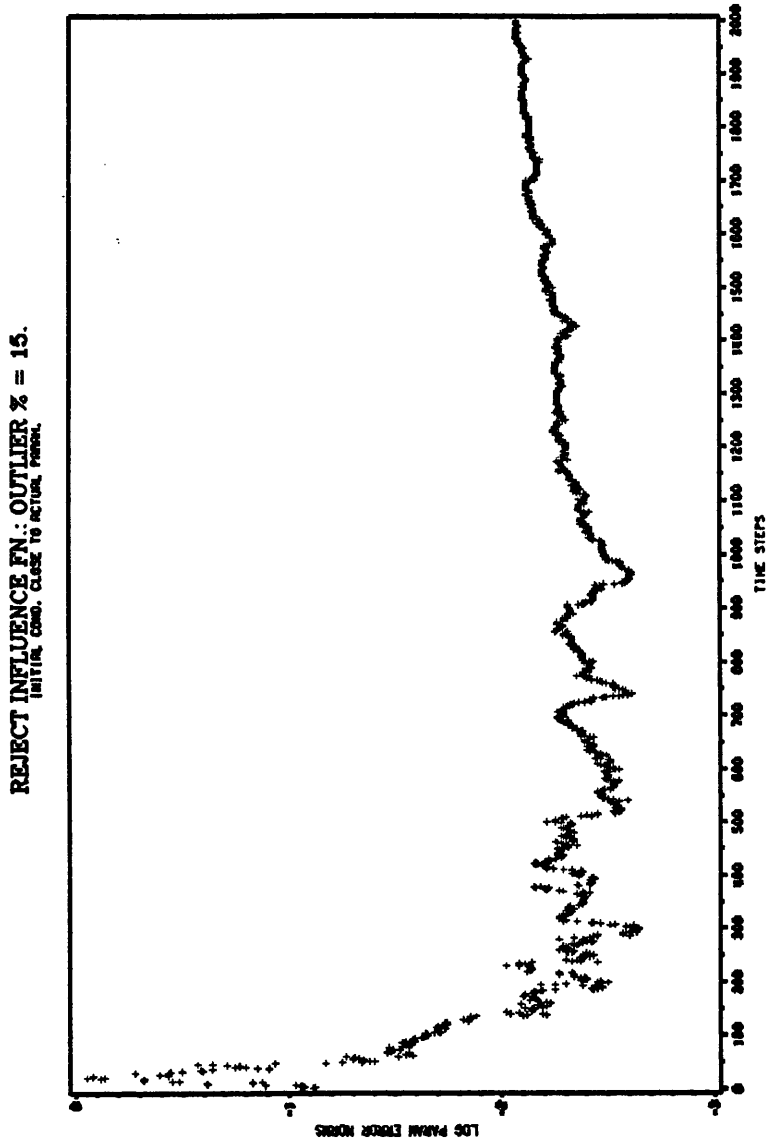

Figure 41. Reject robust PLID.: Logarithmic error norm. 15% outl., S(0) close to true param.

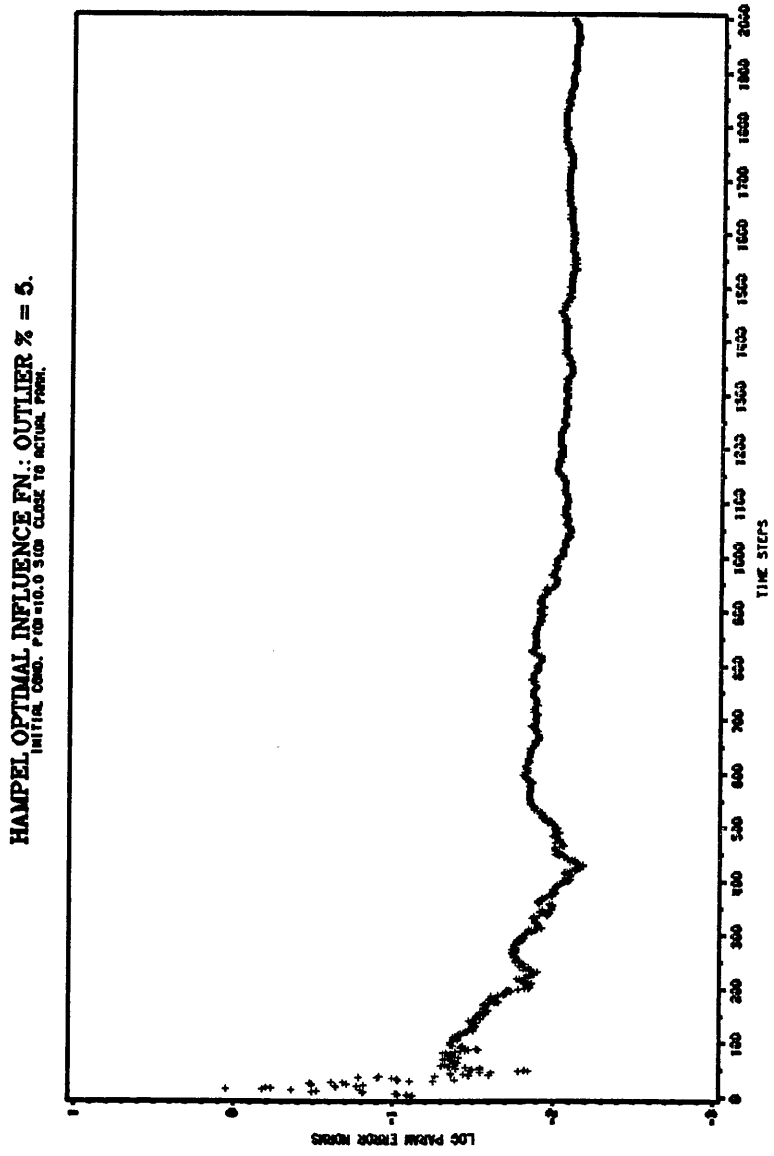

Figure 42. **Hampel robust PLID.:** Logarithmic error norm. 15% outl., S(0) close to true param. P(0) = 10 low sensitivity.

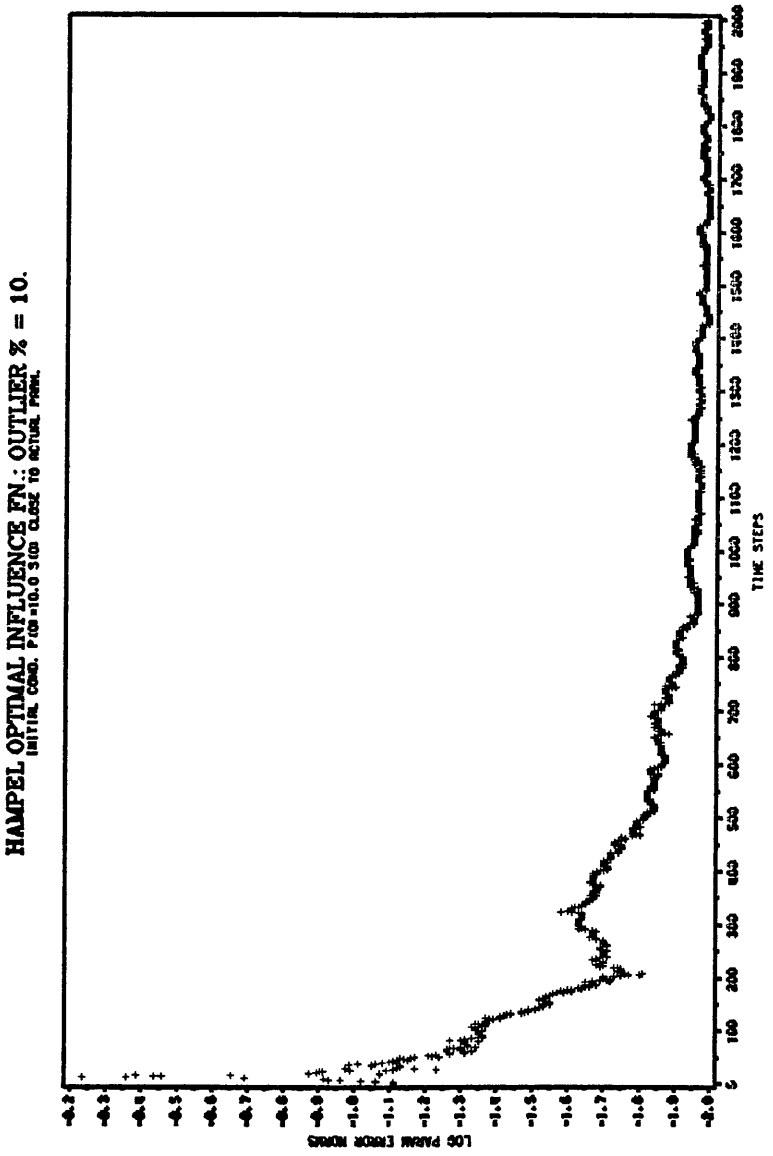

Figure 43. **Hampel robust PLID.:** Logarithmic error norm. 15% outl., S(0) close to true param. P(0) = 10 low sensitivity.

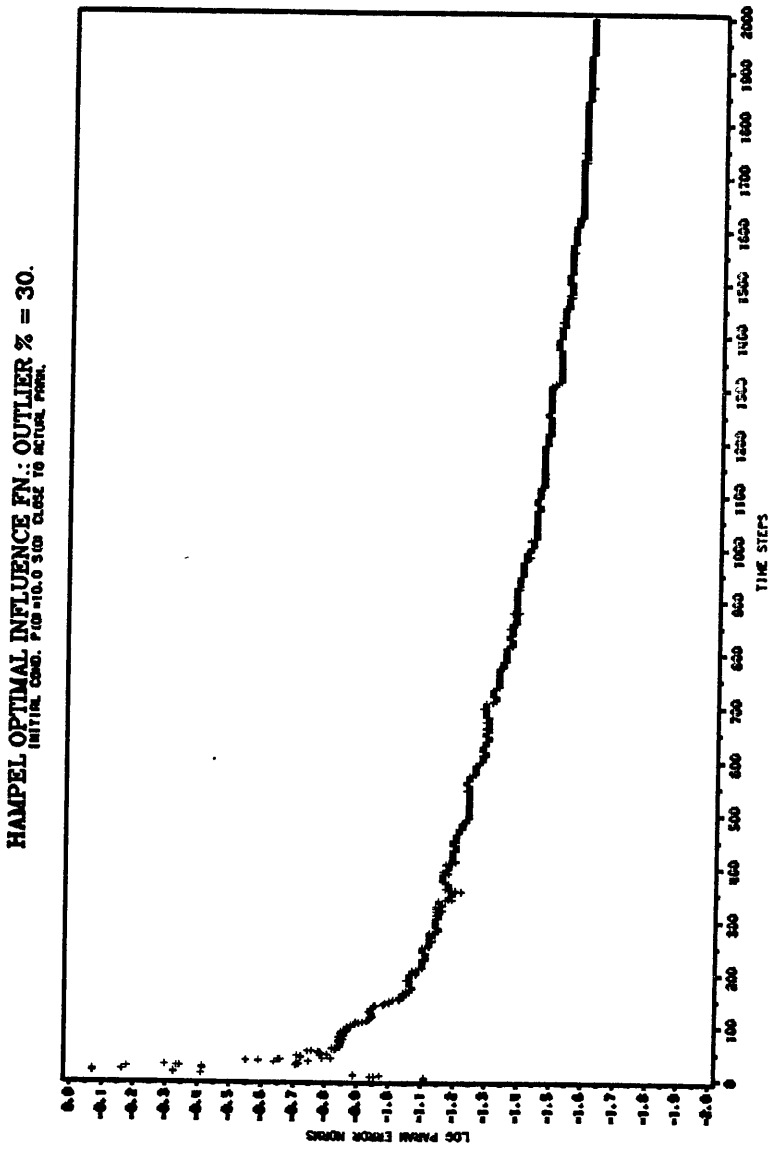

Figure 44. **Hampel robust PLID.:** Logarithmic error norm. 15% outl., S(0) close to true param. P(10) = 10 low sensitivity.

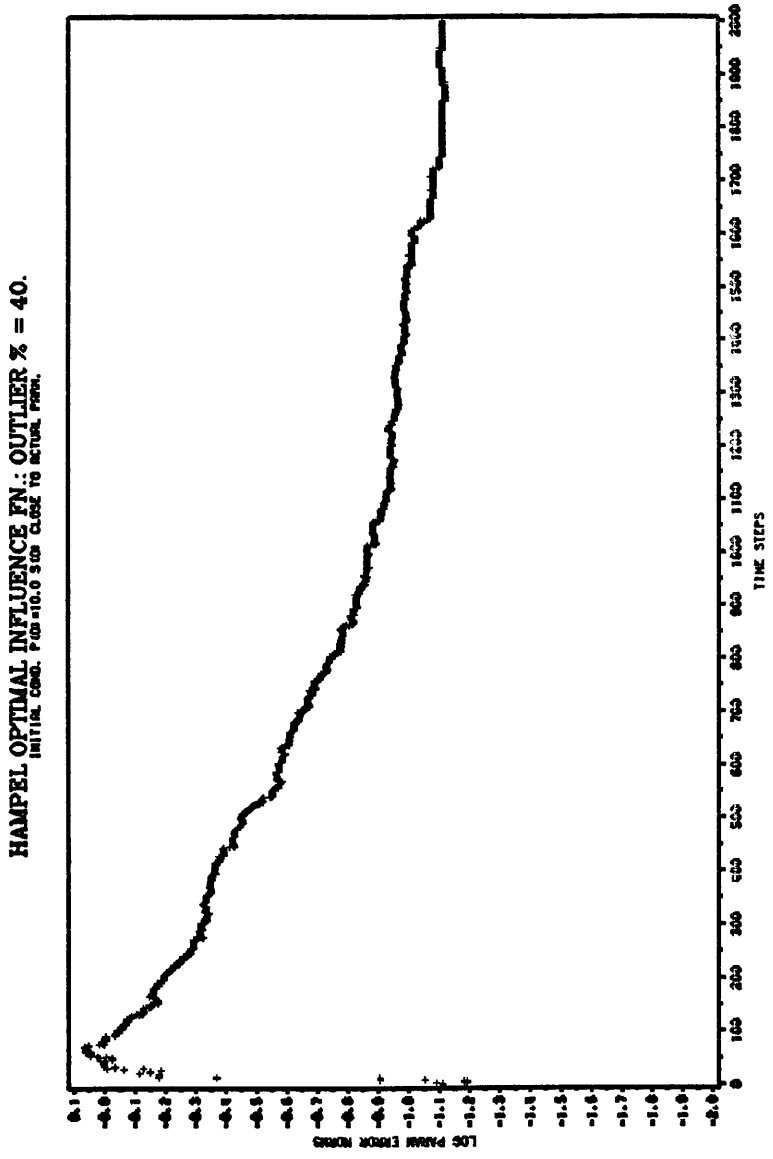

Figure 45. **Hampel robust PLID.:** Logarithmic error norm. 15% outl., S(0) close to true param. P(10) = 10 low sensitivity.

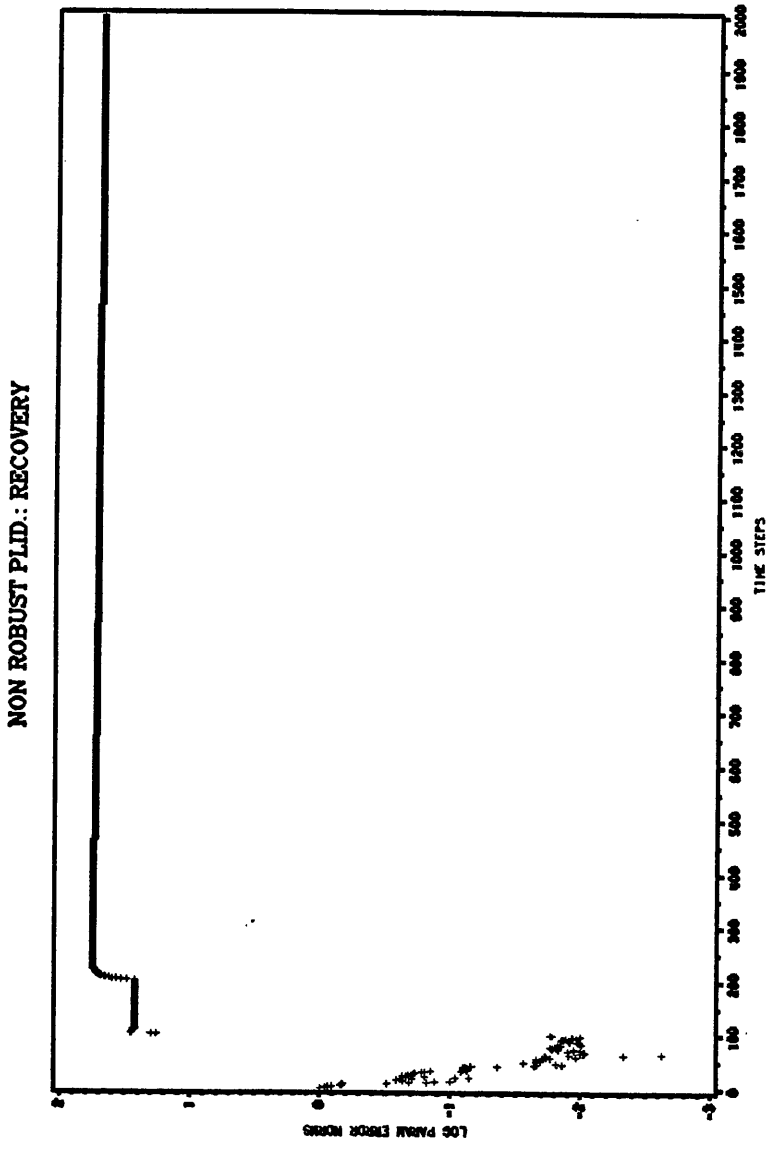

Figure 46. Huber robust PLID.: Logarithmic error norm. Recovery from outlier burst.

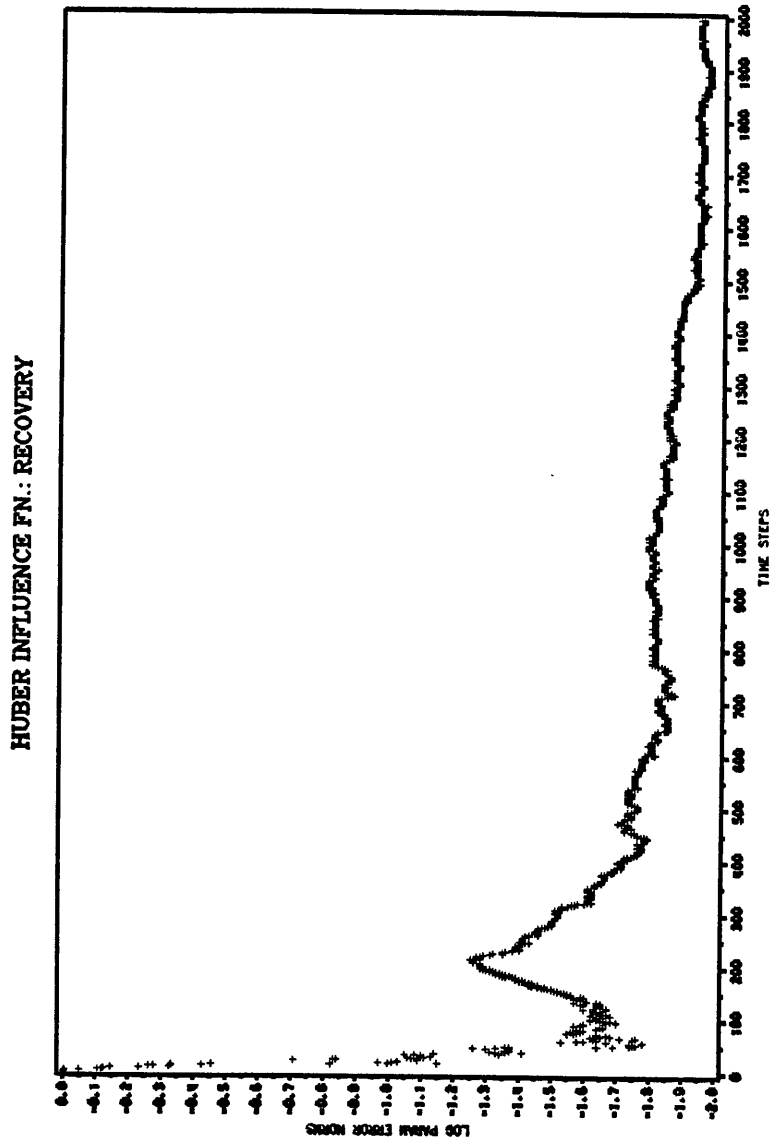

Figure 47. Hampel robust PLID.: Logarithmic error norm. Recovery from outlier burst.

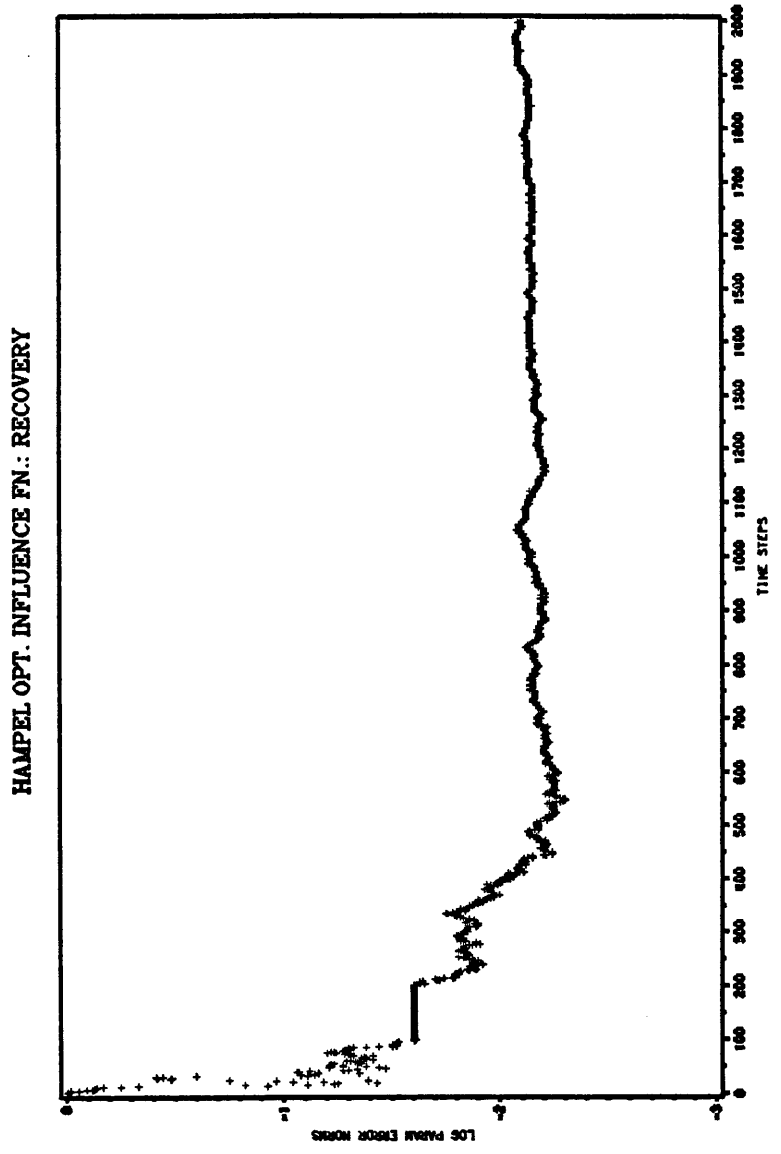

Figure 48. Reject robust PLID.: Logarithmic error norm. Recovery from outlier burst.

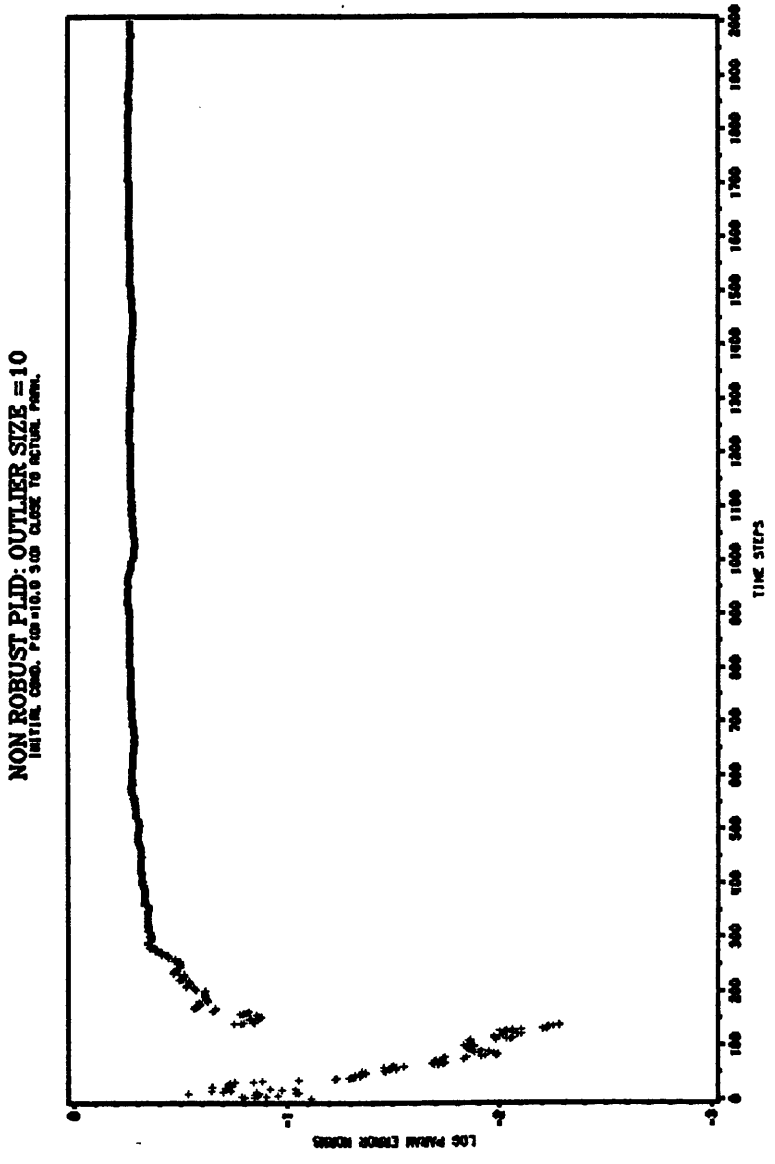

Figure 49. Non robust PLID.: Logarithmic error norm. Small Outlier.

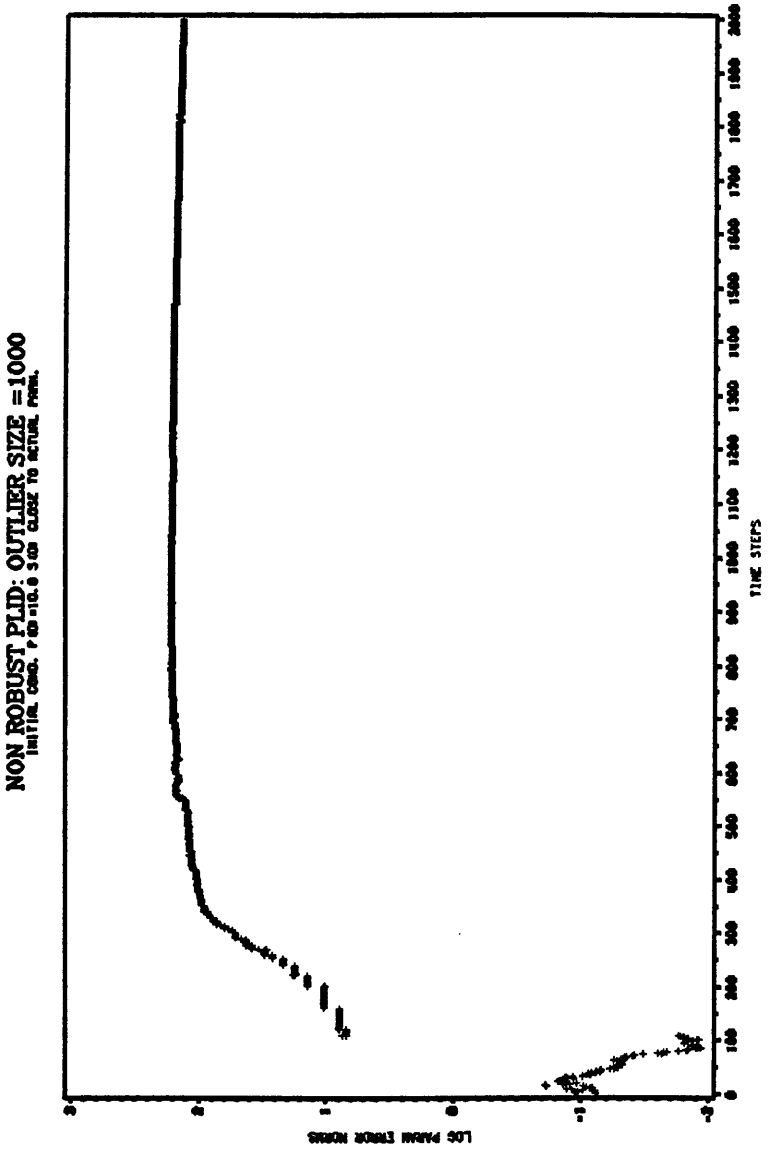

Figure 50. Non robust PLID.: Logarithmic error norm. Large Outlier.

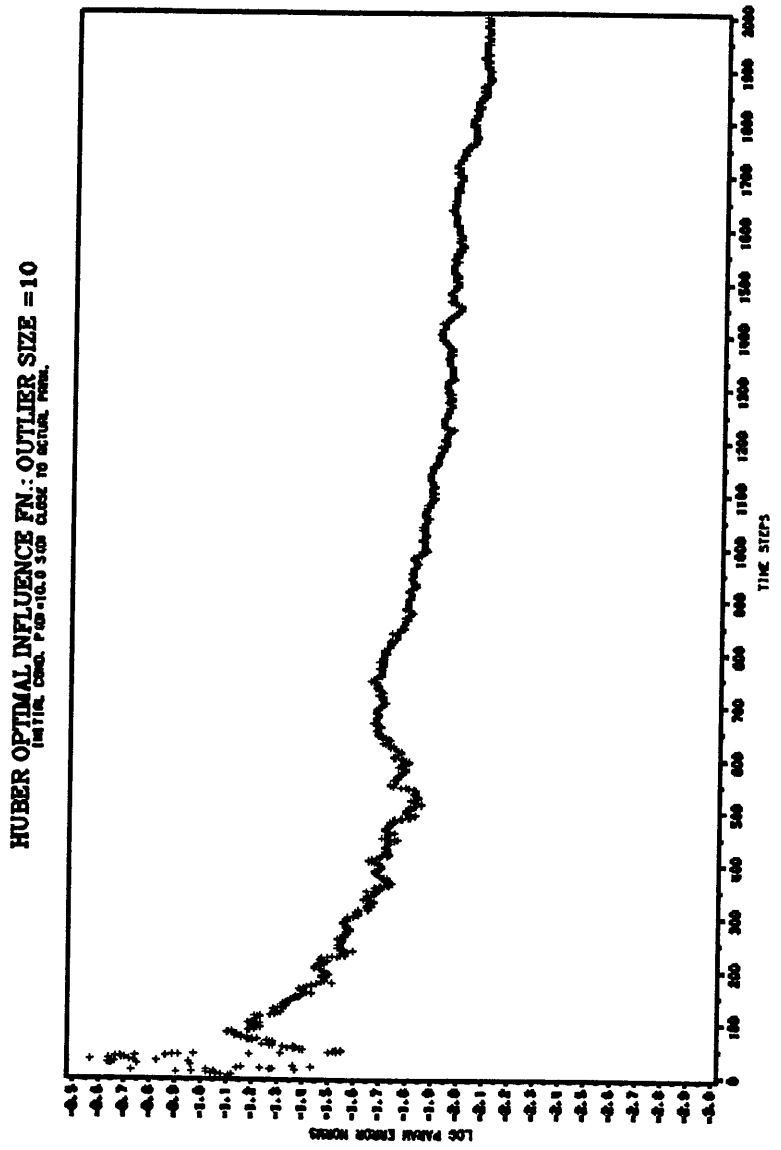

Figure 51. Huber robust PLID.: Logarithmic error norm. Small Outlier.

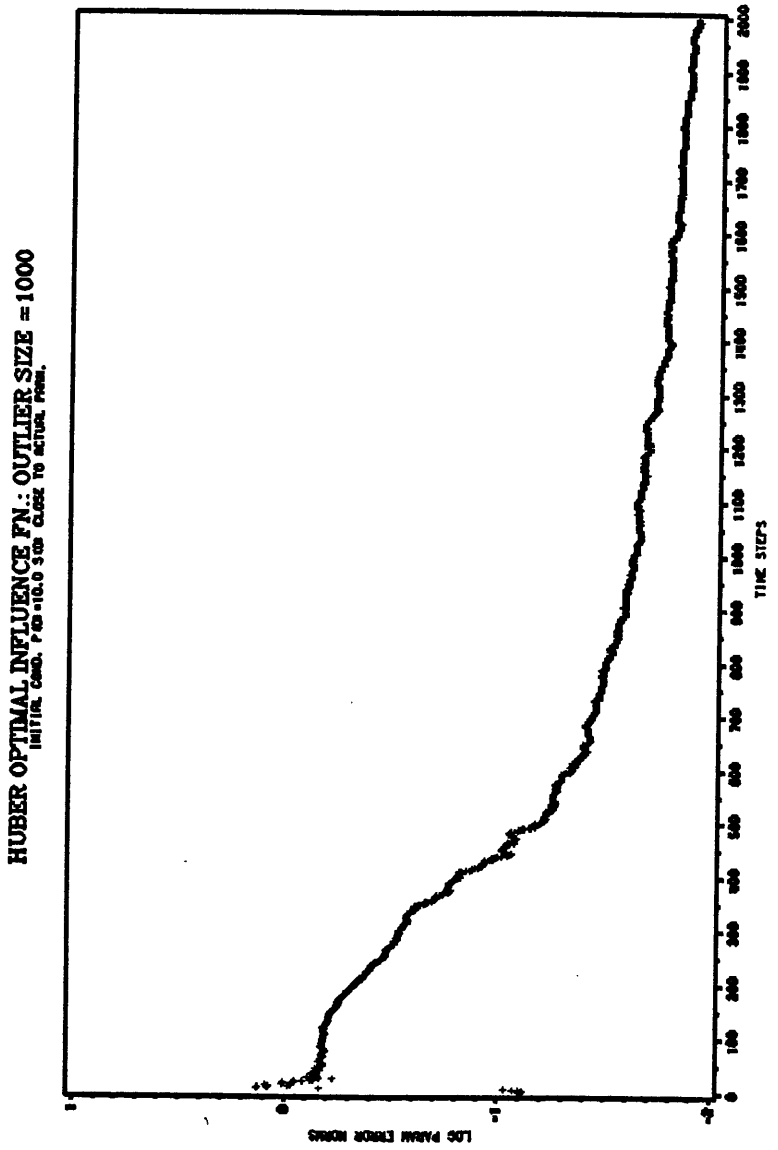

Figure 52. Huber robust PLID.: Logarithmic error norm. Large Outlier.

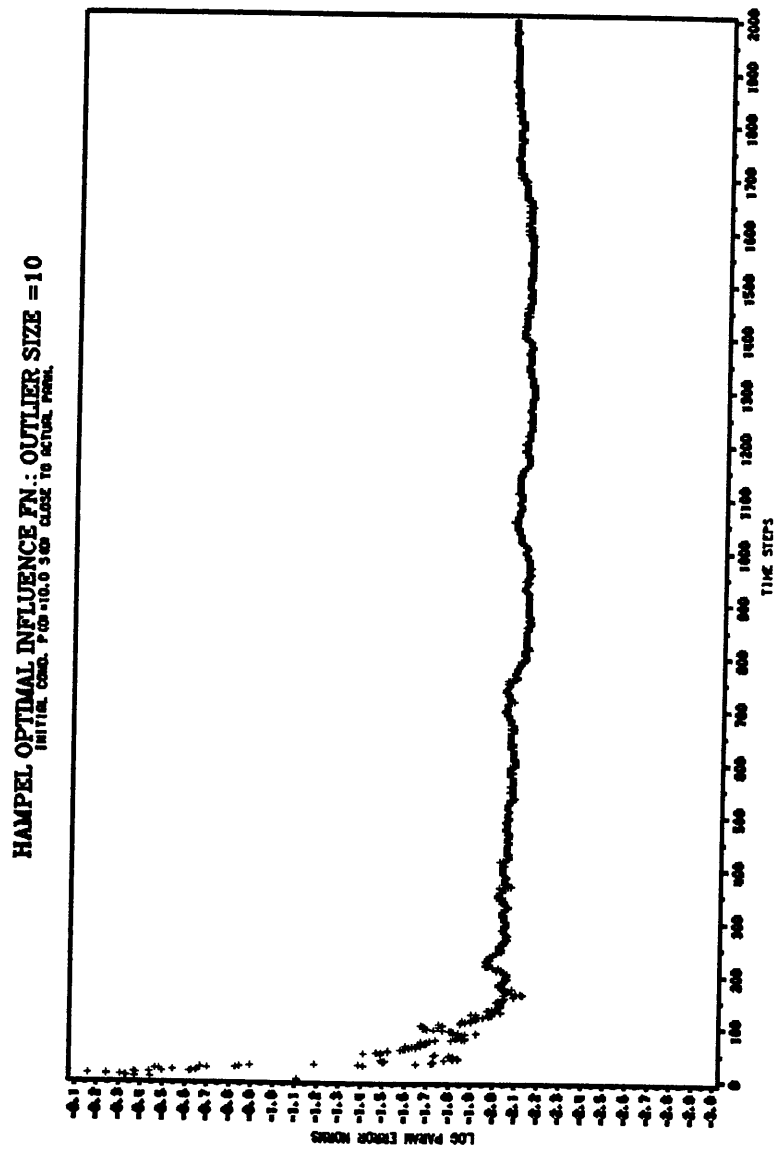

Figure 53. Hampel robust PLID.: Logarithmic error norm. Small Outlier.

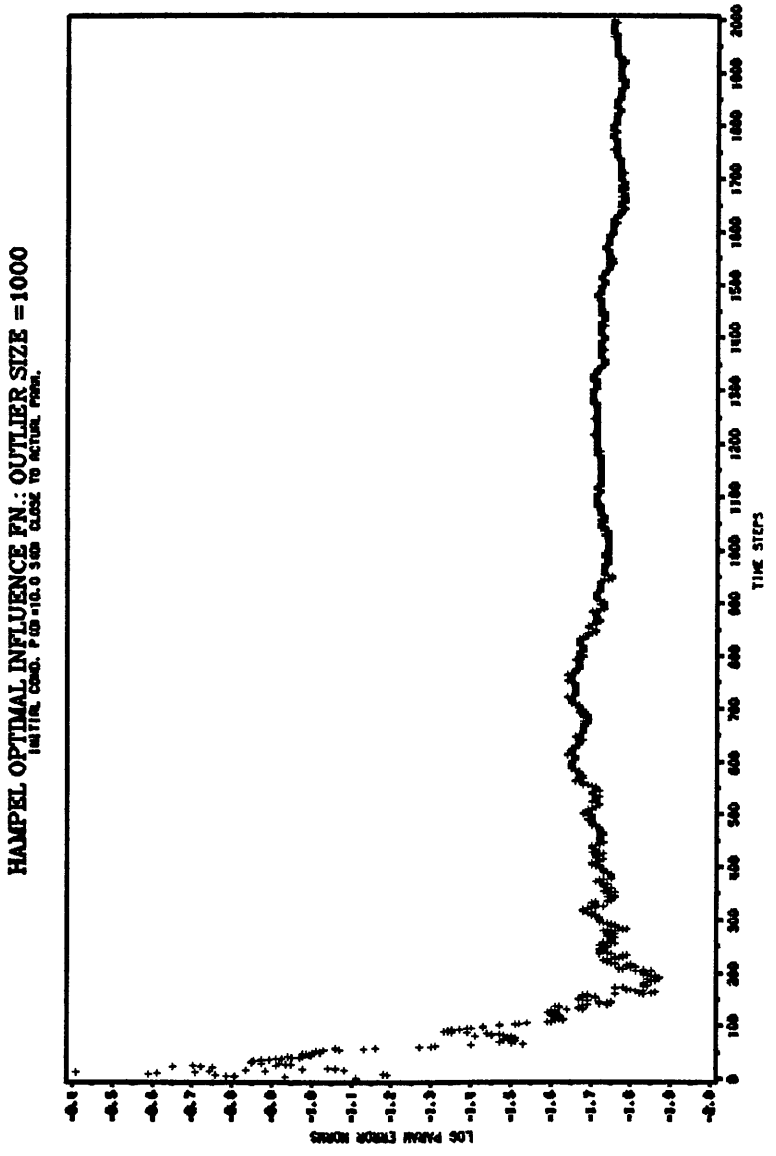

Figure 54. Hampelrobust PLID.: Logarithmic error norm. Large Outlier.

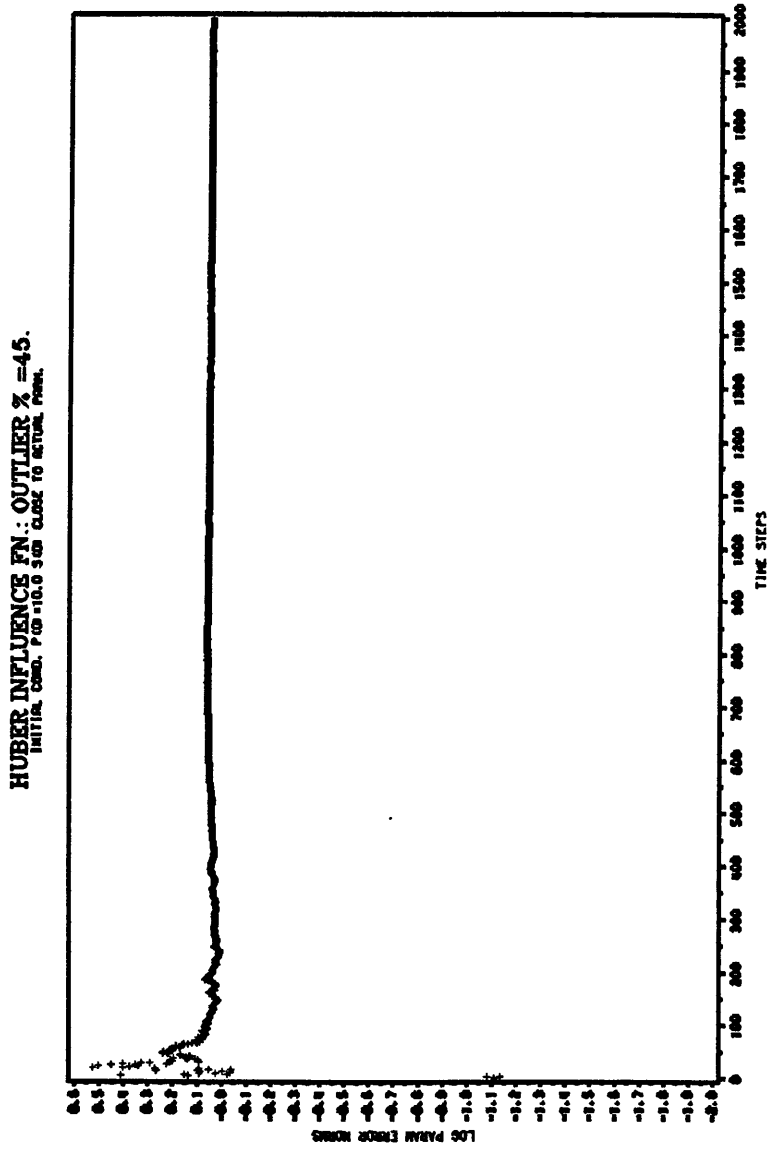

Figure 55. Huber robust PLID.: Logarithmic error norm. Outlier % = 45%.

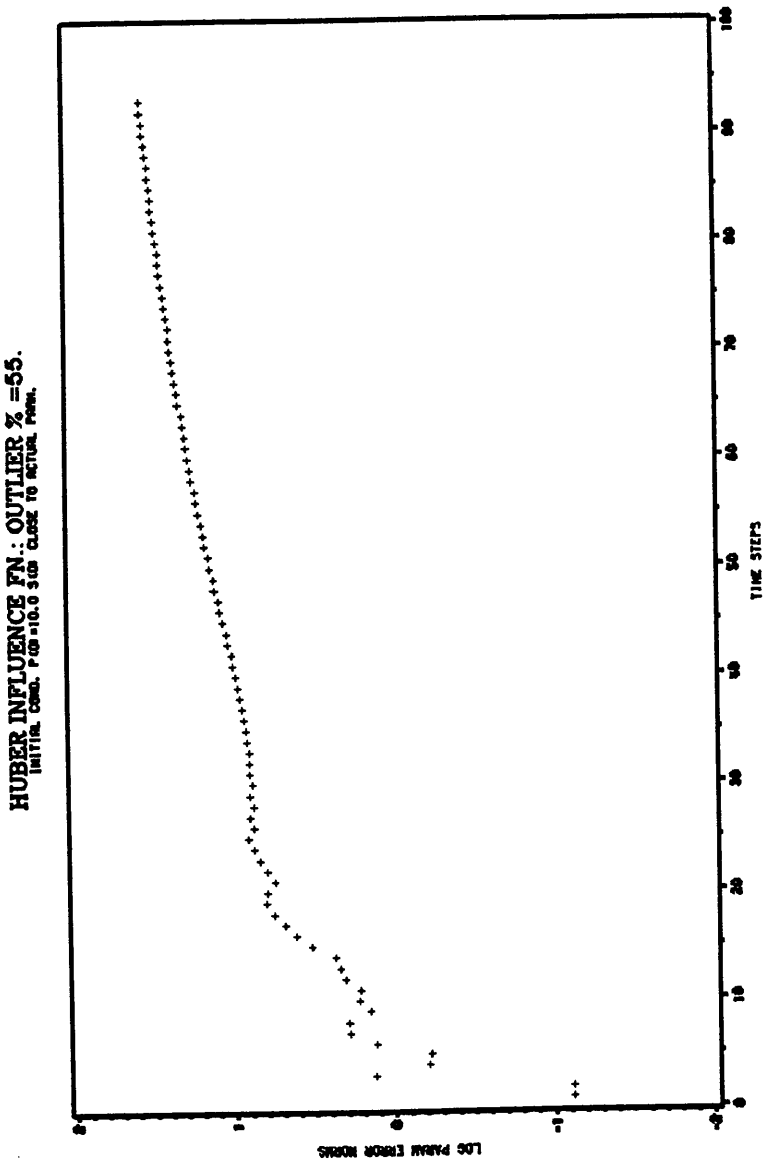

Figure 56. Huber robust PLID.: Logarithmic error norm. Outlier % = 55%.

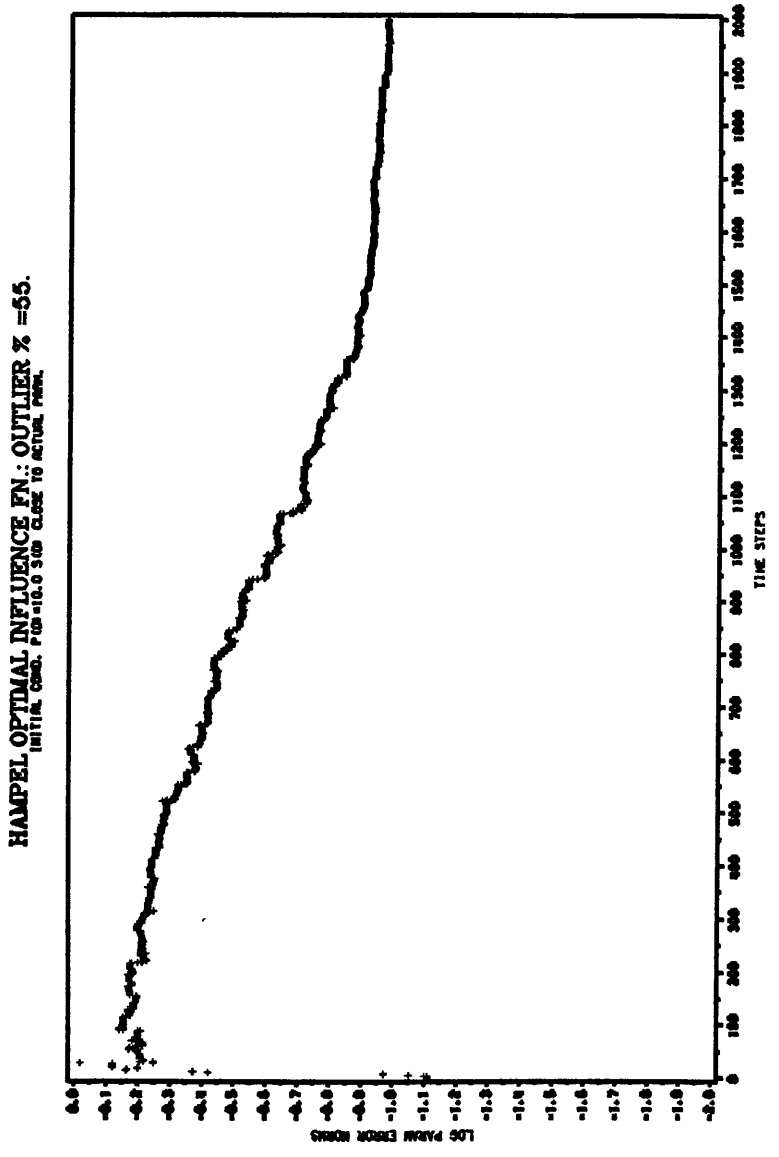

Figure 57. Hampel robust PLID.: Logarithmic error norm. Outlier % = 45%.

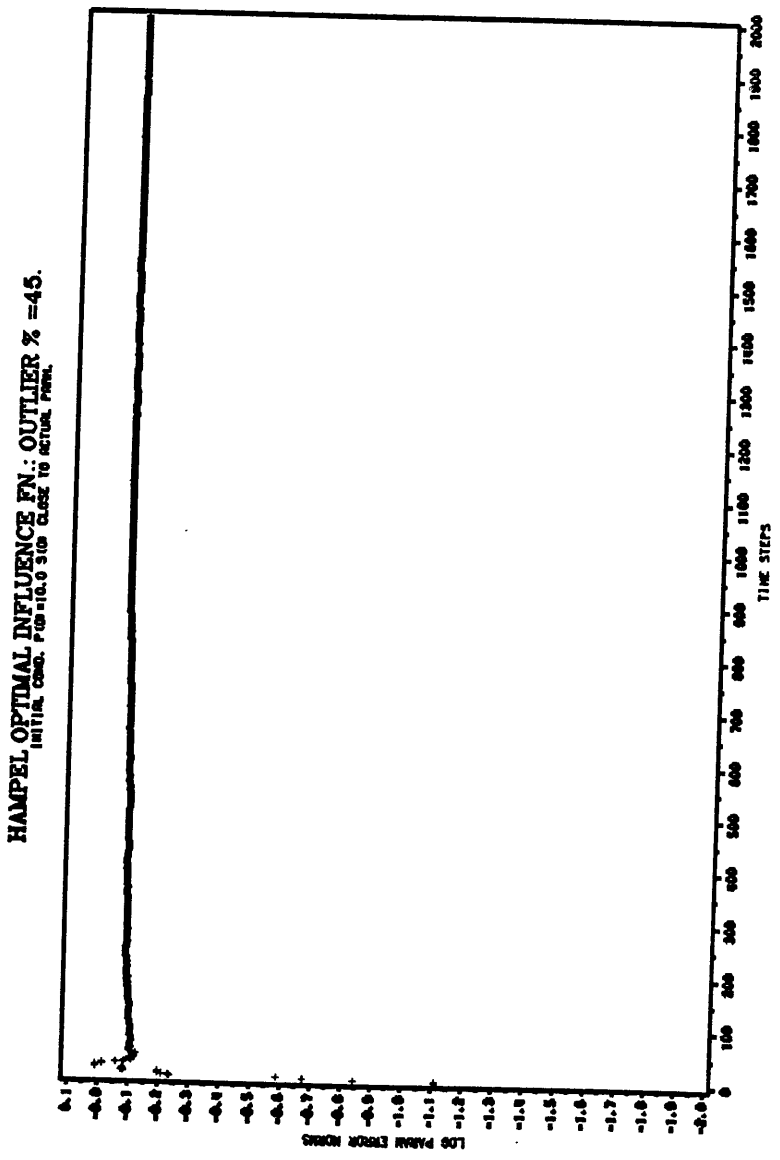

Figure 58. Hampel robust PLID.: Logarithmic error norm. Outlier % = 55%.

The vita has been removed from the scanned document