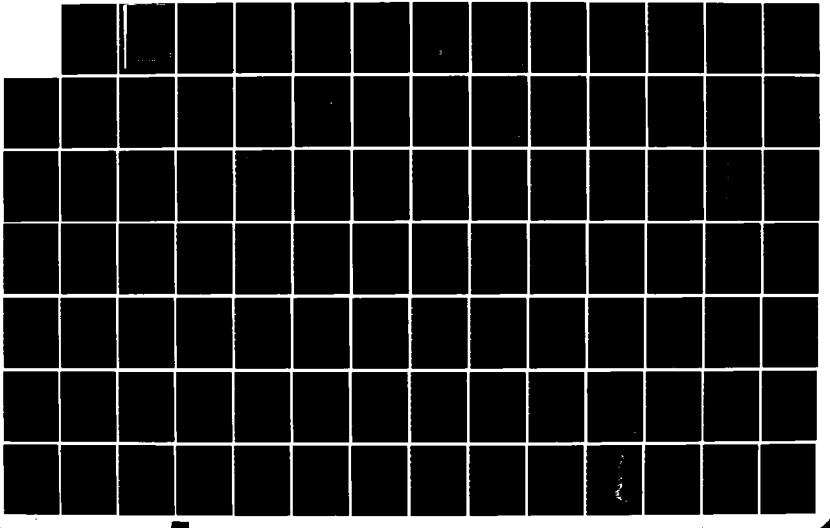
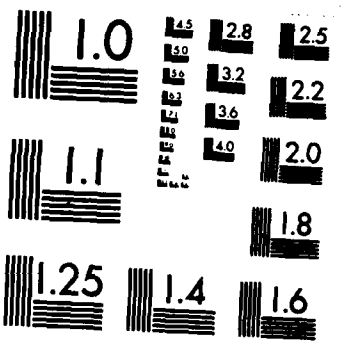
AD-A154 660 APPLICATIONS OF ADAPTIVE LEARNING CONTROLLER TO SYNTHETIC APERTURE RADAR..(U) ENVIRONMENTAL RESEARCH INST OF MICHIGAN ANN ARBOR RADAR DIV

1/2

UNCLASSIFIED D T POLITIS ET AL. FEB 85 ERIM-163800-4-F F/G 17/9 NL





(2)

Final Technical Report

APPLICATIONS OF ADAPTIVE LEARNING CONTROLLER TO SYNTHETIC APERTURE RADAR

DEMETRIOS T. POLITIS
WILLIAM H. LICATA
Radar Division

FEBRUARY 1985

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D.C. 20332
AFOSR Contract Number F49620-82-C-0097

ENVIRONMENTAL
RESEARCH INSTITUTE OF MICHIGAN
BOX 8618 • ANN ARBOR • MICHIGAN 48107

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A154660

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS:
2a. SECURITY CLASSIFICATION AUTHORITY:	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): 163800-4-F	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 85-0455
6a. NAME OF PERFORMING ORGANIZATION: Environmental Research Institute of Michigan - Radar Division; 6b. OFFICE SYMBOL (If applicable):	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): P.O. Box 8618, Ann Arbor, Michigan 48107	7b. ADDRESS (City, State and ZIP Code): Bolling Air Force Base, Washington, D.C. 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Air Force Office of Scientific Research; 8b. OFFICE SYMBOL (If applicable): NL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F49620-82-C-0097
8c. ADDRESS (City, State and ZIP Code): Bolling Air Force Base, Washington, D.C. 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2312	A1	

11. TITLE (Include Security Classification): Applications of Adaptive Learning Controller to Synthetic Aperture Radar

12. PERSONAL AUTHOR(S): Demetrios T. Politis and William H. Licata

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final Technical Report	FROM 10/10/83 TO 12/9/84	February 1985	134

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Adaptive control, aritificial intelligence, synthetic aperture radar, autofocus, learning networks

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this study the application of Artificial Intelligence methods to Synthetic Aperture Radars (SARs) is investigated. It was shown that the neuron-like Adaptive Learning Controller (ALC), operating in the extremizing mode suggested by Klopf, can be used successfully in the motion compensation system of a SAR.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☑ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Berry	(202) 767-5021	NL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES
(Concluded)

LIST OF TABLES

1
INTRODUCTION

The work described in this report is a continuation of the effort initiated two years ago for the purpose of determining whether a special class of neuron-like adaptive learning controllers developed by Barto [1] which operate in the extremizing mode suggested by Klopf [2] could find some engineering application.

The motivation for this work came from our belief that if physical systems operating in this mode could be built, they may possess significant performance advantages over other conventional systems. Ultimately, our goal and hope was to demonstrate conclusively the practical usefulness of these adaptive learning networks by developing some specific engineering application. Since our expertise and interest are in synthetic aperture radars (SAR), the application naturally would be in that area.

The application did not have to be unique, that is, it did not have to be something which only these adaptive networks could perform. It would be sufficient at this stage to show simply that a useful task could be performed. Whether these devices could do better than other conventional systems/techniques, what would have to be done to improve them, and how one would design them so that superior performance would be obtained did not concern us at this time.

In the first year of this study it was decided that the most promising learning network Barto developed that demonstrates problem solving/control capability is the Adaptive Learning Controller (ALC) learning network [3], which is described in some detail in the next section. Briefly stated, in this system two elements are used to implement a learning strategy as follows. One element, termed the Associative Search Element (ASE), constructs associations between

the input and output by searching under the influence of reinforcement feedback. A second element, the Adaptive Critic Element (ACE), constructs a more informative evaluation function than reinforcement feedback can provide, thus improving the performance of the ASE when operating alone. Both of these neuron-like adaptive elements, which constitute the controller of the learning network, were suggested by the work of Klopf.

The most attractive features of this ALC are:

a. Knowledge of system dynamics, i.e., a mathematical model of the system, is not necessary in order to develop a control law. The controller learns to develop it by association of input and output;

b. The system to be controlled can be time varying and/or non-linear;

c. A wide class of measures of performance can be optimized;

d. Only crude measures of the system state are required for feedback;

e. A non-uniform sampling rate can be used, and

f. The algorithms are naturally adaptive. They can be used to directly control the system or to optimize the performance of an existing control system.

The ALC algorithm was implemented at ERIM and tested on the pole-on-cart problem also used by Barto. In this example, the ALC is required to learn to balance a pole which is pivoted on top of a cart, which is free to move along a straight line path by applying a constant force, ± F, through the system's center of gravity. Our results substantiated that the ALC controller after a few trials can indeed learn to keep the pole balanced for as long as desirable.

A study of the ALC algorithm revealed that it has potential application to a class of control problems that can be studied from the perspective of minimizing some performance function. Several possible applications in SAR were identified for investigation. One of these is the problem of cancellation of motion-induced quadratic phase errors, which is discussed in detail in Section 3.

Phase errors in SAR phase histories result in blurred or defocused imagery, with degraded resolution and/or signal-to-noise ratio. The primary cause of these phase errors is inaccuracy in measuring the motion of the antenna phase center over the processing interval.

A Taylor series expansion of the phase error about the center of the processing aperture shows that the second-order term of the series tends to dominate and most phase error correction algorithms are concerned with the estimation and correction of the quadratic term. Image refocusing through quadratic phase error correction is commonly referred to as autofocus processing.

To apply autofocus, however, aircraft velocity at imaging initiation must be accurately known. Errors in initial aircraft velocity values will cause proportionately large quadratic phase errors, which may render autofocus impossible.

At the present time, a two-beacon plus altimeter scheme is used at ERIM to provide these initial values. Though the system has been working satisfactorily, this is a crucial link in good SAR imaging and any improvements in the beacon update procedure would be considered a significant contribution to SAR imaging.

Section 2 provides a brief description of the operation and implementation of the ALC at ERIM. In Section 3, the SAR Autofocus and Motion Compensation Problem are outlined.

In Section 4, the application of the ALC to SAR imaging is considered. It is shown that the ALC adaptive controller can indeed be

used to provide the correct initial aircraft velocity, hence reduce motion-induced quadratic phase errors. This technique, which will be referred to here as ALC motion correction, is automatic and can be done in real time. These are distinct and desirable characteristics which make this technique very attractive and worthy of further exploration and development.

Section 5 describes the study performed and the results obtained at ERIM related to the process of ALC design, in general, and design optimization, in particular.

2
DESCRIPTION OF OPERATION AND IMPLEMENTATION OF THE ALC AT ERIM

Figure 1 shows a system with transfer function G controlled by the ALC learning net. The state vector $\underline{s}$ of the system is sampled at intervals T sec. and is fed into a decoder which is used to discretize the state space of $\underline{s}$ into a finite number of states, thus converting $\underline{s}$ into a binary vector $\underline{X}$, whose components are all zero except the one corresponding to the state of the system at the sampling instant $t_k = kT$. The dimension of $\underline{X}$ is equal to the number chosen for the discrete states of the space of $\underline{s}$.

The vector $\underline{X}$ is fed into the ALC. At the Adaptive Critic Element, its adaptive weighting vector $\underline{v}$, the input vector $\underline{X}$ and the external reinforcement function r(t), are used to generate the internal reinforcement function $\hat{r}(t)$ that is input to the ASE, in accordance with the rule:

$$\hat{r}(k) = r(k) + \gamma p(k) - p(k-1)$$

where

$$p(k) = \sum v_i x_i$$

γ is a non-negative constant less than or equal to one, and the weighting vector $\underline{v}$ updates in accordance with

$$v_i(k+1) = v_i(k) + \delta \hat{r}(k) \hat{x}_i(k)$$

where $\hat{x}_i(k)$ is the value of a trace of the input variable x_i at t_k, evaluated from:

$$\hat{x}_i(k+1) = \beta \hat{x}_i(k) + (1-\beta) x_i(k)$$

and β and δ are positive constants.

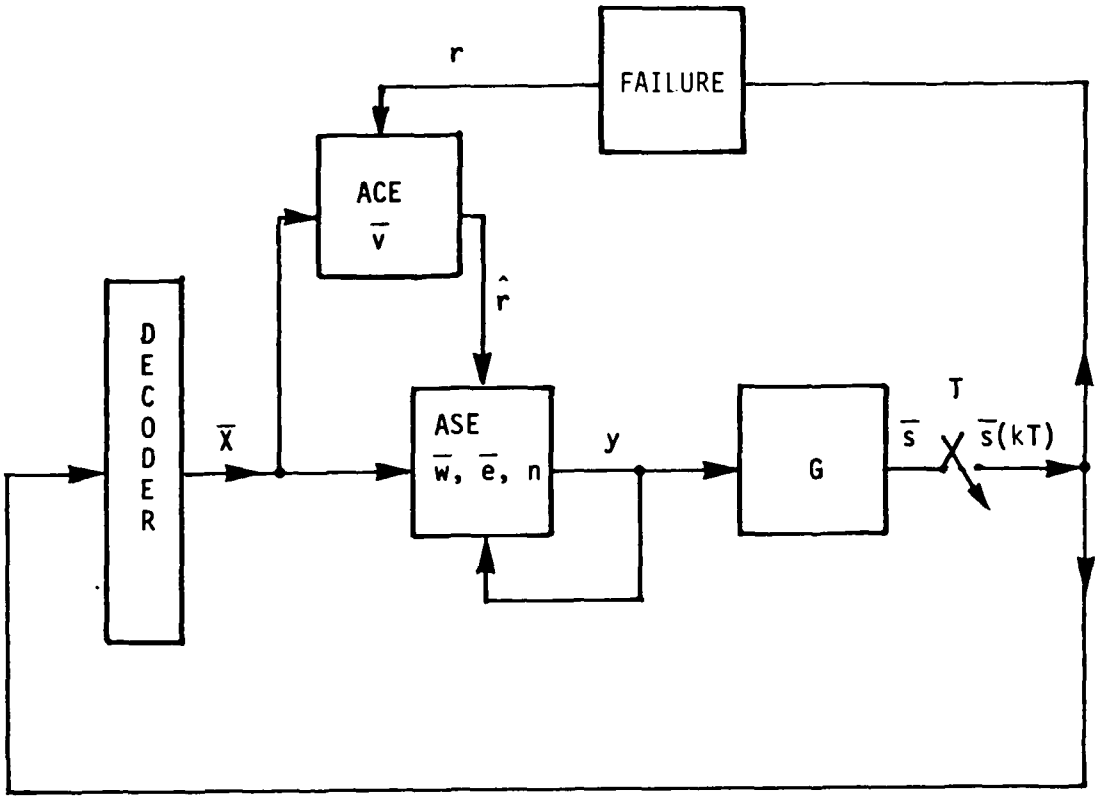


FIGURE 1. THE ASE-ACE CONTROLLER.

4.3 PROCESSING APPROACH

Figure 9 illustrates the SAR image processing chain normally used at ERIM and the modified processing chain used for the ALC application. Radar phase history and motion data are stored on High Density Tape (HDT) in the aircraft. On the ground, a portion of the motion data and radar phase history data are written on computer compatible tape which is used by the image processor to produce a SAR image. Conventional autofocus techniques would normally be applied by the image processor.

In the case of the ALC processing, the original radar phase history CCT was processed offline on a VAX. The motion data are processed by the ALC and the modified motion data, along with the original radar phase history data, are written on a new CCT. The new CCT is now processed by the image processor without using autofocus. Ideally, the ALC may reduce the quadratic motion error to the point where autofocus is not required, but at a minimum will reduce the autofocus pull-in range requirements to a tolerable level.

4.4 ALC VELOCITY BIAS CORRECTION BLOCK DIAGRAM

Figure 10 indicates in block diagram form the processing conducted with the ALC to solve the task defined in Figure 8. The measurement data are $(Y - Y_0)$, where Y_0 is the first measured value of Y. To these data a correction is applied in the form of Y_c. The correction is obtained by integrating the estimate of the velocity bias in $(Y - Y_0)$, $\dot{Y}_c$. The corrected Y motion measurement, $(Y - Y_0 - Y_c)$, feeds two threshold tests. One threshold counts the number of times $(Y - Y_0 - Y_c)$ crosses the upper bound and the other threshold test counts the number of times $(Y - Y_0 - Y_c)$ crosses the lower bound. The variable X is the difference between the number of positive and the number of negative crossings. The correction produced by the ALC is applied through a gain to the

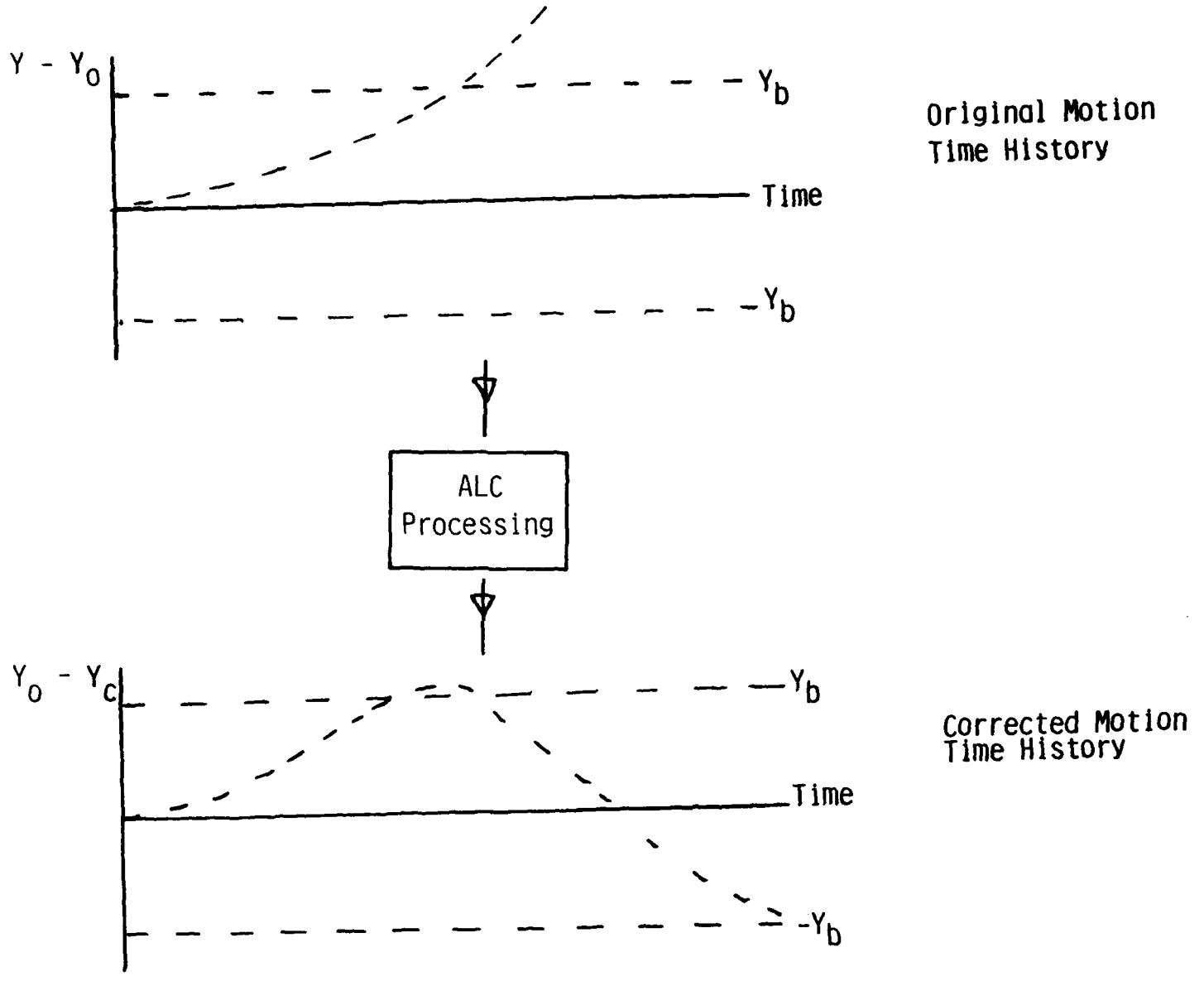


FIGURE 8. ALC TASK DESCRIPTION.

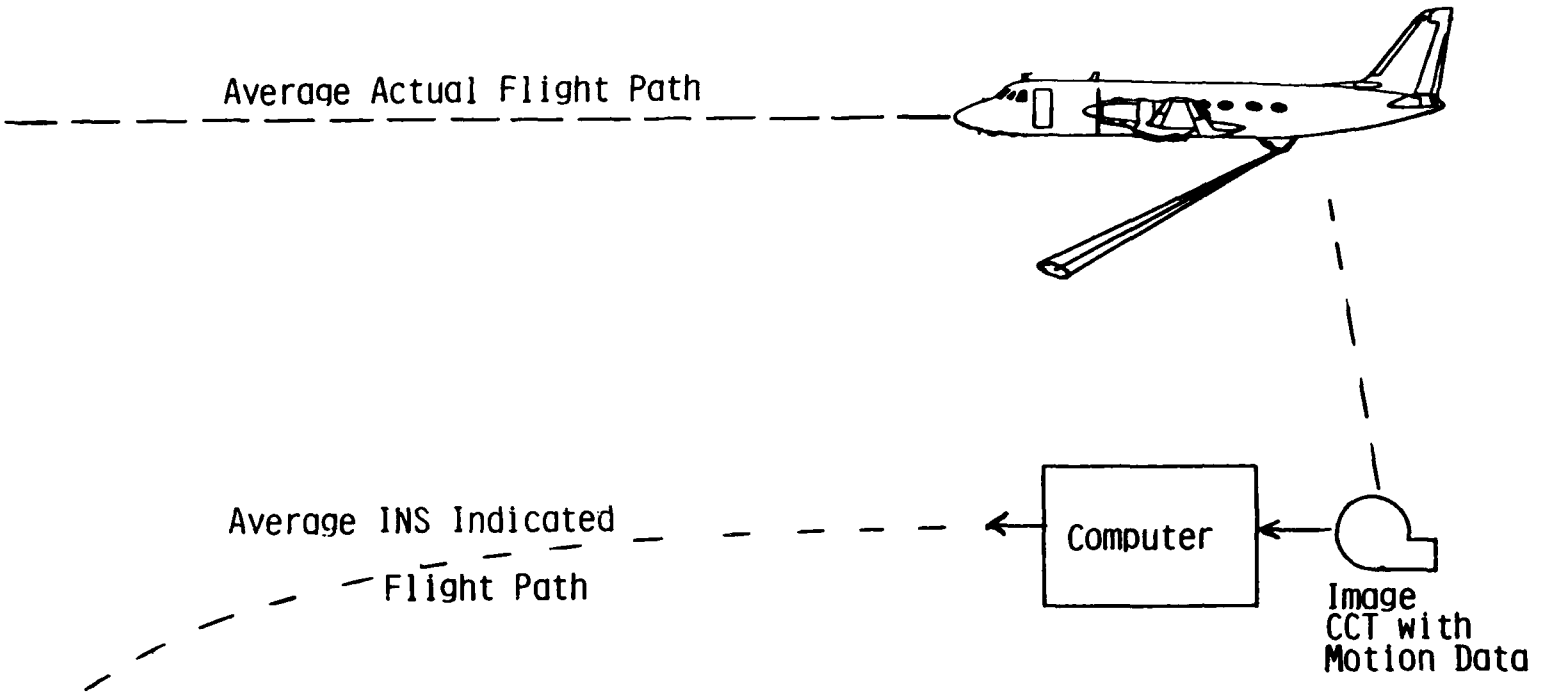


FIGURE 7. FLIGHT PATH RECONSTRUCTION PROBLEM.

As was previously stated, the main source of quadratic (i.e., low frequency) phase error in the SAR signal is due to the motion sensing inertial navigation system (INS) velocity bias. Therefore, the ALC application was concentrated in sensing and correcting the velocity bias. Further, to simplify the problem in the initial research effort, only motion errors due to cross-track velocity bias were considered. This reduced the problem to one dimension, but since the three-dimensional problem can be studied as three one-dimensional problems, the simplification is justified.

Figure 7 illustrates the application of the ALC to the problem of cancelling the lateral (toward right or left wing) or cross track INS velocity bias error. The task assigned to the ALC is to keep the position measurements along any aircraft axis within the expected bounds over the aperture time of the radar. When beacon data are available, zero cross-track velocity estimates will be determined from these data. If beacon data are not available, the assumption is made that during imaging, the aircraft flies on the average a straight line path through space and any deviations from this path are at frequencies high compared to the inverse of the aperture time. The aircraft motion data, recorded on computer compatible tape, when plotted indicate that the aircraft was moving off the desired flight line and had a nonzero average cross-track velocity. This apparent average cross-track velocity is not a true velocity but a bias, and if not cancelled will cause a quadratic phase error.

Figure 8 illustrates the task that the ALC must learn to perform. The actual Y position of the aircraft as a function of time is known to be bounded by the limits $\pm Y_b$. When the recorded lateral aircraft motion is plotted as a function of time, it quickly exceeds these bounds. Thus, the ALC must apply a position correction to the measured motion data to keep it within the defined bounds. These corrected motion data are in turn used to correct the radar phase data instead of the original motion data.

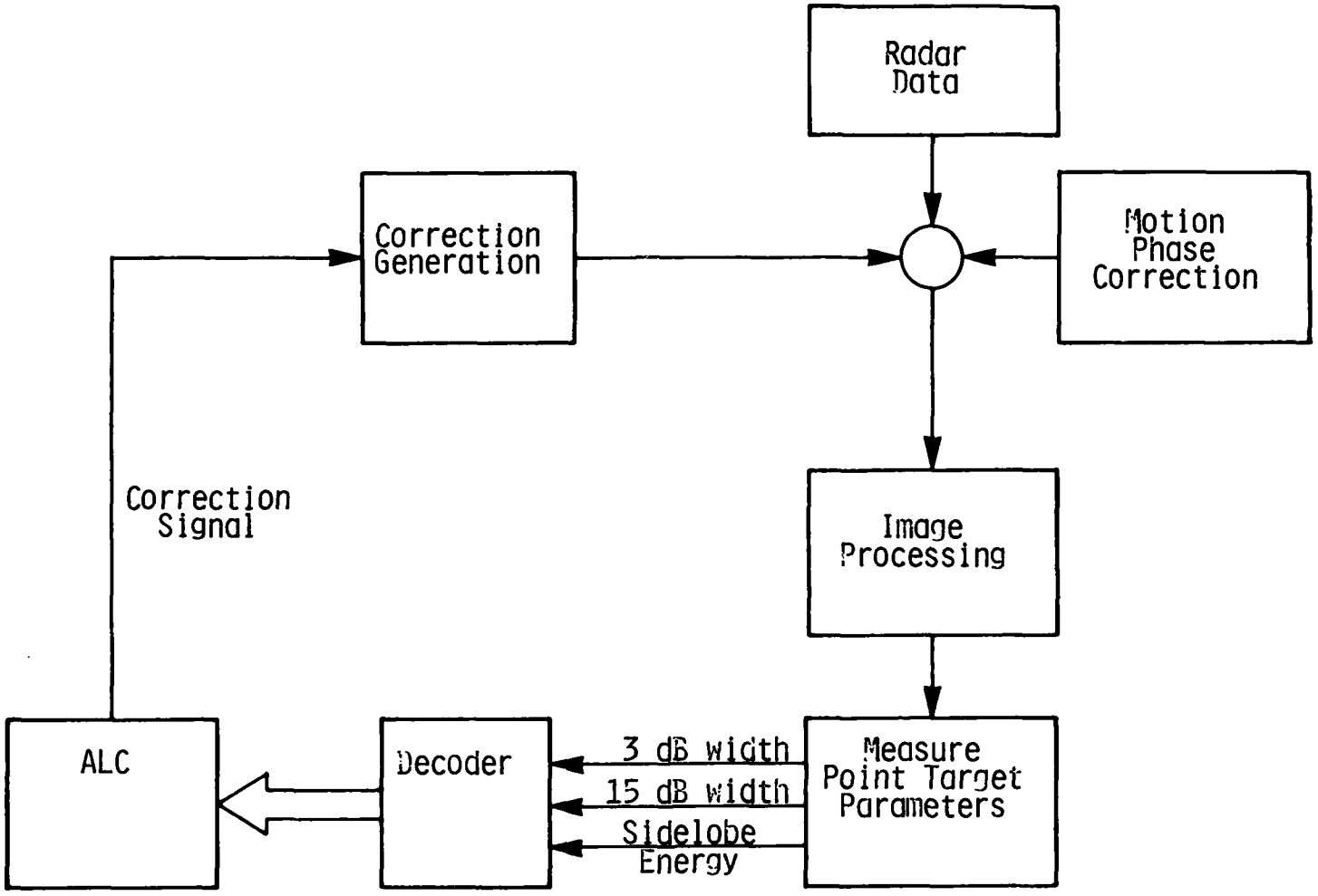


Figure 6. Application of ALC to Autofocus

4
ALC APPLICATION TO SAR IMAGING

4.1 APPLICATION OF ALC TO AUTOFOCUS

The method of applying the ALC to the autofocus problem, which addresses the low frequency phase error correction problem in the same way as present digital autofocus techniques, is illustrated in Figure 6. Combining radar data and motion compensation phase corrections, a radar image can be produced and measurements of 3 dB width, 15 dB width and sidelobe energy can be made on a point target in the scene. These IPR measurements go to the ALC decoder and the ALC decides on the direction of the proper correction. Based on the ALC decision, an autofocus correction is generated so a new image can be produced. This process is repeated until a satisfactory image is obtained.

This approach to applying the ALC to the autofocus problem was not pursued because the computer time associated with producing an image is long and it was believed that the computer costs associated with testing out this approach would be prohibitively high. A second approach which did not require use of the image processing facility was selected in place of this approach.

4.2 MOTION CORRECTION VIA CORRECTION OF VELOCITY BIAS

In this section, we describe the work performed at ERIM in implementing an ALC velocity bias correction system. It was shown that the ALC can be used to reduce motion-induced quadratic phase errors. Cancellation of motion-induced quadratic phase errors may render autofocus correction unnecessary, or at a minimum so reduce the errors as to ensure the success of existing autofocus technology.

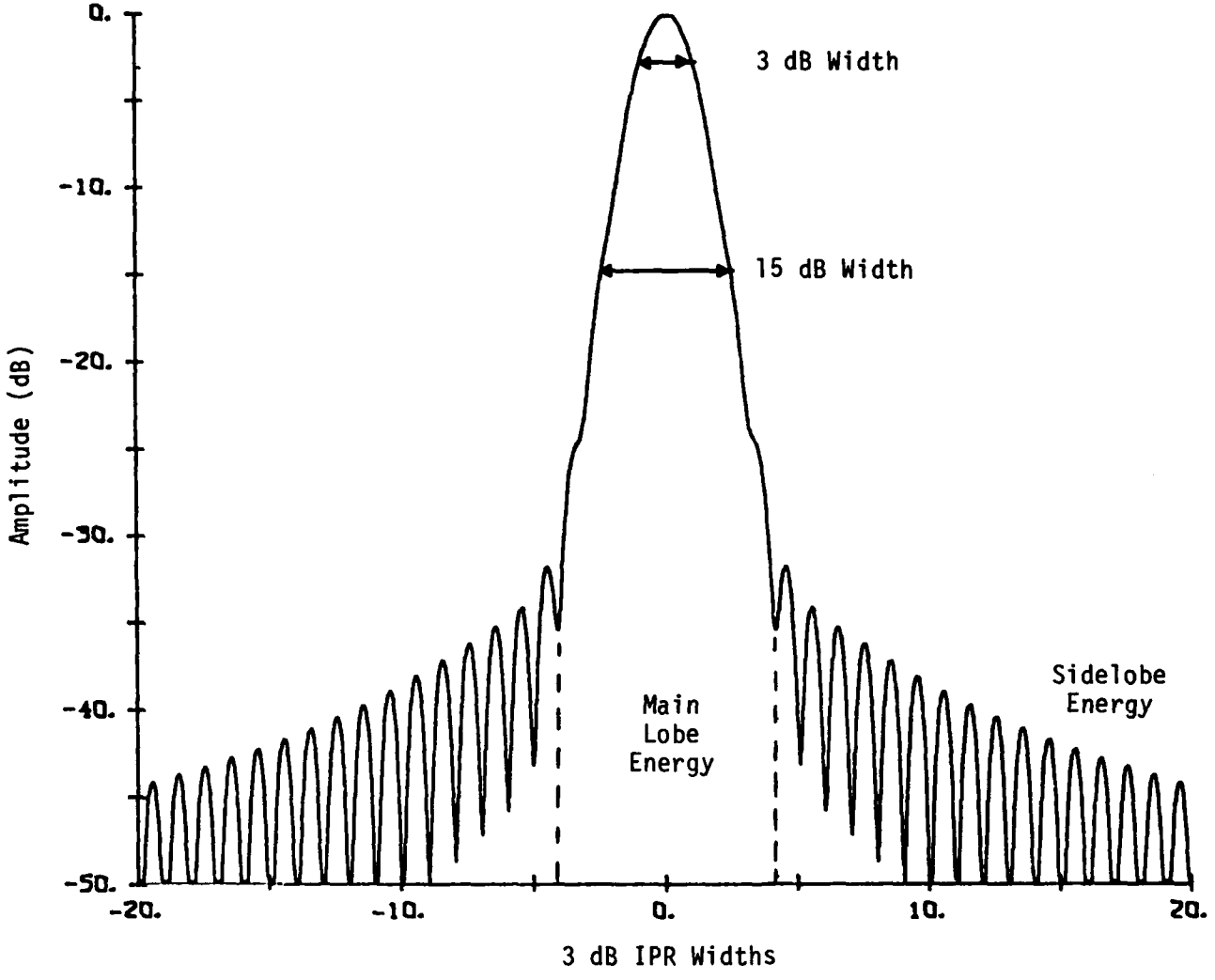


FIGURE 5. THE IPR FUNCTION.

line would look something like Figure 5. The peak of the intensity would be where the physical target is located and then it would roll off to some small value which is eventually lost in the surrounding clutter intensity. Figure 5 is called the azimuth impulse response (IPR) function and is specified by such things as the 3 dB width, 15 dB width and the amount of energy in the sidelobes compared to the amount of energy in the mainlobe. An analogous situation exists in the orthogonal (range or line of sight) direction but in this case high resolution is obtained on a single pulse using a narrow transmitted pulse or pulse compression technique.

Any errors in measuring the motion of the aircraft off the nominal flight path will cause the IPR function illustrated in Figure 5 to degrade. The type of degradation can be related to the frequency of the motion error (i.e., how rapidly the motion error changes with time). Slowly changing errors, such as errors quadratic with time, primarily cause the 3 dB width and 15 dB width to spread. Higher frequency errors primarily cause an increase in the sidelobe energy at the expense of the mainlobe energy. When SAR image processing was done optically, the broadening of the 3 dB width could be corrected by moving a lens in the optical processing chain. When digital image processors were developed, a technique called autofocus was developed to correct out the low frequency phase errors.

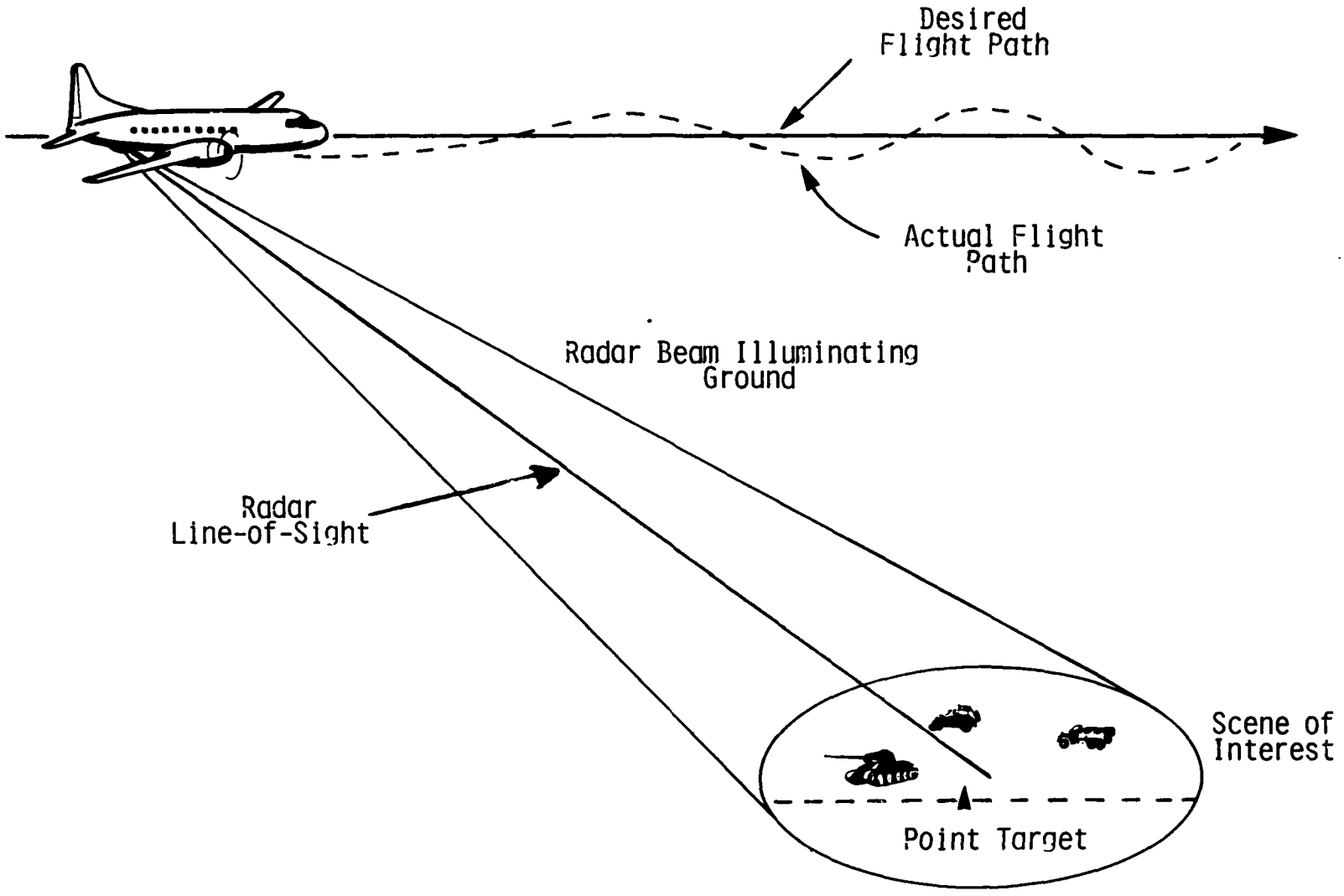


Figure 4. Motion Compensation Problem

corrects the measured phase history so the image focuses properly. Current autofocus techniques, however, do not work very well for certain kinds of imaged scenes. A detailed description of current autofocus methods is given in Appendix A.

SAR MOTION COMPENSATION PROBLEM

The SAR motion compensation problem is illustrated in Figure 4, which shows an aircraft trying to fly a straight line course while illuminating some area of interest with a synthetic aperture radar (SAR). The radar is a pulsed radar and every point in space, where the radar transmits and receives a pulse, is a point in a synthetically-long phased array. If these points form a straight line in space and the pulses are added coherently, a synthetic array is formed in space with a beamwidth much narrower than the radar physical beamwidth, which permits accurate azimuth resolution of closely spaced targets in the scene of interest. In reality, the aircraft does not fly in a straight line but moves about the nominal flight path as it is tossed about by atmospheric gusts. This means that adding the pulses coherently will form a distorted linear array and degrade resolution. Aircraft motion can, however, be corrected by shifting the phase of each pulse proportionally to how much the aircraft is off the nominal flight path, so it will have the same phase as if received on the nominal flight path so the pulses will coherently add and form the proper line array. To compensate each radar pulse for aircraft motion, the aircraft must contain motion sensing and computer equipment that can accurately compute the actual aircraft position relative to the desired flight path. This motion sensing and correcting system constitutes the motion compensation system.

A point target (i.e. an object that appears to the radar to be a single scatterer) is illustrated in Figure 4. If a line is drawn though the point target and parallel to the desired flight path, a plot of the intensity of the scene as seen by the radar along that

3
THE SAR AUTOFOCUS AND MOTION COMPENSATION PROBLEM

3.1 INTRODUCTION

As was previously mentioned, one possible practical SAR application of the ALC learning algorithm is in the limiting of motion-induced quadratic phase errors.

Good radar imaging requires fine-resolution capability (i.e., ability to separate targets which are physically in proximity with each other) both in range and azimuth. While very high range resolution can be achieved by using pulses of appropriately wide bandwidth, high azimuth resolution can be obtained only by synthetic aperture radar systems, which make use of sophisticated data processing techniques. These techniques require a coherent radar and precise and accurate knowledge of the aircraft (radar platform) flight path, which in turn can be obtained through elaborate motion compensation systems.

The purpose of the motion compensation system is to detect and provide an accurate measure of the deviations of the aircraft motion from a straight line path, so the phase of the radar echo can be referenced to the coherent video source in the transmitter. Errors in the motion measurements degrade the image quality, the degradation depending on the frequency content of the motion error relative to the inverse of the aperture time of the SAR system.

Low frequency motion compensation errors, which are quadratic with time, cause the SAR image to smear in the azimuth direction, making a point target appear as a line. This image degradation is called defocusing and in an optical SAR processor it would be corrected by moving a lens until the image came into focus. In a digital SAR processor it is corrected by a technique referred to as autofocus, which makes an estimate of the quadratic phase error and

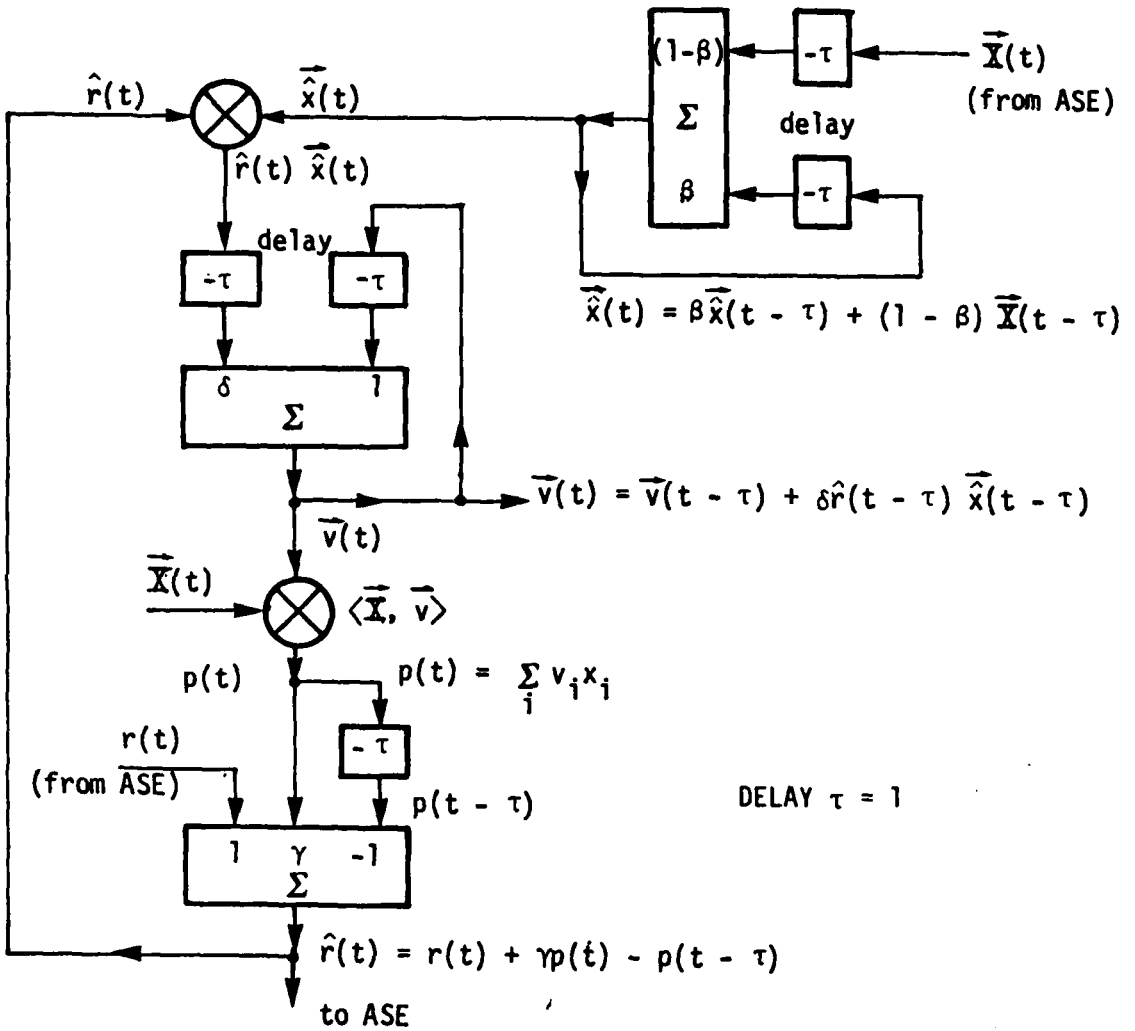


FIGURE 3. IMPLEMENTATION OF ACE.

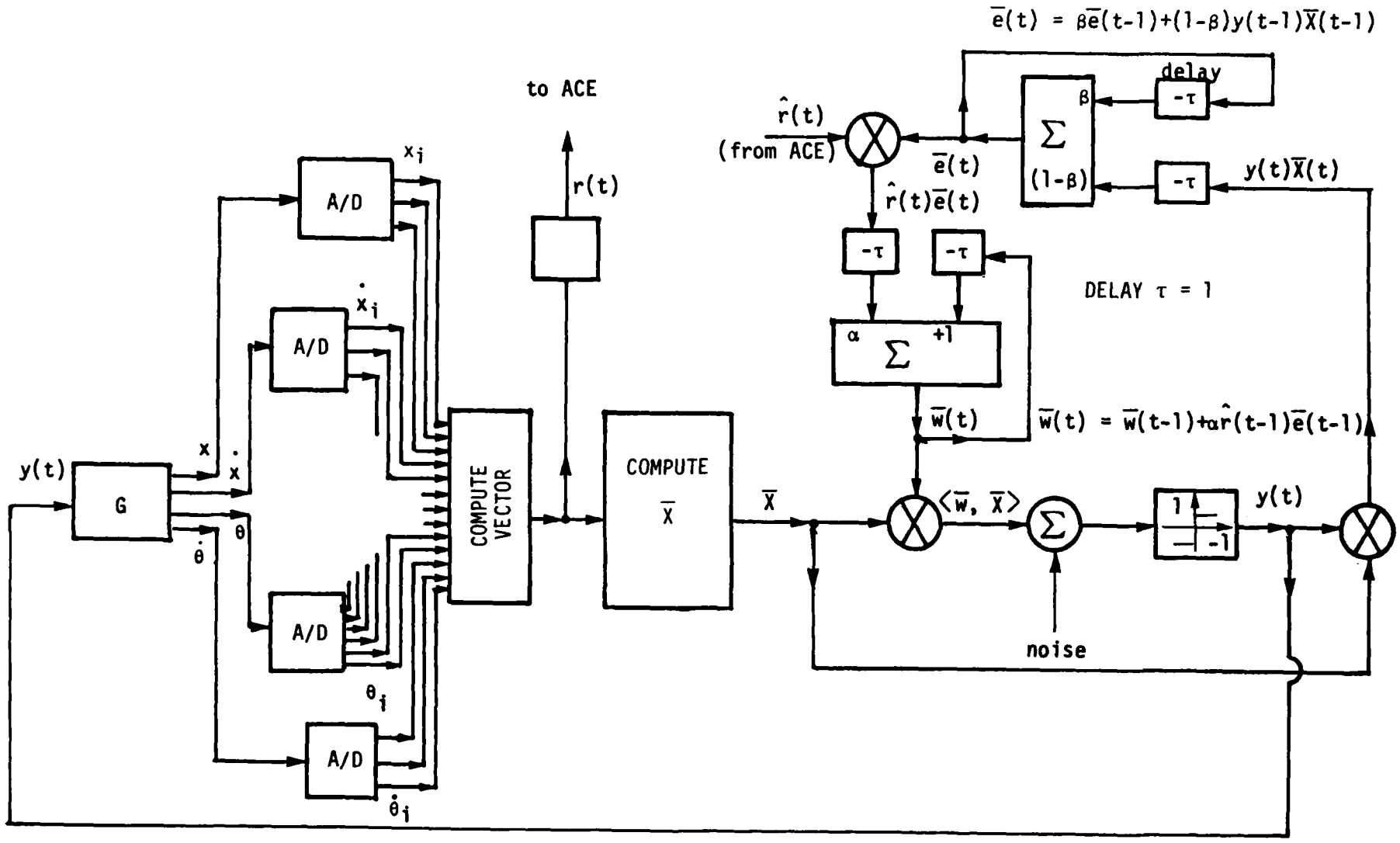


FIGURE 2. IMPLEMENTATION OF ASE.

At the Associative Search Element, the input vector $\underline{X}$ generates the output y:

$$y(k) = \pm 1$$

depending on whether $[\sum w_i x_i + n(t)]$ is non-negative or negative, respectively, where n(t) is additive system noise and the weighting vector $\underline{w}$ updates in accordance with the rule:

$$w_i(k+1) = w_i(k) + \alpha \hat{r}(k) e_i(k)$$

and the function $e_i(k)$ is the eligibility at t_k of path i, adapting in accordance with the rule:

$$e_i(k+1) = \beta e_i(k) + (1-\beta)[y(k)x_i(k)]$$

and $\alpha > 0$; $0 \leq \beta < 1$. Figures 2 and 3 show in block diagram form the implementation of the ALC algorithms at ERIM.

The way the ALC exercises control over the system G is as follows. Let us assume that we wish to maintain the values of the state variables s_j and s_k of the system within certain bounds. We use the external reinforcement variable r(t) to penalize the system when either s_j or s_k take values outside the desired range. When this happens, we will say that the system has failed and r(t) is set equal to -1. Otherwise, r(t) = 0.

With zero initial values for the system state variables and the ALC variables $\underline{w}$, $\underline{v}$, $\underline{e}$, $\underline{x}$, the system is activated and goes through a sequence of admissible states, until it finally fails, either in s_j or s_k. At that time, the system state variables and $\hat{\underline{x}}$ are reset to zero but $\underline{w}$, and $\underline{v}$ are left untouched. Thus, when the next trial for the system starts, the initial values of $\underline{w}$ and $\underline{v}$ are the final values from the previous trial. Hence, the experience, or learning of the system at time t is stored in the values $w_i(t)$ and $v_i(t)$. After a few trials, the system learns to operate without failure, i.e., learns to operate while maintaining the state variables within the desired bounds.

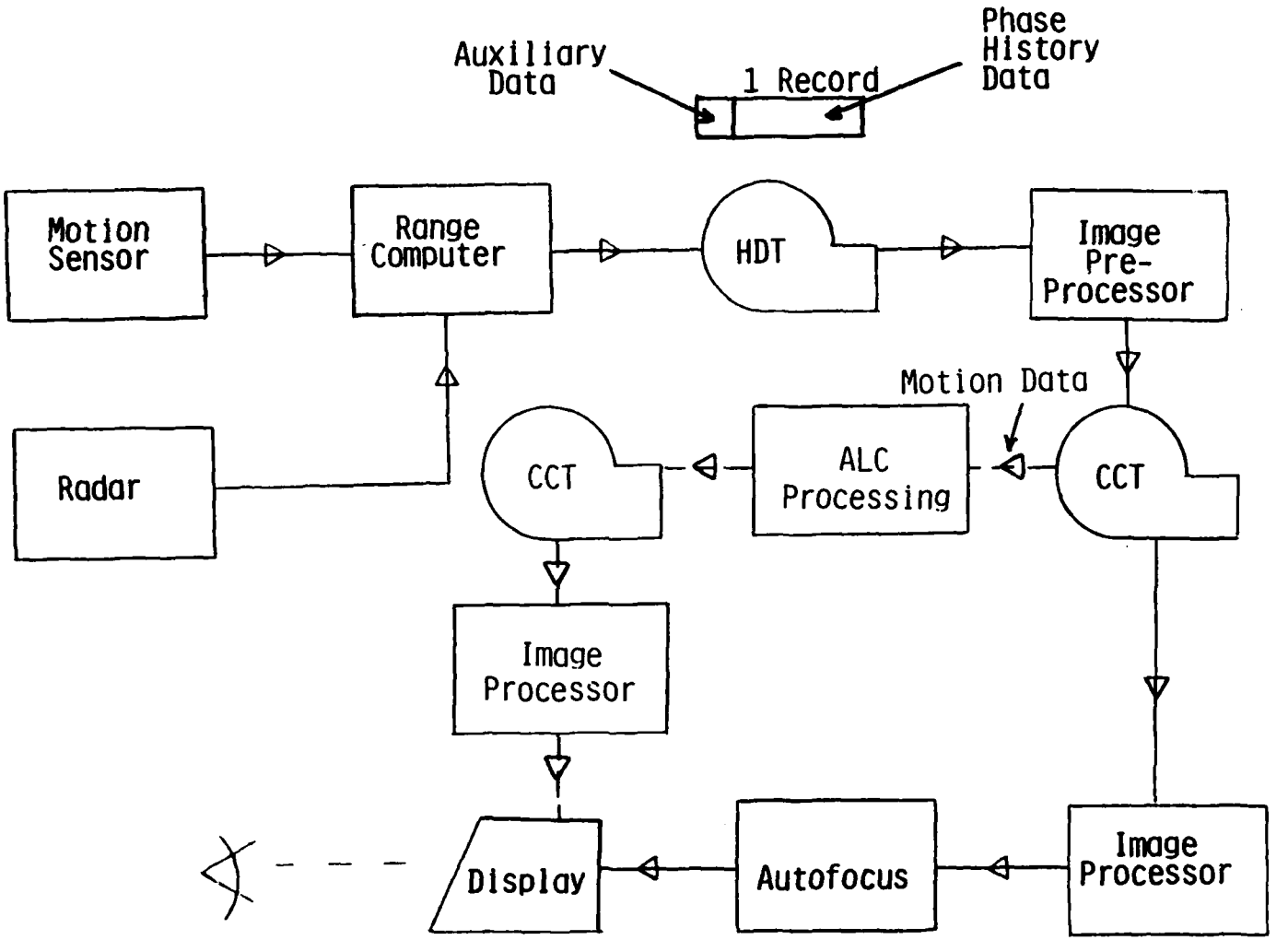


FIGURE 9. OFF-LINE IMAGE PROCESSING.

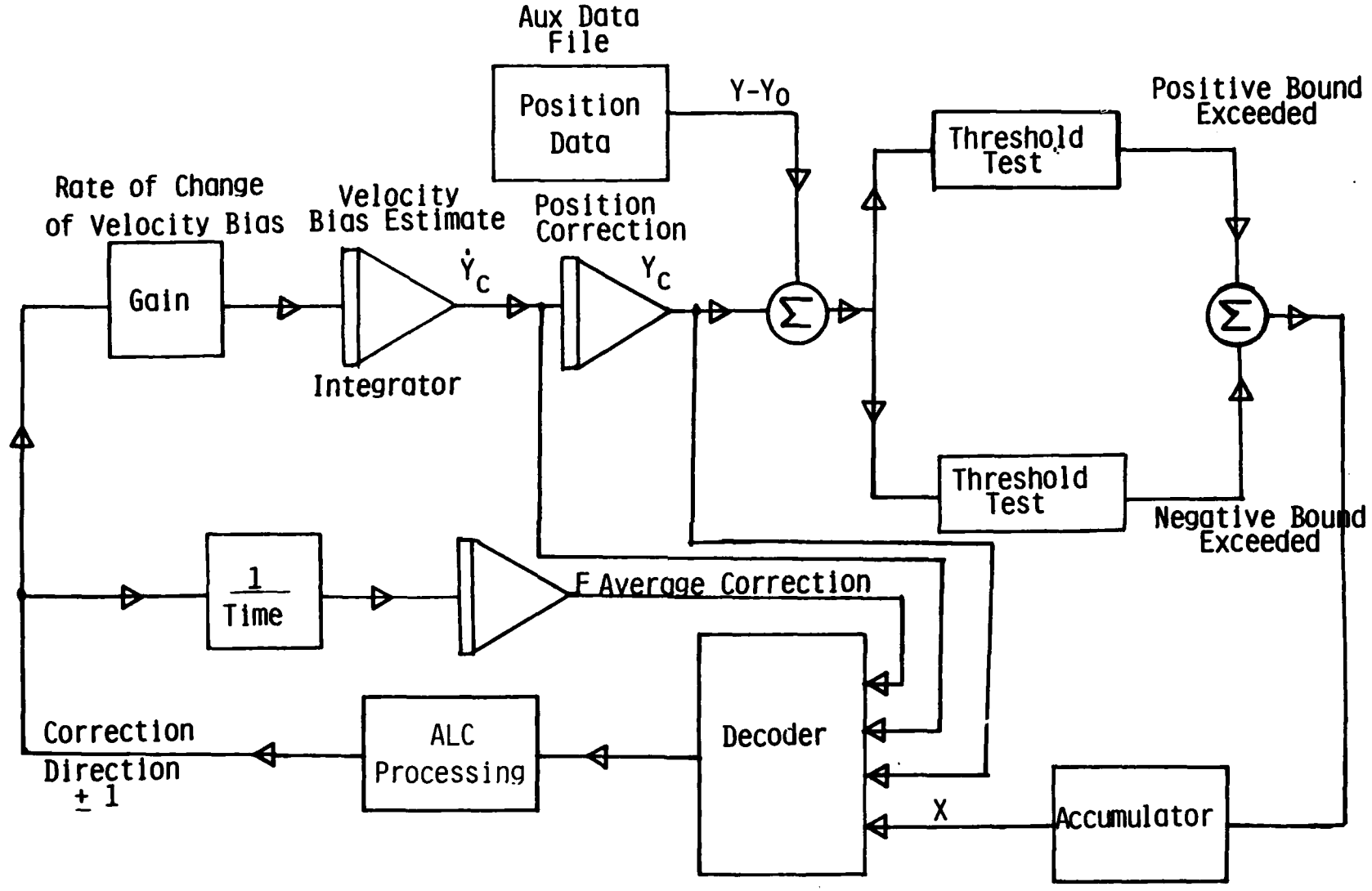


FIGURE 10. ALC APPLIED TO MOTION CORRECTION PROBLEM.

input of the velocity bias integrator. The measurements given to the ALC include: X, Y_c, $\dot{Y}_c$ and F, a measure of the average correction applied. These measurements enter the decoder and a decision is made by the ALC which determines the direction the control signal should be applied. If the ALC applies the proper correction, the resulting motion measurements will stay within the desired bounds.

Failure criteria are defined in Figure 11. If X or F exceed their permissible limits, a failure is declared. This means that either $(Y-Y_o - Y_c)$ continually exceeds the lower or the upper bound, or the ALC is driving the velocity bias integrator as hard as it can in one direction. If Y_c or $\dot{Y}_c$ exceed their limits, a failure is not declared, but since the ALC is not in any box, no learning will occur (i.e., the weights will not change).

The parameters X, Y_c, $\dot{Y}_c$ and F were selected as inputs to the ALC for the following reasons:

1. X is a measure of what Y_c is doing on a long term basis relative to the boundaries and is the primary variable to be controlled;
2. Y_c and $\dot{Y}_c$ are the controls being applied and provide a memory of past control actions;
3. F was added to provide a measure of how hard the ALC has been driving the system in the past so it won't overdrive the system beyond the desired equilibrium point.

The variable X can be thought of as the output of a 2-bit quantizer plus sign followed by an integrator. It allows a certain amount of velocity error to exist before taking corrective action.

For the runs which were performed and are documented here the motion data were read every 0.1 second and the gain was set equal to 0.4 m/sec^2. The decoder was designed as follows:

- The variable X was divided up in 10 states between -20 and 20;

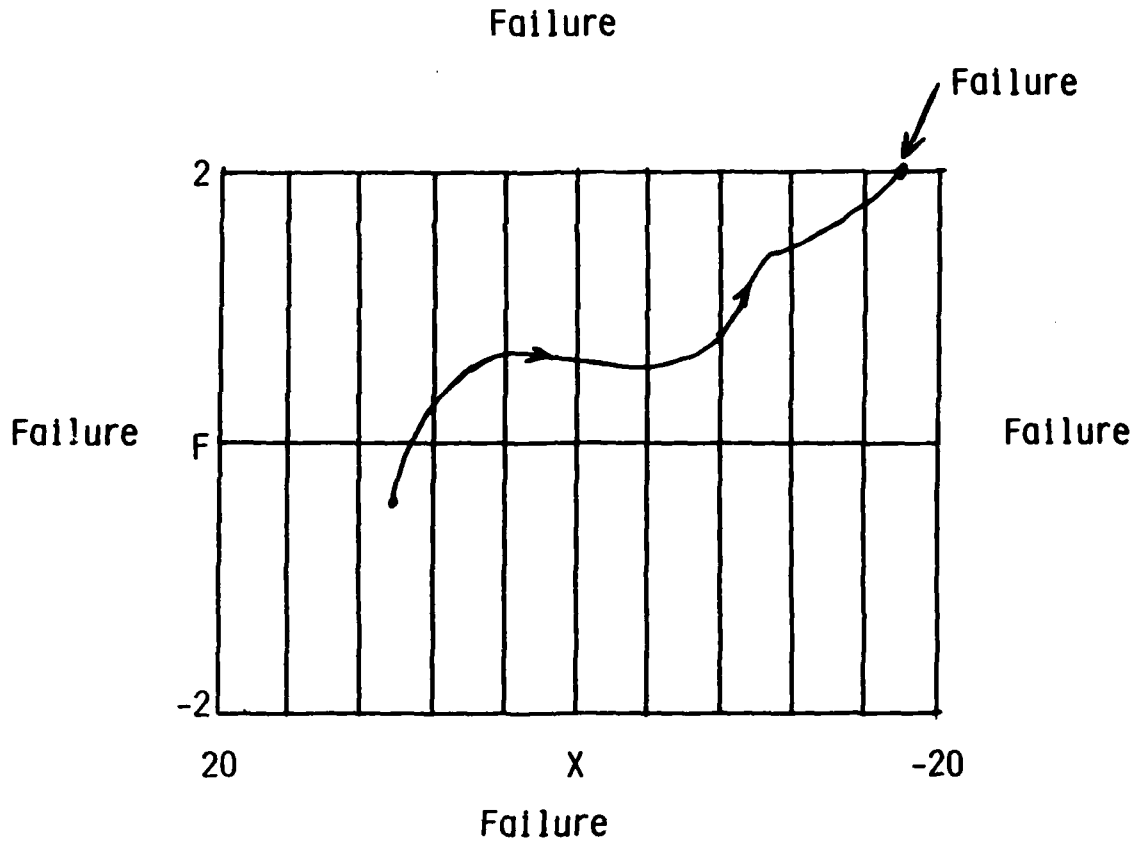


FIGURE 11. FAILURE CRITERIA.

- Y_c was divided up into 10 states between -1 and 1;
- $\dot{Y}_c$ was divided up into 5 states between -1 and 1;
- F was divided up into 2 states between -2 and 2.

Thus the total number of possible states in the decoder was 10 x 10 x 5 x 2 = 1,000. Penalties were assigned to the X states away from zero as follows:

- A penalty of zero for the two intervals on either side of zero,
- A penalty of -0.001 for the next two intervals closest to zero;
- Penalties of -0.002, -0.004 and -0.01 as the intervals moved farther away from zero.

4.5 EXAMPLE RUN OF ALC

4.5.1 Y-BOUND OF 5 METERS

Figures 12 through 15 illustrate the ALC performance for a particular set of motion data with a bias of 1 meter/sec purposely added to simulate performance for a system without beacons and a Y-bound of 5 meters, which was selected to allow 5 sec for the ALC to learn (since with a 1 m/s bias it takes 5 seconds to hit the bound). After 23 trials, the ALC has learned to keep $(Y-Y_o-Y_c)$ within bounds as shown in Figure 12. Figure 13 shows that the ALC has estimated a $\dot{Y}$ bias of 1 meter/second, which is the correct value and Figure 14 shows it has applied a correction of 10 meters after 10 seconds which is also what should be expected. Figure 15 shows F, the average correction applied by the ALC, which is 1 for the first few steps and then decays to zero. The amplitude of the control force used was 0.1 (the gain in Figure 10). Rather than optimize system performance for a Y-bound of 5 meters, an attempt was made to test ALC operation with a lower value of Y-bound since it was believed a more accurate bias estimate would be achieved for a lower Y-bound.

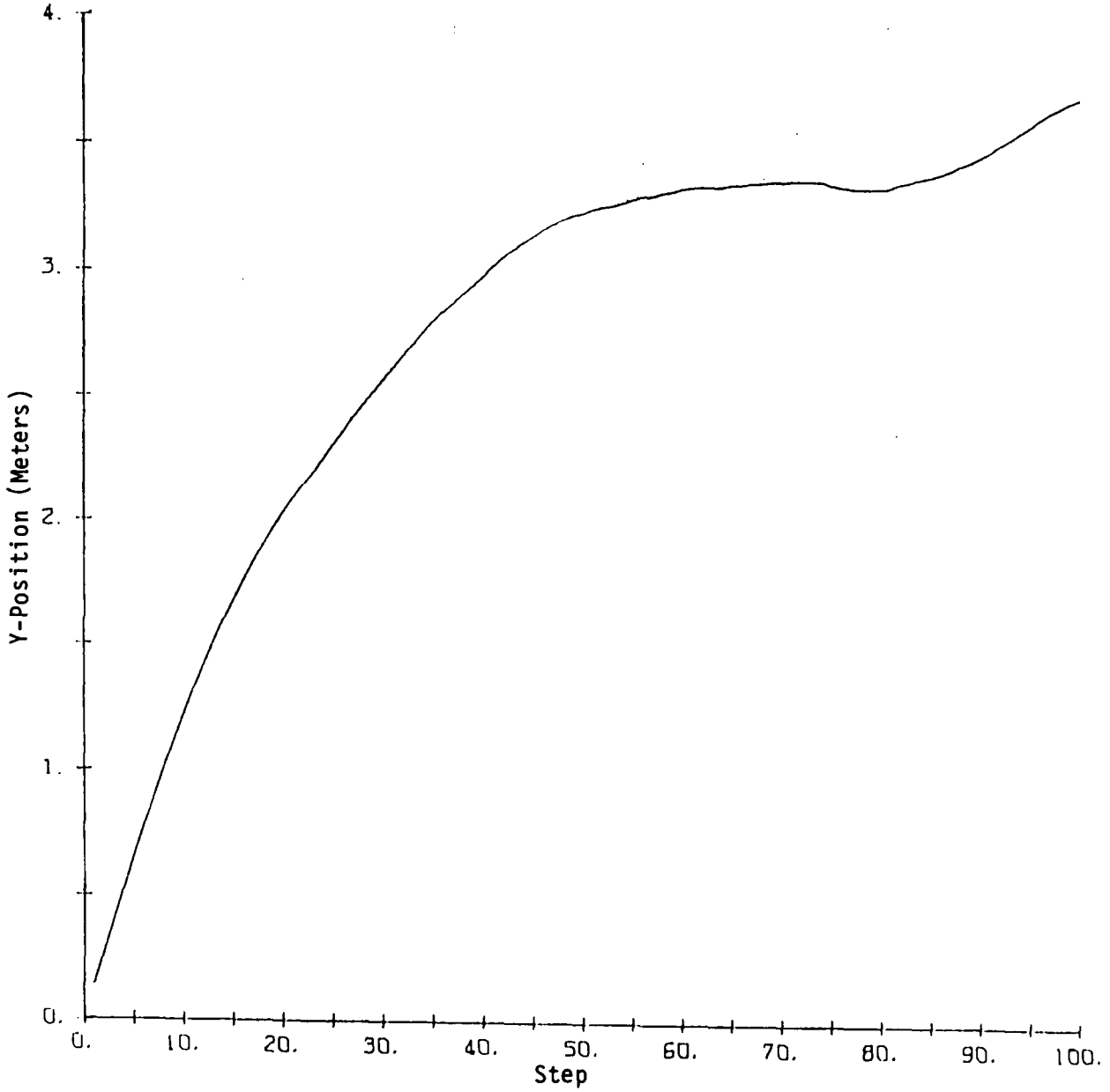


FIGURE 12. PLOT OF $Y - Y_o - Y_c$ VERSUS TIME.
(Y-Bound = ± 5 meters)

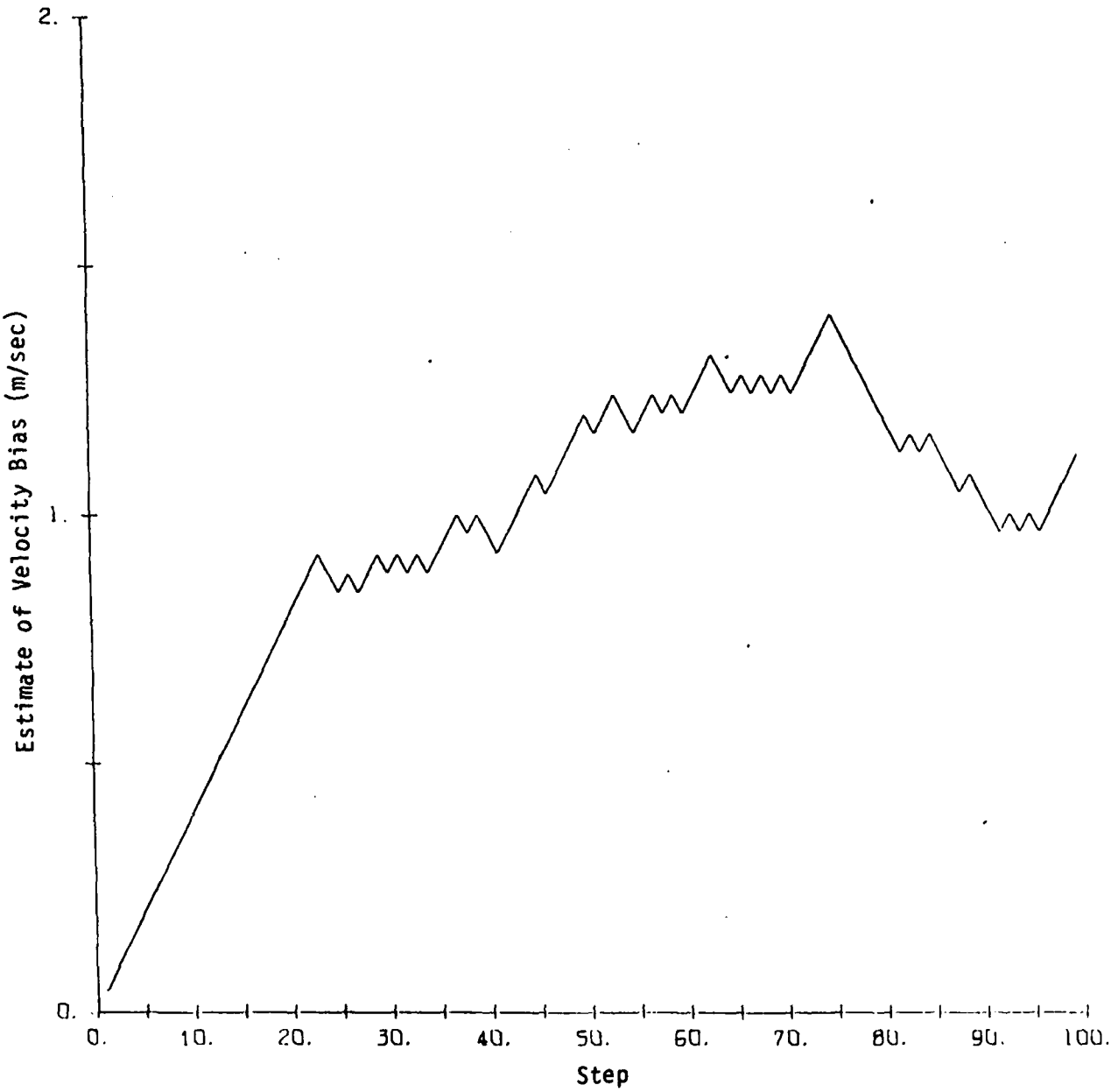


FIGURE 13. PLOT OF $\dot{Y}_c$ VERSUS TIME.
(Y-Bound = $\pm$5 meters)

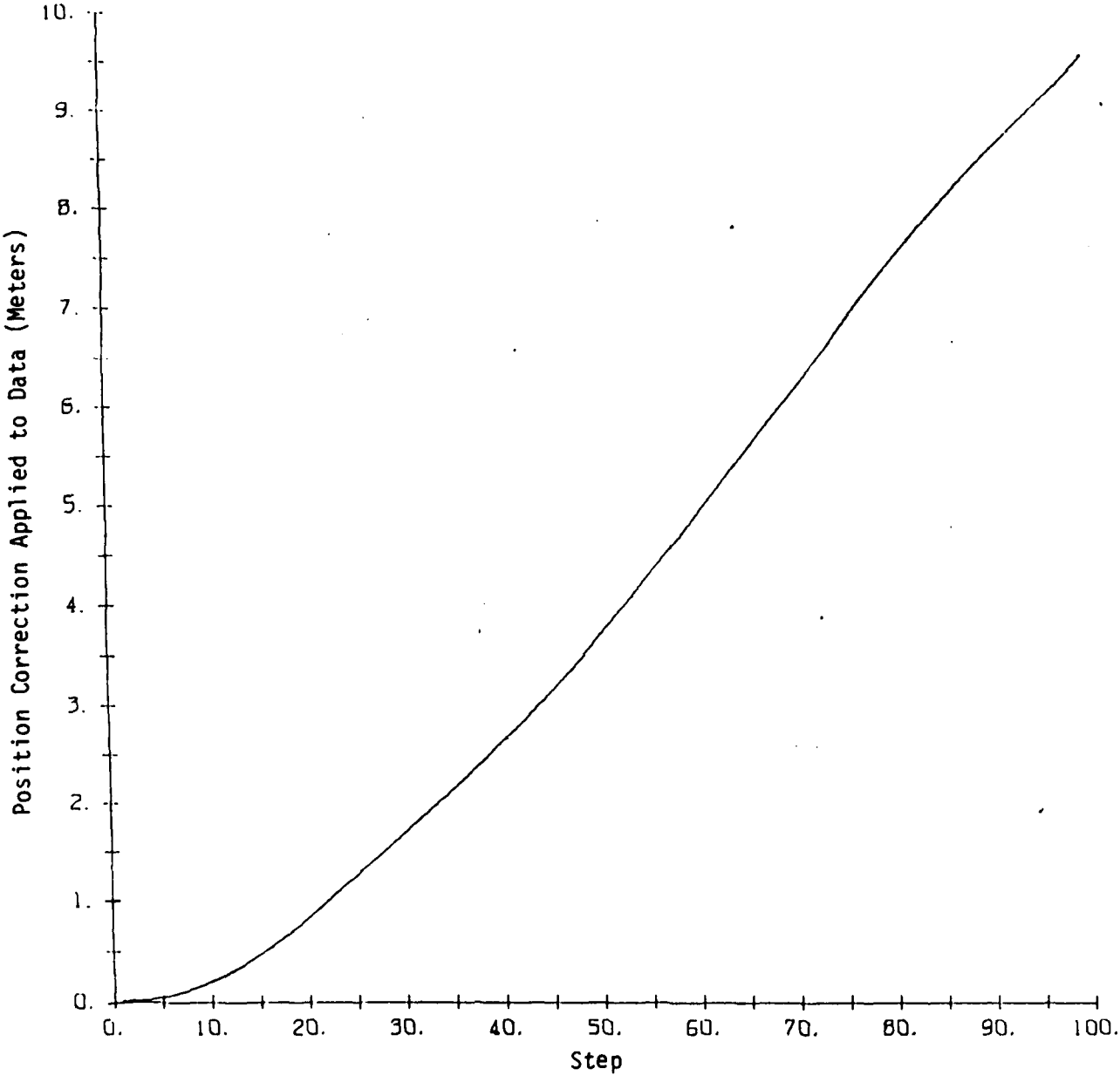


FIGURE 14. PLOT OF Y_C VERSUS TIME.
(Y-Bound = $\pm$5 meters)

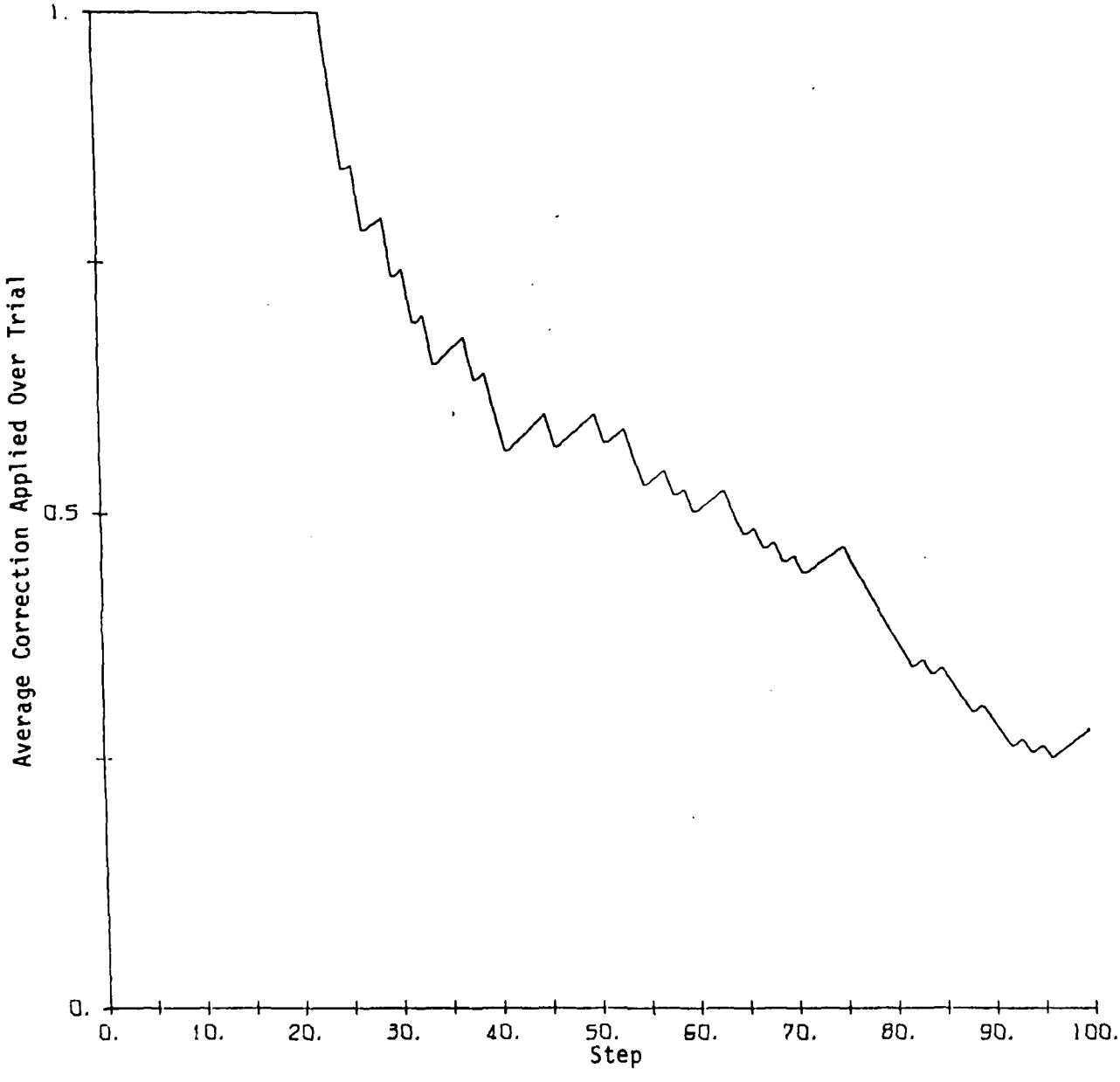


FIGURE 15. PLOT OF F VERSUS TIME.

4.5.2 Y-BOUND OF 2 METERS

Figures 16 through 19 are plots of some of the variables in the ALC learning loop for one data file. Figure 16 is a plot of the corrected Y position as a function of time for a Y-bound of ±2 meters, after the ALC has learned to keep Y within bounds. Each step corresponds to a time interval of 0.1 seconds. Notice that Y never crosses the Y-bound. Again, in order to simulate performance without beacons, a bias of 1 meter per second was purposely added to the original Y measurements. Thus, in order for the ALC to keep the data within bounds, it had to cancel at least a 1 meter per second bias. Figure 17 shows that after 27 seconds, the ALC has corrected out 36 meters of displacement or an average correction of 1.33 meters per second. Figure 18 is a plot of the velocity bias estimate as a function of time and it appears to be oscillating about a value of 1.33 meters per second. The velocity bias estimate of 1.33 meters per second may seem surprising since only 1 meter per second was added to the data. However, the original data file was noted to have a bias of 0.33 meters per second, so a value of 1.33 meters per second should be the expected value. Figure 19 is a plot of F as a function of time and looks similar to an impulse, which is consistent with driving the Y_c integrator to a constant value.

The application of the learning algorithm to SAR imaging was finally demonstrated by generating two images, one by processing SAR data in the normal fashion but without autofocus, and another by correcting the motion data via the ALC and processing, also without conventional autofocus. In order to make the task of the ALC more difficult, a velocity bias of 1 m/sec was also introduced in the motion data. This made the uncorrected image severely blurred. However, the image produced when the motion data were ALC corrected was focused and clear. These images show conclusively that the ALC adaptive learning controller could be used for on-line reduction or cancellation of motion-induced quadratic phase errors.

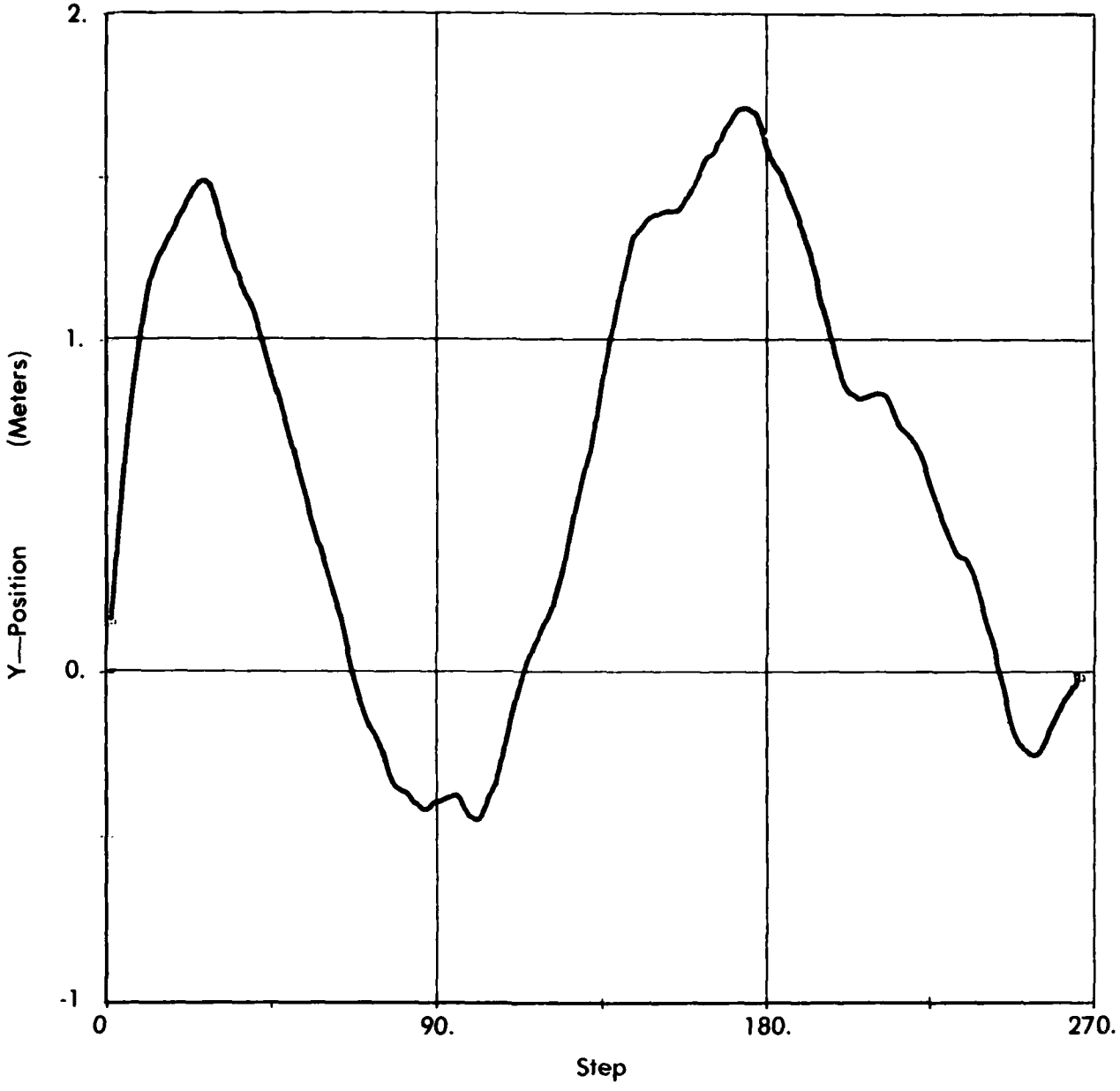


FIGURE 16. PLOT OF $Y-Y_0-Y_c$ VERSUS TIME.
(Y-Bound = ± 2 meters)

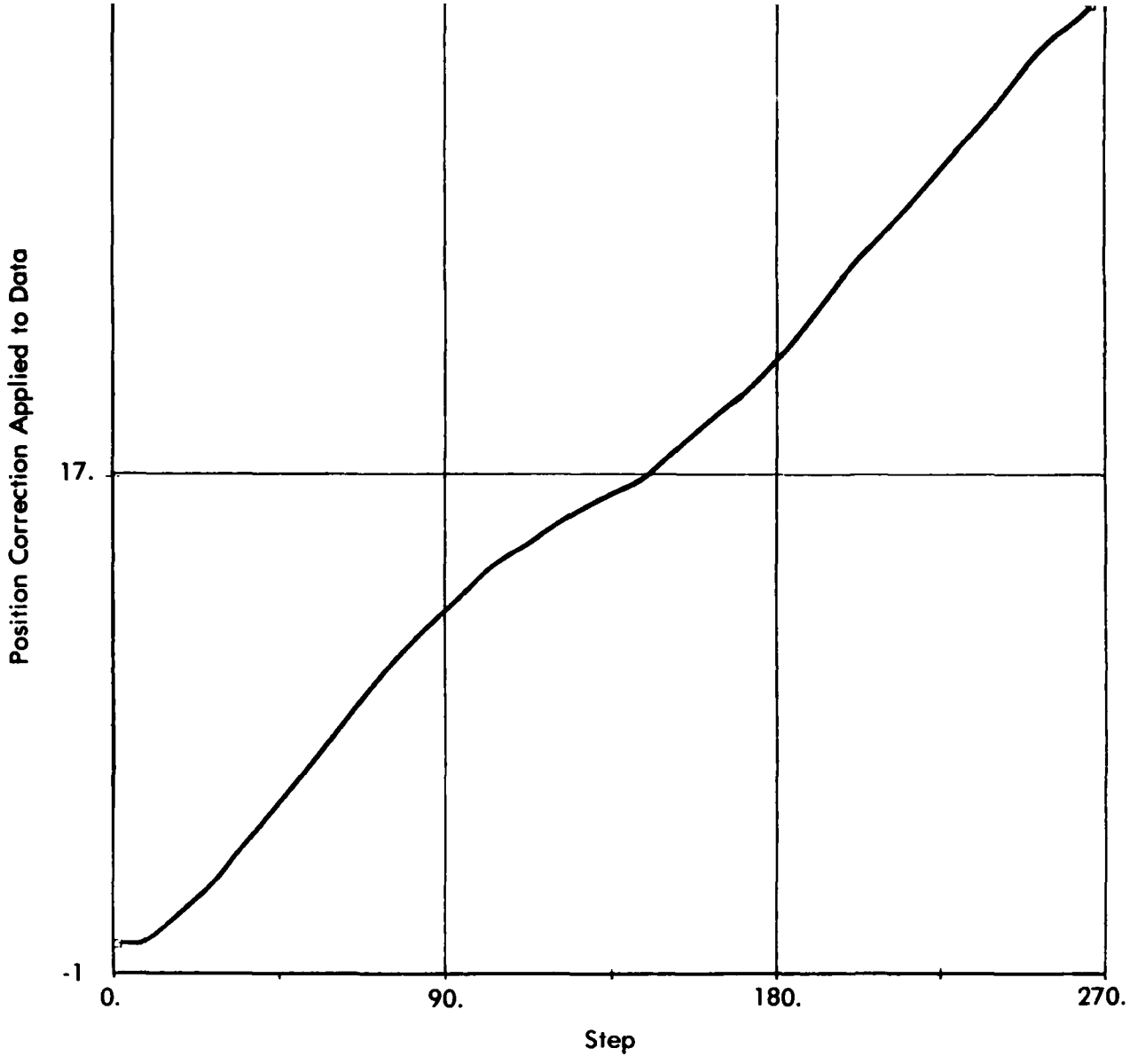


FIGURE 17. PLOT OF Y_c VERSUS TIME.
(Y-Bound = ±2 meters)

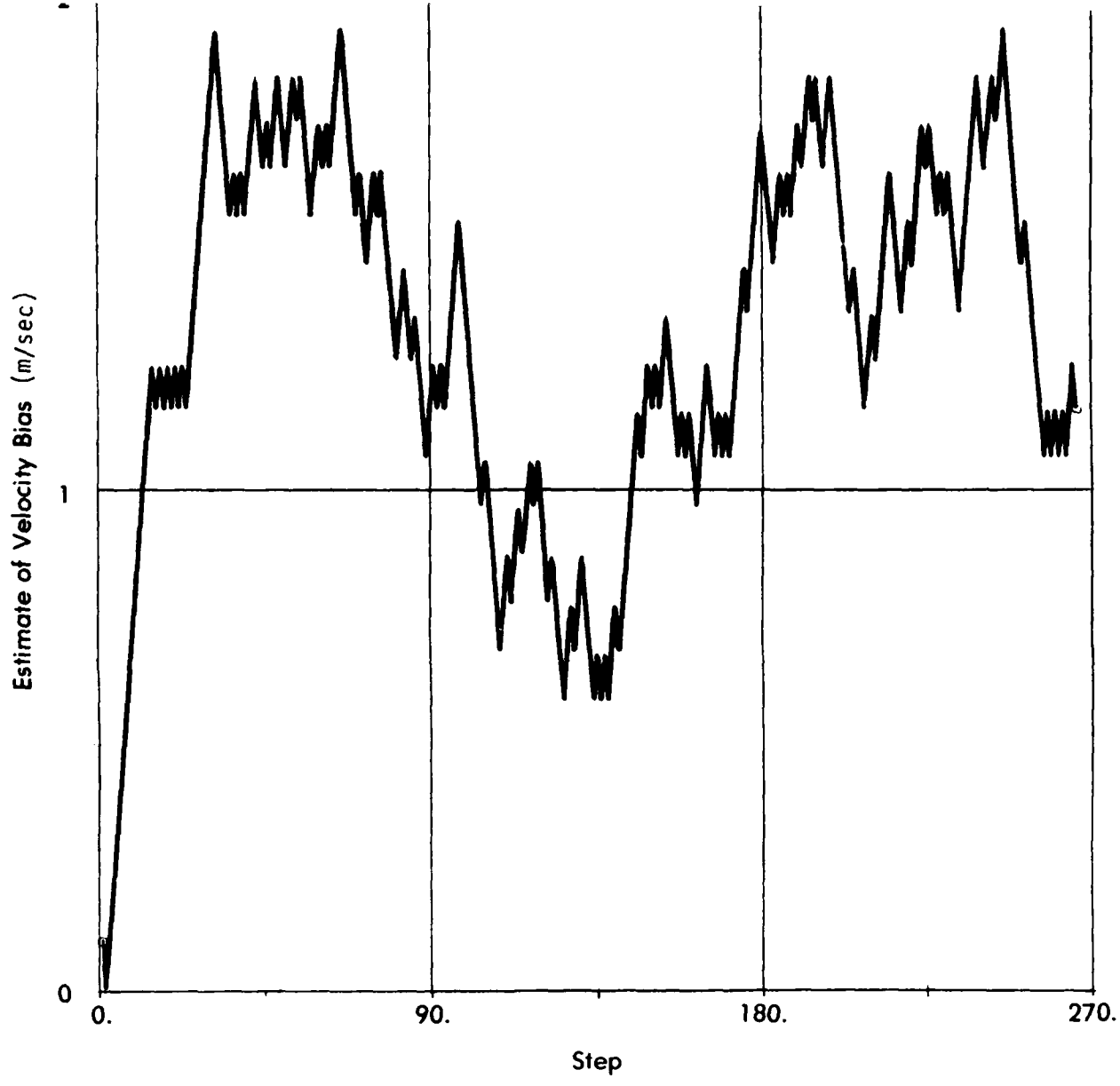


FIGURE 18. PLOT OF $\dot{Y}_c$ VERSUS TIME.
(Y-Bound = $\pm$2 meters)

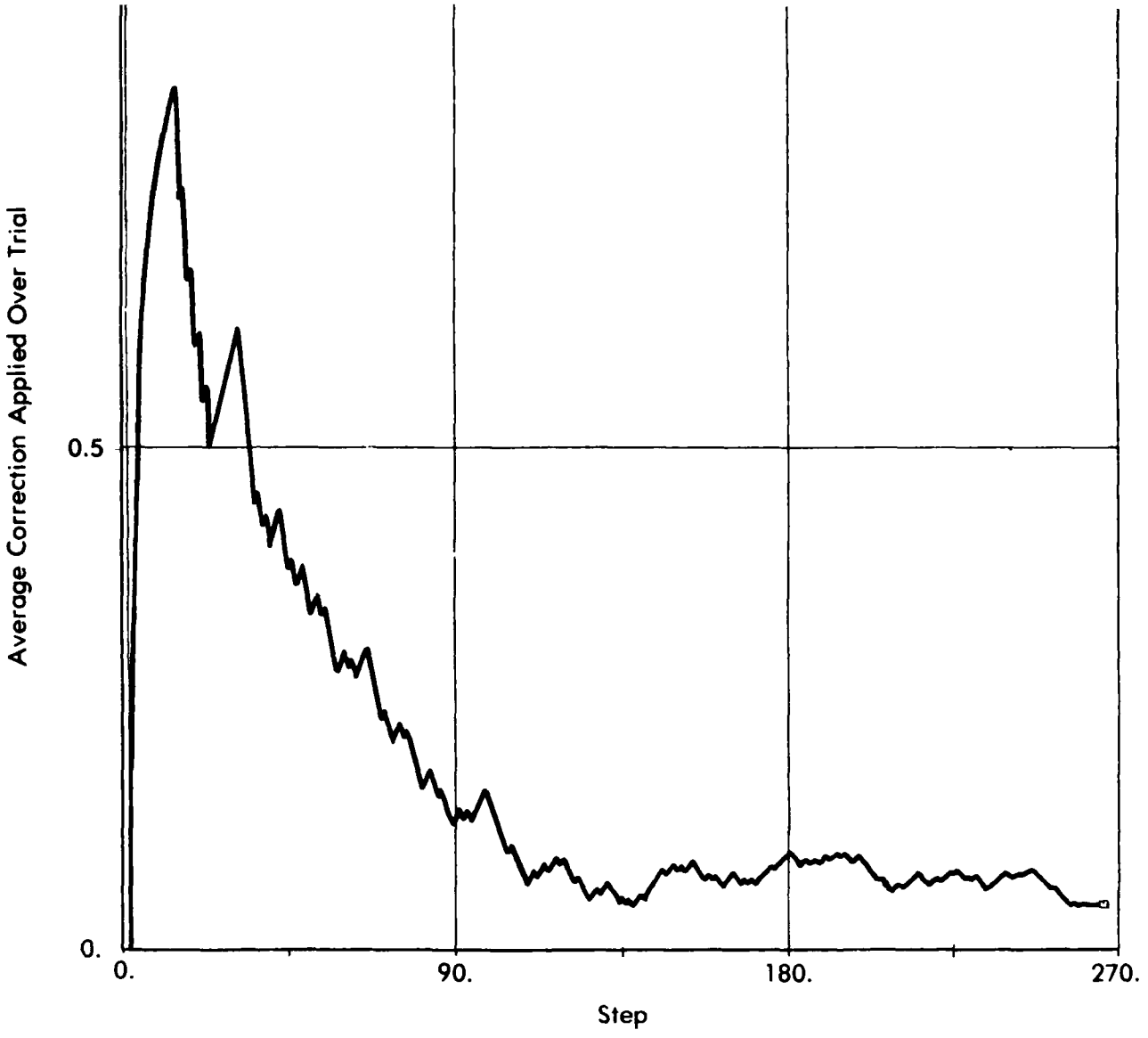


FIGURE 19. PLOT OF F VERSUS TIME.
(Y-Bound = ±2 meters)

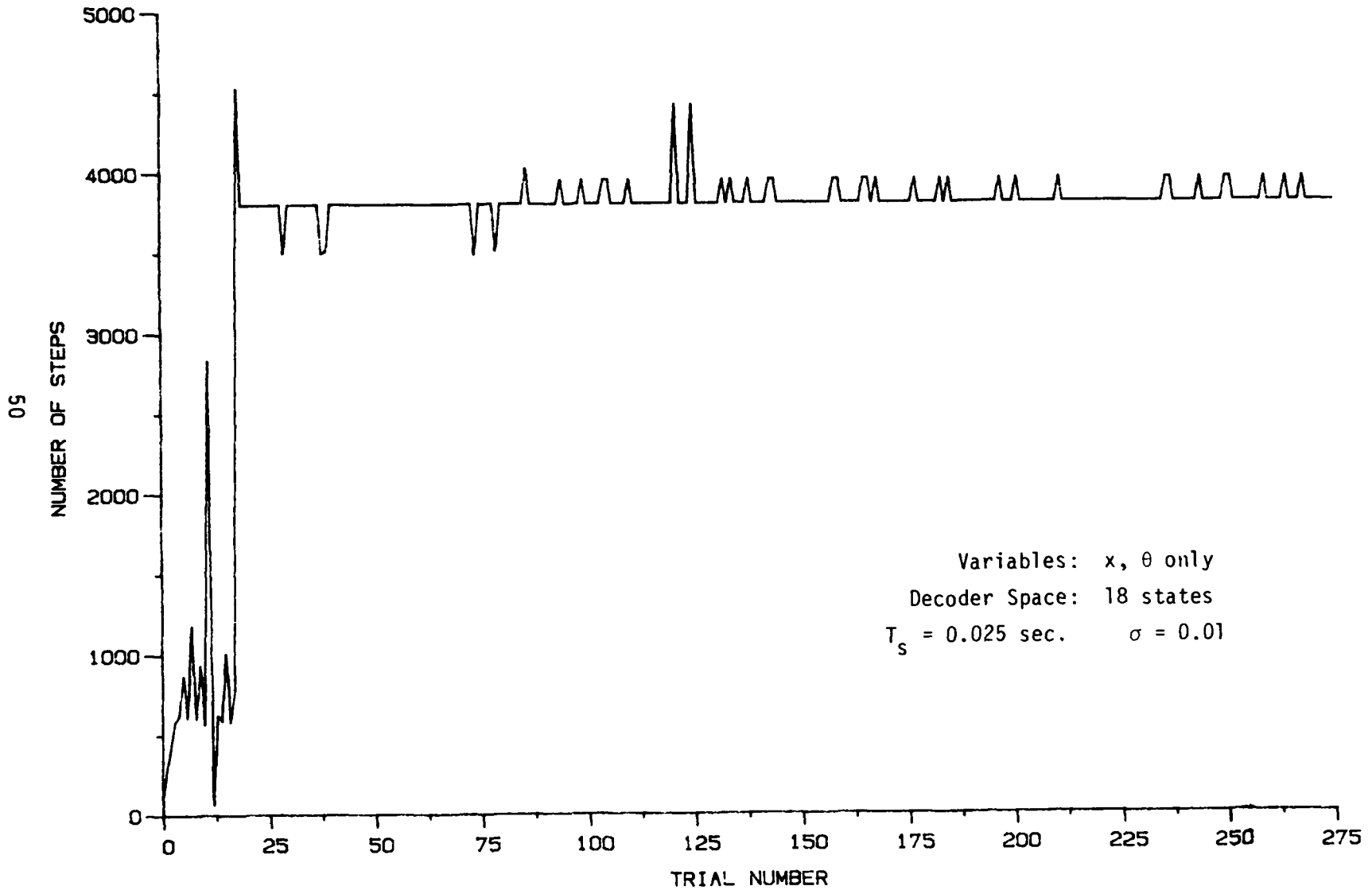


FIGURE 26. ALC LEARNING CURVE.

This was clearly shown when we tested system operation with x and θ inputs only to ALC, for a total of 18 states. Figure 26 shows that the ALC reaches a certain level of control and cannot improve further, because it does not have the information necessary to do it.

5.5 ALC OPERATION WITH BINARY DECODER

Pole and cart control was next investigated with a binary decoder. By binary decoder we mean that the decoder recognizes for each of the state variables only two states, one state for positive values, the other for negative. Failure was set as usual on θ at $\pm 12^{\circ}$ and on x at ± 50 meters.

Figure 27 shows that the ALC quickly reaches a "medium" level of learning, but cannot improve any further. In other words the ALC learns to control the system for about 5 minutes (12,000 steps approximately) but cannot do any better. Obviously, this result indicates that a binary decoder does not supply the ALC with adequate information to learn. Because of the nature of the ALC it appears that the minimum number of required states for each variable is three, one around the desired point of operation and one on each side.

When the test was repeated with failure signals in all four states no improvement resulted. Figure 28 shows that indeed performance deteriorated, the system never being able to run for longer than 30 sec. (1,000 steps). The reason must be that more failure signals impose more severe restrictions on the operation, without on the other hand providing the ALC with more information to be able to cope with the new demands.

Finally, to further confirm and strengthen our understanding of the binary decoder a test was made on a system with a binary decoder but with two variables only, θ and $\dot{\theta}$.

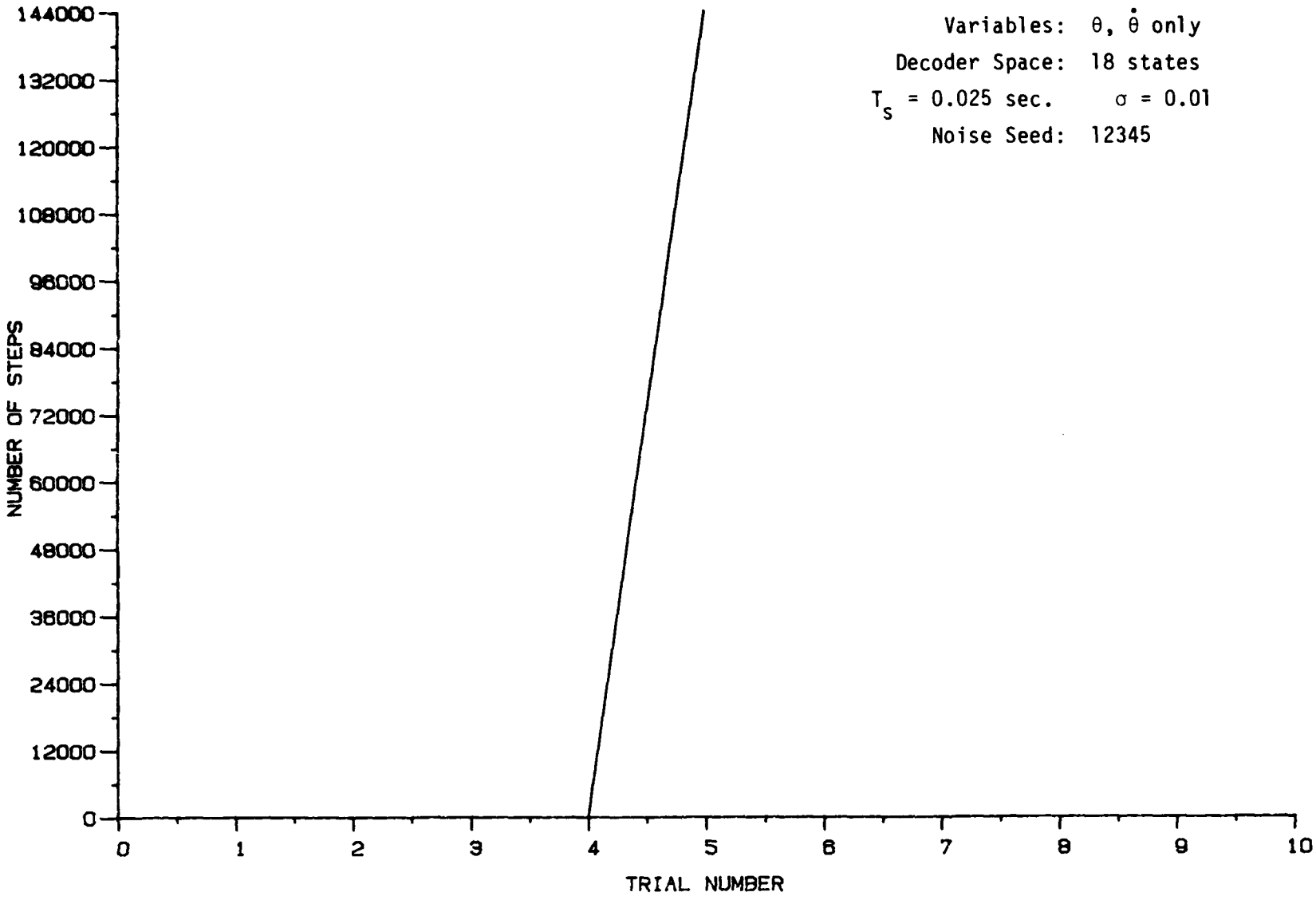


FIGURE 25. ALC LEARNING CURVE.

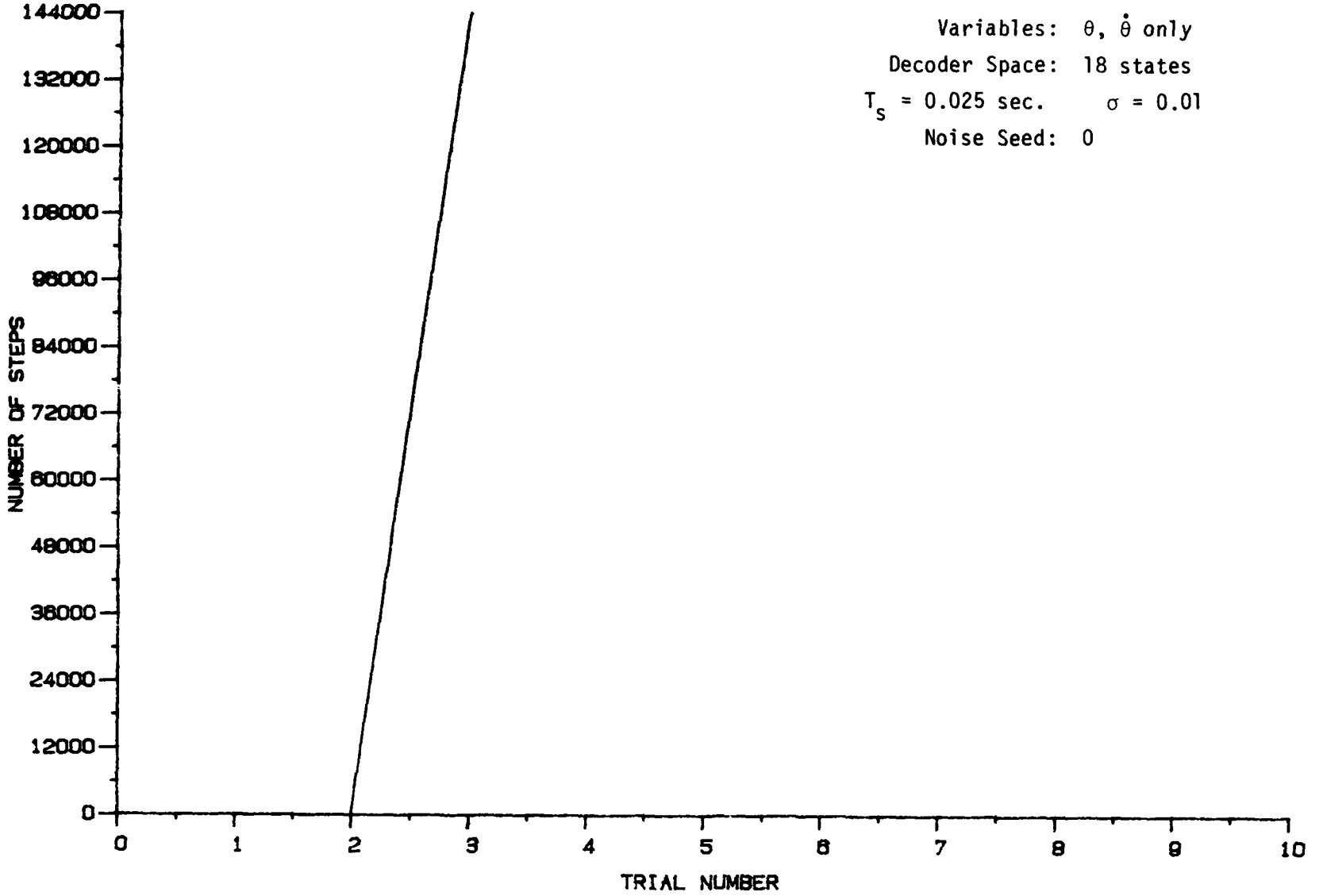


FIGURE 24. ALC LEARNING CURVE

failure $\dot{\theta}$ must take positive and negative values, on the average with equal frequency. Theoretically, therefore, $\dot{\theta}$ can take large values if it would turn around quickly and take equally large values, but of opposite sign. Clearly, the choice of a penalty bound for $\dot{\theta}$, which is equivalent to a desirable θ bound, is difficult to determine.

5.4 ALC OPERATION WITH (θ, $\dot{\theta}$) CONTROLS

Next the ALC behavior with θ and $\dot{\theta}$ only available for control was investigated. This means that the decoder fed to the ALC information on θ and $\dot{\theta}$ only. The θ space was divided into six states and the $\dot{\theta}$ into three states for a total of 18 states for the ALC. Failure bounds were $\pm 12^{\circ}$ for θ and $\pm 30^{\circ}$/sec. for $\dot{\theta}$, as in the standard operation.

By withholding the variables x and $\dot{x}$ from the ALC we have essentially changed and significantly simplified the nature of the control problem. Now the only requirement is that the pole stays within the prescribed bounds of $\pm 12^{\circ}$. Where the cart is located along the x-axis is not important.

The results shown in Figures 24 and 25 indicate that the ALC learned very quickly to control the system. After 14.6 seconds of trials the ALC had the pole balanced for over one hour and operation was automatically interrupted. From the computer printout, where the complete state vector is recorded, it is seen that while the pole was being kept in balance (within $\pm 2^{\circ}$ approximately) the cart was slowly drifting along the x-axis, exceeding the 50 meter mark on the 1013th step. Also, while the ALC was learning the system failed in both θ and $\dot{\theta}$.

This test clearly demonstrated that one does not need to feed into the ALC more information than is necessary to perform the prescribed task. For pole control only θ and $\dot{\theta}$ are needed. For pole and cart control all four state variables are required by the ALC.

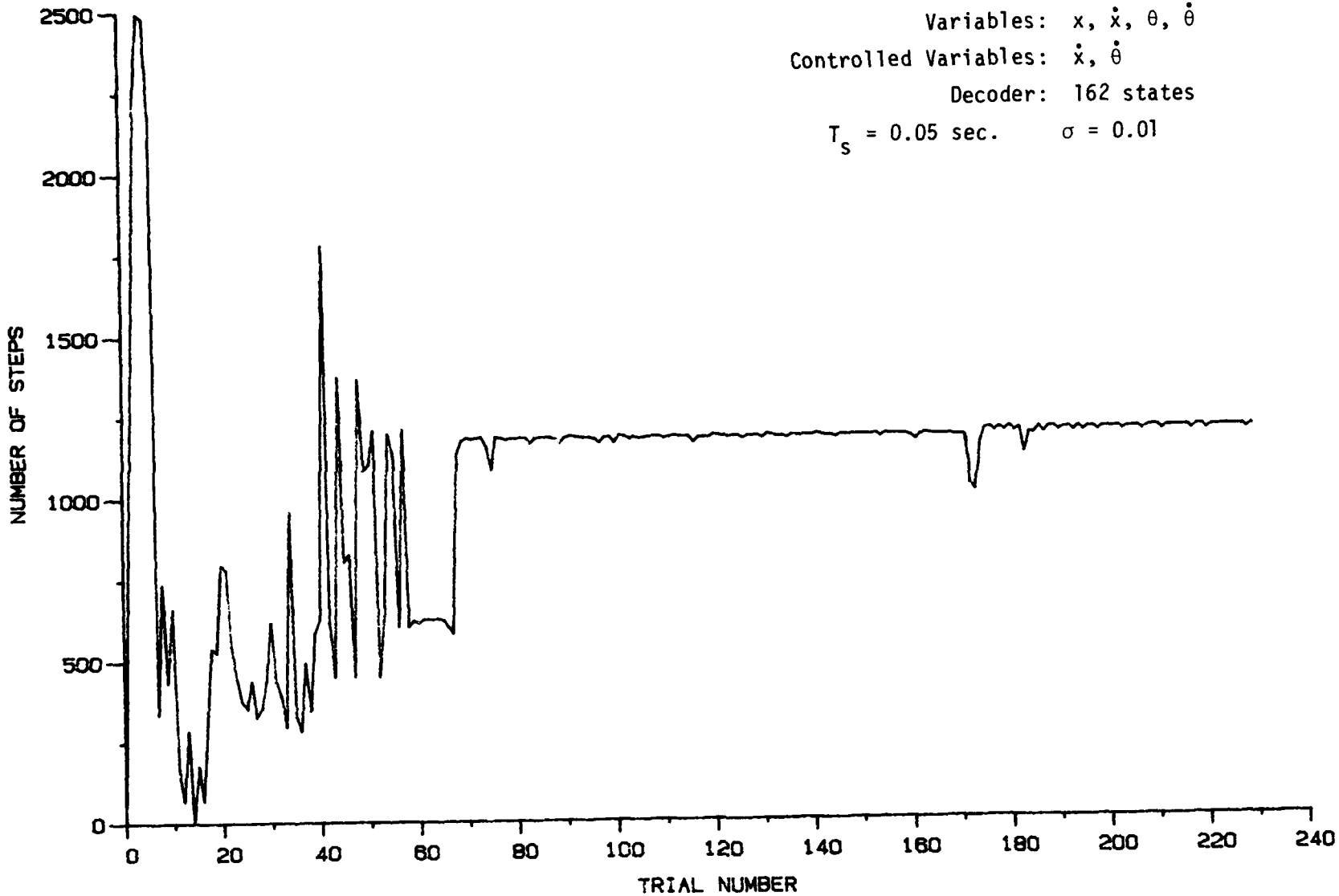


FIGURE 23. ALC LEARNING CURVE.

Hence, if $v \geq 20$ m/sec, the probability of failure in x is high. Similarly, one can reason that if $\dot{\theta} \geq 15^{\circ}$ /sec, θ will quickly exceed the desirable limit of $\theta < 12^{\circ}$.

To test ALC performance with $(\dot{x}, \dot{\theta})$ as controlled variables, several runs were made with the following decoder boundaries:

x: -50, -17, 17, 50 meters

$\dot{x}$: -5, -2, 2, 5 m/sec

θ: -12, -5, 5, 12 degrees

$\dot{\theta}$: -15, -10, -5, 0, 5, 10, 15 deg/sec

The total number of states is the same as before: 3 x 3 x 3 x 6 = 162. Failure was set when $\dot{\theta} \geq 15^{\circ}$ /sec. and T_s was 0.025 sec.

Following the conclusions of last year's study [4], in all the runs we considered that the ALC learned to control the system if it kept the controlled variables within the chosen bounds for over one hour of control time. This means that for T_s = 0.025 sec. the computer was set to cut off when 144,000 steps were exceeded in a run. All runs were performed in the Z-100 microcomputer. The results showed that the system did not learn as well as when the control variables were x and θ (see Figure 23). The reason for this poorer behavior may be that the chosen bounds for $\dot{\theta}$ were not the proper ones, and there is evidence supporting this argument. One of the runs, for example, shows that θ exceeded 12° even though $\dot{\theta}$ stayed below 10°/sec, and by the time $\dot{\theta}$ went over 15°/sec, θ had exceeded 20°.

This points to the difficulty of actually controlling a certain variable, in this case the angle θ, by imposing a penalty on another variable, in this case the derivative of θ. Since θ at t is the integral of $\dot{\theta}$ over the interval (o, t), if θ is to stay below a certain chosen value, the total area under $\dot{\theta}$ has to maintain a value less than the θ bound. This means that for the system to avoid

These results substantiate our findings from the runs we performed last year, testing the learning behavior of the pole-on-cart system for various sampling periods. It was shown then that the learning rate deteriorated significantly as T_s became greater than 0.1 sec. Thus, 0.1 second is the upper bound for the sampling period for the pole-on-cart system with these parameters.

5.3 USE OF $\dot{x}$ AND $\dot{\theta}$ AS CONTROL VARIABLES

The choice of the system variables to be controlled is at the present time made on an intuitive basis. In the pole-on-cart problem, for example, by defining acceptable performance to be that the variables x and θ stay within chosen bounds, almost automatically we must consider that the system has failed when these variables exceed these bounds and we require that a failure signal then be generated. The advantage in this choice is that failure bounds are easily and clearly defined: if it is desired to keep θ below 12° whenever θ exceeds 12°, the system fails. Yet there is no assurance that θ is the best variable to place failure bounds on to achieve optimum performance. It may be possible that using $\dot{\theta}$ as a control variable, better performance may be achieved, since a high $\dot{\theta}$ value at acceptable θ values may imply that the system will fail no matter what action is taken. The difficulty, however, is that it is not at all evident what failure bounds must be set for $\dot{\theta}$.

Let us try to establish some reasonable bounds for $\dot{\theta}$ and test for system performance. We will assume that θ is small.

Using the dynamic characteristics of the system obtained previously, control values for $\dot{\theta}$ and $\dot{x}$ were derived as follows:

First, we will assume that $T_s \leq 0.1$ sec. Since the probability of a large number of steps in the same direction is very small, for the cart velocity to reach a value like 20 m/sec, the system must go through a long series of short bursts of steps in the same direction.

torque while at the same time pulling the center of gravity of the pole down so that pivot point P stays on the moving mass.

For small θ (i.e., $\theta < 6^\circ$, $\sin\theta < 0.1$), the effect of the weight is small compared to F and can be ignored. Figure 22 shows the pole making an angle $\Delta\theta$ with the vertical. The effect of F will be to cause it to rotate about its center of gravity, the angle $\Delta\theta$ equaling

$$\tan \Delta\theta \cong \Delta\theta = \frac{\Delta x}{\ell/2} = \frac{2}{\ell}\Delta x \text{ radian}$$

From Eq. (4) for $t < 5$ seconds, the displacement x of P is given by

$$x(t) = 5t^2 \text{ meters}; \quad 0 < t < 5 \text{ sec}$$

$$dx = 10t\, dt$$

If the sampling period $T_S = 0.1$ sec, $\ell = 20$ meters, $\theta \cong x/10$ radian

$x(0.1) = 5(0.1)^2 = 0.05$ m	$\theta = 0.005$ rad
$x(0.2) = 0.2$ m	$\theta = 0.02$ rad $< 1.2^\circ$
$x(0.3) = 0.45$ m	$\theta \cong 0.045$ rad $< 3^\circ$
$x(0.5) = 1.25$ m	$\theta \cong 0.125$ rad $\cong 7^\circ$

We can see, therefore, that if the sampling period, T_S, is greater than 0.1 sec, the learning rate will be slow because three to four successive moves in the same direction will cause the system to fail. Taking $T_S = 0.05$ sec, in 5 T_s:

$$x(5T_s) = 0.3125 \text{ m and } \theta \cong 0.03125 \text{ rad} < 2^\circ$$

Thus, $T_s = 0.05$ sec will give a much better system performance, by allowing sufficient time for the system to learn and take appropriate action to avoid failure. Further, we note that since T_s must be less than 0.1 sec, the system will take at least fifty steps in 5 seconds. Because the probability of $y = +1$ (or $y = -1$) in more than 50 consecutive steps is very small, it follows that for approximate estimates of cart performance, Eq. (4) is adequate.

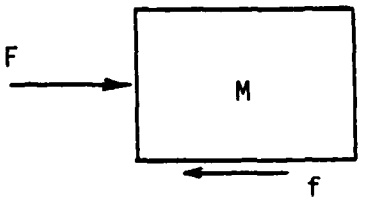


FIGURE 20. THE CART FREE-BODY DIAGRAM.

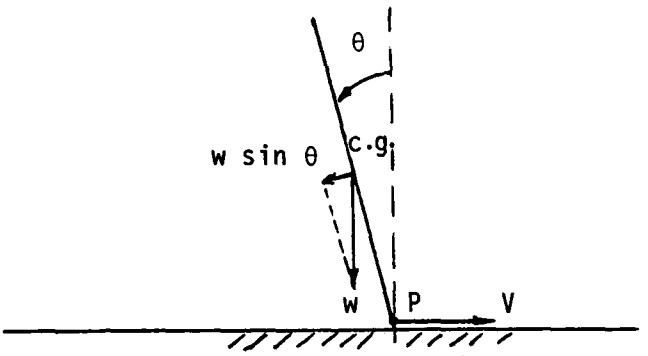


FIGURE 21. THE POLE AT ANGLE θ.

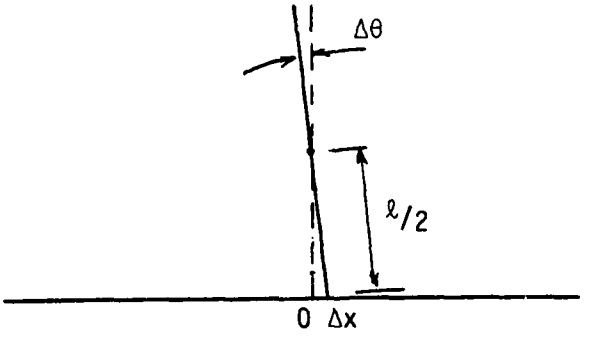


FIGURE 22. THE POLE AT ANGLE Δθ.

The dynamic behavior of the cart can be obtained by looking at the free-body diagram of the cart, which is shown in Figure 20. The equation describing the motion of the system is:

$$F - fv = M \frac{dv}{dt} \tag{1}$$

where f is the coefficient of friction, from which we get, solving for v:

$$v(t) = \frac{F}{f}\left(1 - e^{-\frac{f}{M}t}\right) \quad t > 0 \tag{2}$$

Introducing the chosen values of the system parameters: F = 10 newtons, M = 1 kg, f = 0.01 newton-sec/m, we have:

$$v(t) = 10^3\left(1 - e^{-0.01t}\right) \text{ m/s} \tag{3}$$

So we see that the fundamental cart time-constant is approximately 100 seconds. This is a very large value and, hence, the critical system time-constant with regard to sampling rate will come from the dynamic behavior of the pole. Also, note that for t < 5 sec, or when friction is ignored, the cart velocity is given by

$$v(t) \cong 10t \text{ m/s} \qquad 0 < t < 5 \text{ sec} . \tag{4}$$

Looking now at the pole alone, we see that its behavior can be analyzed by considering it as pivoted to a very large mass, which is moving with a velocity v(t), given in Eq. (3), independently of the motion of the pole.

Figure 21 shows a pole of length ℓ making an angle θ with the vertical while the pivot point P is moving with velocity v. There are two forces acting on the pole: its weight w acting through its center of gravity and a constant force F on P. When the pole makes an angle θ, the weight generates a torque $T = (w \sin \theta)\,\ell/2$, which tends to rotate the pole about the pivot point P so as to increase θ and cause the system to fail. The force F also produces a rotating

If one ALC can control any number of system variables (ignoring running time problems), then it seems that a second ALC would be needed only when it is desired to control two independent but possibly interacting systems.

Two ALC's could also be used if more than one input to a system can be applied. In the pole-on-cart problem, for example, if the cart is free to move on a plane instead of the x-axis only, one ALC can control the force F_x that results in motion in the x-axis direction, and another ALC can control the force F_y that controls the cart motion in the y-axis direction.

5.2 THE DYNAMIC PERFORMANCE OF THE CART-POLE SYSTEM

The learning rate of the ALC is seriously affected by the system sampling rate used for two reasons: (a) the sampling rate must satisfy the Nyquist criterion in order to make sure we are indeed making use of all the information in the output of the system to be controlled, and (b) at every sampling instant a decision is made on whether the input should be plus 1 or minus 1, and the ALC weighting vectors $\underline{w}(t)$ and $\underline{v}(t)$ are updated, hence the system performance in the next operation interval is modified. Also, the way the state-space is partitioned and discretized affects, and is affected by, the choice of the sampling period.

In order to better understand the significance and get quantitative measures of the choice of the sampling period to the learning rate of the ALC, it is necessary that we get a feel for the dynamic performance characteristics of the system.

An approximate but sufficiently accurate and quick estimate can be derived as follows. First, we recognize that since the mass M of the cart (1 kg) is much greater than that of the pole (0.1 kg), which is pivoted on top of it, the motion of the pole has a minor effect on the motion of the cart. Hence, we can analyze approximately the system by looking at each body separately.

There is another set of questions that must be definitively answered before the parametric study is made. Some of these are:

1. How many state variables can one ALC work on and control? Is there a limit?
2. Should the complete state vector be the input to the ALC?
3. How many variables could/should be used to produce failure signals? Is there an optimum number? An optimum set? How does one determine this set?
4. When do we need a second ALC? How do we operate them in series/parallel?

We do not have definitive answers to these questions at the present time. Logically, it would seem desirable or even necessary to input to the ALC the complete system state vector. This way complete information on the state of the system will be available to the ALC at all times. One cannot say, however, that if some of the state variables which are not used for system control are omitted, the ALC cannot learn to control the system. What the effect of a reduced state vector input will be on the learning ability of the ALC to perform an assigned task will have to be investigated. It will certainly reduce the running time of the algorithm, which is always desirable.

A failure signal may be produced, if so desired, by any or all of the state variables. Intuitively, it seems that it would be best to get failure signals only from those variables we wish to directly control, i.e., keep within bounds, if for no other reason because appropriate failure bounds for these variables are easy to recognize. It does not necessarily follow that if these variables are chosen for control best ALC learning behavior will result. For example, in the pole-on-cart problem, the goal is to keep the variables x and θ within chosen bounds. But the pair (x, θ) may not be the best variables to control to achieve this goal.

5
ALC DESIGN STUDY

5.1 INTRODUCTION

Let us assume that we are given an engineering problem in which it has been determined that the ALC can be profitably applied.

To design the appropriate ALC that will match the problem or system requirements, the following decisions must be made regarding the ALC parameters:

1. Determine which variables should be used for control and set the appropriate failure bounds;
2. Decoder design, i.e., decide on the system state space partition (number of states and state bounds);
3. Choose the sampling period, T_s;
4. Set values for the ALC parameters (constants) α, β, γ, δ and noise standard deviation, σ;
5. Choose the magnitude of the force, $\pm F$, the output of ALC; and
6. Finally, we must decide whether one ALC will suffice, or whether two ALC's may be needed.

This is not necessarily the order in which these decisions have to be made, neither are they independent of each other. There is, for example, an upper limit on T_s, which is determined by the dynamics of the system, but how often we should sample also depends on the decoder design.

In order to be able to answer these questions, the behavior of the ALC must be understood well enough to permit optimal design. Since ALC's do not admit to analytical study, laborious parametric study is the only available course of action.

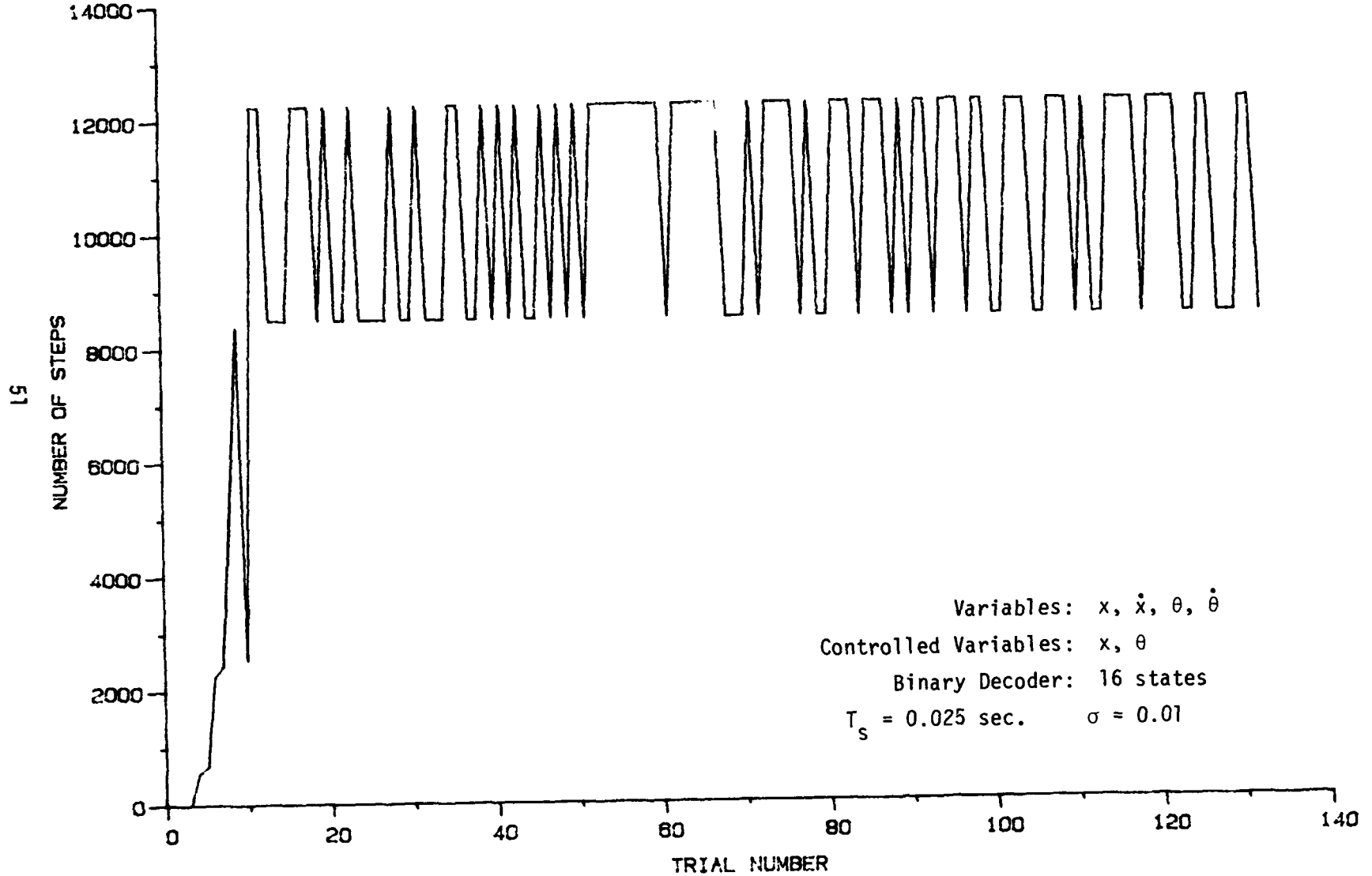


FIGURE 27. ALC LEARNING CURVE.

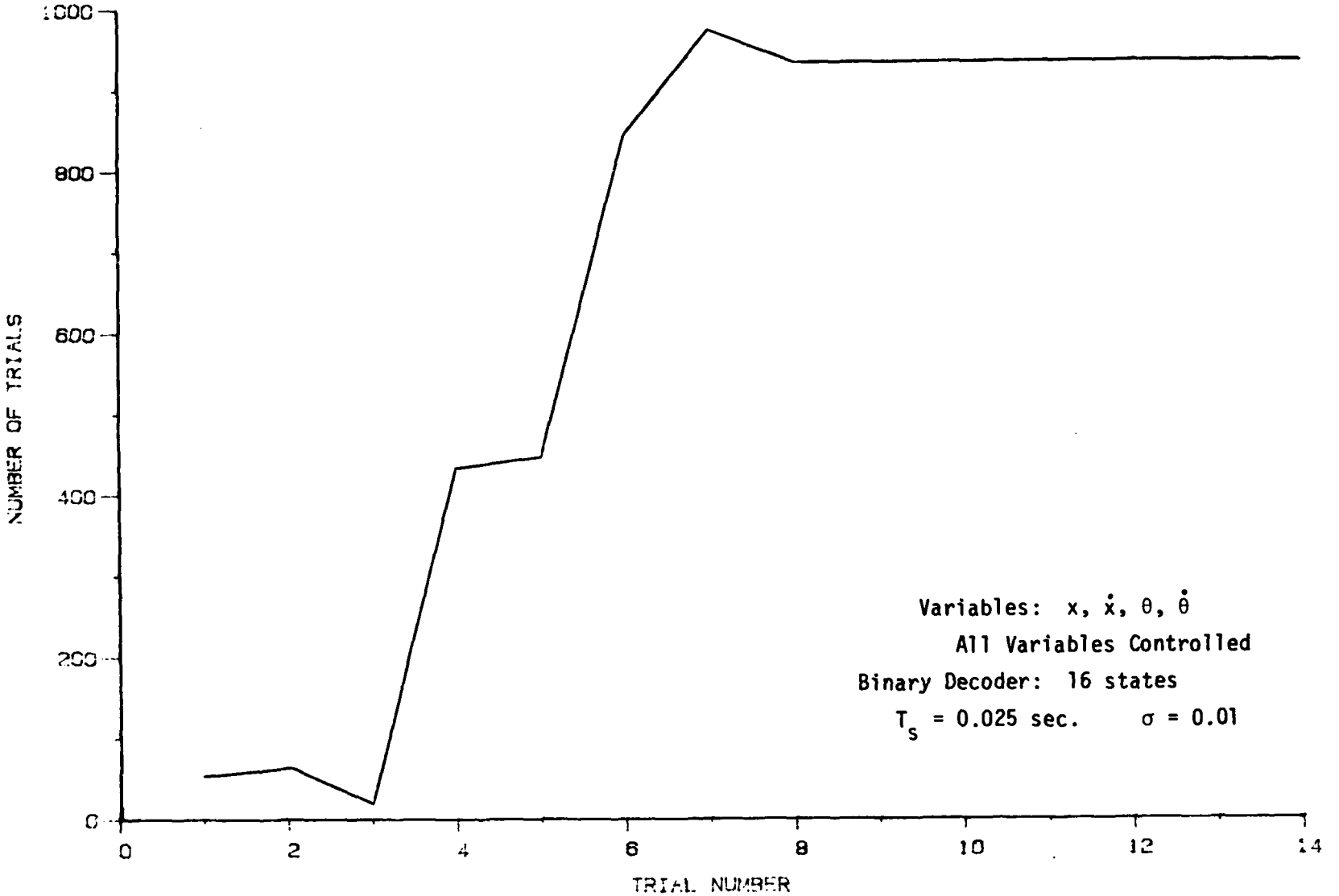


FIGURE 28. ALC LEARNING CURVE.

When this simpler system was run with an 18 state decoder the ALC learned very quickly to control it and performed very well (see Section 5.4). With a binary decoder (a total of four states here) the ALC was never able to run the system for a one hour period. Figures 29 and 30 show that the ALC learned quickly to run for about one-half hour (about 65,000 steps) and then failed. Again this indicates that two states for a control variable do not provide sufficient information (room for comparison of performance) for the ALC to learn to carry out the desired task.

5.6 EFFECT OF NOISE ON ALC LEARNING BEHAVIOR

It has been suggested, without proof, that the injection of noise at the ASE at the point where the output y is generated enhances the learning rate of the ALC. Specifically, it is felt that it forces the system state vector to move quickly through the various system states, thus generating w_i and v_i weights in each state faster than if there were no noise present.

To resolve this question and determine conclusively the effect of noise level on the learning characteristics of the ALC, an exhaustive study was carried out. The system used for this study was again the pole-on-cart and we declared that the ALC learned to control it if it maintained the controlled variables within the chosen bounds for over one hour. When step sizes of various lengths are used this is a better learning criterion.

To test the performance of the ALC on the pole-on-cart system, noise standard deviations and step sizes were varied while the other system parameters were held constant. The combinations tested were noise sigmas of 0, 0.001, 0.01, 0.1, 1., 10. and 100., against step sizes of 0.01, 0.025, 0.05, 0.1, 0.15, 0.2 and 0.25 seconds. Noise sigma multiplies a Gaussian random number generator to produce the noise input in the ALC. Again, step size is the sampling period,

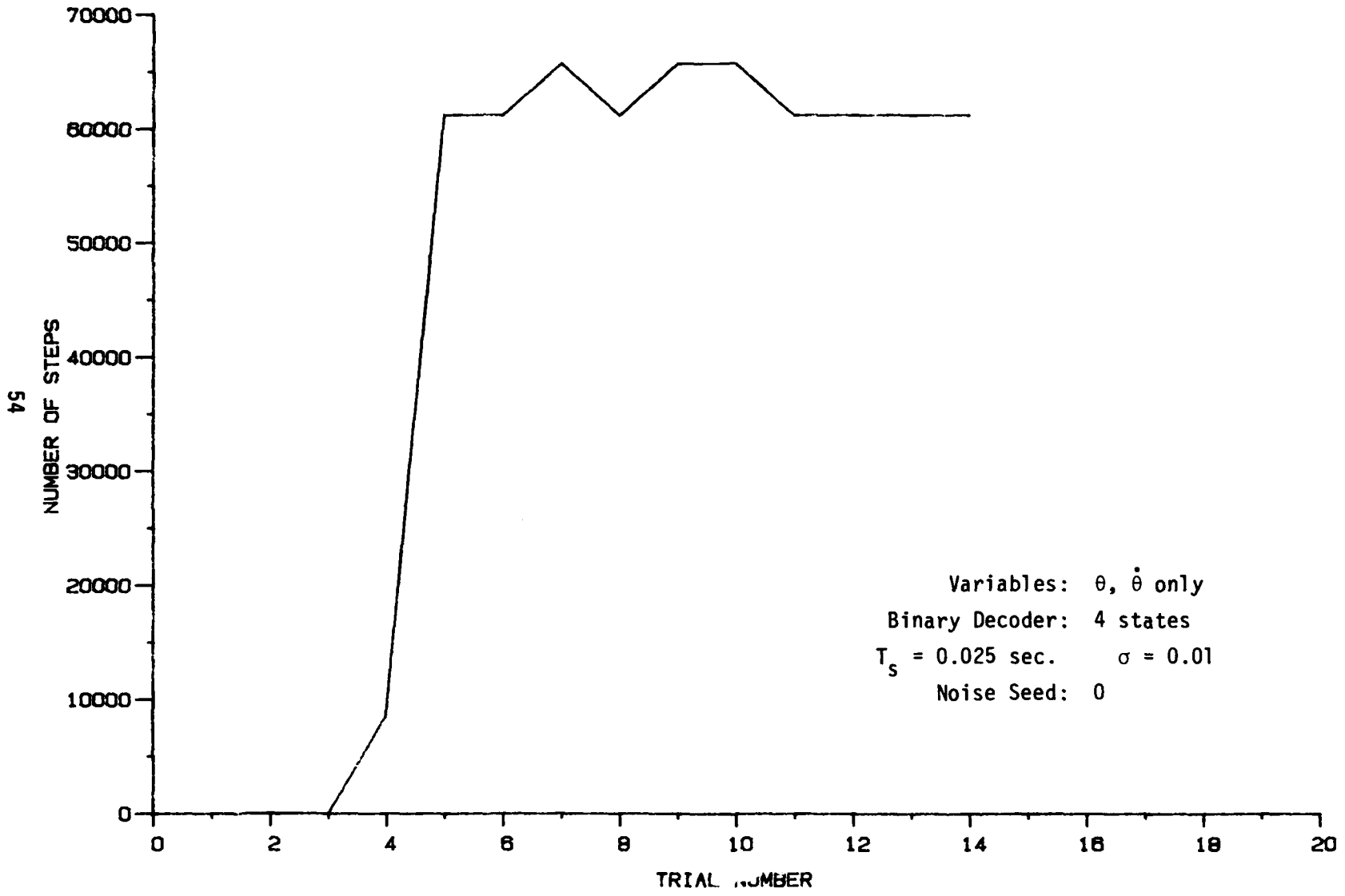


FIGURE 29. ALC LEARNING CURVE.

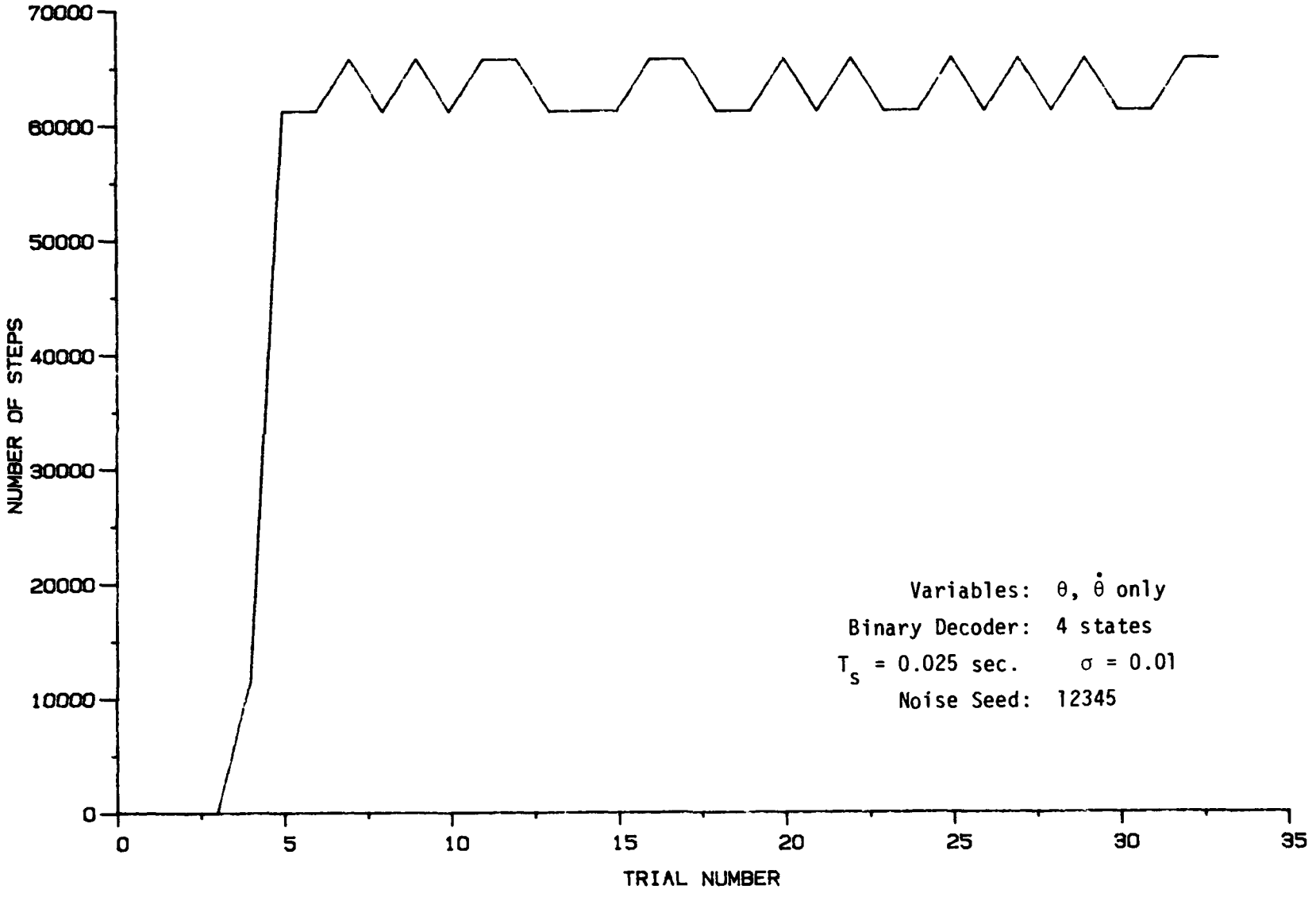


FIGURE 30. ALC LEARNING CURVE.

the time interval between successive decisions on whether the applied force should be +F or -F.

The α, β, γ, and δ values used were those previously determined to be optimal for a step size of 0.025 seconds and a noise σ of 0.01, i.e., $\alpha = 1000.$, $\beta = 0.9$, $\gamma = 0.95$, $\delta = 0.1$ [4]. The integration interval was always set equal to the step size. Two different noise seeds (0 and 12345) were used in order to confirm to our satisfaction the validity of obtained results. All test runs were performed on the Z-100 microcomputer using the POLE2 program.

Table 1 shows the number of trials that were completed for each of these combinations and indicates whether or not learning has occurred. Unfinished test runs are the result of excessively long execution times i.e., a large number of steps per trial, but not long enough to satisfy the learning criterion. The average rate of execution of the POLE2 program is approximtely 15,000 steps per hour.

The results which are summarized in Figures 31 to 33 essentially substantiated our predictions. A low noise level does indeed help the ALC to learn for all step sizes. Without noise input the ALC operation is completely deterministic. It starts always with a plus one output and after it learns to control the system for a considerable length of time, eventually it gets into a pattern of successive states leading to failure and cannot get out of it. A high noise level, on the other hand (i.e., $\sigma \geq 10$) does not allow the system to settle into a stable cycle of states and stay there. With high probability a noise spike will throw the system out of the cycle, thus causing the system to wander.

The average values of the runs made with the two different noise seeds are shown in Figure 31. It is seen that for $T_s = 0.025$ sec and $T_s = 0.05$ sec. the results are almost identical. As T_s increases the time for the ALC to learn increases in general for all σ values but it takes a marked increase for σ greater than one for all T_s. Individual learning curves for the various runs are included

TABLE 1
NUMBER OF TRIALS COMPLETED

Step Size (seconds)	Seed	Noise Sigma 0	0.001	0.01	0.1	1.	10.	100
0.010	0	38*	46	48	30	61*	343	306
0.025	0	86	52*	19*	61*	27*	230	800
	12345	same	22*	45*	48*	46*	87*	
0.050	0	38*	47*	31*	62*	230	39*	551
	12345	same	85	46*	38*	38*	82*	
0.100	0	103*	62*	76*	49*	80*	126*	330
	12345	same	84*	144*	100*	107*	136*	443
0.150	0	411	119*	241	141*	127*	265*	531
	12345	same	124*	199*	129*	136*	284	489
0.200	0	408	402	373	355	410	572	786
0.250	0	848	1000	1000	684	699	720	1000

*POLECART system has "learned".

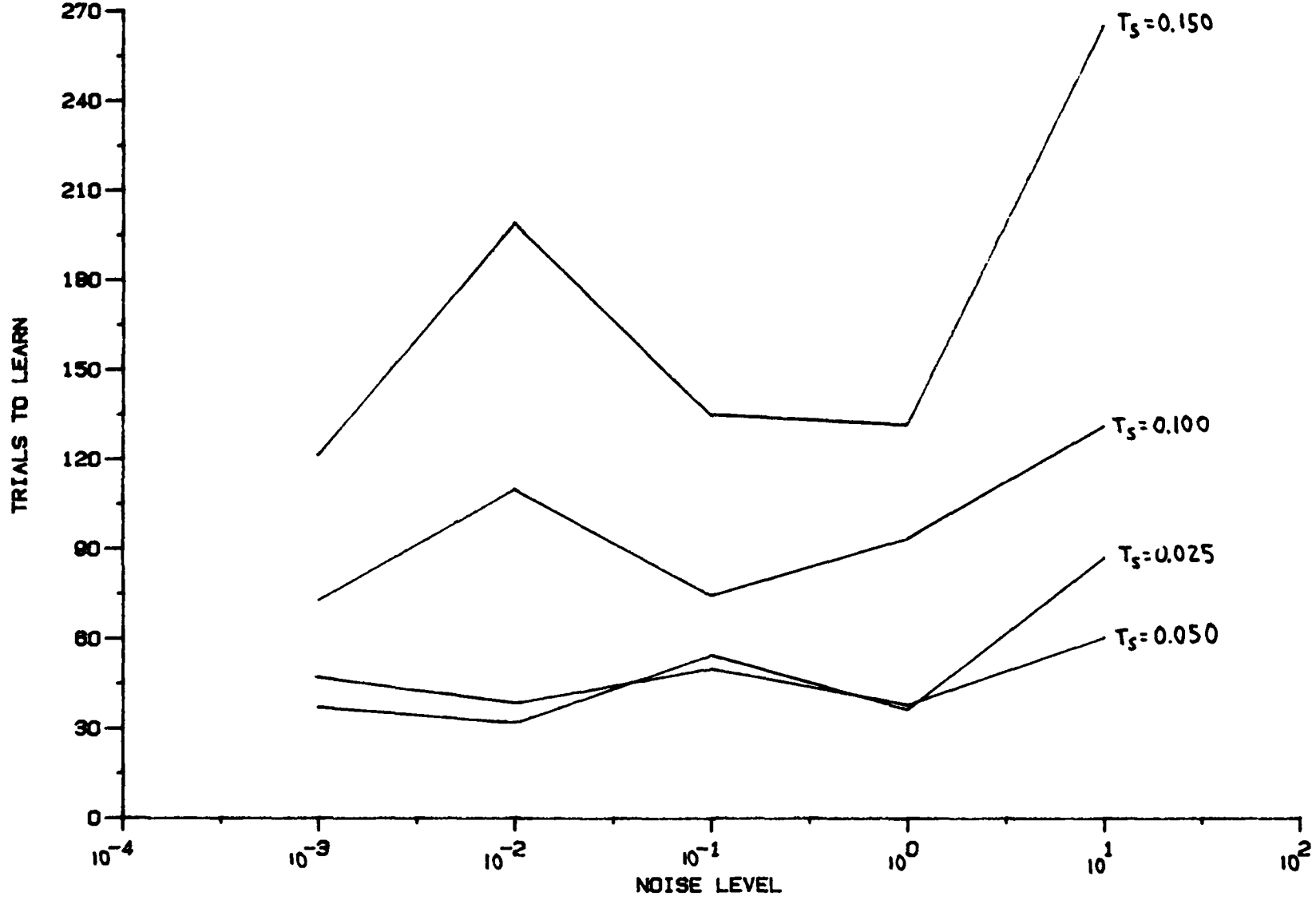


FIGURE 31. ALC TIME TO LEARN AS A FUNCTION OF NOISE LEVEL FOR VARIOUS STEP VALUES, AVERAGED OVER TWO NOISE SEED VALUES.

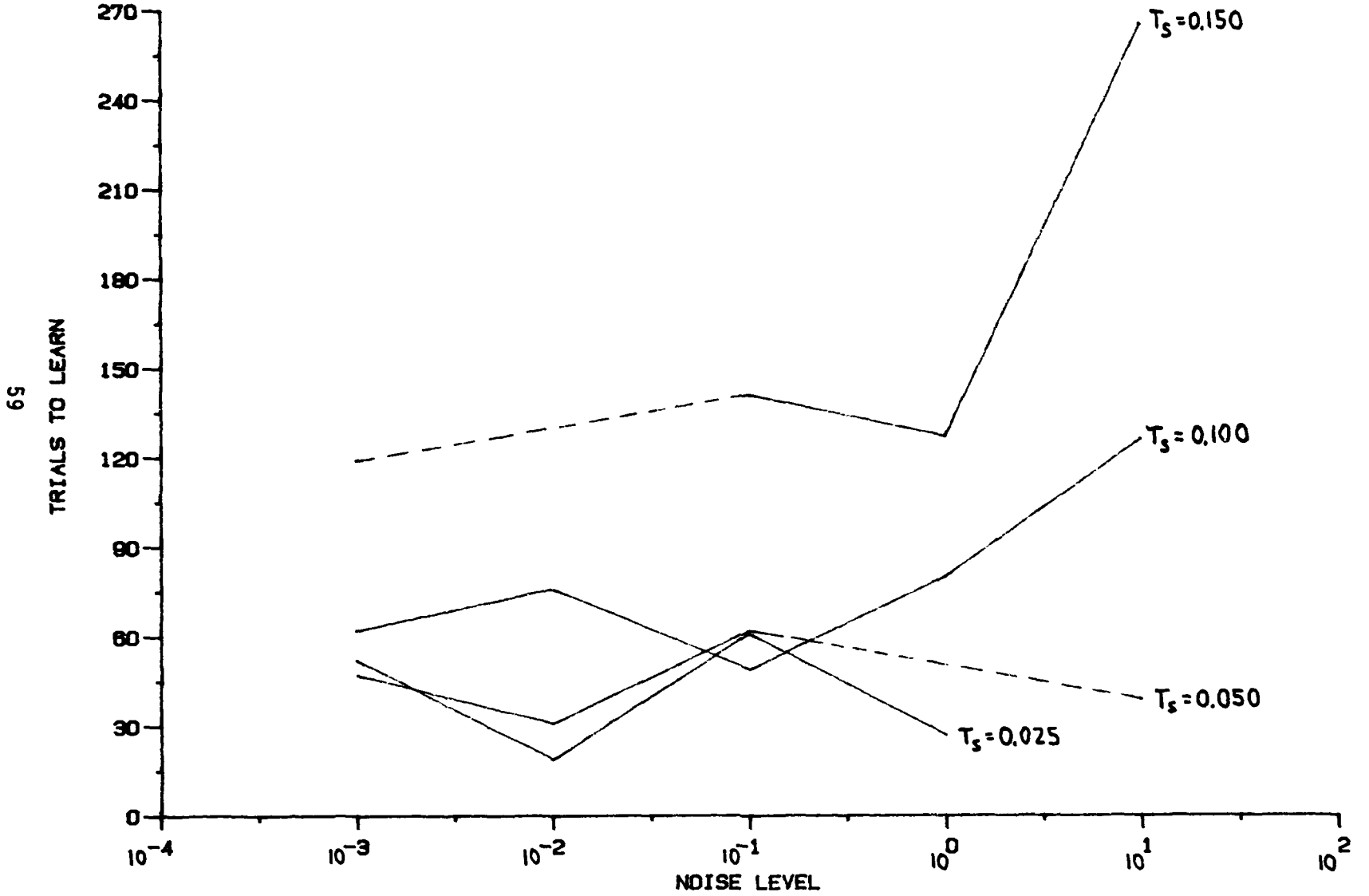


FIGURE 32. ALC TIME TO LEARN AS A FUNCTION OF NOISE LEVEL FOR VARIOUS STEP VALUES. NOISE SEED = 0.

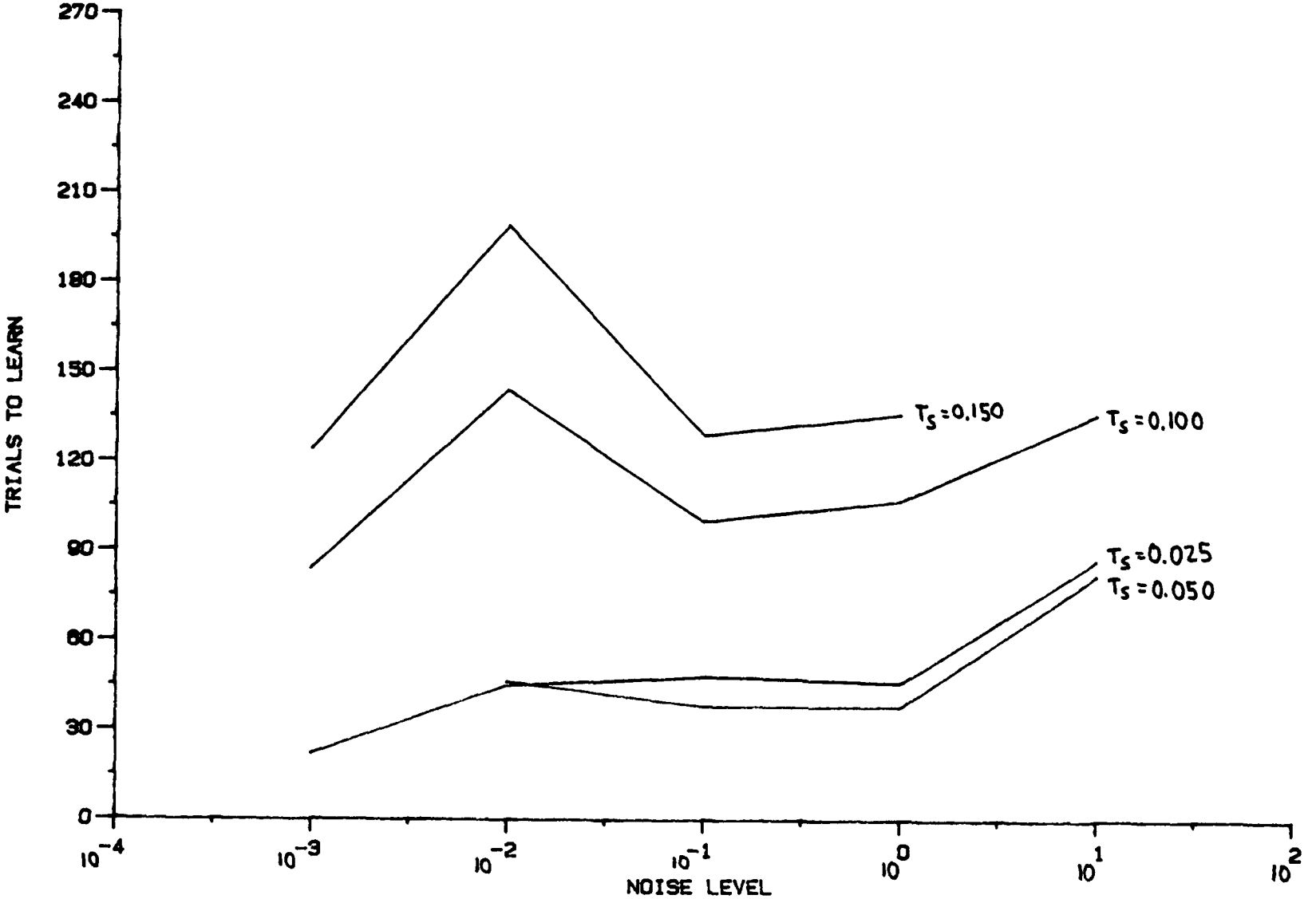


FIGURE 33. ALC TIME TO LEARN AS A FUNCTION OF NOISE LEVEL FOR VARIOUS STEP VALUES. NOISE SEED = 12345.

in Appendix C. We also note that when the step size increased to more than 0.15 sec, the ALC did not learn, as was expected.

There does not seem to be a definite σ value giving optimum performance. The range of values $10^{-2} < \sigma < 1$ give approximately equally satisfactory results. If a "best σ" value irrespective of step size had to be selected, however, $\sigma = 0.01$ would be the choice for the system parameter values used in these runs.

The effect of the level of noise in the ALC learning behavior depends on the value of the parameter α. A larger value of α will offset a larger noise level and allow the ALC to learn. Figures 34 and 35 show that for $\sigma = 100$ the ALC can learn to control the pole-on-cart system if α is set equal to 10^6.

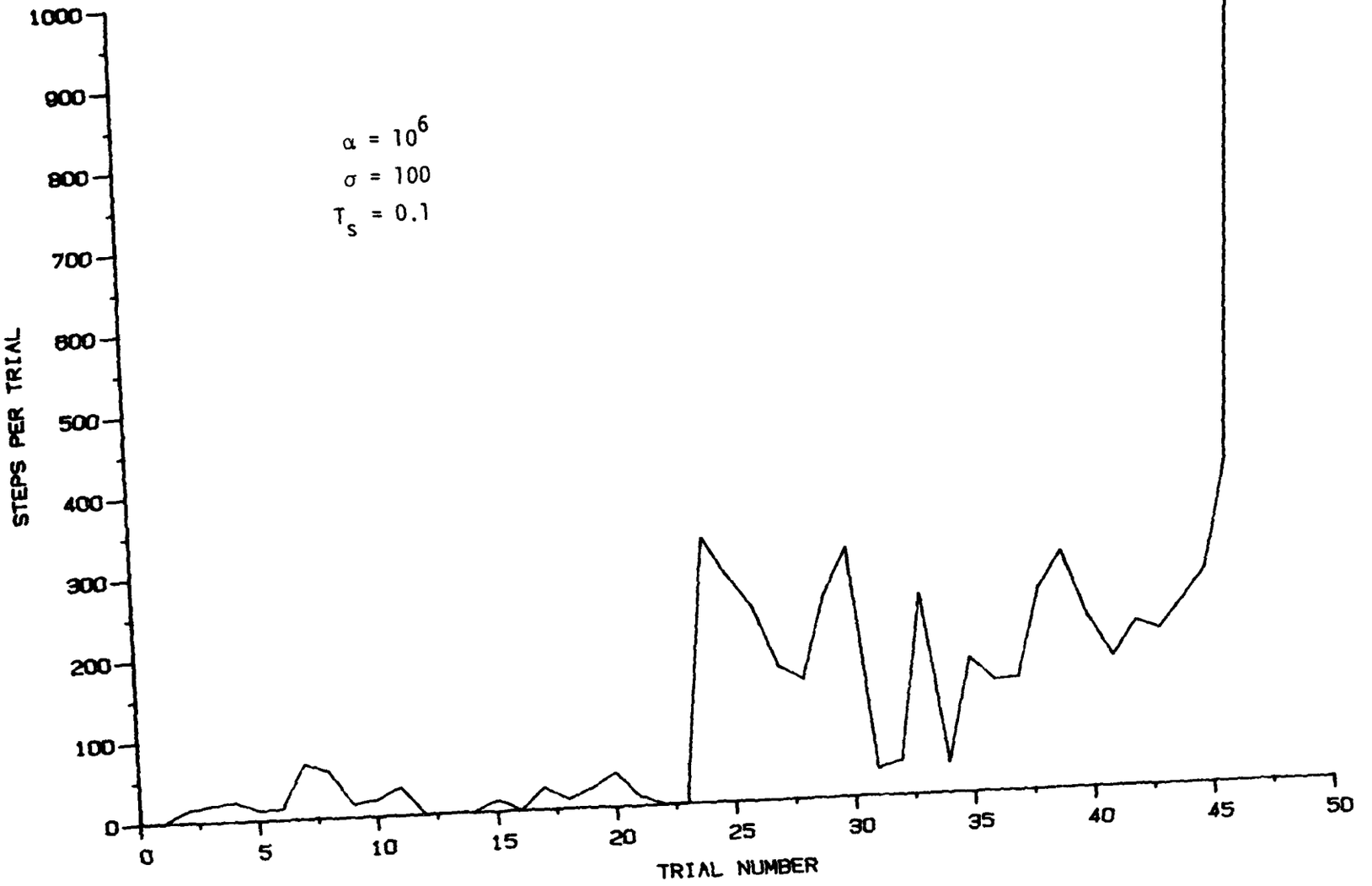


FIGURE 34. ALC LEARNING CURVE.

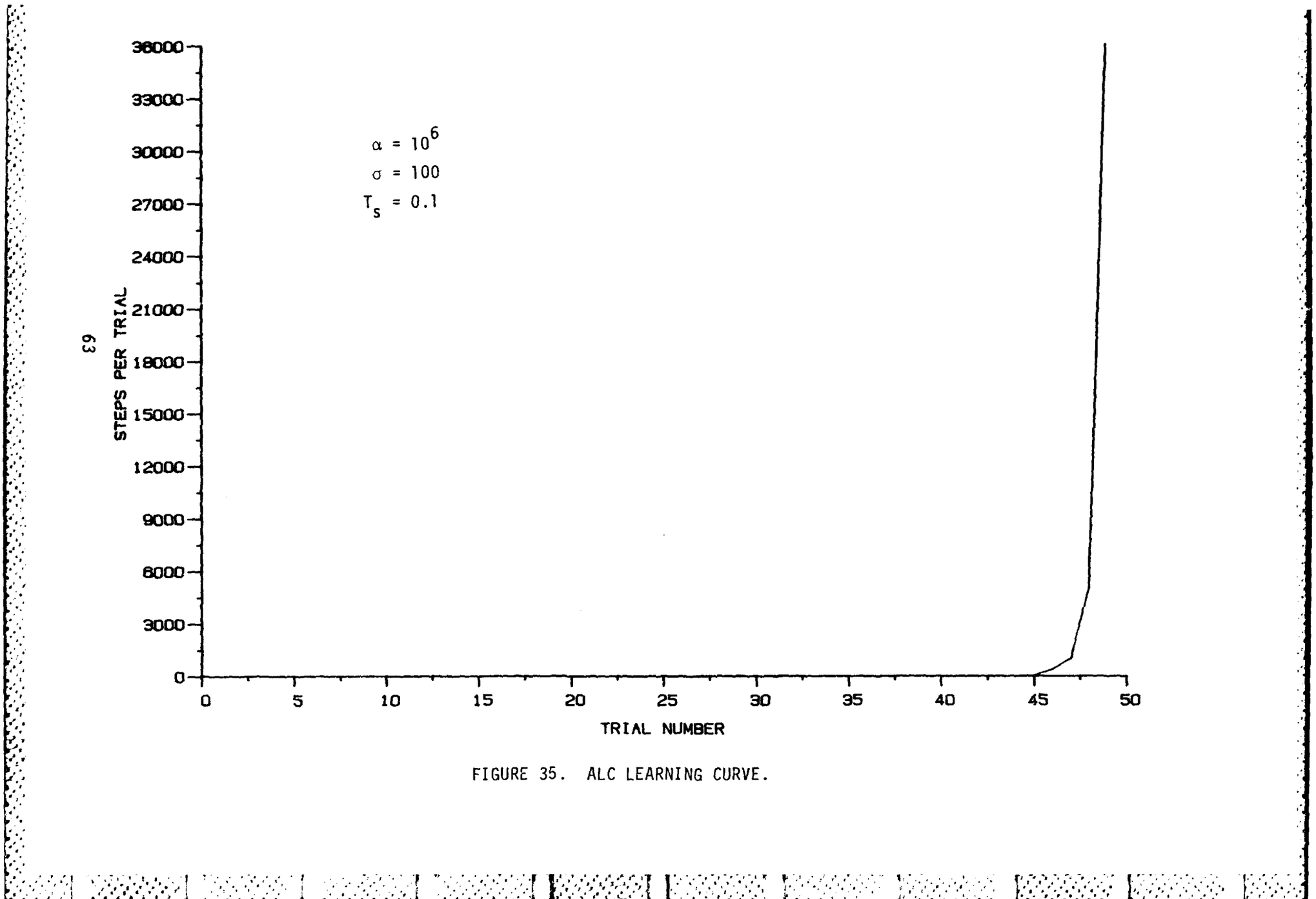


FIGURE 35. ALC LEARNING CURVE.

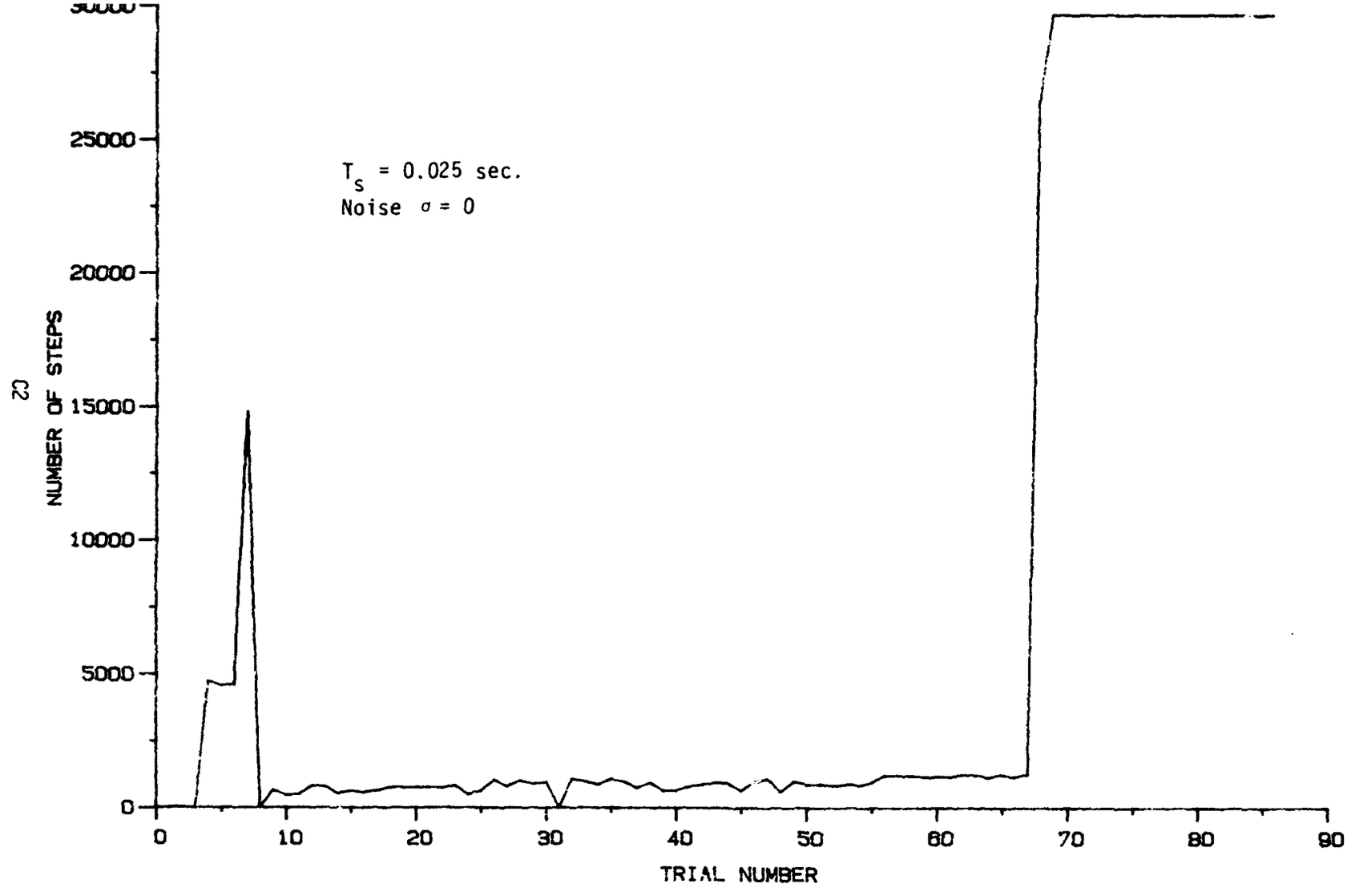
6
CONCLUSIONS

The successful application of the adaptive learning controller to the motion compensation problem of Synthetic Aperture Radar imaging has shown that the ALC can indeed operate in an engineering environment to solve practical problems. While time and funds did not allow us to make a quantitative comparison between the performance of the ALC and conventional techniques used at ERIM in SAR image processing, the usefulness of these learning algorithms has been conclusively demonstrated.

The performance of the adaptive learning controller with various levels of noise inputs was also investigated. It was found that a low level noise input improves learning behavior, while a high noise level does limit learning ability.

A limited decoder design study seemed to indicate that the learning characteristics of ALC with binary decoders are not as good as those with a larger number of states. It was also shown that faster response results when only those variables which contain information on the controlled variables are used in the decoder. This results not only in better learning rate but also in decreased computer time.

Further studies in ALC operation are expected to lead to improved system performance, ALC design optimization capability and extensive engineering application.



Figures 38. ALC Learning Curves with Parameters Indicated.

APPENDIX C

The advantage of the ALC over other possible correction methods becomes more apparent when the problem is nonlinear and/or time variant or the fundamental process involved cannot be modeled for some reason. These situations are very difficult to handle with a Kalman filter or passband type filter, while the ALC can still be applied to these problem. This characteristic of the ALC is very desirable from a SAR application standpoint. Suppose that instead of Y the power spectral density function of Y was used as an input to the ALC. The ALC would know at any instant of time how much energy is present in the portion of the motion spectrum due to motions linear, quadratic or cubic with time, those motions due to rigid body motion, structural bending motions or vibration motions and the ALC could take the correct action depending on what motion is driving the system. The ALC could also be used to cancel the acceleration bias due to gravity bias errors in a strapdown platform which involves a coordinate transformation process. This strapdown acceleration bias problem is nonlinear and a difficult one to handle with a linear passband filter or Kalman filter. These more difficult problems are more important for tactical systems or systems that use low performance motion sensing equipment.

APPENDIX B
ALC AS HIGH PASS FILTER OR KALMAN FILTER

A correction similar to the one being made by the ALC could be made using a Kalman filter or a high pass filter.

A Kalman filter could be designed to estimate the velocity bias in Y, or Y could be differentiated to give Y velocity, passed through a high pass filter to remove the average velocity and then integrated to give a new estimate of Y without any velocity bias. The high pass filter is an undesirable approach since it is difficult to obtain the sharp cutoff required for this application, the filter adds undesirable time lags over the low frequency region where errors have the greatest effect on system performance and differentiation amplifies high frequency noise. A Kalman filter could be used but it involves a major design effort, is computationally complex and can have stability problems. Also, if the task description were changed in order to improve overall performance and became very nonlinear which creates no fundamental problem to the ALC, it might become impossible to use a high pass filter or Kalman filter.

Consider a second order least squares algorithm, which under the right simplifying assumptions is a Kalman filter. Assume that Y is a sine wave with amplitude just below the Y bound so that it never crosses the bound. The ALC will make no correction and will give a velocity bias estimate of zero. The least squares algorithm will give a velocity bias estimate that will oscillate about zero and have some peak value. The ALC gives the correct value while the least squares approach gives a correct value only occasionally. This illustrates one basic difference between the ALC and the least squares approach which is important to the SAR motion compensation problem. Determining other ways in which the two approaches differ requires a more comprehensive tradeoff analysis.

By first ignoring the constant phase error terms, and lumping the identical quadratic and linear terms into the two signal histories, and after Fourier transforming, two low-resolution images are produced. These images look almost the same, but they are displaced in both dimensions relative to each other. Letting S denote the "common" undetected image, the image derived from the left subaperture is approximately $|S(f_x + 2ax_L, f_y + cx_L)|^2$, and that derived from the the left is $|S(f_x - 2ax_R, f_y - cx_R)|^2$. By measuring the amounts of x and y misregistration, estimates a and c may be obtained. If a common x-center had instead been chosen for the two sub-phase histories, then estimates of b and c would become available. If the two patches had neither common x nor y centers, shifts in f_x and f_y would both depend upon two of the three coefficients, and the contributions from each could not be easily determined. The reliability of the estimates can be improved by averaging estimates obtained from several sub-phase history pairs.

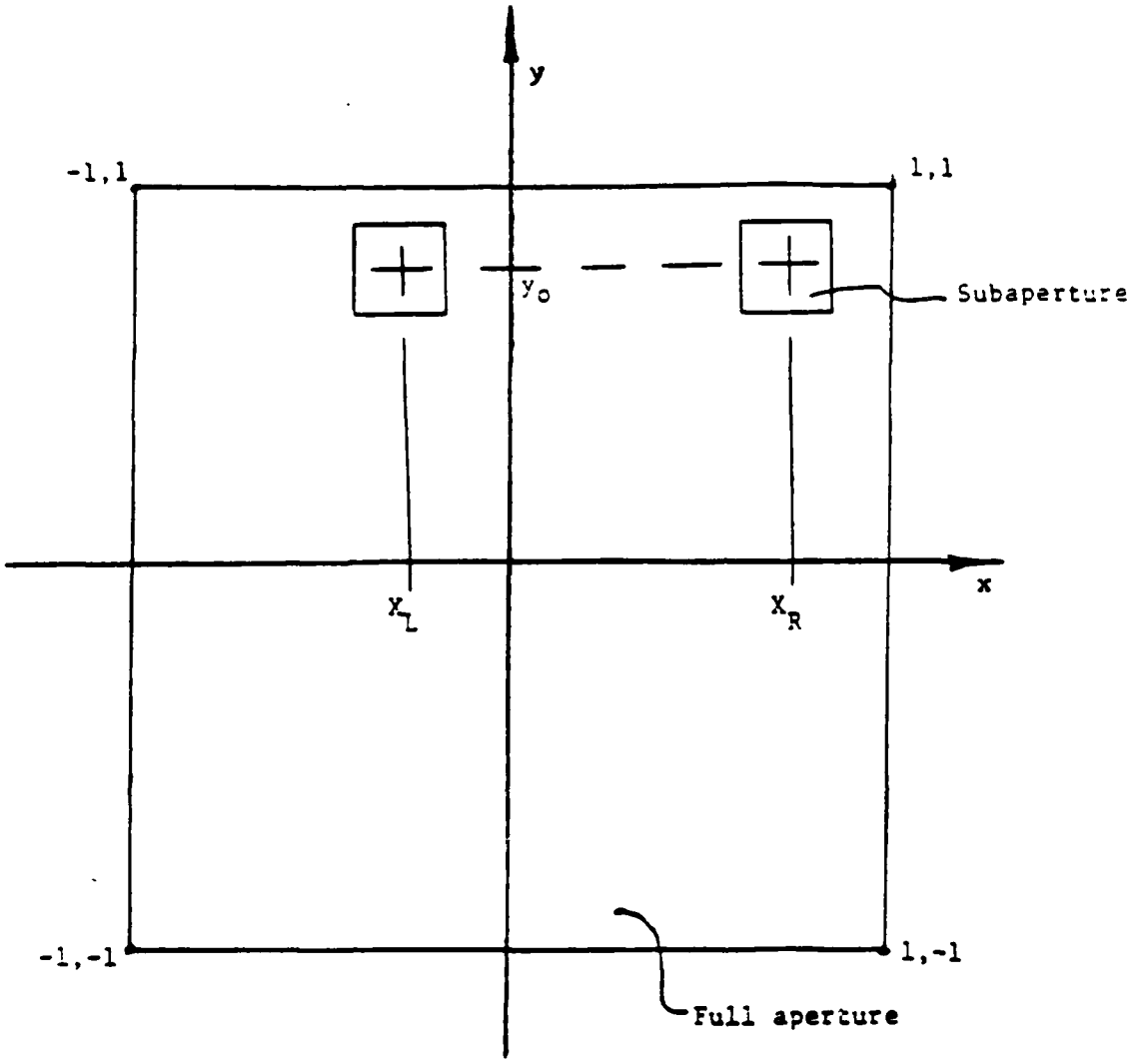


FIGURE 37. LOCATION OF TWO SUB-PHASE HISTORIES TO BE UTILIZED IN ESTIMATING MISFOCUS COEFFICIENTS A AND C.

where the shifts result from quadratic phase error-induced linear phase terms of varying sign and magnitude. To detect phase errors of the form $\phi(x, y) = ax^2 + by^2 + cxy$, we will see below that an estimate of the two-dimensional misregistration between two images is required.

The two-dimensional quadratic focusing problem can be stated as one of obtaining reliable estimates of the misfocus coefficients, a, b, and c, where the phase error is assumed to be of the form $\phi(x, y) = ax^2 + by^2 + cxy$, for $x, y \in [-1, +1]$. Some type of map-drift procedure should be used to find a, b, and c.

In the spirit of the 1-D technique, a 2-D approach suggested by Figure 37 is taken. Two equal-size sub-phase histories centered at $(-x_L, y_0)$ and (x_R, y_0) are culled from the unfocused 2-D phase history. Each of these small patches is translated in x and y to become centered about the origin and Fourier transformed and detected. The translational shifts between the resulting low-resolution images are then used to estimate some of the misfocus coefficients.

Examining the phase errors across the shifted sub-phase histories note that

$$\phi_L(x, y) \stackrel{d}{=} \text{phase error across shifted left sub-aperture history}$$

$$= \phi(x - x_L, y + y_0)$$

$$= ax^2 + by^2 + cxy + \text{constants} - 2ax_Lx + 2by_0y - cx_Lx + cy_0x$$

Similarly,

$$\phi_R(x, y) \stackrel{d}{=} \text{phase error across shifted right sub-phase history}$$

$$= \phi(x + x_R, y + y_0)$$

$$= ax^2 + by^2 + cxy + \text{constants} + 2ax_Rx + 2by_0y + cx_Ry + cy_0x.$$

$$\mathcal{F}\{s_L(x)\} = \mathcal{F}\{s(x - \tfrac{1}{2})\}$$

$$= \int_{-1/2}^{+1/2} s_o(x - \tfrac{1}{2})\, e^{j2\pi a(x - \frac{1}{2})^2}\, e^{-j2\pi fx}\, dx$$

$$= c_L \int s_o(x - \tfrac{1}{2})\, e^{j2\pi ax^2}\, e^{-j2\pi(f+a)x}\, dx, \quad |c_L| = 1,$$

and

$$\left|\mathcal{F}\{s_L(x)\}\right|^2 = \left|\mathcal{F}\{s_o(x - \tfrac{1}{2})\, e^{j2\pi ax^2}\}_{f+a}\right|^2.$$

Similarly, the right-hand phase history subaperture receives the following operations:

$$\left|\mathcal{F}\{s_R(x)\}\right|^2 = \left|\mathcal{F}\{s_o(x + \tfrac{1}{2})\, e^{j2\pi ax^2}\}_{f-a}\right|^2$$

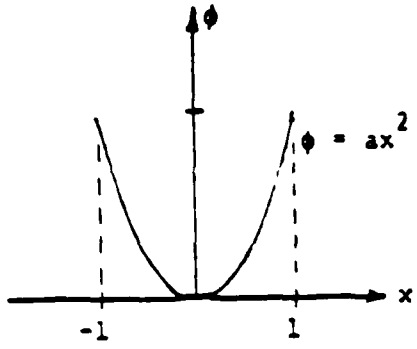
The following approximation can now be made:

$$\left|\mathcal{F}\{s_o(x - \tfrac{1}{2})\, e^{j2\pi ax^2}\}_f\right|^2 \cong \left|\mathcal{F}\{s_o(x + \tfrac{1}{2})\, e^{j2\pi ax^2}\}_f\right|^2; \; x\epsilon[-\tfrac{1}{2} - \tfrac{1}{2}],$$

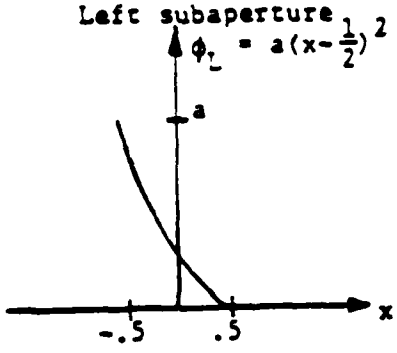
which is heuristically justified by observing that the images produced from two short pieces of a phase history look very much the same, and that the misfocus term will smear all reflectors about the same. Thus, the two images $|\mathcal{F}\{s_L(x)\}|^2$ and $|\mathcal{F}\{s_R(x)\}|^2$ are nearly shifted versions of one another, and the amount of relative shift is dependent only on the misfocus coefficient, a. Various cross-correlation techniques might then be used to estimate this misregistration.

Two-Dimensional Mapdrift Focusing Algorithm

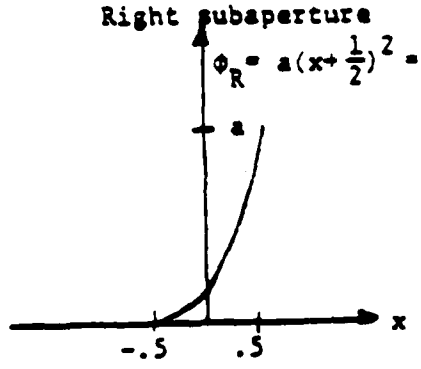
The mapdrift focusing approach described above can be stated more generally as a technique which detects phase errors by exploiting relative shifts among images produced from multiple subapertures,

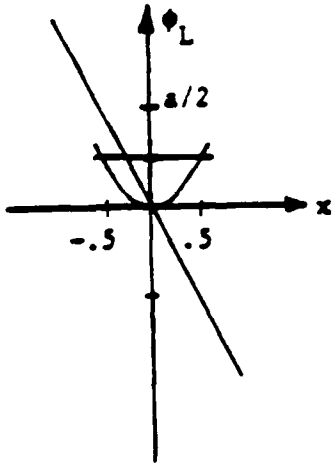


Left subaperture $\phi_L = a(x-\frac{1}{2})^2 = ax^2-ax+a/4$



Right subaperture $\phi_R = a(x+\frac{1}{2})^2 = ax^2+ax+a/4$





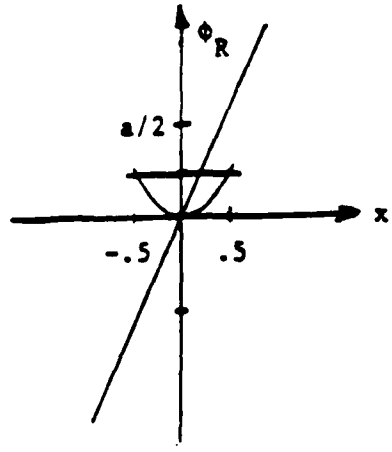


FIGURE 36. MAPDRIFT PRINCIPLE.

This is a satisfactory model of the azimuth phase history when the primary phase error is due to unknown cross beam velocity errors. The task is to estimate a in order to remove the phase error later by multiplying the phase history by the conjugate of the phase error. The tasks required to generate an estimate are:

1. Split the interval [-1, +1] into two intervals [-1, 0] and [0, +1], and shift the signals on these intervals so as to become centered about the origin.
2. Fourier-transform and detect (i.e., image) each of the short phase histories.
3. Derive an estimate of the quadratic misfocus by measuring the relative shift between the resulting two low-resolution images.

The objective of the splitting and shifting is to induce phase errors across two subapertures with linear terms differing in sign (and possibly magnitude). For the situation outlined above and diagrammed in Figure 36, the linear phase error terms are seen to be -ax and +ax for the left and right subapertures, respectively. Since the effect of multiplying a phase history by a linear term is to shift the corresponding image left or right according to the sign of the term and by an amount proportional to the term's magnitude, the low-resolution images produced from the sub-phase histories are shifted in opposite direction by an amount proportional to 2a.

The desired sequence of operations is described algebraically by first forming the left image from the shifted phase history:

$$s_L(x) \triangleq \begin{cases} s(x - \frac{1}{2}), & x \in [-\frac{1}{2}, +\frac{1}{2}] \\ 0 \text{ otherwise} \end{cases}$$

The magnitude of the quadratic phase error due to a cross beam velocity measurement of 1 percent would be 13.3 cycles, assuming a range of 15 statute miles, a 0.1 ft wavelength and a 3 ft resolution. This is figured mid-aperture to first null for a uniformly weighted aperture. A 1 percent velocity error is consistent with the specified accuracy for the velocity measurement generated by the LTN-51 INS of ±(3 fps + 0.5 fps/hr).

A.2 AUTOFOCUS DESCRIPTION

The purpose of the autofocus algorithm in SAR image processing is to estimate the coefficients of the two-dimensional quadratic phase error present in phase histories. The quadratic phase error can be expressed as

$$\phi(x, y) = ax^2 + by^2 + cxy$$

where x, y are phase history coordinates, and a, b, c are constants to be determined. Two methods that are frequently used to estimate these coefficients will be now described briefly.

1-D Mapdrift Method

Suppose the time-limited phase history can be expressed as a product of factors as shown:

$$S(x) = S_o(x)\ e^{j2\pi ax^2}$$

where

$S_o(x)$ is an unperturbed signal

and

$e^{j2\pi ax^2}$ is a quadratic phase error

$x \in [-1, 1]$

Assume that the antenna has a physical aperture L

$$L = 2\rho$$

where

$$\rho = \text{azimuth resolution}$$

If the antenna aperture is uniformly weighted, the null to null antenna beamwidth ψ is

$$\psi = 2\lambda/L$$

$$= \lambda/\rho$$

The time interval Δt over which a target is illuminated is

$$2\Delta t = \psi r/v \sin\theta$$

$$= \frac{\lambda r}{\rho v \sin\theta}$$

The total phase shift across half an aperture centered on broadside

$$\phi = \frac{r(\Delta t)^2}{\lambda}$$

$$= \frac{v^2}{\lambda R_o}\left(\frac{\lambda R_o}{2\rho v}\right)^2$$

$$= \frac{\lambda R_o}{4\rho^2} \quad \text{cycles}$$

If there exists a cross beam velocity measurement error, ξ_v, and an along beam acceleration measurement error, ξ_a, the phase error across half an aperture is

$$\phi_e = \frac{\lambda R_o}{4\rho^2}\left(2\frac{\xi_v}{v} + R_o\frac{\xi_a}{v^2}\right) \text{cycles}$$

APPENDIX A
CONVENTIONAL AUTOFOCUS

A.1 SOURCES OF QUADRATIC PHASE ERROR

The quadratic phase errors present in azimuth phase histories are the indirect result of dc errors in the measurement of along beam acceleration. Along beam acceleration is a function both of radar vehicle acceleration and cross beam velocity. In the absence of vehicle accelerations the range r to scene center is

$$r = \sqrt{R_o^2 + (vt)^2}$$

where

R_o = range at broadside

v = vehicle velocity

t = 0 at broadside

$$\dot{r} = \frac{v^2 t}{r}$$

$$\ddot{r} = \frac{v^2}{r} - \frac{v^4 t^2}{r^3}$$

$$= \frac{v^2}{r} \sin^2 \theta$$

where

θ = squint angle relative to vehicle track

$$\text{Phase } \phi = \frac{1}{2} \ddot{r} \, t^2 / \frac{\lambda}{2}$$

$$= \frac{\ddot{r} t^2}{\lambda} \text{ cycles}$$

where

λ = radar wavelength

REFERENCES

1. Barto, A.G. and R.S. Sutton, Goal Seeking Components for Adaptive Intelligence: An Initial Assessment, Technical Report AFWAL-TR81-1070, April 1981.

2. Klopf, A.H., The Hedonistic Neuron, A Theory of Memory, Learning and Intellitence, Hemisphere Publishing Corporation, 1982.

3. Barto, A.G., R.S. Sutton, and C.W. Anderson, Neuron-Like Adaptive Elements that can Solve Difficult Learning Control Problems, Computer and Information Science Department, University of Massachusetts, Report 82-20.

4. Politis, D.T. and W.H. Licata, Study on Extremizing Adaptive Systems and Applications to Synthetic Aperture Radars, October 1983, Annual Report, ERIM.

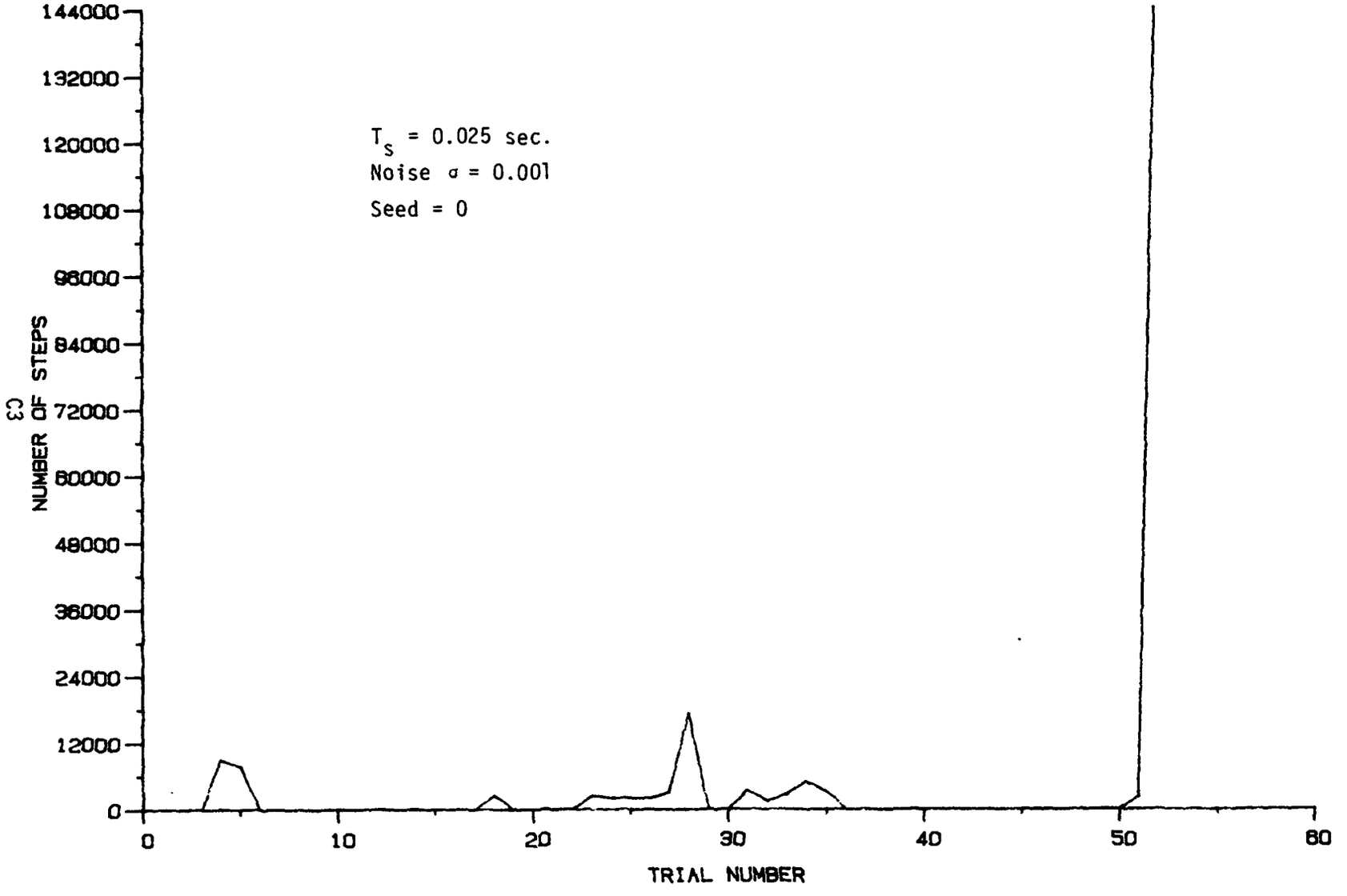


Figure 39. ALC Learning Curves with Parameters Indicated.

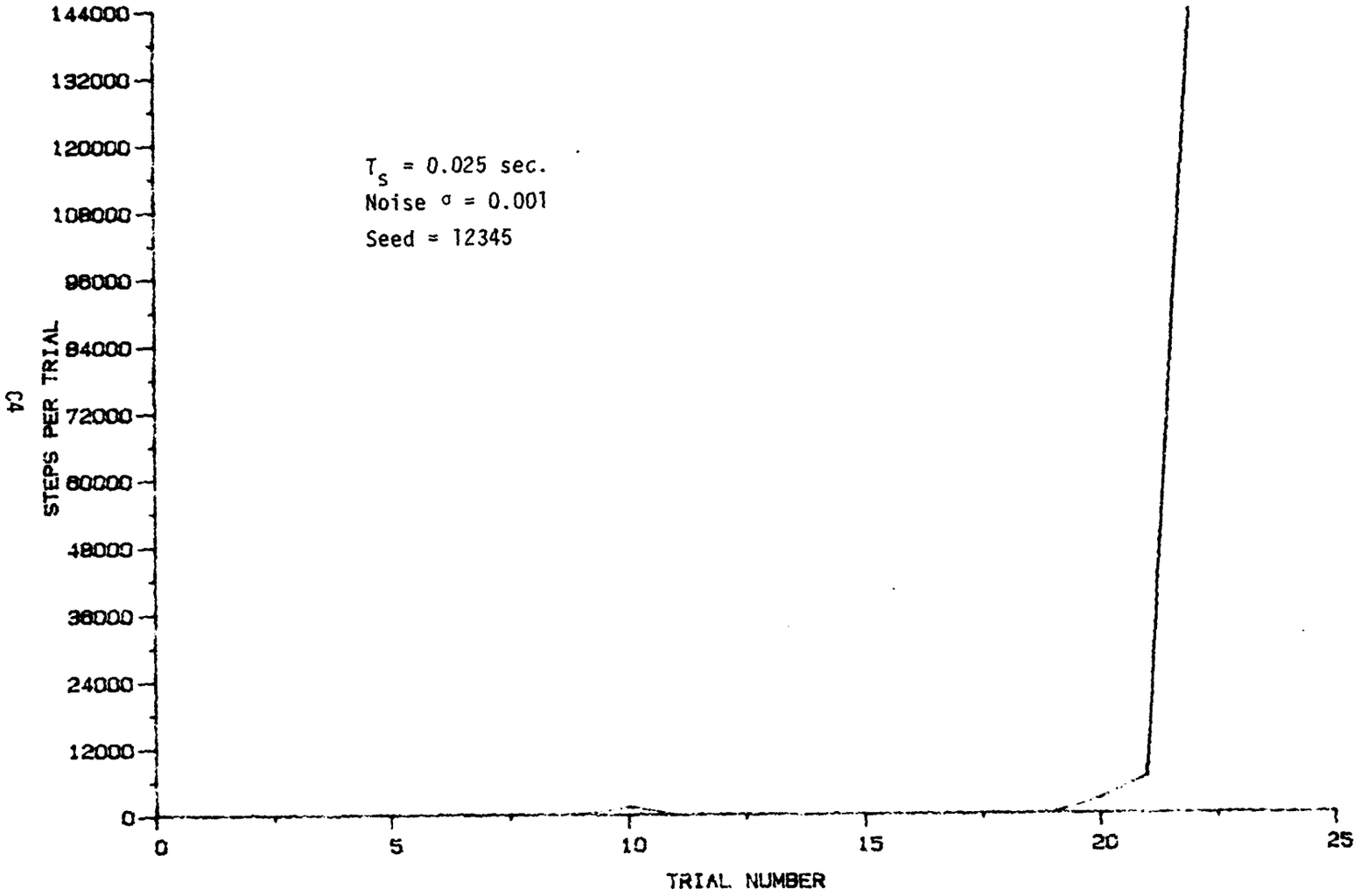


Figure 40. ALC Learning Curves with Parameters Indicated.

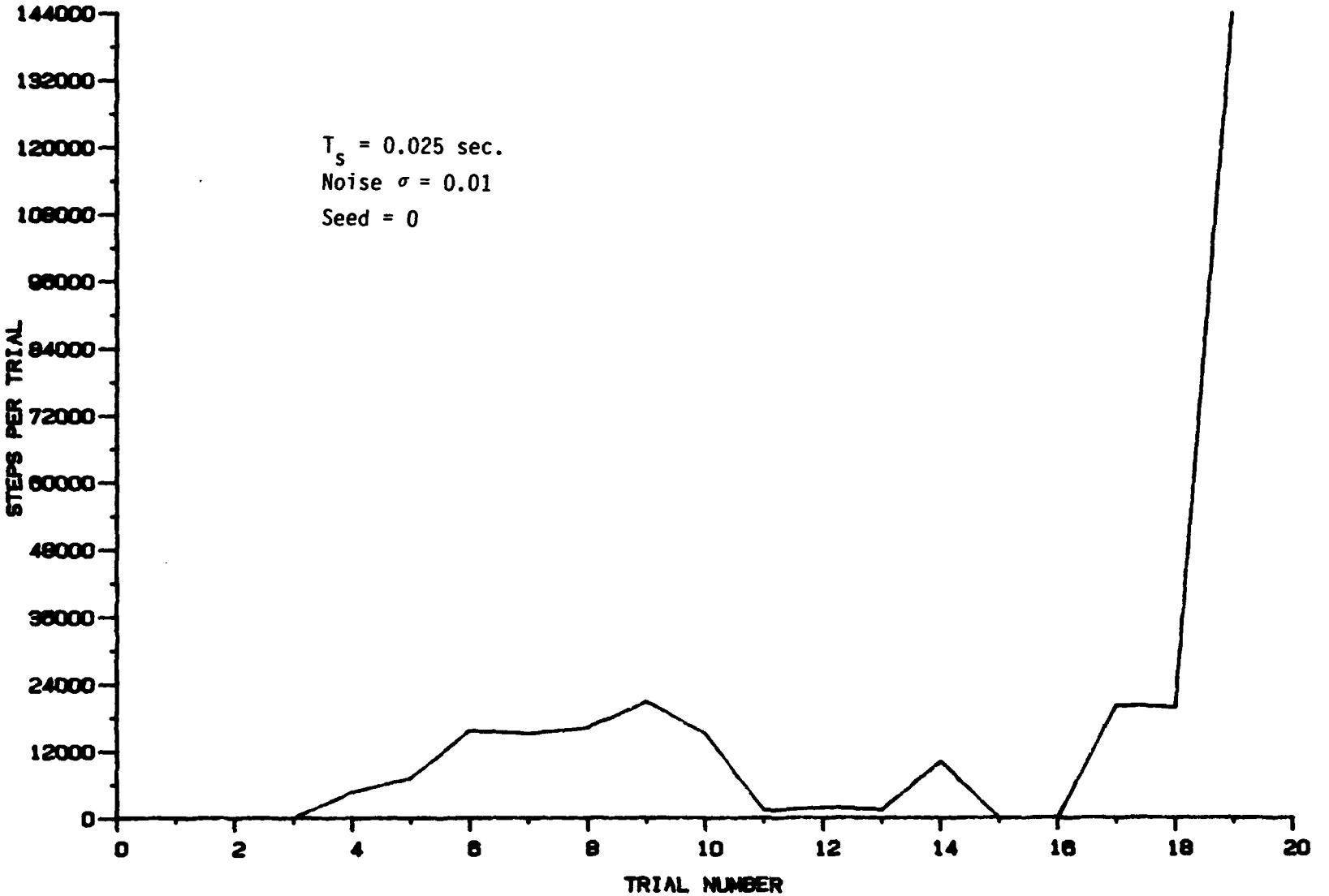


Figure 41. ALC Learning Curves with Parameters Indicated.

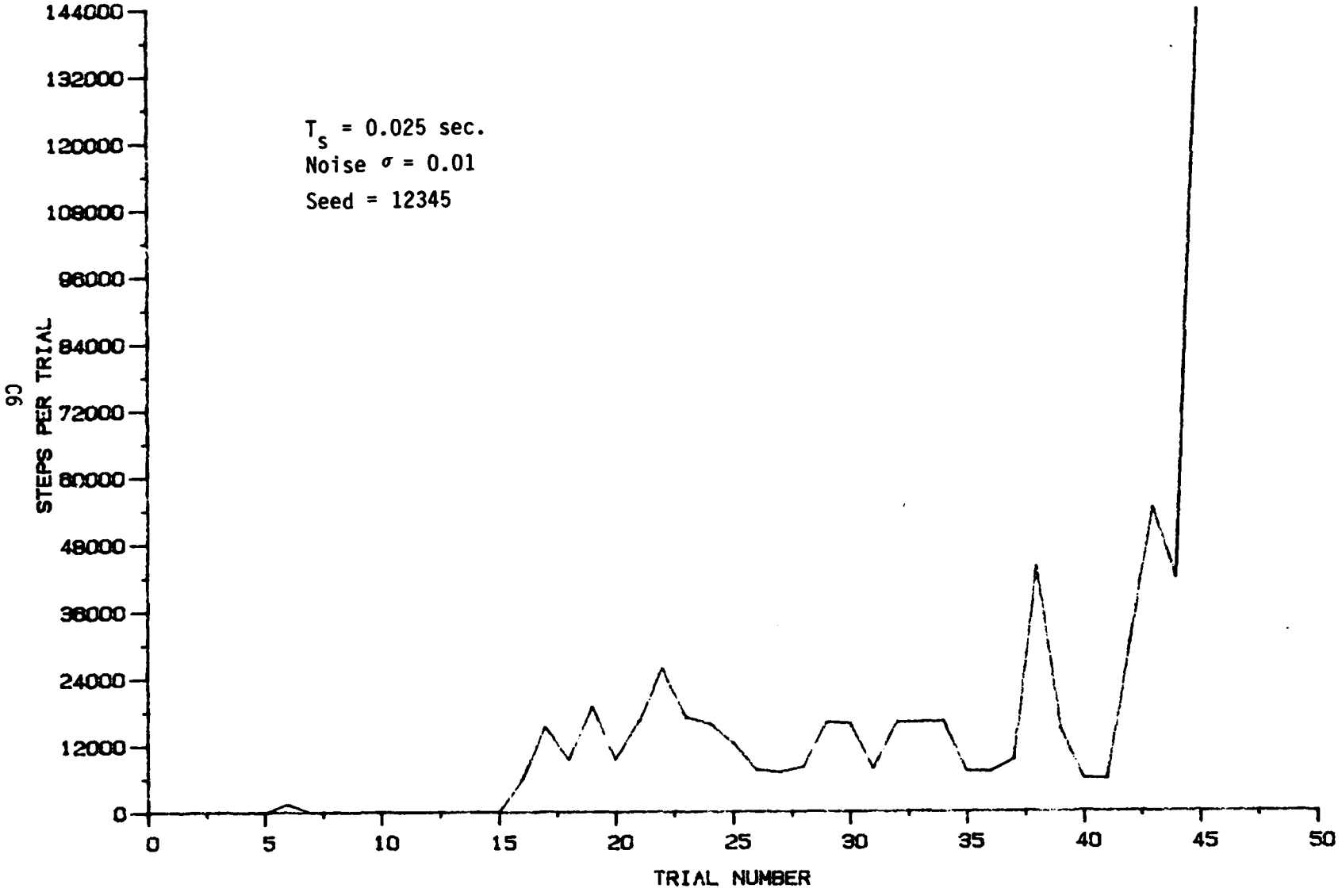


Figure 42. ALC Learning Curves with Parameters Indicated.

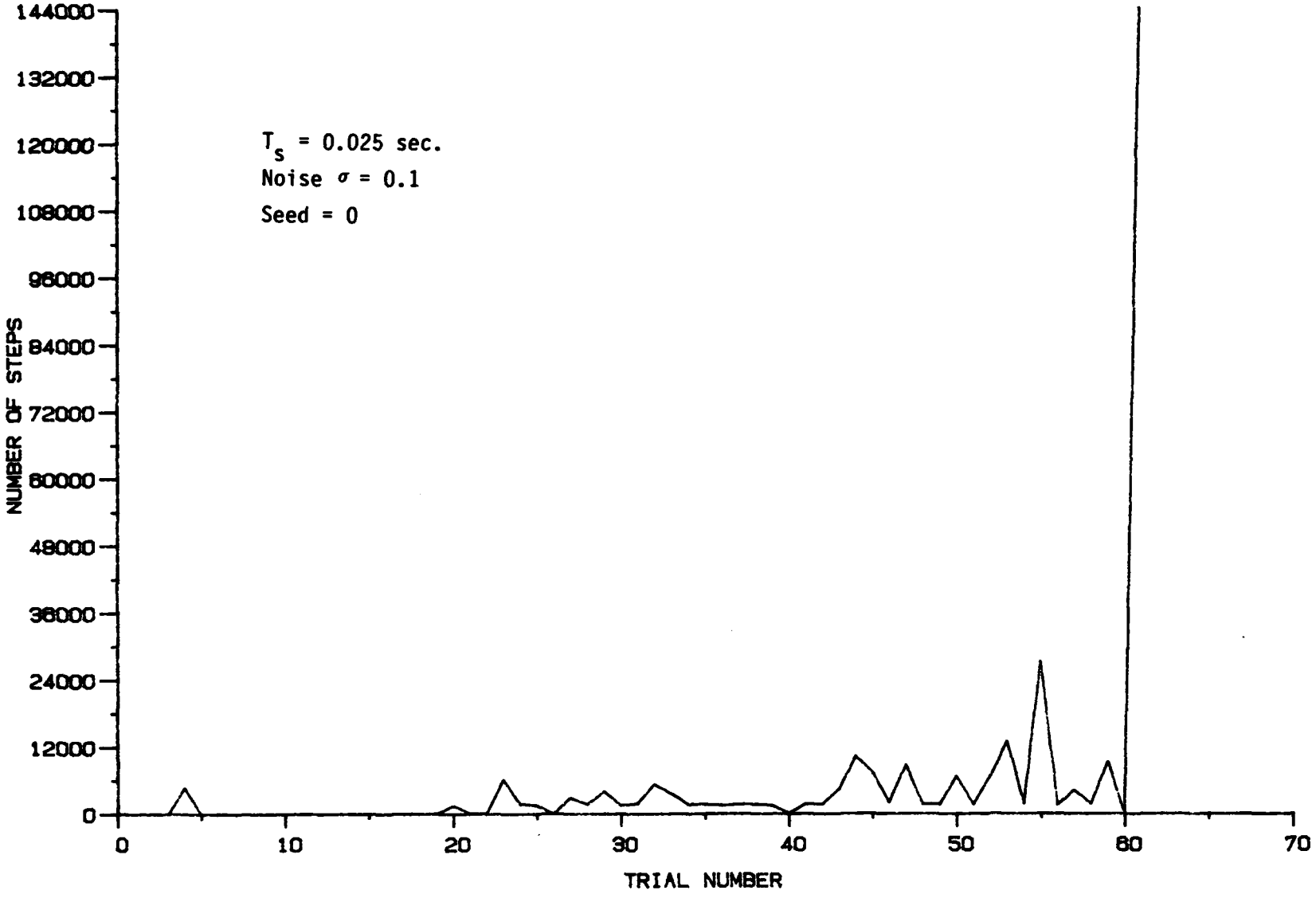


Figure 43. ALC Learning Curves with Parameters Indicated.

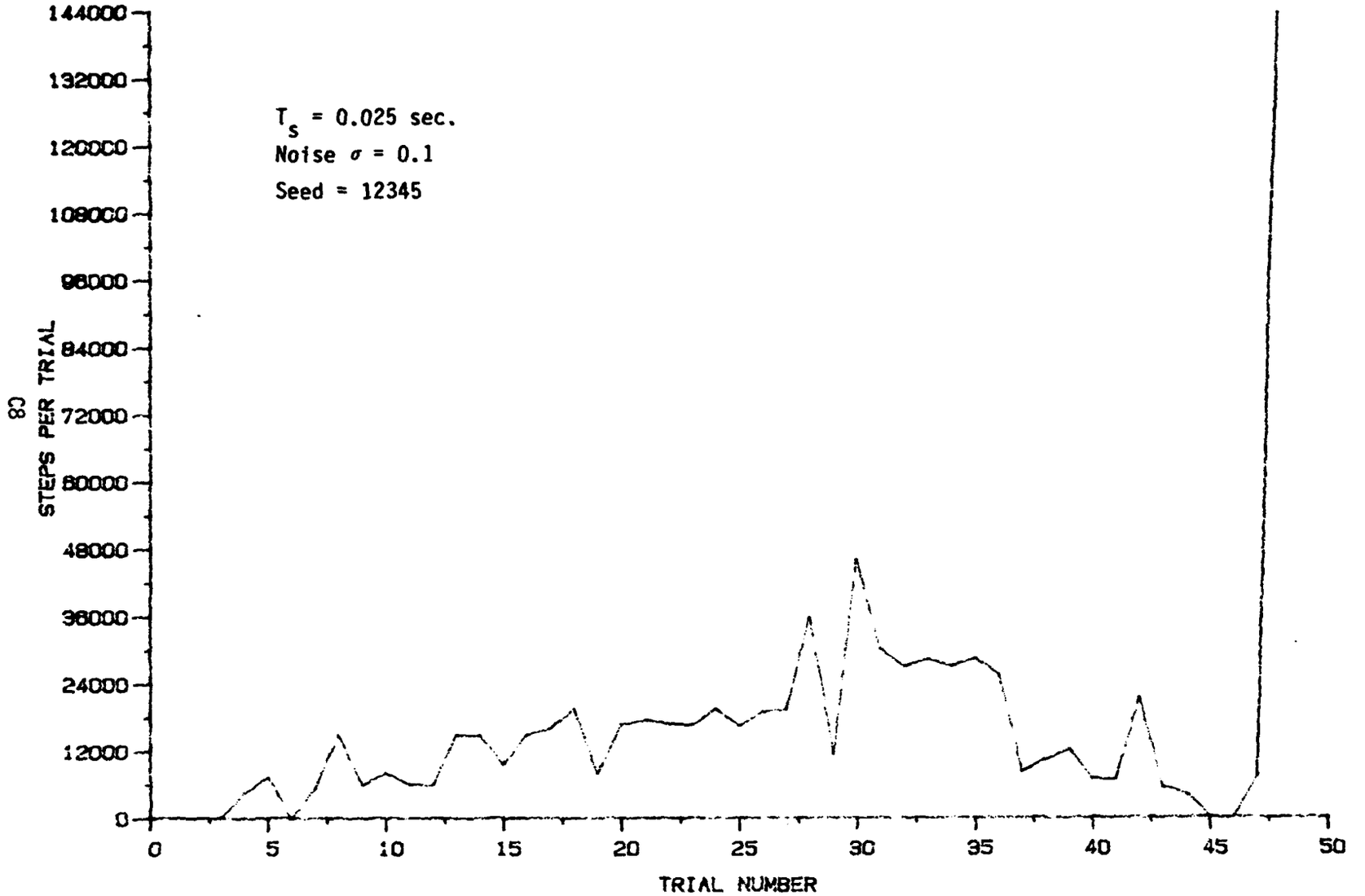


Figure 44. ALC Learning Curves with Parameters Indicated.

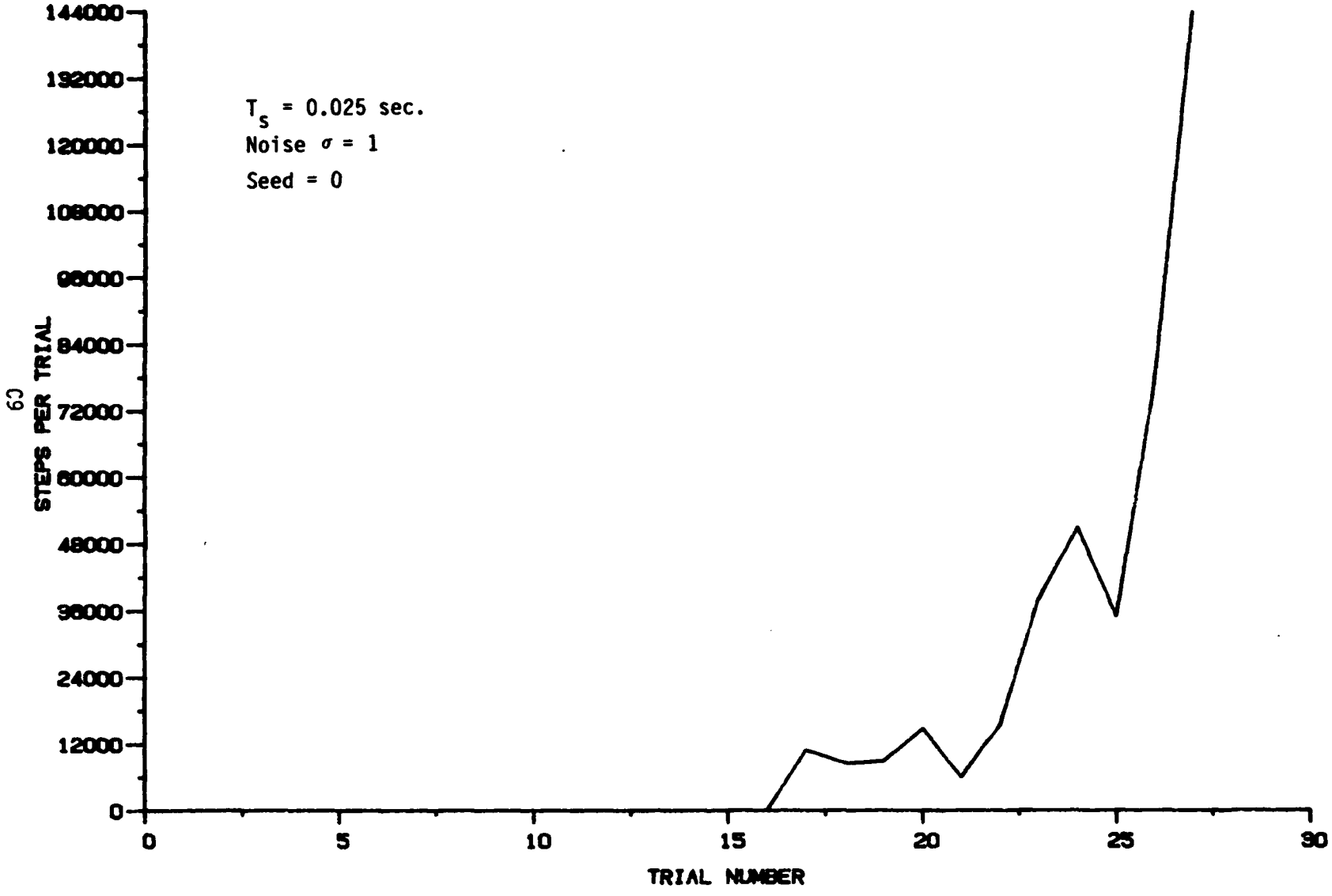


Figure 45. ALC Learning Curves with Parameters Indicated.

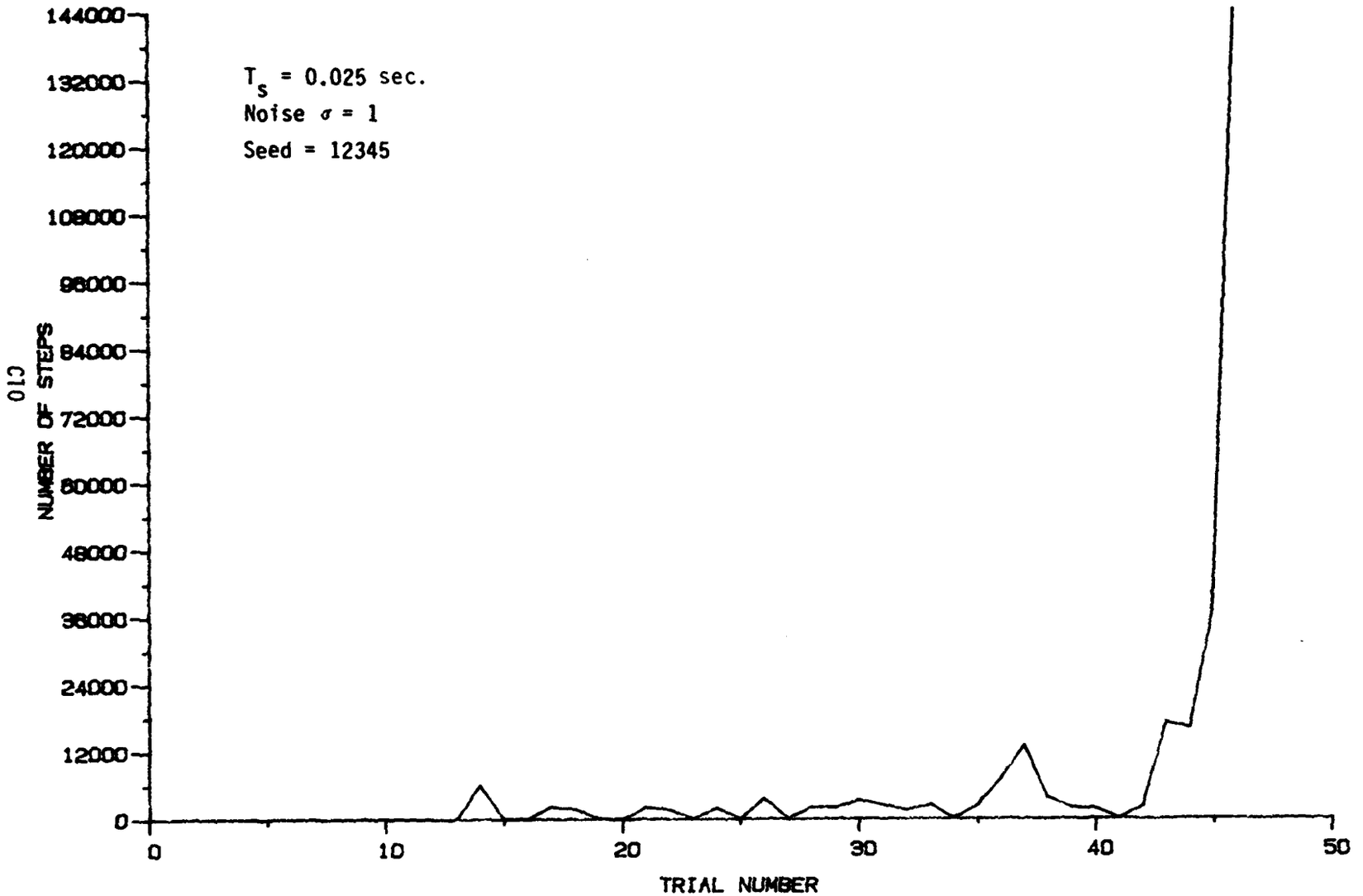


Figure 46. ALC Learning Curves with Parameters Indicated.

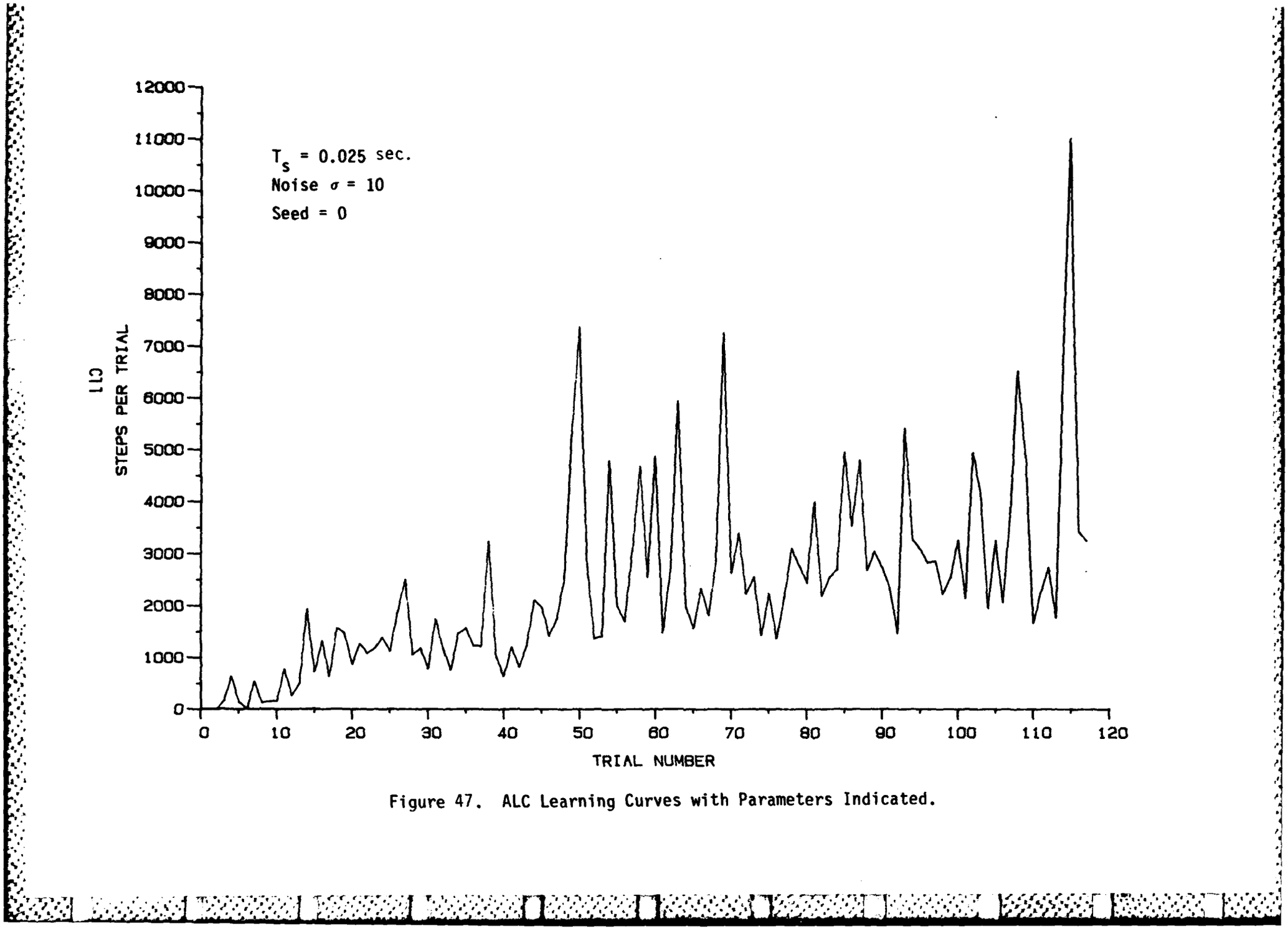


Figure 47. ALC Learning Curves with Parameters Indicated.

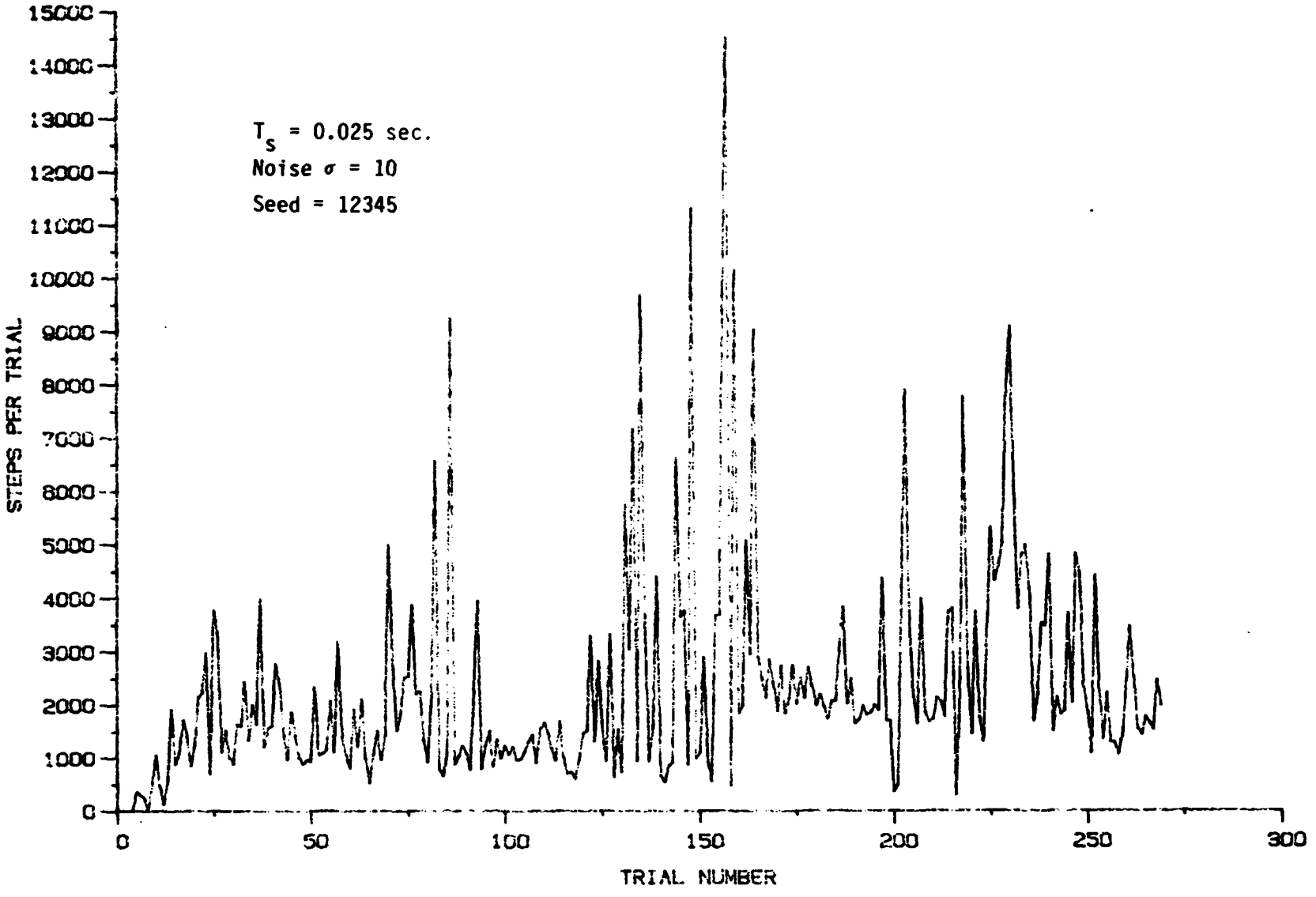


Figure 48. ALC Learning Curves with Parameters Indicated.

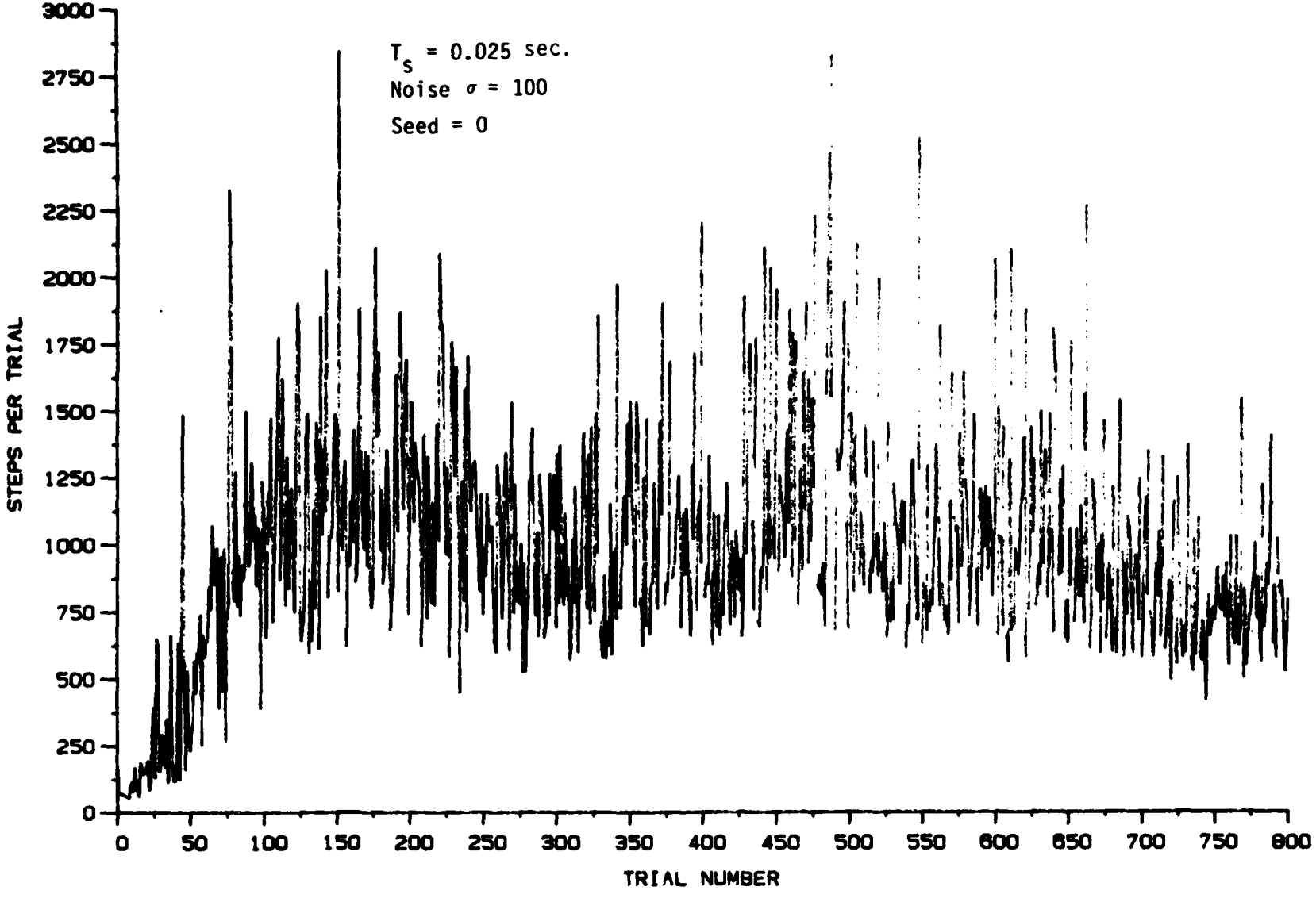


Figure 49. ALC Learning Curves with Parameters Indicated.

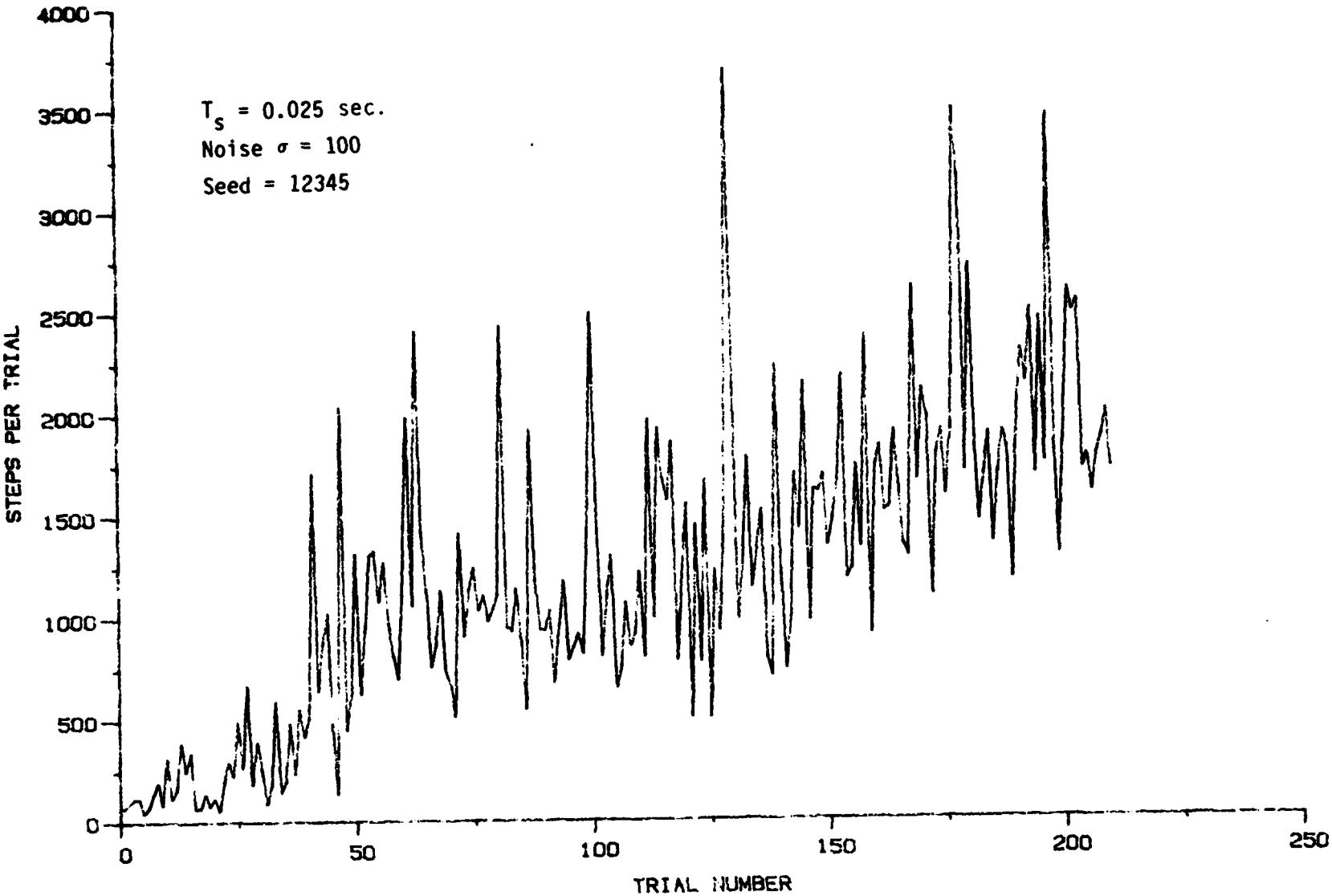


Figure 50. ALC Learning Curves with Parameters Indicated.

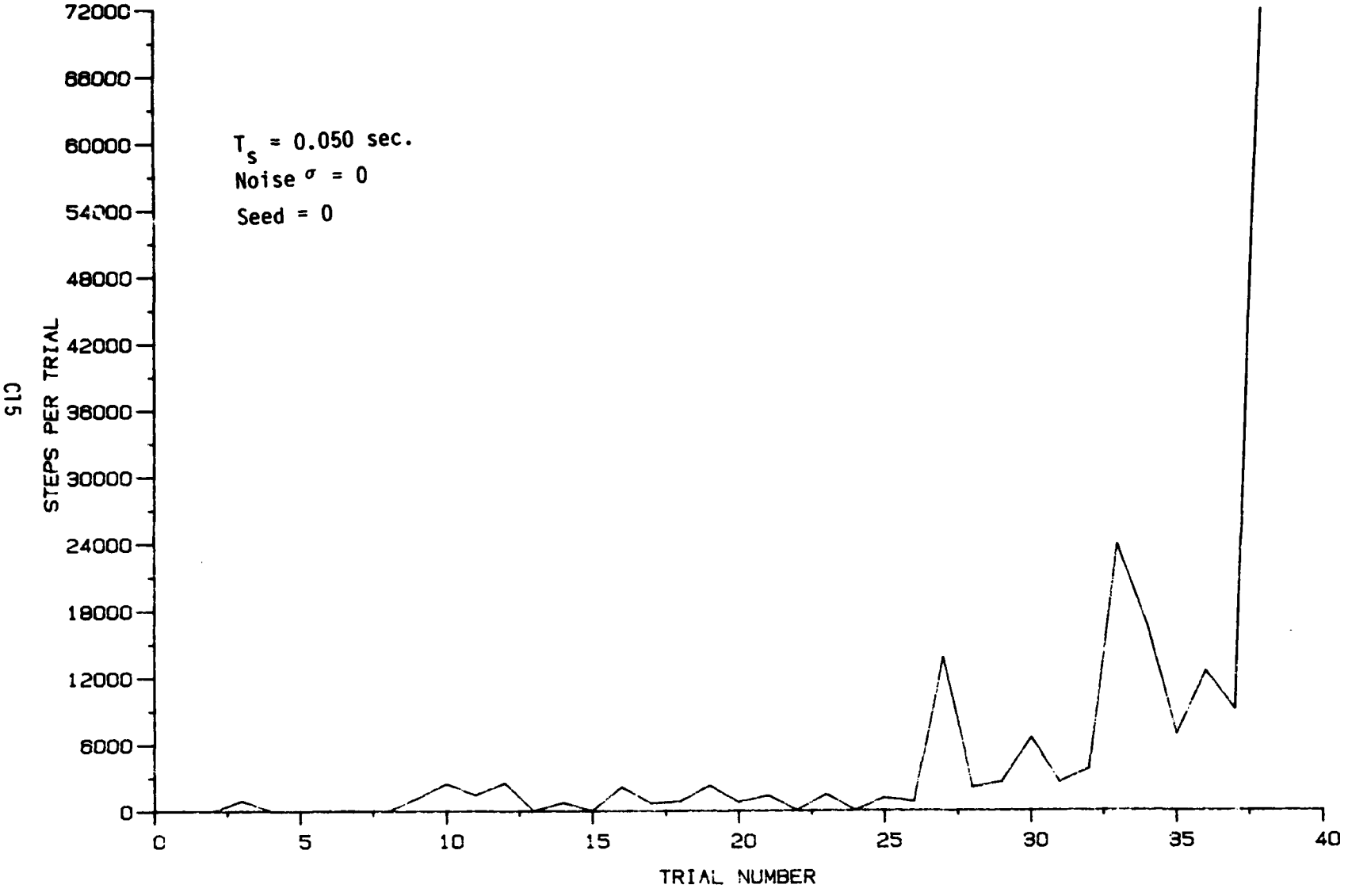


Figure 51. ALC Learning Curves with Parameters Indicated.

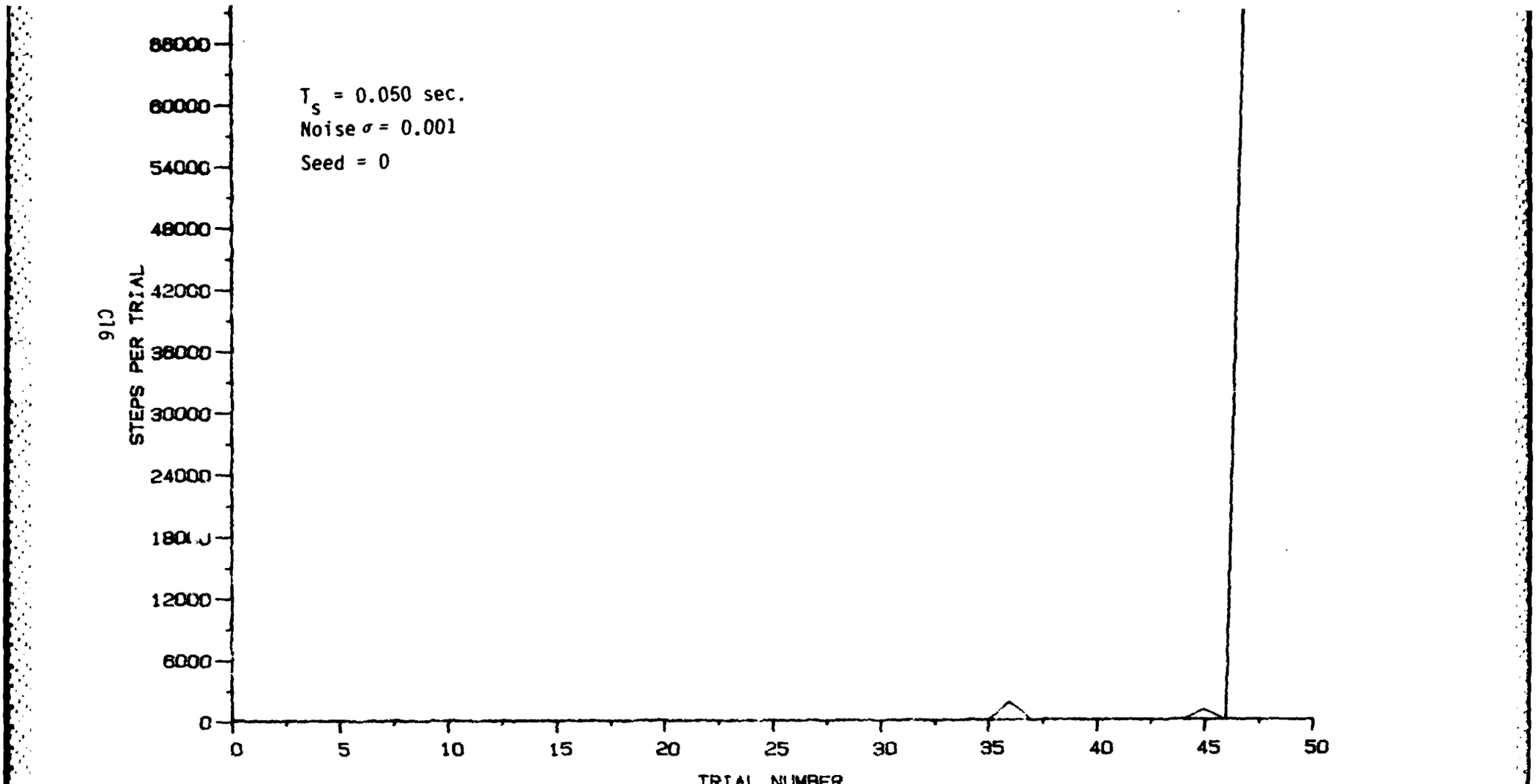
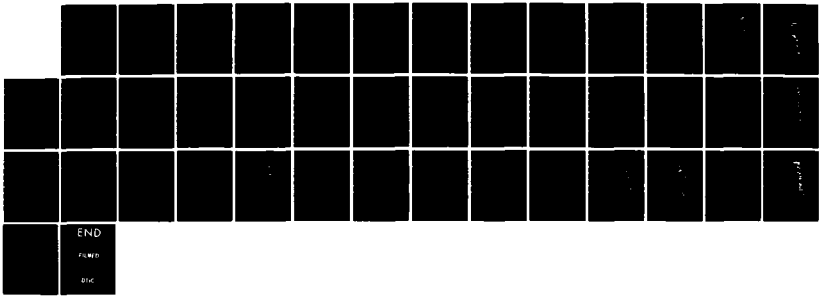
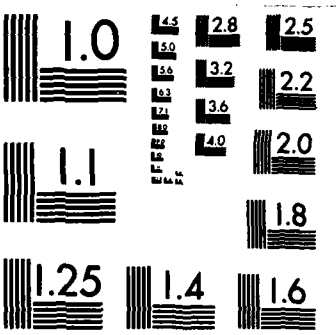


Figure 52. ALC Learning Curves with Parameters Indicated.

AD-A154 660 APPLICATIONS OF ADAPTIVE LEARNING CONTROLLER TO SYNTHETIC APERTURE RADAR..(U) ENVIRONMENTAL RESEARCH INST OF MICHIGAN ANN ARBOR RADAR DIV 2/2

UNCLASSIFIED D T POLITIS ET AL. FEB 85 ERIM-163800-4-F F/G 17/9 NL





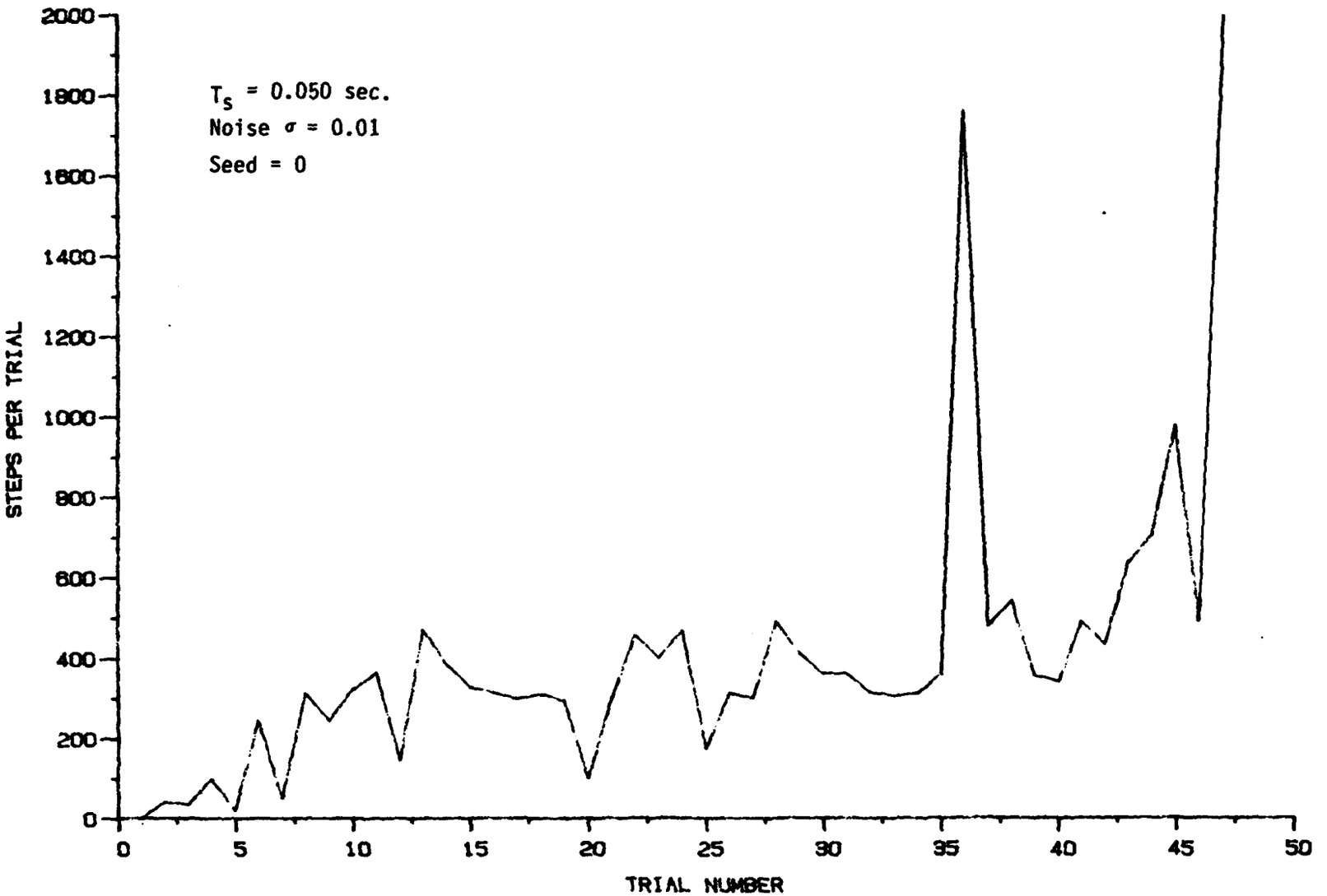


Figure 53. ALC Learning Curves with Parameters Indicated.

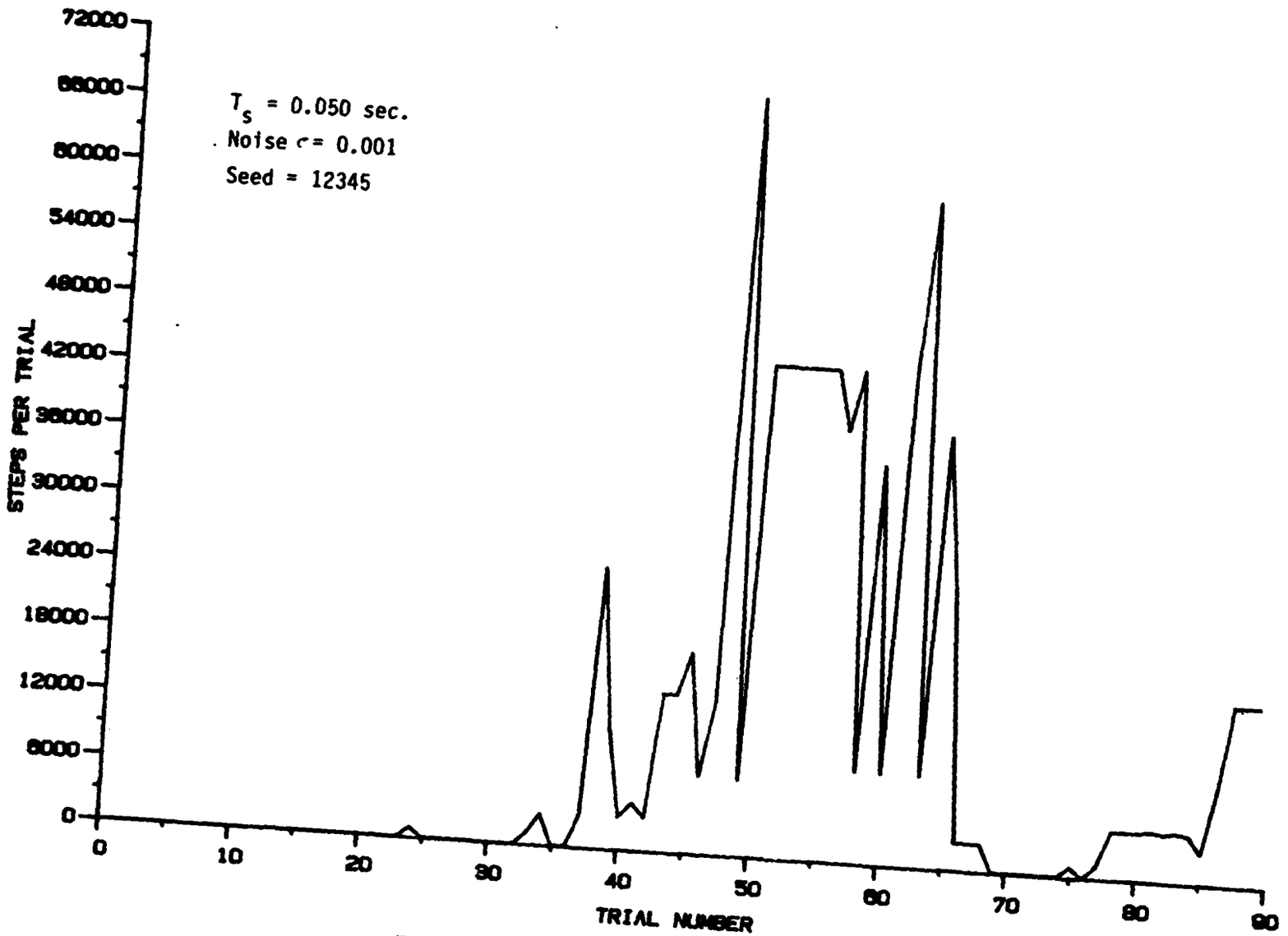


Figure 54. ALC Learning Curves with Parameters Indicated.

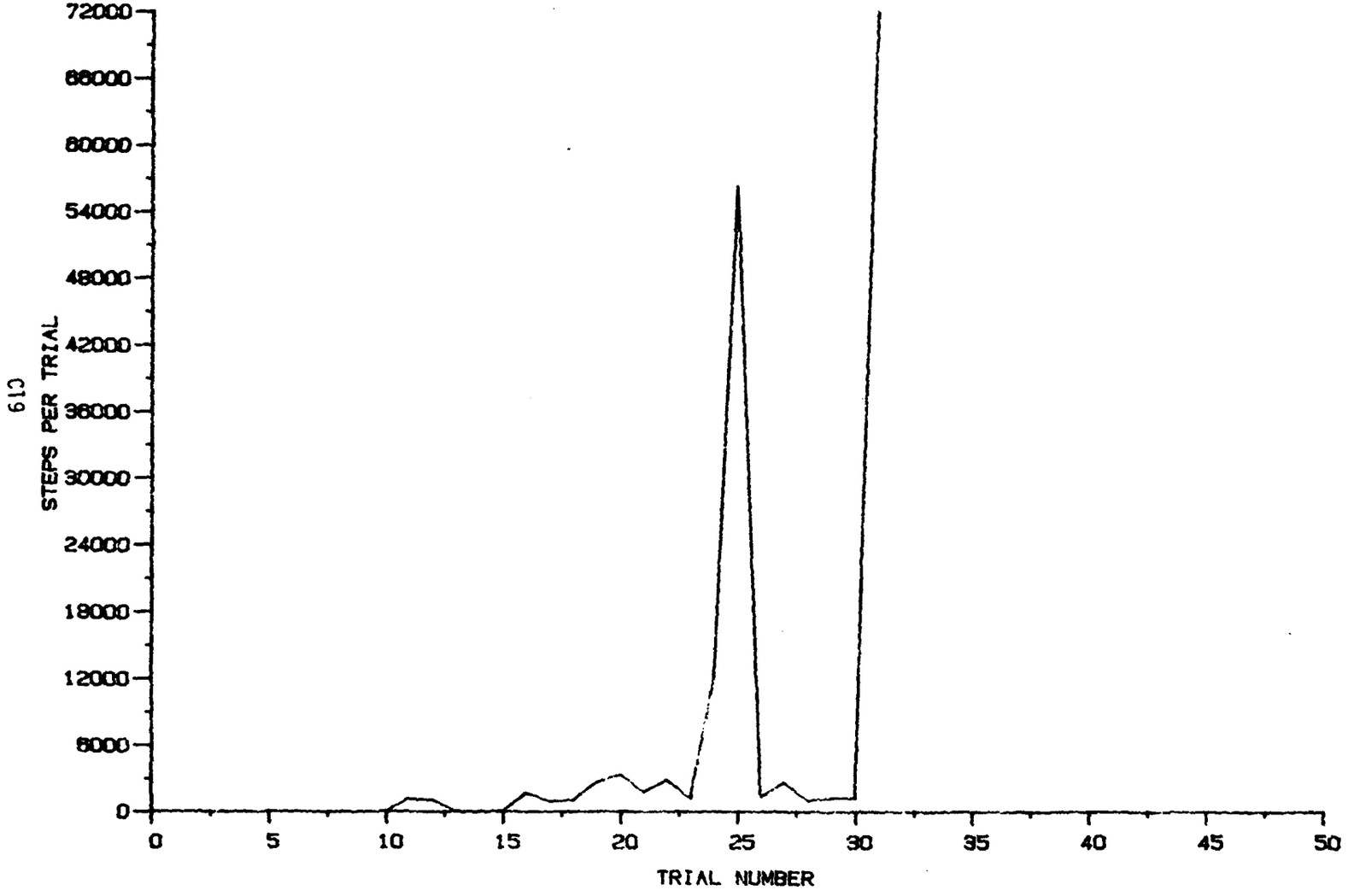
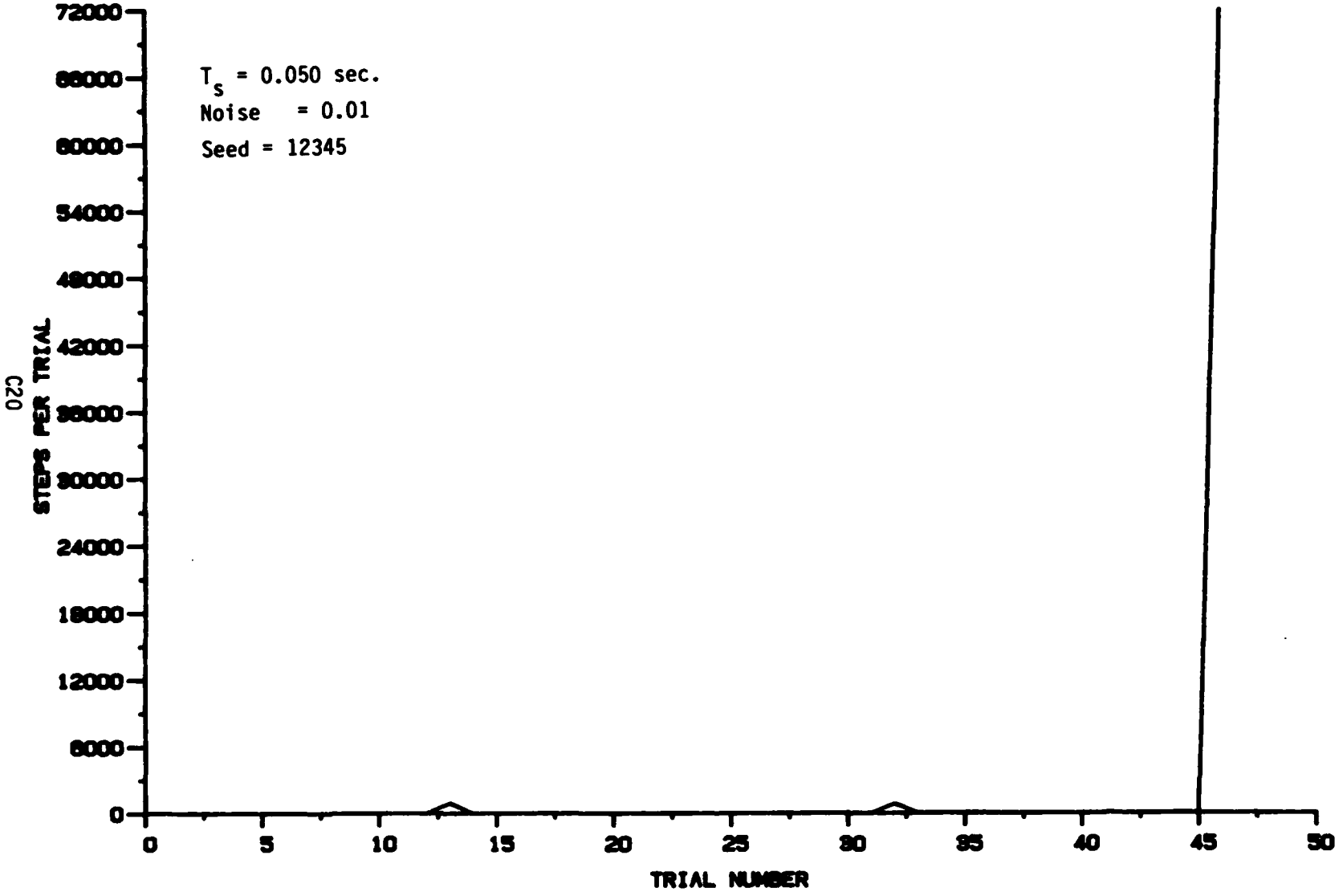


Figure 56. ALC Learning Curves with Parameters Indicated.



C20

Figure 57. ALC Learning Curves with Parameters Indicated.

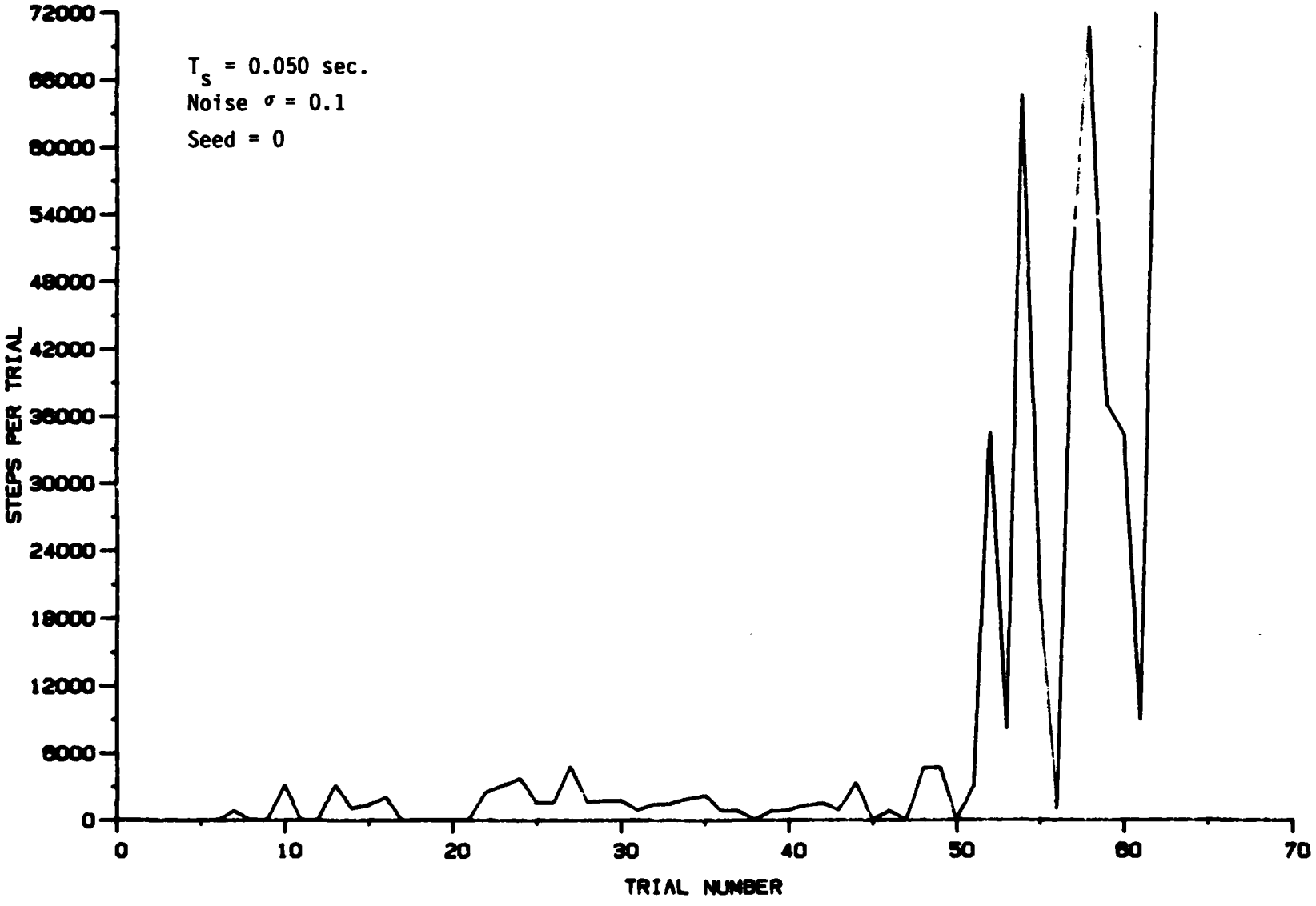


Figure 58. ALC Learning Curves with Parameters Indicated.

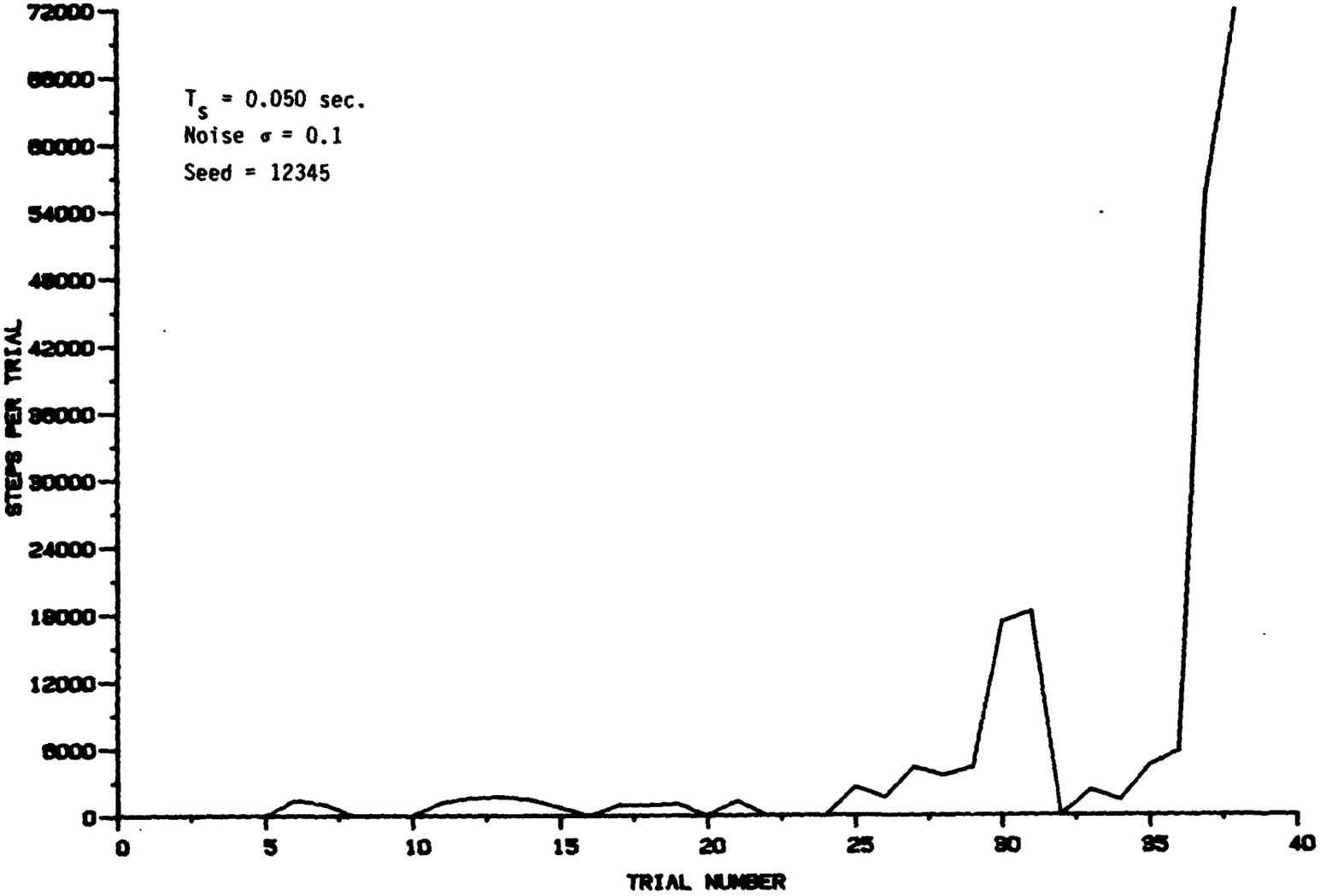


Figure 59. ALC Learning Curves with Parameters Indicated.

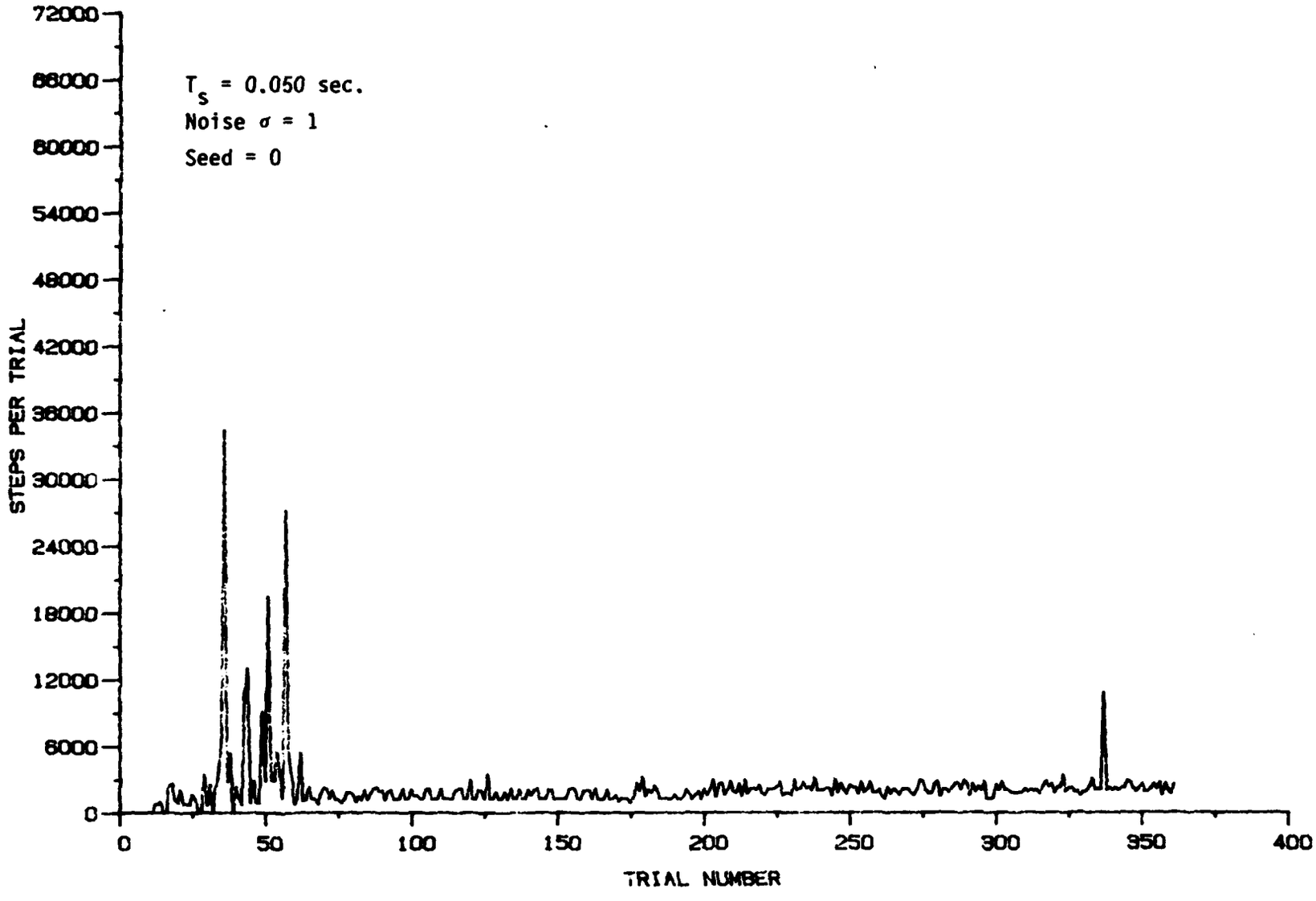


Figure 60. ALC Learning Curves with Parameters Indicated.

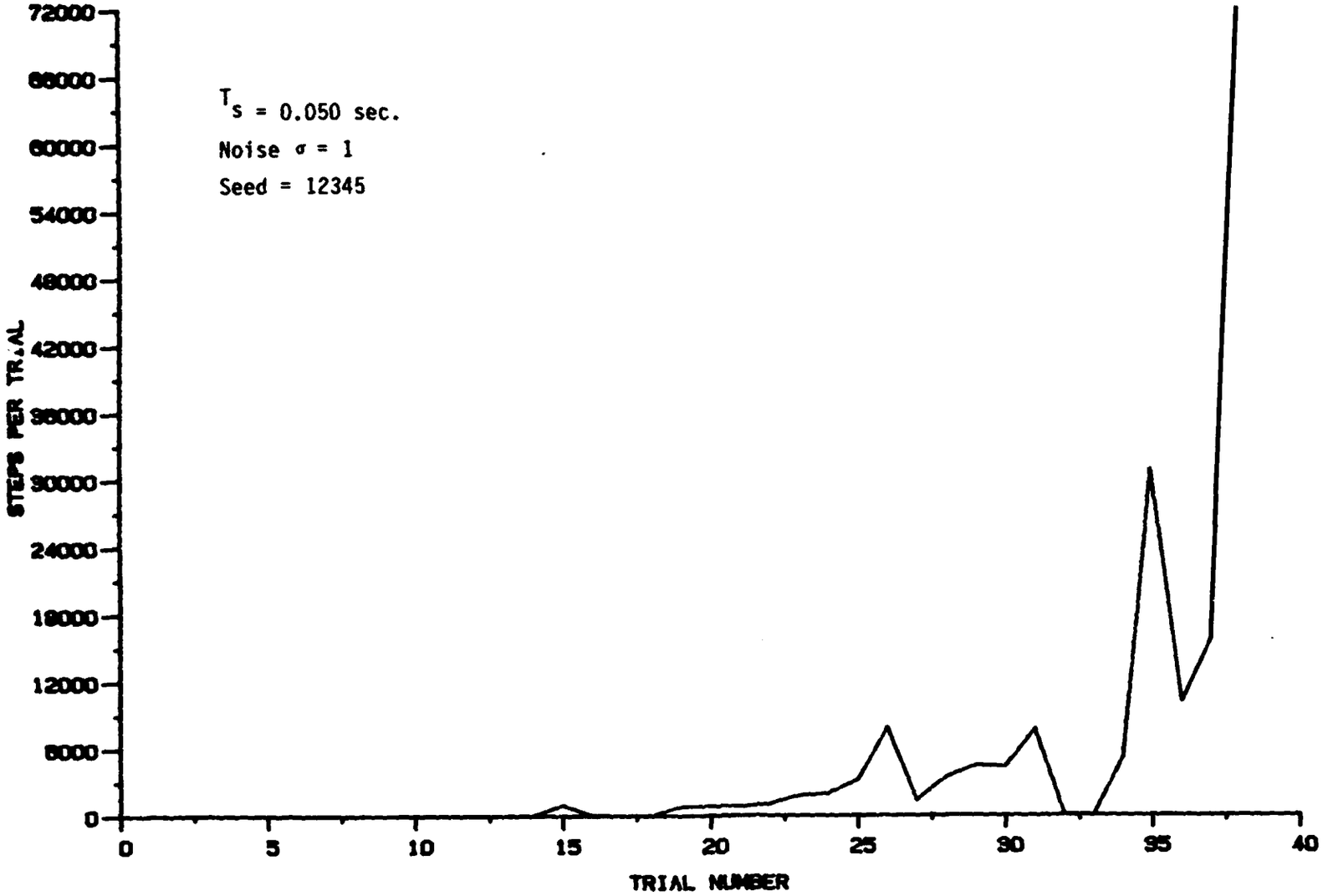


Figure 61. ALC Learning Curves with Parameters Indicated.

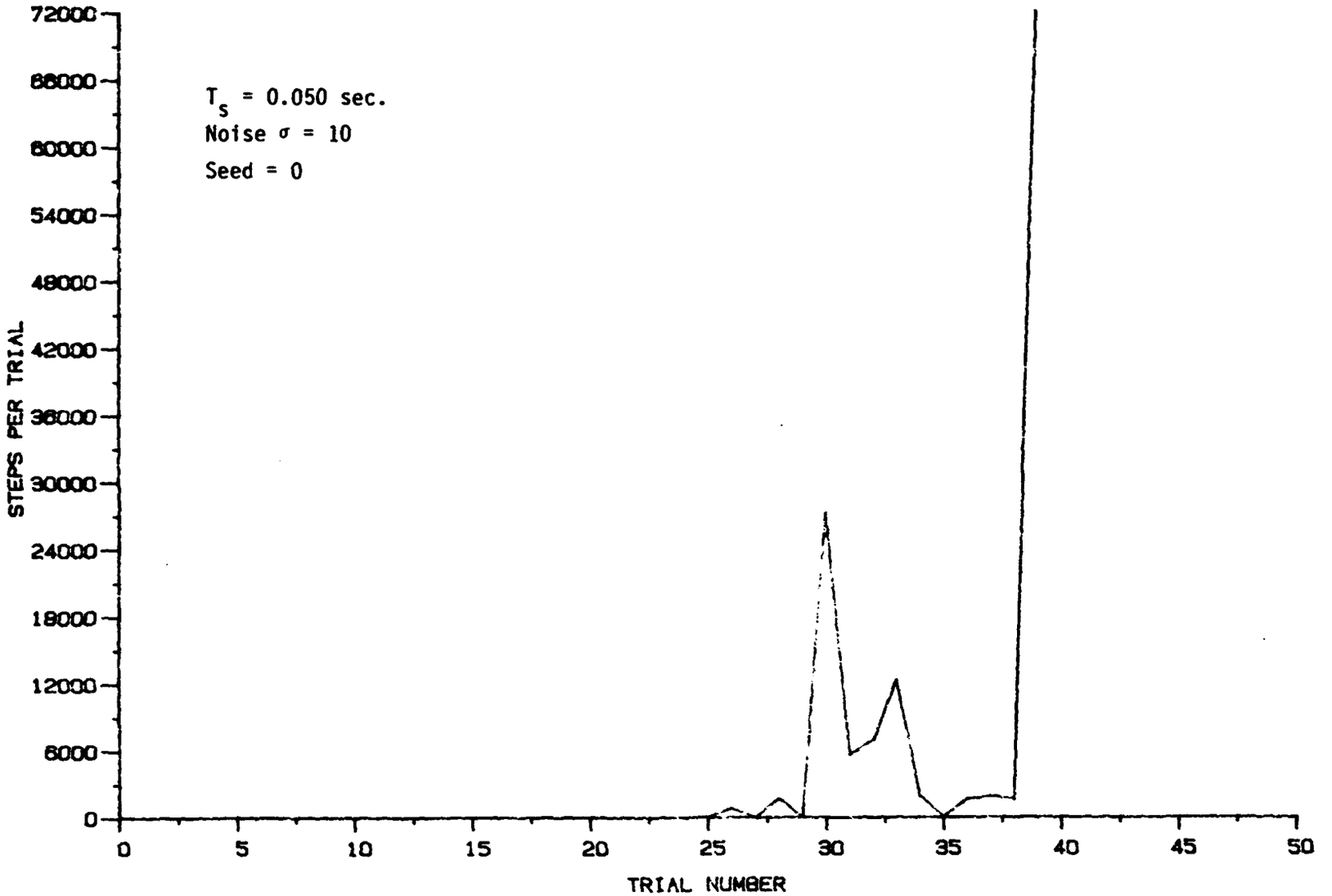


Figure 62. ALC Learning Curves with Parameters Indicated.

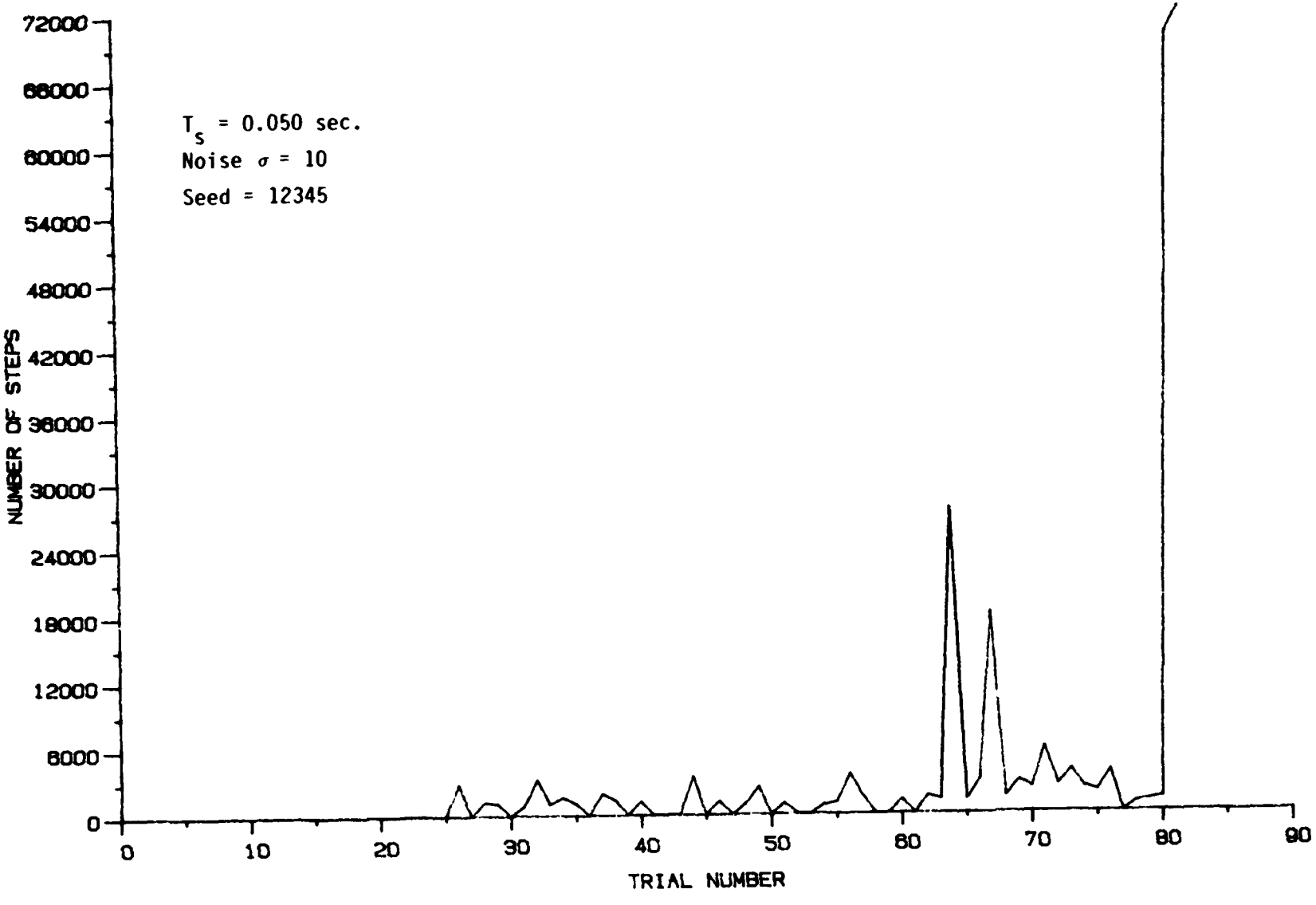


Figure 63. ALC Learning Curves with Parameters Indicated.

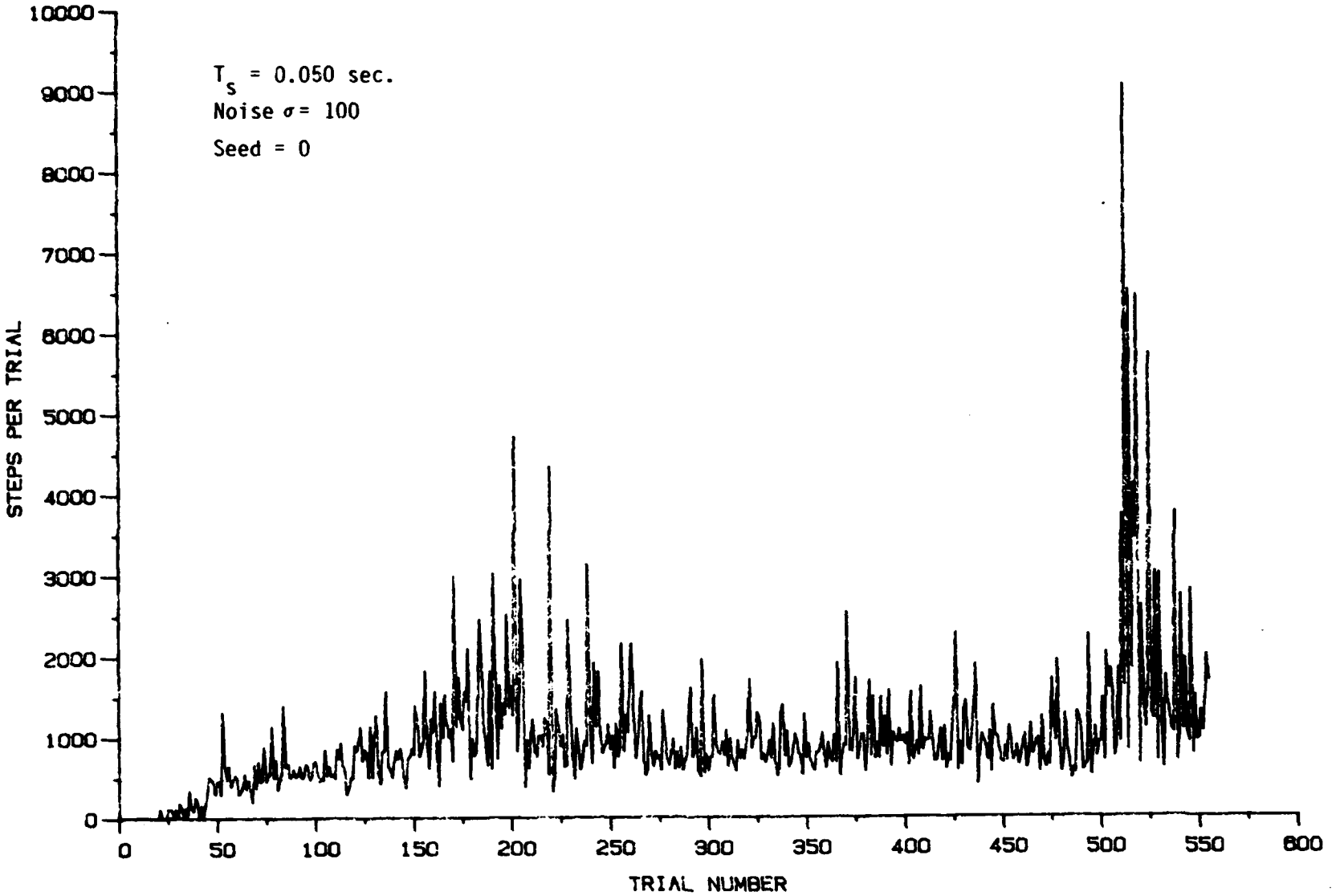


Figure 64. ALC Learning Curves with Parameters Indicated.

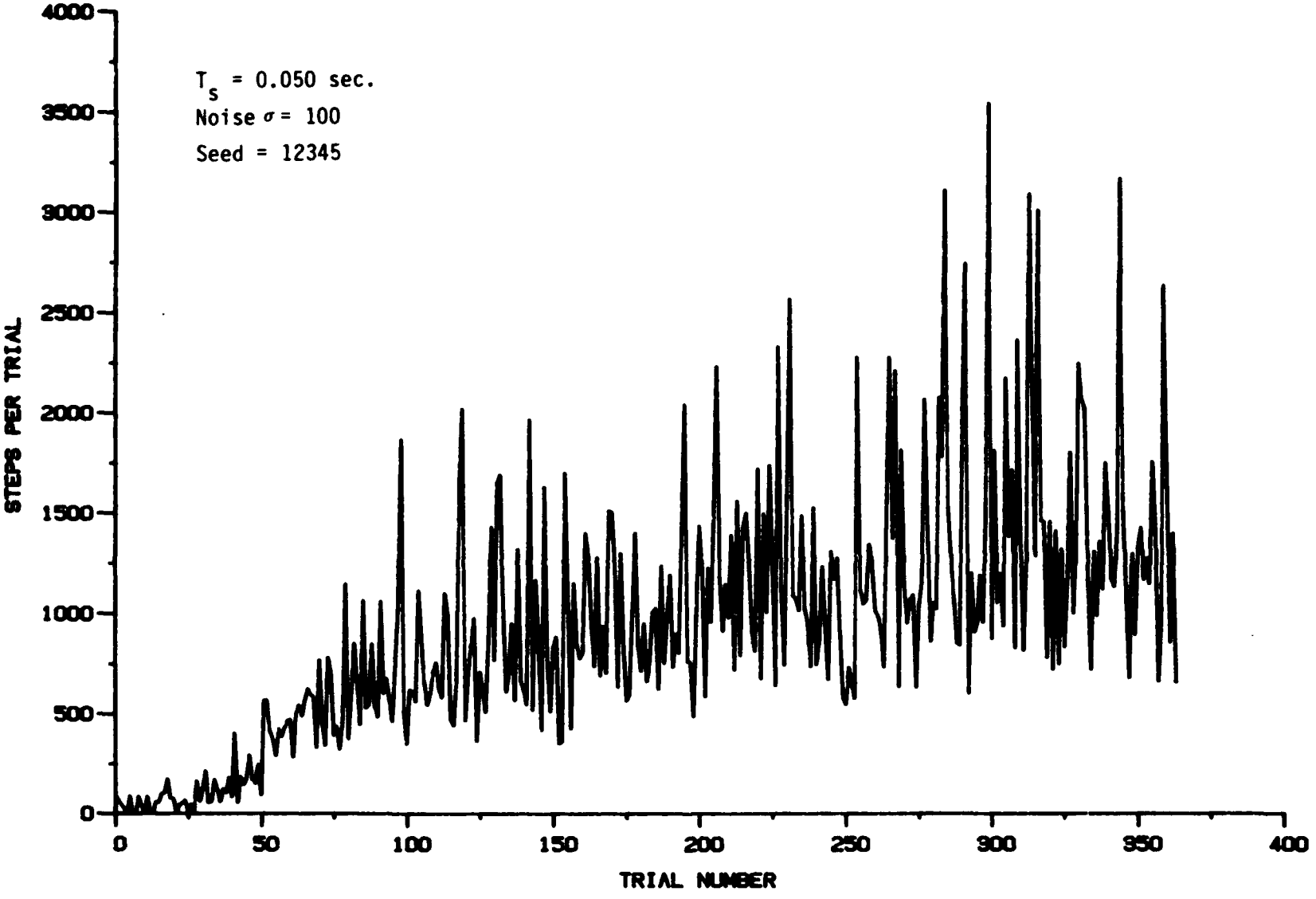


Figure 65. ALC Learning Curves with Parameters Indicated.

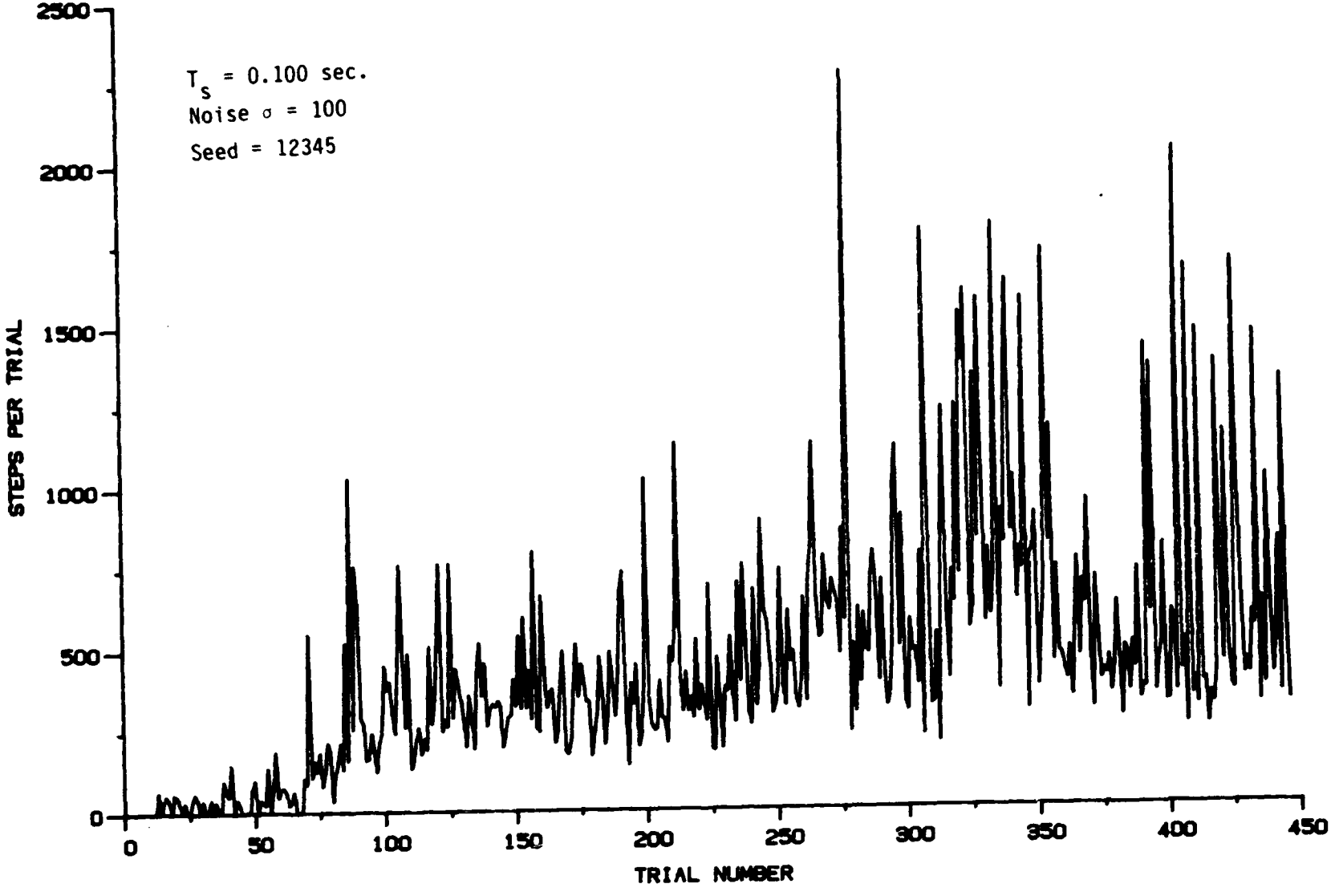


Figure 79. ALC Learning Curves with Parameters Indicated.

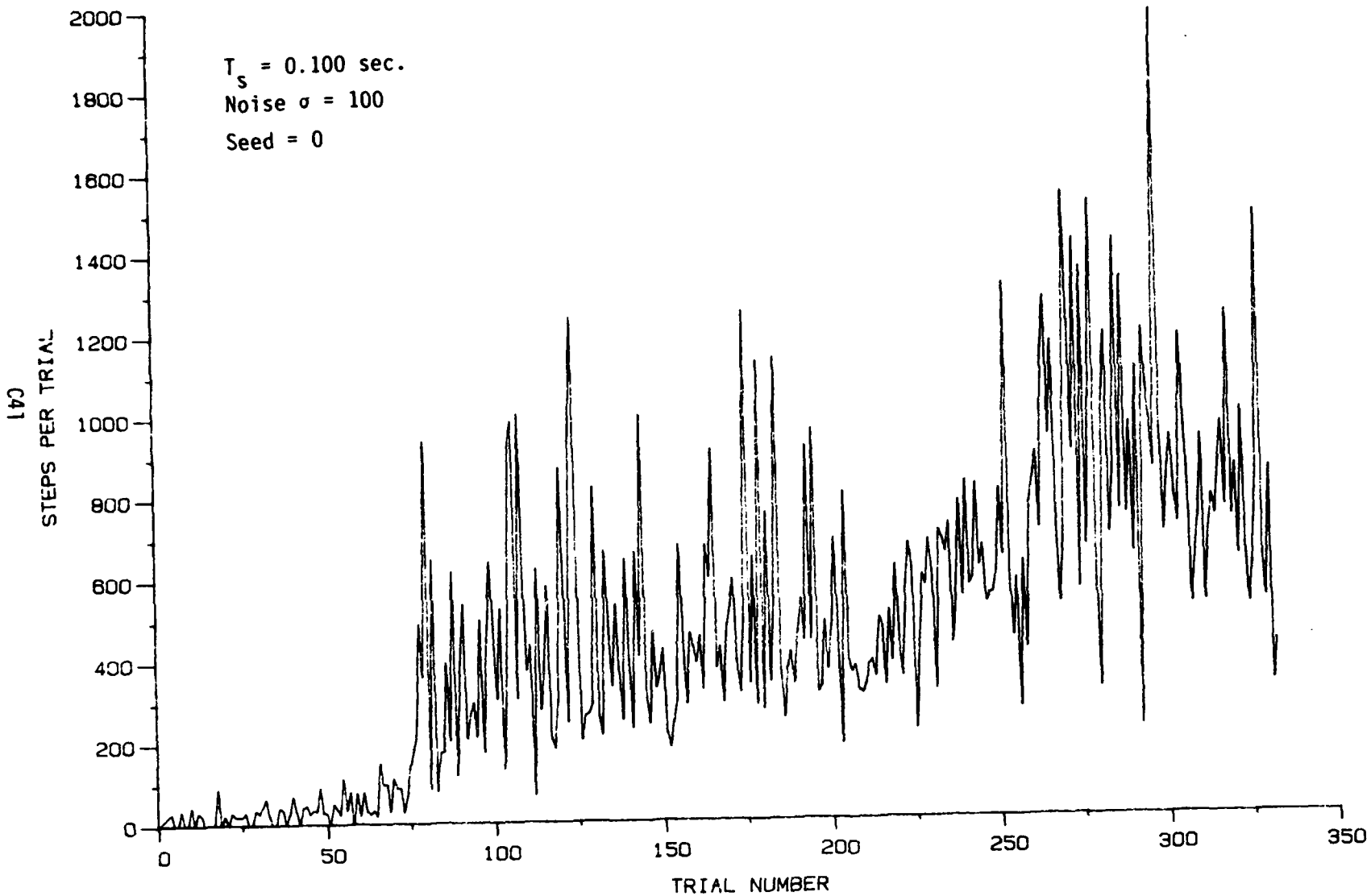


Figure 78. ALC Learning Curves with Parameters Indicated.

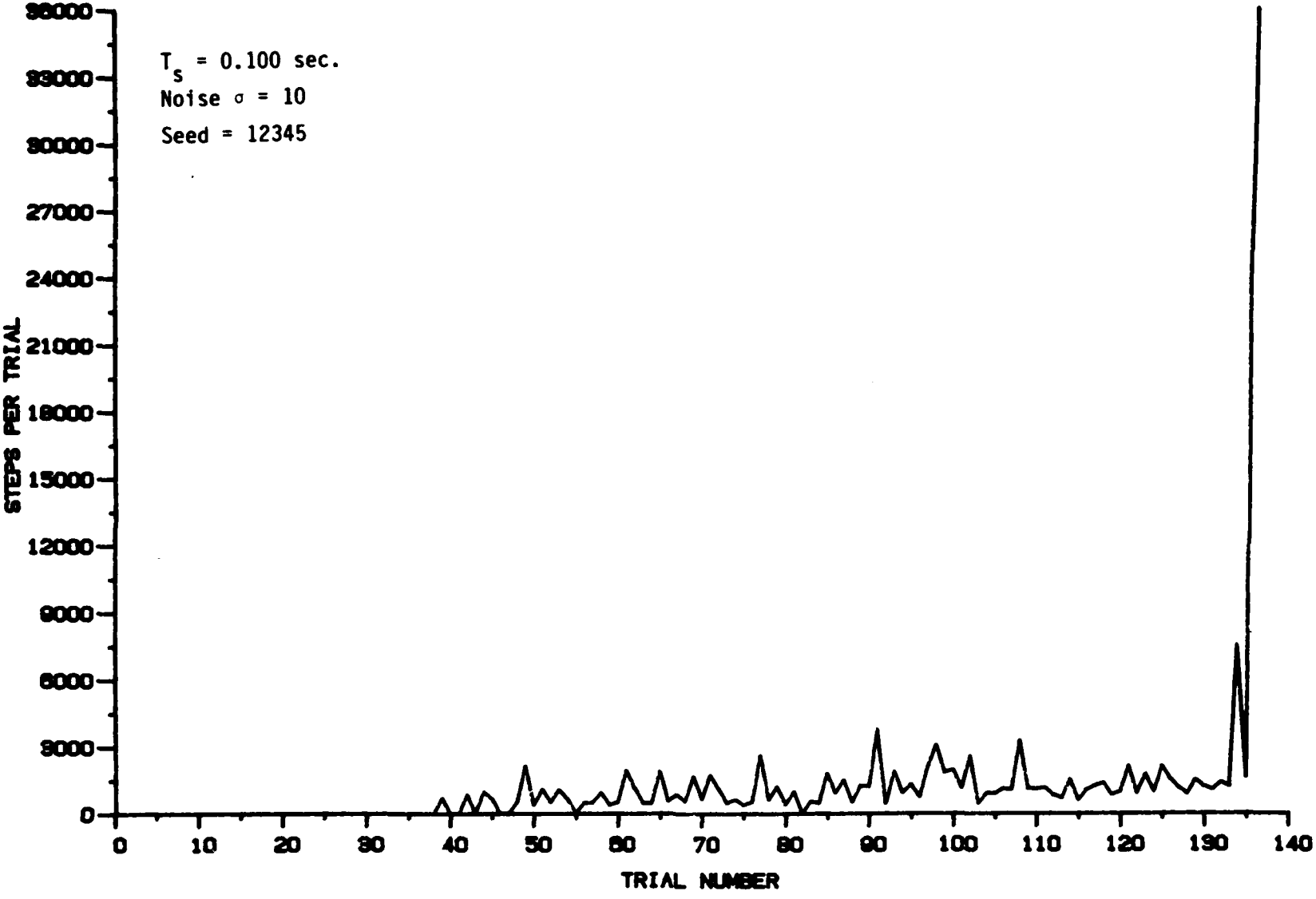


Figure 77. ALC Learning Curves with Parameters Indicated.

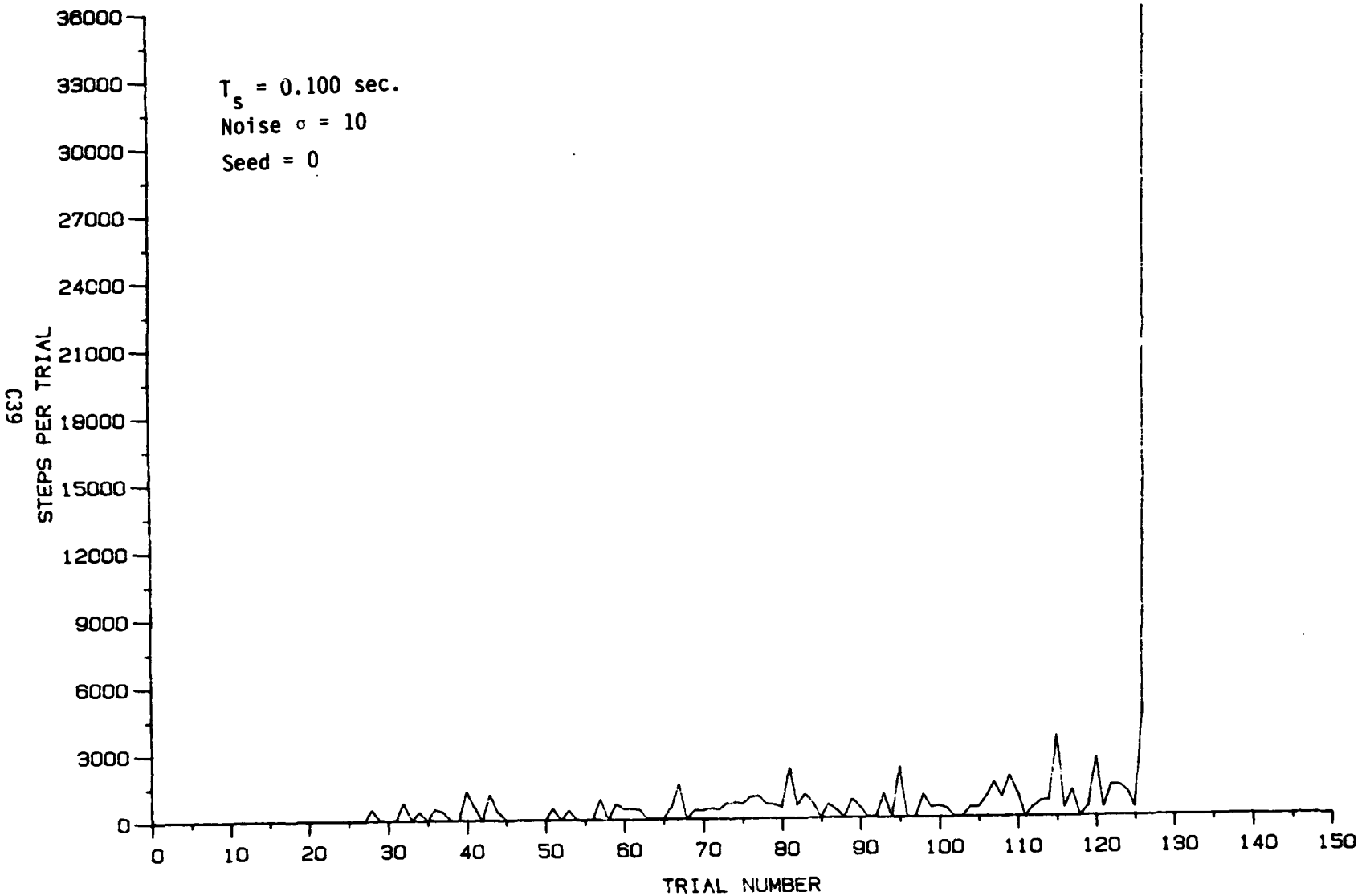


Figure 76. ALC Learning Curves with Parameters Indicated.

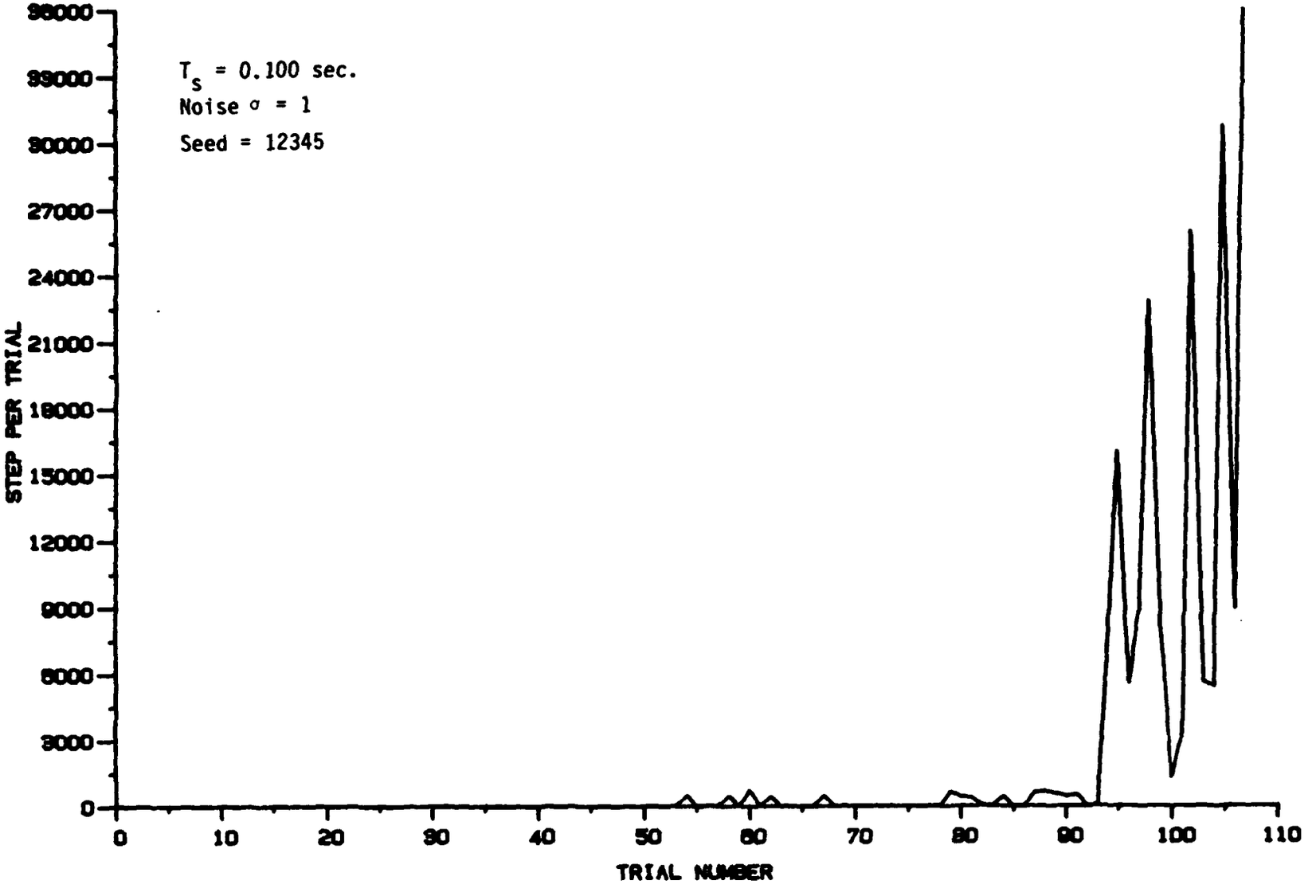


Figure 75. ALC Learning Curves with Parameters Indicated.

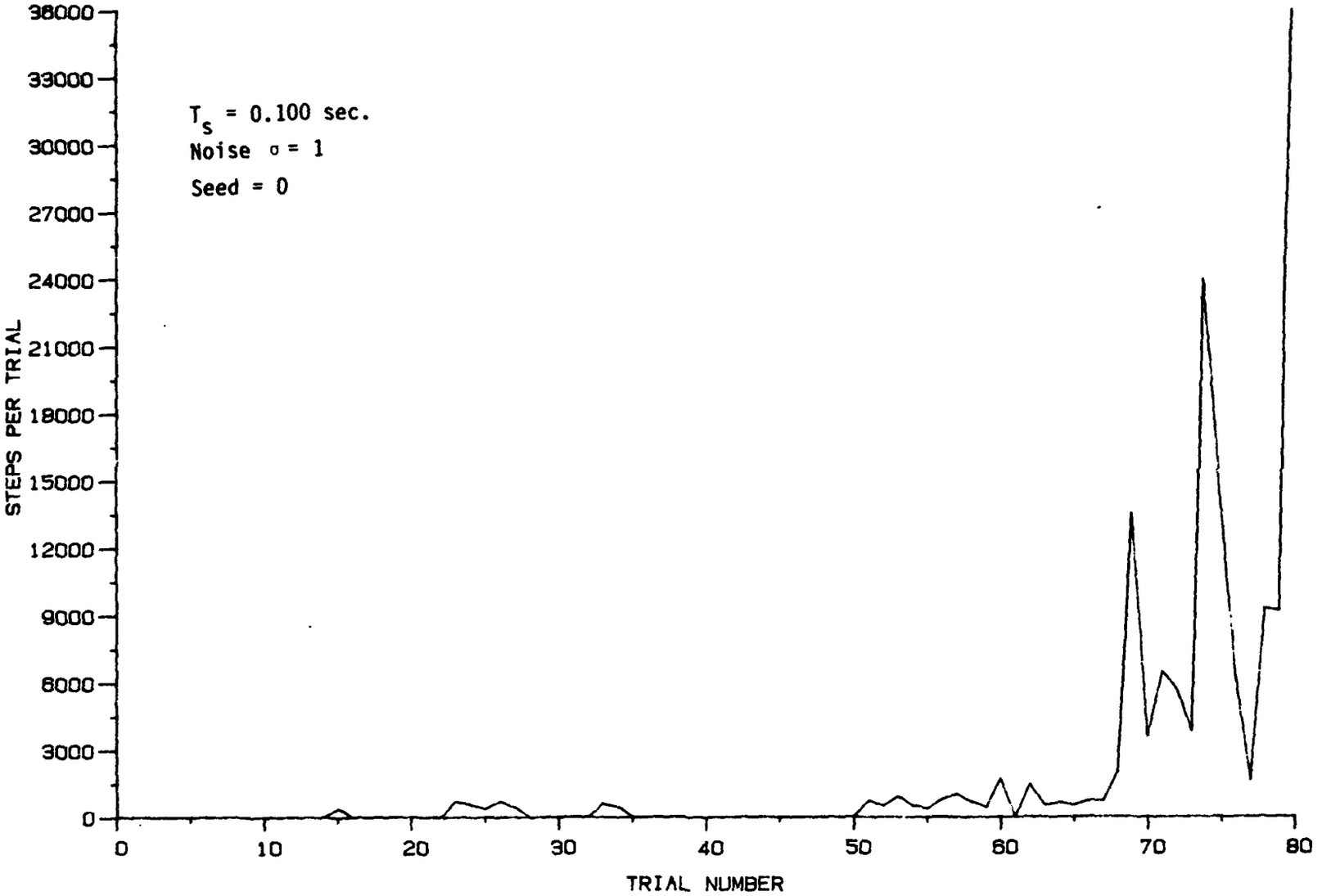


Figure 74. ALC Learning Curves with Parameters Indicated.

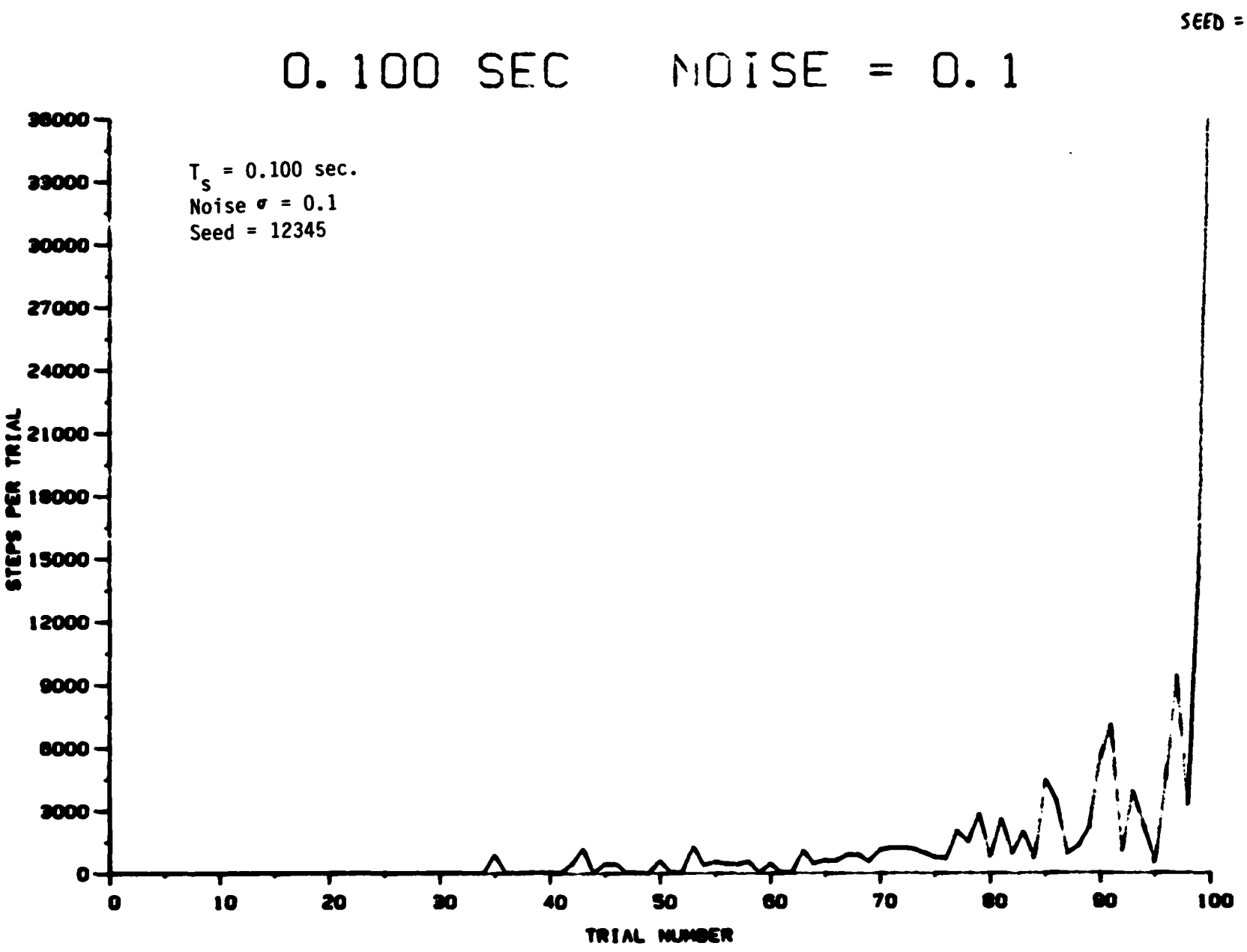


Figure 73. ALC Learning Curves with Parameters Indicated.

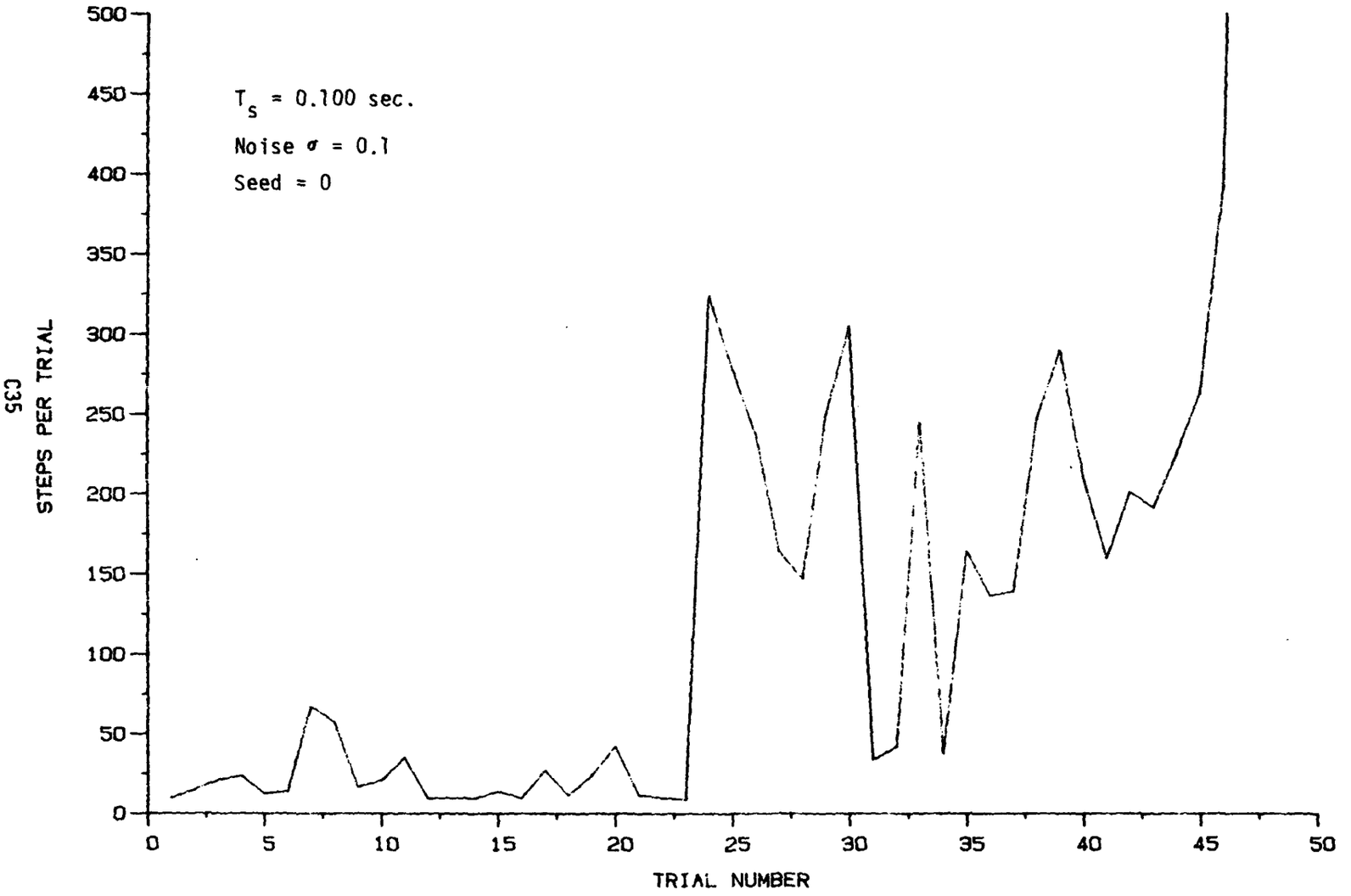


Figure 72. ALC Learning Curves with Parameters Indicated.

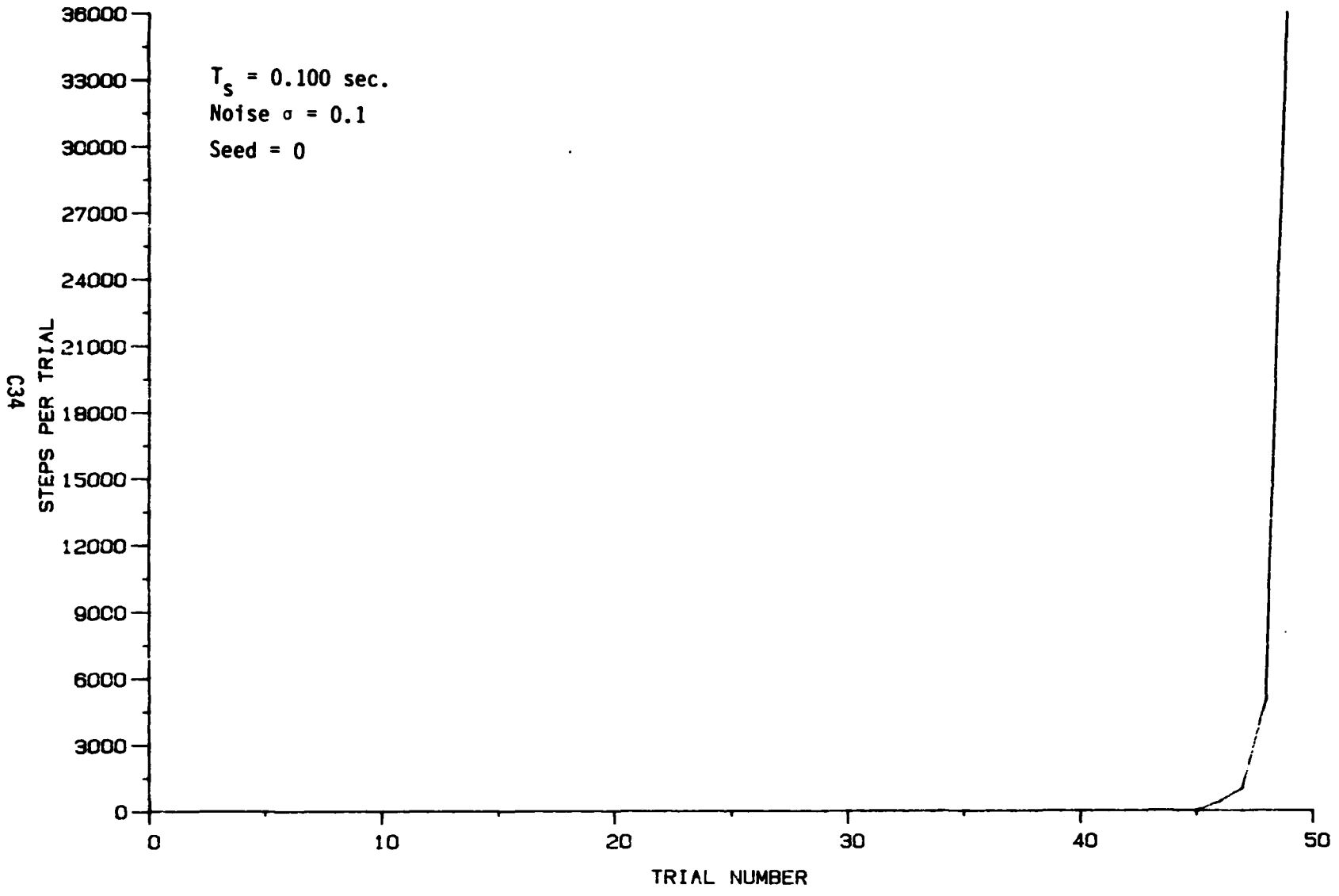


Figure 71. ALC Learning Curves with Parameters Indicated.

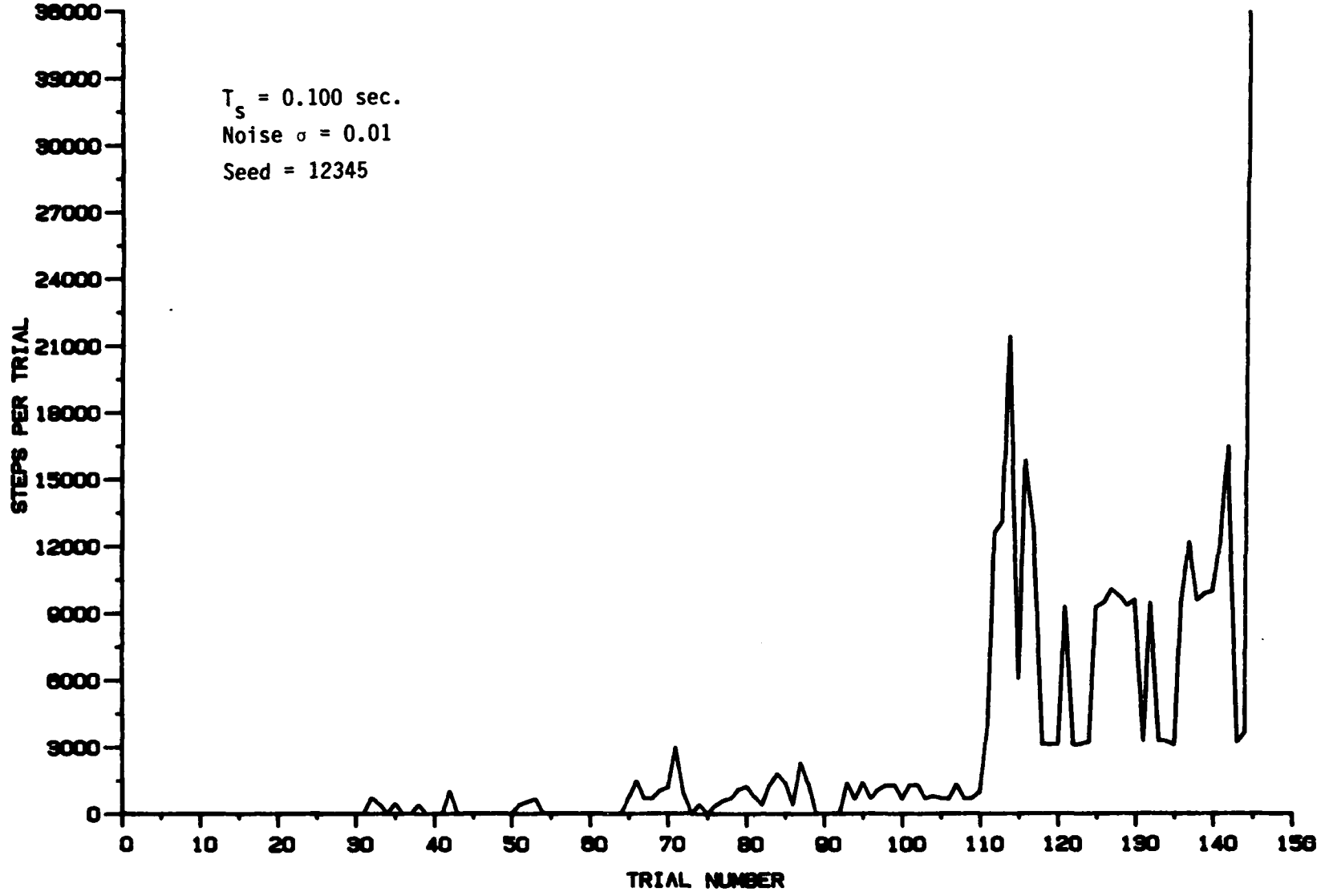


Figure 70. ALC Learning Curves with Parameters Indicated.

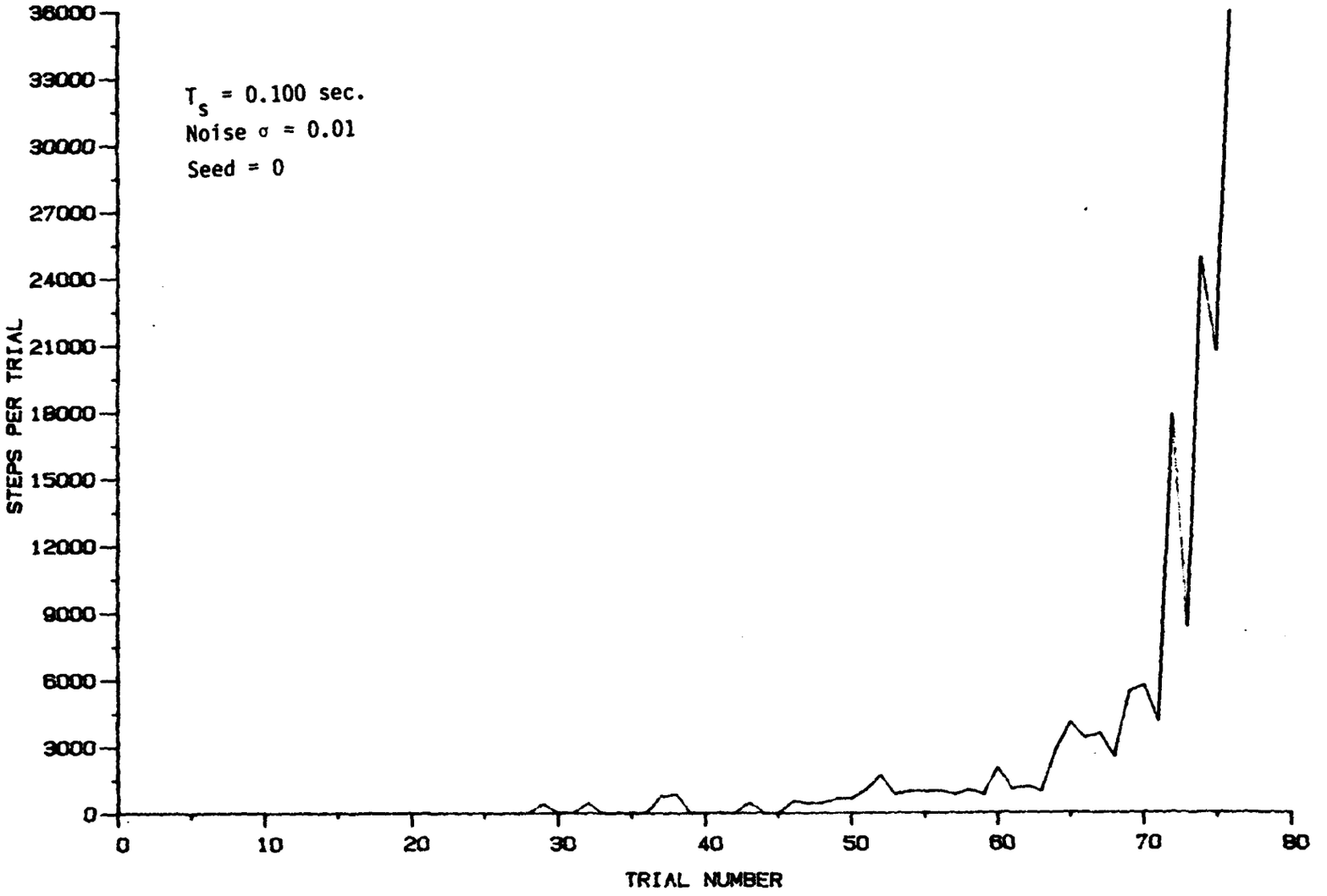


Figure 69. ALC Learning Curves with Parameters Indicated.

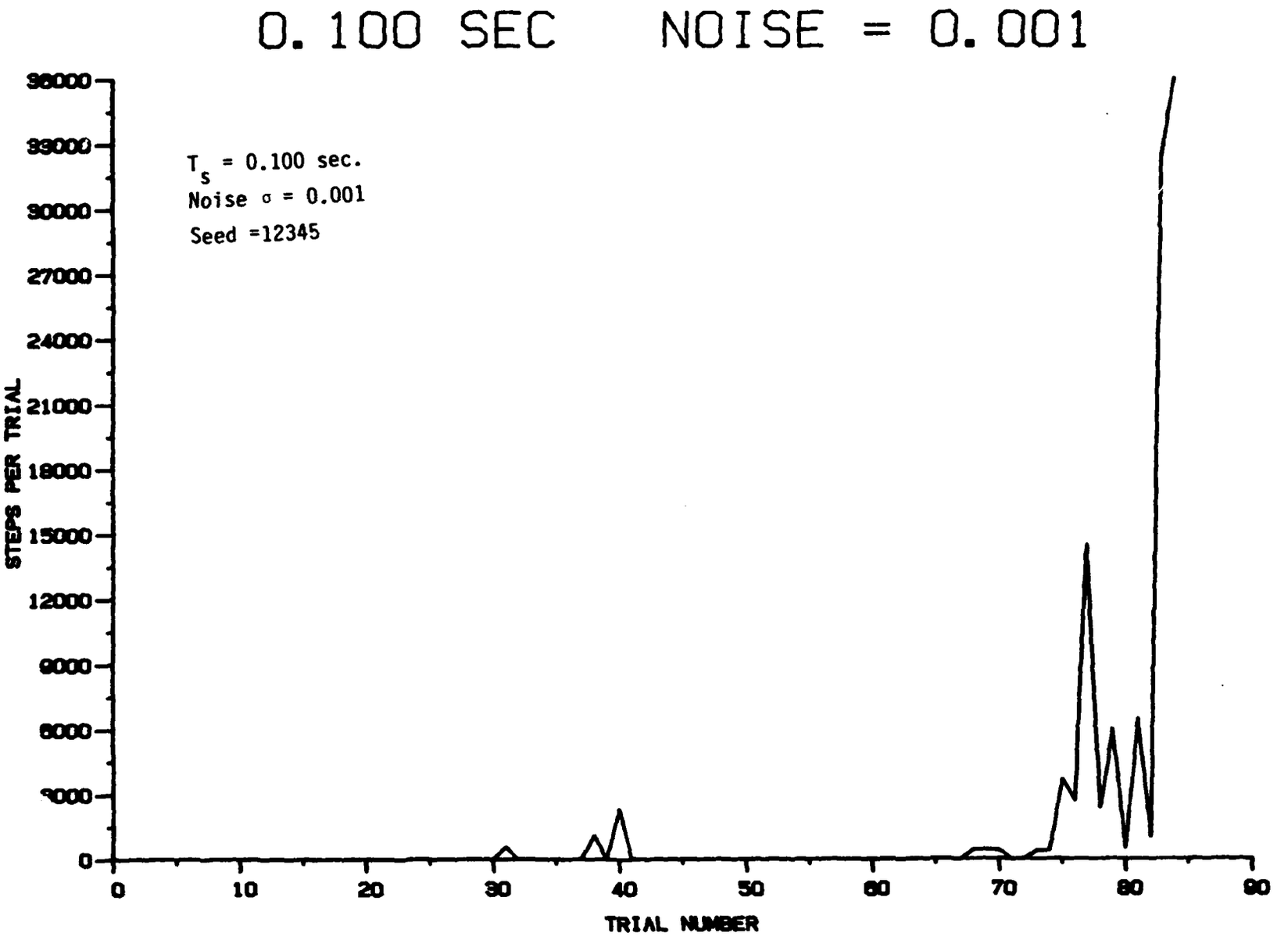


Figure 68. ALC Learning Curves with Parameters Indicated.

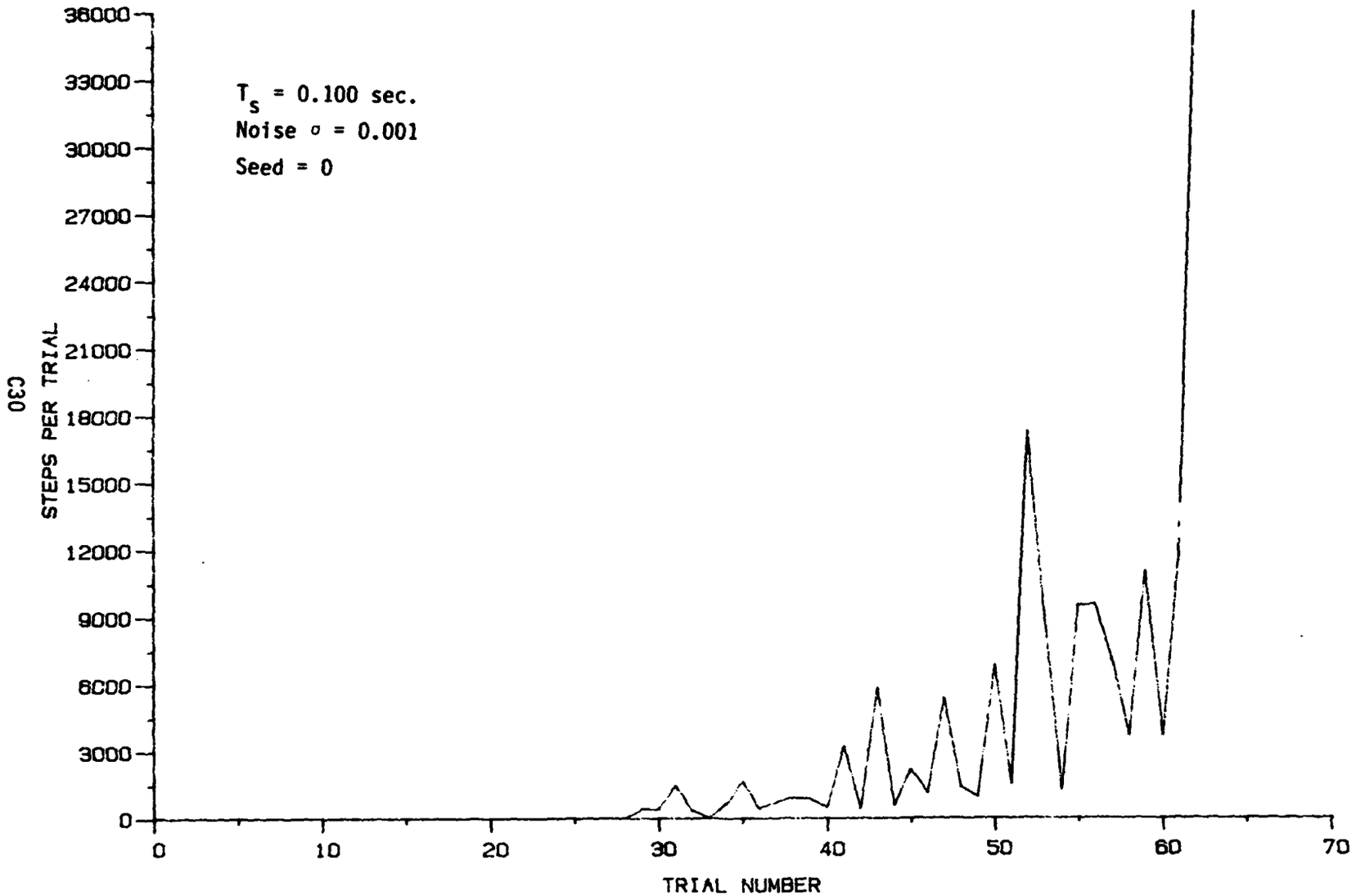


Figure 67. ALC Learning Curves with Parameters Indicated.

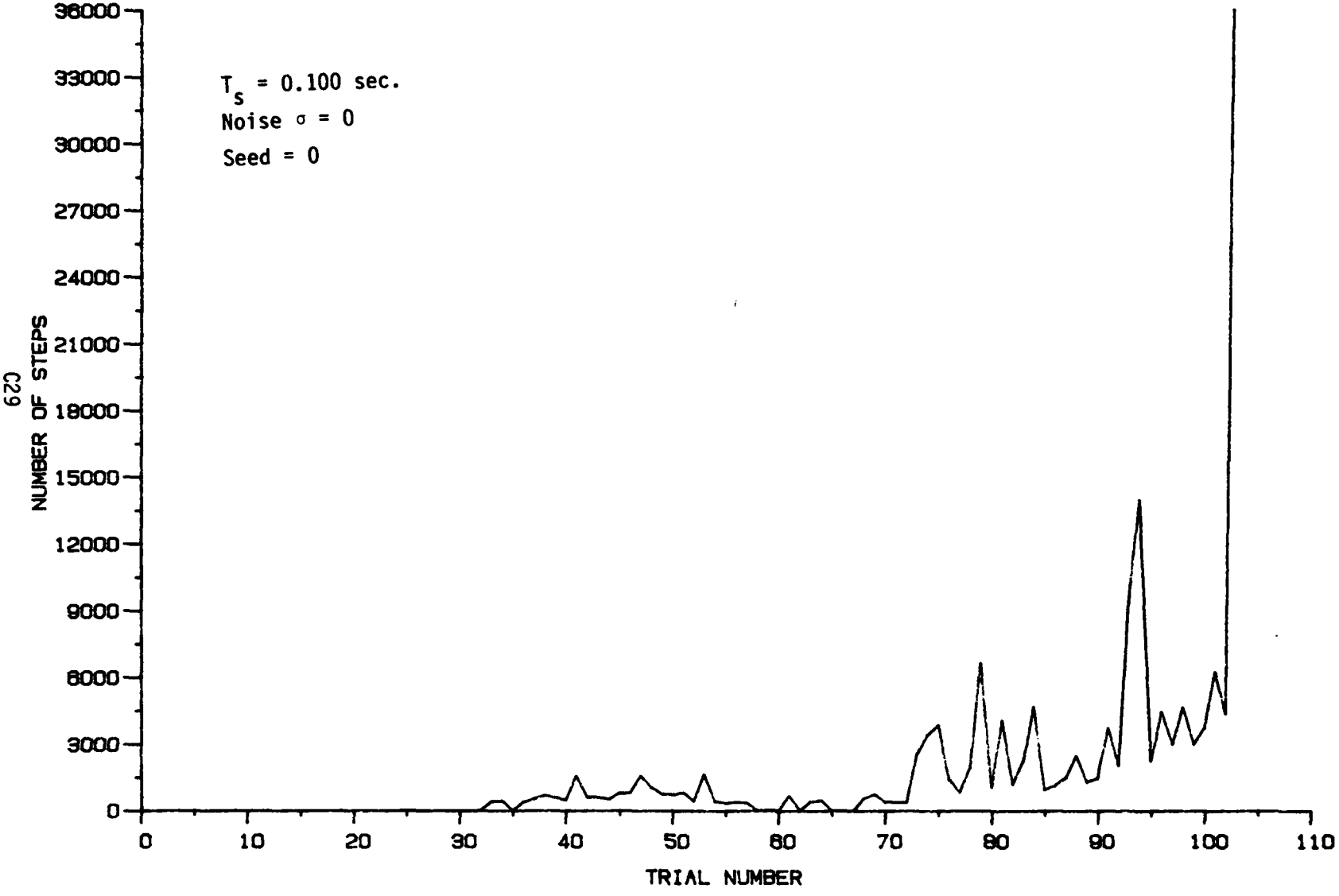


Figure 66. ALC Learning Curves with Parameters Indicated.

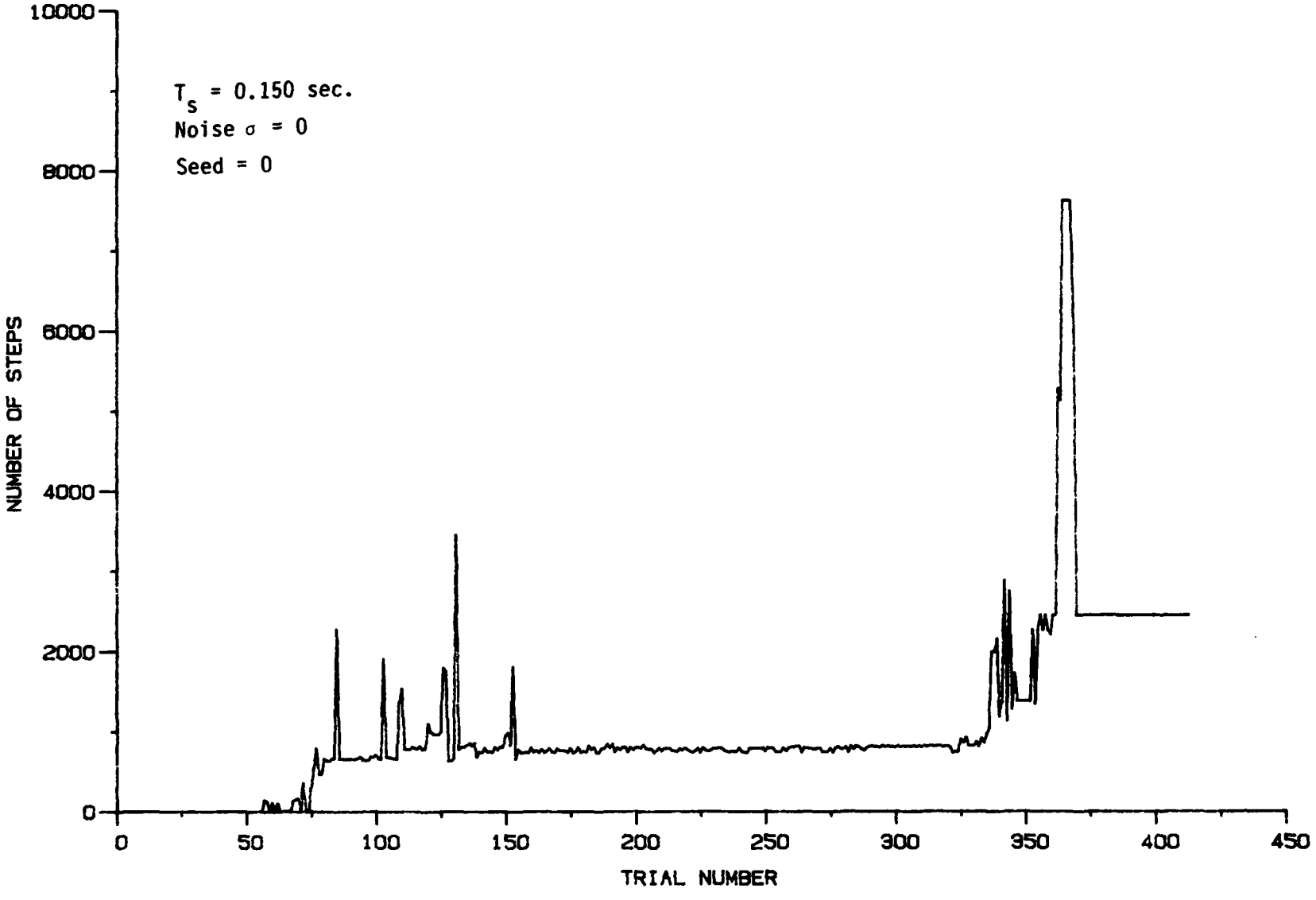


Figure 80. ALC Learning Curves with Parameters Indicated.

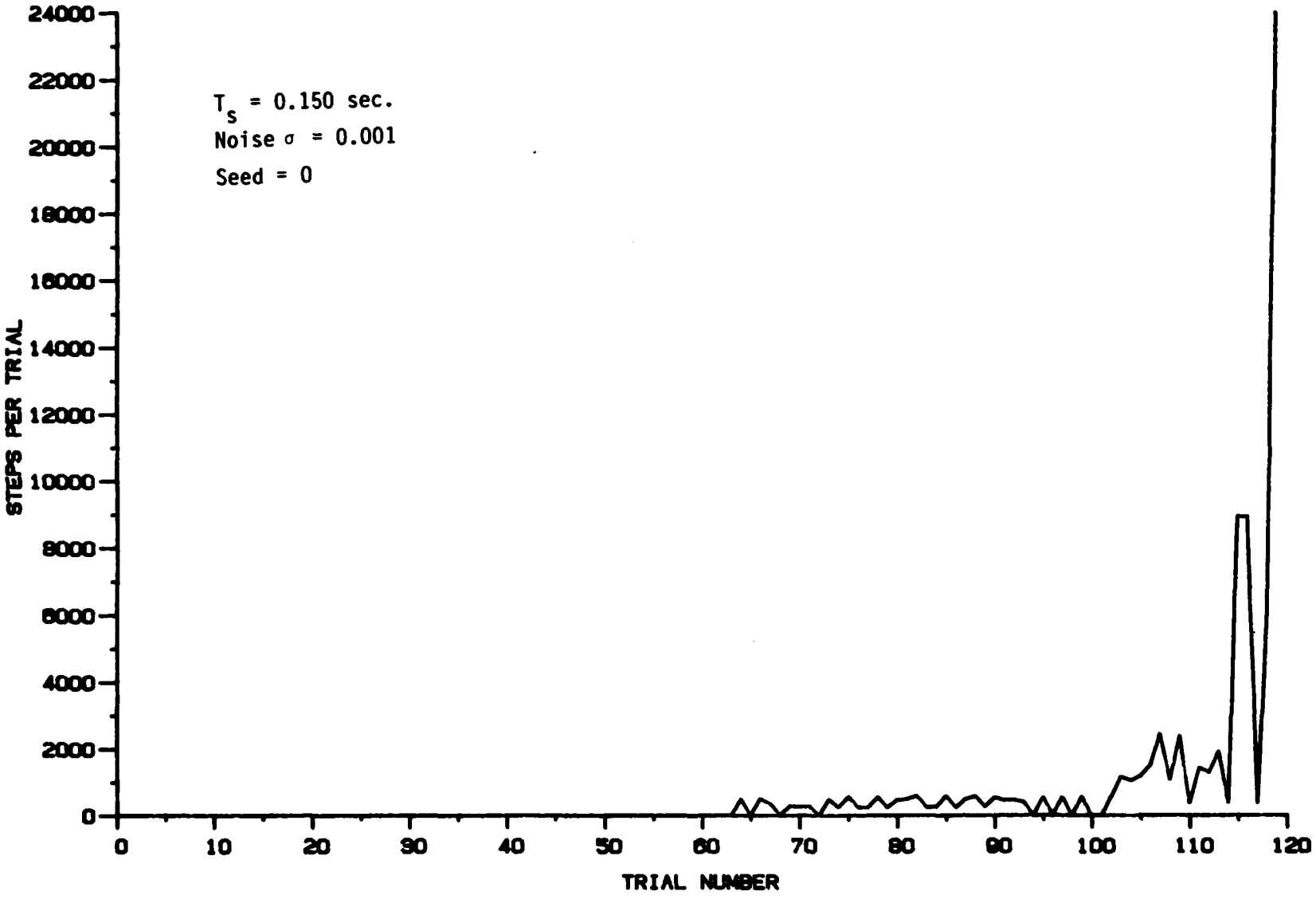


Figure 81. ALC Learning Curves with Parameters Indicated.

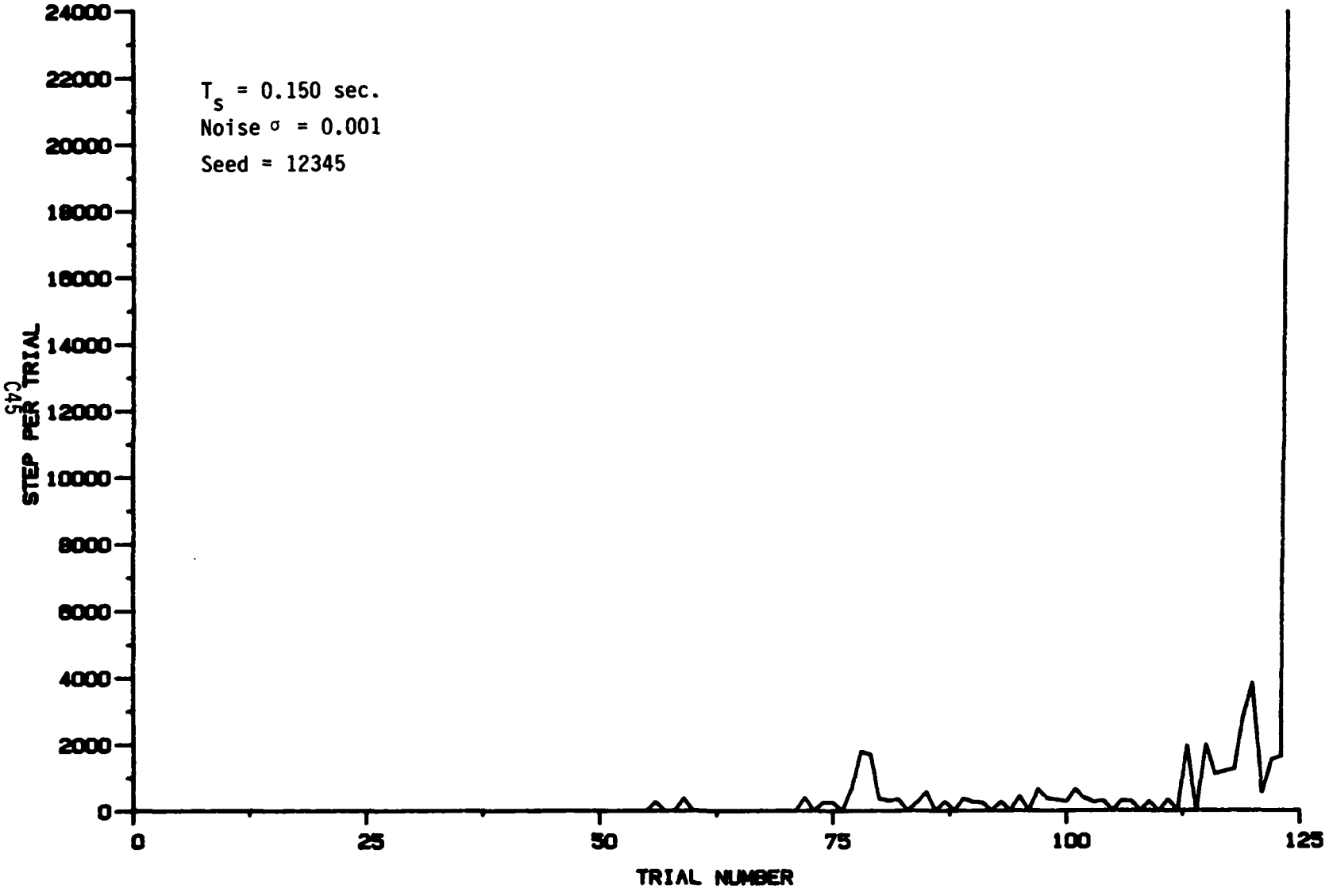


Figure 82. ALC Learning Curves with Parameters Indicated.

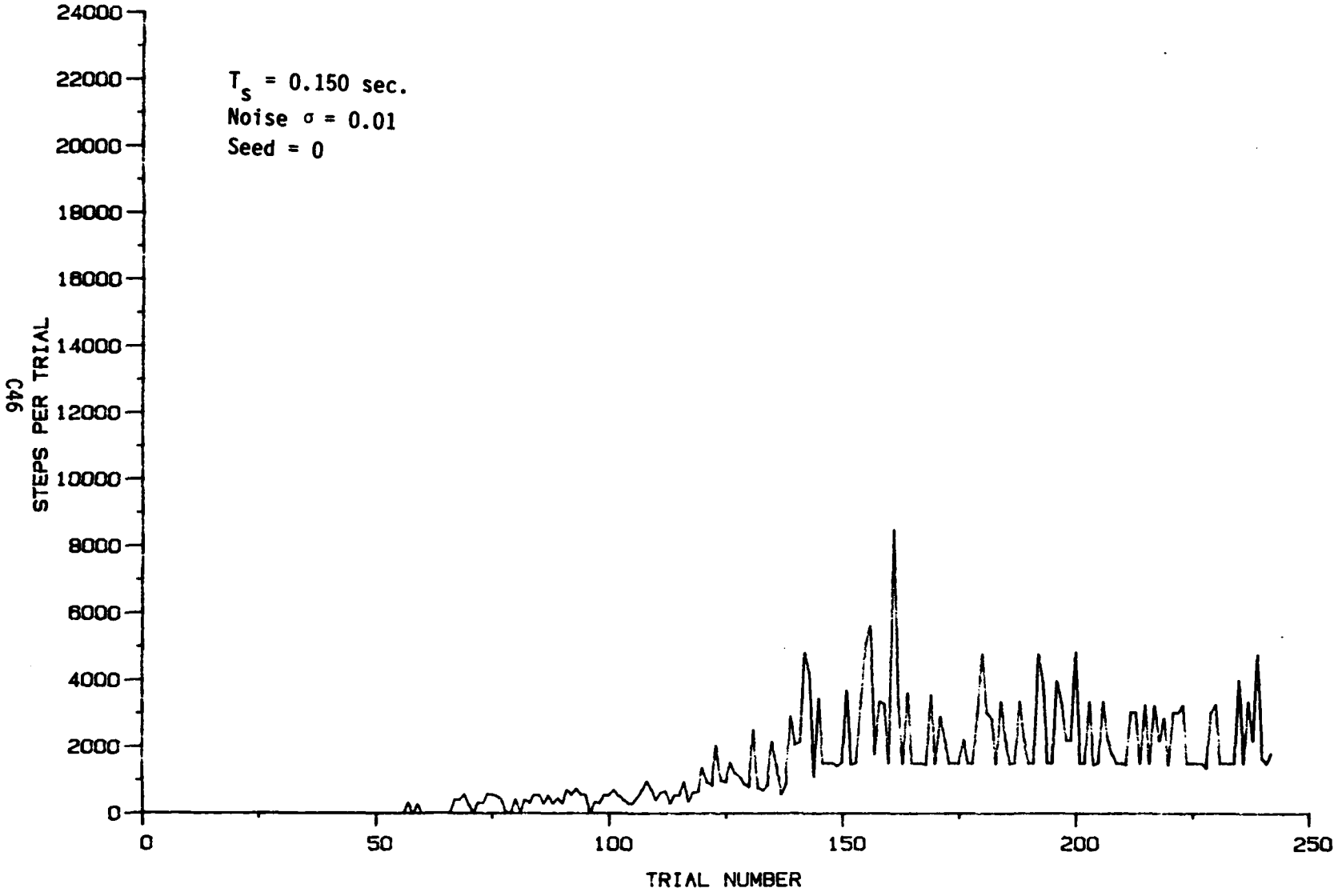


Figure 83. ALC Learning Curves with Parameters Indicated.

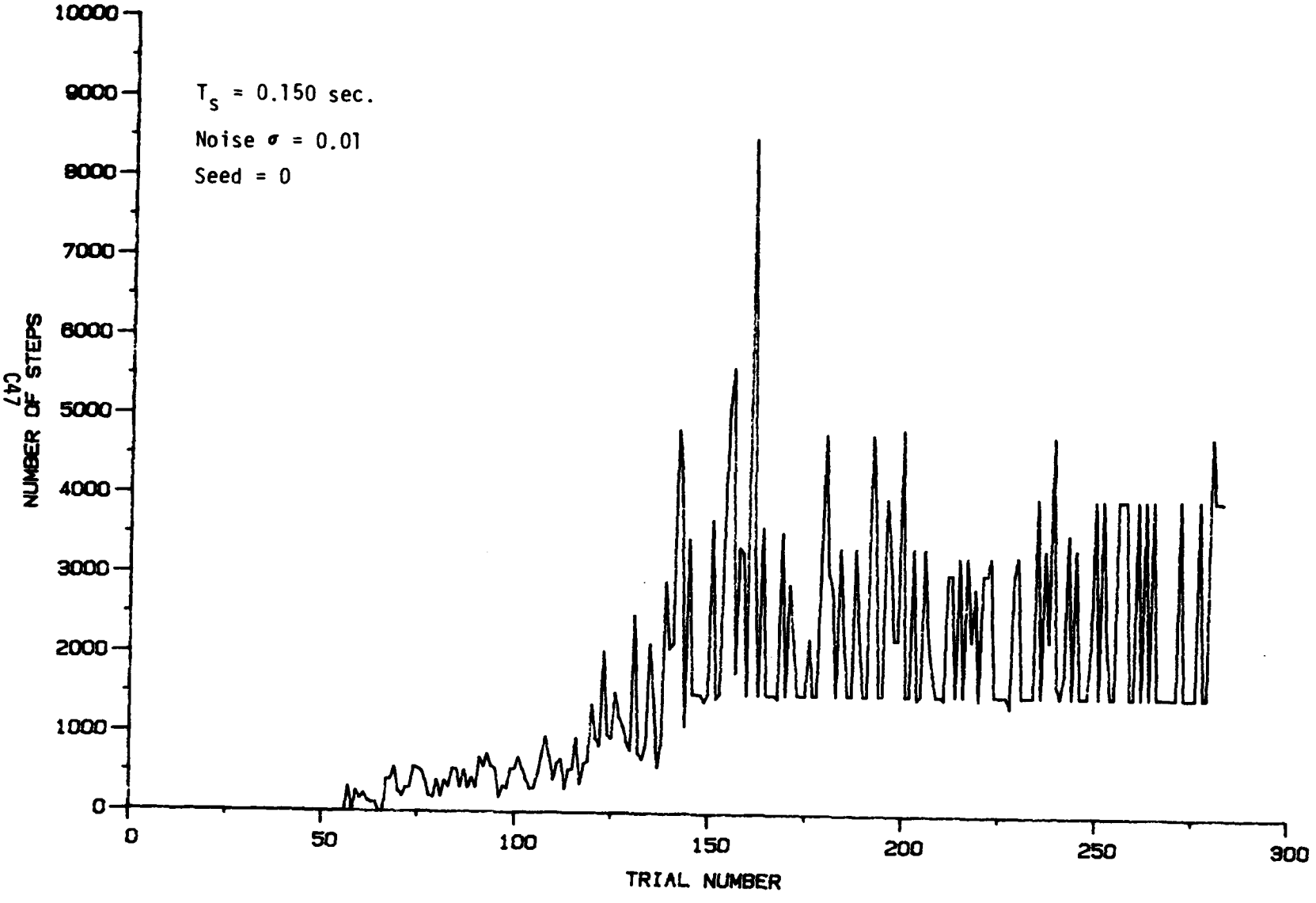


Figure 84. ALC Learning Curves with Parameters Indicated.

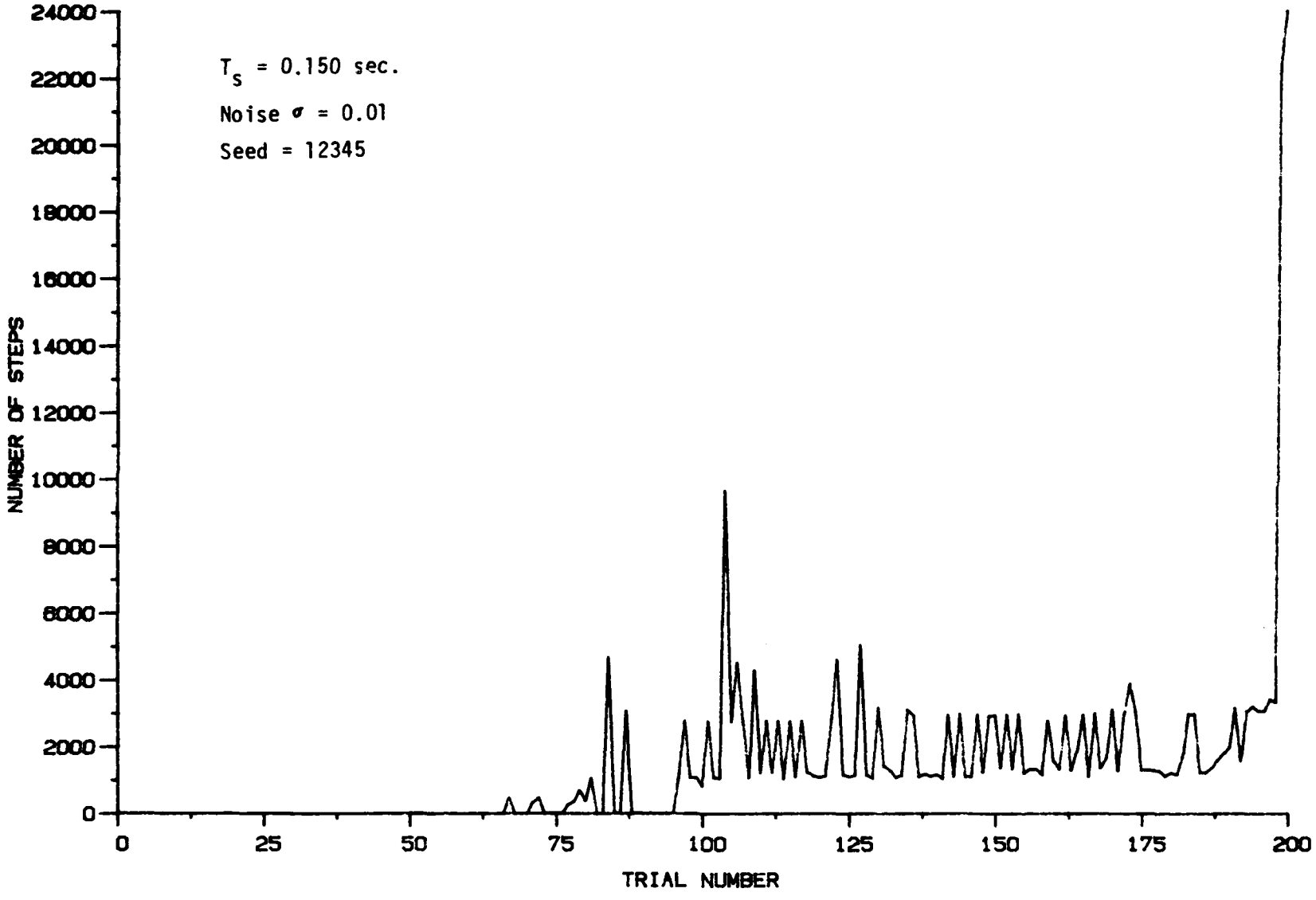


Figure 85. ALC Learning Curves with Parameters Indicated.

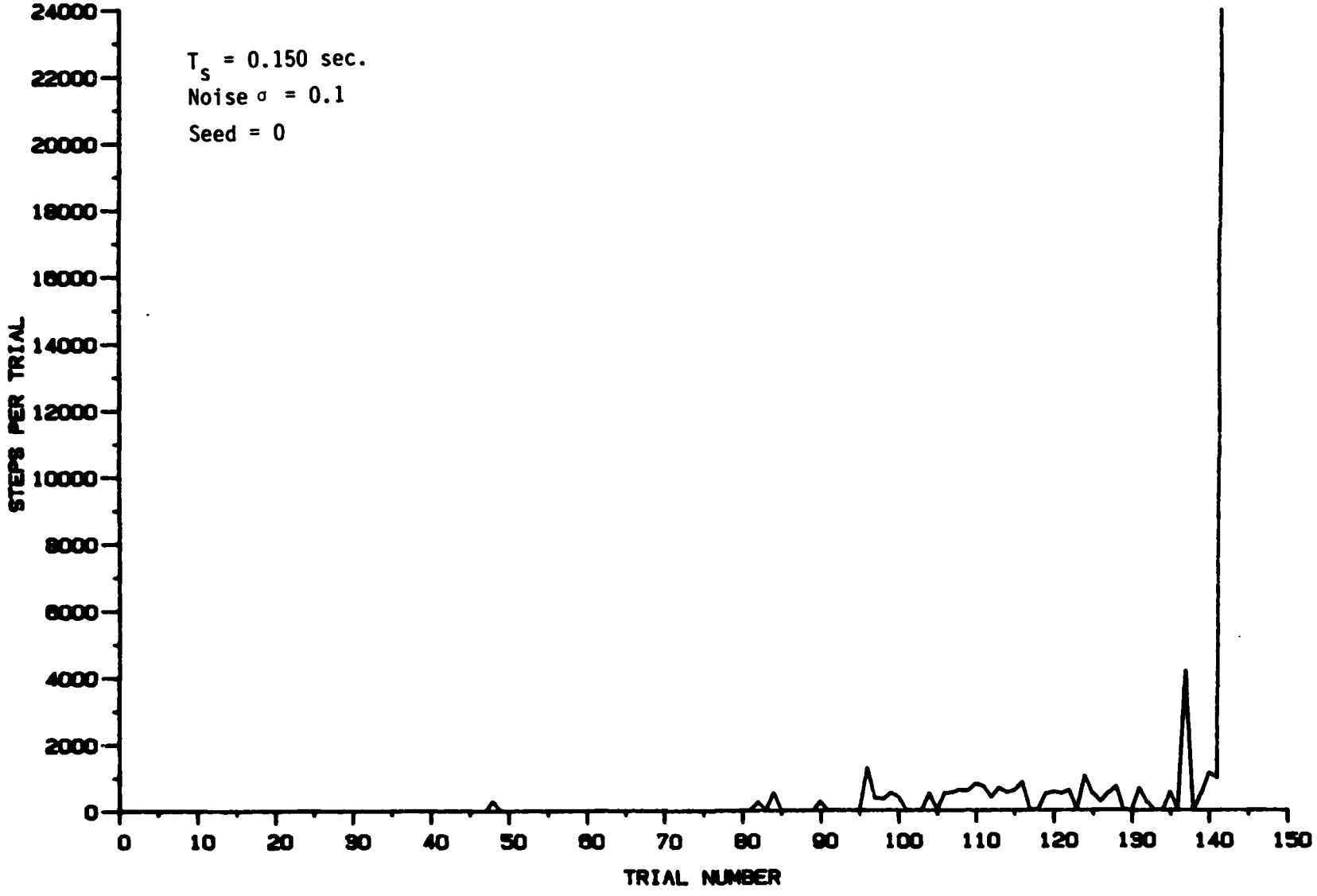


Figure 86. ALC Learning Curves with Parameters Indicated.

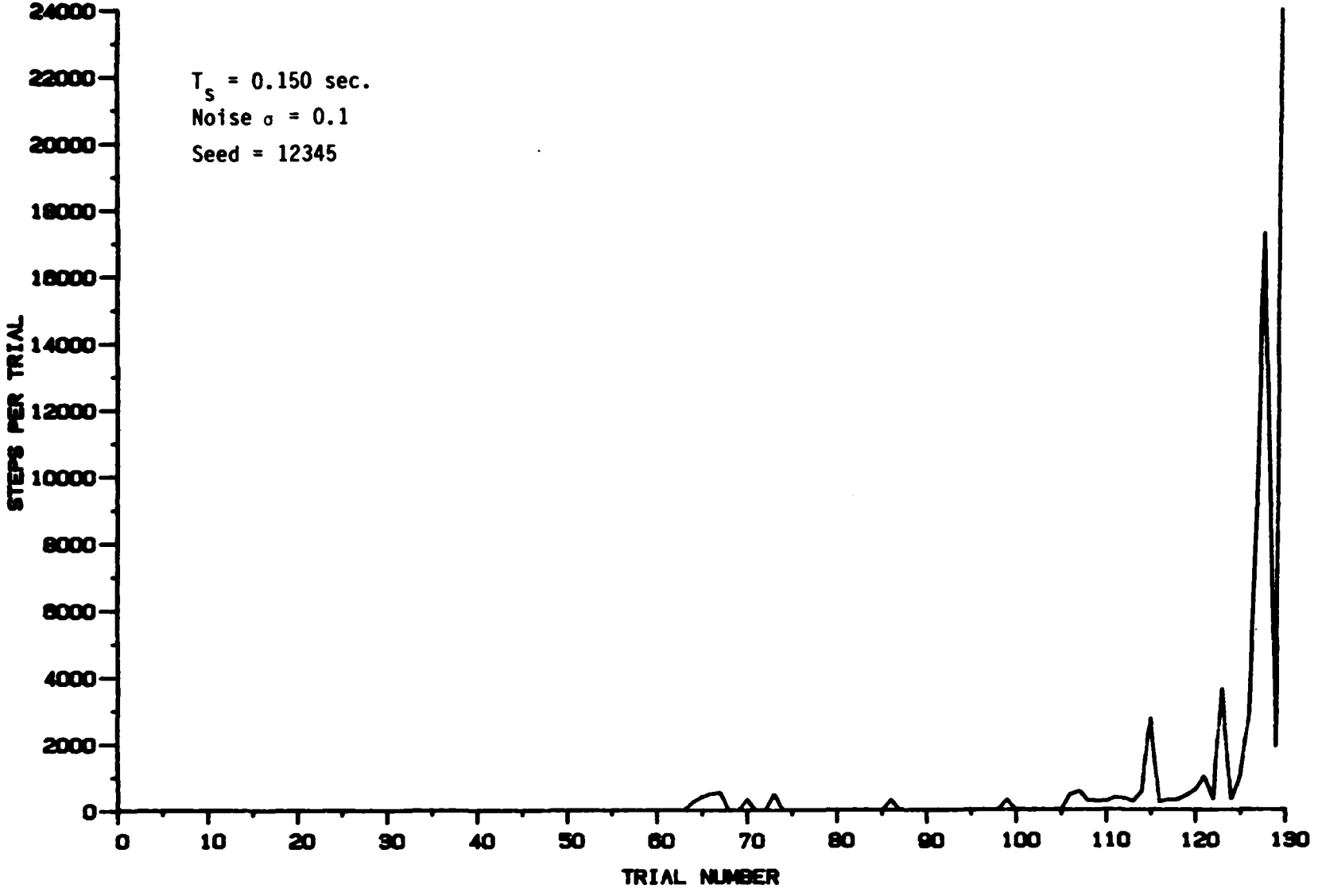


Figure 87. ALC Learning Curves with Parameters Indicated.

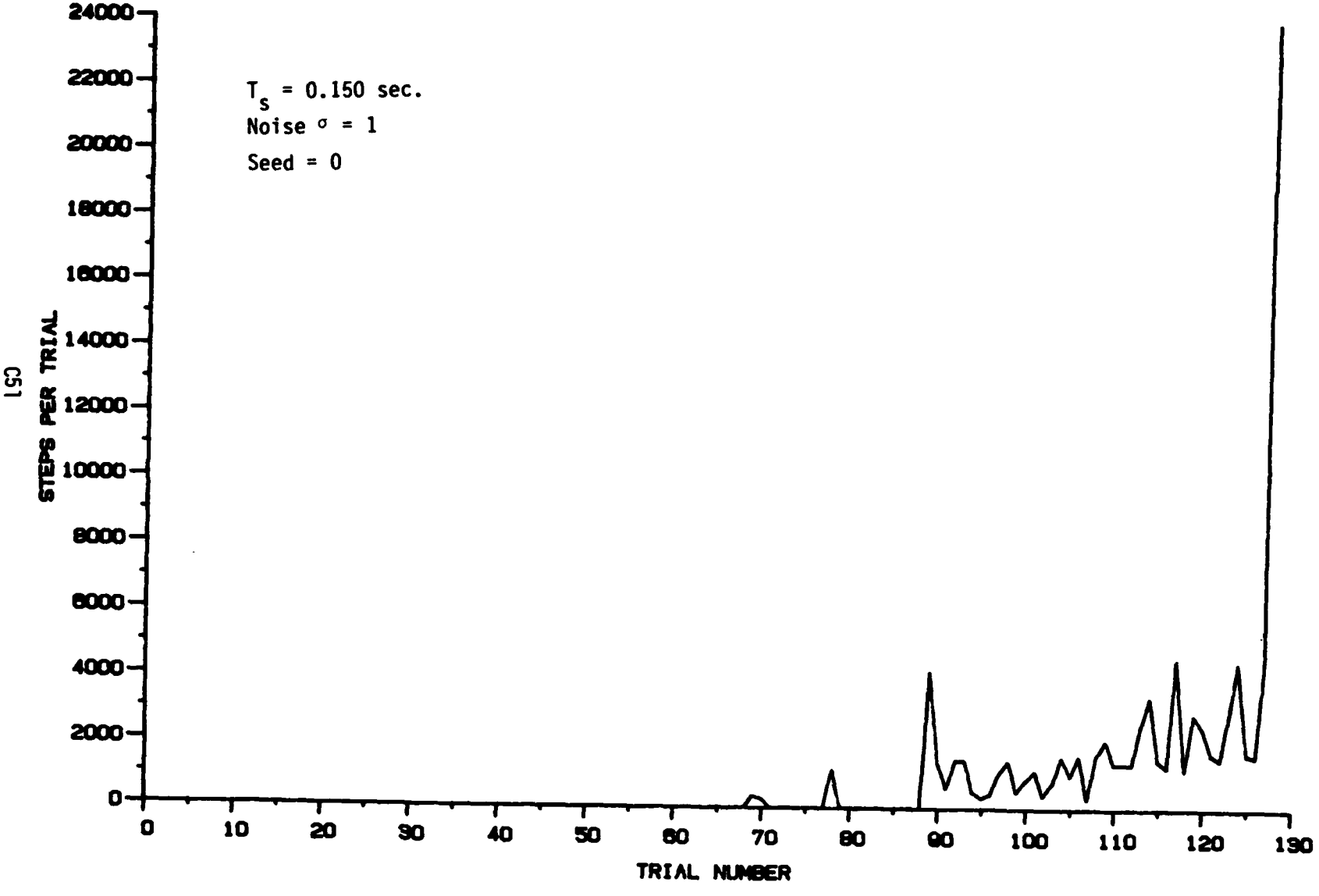


Figure 88. ALC Learning Curves with Parameters Indicated.

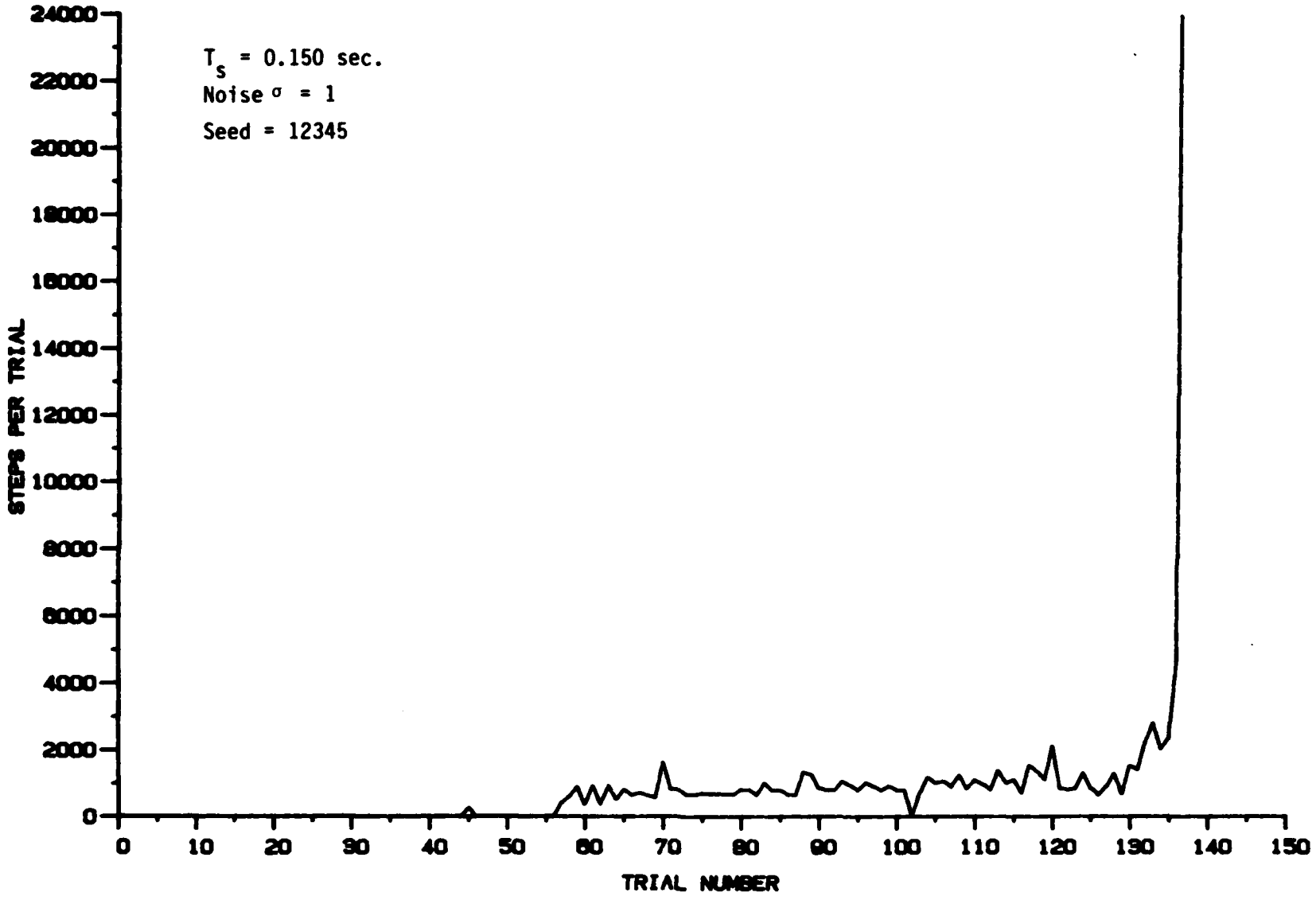


Figure 89. ALC Learning Curves with Parameters Indicated.

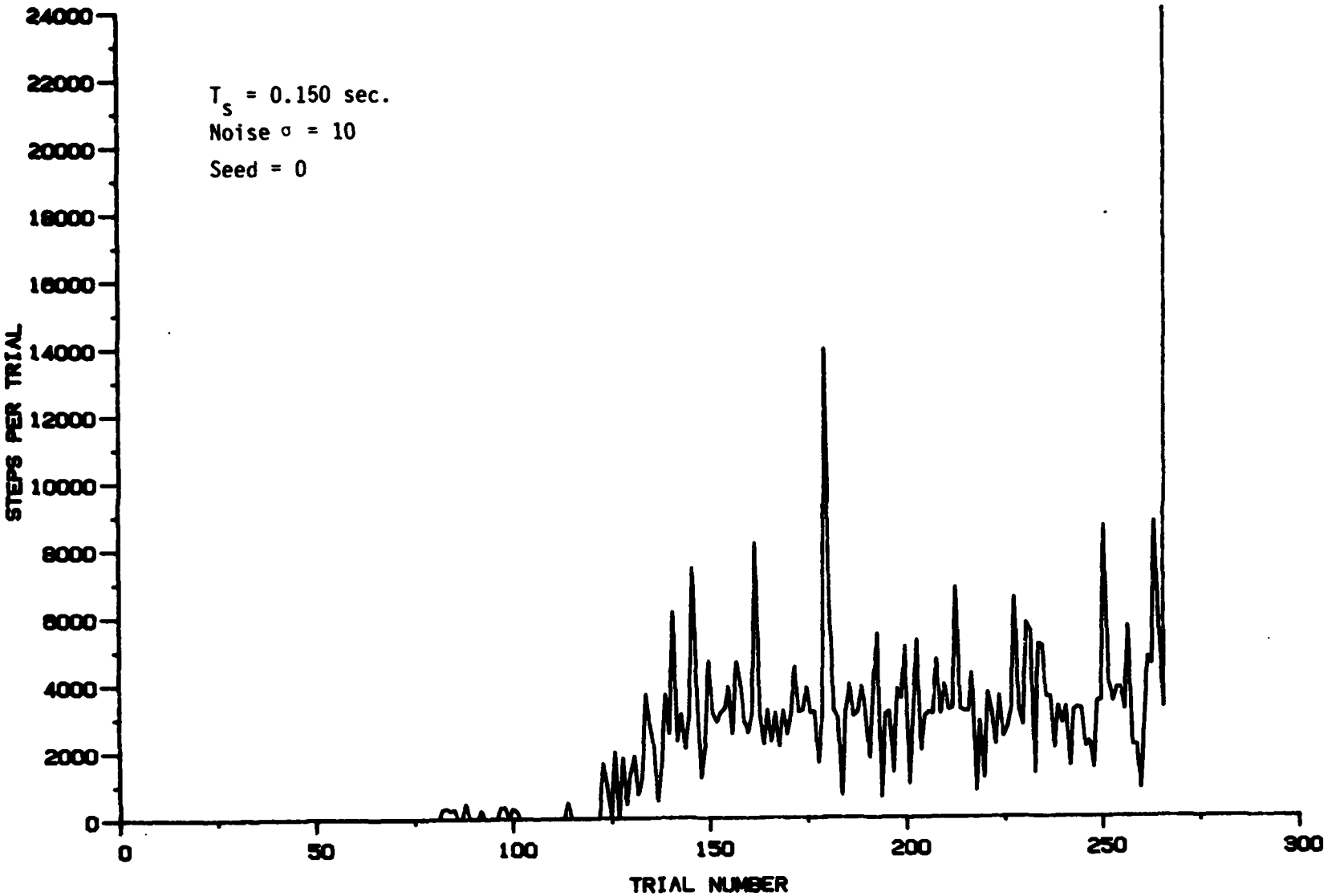


Figure 90. ALC Learning Curves with Parameters Indicated.

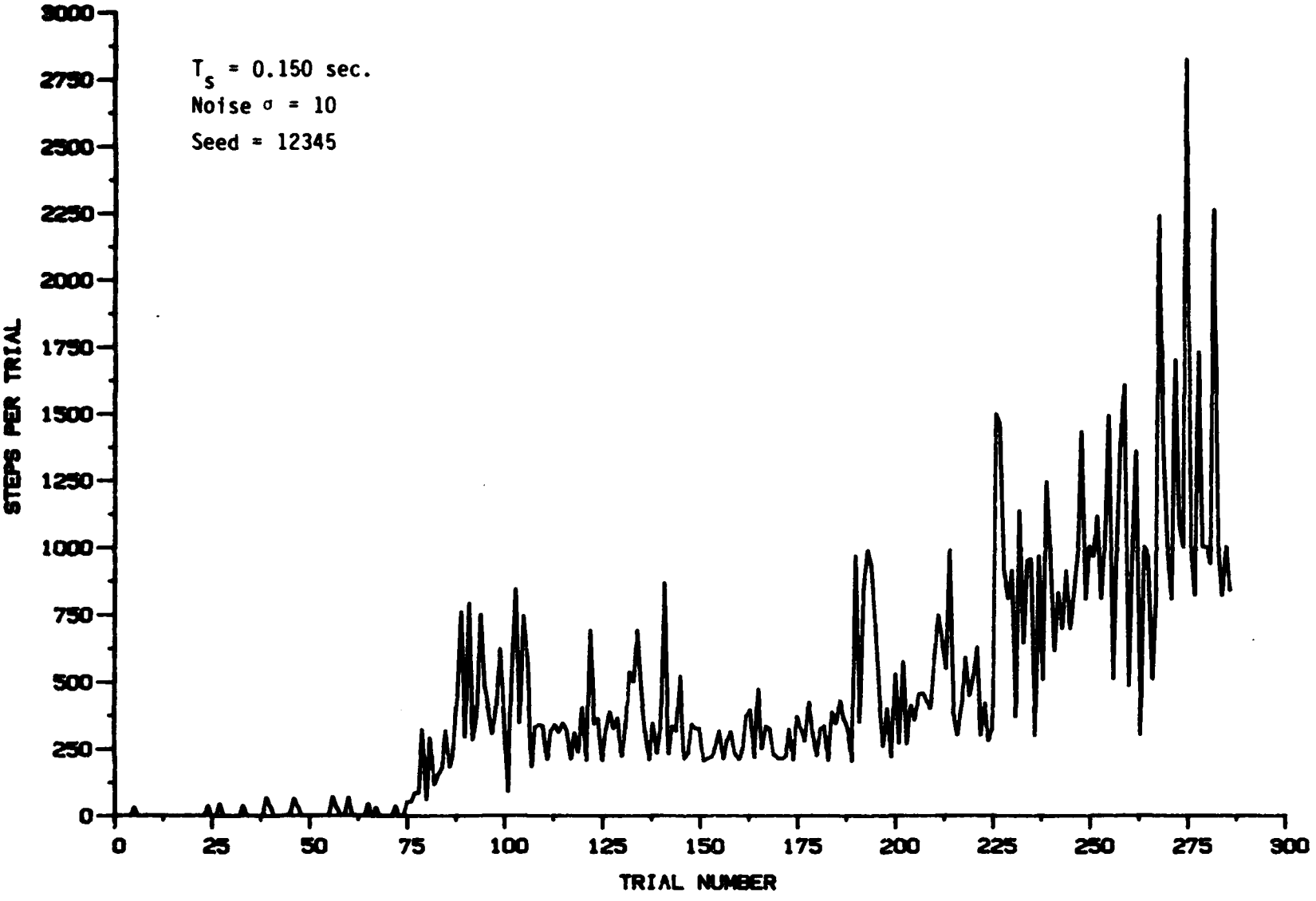


Figure 91. ALC Learning Curves with Parameters Indicated.

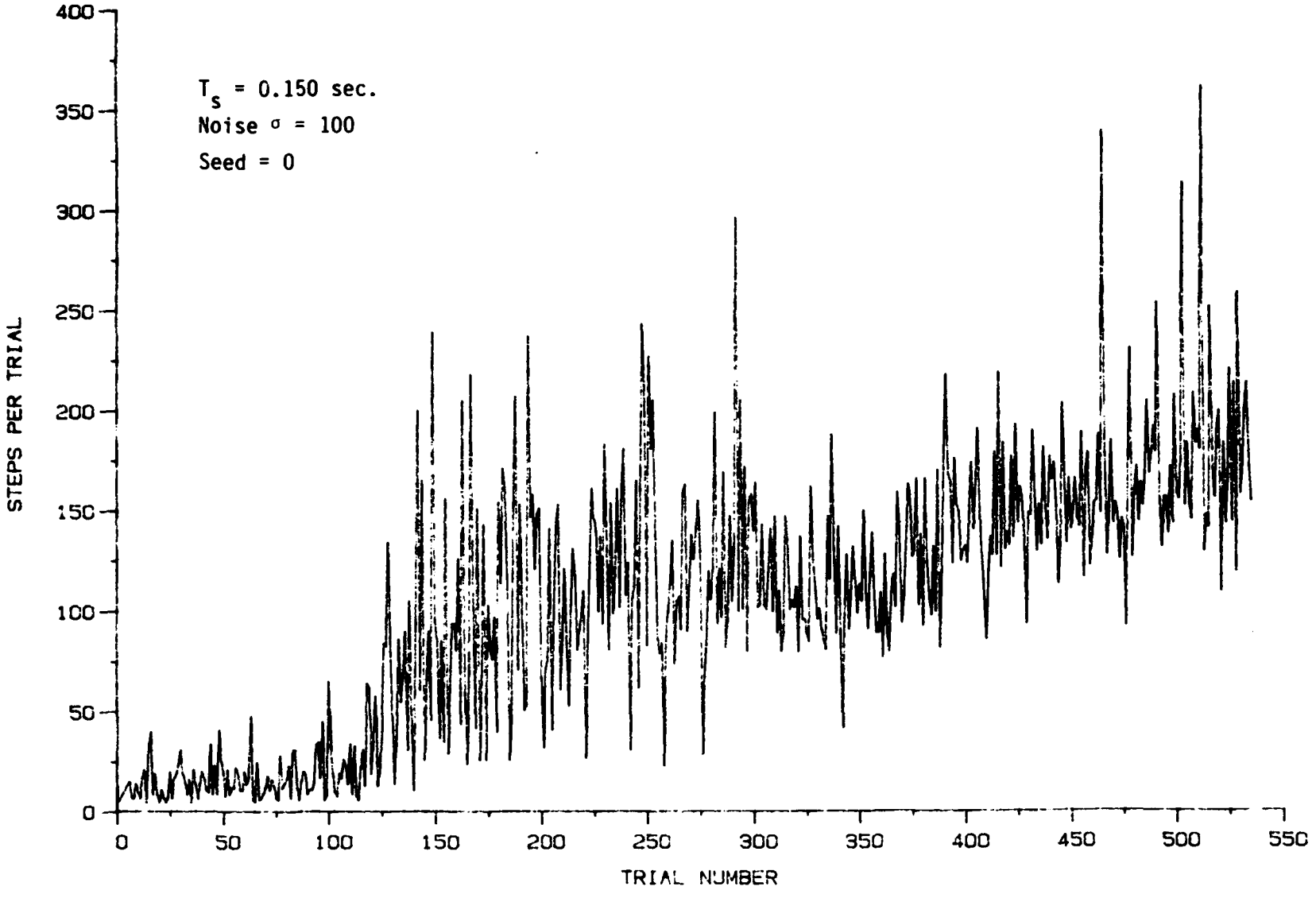


Figure 92. ALC Learning Curves with Parameters Indicated.

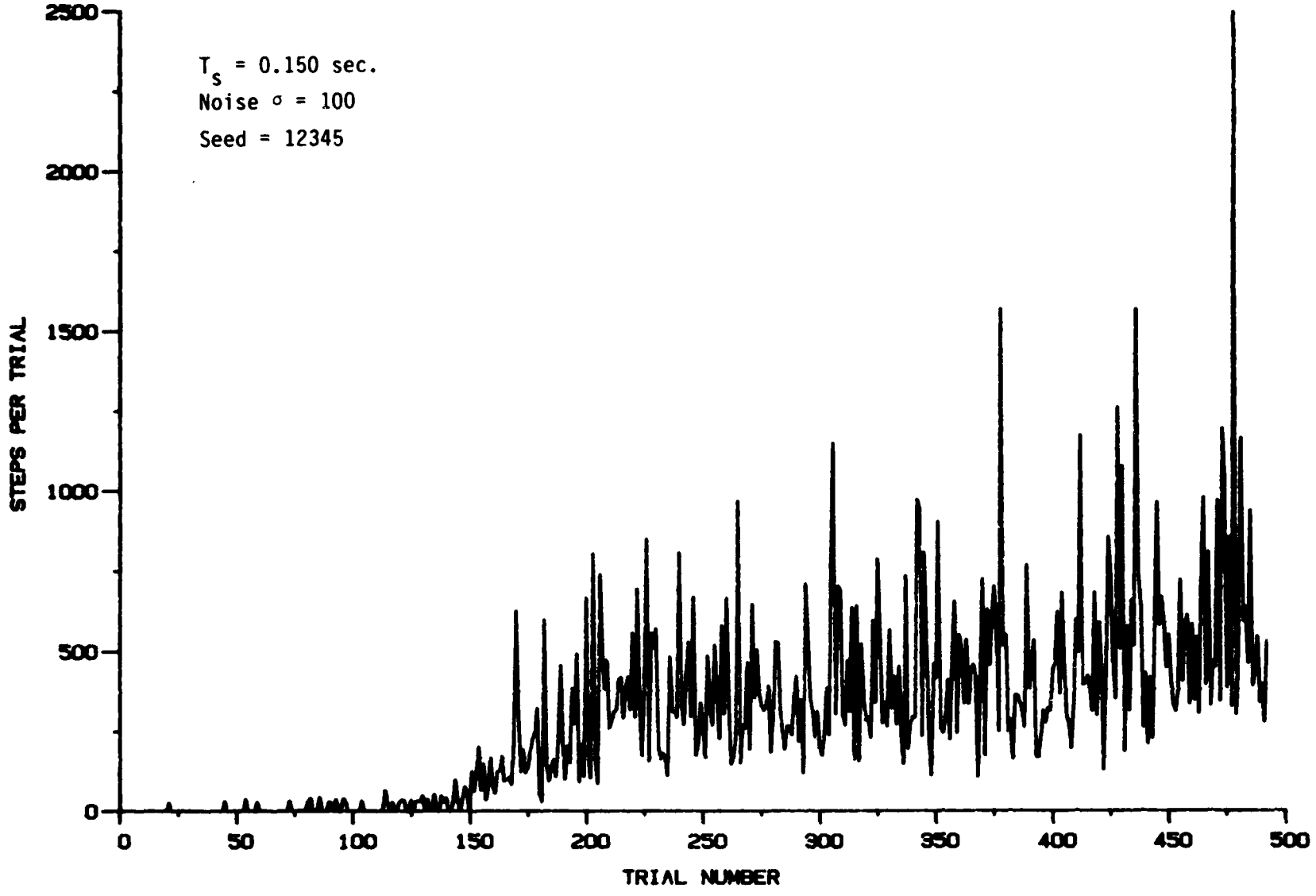


Figure 93. ALC Learning Curves with Parameters Indicated.

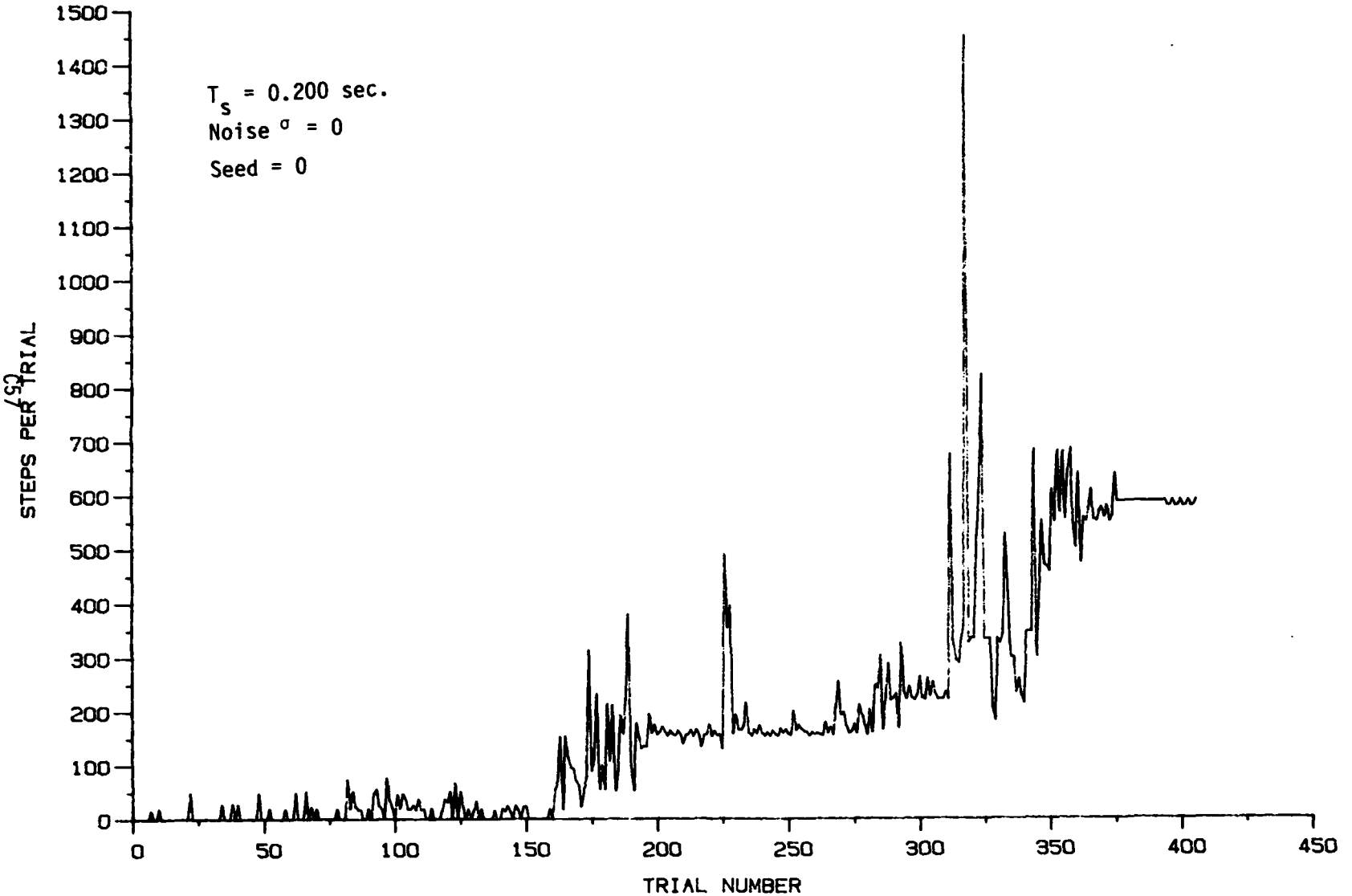


Figure 94. ALC Learning Curves with Parameters Indicated.

END

FILMED

7-85

DTIC