

1985

Trajectory tracking control method for nonlinear systems /

Athanasios Kehagias
Lehigh University

Follow this and additional works at: <https://preserve.lehigh.edu/etd>



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Kehagias, Athanasios, "Trajectory tracking control method for nonlinear systems /" (1985). *Theses and Dissertations*. 4592.
<https://preserve.lehigh.edu/etd/4592>

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

TRAJECTORY TRACKING CONTROL METHOD

FOR NONLINEAR SYSTEMS

by

Athanasios Kehagias

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Applied Mathematics

in

the Mechanical Engineering Department

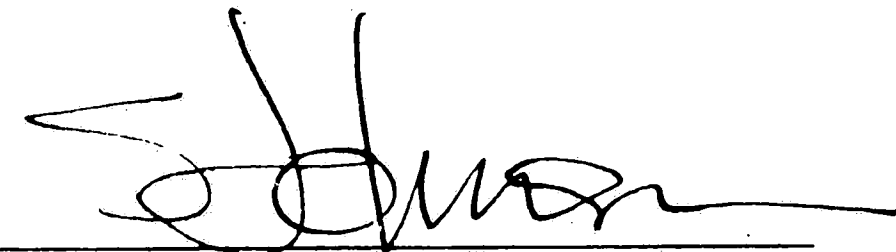
Lehigh University

1985

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

12/16/85

(date)



Professor in Charge

F. Eudozan

Chairman of the department

ACKNOWLEDGEMENTS

Hearty thanks to Professors Stanley Johnson and Gary Harlow, from the Mechanical Engineering Department. Also to the people from the Department for the Application of Mathematics. Writing this thesis has been very frustrating, but also, lots of fun.

Thanasis Kehagias,

Lehigh University, December 1985

TABLE OF CONTENTS

LIST OF CHAPTERS

Abstract.....	1
Chapter 0: Introduction.....	2
Chapter 1: Preliminaries.....	5
Chapter 2: Stability.....	37
Chapter 3: Time optimality.....	65
Chapter 4: Simulations.....	80
References.....	119
Appendix.....	123
On notation.....	125
Vita.....	126

LIST OF FIGURES

Fig. 1.1.1 A driven pendulum.....	7
Fig. 1.2.1 Limit cycles due to variation of parameters...	10
Fig. 1.3.1 Time optimal control trajectory.....	15
Fig. 1.3.2 Trajectory with constant input.....	16
Fig. 1.3.3 Trajectory with square pulse input.....	17
Fig. 1.3.4 2-interval spline.....	21
Fig. 1.3.5 4-interval spline.....	25
Fig. 1.3.6 8-interval spline.....	29
Fig. 1.3.7 12-th order polynomial.....	33
Fig. 2.3.1 Guaranteed stability region.....	54
Fig. 3.1.1 Effect of t_s variation.....	74
Fig. 3.2.1 Plot of total time t_f vs. input U	78
Fig. 4.1.1 Ideal conditions simulation.....	83
Fig. 4.1.2 Different initial conditions simulation.....	89
Fig. 4.1.3 Different switching time simulation.....	94
Fig. 4.1.4 Difference in modelling function simulation..	100
Fig. 4.1.5 Long term tracking simulation.....	106
Fig. 4.1.6 Square pulse input simulation.....	111

ABSTRACT

A method is proposed for the control of nonlinear systems. The method consists in computing and approximating desired trajectories in the state space, and then tracking these trajectories. The method is applied to second order nonlinear systems. A stability analysis is performed, in which the attractive region of the system is computed by the use of a numerical technique. It is proved that if the nonlinearity is bounded and proportional corrective control is applied to the system, the parameters of the controller can be chosen so as to guarantee stability. The case of time optimal control is examined in particular detail, and estimates of the performance deterioration are given. In particular, it is shown that if the system tracks the desired trajectories closely enough, the deterioration of performance is only due to the fact that the available control input is split in two parts: a bang-bang component, and a corrective component. Further, a relationship is developed to approximately compute the increase of total time required to reach a target state. Finally, extensions to higher order systems are discussed.

CHAPTER 0: INTRODUCTION

0.1 MOTIVATION : Industrial manipulators (robots) are strongly nonlinear systems. As interest in them has been increasing in the last years, so has the awareness that we do not have a general method for the control of nonlinear systems. The case of time optimal control is of particular interest. In that case we obviously want to drive the manipulator at high speeds that exacerbate the nonlinearities.

In this document we will present a control method for nonlinear systems. The task we want to accomplish is to get from one initial state (i.e. one point of the state space) to another final state . Our main concern will be with gross motion control, that is control of movement over long distances, at high speeds . In gross motion control we are trying to bring the system close to the target state. Once the system is in a neighborhood of the target state, it can be operated at lower speeds, in a linear terminal control mode that can be as accurate as necessary.

The main idea of our method is that we track in the state space the trajectory that describes the desired response. More specifically, our method consists of the following stages:

1. We formulate a model of the physical system under investigation and we obtain a control law that will get us from the initial to the final state. There may be more than one such control and we may be interested in choosing that control that will also ensure additional properties, e.g. time optimality.

2. We simulate the system using the model and the appropriate control law and record the trajectory it follows in the state space.

3. We store the coordinates of the trajectory for future reference. We use some form of function approximation to facilitate this.

4. We operate the actual physical system using the precomputed control of stage 1 and an additional corrective control that depends on the error, i.e. the difference between the recorded simulated trajectory and the actual trajectory of the physical system. This corrective control can be, for example, proportional to the error.

This permits a linearization around the nominal trajectory. We show that for a certain class of systems the error is guaranteed to be bounded. A case of particular interest is time optimal control, where we want to minimize the time required to get from initial to final state without too greatly missing the target final state.

In this document we will deal with second order systems, discrete or continuous in time. The major thrust of our analysis is in setting the foundation for a practical scheme for the control of real, high-order, strongly nonlinear industrial manipulators. We are particularly interested

in time optimality and the usefulness of superimposing additional control methods (namely, tracking control) onto bang-bang control. We also want to keep on-line computations at a minimum so as to permit adequately rapid sampling rate. We will make maximum use of off-line computation, that can demand significant resources, but does not slow down or complicate the actual on-line control process. So we will be interested in this trade-off of on- and off-line computation.

CHAPTER 1: PRELIMINARIES

1.1. THE SYSTEM: We will look at the following type of systems:

1.1.1 $\dot{x}_1 = x_2$

1.1.2 $\dot{x}_2 = f(x_1, x_2) + u$

This is the equation of motion of a mechanical system of one degree of freedom, where f is a function of $x_{1,2}$, generally nonlinear. In addition we will assume that $|f| < A_1$. If we were considering actual industrial manipulators, we would be dealing with at least three degrees of freedom and all the additional coupling terms between these three degrees of freedom. In that case we would have three groups of two equations each, like (1.1.1-.2), one for each degree of freedom. In all we would have six state variables.

The coupling applied to one degree of freedom can actually be modelled as part of the nonlinear function f , except that f will be a function of all six state variables in the general case. In particular, we can visualize an extension of the 2-D method to higher dimensional cases, where the coupling

terms can be considered as disturbances, uncertainties (in the sense of df that we mention in Chapter 2) etc. If we can justify a boundedness assumption for these extra terms, our analysis of the 2-D case is highly relevant for higher dimensional systems. However, in this work we will limit ourselves to the one degree of freedom system. A particular example we will use a lot in the following is

$$1.1.3 \quad \dot{x}_1 = x_2$$

$$1.1.4 \quad \dot{x}_2 = A \sin x_1 + B x_2 + u$$

These are the equations of a damped nonlinear pendulum with torque applied to control its angular position (see Fig. [1.1.1]). Of course this can be thought of as an elementary industrial manipulator.

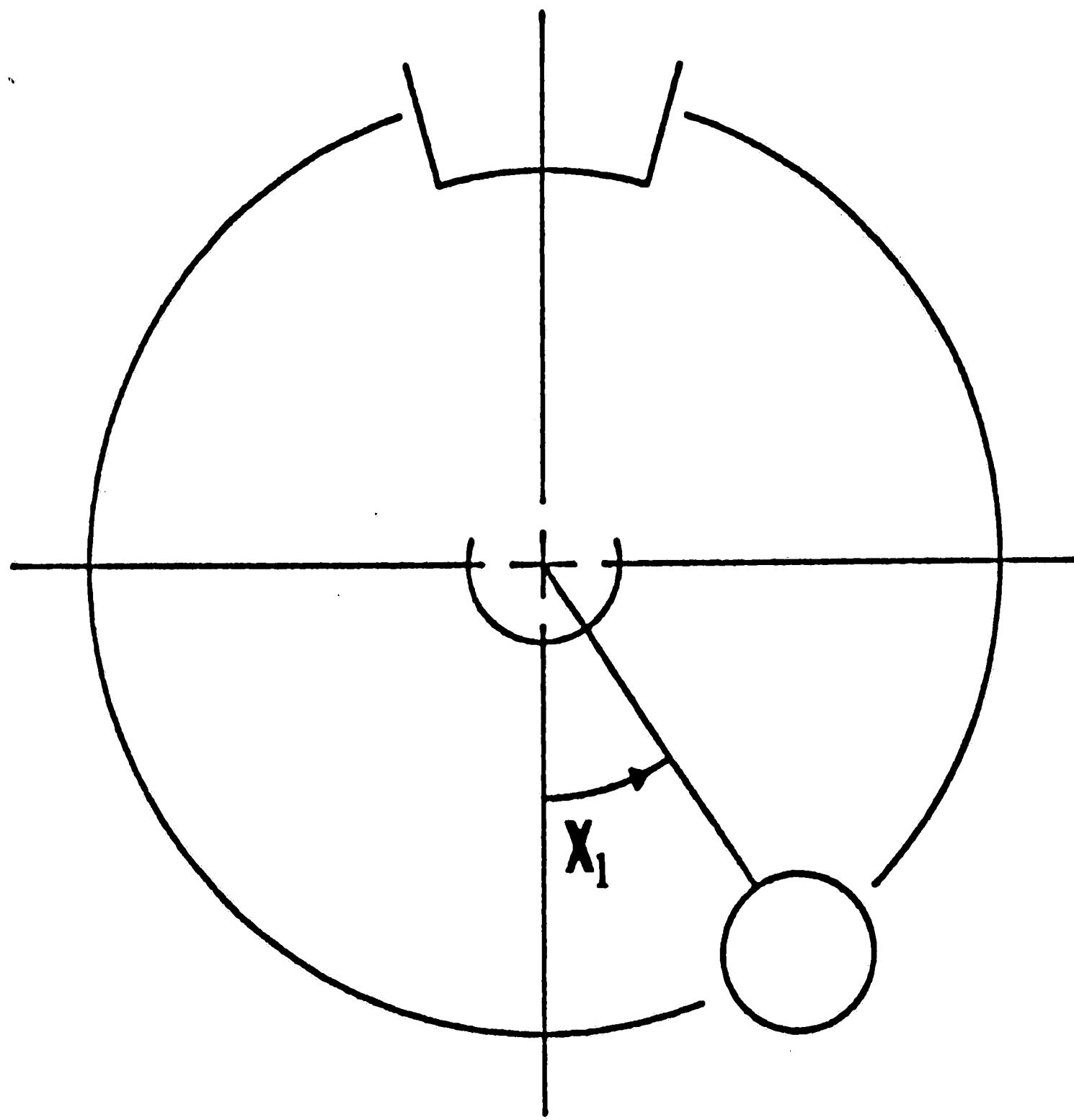


Figure 1.1.1: A driven pendulum

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = A \sin x_1 + B x_2 + u$$

1.2 BANG-BANG CONTROL: Let us now implement the second step of our method. Assume we want to find a control $u(t)$ ($0 < t < t_f$) with $|u(t)| < U$, so as to get from state $\underline{x}_i = \underline{x}(0)$ to state $\underline{x}_f = \underline{x}(t_f)$ in such a way that t_f is minimum. There is a classical real world problem associated with industrial manipulators and productivity improvement that can be cast in this form: given that the actuators of a manipulator have a bounded force/torque output, find the input schedule that will minimize the time it takes the end effector to get from one rest position to another.

The system is linear in the input and so, using Pontryagin's maximum principle, we can find the control $u(t)$ (see e.g. [1,2]). It turns out that

$$1.2.1 \quad u(t) = U \operatorname{sgn}(g(\underline{x}))$$

Where $\operatorname{sgn}(x)$ is the signum function:

$$1.2.2 \quad \operatorname{sgn}(g(\underline{x})) = +1 \text{ if } g(\underline{x}) > 0$$

$$1.2.3 \quad \operatorname{sgn}(g(\underline{x})) = -1 \text{ if } g(\underline{x}) < 0$$

Obviously, u can take two values only, U and $-U$. In addition (see for example Athans [3]) for a second order system linear in the control, there

is only one switch (at time t_s) from one value to the other. In principle, we can solve the problem either analytically or numerically and determine what the switching time t_s will be. Then, when we operate the physical system we can switch the sign of the control at time t_s . This is an open loop control scheme, since there is no feedback, and is fraught with all the problems of open loop control, in particular sensitivity to parameter variations. It has been shown (e.g. Dubowsky [4]) that small errors in the timing of the switching can lead to large errors in the final state.

The other traditional way to implement bang-bang control is to compute the value of $g(x)$ at successive times and when that changes sign to switch the sign of the control as well. However, small variations in the system parameters can lead to limit cycles, an infinite number of switchings and the system's never reaching the target state. See , for example, Fig.[1.2.1]. Such considerations are the subject of several papers [6-11].

Our scheme follows a middle road between the two traditional schemes. We have a totally preprogrammed control component, which in the ideal case ensures time optimality and reachability of the target state. And we also have a corrective control that ensures that the actual system state never drifts too far away from the nominal trajectory. We will deal more extensively with the time optimality question and related topics in Chapter 3.

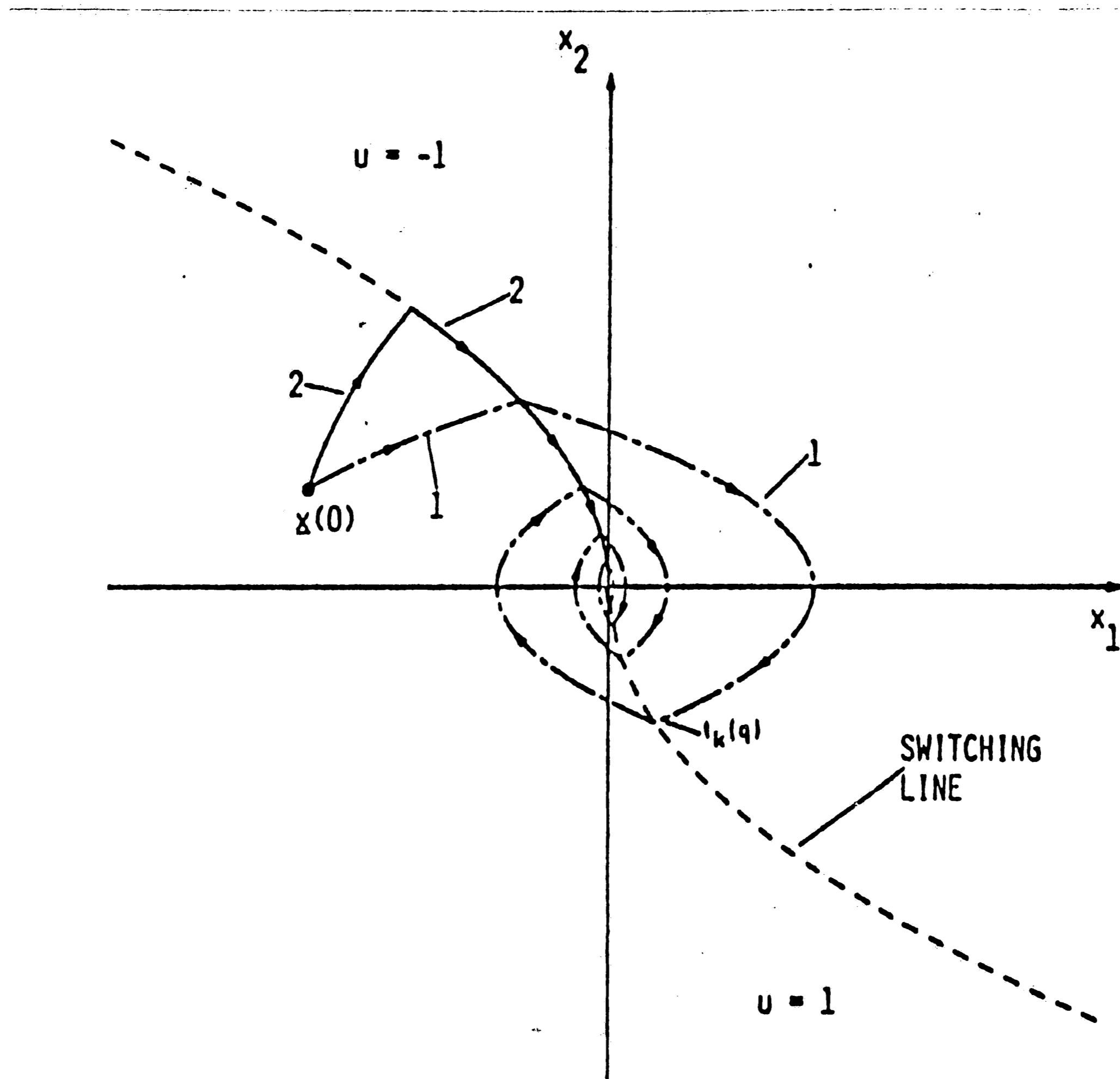


Figure 1.2.1: Limit cycles due to parameter variation

Note it takes an infinite time for the system to reach the target state.

1.3 SIMULATION AND APPROXIMATION: In this section we deal with the preliminaries necessary to apply the control method proposed. We simulate the system (1.1.3-.4) on a digital computer and record the trajectory. In fact we perform several simulations, differing in the specific control function applied.

1. First we look at the optimal time problem. If the initial and target states ($\underline{x}_i, \underline{x}_f$ respectively) and the control bound (U in 1.2.1) are given, then the control is completely characterized by the switching time t_s . We find this by integration of the eqs.(1.1.3-.4) forward in time from initial value \underline{x}_i with $u=U$ and backwards from initial value \underline{x}_f with $u=-U$. We get the solutions $\underline{y}(t)$ and $\underline{w}(t)$ respectively. We check to see at what times t_s and t_{sb} $||\underline{y}(t_s) - \underline{w}(t_{sb})|| < e$, where e is a small enough positive number. Ideally, $\underline{y}=\underline{w}$, but since this is a numerical method we can only examine distinct points. The two trajectories intersect (or, anyway, are closer than e) at the point $\underline{y}=\underline{w}$, which however is characterized by a different time variable for each of the two trajectories. So we have to transform t so that t_{sb} transforms to t_s . The transformation used is $t=t+t_s-t_{sb}$. This ensures, as the reader can check, that if at time t_s we switch the sign of u the system will move forward in time to the target state \underline{x}_f . The trajectory is also illustrated in Fig.1.3.1.

2. The second input we use is constant: $u(t)=U$ for $t>0$. With this input we generate a trajectory to test the tracking performance for longer time intervals, where errors and instabilities accumulate. This trajectory is illustrated in Fig.(1.3.2).

3. In the third case the input is a square pulse train. The motivation in the selection of this input is to test the tracking performance for a trajectory which has lots of transients. This trajectory is illustrated in Figs. (1.3.3a) (state space plot) and (1.3.3b-d) (as a function of time).

Of course we could go on, testing our method with lots of different inputs, but the three inputs we use are a fairly representative sample.

Obviously the simulations are performed on a digital computer and the time domain is discrete. Now, we have the values of $x_{1,2}$ at the times t_i , $i=1,2,\dots$ which we want to use as a reference input for the tracking phase of our method. So we have two options: either (1) store this amount of numerical data, which can be quite significant, especially if we want to apply the method to larger scale systems, or (2) find a suitable approximation scheme which we will use to obtain values of $x_{1,2}$ at times t_i .

We experimented with several sets of approximating functions, such as polynomials, exponentials and splines. The splines gave the best approximation by far, with quite modest parameter storage requirements. Namely, (see, for example, Ahlberg et.al. [12]), a spline of n intervals is completely characterized by $3n+4$ parameters. A 4-interval spline is

characterized by 16 parameters. If each parameter can be stored in two bytes of computer memory all the data we need can be stored in $2 \cdot 16 = 32$ bytes. Also the computation necessary for evaluating a spline at a point is modest: we need four additions and three multiplications for one evaluation.

We list these facts to show that the reference input for our control scheme can be computed quickly, with modest demands in computing power. This is crucial, because it shows we can implement a control scheme with little hardware (e.g. a microprocessor, a few hundred bytes of memory) and accomodate a fast sampling rate.

There is something else that is nice about splines: they guarantee approximation not only of the function but also of its first and second derivatives. The relevant error equations are

$$1.3.1 \quad |S_{D_k}^{(p)} - f^{(p)}| = O(|D_k|^{2-p}) \quad p=0,1,2$$

where S_{D_k} is the spline of intervals D_k and $f^{(p)}$ are the approximated function, its first and second derivatives, and D_k is the length of the k -th interval. But actually these bounds are really conservative. The IMSL spline fitting routine that we used provided error estimates which were significantly below the bounds of eqs.(1.3.1-.3). Besides it is easy to check the error during the off-line computation phase. In Fig.[1.3.4-.7] we

present the approximation of one trajectory of the system of eqs. 1.1.3-4. We can see the error is virtually negligible. Note, in these figures, that we present 2-, 4- and 8- interval splines. For comparison purposes we also present an approximation by a 12-th order polynomial. This polynomial is characterized by twelve coefficients, that is fewer than those needed for a spline, but it takes twelve multiplications and thirteen additions to evaluate it. And, visibly, the fit is much worse than that of the 4- and 8- interval splines.

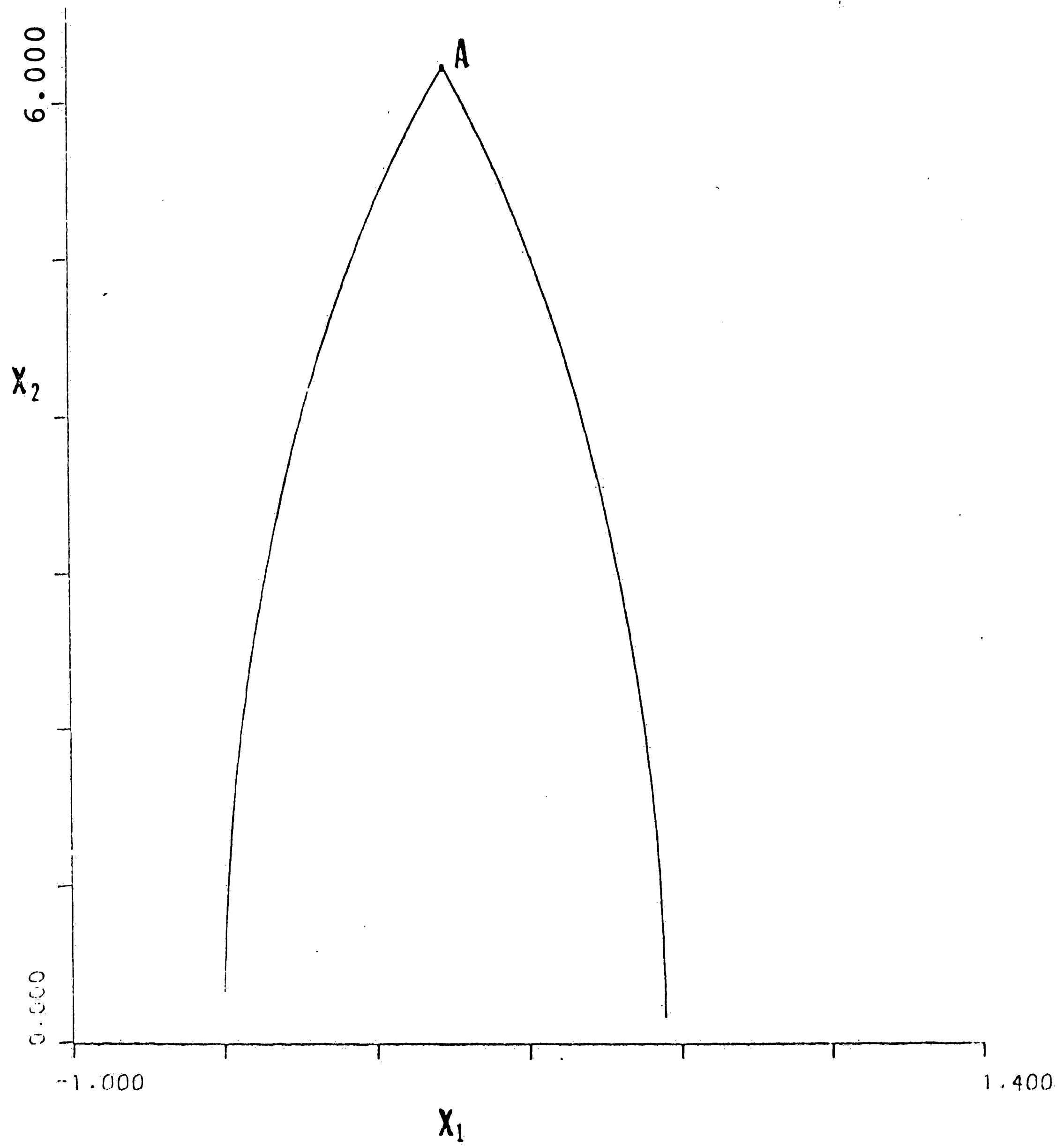


Figure 1.3.1: Time optimal control trajectory

Switching takes place at point A

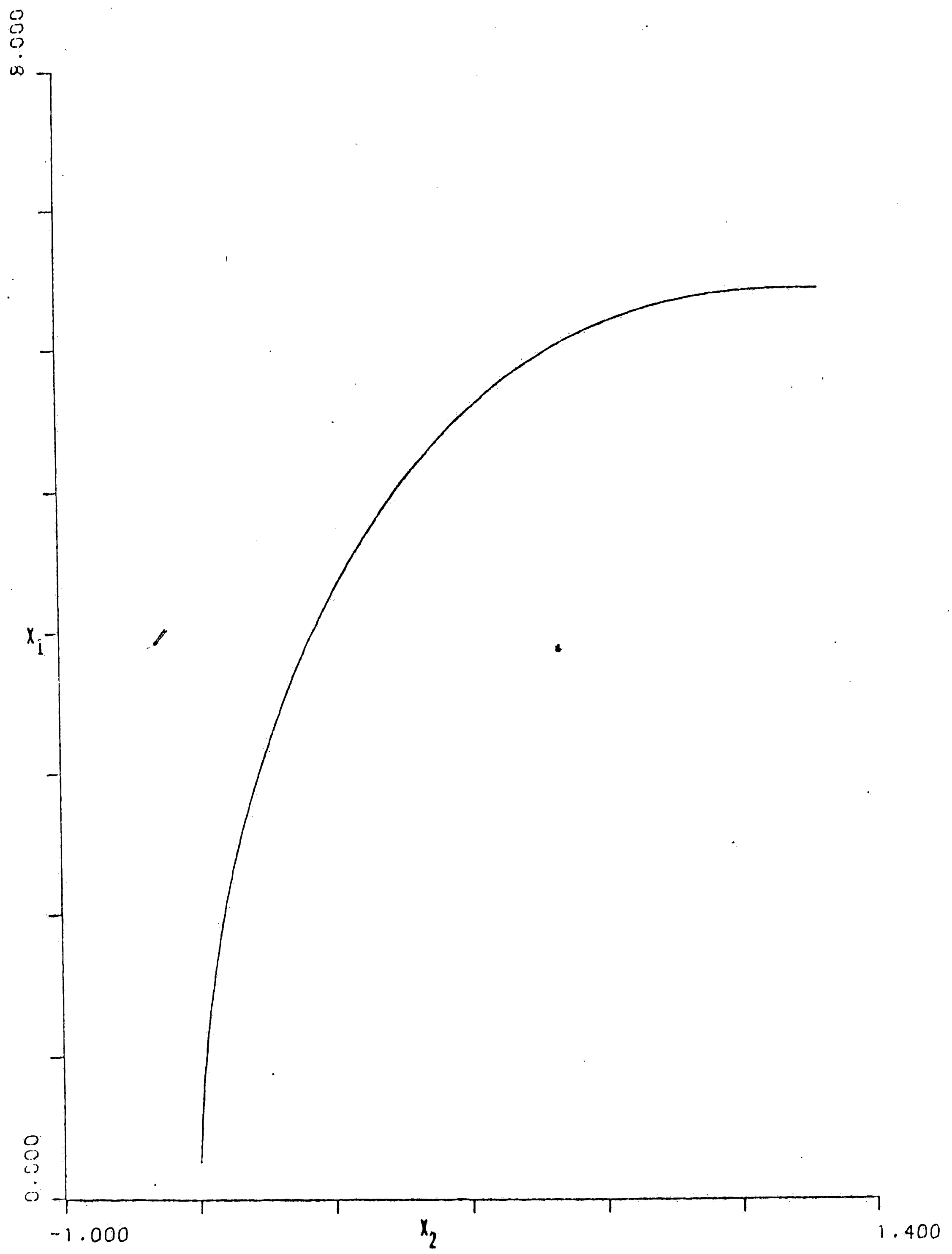


Figure 1.3.2: Trajectory with constant input

($u=16.9$)

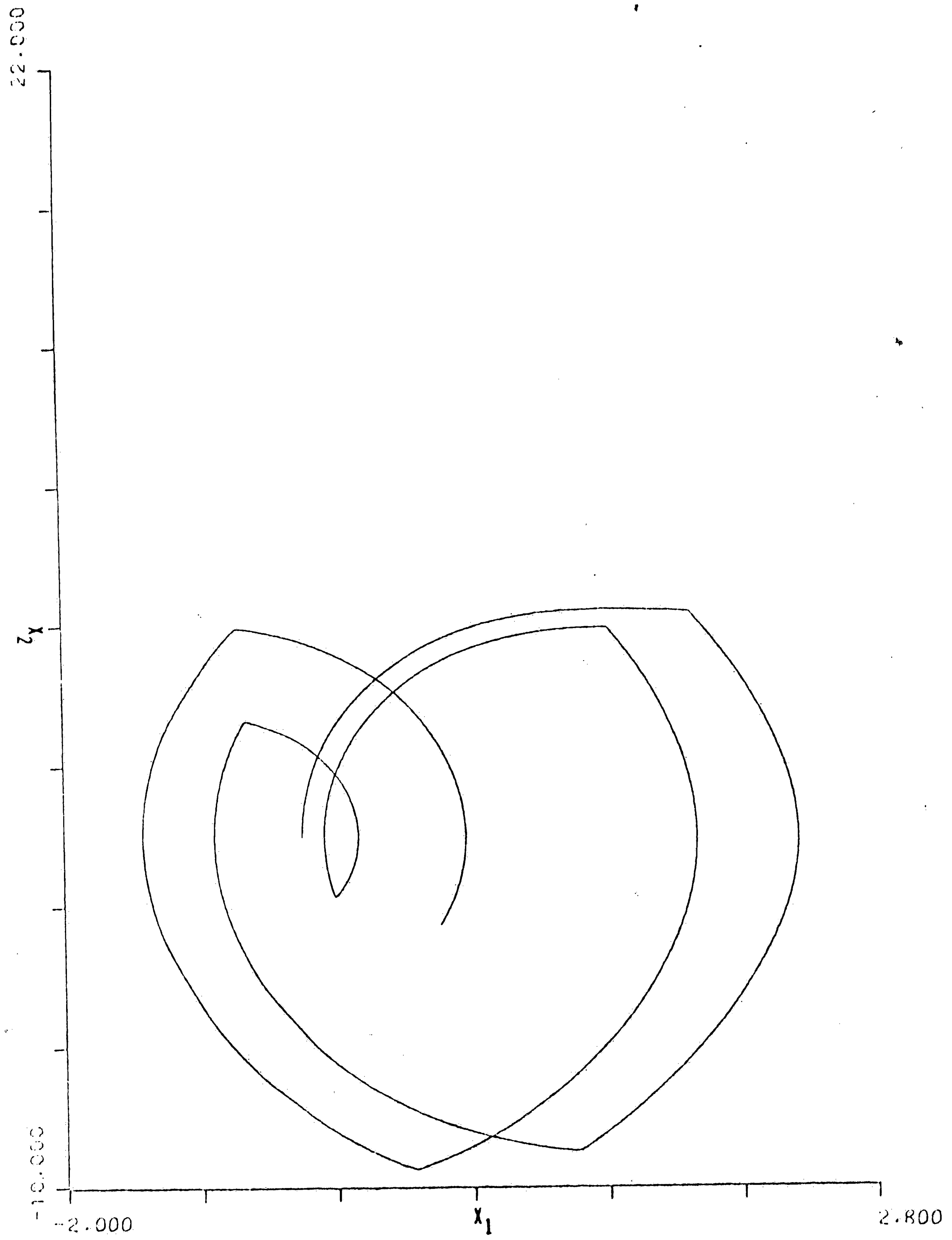


Figure 1.3.3a: Trajectory with square pulse input

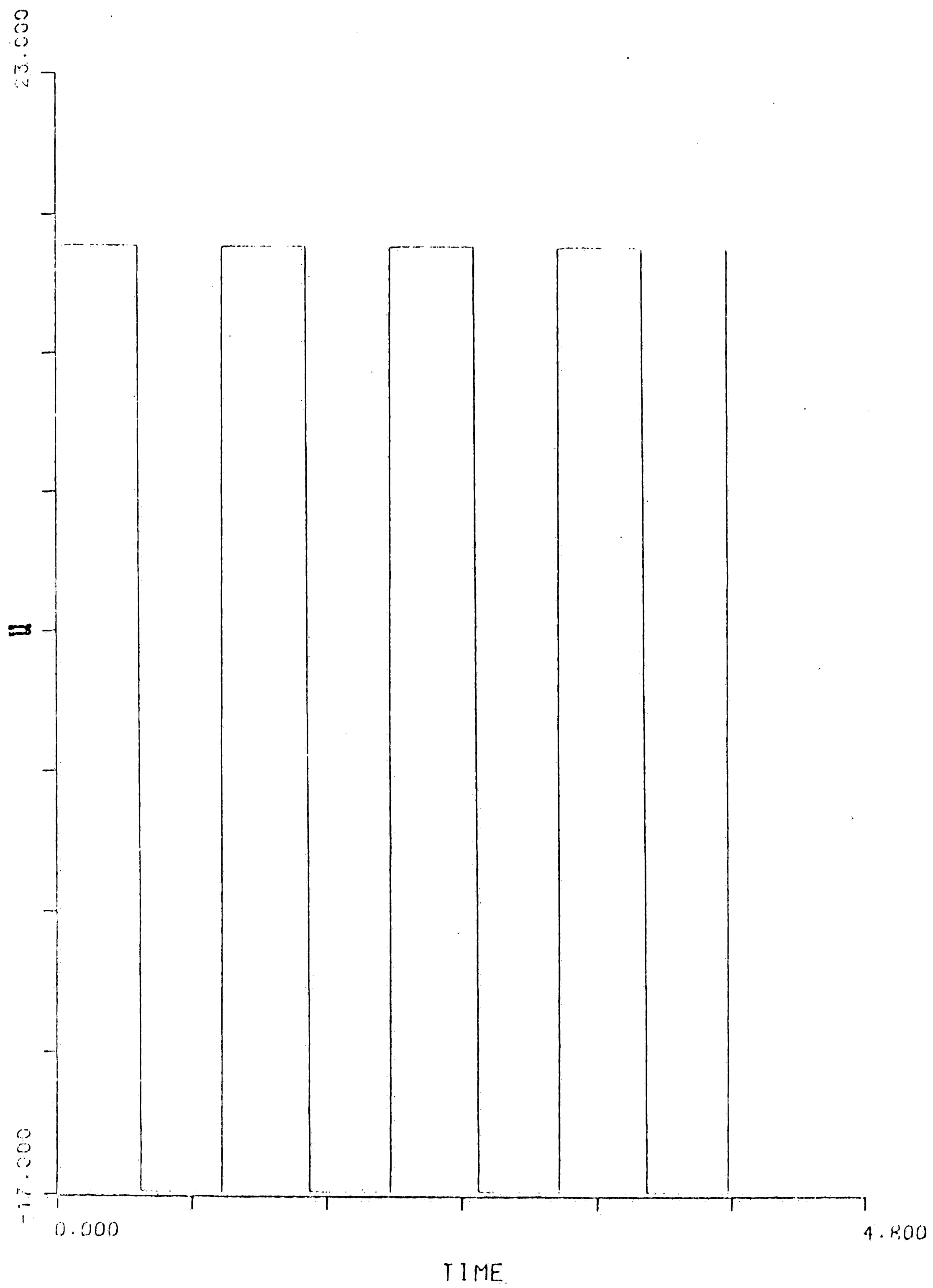


Figure 1.3.3b: Input $u(t)$
(plotted vs. time)

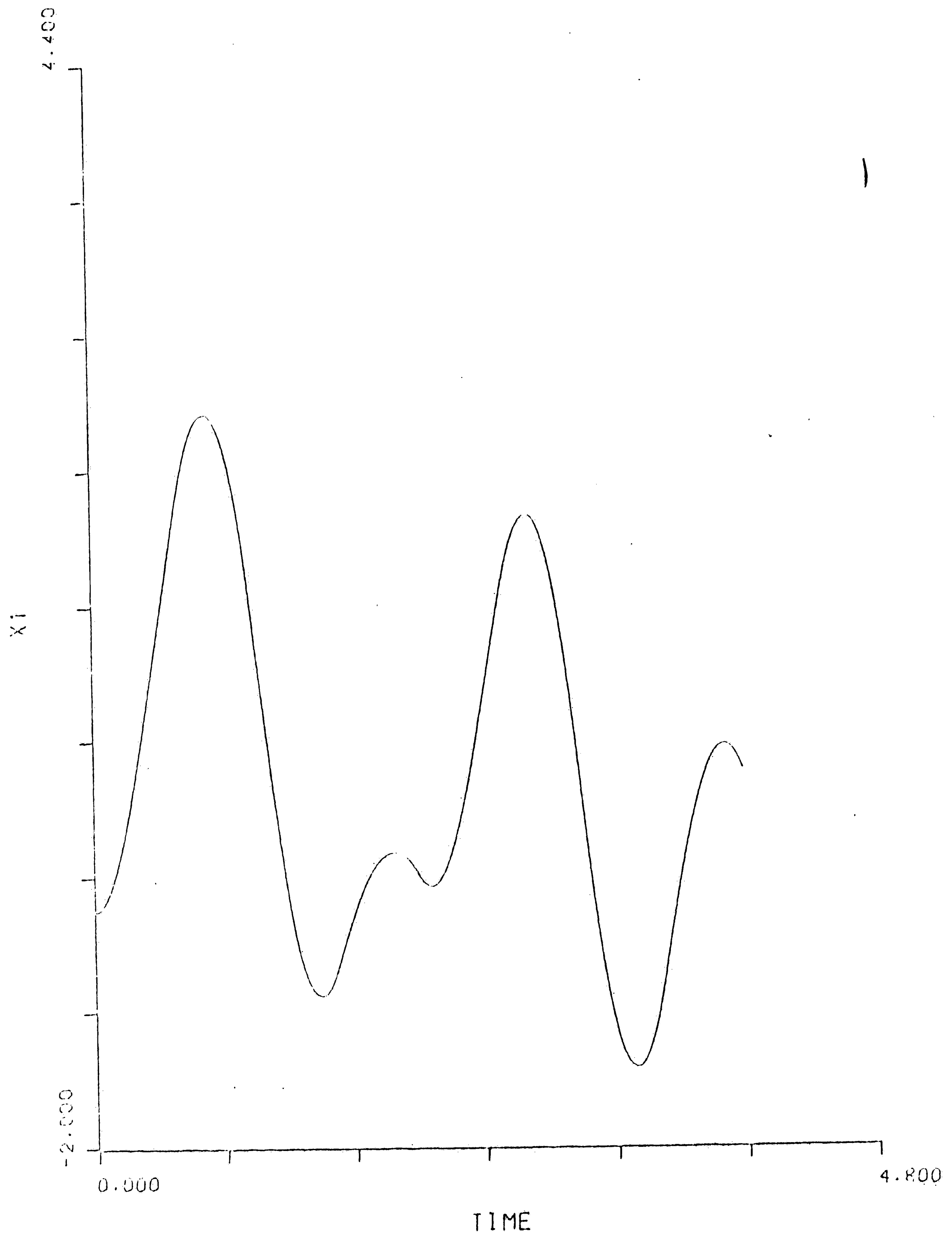


Figure 1.3.3c: x_1
as a function of time

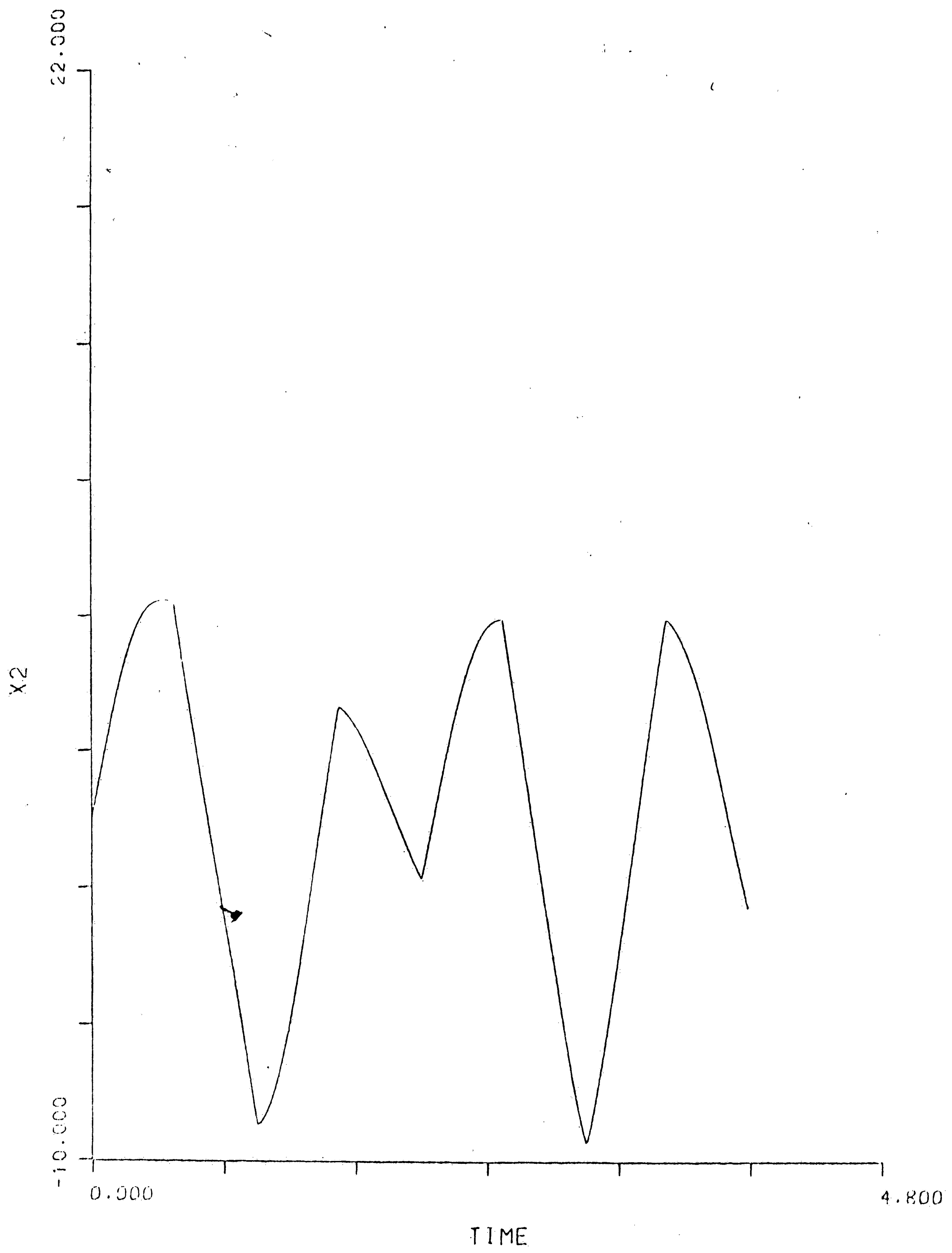


Figure 1.3.3d: $x_2(t)$

plotted vs. time

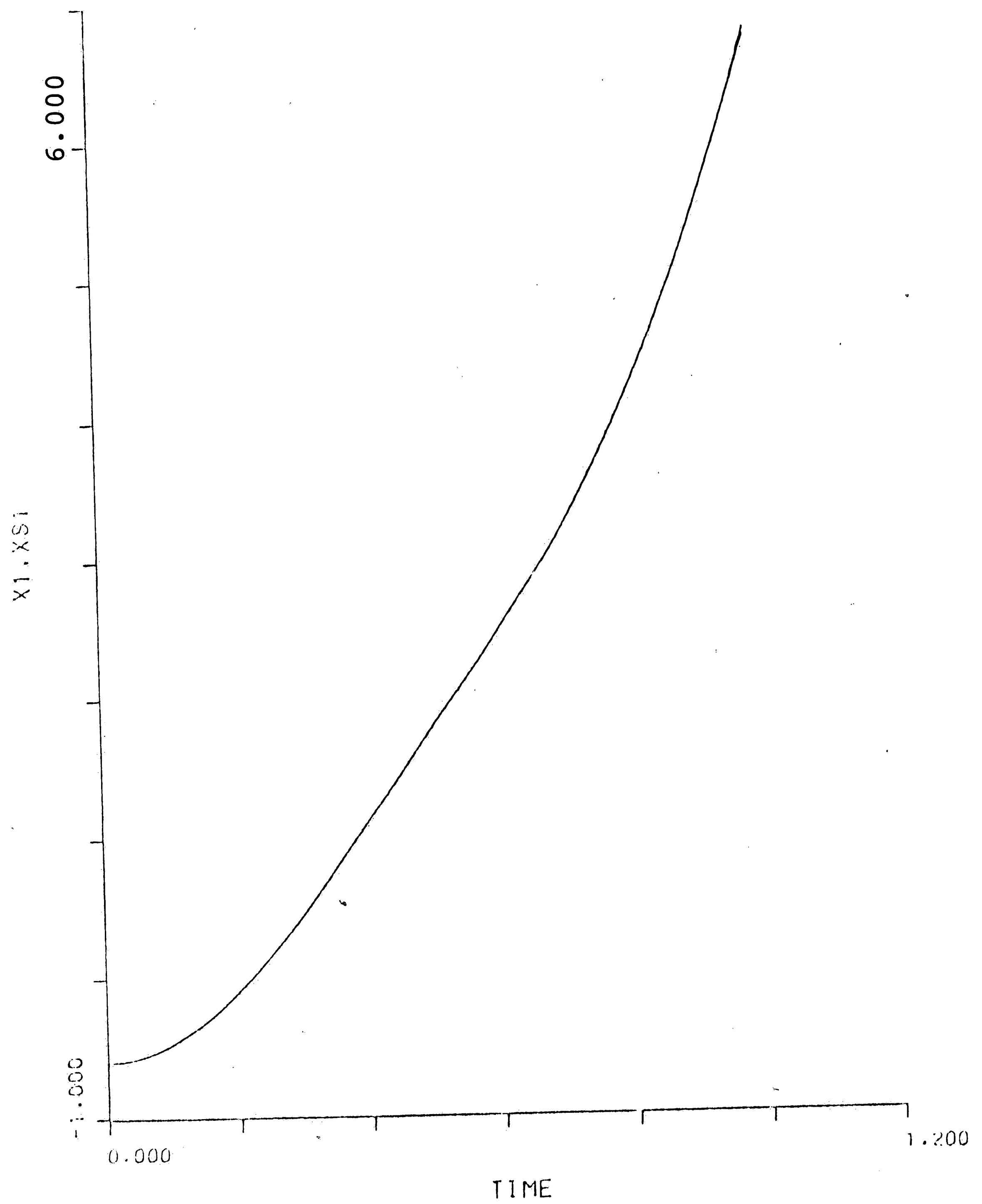


Figure 1.3.4a: 2-interval spline
($x_{s1}(t)$ approximation of $x_1(t)$)

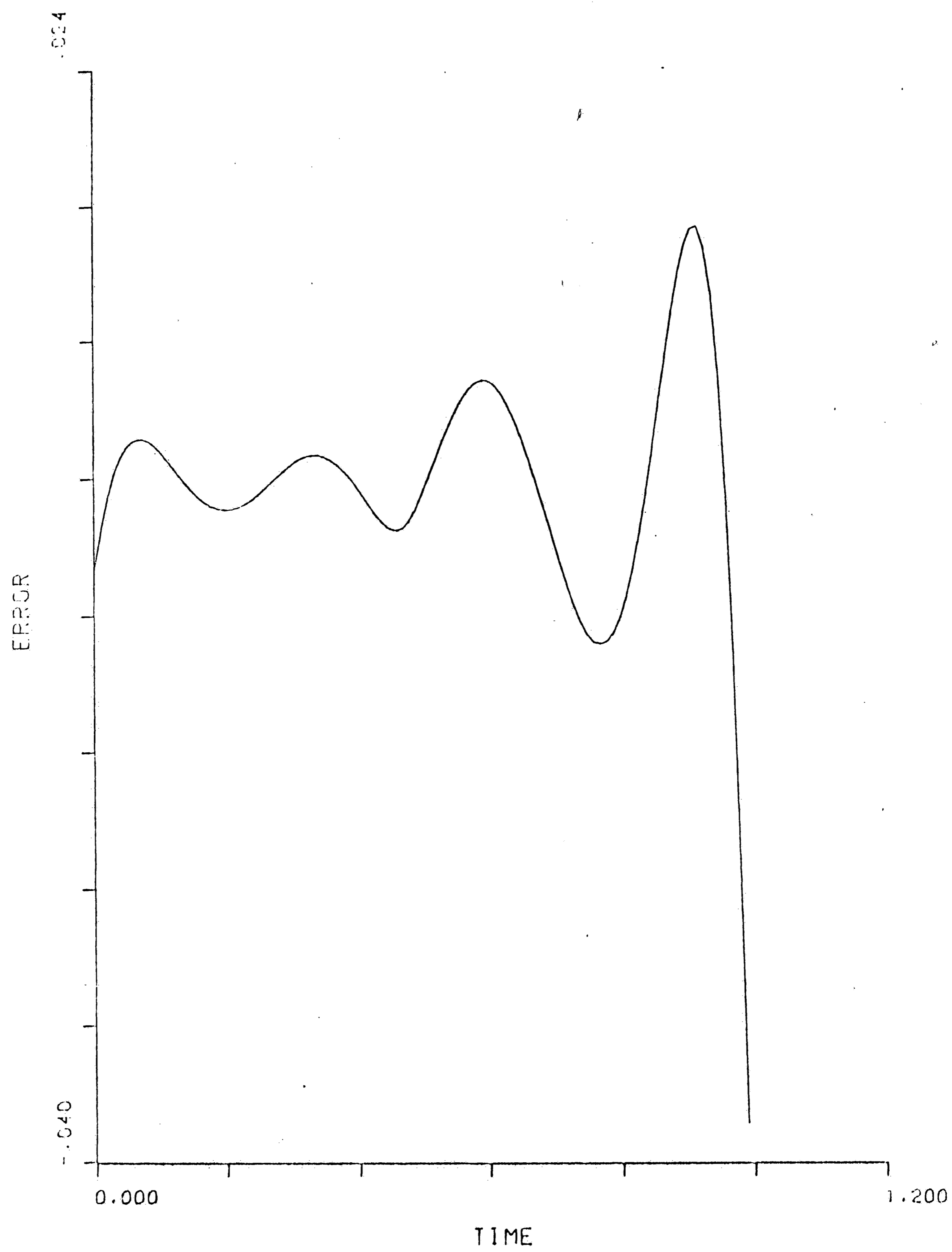


Figure 1.3.4b: Error of approximation

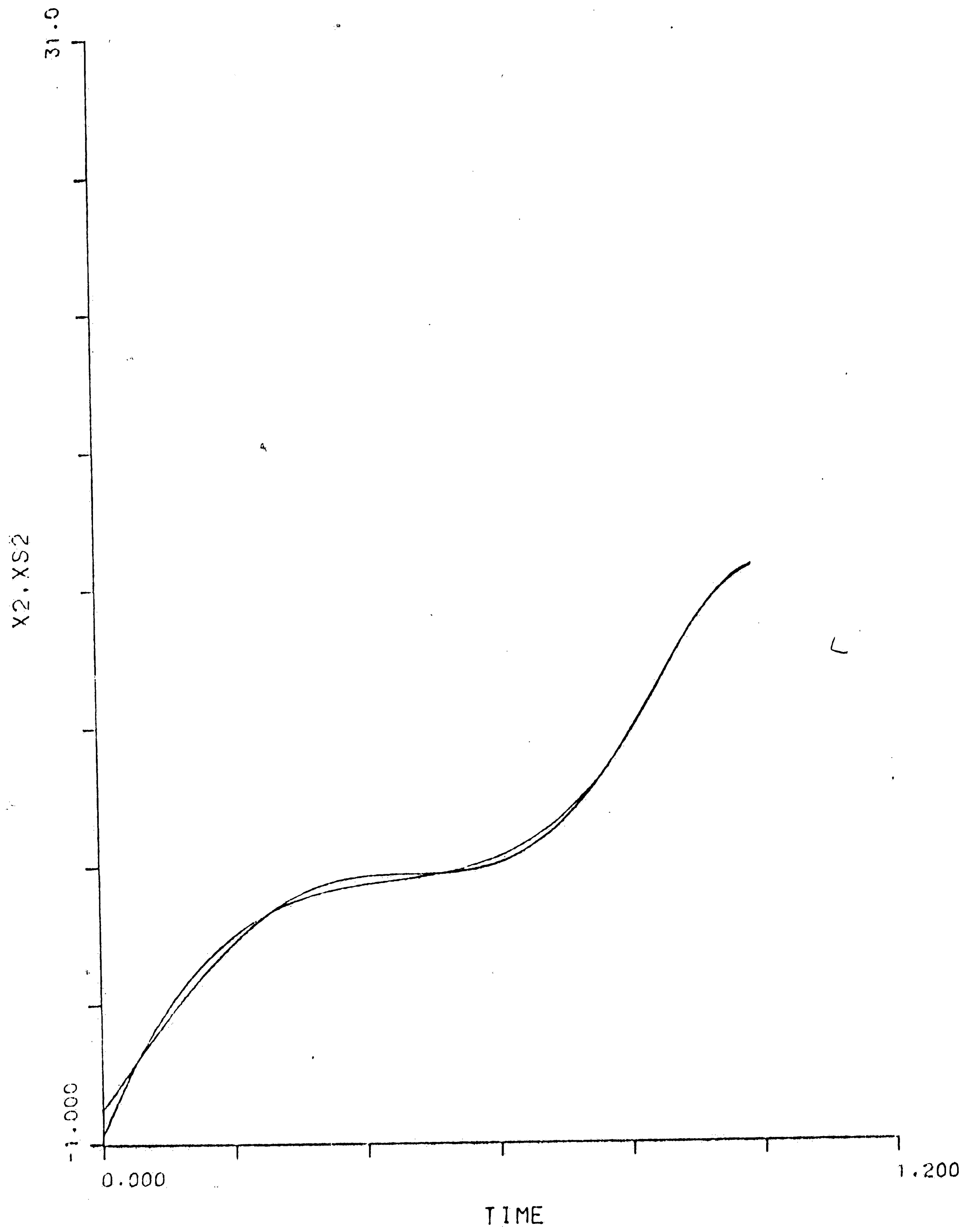


Figure 1.3.4c: 2-interval spline
($x_{s2}(t)$ approximation of $x_2(t)$)

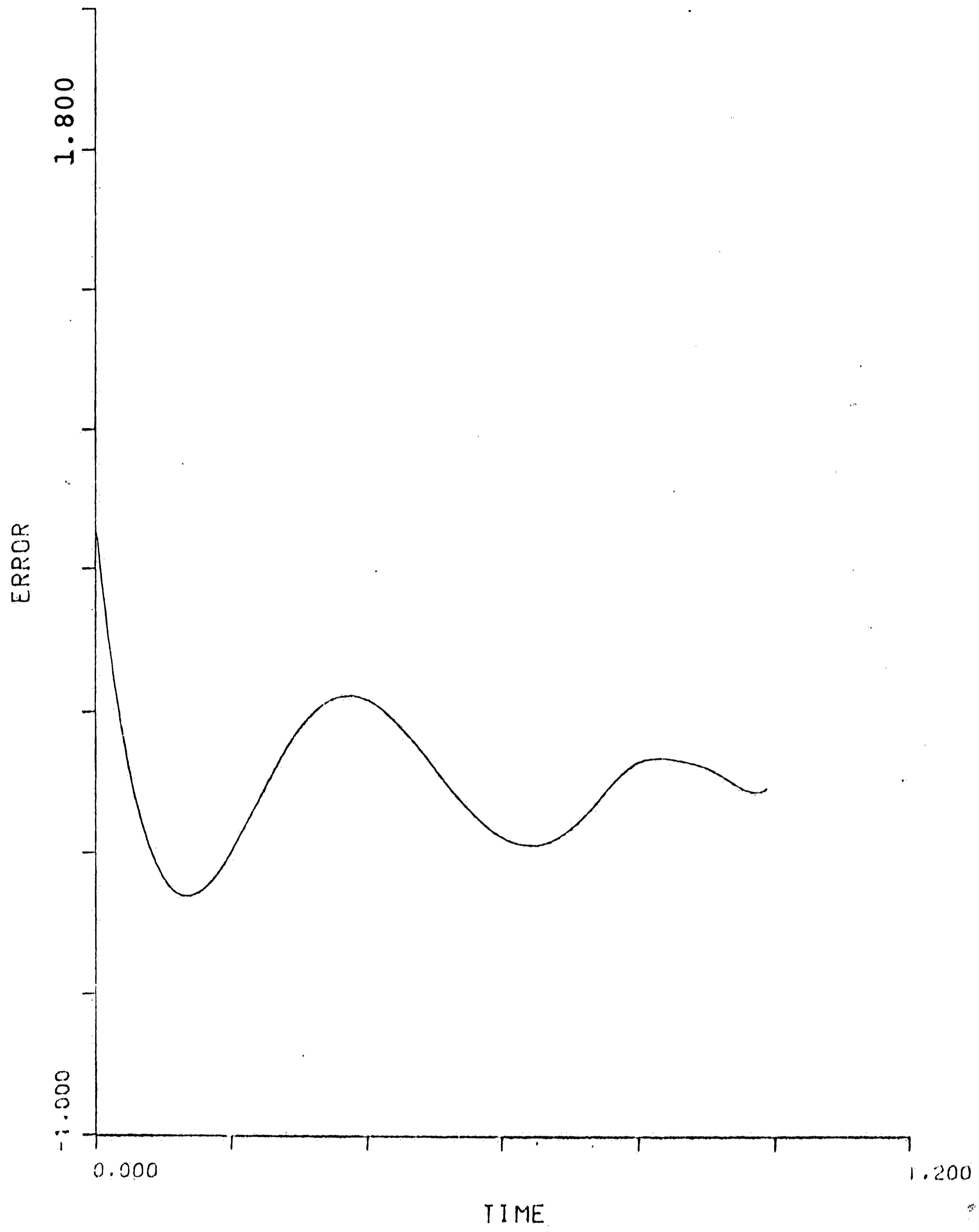


Figure 1.3.4d: Error of approximation

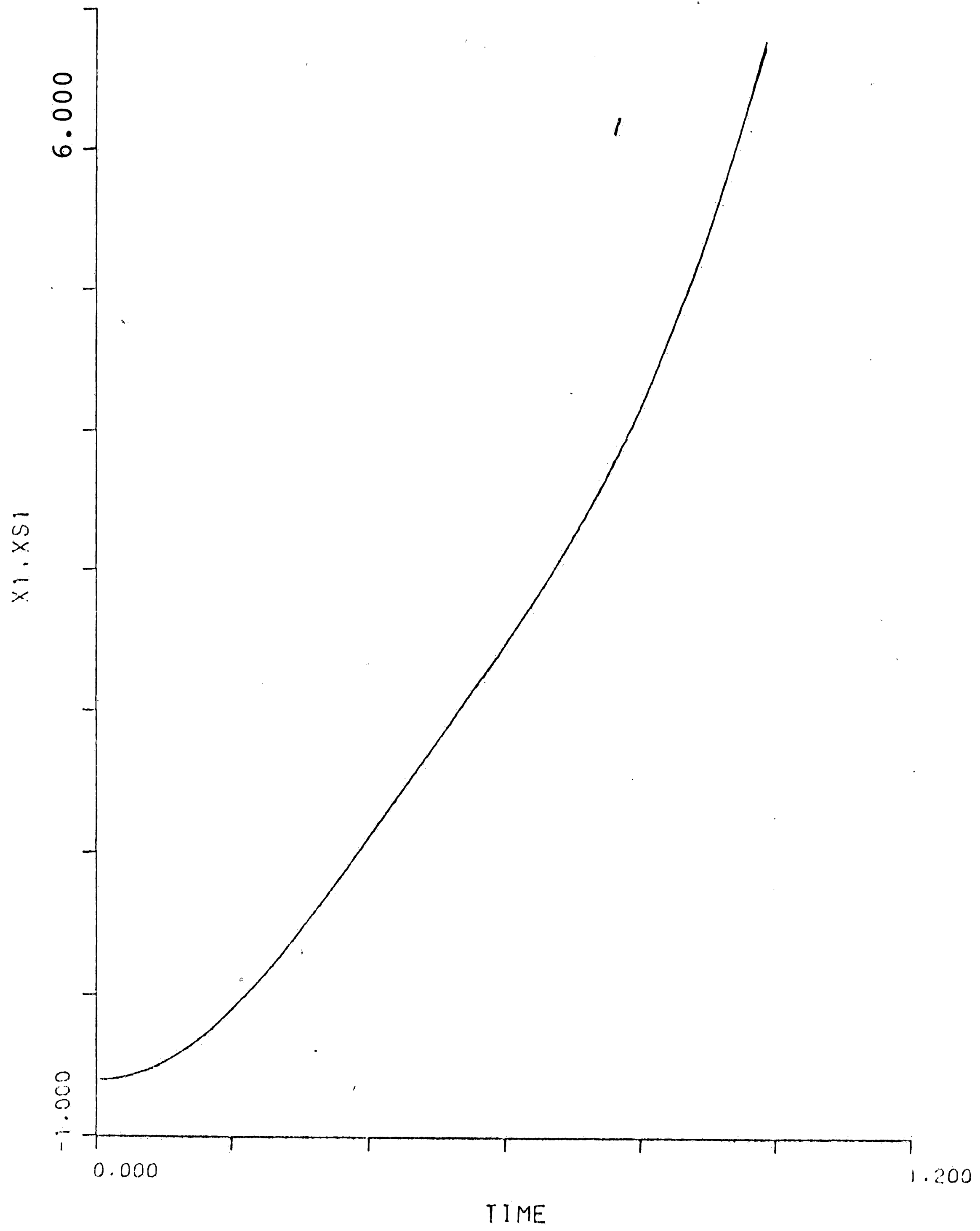


Figure 1.3.5a: 4-interval spline
($x_{s1}(t)$ approximation of $x_1(t)$)

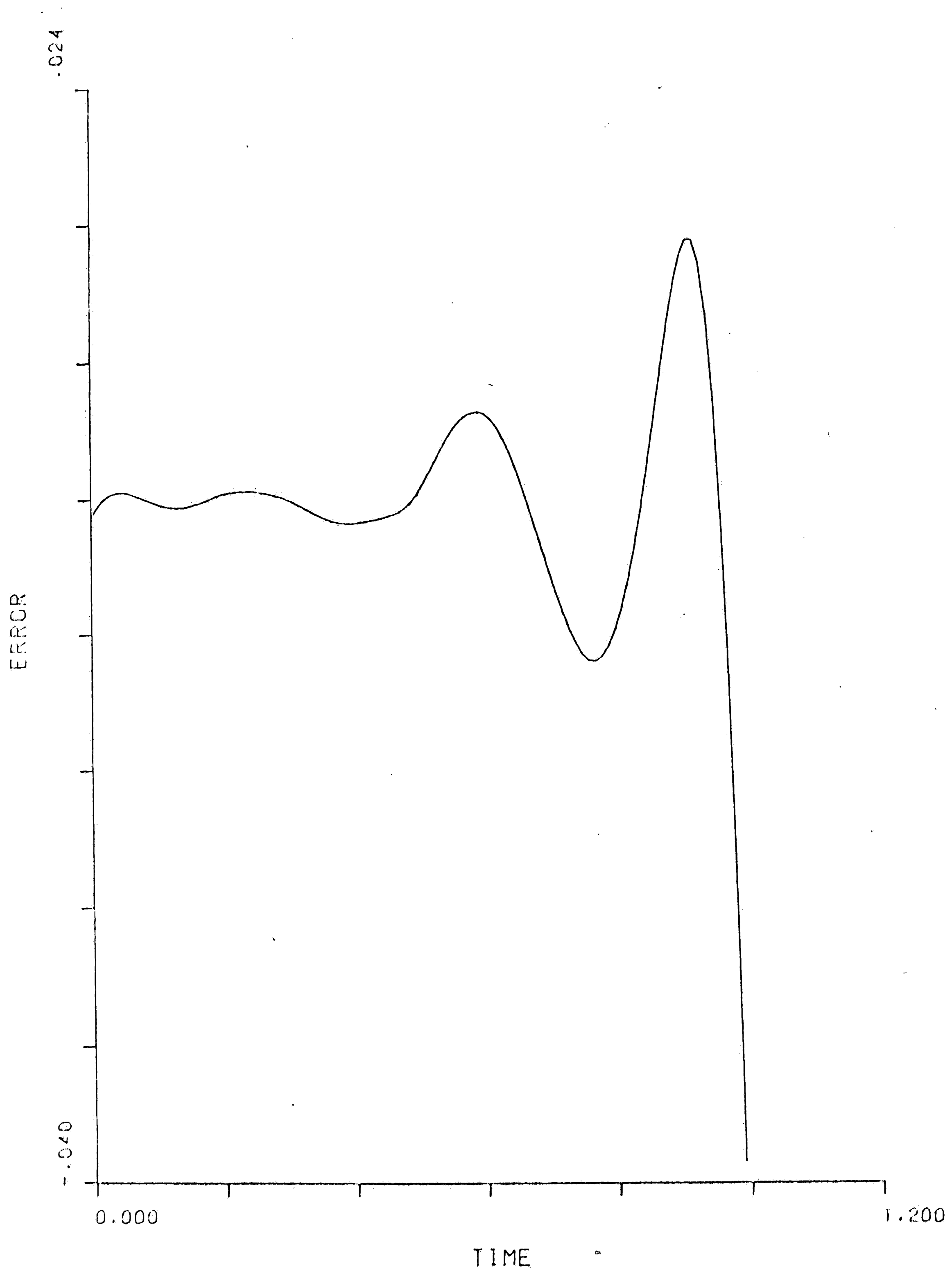


Figure 1.3.5b: Error of approximation

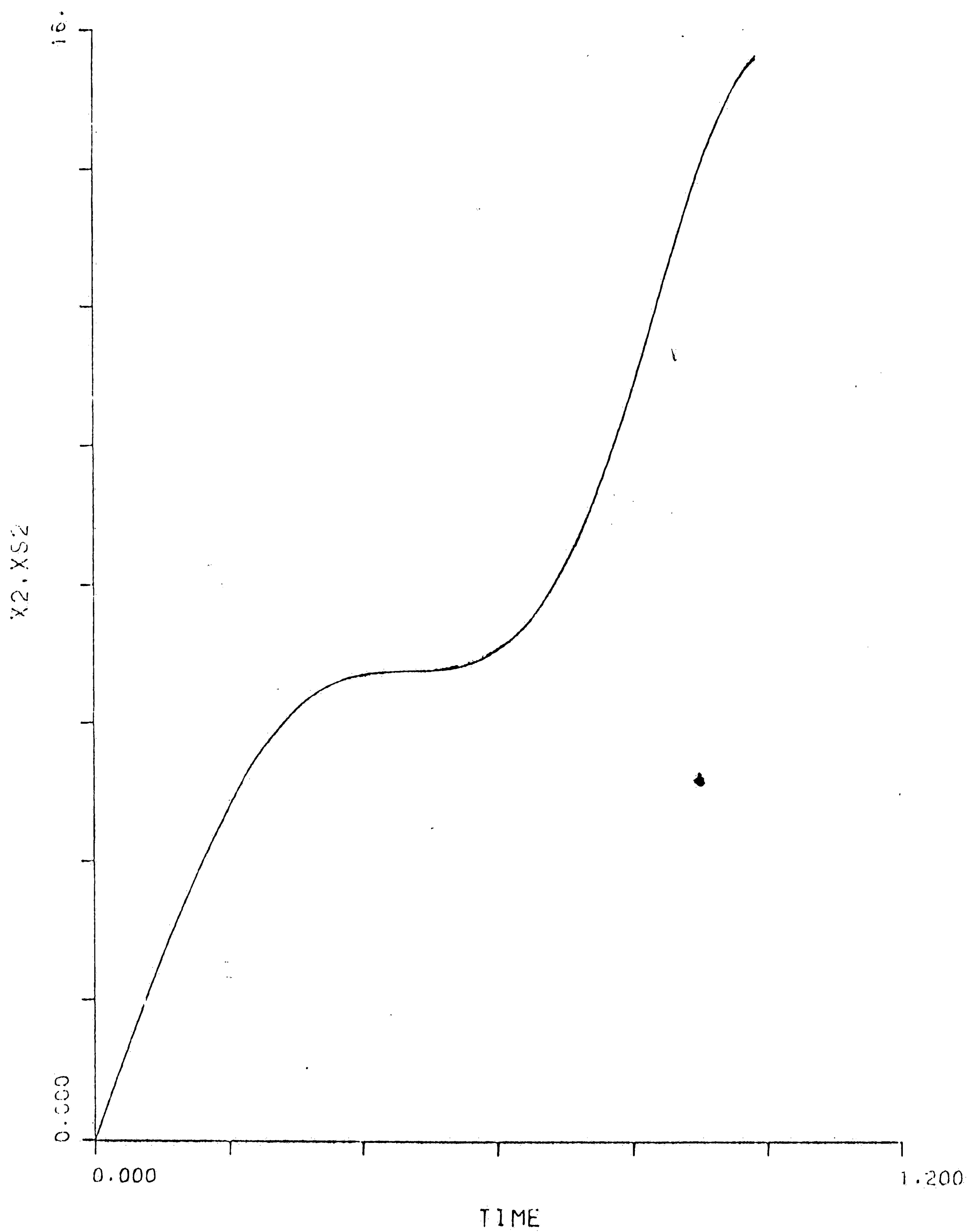


Figure 1.3.5c: 4-interval spline
($x_{s2}(t)$ approximation of $x_2(t)$)

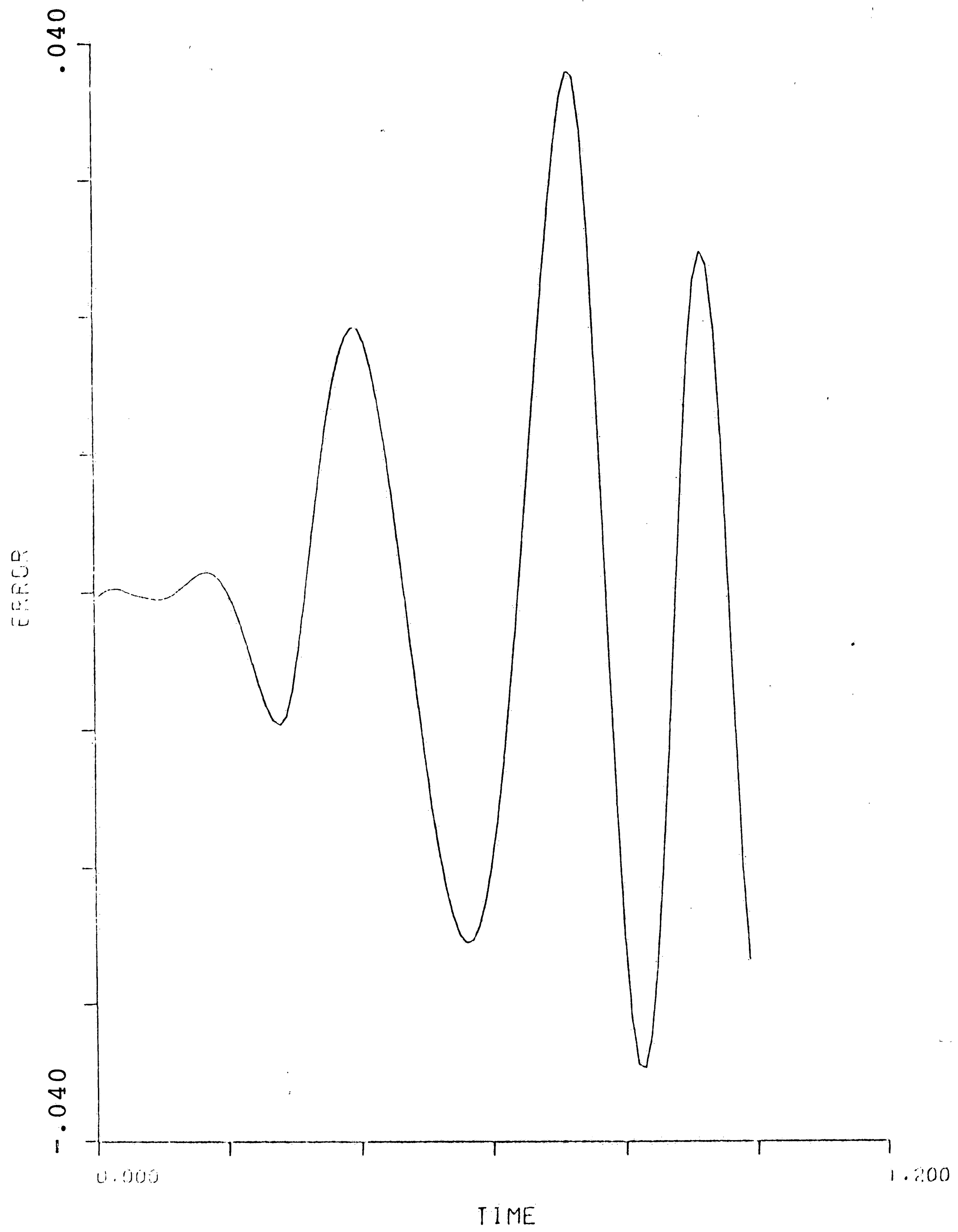


Figure 1.3.5d: Error of approximation

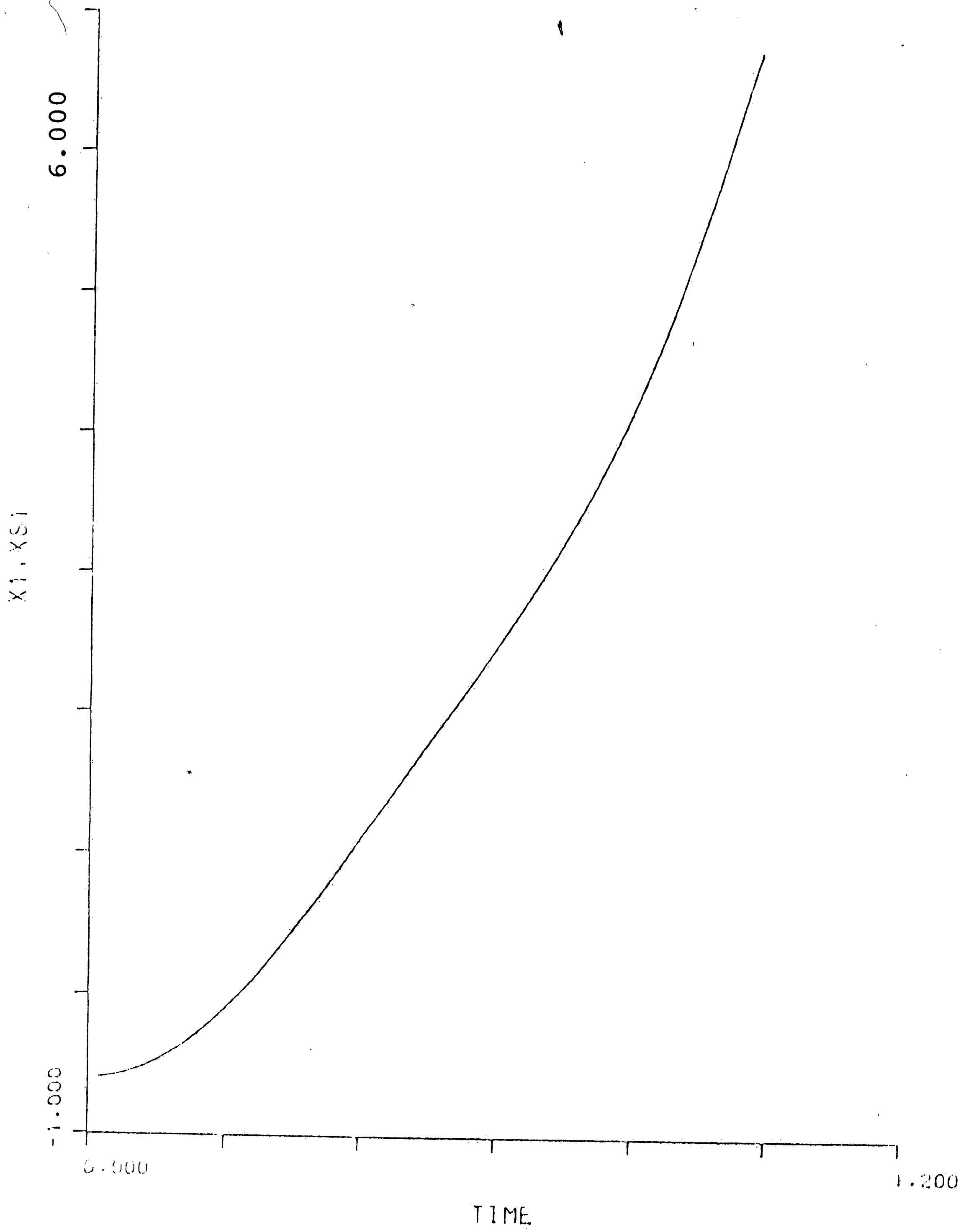


Figure 1.3.6a:~8-interval spline
($x_{s1}(t)$ approximation of $x_1(t)$)

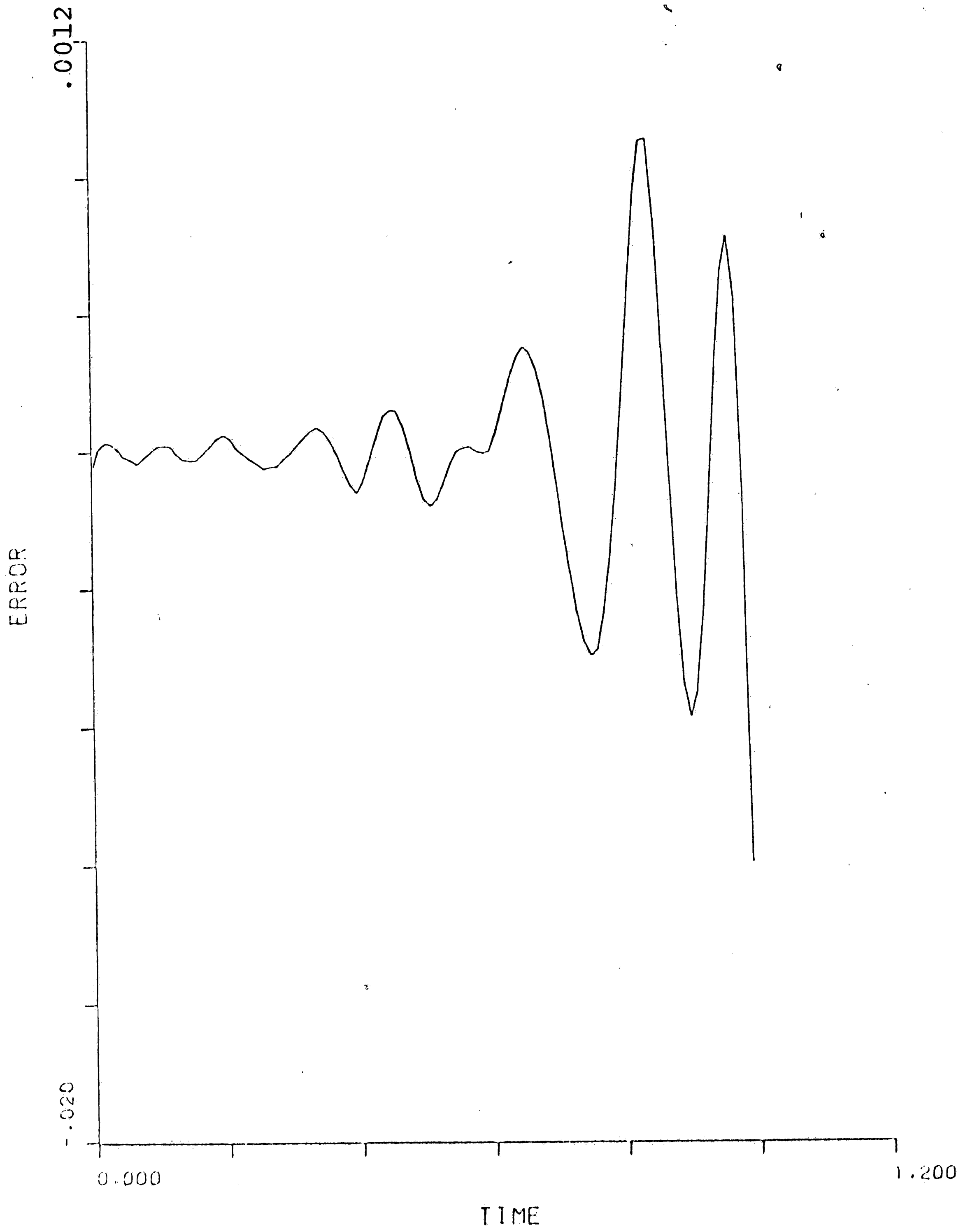


Figure 1.3.6b: Error of approximation

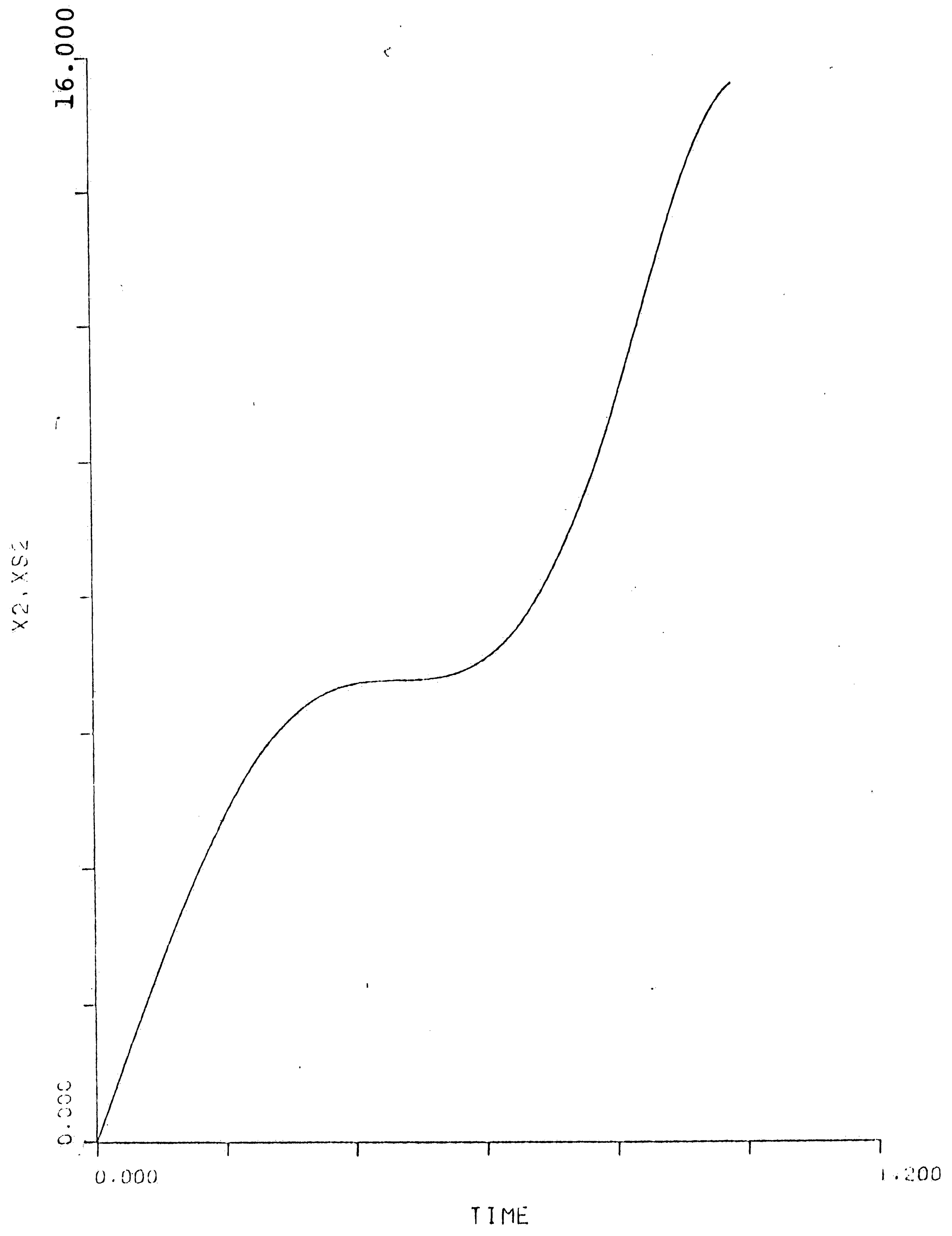


Figure 1.3.6c: 4-interval spline
 ($x_{s2}(t)$ approximation of $x_2(t)$)

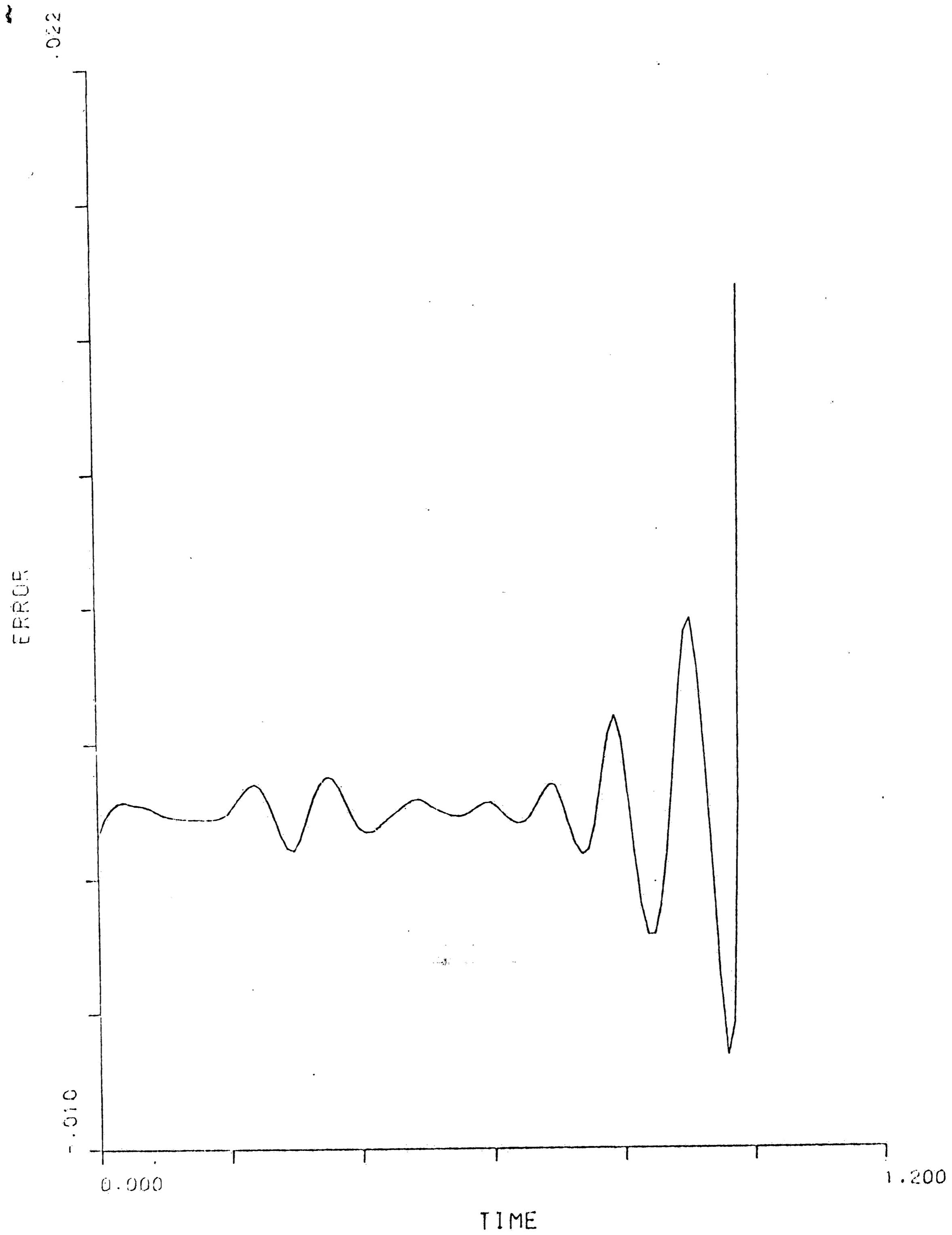


Figure 1.3.6d: Error of approximation

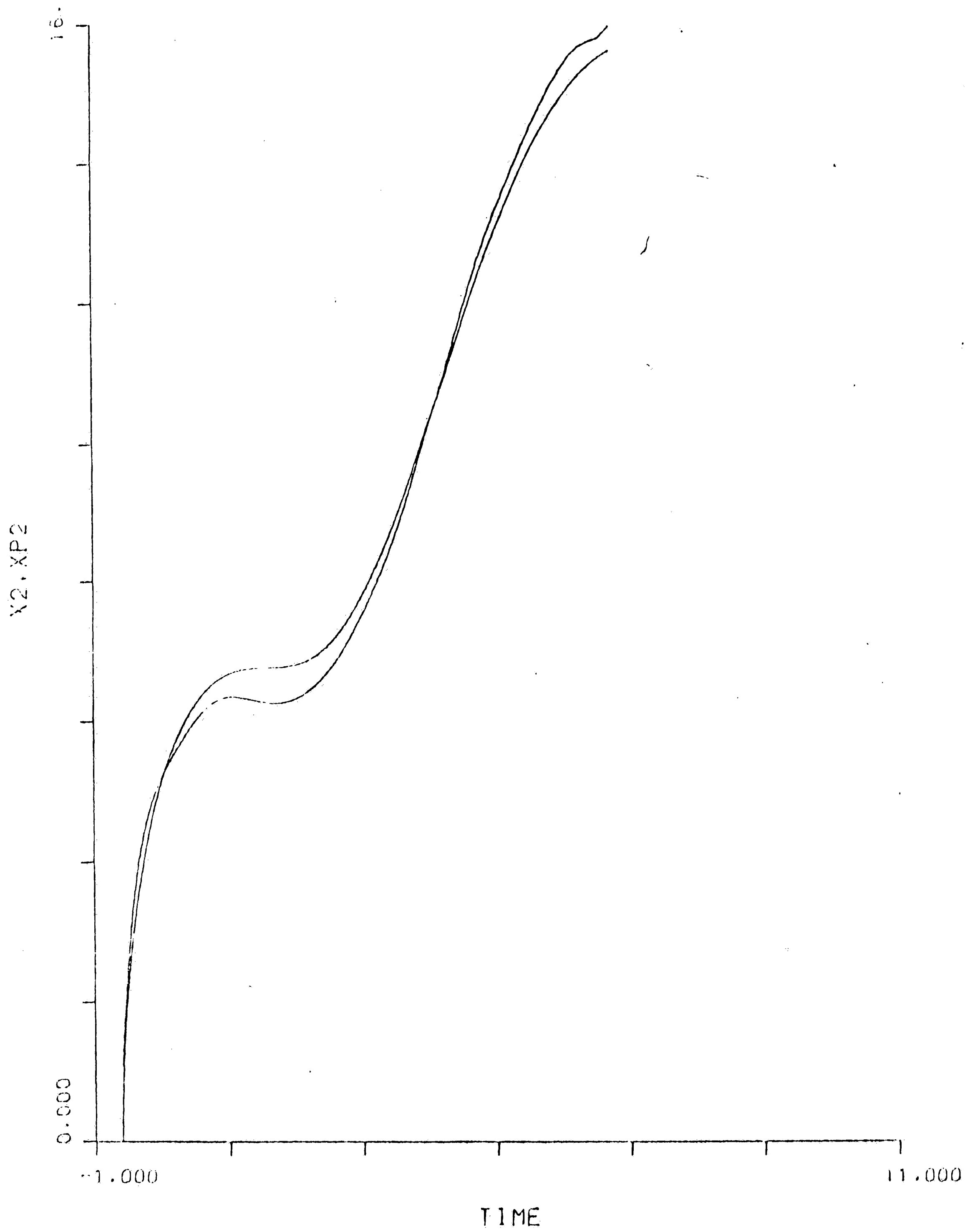


Figure 1.3.7a: 12-th order polynomial

($x_{p2}(t)$ approximation of $x_2(t)$)

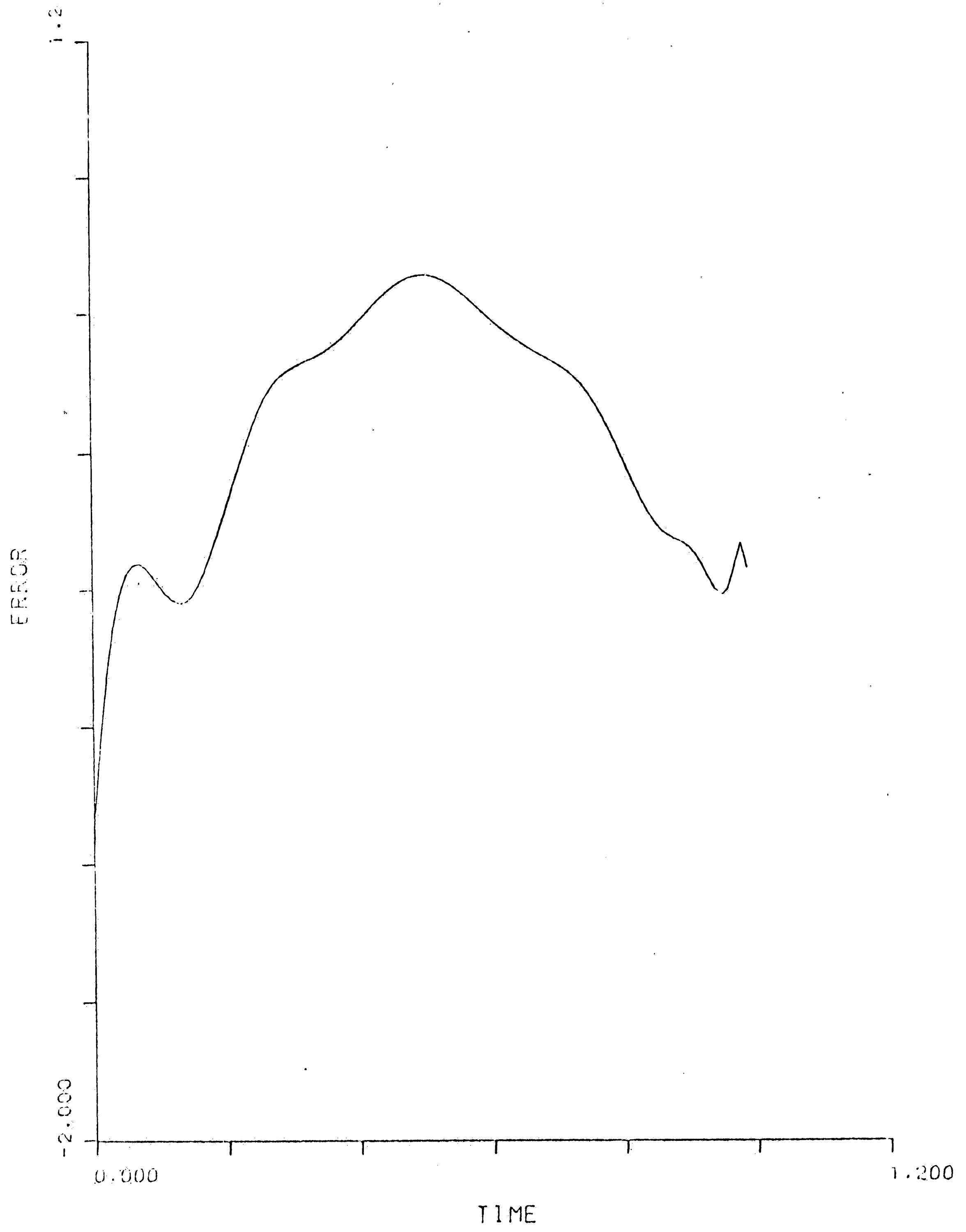


Figure 1.3.7b: Error of approximation

1.4 TRACKING: Once we have the approximations of the trajectories, we track them by the use of some form of corrective control. In the selection of the particular control law, we will be guided by several requirements. Foremost, we want good stability properties, that is, we do not want the tracking error to grow too much. If we cannot ensure asymptotic stability for the error, we might be satisfied with boundedness. We examine this question in Chapter 2 in greater detail. Of the many publications on the subject, we mention [13-20]. In Chapter 3 we present an analysis of time optimality deterioration with the decrease of maximum control input. As we point out, when we commit some of the available input for corrective control purposes, less is available for bang-bang control. Also, we want some form of robustness to parameter variations (see [13,16-18]). The robustness question is related to the stability of the system (see e.g. [20]). A special case of robustness is the sensitivity of time optimality to parameter variations [6-10].

Further desirable characteristics of the control law are low computational demand and the ability to accommodate a high sampling rate. These requirements are complementary. We must be able to perform the necessary control computations in the time interval between two measurements. To assure that we can either use a control scheme that can work with few measurements (ideally at the minimum sampling rate determined by the dynamics of the system), or a control scheme that requires little

computation, or even better, a combination of the two. At the present time, the dynamic control of industrial manipulators involves dedicated mainframes; we are aiming for a microprocessor implementation. To realize this goal, we must maintain a relatively small number of computations between taking two successive measurements. This means low sampling rate and/or few computations are required.

CHAPTER 2: STABILITY

2.1 NOTATION AND ASSUMPTIONS : Given the system 1.1.1-.2, assume that we have the nominal trajectories, characterized by the state variables as functions of time (in closed or tabulated form): $x_{n1}(t)$, $x_{n2}(t)$. Also assume that we have an approximation of $x_{n1}(t)$, $x_{n2}(t)$ (again in closed or tabulated form): $x_{s1}(t)$, $x_{s2}(t)$. In the following we will often drop the time argument, where it is clear from the context. Then we have the following notation:

- 2.1.1 x_1, x_2 actual system state variables
- 2.1.2 x_{n1}, x_{n2} nominal system state variables
- 2.1.3 x_{s1}, x_{s2} approximation to x_{n1}, x_{n2}

We also define the differences

- 2.1.4 $\delta_{s1} = x_{s1} - x_{n1}$ (approximation error)
- 2.1.5 $\delta_{n1} = x_{s1} - x_{n1}$ (correction signal)

$$2.1.6 \quad \delta_1 = x_1 - x_{n1} \quad (\text{tracking error})$$

$$2.1.7 \quad \delta_{s2} = x_{s2} - x_{n2} \quad (\text{approximation error})$$

$$2.1.8 \quad \delta_{n2} = x_{s2} - x_2 \quad (\text{correction signal})$$

$$2.1.9 \quad \delta_2 = x_2 - x_{n2} \quad (\text{tracking error})$$

Note that

$$2.1.10 \quad \delta_1 = \delta_{s1} - \delta_{n1}$$

$$2.1.11 \quad \delta_2 = \delta_{s2} - \delta_{n2}$$

We hope (on the basis of equation (1.3.1) and Figs. 1.3.4-1.3.6, that $\delta_{s1,s2}$ are bounded and in fact quite small. We write

$$2.1.12 \quad |\delta_{s1}| < \epsilon_{s1}$$

$$2.1.13 \quad |\delta_{s2}| < \epsilon_{s2}$$

Now we will write the equations of the simulated system (in nominal operation) and the equations of the actual system when an additional corrective control is being applied.

$$2.1.14 \quad \dot{x}_{n1} = x_{n2}$$

$$2.1.15 \quad \dot{x}_{n2} = f(x_{n1}, x_{n2}) + u$$

And

$$2.1.16 \quad \dot{x}_1 = x_2$$

$$2.1.17 \quad \dot{x}_2 = f^*(x_1, x_2) + (u + u_c)$$

(where u_c is the corrective control)

Now, f and f^* are different functions. The reason we choose two different functions is to represent the possibility of inexact modelling. The difference between f and f^* may be small or great. For example, in the context of the pendulum (eqs. (1.1.3-.4)), we could have

$$2.1.18 \quad f(x_1, x_2) = A \sin x_1 + B x_2 + u$$

$$2.1.19 \quad f^*(x_1, x_2) = A^* \sin x_1 + B x_2 + (u + u_c)$$

And the discrepancy would be just numerical, e.g. $A=20.$, $A^*=21.$ Of course we could have more serious discrepancies, e.g. we could have an x_1^2 term in the second equation, or some totally different function. Practically, we hope that our modelling of the system will be quite accurate. Anyway, we will make the assumption that

$$2.1.20 \quad |\delta f| < \epsilon_0 \quad \text{for all } x_{1,2}$$

where

$$2.1.21 \quad \delta f = f^* - f$$

And

$$2.1.22 \quad |f| < A_1 \quad \text{for all } x_{1,2}$$

Note that this also implies

$$2.1.123 \quad |f^*| < A_2 = A_1 + \epsilon_0$$

Now, we want to choose u_c in eq.(2.1.17) so as to ensure some form of stability. In essence, we want to track the nominal trajectory of the system. To quantify the problem we need to develop the tracking error $(\delta_{1,2})$

equations. We do that in the following fashion. Write the equations of the actual system

$$2.1.24 \quad \dot{x}_1 = x_2$$

$$2.1.25 \quad \dot{x}_2 = f^*(x_1, x_2) + u + u_c = f(x_1, x_2) + \delta f(x_1, x_2) + u + u_c$$

Let us repeat that δf can have any functional form, very different from that of f . This means that we can even deal with the case where our modelling function f^* is quite "wrong", that is, our model is quite different from the actual system, which is characterized by the function f . All we require, is that the difference δf between the two functions be bounded. However, what is usually the problem, and what we are particularly interested in, is the case where we know the functional form of f , but we are not quite certain of some parameter's numerical value. To invoke the example of the pendulum, the constant A depends on the mass of the bob. If we make a mistake in the measurement, or estimation, of the mass, the forms of f and f^* will still be identical. The first one will be $A \sin x_1$ and the second will be $A^* \sin x_1$.

However, the two numerical constants A and A^* might have different values. Now, in eq.(2.1.25) we have separated the "real" function f^* in two parts: the modelling function f and the difference function δf . We would like to develop an error equation, in which f would disappear entirely. A way to achieve this, is to subtract eq.(2.1.14-.15) from eqs.(2.1.24-.25) respectively. In that case, f would appear only in the difference $f-f$, which will hopefully vanish. Unfortunately, the difference is $f(x_1, x_2) - f(x_{n1}, x_{n2})$. In other words, the arguments are different and the difference does not equal zero.

However, we can always write a Taylor series for the function f , around the trajectory x_{n1}, x_{n2} . To do this, note that from eqs.(2.1.6,.9) we have

$x_1 = x_{n1} + \delta_1$, $x_2 = x_{n2} + \delta_2$, and substitute these values in eqs.(2.1.24-.25)

2.1.26 $\dot{x}_1 = x_{n2} + \delta_2$

2.1.27 $\dot{x}_2 = f(x_{n1} + \delta_1, x_{n2} + \delta_2) + \delta f(x_1, x_2) + u + u_c$

Now, write the Taylor series for f , considering f a function of $x_{n1, n2}$.

2.1.28 $\dot{x}_1 = x_{n2} + \delta_2$

2.1.29 $\dot{x}_2 = f(x_{n1}, x_{n2}) + \frac{\partial f}{\partial x_{n1}}(x_{n1}, x_{n2})(x_1 - x_{n1}) + \frac{\partial f}{\partial x_{n2}}(x_{n1}, x_{n2})(x_2 - x_{n2}) + \dots + \delta f(x_1, x_2) + u + u_c$

Where the dots stand for Taylor terms of higher order. Subtracting eq.(2.1.28) from eq. (2.1.24) and eq.(2.1.29) from eq.(2.1.25) and using the tracking error equations (2.1.6-.9) we get

$$2.1.30 \quad \dot{\delta}_1 = \delta_2$$

$$2.1.31 \quad \dot{\delta}_2 = \frac{\partial f}{\partial x_{n1}}(x_{n1}, x_{n2})\delta_1 + \frac{\partial f}{\partial x_{n2}}(x_{n1}, x_{n2})\delta_2 + \dots + \delta f(x_1, x_2) + u_c$$

The purpose of all this maneuver is to develop equations for the error $d_{1,2}$ in which only known, or measurable quantities will occur. We had no way of knowing f^* , so we had to express it in terms of the known model function f (in this case its derivatives) and the known-to-be-bounded function δf . Note, that the nominal trajectory, $x_{n1, n2}$, is known functions of time and so the partial derivatives in eq.(2.1.31) are, as well, known functions of time.

So far what we have been doing is ordinary linearization around a nominal operating trajectory. We will not use this approach to the end but will now follow an alternative approach in the following section, where we will consider tracking methods and their stability properties.

Before we end this section, let us note that when treating discrete time systems we can proceed in exactly the same way and obtain exactly analogous equations for the tracking error.

2.2 STABILITY PROPERTIES EXEMPLIFIED BY THE PROPORTIONAL CONTROL SCHEME: In

this section we will assume the control u_c is proportional to the tracking error and, using this as an example, we will discuss the stability properties of the system. In this section we will discuss the subject analytically just for the case of continuous time. We will propose and use a numerical method in the next section, which, even though based on the same ideas is more tractable and efficient.

Ideally we would wish that the error system described by eqs.(2.1.30-.31) be globally asymptotically stable, but this turns out that to be too strong a requirement. However, boundedness of the error would be almost as good, especially if we can control the bounds by proper adjustment of the controller's parameters.

Assume then that

$$2.2.1 \quad u_c = k_1 \delta_{n1} + k_2 \delta_{n2}$$

Remember that $\delta_{n1} = x_{s1} - x_1$, $\delta_1 = x_{n1} - x_1$. It is clear why we use the nominal error $\delta_{n1,n2}$ instead of the actual $\delta_{1,2}$: we really have available for comparison only $x_{s1,s2}$ and not $x_{n1,n2}$. That is why we want a reasonably good approximation to make sure we do not use a totally misleading reference trajectory. Indeed we will perform an analysis that shows the role of the approximation error in the boundedness of the tracking error.

Let us first note that $x_{n1,n2}$ are given functions of time. Then we can consider a function $f(x_{n1}, x_{n2})$ of the nominal state variables as a

function of time, $g(t)=f(x_{n1}(t),x_{n2}(t))$. Using this fact, equation (2.2.1) and eqs.(2.1.30-.31) we can write

$$2.2.2 \quad \delta_1 = \delta_2$$

$$2.2.3 \quad \delta_2 = (g_1(t) - k_1)\delta_1 + (g_2(t) - k_2)\delta_2 + \delta f(x_1, x_2) + k_1 \delta_{s1} + k_2 \delta_{s2} + (\text{Taylor terms of order 2 and higher})$$

Here we have considered the first partial derivatives of f as functions of t . If we assumed the higher order terms to be negligible, we would end up with a linear time varying system. We could investigate stability of this system, trying for example to find a Lyapunov function, or exploit the boundedness of the time dependent terms (e.g. see Takahashi [22], and also [23]).

Alternatively, we can follow another path. This is our proposed new method of analysis. Instead of assuming that Taylor terms are negligible, we will instead show they are bounded. In fact we will show boundedness for Taylor terms of order 1 and higher.

$$2.2.4 \quad |f(x_1, x_2)| = |f(x_{n1}, x_{n2}) + \text{Taylor}| > |\text{Taylor}| - |f(x_{n1}, x_{n2})| +$$

$$2.2.5 \quad |f(x_1, x_2)| + |f(x_{n1}, x_{n2})| > |\text{Taylor}|$$

But now using eq.(2.1.22)

$$2.2.6 \quad A_1 + A_1 = C > |\text{Taylor}|$$

So we see the Taylor terms of order 1 and higher (that includes first order partial derivatives, i.e. $g_{1,2}(t)$) are bounded. Let us use matrix notation and write the system of eqs.(2.2.2-.3) as follows:

$$2.2.7 \quad \underline{\dot{\delta}} = \underline{A}\underline{\delta} + \underline{b}$$

where

$$2.2.8 \quad \underline{A} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$$

$$2.2.9 \quad \underline{b} = \begin{bmatrix} 0 \\ \text{Taylor} + \delta f + k_1 \delta_{s1} + k_2 \delta_{s2} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Let's take the Euclidean norm ($||\underline{b}|| = (b_1^2 + b_2^2 + \dots + b_n^2)^{1/2}$). In the case of \underline{b} as given by (2.2.9) we have $||\underline{b}|| = |b_2|$, and we can write

$$2.2.10 \quad ||\underline{b}|| < (C + \epsilon_0 + k_1 \epsilon_{s1} + k_2 \epsilon_s)_2$$

Let us now consider another norm, or better let us investigate when the following form is a norm.

$$2.2.11 \quad V = \underline{\delta}^T \underline{B} \underline{\delta}$$

where \underline{B} is a matrix. Here V is a quadratic form and it is a norm if \underline{B} is positive definite (see, e.g. Wiberg, [21]) . We can always symmetrize \underline{B} if

it is not symmetric, so the positive definiteness condition is equivalent to positivity of the eigenvalues. Furthermore, (see again Wiberg, [21]), we have that

$$2.2.12 \quad \lambda_{\max} \|\underline{\delta}\|^2 > V > \lambda_{\min} \|\underline{\delta}\|^2$$

where $\lambda_{\min}, \lambda_{\max}$ are the minimum and maximum eigenvalues of \underline{B} , respectively.

This means that if we can somehow establish that V is bounded then we are assured that $\|\underline{\delta}\|$ is also bounded. Let us now give the following result, which is easily verified with a little algebra.

$$2.2.13 \quad \begin{bmatrix} p & q \\ q & r \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} + \begin{bmatrix} 0 & -k_1 \\ 1 & -k_2 \end{bmatrix} \begin{bmatrix} p & q \\ q & r \end{bmatrix} = -2 \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}$$

when this equation is true and $k_1 \gg 1$, then

$$2.2.14 \quad p = k(k_1/k_2 - k_2/k_1)$$

$$2.2.15 \quad q = k/k_1$$

$$2.2.16 \quad r = k/k_2$$

As can be verified by solving the equation. Let us write (2.2.13) in matrix notation

$$2.2.17 \quad \underline{B}\underline{A} + \underline{A}^T\underline{B} = -\underline{W}$$

Where it is clear what matrices $\underline{A}, \underline{B}$ and \underline{W} stand for. In fact notice that \underline{A} is the system matrix in eq. (2.2.8)

Now we will calculate \dot{V} .

$$2.2.18 \quad \dot{V} = \underline{\delta}^T \underline{B} \underline{\delta} + \underline{\delta} \underline{B}^T \underline{\delta}$$

$$2.2.19 \quad \dot{V} = (\underline{A}\underline{\delta} + \underline{b})^T \underline{B} \underline{\delta} + \underline{\delta}^T \underline{B} (\underline{A}\underline{\delta} + \underline{b})$$

$$2.2.20 \quad \dot{V} = \underline{\delta} (\underline{B}\underline{A} + \underline{A}^T \underline{B}) \underline{\delta} + 2 \underline{\delta}^T \underline{B} \underline{b}$$

Now we can use the result of 2.2.14-.16 for \underline{B} , write \underline{W} as given by (2.2.13, .17) and write \underline{b} as given by (2.2.6)

$$2.2.21 \quad \dot{V} = -2(k \|\underline{\delta}\|^2 - qb_2 \delta_1 - rb_2 \delta_2)$$

$$2.2.22 \quad \dot{V} < -2(k \|\underline{\delta}\|^2 - (q|\delta_1| + r|\delta_2|)(k_1 \delta_{s1} + k_2 \delta_{s2} + C + \epsilon_0))$$

where q and r are given from eqs.(2.2.15-.16). Now, \dot{V} is a function of $\delta_{1,2}$. There are sets of values of $\delta_{1,2}$, which is equivalent to saying there are parts of the δ_1 - δ_2 space, for which \dot{V} is negative. In these regions, V must be decreasing. However, we can always choose $k, k_{1,2}$ such that \underline{B} is positive definite, i.e. V is always positive and hence a norm. First of all, take k to be positive. Then, from eqs.(2.2.14-.16), for \underline{B} to be positive definite (equivalently for \underline{B} to have positive eigenvalues) we must have $p+r$ positive and $pr - q^2$ positive. An easy way to ensure this is to take $k_1 \gg k_2 > 1$. (Check this in the eqs. (2.2.14-.16)). Since V is a norm, it is always greater from $\lambda_{\min} \|\underline{\delta}\|^2$, and since λ_{\min} is a constant, $\|\underline{\delta}\|$ must also be decreasing. In other words, there are regions of the δ_1 - δ_2 space where $\|\underline{\delta}\|$ tends to decrease. Indeed, from looking at relationship (2.2.22), we can see that, with k positive, when $\|\underline{\delta}\|$ is large the second

order term dominates the first order term and, since k is positive, \dot{V} is certainly negative and consequently V and $||\underline{\delta}||$ tend to decrease. This seems to imply asymptotic stability, but we also note that as $||\underline{\delta}||$ gets small the first order terms $\delta_{1,2}$ dominate, and \dot{V} takes the sign of q, r . But because of the way we selected $k_{1,2}$, q and r will be positive. So there is a region close to the origin, where $||\underline{\delta}||$ tends to increase. However, as soon as the system escapes this region, $||\underline{\delta}||$ tends to decrease and it returns. Now, the exact shape of this attractive region, is not easy to determine analytically. It depends on several parameters. First it depends on $k_{1,2}$, that is, the gains of the controller. As $k_{1,2}$ tend to infinity, the attractive region shrinks and the system tends to be asymptotically stable. It is also clear that if the approximation error becomes comparable to the tracking error, then essentially we get in relationship (2.2.22) a second quadratic form, the sign of which may be positive or negative. So, the sign of \dot{V} depends on the modelling error, and this dependenc is expressed by the bounds ϵ_0, C on δf and the Taylor terms, as well as on the approximation errors $\epsilon_{s1, s2}$. Since the bounds on the absolute values of these terms multiply the first order terms in 2.2.21, so they tend to increase the size of the attractive region (= tend to increase the size of the instability region). Let's clarify this: the attractive region is the instability region- when the error vector is in there it tends to escape. So, the smallest we can make it, the smaller are the bounds on the error.

Given the parameters $C, k_{1,2}, \epsilon_{0,s1,s2}$, we can analytically determine the shape of some attractive region. We say "some", because in the process we increase several inequalities and we get an excessively big, that is excessively conservative, estimate. We do that in the Appendix. However, there is a better and easier way to determine the attractive region; we can numerically compute the value of V for a sufficiently broad range of δ_1, δ_2 and for given values $k_{1,2}$. If we want, we can experiment with several different values and choose the most satisfactory configuration. We can also investigate the influence of modelling uncertainty by changing ϵ_0, C , and of approximation accuracy by changing $\epsilon_{s1,s2}$. We give the results with several different combinations of parameters in the following sections. What is clear is that there are attractive regions that can be made almost arbitrarily small.

The reason we cannot shrink the attractive region to zero is that every dynamical system is subject to random disturbances and random variations of its parameters. In fact, this is the reason that we introduce feedback. We have in our analysis taken care of the variation in parameters, by introducing the δf term. And we have showed that we can ensure the existence of a bounded attractive region for the tracking error. But we should also consider the effect of random disturbances. Sure enough, their effect cannot be too great, especially in the case of an industrial manipulator, which is a quite deterministic device. However, they will be above a certain treshold most of the time. All this is rather bad probability theory, since $\delta_{p1,p2}$ are random variables and can take, at least

theoretically, any value. One way or another, by appealing to common sense, or by introducing some kind of argument with variances, equivalence of time and phase means and so on, we can write the inequalities (2.2.25-.26). We express this by assuming bounds on the probabilistic error $\delta_{p1,p2}$. Look at this error as part of $\delta_{1,2}$. (For example the approximation error $\delta_{s1,s2}$ is a random error which does not go to zero, but is not too big, either.) In other words, assume

$$2.2.23 \quad \delta_1 = \delta_{p1} + \delta_{d1}$$

$$2.2.24 \quad \delta_2 = \delta_{p2} + \delta_{d2}$$

$$2.2.25 \quad \epsilon_{p1} < \delta_{p1} < \epsilon^{p1}$$

$$2.2.26 \quad \epsilon_{p2} < \delta_{p2} < \epsilon^{p2}$$

Where $\delta_{d1,d2}$ stands for deterministic error, that is error due to parameter variation, approximation error etc. (Of course, in the final analysis this type of error is as probabilistic as $\delta_{p1,p2}$ but, at least, it is given and determined in the course of the control process)

It is useful for future analysis, to recognize the existence of randomness. For example, we might have taken pains to tighten the bounds of our attractive region as much as possible, but we would realize that after a point the deterministic error is bounded to values comparable to what the ubiquitous probabilistic error will be and we cannot do any better than that. This is not only a philosophical argument, either. We can see in the numerical studies of the attractive regions, and also in the analytical considerations of this section, that a way to shrink the attractive region

is to increase the gains $k_{1,2}$. This would be an ideal solution, since the deterministic error decreases in such a way that the product $k_1\delta_1+k_2\delta_2=u_c$ would go to zero. See for example Fig.2.2.1. This would be true if we could ensure that the probabilistic error goes to zero as well, but this cannot be done. The only problem then is that we cannot increase the gains more than a certain level, because then the product $k_1\delta_{p1}+k_2\delta_{p2}$ would be too high. In other words we would have to commit too much control for corrective action, which would leave too little for bang-bang, time optimal control. We will deal with this problem in section 3.1. However, let us note that in accordance to the gross motion control objective we stated in the Introduction, we can delineate a sufficiently small neighborhood around the nominal trajectory, in which the system will always lie.

The reader has certainly noticed the similarity of this method to Lyapunov stability concepts. In fact our V function is a Lyapunov function candidate, but unfortunately does not fulfill all the requirements, at least not in all of the $\delta_1-\delta_2$ space. However it does quite a lot for us. In fact, as far as we know this is quite an original method of determining stability regions, and even though it does not allow for very fine delineation of the stability region, its simplicity makes it quite attractive. There are several other works on the determination of stability regions, either by analytic methods (e.g. [23-27]), or by numerical methods, e.g. [28-31]).

2.3 NUMERICAL STABILITY ANALYSIS: Here we propose a numerical method for the determination of stability regions. We start by rewriting eq.(2.2.22).

$$2.2.22 \quad \dot{V} < -2(k||\delta||^2 - (q|\delta_1| + r|\delta_2|)(k_1\delta_{s1} + k_2\delta_{s2} + C + \epsilon_0))$$

This inequality determines one stability region of the system. In the Appendix we enhance the inequality and determine a bigger region, which is also a stability region. However, it is clear that we would like to use a method that will place tight bounds on this region. Now, what we can do is actually compute the values of the right-hand part of (2.2.22) and see for what points (δ_1, δ_2) it is negative. Then \dot{V} will also be negative. For given values of $k, k_{1,2}$ we can compute \underline{B} and make sure it is positive definite. Then we will have determined a stability region. Comparing with the analytical method of the Appendix, we see that indeed the numerical method provides tighter bounds.

There is one more point to notice: in the right hand of (2.2.22), everything is known, except $\delta_{s1, s2}$. Actually, we could compute these too, but since they are quite negligible, except when really close to the origin, we will just use the bounds $\epsilon_{s1, s2}$.

In summary then, we construct a grid of points (δ_1^i, δ_2^j) , and compute the values of the expression $J(\underline{\delta}) = -2(k||\delta||^2 - (q\delta_1 + r\delta_2)(k_1\epsilon_{s1} + k_2\epsilon_{s2} + C + \epsilon_0))$ for these points. We present the results for some sets of parameters. In all the

cases, the starred region is the attractive region. When the system gets outside of this, it returns to it. The estimation is conservative, so in reality the system might stay in a sub-region of the starred (attractive) region.

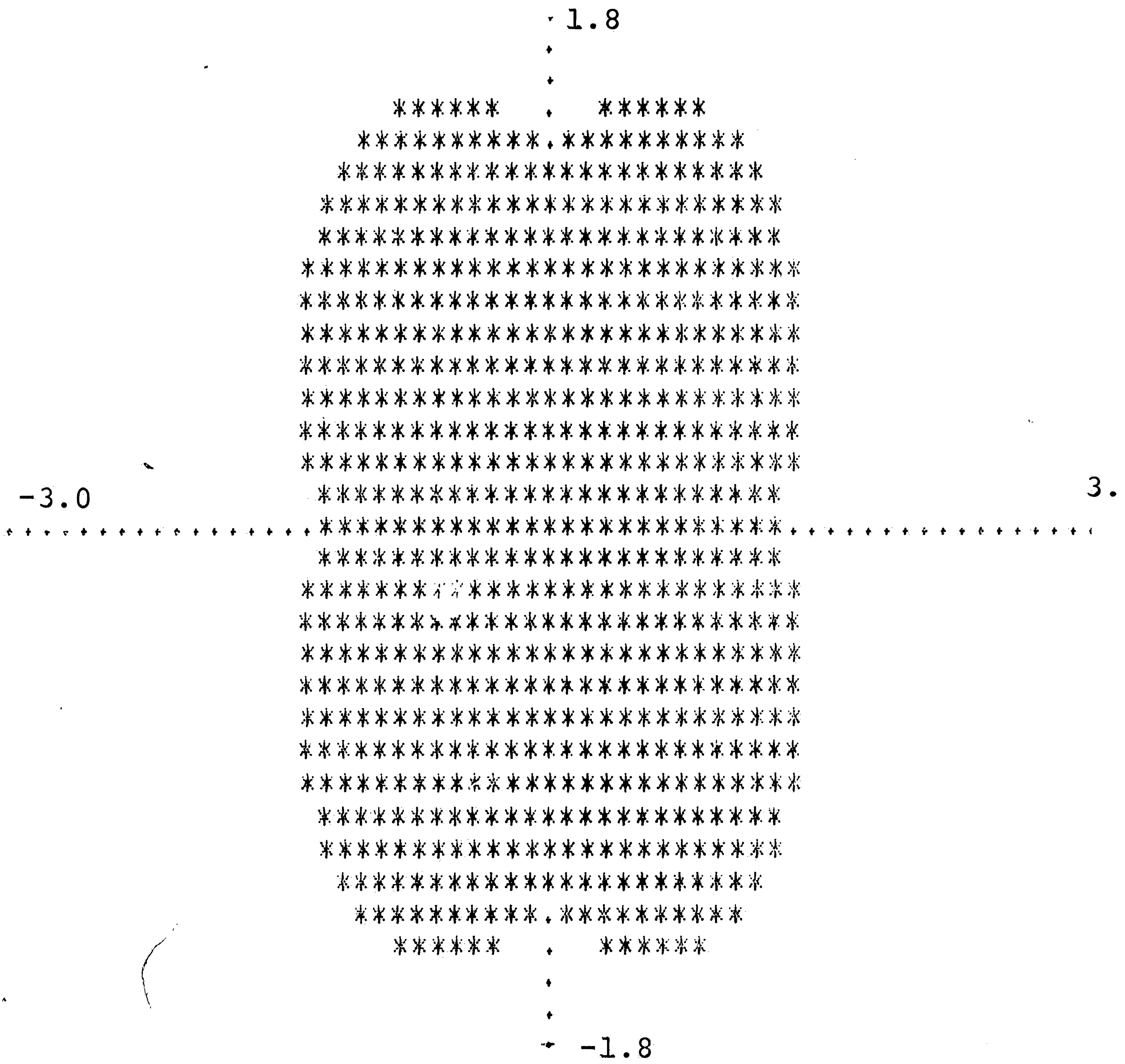


Fig. 2.3.1a: Guaranteed stability region for $k_{1,2}=-20.$, $C=40.$, $e_0=5.$

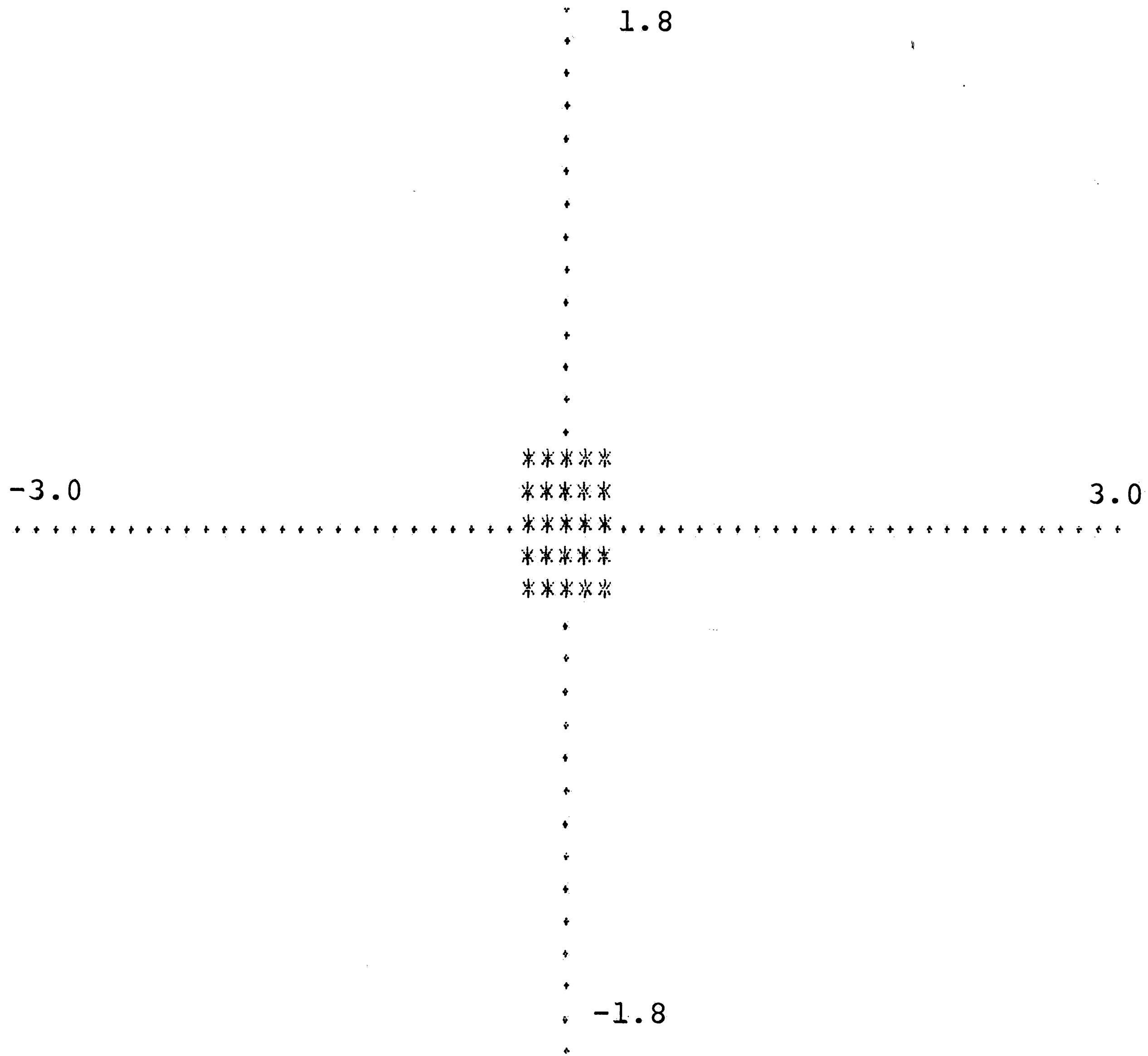


Fig. 2.3.1c: Guaranteed stability region for $k_{1,2}=-100.$, $C=40.$, $e_0=5.$

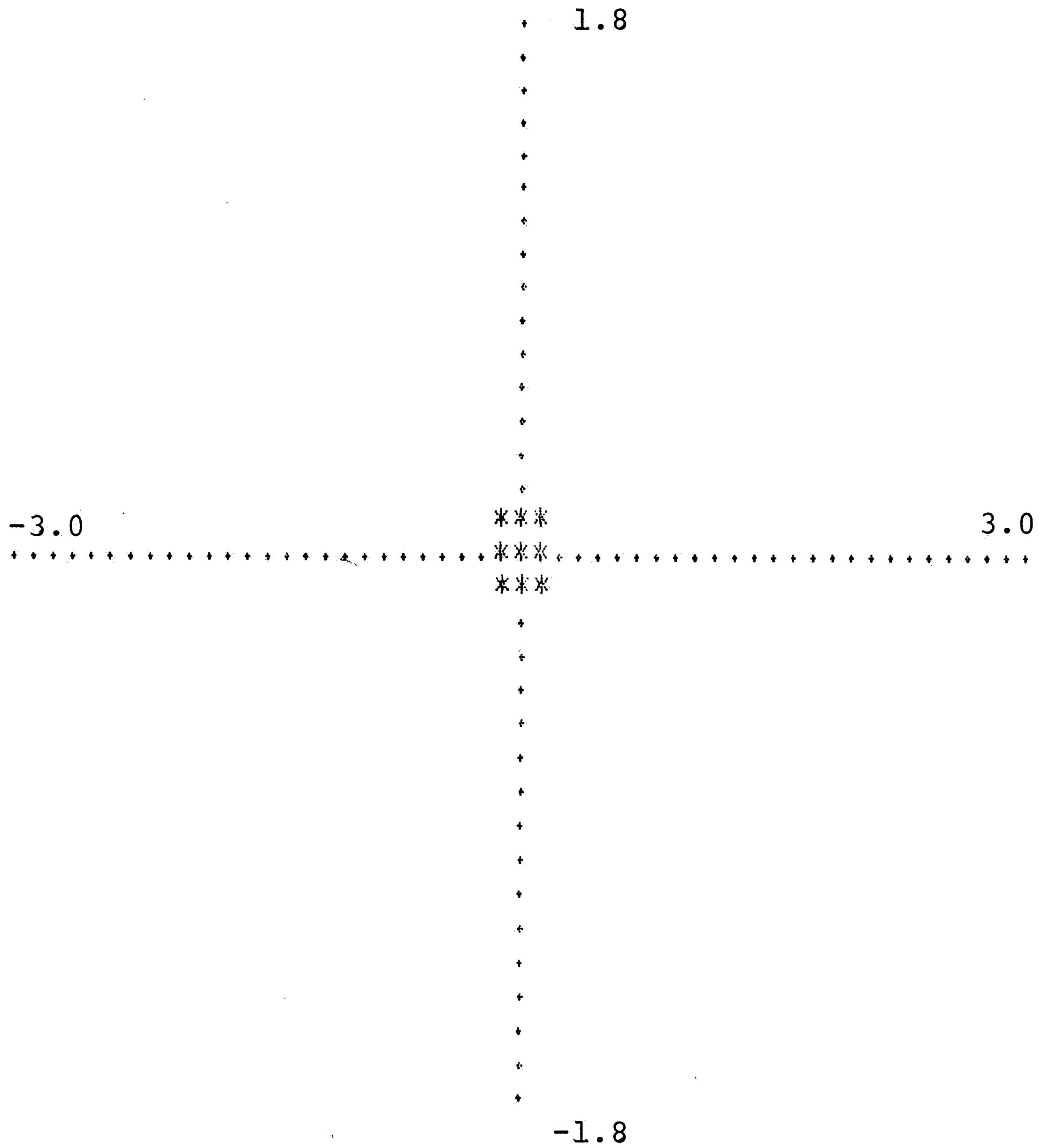


Fig. 2.3.1d: Guaranteed stability region for $k_{1,2}=-200.$, $C=40.$, $e_0=5.$

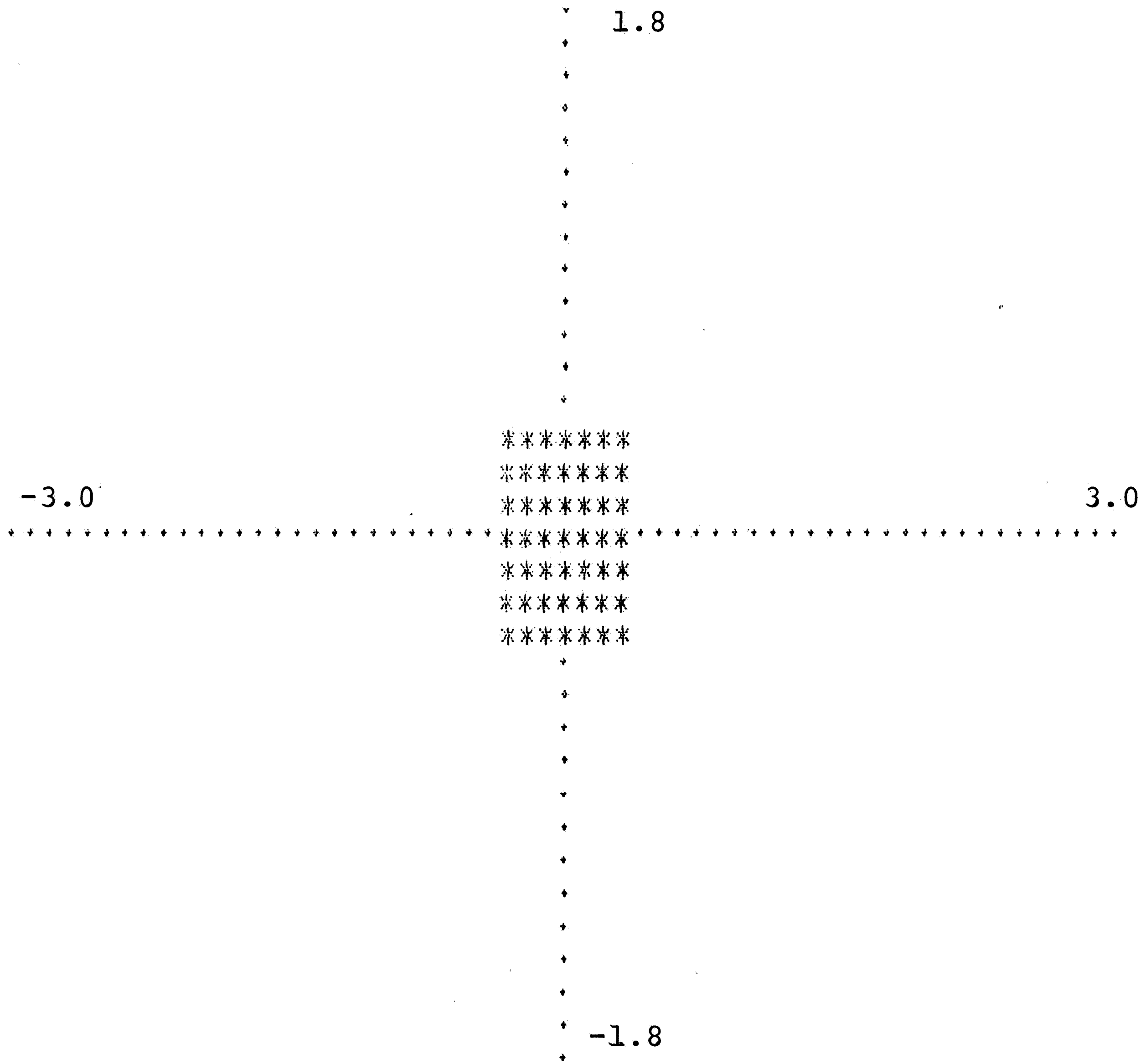


Fig. 2.3.1e: Guaranteed stability region for $k_{1,2}=-100.$, $C=40.$, $e_0=20.$

The preceding analysis can be performed for a discrete time system as well. An analytical method would prove quite tedious, because there is an added parameter, the sampling time h . However, numerical treatment does not present any particular problems.

The equations for a discrete time system, in matrix form are

$$2.3.1 \quad \delta_{k+1} = \underline{A}^* \delta_k + \underline{c}$$

Where the matrix \underline{A}^* is

$$2.3.2 \quad \underline{A}^* = \begin{bmatrix} 1 & h \\ -hk_1 & 1-hk_2 \end{bmatrix}$$

And $\underline{c} = h\underline{b}$. \underline{b} is the same as in the continuous time case and h is the time step.

Now, define $V(\delta_k)$ and $\Delta V(\delta_{k+1}, \delta_k)$

$$2.3.3 \quad V(\delta_k) = \delta_k^T \underline{B} \delta_k$$

2.3.4

$$\Delta V = V(\underline{\delta}_{k+1}) - V(\underline{\delta}_k)$$

And after some computation, we can write

2.3.5

$$\Delta V = \underline{\delta}_k^T (\underline{A}^{*T} \underline{B} \underline{A}^{*T} - \underline{B}) \underline{\delta}_k - 2 \underline{c} \underline{B}^T \underline{A}^{*T} \underline{\delta}_k + \underline{c}^T \underline{c}$$

Now, we can try to determine values of $h, k_{1,2}$, such that the matrix equation

2.3.6

$$\underline{A}^{*T} \underline{B} \underline{A}^{*T} - \underline{B} = -k \underline{I}$$

is satisfied, where \underline{I} is the unit matrix, k is a positive constant and \underline{B} is positive definite, hence a norm. Then, the same kind of arguing as for the continuous time case indicates that V is a norm. Write $\underline{d} = \underline{c} \underline{B}^T \underline{A}^{*T}$. Then we can again write

2.3.7

$$\Delta V < -k \|\underline{\delta}\|^2 + 2(|d_1 \delta_1| + |d_2 \delta_2|) + \underline{c}^T \underline{c}_k$$

And if ΔV is negative in a region of the δ -space, this implies the norm V of $\underline{\delta}$ is decreasing. So again we have defined an attractive region. In the following figures we present the shape of attractive regions for some combinations of parameters $h, k_{1,2}$.

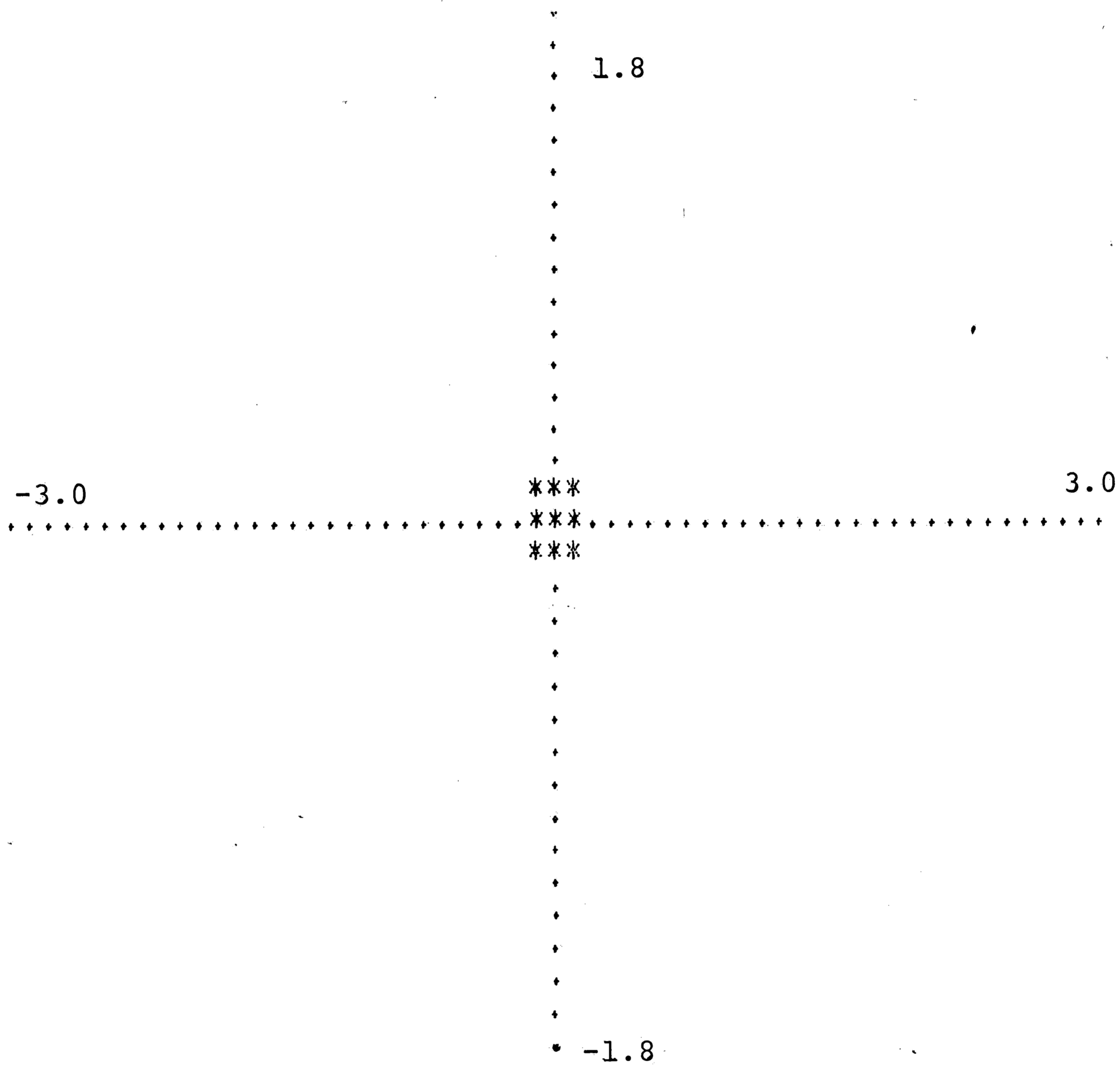


Fig. 2.3.1f: Guaranteed stability region for $k_{1,2}=-100.$, $C=40.$, $e_0=5.$

$h=.001$

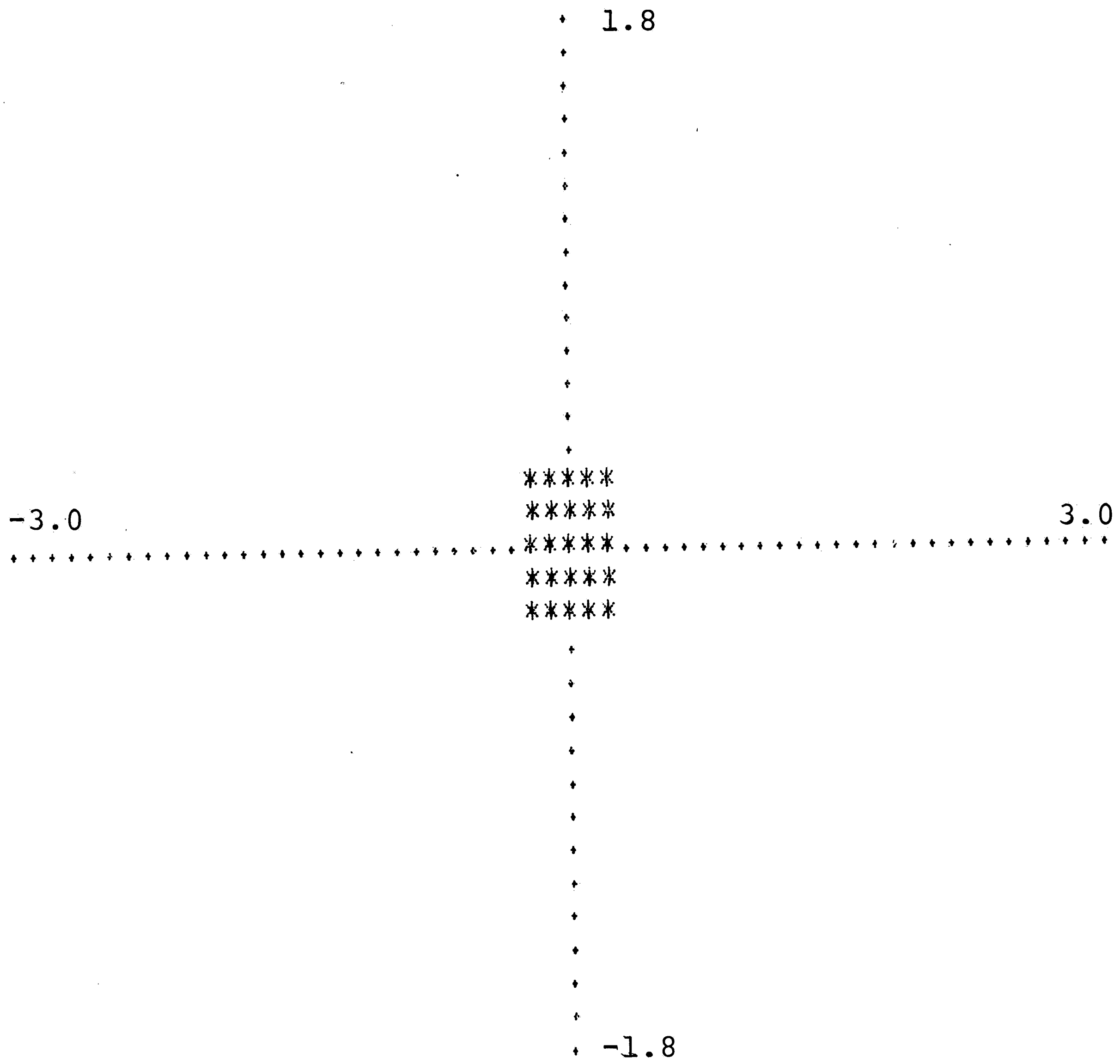


Fig. 2.3.1g: Guaranteed stability region for $k_{1,2} = -100.$, $C=40.$ $e_0 = 20.$ $h=.001$

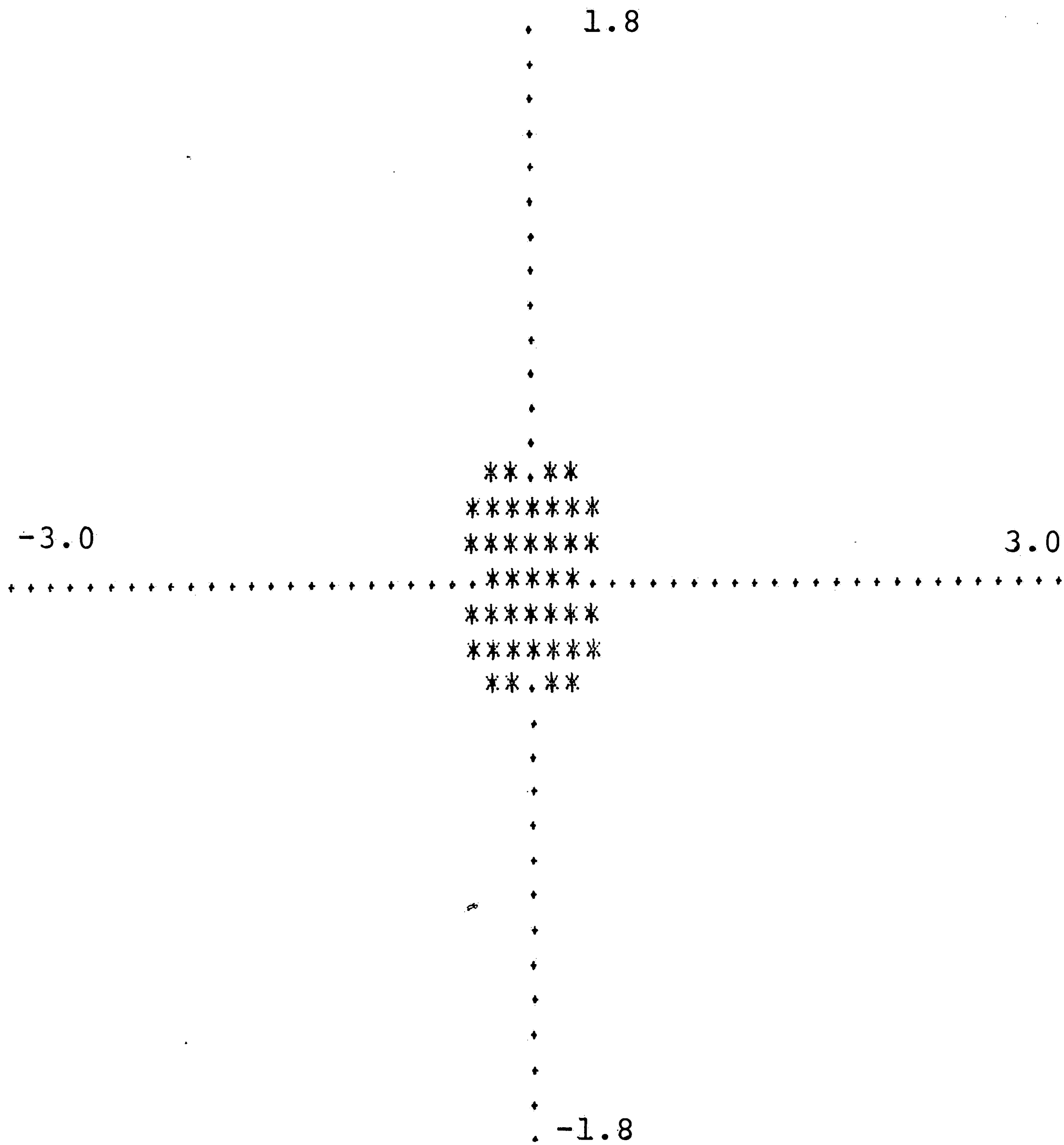


Fig. 2.3.1h: Guaranteed stability region for $k_{1,2}=-200.$, $C=40.$, $e_0=5.$ $h=.001$

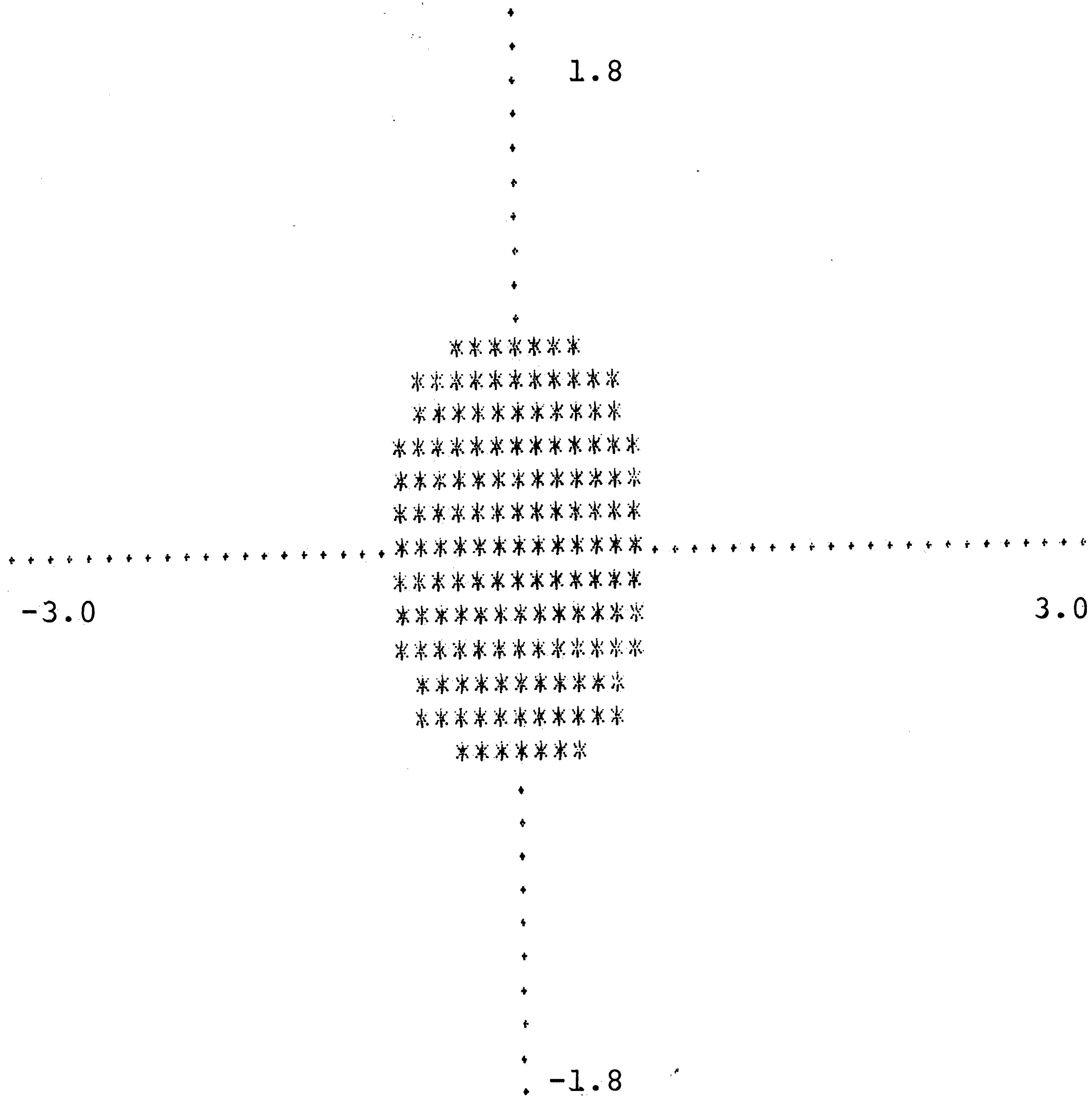


Fig. 2.3.1i: Guaranteed stability region for $k_{1,2} = -50.$, $C=40.$, $e_0 = 5.$ $h=.01$

CHAPTER 3: TIME OPTIMALITY

3.1 THEORY: This chapter is devoted to a study of the time optimality characteristics of our method. Specifically, the problem is as follows:

Suppose that we design a control schedule $u(t)$ so as to minimize the transition time t_f from an initial state $\underline{x}_i(0)$ to a target state $\underline{x}_f(t_f)$.

We have already mentioned that for the type of systems we treat in this work (linear in the control , second order) the appropriate control is bang-bang (See Athans [3], Bellman et. al. [32]) with one switch in the sign of the control function. Now, we have already explained that there are at least three ways to implement the control. The first is switching surface control: a function of the state variables is continuously computed , the sign of which determines the sign of the control. The second method is preprogrammed control: compute the switching time offline and perform the sign switch at that time; this is open loop control. We proposed a third intermediate method. We use the preprogrammed control, but we also add the corrective control that ensures that the actual system trajectory never goes too far away from the nominal trajectory.

We will be interested in the sensitivity of the time optimality in each of the three methods. We will mention some results relevant to the first two, and then we will give the analysis for our own method.

There are several papers ([5-10]) dealing with the closed loop case (first case). The problem is that variations of the system parameters may render the method inadequate, in the sense that as the system approaches the target state we may have an infinite sequence of switchings. Then the system enters a limit cycle, revolving around the target state but never reaching it (see Figs.3.1.1). Obviously, in that case, the time t_f goes to infinity. Since the system might never get close enough to the target state we can not claim even sub-optimality. The system simply performs inadequately.

In the second method, the control is applied, by definition, exactly for time t_f . However, there is no guarantee of arrival at the target state.

Dubowsky brings this point up in [4]: slight variations in the system parameters can cause sharp variations in the final state. We performed several simulations to investigate this effect. See Fig.(3.1.2), where it is clearly demonstrated that small changes in the control can bring a significant change in the final position.

We were motivated to introduce our method to remedy this situation. The addition of the corrective control u_c can guarantee that the tracking error will remain bounded. The bound will be the greater of two quantities: the maximum value of the deterministic error and the maximum value of the random error (as given in eqs. (2.2.25-.26)). For the following analysis of time optimality, we will only need δ_2 to be bounded, so let us write, assuming we have designed the controller so as to get an appropriate attractive region and taking into consideration the existence of a probabilistic error,

3.1.1

$$\delta_2 < \epsilon_d$$

We will also require that the state variable x_2 will be greater than a positive number. Let us justify this assumption. First of all, it is clear that since the control is time optimal, the state x_2 , which is essentially the "speed" of our system, will always be of one sign (for a proof see Athans [3]). So we can take it to be nonnegative without loss of generality. We also assume that it will not be zero. Often, in time optimal problems, we want the system to start from a state of rest ($x_2=0$) and end at a state of rest ($x_2=0$). However, as we have mentioned, we are interested in gross motion control, so we want to start in a neighborhood of the initial state and end in a neighborhood of the target state. As will be clear soon, all we need is that

3.1.2
$$\epsilon_x < |x_2| \quad \text{for all } t, 0 < t < t_f$$

3.1.3
$$\epsilon_d < \epsilon_x$$

A possible objection is that the system starts exactly from a state of rest ($x_2(0)=0$). Even in that case, as soon as it starts moving away from $x_2=0$, (3.1.2) is satisfied, so our time optimality analysis is valid for the rest of the path.

The fact we want to demonstrate is that if the system moves along a given path at approximately the prescribed speed, it will arrive at its destination at approximately the prescribed time.

Let us start with the observation that the state variables x_1, x_2 are functions of the independent variable t , that is $x_1=x_1(t), x_2=x_2(t)$. If we take parts of the trajectory where the state variables are monotone functions of time, we can also invert either one of the two functions and express t as a function of one of the two state variables. So for the description of the actual system we can use as the independent variable either t, x_1 , or x_2 . The same holds true for the nominal system. Denote by t_n and t the time variables for the nominal and actual systems, respectively.

Assume the nominal total time is t_{nf} and the actual total time is t_f .

Finally, $t_{nf}=\int dt_n, t_f=\int dt$. Let us then rewrite eqs.(2.1.14-.16)

$$3.1.4 \quad \frac{dx}{dt} = x_2$$

$$3.1.5 \quad \frac{dx}{dt_n} = x_{n2}$$

Equivalently (remember x_2, x_{n2} are greater than zero)

$$3.1.6 \quad \frac{dx}{x_2} = dt$$

$$3.1.7 \quad \frac{dx}{x_{n2}} = dt_n$$

Now, let us take the independent variables to be x_1, x_{n1} . Since this is just a dummy variable, we can write $x_1=x_{n1}$. On the other hand, $x_2(x_1)$ is a different function from $x_{n2}(x_1)$ and the same holds for $t(x_1), t_n(x_1)$. So, taking $x_1=x_{n1}$, combining eqs.(3.1.6-.7) and writing $\delta t=t_n-t$, which also implies $d(\delta t)=dt_n-dt$, we get

$$3.1.8 \quad \frac{dx}{x_2} (1 - \frac{x_2}{x_{n2}}) = d(\delta t)$$

$$3.1.9 \quad \frac{dx_1}{x_2} (1 - \frac{x_{n2} - \delta_{n2}}{x_{n2}}) = d(\delta t)$$

$$3.1.10 \quad \frac{dx_1}{x_2} - \frac{\delta_{n2}}{x_{n2}} = d(\delta t)$$

And taking absolute values we get

$$3.1.11 \quad \left| \frac{dx_1}{x_2} - \frac{\delta_{n2}}{x_{n2}} \right| = |d\delta t|$$

Now, note that $\frac{dx_2}{x_2} = dt$, t is always positive and $\int |dx| > \int d|x|$. Using these

facts and our boundedness assumptions (eqs. (3.1.2-.3)) we get

$$3.1.12 \quad dt \frac{\epsilon_d}{\epsilon_x} > d|\delta t|$$

And integrating

$$3.1.13 \quad t \epsilon_d / \epsilon_s > \delta t$$

The meaning of this relationship is clear: when the speed error is not great (in particular when it is considerably smaller than the gross motion speed) then the transition time $t_f + \delta t$ cannot differ by much from the optimal time.

However, this is not totally fair, because we have ignored the fact that we have committed a portion of the control to corrective action. If we used all

the available control for bang-bang control, what would the improvement in time optimality be? We will answer this question analytically just for the case of the pendulum, as described by eqs.(1.1.3-.4), but this will offer insights for a general numerical solution.

Consider then, two systems of the form of eqs.(1.1.3-.4), with inputs u_1, u_2 respectively, both of which are driven in the bang-bang mode, u_1 being equal to $+U$ for the first one (system 1) and u_2 being equal to $+(U+\delta U)$ for the second one (system 2). The initial and target states are identical, and there is no parameter variation or probabilistic errors. The appropriate switching times t_{s1}, t_{s2} are calculated for each system and the switchings take place exactly at the right times. What is $\delta t = t_{f1} - t_{f2}$? Here we choose to treat the state variable x_2 as the independent variable, and so 3.1.14

$$t_{f1} = \int dt_1(x_2), \quad t_{f2} = \int dt_2(x_2).$$

Note that we can take δU positive, so that δt will be positive. Second, write expressions for dt for each system. For system 1

$$3.1.15 \quad dt = dx_2 / (A \sin x_1 + Bx_2 + u_1)$$

and for system 2

$$3.1.16 \quad dt = dx_2 / (A \sin x_1 + Bx_2 + u_2)$$

Now, use the following shorthand notation

$$3.1.17 \quad Q = A \sin x_1 + Bx_2 + U$$

$$3.1.18 \quad R = -|A| + Bx_2 + U$$

$$S = A \sin x_1 + B x_2 - U$$

Obviously, if U is big enough Q and R will be positive and S will be negative. Also, for all U , $R < Q$, so we can write (for U big enough)

$$3.1.17 \quad S < 0 < R < Q \quad \text{for all } x_{1,2}$$

Now, dx_2/dt is equal to Q for $t < t_{s1}$ and is equal to S for $t > t_{s1}$. We assumed that U is great enough that for $t < t_{s1}$, dx_2/dt is always positive and for $t > t_{s1}$, dx_2/dt is always negative (This has indeed been the case in all our simulations).

Then for each piece of the trajectory, $x_2(t)$ is a one-to-one, monotone function, that can be inverted and written as $t(x_2)$. To find then the total time t_{f1} , all we have to do is to integrate each part of the trajectory individually, find the times and add them. The same holds true of t_{f2} . Of course, if we go to all that trouble, we could as well run the simulation of the system with inputs U and $U + \delta U$ and see what the total times $t_{f1, f2}$ are. Indeed we will use this technique in what follows. For the time being, however, it is enough to know that there are such times $t_{s1, s2}$, $t_{f1, f2}$ that can be computed if necessary.

To show how the method would work, we will only deal with the first part of the trajectory. Things work out exactly the same for the second half. Now,

we will limit the discussion to the time interval $0 < t < t_{s1, s2}$. In that interval, $Q = dx_2/dt$. And we can write

$$3.1.20 \quad t_{f1} = \int dx_2/Q$$

We integrate then with respect to x_2 from 0 to $x_{s2}(t_{s1})$. Note that for this portion of the path $Q > 0$. We will drop the integration limits from now on. Then, in the same spirit as eq. (3.1.20) we can write

$$3.1.21 \quad t_{s1} = \int dx_2/Q$$

$$3.1.22 \quad t_{s2} = \int dx_2/(Q + \delta U)$$

$$3.1.23 \quad = \int dx_2/Q (1 + \delta U/Q) \\ = \int (dx_2/Q) (1 - (\delta U/Q) + (\delta U/Q)^2 - \dots)$$

By developing the term $1/(1 + \delta U/Q)$ in a Taylor series and retaining only the first order term, we strictly decrease the right hand term of (3.1.23) (no approximation in this!). Then we can write

$$3.1.24 \quad t_{s2} > \int (dx_2/Q) (1 - \delta U/Q)$$

$$t_{s2} > \int (dx_2/Q) - \int (dx_2/Q^2)$$

$$3.1.25 \quad t_{s2} > t_{s1} - \delta U \int dx_2/Q^2$$

$$3.1.26 \quad \delta t_1 < \delta U \int dx_2/Q^2$$

Where we denote $t_{s1} - t_{s2} = \delta t_1$. Now, we will further increase the inequality

by decreasing the denominator of the integrand to R^2 . Then

$$3.1.27 \quad \delta t_1 < \delta U \int dx_2/R^2 = \delta U \int dx_2/(|A| + Bx_2 + U)$$

3.1.28

$$\delta t_1 < \delta U (x_{s2} / ((-|A|+U) (Bx_{s2} - |A|+U)^2))$$

Things work out exactly the same way for the other part of the trajectory.

If we add the two times, we get $\delta t = K\delta U$, where $K = 2x_{s2} / (U - |A|)(U - |A| + Bx_{s2})$.

This shows that the dependence of decrease in time to increase in input is bounded by a linear function of δU . We can easily show that the same will hold true for all systems linear in the control, except it may be harder or impossible to evaluate the integral that yields the factor K . However, we can take, say, two simulations with different controls and interpolate a linear relationship between $\delta t, \delta U$.

If we take more than two simulations, we will see that our linear relationship holds for a neighborhood around a nominal control, but far away from this it becomes nonlinear. This is only to be expected, since we have made a hidden assumption. In eq.(3.1.24) the integral $\int dx_2/Q$ would be t_{s1} if the upper limit of integration were $x_2(t_{s1})$. Instead, it is $x_2(t_{s2})$. However, if δU is not very big, then we expect x_2 not to change much in maximum value (of course the total δt might still be quite significant, if small differences in speed accrue over an interval of time). In particular, in our pendulum example, the $(1/(x_{s2}+a))$ function tends to further smooth variations in x_2 . So for a fairly wide range of δU we get a linear time/control relationship, which is ascertained from the simulations.

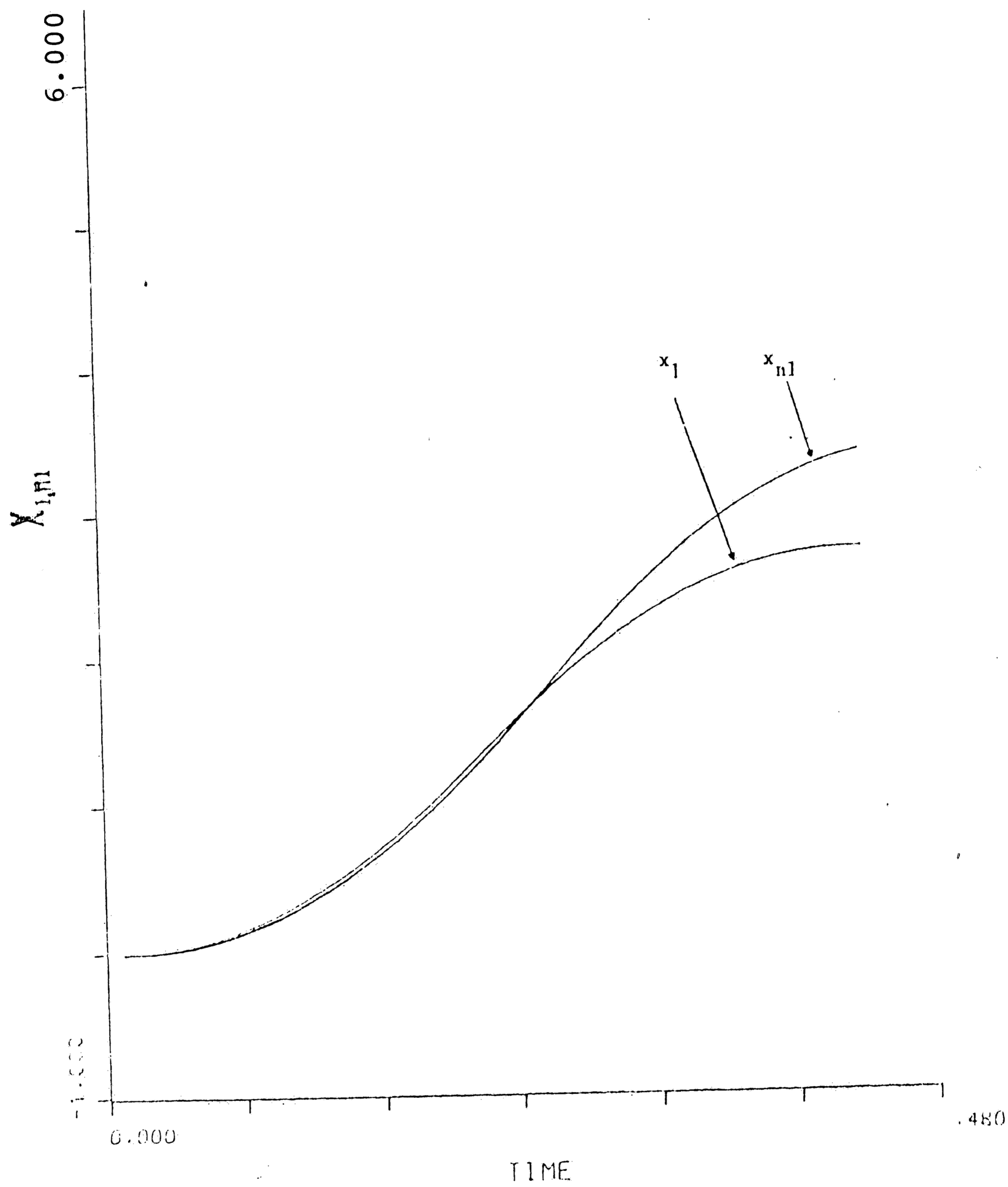


Figure 3.1. a: Effect of t_s variation

Change in x_1 . $t_{ns} = .21$, $t_s = .24$

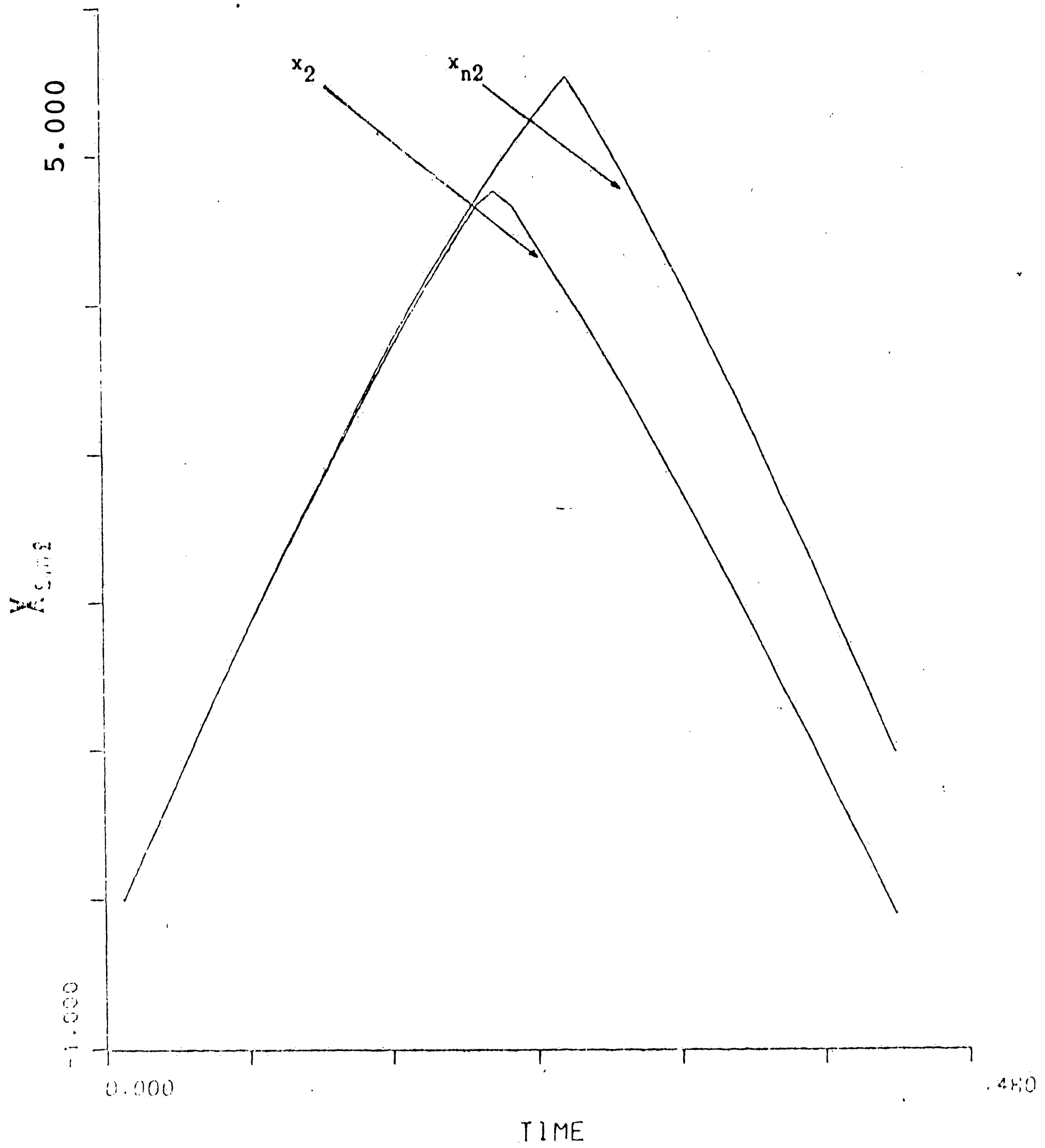


Figure 3.1. b: Effect of t_s variation

Change in x_2 . $t_{ns} = .21$, $t_s = .24$

3.2 NUMERICAL CONSIDERATIONS; Even though we developed a bound for dt , we could treat the problem numerically right from the beginning. In this section we present some results of this numerical study. What we did was to run the optimal time system (1.1.3-.4) for several different values of the input U . Since we have been using the pendulum example for all our simulations in past chapters, we also used it in this case. We summarize our results in the following Table 3.2.1

TABLE 3.2.1

U	t_f
10.	.605
12.	.535
14.	.515
15.	.475
16.	.445
*16.9	.425
18.	.415
19.	.405
20.	.395
22.	.375

We also exhibit these results in Figs. 3.2.1a,b. Here we have plotted the 11 pairs of $dt-dU$, equivalently the 11 pairs t_f-U , which we list in Table 3.2.1. In Fig. 3.2.1a we just plot the discrete points, whereas in Fig. 3.2.1b we interpolate for intermediate points so as to get a smoother curve. We see that around the point $U=16.9$, which is the value used for U in the simulations of Chapter 4, we have a fairly linear behavior, so we can assume that for intermediate values from these actually computed, we can interpolate linearly.

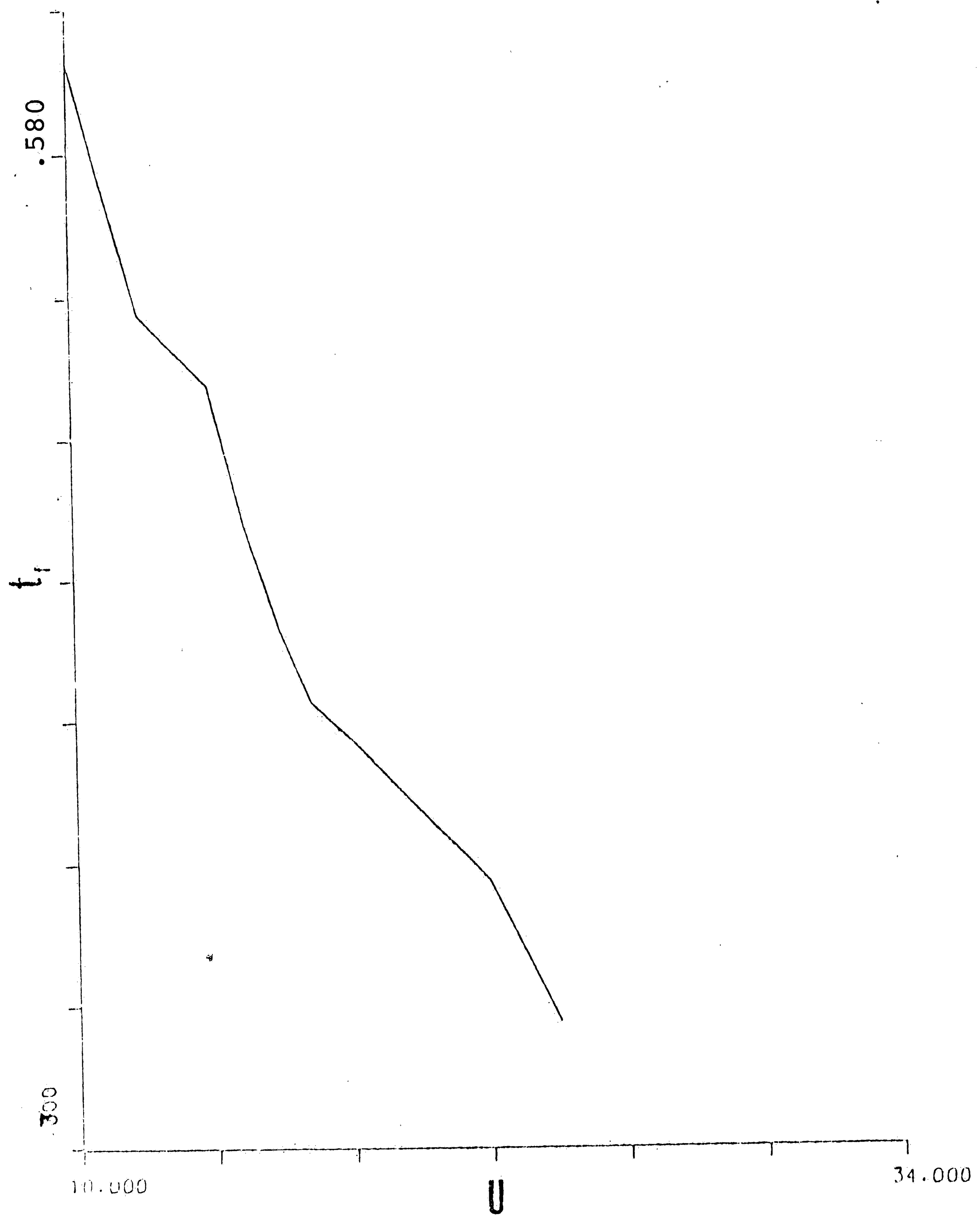


Fig.3.2.1a: Plot of the total time t_f vs. input U

Ten points are plotted and joined by straight lines

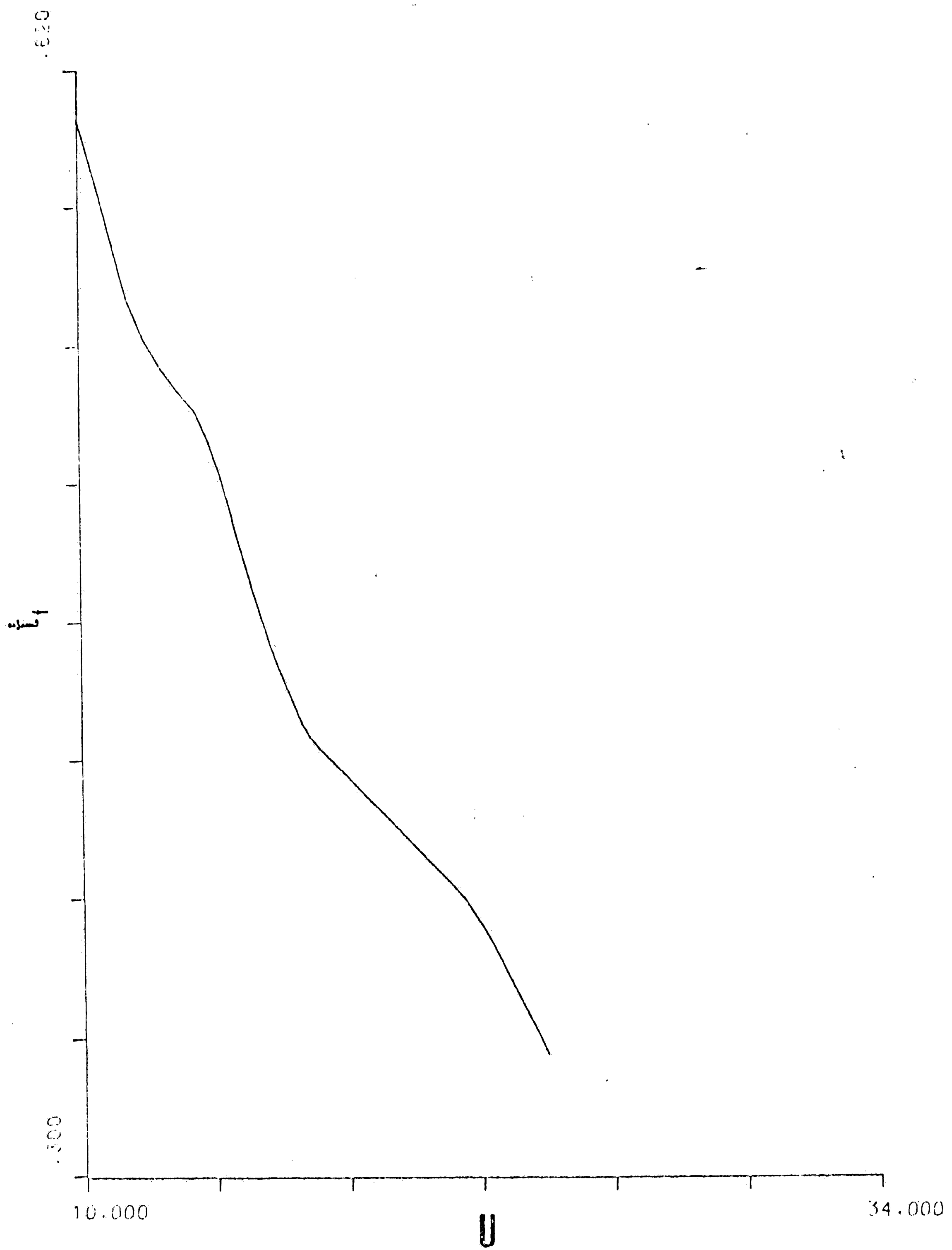


Figure 3.2.1b: Plot of the total time t_f vs. input U

Ten points are plotted and linear interpolation is used to join more intermediate points.

CHAPTER 4: SIMULATIONS

4.1 SIMULATIONS: In this Chapter we will present the results of several numerical simulations that we performed and point out several conclusions that can be drawn from them.

There are some particular characteristics of each one of the simulations. The following factors characterize each particular simulation:

- (a) The task we want the system to perform.
- (b) The differential equation of the model.
- (c) The differential equation of the actual system.
- (d) The nominal initial position of the system.
- (e) The actual initial position of the system.
- (f) The actual switching time (if any) of the system.
- (g) The nominal switching time (if any) of the system.
- (h) The sampling rate .
- (i) The approximation functions used.

Item (a) is fully characterized by the preprogrammed control u . Item (b) is characterized by the fact we always use the equation

4.1.1 $\dot{x}_1 = x_2$

4.1.2 $\dot{x}_2 = -16.6 \sin x_1 - .2x_2 + u$

Item (c) is characterized by df. Items (d), (e), (f), (g), (h) are clearly characterized by one numerical value each. For item (i), we always use splines of eight intervals.

For each of the simulations we list each of these factors, except (b) and (i), which, as we said, are always the same. Then we present plots of the following seven functions of time: $x_1(t)$ and $x_{n1}(t)$, superimposed in the same plot, $x_2(t)$ and $x_{n2}(t)$, again superimposed, $d_1(t)$, $d_2(t)$ and $u(t) + u_c(t)$. The superposition is used to display how well the system performs the required task. In the next section (4.2) we comment on the results.

Figure 4.1.1: Ideal conditions simulation

Here we have an "ideal" setup, in which all of the model parameters agree completely with the actual system.

Input: Bang-bang (nominal $t_s = .21$ sec)

$df=0$.

nominal initial position: $x_{n1}(0) = -.6$, $x_{n2}(0) = 0$.

actual initial position: $x_1(0) = -.6$, $x_2(0) = 0$.

actual $t_s = .21$

sampling interval $h = .01$

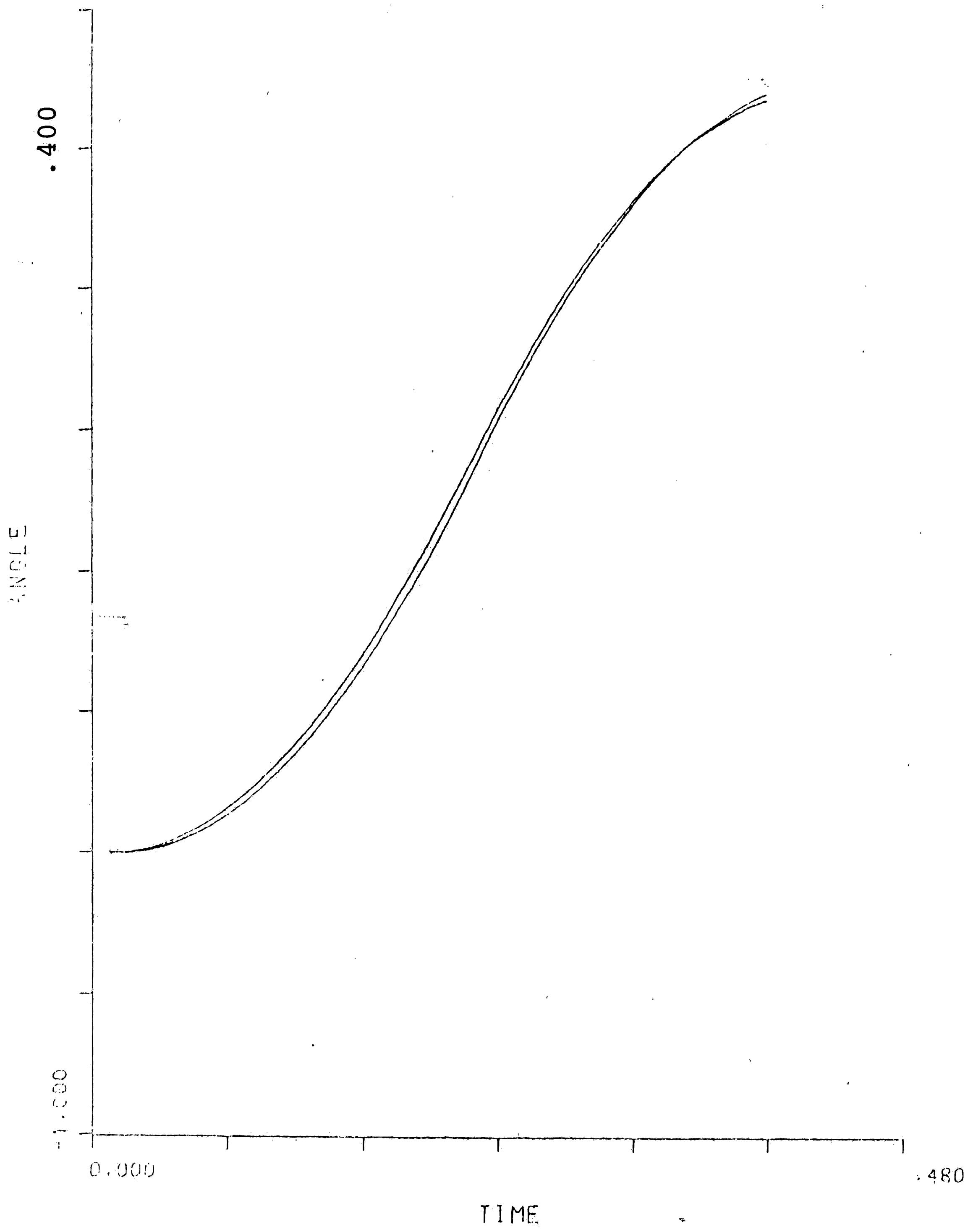


Figure 4.1.1.a

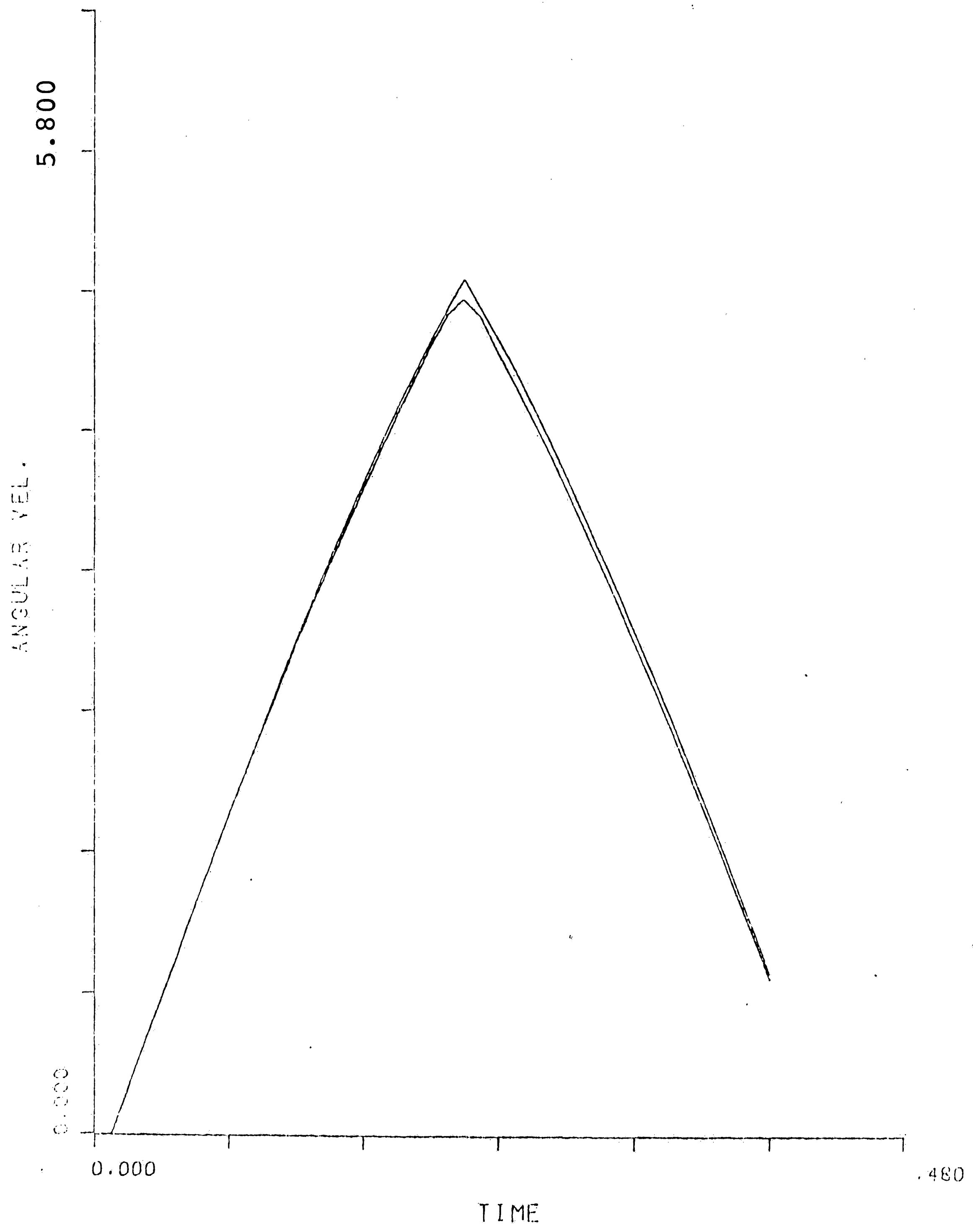


Figure 4.1.1.b

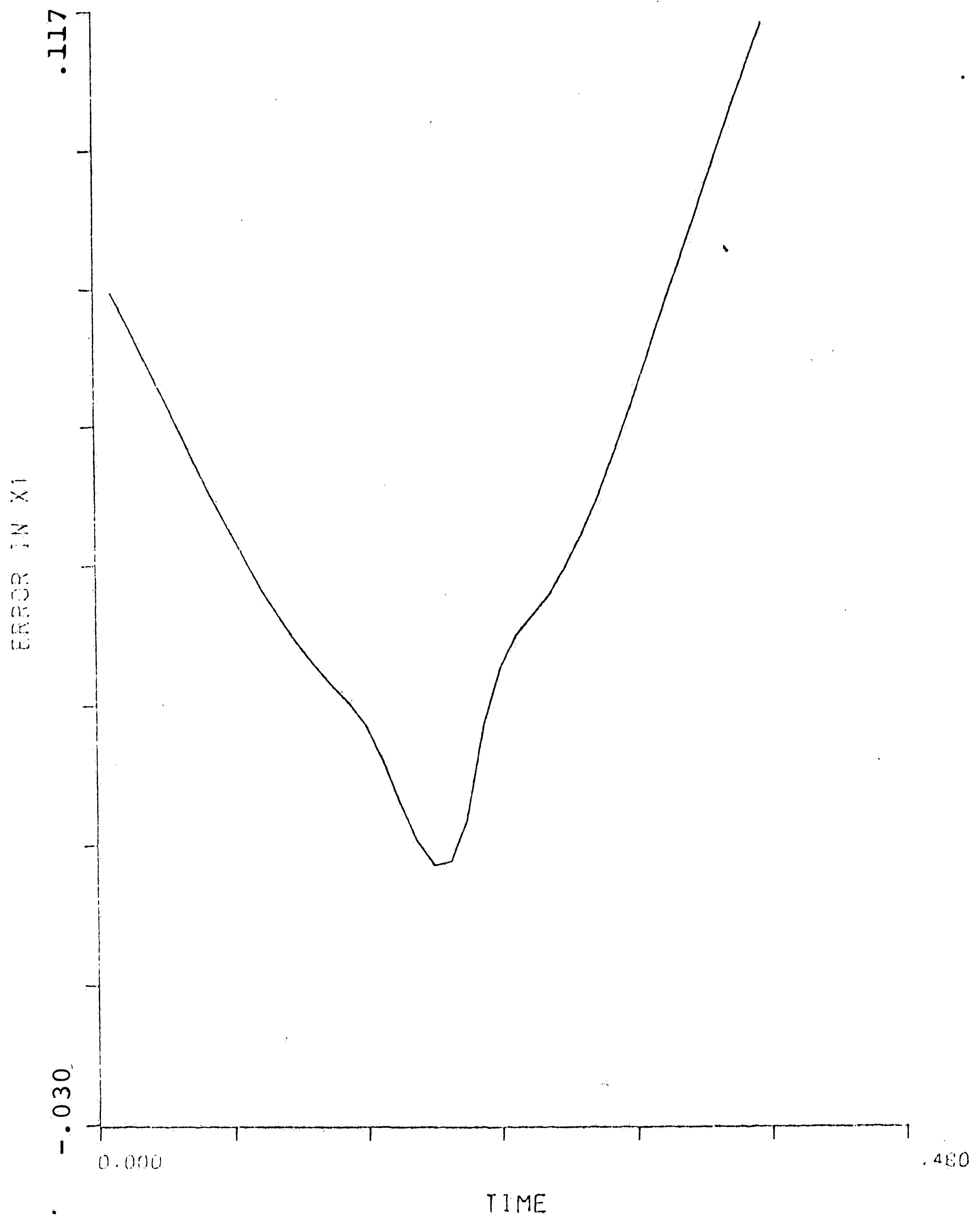


Figure 4.1.1c

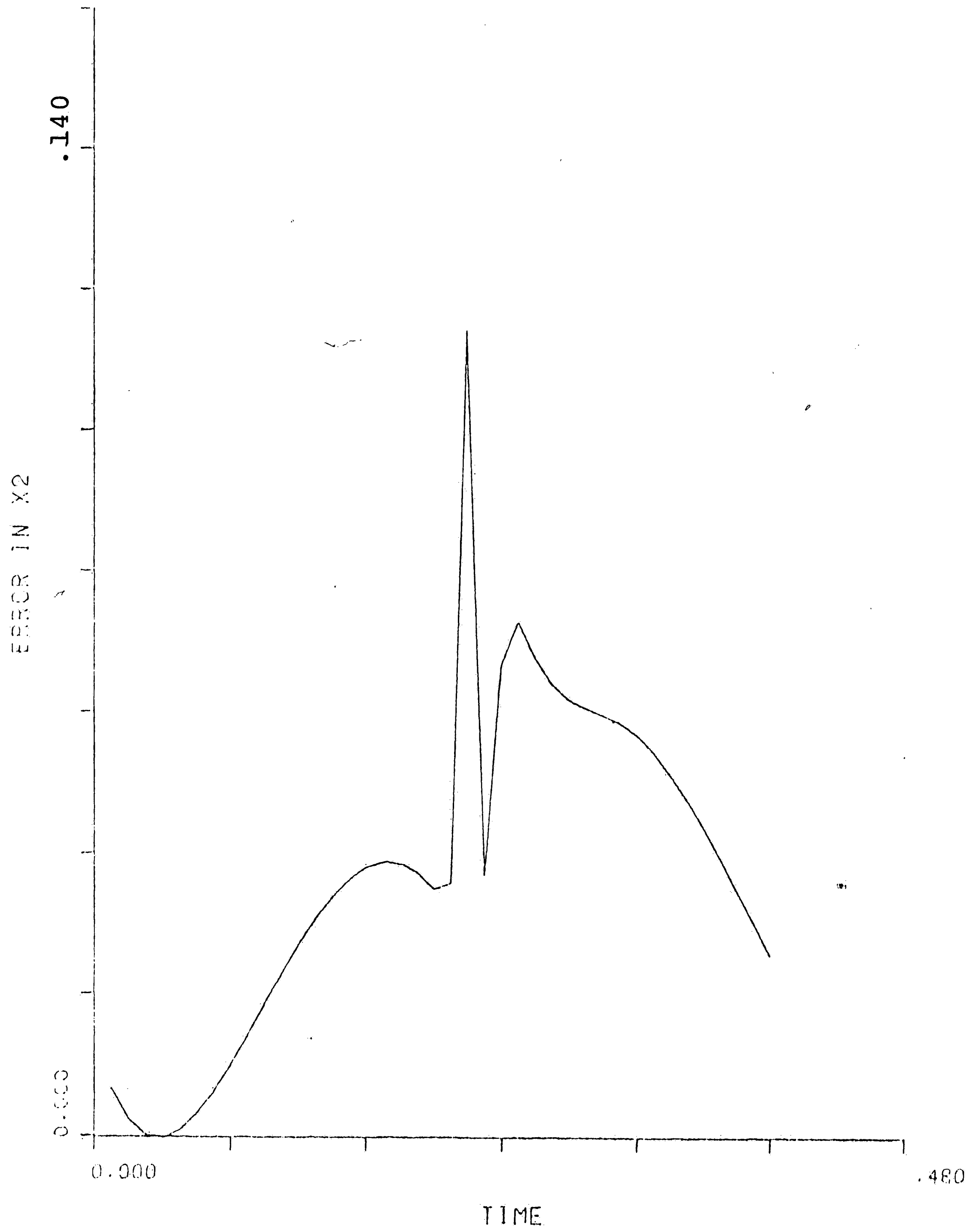


Figure 4.1.1d

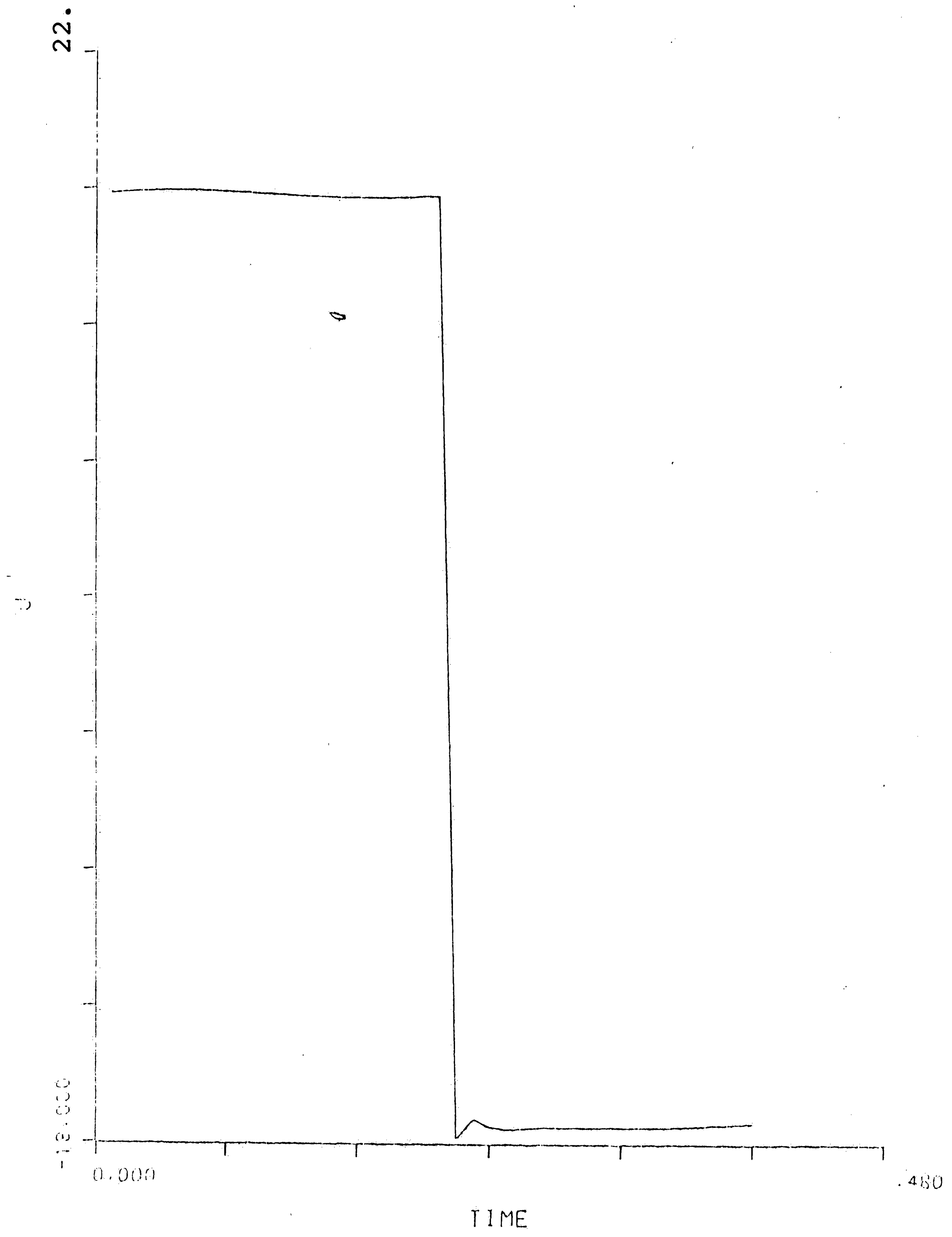


Figure 4.1.1e

Figure 4.1.2 Different initial positions simulation

Here we experiment with the initial position. Namely, we simulate a system which starts from a different initial position than the nominal

Input: bang-bang (nominal $t_s = .21$)

$df=0$.

nominal initial position: $x_{n1}(0) = -.6$, $x_{n2}(0) = 0$.

actual initial position: $x_1(0) = -.5$, $x_2(0) = 0$.

actual $t_s = .21$

sampling interval $h = .01$

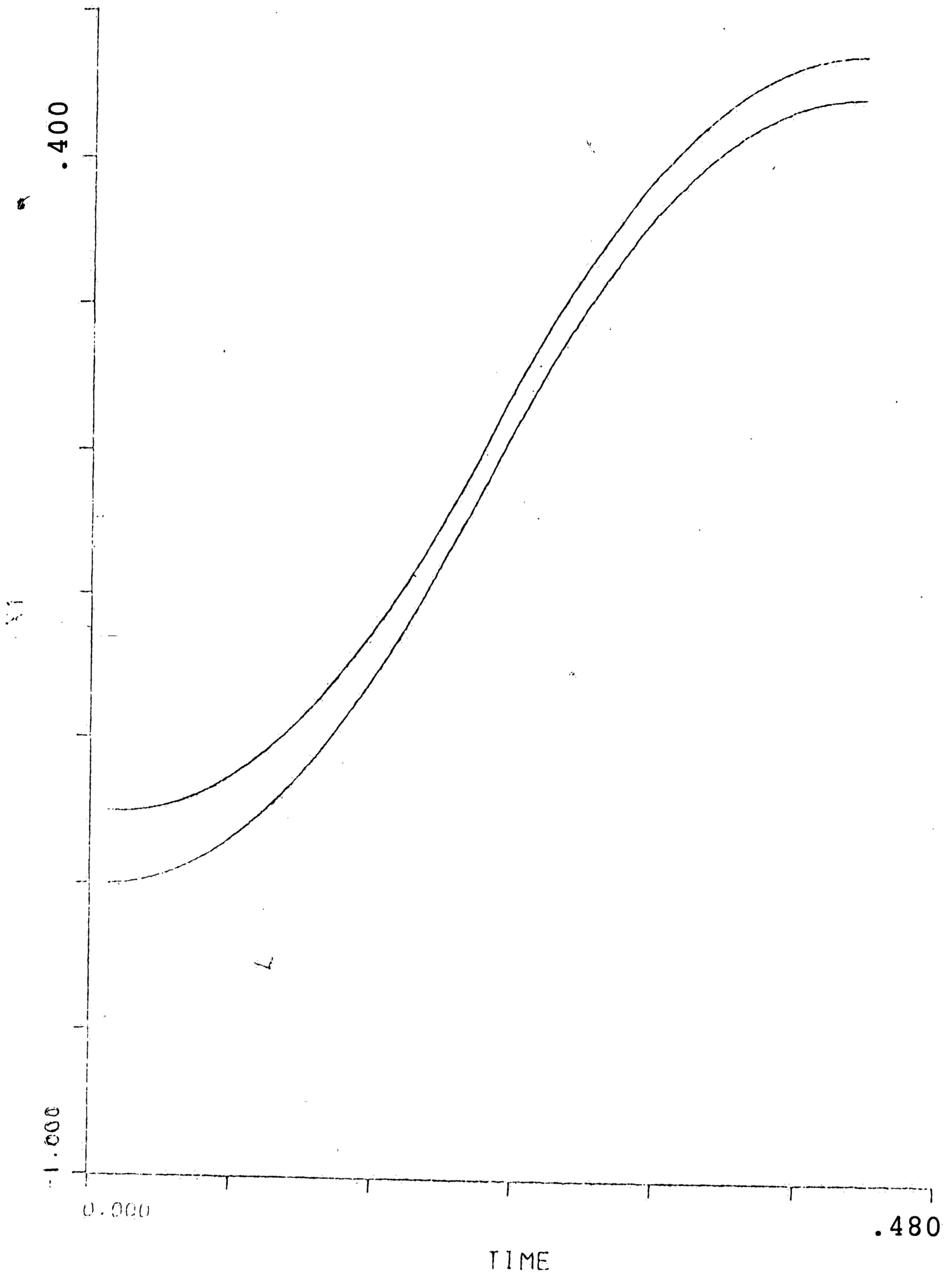


Figure 4.1.2a

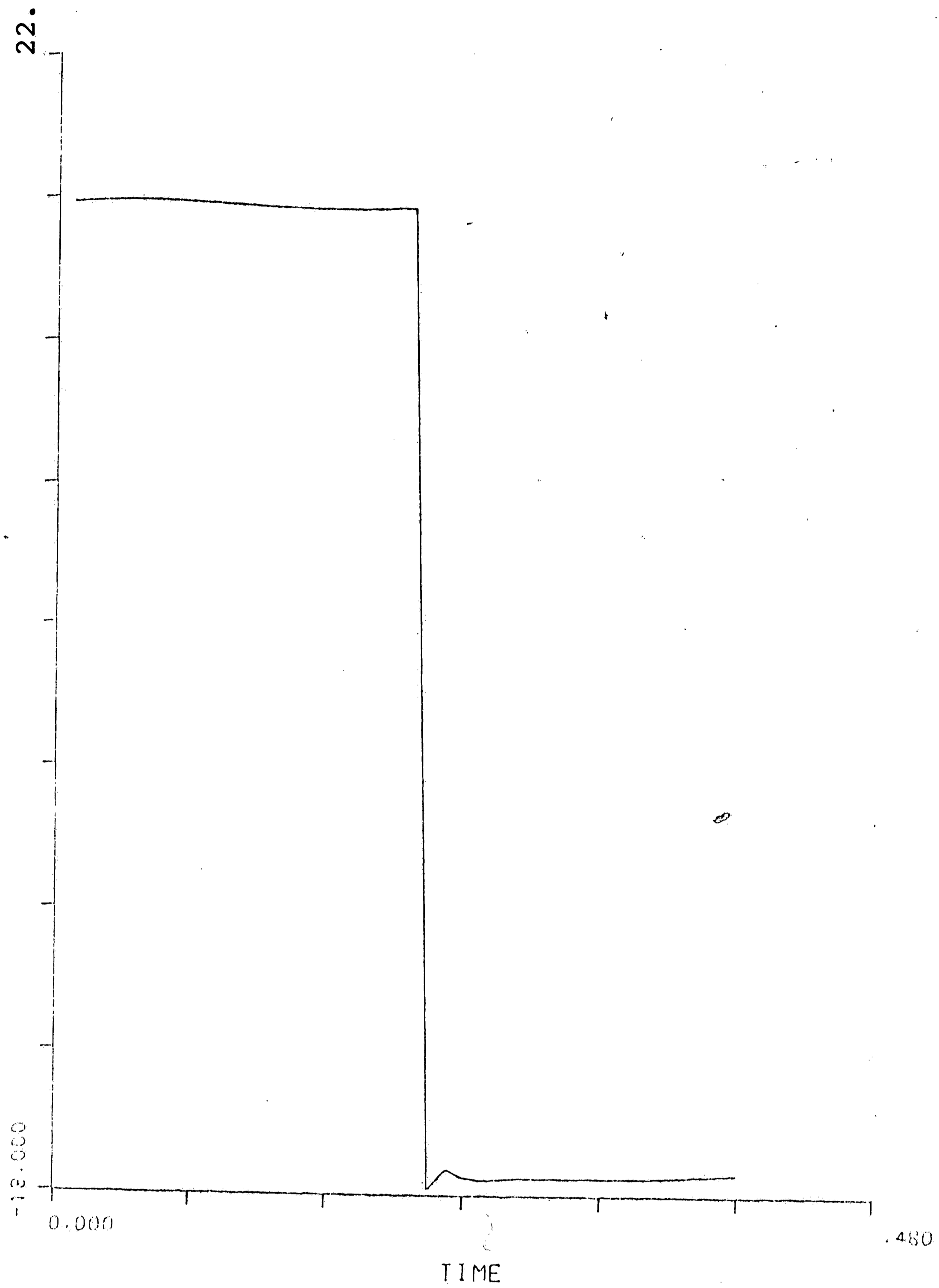


Figure 4.1.1e

Figure 4.1.2 Different initial positions simulation

Here we experiment with the initial position. Namely, we simulate a system which starts from a different initial position than the nominal

Input: bang-bang (nominal $t_s = .21$)

df=0.

nominal initial position: $x_{n1}(0) = -.6$, $x_{n2}(0) = 0$.

actual initial position: $x_1(0) = -.5$, $x_2(0) = 0$.

actual $t_s = .21$

sampling interval $h = .01$

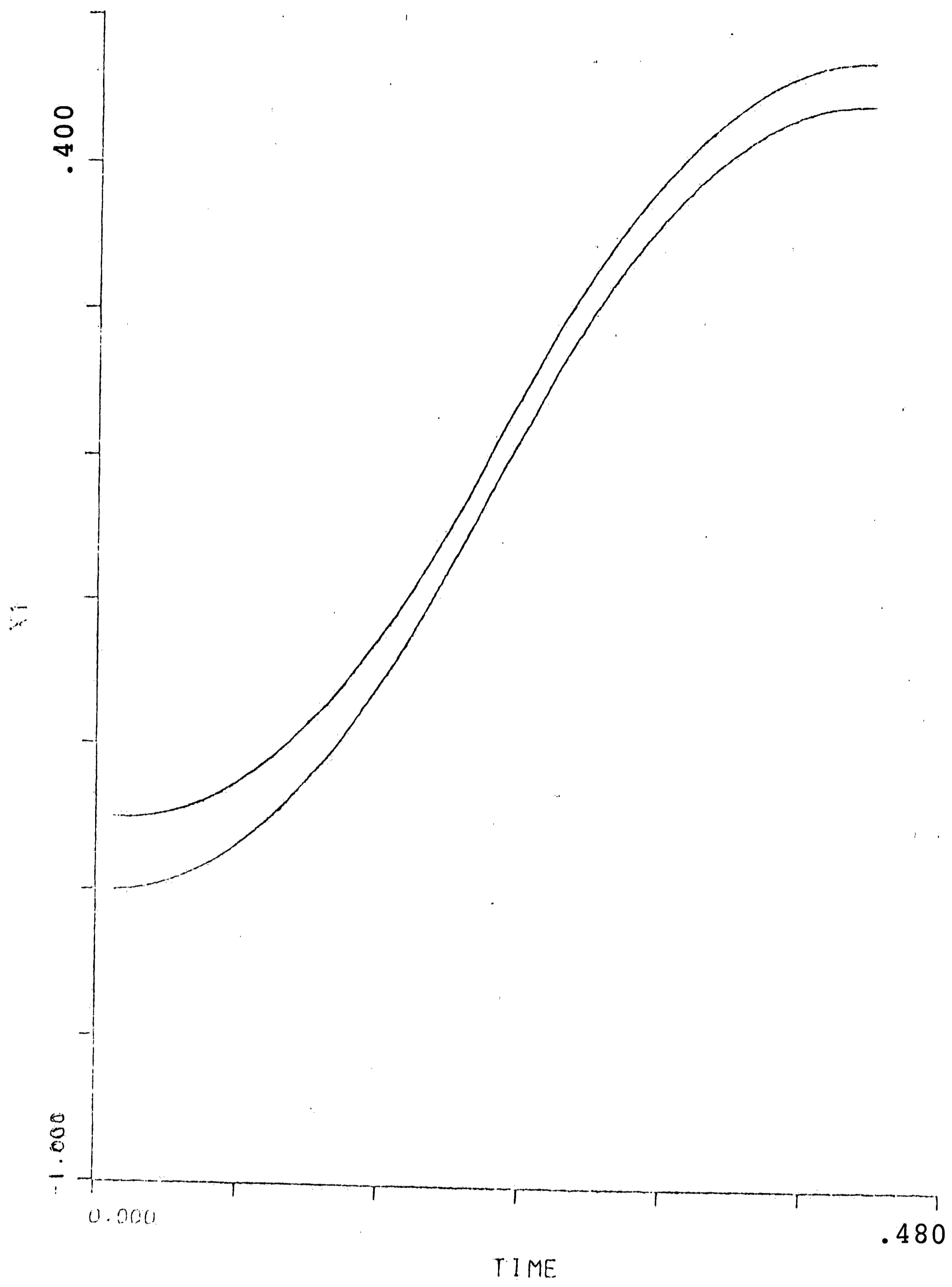


Figure 4.1.2a

DUPLICATE

PAGINATION

PAGE(S) 90

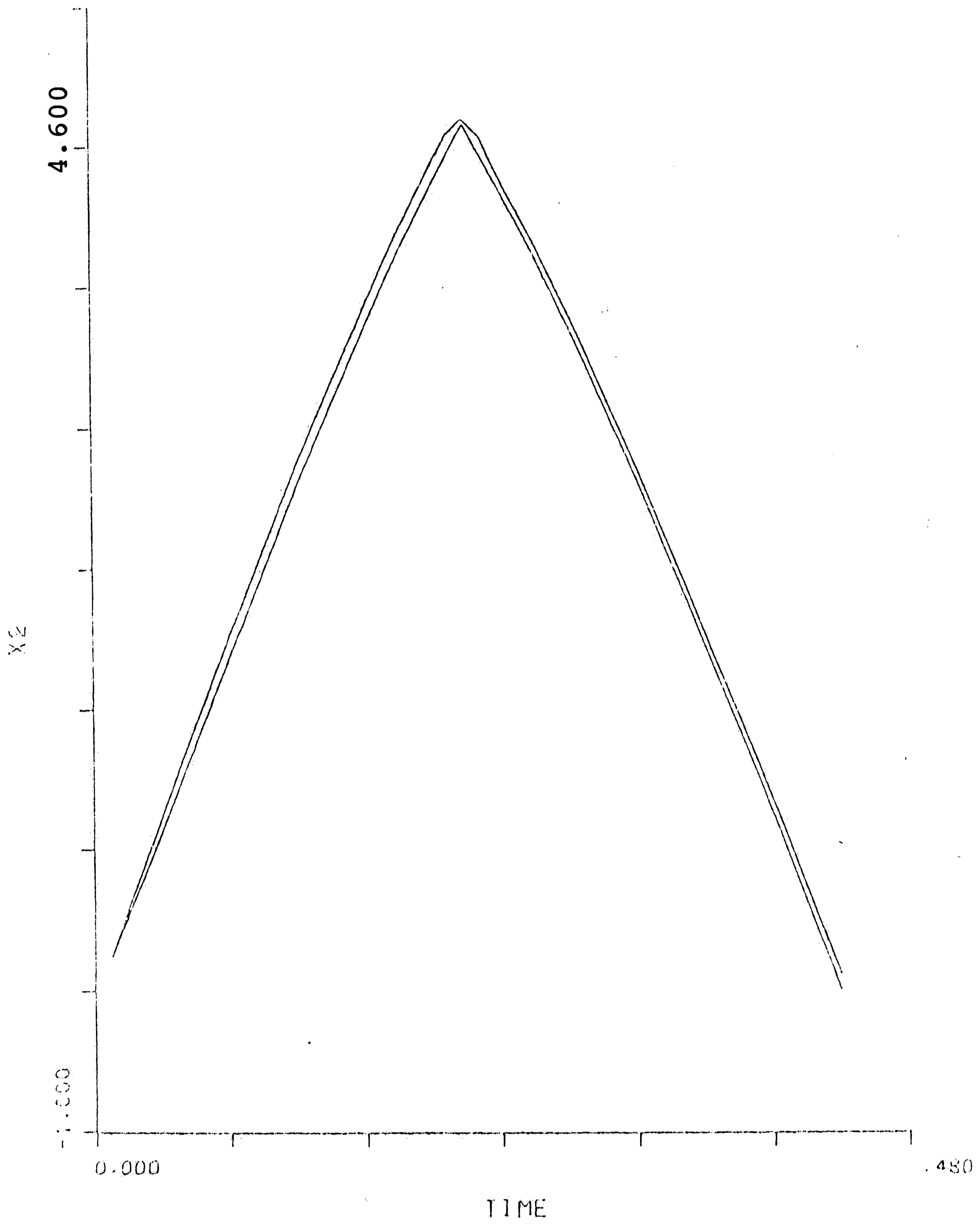


Figure 4.1.2b



Figure 4.1.2c

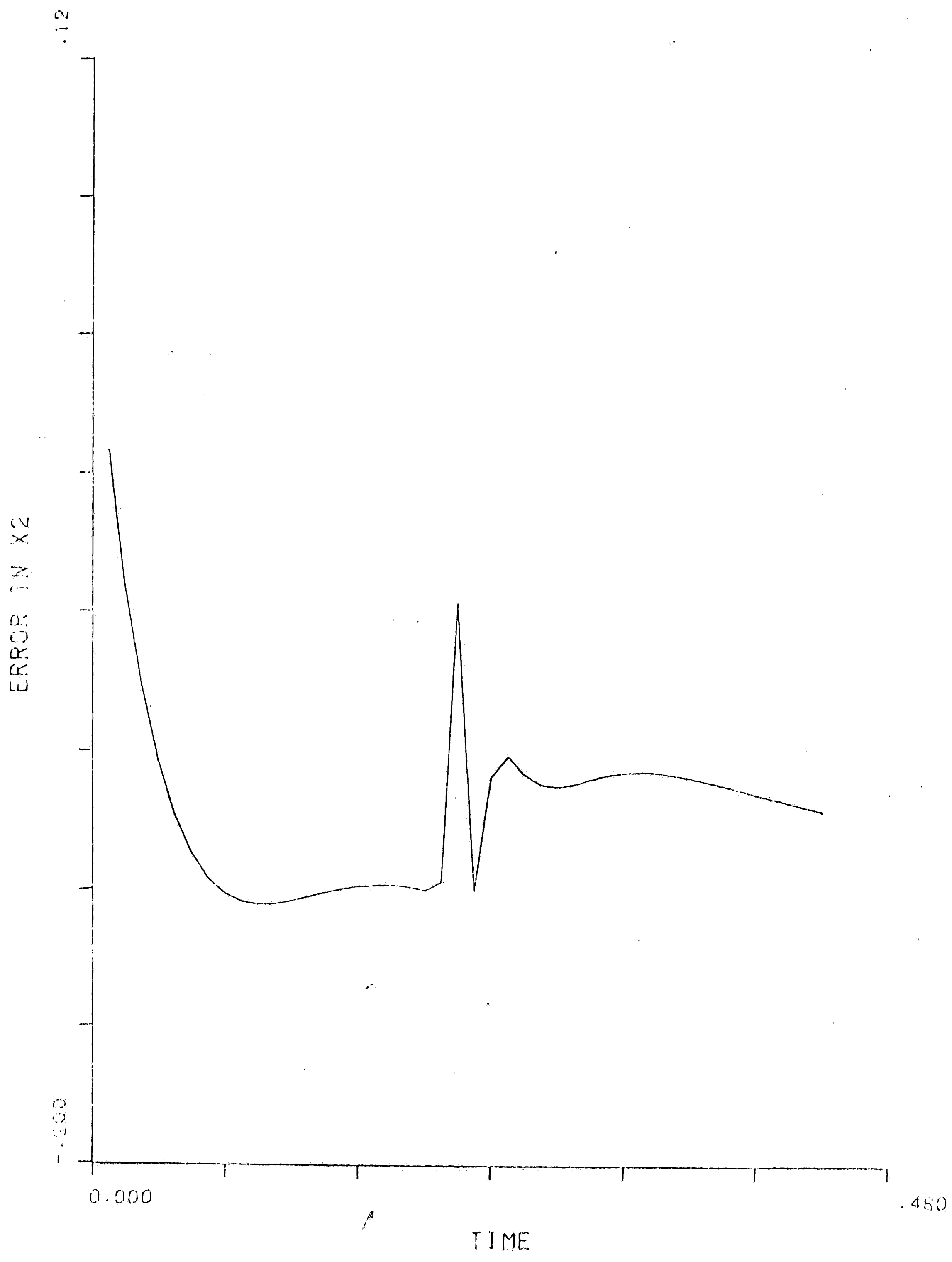


Figure 4.1.2d

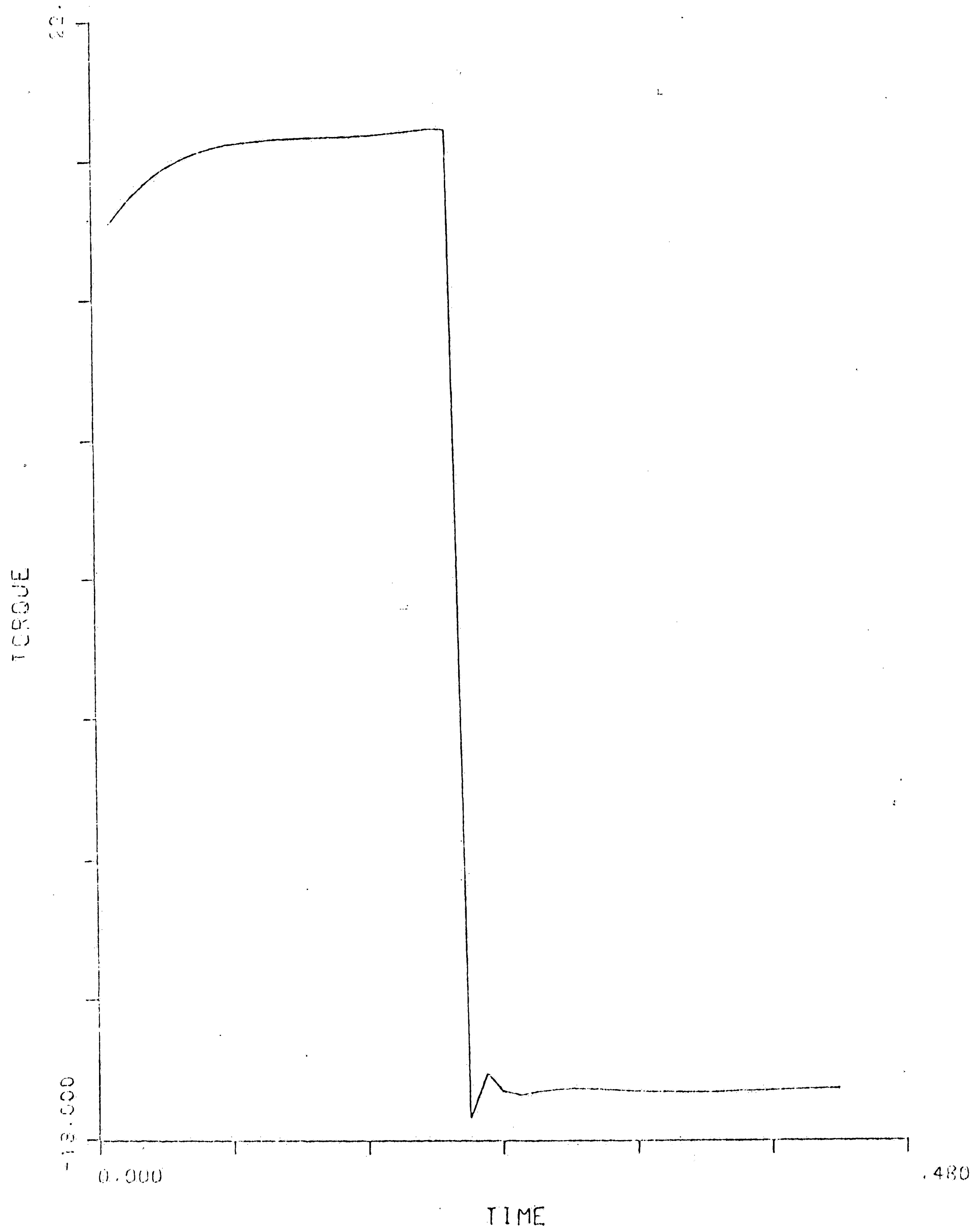


Figure 4.1.2e

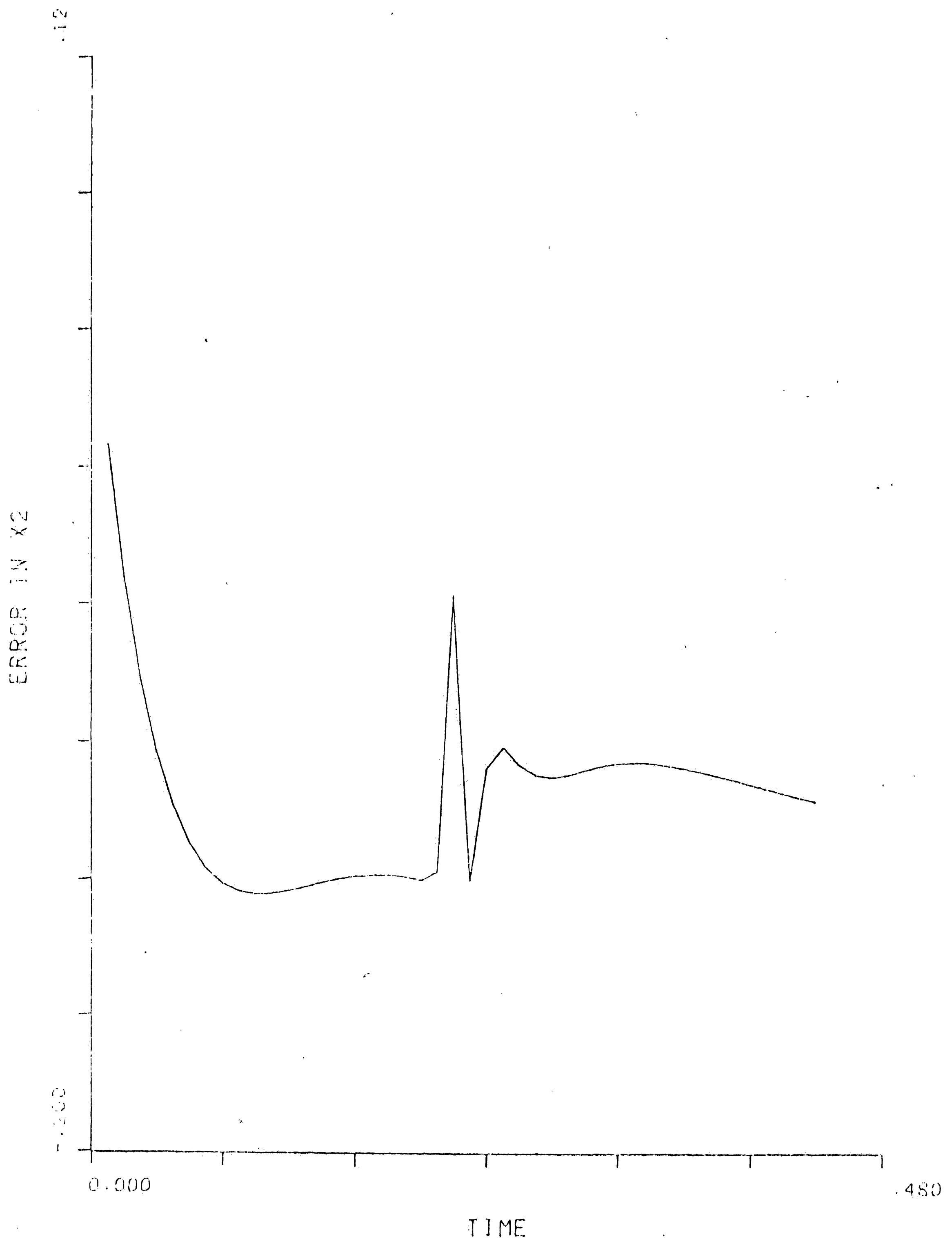


Figure 4.1.2d

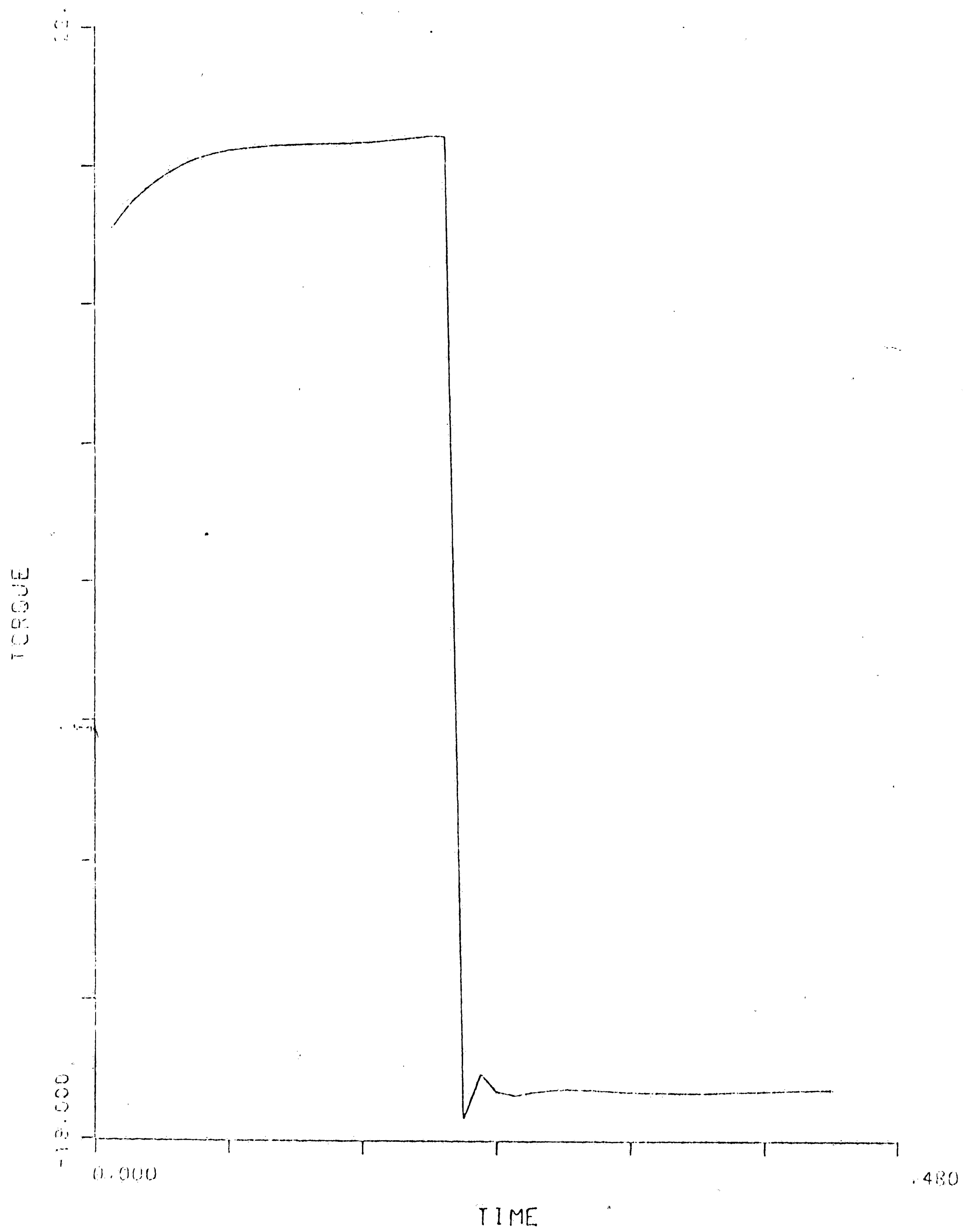


Figure 4.1.2e

Figure 4.1.3 Different switching time simulation

Here we experiment with the switching time.

Input: bang-bang (nominal $t_s = .21$)

$df=0$.

nominal initial position: $x_{n1}(0) = -.6$, $x_{n2}(0) = 0$.

actual initial position: $x_1(0) = -.6$, $x_2(0) = 0$.

actual $t_s = .21$

sampling interval $h = .01$

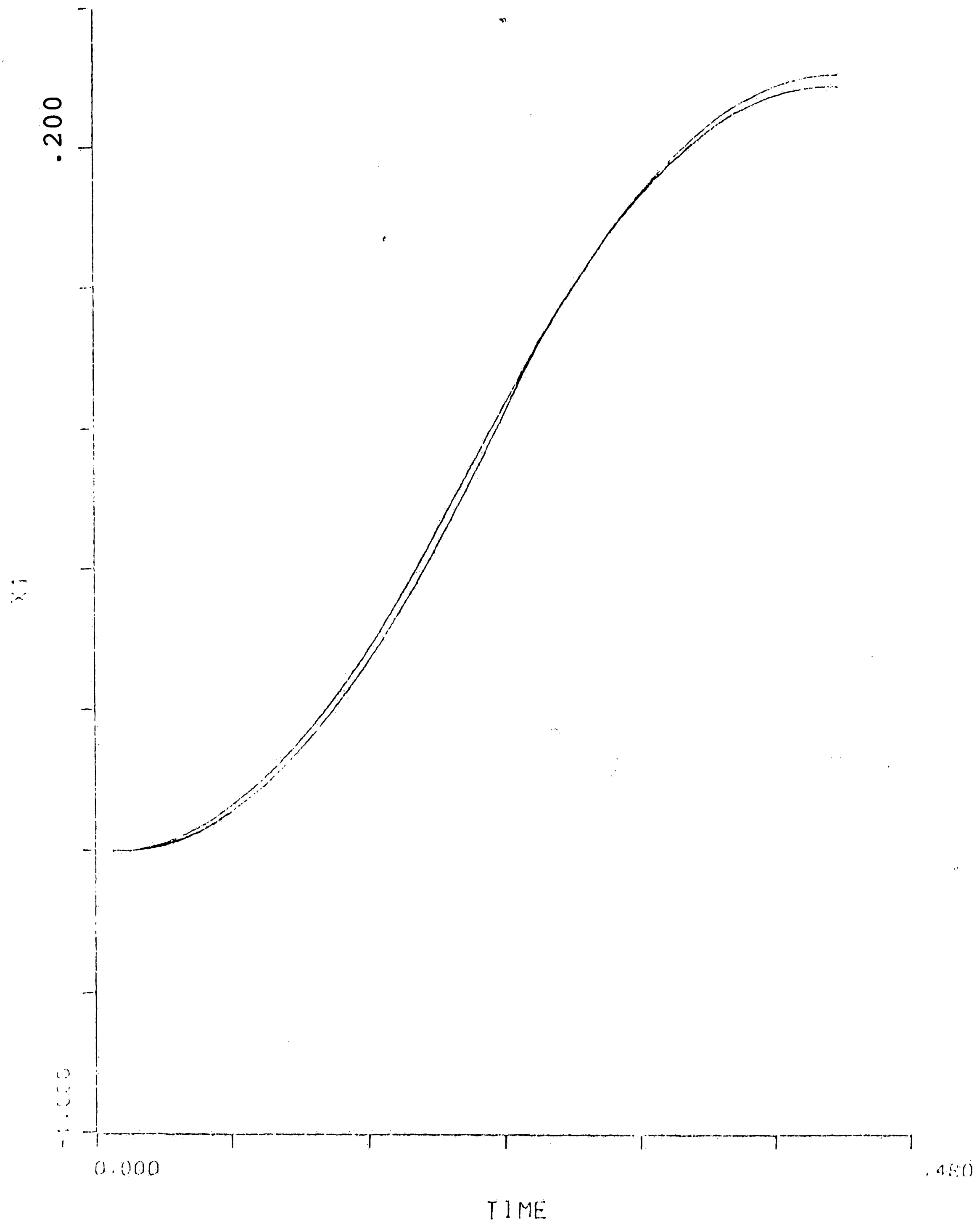


Figure 4.1.3a

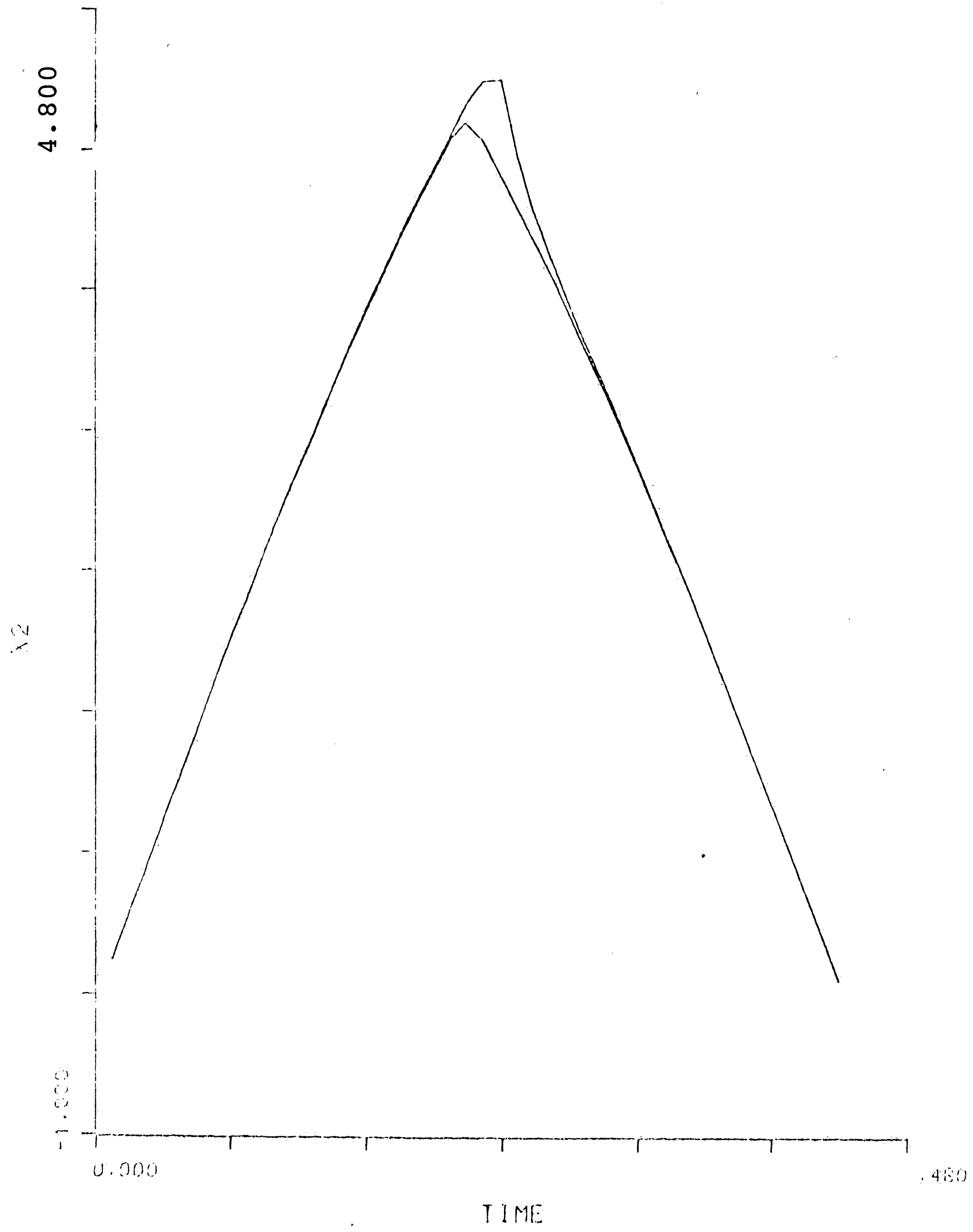


Figure 4.1.3b

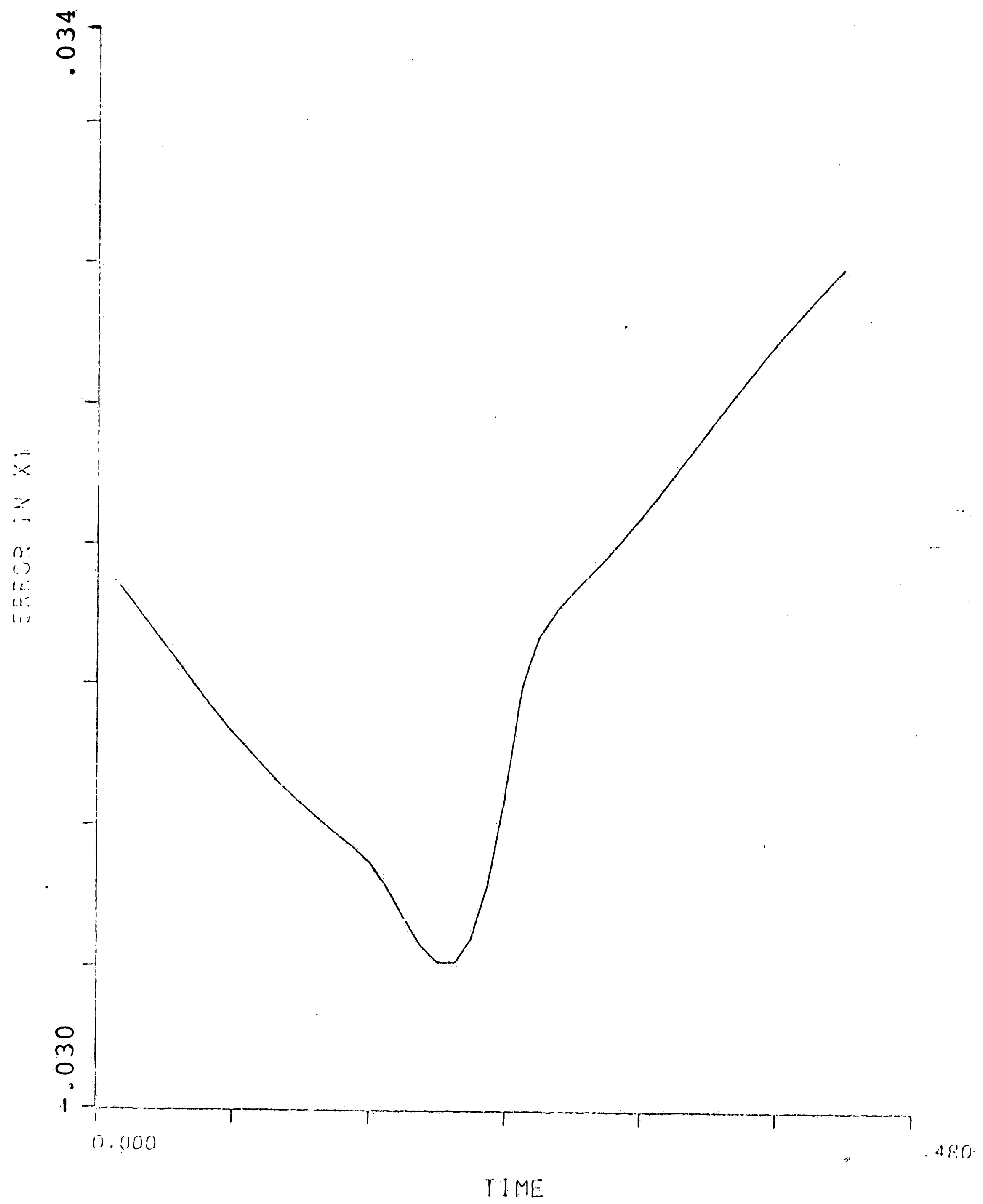


Figure 4.1.3c

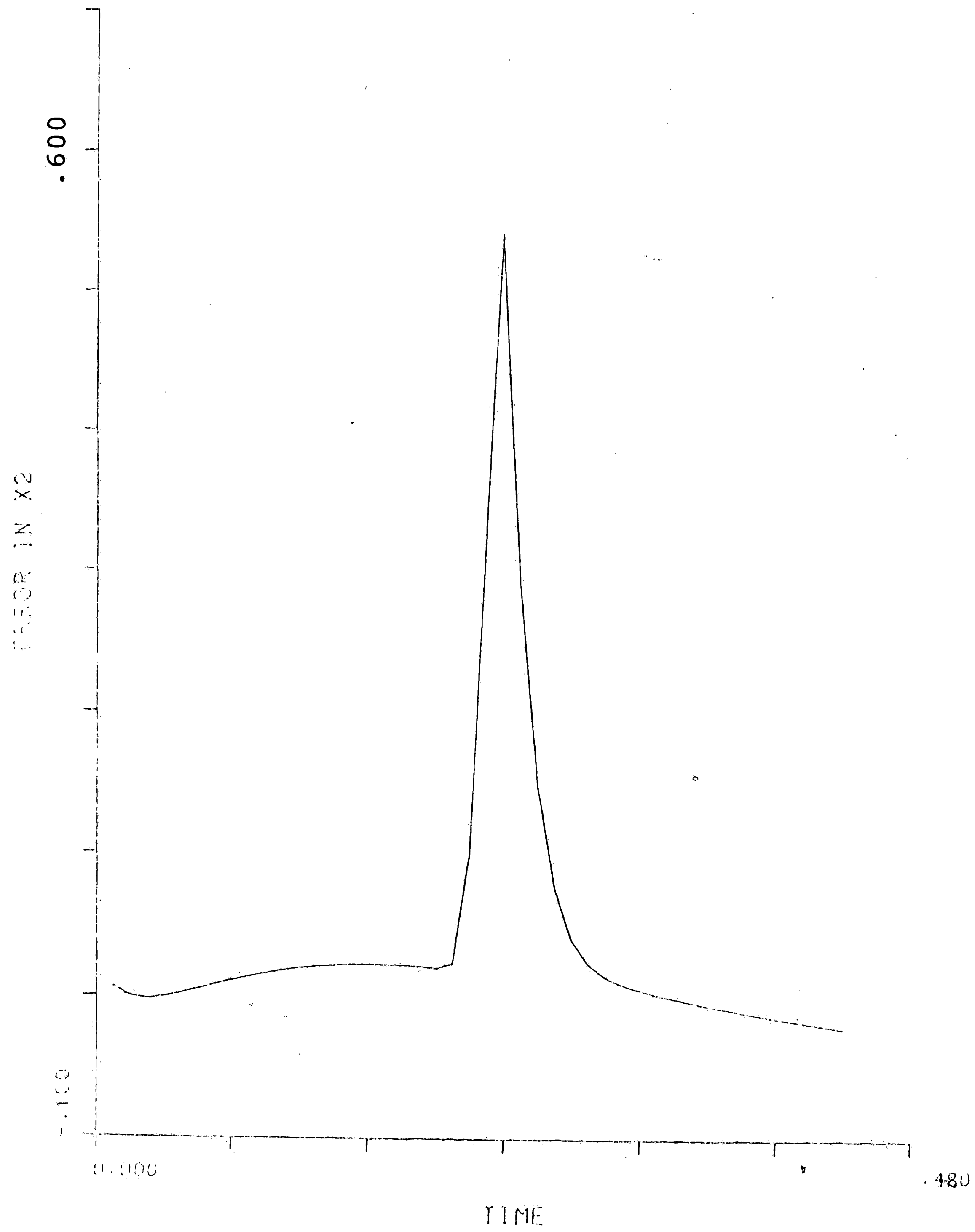


Figure 4.1.3d

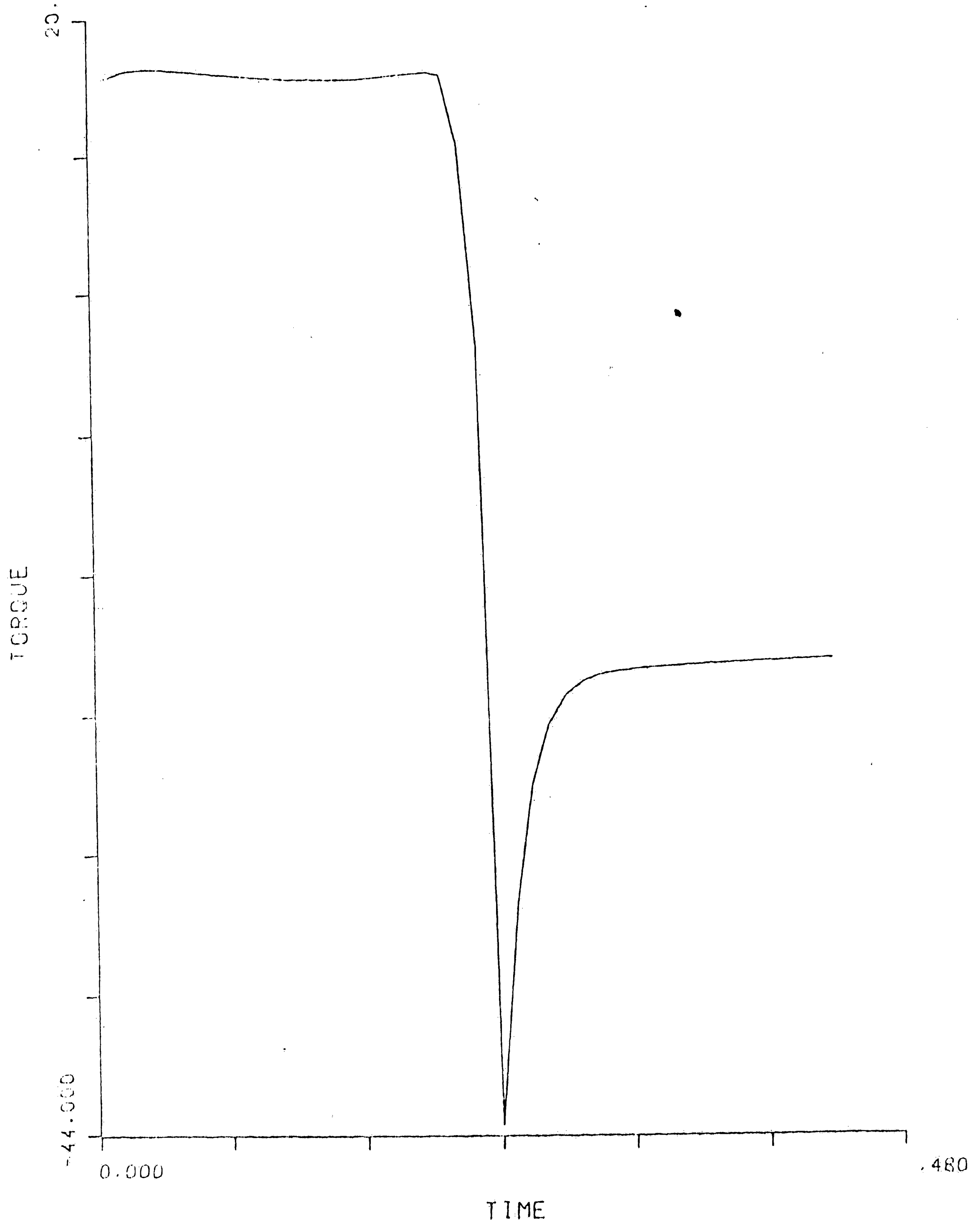


Figure 4.1.3e

Figure 4.1.4 Difference in modelling function simulation

Here we take df to be nonzero.

Input: bang-bang (nominal $t_s = .21$)

$df = -3$.

nominal initial position: $x_{n1}(0) = -.6$, $x_{n2}(0) = 0$.

actual initial position: $x_1(0) = -.6$, $x_2(0) = 0$.

actual $t_s = .21$

sampling interval $h = .01$

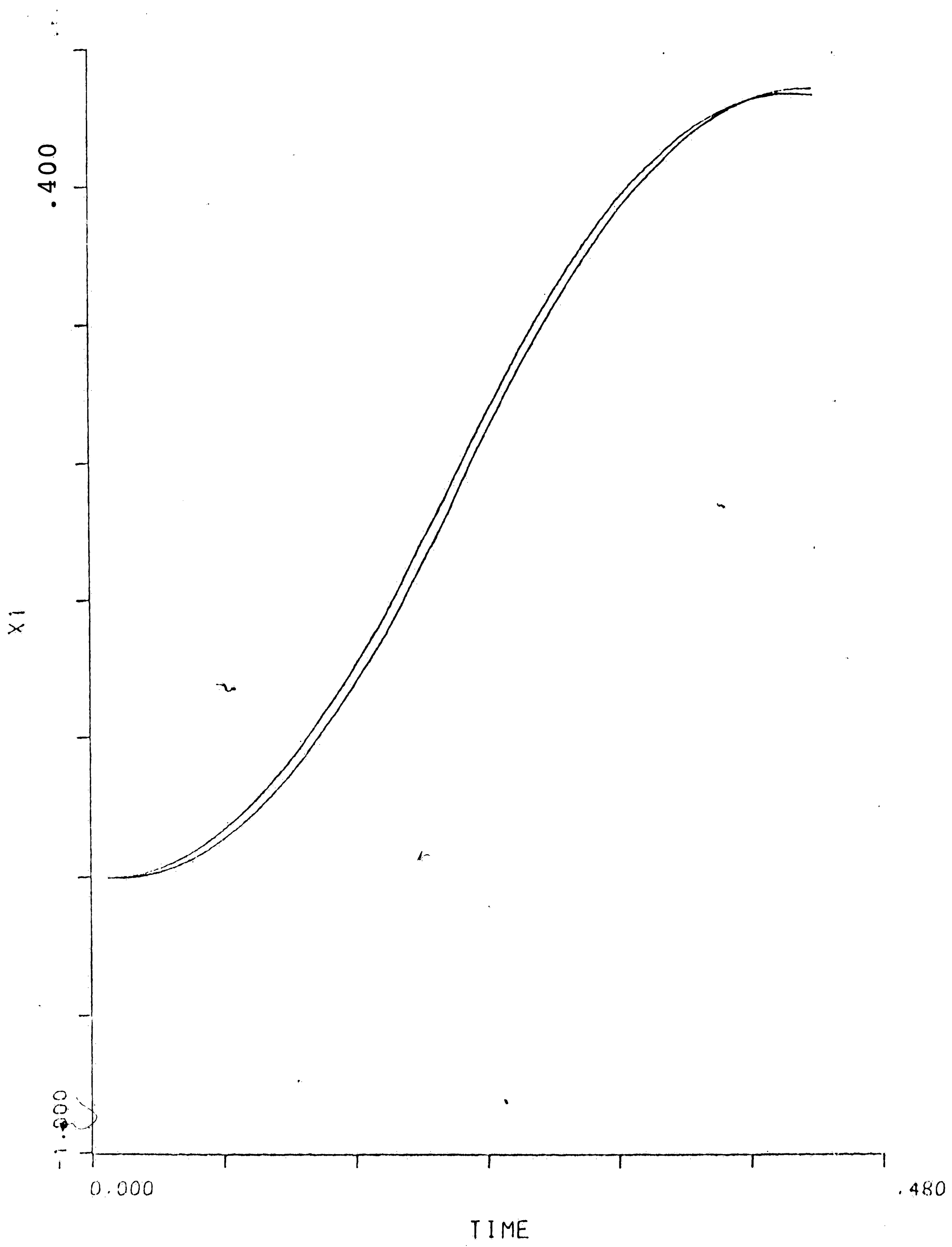


Figure 4.1.4a

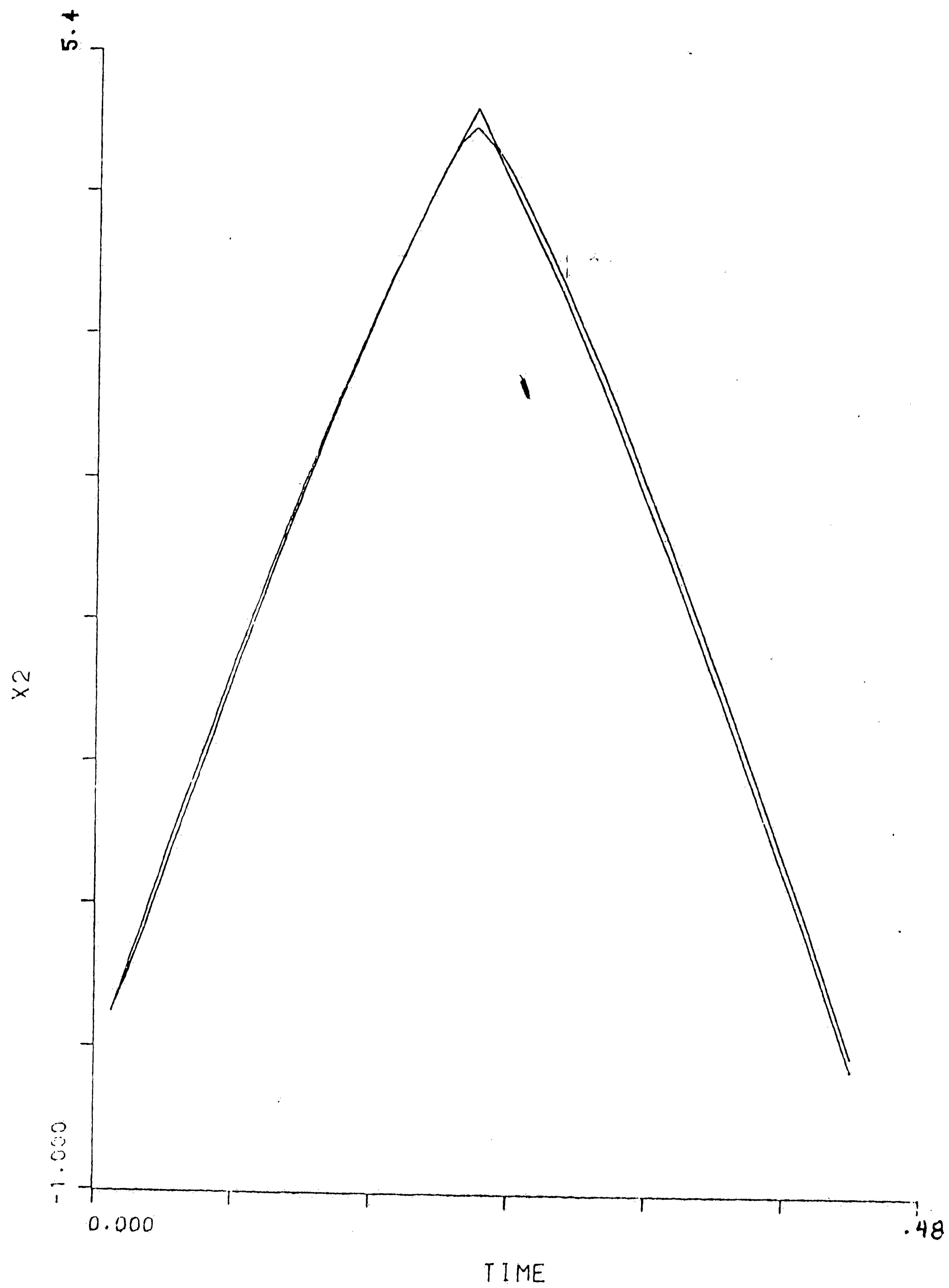


Figure 4.1.4b

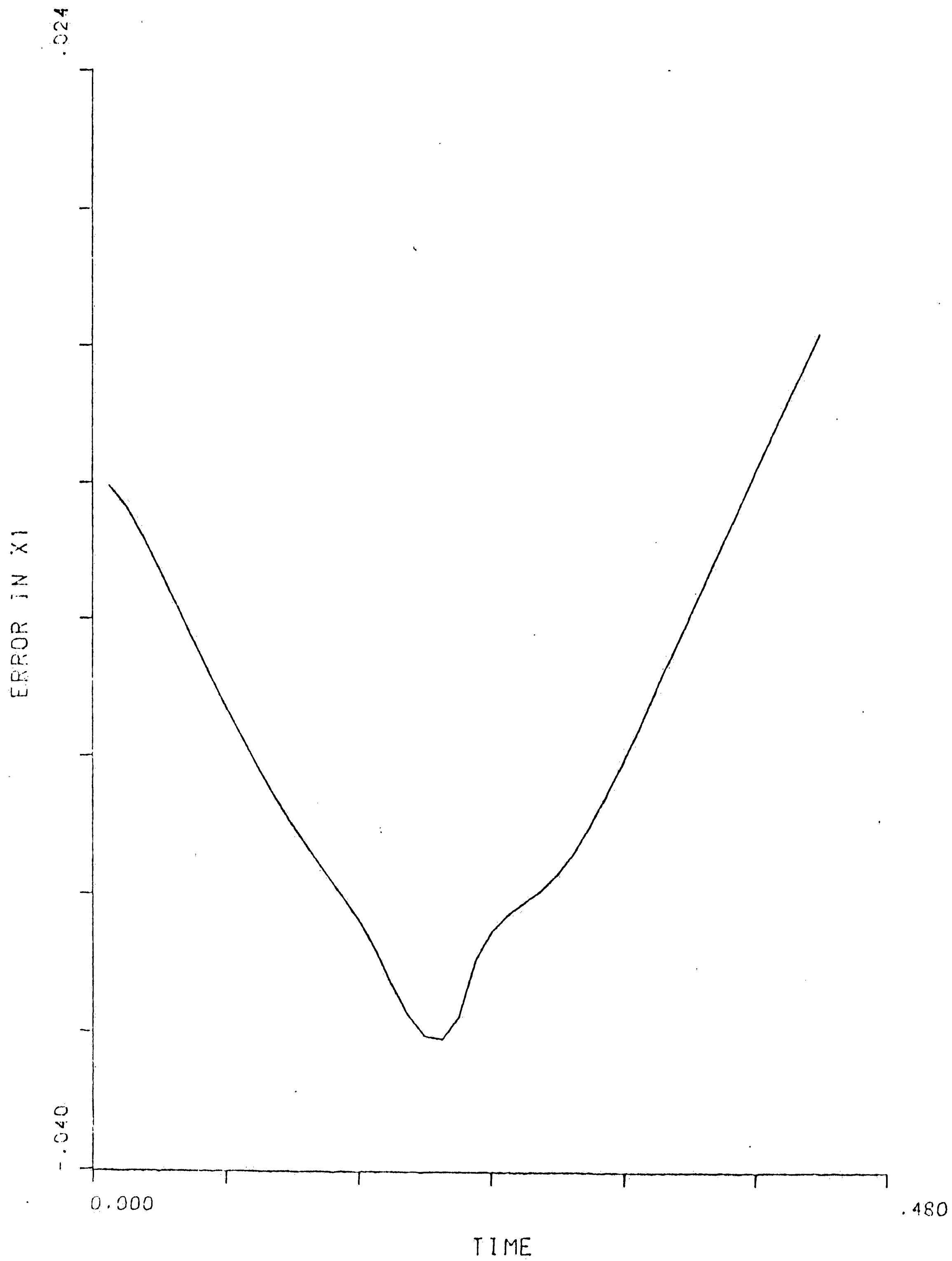


Figure 4.1.4c

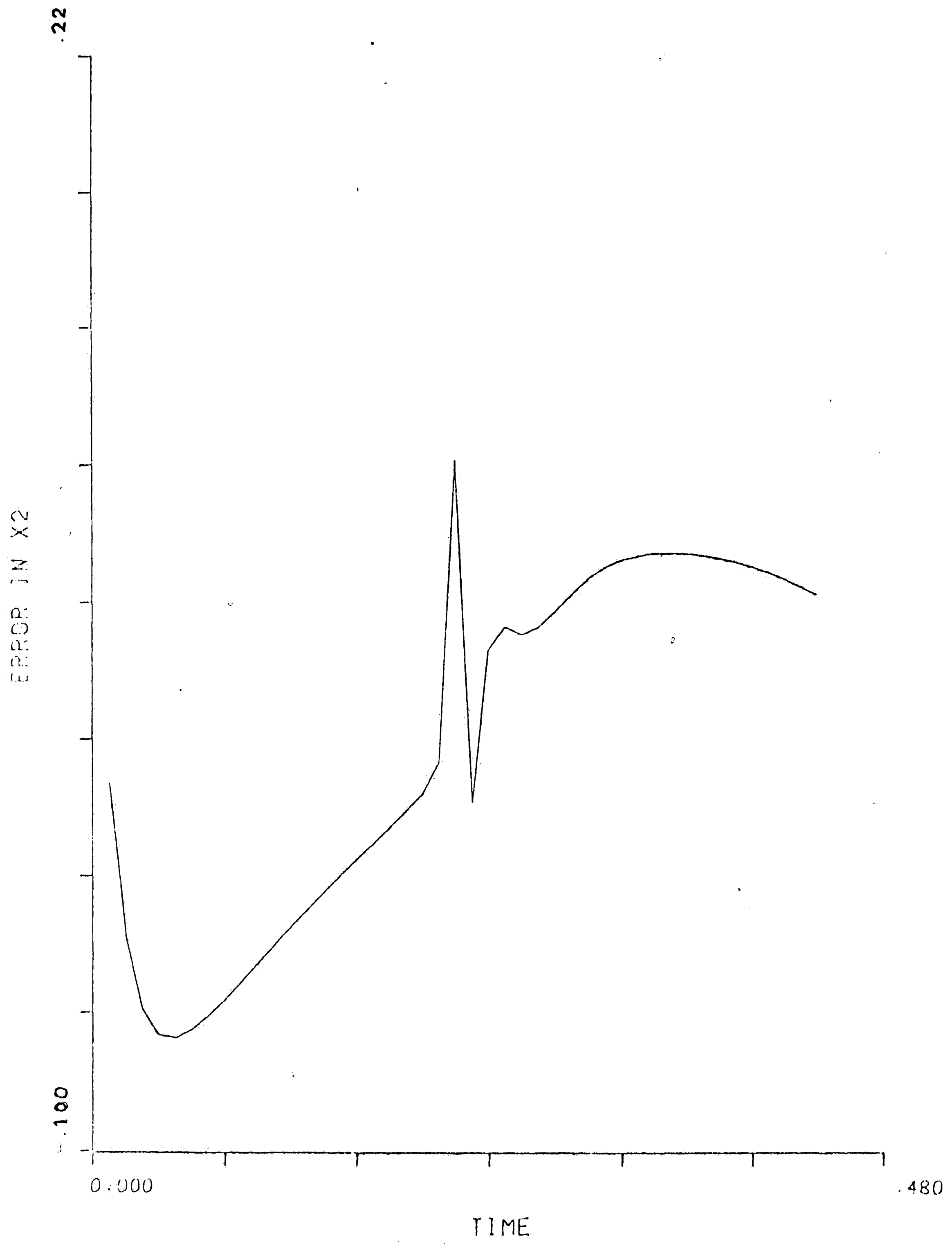


Figure 4.1.4d



Figure 4.1.4e

Figure 4.1.5 "Long term" tracking simulation

Here we use constant preprogrammed control and see how well the system tracks nominal trajectories over long times.

Input: Constant

$df=0.$

nominal initial position: $x_{n1}(0)=-.6$, $x_{n2}(0)=0.$

actual initial position: $x_1(0)=-.5$, $x_2(0)=0.$

sampling interval $h=.01$

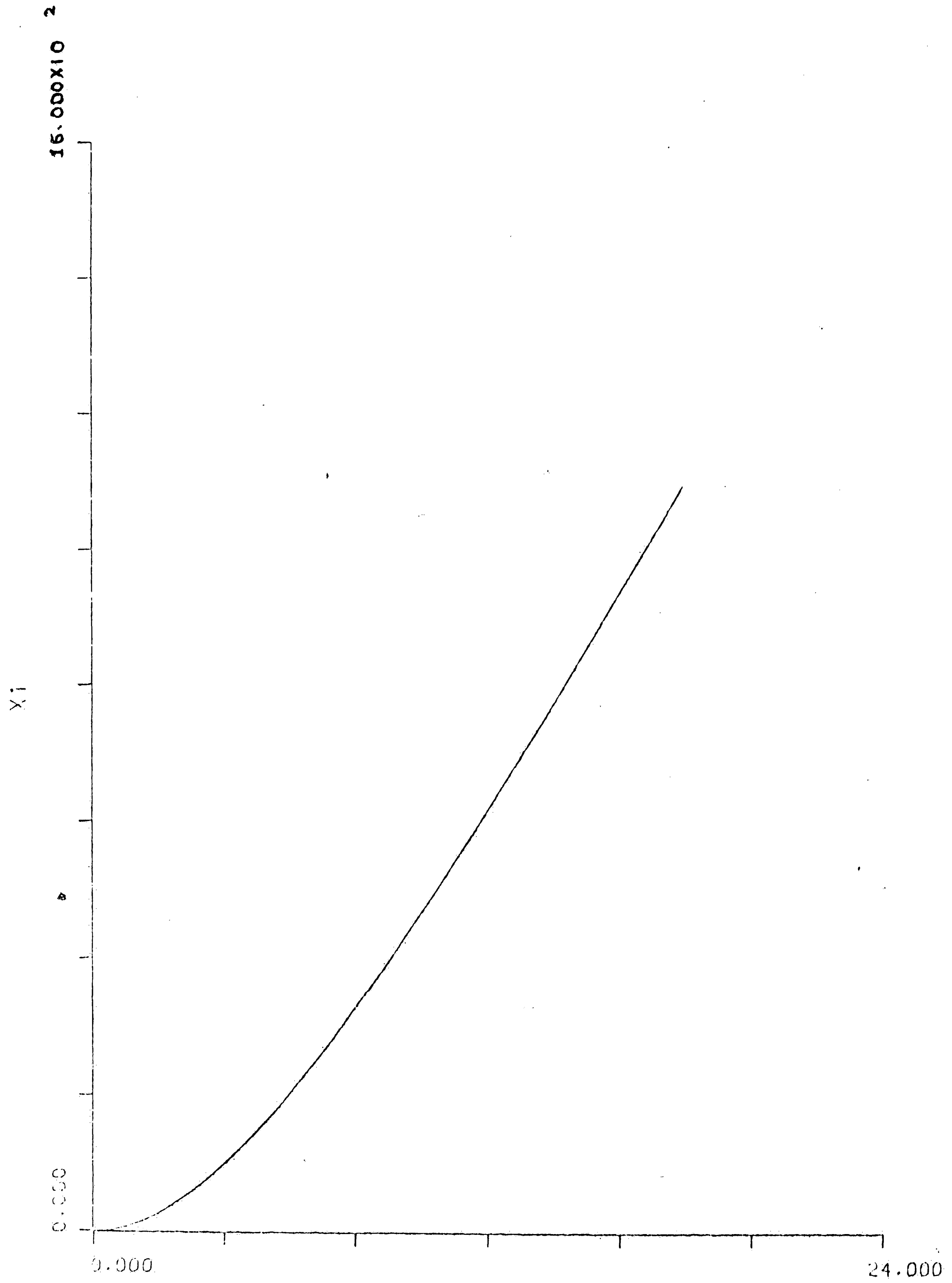


Figure 4.1.5a

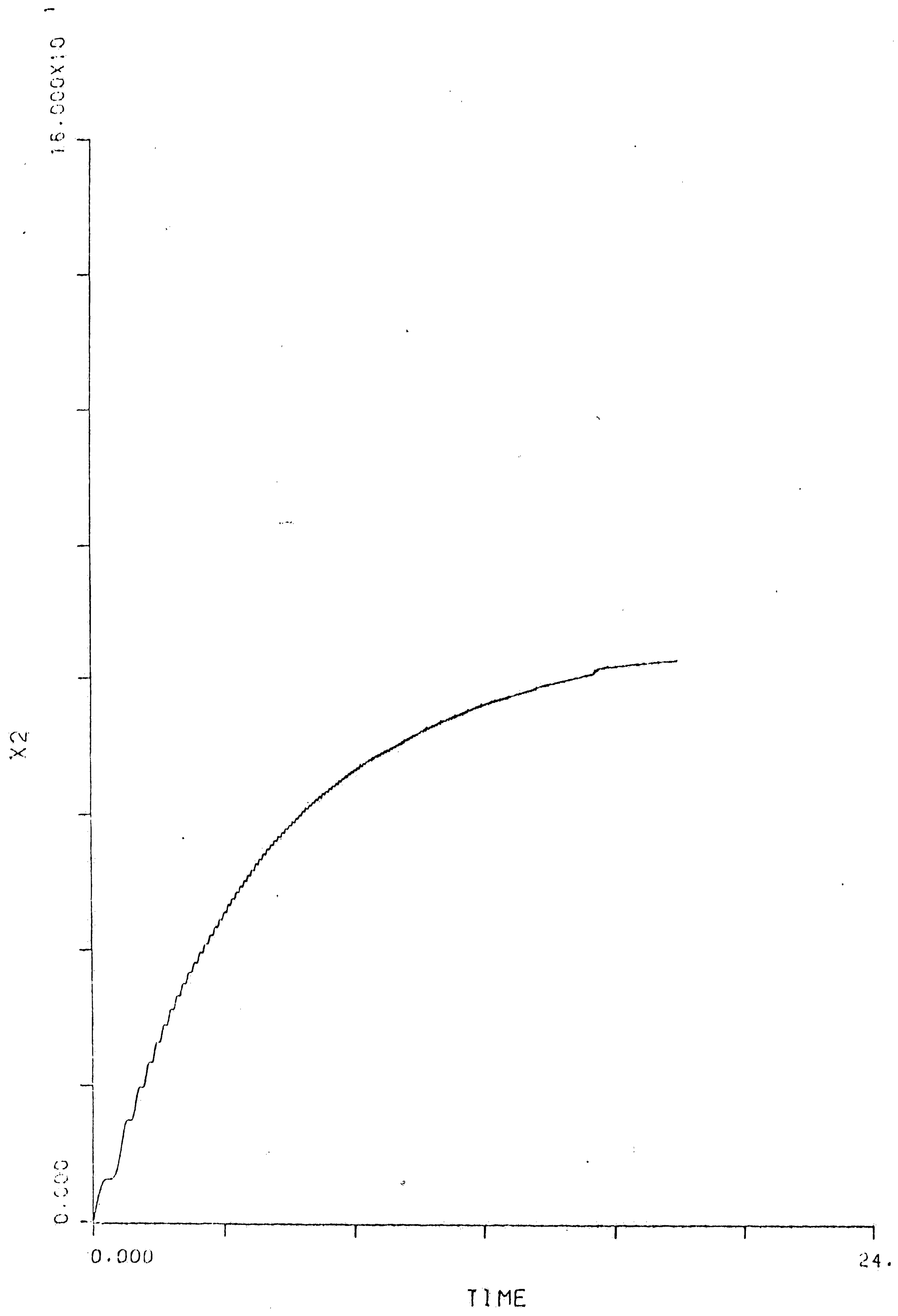


Figure 4.1.5b

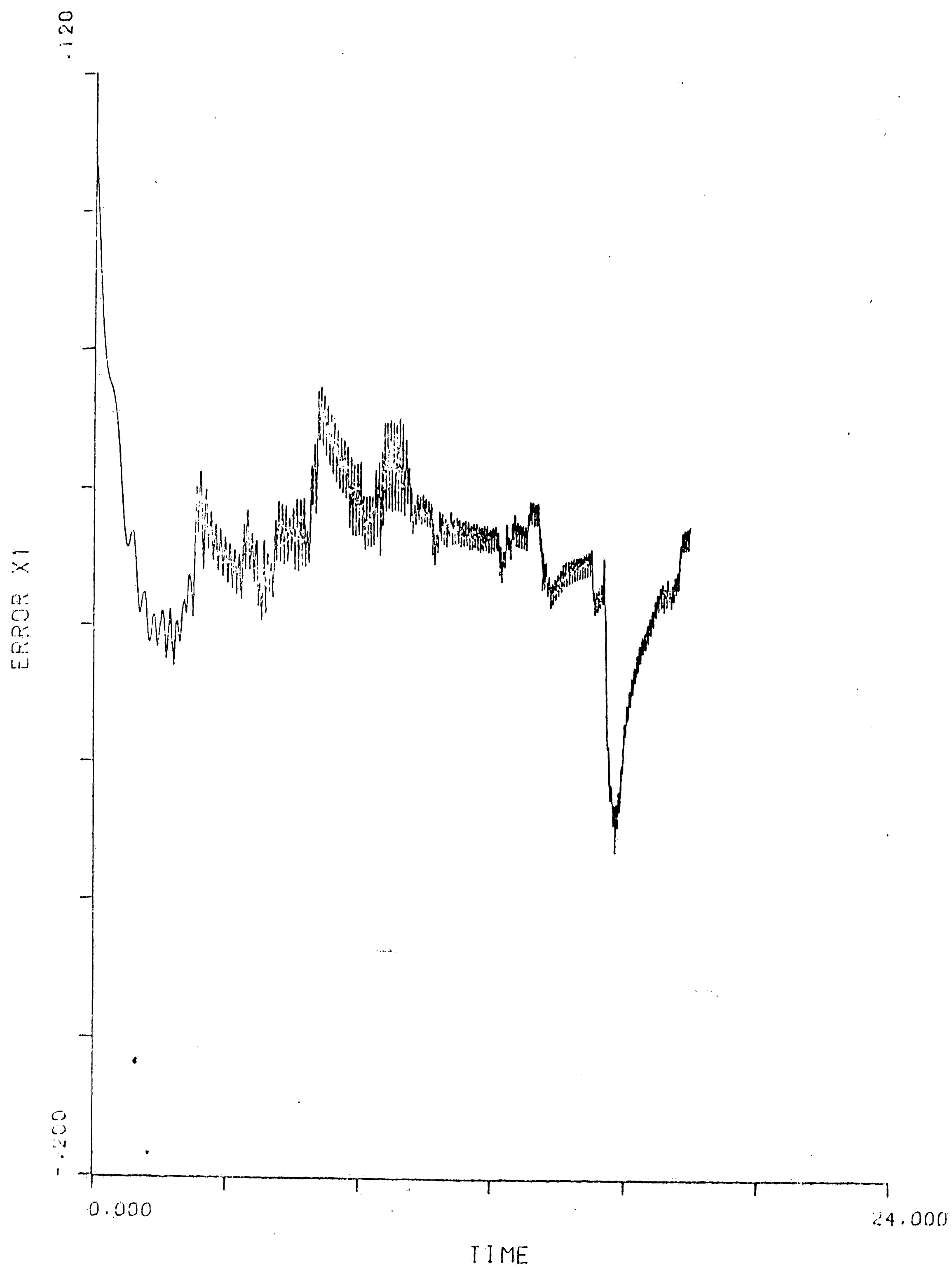


Figure 4.1.5c

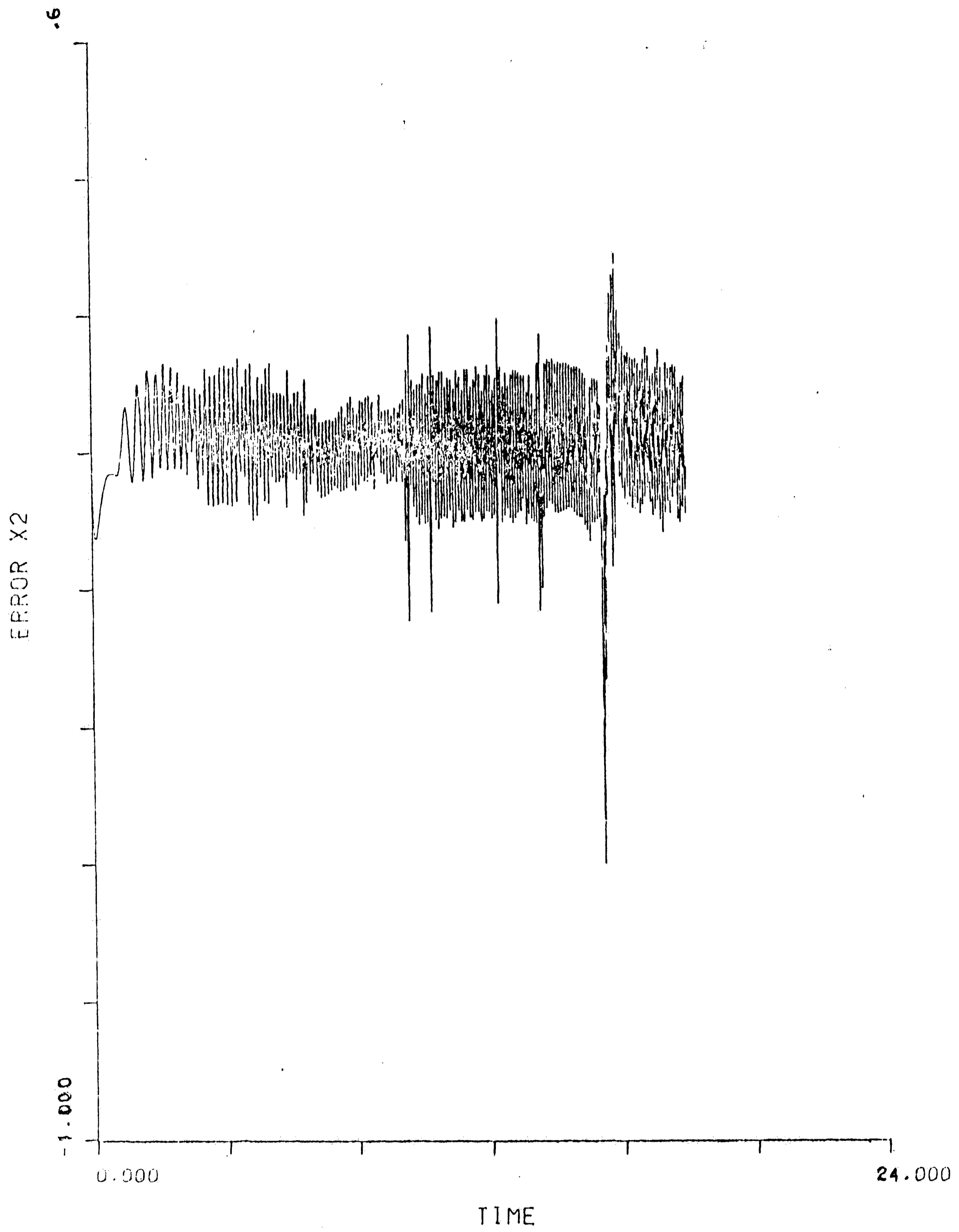


Figure 4.1.5d

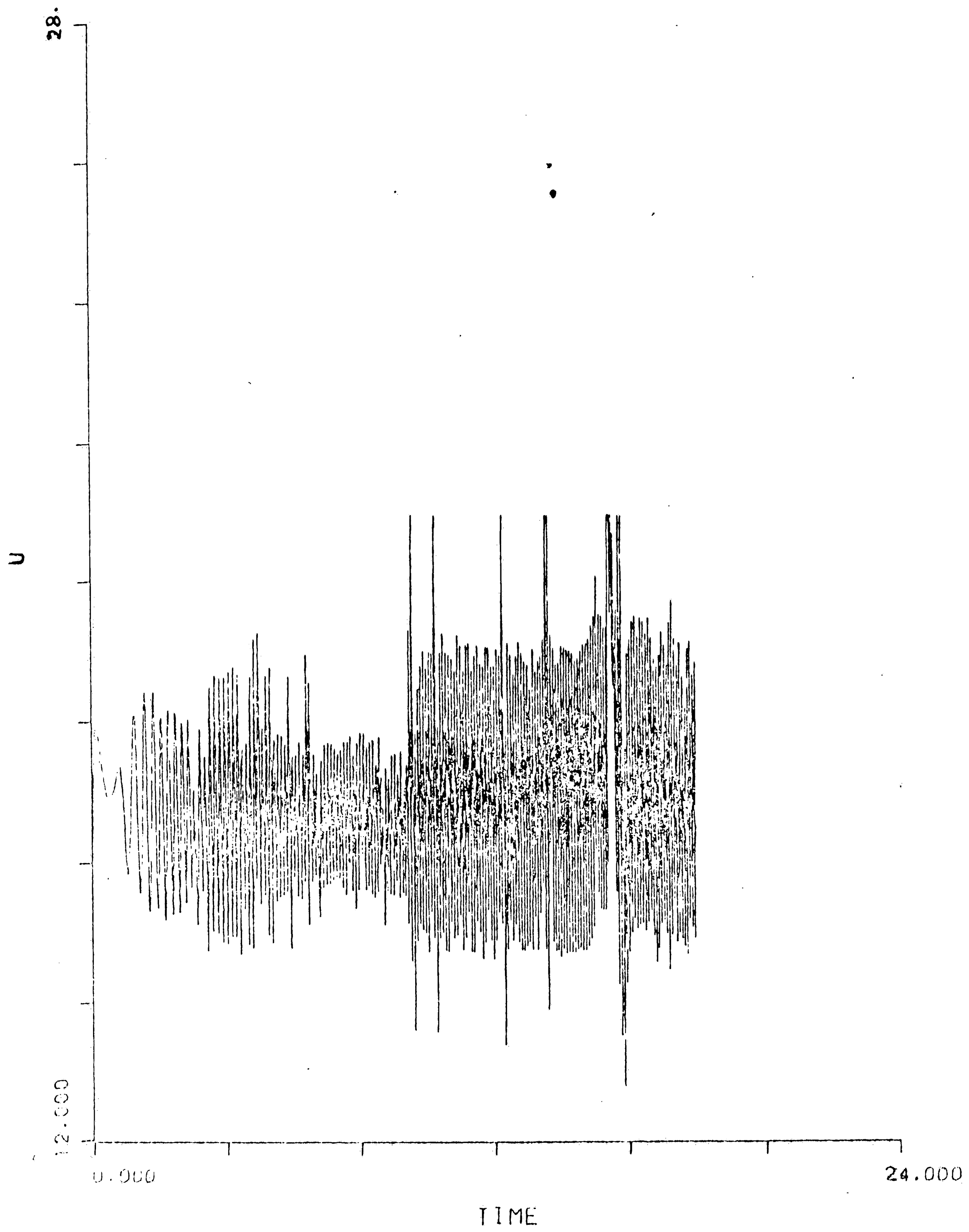


Figure 4.1.5e

Figure 4.1.6 Square pulse input simulation

Here the prerogrammed control is a square pulse train.

We use this type of input to test the system in quick changes.

Input: Square pulse train.

$df=0$.

nominal initial position: $x_{n1}(0)=-.6$, $x_{n2}(0)=0$.

actual initial position: $x_1(0)=-.5$, $x_2(0)=0$.

sampling interval $h=.01$

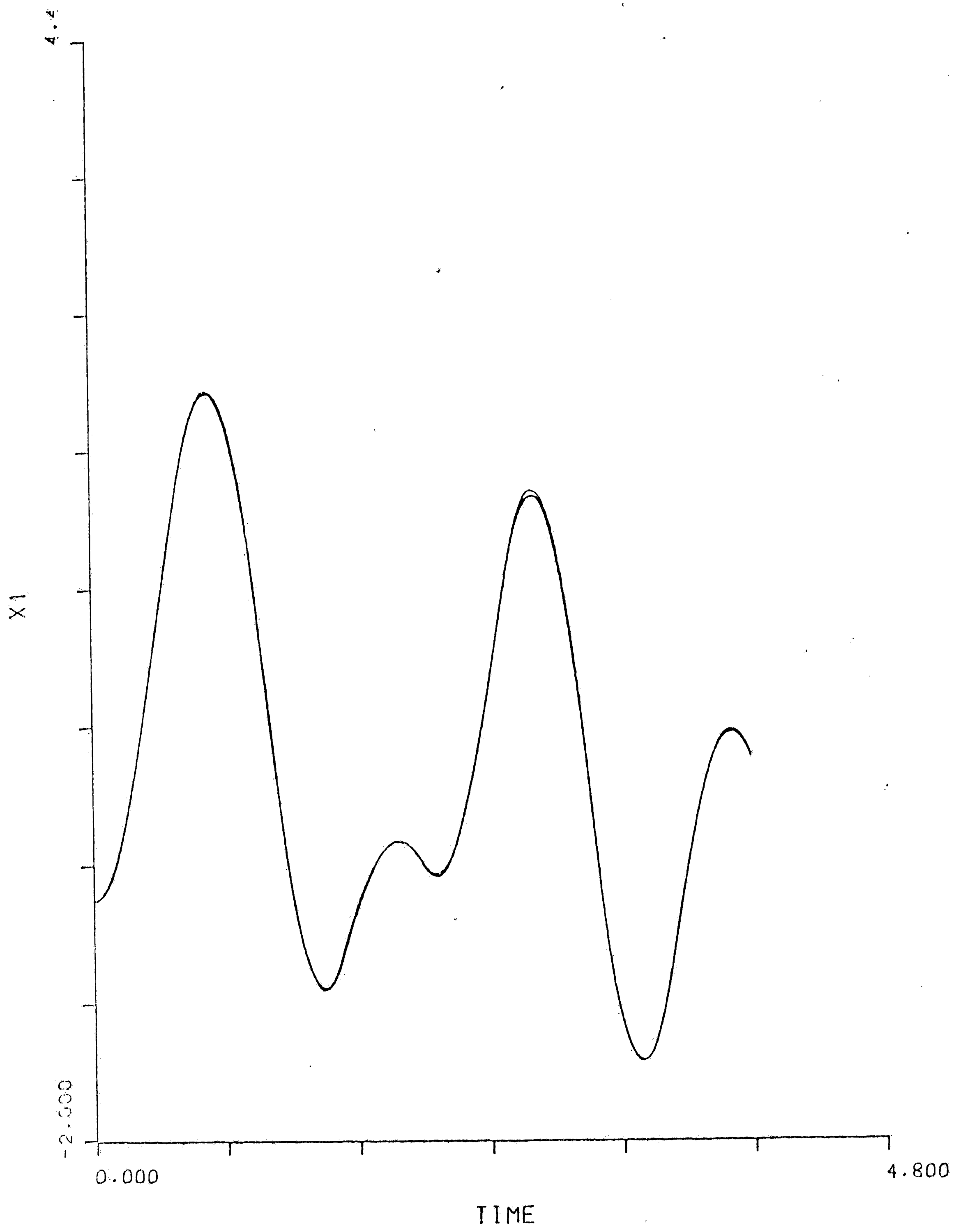


Figure 4.1.6a

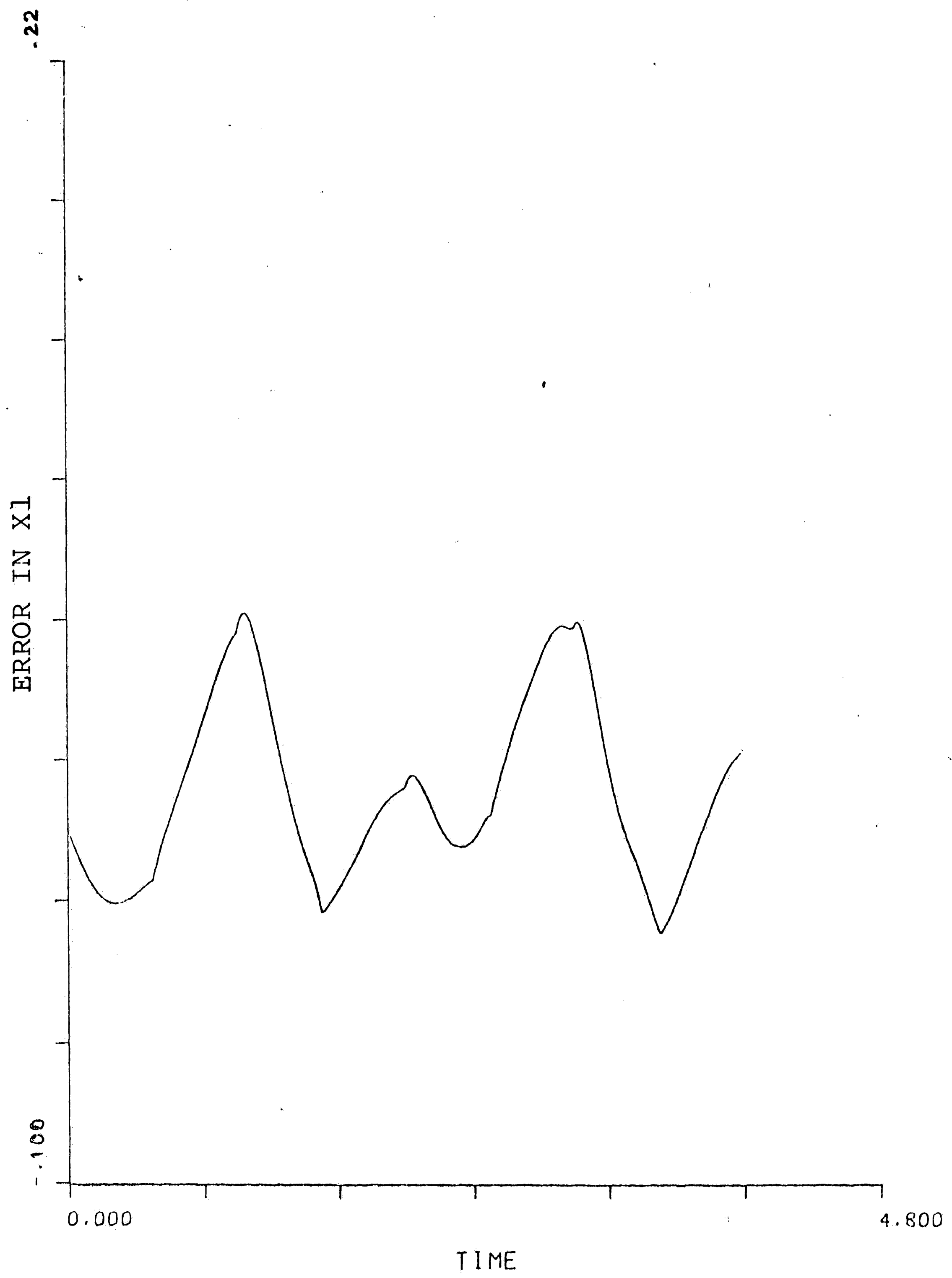


Figure 4.1.6b

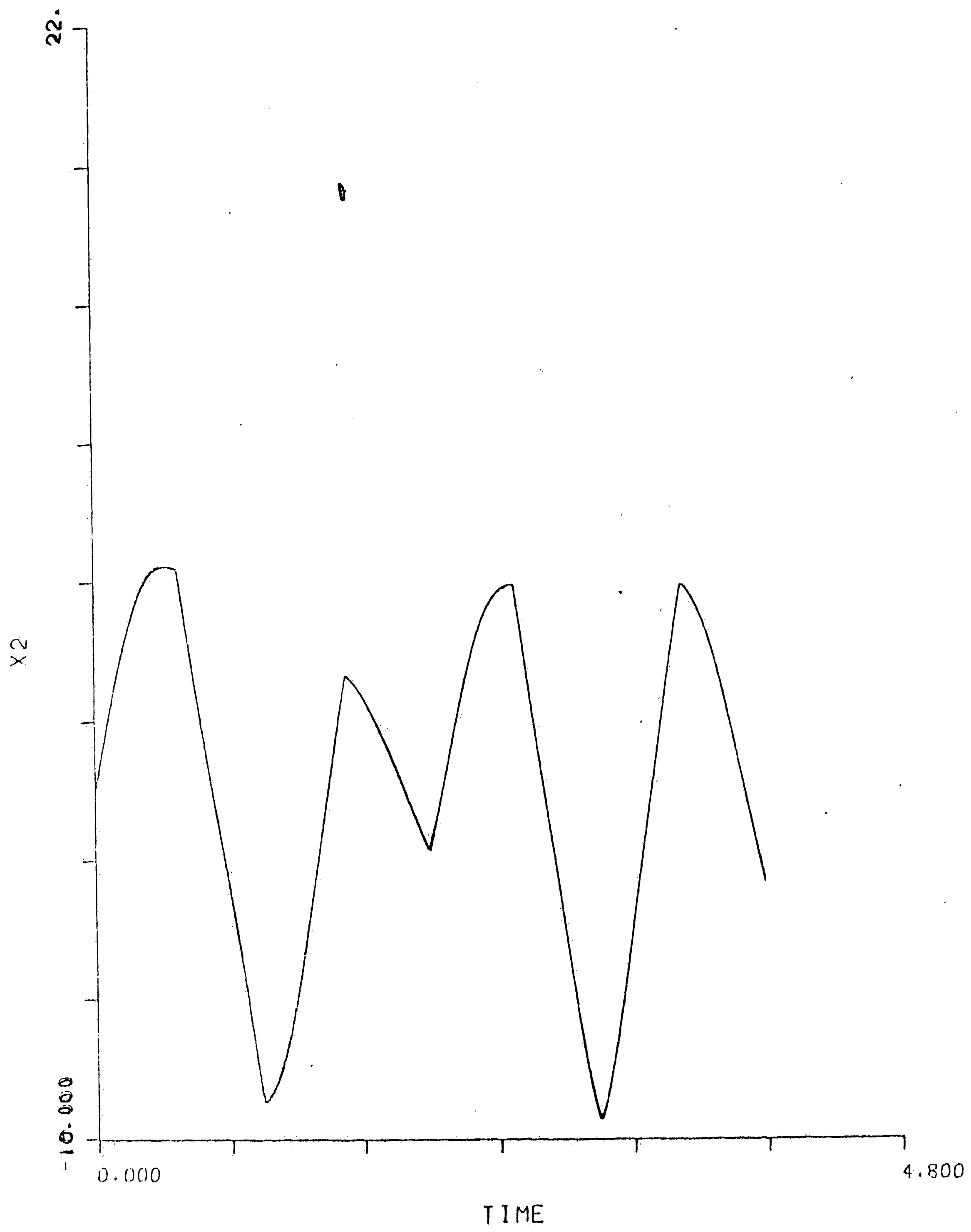


Figure 4.1.6c

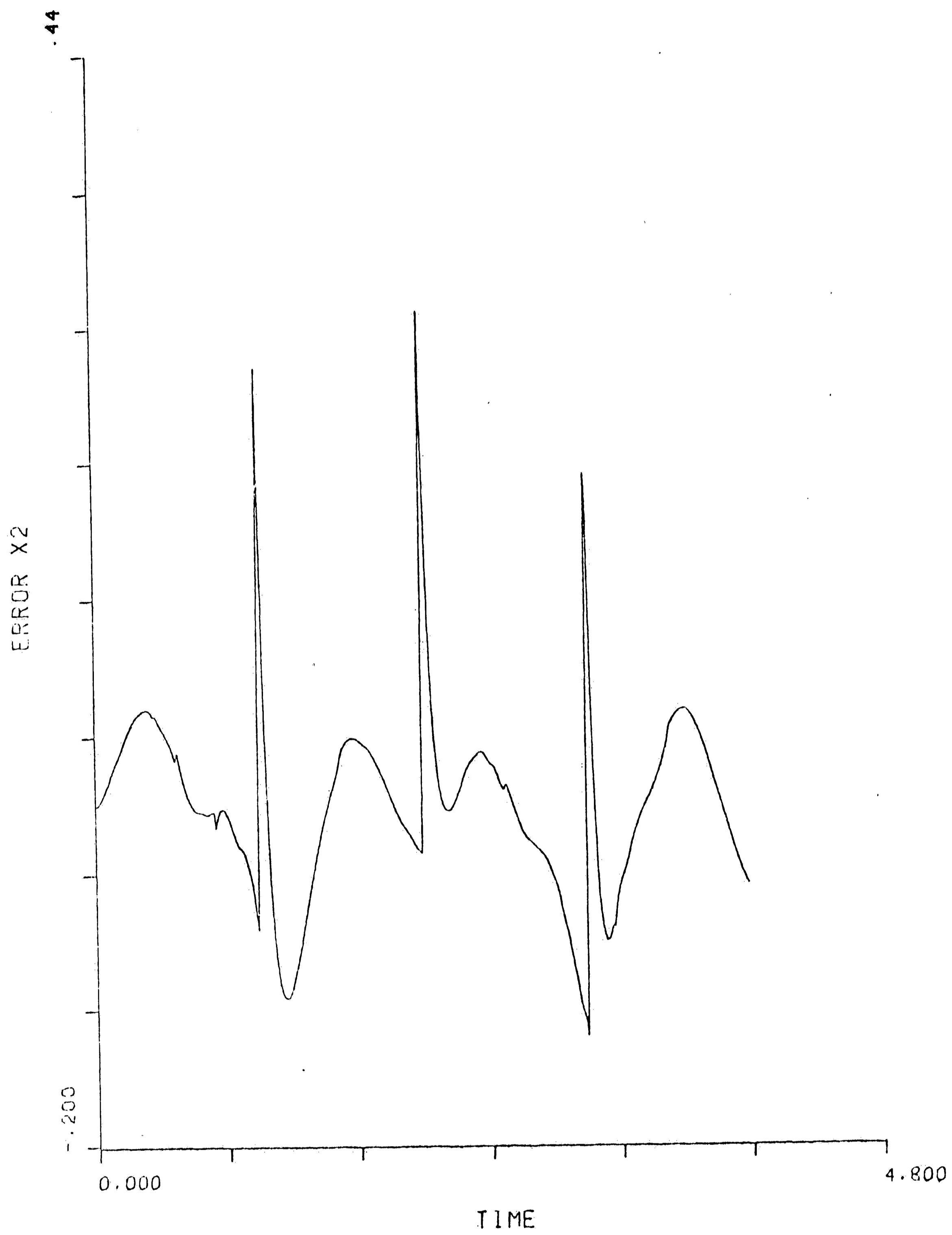


Figure 4.1.6d

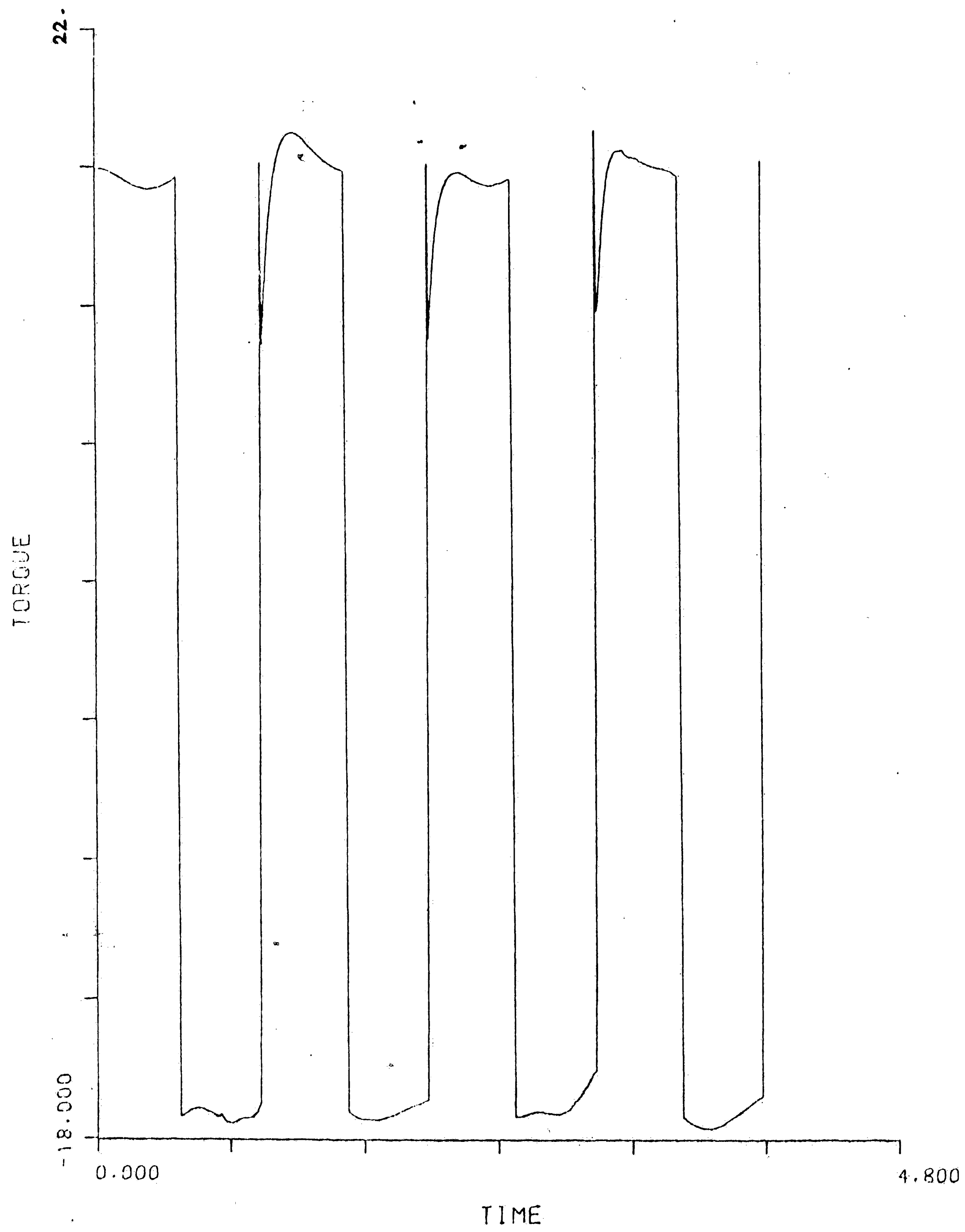


Figure 4.1.6e

4.2 CONCLUSIONS : What conclusions can we draw from the simulations presented ?

It is clear that we have good tracking properties. This is certainly true for the ideal case (see Figs.(4.1.1)), but in addition, our method has good robustness properties. It certainly can handle errors in initial position, as well as in switching time (this latter is essentially a form of initial position error). It also gives good results in the case of parameter uncertainty ($\delta f = 0$). This last case is, also, the most interesting one, since in the case of manipulators, changes in the payload, which are in part unknown , are an essential part of the operation. A method that can handle this kind of problem has a good potential for applications.

Also, we use a sampling rate that is relatively low to the spectral density of the system, especially in consideration of its nonlinearity. The only available results that can be really compared in a meaningful fashion with ours, are these of [16], where a sampling period of .00001 sec, as opposed to our .01, is used. In conjunction to the very modest on line computation required, the sampling rate we propose is quite acceptable, indeed.

As a final point, let us consider the applicability to higher order systems. Though this has not been tested yet, there does not seem to be any major obstacle to the use of our method for the control of higher order systems. There are two interesting points. The one is the method to be used for the approximation of trajectories. But if we consider these as functions of

time, we are again faced with the problem of approximating functions of one independent variable, and the solution can be provided by splines. The second point concerns the bounded nonlinearity hypothesis. It may be that there are unbounded nonlinear terms coupling the states of a high order system (a typical example is the industrial manipulator). Note, however, that the stability analysis of Chapter 2 can still be performed and provide insights on the appropriate selection of a stabilizing control method. A possible solution of the problem would be decoupling of the parts of the system (e.g. the links of a manipulator), which would be realized by an input approximately equal to the nonlinearity, and of opposite sign. Then the nonlinear part of the system equation would be close to zero, and we would get back to the boundedness hypothesis.

In conclusion, the simulations give quite encouraging results for the application potential of our method and it is not hard to foresee possible generalizations of these results.

REFERENCES

1. B.K. Kim, K.G. Shin , " Suboptimal Control of Industrial Manipulator with a Weighted Minimum Time-Fuel Criterion" , IEEE AC-30, n.1, pp.1-10, January 1985
2. M.E. Kahn, B. Roth, " The Near Minimum Time Control of Open Loop Articulated Kinematic Chains " , ASME JDSMC, Sept. 1971, pp.164-172
3. M. Athans, P.L. Falb, Optimal Control, McGraw-Hill, New York, 1966
4. S. Dubowsky, IEEE Robot Dynamics and Control Videoconference, Feb. 1985
5. N. Becker, "A Note on Performance Index Sensitivity of Time Optimal Control Systems " , IEEE AC-25, pp.819-823, 1980
6. E.P. Ryan, " On the Sensitivity of a Time Optimal Switching Function" , IEEE AC-25, n.2, pp.274-279, April 1980
7. A.S.I. Zinober, A.T. Fuller, "The Sensitivity of Nominally Time Optimal Control Systems to Parameter Variation" , Int. Journal of Control, vol.17, n.4, pp.673-703, 1973
8. W. Hejmo, " Sensitivity to Switching Function Variations in a Time Optimal Positional Control System" , Int. Journal of Control, vol.39, n.1, pp.19-30, 1984

- 9 W. Hejmo, "On the Sensitivity of a Time Optimal Positional Control" ,
IEEE AC-28, n.5, pp.618-621, May 1983
10. J. Rootenberg, R. Courtin, "Sensitivity of Optimal Control Systems with
Bang-Bang Control" , Int. Journal of Control, v.18, n.3, pp.537-543, 1973
11. P.M. Lynch, " Minimum Time Sequential Axis Operation of Cylindrical Two
Axis Manipulator", Proceedings of the JACC, 1980
12. J. Ahlberg, E. Nilson, J. Walsh, The Theory of Splines and Their
Applications, Academic, New York 1967
13. G. Ambrosino, G. Celentano, C. Garofalo, " Robust Model Tracking for a
Class of Nonlinear Plants", IEEE AC-30, n.3, pp. 275-279, March 1985
14. I.V. Emelyanov, "Reproduction of a Reference Input by Astatic Tracking
Systems with Variable Structure", Automation and Remote Control, n.1, pp.62-
72, 1965
15. J.J. Slotine, S.S. Sastry, "Tracking Control of Nonlinear Systems Using
Sliding Surfaces, with Applications to Robot Manipulators" , Int. Journal of
Control, vol.138, n.2, pp.464-492
16. G. Leitmann, "On the Efficacy of Nonlinear Control in Uncertain Linear
Systems" , ASME JDSMC, June 1981, vol.102, pp.95-101

17. J.N. Singh, A.A.R. Coelho, "Nonlinear Control of Mismatched Uncertain Linear Systems and Application to the Control of Aircraft", ASME JDSMC, Sept. 1984, vol. 106, pp.203-210
18. S. Jayasuriya, M.J. Rabins, R.D. Barnard, " Guaranteed Tracking Behavior in the Sense of Input-Output Spheres for Systems with Uncertain Parameters", Dec. 1984, ASME JDSMC, vol. 106, pp.273-279
19. S. Gutman, "Uncertain Dynamical Systems, a Lyapunov Min-Max Approach", IEEE AC-24, n.3, June 1979, pp.437-443
20. G. Leitmann, " Guaranteed Asymptotic Stability for Some Linear Systems with Bounded Uncertainties", ASME Trans., vol.101, Sept. 1979, pp.212-216
21. D.M. Wiberg, State Space and Linear Control Systems, McGraw-Hill, New York, 1970
22. Y. Takahashi, M.J. Rabins, D.M. Auslander, Control and Dynamic Systems, Addison-Wesley, Reading, 1970
23. G.P. Szego, " On the Application of Zubov's Method of Constructing Lyapunov Functions for Nonlinear Control Systems ", ASME Trans., vol.85, June 1963, pp.137-142
24. S.G. Margolis, W.G. Vogt, " Control Engineering Applications of V.I. Zubov's Construction Procedure for Lyapunov Functions ", IEEE AC-8, n.2, April 1963, pp.104-113

25. S. Gutman, " Uncertain Dynamical Systems - A Lyapunov Min-Max Approach ", IEEE AC-24, n.3, June 1979, pp.437-442
26. S. Weissenberger, "Stability Regions of Large Scale Systems ", Automatica, v.9, pp.653-663
27. R. Genesio, A. Vicino, " Some Results on the Asymptotic Stability of Second-Order Nonlinear Systems " , IEEE AC-29, n.9, September 1984, pp.857-863
28. M. Hayashi, Y. Ohsawa, "Transient Stability Region of Power System Using Series Expansion of Lyapunov Functions ", Int. J. Control, vol. 29, n.6, June 1979, pp.1059-1066
29. J.R. Hewitt, C. Storey, " Comparison of Numerical Methods in Stability Analysis ", Int. J. Control, v.10, n.6, June 1969, pp.687-701
30. E.J. Davison, K.C. Cowan, " A Computational Method for Determining Stability Regions of a Second Order Nonlinear Autonomous System ", Int. J. Control, v.9, n.3, March 1969, pp.349-357
31. M. Abu Hassan , C. Storey, " Numerical Determination of Domains of Attraction for Electrical Power Systems Using the Method of Zubov ", Int. J. Control, v.34, n.2, February 1981, pp.371-381
32. R. Bellman, I. Glicksberg, O. Gross, " On the Bang-Bang control problem ", Quarterly of Applied Math., v.14.n.1, March 1956.

APPENDIX

In this appendix we will give an analytical estimate of the attractive region of the system discussed in section 2.2

The relevant inequality is is 2.2.22, which we rewrite

$$\text{AP.1} \quad \dot{V} < -2(k||\underline{\delta}||^2 - (q|\delta_1| + r|\delta_2|)(k_1\delta_{s1} + k_2\delta_{s2} + C + \epsilon_0))$$

However,

$$\text{AP.2a} \quad |\delta_1| < ||\underline{\delta}||$$

$$\text{AP.2b} \quad |\delta_2| < ||\underline{\delta}||$$

So we can write the following inequality in $||\underline{\delta}||$

$$\text{AP.3} \quad k||\underline{\delta}||^2 - h||\underline{\delta}|| > 0$$

Where

$$\text{AP.4} \quad h = (q+r)(k_1\epsilon_{s1} + k_2\epsilon_{s2} + A + \epsilon_0)$$

The solution to this inequality is

$$\text{AP.5} \quad 0 < ||\underline{\delta}|| < h/k$$

And this defines one attractive region of the system, i.e. a region of the δ_1 - δ_2 plane, such that, when the system is outside of it, it tends to go back into it. Of course, a more sophisticated analysis, would shrink the bounds of this region and hence find another, "better" attractive region.

ON NOTATION

Underlined capital letters indicate matrices, e.g. A.

Underlined lowercase letters indicate vectors, e.g. a.

Subscripted lowercase letters indicate elements of either matrices (two subscripts) or vectors (one subscript). E.g. a_{ij} is an element of the matrix A, but a_i is an element of the vector a. If two or more subscripts are separated by a comma, this indicates two or more elements of a vector. E.g., $a_{i,j}$ indicates a_i, a_j .

CURRICULUM VITAE

Name: Athanasios Kehagias

Date of birth: April 14th 1961

Place of birth: Thessaloniki, Greece

Father's name: Ioannis

Mother's name: Soultana

Undergraduate School attended: School of Technology, Aristotelian
University of Thessaloniki, Thessaloniki, Greece.

Degree: B.Sc. in Electrical Engineering, June 1984.