THE UNIVERSITY OF ALBERTA

PERFORMANCE OF AN ADAPTIVE CONTROL ALGORITHM IN THE PRESENCE OF PLANT MODEL UNCERTAINTIES

by

© XIA, FEI

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

IN

CONTROL SYSTEMS

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1986

ISBN 0-315-32339-6

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR XIA, FEI

TITLE OF THESIS PERFORMANCE OF AN ADAPTIVE CONTROL ALGORITHM IN THE PRESENCE OF PLANT MODEL UNCERTAINTIES

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED FALL, 1986

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Fei Xia

PERMANENT ADDRESS:

Dept. of Automation

Qinghua Univ.

Beijing China

DATED Sept. 24 1986

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled PERFORMANCE OF AN ADAPTIVE CONTROL ALGORITHM IN THE PRESENCE OF PLANT MODEL UNCERTAINTIES submitted by XIA, FEI in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in CONTROL SYSTEMS.

V. G. Gourishankar
....
Supervisor

....

R. E. Rink
....

Date.....September 24 1986.....

Abstract

The adaptive control approach has been explored for many years. In the late 1970's, several algorithms of the model reference adaptive control class were proved to be globally asymptotically stable under certain assumptions. However they have not seen extended application as a result of these research. A main reason for this has been the performance of these adaptive control algorithms if the assumptions do not hold, especially when the upper bound of the plant order is not known.

For the theoretical proof of stability of these algorithms, the structure of the plant has to be known. When designing a control system, the plant structure is usually estimated. The actual plant dynamics which are outside the estimated plant structure are called unmodeled dynamics. It has been shown that the model reference adaptive control approach works poorly if there are unmodeled dynamics in the system. System stability is adversely affected by the presence of unmodeled dynamics. Hence the successful applications of adaptive control algorithms have so far involved ad-hoc methods to cope with this problem in most cases.

The effect of unmodeled dynamics on the performance of adaptive control systems has been investigated in a number of theoretical and numerical studies. This thesis contains numerical studies of one particular model reference adaptive control algorithm. These simulation studies are concentrated on the effects on system performance of a wide range of factors. These include the locations of the unmodeled dynamics poles, the control and adjustment interval, the order of the control law and the type of the input signal, etc.. The investigations try to clarify the areas which are not satisfactorily covered by the research results published in the literature, and to examine the "design guidelines" obtained with insufficient data in some publications.

A way of numerically measuring the system performance is introduced in this thesis to make the investigation work possible. Simulation instead of theoretical approach is employed because no satisfactory theoretical tool is available for the specific research area.

Acknowledgements

The author wishes to express special gratitude to the supervisor of the research leading to this thesis, Dr. V. G. Gourishankar, for his continuous help during the course of the study. Without his insightful guidence this project would not have been a reality. The more than two years of study under Prof. Gourishankar's supervision has been a fruitful and happy time for the author owing to his numerous advices not only on academic matters, but also on how to lead a useful life.

The author also wishes to thank the members of the advisory committee, Dr. R. E. Rink and Dr. D. G. Fisher, for their time and effort in helping to carry out this project and their valuable advices on problems in the thesis.

The author is greatly indepted to fellow graduate student, Mr. Sun, Xiaoqing. His helps have always been available when needed, including some of his software subroutines which proved to be indispensible to the work for this thesis. The author is very lucky to have found a real friend in him.

Next the author wants to thank a former teacher, Ms. Jin, Yihui, whose encouragement and example has been instrumental for the success of this work. As for the numereous helps received from many selfless friends, words are not enough to express the author's gratitude.

This project is conducted with a student grant from the Government of the P.R.C., as well as additional financial suport from the University of Alberta, the author is gratified for the availability of these funds.

Table of Contents

List of Tables

List of Figures

Figure		Page

Figure | Page

Figure		Page

1. INTRODUCTION

1.1 Adaptive Control

Designing a control algorithm to perform well usually requires a knowledge of the plant, commonly described in terms of a mathematical model. Since plant behavior changes with environment, it is not surprising that controllers which automatically adapt to the changing plant parameters were thought desirable even in the early days of control. However, the idea was not seriously considered owing to the almost impossible task of realizing such a controller in the absence of a digital computer. With the availability of fast digital computers at reasonable cost, the development of a class of controllers which hopefully would compensate for plant parameter changes mushroomed and a large number of algorithms are available today in the literature. In short, an adaptive controller is a controller whose parameters "adapt" to changes in the parameters of the plant being controlled.

However, in spite of the great amount of work devoted to this problem and the relatively long period after the first serious attempts were made to design such a control system (late 1950's according to [5]), the applicability of an adaptive controller to a practical system is still an open question generally. This is partly due to the fact that the adaptive control algorithms introduced up till now have one common characteristic: they are basically non-linear time-varying controller-plus-estimators. The analysis and prediction of system performance (e.g. theoretical proofs of stability) are not easy because of the mathematical complexities due to nonlinearity and time-variance.

Another reason is tendency of the algorithm to be unstable in the presence of plant unmodeled dynamics. If a plant or process being controlled is described by a finite order linear mathematical model, and the actual plant behaves differently from the model (including exhibiting higher order modes), the part of the plant that is not covered by the model is usually referred to as unmodeled dynamics (especially in adaptive control). For instance, if a plant has three degrees of freedom and includes some nonlinear factors, and a second order linear model which approximates the plant is used in designing the adaptive control algorithm, the least significant pole as well as the nonlinear part of the plant constitute the unmodeled dynamics. In the case of linear plants, "unmodeled dynamics" refer to the extra poles, zeros and time delays of a plant that are not covered by the lower order model chosen to represent the plant.

The general configuration of an adaptive controller is shown in Figure 1.1 Such a controller is employed to either cope with changing plant parameters or to obtain superior performance with a plant whose parameters cannot be predetermined. The adaptor part contains an estimator for the plant parameters, and adjusts the controller parameters continuously according to the most current estimates. The "estimator" is based on the knowledge of the plant configuration, including its order. With this knowledge the unknown or changing parameters can then be estimated using plant input/output history. Figure 1.1 represents therefore a configuration designed with a "known structure but missing parameters" situation in mind, not as one to cope with plants of unknown order.

Why, then focus on unmodeled dynamics which imply order uncertainty? It is known that few real life systems can be represented by a fixed order linear model. Many systems are nonlinear, and many nonlinear systems could not be exactly described by a linear model of finite order. For adaptive control algorithms to see extensive use in commercial, industrial and other practical applications, their performance in the presence of unmodeled dynamics would have to be satisfactory.

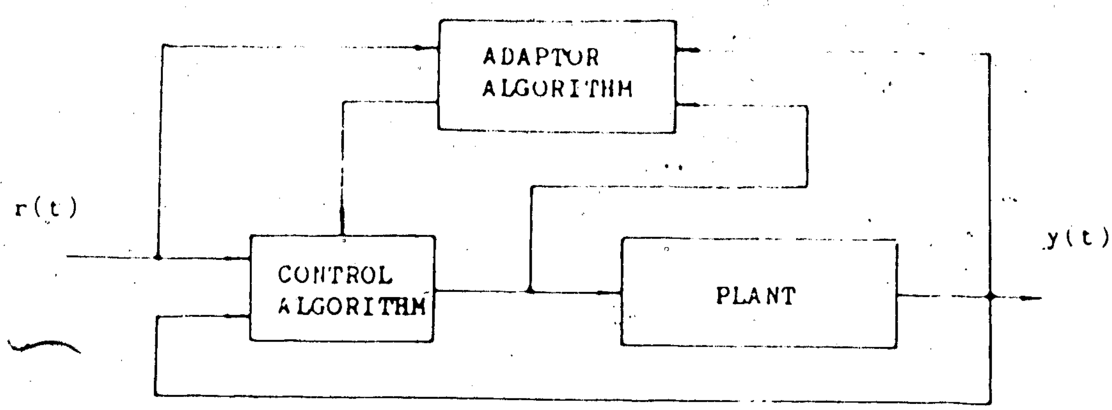


Figure 1.1 Adaptive Control System

1.2 A Survey of Recent Results

In spite of the many years of work in the area of adaptive control, the greatest obstacle to extensive applicability of this type of control concept still is the question of stability of the system when adaptive control is employed. Since adaptive controllers have a more complex mathematical description than conventional controllers, stability analysis of adaptive control systems is much more difficult (By a conventional controller is meant a linear feedback controller for a constant parameter plant.). And because of the "changing controller parameter in accordance with plant parameter estimate" scheme, the controller coefficients run the risk of being adjusted to the unstable regions owing to poor estimates [4].

Adaptive control algorithms reported recently in the literature generally perform better under certain conditions than a conventional controller. Some of the algorithms have been known to be globaly stable under certain assumptions [1-4]. The assumptions are:

1. an upper bound of the plant order is known;
2. the plant is minimum phase single input single output, (ie., all zeros of the open loop transfer function are in the left half s-plane in the continuous case and inside the unit circle in the z-plane in the discrete

time case);

3. the relative degree of the plant, ie., difference between the degree of the denominator polynomial and the degree of the numerator polynomial, is known; and
4. the pure time delay in the plant is known.

It should be noted that the assumptions do not require that the exact plant order be known but only its upper limit and the relative degree of the plant should be known. This has the effect of saying that a correct estimate as well as overestimate of the plant order while designing the control algorithm will produce stable results as long as the other assumptions are satisfied. What would be interesting to check is whether stability is retained if underestimation of the plant order occurs since this is almost always the case with real plants.

With proven stability when the assumptions hold true, work done so far in [1-4] shows that the adaptive control algorithms do their main job well, ie., to follow plant parameter changes, and generally give better performance than the conventional controllers if the algorithm design is executed carefully. However, the assumptions generally don't hold in real world plants (since no real plant could be exactly described by a finite-order model). Hence the system performance, especially stability, with one or more of the assumptions violated, naturally becomes an interesting topic of research after the theoretical proofs were introduced.

It should also be noted that the analytical methods used in [1-4] to prove the stability of the adaptive control algorithms are very special in the sense that they depart far from the Lyapunov methods commonly used to study stability in the case of linear systems. The approaches employed in these papers also differ from one another so much that each algorithm has its own unique stability proof method. This is mainly due to the nonlinear and time-varying nature of adaptive control. Ideally, it would be highly desirable if a unified approach could be developed for the performance and stability analysis of all adaptive control algorithms.

It would seem this had been achieved partially when in 1982, Rohrs in [4] developed a "unified approach" to analyze the performance, especially the stability, of several algorithms published in [1-3] and a few others. The research by Rohrs isolated and highlighted the effect of unmodeled dynamics on the performance of these algorithms:

Rohrs, in his thesis, proceeded to evaluate the behavior of these algorithms when the "known relative degree" assumption does not hold. Here is where the "unmodeled dynamics" come in.

By extensive computer simulation studies, Rohrs was able to show that the algorithms in [1-3] do not necessarily retain their stability when unmodeled dynamics are present in the plant. The primary reason of this is that the type of adaptive algorithms (model reference, dead beat, etc. found in [1-3]) will eventually "tune" the system gains to infinity (especially the high frequency gains). And the infinite gains in turn will make the system signals "blow up". While the analytical work of this thesis cannot be accepted as a mathematical proof that instability would occur should there be unmodeled dynamics involved in the system (more on this later), the very well designed computer simulations do give strong support to the following conclusion which could be found in Section3.3.8 of the thesis (in fact, one counter example in application is good enough to indicate that there may be problems in the usage of the algorithms): "When unmodeled dynamics are present in the plant, large constant reference inputs may prevent the algorithm from matching the model response to a change in the reference input r(t), which is its main task. Instability may result if the reference input further increases."

Ever since the problem of the stability of adaptive controllers in the presence of unmodeled dynamics was analysed by Rohrs [4, 4a], efforts have been made to obtain algorithms which will be "immune" to this problem or at least will lessen the inpact of unmodeled dynamics on system performance. Meanwhile, almost every successful application of some adaptive control algorithm in real systems so far employs some kind of modifications upon the basic algorithm and seems to be only good for one particular plant. This means, in effect, that only local stability can be achieved if unmodeled dynamics are present even if great care is taken in the algorithm design stage. These indirectly support Rohrs' statement that instability is the result of a mechanism inherent in the algorithms which tunes the controller gains to infinity if unmodeled dynamics are present.

1.3 Thesis Objective

Although a certain amount of work has been done concerning the problem of unmodeled dynamics in adaptive control systems, it has been confined only to stability issues and qualitative studies. The conclusions and results have pointed out that unmodeled dynamics in adaptive control systems may cause instability. The field is still open for quantitative studies that observe the gradual change of system performance with the changes of factors such as unmodeled pole positions and sampling frequencies. The "design guidelines" offered by Rohrs could also be improved if system behavior is examined in more detail. For instance, it was indicated as a guideline in [4b] that the slower the sampling rate is, the more stable the system will be. This conclusion was arrived at with comparisons of only two different sampling rates. Quantitative studies can help clarify these results.

This thesis will try to make a contribution to design guidelines by studying the performance of a particular adaptive control algorithm in the presence of unmodeled dynamics. This will include quantitative studies and comparisons with conventional control approaches so that an appropriate range of its application can be established.

1.3.1 Outline of the Thesis

In this thesis, the behavior of one particular adaptive control algorithm in the presence of different unmodeled dynamics is investigated through computer simulations as well as analysis using currently available methods. The results of these investigations will help to gain an insight into the role which unmodeled dynamics play in adaptive control systems, especially their effects on system stability.

In Chapter 2 we will proceed to describe a limitation of Rohrs' "unified analytical approach" and the its inadequacy as a theoretical analysis procedure, as well as other minor shortcomings in his thesis. However, we will also show that beside being part of the set of sufficient conditions for global stability of the algorithm of (1.3.4-8) below, the relative degree as well as order upper limit assumptions are also necessary ones as indicated by Rohrs.

Also in Chapter 2, the results of a series of computer simulations will be described which will establish the relationship of system performance and unmodeled dynamics using the algorithm proposed in (1.3.4-8). The effect of changes in the locations of the unmodeled dynamics poles and control sampling periods will be observed.

In all of Chapter 2, the control algorithm will be used in its first order form, with a third order plant. In Chapter 3, higher order simulations will be conducted. Also, the consequences of overestimating the plant order will be analyzed through simulations of a lower order plant model controlled by a higher order algorithm.

In chapter 4, we will try to establish the applicable range of the particular algorithm. In doing so, we hope to identify the type of situations where the algorithm in (1.3.4-8) performs better than conventional control algorithms. If such situations could not be found in the case of a particular algorithm or approach, its existence cannot be justified. The algorithm's advantages and disadvantages will be carefully observed and analyzed again through a number of simulations and compared with conventional algorithms. The results will be helpful in determining whether an algorithm like that in (1.3.4-8) should be considered in a particular situation.

1.3.2 Controller Algorithm investigated in This Thesis

Numerous adaptive control algorithms have been [illegible]ed in the literature. These are often divided into groups by the nature of their design. Model reference adaptive control algorithms constitute one of the classes. Several adaptive control algorithms that have had their stability proven belong to this group. The study covered by this thesis is conducted within this group. Figure 1.2 shows the general configuration of model reference adaptive control algorithms.

Since the time available is limited, and fairly extensive simulations are required, only one of the many available algorithms is singled out in this research. This algorithm was developed by Goodwin, Ramadge and Caines [19, 20]. It is one of the simplest and its stability proof is quite straightforward. A detailed study of the effect of different unmodeled dynamics on one algorithm should provide a reasonable guidance as to what one can expect with the

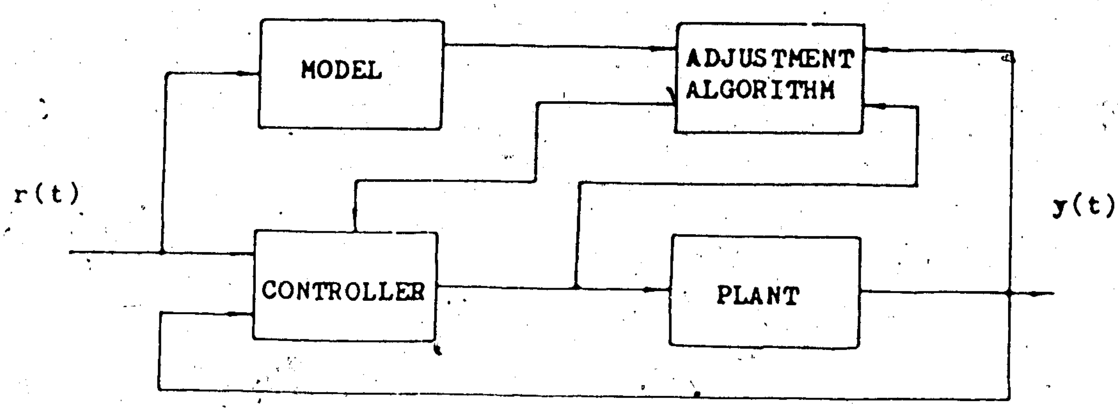


Figure 1.2 Model Reference Adaptive Control

others, if it could be shown that the problem with unmodeled dynamics is caused by some common characteristic of the adaptive control approach.

Since adaptive control algorithms are usually implemented with digital computers, the work in this thesis will be conducted in the discrete-time domain. To compare effects of unmodeled dynamics on the performance of different control configurations, adaptive and non-adaptive, a model reference control algorithm without adjustment mechanism will be introduced as well.

In this thesis, the single-input single-output plant to be controlled is represented by the discrete-time model:

$$A(z^{-1})y(t)=z^{-d}B(z^{-1})u(t) \qquad (1.3.1)$$

where u(t) is the input and y(t) is the output of the plant, and A and B are polynomials defined by

$$A(z^{-1})=1+a_1z^{-1}+\ldots\ldots+a_nz^{-n}$$

$$B(z^{-1})=b_0+b_1z^{-1}+\ldots\ldots+b_mz^{-m},\ b_0\neq0,\ (n\geq m+d),$$

where z is the Z-transform operator, d is the pure time delay (presumed to be an integral multiple of the sampling period), and t is the discrete time 0,1,2,3.... etc..

The main idea behind the model reference approach is to have the plant output follow that of a specific model which is considered desirable. This model is called the reference model. When m, n and d are known, a reference model is chosen to be as follows:

$$A_M(z^{-1})y^*(t)=z^{-d}gB_M(z^{-1})r(t) \qquad (1.3.2)$$

where A_M and B_M are polynomials satisfying

$$A_M(z^{-1})=1+a_{M1}z^{-1}+......+a_{Mn}z^{-n}$$

$$B_M(z^{-1})=1+b_{M1}z^{-1}+......+b_{Mm}z^{-m}$$

where g is a constant gain, and r(t) is the reference input. $A_M(z^{-1})$ has no zeros outside the unit circle in the Z plane.

If the exact degrees of the A and B polynomials are not known, n would be chosen as the upper limit of the degree of A (which is desired to be known for stability). m is then obtained as n minus the relative degree (which is the difference of degrees of A and B).

The objective is to apply feedback to the system so that the output of the plant matches that of the model driven by the same reference input. When the parameters of the plant are known, the following non-adaptive model reference control algorithm found in [20]:

$$B(z^{-1})A_M(z^{-1})u(t)=gB_M(z^{-1})A(z^{-1})r(t) \quad (1.3.3)$$

will attain the objective. In effect, this u(t) will drive the plant in such a way that

$$y(t+d)=y^*(t+d)$$

will be achieved.

This control law is very similar to a deadbeat control law and the knowledge of plant parameters is, of course, mandatory.

When the plant parameters are not known, the following model reference adaptive control algorithm found in [20] and [21] and referred to as DA2 in [4] adjusts controller gains to cause y(t) to follow the model output.

Plant input:

$$u(t)=\underline{\theta}^T(t)\underline{\phi}(t) \quad (1.3.4)$$

where $\underline{\theta}$ is the controller gain vector and $\underline{\phi}$ is called the auxiliary variable vector. They are of the same order, namely n+m+d-1. Vector $\underline{\phi}$ is defined as:

$$\underline{\phi}^T(t)=[y(t),\ldots,y(t-n+1),u(t-1),\ldots$$
$$\ldots,u(t-m-d+1),\phi_r(t)] \tag{1.3.5}$$

where

$$A_M(z^{-1})\phi_r(t)=gB_M(z^{-1})r(t) \tag{1.3.6}$$

or

$$\phi_r(t)=y^*(t+d) \tag{1.3.7}$$

with gain adjustment mechanism:

$$\underline{\theta}(t)=\underline{\theta}(t-d)-f\underline{\phi}(t-d)[1+\underline{\phi}(t-d)^T\underline{\phi}(t-d)]^{-1}e(t) \tag{1.3.8}$$

where f is a user chosen parameter to control adjustment. It is chosen so that $g \times f \leq 2$.

This algorithm will ultimately achieve the goal:

$$\lim_{t\to\infty} y(t)=y^*(t)$$

The performance of this algorithm will be compared with the non-adaptive model reference control algorithm of 1.3.3.. For ease of reference, algorithm (1.3.4-8) will be referred as A1 and (1.3.3) as A2 from this point on.

1.3.3 Plant Models used in This Thesis

Two kinds of plants will appear in this thesis. Both are linear, and the unmodeled dynamics are represented in terms of extra poles. For the sake of convenience and ease of understanding, the plant models will be represented in the continuous domain, by poles and zeros in the s-plane.

In Chapter 2., a plant model developed from the Rohrs version (introduced in [4]) will be used. The Rohrs plant model is as follows:

$$(s+1)(s^2+30s+229)Y(s)=458U(s) \qquad (1.3.9)$$

which contains a dominant pole at $s=-1$ and a pair of complex poles at $s=-15\pm j2$ (serving as unmodeled dynamics).

The plant model in Chapter 2. will be based on this model. However in order to investigate the effect of changing unmodeled dynamics, which is one of the objectives of this thesis, the location of the poles will have to be flexible. The model is shown here:

$$(s+a)(s+\alpha+j\beta)(s+\alpha-j\beta)Y(s)=2a(\alpha^2+\beta^2)U(s) \qquad (1.3.10)$$

where $a>0$, $\alpha>0$, $\beta>0$, and s is the Laplace transform operator. In this model, the complex pole pair will represent the unmodeled dynamics and $\alpha > a$ is required.

In Chapter 3., in order to examine the effect of unmodeled dynamics in a higher order system and the effect of changing the controller order, a 4th order plant containing a pair of complex poles is used. Its model is as follows:

$$(s^2+2\alpha s+\alpha^2+\beta^2)\prod_{i=1}^{2}(s+\alpha_i)Y(s) = 2\alpha_1\alpha_2(\alpha^2+\beta^2)U(s) \qquad (1.3.11)$$

where $\alpha_2 > \alpha_1 > 0$, and $\alpha > 0$, $\beta > 0$. Depending on the controller order, some of these poles will serve as unmodeled dynamics.

In analysis and simulations, which will be carried out in the discrete-time domain, discrete versions of these models will be used. For ease of reference, the algorithms A1 and A2 as well as plant models (1.3.10) and (1.3.11) are collectively listed in Tables 1.1 and 1.2 below.

Algorithm A1:

$$u(t)=\underline{\theta}^T(t)\underline{\phi}(t) \tag{1.3.4}$$

$$\underline{\phi}^T(t)=[y(t),\ldots,y(t-n+1),u(t-1),\ldots$$
$$\ldots,u(t-m-d+1),\phi_r(t)] \tag{1.3.5}$$

$$A_M(z^{-1})\phi_r(t)=gB_M(z^{-1})r(t) \tag{1.3.6}$$

$$\underline{\theta}(t)=\underline{\theta}(t-d)-f\underline{\phi}(t-d)[1+\underline{\phi}(t-d)^T\underline{\phi}(t-d)]^{-1}e(t) \tag{1.3.8}$$

Algorithm A2:

$$B(z^{-1})A_M(z^{-1})u(t)=gB_M(z^{-1})A(z^{-1})r(t) \tag{1.3.3}$$

Table 1.1 Algorithms used in The Thesis

$$(s+a)(s+\alpha+j\beta)(s+\alpha-j\beta)Y(s)=2a(\alpha^2+\beta^2)U(s) \quad (1.3.10)$$

$$(s^2+2\alpha s+\alpha^2+\beta^2)\prod_{i=1}^{2}(s+\alpha_i)Y(s)= 2\alpha_1\alpha_2(\alpha^2+\beta^2)U(s) \quad (1.3.11)$$

Table 1.2 Plant Models used in The Thesis

In Chapter 2, plant model (1.3.10) will be used with $a=1$. In Chapter 3, plant model (1.3.11) will be used with $\alpha_1=1$ and $\alpha_2=1.5$. In Chapter 4 plant model (1.3.10) will be used with changing parameters.

2. Effect of Unmodeled Dynamics on a First Order Algorithm

2.1 Introduction

In this chapter, the plant is presumed to follow (1.3.10). It is assumed to be of first order when the control algorithm is designed, although the actual plant is of third order. Hence the complex pole pair will act as unmodeled dynamics, because of the relative dominance of the real pole at $s=-a$.

The studies in this chapter is concentrated on the performance of algorithm A1 in its first order form, ie., $n=1$, $m=0$ and $d=1$ in (1.3.4-8). The effect of unmodeled dynamics on the performance of this algorithm is observed through a series of simulations. The effectiveness of using longer control intervals to improve system stability will also be studied.

2.1.1 The Linearization Approach

In his thesis [4], Rohrs concluded that when plant (1.3.9) is controlled by algorithm A1 designed to follow a first order model

$$(s+3)Y^{*}(s)=3R(s) \tag{2.1.1}$$

the unmodeled dynamics of the complex pole pair $s=-15\pm j2$ in (1.3.9) introduce instabilities into the system.

The analytical approach employed by Rohrs involves the linearization of the adaptive control system about an operating point. After the system is linearized, the root locus method is used to analyze its stability. The applicability of this approach to adaptive control systems is questionable for the reason stated below.

To linearize an adaptive control system like Rohrs did in [4], it has to be presumed that the adjustable controller parameters remain constant or do not change too far from an operating point. If the system is stable, these parameters reach certain steady-state values provided there is no disturbance. If there are no unmodeled dynamics in the system, the algorithm A1 asymptotically adjusts its control parameters to those of A2. These could be used as the operating point for linearization. In fact, the final linearized model will be the same as the plant controlled by A2, no matter what values other factors like the reference input and the adjustment factor f are set at. Linear system analysis techniques could then be applied to this system.

If there are unmodeled dynamics in the system, however, the steady-state values of the control parameters depend on the position of the unmodeled dynamics poles as well as other operating factors like f and the reference input, even if the system is stable. If the system is not stable, the controller gains or parameters will approach infinity. In this case, it is obvious that no steady-state will be reached, and the linearization process is completely useless. If the system is stable in spite of the unmodeled dynamics being present, the linearization approach could be applied in a way like Rohrs did. First a computer simulation is done to determine the steady-state values of the control parameters for a certain set of unmodeled dynamics and at a certain operating condition (ie., r(t) and f values). Then the system is linearized around this operating point. The stability analysis of this resultant linear system means little, however, since the simulation results has shown that the system is stable. In the Rohrs thesis [4], this approach is used to plot root loci for the system when r and f are changed. These root loci are at best approximations since the operating point, ie., the steady-state values of the control parameters, changes with r and f with unmodeled dynamics present in the system. For this approach to be even marginally meaningful, changes of r and f have to be small for the actual operating point to be in a limited vicinity of the presumed one.

Rohrs' approach is therefore not mathematically rigorous and of very limited use at best. Since the studies in this chapter and the rest of the thesis involve changes of the operating conditions, unmodeled dynamics, and other factors of systems, the linearization approach cannot be successfully employed here. Simulations, not theoretical analysis will be the main tool used here because the objective is to quantitatively observe the effects of several factors (including unmodeled dynamics) on the performance of algorithm A1.

2.1.2 The Simulation Approach, Time Periods

The simulation methods employed in [4] also leave some things to be desired. First of all, for simulations of a control system, ideally the plant should be modeled on a precise analog computer. Lacking such a device, a discrete time model of the plant should be constructed and used uniformly throughout the whole ocess of analysis.

The sampling period used here should be small enough to represent all the modes in the plant, and should not be changed even if the control and adjustment intervals are changed. In [4], however, the sampling period for obtaining the discrete model of the plant is set to be equal to the control interval. While this simplifies the programming, a price has to be paid in terms of the accuracy of the representation of the plant. This is especially true when the sampling period was set at 0.4 seconds, since this violated the Shannon sampling law for the unmodeled dynamics modes at $-15 \pm j2$. These unmodeled dynamics modes are of high frequency compared to the dominant pole at $s=-1$. Without knowing the plant output in between the sampling points, the conclusions concerning the effect of changes in the sampling period on system performance reached in [4] need clarification.

In the simulation studies in this thesis, there are two different time periods involved. The first is the interval of the control parameter adjustment. The representation of algorithm A1 and A2 in Table 1.1 is based on this interval. This time period will be denoted as T throughout the thesis and different values will be used in the simulations. The plant input $u(t)$ will also be adjusted in this interval. The discrete time $t=1,2,3,...$ of (1.3.3-8) is actually in intervals of T, ie., T,2T,3T,... etc..

The second time period is the sampling period used for the discrete model of the plant. For ease of analysis, the plants used in this thesis will be discretizations of models in the continuous time domain. The discretization, however, has to be done with a fast enough sampling period for the discrete time model to represent all the modes in the original plant. This sampling period of the plant model discretization does not have to be the same as T. In the simulations of this thesis, this sampling period will be denoted as T_p. For uniformity, T_p=0.01sec. is kept through all the thesis, which is fast enough for all the plants used in the simulations.

In all the simulation studies of this thesis, $T>T_p$. This means in each control interval, the plant output is calculated for more than once while its input remains the same. This improves the accuracy of the simulations, especially when the effect of unmodeled dynamics is studied, and T is too long to be used for the discrete model of the plant.

2.2 Qualitative Studies

The simulations in [4] are also flawed in another sense, because the results do not include comparisons of systems with and without unmodeled dynamics. Consequently they do not offer proper isolation of the effect of unmodeled dynamics on system performance. However a qualitative examination of Rohrs' results will be helpful in preparing for further studies of the problem.

It is now known that unmodeled dynamics may introduce instabilities into an adaptive control system [4, 4a, 4b]. It is worth examining whether the same is true when the algorithm is not adaptive. Rohrs claims that he has identified the reason for the instability as being due to the adjustment mechanism in all the model reference adaptive control algorithms tuning the controller gains ($\underline{\theta}(t)$) to infinity in the presence of unmodeled dynamics. A direct comparison of simulation results of systems with the same plant controlled by algorithms A2 and A1 seems to offer a good starting point for helping to clarify whether Rohrs' conclusions are fully justified.

From the discussions in the previous chapter, it should be clear that algorithm A2 is the same as algorithm A1 without its gain adjustment mechanism. In other words if there are no unmodeled dynamics and if the initial values of gains are selected properly, the adaptive algorithm will perform exactly as the non-adaptive algorithm would. In this section, simulations will be conducted of both algorithms with and without unmodeled dynamics, and comparisons made to identify the roles of both unmodeled dynamics and adaptation.

2.2.1 Performance of Algorithm A1 in the Presence of Unmodeled Dynamics

The first two simulations of this section are excecuted under the same conditions as those represented by Figures 5-7. and 5-9. in [4]. These are chosen since they represent the kind of situation where the system is still stable but the instability effects of unmodeled dynamics are felt. The third and last simulation uses the condition of Figure 5-10. of [4], since it is supposed to be the particular point when the system starts to go unstable. These figures are reproduced here directly from [4] for reference purposes.

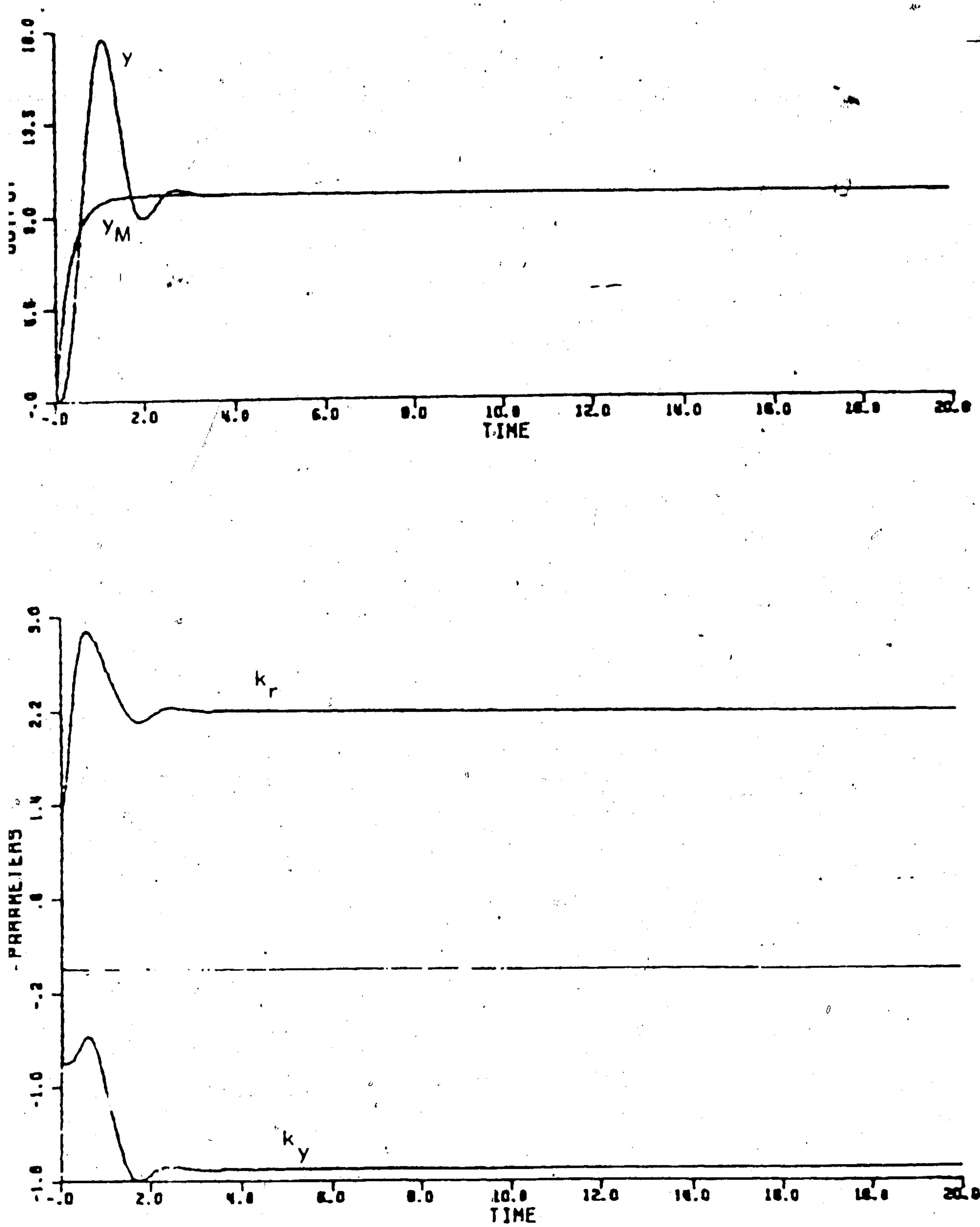
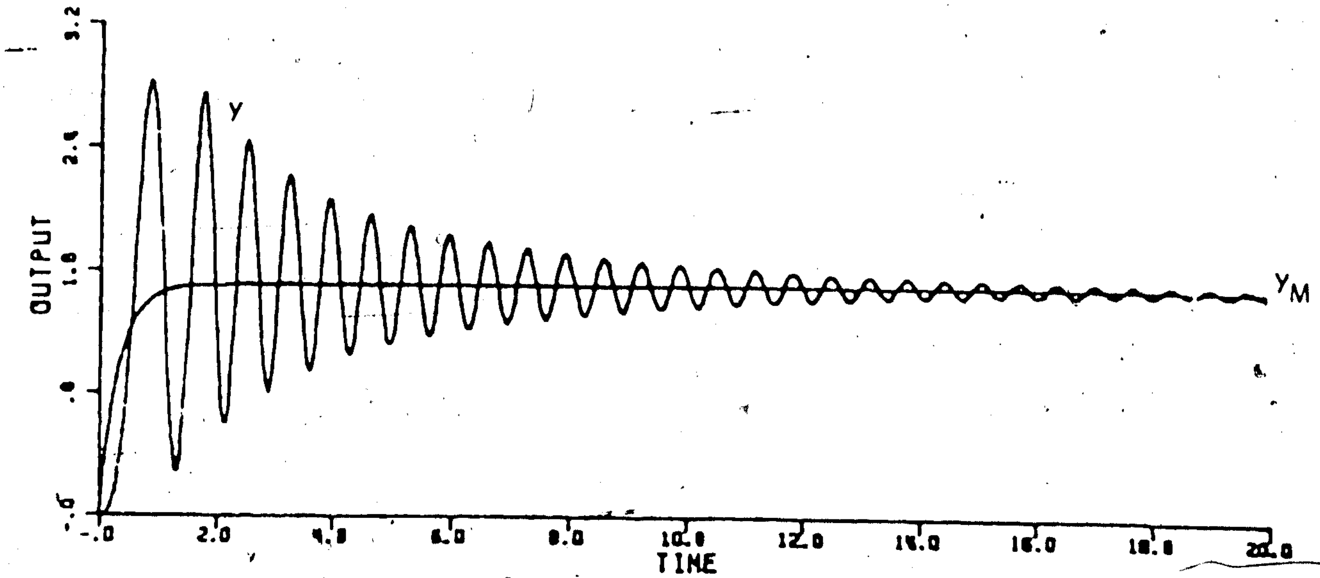


Figure 5-7. Simulation of DA2 with unmodeled dynamics, $r=10.0$, and $\gamma=0.2$



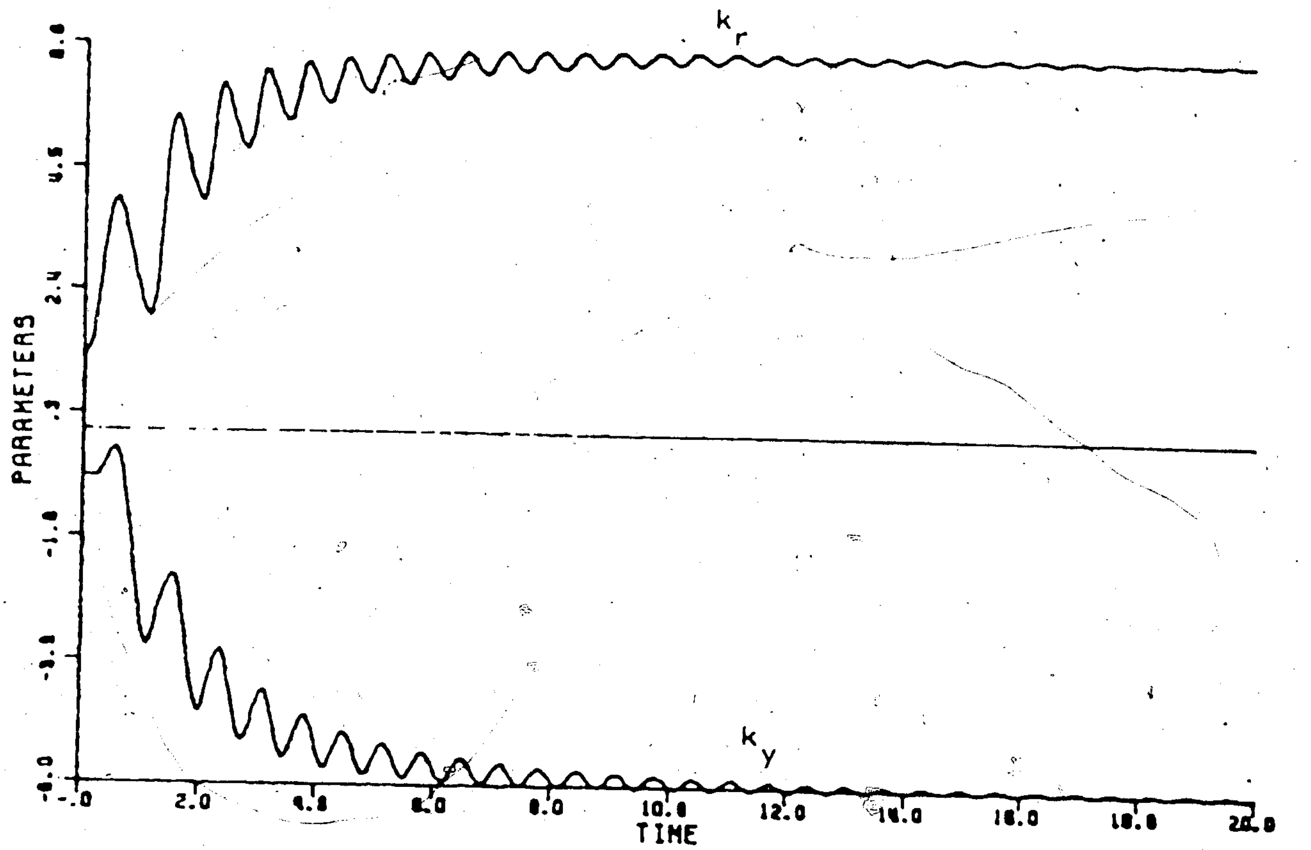


Figure 5-9. Simulation of DA2 with unmodeled dynamics, r=1.5, and γ =1.0.

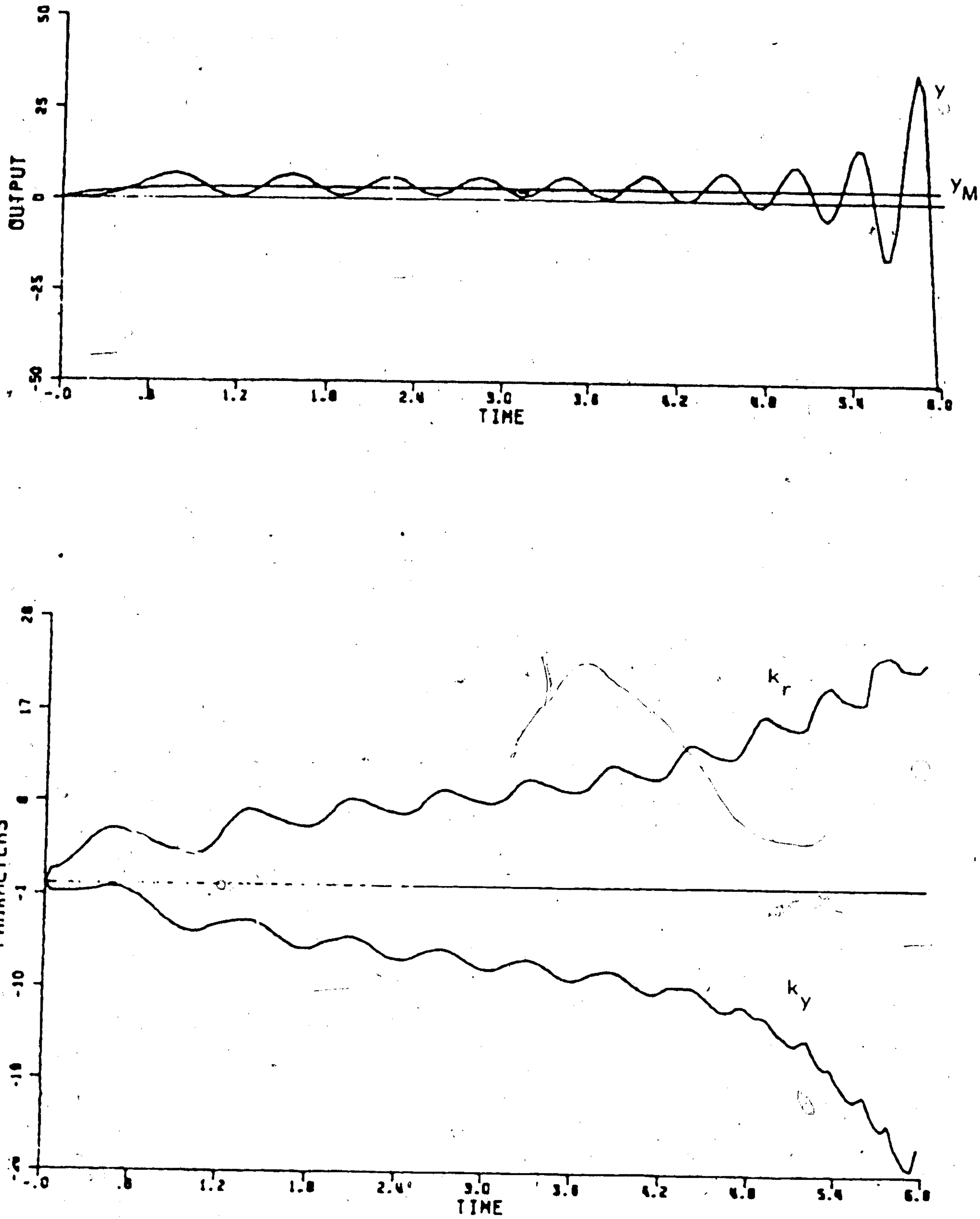


Figure 5-10. Simulation of DA2 with unmodeled dynamics, r=1.62, and γ=1.0
(System eventually becomes unstable.)

Figures 2.1, 2.2 and 2.3 show the results of the simulations. The simulations are conducted using almost exactly the same method as Rohrs used, except that the plant is represented by a more accurate discrete model of the analog transfer function (1.3.9) compared to [4] because of the higher sampling period T_p used here. Also different is the examination of system performance when unmodeled dynamics are not present.

In these plots (Figures 2.1 to 2.3), y(t) is the system output with unmodeled dynamics, when the algorithm A1 is applied to plant (1.3.9); $y^*(t)$ is the output of the reference model the system is supposed to follow; while $y_p(t)$ is the system output if there is no unmodeled dynamics, or when A1 is applied to the alternative plant

$$(s+1)Y(s)=2U(s) \qquad (2.1.2)$$

which is (1.3.9) minus the unmodeled poles. The adjustable gains or control parameters are marked as k_r and k_y. They constitute the vector $\underline{\theta}(t)$ of (1.3.4). That is,

$$\underline{\theta}^T(t)=[k_y(t),k_r(t)]. \qquad (2.1.3)$$

From the results, it is very clear that unmodeled dynamics do cause instability in adaptive control systems.

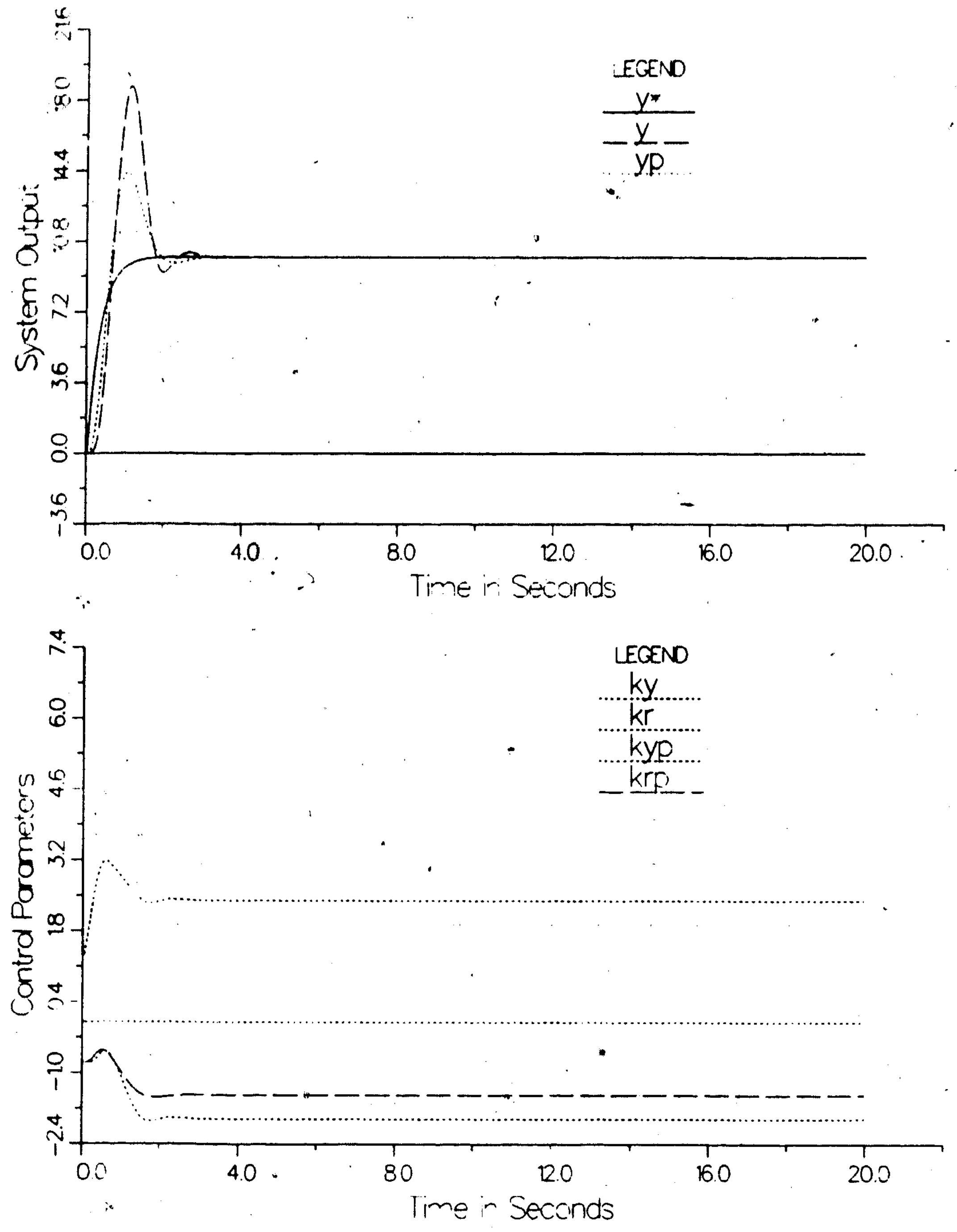


Fig. 2.1 Response for A1, r=10.0, f=0.2

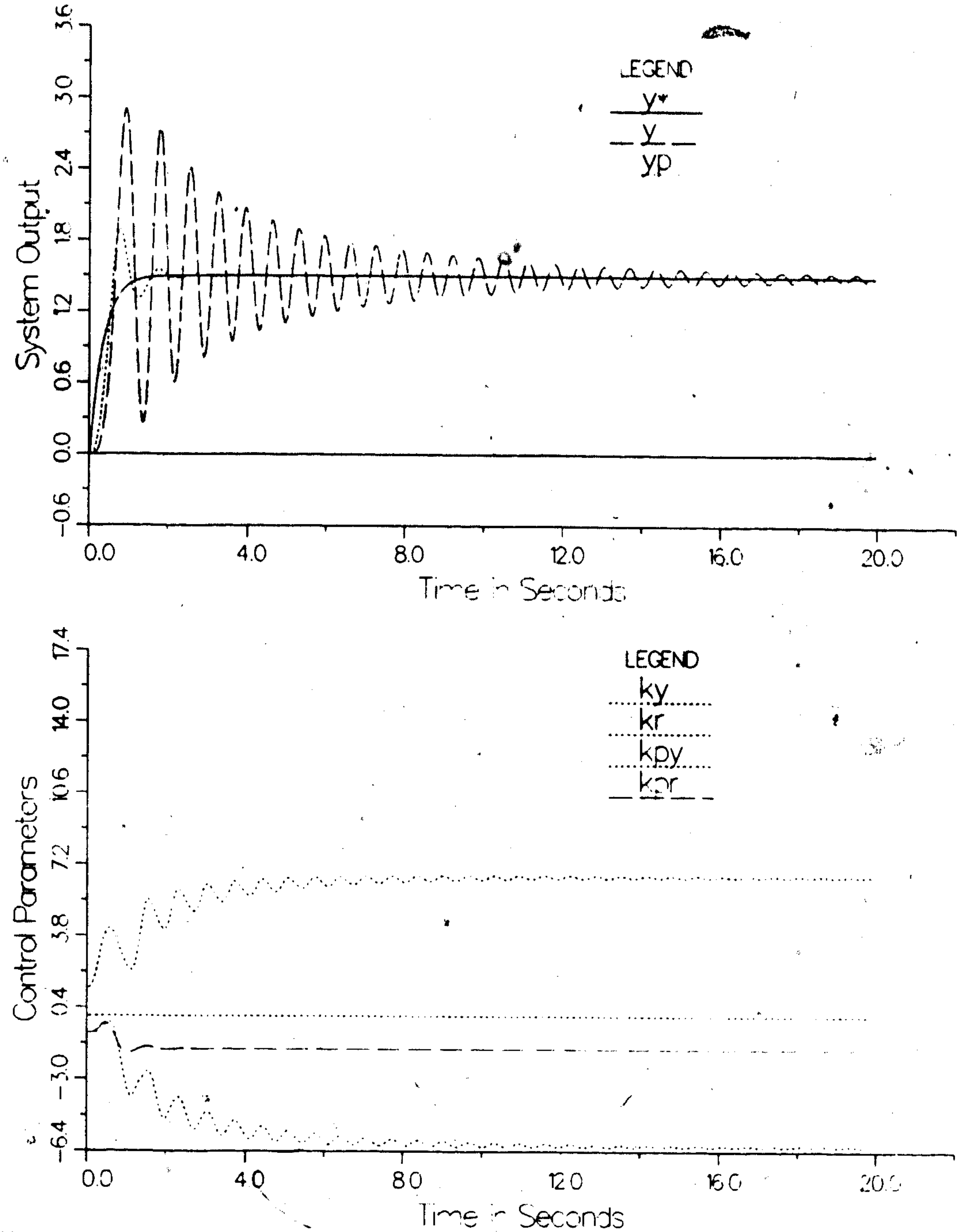


Fig. 2.2 Response for A1, r=1.5, f=1.0

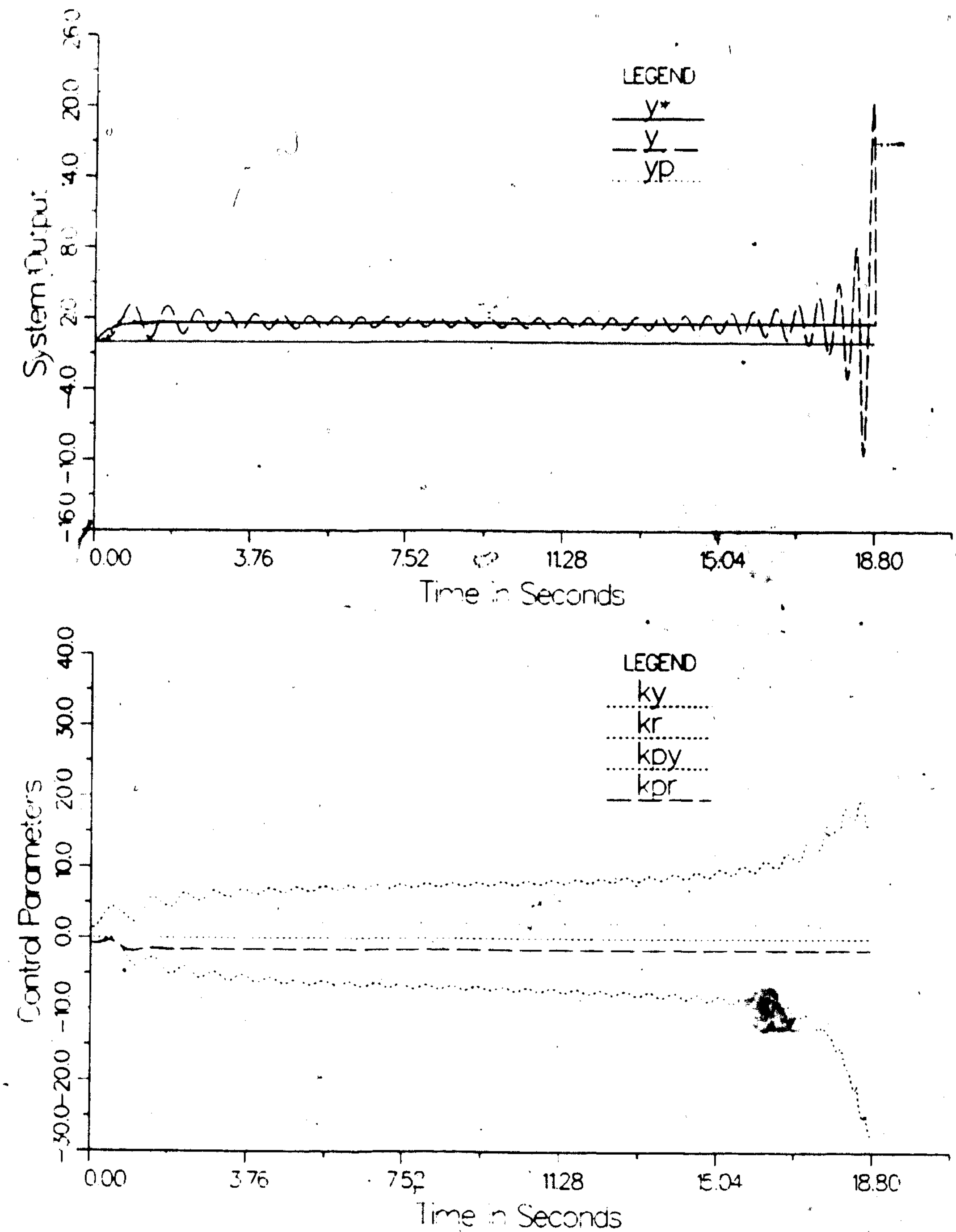


Fig. 2.3 Response for A1, r=1.02, f=1.0

However, how does other control schemes perform in the same kind of situation? Is the problem with unmodeled dynamics unique for adaptive control only? We try to answer this question in the next subsection.

2.2.2 Performance of Algorithm A2 in the Presence of Unmodeled Dynamics

The algorithm A2, designed to match the output of the system with the model in a finite number (d) of control intervals, is a kind of 'deadbeat' controller.

The design of such a controller demands precise knowledge of the mathematical model of the plant, since the determination of u(t) using A2 requires knowledge of plant polynomials A and B. Any uncertainty in these parameters would make it impossible to implement such a controller. Also, if a controller is designed with false or inadequate knowledge of the plant, the desired result, ie., $y(t+d)=y^*(t+d)$ will not be achieved. It is interesting to see the performance of this algorithm because it could be compared with that of A1 to identify the function of the adjustment mechanism. If its performance with unmodeled dynamics is significantly better than that of A1, we would be able to come to the conclusion that the part of A1 that reacted unfavorably to unmodeled dynamics is definitely its adaptive mechanism. Also it should be obvious that most other conventional controllers would perform better with unmodeled dynamics present than adaptive ones since they do not impose as stringent conditions on plants as even A2 does.

The plots in Figures 2.4, 2.5 and 2.6 show the results of simulations with algorithm A2 and the same plants used with A1 in Figures 2.1, 2.2 and 2.3. The same notation is used. In all the simulations the system output $y(t)$ closely followed the model output $y^*(t)$ as it is supposed to, although $y(t)$ did not equal $y^*(t)$ after $d=1$ step as designed ($y_p(t)$ did, of course). No stability problems whatsoever turned up.

2.2.3 Discussion

From simulation results above, we see clearly that when unmodeled dynamics are present, the adaptive algorithm A1 will adjust its control gains to steady state values different from those obtained with no unmodeled dynamics, even if the system does not eventually get to be unstable. The third order plant of (1.3.10) does not have the same input/output relationship of the supposed first order plant structure for which the algorithm is designed. Any differences of the input/output relationship of the plant

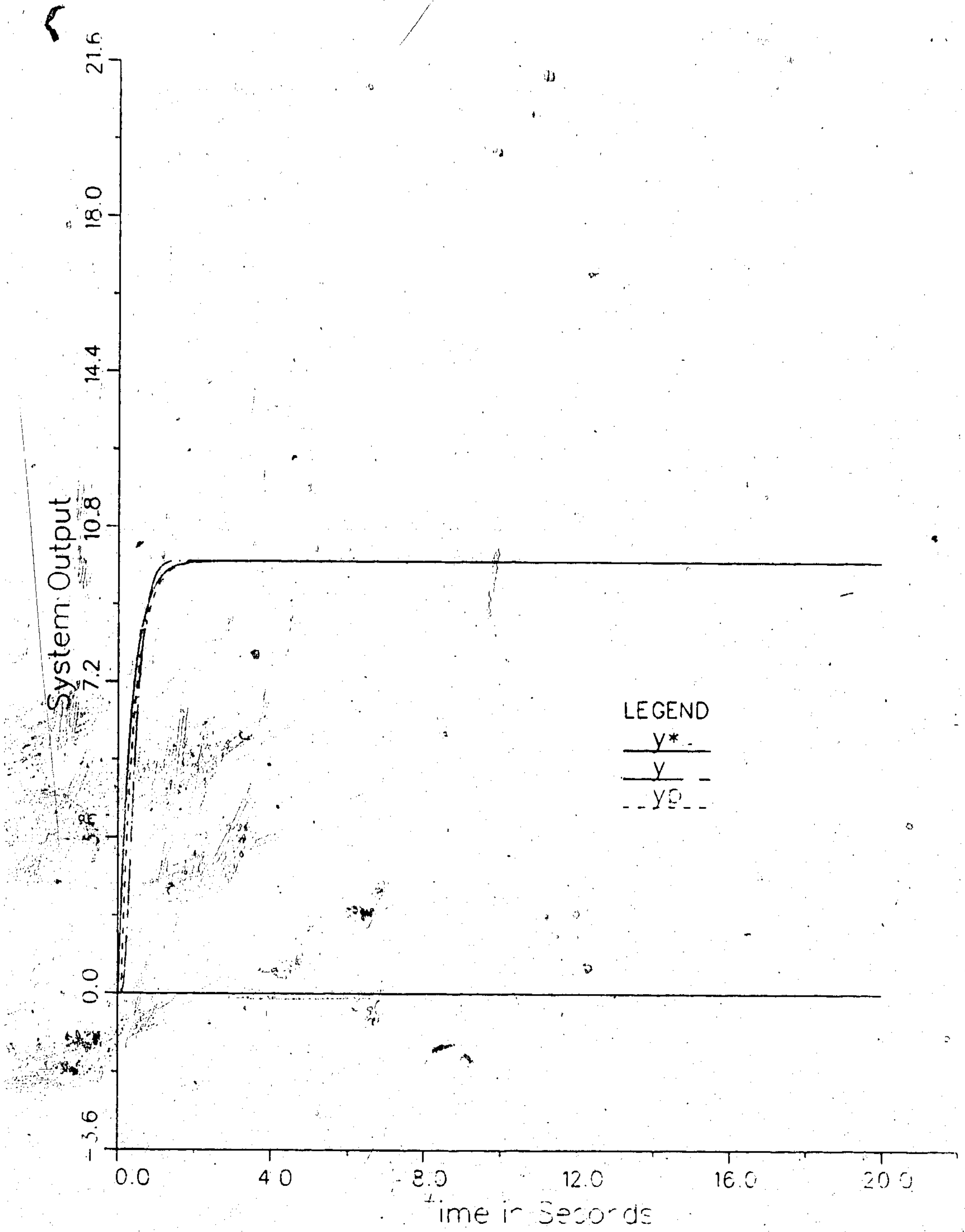


Fig. 2.4 Response for A2, r=10.0

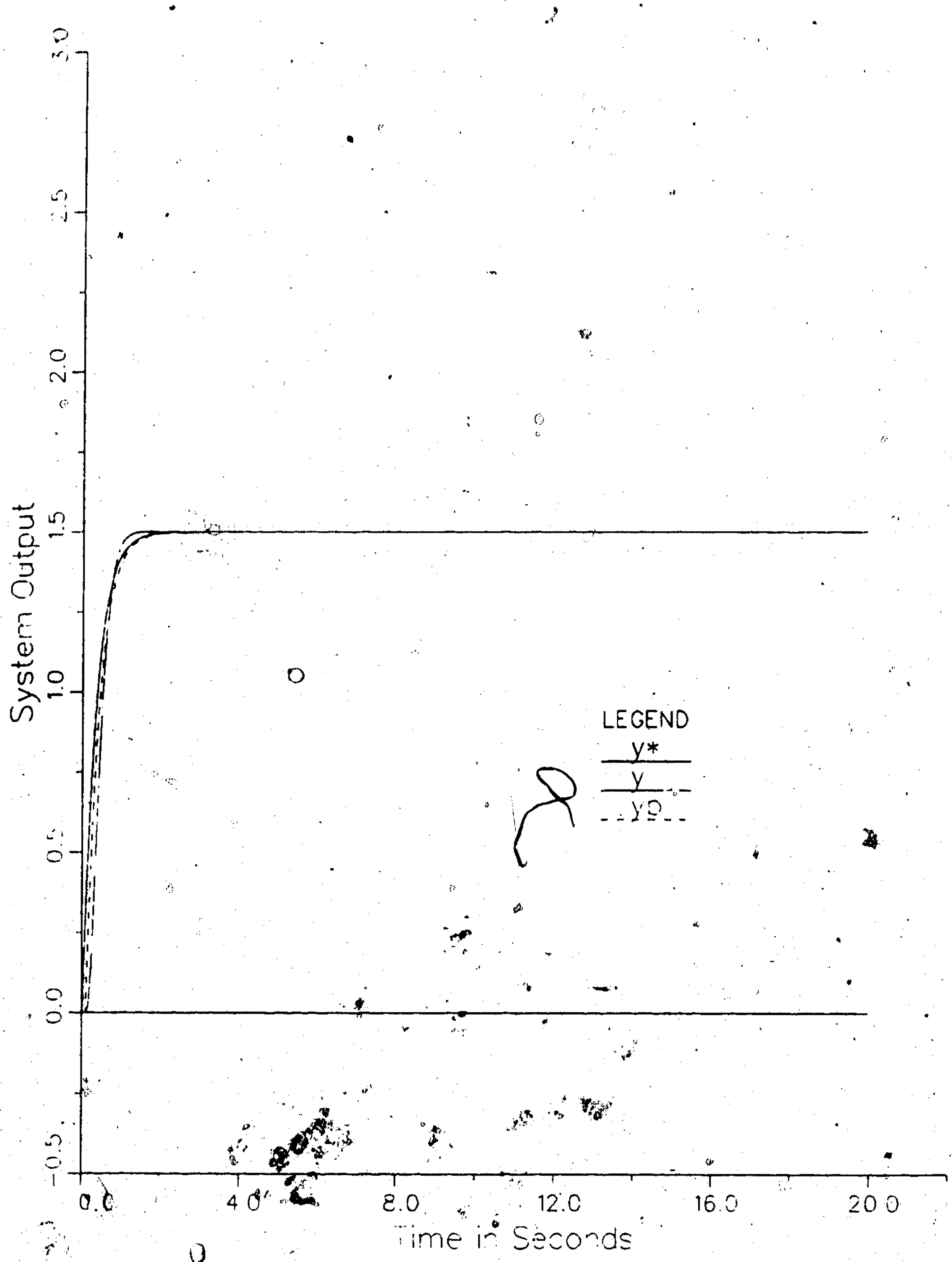


Fig. 2.5 Response for A2, r=1.5

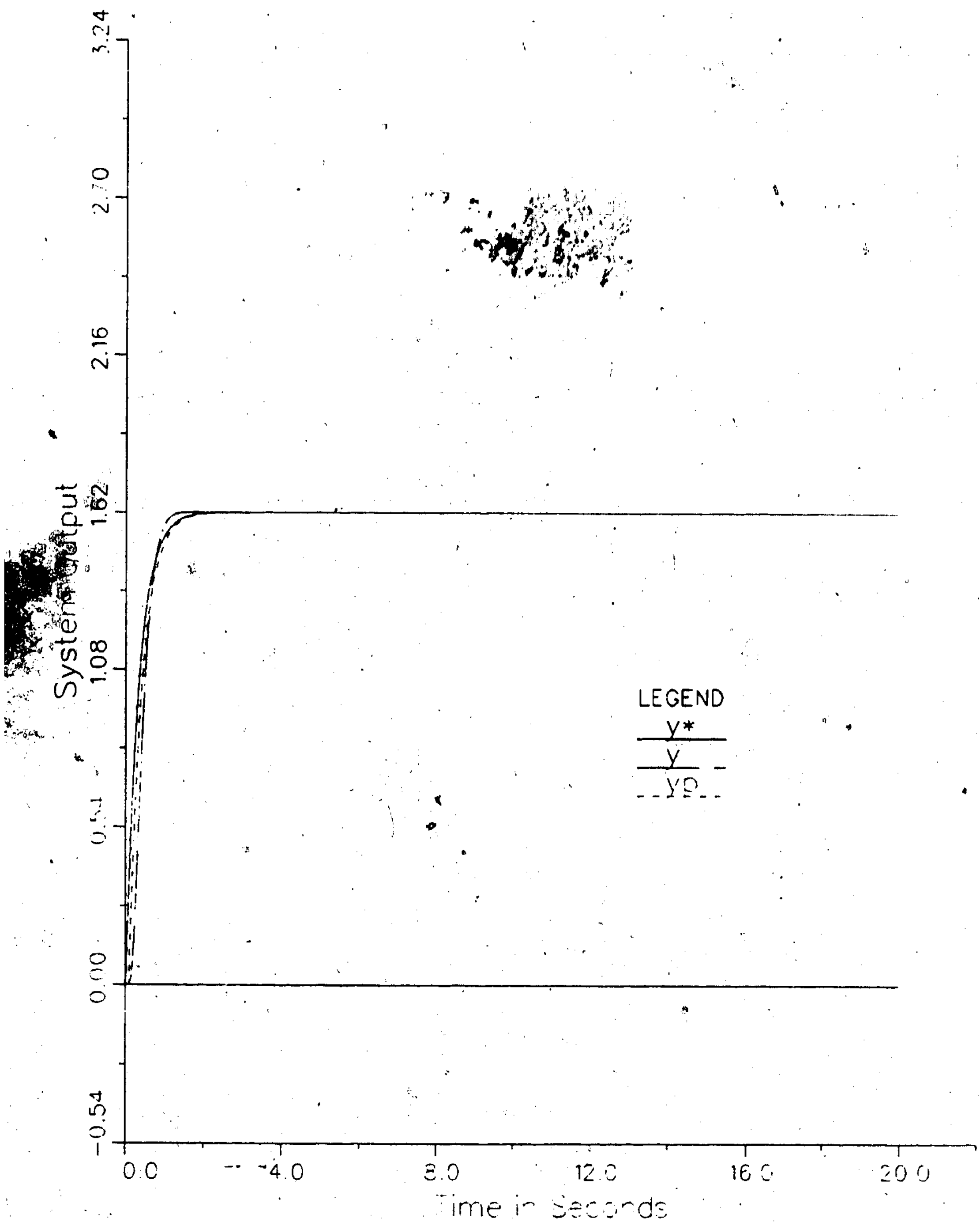


Fig. 2.6 Response for A2, r=1.62

caused by the presence of unmodeled dynamics pose problems to the estimator part of the algorithm. The parameter estimates are naturally different if the plant contains unmodeled dynamics, hence the difference in the control gains.

When no unmodeled dynamics are present, the controller gains have the same steady state values no matter what the initial conditions, the reference input as well as factor f are. These steady state gains correspond with the equivalent constant gains in algorithm A2.

When unmodeled dynamics are present, the gains are adjusted differently, ending with larger absolute values and sometimes adjusted towards infinity. This "tuning to infinity" action can definitely be attributed to the gain adjustment mechanism of A1, since nothing of the sort occured with A2. Hence the result of the study of this section supports Rohrs' conclusions that the presence of unmodeled dynamics in an adaptive control system pose stability problems because the adaptive mechanism tends to adjust controller gains to infinity.

2.3 Effects of Different Unmodeled Dynamics on System Performance

Having established the fact that the presence of unmodeled dynamics in an adaptive control system causes instability, the question of the degree of instability in relation to the position of the unmodeled poles becomes interesting. In this section simulations are done with (1.3.10) as plant model and no arbitrary confinement is imposed on the unmodeled poles as was done in the previous section.

By changing the position of the complex pole pair of (1.3.10) and observing the resulting change of system performance under algorithm A1, a broader picture of how different unmodeled dynamics affect system performance including stability could be obtained. For this purpose, some yardstick or index to measure and compare effects on system behavior of individual unmodeled pole positions is desired.

An accepted measure of system stability is the value of the linear quadratic cost functional of the Lyapunov type:

$$J' = \int_{t_0}^{\infty} \underline{x}^T(t) P \underline{x}(t) dt \qquad (2.2.1)$$

where $\underline{x}(t)$ is the state variable vector and P a positive definite matrix.

A finite value of J' ensures stability. With stable systems, the degree of stability is measured quantitatively by J'. The relative stability of a system decreases with increasing values of J'.

In the simulations of this section, the plant model (1.3.10) is used and t_0 is set equal to zero without loss of generality. The plant is supposed to be dominated by its real pole, and displays near first-order characteristics (otherwise the complex poles would have significant effects on plant behavior). Hence in analyzing system stability, one state variable, ie., the output y(t), could be used in place of $\underline{x}(t)$ as in (2.2.1).

In this section, we not only want to measure and compare the stability of systems, but to isolate the effect of unmodeled dynamics as well. Hence a quantity which measures the instability caused by *the addition of unmodeled dynamics into the system* is needed. We are also interested to the system behaviors as shown by its output (eg. how far away it is from the desired output as exhibited by the output of a plant containing no unmodeled dynamics under the same conditions). The idea of comparing the performance of systems both with and without unmodeled dynamics has been introduced in the previous section. The same approach is used in this section.

Define the quantity

$$J=\int_0^{\infty}[y(t)-y_p(t)]^2dt \qquad (2.2.2)$$

as the functional for the above mentioned purpose.

Clearly, this quantity measures the deviation of output caused by the addition of the unmodeled poles. It is also in the Lyapunov quadratic form (P is the identity matrix). Since the system without unmodeled dynamics is always stable (proved in [1-3]) and both y(t) and $y_p(t)$ asymptotically go to $y^*(t)$ if y(t) is also stable, this quantity could be regarded as a measure of stability similar to J' in (2.2.1) as well.

As stated above, in plant form (1.3.10), the real pole at s=-a is dominant. This, however, does not prevent the non-dominant unmodeled poles from being moved around. In real world situations these poles could be anywhere so long as they do not compete with the real pole for dominance (if they do, it is unlikely that the designer of the control system would mistakenly decide that the plant is of lower order). Hence an attempt is made in this section to move them in simulations and measure the corresponding changes in J to get more insight into the effect of pole positions on the performance of algorithm A1.

As long as they are not too close to the s-plane origin to challenge the dominance of the real pole, the unmodeled poles could be positioned relatively far or near, and different magnitudes of effect on system behavior would result. It is surmised that the nearer these poles are to the s-plane origin the more effect they will have. Whether it is true will be checked in the following simulations.

Another way of measuring the effects of unmodeled dynamics on the system would be to observe how much the system gains change to accomodate them. As Rohrs has noted in his simulations, the gain adjustment mechanism of algorithm A1 tends to tune the gains to large values when unmodeled dynamics is present. He also noted the fact that this trend will develop as the effect of unmodeled dynamics gets bigger and eventually system will become unstable as the gains are tuned to infinity. The simulations in the previous section supported his views. They also offer comparisons with systems without unmodeled dynamics. From the results of these simulations it could be seen that for the system to be stable the gains have to reach a steady state, and the steady state values of system gains with and without unmodeled dynamics present are quite different. This difference is obviously caused by the addition of unmodeled dynamics into the system. And it offers another good measure of its effects.

We set up another quadratic quantity

$$D=0.5\cdot\lim_{t\to\infty}\left[(k_r(t)-k_{pr}(t))^2+(k_y(t)-k_{py}(t))^2\right] \qquad (2.2.3)$$

which is the average quadratic deviation of gains caused by adding unmodeled dynamics into the system.

D, like J, represents the effect on system behavior of a particular pair of unmodeled poles. Especially, its value indicates the extent to which the presence of unmodeled dynamics in the plant affects the parameter adjustment mechanism of the adaptive control algorithm A1. By carrying out simulations with changing unmodeled pole positions covering an area of the s-plane and observing the change of these quantities, a broad picture of unmodeled dynamics effects on system performance could be formed.

2.3.1 Simulation Results with J and D vs. Separation Ratio

Simulations of this section will be concentrated in those unmodeled dynamics pole positions that keep the system stable, for otherwise both J and D will be rendered meaningless. Preliminary tests show that the pole range in Figure 2.7 is appropriate (the system remains stable under our simulation conditions when the unmodeled dynamics poles are in this range). The unmodeled pole pair will be moved around within this range for the simulations that follows. (It should be noted that there are unmodeled pole positions outside this region which do not cause the system to be unstable. However everywhere inside this region the system is stable according to results of preliminary tests and this will be confirmed by the finite performance measure values shown in later simulation results.)

In these simulations, the values of system parameters that are not changed are the same as those used by Rohrs for his simulation in Figure 5-7. of his thesis [4]. Following is a list of these parameters and their values:

$k_y(0)=k_{py}(0)=-0.8$

$k_r(0)=k_{pr}(0)=1.32$

Sampling period for adjustment and control T=0.04(sec)

Reference input r=10.0

Adjustment factor f=0.2

Plant dominant pole a=1

The relative significance of one group of system poles as opposed to another group is usually loosely measured by the "separation ratio" (SR), which is defined as the norm of the most significant pole in the unmodeled group divided by that of the least significant pole in the modeled group. In the (1.3.10) plant used here, this is as follows:

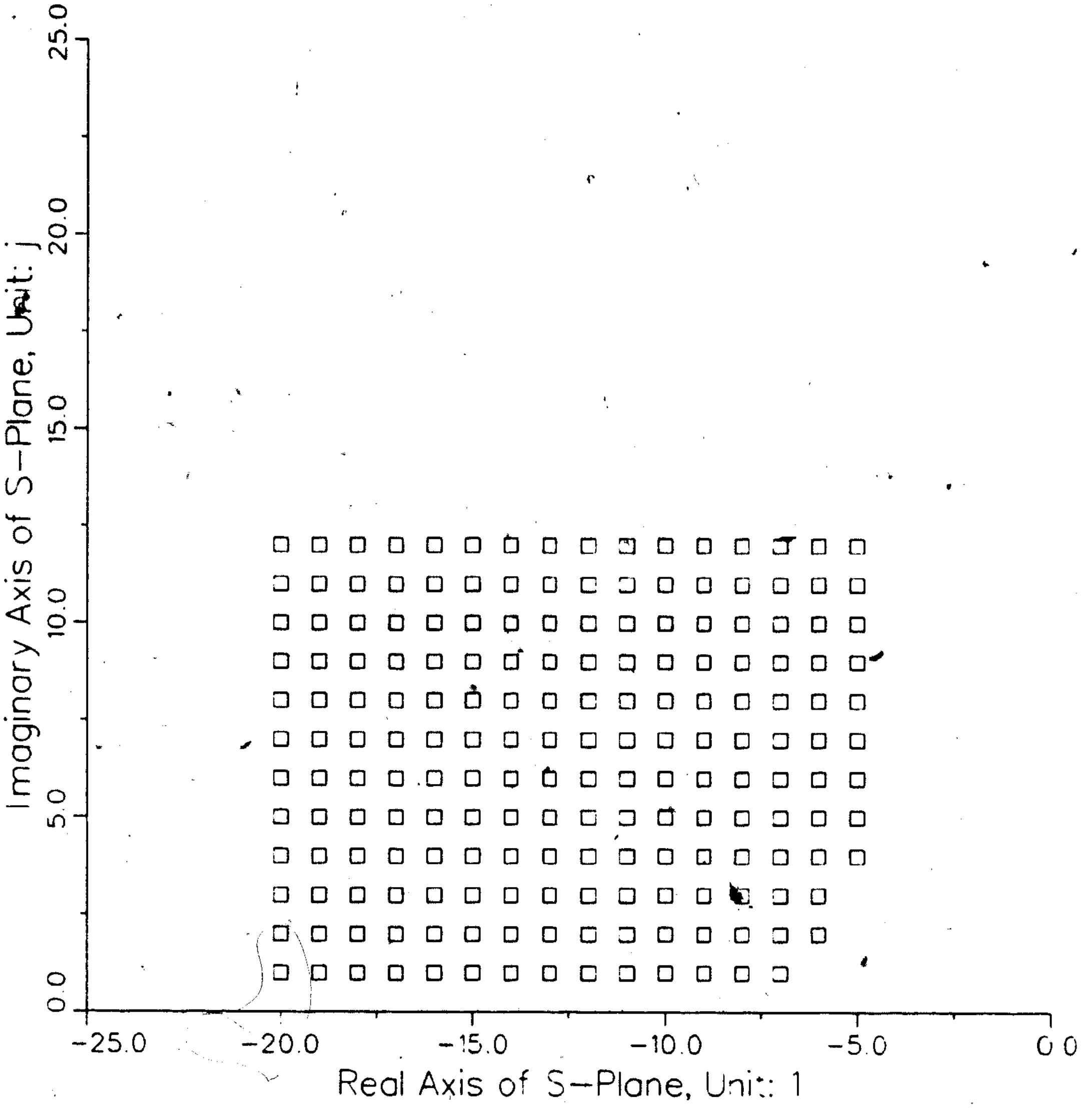


Fig. 2.7 The Distribution Range of U.D. Poles, I.

$$SR=\sqrt{\alpha^2+\beta^2}/a \qquad (2.2.4)$$

It is generally accepted that normally, the larger the SR value (hence the greater physical separation of the two group of poles on the s-plane), the more dominant the dominant group of poles become, while the other group becomes less significant. In our case it is reasonable to expect that with larger SR values the unmodeled dynamics will affect the system performance less. Whether this is true will be answered by the results of simulations that follow.

In these simulations, the performance of the non-adaptive algorithm of A2 are presented as reference comparisons for the same reason for which they were included in the previous section. Figure 2.8 shows the results of J changes with unmodeled dynamics changes by plotting it versus SR. Figure 2.9 does the same with D.

From the results of these simulations, it is clear that generally, the greater the separation ratio, the less effect the unmodeled pole pair has on system performance. However, the plotted results cover an area, not forming a curve, which indicates that there is no one to one relationship between the SR value and J or D. Also, the sharp fall of J and D plots while SR becomes larger suggests movement of the unmodeled poles is relatively free without causing much

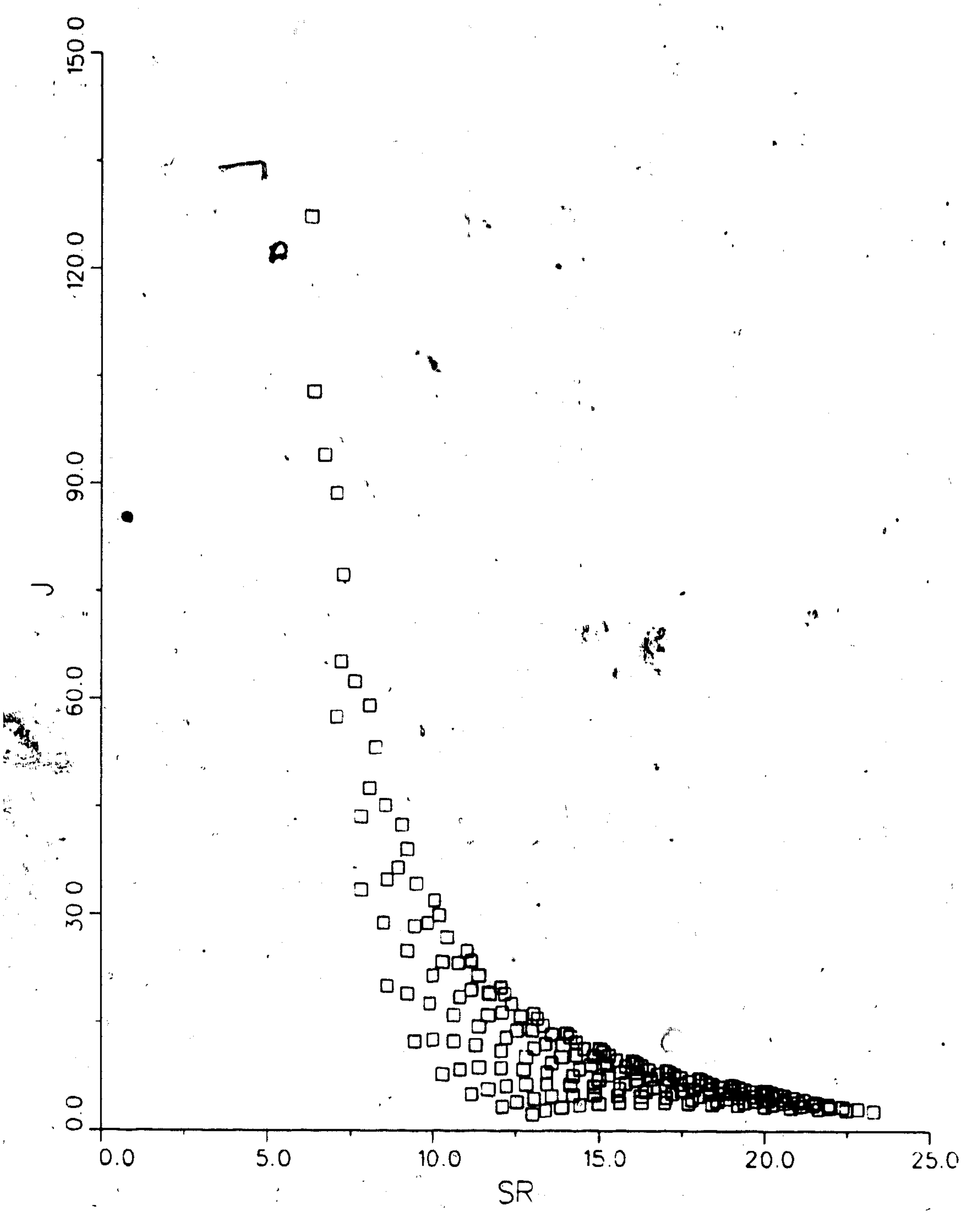


Fig. 2.8 Performance of A1 for Different U.D., I.

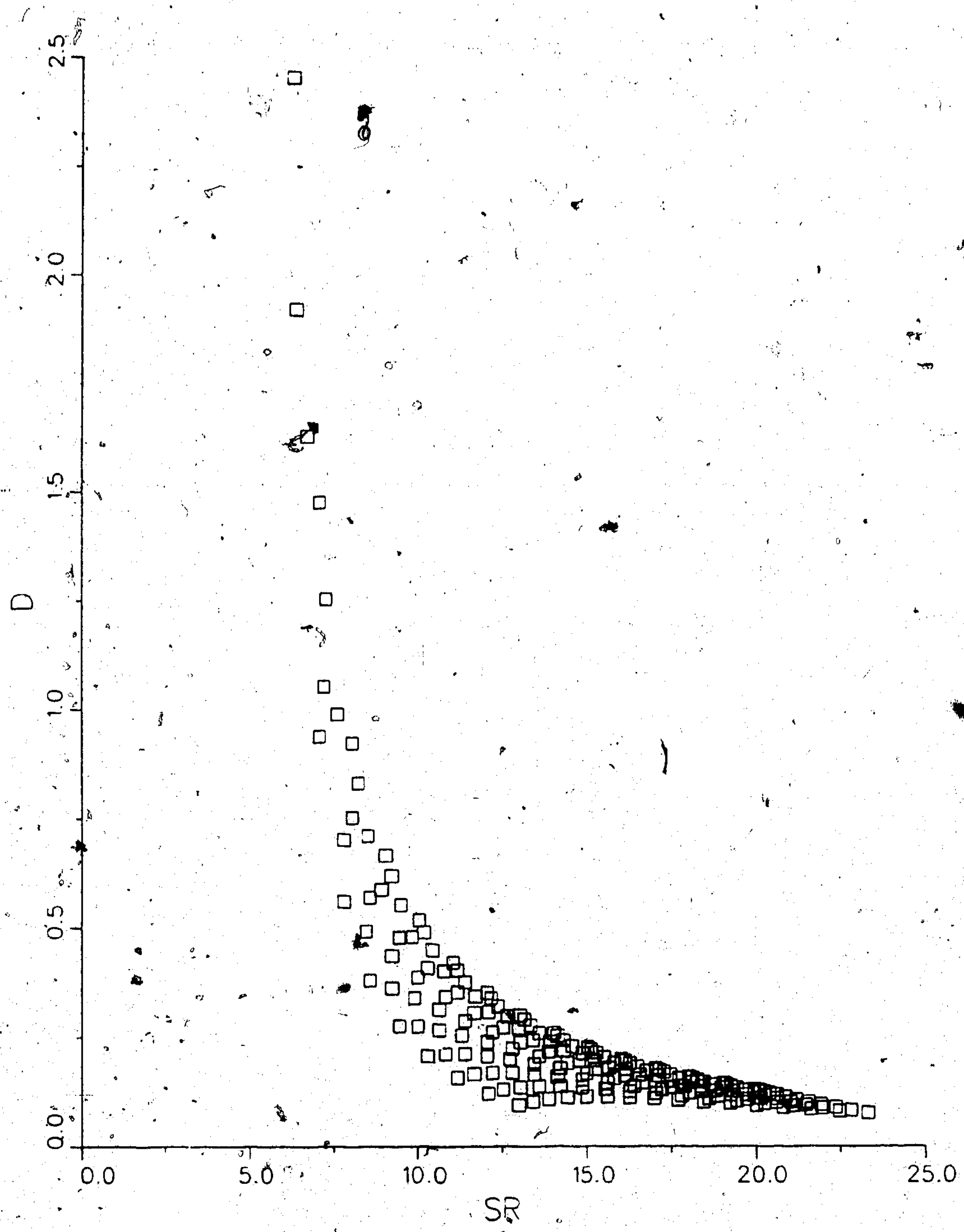


Fig. 2.9 Performancs of A1 for Different U.D., II.

change in the system performance provided the separation ratio is kept at larger than some critical value. Once this value has been reached, however, the system performance deteriorates rapidly. Under the conditions of this simulation, the critical value of SR seemed to be between 10 and 15. Both values J and D behaved in a very similar fasion, supporting the view that it is the deviation on system gains caused by the unmodeled dynamics that causes the instability.

For the particular positions of unmodeled poles which happened to have the same SR value, it is a rule that those farther away from the real axis of the s-plane have less effects on the system behavior. For instance, poles $-5\pm j9$ and $-9\pm j5$ both have SR value of 10.29, but result in very different J values of 7.8 and 23.4 respectively; poles $-6\pm j8$, $-8\pm j6$ and $-10\pm j1$ all have SR value of 10 or so, but with J values of 12.6, 21.5 and 32. The D values behave in a similar pattern. This does show that SR is not an ideal measure of the relative importance of the unmodeled poles in the adaptive control system.

2.3.2 Simulation Results with J and D vs. Plant Deviations

Adaptive control algorithms are used in situations where the plant models are not known. Although the separation ratio has been used rather widely to indicate the relative dominance of the dynamics in a system, it is often not possible to estimate its value without the knowledge of the plant dynamics. Hence as a guide to estimate the potential adverse effect of the presence of unmodeled dynamics in an adaptive control system the SR value is of limited usefulness.

The procedure of designing an adaptive control algorithm for a system always involves the determination of the probable plant order. This is done either from experience or by simple response tests. A plant whose open-loop response is not far from that of a known model (in our case the presumed plant model with $-a$ as its single pole), is estimated as of that order. And the design of the algorithm is based on this estimate of plant order.

In the case of (1.3.10), its open-loop performance approximates that of

$$(s+a)Y(s)=2U(s) \tag{2.2.5}$$

if the real pole at $-a$ is dominant.

If the unmodeled poles change position, the (1.3.10) open-loop response to a step input signal will also change.

By observing this response and measuring its deviation from the response of (2.2.5) to the same signal, we could get a good idea as to how much the addition of a particular unmodeled pole pair affects the plant behavior. This gives us an alternative to the separation ratio as a measure of the effects of certain unmodeled pole positions.

Define the open-loop deviation as

$$J_o=\int_0^{\infty}[y_o(t)-y_{op}(t)]^2dt \qquad (2.2.6)$$

where $y_o(t)$ is the open-loop response of (1.3.10) to a step input and $y_{op}(t)$ is the same response of (2.2.5).

In the following simulations, the performance values J and D are compared with the value J_o, using it as a substitute for the separation ratio used before. An advantage of this value is that it could be obtained before the controller is designed by subjecting the plant to open-loop step input and comparing it with the nearest first order response. The disadvantage however is it changes with the magnitude of the signal used, unlike the SR value which is fixed for a particular pole position. Here a step input of 5 is used to bring the final value of $y_o(t)$ to 10, same as y(t)'s final value in the simulations (normalization procedure). Other factors used are the same as in Subsection.

2.2.2.. The results of these simulations are shown in Figures 2.10 and 2.11

It is clear that the system performance, measured by J and D, deteriorated with increase of J_o, the deviation shown by the plant with the addition of the unmodeled poles. Generally speaking, especially when the J_o value is low, the further the (1.3.10) response is from that of (2.2.5) the more effect the unmodeled dynamics has on system performance. However, with higher J_o values, the final system performance are quite different from one another depending where the poles are even if they have the same J_o value.

For instance, pole positions $-5\pm j6$ and $-10\pm j4$ both have J_o values of nearly 1.50, but turn out very different J values of 33.4 and 23.3 respectively. Pole positions $-5\pm j5$ and $-9\pm j3$ all have J_o values near 2.00, but give J values of 57.5 and 34.3 respectively. It is a rule that for the same J_o value, the poles that have smaller separation ratios have more effect on the system performance. The D measure behaves similarly as that of J.

2.3.3 Discussion

From the simulation results accumulated in this section, we have obtained a general picture of the effects

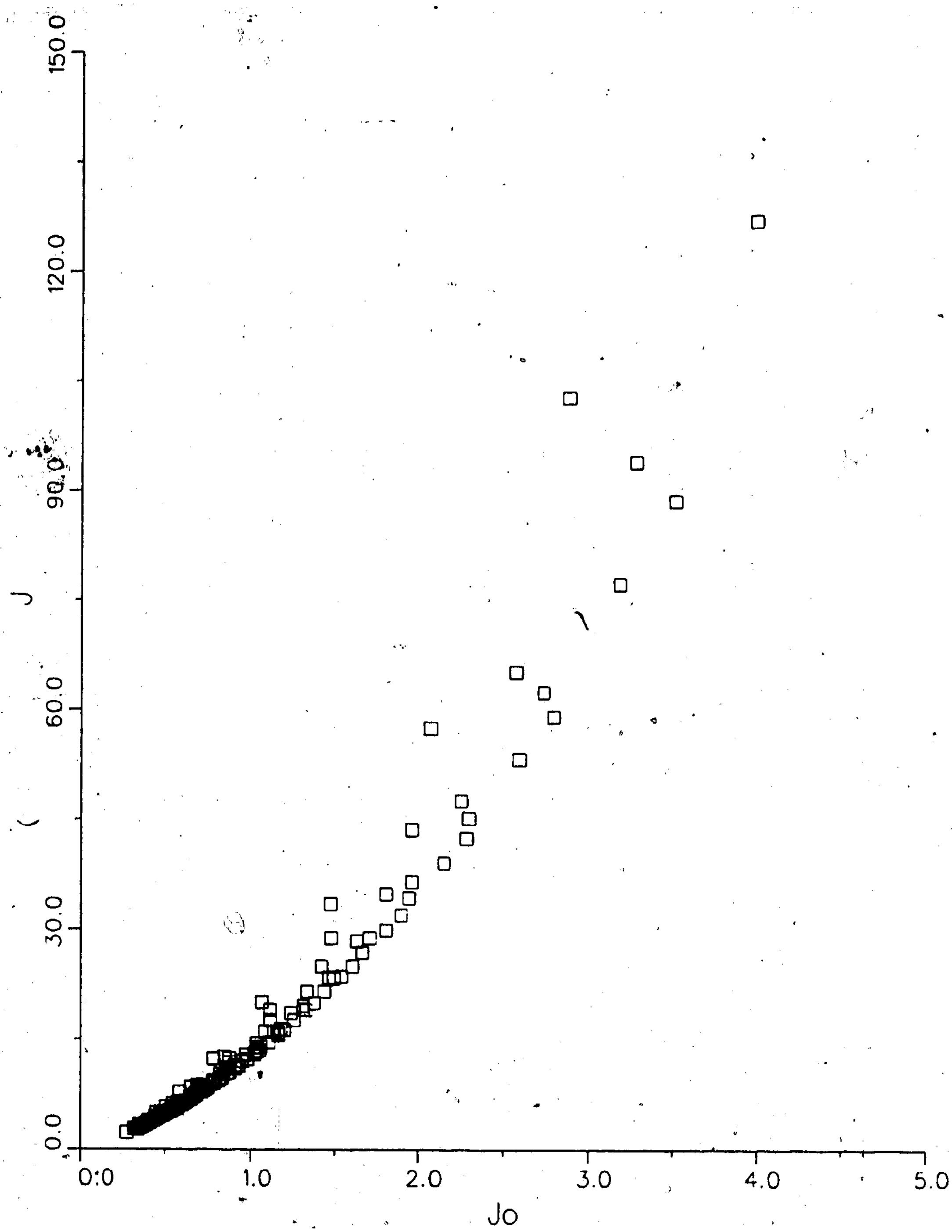


Fig. 2.10 Performance of A1 for Different U.D., M.

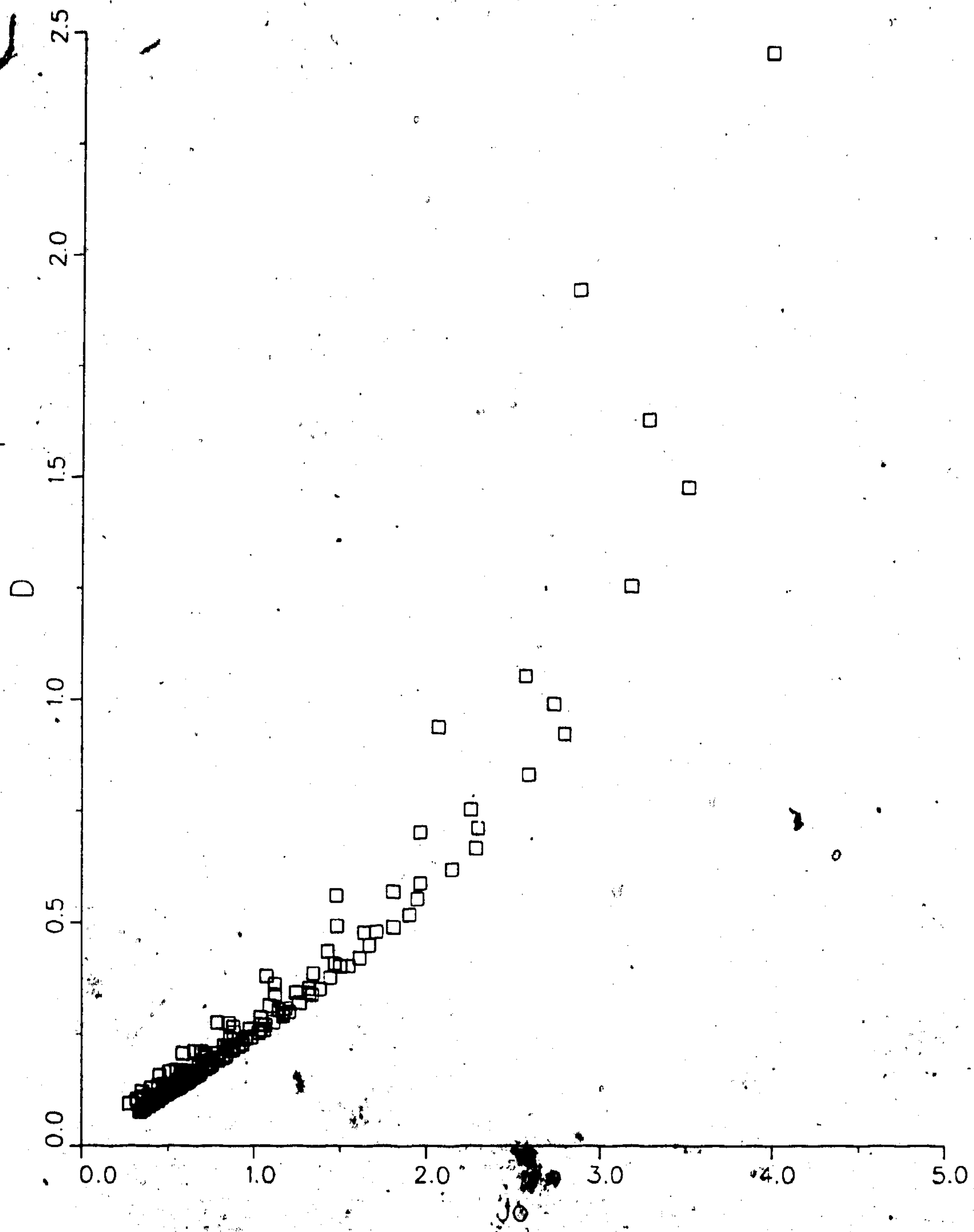


Fig 2.11 Performance of A1 for Different U.D., IV.

on system performance by different unmodeled pole positions. By keeping everything else identical, these results show what the changing of the position of the unmodeled poles do to system performance as measured by quantities D and J.

These quantitative measures J and D enable us to study the trend and direction of change. Without this type of study, one could only speculate on the effects of unmodeled dynamics on system performance in a general qualitative way. Now we are in a position to say with some confidence that in a system with a pair of high frequency complex poles as unmodeled dynamics and a real pole as the dominant dynamics, the less dominant the real pole is, the more effect the unmodeled dynamics would have on system performance. This effect always deteriorates system performance and potentially could make system unstable for all the cases investigated so far. The relative dominance of the real pole could be related to the separation ratio and/or the deviation of the plant response to a step input. Depending on the availability of a particular set of data, one could set up a series of simulations before hand to decide whether the unmodeled dynamics will affect the system performance enough to warrant an increase of controller order. With the same separation ratio, the effect on system performance increases with the unmodeled poles approaching the real axis. While unmodeled poles far apart and with much different separation ratio values could result in the same amount of deviation on the plant open-loop response, they will affect system performance quite differently if their effects are serious enough to be felt. In this situation, the smaller the separation ratio, the larger will be the adverse effect of the unmodeled dynamics. When the deviation of the plant open-loop response is not large, it is safe to say the larger this deviation, greater will be the effect of the unmodeled dynamics on the overall adaptive control system.

Lastly, since two different quantities are used to measure the effect of unmodeled dynamics, and all the simulation results so far point to the fact that they interrelate closely, it is useful to display their relationship. The final deviation of the system gains mark the effect of the unmodeled dynamics on the adjustment mechanism. The suggestions are that this erratical gain tuning finally causes the instability. The simulations so far supported this speculation and in Figure 2.12 the quantity J, which measures system stability, and D, which measures how affected the gain adjustment mechanism is, are plotted versus each other.

In Figure 2.12 the points are virtually lined up in a curve, which means as a rule it is true that the greater D is, the greater J would be. In other words, the two values chosen to measure the effect of unmodeled dynamics in our

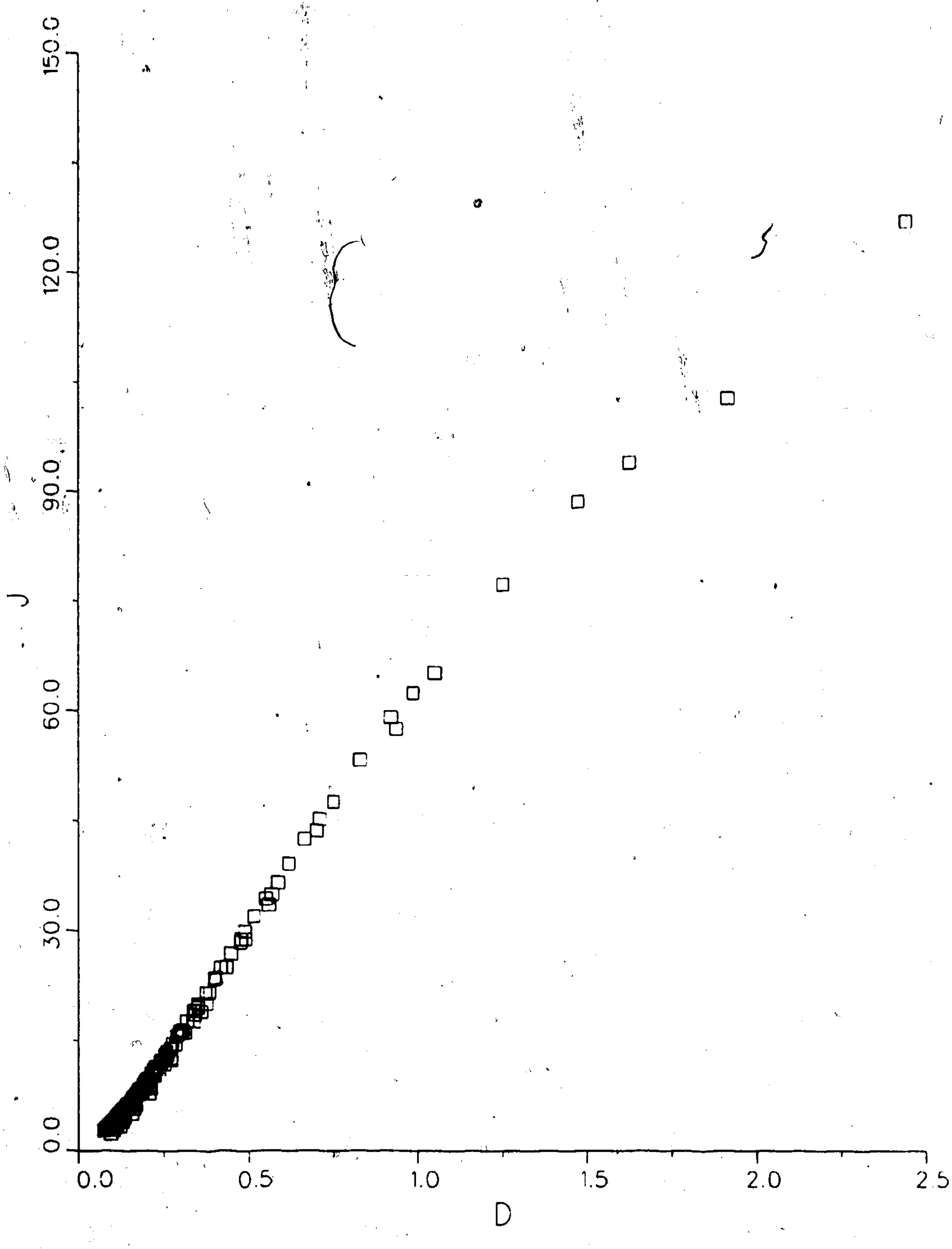


Fig. 2.12 Relationship of the Two Measures D and J

system are identical in their effectiveness. Also, it proves that it is true that the further away the system gains are tuned as a result of the presence of unmodeled dynamics, the more unstable the system becomes. Hence it is the same as saying that the presence of unmodeled dynamics causes the controller gains to be adjusted to larger absolute values than they should be as to say it causes instability.

2.4 Effects of the Adjustment and Control Interval

2.4.1 Introduction

Rohrs did some exploration work on the effect of varying the adjustment and control interval on system performance in an adaptive control system with unmodeled dynamics present. The general conclusion obtained by him suggests that for a given system, with every other factor kept constant, the longer this interval T is, the more stable the system. This is, in effect, to say that with unmodeled dynamics present, more frequent adjustments of control parameters result in more instabilities.

This point of view is easily understood because unmodeled dynamics are composed of much faster dynamics than the dominant poles. In Rohrs' case, the unmodeled pole pair is at $-15\pm j2$ as opposed to -1 of the dominant pole. In this situation, a T large enough (hence infrequent sampling) could in effect block out information from the response of unmodeled dynamics while still being fast enough for the estimator to do a good job on the slower dominant dynamics. This scheme will to some extent prevent erratical adjustments of controller parameters as a result of the estimator's taking unmodeled dynamics as part of the lower order system.

Rohrs tried T values of 0.04sec. and 0.4sec. in his analysis and simulations. He found that system performance was vastly improved when the latter value is used. Recalling plant dynamics in model (1.3.9) we note that the fast unmodeled poles are at $-15\pm j2$, which are definitely not reconstructible when a sampling period of 0.4sec. is used (in 0.4sec. only 0.25% of its original response is left).

Unfortunately, Rohrs used the same numerical values for T and the plant sampling period T_p in his simulations and analysis. By choosing a large value of T, he has made the sampling period too long to represent faithfully the unmodeled dynamics poles of the plant. The resultant models could not be regarded as good enough for simulation purposes.

As stated above in the introduction of this chapter, in simulations the sampling period should be small enough to bring the information on all the plant dynamics out. Otherwise, the unmodeled dynamics will be blocked out from the model of the plant being simulated and the whole process will become pointless.

The sampling period used in the discretization of the plant model, T_p, is chosen to be 0.01sec. in this thesis. This period is small enough for the resultant models to satisfactorily represent all the plant dynamics encountered here. The uniformity of the plant model is also kept when the control interval T is changed.

The analysis and simulations in [4] on the effects of different adjustment and control intervals, while comprehensive, did not provide a complete picture on how different unmodeled pole positions react to changes of T. It also contained only two different T values and did not provide any information of how the performance change as T changes continuously. Moreover, the accuracy of the simulations of T=0.4 is not satisfactory. The conclusion about larger T's bring better stability was obtained without concrete evidence (eg. what if at T=0.2 the performance should be better than at T=0.4?).

2.4.2 The Simulation Results for J and D values

In the following simulations, plants of the (1.3.10) type with different unmodeled pole positions are subjected to control algorithm A1 with gradually changing T, and the resultant performance values J and D observed.

In these simulations, the adjustment and control period T is changed from 0.04 to 0.4 in small increments (0.01), while the resultant performance values J and D observed. The plant unmodeled dynamics are also changed to observe whether the method of using as large a T as possible has the same effectiveness for different unmodeled pole positions.

First of all, three different unmodeled pole positions are chosen for this simulation: $-5\pm j7$, $-10\pm j5$ and $-12\pm j1$, because at T=0.04, they all result values of J and D around 20.0 and 0.36 respectively. If the simulations also show that the effects of changing T on D and J are similar for these different plants, it could be said that regardless of pole positions, the change in the value of T has the same effect on system performance as long as the different unmodeled dynamics have the same effect under one T. Figures 2.13 and 2.14 show the J and D plots respectively.

From these results, we could say that indeed, the systems respond similarly to changes in T. The plots of D are on top of one another, while there are minor differences among the J plots. Here the conclusion is that different

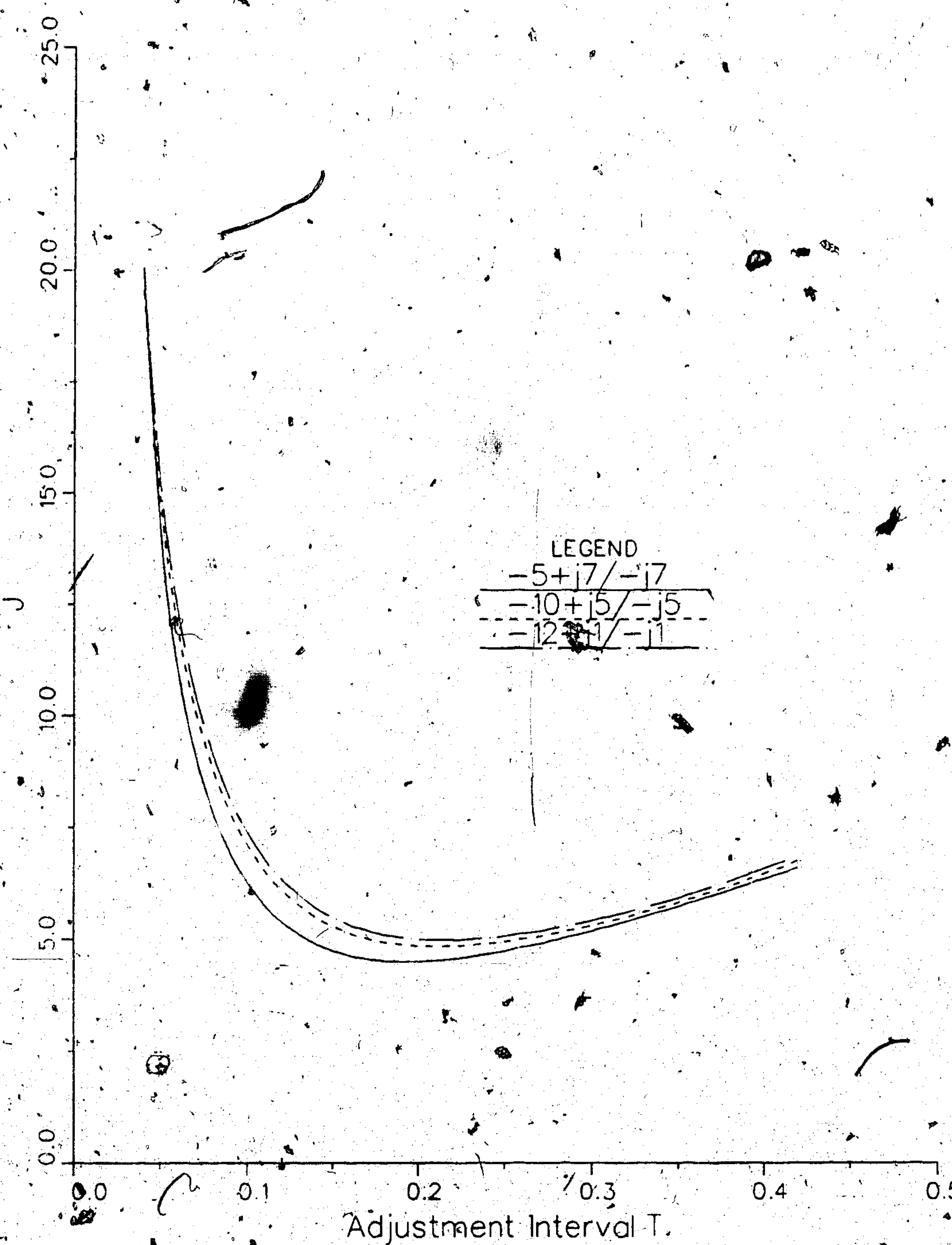


Fig. 2.13 Change of System Performance with T. I.

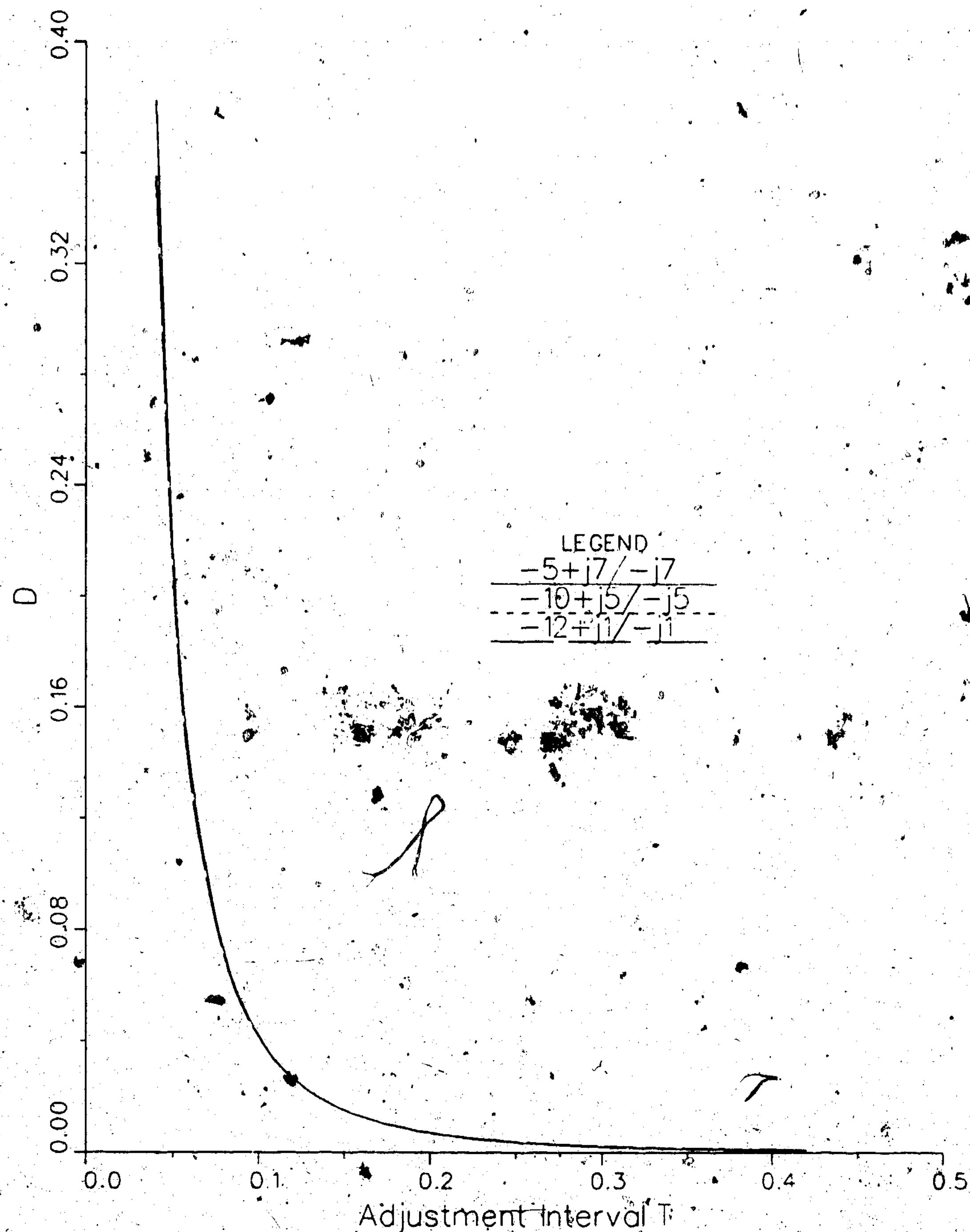


Fig. 2.14 Change of System Performance with T, II.

unmodeled pole positions will have similar change patterns of their performance measures with changes in T, provided that for one value of T, their D and J values coincide.

An interesting aspect of these simulation results is the sharp fall of D and J values with initial increase of T. This shows that the effectiveness of the method of seeking stability by increasing T (recommended as a design guideline by Rohrs in [4-4b]) is not uniform. For a certain plant, a certain controller factor set, input level, etc., the increase of T will greatly increase system stability to some degree. After T has been increased to a certain level, however, any further increase would not be as effective. There seems to be a 'saturation point' for the effectiveness of increasing T. After this point, the system performance as measured by J actually deteriorates slowly, owing to the fact that T has become too large for the estimator to do a good job on the dominant dynamics of the system, although the D value continues to decrease.

Figures 2.15 and 2.16 show the results of simulations carried out with another group of pole positions, namely $-5\pm j6$, $-9\pm j7$ and $-12\pm j1$. T is again changed in increments of 0.01 and the same saturation in effectiveness of its increase is noted.

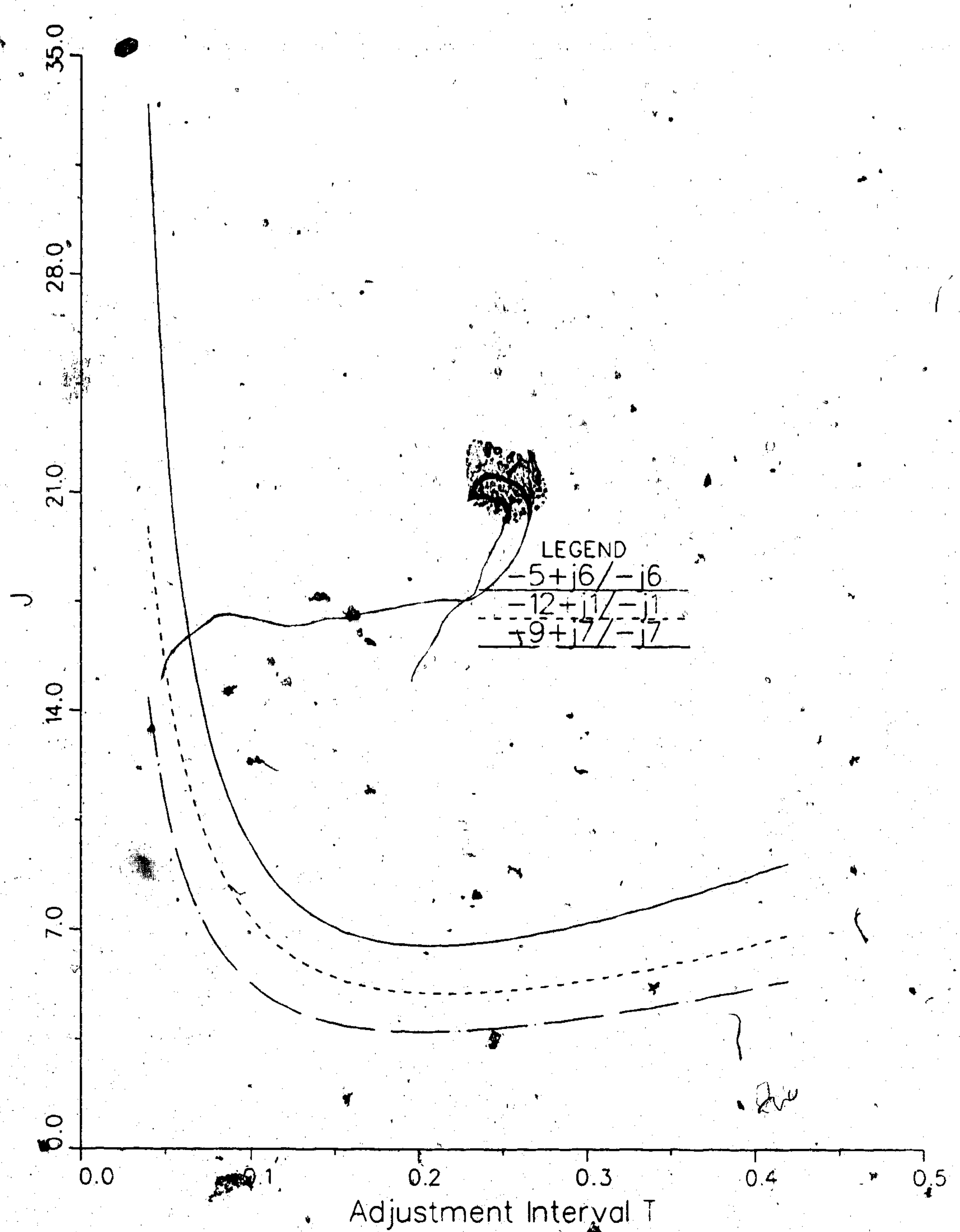


Fig. 2.15 Change of System Performance with T, III

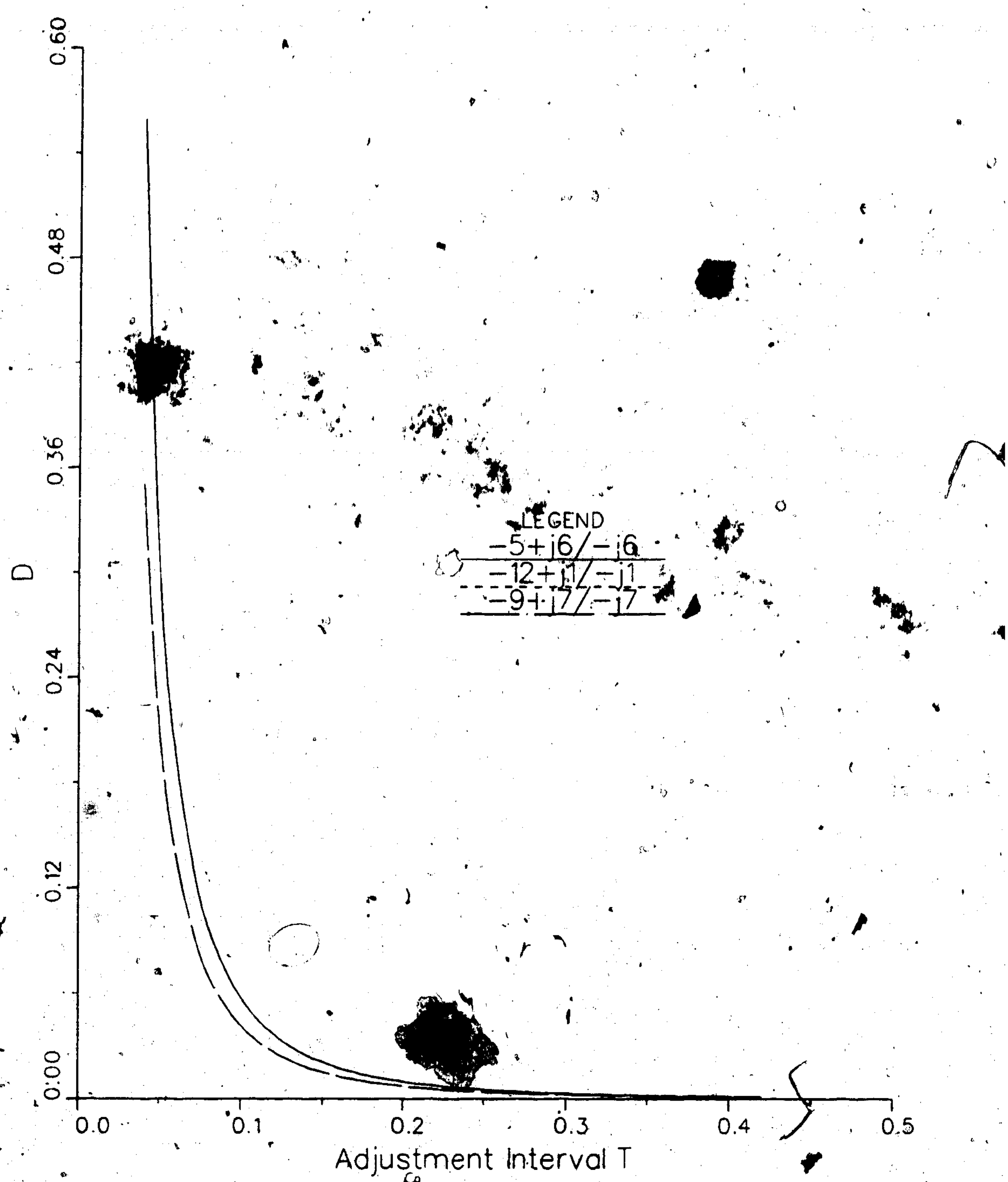


Fig. 2.16 Change of System Performance with T, IV.

With the increase of T the D value decreases rapidly for a while and then slow down when it approaches 0. At the same time the J value falls sharply with initial increase of T but start to increase slowly after a certain point is reached. This seems to be true no matter where the unmodeled poles are positioned.

2.4.3 The Effectiveness of Increasing T as a [illegible] improving Stability

As noted above, when the adjustment and control interval T is increased, the system performance improves markedly up to a point. After that, however, this trend stops and any further increase in T does not result in further improvement of system performance.

This characteristic is more clearly demonstrated with the plots shown in Figures 2.17 and 2.18. Here the derivatives of the variables in Figures 2.15 and 2.16 are shown versus T. These new variables measure the speed of performance improvement. For instance, when T is greater than 0.2 the value dD/dT is approaching 0, which means after T is increased to 0.2 the system performance is about improved to its limit. This agrees with what is shown on the dJ/dT graph. There all the curves pass cross the zero value at around T=0.2, which means J reaches its lowest value at this point.

On the other hand, when T is relatively small, both derivatives have very small negative values, which means that the value of the performance measure decreases very rapidly with increasing T. This is the range in which the recomended design guideline of using relatively larger T's to insure stability is the most efficient.

Finally, the changing T does have different degrees of effectiveness with different unmodeled pole positions. For instance, when T is at a value of 0.65, the value of dD/dT is at -6 for the unmodeled dynamics position of -5±j6, -4.1 for -12±j1 and -3.4 for -9±j7, which means at T=0.65 the further increase of T is nearly twice as effective in improving system performance for the first pole position comppared to the last pole position.

Next we seek to compare two particular unmodeled dynamics positions as related to changes in T. The poles chosen are -15±j2 which is our old standard, and -6±j2 which is the position when the system is on the brink of going unstable at T=0.04. The performance of these two unmodeled pole positions are vastly different at T=0.04, with D values of 0.22 and 2.45 and J values of 11 and 127, respectively. These unmodeled poles also have very different separation ratio values of 15.1 and 6.32. By changing the control

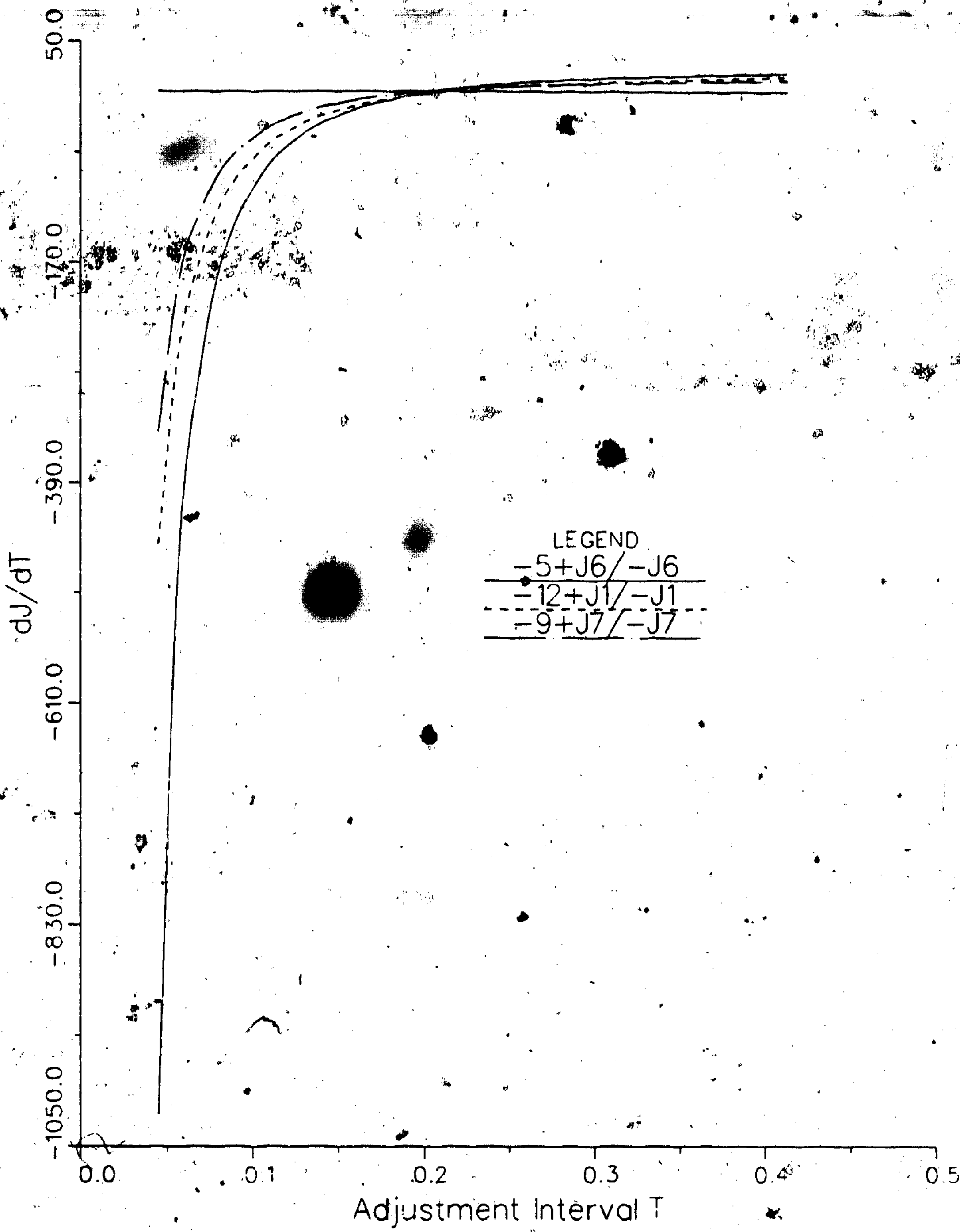


Fig. 2.17 Speed of Performance Improvement vs. T, I.

Fig. 2.18 Speed of Performance Improvement vs. T, II.

interval, the effect of these unmodeled dynamics could be lessened. For the D value to be reduced to 0.1, the pole pair -15±j2 requires T be increased to 0.054, while the poles at -6±2 require it to be 0.11. The lowest value of J for the former pole position is reached at T=0.205 and that the lowest value of J for the latter pole position is reached at T=0.26. It should be noted that the change in the values of T is relatively small, although these two positions of unmodeled poles are far apart. With a separation ratio difference of about 2.5 times, the values of T needed to achieve the best system performance are only 0.055sec. apart, or a 25% difference.

The simulations involving changes in the adjustment and control interval T have yielded some interesting results. The most important among them is the fact that the effectiveness of "increasing T to improve system performance" scheme is by no means uniform. When the effect of unmodeled dynamics on a adaptive control system is relatively large or the instability is high, a little increase of T does a lot of improvement on the system performance. When the unmodeled dynamics do not have a pronounced effect on system performance, however, increasing T does not seem to be as effective. Also another result is that there appears to be a point beyond which the increase of T actually makes the system performance deteriorate. Finally, it is a rule that if with one value of T a particular unmodeled dynamics position affects the system more than another, it will remain so when the T value is changed. Hence the relative effectiveness on system performance of unmodeled pole positions does not change with T.

2.5 Discusions and Conclusions

In this chapter, all the simulations have been conducted with plant model (1.3.10). From the [illegible] of these simulations, several conclusions could be [illegible]. They are:

1. The presence of unmodeled dynamics in a system controlled by adaptive control algorithm A1 degrades system performance.
2. The instability is caused by the parameter adjustment mechanism's tuning system gains to large absolute values in response to the unmodeled dynamics.
3. The less relative dominance the "modeled" dynamics have compared with unmodeled dynamics, the more effect the unmodeled dynamics will have on the system performance.
4. The D value, measure of unmodeled dynamics effect on the parameter adjustment mechanism, has almost

identical relationship with the J value, measure of unmodeled dynamics effect on system performance, supporting conclusion No. 2 above.

5. The design guideline of enhancing stability by choosing large control intervals is not uniformly effective.

6. Different unmodeled pole positions react similarly to changes in T.

These conclusions are based on the simulation work shown here. The simulations of Section 2.1 mainly involved studies of what causes the instability. By excluding one factor then another, it was established that unmodeled dynamics and adaptive control algorithm put together may result in instabilities. In Section 2.2, a comprehensive study of a large number of different unmodeled dynamics and the careful choice of two performance measures enabled a broad picture of how different unmodeled dynamics have different roles to play in such a system. It also helped to verify the cause of the instability factor always present whenever unmodeled dynamics and adaptive control is put together from another angle, as the adjustment mechanism's inclination to tune the gains larger in this situation. In Section 2.3 the simulation results showed the pattern in which system performance changes with T changes. The conclusion could be put in another way: when T is very large, the adaptive control algorithm performed like an non-adaptive algorithm, with the unstable factor introduced into the system by unmodeled dynamics all but non-present; with the decreasing of T, the adaptive algorithm began to adjust controller gains and for a range of T values the system performance is steady, without much increase of unmodeled dynamics effect; and then it comes to a point where with further decreases of T, the unmodeled dynamics effect increases very rapidly. There are apparently two reasons for this behavior. Decreasing the T value while keeping every other factor constant effectively cause the gain adjustment to work faster (f=0.2 with T=.04 and f=0.2 with T=.4 are entirely different). In other words the f value could be regarded as effectively enlarged by the decrease of T. Moreover, smaller values of T permit more information from the unmodeled dynamic poles to reach the gain adjustment mechanism, while larger T's might in effect block out this information.

3. Evaluation of Algorithm A1 in Its Second Order Form

3.1 Introduction

In the previous chapter, the effects of unmodeled dynamics on the algorithm A1 are investigated through simulation studies. In these simulations, algorithm A1 appeared in its first order form, ie., n=1, m=0 and d=1. The third order model (1.3.10) was used as the plant, with the non-dominant complex pair of poles acting as unmodeled dynamics.

These studies, while relatively simple because of the low-order of the systems involved, offer only a limited field of results. A study of the algorithm in a higher order environment will be helpful to clarify certain questions which cannot be answered by a study of first order algorithms. These include the effect of overestimating the plant order while designing the controller, the situation where no group of plant poles can be classified as unmodeled dynamics, etc..

In this chapter, the algorithm A1 will be studied in its second order form, and the plant will follow the model (1.3.11). The consequences of overestimating the plant order in the control law design will be analyzed in Section 3.2..

In 3.3., studies similar to those conducted in the previous chapter, especially in 2.2. and 2.3., are done with second order control law on the plant model (1.3.11), again with the complex pair of poles as unmodeled dynamics. The purpose of these studies is to try to confirm the conclusions reached in Chapter 2, or to see whether these results are true only in a first order situation.

In Section 3.4., we will carry out simulation studies for a specific case, ie., the algorithm A1 will be used in its second order form for a plant of the type used in the previous chapter. The comparisons of the performance of A1 in different orders on the same plant will help resolve the problem of whether increasing control law order improves system performance. In this case, no clear cut classification of unmodeled dynamics could be given to a group of plant poles.

3.2 Higher Order Control Law for Lower Order Plant

3.2.1 General Comments

In Chapter 1., the conditions required for the stability of a system controlled by an algorithm like A1 were introduced. According to these conditions, the exact degree of the plant model polynomials A_M and B_M is not required to be known. It is only necessary that the upper bound of the plant order (ie., the upper bound of the degree of polynomial A_M) as well as the relative degree of the plant be known. If the upper bound of the degree of A_M is N and relative degree is n_1 then the control algorithm should be chosen so that n is an integer at least equal to [illegible]d no less than N, and $m=n-n_1$. The system will then be stable [19, 20]. This does provide mathematical assurance that overestimation of plant order does not affect system stability adversely.

From the simulation studies conducted in the previous chapter, it was concluded that underestimating the plant order may result the degradation of system performance, especially stability. The tendency therefore to apply the control algorithm in the highest order permisible under the operating conditions whenever the plant order could not be exactly predetermined.

However such an approach has its limitations, not the least of which is the cost involved. There also exist types of processes which could not be accurately represented by a linear model. For these "infinite order" systems unmodeled dynamics is always present no matter how high the order of the adaptive control algorithm is set at. There is also the probability of a higher order control algorithm being used on a lower order plant. This is the result of overestimating the plant order.

The simulations in this section are conducted to observe the effect on system performance of overestimating plant order. Comparisons will be made between the performance of algorithms resulting from a correct plant order estimate and an overestimate. The plant being controlled is described by transfer function (2.1.2), reintroduced here as

$$(s+1)Y(s)=2U(s) \tag{3.2.1}$$

Two forms of the control algorithm A1 will be compared, one with n=1, m=0, d=1; and the other with n=2, m=1, d=1.

3.2.2 Simulations with Step Input

Simulations here are conducted with step reference input signals. The results are in Figures 3.1 to 3.4. In each, the responses of the same plant (3.2.1) controlled by the first order algorithm and the second order algorithm are shown together for comparison. The initial values of the control gains are set at zero for all simulations in this section. The second order algorithm has four control gains while the first order alternative has two. All their values are set to zero at the start of the simulations to provide some kind of consistency. Other initial values like those used in the previous chapter for the first order controller are impossible to match in the case of the second order algorithm.

In order to make meaningful comparisons, it is necessary to make both algorithms get the plant to follow the response of the same reference model. The model used in this section is the same one used in the previous chapter, whose transfer function is reintroduced below:

$$(s+3)Y(s)=3U(s) \tag{3.2.2}$$

Strictly speaking, a second order algorithm should be used to get the plant to follow a second order model. However, since only the output of the reference model is used in any way by the control algorithms in this system, model (3.2.2) could be regarded as the approximation of a second order system with an additional pole near $-\infty$ on the s plane.

Simulations with various conditions for input reference values and adjustment factors were conducted. As stated earlier, the results are plotted in Figures 3.1 to 3.4.
In these figures, the variables are defined as:

y_1......output of system controlled by algorithm A1 in its first order form,

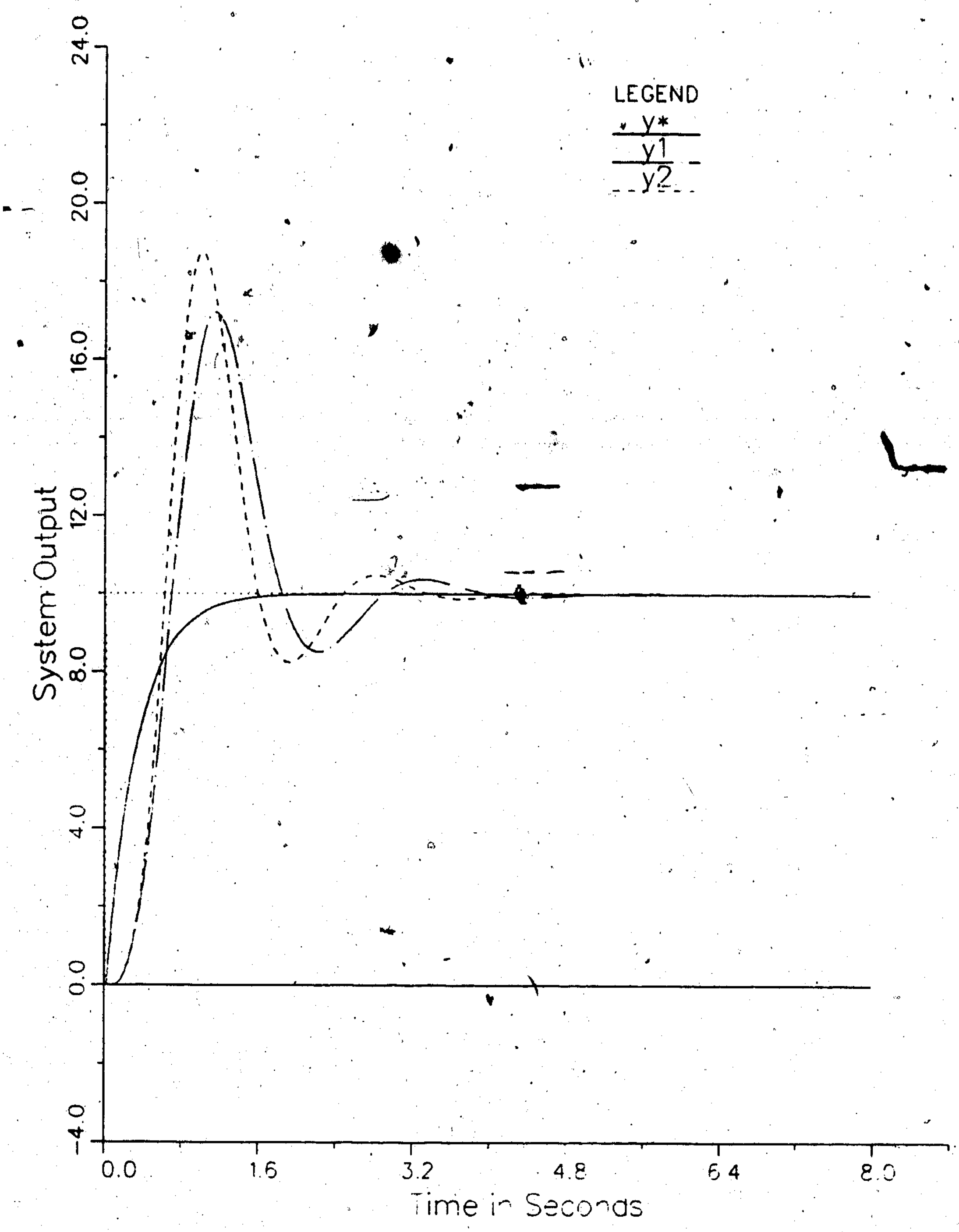


Fig. 3.1 Comparison of 1st and 2nd Order Algorithms, f=0.2, I.

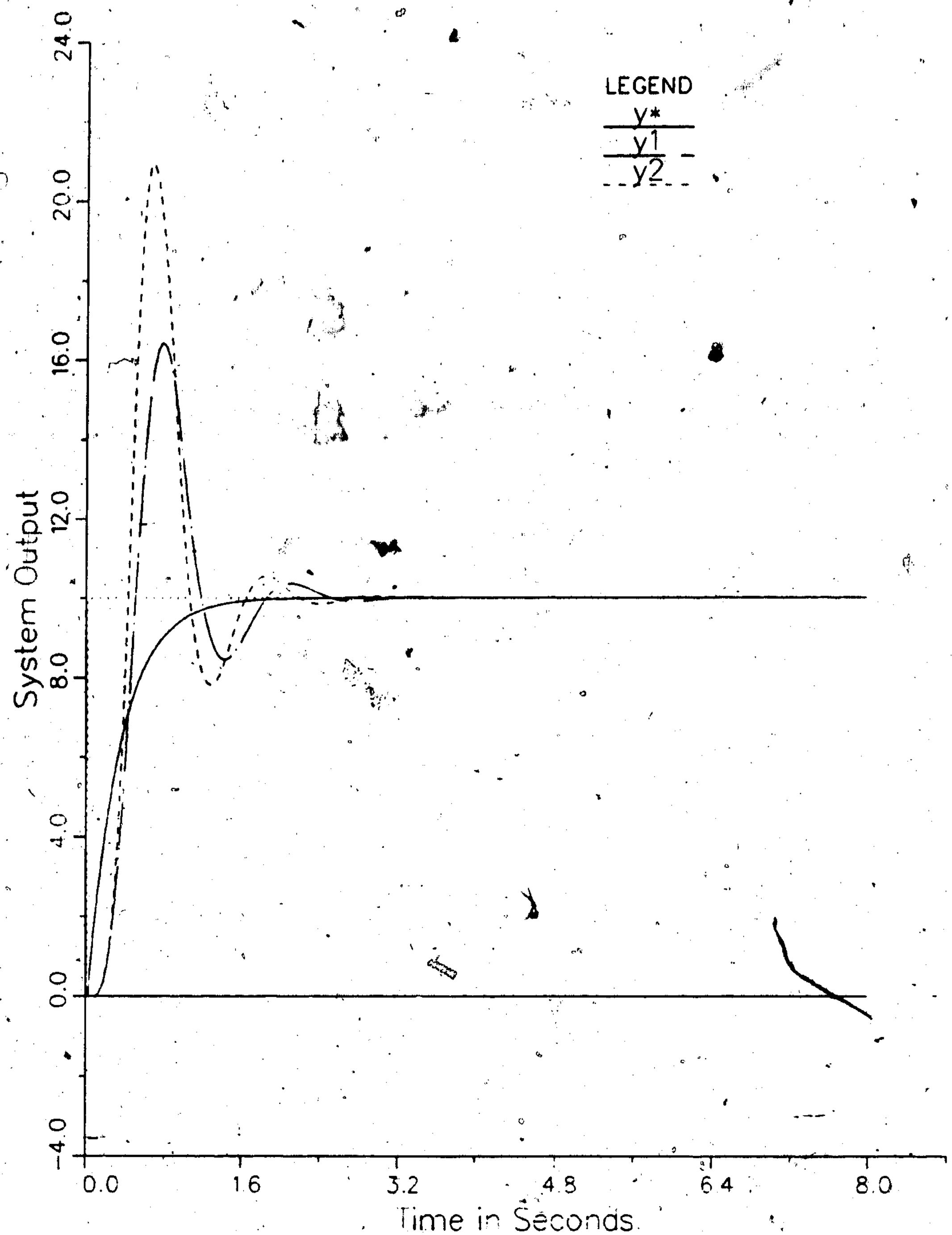


Fig. 3.2 Comparison of 1st and 2nd Order Algorithms, f=0.5, I.

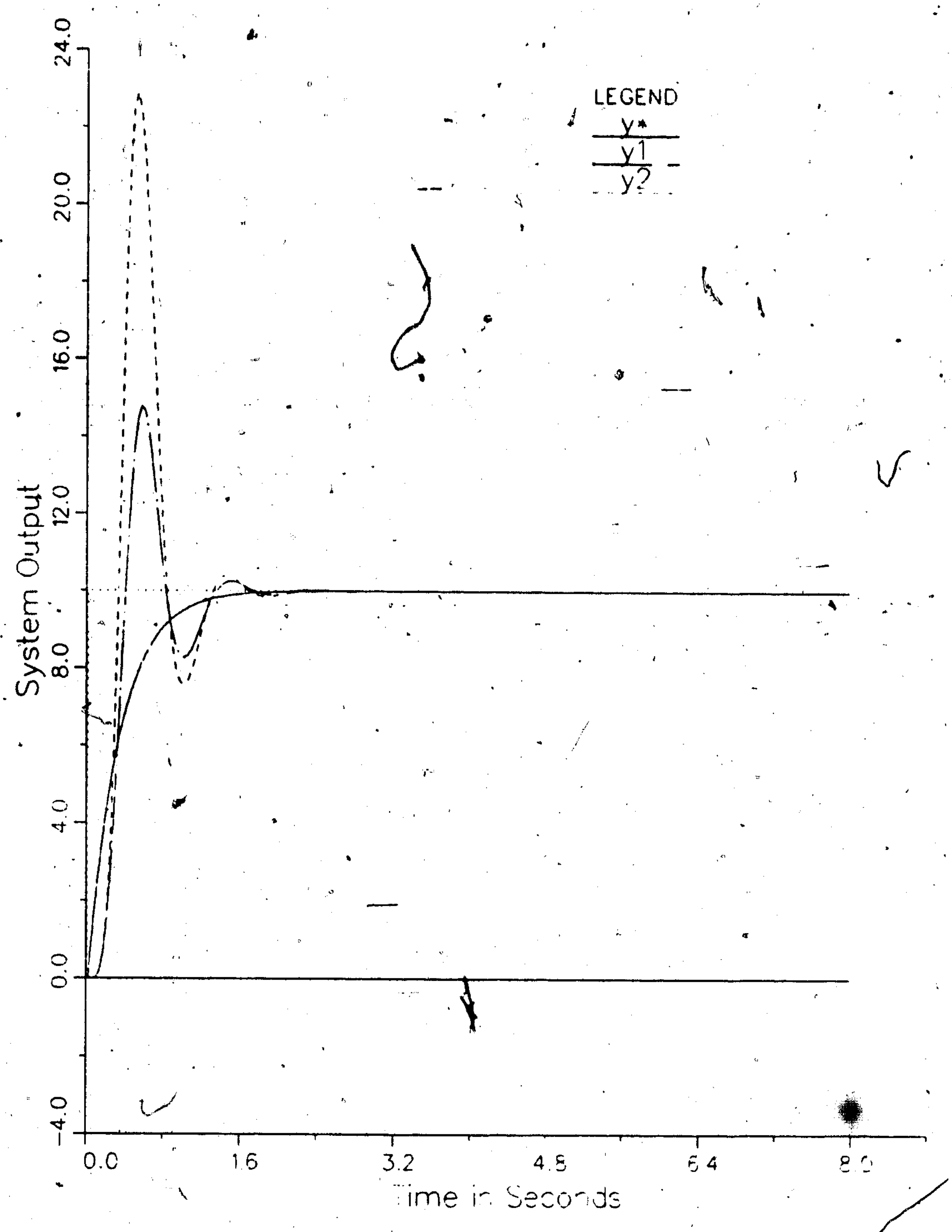


Fig. 3.3 Comparison of 1st and 2nd Order Algorithms, f=1.0, I.

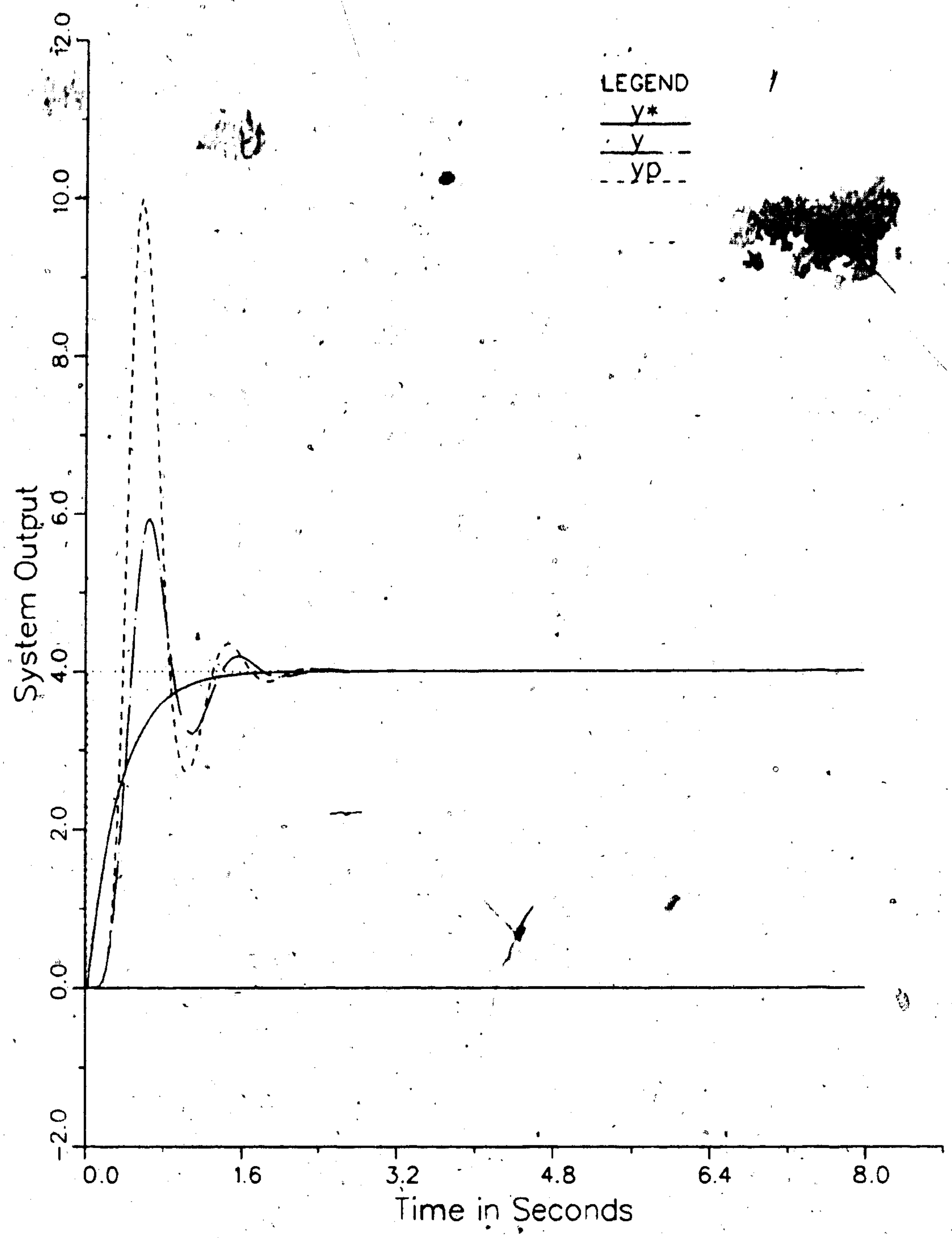


Fig. 3.4 Comparison of 1st and 2nd Order Algorithms, f=1.0, II.

y_2.......output of system controlled by algorithm A1 in its second order form,

y^*......reference model output, and

r.......reference input to the model.

In Figures 3.1, 3.2 and 3.3 the reference input is set at 10, and in Figure 3.4 the reference input is r=3. The adjustment factors for these simulations are f=0.2 in Figure 3.1, f=0.5 in Figure 3.2, and f=1.0 in Figures 3.3 and 3.4.

From these simulation results it should be noted that for a first order plant, the first order algorithm performs better than the second order one. In the end, both algorithms are stable but the second order algorithm has more trouble in its initial adjustment stage, resulting in greater overshoot. The additional effort (which could be measured by a quadratic cost function) is probably spent by the estimator part of the control algorithm before it "realizes" that the plant is actually of a lower order than expected. The higher order algorithm, however, does speed up the system response somewhat.

3.2.3 Simulations with Rectangular Wave Input Signals

To further illustrate the effect of overestimating the plant order on system performance, another form of input signals will be tried in this section. Traditionally, in simulations of adaptive control systems the rectangular wave input have been frequently used because of its unique characteristic of exciting all the frequencies uniformly. This characteristic normally enhances robustness of the system in the presence of disturbances. If the input signal does not offer continuous excitation of all the frequencies, sooner or later the estimator part of the algorithm will be working on the noise or disturbance signals alone, which could result in faulty estimations and adjustments. In the present discussion, no noise is included.

The rectangular waves used here have periods of 8 seconds, and 2.5 periods in the 20 seconds of simulation time. The simulation results are shown in Figures 3.5 to 3.8.

Again, the results show that the unnecessarily high order of the adaptive control algorithm A1 does not improve the system performance. Rather, the transient response of the system is somewhat degraded. The overshoots are higher. And furthermore, in the subsequent periods the second order algorithm has some erratic output harmonics while clearly the first order algorithm has already stablized. This shows that the second order algorithm, while not slower in its gain adjustments, is more sensitive to changes of the reference input.

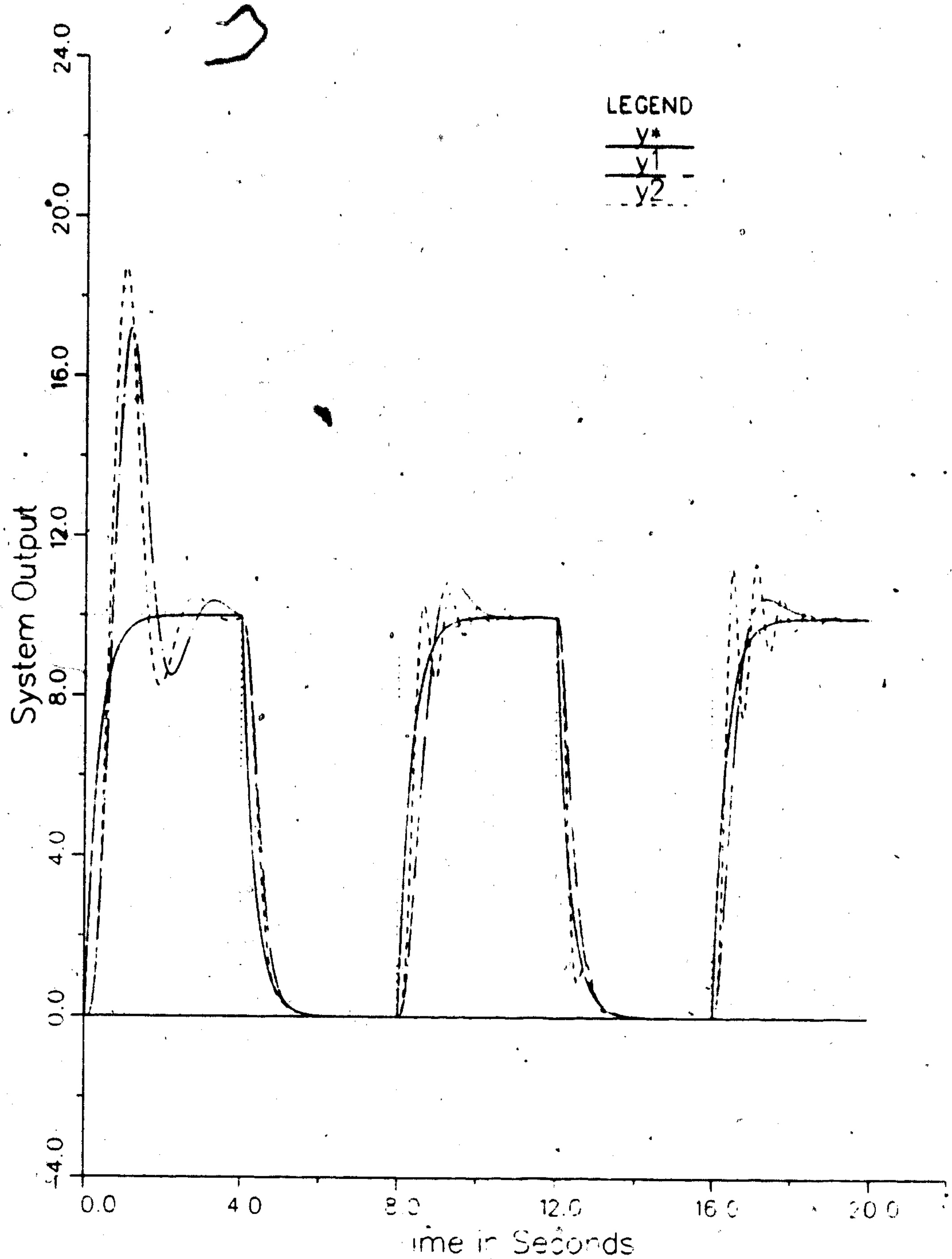


Fig. 3.5 Comparison of 1st and 2nd Order Algorithms, f=0.2, II.

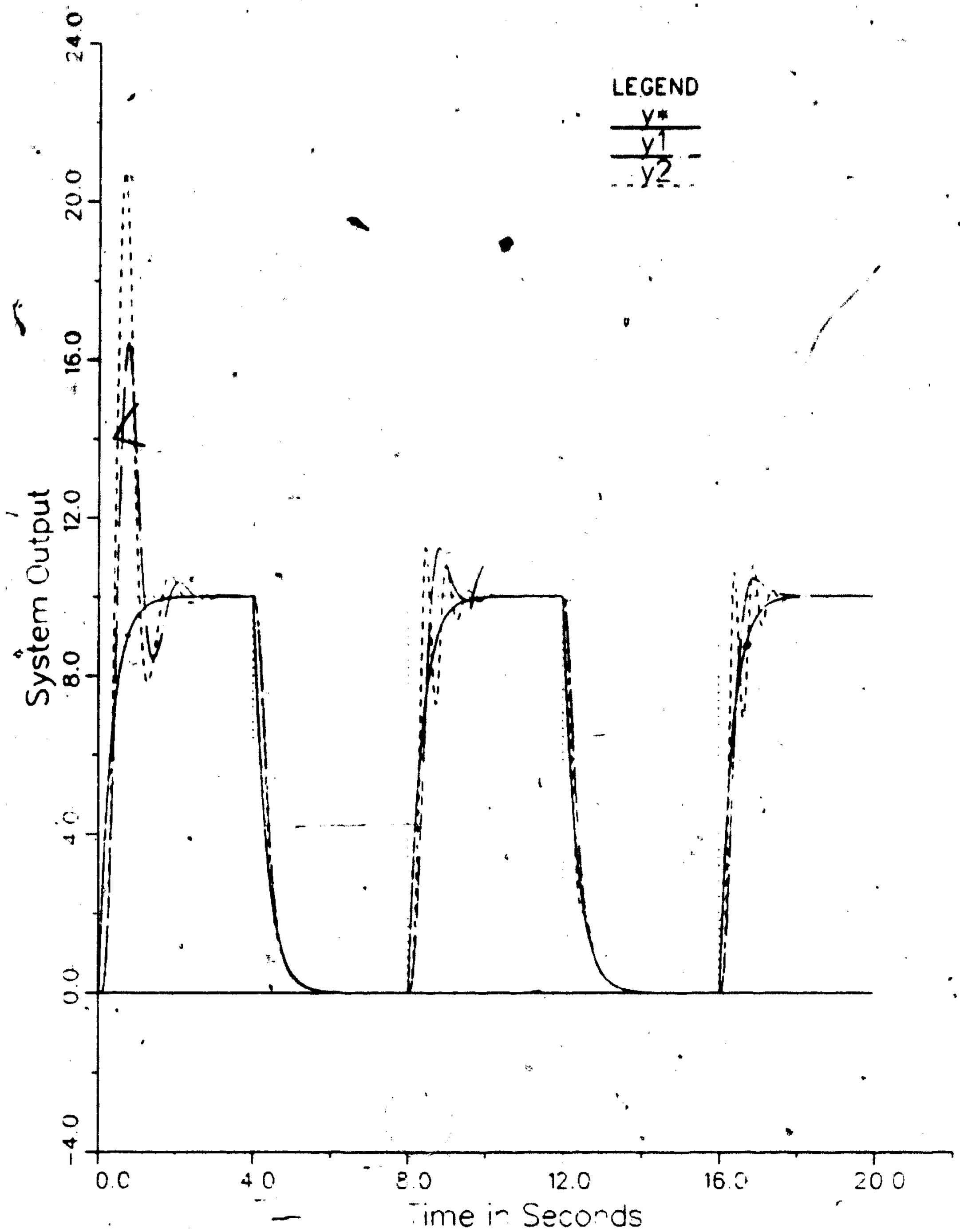


Fig. 3.6 Comparison of 1st and Second Order Algorithms, f=0.5, II.

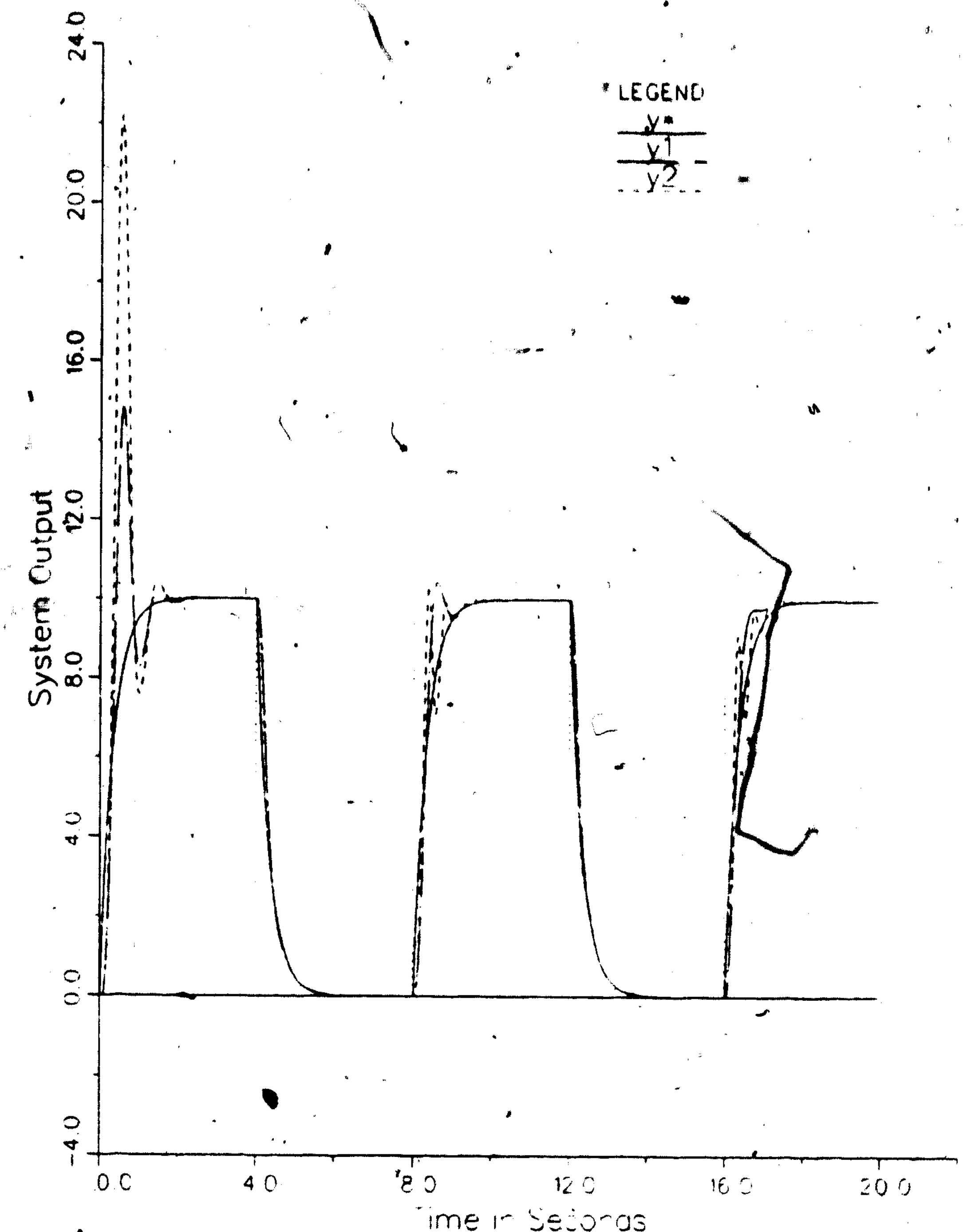


Fig. 3.7 Comparison of 1st and 2nd Order Algorithms, f=1.0, III.

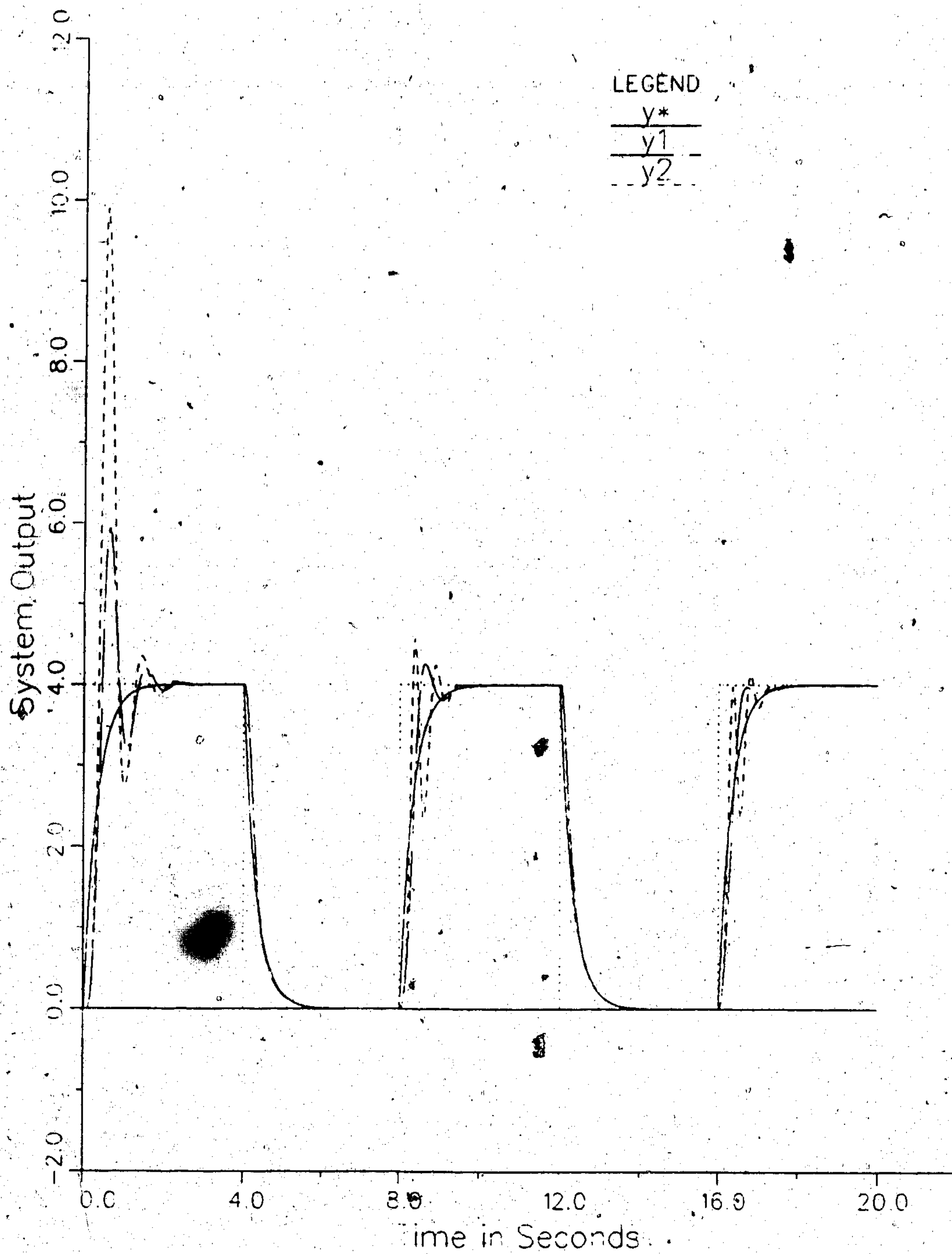


Fig. 3.8 Comparison of 1st and 2nd Order Algorithms, f=1.0, IV.

3.2.4 Simulations with Sinusoidal Input Signals

In this subsection the sinusoidal signal is used as the reference input signal. Since this kind of input signal does not contain all the frequencies, it will not excite all the modes in the system. Hence it is not very suitable for the adaptive control approach because the estimator part of the control algorithm would not be supplied with the complete information it needed to reconstruct plant parameters. However, if the system is globally stable, its stability will not be affected by the reference input being sinusoidal.

The sinusoidal signals employed here have the same period as the rectangular waves used in the previous section, ie. 8 seconds or 2.5 periods in the 20 second simulation duration. The resultant system responses are plotted in Figures 3.9 to 3.12.

From these results, it is again very clear that overestimating of the plant order results in degradation of system response. This time, the response of the second order control algorithm lags further behind that of the first order one, in addition to having greater overshoot.

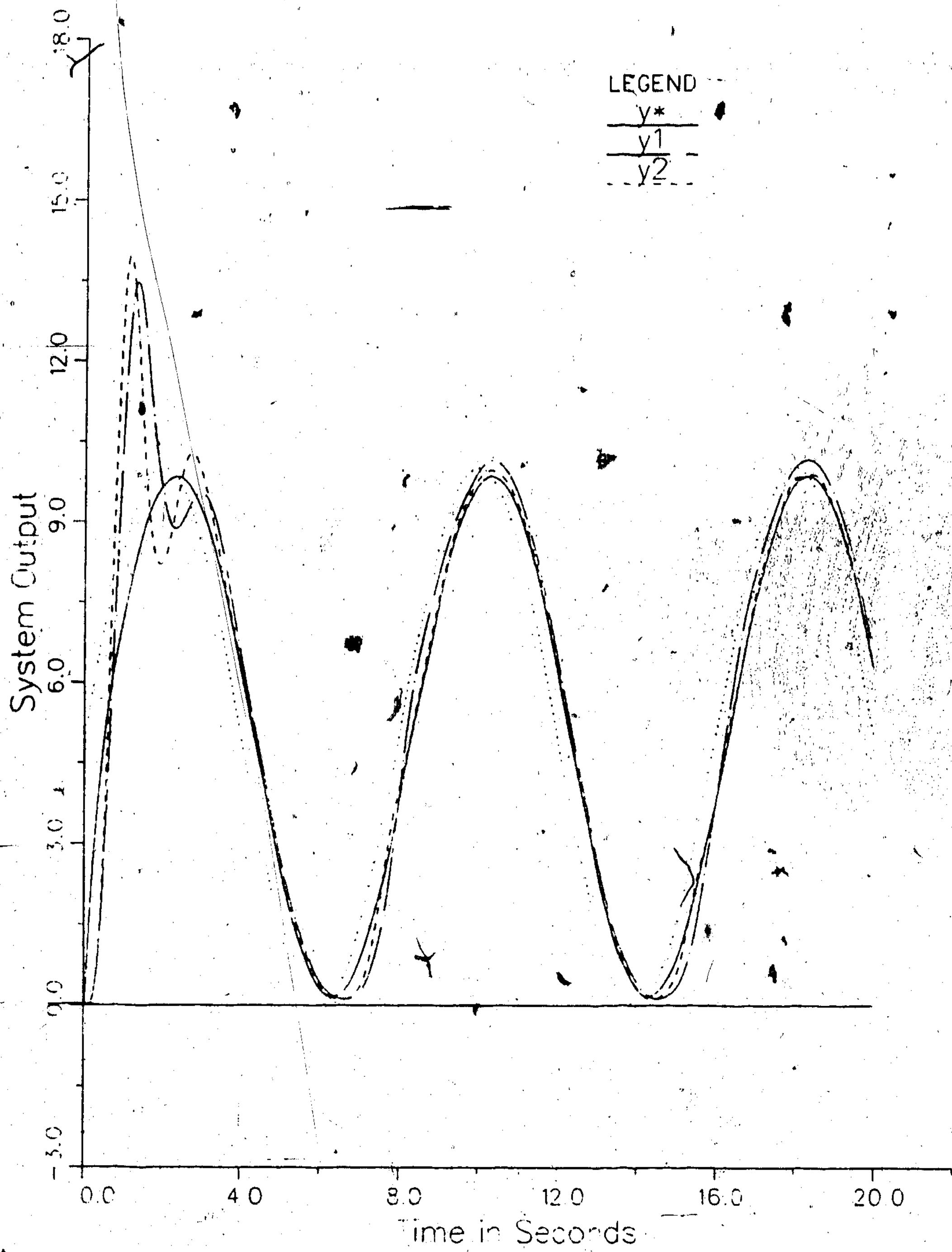


Fig. 3.9 Comparison of 1st and 2nd Order Algorithms, f=0.2, III.

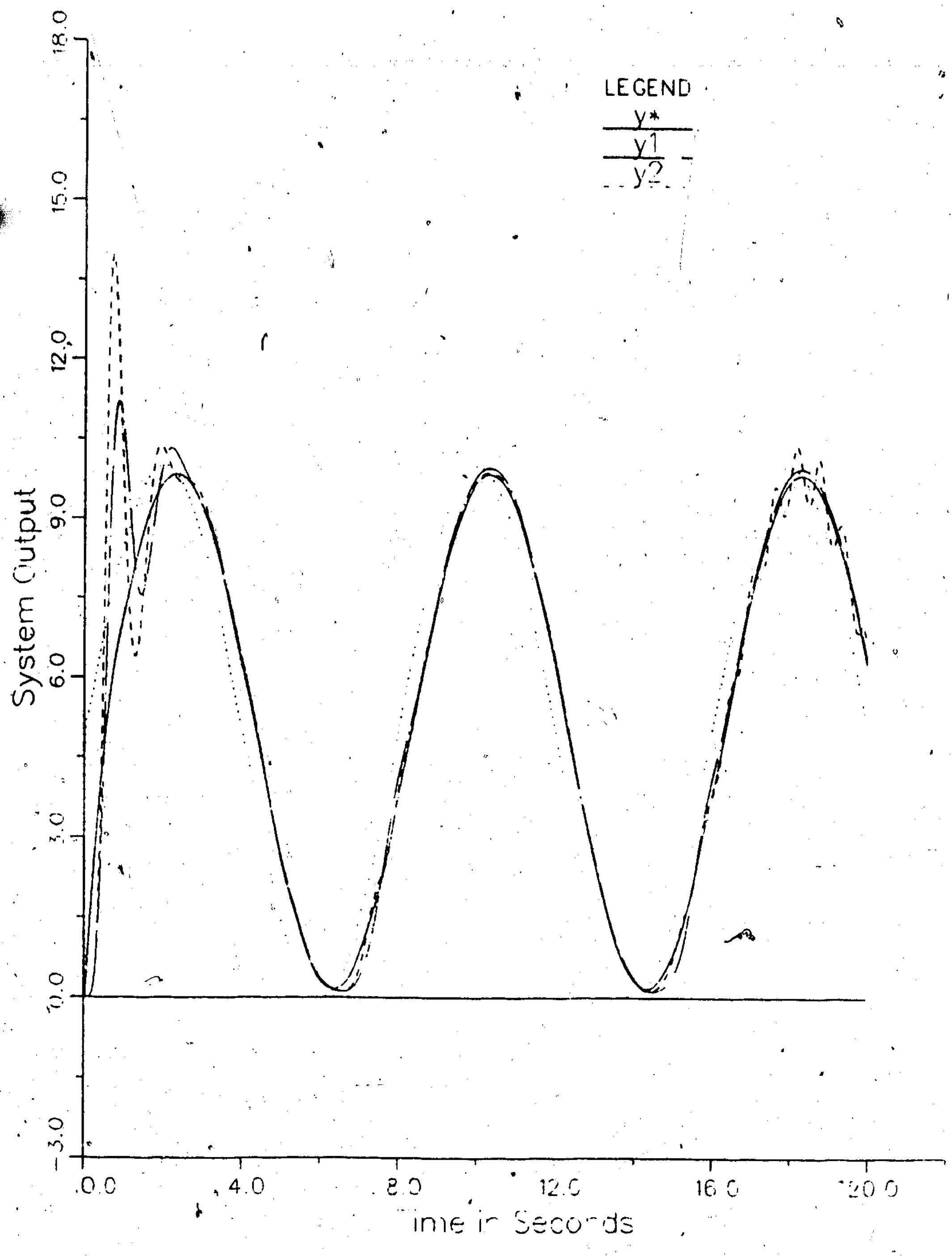


Fig. 3.10 Comparison of 1st and 2nd Order Algorithms, f=0.5, ...

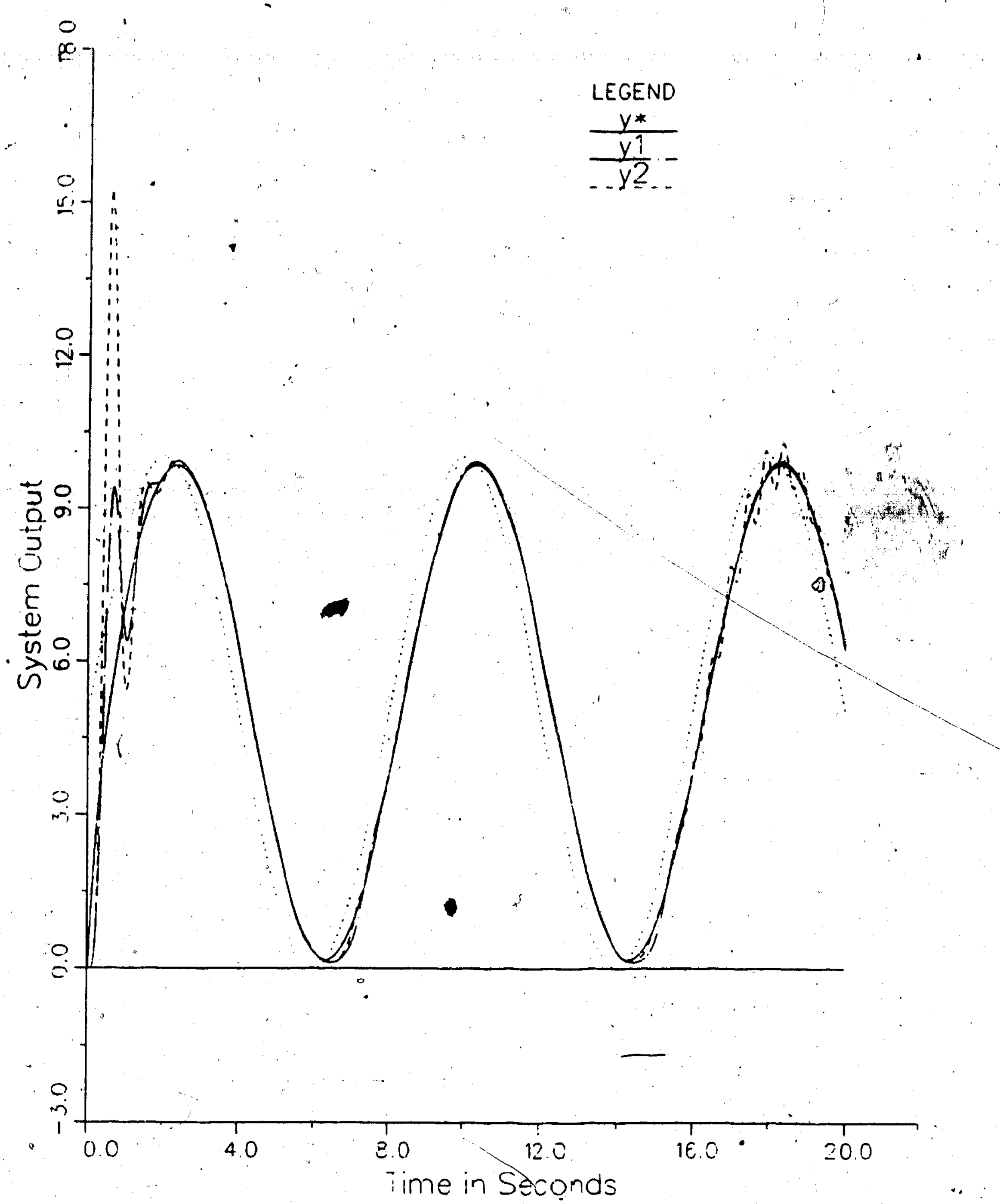


Fig. 3.11 Comparison of 1st and 2nd Order Algorithms, f=1.0, V.

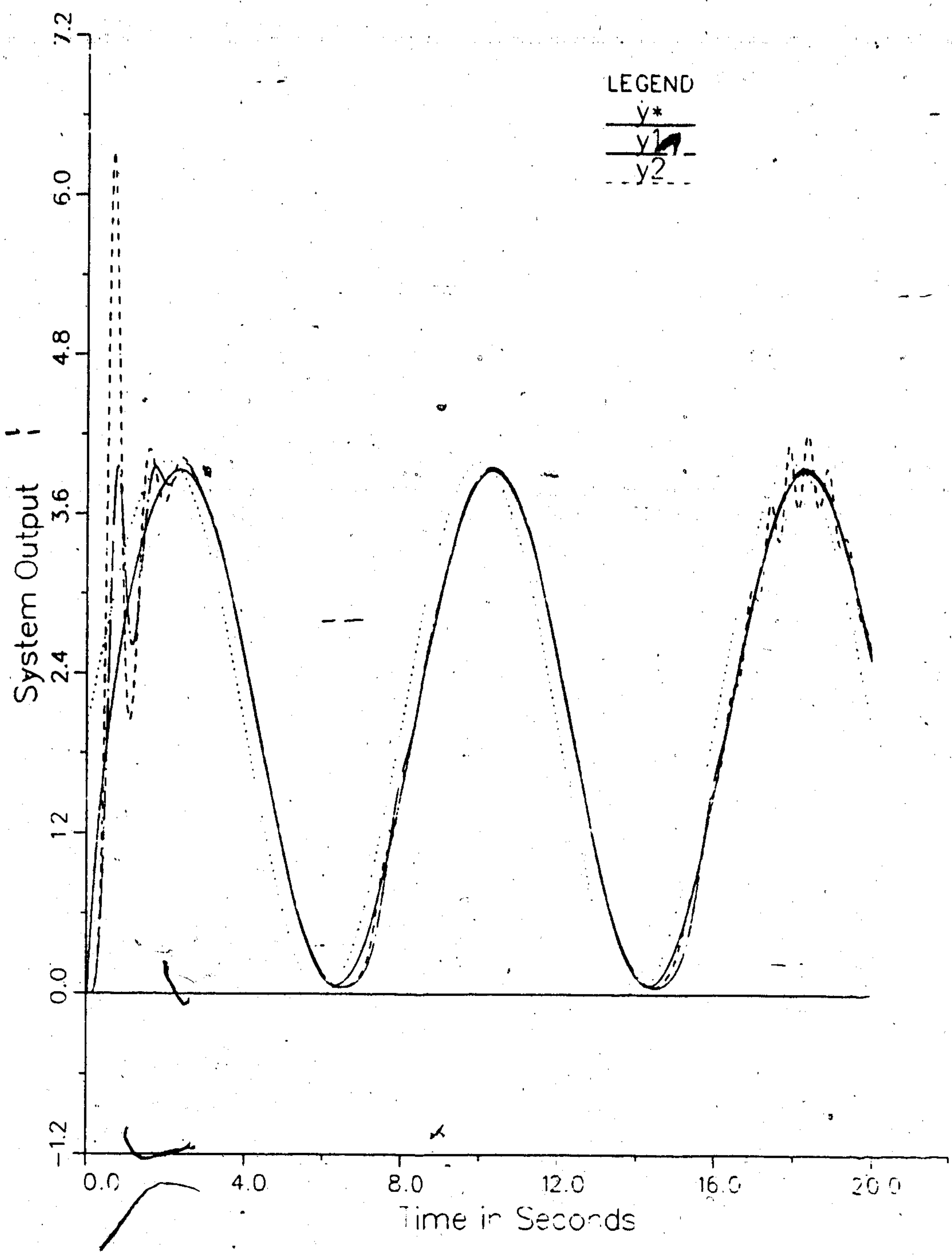


Fig. 3.12 Comparison of 1st and 2nd Order Algorithms, f=1.0, V!.

3.2.5 A Quantitative Comparison

It is useful to examine the quantitative aspect of the problem of overestimating the plant order. It was stated earlier that the difference of the system performance caused by using a higher order controller algorithm on a lower order plant could be measured by a quadratic cost function. Such a function is introduced here:

$$J_2=\int_0^\infty [y_2(t)-y^*(t)]^2 dt \qquad (3.2.3)$$

$$J_1=\int_0^\infty [y_1(t)-y^*(t)]^2 dt \qquad (3.2.4)$$

and

$$J_r=J_2-J_1 \qquad (3.2.5)$$

where y_1 and y_2 are defined in Subsection 3.2.3. above.

It will be interesting to examine the effect of changing of operating conditions such as reference input level r and adjustment factor f on the value of J. Figures 3.13 and 3.14 show the pattern of these changes. In Figure 3.13, f is fixed at 0.6, while r changes from 2 to 20. In Figure 3.14, r is fixed at 10, while f changes from 0.1 to 1. All the results are obtained with step input signals.

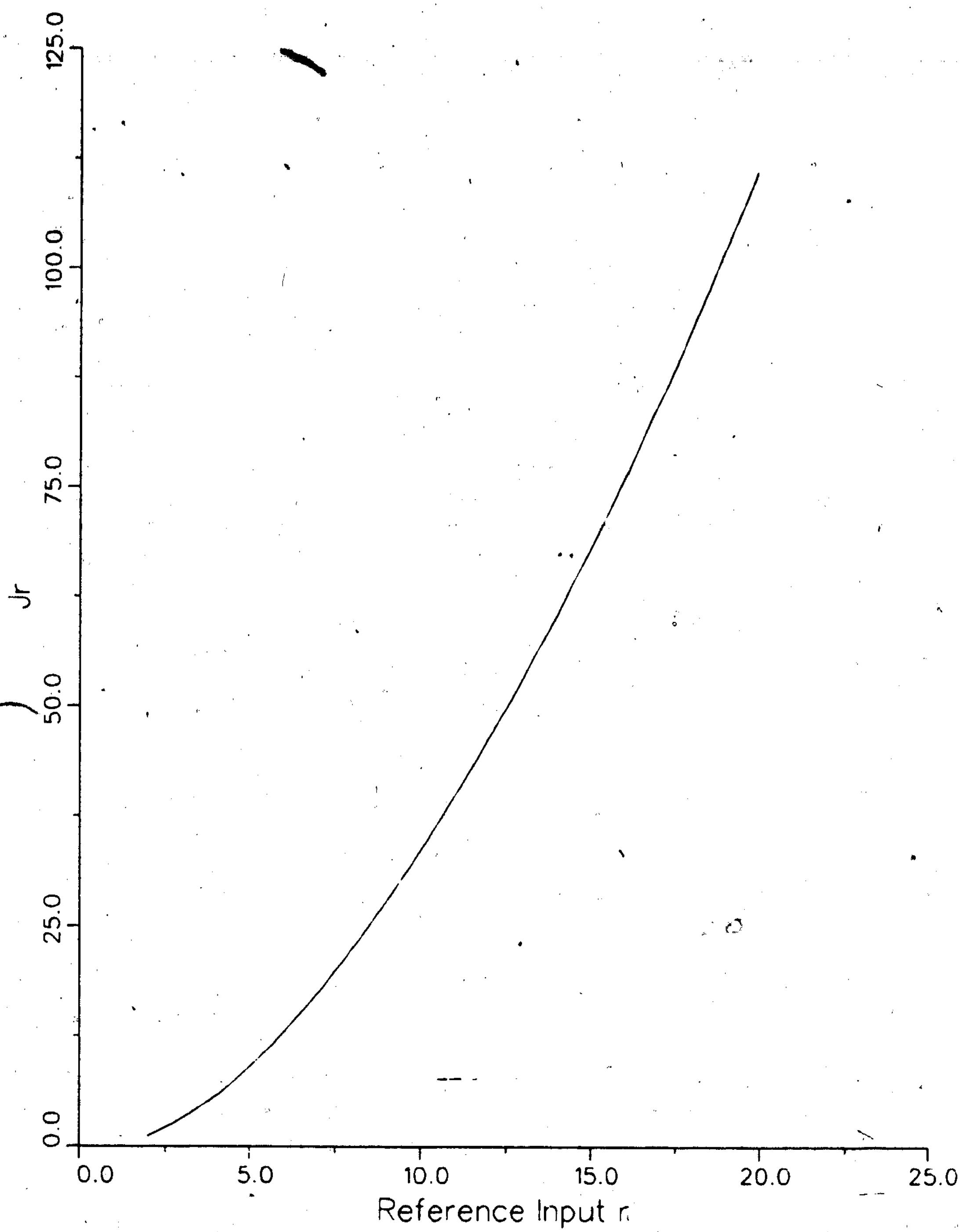


Fig. 3.13 Performance of A1 in 1st and 2nd Order Forms, f=0.6

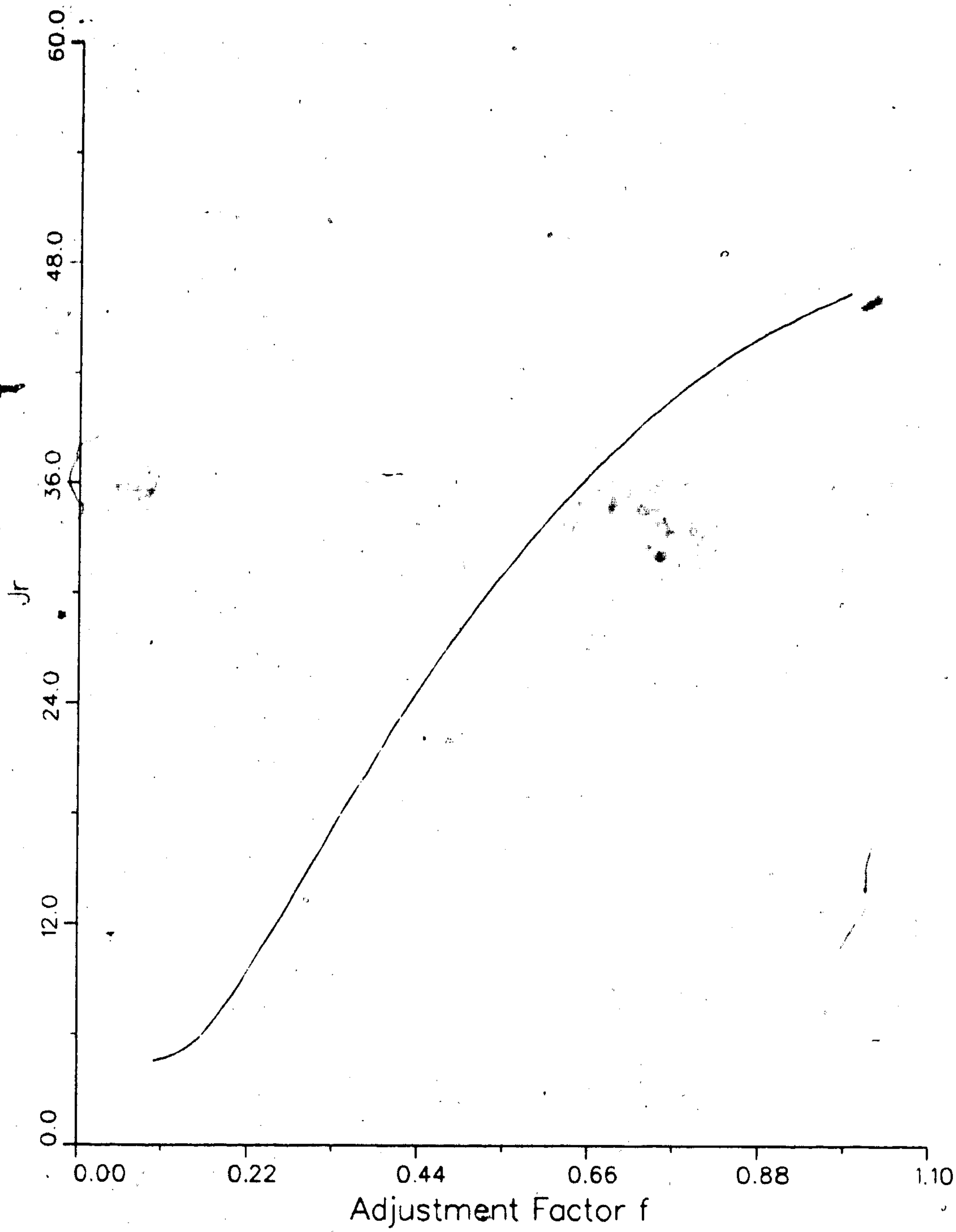


Fig. 3.14 Performance of A1 in 1st and 2nd Order Forms, r=10.0

From these results, it is clear that J_r is always of positive value, which means the algorithm A1 performs better in its first order form than in its second order form on first order plants. The J_r values increase with [illegible] because system output is increased. Its increase with f is rather interesting because it shows that the greater the adjustment rate, the greater is the difference between the performance of the two algorithms.

The simulation results of this section all point out to the same conclusion: overestimating the plant order result in poorer performance of the complete system. This characteristic of algorithm A1 is rather important because while designing a adaptive control system, some people tend to set the controller order as high as possible. While this will not adversely affect the stability of the system, it might degrade its performance if the order of the control algorithm is chosen to be higher than that of the plant. This degradation is, however, small compared to what will happen if the plant order is underestimated.

3.3 Effects of Unmodeled Dynamics on Algorithm A1 in Its Second Order Form

3.3.1 General Comments

In this section, simulations similar to those done in Sections 2.3. and 2.4. will be conducted, with the algorithm A1 used in its second order form, while the plant follows the model (1.3.11). This type of work is done to reinforce the conclusions reached in Section 2.4. with new data from a higher ordered plant and control algorithm.

By changing the position of the complex pole pair of (1.3.11) and observing the resulting change in system performance under algorithm A1 in its second order form, a broader picture of how different unmodeled dynamics affect system performance including stability could be obtained. It is possible to determine whether the results from Section 2.3. are restricted to first order systems. For this purpose, the same measure used earlier to compare effects on system behavior of individual unmodeled pole positions is employed again, namely,

$$J=\int_0^\infty [y(t)-y_p(t)]^2 dt \qquad (3.3.1)$$

The simulations will be conducted with the complex pole pair again serving as unmodeled dynamics. And their positions will be moved around. The other measure of system performance used in Chapter 2, ie., the gain deviation D will not be used since with a different number of gains it is difficult to compare the results numerically.

The difference between these simulations and those of Section 2.3. is related to the area of movement of the unmodeled dynamics poles. These simulations require the system to be in a stable situation throughout for the quantity J to be finite and meaningful. The second order system in this section turns out to be less stable, so that the area of unmodeled pole positions is smaller here. The area in the s plane is shown in Figure 3.15.

In order determine whther the results obtained through the simulation studies in Section 2.4. are restricted to the first order algorithm situation, similar studies will be conducted in this section. The difference will be in the order of the plant and the algorithm, and the position of the unmodeled poles. The effect of changes in the control and adjustment period T on system performance will be examined in similar fashion.

3.3.2 Simulation Results with J vs. Separation Ratio

Preliminary tests show that the pole range in Figure 3.15 is appropriate (the system remains stable under our simulation conditions when the unmodeled dynamics poles are in this range) for the simulations in this section. The unmodeled pole pair will be moved around within this range for the simulations that follow.

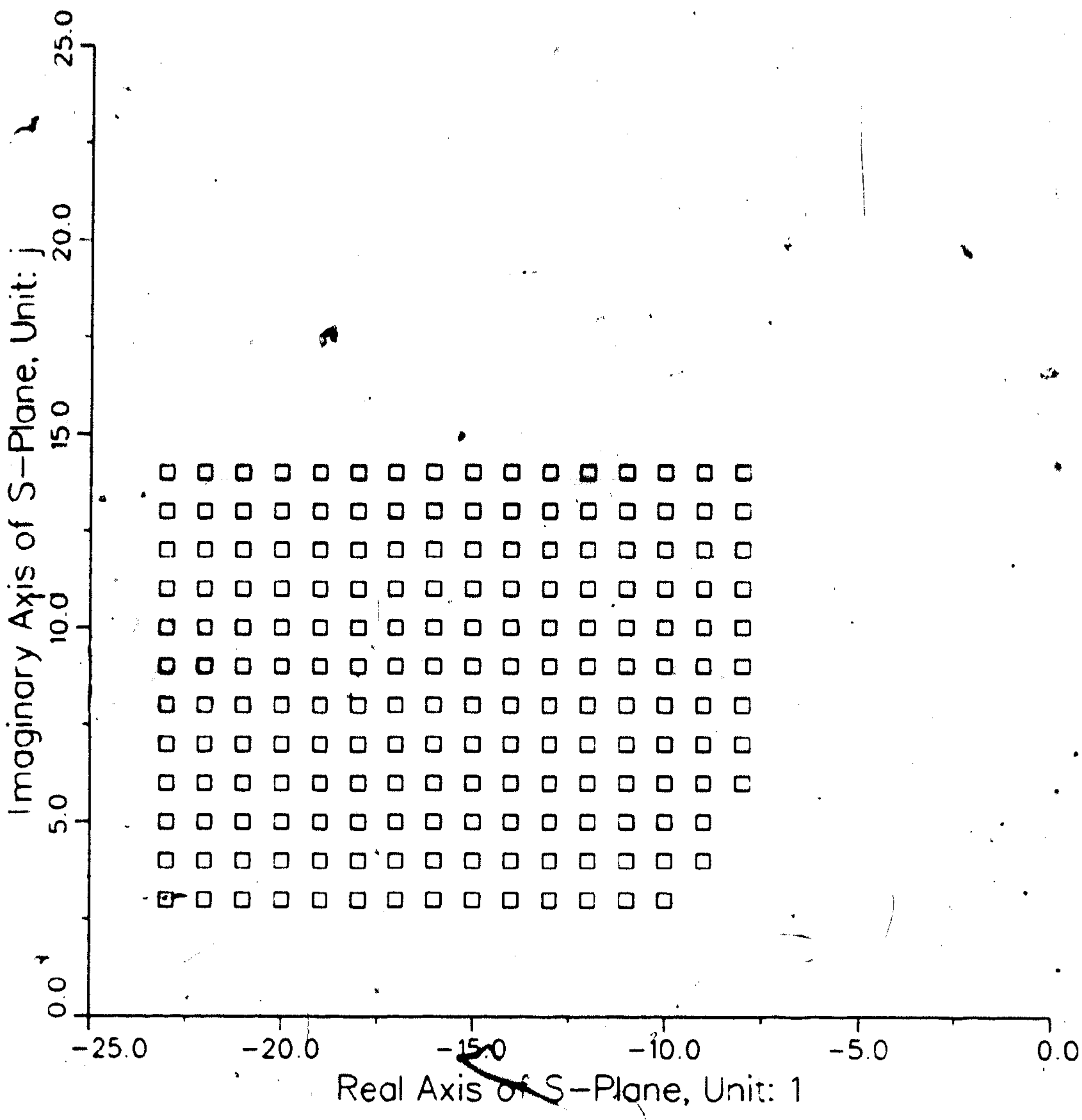


Fig. 3.15 The Distribution Range of U.D. Poles, II.

In these simulations, system parameters that are not changed follow the same pattern used in Section 2.3. except for the initial values of the control gains. Following is a list of these parameters which will remain unchanged in this section:

sampling period for adjustment and control T=0.04(sec),
reference input r=10.0;
adjustment factor f=0.2;
plant dominant poles at $\alpha_1=1$ and $\alpha_2=1.5$, and
all gains have the initial value of zero.

The separation ratio SR used here is defined as

$$SR=\sqrt{\alpha^2+\beta^2}/\alpha_2 \tag{3.3.4}$$

The reference model in the following simulations is:

$$(s+3)(s+10)Y(s)=60U(s) \tag{3.3.5}$$

Figure 3.16 shows the results of changes in J due to changes in the unmodeled pole positions. From these results, it is clear that generally, the greater the separation ratio, the less is effect the unmodeled pole pair has on system performance. However, the plotted results again cover an area, not forming a curve, meaning there is no one to one relationship between the SR value and J. Recalling the result in Figure 3.5, the similarity is obvious.

3.3.3 Variations in J due to Plant Deviations

This subsection covers the same ground as 2.3.2.. The open-loop response of the plant model (1.3.11) approximates that of

$$(s+\alpha_1)(s+\alpha_2)Y(s)=2\alpha_1\alpha_2U(s) \qquad (3.3.5)$$

if the real poles are dominant.

If the unmodeled poles change position, the (1.3.11) open-loop response to a step input signal will also change. Again define the open-loop deviation as

$$J_o=\int_0^\infty[y_o(t)-y_{op}(t)]^2dt \qquad (3.3.6)$$

where $y_o(t)$ is the open-loop response of (1.3.11) to a step input and $y_{op}'(t)$ is the same response of (3.3.5). This is similar to the definition in (2.2.6).

In the following simulations, the values of J are compared with those of J_o, using the latter as a substitute for the separation ratio as a measure of the deviation of plant model caused by the presence of unmodeled dynamics.

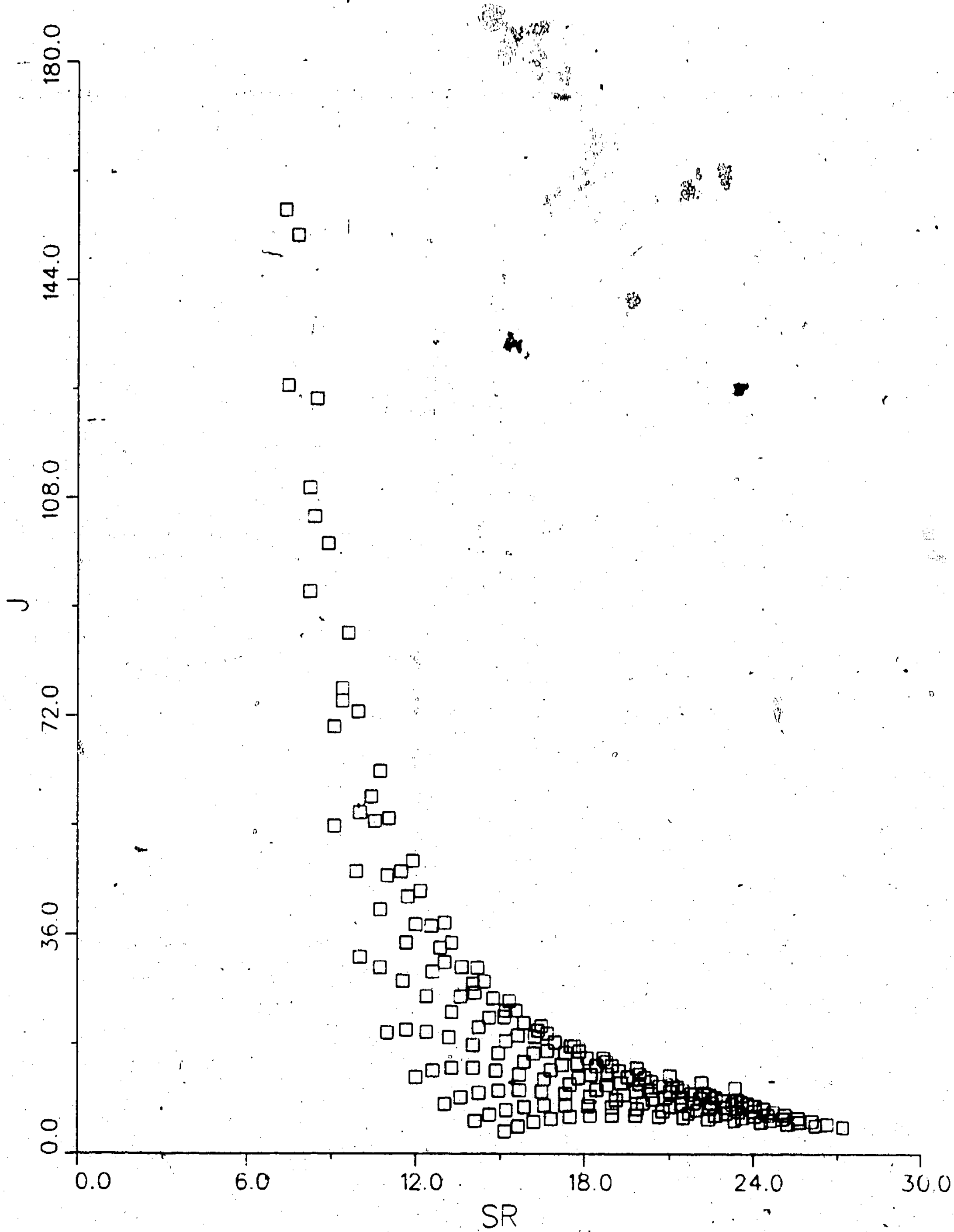


Fig. 3.16 A1, Different U.D., 4th Order Plant, I.

The results of the simulations are shown in Figure 3.17. It is clear that the system performance, measured by J, deteriorated with increase of J_o. The simulation results of Figures 2.10 and 3.17 are remarkably similar.

3.3.4 The Effect of Changing T on System Performance

The simulation work in this subsection is similar to that in Section 2.4.. The difference again is in the system order. Here a fourth order plant is controlled by an algorithm of second order and a pair of complex poles act as unmodeled dynamics. The purpose of this section is to see whether the results and conclusions reached in Section 2.4. are isolated cases or not. The design guideline proposed by Rohrs in [4b] concerning the control and adjustment rates is further tested.

In the following simulations, plants of the (1.3.11) type with different unmodeled pole positions are subjected to control algorithm A1 in its second order form with gradually changing T, and the resultant performance value of J is observed.

In these simulations, the adjustment and control period T is changed from 0.04 to 0.4 in small increments (0.01), while the resultant performance value J is observed. The

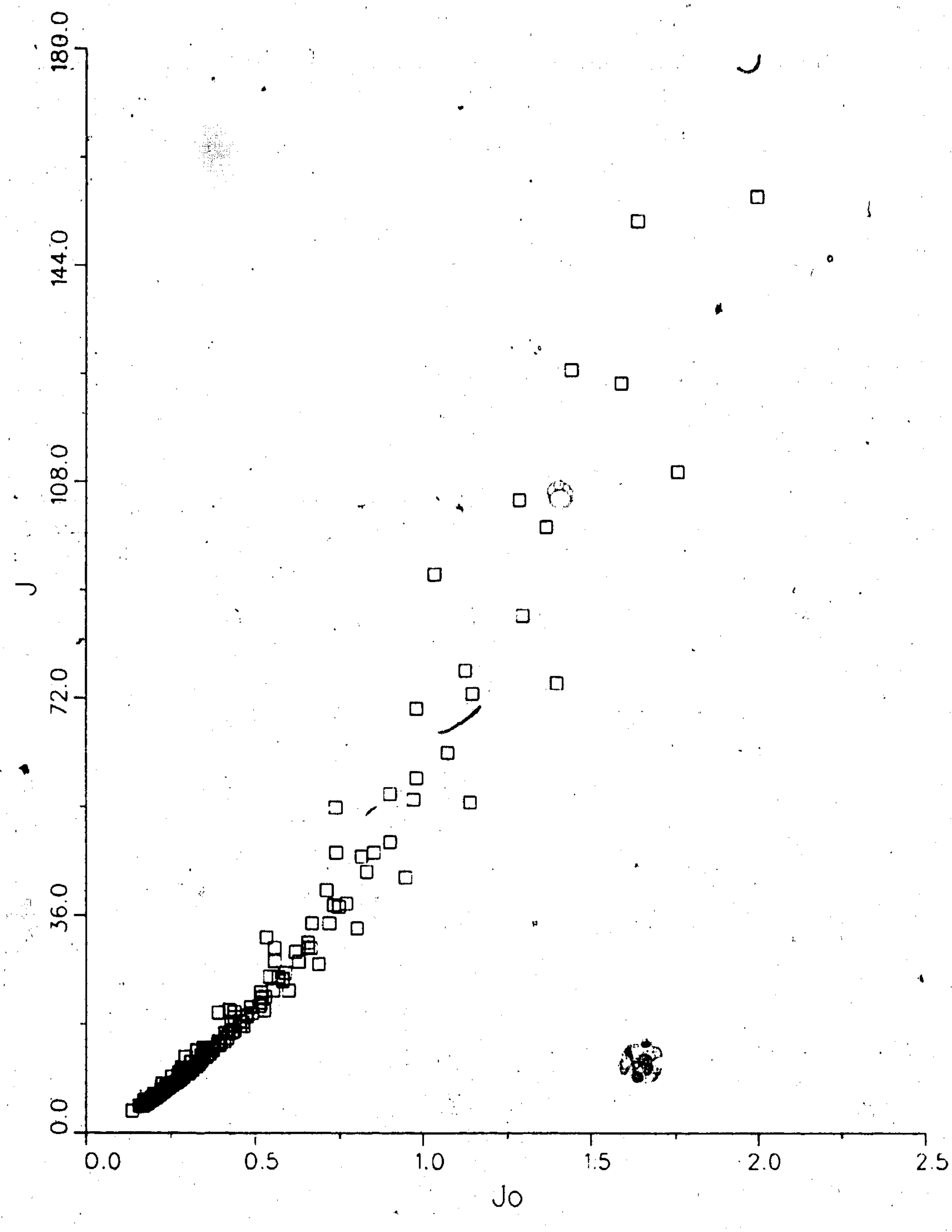


Fig. 3.11 A1, Different U.D., 4th Order Plant, II.

results of these simulations are made relevent by the small sampling period (T_p=0.01) used in the plant model discretization. The model of the plant and the accuracy of its representation of the actual plant do not change when T is changed. The plant unmodeled dynamics are also changed to observe whether the method of using as large a T as possible has the same effectiveness for different unmodeled pole positions.

First of all, three different unmodeled pole positions are chosen for this simulation: -9±j8, -14±j6 and -15±j3, because at T=0.04, they all result in similar values of J around 30.0. The simulation results are shown in Figure 3.18 (c. Figure 2.13).

Comparing with Figure 2.13, it is clear that the patterns are very similar. From these results, we could say that indeed, the systems respond similarly to changes in T.

Again the J values fall sharply with initial increase of T. This shows that the effectiveness of the method of seeking stability by increasing T is not uniform in a higher order situation either. There seems again to be a 'saturation point' where the increase of T does not improve system performance anymore.

Figure 3.19 shows another group of pole positions, that of -9±j7, -13±j8 and -16±j2, as their effects on system

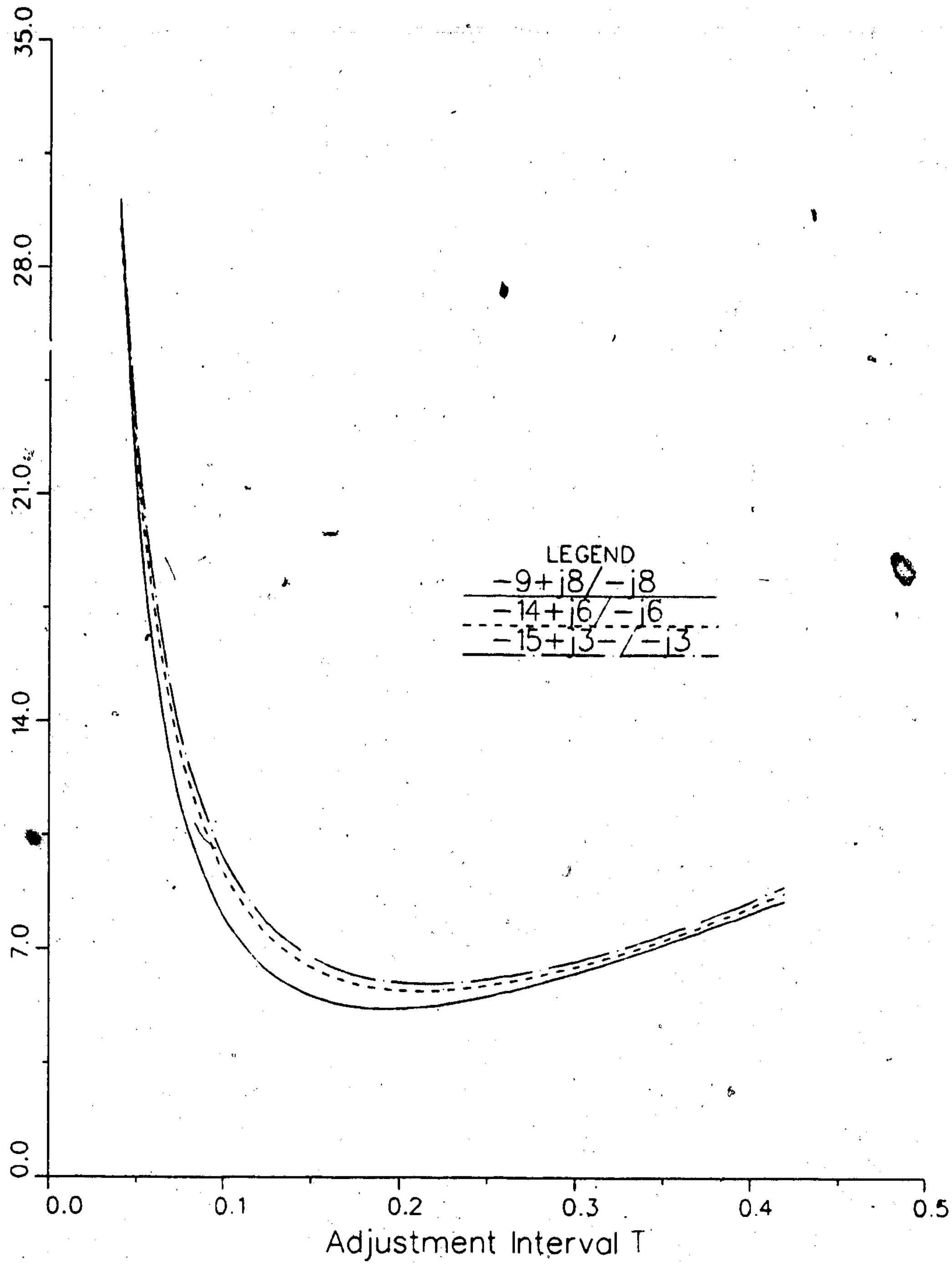


Fig. 3.18 System Performance vs. T, 4th Order Plant, I.

performance change with changes of T. The similarity of these results eith those shown in Figure 2.15 is clear.

3.3.5 The Square Root Values of J

As noted in the previous chapter and confirmed above when the adjustment and control interval T is increased, the system performance improves markedly for a while. After a certain point, however, this trend stops and further increase of T do not give further improvement of system performance anymore.

Is this because of the nature of the performance measures that are chosen? Sure, they are quadratic quantities. This could have been the reason for the sharp fall of J with the increase of T.

In Figures 3.20 and 3.21 the square root of J is plotted versus the control period T. Figure 3.20 represent the same simulations as Figure 3.18, and Figure 3.21 is from the same simulations as Figure 3.19.

The results of these simulations further extended those from the previous chapter and give full support to the conclusions reached in Section 2.4.. It is very clear that the sharp fall of performance with the increase of T and the 'saturation point' characteristics are as marked in these

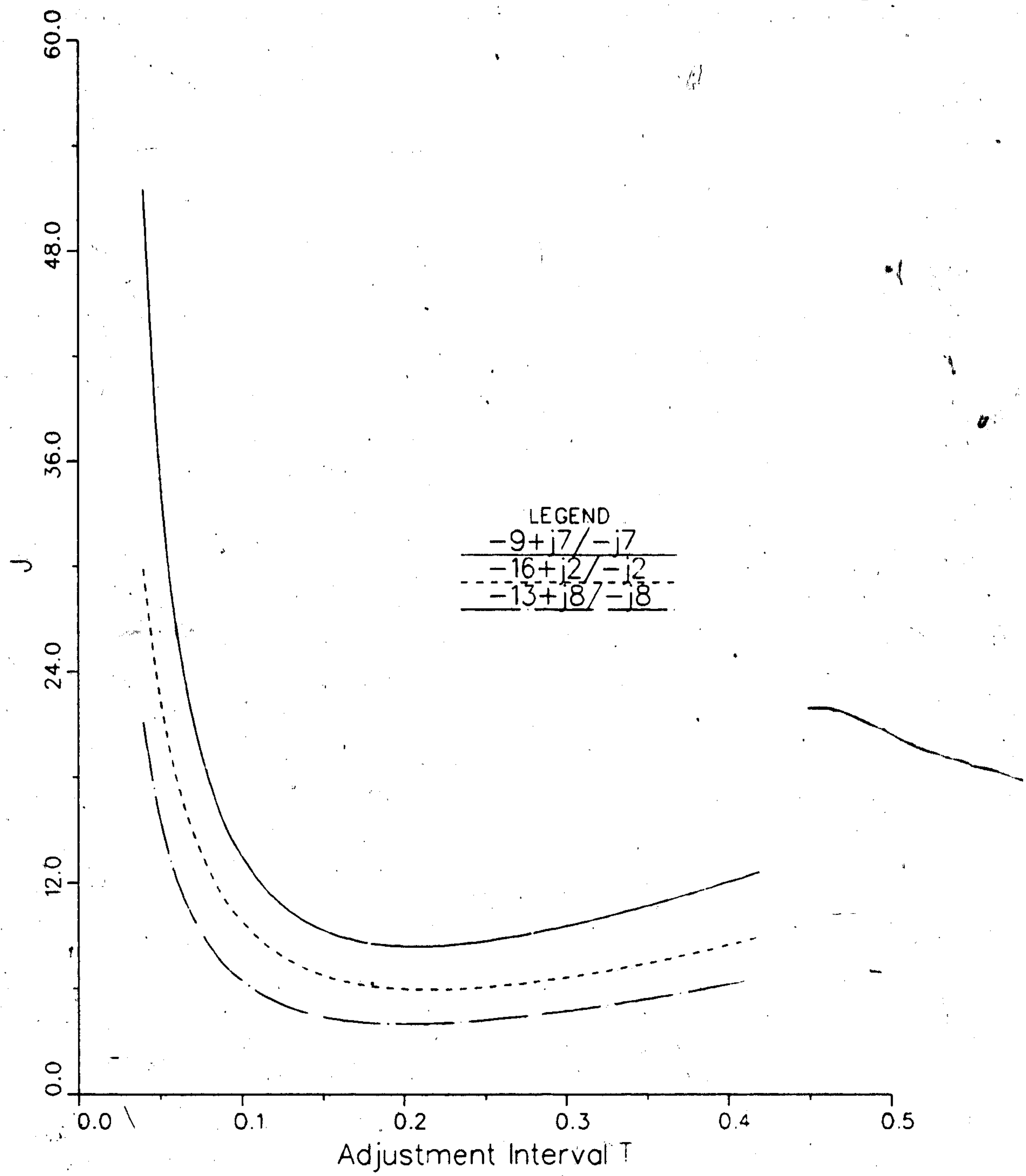


Fig. 3.19 System Performance vs. T, 4th Order Plant, II.

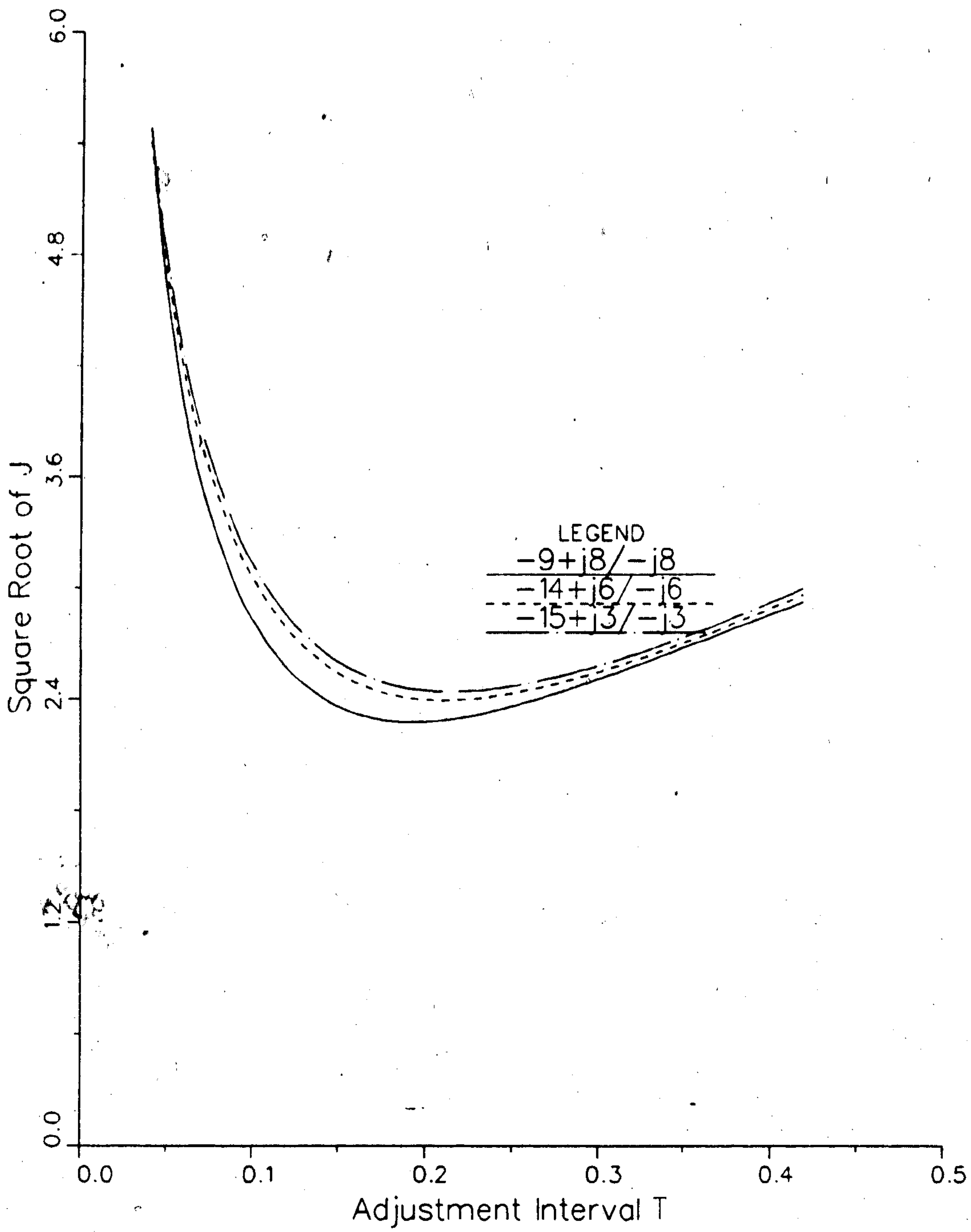


Fig. 3.20 System Performance vs. T, 4th Order Plant, III.

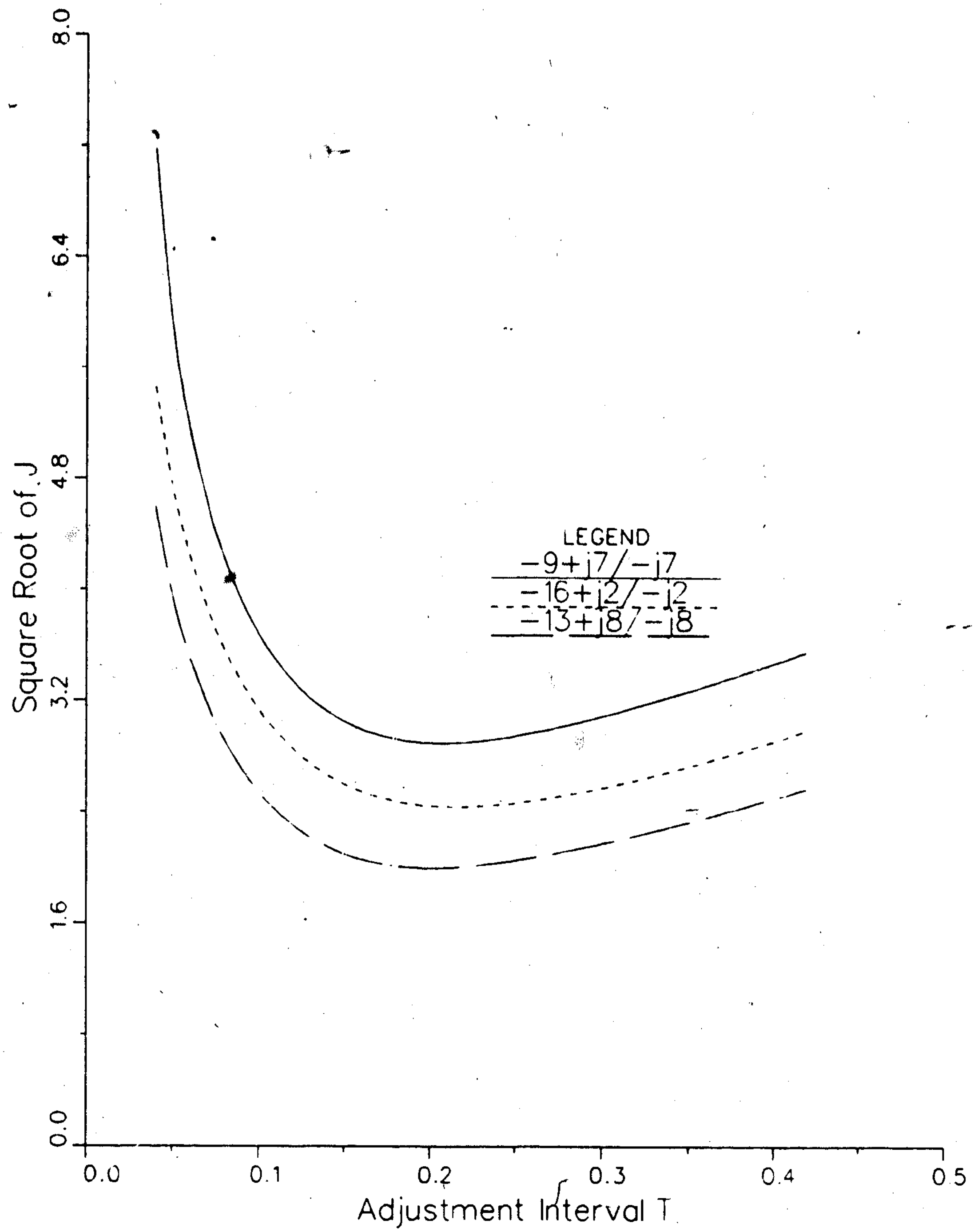


Fig 3.21 System Performance vs. T, 4th Order Plant, IV.

plots as in the earlier ones. Hence it is safe to say that the conclusions concerning the effect of T reached in the previous chapter and confirmed in this section are valid even if $\sqrt{J}$ is used as the performance index.

3.3.6 Discusion

From the results obtained in this section, all the conclusions reached in Sections 2.3. and 2.4.. are supported. These include the changing pattern of the performance index J with the change of unmodeled pole positions, and with the changes in the adjustment and control interval T.

The patterns of the changes of J as shown in the simulation results here are very similar to those in Chapter 2.. The difference is that generally, the higher order systems of this chapter are less stable compared with the lower order systems of the previous chapter. In this chapter, two poles act as the "modeled" dynamics instead of the one pole in Chapter 2.. The effect of the unmodeled poles on the plant open-loop responses are less evident here, while their effect on the closed-loop system is more marked. This means that with the same number of high frequency poles as unmodeled dynamics, a higher order adaptive control system performs better than a lower order one.

3.4 Effect of Changing the Controller Order

In section 3.1., the simulation results showed that using a higher order form of the control algorithm A1 on a lower order plant degrades the system performance somewhat, although the system stability is not adversely affected. Some people tend to use higher order control laws when the actual plant order is not known. When there are unmodeled dynamics in the system, would a higher order control law perform better than a lower order one? Combining the theoretical conclusions from [19, 20] with the simulation results reached so far in this thesis, it is clear that if the increase of the order of the control law eliminates the unmodeled dynamics, the system performance will be improved. What if such an increase of the controller order does not eliminate unmodeled dynamics?

In this section, a series of simulations will be carried out to answer this question. Here the algorithm involved is A1 in its second order form but the plant is the same to what used in Chapter 2., ie., that described by the model (1.3.10). In this case, The aim is to see is whether the increase of the algorithm order necessarily enhance system stability if this increase does not make the order of the control law as high as that of the plant.

This is a situation where there is no clear cut way of defining particular plant modes as the unmodeled dynamics. Since the algorithm is of second order, the plant is presumed to be of the form:

$$(s+a)(s+b)Y(s)=g\times a\times b\times U(s) \tag{3.3.6}$$

or

$$(s^2+2\alpha s+\alpha^2+\beta^2)Y(s)=g(\alpha^2+\beta^2)U(s) \tag{3.3.7}$$

by the designer of the system. The real plant model, however, is (1.3.10), which has one real pole and a complex pole pair. For comparisons with results obtained in 2.2, the complex poles will be moved in the same fashion and the real pole will remain dominant. While it is not possible to classify one of the complex poles as the unmodeled dynamics (certainly not the real pole either), there is no doubt that unmodeled dynamics are present in the system. The area of locations for the complex pole pair will be the same as in Chapter 2. (Figure 2.7). The results of the simulations are shown in Figure 3.22.

Comparing these results with those shown in Figure 2.10, it is clear that the second order algorithm, while not

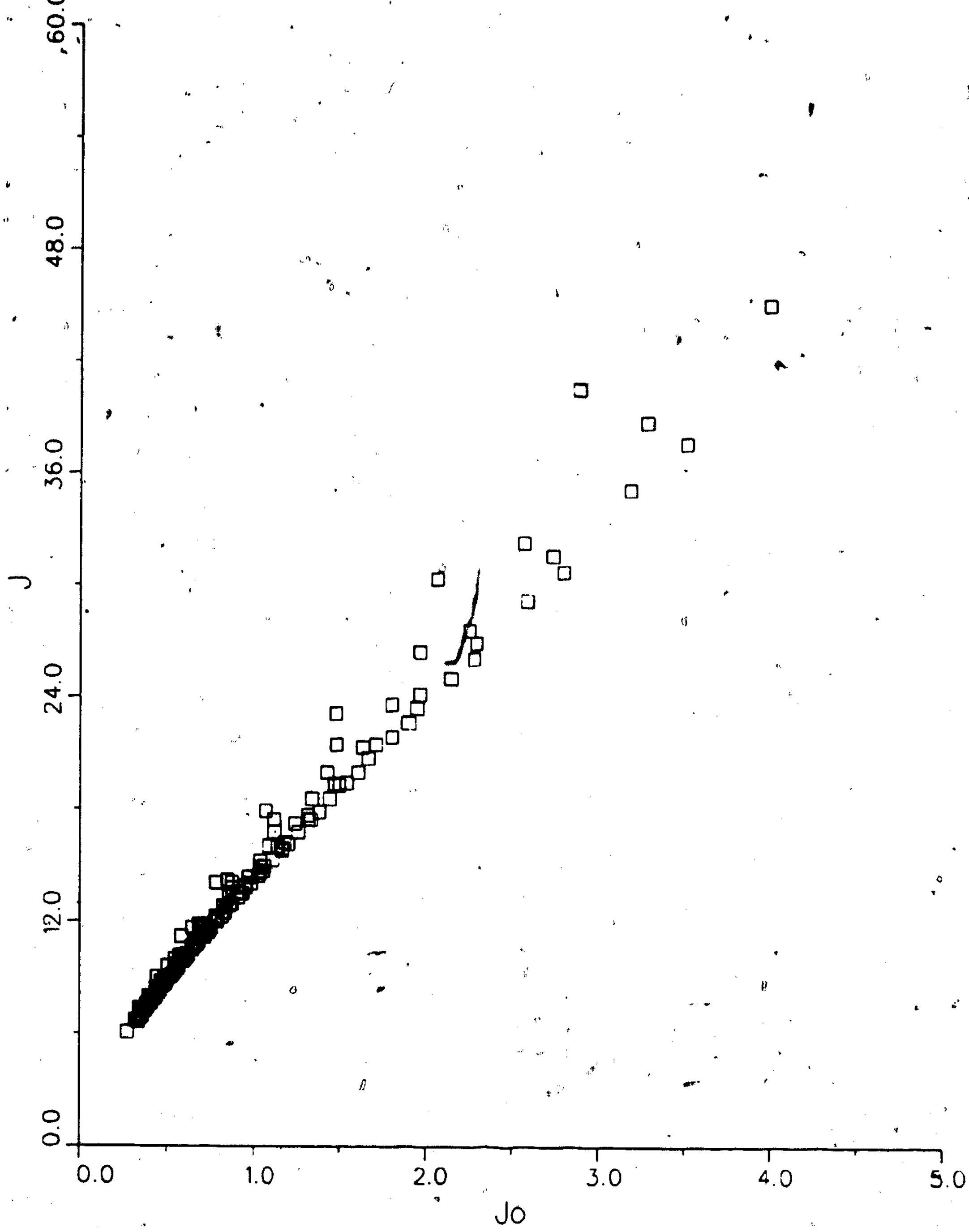


Fig. 3.22 2nd Order A1, 3rd Order Plant, Different U.D.

eliminating instability completely, does perform much better than the first order one in most of the unmodeled poles tried. Especially when the unmodeled dynamics modes are relatively important, adding one order to the control law A1 improves the system performance greatly. When the relative dominance of the real pole is very high, however, changing the first order control law to a second order one degrades the performance a little. It is understandable since when the complex pole pair is relatively unimportant, the plant behaves rather like a first order model. Applying a second order control law on such a plant is like overestimating the plant order in designing the system (c. the results obtained in Section 3.2.).

3.5 Discussions and Conclusions

In this chapter, simulations have been conducted with plant model (3.11) and the algorithm A1 in its second order form. In Sections 3.2. and 3.3., simulations run parallel to those conducted in Sections 2.2. and 2.3., with a higher order plant and a higher order algorithm. All the results and cnclusions of the previous chapter which are reached in a lower order situation has been confirmed.

Besides confirming the conclusions of the previous chapter, further explorations which could only be performed in higher order situations are conducted, with the following results:

1. Controlling a lower order plant with a higher order algorithm degrades system performance somewhat, but does not make the system unstable;
2. The algorithm A1 always has better stability assurance in a higher order form than in a lower order one, but does not necessarily give better system performance even if both orders are lower than the plant order;
3. For a plant whose dynamics include distinctive groups of poles that are dominant and non-dominant, the algorithm A1 might perform better if its order is set at the number of dominant poles in the plant than a higher order, provided that the higher order does not cover all the plant dynamics;
4. For a plant whose dynamics could not be divided into dominant and non-dominant pole groups, if the algorithm A1 has to be used in a lower order than the plant order, the higher its order the better.

These results could be simply stated as: when algorithm A1 is used in a system, it is not always true that the higher its order is, the better the system performance will be. For a designer of a control system using A1, the choice of its order is very important. Also, the guideline of "the greater T is the more stable a system will be" is not always correct either.

These results are reached without any consideration of possible disturbances. Greater T and higher order controller mean more trouble if disturbances are present because the slower response and more complications in system structure. The choice of T and the order of the algorithm should be done very carefully.

4. Further Evaluations of Algorithms A1 and A2

For any control approach to be regarded as useful, it has to perform better than other approaches at least in some situations. As for the algorithm A1 examined in this thesis, we have not so far pinpointed the class of situations where it is superior to other algorithms. This we shall attempt to do in this chapter.

As mentioned in Chapter 1., adaptive control schemes are usually designed to cope with uncertain plant parameters; not uncertain plant orders. The uncertainty of a plant, presuming it is linear, unfortunately could be in both its parameter and its order. In this chapter, studies will be done to observe the effect of addition of adaptation into algorithm A2 (ie., the introduction of A1). The two cases, namely the plant order uncertainty and the plant parameter uncertainty will be studied. Comparisons will be made between the performance of these algorithms to determine the situation or situations where the introduction of the adaptation scheme improves the system performance.

4.1 Uncertainty in The Plant Order

In Section 2.2., comparisons are made between the performance of A1 and A2, and the results show that when the plant order is underestimated, or when there are unmodeled dynamics in the system, A2 performs better than A1. The simulations in that section, however, do not cover enough ground. In this section more simulations will be conducted for the same purpose and different kinds of input signals will be used. Moreover, the performance of the algorithms A1 and A2 will be quantitatively compared. Unlike in Section 2.2., the unmodeled dynamics pole positions will be flexible in the studies of this section.

In the simulations that follow, algorithms A1 and A2 will be used in their first order forms, ie., n=1, m=0, and d=1. The plant is supposed to be described by the model (1.3.10) with a=1.0 and flexible α and β as unmodeled dynamics. The reference model is (2.1.1). Other conditions of the simulations are as follows:

$k_y(0)=-0.8$

$k_r(0)=1.32$

$T=0.04$ and

$f=0.2$

These conditions are consistent with Section 2.2..

4.1.1 Simulations with Rectangular Wave Input Signals

In this subsection, the reference input signal will be in the rectangular wave form. The signal has a period of 8 seconds. Under this reference input, the system output of plant (1.3.10) controlled by A1 and A2 are plotted and compared.

Figure 4.1 is the result of the simulation with the unmodeled poles at $-15\pm j2$ and control algorithm A1. Figure 4.3 is the performance of A2 under the same conditions. Figure 4.2 shows the result of the simulation with the unmodeled poles at $-10\pm j2$ and control algorithm A1, while Figure 4.4 shows the simulation with A2 in the same situations.

These results agree with the results of Section 2.2., ie., the algorithm A1 performs relatively poorly compared to A2 when there are unmodeled dynamics in the system. The system controlled by A1 has initial overshoot which are absent in the system controlled by A2. The A1 system with unmodeled poles at $-10\pm j2$ will actually go unstable if the simulation is continued (such a system is stable if r is a step input of 10, since $-10\pm j2$ is in the stable region shown in Figure 2.7). This shows that in adaptive control systems where unmodeled dynamics are present, an input signal which

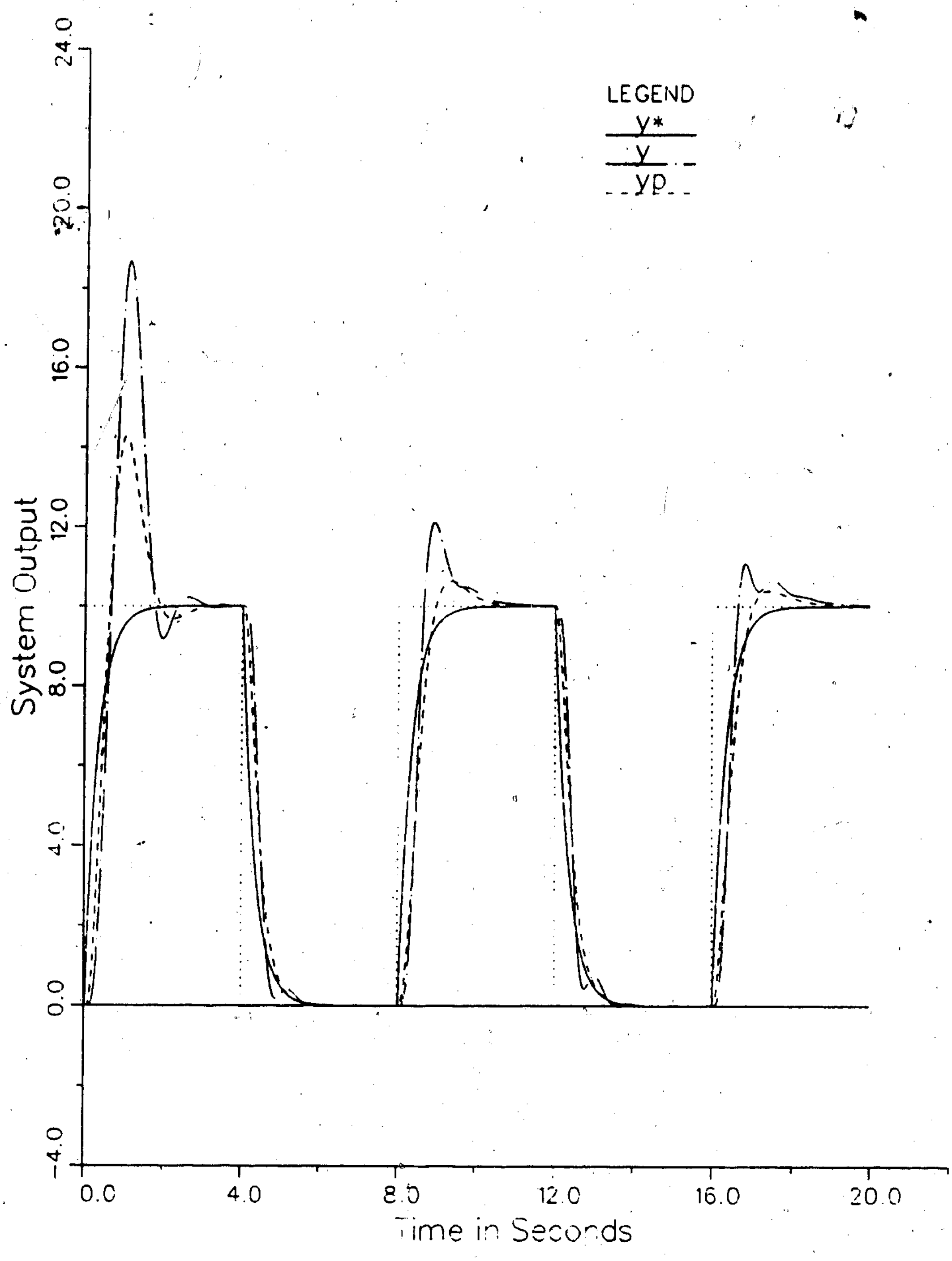


Fig. 4.1 Response of A1, U.D. at −15+j2, −15−j2, f=0.2, I.

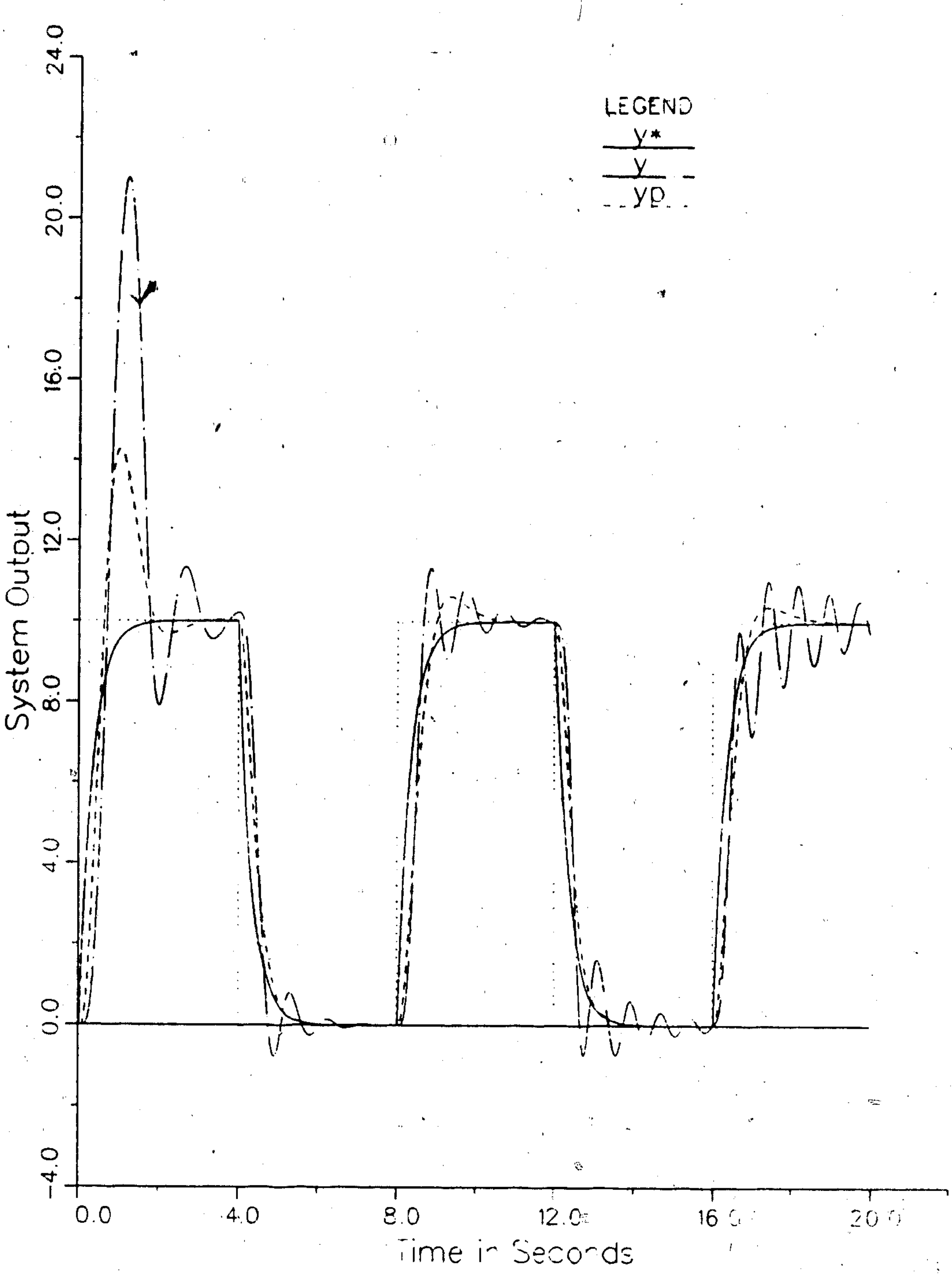


Fig. 4.2 Response of A1, U.D. at $-10+j2$, $-10-j2$, $f=0.2,-1$.

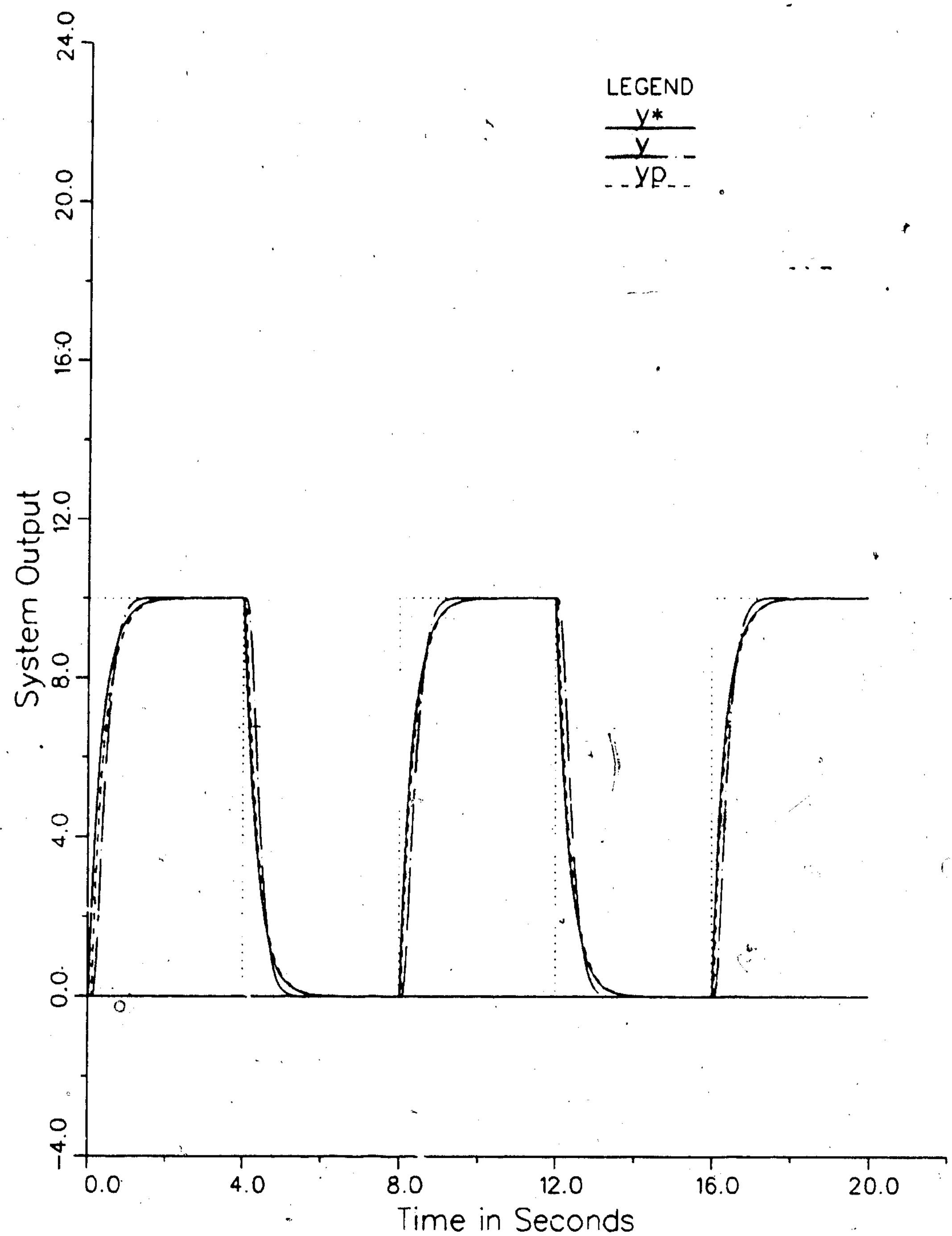


Fig. 4.3 Response of A2, U.D. at −15+j2, −15−j2, f=0.2, I.

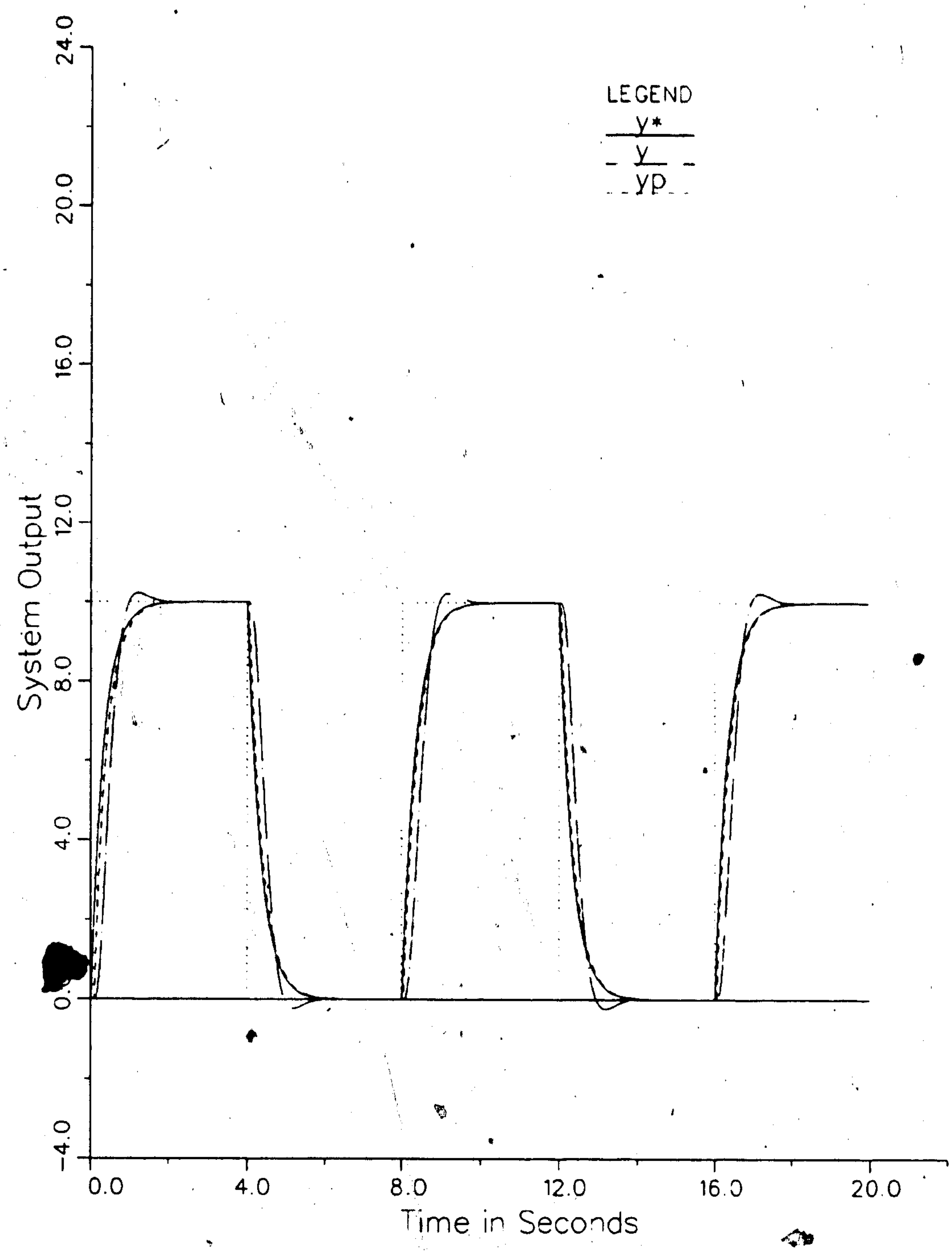


Fig. 4.4 Response of A2, U.D. at −10+j2, −10−j2, I.

offers continuous excitation of all the frequencies might make the system less stable by constantly renewing the information from the higher order modes. This information might otherwise havw been lost quickly since they are of higher frequency.

The presence of higher frequency modes as unmodeled dynamics seems to affect the non-adaptive model reference algorithm very little. A2 does a very good job even when the third order plant is presumed to be of first order although it fails to perform as a deadbeat controller.

4.1.2 Simulation Results with Sinusoidal Input Signals

In this subsection, the reference input signal will be in the sinusoidal wave form. The signal is again similar to that used in Section 3.2.. The following figures show the simulation results of the plant (1.3.10) controlled by A1 and A2.

Figure 4.5 is the result of the simulation with the unmodeled poles at -15 ± 2 and control algorithm A1. Figure 4.7 shows the performance of A2 under the same conditions. Figure 4.6 shows the result of the simulation with the unmodeled poles at -10 ± 2 and control algorithm A1, while Figure 4.8 shows the results with A2 in the same situation.

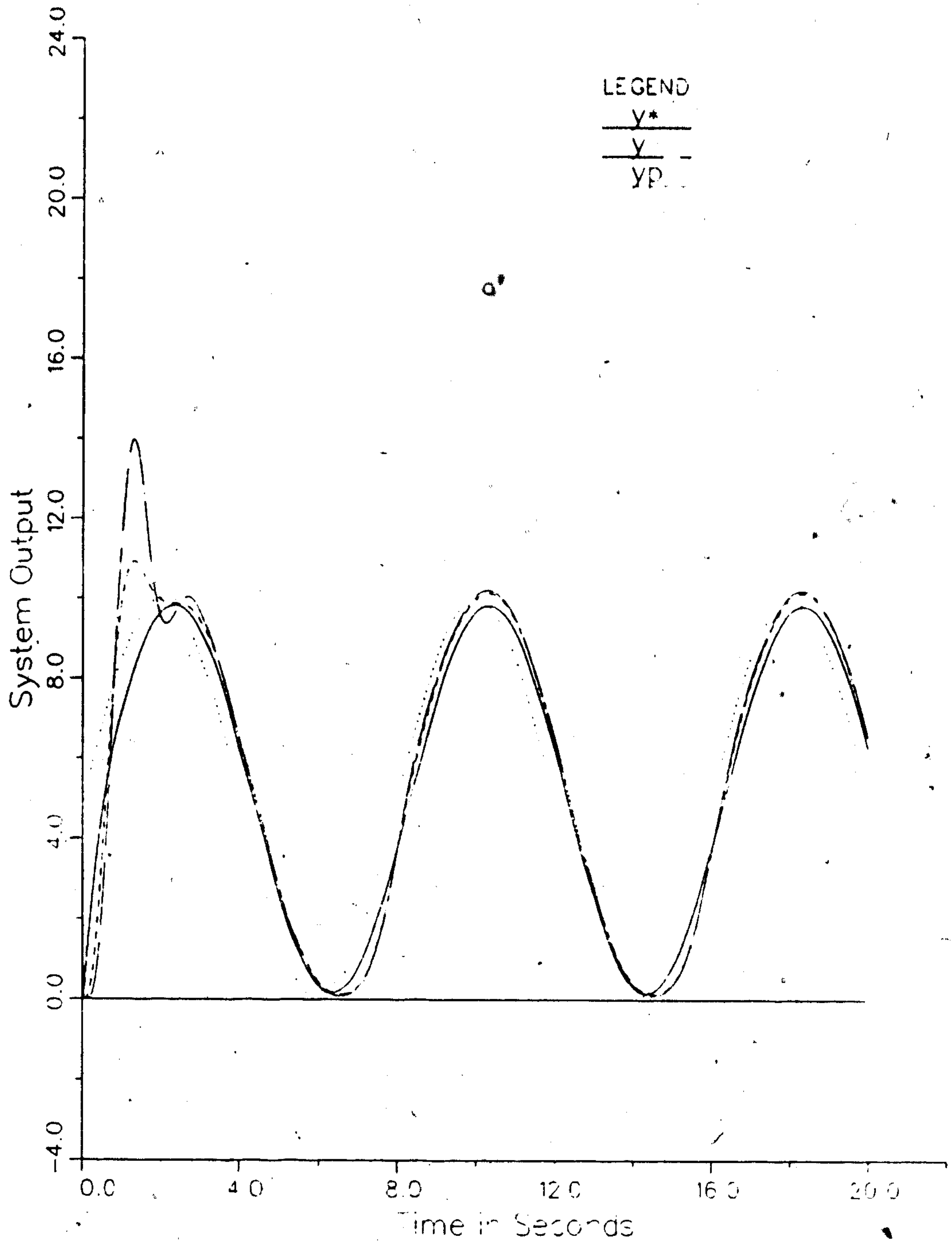


Fig. 4.5 Response of A1, U.D. at −15+j2, −15−j2, f=0.2, II

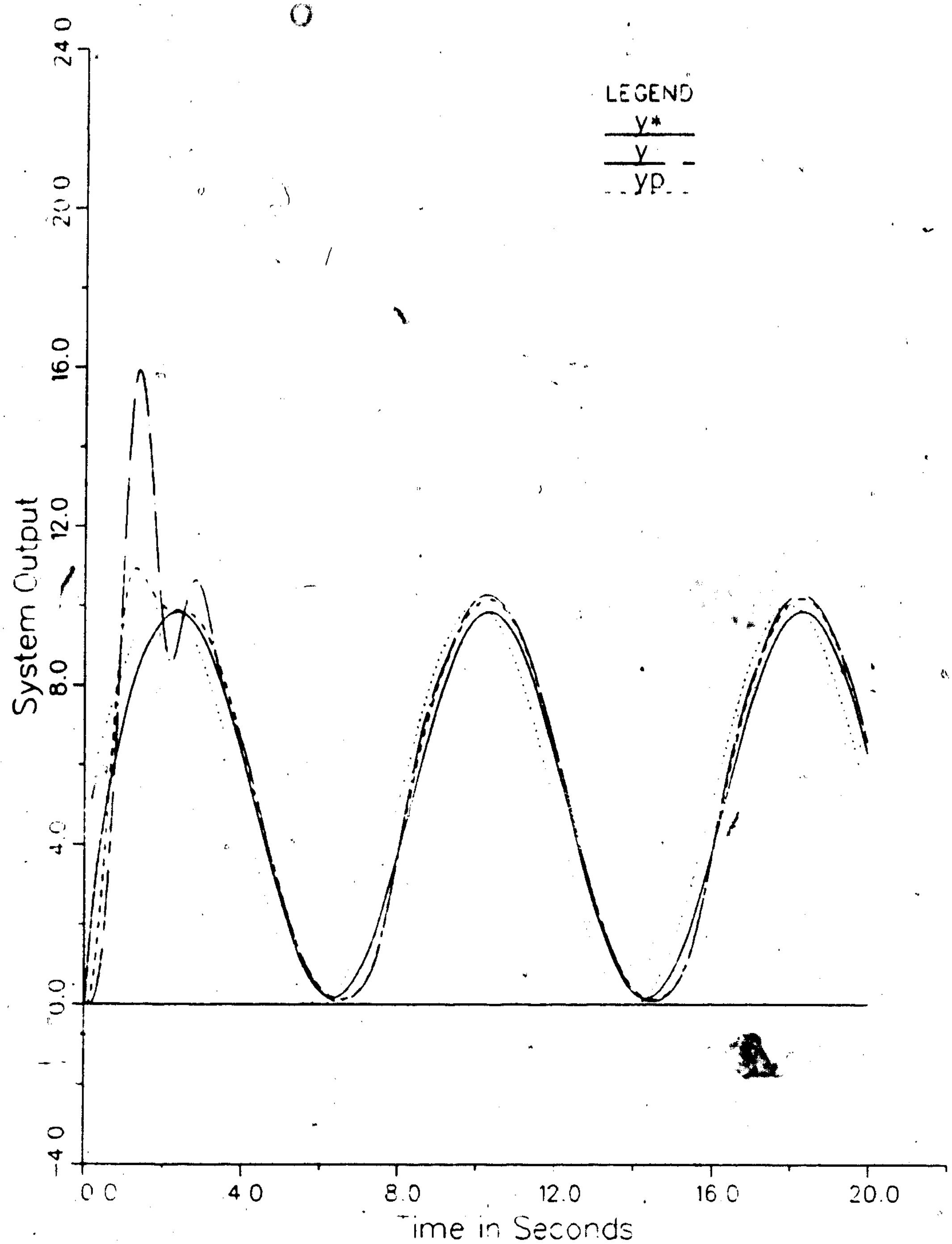


Fig. 4.6 Response of A1, U.D. at −10+j2, −10−j2, f=0.2, II.

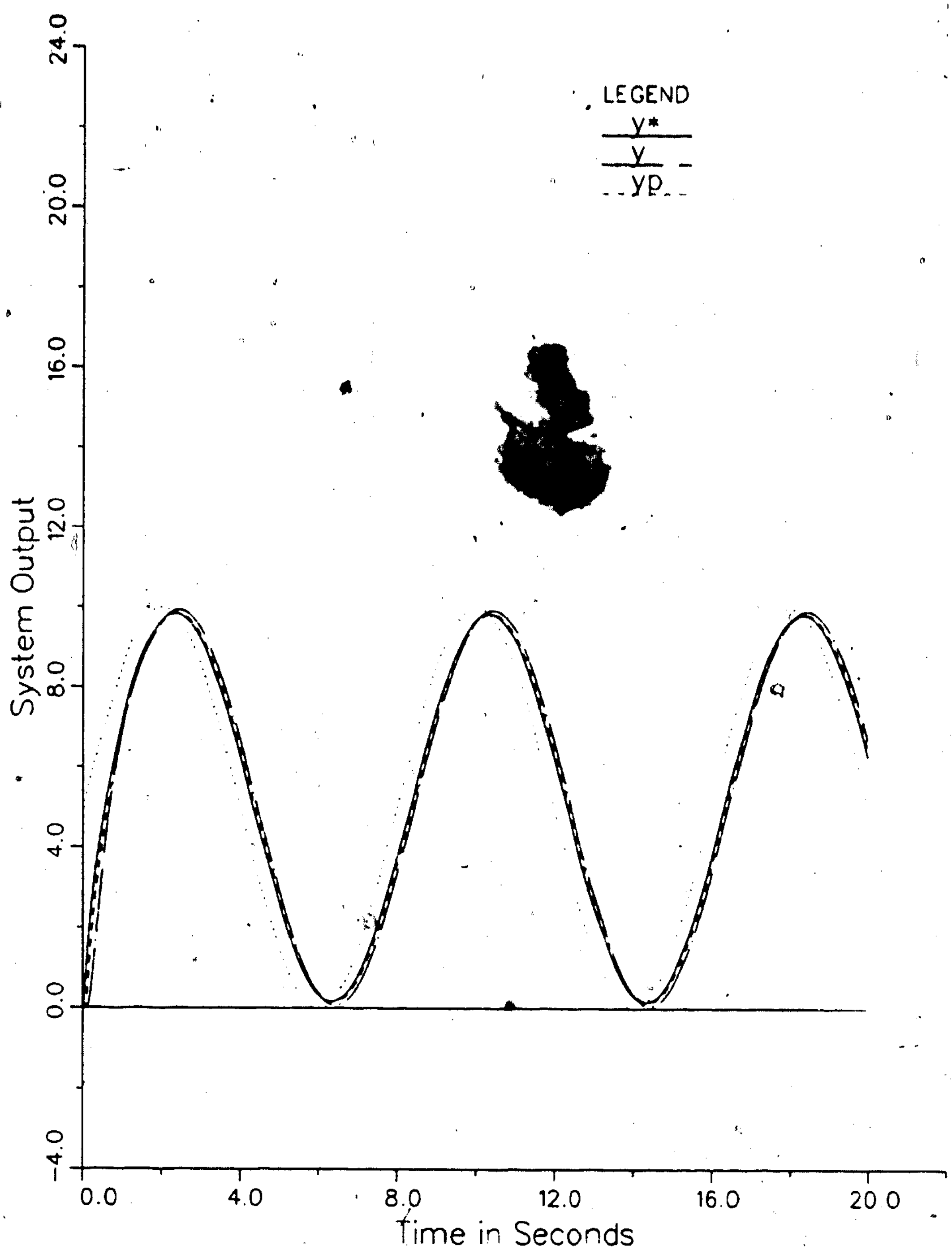


Fig. 4.7 Response of A2, U.D. at −15+j2, −15−j2, II.

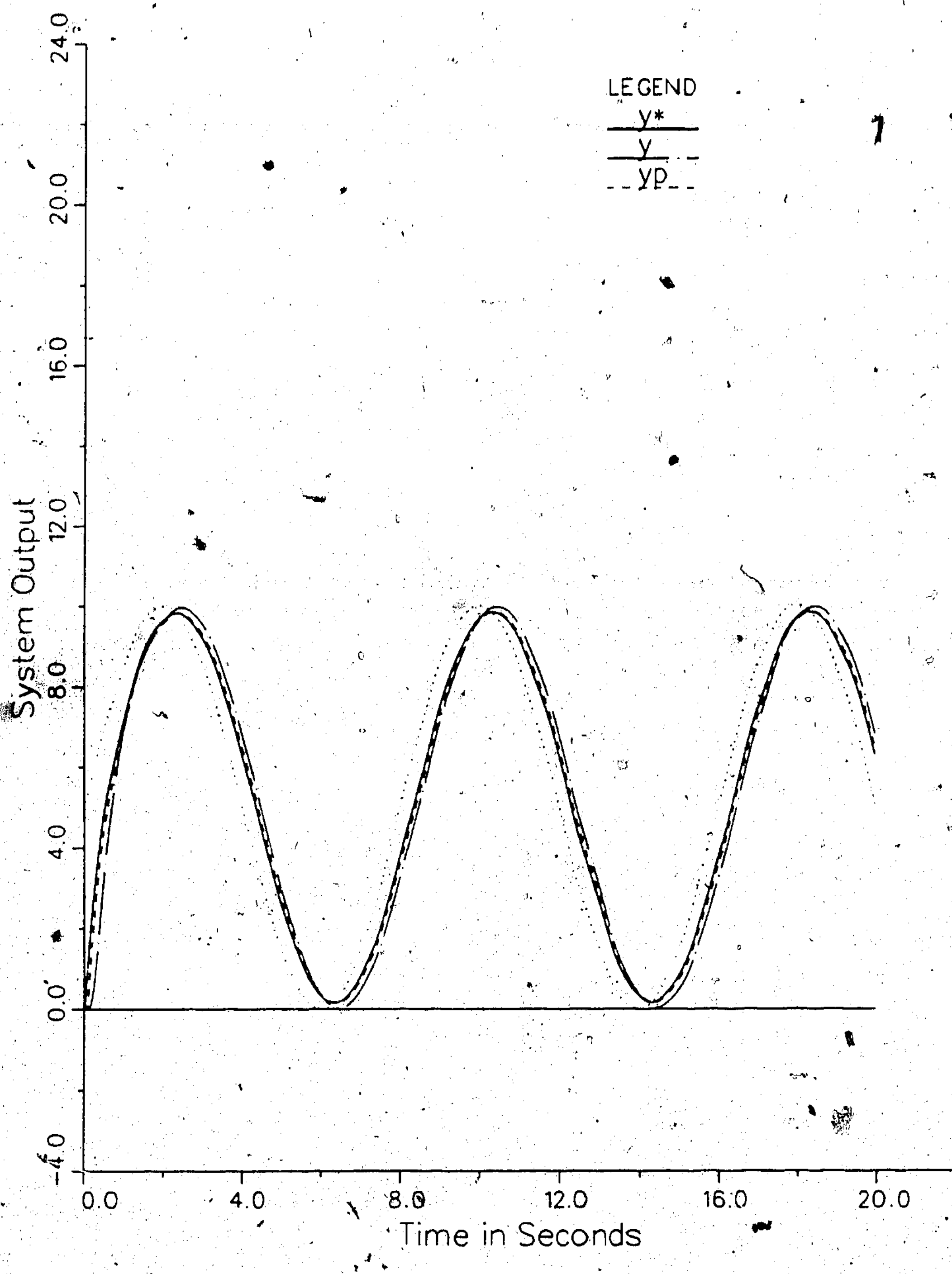


Fig. 4.8 Response of A2, U.D. at −10+j2, −10−j2, II.

These results again show that the algorithm A1 does not perform as well as A2 when there are unmodeled dynamics in the system. The overshoot is again the biggest difference. Again the unmodeled poles made little difference in the performance of A2.

4.1.3 Comparisons of the Quadratic Performance Index of A1 and A2

The results so far have uniformly supported the conclusion that algorithm A1 does not perform as well as A2 when the plant order is underestimated, ie, when there are unmodeled dynamics in the system. If this view is supported by studies of a large number of unmodeled pole positions in a quantitative sense, the conclusion will be well justified.

Recall that in Sections 2.3. and 3.2. the quantity J is used as a performance index for the control algorithm used. The same approach is used here. For ease of comparison, simulations here are done under the same conditions as in Section 2.3.. The unmodeled pole positions are changed within the area shown in figure 2.7, and the definition of J, SR, and J_o remain the same as before. The only difference is that in the present simulations the system is controlled by the algorithm A2, instead of A1.

The simulation results are shown in Figures 4.9 and 4.10. In Figure 4.9, the performance measure J is plotted versus the separation ratio SR, while in Figure 4.10 it is plotted versus the plant open-loop deviation J_o.

It is clear when comparing these results with those shown in Figures 2.8 and 2.10, that the unmodeled dynamics adversely affect the performance of A1 much more than they do that of A2. Especially when the system controlled by A1 is approaching instability, the superiority of A2 is significant. Nowhere in this range of unmodeled dynamics pole positions does algorithm A1 perform as well as A2.

Note the fact that the performance measure J is not much numerically greater than J_o, which is the deviation of the plant open-loop response.

4.1.4 Discussion

From the results of the studies in this section, it can be said with confidence that when the plant model uncertainty concerns its order, adding gain adaptations into algorithm A2 makes the system performance deteriorate. When there are unmodeled dynamics in the system, the performance of A1 just does not begin to compare with that of A2. Furthermore, the application of A1 in cases where the plant

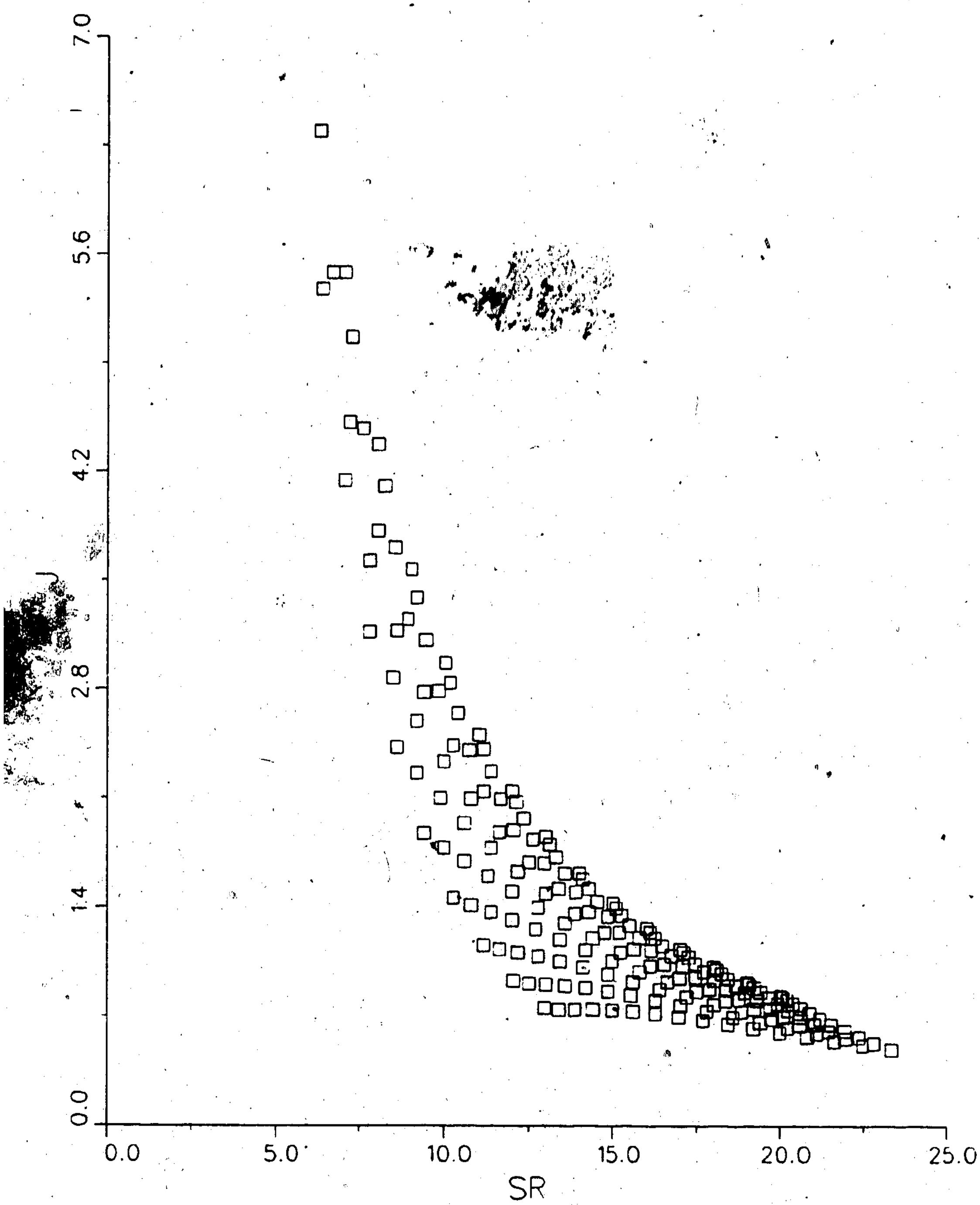


Fig. 4.9 Performance of A2 for Different U.D., I.

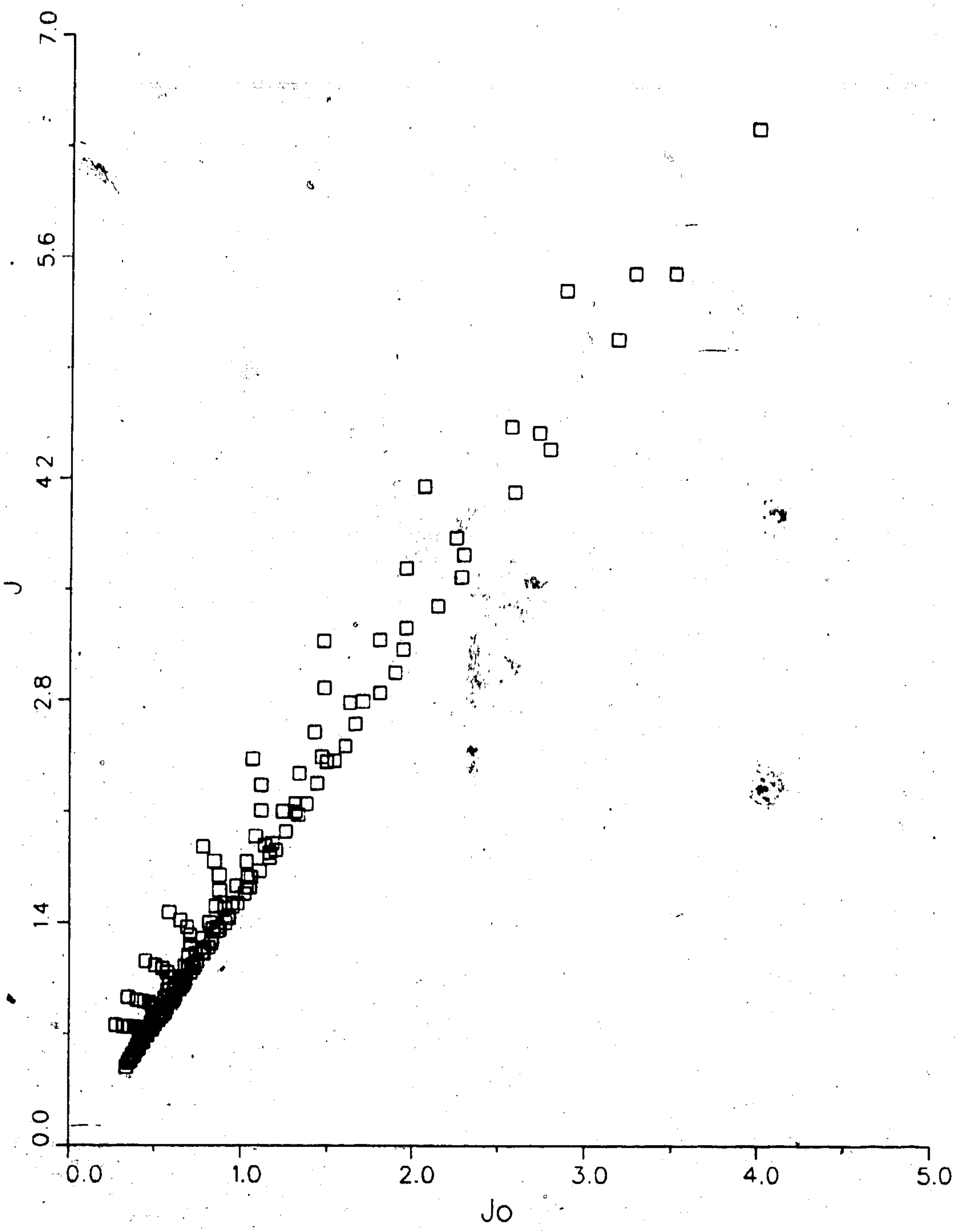


Fig. 4.10 Performance of A2 for Different U.D., II.

order is underestimated runs the danger of the final control system being unstable. Hence, if the order of the plant is not known but the position(s) of its dominant pole(s) are known, a better choice wold be control algorithm A2.

4.2 Uncertainty in Plant Parameters

In the studies above, it is presumed that the position of the dominant pole of the plant model is fixed and known before the control algorithms are designed, while the plant contains unknown higher frequency modes. In real applications, it is not always possible to establish the positions of the dominant poles of the plant. The dominant poles might even change in the process of control. This is precisely the situation that the adaptive control scheme is designed to cope with.

In this section, simulations similar to those of the previous section will be conducted under different conditions, ie., the order of the plant model is presumed known while its parameters is assumed to be unknown.

In the simulations that follow, algorithms A1 and A2 will be used again in their first order forms, ie., n=1, m=0, and d=1. The plant is to be described by the model

$$(s+a)Y(s)=2aU(s) \tag{4.2.1}$$

where a is a variable which will be changed.

Other conditions of the simulations are the same as in the previous section. As much consistency as possible is kept this way.

4.2.1 With Step Input Signals

The reference input signal of this subsection is a step function, ie., $r(t)=10.0$ for $t \geq 0.0$. The fixed parameters of the algorithm A2 are designed to make the system have deadbeat response when the plant parameter a is 5.0. This is to say that the estimate of the plant pole position is at -5 on the s plane before the controller is designed. This particular design of A2 will be used throughout Section 4.2..

As for the adaptive control algorithm A1, the pole position of the plant model is not required to be known. Hence the design of A1 only used the order of the plant, ie., $n=1$, $m=0$, and $d=1$. The position of the plant pole, denoted by a, is changed in the following simulations, which means that the estimate of $a=5$ could be inappropriate for the design of A2.

The reference model for the system to follow is:

$$(s+8)Y(s)=8U(s) \tag{4.2.2}$$

The simulation results are shown in Figures 4.11 to 4.13. In 4.11, a is set at 1. In 4.12, a is set to be 5, while Figure 4.13 shows the system output when a=15. In these figures, y_1 denotes the output of the system controlled by A1, while y_2 denotes that of the system controlled by A2.

These results show that the algorithm A1 works well for all the pole positions at -1, -5 and -15 of the s plane. The algorithm A2 however, does not work uniformly well. For instance, at a=5, it performed very well, acting as a deadbeat controller. At a=1, however, $y_2(t)$ lags behind $y^*(t)$ quite a bit, while at a=15, $y_2(t)$ leads $y^*(t)$. Although the overshoot shown in y_1 is absent in y_2, it could not be said that A2 performed better than A1. It is apparent that the performance of A2 changes more than that of A1 when a is changed. In other words, A2 is more sensitive to changes of plant parameters than A1.

4.2.2 With Rectangular Wave Input Signals

In this subsection, the reference input signal will be in the rectangular wave form, with a period of 8 seconds.

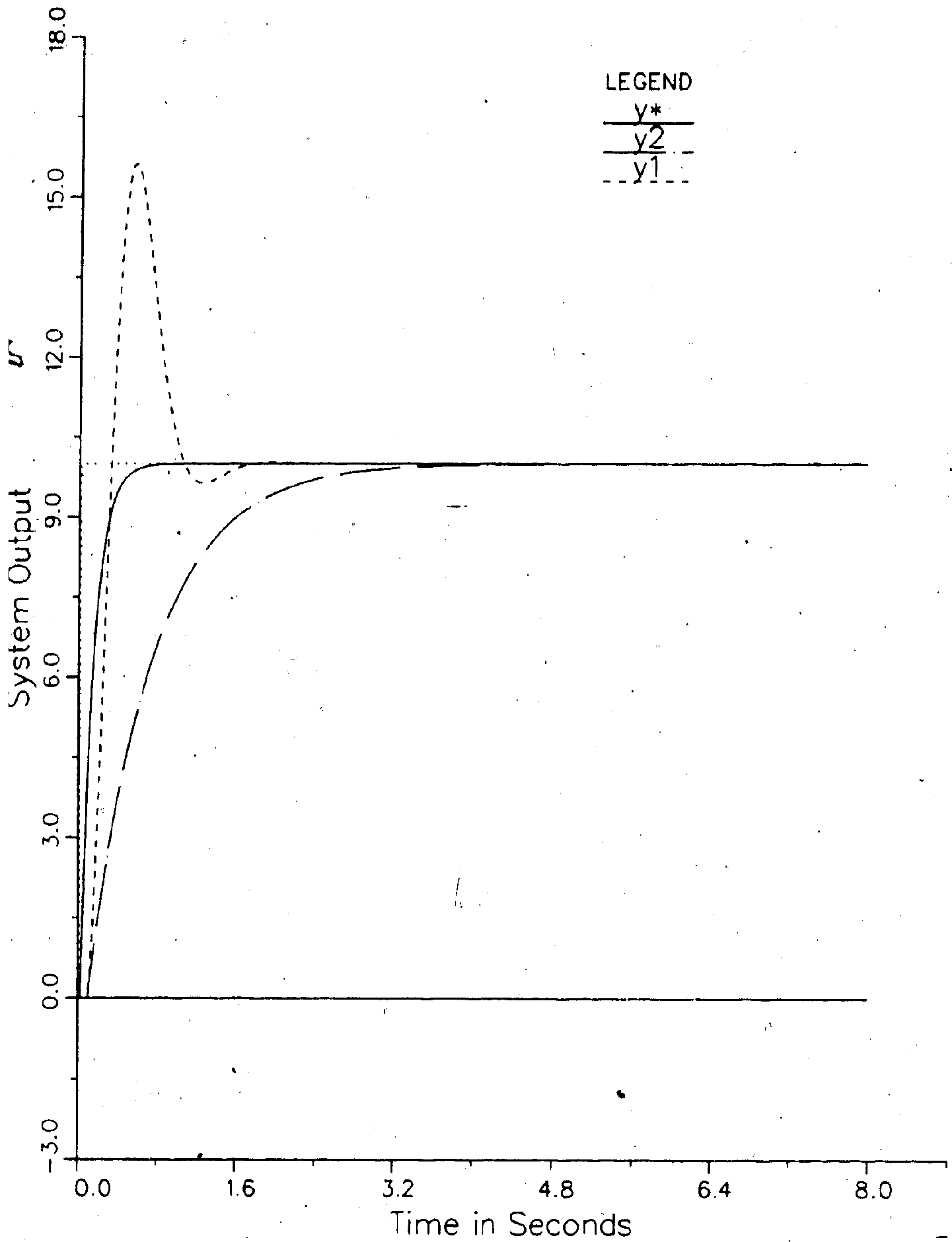


Fig. 4.11 Comparison of A1 and A2, f=0.6, a=1.0, I.

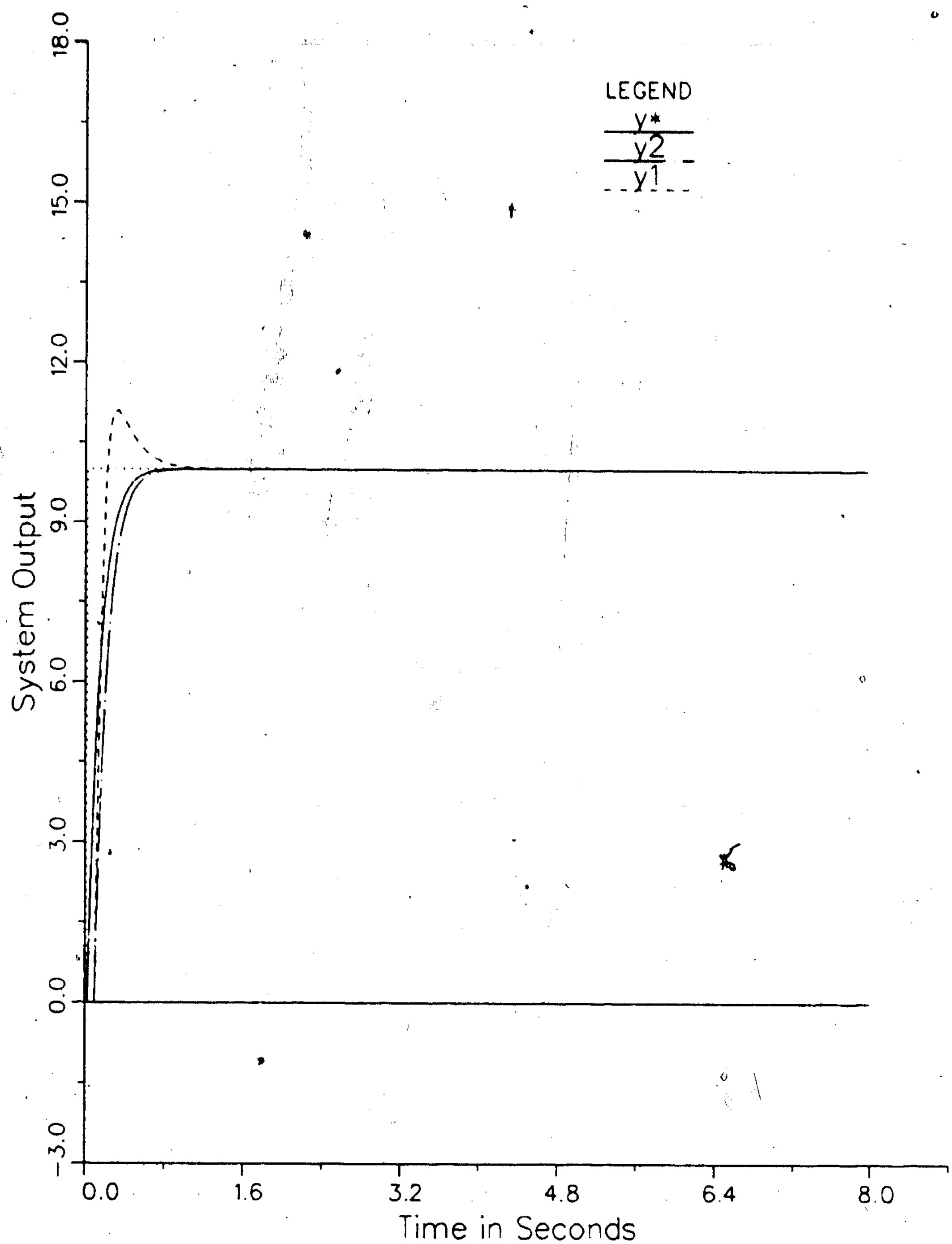


Fig. 4.12 Comparison of A1 and A2, f=0.6, a=5.0, I.

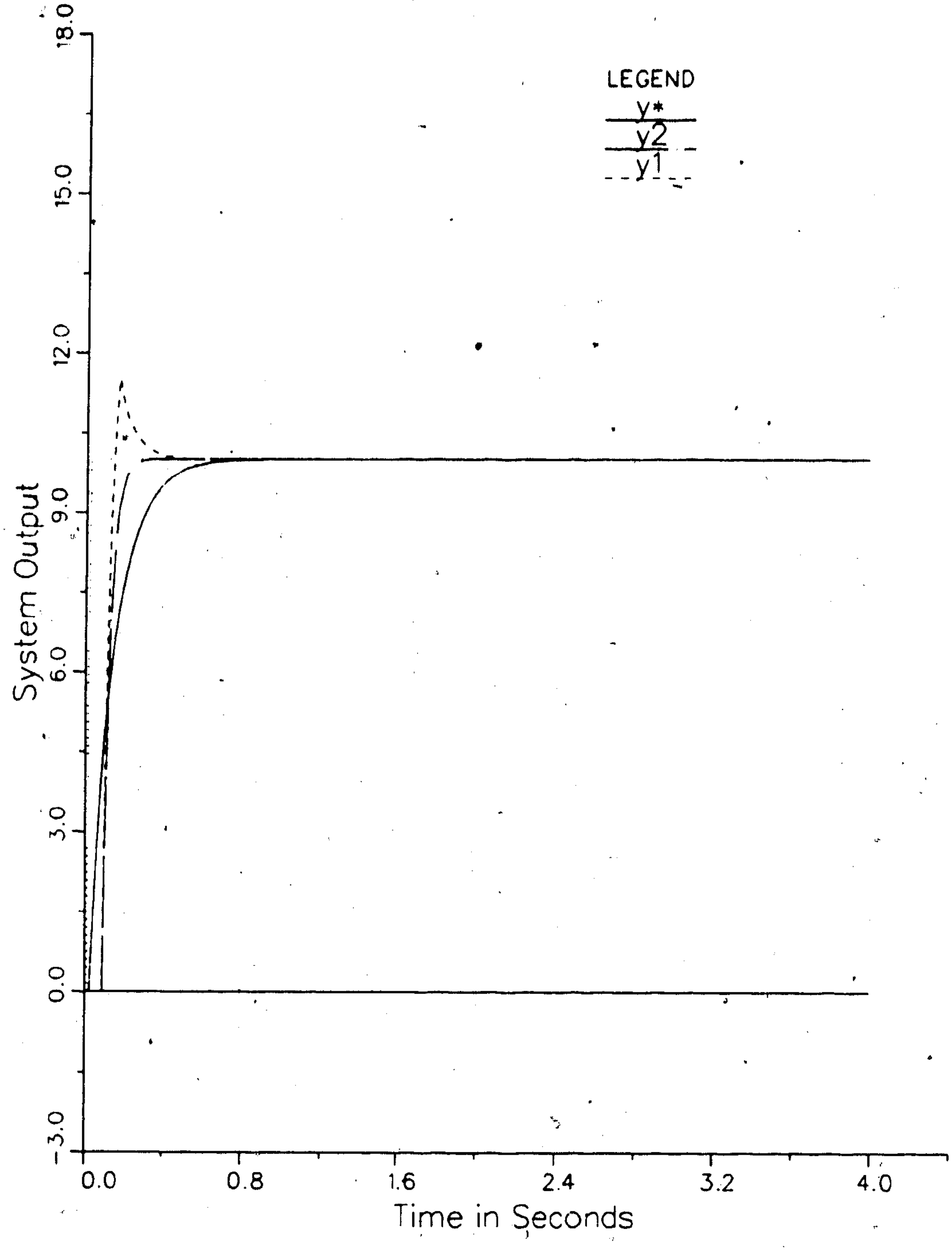


Fig. 4.13 Comparison of A1 and A2, f=0.6, a=15.0, I.

With this reference input, the system output of plant (4.2.1) controlled by A1 and A2 are plotted and compared.

In Figure 4.14 are shown the results of the simulations with plant pole at -1 on the s plane. Figure 4.15 shows the results of simulations with a=5 or plant pole at s=-5. Figure 4.16 shows the results with a=15. The variables y_1 and y_2 take on the same meanings as in the previous subsection.

Here it is very clear that the rectangular wave signal is a type of reference input signal under which the adaptive control scheme is superior to conventional controls. In both cases when the plant pole is incorrectly estimated, A1 performs much better than A2. The fixed control gains of A2 produce the same response each time the input level is changed, while the adjustment of gains in A1 makes it possible for the system to follow the reference model much more closely in the latter cases of changes in the input level. When the estimate of the plant pole is correct, A1 eventually is adjusted to A2, and their performance are the same asymptotically.

4.2.3 Comparison of the Quadratic Performance Measurements

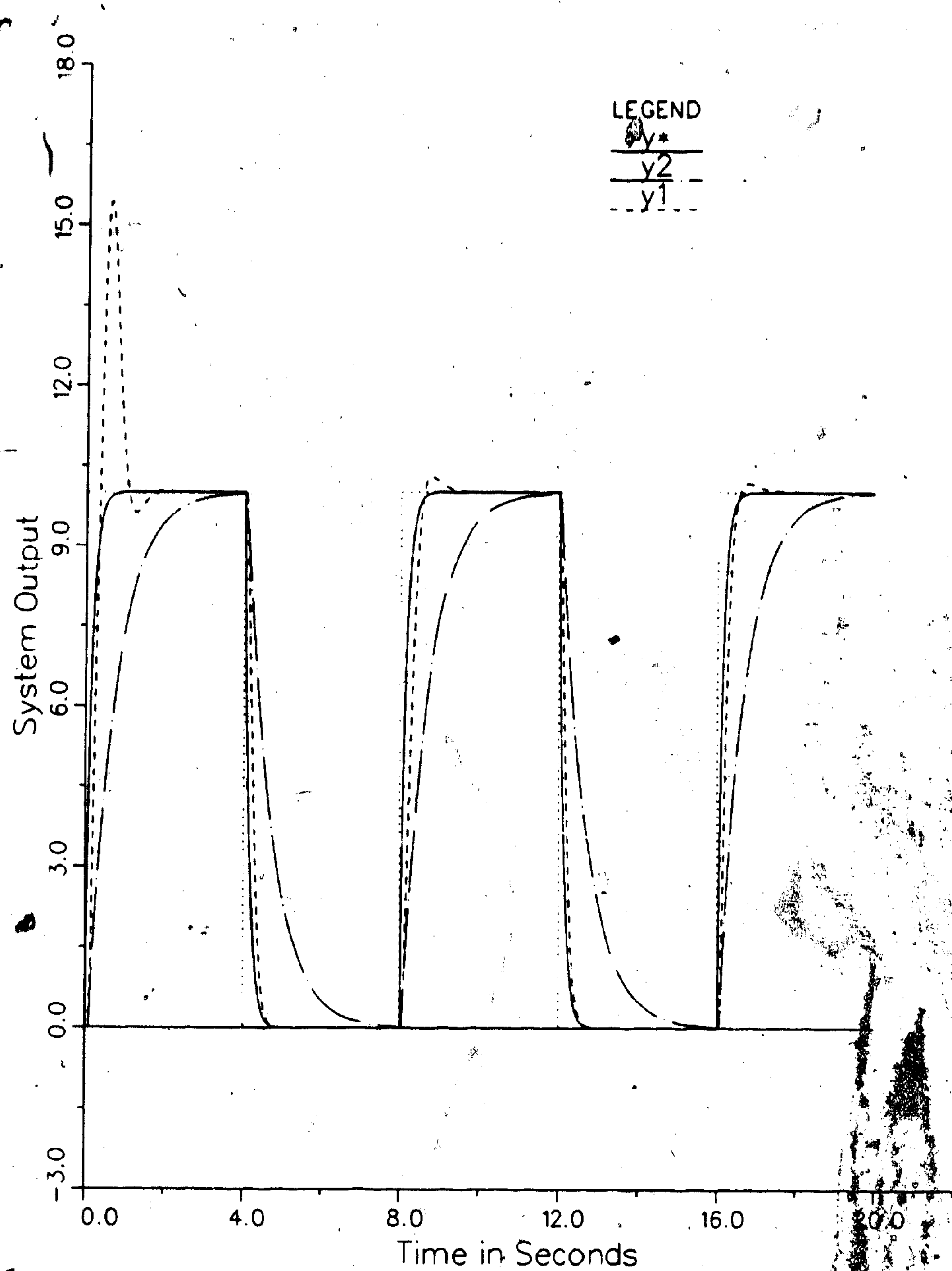


Fig. 4.14 Comparison of A1 and A2, f=0.6, a=1.0, II.

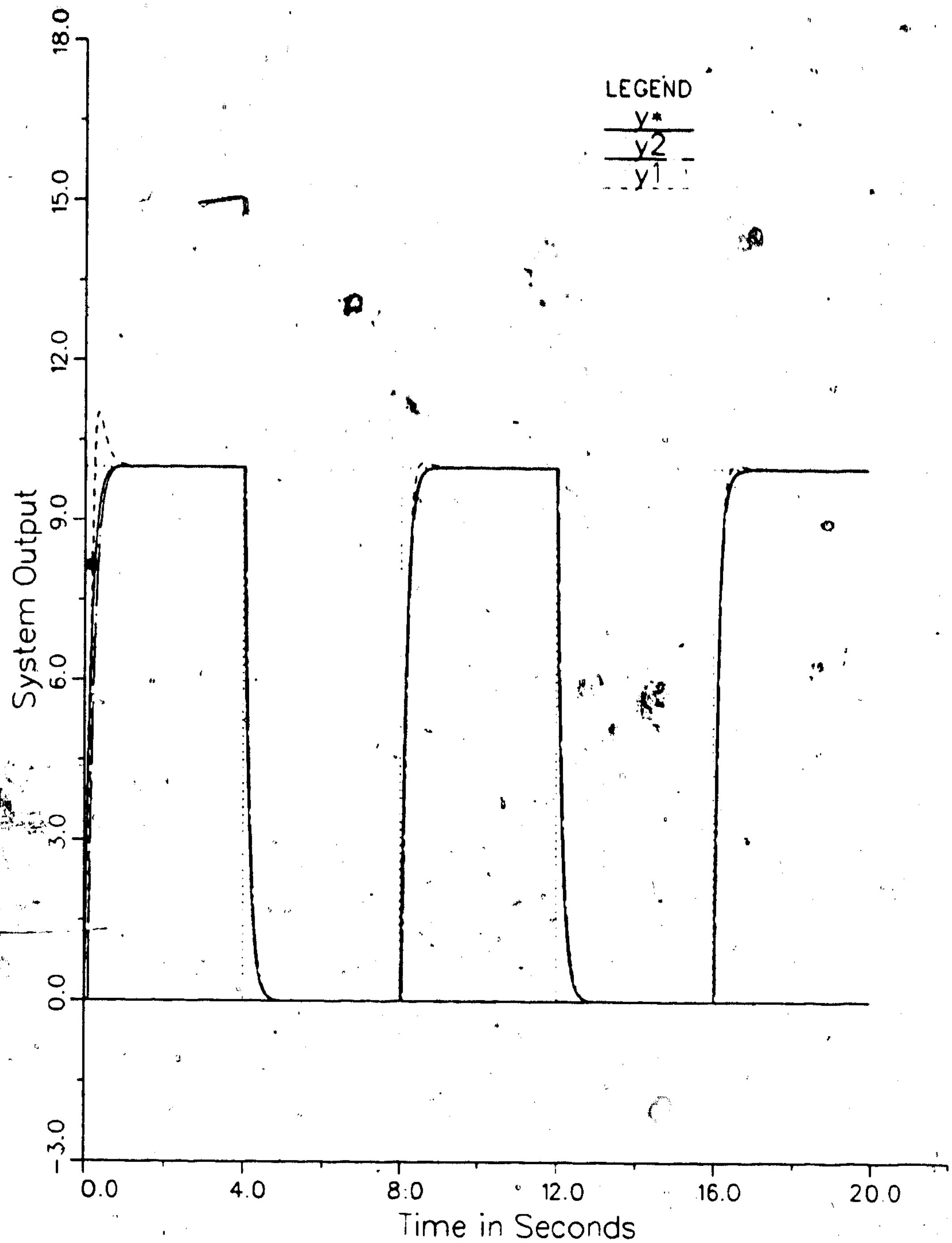


fig. 4.15 Comparison of A1 and A2, f=0.6, a=5.0, II.

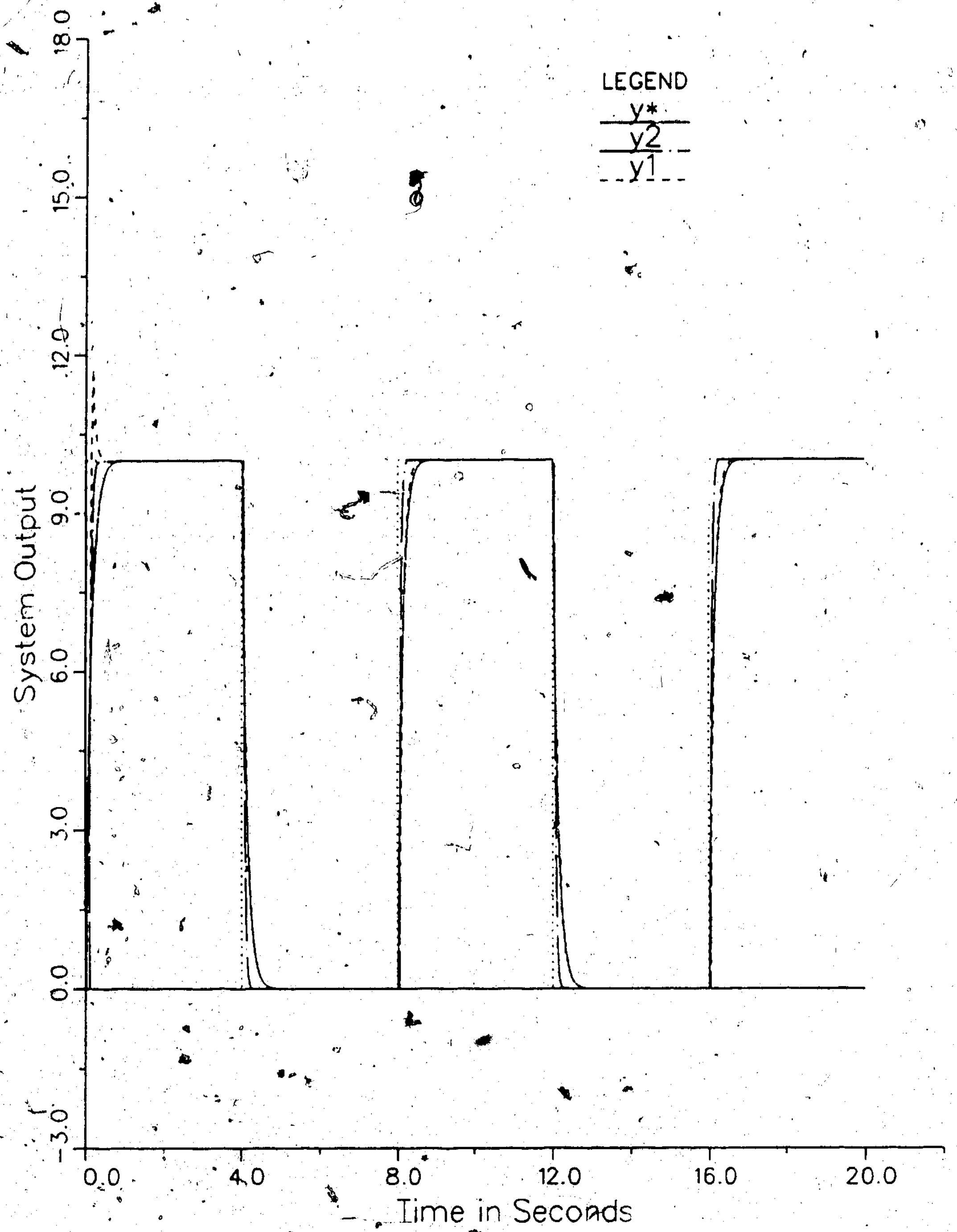


Fig. 4.16 Comparison of A1 and A2, f=0.6, a=15.0, II.

In order, to see how different positions of the plant pole affect the performance of algorithms A1 and A2, the following measures are introduced:

$$J_2=\int_0^{\infty}[y_2(t)-y^*(t)]^2dt \qquad (4.2.3)$$

$$J_1=\int_0^{\infty}[y_1(t)-y^*(t)]^2dt \qquad (4.2.4)$$

Note that these are similar to the definitions in (3.2.3-4). The definitions of the output variables y_1 and y_2 are however different here from those in Chapter 3.. These quantities measure the quadratic deviations of the system output y's from the reference model output y^*.

In Figure 4.17, the reference input is r=10.0; A2 is designed for the plant to follow the reference model (4.2.2) if a=5; for A1, the adjustment factor is set to be f=0.6. In Figure 4.18, the same conditions are used except that the input signal is a rectangular wave function. In this case, since the J quantities defined by (4.2.3-4) are not bounded, the integration is only done for the first 20 seconds of the simulations, which covers 2.5 periods of the input signal.

From these results, it could be seen that algorithm A2 does not work as well as A1 if the estimate of the plant model parameter a is far off, especially when the plant is actually slower than what is estimated to be. If the reference input is in the form of rectangular waves, the

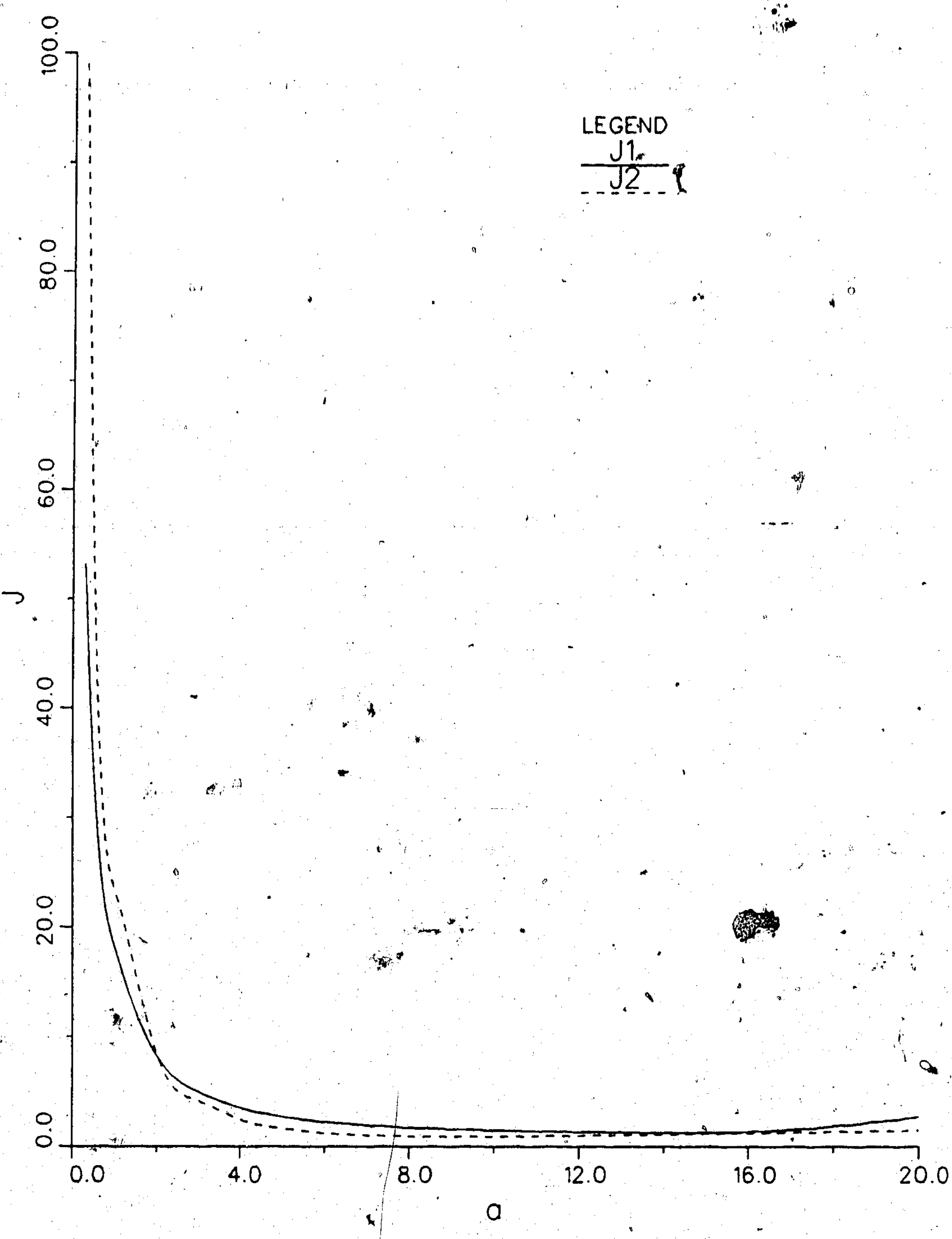


Fig. 4.17 Performance of A1 and A2, f=0.6, r=10.0

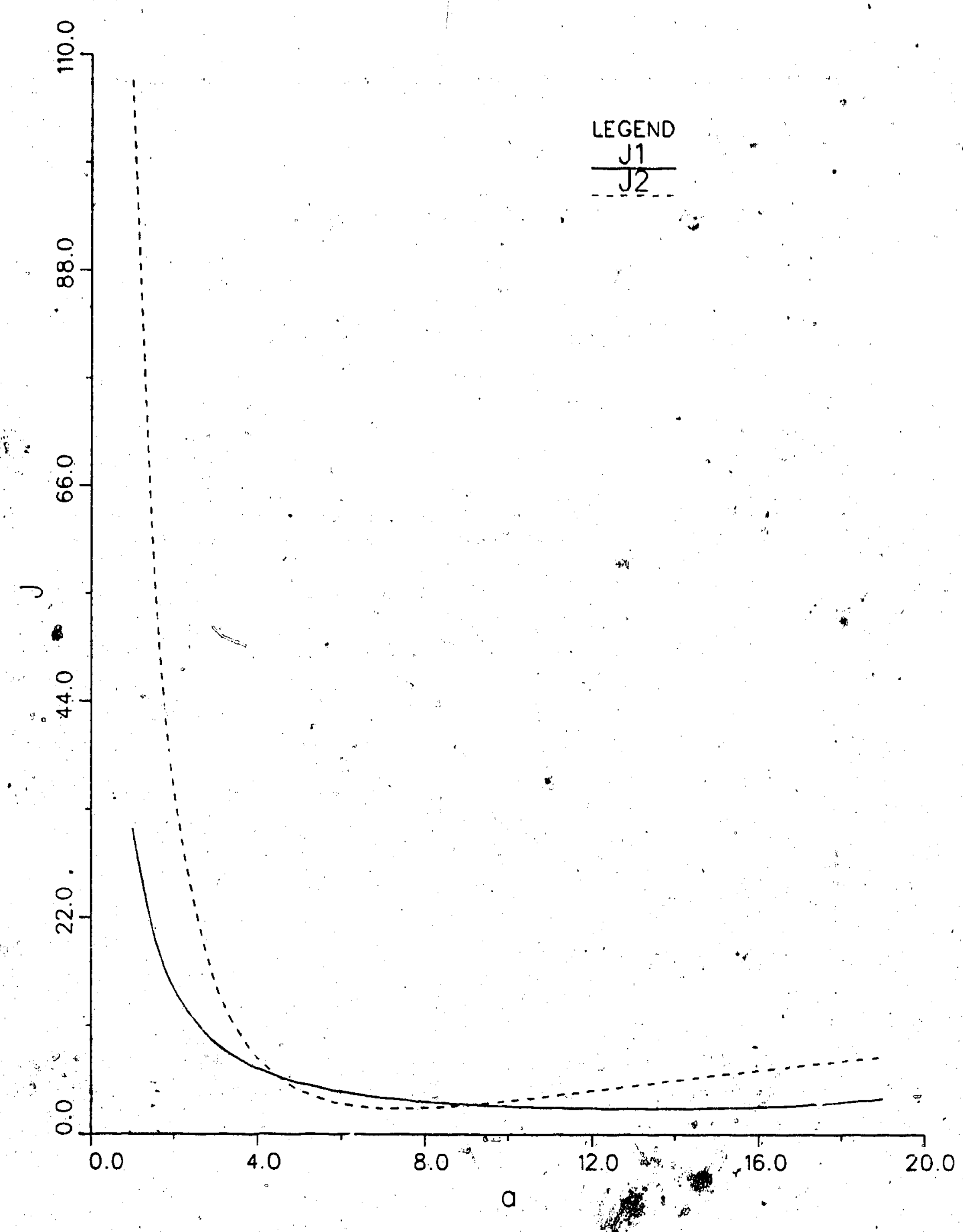


Fig. 4.18 Performance of A1 and A2, f=0.6, Rect. Wave Input

superiority of A1 to A2 is more marked. If the actual plant pole is near its estimate or a little faster, however, the performance of A2 and A1 is roughly the same with J_1 slightly greater than J_2 because of the initial overshoot in the systems controlled by A1.

4.2.4 Discussion

The simulation results in this section have shown that the algorithm A2 is more sensitive to changes of the plant parameter than A1, especially when the reference input signal has level changes like in a rectangular wave signal. If A2 is used as the control algorithm, the system response will either lag or lead the reference model response, if the estimate of the plant pole is incorrect. This happens in all the level changes in the reference input signal. On the other hand, if A1 is used, after the initial adjustment of the control gains is done, the controller performs like a deadbeat one for future level changes of the input.

4.3 Conclusions

By comparing the performance of A1 and A2 with two different kinds of plant uncertainty, we have been able to find the appropriate application range of each. The advantages and disadvantages of the two algorithms are now clear. The addition of the adaptive scheme into a model reference control approach like A2 improves the system performance in some specific operating conditions. The system performance is degraded in some other situations. Here is a brief summary of the conclusions of this chapter:

1. When the plant model's uncertainty involves its order, ie., when there are unmodeled dynamics in the system, the model reference control approach works better in its non-adaptive form than its adaptive form. It is more so when the dominant dynamics of the plant are constant and known. The adaptive algorithm A1 shows a lot more sensitivity to the plant order uncertainty than the non-adaptive A2.
2. When the structure of the plant model is known and the uncertainty is in the values of its parameters, the adaptive algorithm A1 does perform as well as or better than the non-adaptive A2. There are other factors to be considered in these comparisons. If the estimate of the plant parameter is wrong, especially if the actual plant is slower, A2 generally does not perform as well as A1.
3. From the simulation results of this chapter, the capability of the two algorithms to cope with changing plant parameters in the control process of the two algorithms could be deduced. A1 should be

much better than A2 in this respect.

4. Should the plant order change during the control process, A2 should do much better than A1 if the dominant dynamics of the plant do not change much. This is presuming that these dominant dynamics are known and used in the design of A2.

5. Final Disscusions of the Results

This thesis contais numerous simulation studies of the algorithm A1. The main part of this work is concentrated on the effect of unmodeled dynamics on the performance of systems controlled by A1. In this chapter the results obtained through the studies of the last three chapters are put together and discussed.

5.1 The Effects of Different Unmodeled Dynamics

In order to investigate the effects of different unmodeled dynamics on the performance of systems controlled by A1, plant models with variant unmodeled poles are introduced. With the position(s) of the dominant plant pole(s) constant, the unmodeled poles of the plant are moved around in a region on the s plane. The performance of the adaptive control control systems are then measured using index J, which is a quadratic quantity.

This quantity means little by itself. However in a comparative sense, it represents the relative system performance. For instance, the effect on system performance of the changes in a certain factor of the control law or the plant can be measured with the resultant changes of J. By keeping all the other factors in the system constant and changing only the unmodeled pole positions, we are able to confirm the presumption that generally, the more effect a particular group of unmodeled dynamics has on the plant open-loop response, the more is its effect on the adaptive control system.

Another quadratic quantity is used to measure directly the effect of unmodeled dynamics on the system gain values. The almost linear relationship this quantity (D) has with J (Figure 2.12) suggests that the degradation of system performance resulted from the presence of unmodeled dynamics is closely related to the changes in the control gains caused by it. The conclusion here is that the gain adjustment mechanism works erratically when there are unmodeled dynamics in the system.

5.2 The Effect of Employing Higher Order Control Laws

Since the presence of unmodeled dynamics in the system is caused by the underestimating the plant order while designing the control law, people have preferred higher order control laws when they have the choice. In this thesis simulations have been carried to examine the effectiveness of this approach.

The results in Chapter 3. show that generally speaking, using a higher control law make the system more stable in the presence of unmodeled dynamics. If the increase of the controller order eliminates unmodeled dynamics, the system will be rid of its unstable factor. If this order increase is not great enough to get rid of all the unmodeled dynamics, the system stability will still be improved.

The system performance measured by J, however, does not always improve with the increase of the control law order. Especially when the original controller order is already high enough, the further increase of it actually degrades the system performance somewhat. This happens because in a higher order control law there are more gains to adjust, which makes it harder to perform as well as the control law of the exact right order. This degradation of system performance is quite insignificant numerically, compared with the loss in the quality of the system performance caused by unmodeled dynamics.

5.3 The Effect of Changes in T on System Performance

Employing the quantitative measures of system performance, we are able to investigate the effect of using different control and adjustment intervals. Since unmodeled dynamics are usually high frequency modes, it is presumed that greater values of the control interval T help reduce their effect on system performance.

The simulation results in Chapters 2. and 3. support this presumption. However it is not always true that the increase of T makes the system perform better. System performance, as measured by J, improves very quickly with the increase of T if the effect of the unmodeled dynamics is relatively significant. When this effect is diminished after the increase of T, however, continuing to increase it does not improve the system performance any more.

There appears to be an optimal T value for each system controlled by algorithm A1 that contains unmodeled dynamics. Aimlessly using large values of T to cope with the unmodeled dynamics problem is unnecessary and sometimes undesirable even if there is no disturbance in the system. If the situation permits, the control interval T should be decided through trials. An appropriate method is not found to determine how much of the effect of T on system performance is linked to the fact that f is effectively dependent on T, and how much is due to the filter effect of larger T values. This might be an interesting topic for further research.

5.4 The Advantages of The Adaptive Scheme

The adaptive control approach was introduced to cope with plant parameter uncertainties. The simulation results of Chapter 4. show that it works well in this situation. No conventional approach could compare with the adaptive controller if the plant parameter changes widely in the control process, especially with frequent changes in the reference input level.

The changes in the plant parameters have to be relatively large, however, to justify the deployment of an adaptive algorithm like A1. If these changes are small, even a control algorithm like A2 which is relatively sensitive to plant parameter uncertainties among conventional controllers might perform well.

If a decision has to be made as to whether an adaptive control algorithm should be employed in a particular design situation, the type of plant uncertainty present has to be investigated. The adaptive approach is very sensitive to plant order uncertainties, while its advantage in coping with plant parameter uncertainties is not significant unless the parameters change in a very wide range. Only in the cases where the sensitivity of the conventional control approaches to plant parameters outweigh the sensitivity of the adaptive approach to plant orders should one consider the deployment of an algorithm like A1. This is likely to happen in cases where the plant order is fixed while its parameters might change widely.

REFERENCES

[1] Feuer, A. and Morse, A.S., "ADAPTIVE CONTROL OF SINGLE-INPUT SINGLE-OUTPUT LINEAR SYSTEMS", *IEEE Trans. Aut. Contr.*, Vol.AC-23, pp.557-570, 1978.

[2] Morse, A.S., "GLOBAL STABILITY OF PARAMETER ADAPTIVE CONTROL SYSTEMS", *IEEE Trans. Aut. Contr.*, Vol.AC-25, pp.433-440, 1980.

[3] Egardt, B., "STABILITY ANALYSIS OF DISCRETE-TIME ADAPTIVE CONTROL SCHEME", *IEEE Trans. Aut. Contr.*, Vol.AC-25, No.4, pp.710-716, 1980.

[4] Rohrs, C.E., *ADAPTIVE CONTROL IN THE PRESENCE OF UNMODELED DYNAMICS*, Ph.D. Thesis, MIT, 1982.

[4a] Rohrs, C.E., Valavani, L., Athans, M. and Stein, G., "ROBUSTNESS OF ADAPTIVE CONTROL ALGORITHMS IN THE PRESENCE OF UNMODELED DYNAMICS", *21st IEEE Conf. of Dec. & Contr.*, Orlando, 1982.

[4b] Rohrs, C.E., Athans, M., Valavani, L. and Stein, G., "SOME DESIGN GUIDELINES OF DISCRETE-TIME ADAPTIVE CONTROLLERS", *Automatica*, Vol.20, No.5, pp.653-660, 1984.

[5] Goodwin, G.C., Hill, D.J. and Palaniswami, M., "A PERSPECTIVE ON CONVERGENCE OF ADAPTIVE CONTROL ALGORITHMS", *Automatica*, Vol.20, No.5, pp.519-531, 1984.

[6] Egardt, B., *STABILITY OF ADAPTIVE CONTROLLERS*, Springer-Verlag, Berlin, 1979.

[7] Samson, C., "STABILITY ANALYSIS OF ADAPTIVE CONTROLLED SYSTEMS SUBJECT TO BOUNDED DISTURBANCES", *Automatica*, Vol.19, No.1, pp.81-86, 1983.

[8] Ioannou, P.A. and Kokotovic, P.V., ADAPTIVE SYSTEMS WITH REDUCED MODELS, *Lecture Notes in Control and Information Sciences*, Springer-Verlag, New York, 1983.

[9] Kosut, R.L. and Johnson, C.R., Jr., "AN INPUT-OUTPUT VIEW OF ROBUSTNESS IN ADAPTIVE CONTROL", *Automatica*, Vol.20, No.5, pp.569-581, 1984.

[10] Johnson, C.R., Jr., Anderson, B.O.D. and Bitmead, R.R., "A ROBUST LOCALLY OPTIMAL MODEL REFERENCE ADAPTIVE CONTROLLER", *Proc. 23rd IEEE Conf. on Dec. & Contr.*, Las Vegas NV, 1984.

[11] Glasson, D.P., *RESEARCH IN MULTIRATE ESTIMATION AND CONTROL*, The Analytic Sciences Corp., Mass. 01867, 1980.

[12] Amit, N., *OPTIMAL CONTROL OF MULTIRATE DIGITAL CONTROL SYSTEMS*, Stanford University Paper, SUPAAR #523, 1980.

[13] Astrom, K.J., "THEORY AND APPLICATIONS OF ADAPTIVE CONTROL - A SURVEY", *Automatica*, Vol.19, No.5, pp.471-486, 1983.

[14] Gourishankar, V.G., Trybus, L. and Rink, R.E., "AN ADAPTIVE FILTER FOR DYNAMIC POSITIONING", *Optimal Control Applications and Methods*, (accepted for publication, March, 1985).

[15] Harris, C.J. and Billings, S.A. (ed.), *SELF-TUNING AND ADAPTIVE CONTROL: THEORY AND APPLICATIONS*, Peter Peregrinus Ltd., 1981.

[16] Anderson, B.D.O. and Johnson, C.R., Jr., "EXPONENTIAL CONVERGENCE OF ADAPTIVE IDENTIFICATION AND CONTROL ALGORITHMS", *Automatica*, Vol.18, No.1, pp.1-14, 1982.

[17] Kosut, R.L. and Friedlander, B., "ROBUST ADAPTIVE CONTROL: CONDITIONS FOR GLOBAL STABILITY", *IEEE Trans. Aut. Contr.* (to appear).

[18] Riedle, B.D. and Kokotovic, P.V., "STABILITY ANALYSIS OF AN ADAPTIVE SYSTEM WITH UNMODELED DYNAMICS", *Int. J. Contr.* (to appear).

[19] Goodwin, G.C., Ramadge, P.J., and Caines, P.E., "Discrete Time Multivariable Adaptive Control", *IEEE Trans. Aut. Contr.* Vol. AC-25, pp. 449-456, June 1980

[20] Goodwin, G.C., and Sin, K.S., *ADAPTIVE FILTERING PREDICTION AND CONTROL*, Prentice-Hall, Inc., N.J. 07632, 1984