AUTOMATIC MODEL STRUCTURE DETERMINATION FOR ADAPTIVE CONTROL

By

Anat Kotzev

B.Sc. Chemical Eng., Technion (Israel Institute of Technology), Haifa, Israel, 1979

M.Sc. Chemical Eng., Technion, Haifa, Israel, 1982

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

MECHANICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

June 1992

© Anat Kotzev, 1992

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Eng.

The University of British Columbia
Vancouver, Canada

Date Aug 14, 1992

Abstract

This work is a study of adaptively controlled systems with plant model structures that may vary due to changing operating conditions. Most closed loop adaptive control algorithms use identification methods for determination of the parameters in fixed structure models. Those parameters, once estimated, are assumed to be correct and uncertainties in the values are ignored. If the structure of the plant dynamics changes on-line, the incorrect model can lead to poor performance and instabilities.

The adaptive algorithm used in this work is the Generalized Predictive Control (GPC) algorithm. It is reported to be capable of handling a number of simultaneous problems and therefore was chosen. Along with handling on-line changes of parameters, it claims to overcome nonminimum-phase plants, open loop unstable plants, plants with badly damped poles, plants with variable or unknown time delay, and plants with unknown order.

The goal of this research is to investigate and study GPC with the on-line changes in the model structure of the plant, and corresponding changes in the order of the estimated model for GPC and the structure of the controller and as well as to propose a method that detects on-line, the need for model order changes and determines the correct one.

There are at least two major sources for structure variations in the estimated model. The first is the model actually being time variant and the second resulting from the use of inherently nonlinear systems and mis-modeling. Two applications exemplifying these variants were selected to examine the techniques developed in the thesis. The first is a single flexible link manipulator, whose changes in model structure are due to new excited vibration modes. The second is a two link rigid manipulator with hydraulic actuators causing the system to be highly nonlinear, whose model could change due to changes in operating points. The effect of mis-modeling on the total system performance and stability was assessed.

A cost function was used as a measure of the closed loop controlled system reaction to under, correct and over-modeling. Its effectiveness in terms of stability and performance was measured in context of the two applications. In addition, experimental data from open loop identification of the dynamic model of a 215B Caterpillar, an excavator type machine, confirms the study of the behavior of the cost function for those conditions.

Based on the behavior of the cost function a new algorithm was developed. The MOD (Model Order Determination) algorithm detects, determines and executes, on-line, changes to the model order. It was implemented for both application which were controlled with the GPC algorithm. The results show that good performance and stability can be achieved.

The main contributions of this work are:

- The MOD algorithm which based on the behavior of a cost function, corrects on-line mis-modeling of adaptively controlled systems while maintaining good performance.

- GPC was successfuly implemented for hydraulically actuated manipulators. On-line automatic change of the GPC output horizon was introduced to achieve sufficiently fast transient response and avoid overshoots.

- Experimental data from a 215B Caterpillar manipulator proved the need for a closed loop approach.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisor, Professor Dale B. Cherchas for the support, guidance and encouragement he provided in the development of this work.

I would like to express my gratitude and appreciation to Professor Peter D. Lawrence, for fruitful work and valuable suggestions.

I am highly grateful to my colleague Doug Latornell, for the friendship, support and help I received during all the years I worked on this thesis. Also, I would like to thank Alan Steeves for his good advice, patience and support in the use of the department computer system.

All the staff in the department of Mechanical Engineering provided me with all the help I needed, and always with a smile.

The VAXStation 3200 computer system was provided through an equipment grant from the British Columbia Advanced Systems Institute (ASI).

Last but not least I would like to thank the Technion (Israel Institute of Technology), especially express my warmest gratitude to Professor Ram Lavie, from the faculty of Chemical Engineering, who made the last period of working on this thesis smooth and comfortable by granting me the status of a guest fellow at the Technion. A key figure to continuing my research work in Israel was provided, very professionally , by the computing center, by Miriam Ben-Haim and Ben Pashkoff - Thank You.

To Milush Roi and Reut

Chapter 1

INTRODUCTION AND STATEMENT OF OBJECTIVES

1.1 Introduction

The research in this thesis deals with adaptive control systems whose plant model parameters and structure may vary due to changing operating conditions. There are numerous examples of systems that need adaptive control algorithms, (Astrom et al [1]). One example is a robotic manipulator whose moment of inertia may vary within a working cycle. A flexible robot may have unexpected modes of vibration occurring and changing its model structure. Process control also has changes in dynamics, which depend on operating parameters, such as flow through tanks and pipes that change with production rate.

The structure of the plant's model is usually determined by its order and the nature of its nonlinear terms. The model generally used for the adaptive algorithms is linear and therefore its structure is actually its order. In most stability proofs for adaptive systems, the basic assumption is that the model order is known, or at least the upper bound of the system's order is known. In the presence of a change in the system's order, such as new significant modes in a flexible mechanical system, or in the presence of any unmodeled dynamics, instabilities can occur due to incorrect model structure and therefore incorrect parameters. The on-line changes in the plant's model structure if they occur, may result in the need to identify those changes accordingly, and adjust the order of the model for the adaptive algorithm in addition to the parameter identification.

The identification methods of a model off-line can have the advantage of choosing a model out of a set of proposed model structures. The model, its structure and parameters can later be verified with model validation methods. Many closed loop adaptive control algorithms use identification methods for on-line determination of the model parameters (Astrom et al [1]). We are not aware of any effective on-line structure validation methods. It is not surprising that most if not all identification methods use a fixed model structure to estimate the parameters. In some cases, the uncertainties in the values are ignored, i.e. the identified parameters are assumed to be correct, and are used as if they were the true ones. This is called the **certainty equivalence principle** (Astrom et al [1] and Middleton and Goodwin [2]). Adaptive systems have been said to be inherently nonlinear (Astrom et al [1]) and reliance on that principle can lead to instability.

1.2 Objectives and Motivation

One objective of the research is to investigate and study a chosen adaptive control algorithm with a changing model structure for the plant, and with it, the change in estimated model order and of the controller structure. A second objective is to develop a method to detect the need for a model order change, determine the correct order, execute it on-line and ensure that the control system will maintain its performance and stability. Changes in the order of the model along with the accompanying parameter estimation are to be integrated into a closed loop adaptive system. (All calculations and estimations are done on-line in real time).

The approach taken to the problem is listed below:

1. An approach to on-line order change detection and estimation was developed. An assumption considered is that order changes are less frequent and converge slower than the changes in the model parameters. Most existing techniques choose the order, estimate the model parameters, validate the order and if incorrect go through the whole procedure again, off-line and in open loop, (Ljung [3]).

2. Experimental evidence from a real machine, a 215B Caterpillar excavator was examined for open loop operation. The model output and representing cost function values were calculated and investigated. (The results corroborate the numerical results, see Appendix A). It was shown that a closed loop approach was needed.

3. The model determination algorithm was implemented with GPC for two examplifying applications. The Generalized Predictive Control (GPC) algorithm was chosen for the study, since it claims to effectively handle a number of problematic system characteristics at the same time (Clarke [4], [5]). There are at least two reasons for plant model structure variations. First, the machine model itself changes such as vibration modes in a flexible link. Second, use of a linear representation of a highly nonlinear machine model tracking an operating point, such as a two rigid link manipulator with hydraulic actuators. The structure flexibility of the single flexible link manipulator may give rise to modes of oscillations. During a work cycle of a robot, new modes can occur due to movement and change in the tip's load. This resembles an 'infinite order' system. (see Chapter 3). The two link rigid body manipulator with hydraulic actuators resembles a 'finite order' system. The hydraulic actuators can be modeled with several model structures and are highly nonlinear. Coupling between different motions of the arm's parts generates nonlinear terms. (see Chapter 4).

1.3 The Thesis Outline

Chapter 2 presents a review of previous work done in areas relevant to the research. Chapters 3, 4 and 5 present in detail the work done tn this research. Chapter 3 deals with a single flexible link manipulator, its dynamic equations of motion, the control strategy, results of closed loop simulation and the effects of under and over-modeling. Chapter 4 deals with the two link manipulator actuated by hydraulic actuators and reports on the same topics as Chapter 3. Chapter 5 introduces the chosen cost function as a measure of plant mis-modeling. The behavior of the cost function and its derivative is investigated. The conclusions drawn lead to the proposal of a method that detects on-line the need for a model order change, determines the correct order and executes it while developing or maintaining good system performance. Chapter 6 presents the implementation of the MOD algorithm on both applications and results are presented. Chapter 7 discusses the conclusions drawn from this research. Appendix A provides the experiment results from operating the heavy duty manipulator, the 215B Caterpillar. Appendix B presents the modal analysis of a cantilever beam, the results of which were used for the equations of the single flexible link in Chapter 3.

1.4 Thesis Contribution

This work is a study of robotic systems controlled with an adaptive algorithm (GPC). The main focus is on the behavior of those systems when the plant is mis-modeled and on restoring its desired performance and stability if needed.

The main research contributions are described here as:

1. We developed a method called the Model Order Determination (MOD) algorithm, to detect mis-modeling and implement its correction on-line.

   - A cost function was studied for correct, under and over-modeling, it was found it has significantly different behavior for each case.
   - The cost function behavior for correct, under and over-modeling was similar for two exemplifying applications.
   - When MOD was implemented, the desired performance was restored for mis-modeling for both applications.
   - Rules for choosing the parameters for the MOD algorithm were defined.

2. We have found that GPC can be successfuly implemented for heavy duty manipulators.

    - This work examined some complex considerations, such as the effects of nonlinearities in the application of GPC to a broad category of hydraulically actuated manipulators.
    - The work introduced on-line automatic change of the output horizon (for GPC) so transient response can be sufficiently fast and undesirable overshoot avoided.

3. Experimental results from an open loop experiment on a 215B Caterpillar indicate that the cost function behavior in open loop does not vary strongly to be reliably determined and a closed loop approach is required. This was also verified by numerical simulations with other applications.

Chapter 2

REVIEW OF PREVIOUS WORK

2.1 Outline

This chapter reviews some of the relevant work published in the literature considering some of the topics discussed in this thesis. Adaptive control in general and GPC in particular, are described. Some review on model order and parameter determination is also given.

Adaptive control is the basic motivation for the search for a method to change the model structure on-line. Section 2.2 describes adaptive control in general terms. A block diagram (Figure 2.1) is presented to clarify how all the components (identification, plant model, control algorithm, etc.) are implemented in the complete configuration.

In the last two decades, much research on modeling, system identification, model structure determination, and adaptive control algorithms has been done (Ljung [6], [3]). Most of the work concerning model structure determination and its validation is done *off-line*. The advantage of the off-line methods is in the possibility of choosing a model out of a set of likely ones. On the contrary, other works which involve *on-line* (recursive) identification demand an assumption of a fixed model structure, (Ljung [6], [3] and Astrom [1]). A brief review of the above modeling and structure determination will be given in Section 2.4.

In Section 2.3 we present the GPC (General Predictive Control) algorithm in detail. The GPC has been used in the work done so far, from which the results will be presented in Chapters 3 and 5.

Flexible structure models are dealt with in this work. The need for flexibility arises when

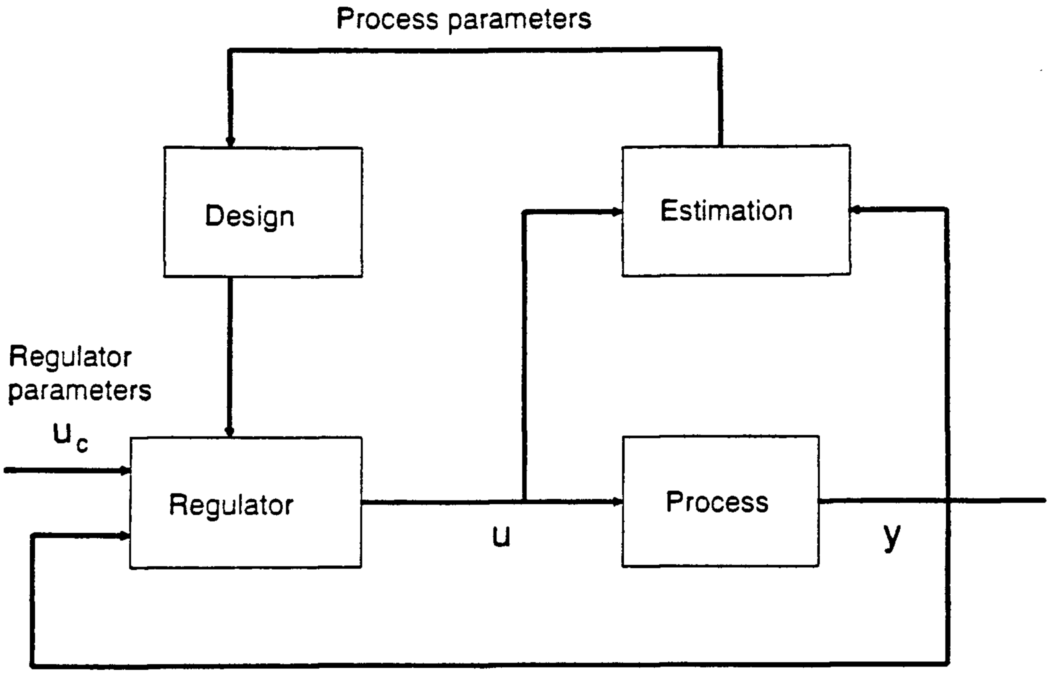

Figure 2.1: Block diagram for a self tuning regulator, (Astrom [1])

the certainty equivalence principle is used, or in the presence of unmodeled dynamics, or when changes occur in the structure. In order to maintain stability of the adaptive controlled system, the flexible model structure is particularly important.

Section 2.5 includes a discussion of the prior work done in the area of linear model's order determination in the 1970's and some later work on linear and nonlinear model structure determination.

2.2 Adaptive Control - General Description

2.2.1 Introduction

Much work has been done and published on adaptive control (Åström and Wittenmark [1], [7], Åström and Borrison and Ljung and Wittenmark [8]. Landau [9], Edgar [10]). Most of the techniques for the design of control systems assume that the plant and its environment are known. This is not often the case, since the plant might be too complex, or basic relationships may not be fully understood, or the process and the disturbances may change with operating conditions. Adaptive control deals with the above problems. There are four main categories of adaptive control:

1. Self Tuning Regulators - STR
2. Model Reference Adaptive Systems - MRAS
3. Auto-Tuning
4. Gain Scheduling

The STR and the MRAS are two widely discussed approaches to solving the problem for plants with unknown parameters. The proposed research concentrates on self tuning regulators.

2.2.2 Self Tuning Regulators - (STR)

STR are based on a fairly natural combination of identification and control. In Figure 2.1 a block diagram of the structure of an STR control loop is shown. It has two feedback loops, i.e. an inner loop and an outer loop. The inner one is an ordinary feedback loop with a process and a regulator. The regulator has adjustable parameters which are set by the outer loop. The adjustments are based on feedback from the process inputs and outputs. The outer loop is composed of a recursive parameter estimator and a design calculation. Estimations can be done on the process parameters or on the regulator parameters, depending on the control algorithm. The starting point is a design method for known plants. Since the parameters are not known, their estimates are used. The assumption is that there is a separation between identification and control, and the parameters' uncertainties are initially not considered here. As a simple example, consider the plant modeled by Equation 2.1:

$$y(t) + ay(t-1) = bu(t-1) + e(t) \tag{2.1}$$

Where u is the input, y is the output and e(t) is a sequence of independent, zero mean random variables. A control law that will give minimum variance control is :

$$u(t) = \frac{a}{b}y(t) \tag{2.2}$$

If a and b are unknown, the algorithm by Åström and Wittenmark [1] can be applied. It consists of two steps, each repeated every sampling period:

- Estimate the parameter α in the model:

$$y(t) = \alpha y(t-1) + \beta_0 u(t-1) + \epsilon(t) \tag{2.3}$$

  Where ϵ is the error. The resulting estimate is $\hat{\alpha}$.

- Use the control law:

$$u(t) = \frac{\hat{\alpha}}{\beta_0}y(t) \tag{2.4}$$

The estimation of α can be done recursively and on-line.

The above algorithm was also generalized in Åström & Wittenmark [1]. Åström & Wittenmark[1], [11] and Clarke & Gawthrop [12] proposed a generalization of the above basic algorithm. STR are not confined to minimum variance control. Edmunds [13], Åström et al. [1] proposed algorithms based on pole placement. Multivariable formulations were given by Borrisson [14].

2.2.3 Stability

Stability is a key requirement for a control system; however stability analysis of adaptive systems is difficult because the behavior of such systems is complex as a result of their nonlinear character. The stability problem can be approached in several different ways. A local stability technique is of limited value since it reveals little about global properties. The fundamental stability concept for nonlinear systems refers to the stability of a particular solution. One possibility is to apply Lyapunov's theory (Edgar [10] and Astrom [1]). However, it is often difficult to find a suitable Lyapunov function. Closed loop systems with bounded input/output signals and desired asymptotic properties can be achieved, provided that certain assumptions are made as in Åström et al. [1], Edgar[10], as noted below:

Given a plant model of the type:

$$A(q^{-1})y(t) = b_0 q^{-(d+1)} B(q^{-1})u(t) + e(t) \tag{2.5}$$

Where $A(q^{-1})$ & $B(q^{-1})$ are polynomials of degree n & m of the output y(t) and the input u(t) respectively, d is the time delay and e(t) is a disturbance that can not be measured, and q^{-1} is the backward shift operator. Also given are the following:

The time delay d is known. The upper bounds on the degrees of the polynomials A & B are known, i.e. the order of the system is known. The plant is a minimum phase process. The sign of b_0 is known.

Considering those assumptions, in Astrom and Wittenmark[15], it is shown that the closed loop system is stable if bounded disturbance and command signal (u_c) gives bounded input (u) and output (y).

2.3 Generalized Predictive Control - GPC

2.3.1 Introduction

Equations of motion of a robotic manipulator contain nonlinearities, inertial characteristics and disturbances that vary during a working cycle and may not always be predictable (Fu [16]). In many cases, performance obtained with fixed time invariant controllers may not be satisfactory. Lately, self tuning predictive algorithms have been used, since the results are more robust compared with other self tuning control algorithms, such as Pole Placement and Minimum Variance. The robustness of predictive algorithms is due to the minimization of a multi-step cost function, Clarke et al [4]. The basic predictive method contains the following steps:

1. Prediction of the output.
2. Choice of the future set points, and minimization of a cost function calculated from the future errors, between the future outputs and future set points, which yields a set of future control signals.
3. The first time step of the control signals is that actually used, and the whole procedure is repeated. This is a receding-horizon controller.

2.3.2 The GPC Algorithm

The type of controllers mentioned above consider the output at one point in time in the future. The Generalized Predictive Control (GPC), Clarke et al [4], [5], algorithm minimizes a cost function that considers the future predicted outputs j steps ahead, the future set points and future control signals. The GPC is robust and deals with overparametrization because of its predictive capabilities, and with dead time since it uses an explicit plant model. The robot manipulator can be nonminimum phase and incorrectly parametrized (especially when there is some flexibility in the links), can have dead time in the hydraulic system, and if sampled fast, can have instabilities. Many discussions about adaptive control and GPC can be found in the literature, such as Åström [11],Tomizuka [17], Demircioglu [18], Latornell [19], etc. .

The GPC uses a plant model which is a CARIMA type (Controlled Auto Regressive Integrating Moving Average): i.e.

$$\begin{aligned} A(q^{-1})y(t) &= B(q^{-1})u(t-1) + e(t) \\ e(t) &= c(q^{-1})\frac{\xi(t)}{\Delta} \\ \Delta &= 1 - q^{-1} \end{aligned} \tag{2.6}$$

Where $y(t)$ is the measured output, $u(t)$ is the control input, $e(t)$ is the unmeasured disturbance term, ξ is uncorrelated random sequence, q^{-1} is the backward shift operator, Δ is the differencing operator and $A(q^{-1})$, B(q^{-1}) and C(q^{-1}) are polynomials of degrees n_a, n_b and n_c respectively.

The algorithm minimizes a cost function of the form (Clarke et al [4], [5]):

$$J(N_1, N_2, N_u) = E\left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)\,[\Delta u(t+j-1)]^2 \right\} \tag{2.7}$$

Where N_1 is the minimum output horizon, N_2 is the maximum output horizon, N_u is the control horizon $\lambda(j)$ is a control weighting sequence, $y(t+j)$ is the output j steps ahead and $w(t+j)$ is the future set point.

The GPC algorithm predicts future outputs and aims at good performance a few steps ahead by minimizing the above cost function, which gives a sequence of future control signals and avoids large input signals with saturation.

To derive a j step ahead predictor of $y(t)$, Equation 2.6 should be multiplied by $q^{-j}E_j(q^{-1})\Delta$ and the following identity used:

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \tag{2.8}$$

This is the Diophantine equation, where E_j and F_j are polynomials in the backward shift operator. In order to get a unique solution, the degree of the following polynomials is chosen as:

$$\deg(E_j(q^{-1})) \leq j - 1$$

and

$$\deg(F_j(q^{-1})) \leq \deg(A(q^{-1}))$$

So the j step ahead output y(t+j) is:

$$y(t+j) = F_j(q^{-1})y(t) + E_j(q^{-1})B(q^{-1})\Delta u(t+j-1) + E_j(q^{-1})\xi(t+j) \tag{2.9}$$

$C(q^{-1})$ is chosen to be 1.

The disturbance sequence consists only of future values which are unknown, so the optimal predictor is:

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-1) \tag{2.10}$$
$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$$

The objective of the predictive control law is to derive future plant outputs $y(t+j)$, given a future set point sequence $w(t+j)$. It is done as follows:

1. The future set point sequence $w(t+j)$ is determined.

2. A set of predicted errors is calculated:

$$\epsilon(t+j) = \hat{y}(t+j) - w(t+j)$$

3. The cost function in Equation 2.7 is minimized to provide a suggested sequence of future control increments, assuming that after some control horizon N_u, future increments in the control are zero, and the control signal is kept constant. The weighting factor was initially selected as recommended by Clarke et al [4], [5], followed by adjusting the values according to the controlled system.

The optimal prediction of $y(t+j)$ can be written as:

$$\hat{\mathbf{y}} = \mathbf{G}\ \mathbf{u} + \mathbf{f} \tag{2.11}$$

Where **f** includes the components of the predicted output $\hat{\mathbf{y}}$, which are known at time t, and **G** is a lower triangular matrix of dimension $N_2 \times N_2$.

Minimization of equation 2.7 yields the control increment vector:

$$(\mathbf{u}) = (\mathbf{G}^{\mathbf{T}}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^{\mathbf{T}}(\mathbf{w} - \mathbf{f}) \tag{2.12}$$

The first element of **u** is $\mathbf{\Delta u(t)}$ so that the current control $u(t)$ is given by:

$$\mathbf{u(t)} = \mathbf{u(t-1)} + \bar{\mathbf{g}}^{\mathbf{T}}(\mathbf{w} - \mathbf{f}) \tag{2.13}$$

Where $\bar{\mathbf{g}}^{\mathbf{T}}$ is the first row of $(\mathbf{G}^{\mathbf{T}}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^{\mathbf{T}}$.

The design parameters for this algorithm are:

1. **Minimum Output Horizon,** N_1**:** is set to the time delay K, if known, to save computation load since the control signals have no effect earlier. The time delay, K, in discrete

systems corresponds to the degree of the polynomial B, so when K is not known, N_1 is given the minimal value for the time delay, $N_1 = 1$.

2. **Maximum Output Horizon, N_2:** the authors recommend to set it approximately at the value of the rise time of the system. Both parameters N_1 and N_2 are used in Equation 2.11 in which the number of prediction steps j vary from N_1 to N_2 and are then used in calculating the control law in Equation 2.13.

3. **The Control Horizon,** N_u: A major advantage of the GPC is in the assumption about future control signals. After an interval of $N_u < N_2$, the projected control increments are assumed to be zero. This reduces the computation burden, since $\dim(\mathbf{u}) = N_u$ and $(\mathbf{G}^\mathbf{T}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^\mathbf{T}$ is an $N_u \times N_u$ matrix. Usually, the control horizon can be chosen as $N_u = 1$ (for stable plants with delay or with nonminimum phase), but when poorly damped or unstable poles are present, N_u should equal the number of those poles.

4. **The Control Weighting Sequence, $\lambda(j)$:** Acts as damping of the control action when greater than zero.

Final goal of model application	*Type of process model*	*Required accuracy of model*	*Identification method*
verification of theoretical models	linear/continuous time nonparametric/ parameters	medium/ high	off-line step response, frequency response, parameter estimation
controller parameter tuning	linear nonparametric, continuous time	low for input/output behavior	off-line step response
computer aided design of digital control	linear parametric (nonparametric) discrete time	medium for input/output behavior	on-line, off-line parameter estimation
self-adaptive digital control tuning	linear parametric discrete time	medium for input/output behavior	on-line parameter estimation in close loop
process parameter monitoring and failure detection	linear/nonlinear parametric continuous time	high for parametric continuous time	on-line parameter process parameters estimation

Table 2.1: Different final goals and specifications for identification cases.

2.4 Introduction to Model Structure and Parameter Determination

System identification is basically a function of building a mathematical model by analyzing the relations between observed input and output. A considerable amount of work has been done and can be found in references Ljung and Söderstöm [6], Ljung [3] and Isermann [20]. It is important to first consider the final goal for the application of the process model, since this determines the type of model, its accuracy requirements, and the identification method.

Table 2.1 (Isermann [20]) shows some examples for relationships between different final goals and some specifications of process identification. The a priori knowledge of the process is, for example, based on general process understanding, on principal laws and on premeasurements.

The choice of a model structure is essential for successfully identifying a system. Ljung [3] in his book describes techniques and procedures about model structure selection and model validation. The choice of a model structure should be based on good understanding of the identification method and a priori knowledge of the system. There are three basic steps in choosing a model structure i.e.

1. Choice of type of the model set: linear or nonlinear model, input-output or black box models, etc. .

2. Choice of the model size: model order for linear systems.

3. Choice of model parametrization: identification of model parameters.

The quality of the model is usually measured by minimizing a criterion. There is a tradeoff between flexibility and parsimony. Flexibility will give, for a larger number of parameters, a better fit for the minimization of the criterion since the data set is larger. On the other hand it is important, in practice, to employ the smallest number of parameters for adequate representation of physical systems.

The type of the chosen model is usually based on a priori knowledge about the system and intuition. Generally it is advisable to start with the simplest possible model.

Order estimation of a linear system is usually based on preliminary data analysis. The relevant methods fall into the next categories:

- Spectral analysis of the transfer function.

- Testing ranks of sampled covariance matrix.
- Correlating variables.
- Examining the information matrix.

Model validation methods can be as follows:

- Comparing linear models with a priori knowledge of the system.
- Comparing measured and simulated outputs.
- Testing residuals for independence of past inputs.
- Comparing criteria fit obtained in different model structures.

Off-line methods are capable of choosing a model structure and identifying its parameters. There are many cases where on-line identification of the model is needed. There are two disadvantages to recursive (on-line) identification, in contrast to off-line identification. The first is that the decision on which model structure to use has to be made and fixed a priori, before starting the recursive identification procedure. It is well known that parametric models can give large errors when the order of the model does not agree with the order of the process (see Åström [21]). In the off-line situation, different types of models can be examined. The second disadvantage is that recursive methods generally are not as accurate as off-line ones (Ljung [3]).

For off-line identification, the basic three steps are:

- *data recording* - good choice of the data record makes it maximally informative.
- *a set of candidate models* - the most important and difficult choice of the system identification. This is the actual determination of the model structure.

- *determining the best model in the set* - guided by the data (this is the identification method).

After having settled on the preceding three choices, one must validate the model, i.e. establish a criterion to examine if the model accepted is good enough. If it is not, the entire procedure is repeated again as shown in Figure 2.2 .

After determining the system model‘s structure, the user may choose from the identification and adaptive control techniques that are available. Considerable work in these areas has been done and will not be discussed at this point, but some important references are Ljung and Söderstöm [6], Ljung [3], Åström and Eykhoff [21], Åström [22], Åström and Wittenmark [1], Strejc [23], Goodwin and Payne [24].

2.5 Review of Previous Work in Order Determination

To emphasize the effect of incorrect modeling, we refer to Rohrs et al [25] which analyzed the effect of unmodeled dynamics on the robustness and stability of continuous time adaptive control algorithms. The conclusion in this paper is that adaptive algorithms published in the literature are likely to produce unstable control systems if they are implemented on physical systems directly as they appear in the literature. Unfortunately, stability proofs of all those algorithms have in common a very restrictive assumption that the order of the system is known. So if there are errors in the structure assumptions, instabilities can occur. It is also noted that this problem can be partially alleviated by sufficient excitation, but the amount of modeling error or the amount of disturbance for which the adaptive system can maintain stability may be extremely small.

In the 1970's, several works were published, such as Akaike [26] and Akaike [27], Isermann [20], Schwarz [28], primarily establishing estimates of a measure of fit of the model. Those procedures are off-line ones and are part of the model validation techniques.

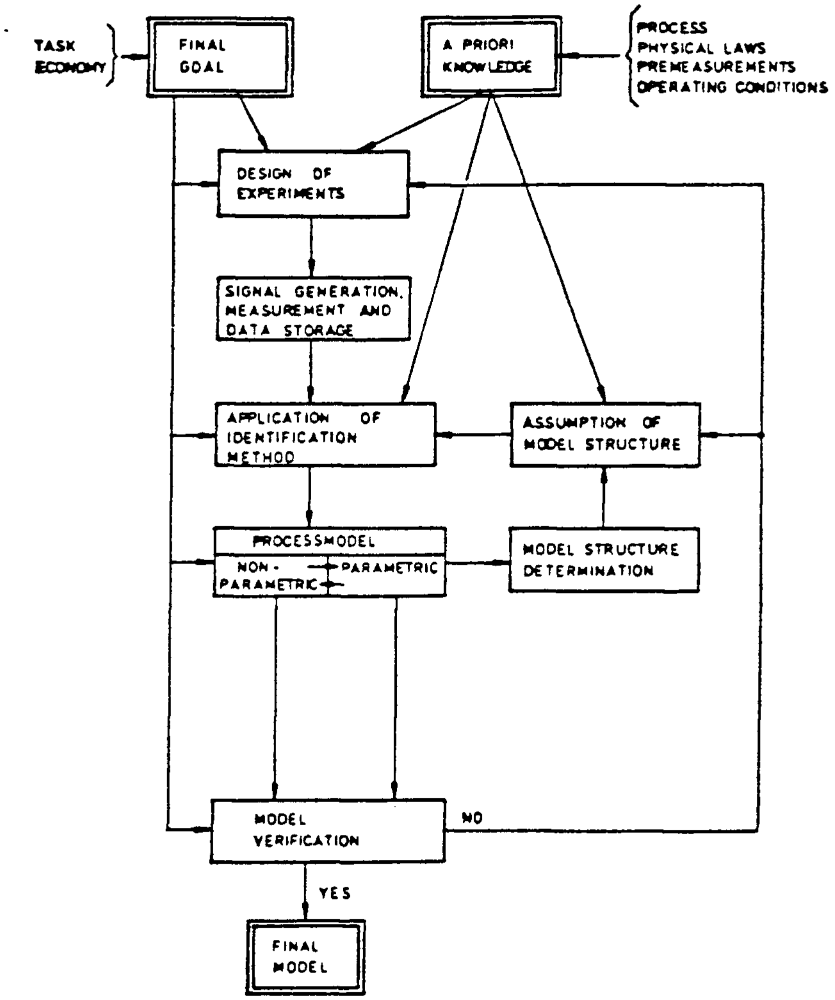

Figure 2.2: General procedure of process identification, (Isermann [20])

As mentioned, several criteria for order estimation were published. Rissanen [29] proposes the following cost function:

$$U(x,n,m,\xi) = N\ln\hat{r} + \sum_{i=1}^{n+m} \ln\left[\hat{\xi}_i^2 \frac{\partial^2 \ln\hat{r}}{\partial\hat{\xi}_i^2}\right] + (n+m+1)\ln(N+2) + 2\ln(n+1)(m+1) \quad (2.14)$$

Where

x is the model:

$x(t) = f_\alpha[x(t-1),...,x(t-n),e(t),...,e(t-m)]$

N - number of sampled observations.

e - disturbance.

ξ - consist of real valued parameters.

$N\log\hat{r}$ - the minimized log likelihood function.

The first and third terms together virtually coincide with a criterion derived by Schwartz [28]. The most commonly used criterion, derived by Akaike ([26], [6]), is the AIC (Akaike's Information Theory Criterion) which is:

$$\text{AIC} = N\log\hat{r} + 2k \quad (2.15)$$

Where

$N\log\hat{r}$ - the minimized log-likelihood function.

k - the number of parameters in the model.

More criteria were introduced by Akaike in 1977 [30] and by Hannan and Quinn in 1979 [31].In 1989, Gou [32] introduced a new criterion for order estimating of a CARAMA model. The final conclusion is that estimates presented are non-recursive and require availability of upper bounds for unknown orders; therefore further research is required.

Approaches for different structures have been explored such as canonical structures (Guidovzi [33] and [34]). These try to close the gap by exploring a class of state space canonical models with particularly simple relations to input/output difference descriptions that can be directly identified from input/output sequences. However, in order to avoid error, the previous estimate of the order and structure of the process, out of a class of models, is required.

Further off-line methods, including MIMO (Multi Input Multi Output) systems, have been published and will not be discussed here. Most of the recursive identification methods for MIMO also assume a model structure a priori. Such a method by Gauthier [35] extends the use of input/output description in terms of polynomial matrices for recursive identification in canonical state space form. However, before identifying the polynomials' coefficients, one must define the structure and the polynomials' degree or at least, their upper bounds.

Ljung and Sodestrom [6] discuss in their book the concept of identifying overparametrization of a model set and the choice of a model order. The choice of a model order is a delicate trade off between good description of the data and the model complexity. Most methods for model order selection are developed for off-line techniques. The basic approach is to compare performance of models with different orders and test whether a higher order model is worth while. Recursive algorithms in on-line applications require identification of several models simultaneously. A model set is said to be identifiable if its parameters can be identified i.e. parameters can be uniquely determined from the data. Lack of identifiability can be caused by non-exciting inputs and overparametrization.

MIMO systems, can be parametrized depending on the choice of structure of the system. The problem has been avoided by some researchers, assuming that the designer has enough a priori knowledge of the structure of the system to select stable parametrization. Overbeek and Ljung [36] suggested a procedure that provides a means of obtaining the best model structure. The model Structure Selection (MSS) algorithm, the structure dealt with in that work, is the parametrization of systems and can be performed in a number of ways. The technique does not deal with how to select an appropriate order. The algorithm receives as input a given system with a given parametrization. It tests whether this parametrization is well conditioned for identification purposes. If it is not well conditioned, another structure is considered. The best structure of a possible set is decided upon a priori. Nagy and

Ljung [37] describe in their paper the subject of computer- aided model structure selection. In order to use a software package for system identification, an appropriate model structure should be chosen. A common feature for these methods is that they mix extensive numerical computations, code generation and symbolic algebra.

Davison [38] in his paper describes a method of model size reduction. Many physical plants can be represented by simultaneous linear differential equations with constant coefficients, of the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

Where the order of matrix $\mathbf{A}$ can be large, for example chemical plants or nuclear reactors, which can pose numerical problems, the method suggests the reduction of the rank of such matrices by constructing a matrix of lower order with the same dominant eigenvalues and eigenvectors as the original system.

The paper by Niu, Xiao and Fisher [39] presents a simultaneous recursive estimation of the model parameters and loss functions for all possible model orders from zero through n is done by using augmented information matrix (AIM) and a UDU^T factorization algorithm.

The AIM matrix is:

$$C_n(k) = S_n^{-1}(k) = \left[\sum_{j=1}^{k} \Phi_n(j)\Phi_n^T(j)\right]^{-1} \tag{2.16}$$

Where Φ_n is the regression vector and C_n is the AIM matrix.

$$\phi_n = [-y(t-n), v(t-n), u(t-n) \cdots - y(t-1), v(t-1), u(t-1), -y(t)]$$

Where y is the output, v is the noise and u is the input.

The algorithm is reported to be computationally efficient and have good numerical properties due to the use of the UDU^T algorithm. In a second paper Niu and Fisher [40] report on a

MIMO system identification technique using augmented UD factorization. This work extends the AUD algorithm from SISO systems to MIMO systems. It is based on the canonical state space representation be Guidorzi [33] [34] the Bierman's UD factorization [41]. This algorithm too, is reported to poses excellent numerical properties.

The work reported next, (Wulich and Kaufman [42]) is a trial and error model order estimation procedure. The estimation is based on sampling a signal and calculating its autocorrelation function:

$$R(n) = \frac{1}{L}\sum_{j=0}^{L-k-1} x(j)x(j+k) \qquad (2.17)$$
$$k = 1, 2, \cdots, L_k \ll L$$

Where $x(t-j)$ are sampled values of the signal $x(t)$ and L is the number of samples. A system of n linear equations is generated:

$$\mathbf{R}_n A = 0$$

Where A is a nonzero vector of coefficients and $\mathbf{R}_n$ is a matrix. The order is estimated by examining the determinant of the matrix $\mathbf{R}_n$:

If $\det \mathbf{R}_n = 0$ then $n > N$

and

If $\det \mathbf{R}_n \neq 0$ then $n \leq N$

Where $n = 1, 2, 3, \cdots$.

and N is the correct order.

A method for simultaneously selecting the order and identifying the parameter of an Autoregression (AR) model, has been developed, (Katsikas et al [43]). The AR model is defined as:

$$y(k) = \sum_{i=1}^{N} a_i(k)y(k-i) + v(k)$$

Where a_i are the coefficients, $y(k-i))$ are previous outputs and $v(k)$ is zero mean white noise. The order of the system N in unknown but it is in the range of $1 \leq N \leq M$, where M is known. The true model will be one of a family of models with the above range of order.

A paper by (Birch, Lawrence et al [44]) deals with fitting and estimating a model to EEG (Electroencephalography) signals. The EEG signal is modeled with an AR model type and a spectral estimation procedure is performed. The selection of the proper model order is done by some a priori knowledge of expected results.

Very little work has been published on structure determination for nonlinear systems. One which treats mathematical models for representation of the dynamic of ship rudder-yaw and roll motions is in Zhou et al [45]. The work checks suggested models given in the literature for the problem with the Recursive Prediction Error (RPE) identification methods. It is an off-line method that finds the best nonlinear terms for the model. A non linear on-line method has been described by Zervos and Dumont [46]. The plant is modeled by an orthogonal Laguerre network put into state space form. The number of the Laguerre filters used depends of the presence of time delay and undamped modes. The actual plant order does not influence the number of Laguerre filters (N) used. Usually the choice is $5 < N < 10$, and N can be changed on line.

Hemerly [47] presents a method for order and parameter identification of industrial processes. The processes to be identified are described by ARX mode:

$$y(t) = -a_1 y(t-1) - \cdots - a_n y(t-n) + b_1 u(t-1) + \cdots + b_n u(t-n) + w(t)$$

Where $y(t)$ is the systems output, $u(t)$ is the input, a_i and b_i are the coefficients and $w(t)$ is white noise. The parameters are identified by Recursive Least Squares (RLS) algorithm and for the order estimation the Predictive Least Squares (PLS) criterion is used, (Rissanen [48]).

Where:

$$PLS(n.t) = \frac{1}{t}\sum_{i=0}^{t-1} e^2(n, i+1)$$

The best estimate of the order should be

$$\hat{n}(t) = \arg\min PLS(n, t)$$

with

$$e(n, t+1) = y(t+1) - \hat{\Theta}^T(n, t)\Phi(n, t)$$

where

$$\hat{\Theta} = [-a_1, \cdots, -a_N, b1, \cdots, b_N]^T$$

$$\Phi(n, j) = [y(j), \cdots, y(j-n+1), u(j), \cdots, u(j-n+1)]^T$$

The PLS criterion is highly intuitive and at time t the order estimate $\hat{n}(t)$ is the order of the model which has given the least mean square prediction error up to that time. The process can be identified for different operating points by varying the excitation amplitude and therefore getting several linear models. A controller can be designed for each model and changed in real time if necessary. Medeiros and Hemerly [49] integrated lattice form for constructing a minimum variance adaptive controller with parameter and order estimation. As described in the former paper the order is estimated with the PLS criterion. The lattice filter is a way to parametrize as following:

$$\hat{\Theta}_{n+1}(t) = \hat{\Theta}_n(t) + \hat{\rho}\hat{r}_n(t-1)$$

$$\hat{r}_{n+1}(t) = \hat{r}_n(t-1) + \hat{\rho}\hat{\Theta}_n(t)$$

$$\hat{e}_0(t) = \hat{r}_0(t) = y(t)$$

where:

$$\hat{e}_n(t) = y(t) - \hat{y}_n(t/\hat{\Theta}^n)$$

$$\hat{r}_n(t-1) = y(t-n-1) + \hat{a}_1^n y(t-n) + \cdots + \hat{a}_n^n y(t-1)$$

Where $\hat{\rho}_n$ are coefficients, $\hat{\Theta}_n$ estimated parameters for order n, $y(t)$ is the measured output, $\hat{y}_n$ is the estimated output for model order n, and $\hat{e}_n$ is the error. A set of prediction errors is calculated and the estimated order of the model is the one that has a minimal least squares error.

As can be inferred from the previous discussions, much work has been published on identification and adaptive control algorithms, mostly for fixed order and structure models. Very little research work has been done for flexible model structures particularly for on-line methods and there is definitely a need for research in the area. This work constitutes a contribution to the research for on-line model structure determination.

Chapter 3

SINGLE FLEXIBLE LINK MANIPULATOR

3.1 Outline:

In this Chapter, we present the work done with the flexible link manipulator. The mathematical model and equations of motion have been developed. Numerical results for the control of the manipulator are presented, as well as the effects of mis-modeling on the magnitude of a cost function and an outline for an iterative method for the order estimator.

Section 3.2 develops all mathematical modeling involved in the numerical simulation of a single flexible link manipulator, including the linear equations of motion and control strategy. Section 3.3 presents the numerical results and analysis for the flexible link application.

3.2 Single Flexible Link Manipulator

3.2.1 Introduction

A robot is a complex system to control not only because it is a nonlinear system and has variations in the moment of inertia, but also because flexible link structure or nonlinearities, such as hysteresis or backlash. There are two approaches for the design of controllers for such systems: i.e. to design one that will not excite the poorly damped modes, or one that actively damps oscillatory modes. The second option is not used in industrial robots. Such control systems are complicated, since the frequencies of the oscillatory modes vary with orientation and load. The variations in the oscillatory modes are the reason for choosing the flexible link as an application. New vibration modes that arise mean that the structure of the model has changed on line, so the model and the control system should be updated. Data for the chosen flexible link can be found in [50] and [19]. The arm is a 1 meter long, flexible mechanical structure which can bend freely in the horizontal plane but is stiff in vertical bending and in torsion. Its motion is only in the horizontal plane i.e. gravity effects are not important.

3.2.2 Equations of Motion for the Single Flexible Link

The flexible arm is comparable to a cantilever beam. Figure 3.1 describes the flexible link configuration.

Where:

x_0 - is the reference axis.

x - is the position of a rigid arm at θ [rad.] from x_0.

$w(x,t)$ - is the deflection from the rigid body.

I_B - is the moment of inertia about the hub $[kg \cdot m^2]$.

I_H - is the motor's moment of inertia $[kg \cdot m^2]$.

τ_H - is the torque applied by the motor $[N \cdot m]$.

E - is Young's module $[N/m^2]$.

I - is the beam cross sectional moment of inertia $[m^4]$.

The displacement of any point P along the beam at a distance x from the hub is given by $\theta(t)$ and the deflection $w(x,t)$, measured from the line Ox which would be the arm, had it been rigid. The assumptions made are:

- the deflection is small - $w(L,t) \ll L$
- shear deformation and rotary inertia effects are neglected.
- gravitation effects for deflection and movement in a horizontal plane are neglected.

The displacement $y(x,t)$ of a point p along the arm is defined as:

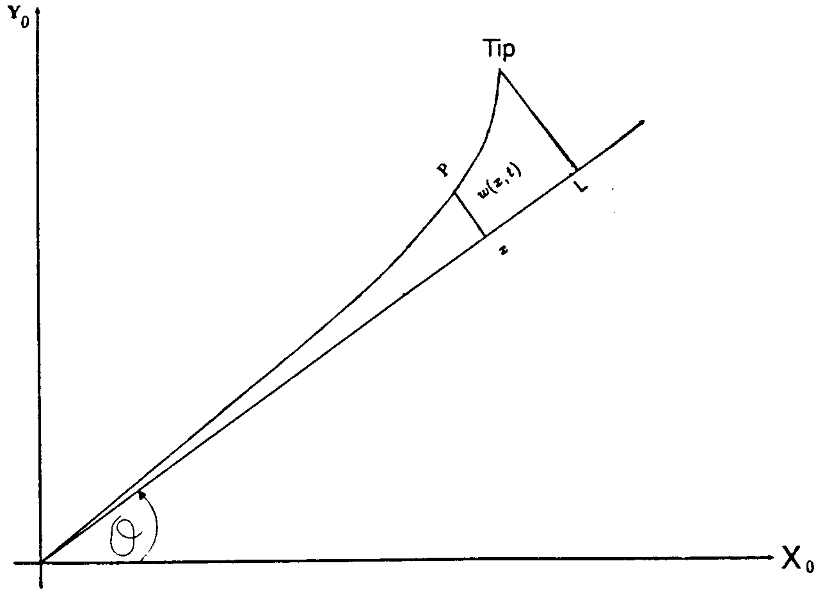

Figure 3.1: Configuration of the single link flexible arm

$$y(x,t) = w(x,t) + x\theta(t) \tag{3.1}$$

Let:

$$w(x,t) = \sum_{i=1}^{n} \phi_i(x) q_i(t) \tag{3.2}$$

Where $\phi_i(x)$ is the i^{th} mode shape and $q_i(t)$ is the i^{th} mode generalized coordinate. $n = 1, 2, 3, 4, \ldots$ is the number of vibration modes. (Note that when $i = 0$, $\phi_0(x) = x$, $\frac{d\phi_0(x)}{dx} = 1$, $q_0(t) = \theta$ and $\dot{q}_0(t) = \dot{\theta}$ are the parameters for a rigid body.)

All derivatives as $\dot{\theta}$ are with respect to time, all derivative as θ' are with respect to x.

The kinetic energy of the system is:

$$T_K = \frac{1}{2} I_H \dot{\theta}^2 + \frac{1}{2} m \int_0^L (\frac{dy(x,t)}{dt})^2 dx$$

$$2T_K = I_H \dot{\theta}^2 + m \int_0^L (\frac{dw(x,t)}{dt} + x\dot{\theta})^2 dx$$

Using Equation 3.2 the kinetic energy is:

$$2T_k = I_T \dot{\theta}^2 + \sum_{i=1}^{n} I_{2i}\, \dot{q}_i^2(t) + 2\, \dot{\theta} \sum_{i=1}^{n} I_{1i}\, \dot{q}_i(t) \tag{3.3}$$

The inertia integrals I_{1i}, I_{2i}, I_{3i} are described in Appendix B and their values are presented in Table B.1. (The number indicates if it is I_1, I_2 or I_3 as indicated in Appendix B, and the i shows for which mode the integral is calculated).

The potential energy is strain energy due to bending deformation is:

$$2\,P = \int_0^L EI\,(\frac{d^2w}{dx^2})^2\,dx$$

$$2\,P = \sum_{i=1}^{n} \sum_{j=1}^{n} \int_0^L EI\,\phi_i''\,\phi_j''\,q_i\,q_j\,dx$$

Using orthogonality relations:

$$EI\,\phi_i''\,\phi_j''\,dx = \begin{cases} \omega^2 \int \phi_i^2\,\rho\,dx = I_{2i}\,\omega^2 & i = j \\ 0 & i \neq j \end{cases} \tag{3.4}$$

the potential energy becomes:

$$2P = \sum_{i=1}^{n} I_{3i}\,q_i^2 \tag{3.5}$$

Introducing a dissipation function D that may be defined as:

$$D = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\,\dot{q}_i \dot{q}_j$$

the damping force will be:

$$-\frac{dD}{d\dot{q}_j} = -\sum_{i=1}^{n} c_{ij}\,\dot{q}_i \tag{3.6}$$

Combining all together and applying Lagrange's equation,

the **equations of motion** are:

for $i = 0$ - rigid body:

$$I_T\,\ddot{\theta} + \sum_{i=1}^{n} I_{1i}\,\ddot{q}_i(t) = \tau_H - c_0\,\dot{\theta} \tag{3.7}$$

for $i = 1, 2, 3, \dots$:

$$I_{1i}\,\ddot{\theta} + I_{2i}\,\ddot{q}_i(t) = -I_{3i}\,q_i(t) - c_0\,\dot{q}_i(t) \tag{3.8}$$

Where: $I_T = I_B + I_H$ is the total moment of inertia, and c_i where $i = 1, 2, 3, \dots, n$ is the damping coefficient.

3.2.3 The State Space Model

Equations 3.7 and 3.8 can be put into a matrix form, such as:

$$\mathbf{M}\dot{\mathbf{x}} = \mathbf{K}\mathbf{x} + \mathbf{b}\tau_H$$

Where x is the state space vector defined as:

$$\mathbf{x} = \begin{bmatrix}\theta, & q_1, & q_2, & \cdots & q_n, & \dot{\theta}, & \dot{q}_1, & \dot{q}_2, & \cdots & \dot{q}_n\end{bmatrix}^T \tag{3.9}$$

M, **K** and **b** are matrices defined as:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \cdots & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & \cdots & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & I_T & I_{11} & I_{12} & I_{13} & \cdots & I_{1n} \\ 0 & 0 & 0 & \cdots & \cdots & 0 & I_{11} & I_{21} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & I_{12} & 0 & I_{22} & \cdots & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & \cdots & 0 & I_{1n} & 0 & 0 & \cdots & \cdots & I_{2n} \end{pmatrix} \tag{3.10}$$

$$\mathbf{K} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 1 & \cdots & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & \cdots & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & -c_0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -I_{31} & 0 & \cdots & \cdots & 0 & 0 & -c_1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & -c_2 & \cdots & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & \cdots & -I_{3n} & 0 & 0 & 0 & \cdots & \cdots & -c_n \end{pmatrix} \tag{3.11}$$

$$\mathbf{b} = [0\ 0\ 0\ \cdots\ 1\ 0\ 0\ 0\ \cdots\ ]^T \tag{3.12}$$

The state space model in its final form is:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \tau_h \tag{3.13}$$

$$y_{tip} = \mathbf{C} \cdot \mathbf{x}$$

Where:

$$\mathbf{A} = \mathbf{M}^{-1} \cdot \mathbf{K}$$

$$\mathbf{B} = \mathbf{M}^{-1} \cdot \mathbf{b}$$

$$\mathbf{C} = [1\ 1\ 1\ \cdots\ 0\ 0\ 0\ \cdots\ ] \tag{3.14}$$

The order of the system depends on the number of modes that are included. For example, a 3 mode model will be an 8^{th} order model. Table 3.1 shows the order of the system versus the number of modes considered.

3.2.4 The Discrete Time Model

The system was converted with a zero order hold sampling from a continuous state space form into a discrete time form.

For sampling with period h, the time is:

$$t_k = k \cdot h$$

The state space discrete time model has the following structure:

$$\begin{aligned} x((k+1)h) &= \mathbf{\Phi}\, x(k\,h) + \mathbf{\Gamma}\, u(k\,h) \\ y(k\,h) &= \mathbf{C}\, x(k\,h) \end{aligned} \tag{3.15}$$

Where:

$$\begin{aligned} \mathbf{\Phi} &= \exp^{\mathbf{A}\,h} \\ \mathbf{\Gamma} &= \int_0^h \exp^{\mathbf{A}\,S}\, dS\, \mathbf{B} \end{aligned} \tag{3.16}$$

In order to simplify calculations, we expanded it into a series, i.e.:

$$\mathbf{\Psi} = \int_0^h \exp^{\mathbf{A}\,S} dS = Ih + \frac{\mathbf{A}\,h^2}{2!} + \frac{\mathbf{A}^2\,h^3}{3!} + \cdots + \frac{\mathbf{A}^i\,h^{i+1}}{(i+1)!} + \cdots \tag{3.17}$$

Now the matrices are given by:

$$\begin{aligned} \mathbf{\Phi} &= I + \mathbf{A}\,\mathbf{\Psi} \\ \mathbf{\Gamma} &= \mathbf{\Psi}\,\mathbf{B} \end{aligned} \tag{3.18}$$

3.2.5 Open Loop Discrete Time Models for Different Number of Modes

The discrete time models for y_{tip} ver. τ_h were developed for cases with a different number of modes. The structure of the discrete model is in the form of Equation 2.6, where the noise sequence is equal to zero. The sampling period is $h = 0.01$ [sec.]. The damping factors are, $c_i = 0.05$.

Remark: All the following discrete time models for the different orders were checked with continuous time simulation for the flexible link with the same number of vibration modes (i.e. the same model order). The results show the same behavior for the continuous time and the discrete time models.

For rigid body - $i = 0$, model order is 2 :

$$\begin{aligned}\mathbf{y_{tip}(t_k)_{/0}} = \quad & +1.9982\, y_{tip}(t_k - 1) - 0.9982\, y_{tip}(t_k - 2) \\ & +0.17774 \cdot 10^{-3}\, \tau_H(t_k - 1) + 0.1773 \cdot \tau_H(t_k - 2)\end{aligned} \tag{3.19}$$

For one vibration mode $i = 1$, model order is 4 :

$$\begin{aligned}\mathbf{y_{tip}(t_k)_{/1}} = \quad & +3.1423\, y_{tip}(t_k - 1) - 4.1531\, y_{tip}(t_k - 2) \\ & +2.8781\, y_{tip}(t_k - 3) - 0.8673\, y_{tip}(t_k - 4) \\ & +0.1458 \cdot 10^{-3} \tau_H(t_k - 1) + 0.0173 \cdot 10^{-3} \tau_H(t_k - 2) \\ & -0.0405 \cdot 10^{-3} \tau_H(t_k - 3) + 0.1344 \cdot 10^{-3} \tau_H(t_k - 4)\end{aligned} \tag{3.20}$$

For two vibration modes $i = 2$, model order is 6 :

$$\begin{aligned}\mathbf{y_{tip}(t_k)_{/2}} = \quad & +1.5506\, y_{tip}(t_k - 1) - 0.4874\, y_{tip}(t_k - 2) \\ & +0.5763\, y_{tip}(t_k - 3) - 1.2837\, y_{tip}(t_k - 4) \\ & +1.0311\, y_{tip}(t_k - 5) - 0.3870\, y_{tip}(t_k - 6)\end{aligned} \tag{3.21}$$

$$\begin{aligned}
&+0.0534\,10^{-3}\tau_H(t_k-1)+0.4899\,10^{-3}\tau_H(t_k-2)\\
&+0.1043\,10^{-3}\tau_H(t_k-3)-0.1471\,10^{-3}\tau_H(t_k-4)\\
&+0.1764\,10^{-3}\tau_H(t_k-5)+0.0187\,10^{-3}\tau_H(t_k-6)
\end{aligned}$$

For three vibration modes $i=3$, model order is 8 :

$$\begin{aligned}
\mathbf{y_{tip}(t_k)_{/3}} = \quad &+0.0662\,y_{tip}(t_k-1)+2.0013\,y_{tip}(t_k-2)\\
&+0.6734\,y_{tip}(t_k-3)-1.7043\,y_{tip}(t_k-4)\\
&-1.1544\,y_{tip}(t_k-5)+0.9074\,y_{tip}(t_k-6)\\
&+0.6223\,y_{tip}(t_k-7)-0.4120\,y_{tip}(t_k-8)\\
&+0.0576\cdot10^{-3}\tau_H(t_k-1)+0.5583\cdot10^{-3}\tau_H(t_k-2)\\
&+0.8196\cdot10^{-3}\tau_H(t_k-3)-0.1301\cdot10^{-3}\tau_H(t_k-4)\\
&-0.6113\cdot10^{-3}\tau_H(t_k-5)+0.0127\cdot10^{-3}\tau_H(t_k-6)\\
&+0.1971\cdot10^{-3}\tau_H(t_k-7)+0.0217\cdot10^{-3}\tau_H(t_k-8)
\end{aligned} \tag{3.22}$$

For four vibration modes $i=4$, model order is 10 :

$$\begin{aligned}
\mathbf{y_{tip}(t_k)_{/4}} = \quad &+1.1390\,y_{tip}(t_k-1)+1.2433\,y_{tip}(t_k-2)\\
&-1.3167\,y_{tip}(t_k-3)-1.1016\,y_{tip}(t_k-4)\\
&+0.9544\,y_{tip}(t_k-5)+0.8454\,y_{tip}(t_k-6)\\
&-1.0645\,y_{tip}(t_k-7)-0.3661\,y_{tip}(t_k-8)\\
&+0.8878\,y_{tip}(t_k-9)-0.3062\,y_{tip}(t_k-10)\\
&+0.0583\cdot10^{-3}\tau_H(t_k-1)+0.4965\cdot10^{-3}\tau_H(t_k-2)\\
&+0.2602\cdot10^{-3}\tau_H(t_k-3)-0.6346\cdot10^{-3}\tau_H(t_k-4)\\
&+0.0219\cdot10^{-3}\tau_H(t_k-5)+0.4743\cdot10^{-3}\tau_H(t_k-6)\\
&-0.2846\cdot10^{-3}\tau_H(t_k-7)-0.1749\cdot10^{-3}\tau_H(t_k-8)
\end{aligned} \tag{3.23}$$

$$+0.1224 \cdot 10^{-3}\tau_H(t_k - 9) + 0.0172 \cdot 10^{-3}\tau_H(t_k - 10)$$

The structure of all models is such that y_{tip} at the present value in time depends on a sequence of previous measured outputs $y_{tip}(t_k - i)$ and previous inputs $\tau_h(t_k - i)$.

3.2.6 Control Strategy for the Flexible Link Manipulator

Self tuning adaptive control algorithms are the control strategy used in this work. These algorithms can be direct STR's or indirect ones. In the direct algorithms, the controller parameters are estimated directly whereas the estimation for the indirect ones is done on the plant's model parameters rather than the regulator's parameters, which are calculated later. The first algorithm, which has already been implemented, is the General Predictive Control algorithm (GPC), and its results are presented in Section 2.3. As was presented in the previous section, the basic structure of the linear model for the flexible link is (without noise):

$$A(q^{-1})\,y(t) = b(q^{-1})\,u(t) \tag{3.24}$$

Figure 2.1 presents a block diagram of the control which was designed to deal with one structure of the plant's model at a time. We will examine the effects of changes in that structure. Actual changes should be done in the parameter estimation block once their number changes and in the controller calculations, since the dimensions of all polynomials and matrices will change.

3.3 Analysis and Results of Simulation and Control Work Performed

3.3.1 Introduction

The results presented in this section are for numerical simulations of the single flexible link controlled with the GPC algorithm. The "measured" outputs are produced by a simulation solving the dynamic equations for the flexible link as presented in Section 3.2 . Those equations are referred to as the "real plant" whose model is to be estimated. The unknown model is chosen in the form of Equation 2.6 (without the noise term). Different models for the different number of modes are presented in Subsection 3.2.5. In addition, the parameters of those models are presented, even though in actual situations they are unknowns. They are found by an estimation technique fitting for the plant to be estimated. The method used in this work so far has been Recursive Least Squares (RLS).

3.3.2 Effects of Under-Modeling and Over-Modeling on the Controlled Flexible Link

This section will present the simulation results of investigating under and over modeling of the actual plant, with regard to the structure. It means that for each figure, the actual number of modes are shown, i.e. modes that are used in the dynamic equations (Section 3.2), as well as the number of modes taken into account in the estimator model. With every choice of estimated structure for the plant's model, the number of parameters to be estimated changes.

In Figure 3.2, the flexible link was modeled with two vibration modes, which means that the "real" model of the link is of 6^{th} order (see Table 3.1). The estimated structure for the discrete time linear model was also of order 6. As mentioned before, the tip position of the flexible link is controlled with the GPC algorithm. The set point is a square wave between the values of ±1. The output, the tip position, follows it very well. Figure 3.3 shows the behavior of the angle θ and its derivatives for the same conditions as in Figure 3.2. Since it is a linear model and the length of the arm is 1 $m.$, and the effect of the vibration modes is small, y_{tip} and θ appear to be the same. Figures 3.4 and 3.5 present the vibration modes generalized coordinates q_1 and q_2 and their derivatives, and Figure 3.6 shows the torque input

.

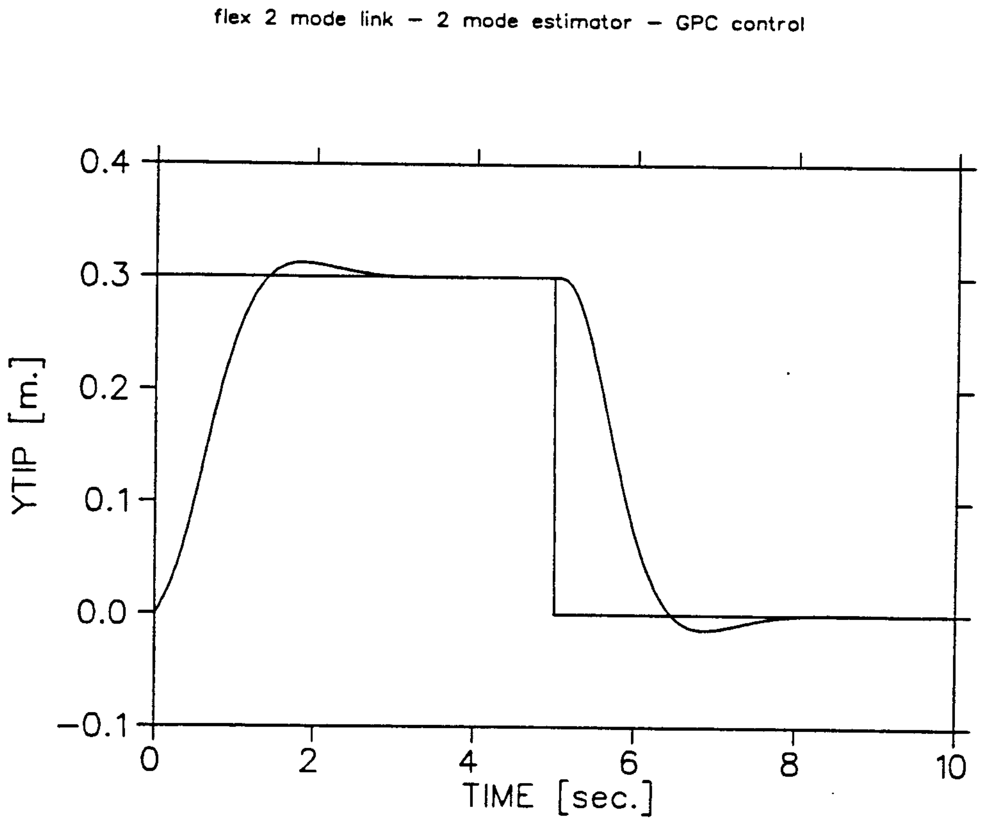

Figure 3.2: Two mode flexible link with two mode estimator model

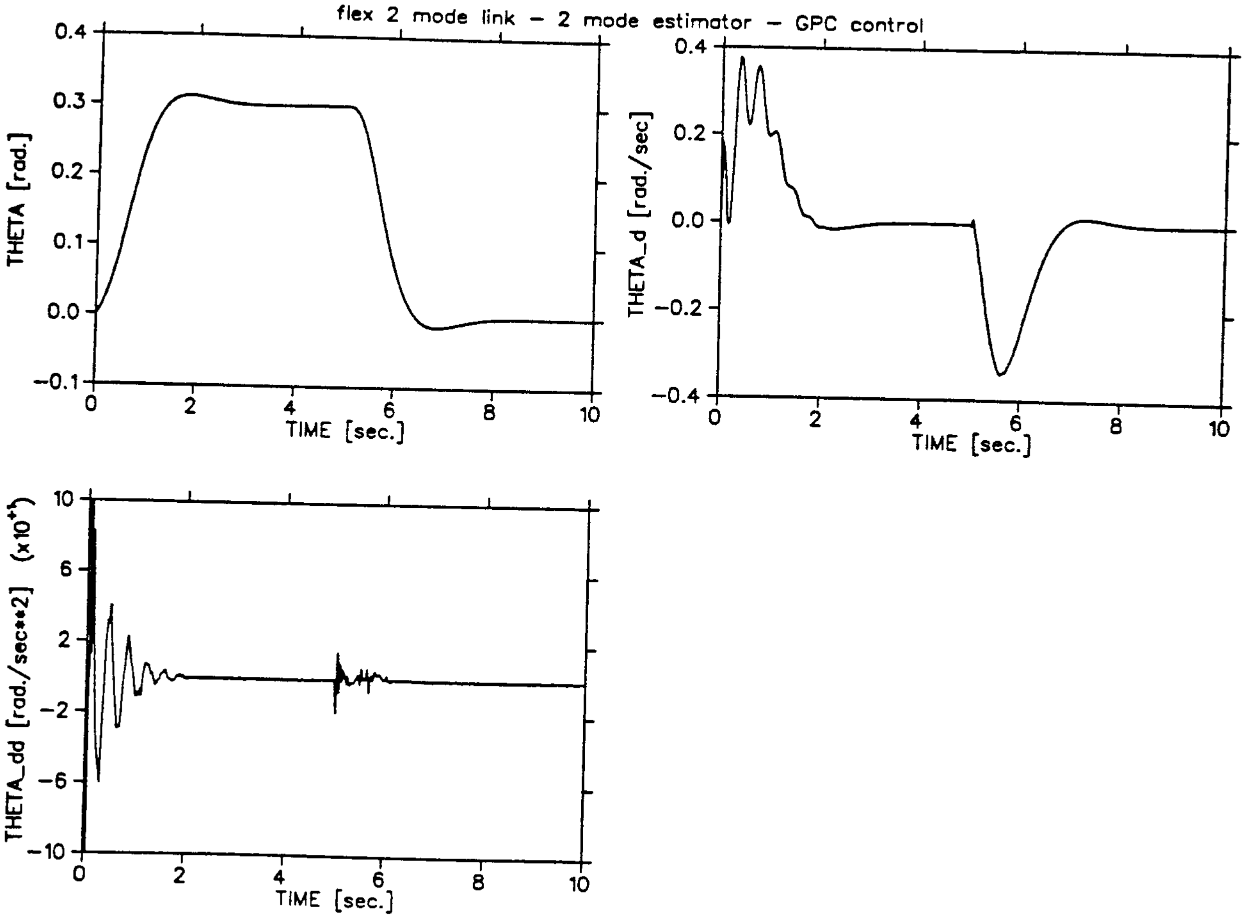

Figure 3.3: The angle θ and its derivatives

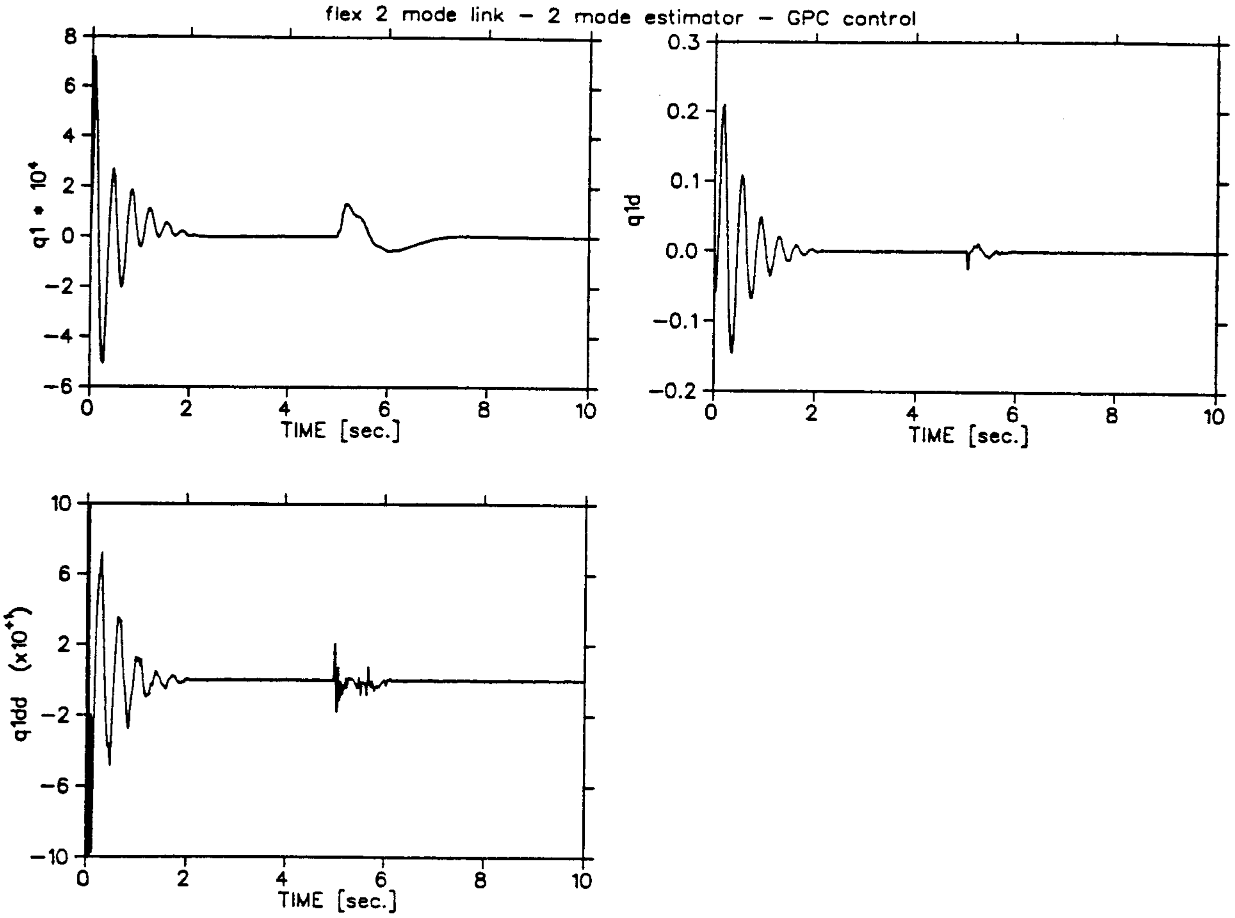

Figure 3.4: The generalized goordinate q_1 and its derivatives

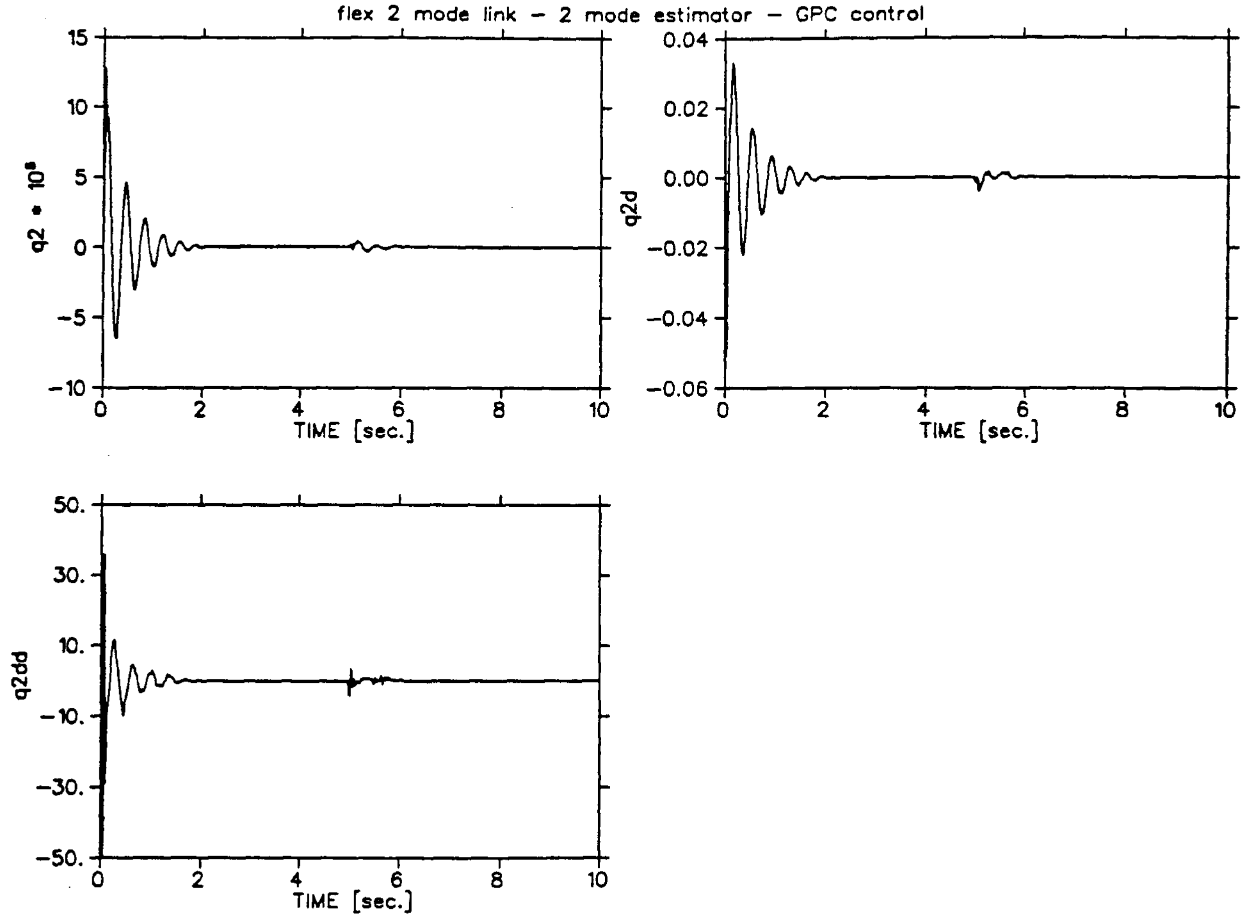

Figure 3.5: The generalized coordinate q_2 and its derivatives

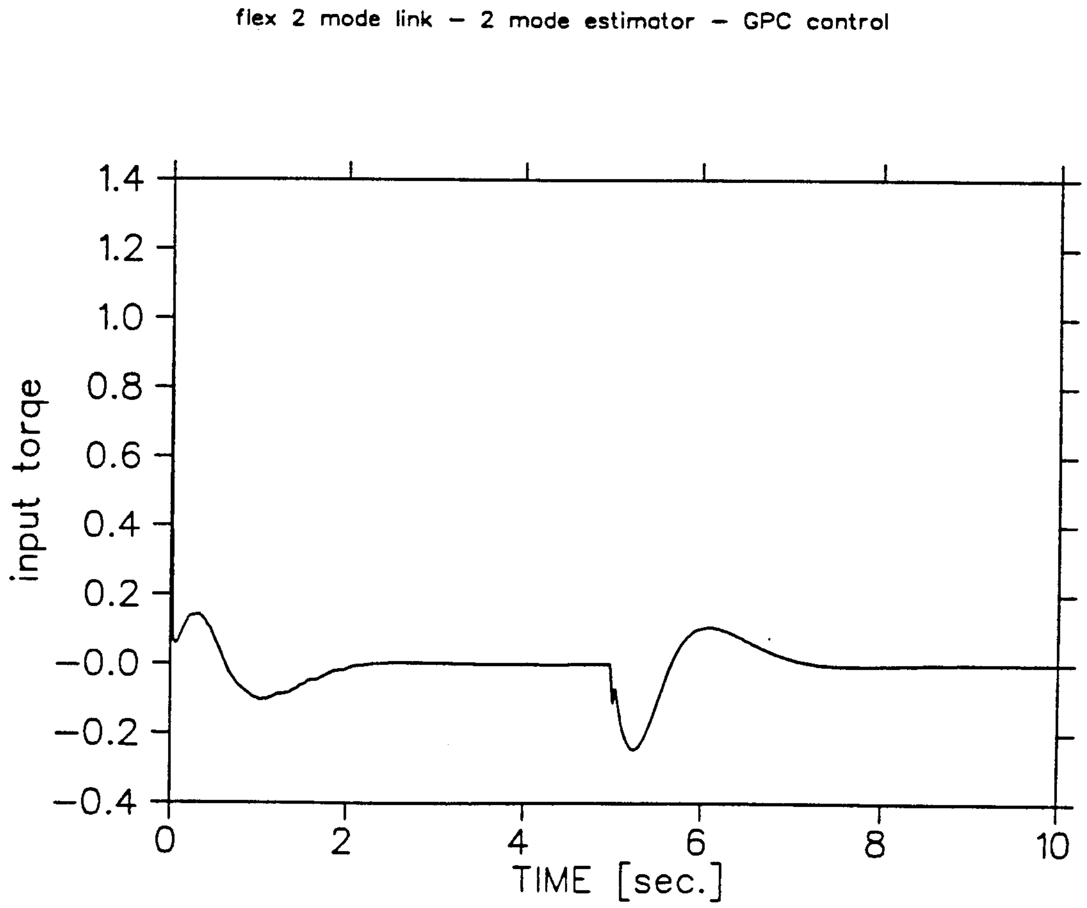

Figure 3.6: The torque input

The effect of *under-modeling* can be seen in Figure 3.7, where a "real" flexible two mode link (6^{th} order plant) is estimated for a model with a structure of one vibration mode (4^{th} order estimated model). The result is unstable. This unstable result was expected since stability analyses usually assume that the estimator model should be at least as complex as the plant itself.

Figure 3.8 presents the opposite effect of *over-modeling* where a flexible one mode link (4^{th} order) is modeled with a four mode estimator (10^{th} order). The output develops oscillations which diverge and ends with instability. The frequency of oscillations is about 14 [rad./sec.] which is the frequency of the first modes.

When the over-modeling is closer, as in Figure 3.9, where a 2 mode flexible link (6^{th} order) is modeled with an estimator model of 3 modes (8^{th} order), the response is not unstable. Here the output tracks the square wave of the set point as shown in Figure 3.2, where the simulated model and the estimator's model match in structure. These results are also checked in Chapter 5 for the two link rigid manipulator with the hydraulic actuators.

The *conclusions* so far are that under-modeling of a plant with an adaptive control system will most probably result in an unstable system. This is expected, if one observes assumptions made in stability proofs in the literature (Edgare [10] and Astrom [1]), which say that the estimated model should be at least as complex as the real plant model. There is more freedom in the choice of structure for over-modeling a plant. If the estimator is close to the actual model as in Figure 3.9, then the system behaves very well, but when the difference grows, as in Figure 3.8, instability can occur.

It seems that the instabilities in the over-modeling case are due to dynamics introduced in the control algorithm through the estimator model-dynamics which do not actually exist in the system but which are present for the control algorithm, since the estimator will give non-zero values to over modeled model parameters. There is an effort made to control an entirely different plant than the actual one, which is projected through the "measured" values

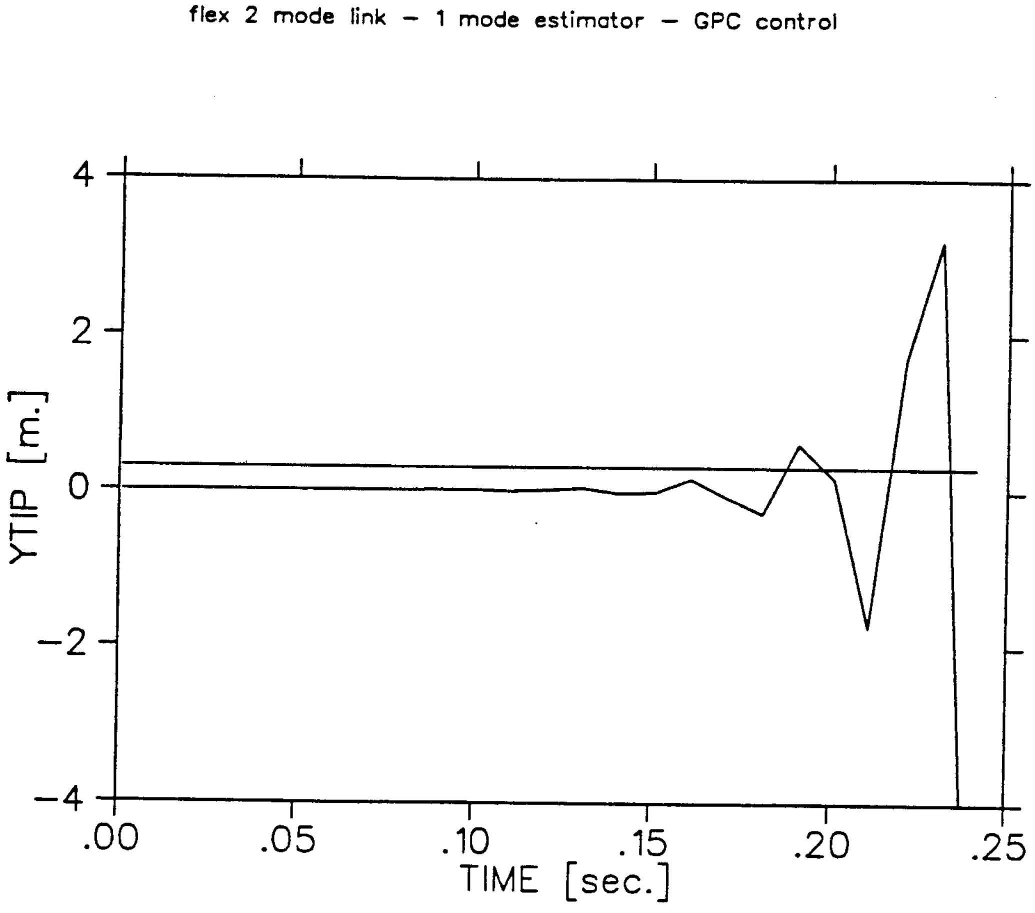

Figure 3.7: The effect of under-modelling

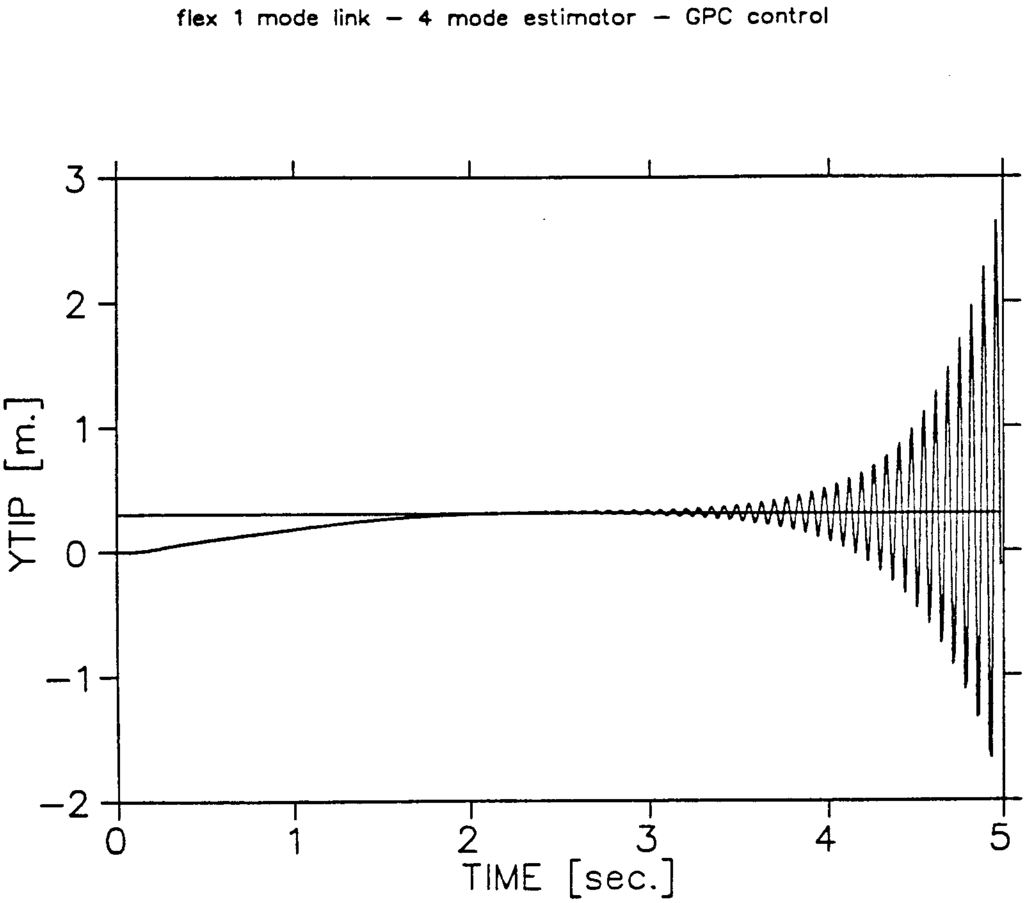

Figure 3.8: The effect of over-modelling

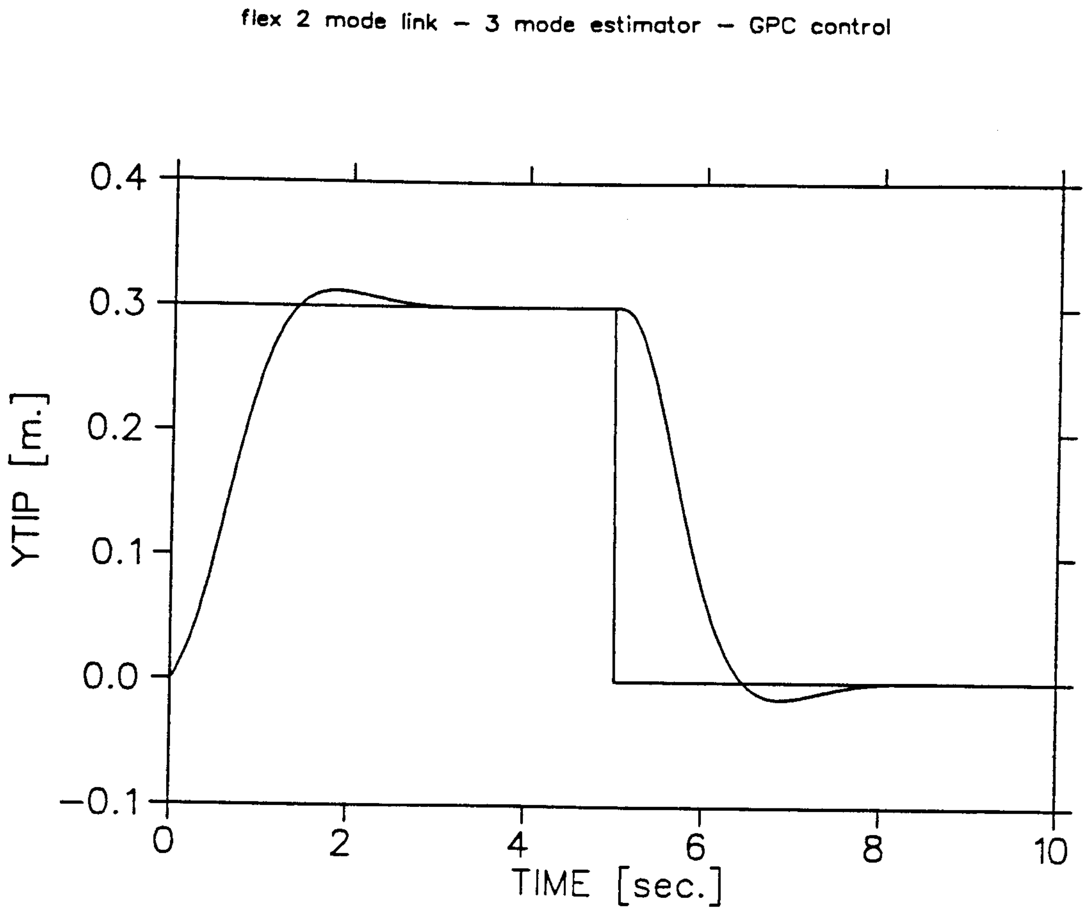

Figure 3.9: Over-modelled 2 mode link with 8^{th} order estimator

of y_{tip}. The next subsection describes the behavior of a cost function and a possible method of detecting the need to change the structure and integrate the new one, on line, into the control system.

3.3.3 Use of an Estimation Cost Function as a Criterion for Changing the Structure of the Plant's Model

The cost function is a tool for attaining an optimal behavior of a physical property of the system. One may want to optimize a trajectory for a robot arm or to optimize time or output error, as may be logical for this case. The cost function chosen, Equation 3.25, minimizes the output error between the tip position y_{tip}, the "measured" output as calculated from the equations of motion, and the one from the estimator model y_{est}.

$$J(y_{tip}, y_{est}) = \sum_{k=0}^{t}[y_{tip} - y_{est}]^2 \tag{3.25}$$

Figure 3.10 presents the behavior of the cost function in Equation 3.25. Its value rises initially when there is a difference between the model and plant dynamics, and then when the error goes to zero, it settles to a constant value.

There are, of course, additional possibilities for the choice of a cost function which are not limited to the one mentioned.

Another interesting cost function, described in Section 2.3, is the one used for the GPC algorithm (Equation 2.7). There an error is also minimized; however, the outputs are the ones predicted. Not only should the present output track a set point, but the values to be minimized are predicted ones, so the cost function ensures that the future error will be minimal. In order to have reasonable control inputs and not to demand, for example, extremely high input signals that may drive the system to saturation, the total sum of control increments is also minimized. The result is minimal output error, with minimal control effort.

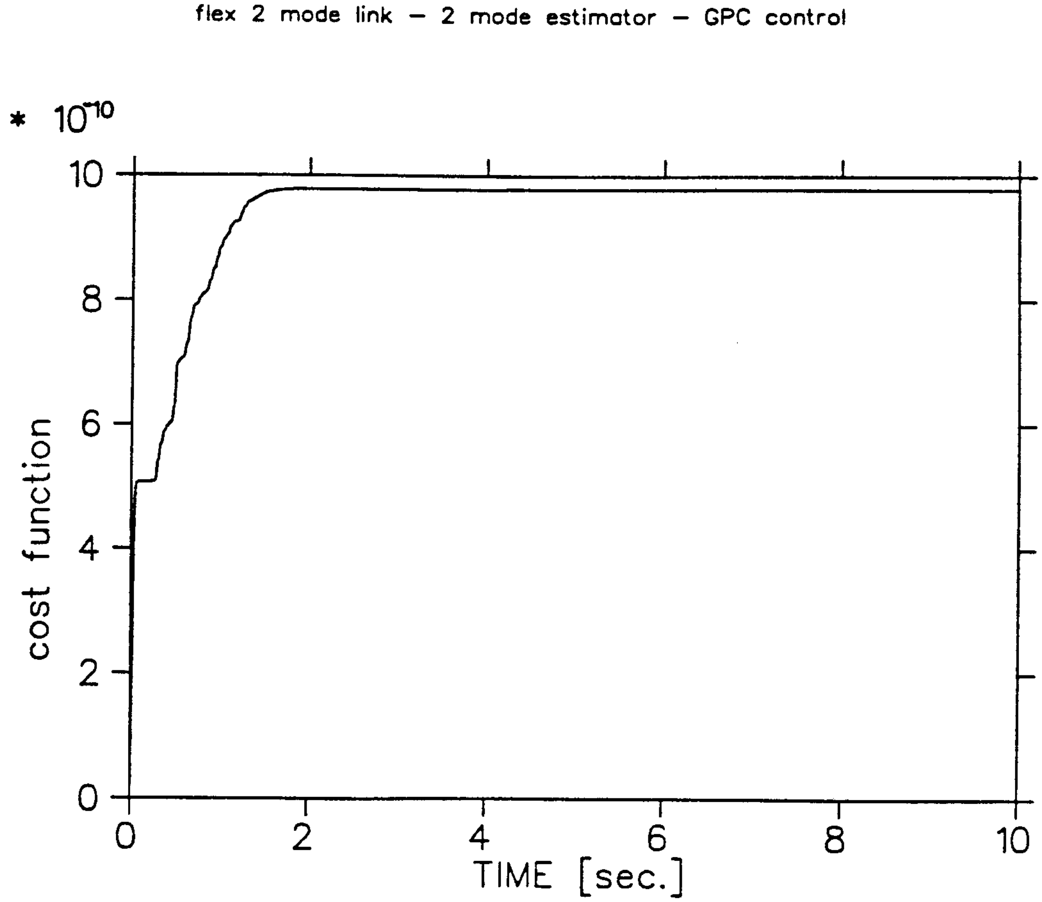

Figure 3.10: Estimator cost function for correct modelling

The behavior of the two cost functions for the flexible link application will be compared later in this work.

3.3.4 Effects of On Line Changes in Model order

Chapter 5 will present a full discussion on the effects of under and over-modeling on the cost function for both applications of this work, the flexible link manipulator and the hydraulically actuated two link manipulator. It will also present a method to detect structure modeling errors and correct them. In this section, preliminary discussion on the effects of the change of the estimator model structure on the controlled tip position of the flexible link is presented. The link itself, the "real" model, was chosen to have 2 vibration modes (i.e. 6^{th} order). Two cases are presented; in the first, the estimator model is initially a 2 mode model (i.e. 6^{th} order) and is then changed into a 1 mode model (i.e. 4^{th} order). In the second case, the estimator model is initially a 1 mode model (4^{th} order) and is then changed into a 2 mode model (6^{th} order) to match the "real" model mentioned above. The next figures will show numerical simulation results for on line model changes. For each case, the tip position, the estimator cost function, and the estimator output error behavior will be presented. It should be noted that the criterion used to change the structure of the estimator in the cases presented was time, which is not the final one (Chapter 6 presents the full criteria). The actual criteria are the changes in the values of the chosen cost function and its derivatives. In Figures 3.11 and 3.12 the 6^{th} order model converges to the set point, and after 5 sec., the estimator model has been changed to an under-modeled situation ($order = 4$) when the whole system goes unstable. In Figure 3.11 y_{tip} reaches instability after the change of the estimated model structure. Figure 3.12 shows the changes in the cost function and output error.

In Figure 3.13, Figure 3.14, Figure 3.15, Figure 3.16 the process starts with the wrong estimator model and is changed on line to the correct one.

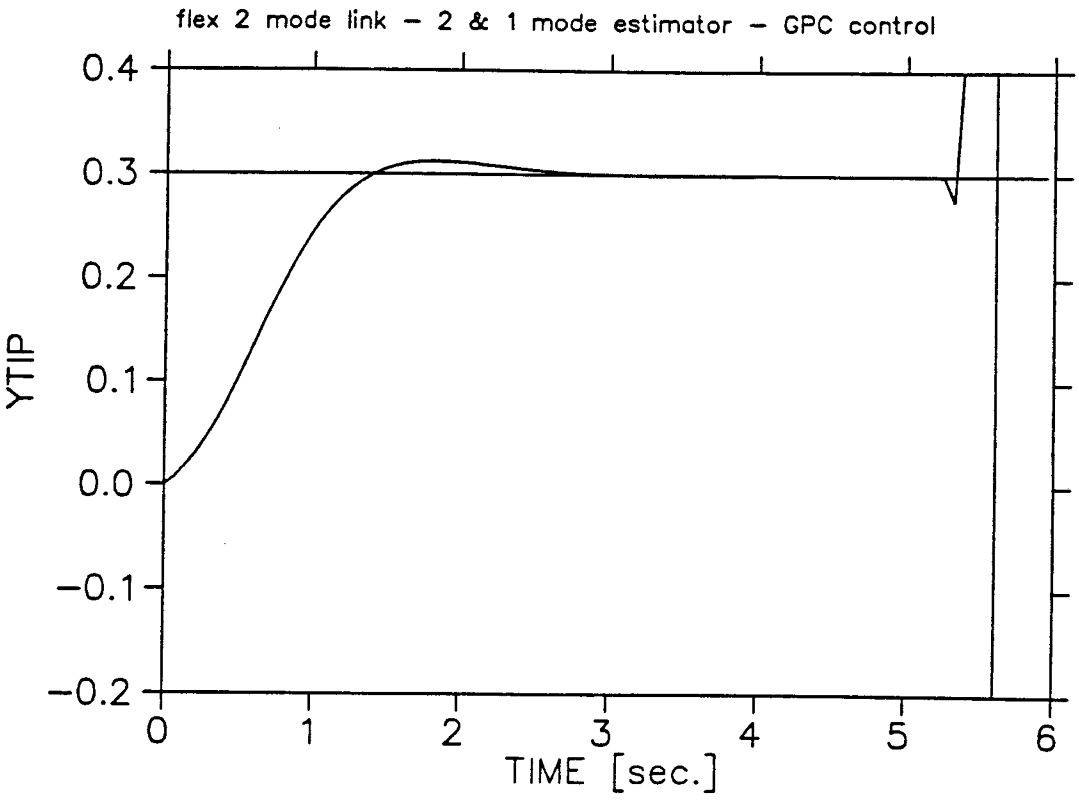

Figure 3.11: Online change of estimator model - correct to under- modeling

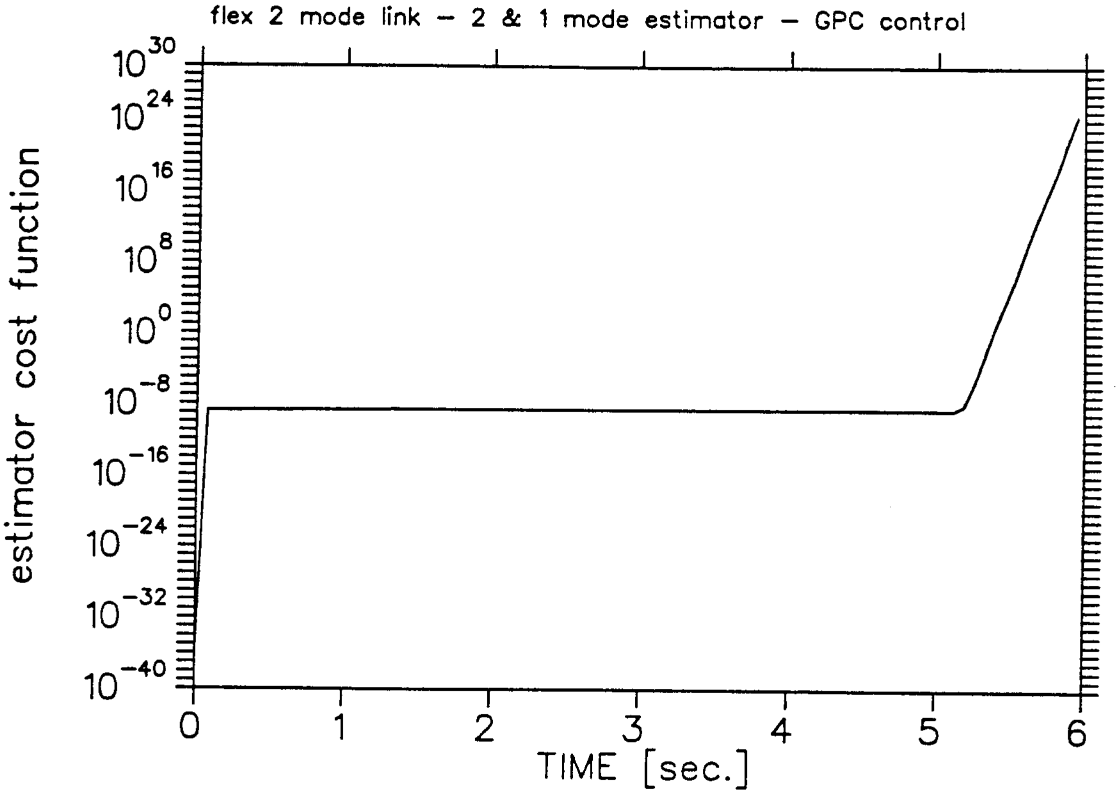

Figure 3.12: Estimator cost function for the on line change in model

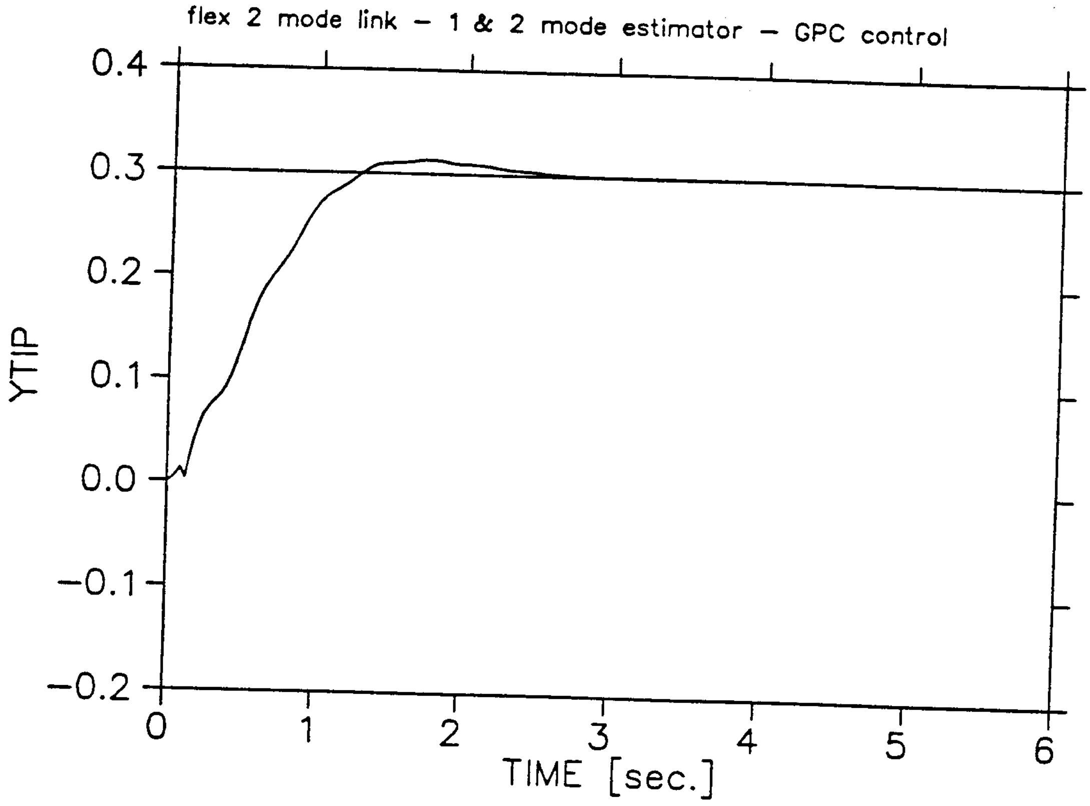

Figure 3.13: On line change of estimator model - under modelling to correct

In the examples presented, the output converges and tracks the set point. In Figures 3.13 the change in models is done at an early stage (0.1 sec which are 100 sampling steps), so y_{tip} converges well. Compared with Figure 3.2 it is slower, but the results are still satisfactory. The estimator cost function (Figure 3.14) converges to a higher value (order of magnitude of 10^{-4}) than the one (order of magnitude 10^{-9}) which exists when both models match. In Figures 3.15 and 3.16 the correction of the model is done later, so the system gains more error from the wrong estimator model. The convergence takes longer than in Figure 3.13, and the estimator cost function has much larger values.

3.3.5 Comparison Between the Behavior of Two Different Cost Functions

It is of interest to compare the estimator cost function behavior as presented in Equation 3.25 and the GPC control algorithm cost function as presented in Equation 2.7. Figures 3.17 3.18 3.19 present such comparisons for different mis-modeling cases.

When writing GPC error or estimator output error, the calculations are of the terms within the sum symbols in both Equation 2.7 and 3.25, respectively. Those terms are calculated at each time step. When writing the GPC cost function or the estimator cost function, the calculations are accumulated with time. Figure 3.17 is the cost function for the changing estimator model. It starts with the correct 6^{th} order model that changes to a 4^{th} order one after 5 seconds, as in Figure 3.11. Figures 3.18 and 3.19 present the opposite case, where the difference between the two is the time of switching models (as in Figures 3.13 and 3.15). The two cost functions are functions of different variables. The estimator cost function (Equation 3.25) is the sum of the square estimator error, which is the difference between y_{tip}, the measured value, and y_{est} , the estimator output. The GPC cost function is the sum of the square error between the predicted output and future set points and the sum of the weighted square future control effort. Yet even though the behavior of both cost functions is very similar, their values are different. This may promote the use of different possible cost

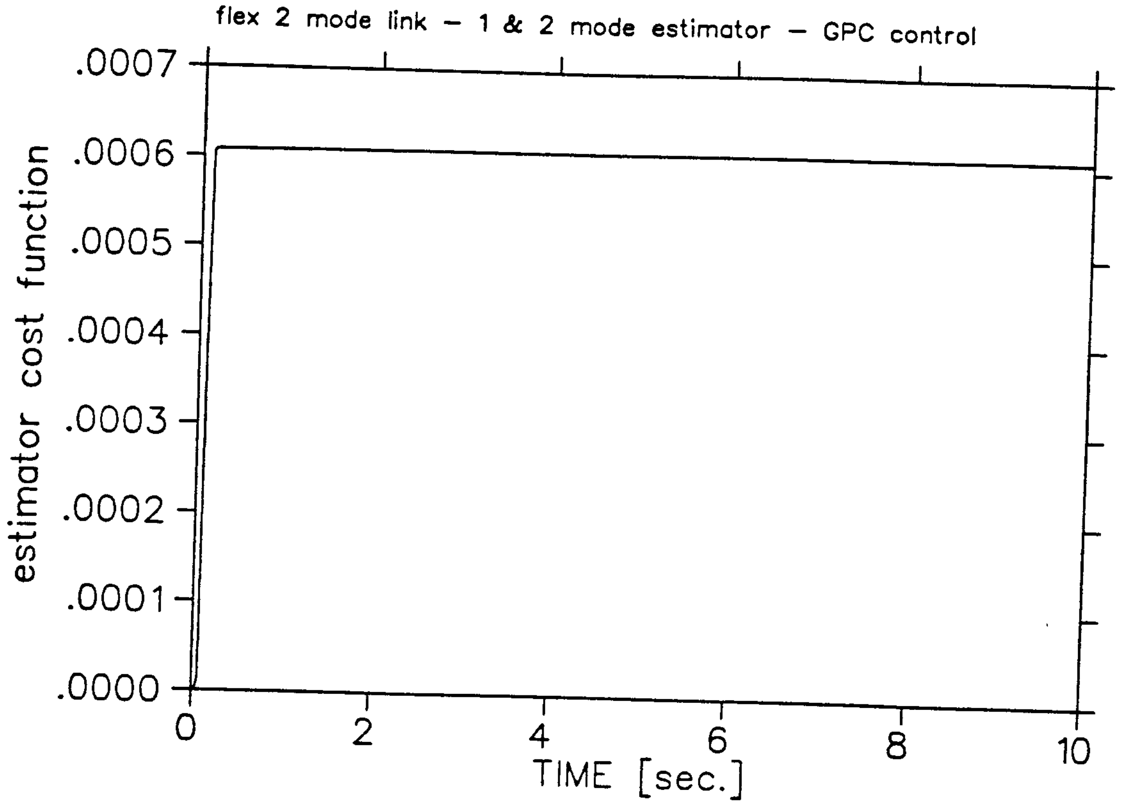

Figure 3.14: Estimator cost function for the on line change in model

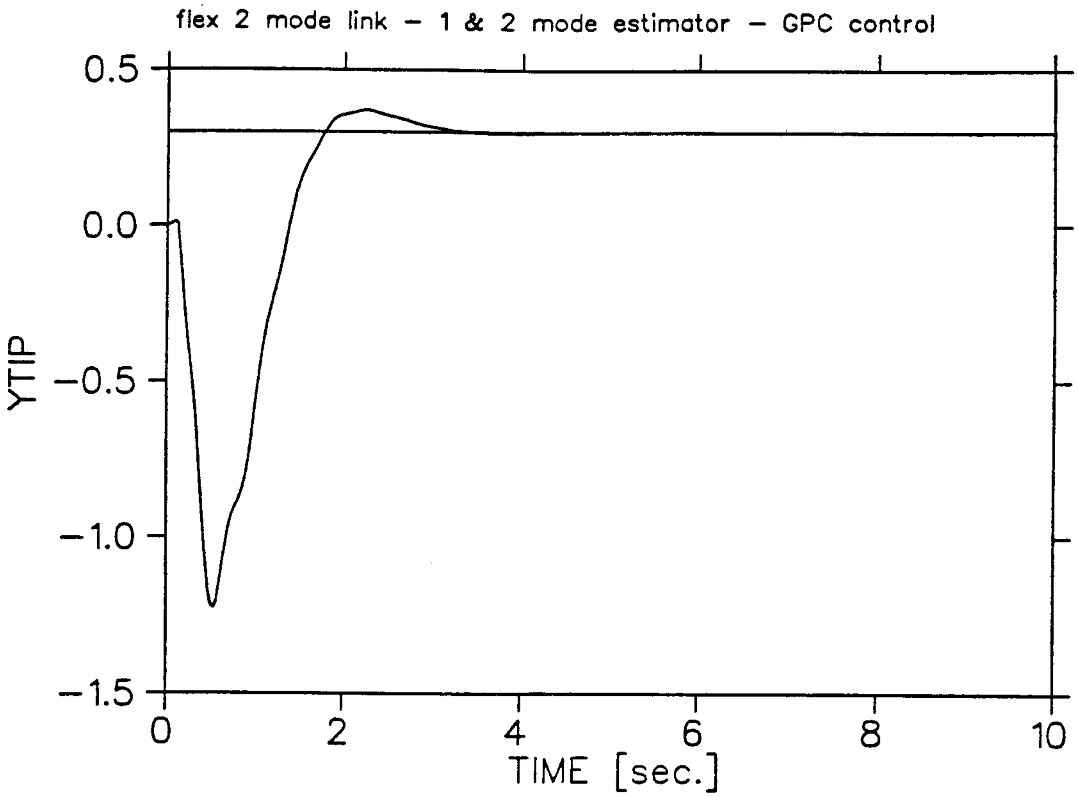

Figure 3.15: On line change of estimator model - under modelling to correct

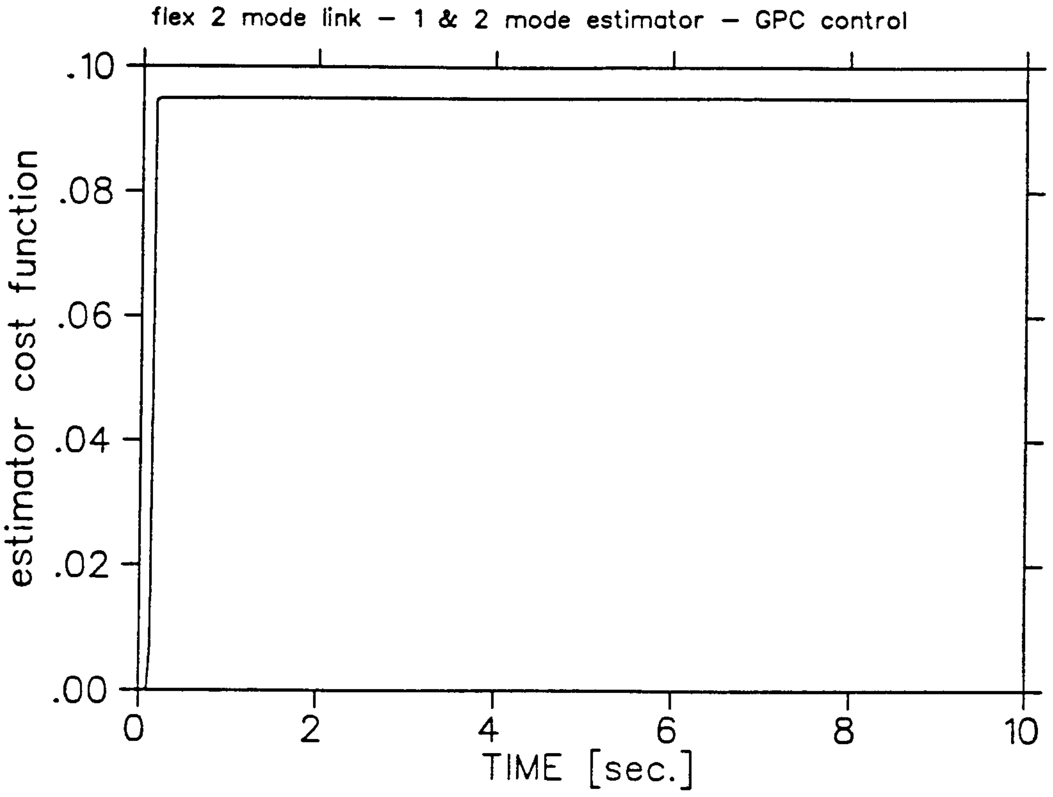

Figure 3.16: Estimator cost function for the on line change in model

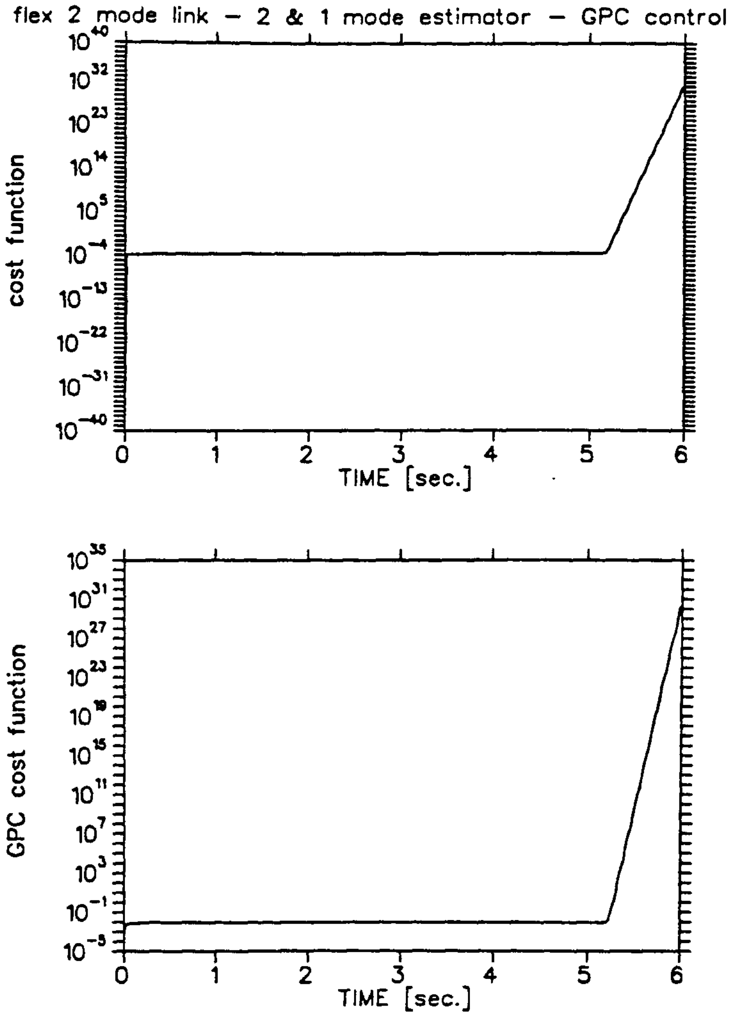

Figure 3.17: GPC and estimator cost function - online correct to under-modeling

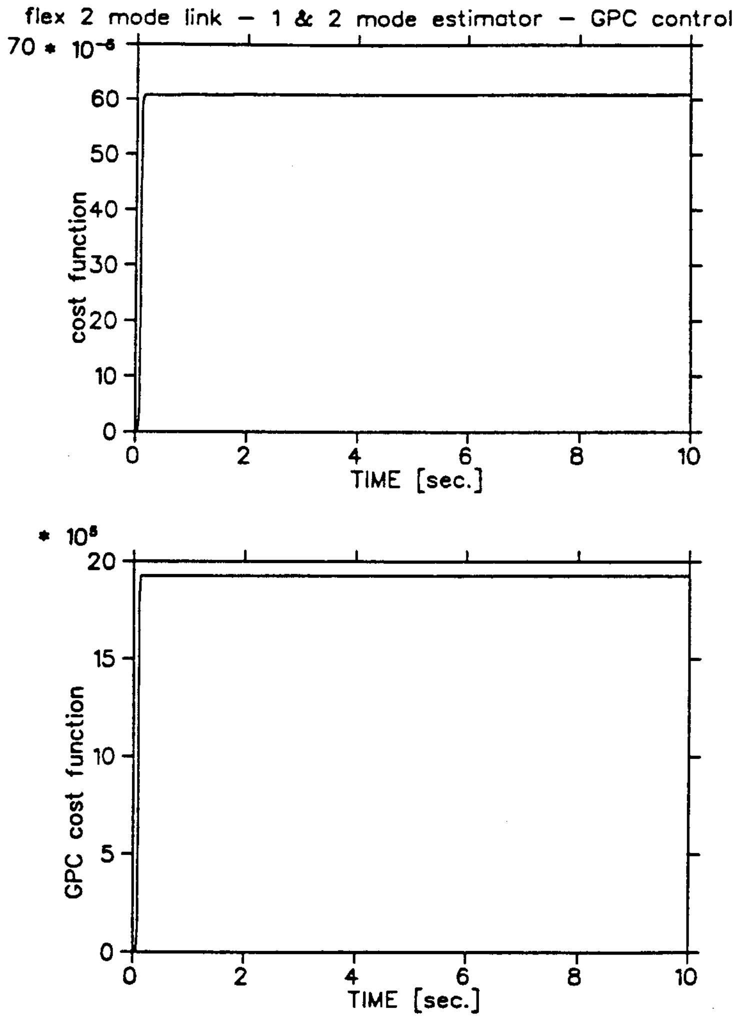

Figure 3.18: GPC and estimator cost function - online under-modeling to correct (0.1 seconds)

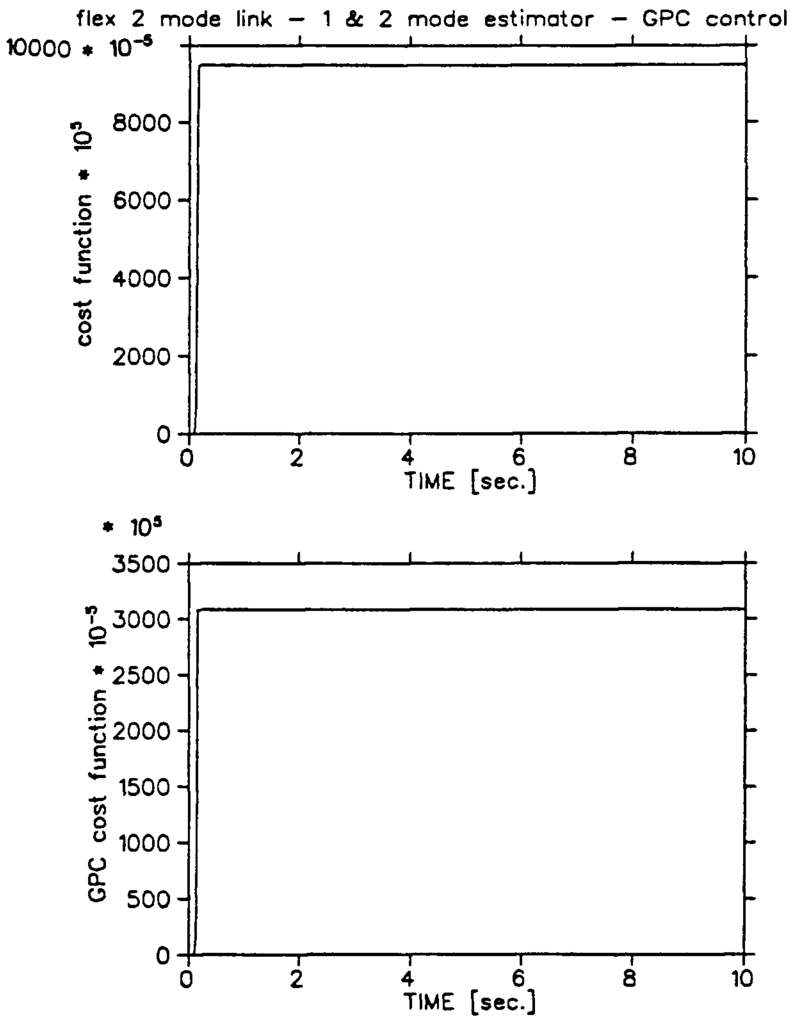

Figure 3.19: GPC and estimator cost function - online under-modeling to correct (0.2 seconds)

functions for the model structure changes algorithm.

3.4 Conclusions

An interesting point of this research is the response of adaptively controlled systems to on-line changes in the model structure due to variations in operating conditions. Adaptive algorithms usually use an estimation procedure for the plant or controller parameters in which the structure of the plant's model is assumed to be fixed. Estimated values are considered to be correct, and uncertainties in those values are ignored (the certainty equivalence principle). Reliance on that principle can lead to instability in the system.

A good example is a flexible link manipulator, where changes in load during a working cycle can result in the rise of vibration modes which were not there before. This chapter presents the equations of motion for a single flexible link manipulator which is controlled with a General Predictive Control algorithm (GPC) and the parameters estimated with the Recursive Least Squares (RLS) algorithm. Simulation results of the controlled system are presented. Under-modeling of the plant's dynamics (i.e. the order of the estimator is smaller than the "real" order) leads to instability. Over-modeling could also lead to instability when the gap between the estimated model order and the actual system's model is too large. However, there are conditions under which the system behaves well with over-modeling. It is also shown that a system which begins with an under-modeled estimator plant, and is then changed to the correct one, will not become unstable under the right conditions, as it would have if the change in the estimator had not been done.

It is suggested that the cost function presented in Equation 3.25 may be a criterion to detect the need for the estimator's order change. A change of the estimator's model structure on line requires a change in the controller's structure as well.

The work in Chapter 5 puts together the results in Chapter 3 that detect the need in model structure change and execute it when the system is controlled with the GPC adaptive algorithm. This chapter shows the results of correcting mis-modeling by using a time criterion, i.e., a system that could be unstable, but with the correction has an acceptable performance.

No. of Modes	*0*	*1*	*2*	*3*	*4*	*5*
Order of the system	2	4	6	8	10	12

Table 3.1: No. of flexible modes vs. order of system

Chapter 4

TWO LINK MANIPULATOR WITH HYDRAULIC ACTUATORS

Outline

Robotic manipulators consist of links (rigid or flexible), connected by joints that control the relative motion of neighboring links. The joints usually have position sensors which measure the relative motion and are actuated by electric, pneumatic or hydraulic drives [51]. These systems are subject to nonlinearities such as coupling, coulomb friction and backlash. Their inertial characteristics and loads vary during operation and are not always predictable [17]. Hydraulically actuated manipulators are widely used in industry today. Hydraulic systems have relatively large torque to weight ratios, higher loop gains and wider bandwidths compared with electrical motors [52]. Hydraulic robots are used for heavy duty tasks requiring position accuracy, rapid dynamics and rapid start and stop. However, hydraulic systems are complex , nonlinear and difficult to analyze for control purposes [53], [54].

In the design of a manipulator control strategy two kinds of physical quantity should be considered, those that can be determined accurately with the values remaining relatively constant, and those that vary within a range of values during a working cycle. The second type of quantities cannot always be avoided in a control system and may require an on-line change of the controller parameters. Examples in hydraulic systems would be external and internal leakages, size of orifices, temperature changes, accumulation of oil contamination, viscosity changes of the hydraulic fluid, damping coefficient etc. In the modeling of the links, one can find changes in the moments of inertia during a working cycle when an external load (in some cases an unknown external load) is being picked up and put down. Compliance in the links may give oscillatory dynamics with low damping, i.e. excite vibration modes that change the model of the controlled system.

Most of the techniques for control system design assume the plant and its environment are known. In many cases however, this is not so, since the plant might be too complex, the model not fully understood, or the process and the disturbances changing with operating conditions. When a system's dynamic model is uncertain or has the possibility of changing its parameters on-line, adaptive control may be considered.

Control of robotic systems has been widely discussed in the literature before, (Fu et al [16], An et al [55], Asada [56], Craig [51] and others.) The dynamics of the actuators are usually ignored and the link motion provides second order equations with coupling effects [16]. Sepehri et al [54] show a control strategy in which the link motion is controlled by a self tuning algorithm (minimum variance control), and the hydraulics is controlled by a classical control algorithm.

A study on a hydraulic manipulator controlled by an adaptive algorithm was presented by Vaha in 1988 [53]. The control algorithm was based on a one-step-ahead self-tuning controller proposed by Clarke et al in 1975 [12]. An integral term was introduced to a quadratic performance criterion which was minimized to find a control law that was applied to a heavy duty manipulator in two ways. An experimental study was performed in order to evaluate the applicability of the adaptive algorithm to control the movement of the manipulator's links. The autoregressive model chosen for the adaptive control process experienced difficulties caused by mechanical and physical characteristics and measurement noise. As well, the study considered simulation evaluation of the problem. The model chosen to simulate the actual manipulator was a linearized second order model.

As previously mentioned, in 1987 Clarke et al [4], [5] developed the Generalized Predictive Control technique which may have advantages for the control of complex systems such as heavy duty manipulators.

The present work applies the GPC algorithm to an extensive mechanical and hydraulic system model of an industrial hydraulic manipulator, to assess the control capability of this more recent algorithm on such a nonlinear system, and to study the effect of the design parameters, Kotzev et al [57] .

First the dynamic model of the manipulator is presented as well as the equations of motion of the hydraulic actuator including compliance, dead time and full dynamics of the servovalve, resulting in a rather complex nonlinear system in which the order of the estimated linear model for the GPC may vary from 6 to 10 . The GPC algorithm is also presented in detail. It uses for control purposes a linearized model of the system. The control law derived depends on values of the measured output from the nonlinear system, and uses assumed and estimated parameters for the linear model.

The control strategy in this work, consists of two adaptive loops, in which the process model contains the manipulator link with the hydraulic actuator. There is an advantage in combining all the system states into one control loop, where the system is represented by an input/output model in the GPC, since the estimated parameters can reflect all changes in the system as well as the uncertainties, disturbances, nonlinearities and coupling, provided that safety limits on the required system variables exist. This approach can be implemented on any hydraulic manipulator with as many links and actuators as required. It can also be implemented on manipulators with other actuators such as electric motors. The results show the effects of the different control tuning parameters on the controlled system performance. The GPC has an inherent integrator which helps overcome offsets but results in undesirable overshoot when operating robot manipulators.

In advancing the state of the art of predictive control, in this work special attention is given to the maximum output horizon, which for larger values (i.e. larger prediction horizon), has stabilizing effects and damps the output behavior but slows the transient response. The work also introduces an on-line automatic change of the maximum output horizon so that the transient response can be sufficiently fast and undesirable overshoots avoided. Further advances are also made in the selection of other GPC design parameters.

The dynamic equations of the manipulator and its actuators have been simulated in a FORTRAN program along with the control algorithm and the numerical simulations were performed on a VAX 3200 computer.

4.1 Rigid Two Link Manipulator with Hydraulic Actuators

4.1.1 Introduction

A two link rigid manipulator is a complex and nonlinear system. The coupling between the motion of the arm's components introduces nonlinearities. The hydraulic actuators consist of servovalves and cylinders and may be described as a third or a fifth order system. The next two sections will present the equations of motion of the dynamics of the manipulator links, and the equations of motion of the hydraulic actuators, which will be expressed in equations for solving $\dot{P}_{i_{in}}$,$\dot{P}_{i_{out}}$ and $\ddot{\theta}_i$ The state space vector, of order 8, for this specific system is:

$$\mathbf{x} = \left[P_{1_{in}} \;\; P_{1_{out}} \;\; \theta_1 \;\; \dot{\theta}_1 \;\; P_{2_{in}} \;\; P_{2_{out}} \;\; \theta_2 \;\; \dot{\theta}_2\right] \tag{4.1}$$

4.1.2 Equations of Motion for the Rigid Two Link Manipulator

Figure 4.1 shows the configuration of the two link manipulator.

Using a general formulation,^{16}, the dynamic equations of motion derived via Lagrange's approach are:

$$\tau_i = \sum_{j=0}^{n} D_{ij}\ddot{\theta}_j + \sum_{j=0}^{n}\sum_{k=0}^{n} D_{ijk}\dot{\theta}_j\dot{\theta}_k + D_i \tag{4.2}$$

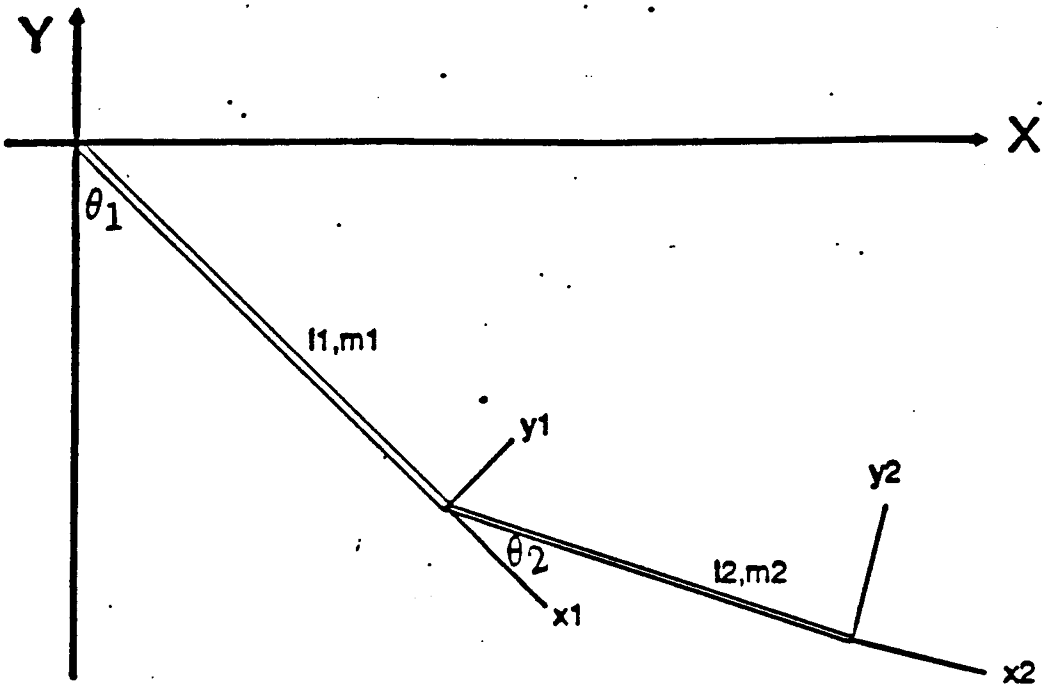

Figure 4.1: Configuration of the two link manipulator

Figure 4.1: Configuration of the two link manipulator

Where n is the number of degrees of freedom, D_{ij} terms for effective and coupling inertia at joint i due to link j motion, D_{ijk} terms for the Coriolis and centripetal forces at joint i as a result of motion in links j and k, and D_i are terms for gravity loading at joint i.

The kinetic energy of the system is:

$$\begin{aligned} T_k = \frac{1}{2}m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ + m_2 l_1 l_2 \cos\theta_2(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2) \end{aligned} \tag{4.3}$$

The potential energy of the system is:

$$P = -m_1 g l_1 \cos\theta_1 - m_2 l_1 \cos\theta_1 - m_2 g l_2 \cos(\theta_1 + \theta_2) \tag{4.4}$$

Combining and applying Lagrange's equation, the nonlinear equations of motion are:

$$\begin{aligned} \tau_1 = \quad & \left[(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos\theta_2\right]\ddot{\theta}_1 \\ & + \left[m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2\right]\ddot{\theta}_2 \\ & -2m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_2 \\ & \sin\theta_2 \dot{\theta}_2^2 + (m_1 + m_2) g l_1 \sin\theta_1 + m_2 g l_2 sin(\theta_1 + \theta_2) \end{aligned} \tag{4.5}$$

$$\begin{aligned} \tau_2 = \quad & \left[m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2\right]\ddot{\theta}_1 \\ & + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1^2 \\ & \dot{\theta}_1^2 + m_2 g l_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{4.6}$$

Where τ_1 and τ_2 are input torques to the joints.

4.1.3 Equations of Motion of the Hydraulic Actuator

The links of the manipulator are actuated by hydraulic actuators. Each link is activated by a hydraulic motor which is connected to a servovalve through expandable hoses. The servovalve monitors the flow of the hydraulic liquid. Figure 4.2 describes a critical center symmetric valve.

The supply pressure is kept constant which allows each servovalve to function independently. The return pressure is the atmospheric pressure, since it is connected to a storage tank. Components such as check valves and relief valves are for machine safety. The servovalves control the fluid power. The most widely used valve has a spool valve type construction, and is classified by the way flow goes through the valve. The valve variables are the spool displacement (X_{V_i}), the flows in and out of the valve $q_{i_{in}}$ and $q_{i_{out}}$, the supply pressure (P_{sup}), the return pressure (P_{res}), and the line pressures ($P_{i_{in}}$ and $p_{i_{out}}$). The equations describing the equations of motion for the valves are nonlinear. The flow equations are ^{52}:

$$X_{V_i} > 0, \qquad (positive \quad direction)$$

$$q_{i_{in}} = K_{valve} \; X_{V_i} \sqrt{P_{sup} - P_{i_{in}}} \tag{4.7}$$

$$q_{i_{out}} = K_{valve} \; X_{V_i} \sqrt{P_{i_{out}} - P_{res}} \tag{4.8}$$

$$X_{V_i} < 0, \qquad (negative \quad direction)$$

$$q_{i_{in}} = K_{valve} \; X_{V_i} \sqrt{P_{i_{in}} - P_{res}} \tag{4.9}$$

$$q_{i_{out}} = K_{valve} \; X_{V_i} \sqrt{P_{sup} - P_{i_{out}}} \tag{4.10}$$

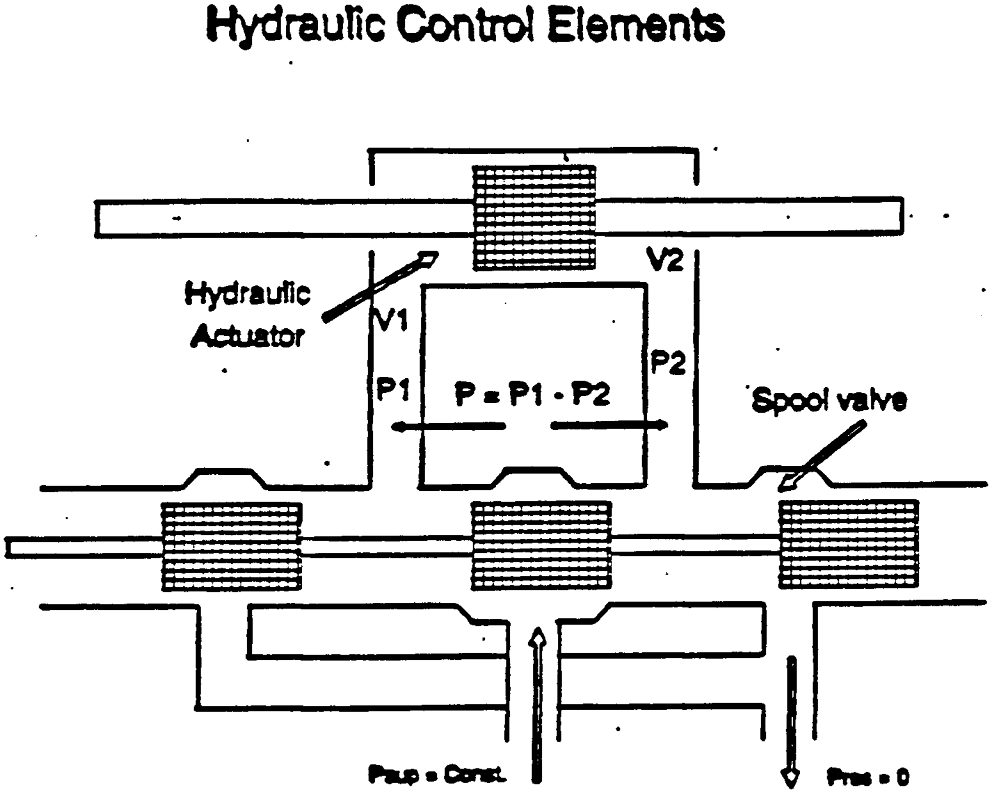

Figure 4.2: Electrohydraulic actuator

Linearization of these equations with a Taylor series expansion about zero spool displacement, for initial design purposes only, gives:

$$q_{i_{in}} = K_{x_i}\, X_{V_i} - K_{p_i}\, P_{i_{in}} \tag{4.11}$$

$$q_{i_{out}} = K_{x_i}\, X_{V_i} + K_{p_i}\, P_{i_{out}} \tag{4.12}$$

Where K_{x_i} and K_{p_i} are the flow gain and the flow pressure coefficients, respectively.

A first order model, [58], describing the equations of the pipes model are:

$$\dot{P}_{i_{in}} = \frac{\beta}{V}\,(q_{i_{in}} - D_m\, \dot{\theta}_i) \tag{4.13}$$

$$\dot{P}_{i_{out}} = \frac{\beta}{V}\,(D_m\, \dot{\theta}_i - q_{i_{out}}) \tag{4.14}$$

Where D_m is the volumetric displacement of hydraulic motor, and $\frac{V}{\beta}$ is the hydraulic compliance.

The motor and link dynamic model is :

$$T_i = (P_{i_{in}} - P_{i_{out}})\, D_m = j_{mi}\, \ddot{\theta}_i + b_m\, \dot{\theta}_i + \tau_i \tag{4.15}$$

Where the first term expresses the movement of the hydraulic motor, the second is a damping term, τ_i expresses external load by the links movement and T_i is the applied torque to the link i.

4.2 Control Strategy

4.2.1 Introduction

As mentioned before, the equations of motion of a robotic manipulator contain nonlinearities, inertial characteristics and disturbances that vary during a working cycle and may not always be predictable. Lately self tuning predictive algorithms have been used since the results have better robustness compared with other self tuning control algorithms such as Pole Placement and Generalized Minimum Variance (Astrom [1]). The robustness of predictive algorithms is due to the minimization of a multi-step cost function [4]. The basic predictive method has the following steps:

1. Prediction of the output in the future.
2. Choice of the future set points, and minimization of a cost function calculated from the future errors, between the future outputs and future set points, yields a set of future control signals.
3. The first element of the control signals is actually used and the whole procedure is repeated. This is a receding-horizon controller.

The type of controllers mentioned above consider the output at one point of time in the future. The Generalized Predictive Control (GPC) [4], [5], algorithm minimizes a cost function that considers the future predicted outputs j steps ahead, the future set points and future control signals.

4.2.2 Control Strategy for the Two link Rigid Manipulator

Figure 4.3 presents a block diagram for controlling the tip position of the two link manipulator with the hydraulic actuators.

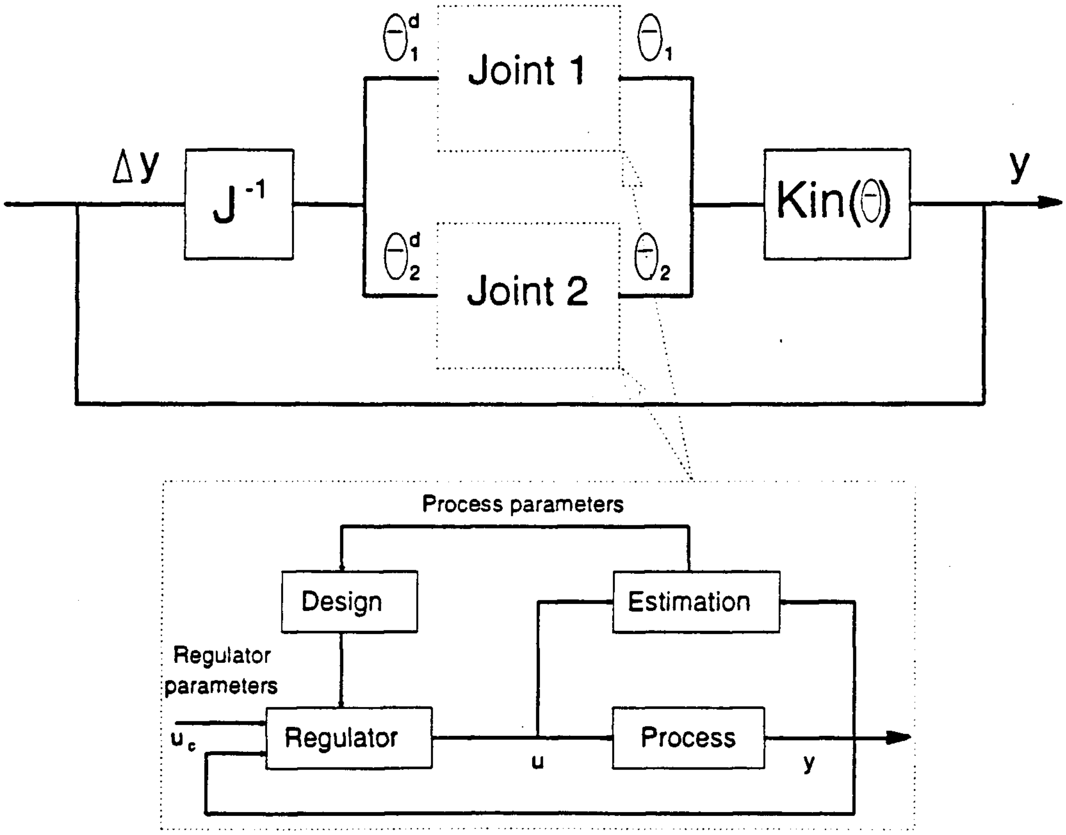

Figure 4.3: Control strategy for the two link manipulator

The tip location error is translated to angle changes in the joints. The control consists of two adaptive loops, in which the process model contains the manipulator link with the hydraulic actuator. Each of the joint links is controlled separately with the general predictive control algorithm. The model for each loop will be an Input/Output type of model in the form of :

$$A(q^{-1})y(t) = b(q^{-1})u(t) + c(q^{-1})e(t) \tag{4.16}$$

Where:

$y(t)$ is the output - joint angle.

$u(t)$ is the input to the process - spool displacement

$e(t)$ is the noise sequence.

4.3 Analysis and Results of Simulation

4.3.1 System Parameters

The link parameters are:

$$l_1 = 50 \ cm.$$

$$l_2 = 50 \ cm.$$

$$m_1 = 1 \ kg.$$

$$m_2 = 1 \ kg.$$

The hydraulic actuator parameters are:

$$K_{valve} = 243. \sqrt{\frac{cm^5}{kg}}$$

$$D_m = 8.2 \ cm^3$$

$$\frac{\beta}{V} = 3. \ \frac{kgf}{cm^5}$$

$$P_{sup} = 65. - 250. \ \frac{kgf}{cm^2}$$

$$P_{res} = 0. \ \frac{kgf}{cm^2}$$

$$K_{x_i} = 1387. \ \frac{\frac{cm^3}{sec}}{cm}$$

$$K_{p_i} = 4.65 \ \frac{\frac{cm^3}{sec}}{\frac{kgf}{cm^2}}$$

The range of the spool displacement is:

$$-0.5 \ cm. \ \leq X_{V_i} \geq 0.5 \ cm.$$

Where $i = 1, 2$ for the number of links.

4.3.2 Open Loop Analysis

The linearized equations of motion, i.e. Equation 4.11, Equation 4.12, Equation 4.13, Equation 4.14, Equation 4.15 and linearized Equation 4.5, Equation 4.6 produces the following state space form, for a single link and are used only for preliminary study and design :

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}X_{V_i} \tag{4.17}$$

where the state space vector for one loop is:

$$\mathbf{x} = \left[\; P_{L_i} \;\theta_i\; \dot{\theta}_i \;\right]^T$$

Allow $P_{L_i} = P_{i_{in}} - P_{i_{out}}$, and:

$$\mathbf{A} = \begin{pmatrix} -\frac{\beta}{V}K_{P_i} & 0 & -2\frac{\beta}{V}D_m \\ 0 & 0 & 1 \\ \frac{D_m}{Jl_i} & 0 & 0 \end{pmatrix} \tag{4.18}$$

Where the eigenvalues are:

$$[\quad 0.9997413 \quad 1.072178 \quad 0.9996837 \quad]^T$$

$$\mathbf{B} = \begin{pmatrix} 2\frac{\beta}{V}K_{x_i} \\ 0 \\ 0 \end{pmatrix} \tag{4.19}$$

The transfer function between the angle and spool displacement is:

$$\frac{\theta_i(s)}{X_{V_i}} = \frac{2\frac{\beta}{V}\,D_m\,K_{x_i}}{s\;(Jl_i\;s^2 + Jl_i\;\frac{\beta}{V}K_{p_i}\;s + 2\frac{\beta}{V}D_m^2)} \tag{4.20}$$

The system was converted with a zero order hold sampling into a discrete time form, the input/output model is:

$$\begin{aligned} \theta_i(t_k) = \quad & -a_1\,\theta_i(t_k-1) - a_2\,\theta_i(t_k-2) - a_3\,\theta_i(t_k-3) \\ & +b_1\,X_{V_i}(t_k-1) + b_2\,X_{V_i}(t_k-2) + b_1\,X_{V_i}(t_k-1) + e_i(t_k) \end{aligned} \tag{4.21}$$

For sampling period of $h = 0.005$ $[sec]$ the parameters of the A and B polynomials are: $a1 = -2.9326$, $a2 = 2.8652$, $a3 = -0.9326$, $b1 = 0.000000554$, $b2 = 0.000002177$, $b3 = 0.0000005348$. Where $e(t)$ is the unmeasured disturbance term which includes two components, the first, dynamic coupling between the links and gravitation effects, and the second an uncorrelated random noise sequence if exists.

4.3.3 Simulation Study and Results

Subsection 4.3.2 developed the open loop analysis which was used to determine the order of the input/output model for the GPC algorithm, and the initial values for identification of its parameters. The actual system is a nonlinear system resulting from coupling terms due to relative motion of the links, the gravity term, saturation limits on variables, the hydraulic system etc. The nonlinearities were incorporated into the simulation model while the model estimated and used by GPC is linear by its nature. Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7 present the controlled nonlinear system, (the two link manipulator actuated by hydraulic actuators) which exhibit good performance in output tracking of the given set points. The nonlinearities are treated as unknown deterministic disturbances in that the GPC assumes a linear model for the actual system.

Figure 4.4 and Figure 4.5 show the behavior of the outputs θ_1, and θ_2 and their derivatives to square wave setpoints. Figure 4.6 presents the spool valves displacements and the control action $\Delta u(t)$ for both links and actuators. Figure 4.7 presents the hose pressures for both actuators.

The design parameters used for tuning the GPC are noted at this point. N_2, the maximum output horizon, is changed on line. N_1, the minimum output horizon, N_u, the control horizon and λ, the weighting factor are additional design parameters. At first a lower value of N_2 ($N_{2_{v_{i_1}}}$)was used, to achieve a faster transient response and later it was increased to $N_{2_{v_{i_2}}}$ to avoid overshoots. In the case presented in Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7

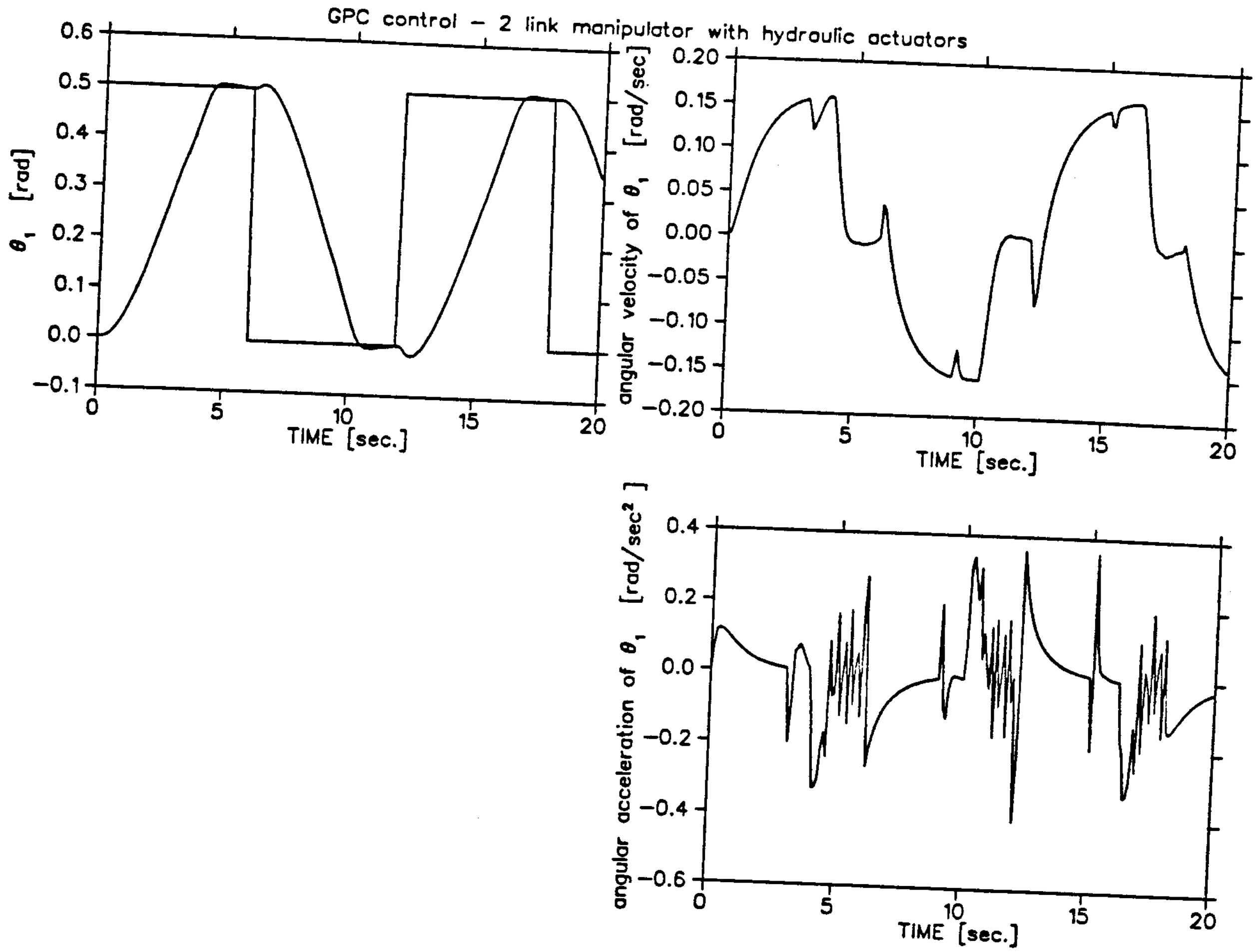

Figure 4.4: θ_1,$\dot{\theta}_1$,$\ddot{\theta}_1$ for square wave input

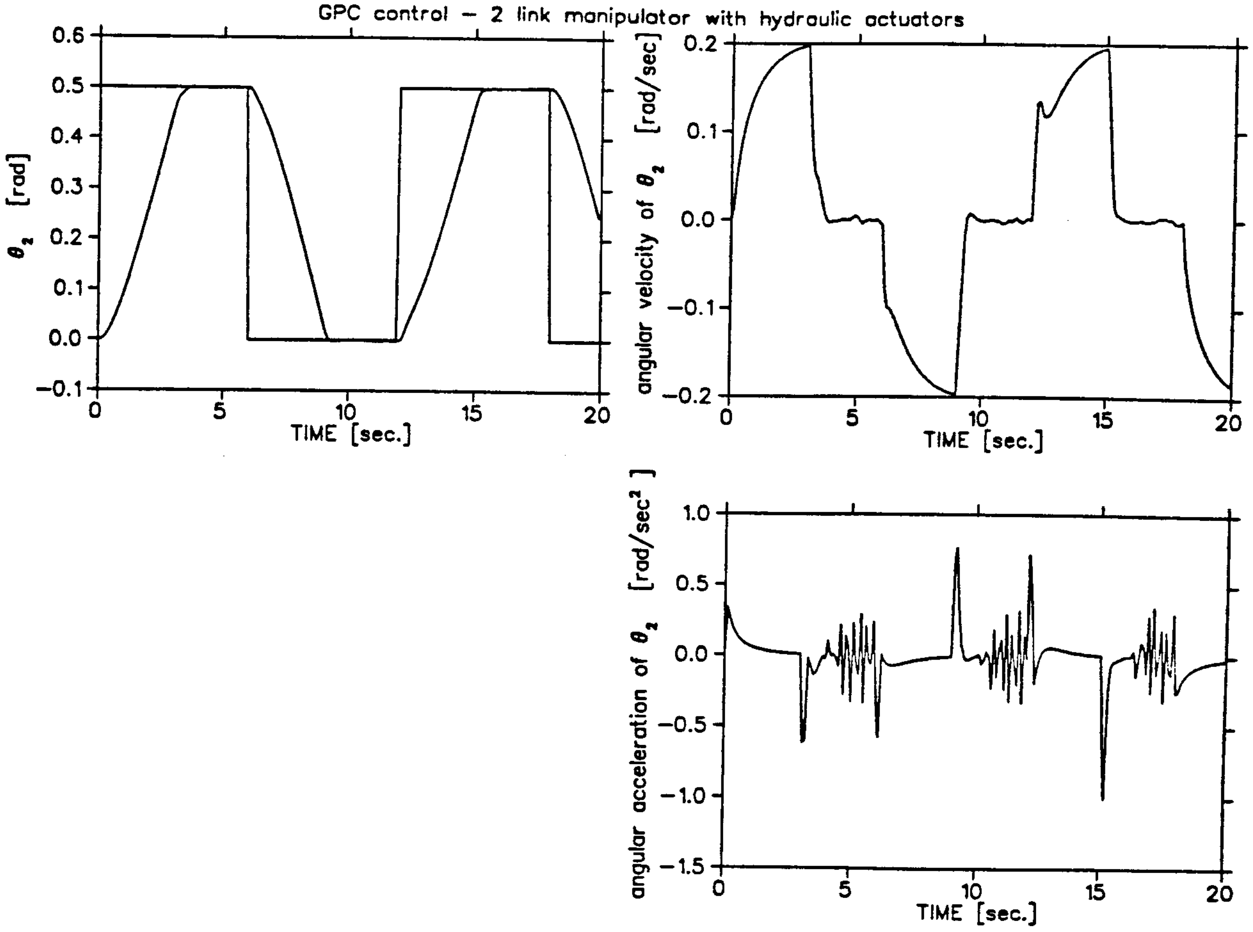

Figure 4.5: θ_2,$\dot{\theta}_2$,$\ddot{\theta}_2$ for square wave input

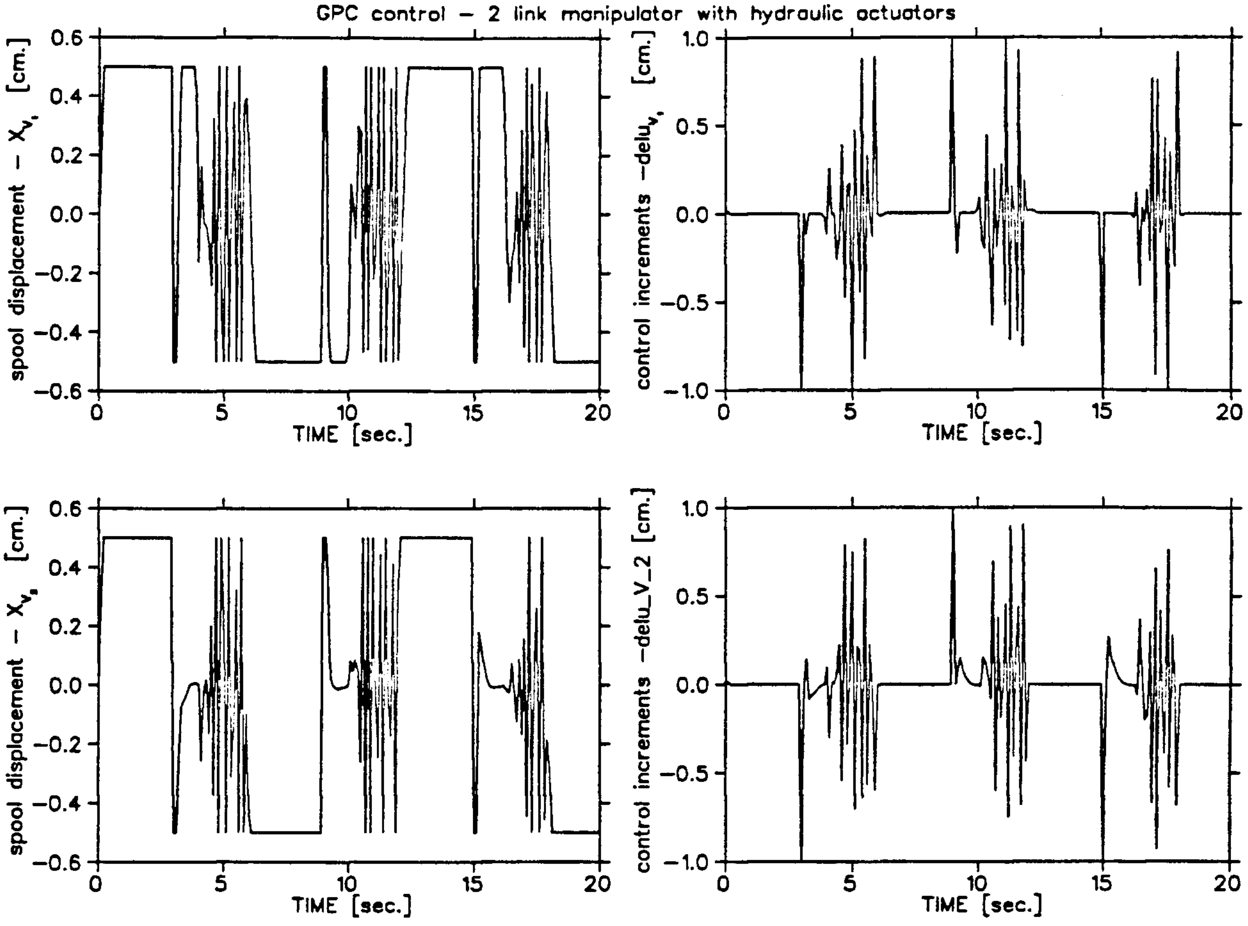

Figure 4.6: Control action and spool displacement for θ_1 and θ_2

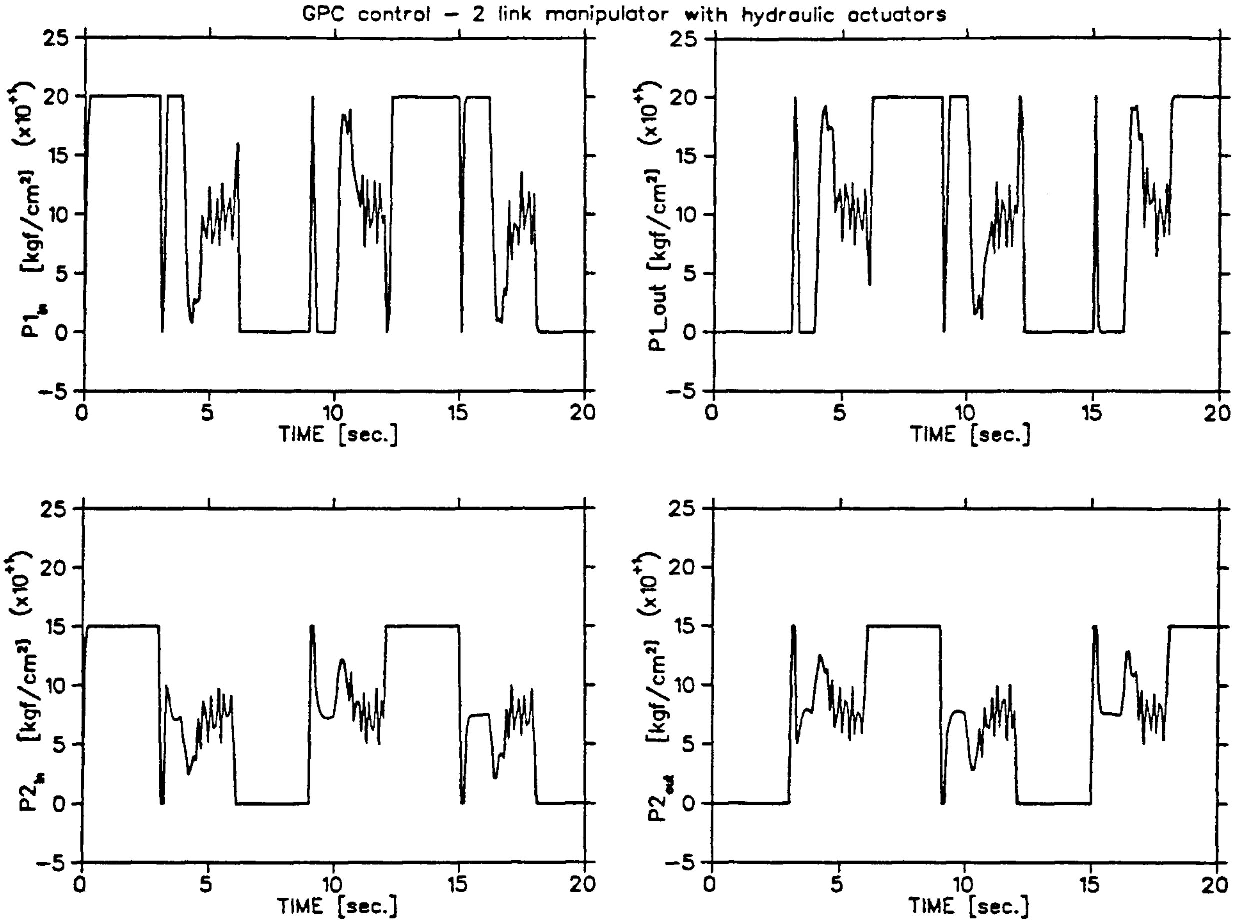

Figure 4.7: Pressures for θ_1 and θ_2

the system's performance was achieved with the following design parameters:

1. for θ_1 :

   $N_{1_{v_1}} = 1$, $N_{2_{v_{1_1}}} = 70$, $N_{2_{v_{1_2}}} = 100$, $N_{u_{v_1}} = 1$, $\lambda_{v_1} = 0.05$

2. for θ_2 :

   $N_{1_{v_2}} = 1$, $N_{2_{v_{2_1}}} = 40$, $N_{2_{v_{2_2}}} = 60$, $N_{u_{v_2}} = 1$, $\lambda_{v_2} = 0.05$

Note that in steady state there is a chatter in some of the parameters, such as the spool valve displacement, the control increment signal, the hose pressures and the accelerations. In steady state the spool valve chatters around the zero value which it cannot maintain due to nonlinearities in the system and the mis-match between the nonlinear model that simulates the actual system and the representing linear model used by the adaptive algorithm. Figure 4.8 and Figure 4.9 show the results for the same design parameters as Figure 4.6 and Figure 4.7, only in this case the model of the hydraulic actuator system was linearized. The nonlinearities due to coupling between the movement of the links or due to saturation in the displacement of the spool valve remain in the simulated model of the system. The chattering has been reduced significantly.

Changes in the values of the design parameters above will change the behavior of the system. The effects of the output horizon N_2 were checked in both loops. The larger the value of N_2, the slower the response. Larger values have the tendency to stabilize the system since it uses predicted errors over a larger period of time. On the other hand fewer prediction steps will result in a more rapid control action, and the inherent integration term of the model used for GPC causes the response to have more overshoot and more oscillations. θ_1 in the upper part of Figure 4.10 is the same as the one in Figure 4.4 but the setpoint is constant instead of a square wave. The lower part shows θ_1 in response to the same setpoint, but the output horizon is constant and its values are the higher values for $N_{2_{v_{1_2}}} = 100$ and $N_{2_{v_{2_2}}} = 70$ used in Figure 4.4. The response for the larger values is slower.

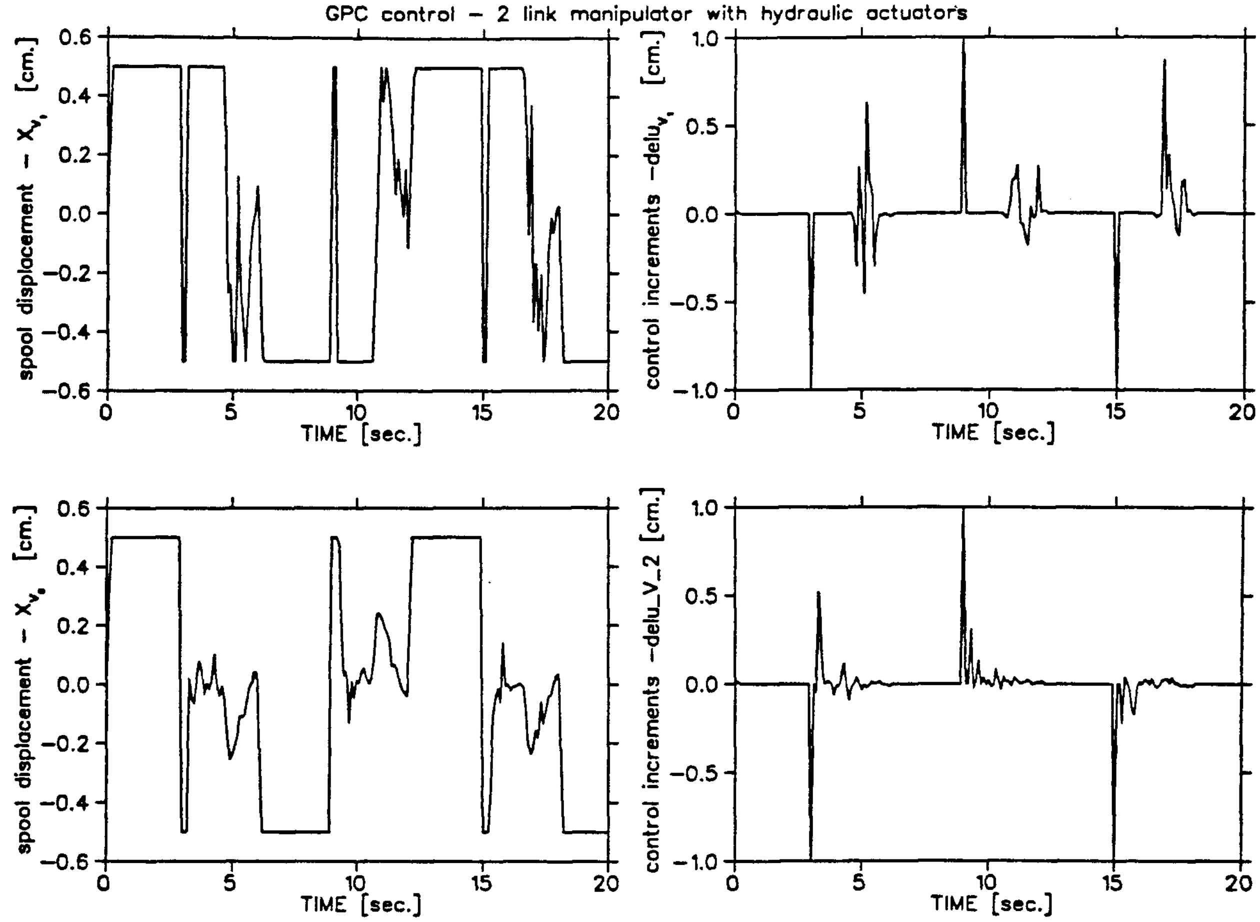

Figure 4.8: Control action and spool displacement for hydraulic linearized model.

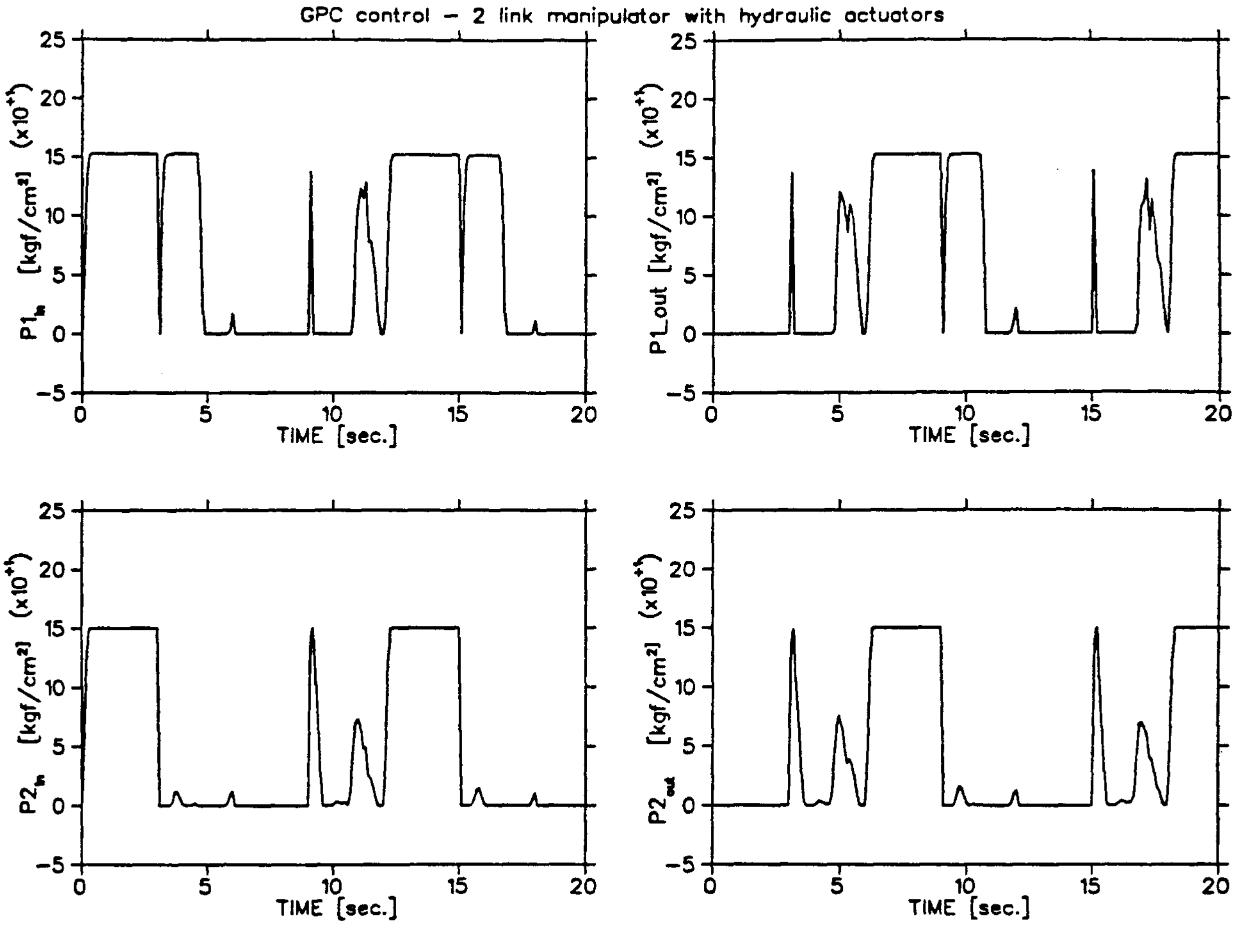

Figure 4.9: Pressures for θ_1 and θ_2 for hydraulic linearized model.

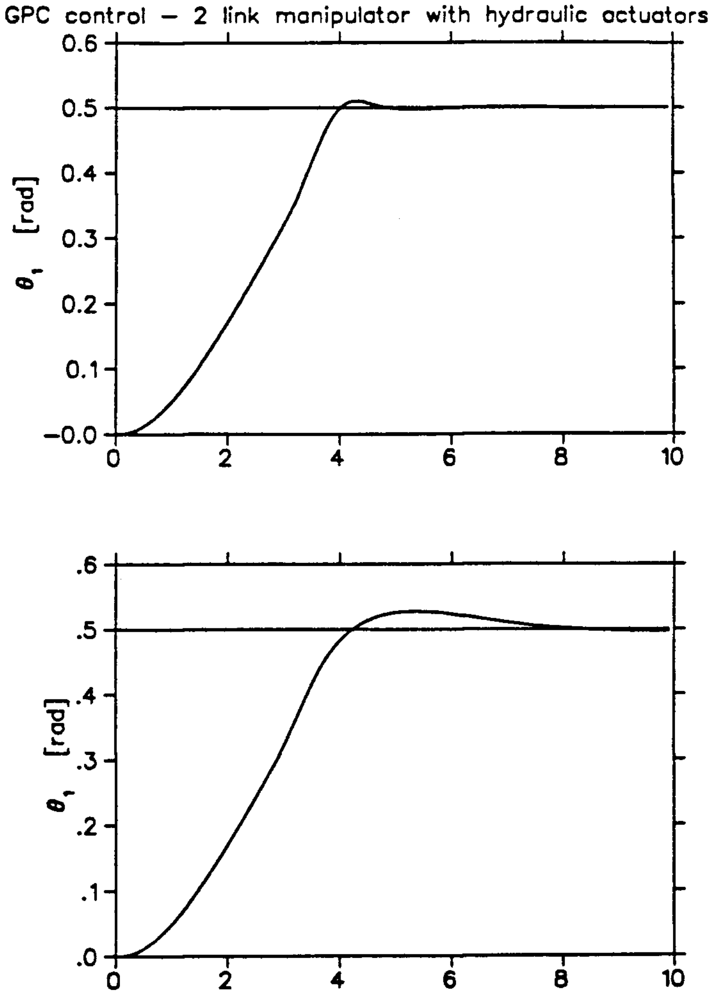

Figure 4.10: The effect of higher values of N_2 (lower case)

PAGINATION ERROR.

TEXT COMPLETE.

ERREUR DE PAGINATION.

LE TEXTE EST COMPLET.

NATIONAL LIBRARY OF CANADA.

CANADIAN THESES SERVICE.

BIBLIOTHEQUE NATIONALE DU CANADA.

SERVICE DES THESES CANADIENNES.

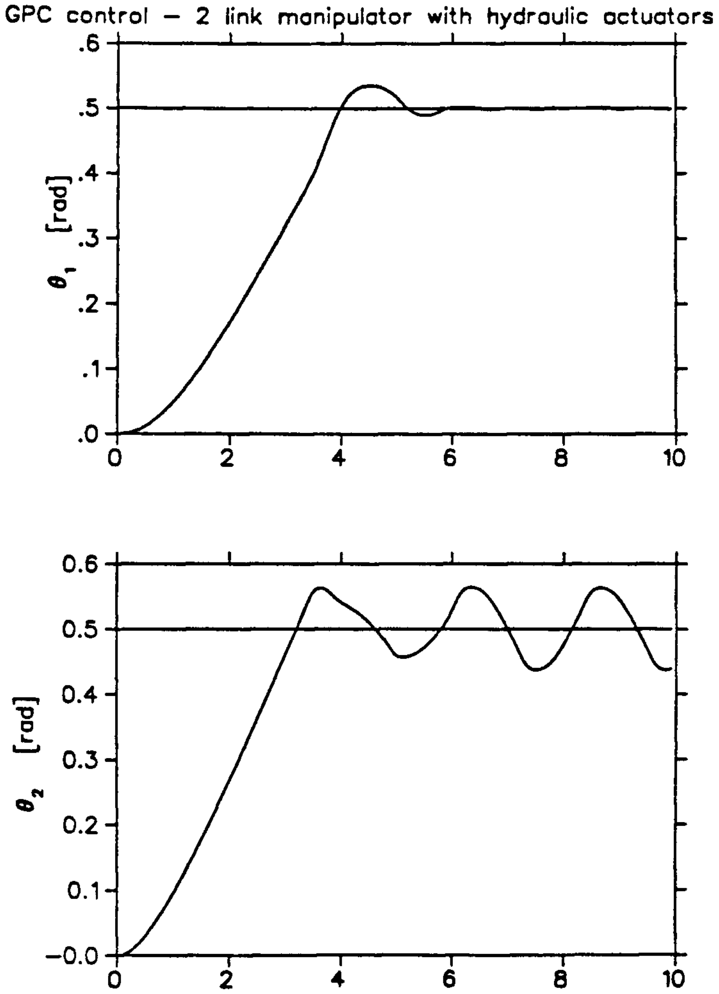

Figure 4.11: θ_1 and θ_2 for $N_{2_{V_1}} = 50$ and $N_{2_{V_2}} = 20$

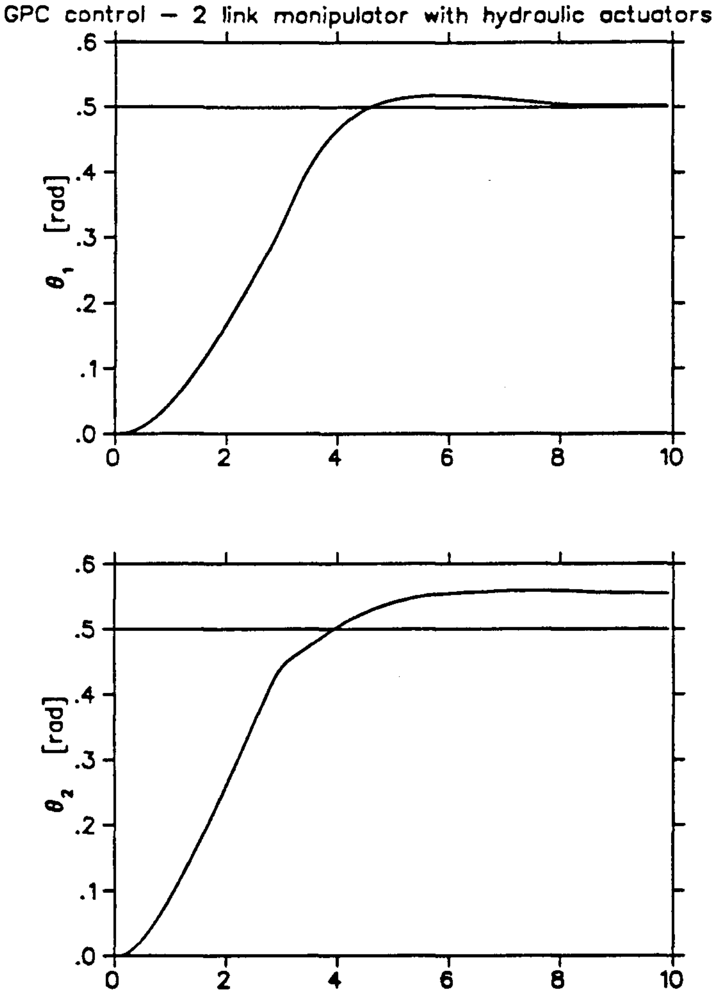

Figure 4.12: θ_1 and θ_2 for $N_{2_{V_1}} = 200$ and $N_{2_{V_2}} = 200$

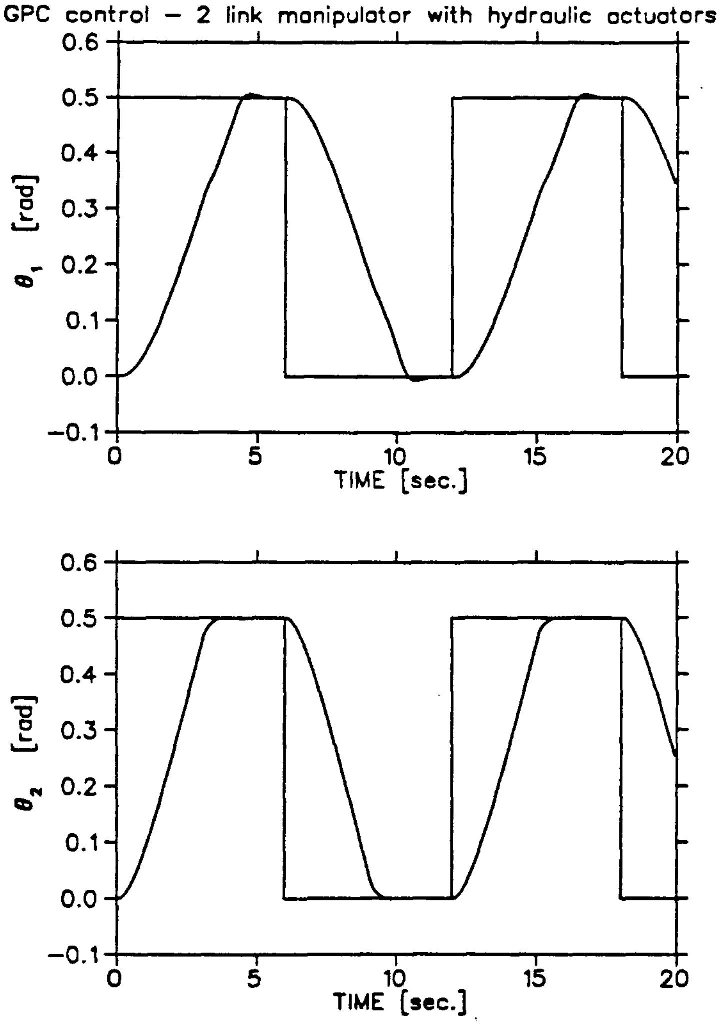

Figure 4.13: θ_1 and θ_2 for $N_{u_{V_1}} = 3$ and $N_{u_{V_2}} = 3$

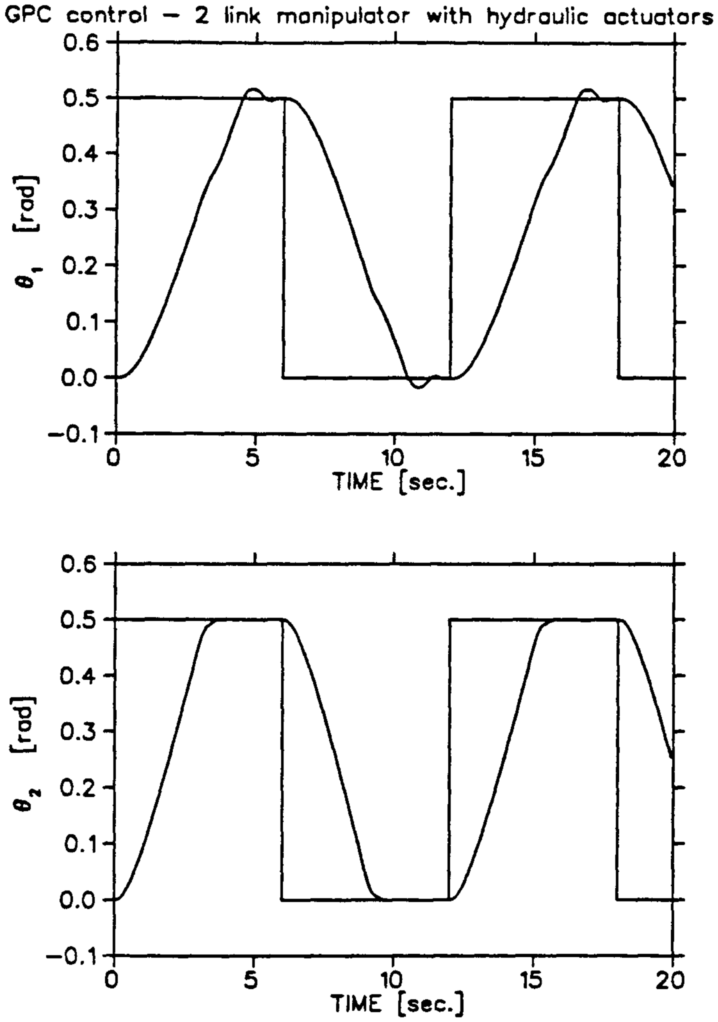

Figure 4.14: θ_1 and θ_2 for $N_{u_{V_1}} = 10$ and $N_{u_{V_2}} = 10$

4.3.4 Effects of On-line Changes in Model Order

An analysis of the effect of under and over modeling for the two link hydraulic manipulator is given in Chapter 5. The analysis is based on the behavior of a cost function for the cases of correct, under and over-modeling.

4.4 Conclusions

In this chapter a rigid two link manipulator with hydraulic actuators controlled by a GPC algorithm is presented. The system is highly nonlinear, controlled with a GPC algorithm that assumes a linear structured input/output model, whose parameters are estimated online. The control strategy treats the control of each link and its actuator as one model. The changes in the system parameters are handled without the need to identify the exact cause of the change. Such an approach may need safety measures and bounded values on some of the system variables. The system is well behaved when controlled with the set of design parameters shown in Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7. GPC overcomes the effects of the nonlinearities in the actual model of the system, (coupling between links, saturations etc.)

In terms of advancing the state of the art of predictive control, the influence of the main GPC design parameters, N_2 and N_u, was studied. It was recommended, by Clarke et al, that the control horizon N_u, should be chosen as high as the number of poorly damped poles of the system. In this study there was no significant effect on the output by choosing it so, (Figure 4.13). When N_u is chosen as high as 10 the outputs have a more oscillatory nature. As well, the maximum output horizon N_2 is shown to have a stabilizing effect if the prediction margin is wide enough. The larger the value N_2, the slower and more damped the response. The output horizon was found to play a role in reducing the inherent overshoot of the GPC algorithm (Figure 4.10 compared to Figure 4.11).

In this work, its value was changed on line and resulted in a relatively quicker response at first and significantly reduced overshoot (if at all) as shown in Figure 4.4 and Figure 4.5 compared to Figure 4.10. The sluggish response to higher values of N_2 and the oscillatory nature of it for lower values was shown in Figure 4.12 compared to Figure 4.11.

Many different industrial manipulators in use today are represented by 2 link hydraulically actuated mechanisms of the type in this work. This work studies the behavior of this category of manipulator when controlled by an adaptive control algorithm. The study can be expanded to a manipulator with additional links. Hydraulically actuated machines are highly nonlinear systems and their parameters may vary online during a working cycle. When controlled by GPC a good performance of the output in tracking a sequence of set points was achieved.

Thus in addition to advancing the state of the art in certain areas of predictive control related to design parameters, the work described in this chapter has also examined some complex considerations such as the effect of hard nonlinearities in the application of GPC to a broad category of hydraulically actuated manipulators.

Chapter 5

MODEL ORDER DETERMINATION

5.1 Introduction

Adaptive control algorithms are designed assuming that the plant model is defined by a fixed structure. A question asked in this work is how will an adaptive control algorithm (the GPC in this case) behave when the true plant is not perfectly described by any model of a given class.

The behavior of a specific algorithm is understood through analyzing stability and performance. Stability proofs usually require restrictive assumptions,for example, assumption on the structure of the model (number of poles, number of zeros, time delay, etc.). In many research studies, stability proofs for adaptive controlled systems are done for examples which deal primarily with linear systems, and the signals are bounded with small perturbations (for example in Astrom et al [1]). If modeling errors are sufficiently small, robust stability of adaptive systems can be achieved (Bahnasawi and Mahmoud [59]). Modeling errors (such as unmodeled or over-modeled dynamics, nonlinearities, etc.) appear as disturbances in the adaptive process.

The present research deals with changes in the structure of the plant, and therefore changes in the model order. When the model order is not accurate, the modeling error can be large and instabilities can appear. Both applications used in this work are analyzed with modeling errors.

The GPC uses a linearized model of the system for control purposes. Any diversion from the linear form is as a disturbance to the adaptation mechanism. The hydraulic manipulator example is a highly nonlinear system which is represented by a linear model for the GPC. The modeling errors act as disturbances caused partly by the nonlinearities. Chapter 4 analyzed this system and good performance is achieved. When the order of the system is changed, for example when time delay or valve dynamics are introduced, and the control algorithm has not been updated with the changes, the modeling error is large and instabilities occur (the nonlinear terms remain unchanged). The single flexible link manipulator, is modeled by a linear model that matches the order of the one for the adaptive algorithm. Good performance is shown in Chapter 3. When the order of the system does not match the one representing it for the adaptive algorithm due to vibration modes, the modeling errors are large and instabilities occur.

This chapter presents a method for detecting and correcting the model order and hence minimizing the modeling error. It evaluates a cost function and its derivative. If necessary, the represented model order for the adaptive algorithm is changed on-line to reduce modeling errors and the uncertain parameters in the model estimated as is normally done in an adaptive algorithm.

5.2 Cost Function - For Detection Of The Model Structure

The goal, in order to achieve good performance, is to reduce modeling errors. As a measure of the modeling error, we choose a cost function which is the square of the difference between the output measured from the actual system and the one of the linear model as used by the adaptive control algorithm i.e.

$$J(\mathbf{y}_{meas}, \mathbf{y}_{est}) = \sum_{k=1}^{t} [\mathbf{y}_{meas} - \mathbf{y}_{est}]^2 \tag{5.1}$$

A cost function is usually chosen as a criterion to be minimized according to the final target. For the estimation problem, for example, the process output and the estimated model output are compared and some optimal adjustment between the two should be found. The optimum is defined using a criterion with respect to output signals or to the expected error of the estimated parameters values.

For control purposes the criterion is optimized in order to achieve a desired control law. The criterion can be a quadratic form which, for example, could be a function of the state vector and the input signal (see Astrom and Wittenmark [15], Ljung [6] and Eykhoff [60]). In Chapter 3 the behavior of the cost function of the GPC algorithm (Equation 2.7) is compared with the one presented in Equation 5.1 and was found to be similar.

The behavior of the cost function in Equation 5.1 was studied with open loop and closed loop control in order to determine a method of detecting an on-line change in the actual system's model and of changing the model for the control algorithm accordingly. The cost function behavior was studied in two ways, first as a function of order changes (Figures 5.1 and 5.2) i.e. the value of the cost function was recorded at a certain time as a function of different estimated model orders (the plant's model remained unchanged) and second, as a function of time. In each run the plant model and order of the estimated one remain unchanged for the desired period of time.

To examine the open loop behavior of the cost function as a function of the order, the estimated model order was changed while the actual system structure remained unchanged. For example, for the flexible link (as described in Chapter 3), the actual system had two vibration modes (plant model order of 6), but the estimated model order was changed (from 2 which is a rigid body, to 10 which is four vibration modes). Figure 5.1 shows the cost function behavior as a function of the model order. The values at each order are relatively small, and it is hard to differentiate between them. On the other hand, Figure 5.2 shows the behavior of the same cost function with the same model order changes to a closed loop situation, with the GPC algorithm. This time, the results for each chosen order of the estimated model differ

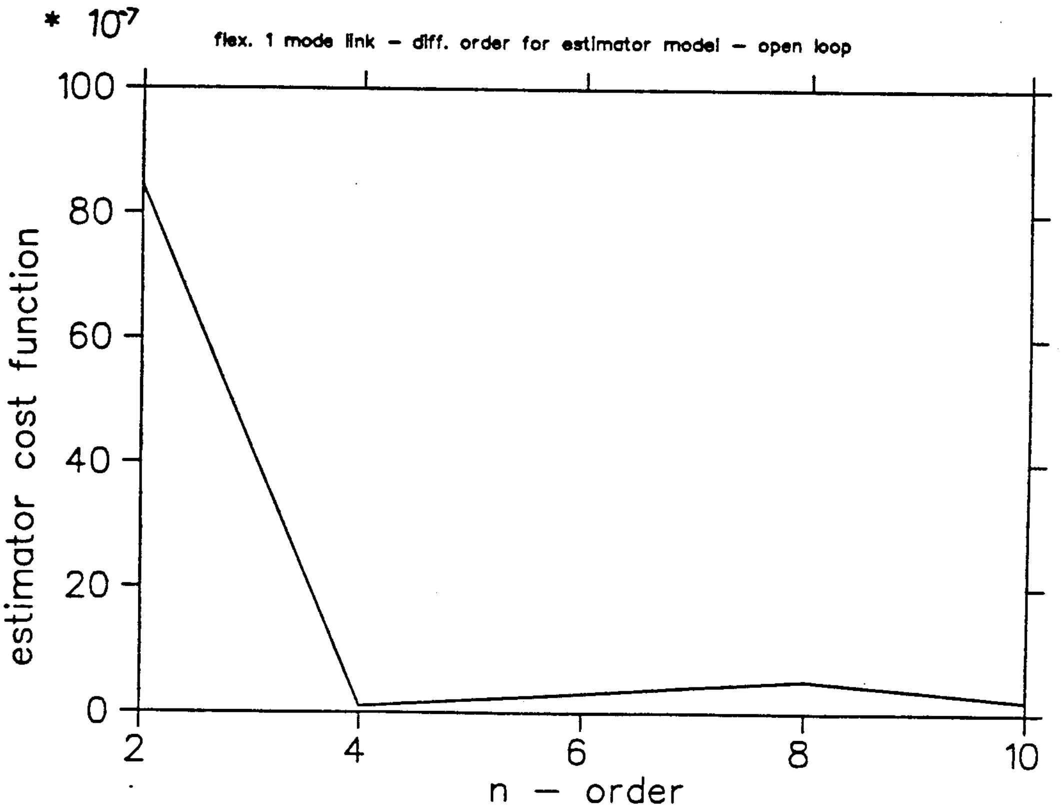

Figure 5.1: Cost function behavior for open loop flexible link

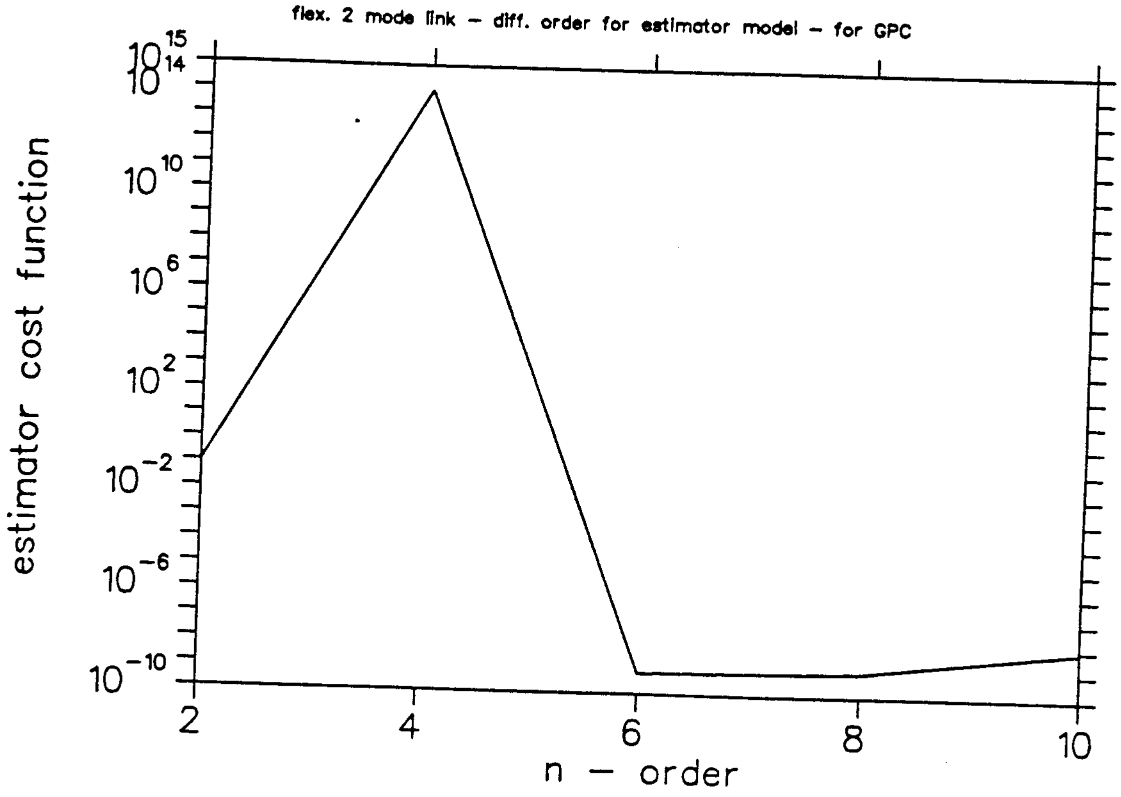

Figure 5.2: Cost function behavior for closed loop flexible link

extensively, and the cost function can indicate the size of the modeling error.

Appendix A shows the results of experimental data obtained for the identification of the dynamic model of a Caterpillar 215B excavator, which is a two link manipulator actuated by hydraulic actuators. The identification done was an open loop one in which no adaptive control algorithm was introduced. The results in Appendix A show that after the cost function becomes flat at high orders it is easy to mis-choose the order for the system's model. (see Figures A.8 and A.9)

Observing the time behavior of the cost function reveals three parameters that can be used for detecting mis-modeling. The parameters are: the Rise Slope $\dot{J}_R$, the Zero Slope $\dot{J}_Z$, and Change Time T_c. Figure 5.3 also shows a behavior format of the cost function (C.F.) (which are backed up, later in the work, with figures showing the actual behavior of the C.F.) for several configurations, for the two applications, where the three parameters are shown. In all of the configurations shown in Figure 5.3 the model of the actual system matches the order of the estimated model by the GPC. One and two mode flexible links, and a linear and nonlinear model of the hydraulically actuated manipulator all have the same nature of behavior, i.e. all three parameters, for linearized models of the systems, have a similar behavior. The Rise Slope $\dot{J}_R$ has the order of magnitude of 10^{-9} [deg/sec or cm/sec depending on the system), the Change Time T_c, from 1.5 [sec.] to 5.4 [sec.] depending on the system's nature and the control parameters. The Zero Slope stabilizes on different values with an order of magnitude of 10^{-9} [deg. or cm.].

The values of the parameters do not necessarily have the same order of magnitude, as it all depends on the nature of the system being controlled. However the nature of the behavior is the same. This is important, since regions can be defined for each of the parameters for a specific system so that detection of mis-modeling can be achieved. It will be shown below that the behavior of the C.F. is different when a mis-modeling occurs. Figure 5.3 also shows that the behavior of the C.F. for a nonlinearized model of the hydraulic actuated manipulator.

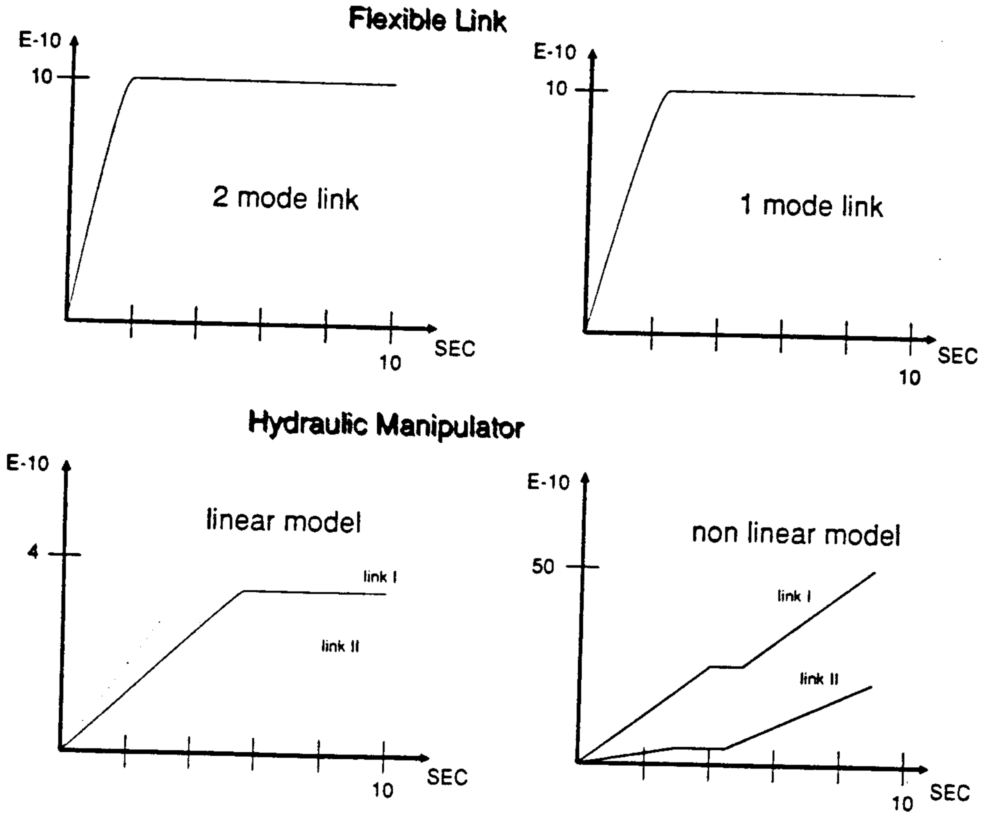

Figure 5.3: Schematic description of the C.F. behavior for the different applications

When the system is under or over-modeled, the behavior of the cost function changes extensively in some cases, and moderately in others. Since those changes can be detected, the assumptions about the structure of the estimated model will be updated and the results improved.

5.2.1 The Cost Function for the Flexible Link Manipulator

Cost Function of the Flexible Two Mode Link

When a flexible two mode link (order 6) is estimated by a two mode link, the modeling error is small and good control is achieved, as mentioned in Chapter 3. Figure 5.4 shows the behavior of the cost function, which has low values in the order of magnitude of 10^{-10}, the Change Time is $T_c = 1.5[sec.]$ to a zero slope, which means that for $t > T_c$, the error has very small values (Figure 5.5).

Mis-modeling can be classified into two categories: under-modeling and over-modeling. When this system is under-modeled with a one mode estimated model (order 4), instabilities occur, since the control algorithm does not account for the unmodeled dynamics and can not overcome it as a disturbance. Figure 5.6 presents the behavior of the cost function in this case, showing that its values rise very high even before the Change Time (1.5 sec.); thus the mis-modeling can be detected and changed on-line. Figure 5.7 produces very similar results for under-modeling of the estimated model with the order of 2.

When the system is over-modeled, the reaction to the mis-modeling is more moderate. When the estimated model is a 3 mode one (order 8), the cost function and the other parameters behave as if there is no mis-modeling (Figure 5.8). When a 4 mode model (order 10) is introduced, the modeling error is larger, and the cost function value rises beyond the desired value (order of magnitude goes to 10^{-8} instead of 10^{-10}); after T_c it keeps on rising and does not achieve the zero slope. (Figure 5.9).

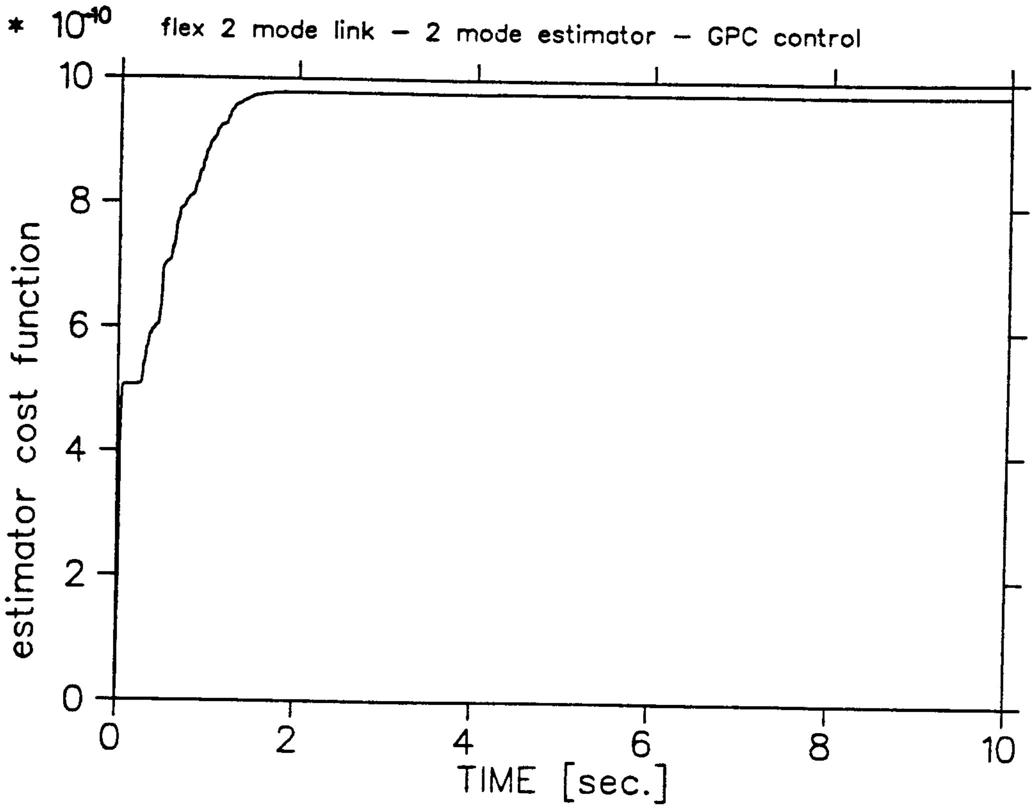

Figure 5.4: Cost function behavior for 2 mode Link and 2 mode estimated model

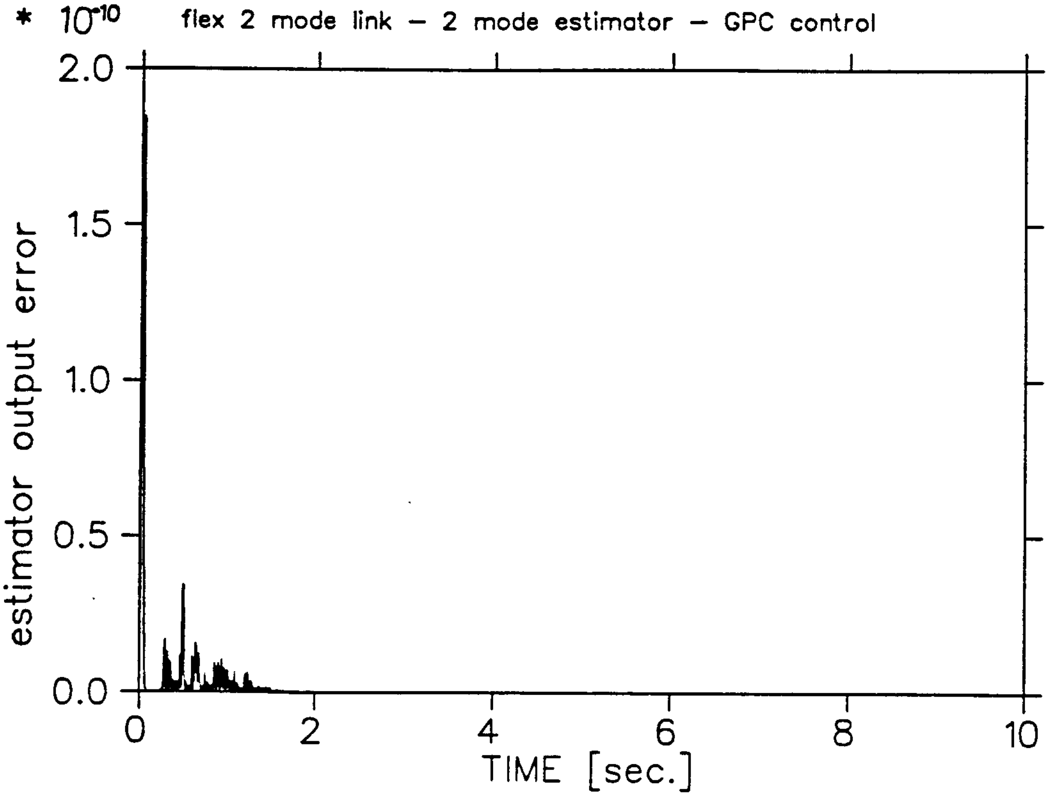

Figure 5.5: Output error behavior for 2 mode link and 2 mode estimated model

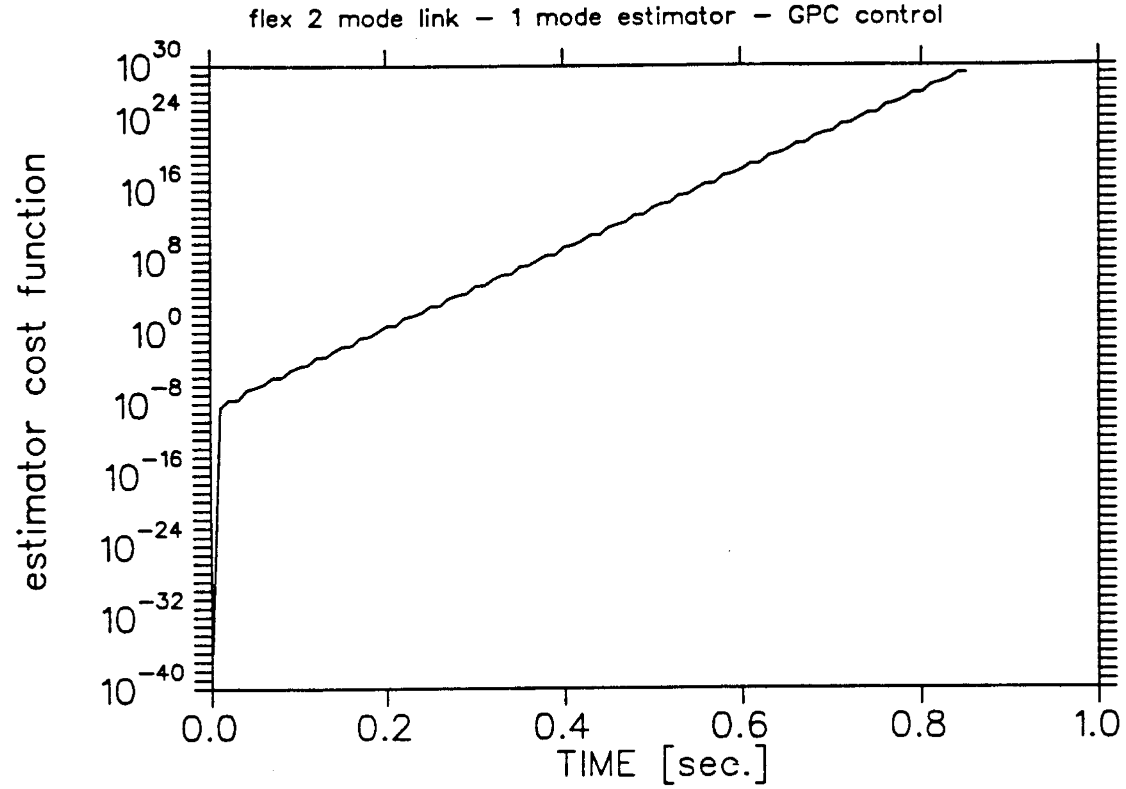

Figure 5.6: Cost function behavior for 2 mode link and 1 mode estimated model with logarithmic axis

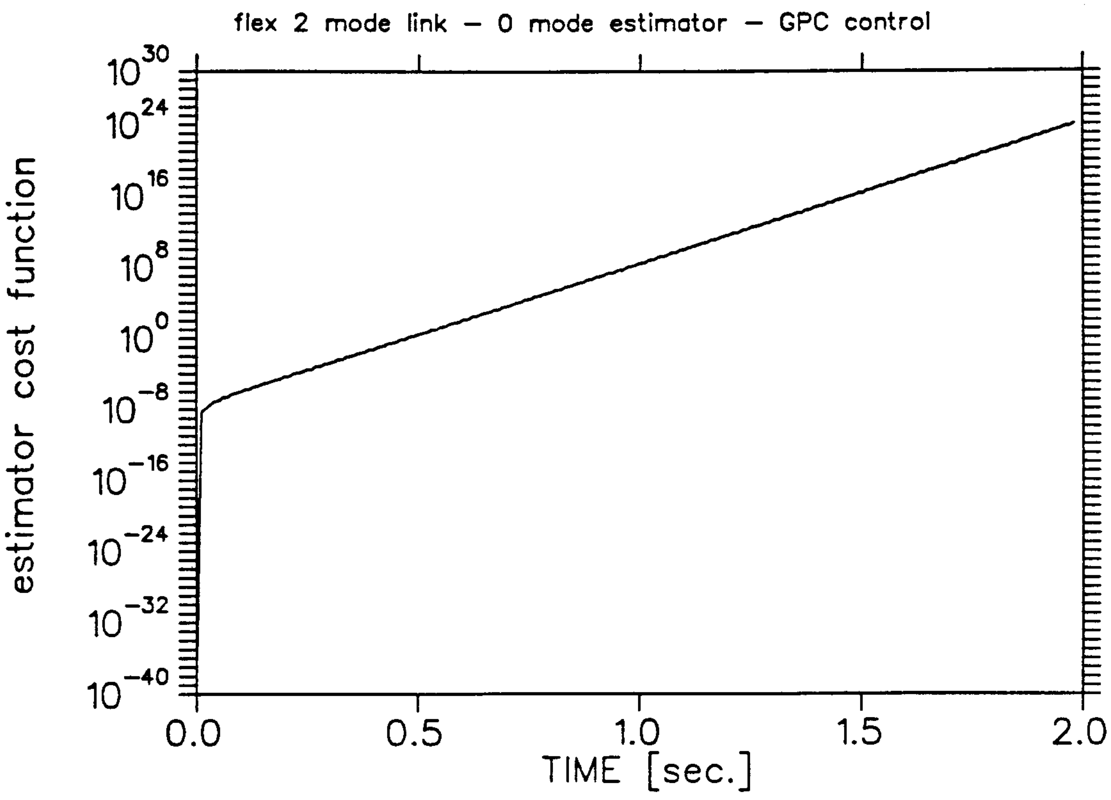

Figure 5.7: Cost function behavior for 2 mode link and 0 mode estimated model with logarithmic axis

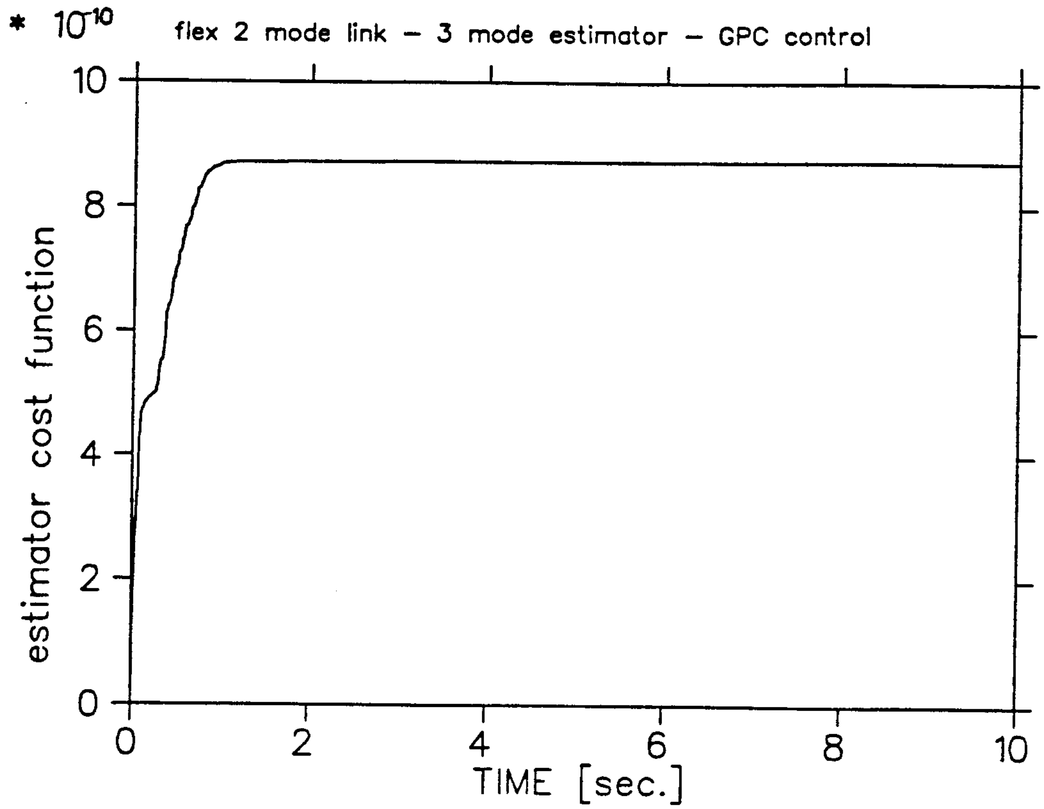

Figure 5.8: Cost function behavior for 2 mode link and 3 mode estimated model

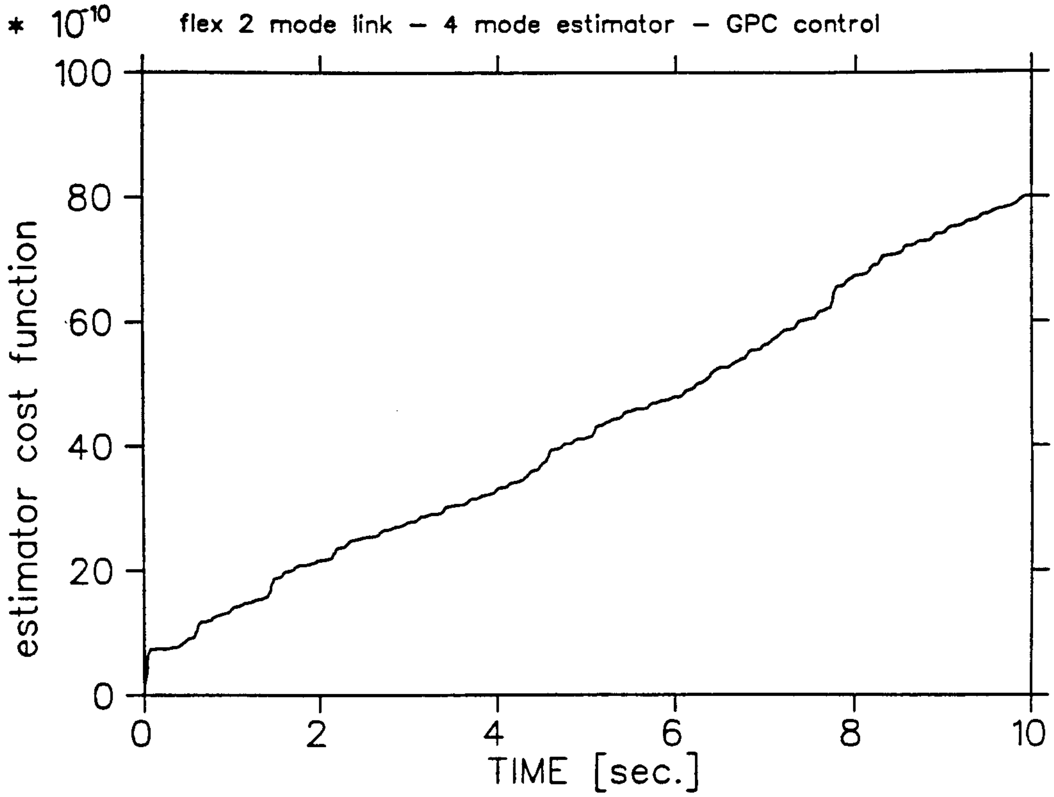

Figure 5.9: Cost function behavior for 2 mode link and 4 mode estimated model

The conclusion so far is that unmodeled dynamics affect the performance and stability of the system faster and in a more intensive manner than the over- modeling does. This fact will help the on-line model structure detection differentiate between under and over-modeling. These conclusions will be backed up by further results.

Flexible One Mode Link

The one mode link (order 4), when estimated with a one mode estimated model, produces a controlled system with good performance. Figure 5.10 shows the behavior of the cost function for these conditions. The C.F. (like the one in Figure 5.4) has a zero slope and stabilizes at the order of magnitude of 10^{-9} and $T_c = 2sec.$. The behavior of the two cost functions is the same but the values are different. Here too, the mis-modeling is addressed by the two categories: under-modeling and over-modeling. Under-modeling, as in Figure 5.11, causes the unstable response if an estimated model of a rigid body is chosen. It produces an oscillation frequency of about $14.5Hz.$, the frequency of the first mode not accounted for by the control algorithm (see Table B.1).

Figure 5.12 shows that the cost function grows, and at $T_c = 2$, its value is approximately 50, whereas in Figure 5.10 it was at the order of magnitude of 10^{-9}. This is a clear indication that the model chosen was not the right one. These results match the behavior of the 2 mode system for under-modeling as presented in Figure 5.6 and Figure 5.7.

Over-modeling has a more moderate response. A two mode estimated model (Figure 5.13) behaves like the 1 mode model, but a 3 mode estimated model (Figure 5.14) at T_c has a value of the order of magnitude of 10^{-5} where the 1 mode estimated model had the value of 10^{-9}. This case shows that the C.F. continues increasing, and the whole process becomes unstable. The rate of approaching bad performance or even instability is much slower than the one for under-modeling. A 4 mode estimator (Figure 5.15) has a similar behavior to the previous case, but the cost function value rises quicker. At T_c the C.F. has the order of magnitude

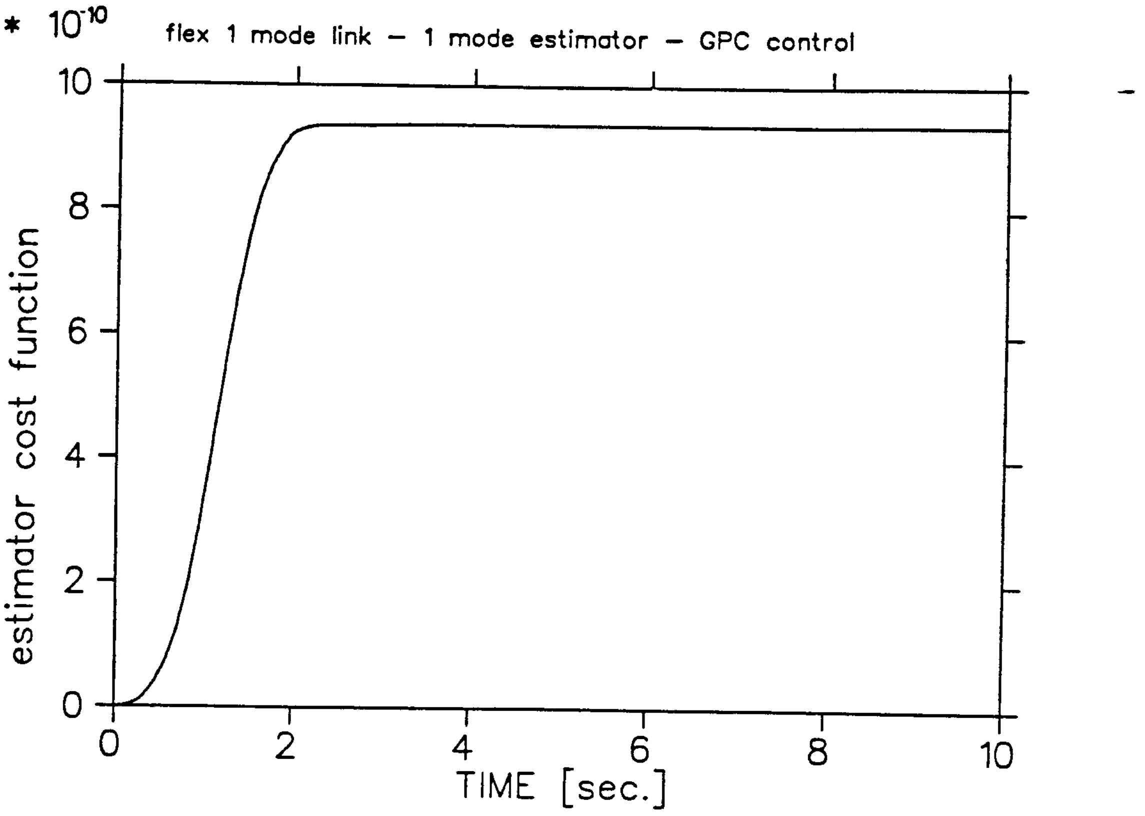

Figure 5.10: Cost function behavior for 1 mode link and 1 Mode estimated model

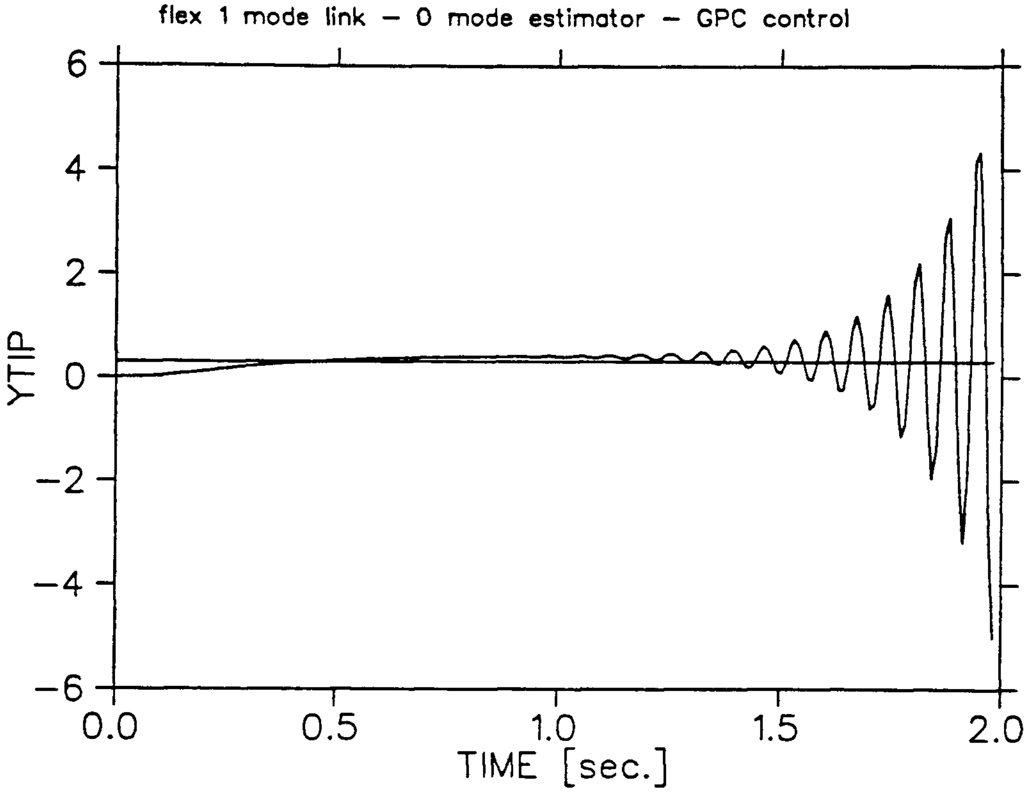

Figure 5.11: Output behavior for 1 mode link and 0 mode estimated model

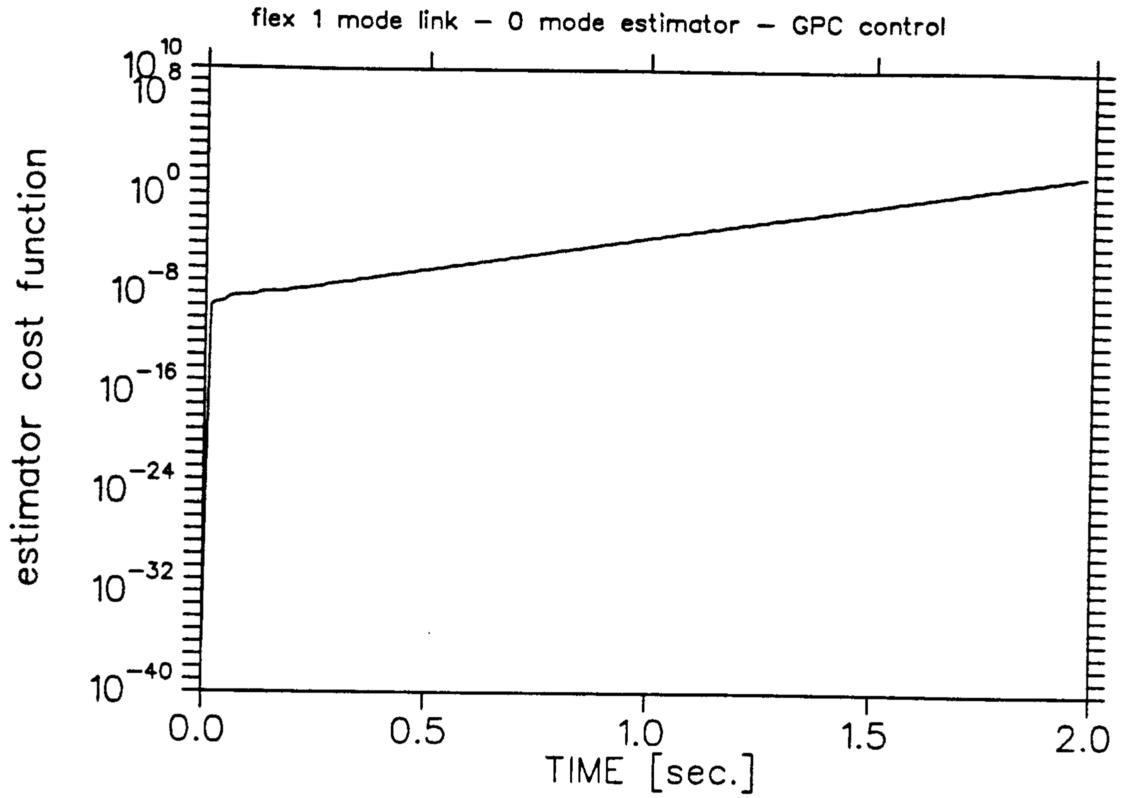

Figure 5.12: Cost function behavior for 1 mode link and 0 mode estimated model with logarithmic axis

of 10^{-4} and at $t = 5sec.$ the order of magnitude of 10 where the order of magnitude for the previous case is 10^{-1}.

The conclusion that can be drawn so far from analyzing the flexible link, is that for under, over or correct modeling, the cost function behavior is significantly different. So a model plant mis-match could be detected by examining the behavior of the cost function and its derivative.

5.2.2 The Cost Function for the Two Link Manipulator with Hydraulic Actuators

The two link manipulator is highly nonlinear, widely used in the industry, and is therefore of interest in this investigation. Chapter 4 presents a thorough discussion of such a system controlled with the GPC algorithm which achieves good performance. In this chapter, the behavior of a cost function for such a system will be studied. The hydraulically manipulated robotic link is basically a third order system, order of 2 for the dynamics of the links, and order of 1 for the hydraulic systems. The order of such a system can change if the link is not rigid, but flexible with an unknown number of vibration modes, or if the hydraulic system contains a time delay (which will add an order of one to the basic system), or if the spool valve dynamics influence the process (then an order of 2 is added to the basic process bringing it to order 5). In this discussion the order change will be in the hydraulic part, where the results will be divided into two categories. First a linearized model of the machine is introduced and the behavior of the cost function studied, and then the full nonlinear model for the system is used in the simulation and the behavior of its C.F. studied as well.

Figure 5.16 shows the behavior of θ_1 and θ_2 to a step function when the linearized model of the system and the model assumed for GPC match and are both of order 3. Figure 5.17 shows the behavior of the cost function and its derivative for both links The same pattern of behavior can be observed (as in the flexible link). The Rise Slope, the Time Change T_c,

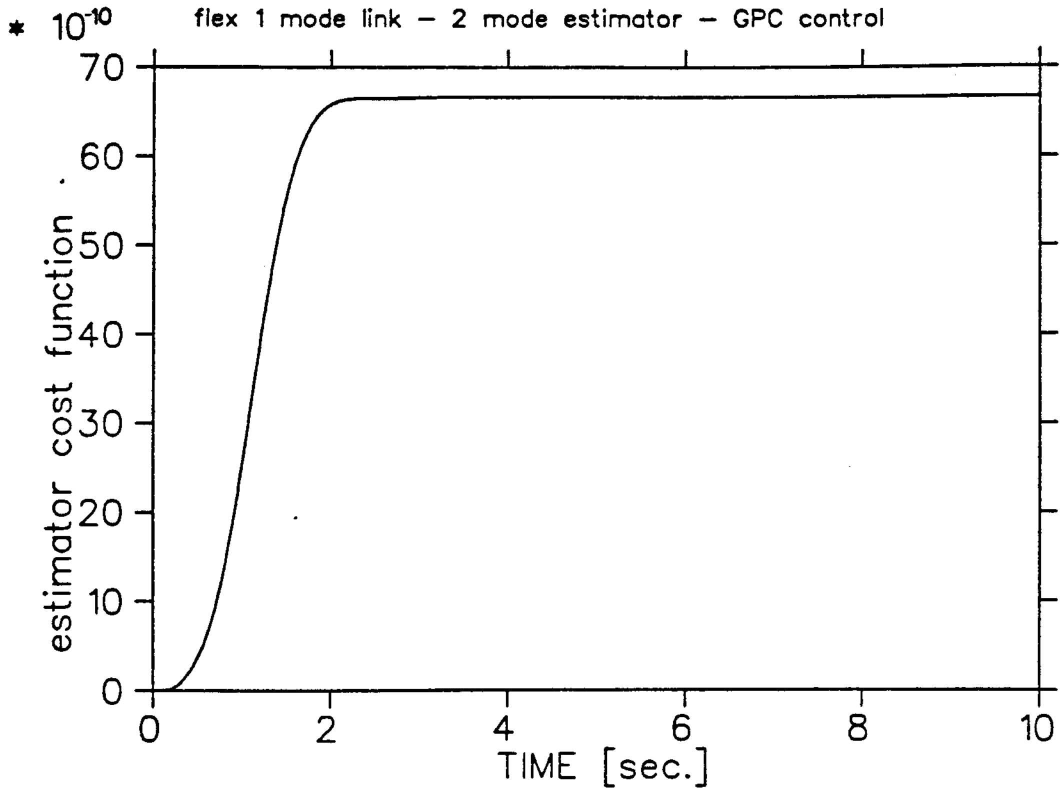

Figure 5.13: Cost function behavior for 1 mode link and 2 mode estimated model

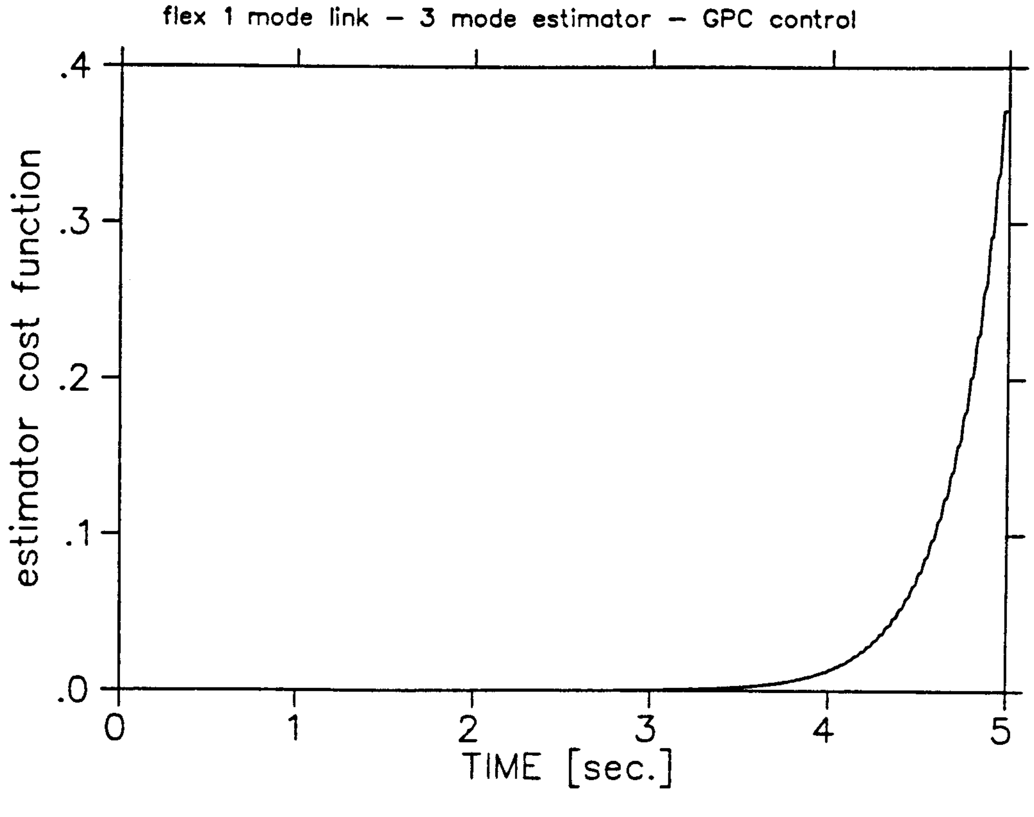

Figure 5.14: Cost function behavior for 1 mode link and 3 mode estimated model

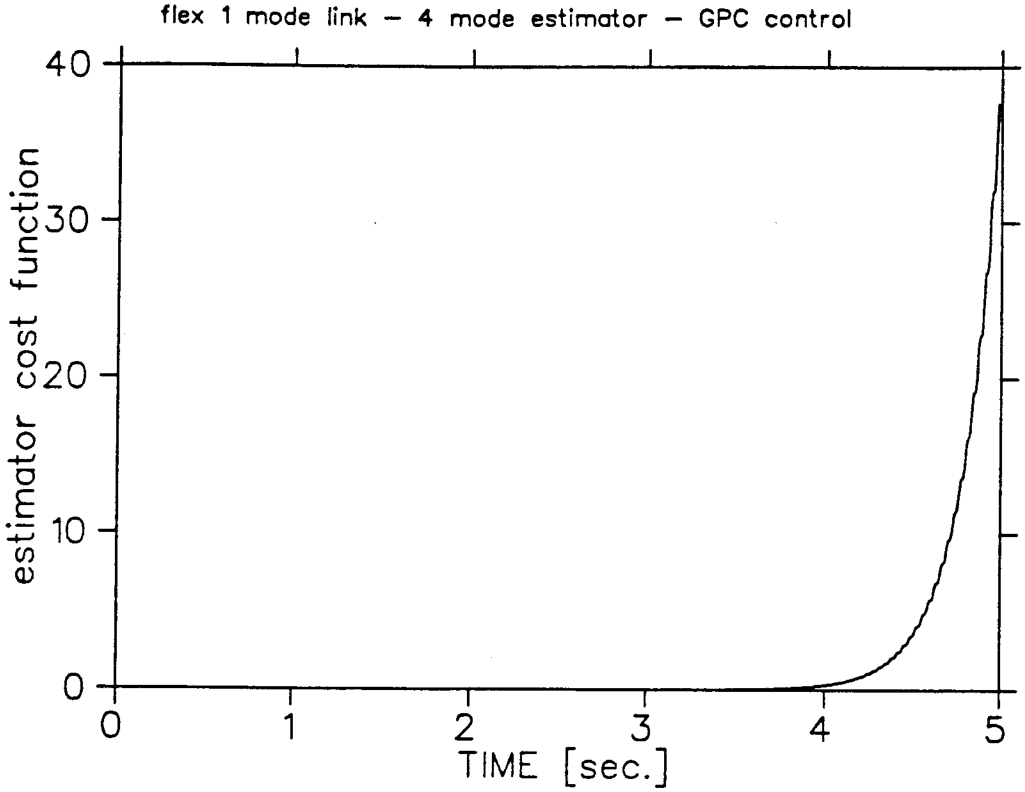

Figure 5.15: Cost function behavior for 1 mode link and 4 mode estimated model

and the Zero Slope are found in this case too (schematic description of these parameters and their values are described in Figure 5.3).

When the actual nonlinear system is introduced (the order remains 3), the behavior of the cost function changes and so does the behavior of the system. In order to maintain steady state values for some of the hydraulic parameters, the spool valve chatters around its zero value. As a result, the output error is constant (not zero), and the cost function rises constantly. Figure 5.18 shows that the C.F. behaves similarly to the behavior in Figure 5.17 up to the time when the chattering begins. It can be detected clearly on the C.F. derivative plot .

Since good performance is achieved in controlling the nonlinear system with GPC (see Chapter 4), the cost function indicates that the order of the estimated (linearized) model matches the order of the actual nonlinear machine model and can be used to detect modeling errors of the system. When the order of the estimated model does not agree with that of the actual system, it is also evident in the cost function behavior. Figure 5.19 shows an under-modeled first link, in which its estimated model was of order 2, whereas link 2 had a matching estimated model of order 3. The cost function of Link 2 has a very similar behavior to the one seen in Figure 5.18 and its output (Figure 5.20) stabilizes on its set point. However, it has a larger overshoot due to the coupling with link 1, which is under-modeled by its estimated model.

Over-modeling, as in the flexible link case, reacts in a more moderate way than the under-modeling. In Figure 5.21 both links were over modeled with an estimated model of order 4. The cost function for both links grows, indicating the mis-match between the models. The derivative however, decreases eventually. The cost function and its derivative change in a moderate manner compared with the under-modeling case.

In order to get a realistic fifth order model for the system, dynamics should be introduced to the servovalve of the system. Usually, for most practical purposes, the servovalve dynamics

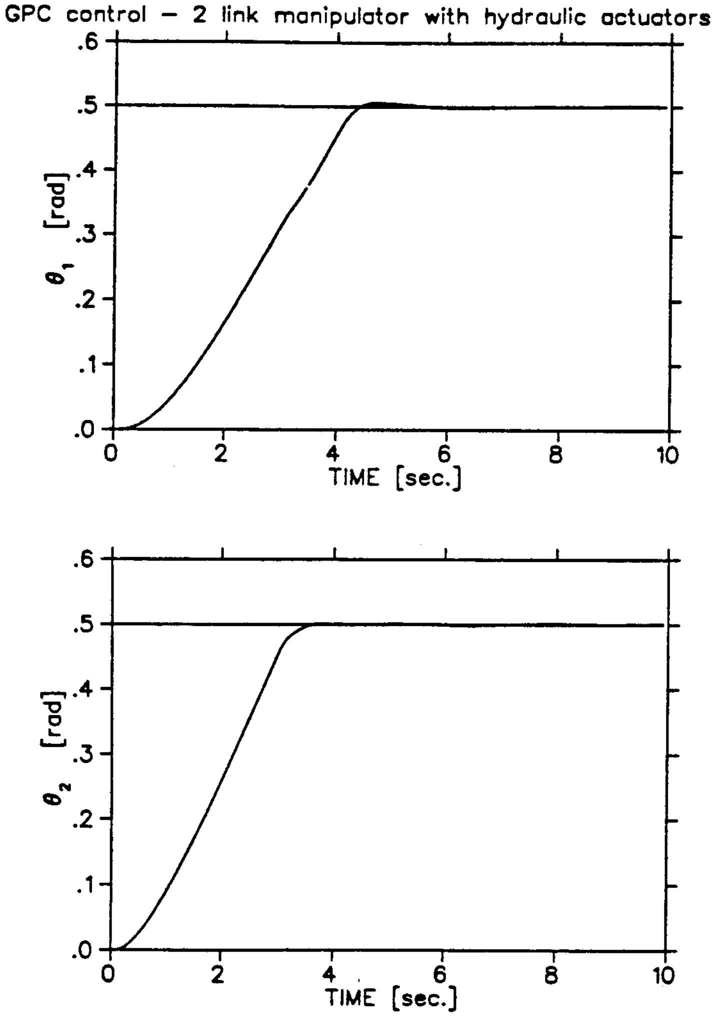

Figure 5.16: θ_1 and θ_2 behavior for 3 mode model and 3 mode estimated model

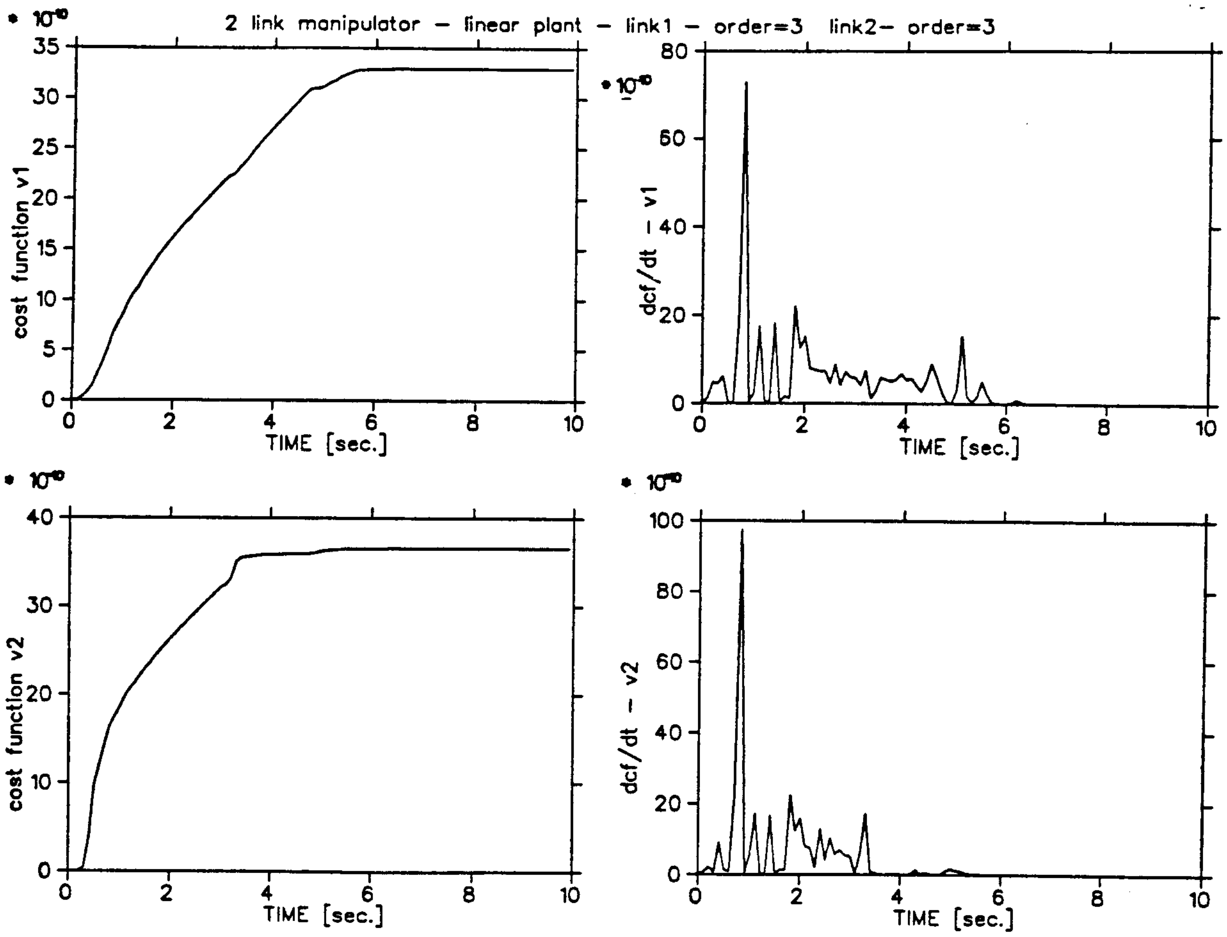

Figure 5.17: Cost function behavior for 3 mode hydraulic links (linearized plant model) and 3 mode estimated models

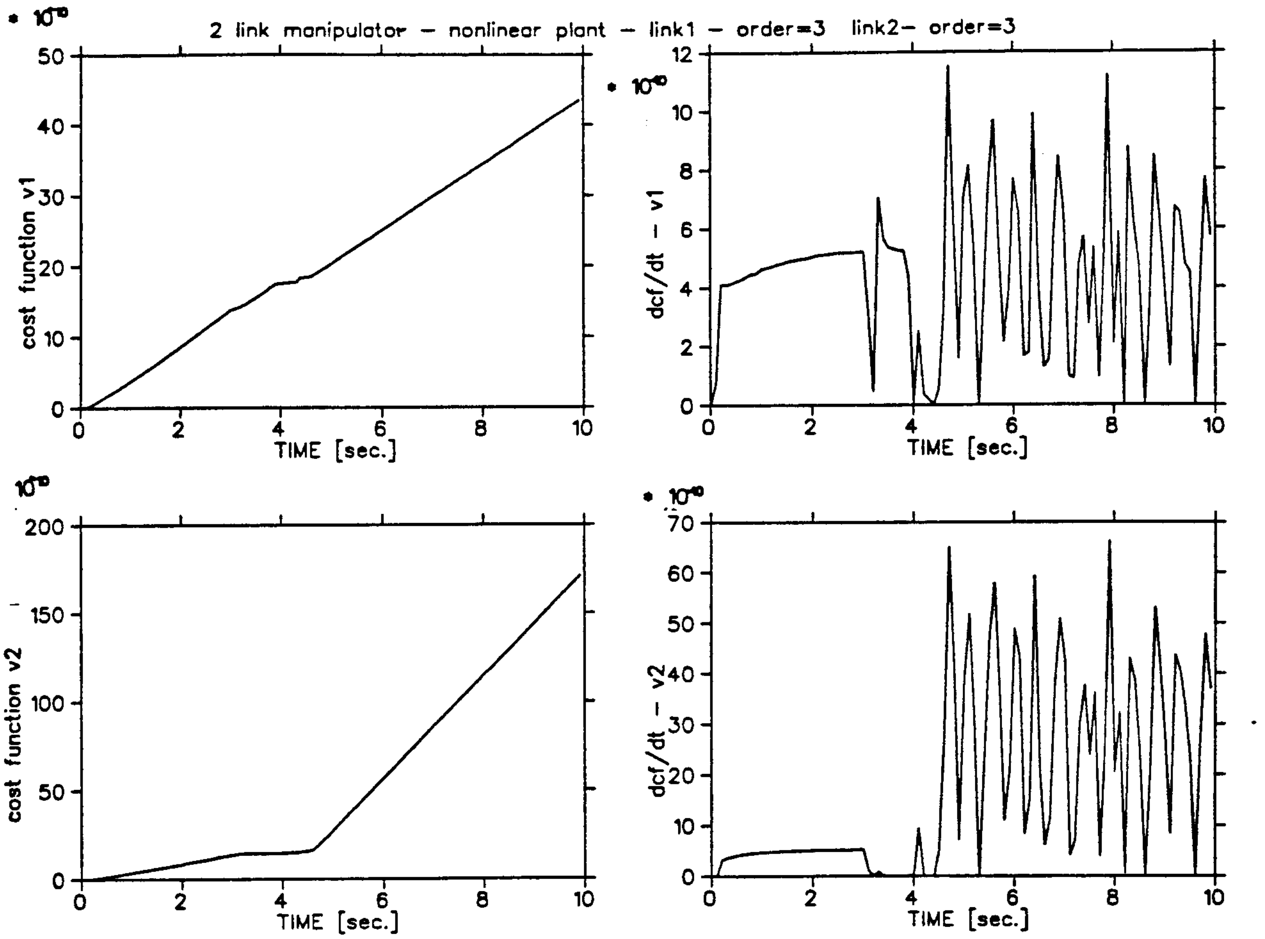

Figure 5.18: Cost function behavior for 3 mode hydraulic links (nonlinear plant model) and 3 mode estimated models

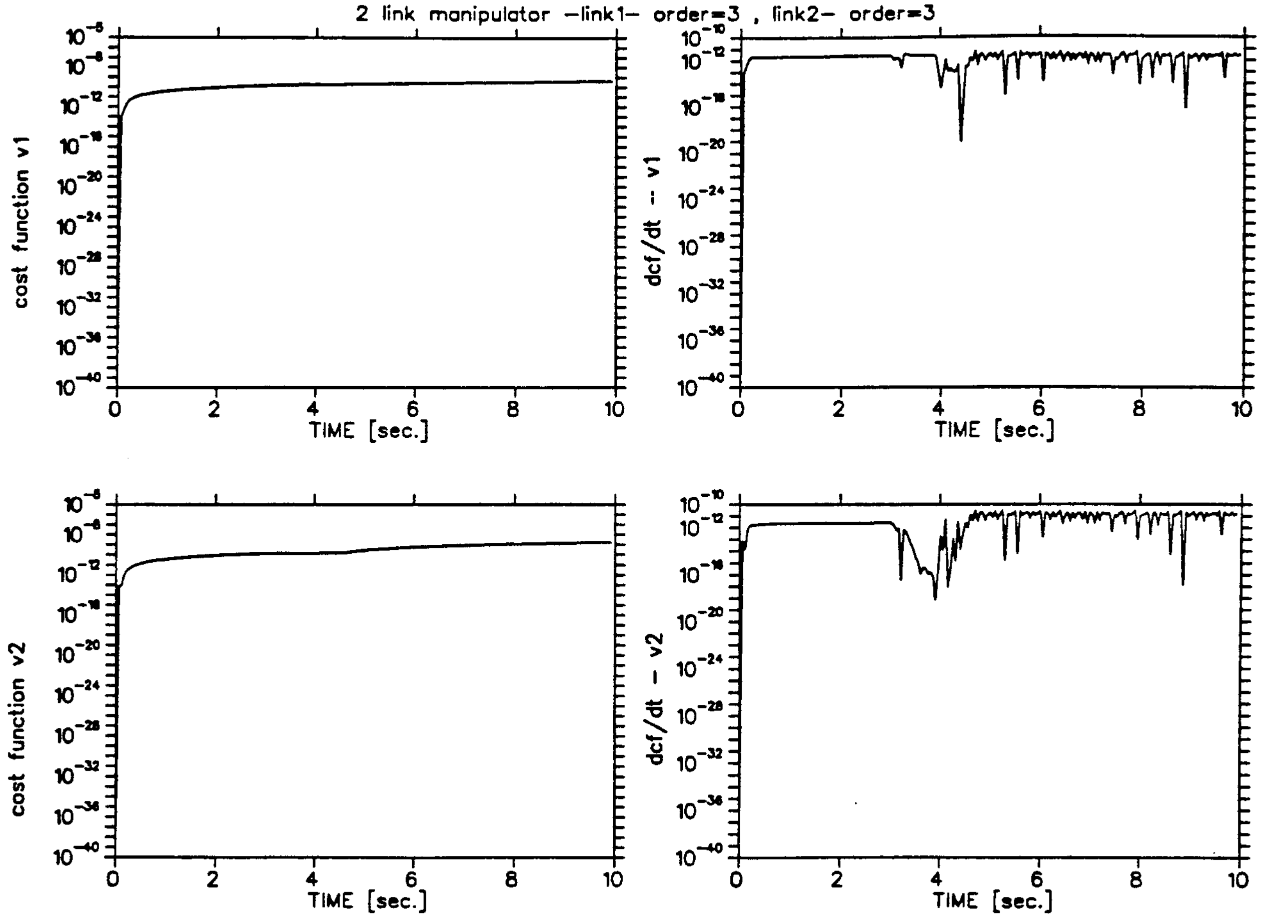

Figure 5.19: Cost function behavior for 3 mode hydraulic links and 2 mode estimated model

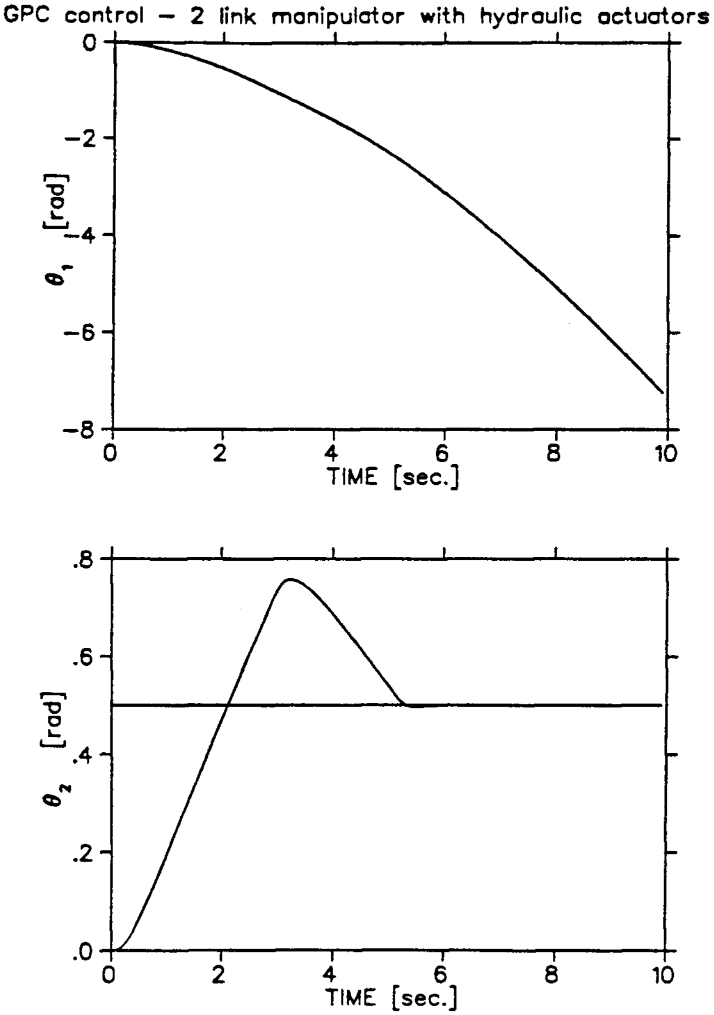

Figure 5.20: θ_1 and θ_2 behavior for 2 mode model and 3 mode estimated model

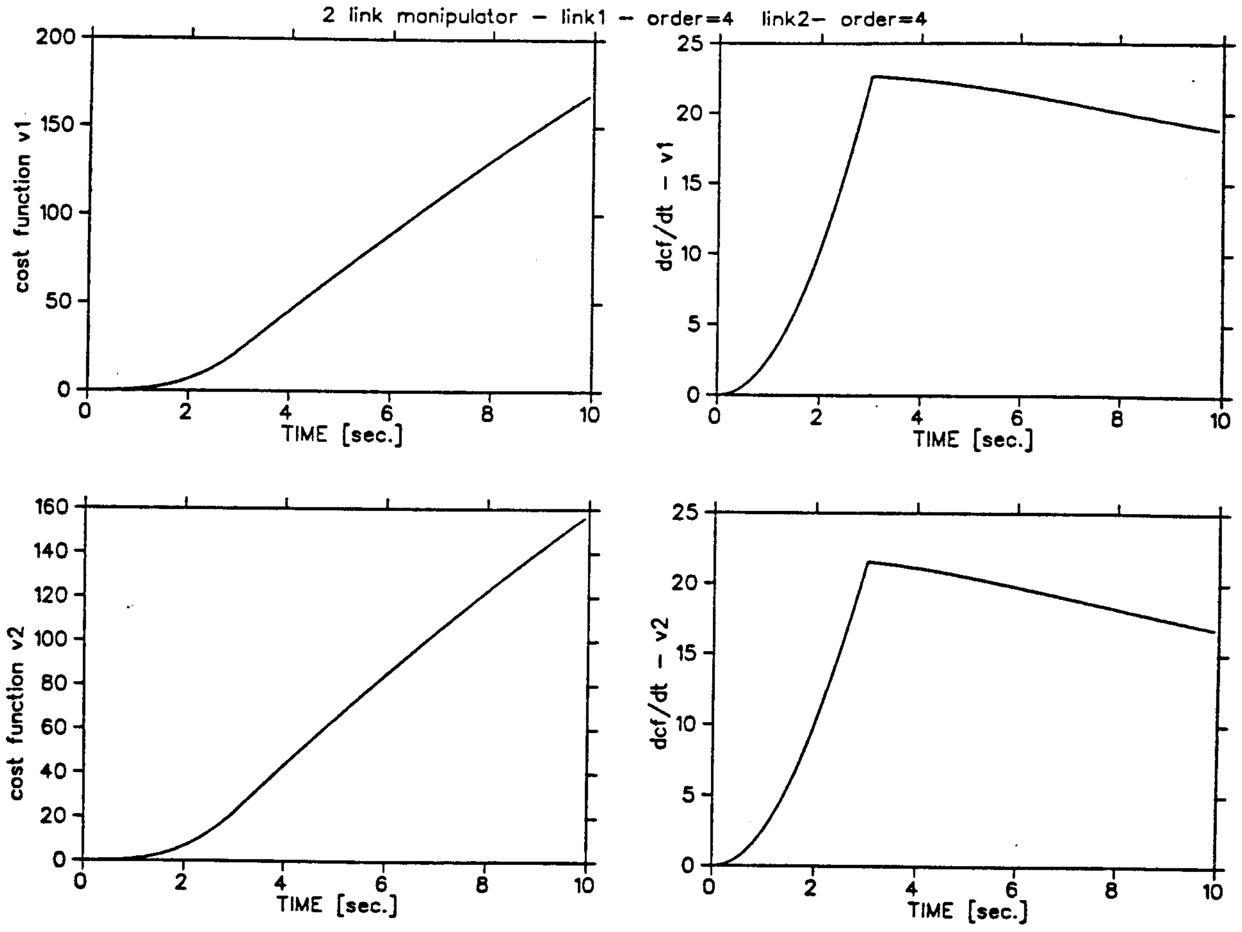

Figure 5.21: Cost function behavior for 4 mode hydraulic links and 4 mode estimated model

are fast enough to be ignored. In this case, an equivalent second order system was added to the structure of the estimated model. The initial values for the estimated parameters included the data for the servovalve dynamics with natural frequency of $20Hz$ and a damping ratio of 0.6 (see Catalog, Moog Inc. [61]). In Figure 5.22 link 1 has an estimated model of order 5, which grows constantly due to the error between the models. For link 2, the order is 3 and the cost function behaves like the one in Figure 5.18. Figure 5.23 shows the behavior of the outputs. θ_1 can not achieve the goal of its set point due to the mis-match of the models, and θ_2 behaves well since the models match each other.

The conclusions drawn from this section are similar to the ones from the flexible link. It is possible to identify, through the cost function and its derivative, the case in which the estimated model matches the actual system's model. The cost function also indicates under-modeling and over-modeling. This information forms the basic data for the method for detecting on-line the order of a system model and its changes.

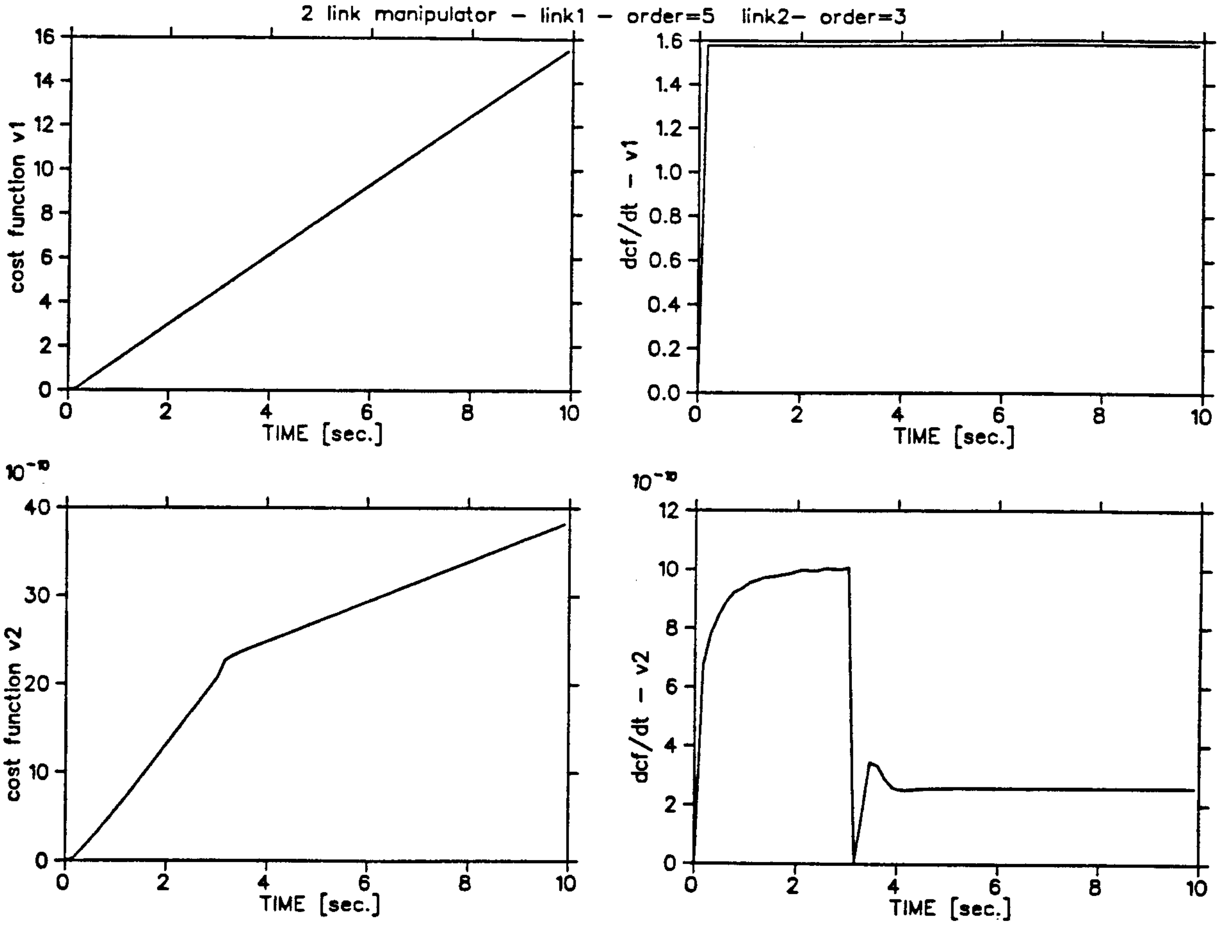

Figure 5.22: Cost function behavior for 5 mode hydraulic links and 3 mode estimated model

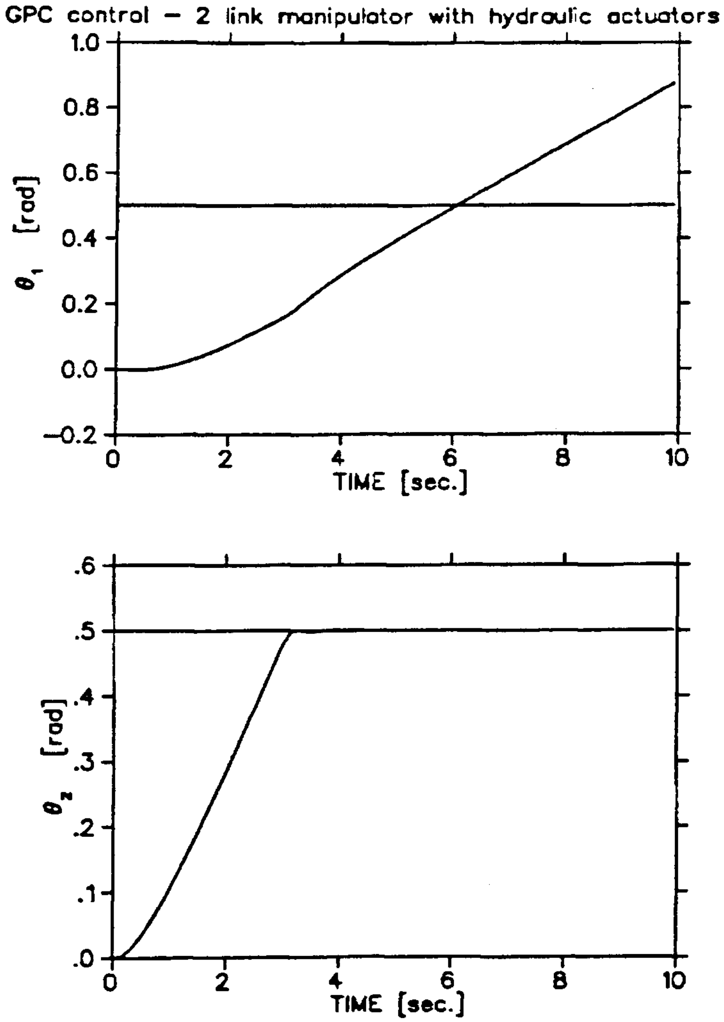

Figure 5.23: θ_1 and θ_2 behavior for 5 mode model and 3 mode estimated model

5.3 Reasons for Under and Over-Modeled Behavior

Most adaptive control algorithms assume that plant dynamics can be modeled by one member of a specified class of models. Usually, there are uncertainties in the estimated model due to unknown but estimated parameters or disturbances. These can be from external sources or internal ones such as nonlinearities in the plant dynamics which are not included in the estimated model. If the disturbances are bounded and there is sufficient excitation by the input signal to estimate the model parameters, then the system can be controlled and stability retained. This was demonstrated in Chapter 3 by controlling a single flexible link manipulator, and in Chapter 4 by controlling a hydraulically actuated two link manipulator which is a highly nonlinear system. Both systems were modeled by a linear model for control purposes, with parameters estimated on-line, and the nonlinearities are considered to be disturbances. By tuning the GPC parameters, acceptable and good performance can be achieved (see Figures 4.4, 4.5, 4.6, 4.7).

This work deals with model/plant mis-match in which the estimated model for the GPC algorithm has a different structure (i.e. linear and different order) from that of the real plant. Such mis-match is another form of uncertainty in the adaptive controller. The cost function (Equation 5.1) and its time variations were chosen as a measure of that phenomenon. When the plant and model match, the cost function rises initially for the time period that it takes for the estimated model to adjust, and then stabilizes on a close to constant value, since the error between the models becomes very small.

The behavior of the cost function J as a function of the estimated model order (Figure 5.1, for open loop investigation) shows that under-modeling, and over-modeling are hard to detect by comparison with the correct structure.

On the other hand, Figure 5.2 (closed loop calculations) shows that model mis-match in most forms is significant and can be detected. The real processes have complex nonlinear dynamics (as in the two link hydraulically actuated manipulator), and the adaptive controller attempts to control the dynamics by a simple linear model.

The parameters of the linear estimated model depend strongly on the properties of the input signal and its frequency content. Proper excitation is needed for good estimation results. There is self excitation when the estimation is done in closed loop (as when the adaptive controller is used), since the estimation process is excited by the signal from the feedback. The feedback could cause dependencies between the elements of the regression vector (Equation 5.4 which means that the parameters cannot be determined uniquely (Astrom [1]). Errors due to modeling errors arise when the chosen model does not describe the system completely, it can cause poor performance depending on the value of the modeling error and its nature.

In the following material, a discussion on the estimation process RLS (Recursive Least Squares), used with GPC (see Ljung [6] and Astrom and Wittenmark [15]), and the effect of under, and over-modeling is given.

When the plant is linear and its order is known, it can be described by the mathematical model:

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \cdots + b_0 u(t-1) + b_1 u(t-2) + \cdots \tag{5.2}$$

or:

$$y(t) = \Phi^T(t)\Theta \tag{5.3}$$

where Φ is the regression vector:

$$\Phi^T = [-y(t-1), \ \ -y(t-2), \ \ \cdots, \ \ u(t-1), \ \ u(t-2)\cdots] \tag{5.4}$$

and Θ is a vector of unknown parameters:

$$\Theta^T = [a_1, \ a_2, \ \cdots, \ b_0, \ b_1, \ \cdots] \tag{5.5}$$

The estimated model is:

$$\hat{y}(t) = \Phi^T(t)\hat{\Theta} \tag{5.6}$$

where $\hat{\Theta}$ is a vector of the estimated parameters, and $err(t)$ is the error.

$$err(t) = y(t) - \hat{y}(t) \tag{5.7}$$

Thus the RLS algorithm is:

$$\hat{\Theta}(t) - \hat{\Theta}(t-1) = \frac{\alpha P(t-1)\Phi(t)(y(t) - \hat{y}(t))}{\gamma + \Delta T \Phi^T(t) P(t-1)\Phi(t)} \tag{5.8}$$

$$P(t) - P(t-1) = \frac{-\alpha P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{\gamma + \Delta T \Phi^T(t) P(t-1)\Phi(t)} \tag{5.9}$$

where: $\alpha(t) \in [0,1]$ is a gain, $\gamma(t) > 0$ is a normalization term, ΔT is the sampling period, $y(t)$ is the measured output, and $\hat{y}(t)$ is the estimated output.

Under-Modeling

When under-modeling is considered the measured output can be expressed as:

$$y(t) = \Phi^T(t)\Theta + \eta(t) \tag{5.10}$$

where $\eta(t)$ contains the unmodeled terms.

rearranging Equation 5.8, Equation 5.10 and Equation 5.6 yields:

$$\tilde{\Theta}(t) - \tilde{\Theta}(t-1) = -\frac{\alpha P \Phi \Phi^T P \tilde{\Theta}}{\Gamma + \Delta T \Phi^T P \Phi} + \frac{\alpha P \Phi \eta}{\Gamma + \Delta T \Phi^T P \Phi} \tag{5.11}$$

Where:

$$\tilde{\Theta} = \hat{\Theta} - \Theta$$

Φ, the regression vector, contains information on previous outputs and inputs to the plant and therefore information about the feedback to the controller. The unmodeled dynamics, although unknown, are part of the plant's output and of the feedback signal. It can thus be concluded that Φ and η are dependent.

Equation 5.11 shows that even when the estimated parameters match the ones in Θ the term with $\Phi\eta$ in it can cause $\tilde{\Theta}$ to drift. This effect can also be seen in the equations describing the cost function. Basically when the error grows the cost function grows in value i.e.: Based on equations 5.10 5.5 5.6 and 5.7, the error in terms of the regression vector Φ^T, the difference between estimated and true parameters and the unmodeled dynamics is:

$$err(t) = \Phi^T(t)\tilde{\Theta} + \eta(t) \tag{5.12}$$

Where:

$$\tilde{\Theta} = [\tilde{a}_1, \; \cdots, \; \tilde{a}_{n_*}, \; \tilde{b}_1, \; \cdots, \; \tilde{b}_{n_*}]$$

and the regression vector for this case is:

$$\Phi^T = [-y(t-1),\ \cdots,\ -y(t-n_u),\ u(t-1),\ \cdots,\ u(t-n_u)] \tag{5.13}$$

its dimension is: $dim(\Phi^T) = 2n_u$, where n_u is the order of the modeled dynamics for the under-modeling case, and the terms $y(t-i)$ are:

$$y(t-i) = \Phi^T(t-i)\Theta + \eta(t-i) \tag{5.14}$$
$$i = 1, 2, \cdots, n_u$$

The term $\Phi^T(t-i)\Theta$ is derived using equation 5.5:

$$\Phi^T(t-i)\Theta = \sum_{j=i+1}^{n_u+i} -a_j y(t-j) + b_j u(t-j) \overset{\text{def}}{=} \Psi(y,u) \tag{5.15}$$

The cost function is defined as:

$$J = \sum_{k=1}^{t} err^2 \tag{5.16}$$

Based on Equation 5.12 it follows that:

$$err^2(t) = (\Phi^T(t)\tilde{\Theta})^2 + 2\Phi^T(t)\tilde{\Theta}\eta(t) + \eta^2(t) \tag{5.17}$$

The second term on the right hand side of Equation 5.19 contains the regression vector Φ, and the unmodeled dynamics η, which as can be concluded from Equations 5.14 and 5.15, are dependent. As can be seen from Equation 5.17 there are two contributions to the error, the modeling reflected in the unmodeled dynamics term, and the estimation which is reflected in the parameters. By expressing the regression vector Φ terms, with Equation 5.15 the term $\Phi^T(t)\tilde{\Theta}$ is calculated as:

$$\Phi^T(t)\tilde{\Theta} = \sum_{j=1}^{n_u}[-\tilde{a}_j(\eta(t-j) + \Psi(y,u))] + \tilde{b}_j u(t-j) \tag{5.18}$$

Substituting Equations 5.17 and 5.18 into Equation 5.16 the cost function is described by:

$$J = \sum_{k=1}^{t}(\Phi^T(k)\tilde{\Theta})^2 \tag{5.19}$$

$$+2\sum_{j=1}^{n_u}[-\tilde{a}_j \sum_{k=1}^{t} \eta(k)\eta(k-j) - \tilde{a}_j \sum_{k=1}^{t} \eta(k)\Psi(y,u) + \tilde{b}_j \sum_{k=1}^{t} \eta(k)u(k-j)]$$

$$+\sum_{k=1}^{t} \eta(k)\eta(k)$$

The unmodeled dynamics η is a physical signal and though unknown it is part of the plant's output. The second term in the right hand side of Equation 5.17 $\Phi^T(k)\tilde{\Theta}$, will be strong for under-modeling because of $\tilde{\theta}$ and because of the correlation between the regression vector Φ and the unmodeled dynamics η. When evaluating the correlation between measurements of pairs of variables, the correlation is determining whether there exists a physical relationship between the two, or whether the variations in the observed values of one quantity are correlated with the variations in the measured values of the other. In Press [62] the discrete correlation of two sampled fanctions is defined by:

$$Corr(g,h)_j = \sum_{k=1}^{t} g(j+k)h(k) \tag{5.20}$$

When g and h are the same function the above is the autocorrelation of the signal. The correlation will be large at some value of k if the first function g is a close copy of the second h but lags it by k. In Equation 5.19 several terms are summed with respect to time. Since η, y and u are real physical signals there are terms of autocorrelation and correlation. These terms do not exist in the over modeling case as will be shown in the discussion on over-modeling. The third term in the right hand side is an autocorrelation of two η signals shifted in time. The fourth and the fifth terms in Equation 5.19, are correlation terms between η and the output y or η and the input u. In the under-modeling case η is a part of the output y and since there is feedback of y it is correlated with the input u and therefor the unmodeled dynamics η as well. In the under-modeling case, the correlation between the different variables in Equation 5.19, is the reason for the rapid rise in the cost function's values as was shown in Section 5.2 for both applications.

Over-Modeling

When over-modeling is concerned y and $\hat{y}$ are:

$$y(t) = \Phi^T(t)\Theta \tag{5.21}$$

and:

$$\hat{y}(t) = \Phi^T(t)\hat{\Theta} + \eta(t) \tag{5.22}$$

The regression vector for the over modeling case is:

$$\Phi^T = [-y(t-1),\ \ -y(t-2),\ \ \cdots,\ \ -y(t-n),\ \ u(t-1),\ \ u(t-2),\ \ \cdots,\ \ u(t-n)] \tag{5.23}$$

its dimension is: $dim(\Phi^T(t)) = 2n$, where n is the correct order of the system, and $dim(\eta(t)) = n_o - n$, where n_o is the over-modeled model order. The error is then:

$$err(t) = \Phi^T(t)\tilde{\Theta} - \eta(t) \tag{5.24}$$

and :

$$err^2(t) = (\Phi^T(t)\tilde{\Theta})^2 - 2\Phi^T(t)\tilde{\Theta}\eta(t) + \eta^2(t) \tag{5.25}$$

$$\Phi^T(t)\tilde{\Theta} = \sum_{j=1}^{n} -\tilde{a}_j y(t-i) + \tilde{b}_j u(t-j) \tag{5.26}$$

Substituting Equation 5.26 into Equation 5.25 and into Equation 5.16 results in the cost function for over-modeling:

$$\begin{aligned} J &= \sum_{k=1}^{t}(\Phi^T(k)\tilde{\Theta})^2 \\ -2\sum_{j=1}^{n_u}[-\tilde{a}_j \sum_{k=1}^{t}\eta(k)y(k-j) + \tilde{b}_j \sum_{k=1}^{t}\eta(k)u(k-j)] \\ &+ \sum_{k=1}^{t}\eta(k)\eta(k) \end{aligned} \tag{5.27}$$

In this case, η contains all the extra terms of the estimated model. These dynamics are just in the estimated model and not in the real system. This means that there is no correlation between the regression vector Φ and the extra terms η, i.e. both variables are independent. The correlation and autocorrelation terms in Equation 5.19 do not exist in equation 5.27, and $\eta(k)$ in the third and fourth term of the right hand side acts as a time varying coefficient. η therefore influences the control parameters which influence the input to the process (u), but not the feedback of the controlled system.

Closed Loop Poles for Under, Over and Correct-Modeling

The change in the controller parameters is a change in the controller dynamics which determine the location of the closed loop poles. The minimization of the cost function for the estimation will determine the parameters uniquely, only when the model order is correct. When the model is over parametrized it can result in any one of several solutions, and the correct parameters cannot be determined. The closed loop poles show in some cases unstable modes indicating that the excess model dynamics add poles which are close to the dominant poles of the system driving the error into the higher values. This can result eventually in poor performance or in instability (especially if there is not enough excitation in the process). The reaction for the over-modeled dynamics is not as extreme as to the under-modeled ones.

Next we discuss some examples, from the flexible link application, for the behavior of the closed loop poles which show the unstable modes of the controlled system when mis-modeling poses a problem. The calculations of the closed loop poled are based on Clarke [4] and on Latornell [63]. First the closed loop poles for correct modeling:

correct-modeling - order 6 for plant and model

$$p1 = +0.9871 + 0.0140j$$

$$p2 = +0.3955 + 0.4982j$$

$$p3 = -0.6192 + 0.7582j$$

$$p4 = +0.0360 + 0.0000j$$

$$p5 = -0.6192 - 0.7582j$$

$$p6 = +0.3955 - 0.4982j$$

$$p7 = +0.9871 - 0.0140j$$

All poles for the correct modeling of a two mode flexible link (order 6) are within the unit circle indicating a stable system. The system behavior is presented in Figure 3.2 and the cost function in Figure 5.4.

correct-modeling - order 4 for plant and model

$$p1 = +0.9869 + 0.0132j$$

$$p2 = +0.5720 + 0.7359j$$

$$p3 = +0.0337 + 0.0000j$$

$$p4 = +0.5720 - 0.7359j$$

$$p5 = +0.9869 - 0.0132j$$

When one mode occurs for the flexible link and the system is correctly modeled the GPC conrtol achieves good results and the cost function behaves as presented in Figure 5.10. The closed loop poles as shown above are all in the stable region with in the unit circle.

over-modeling - order 6 for plant and order 8 for model

$$p1 = +0.9987 + 0.0299j$$

$$p2 = +0.6295 + 0.0000j$$

$$p3 = +0.5045 + 0.6561j$$

$$p4 = -0.1567 + 0.8209j$$

$$p5 = +0.3912 + 0.5150j$$

$$p6 = -0.6192 + 0.7582j$$

$$p7 = -0.6910 + 0.2549j$$

$$p8 = -0.6910 - 0.2549j$$

$$p9 = -0.6192 - 0.7582j$$

$$p10 = +0.3912 - 0.5150j$$

$$p11 = -0.1567 - 0.8209j$$

$$p12 = +0.5045 - 0.6561j$$

$$p13 = +0.9987 + 0.0299j$$

As previously mentioned when a system is over parametrized there is no unique solution to the identification process. If the excess dynamics add closed loop poles that are close to the dominant ones it could drive the controlled system into instabilities. In this case all closed loop poles are stable and it was shown that good performance was achieved. However $p6$ and

$p9$ are close to the circle at a radius of 0.97889 and $p1$ and $p13$ are at the radius of 0.9915. Any small change in the systems parameters or even in the control parameters (which are chosen for correct modeling of the best system known to the designer) could drive the system to instability.

over-modeling - order 4 for plant and order 10 for model

$$p1 = +1.2989 + 0.3954j$$

$$p2 = +0.8569 + 0.7815j$$

$$p3 = +0.2784 + 0.9573j$$

$$p4 = -1.0204 + 0.7592j$$

$$p5 = -0.0348 + 0.0000j$$

$$p6 = -1.0204 - 0.7592j$$

$$p7 = +0.2784 - 0.9573j$$

$$p8 = +0.8569 - 0.7815j$$

$$p9 = +1.2989 - 0.3954j$$

This over-modelig case is one with six unstable modes: $p1$ and $p9$ at a radius of 1.3577, $p2$ and $p8$ at 1.1597 and $p6$ and $p4$ at 1.2718, the system is unstables as the cost function indicates Figure 5.15.

under-modeling - order 6 for plant and order 4 for model

$$p1 = +1.2795 + 0.0000j$$

$$p2 = +0.7764 + 0.4191j$$

$$p3 = +0.2420 + 0.6053j$$

$$p4 = -0.5956 + 1.3359j$$

$$p5 = +0.3840 + 0.5073j$$

$$p6 = -0.7740 + 0.0000j$$

$$p7 = -0.3511 + 0.0000j$$

$$p8 = -0.5956 - 1.3359j$$

$$p9 = +0.2420 - 0.6053j$$

$$p10 = +0.7764 - 0.4191j$$

$$p11 = +1.2795 - 0.0000j$$

When under-modelimg occurs the reaction of the cost function was more rapid and the system became unstable faster than the under-modeling case. The unstable modes are: $p1$ at radius of 1.2795, and $p4$,$p8$ at 1.3002. The over-modeling of the two mode manipulator presented above was stable, in this case the unstable poles are quite far in the unstable region.

under-modeling - order 4 for plant and order 2 for model

$$p1 = +1.8581 + 0.0000j$$

$$p2 = +0.6089 + 1.3839j$$

$$p3 = +0.7904 + 0.8734j$$

$$p4 = -1.1640 + 0.0000j$$

$$p5 = -0.2321 + 0.0000j$$

$$p6 = +0.6089 - 1.3839j$$

$$p7 = +0.7904 - 0.8734j$$

In this under-modeled case the unstable modes are: $p1$ at radius of 1.8581, $p2$ and $p6$ at 1.5119, $p4$ at 1.164, and $p3$ and $p7$ at radius of 1.1779. in this case too, the unstable poles are further in the unstable zone that the ones for the over-modeling case (for plant order 4 and model order 10).

5.4 MOD - Model Order Determination Algorithm

Section 5.2 presents the behavior of the cost function J and its time variations for both applications, the flexible link manipulator and the hydraulically actuated manipulator, for under, over and correct modeling. The cost function is a measure of the accumulated error between the plant and the model dynamics. The difference between under and over-modeling is clear in the behavior of J its time derivatives as discussed in Section 5.2. This section presents an algorithm to detect mis-match between plant and model, based on the results above, and to correct the order. It should be noted that correcting mis-modeling is not a target in itself, but rather, is to detect a possible route to instability and poor performance. Thus, if an over-modeled system is well controlled, there is no reason to interfere. The goal of the method presented is to detect problematic mis-match cases, to identify their nature, and to correct them regardless of their cause.

As mentioned in Chapter 2 (Figure 2.1), the adaptive system contains two loops; one is an ordinary feedback loop, and the second loop identifies the estimated model parameters and updates the parameters of the controller. Figure 5.24 shows an adaptive system block diagram with a model determination block which is an addition to the two loops mentioned above. In the procedure a feedback loop is added to the identification loop. This loop calculates the error between the measured and estimated outputs and minimizes it with the algorithm given below. Figure 5.25 shows the block diagram of the order determination method.

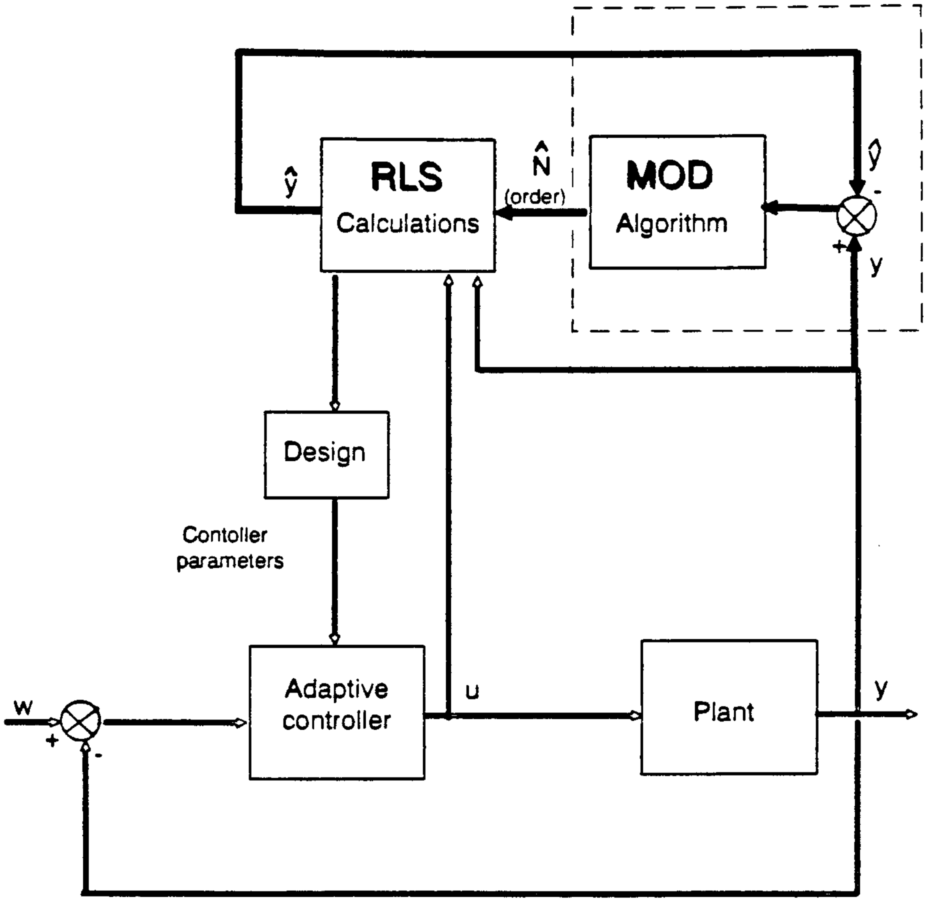

Figure 5.24: Flow chart of an adaptive control loop with model order determination

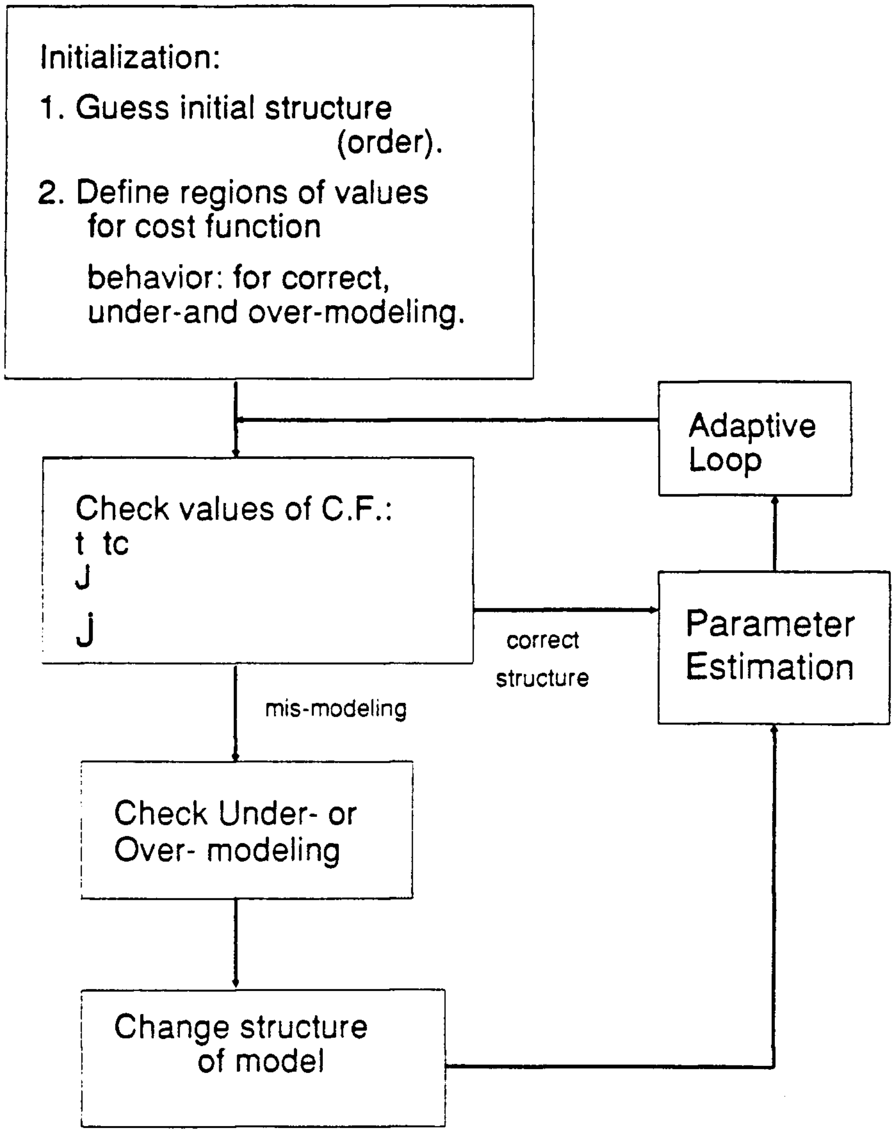
Initialization:
1. Guess initial structure (order).
2. Define regions of values for cost function
behavior: for correct, under-and over-modeling.
Adaptive Loop
Check values of C.F.:
t tc
J
j
correct structure
Parameter Estimation
mis-modeling
Check Under- or Over- modeling
Change structure of model

Figure 5.25: Flow chart of the order determination procedure

The order change is calculated with the MOD algorithm which is presented next:

$$N(t) = N(t-1) + \Delta N(J,\ \dot{J},\ t_{order},\ T_U,\ T_C,\ NDT) \tag{5.28}$$

Where J is the cost function which is described in Equation 5.1. Its derivative $\dot{J}$ is the following:

$$\dot{J} \simeq \frac{J(t) - J(t-1)}{\Delta T} \tag{5.29}$$

Where:

- t_{order} is the MOD's time scale. In the event of several order changes during a working cycle, in every change t_{order} is set to zero. This moves the origin of the time scale relative to the absolute time t, and enables the time parameters (that will be stated next) for each order change to be considered.
- T_U is the time for under modeling detection.
- T_C is the time when the cost function changes to Zero Slope for correct modeling. Its value will be within the region $T_{C_{min}} \leq T_c \geq T_{C_{max}}$.
- NDT is the number of time steps to wait for convergence after an order change.
- N_0 is the initial guess for the order.
- K_{wait} is the number of time steps to wait between indication of possible mis-modeling and its acceptance.

The order change function, ΔN based on the behavior of J in Equation 5.19 for under-modeling and Equation 5.27 for over-modeling and for correct modeling, is as follows:

$$\Delta N = \begin{cases} 0 & t_{order} \begin{cases} < \Delta T \cdot NDT & \\ < T_C & J \leq J_{R_m} \quad \dot{J} \leq \dot{J}_{R_m} \\ \geq T_C & J \leq J_{Z_m} \quad \dot{J} \leq \dot{J}_{Z_m} \end{cases} \\ & \\ DUM & t_{order} \begin{cases} < T_U & J \geq J_{R_U} \quad \dot{J} \geq \dot{J}_{R_U} \end{cases} \\ & \\ -DOM & t_{order} \begin{cases} < T_C & J \geq J_{R_o} \quad \dot{J} \geq \dot{J}_{R_o} \\ \geq T_C & J \geq J_{Z_o} \quad \dot{J} \geq \dot{J}_{Z_o} \end{cases} \end{cases} \tag{5.30}$$

where :

- J_{R_U} and $\dot{J}_{R_U}$ are minimum values for the cost function and its derivative, for identifying under-modeling.
- J_{R_o} and $\dot{J}_{R_o}$ are minimum values for the cost function and its derivative, to identify over-modeling, for $t < T_C$.
- J_{Z_o} and $\dot{J}_{Z_o}$ are minimum values for the cost function and its derivative, to identify over-modeling, for $t \geq T_C$.
- J_{R_m} and $\dot{J}_{R_m}$ are maximum values for the cost function and its derivative, to identify correct-modeling.
- J_{Z_m} and $\dot{J}_{Z_m}$ are minimum values for the cost function and its derivative, to identify correct-modeling.
- DUM is the under-modeling addition to the order at each order change step.
- DOM is the over-modeling subtraction from the order at each order change step.

This is a gradient algorithm designed to minimize the number of steps to achieve the correct order. Based on the results from the investigation of the robotic applications presented in this work parameters initial values were determined. The MOD algorithm was implemented on the flexible and the hydraulic manipulator and was found to be stable in behavior due to several factors:

- K_{wait} is the number of time steps to wait and verify the need for order change. This prevents a random increase in the values of the parameters and on unnecessary order change.
- t_{order} a relative time origin is used and reset after an order change to what is believed is the correct value.

Chapter 6

IMPLEMENTATION OF THE MOD ALGORITHM

6.1 Implementation of the Order Determination Algorithm

The algorithm implementation is described as follows:

1. Initialization: definitions by the user

   (a) N_0 : initial value for the estimated model order.

   (b) $T_{C_{min}} \leq T_C \geq T_{C_{max}}$: time region for the time change.

   (c) T_U : $T_U \leq T_{C_{min}}$: time for under-modeling detection

   (d) K_{wait}: number of time steps to wait between indication of possible mis-modeling and its acceptance.

   (e) $NUDT$: the number of time steps to wait for convergence, when an order change was done due to under-modeling.

   (f) $NODT$: the number of time steps to wait for convergence, when an order change was done due to over-modeling.

   (g) data for correct-modeling:

      i. at - $t \leq T_C$

         - J_{R_m} : maximum value for rising cost function: $J \leq J_{R_m}$
         - $\dot{J}_{R_m}$: maximum value for rising slope $\dot{J} \leq \dot{J}_{R_m}$

      ii. at - $t \geq T_C$

- J_{Z_m} : maximum value for zero slope cost function: $J \leq J_{Z_m}$
- $\dot{J}_{Z_m}$: maximum value for zero slope $\dot{J} \leq \dot{J}_{Z_{nl}}$
- $J_{Z_{nl}}$: influence of nonlinearities $J_{Z_{nl}} \geq J_{Z_m}$
- $\dot{J}_{Z_{nl}}$: influence of nonlinearities $\dot{J}_{Z_{nl}} \geq \dot{J}_{Z_m}$

(h) data for under-modeling:

   i. at - $t \leq T_U$

   - J_{R_U}: minimum value for cost function (to identify under-modeling): $J \geq J_{R_U}$
   - $\dot{J}_{R_U}$: minimum value for cost function slope (to identify under-modeling): $\dot{J} \geq \dot{J}_{R_U}$

(i) data for over-modeling:

   i. at - $t \leq T_C$

   - J_{R_O}: minimum value for cost function (to identify over-modeling): $J \geq J_{R_O}$
   - $\dot{J}_{R_O}$: minimum value for cost function, slope (to identify under-modeling): $\dot{J} \geq \dot{J}_{R_O}$

   ii. at - $t \geq T_C$

   - J_{Z_O}: minimum value for cost function (to identify over-modeling): $J \geq J_{Z_O}$
   - $\dot{J}_{Z_U}$: minimum value for cost function, slope (to identify under-modeling): $\dot{J} \geq \dot{J}_{Z_O}$

2. The MOD Algorithm

   (a) Check time relative to T_C and values of the cost function J, and its slope $\dot{J}$.

(b) Determine if mis-modeling is indicated, then:

- Determine under or over-modeling
- Wait for K_{wait} time steps
- The algorithm has logic to handle model order changes according to the type of mis-modeling
- Convergence to acceptable structure when at $t \geq T_C$: $J \leq J_{Z_m}$ and $\dot{J} \leq \dot{J}_{Z_m}$

(c) When mis-modeling is not indicated:

- convergence for verification of the model model occurs: when at $t \geq T_C$: $J \leq J_{Z_m}$ and $\dot{J} \leq \dot{J}_{Z_m}$

(d) At each time step, the control algorithm is activated with the present model order.

6.1.1 The Method For The Flexible Link Manipulator

Data for the Flexible Link

The values for the parameters presented in this section are based on the investigation done in Section 5.2 for the behavior of the cost function of the flexible link. In Section 5.2 the MOD parameters were determined from simulation results. In other applications of the algorithm, such simulations would first be run on-line with MOD turned off, the order of the model will be changed and based on the cost function behavior parameters will be determined. The values for TC_{min}, $T_{C_{max}}$, J_{R_m}, $\dot{J}_{R_m}$, J_{Z_m}, $\dot{J}_{Z_m}$, $J_{Z_n l}$, $\dot{J}_{Z_n l}$ were determined from correct modeling results shown in Figure 5.4 and Figure 5.10. Values for J_{R_U}, $\dot{J}_{R_U}$ and T_U are from data based on under-modeling Figure 5.6, Figure 5.7 and Figure 5.12. Data for the over-modeling case, J_{R_o}, $\dot{J}_{R_o}$, J_{Z_o}, $\dot{J}_{Z_o}$, was obtained from Figure 5.8, Figure 5.9, Figure 5.13, Figure 5.14, Figure 5.15.

Figure 6.1 shows the regions in which J and $\dot{J}$ indicated the mis-modeling. The shaded areas show for the flexible link at what values under, over or correct modeling occur.

1. Time data
   (a) $T_{C_{min}} = 1.4[sec.]$
   (b) $T_{C_{max}} = 2.5[sec.]$
   (c) $T_U = 0.5[sec.]$
2. Data for correct modeling
   (a) $J_{R_m} = 5 \cdot 10^{-9}$
   (b) $\dot{J}_{R_m} = 5 \cdot 10^{-9}$
   (c) $J_{Z_m} = 10^{-9}$
   (d) $\dot{J}_{Z_m} = 10^{-12}$
   (e) $J_{Z_nl} = J_{Z_m}$
   (f) $\dot{J}_{Z_nl} = \dot{J}_{Z_m}$
3. Data for under-modeling
   (a) $J_{R_U} = 10^{-4}$
   (b) $\dot{J}_{R_U} = 10^{-3}$
4. Data for over-modeling
   (a) $J_{R_O} = 10^{-6}$
   (b) $\dot{J}_{R_O} = 10^{-5}$

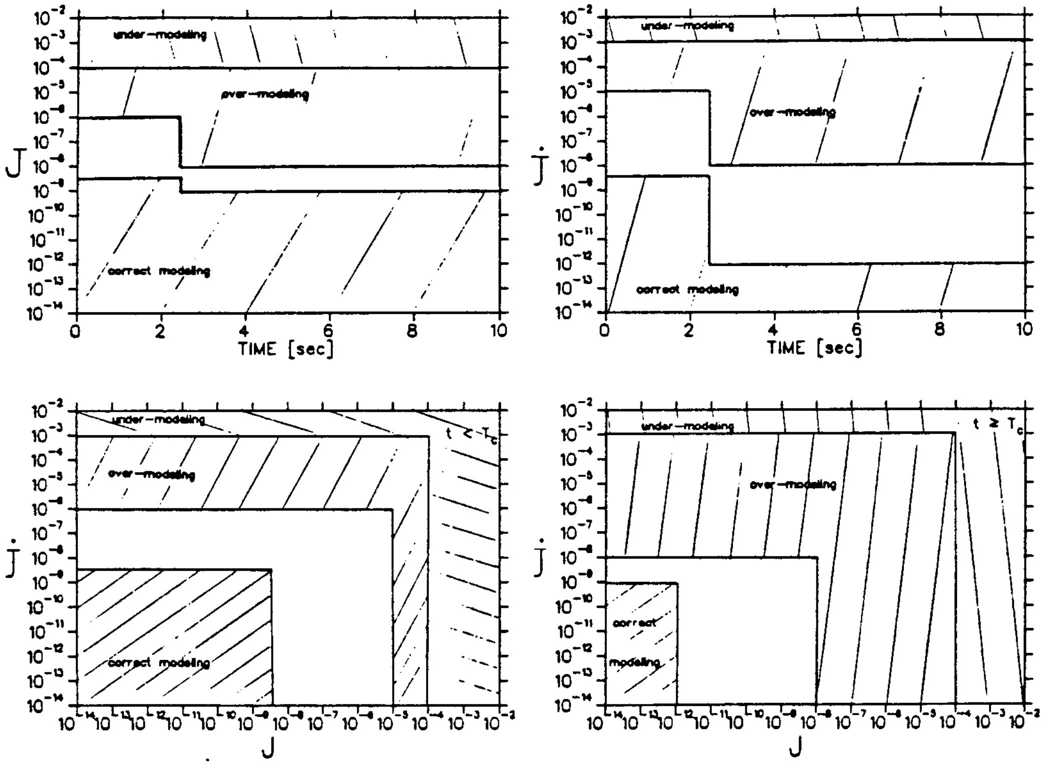

Figure 6.1: Regions for under, over and correct modeling

(c) $J_{Z_O} = 10^{-8}$

(d) $\dot{J}_{Z_O} = 10^{-8}$

5. Data for other parameters

(a) $DUM = 2$, (4), (6)

(b) $DOM = 1$, (2)

Results for the Flexible Link

The data for both the two mode and the one mode examples when modeled correctly, show that when $t < T_c$, the values of $\dot{J}$ are approximately $0.5\,10^{-9}$ and at $t \geq T_c$ J is of the order of magnitude of 10^{-9}. When under-modeled at small t, the values of $\dot{J}$ rise to the order of magnitude of 10^{-3} and higher, and so do the values of J. For over-modeling the changes are more moderate. In the case of small over-modeling (by order of 2), the mis-modeling can not be detected since the control algorithm works well and the C.F. derivative's values are small. When over-modeling is larger, the values of J change in a more moderate slope. The observation of the values of J and $\dot{J}$ were made on the basis of results in Section 5.2. J accumulates its values with time, and order changes on-line. Its values may be higher, but its shape remains the same and important. $\dot{J}$ values do not change and are very important. The results to be shown describe the behavior of the systems dealt with in this work in under and over-modeling for different values for the data needed by the order determination algorithm. (Note: each time an order change is done, the algorithm sets its internal time to zero, so all parameters for the value regions can be treated in the proper time frame and not in the absolute one).

Figures 6.2, 6.3, show the behavior of the 2 mode (*order* = 6) flexible link when under-modeled with a second order estimated model. Case A is the behavior of the system when no order correction is done. Ytip goes unstable. In case B, the system converges to the desired set point, but has a very high overshoot. The algorithm detects the under-modeling soon enough, but does not have the needed time to settle on *order* $= 6$. It understands it as over-modeling again and goes to *order* $= 4$, which drives the system into instability. The algorithm then changes the order to the value of *order* $= 10$, which stops the rapid change in the values of J. After $T_{c_{max}}$ the over-modeling has been detected and the order is reduced to 8, $\dot{J}$ is reduced to the region accepted as the correct modeling and the system goes to the value of the set point. In case C some of the parameters have been adjusted. First, more time has been given for the system to settle after the under-modeling was detected and changed. In addition, the values for J_{Z_O} and $\dot{J}_{Z_O}$ have both been increased to 10^{-7}. All the changes made in the parameters of this case were made done order to increase the time of convergence between order changes. As a result, YTIP has almost no overshoot. It takes a little more time to converge to the set point (about 4 seconds instead of about 1.5 seconds); yet, it is far better than the under-modeled response presented in case B and ofcourse case A.

The next set of results combines under and over-modeling in the process of correction an under-modeling case of a two mode (*order* $= 6$) flexible link initially under-modeled with estimated model of *order* $= 4$. Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7 show three such cases. Case A shows the unstable behavior of a two mode flexible link, controlled with an estimated model of order 4. In case B, the output converges to the set point slower (about 5 seconds in contrast to 1.5 seconds) and has an overshoot of 27 percent, but it does not go unstable like case A. The parameters for the order determination algorithm are $DOM = 2$, $J_{Z_O} = 10^{-5}$ and $\dot{J}_{Z_O} = 10^{-5}$. The algorithm detects the under-modeling soon enough so as not to have a very large overshoot, and the order is set to the correct value of 6. Therefore, the cost function derivative rises at first, but after the order change, it drops and settles on an order of magnitude of 10^{-13} which is an indication for convergence. In case C, $DOM = 2$, $DUM = 6$, $J_{R_O} = 10^{-9}$ and $\dot{J}_{R_O} = 10^{-9}$. The output has a larger overshoot (Figure 6.4) since

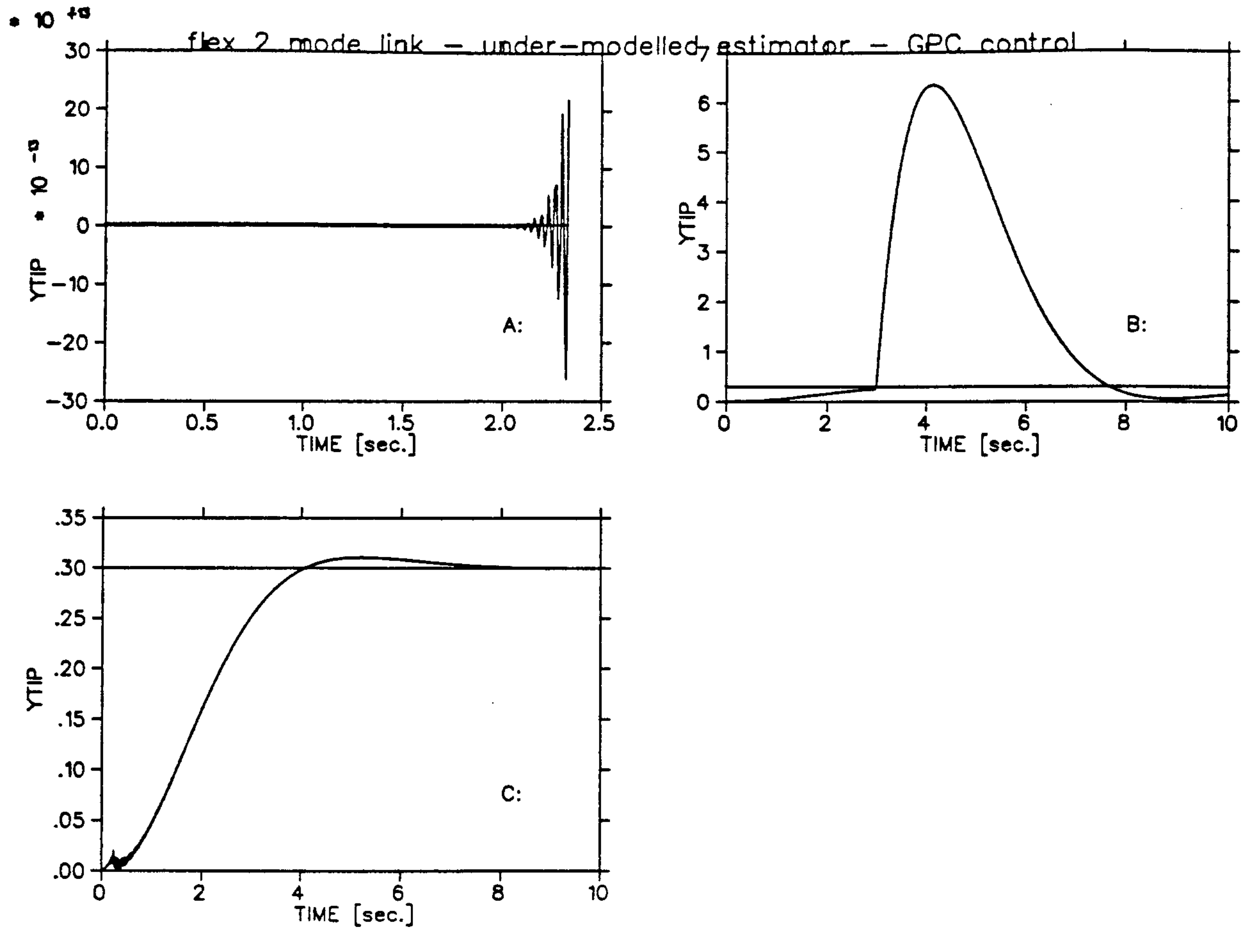

Figure 6.2: The output behavior of a two mode flexible link estimated initially with an order 2 model

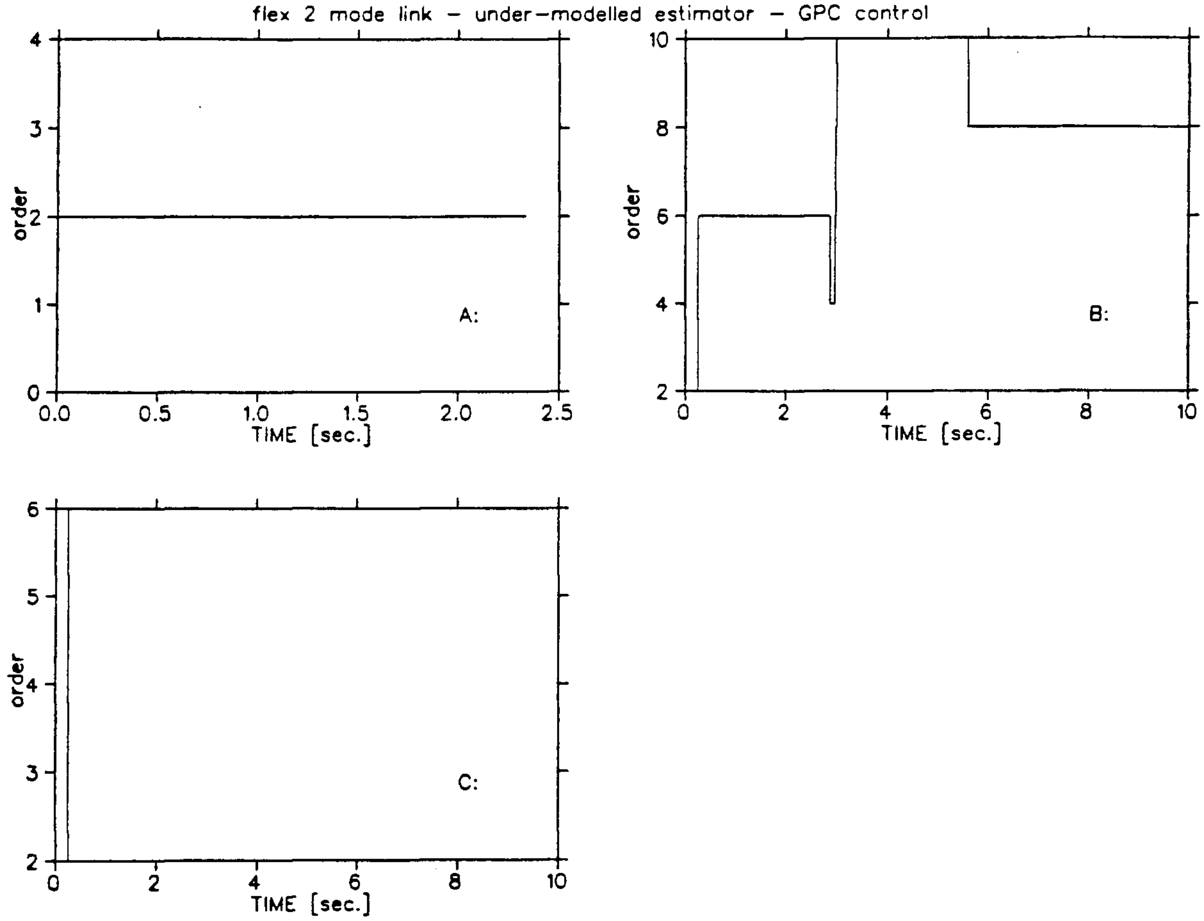

Figure 6.3: Order changes of the estimated model for a two mode flexible link estimated initially with order 2

the order of the estimated model changes from under-modeled ($order = 4$) to over-modeled ($order = 10$) to ($order = 8$) and is inaccurate for a longer time. But the output converges at the same time as case B.

Figure 6.8, Figure 6.9, again show the behavior of the 2 mode flexible link when initially under-modeled ($order = 4$) and depending on the order change of algorithm parameters, later over- modeled. Case A has a larger $NUDT_U$, $DUM = 6$, $DOM = 2$. $NUDT_U$ is the parameter defining the number of time steps, after under-modeling is detected. When it has a larger value, the system is in the unstable mode longer; thus, there is a large overshoot (Figure 6.8 case A). $DUM = 6$ changes the order to the value of 10, over-modeling that yields higher values for the cost function and its derivative; with time, the order is reduced to 6. The whole procedure resulted, as mentioned, in a high overshoot and in a slower convergence. In case B the time to change the order after under-modeling is detected was reduced, while all other parameters remained unchanged. The result is a much smaller overshoot and a faster convergence. Case C presents the best result of the three, with an overshoot of about 15 percent and convergence to the setpoint within less that 6 seconds.

The one mode flexible link ($order = 4$) when under-modeled, reacts like the two mode link to under-modeling. Figures 6.10, 6.11, present 3 cases. Case A is a one mode ($order = 4$) flexible link under-modeled with an estimated model of order 2. The output is unstable like the corresponding in Figure 5.11. The cost function and its derivative rise to high values which indicate the responding instability. In case B, the order is changed on line to stabilize the response. When under-modeling is detected, the order is changed, by the parameters given a priori, to an over- modeling value of 8, and then goes down gradually to the correct value of 4. The response has an overshoot of 83 percent which is mainly due to the initial instability and the later over-modeling. But the overall response, instead of going unstable, converges to the set point after an acceptable time (about 4 seconds). Case C presents a better behavior of the system, in which the overshoot is smaller. The order change of algorithm parameters

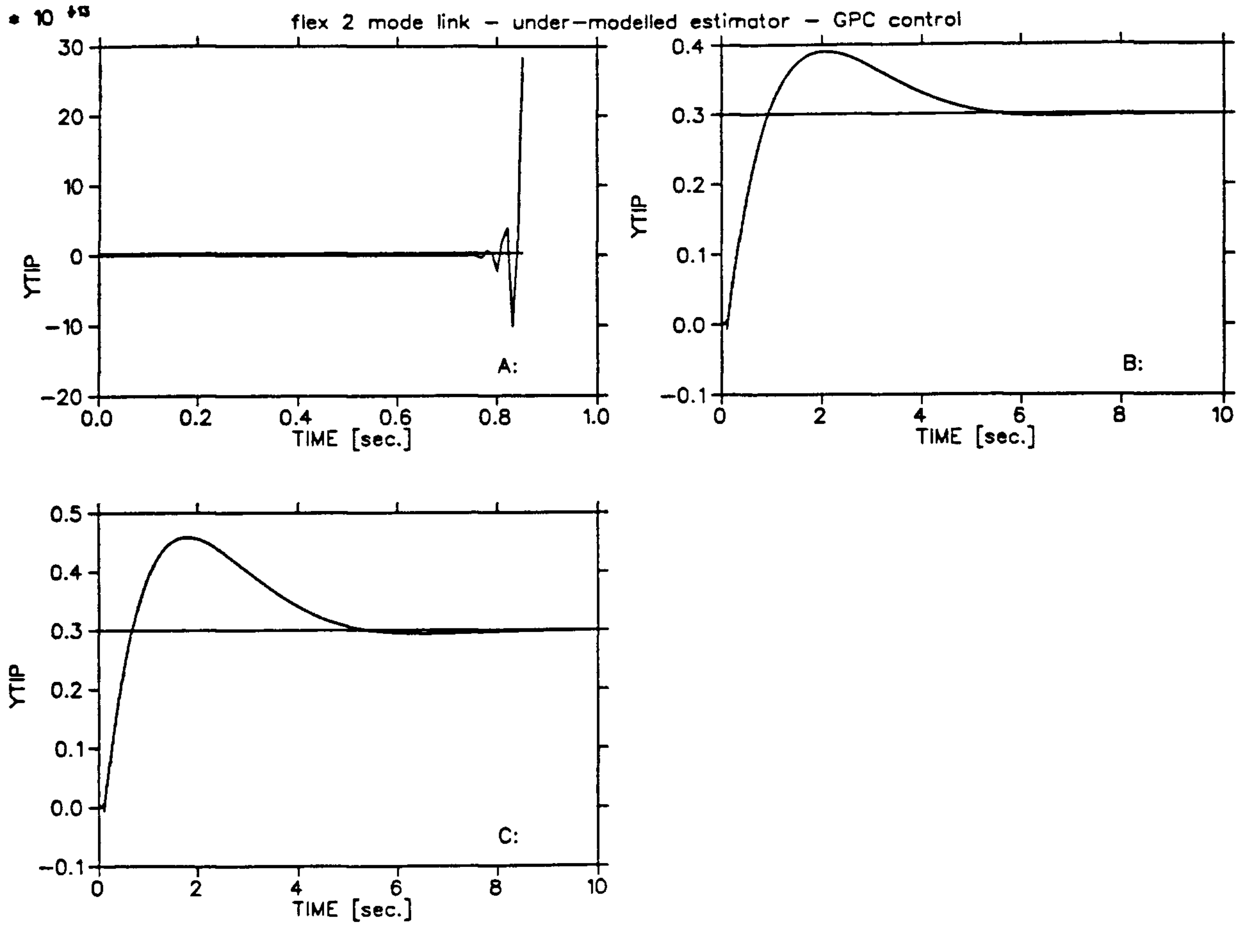

Figure 6.4: The output behavior of a two mode flexible link estimated initially with an order 4 model

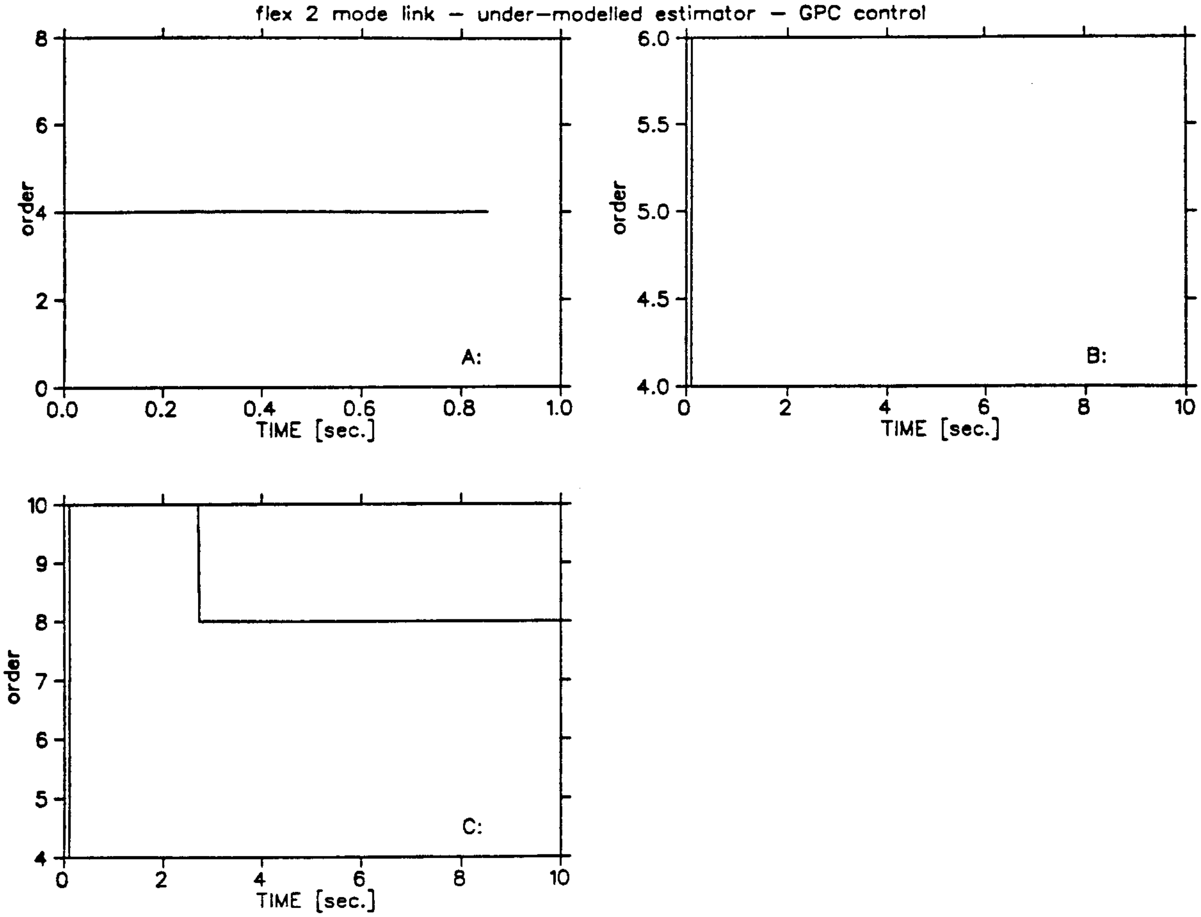

Figure 6.5: Order changes of the estimated model for a two mode flexible link estimated initially with order 4

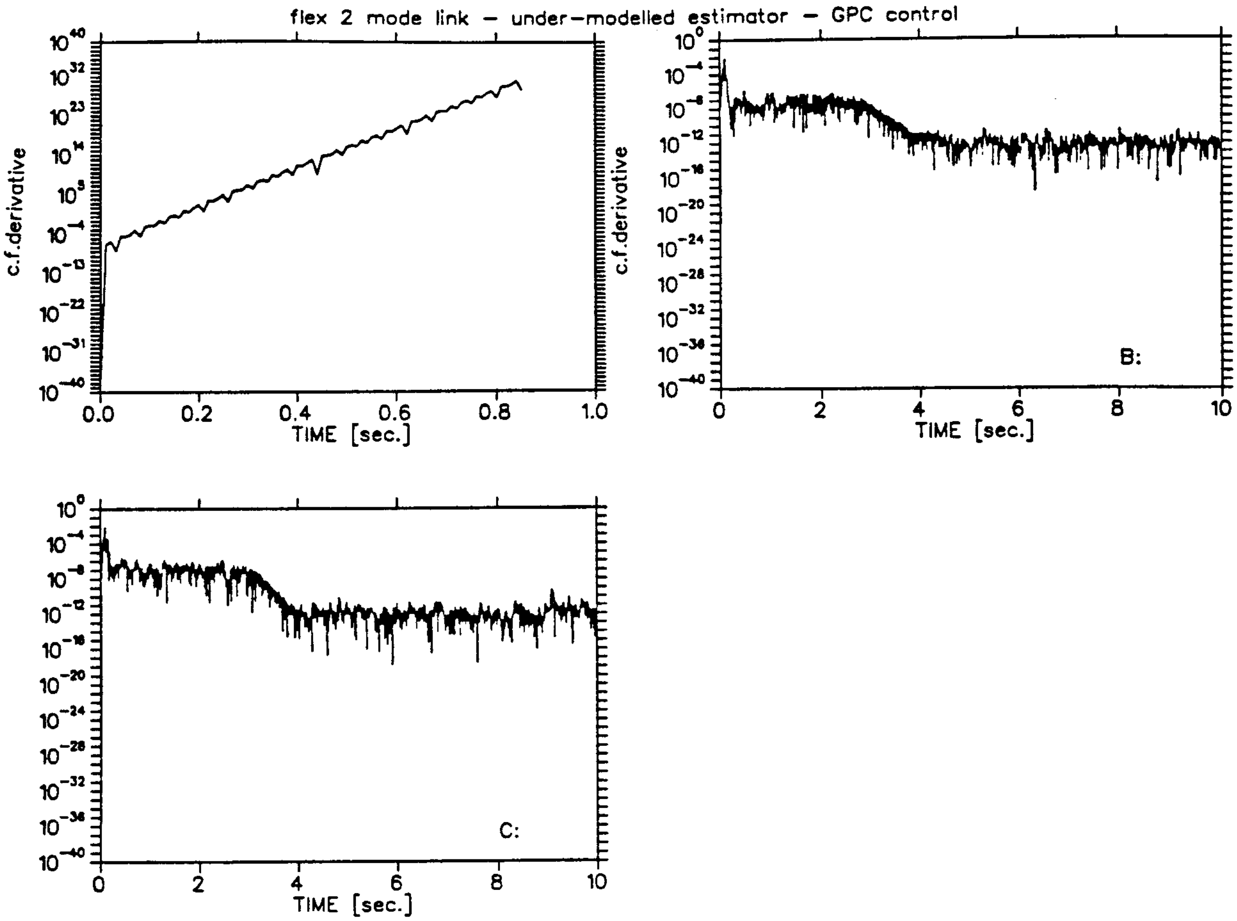

Figure 6.6: The cost function derivative behavior of a two mode flexible link estimated initially with an order 4 model

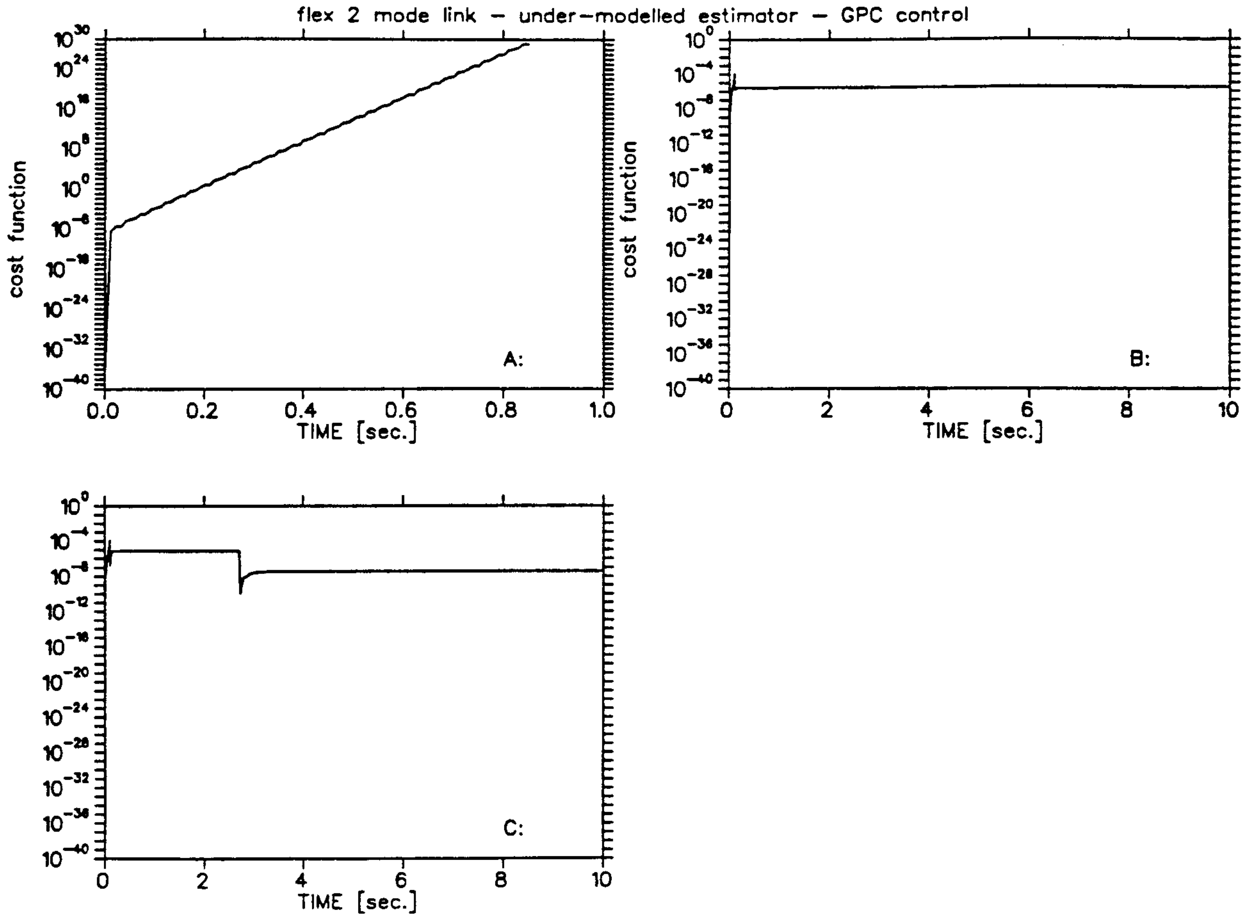

Figure 6.7: The cost function behavior of a two mode flexible link estimated initially with an order 4 model

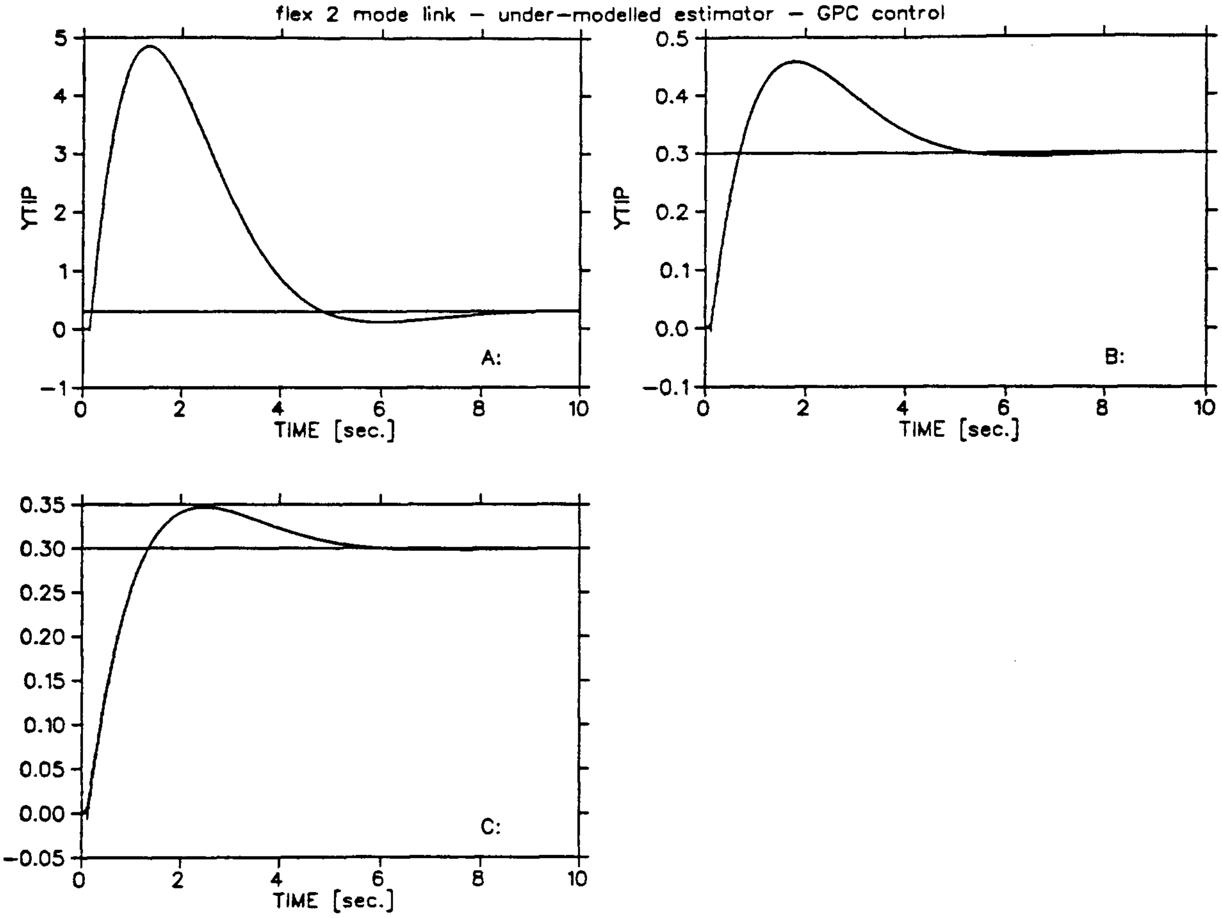

Figure 6.8: 3 more cases of output behavior of a two mode flexible link estimated initially with an order 4 model

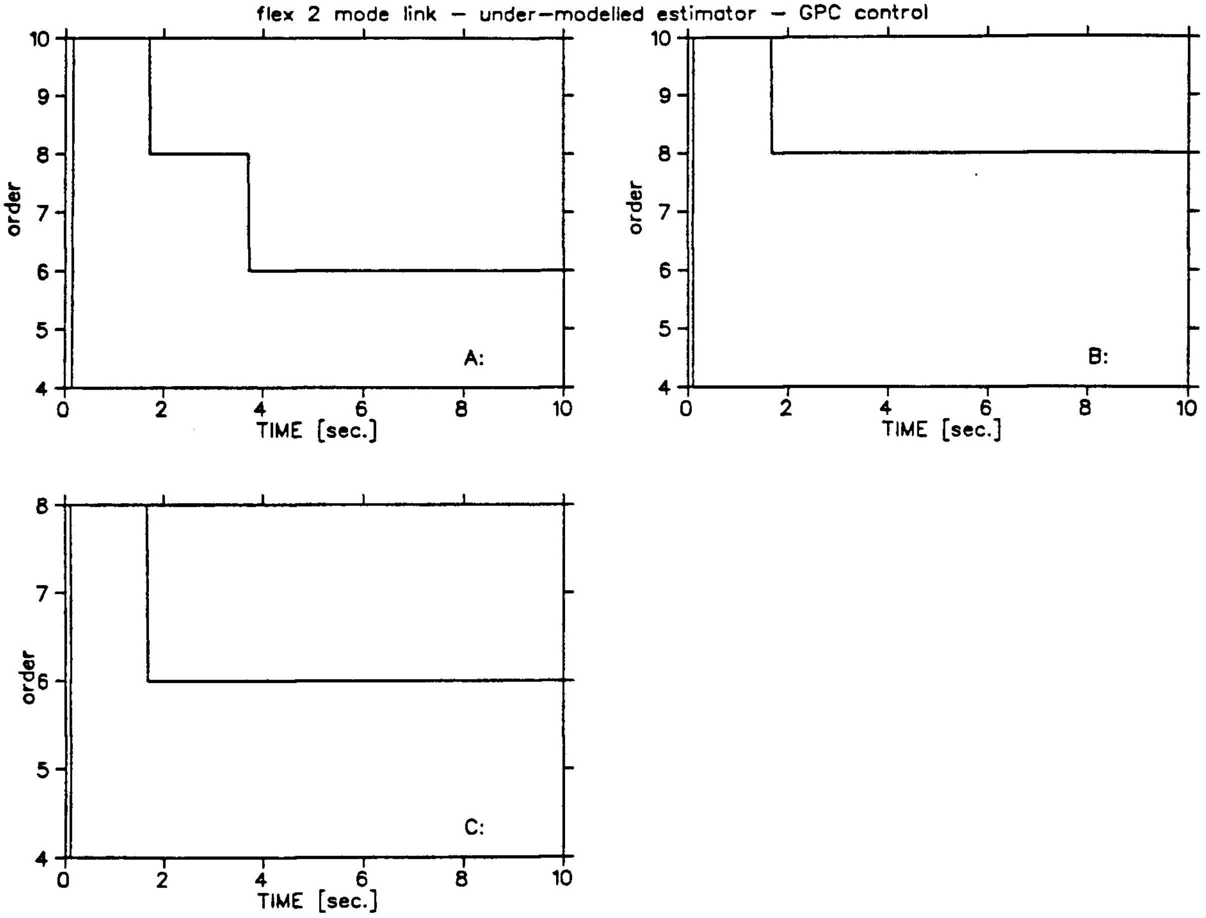

Figure 6.9: 3 more cases of order changes of the estimated model for a two mode flexible link estimated initially with order 4

was changed, with DUM that has a smaller value of 4 instead of 6. Thus after detecting the under-modeling, the order rises to the value of 6 (the correct one is 4); yet the overshoot is still quite high due to the initial instability.

Figures 6.12, 6.13, show 3 cases for initial over-modeling (for a one mode link) where the different cases present different parameters of the order determination algorithm. In all of the 3 cases, the estimated model order for the control algorithm is initially 10, and the outputs vary slightly in the overshoot and time of convergence. In case A, $DOM = 4$, and the algorithm, after detecting over-modeling, sets the order on 6. In case B, the parameters changed are:$DOM = 2$, $T_{C_{max}} = 2.5$ sec., $J_{R_O} = 10^{-6}$, $\dot{J}_{R_O} = 10^{-6}$. The order detection algorithm detects the over-modeling and changes it to the value of 8, then after 4.3 sec. to the order of 6. Case C has $T_{C_{max}} = 1.5$ sec., $J_{R_O} = 10^{-9}$, $\dot{J}_{R_O} = 10^{-9}$, which brings a quicker change of the order from 8 to 6. The differences between the three results is small. Cases A and C have almost no difference because of the quick change of $order = 8$ or 10 to $order = 6$; Case B takes more time and the cost function and its derivative limits are higher, so the result has a slightly higher overshoot and takes a little longer to converge.

The conclusions drawn so far from the results of the flexible link (one or two modes) is that under-modeling creates instability, which can be controlled by detecting the under-modeling and changing it to the correct one or one close to it. The results may take longer to converge and have an undesired larger overshoot (than the correct modeling), but the output is not unstable. The over-modeling has a much more moderate response, which is easier to control after detecting it and changing the order of the estimated model.

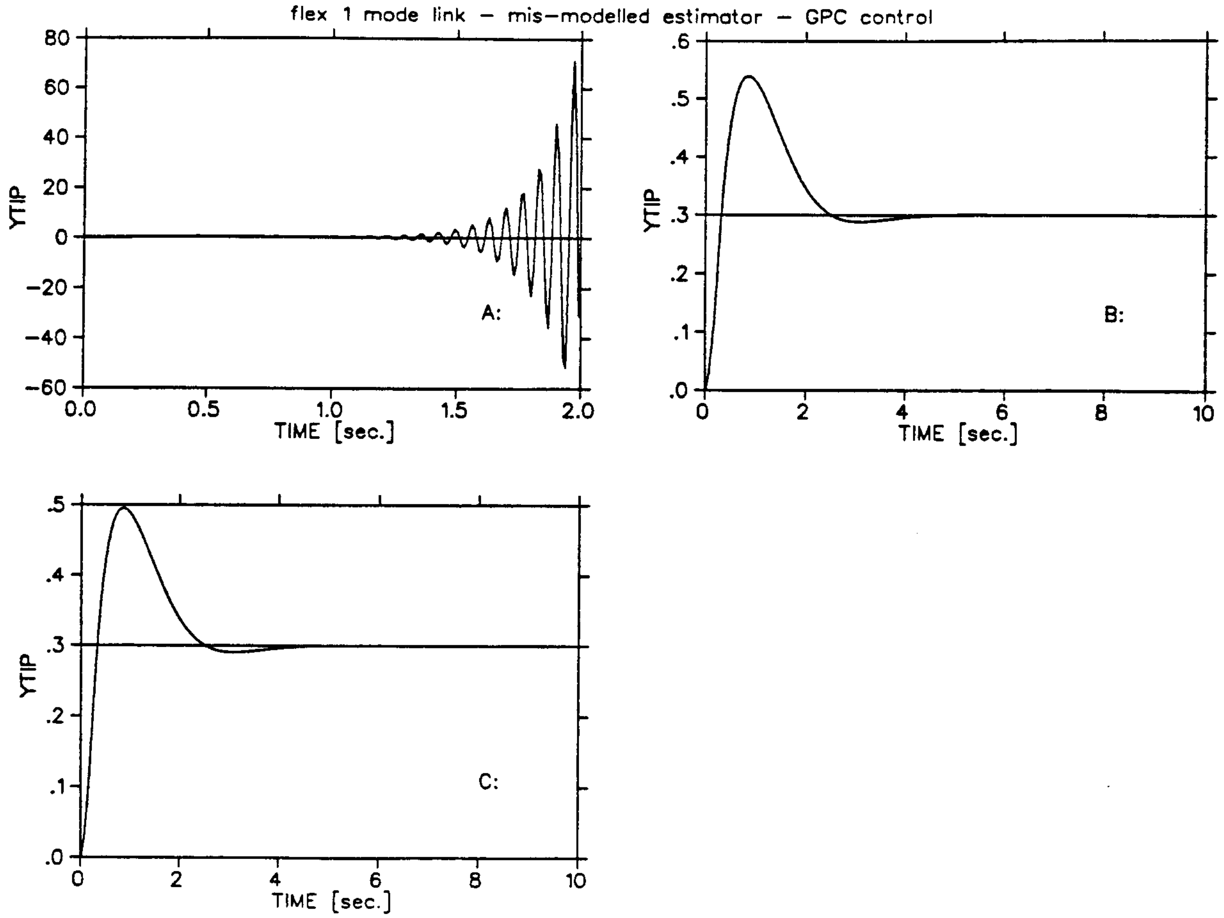

Figure 6.10: The output behavior of a one mode flexible link estimated initially with an order 2 model

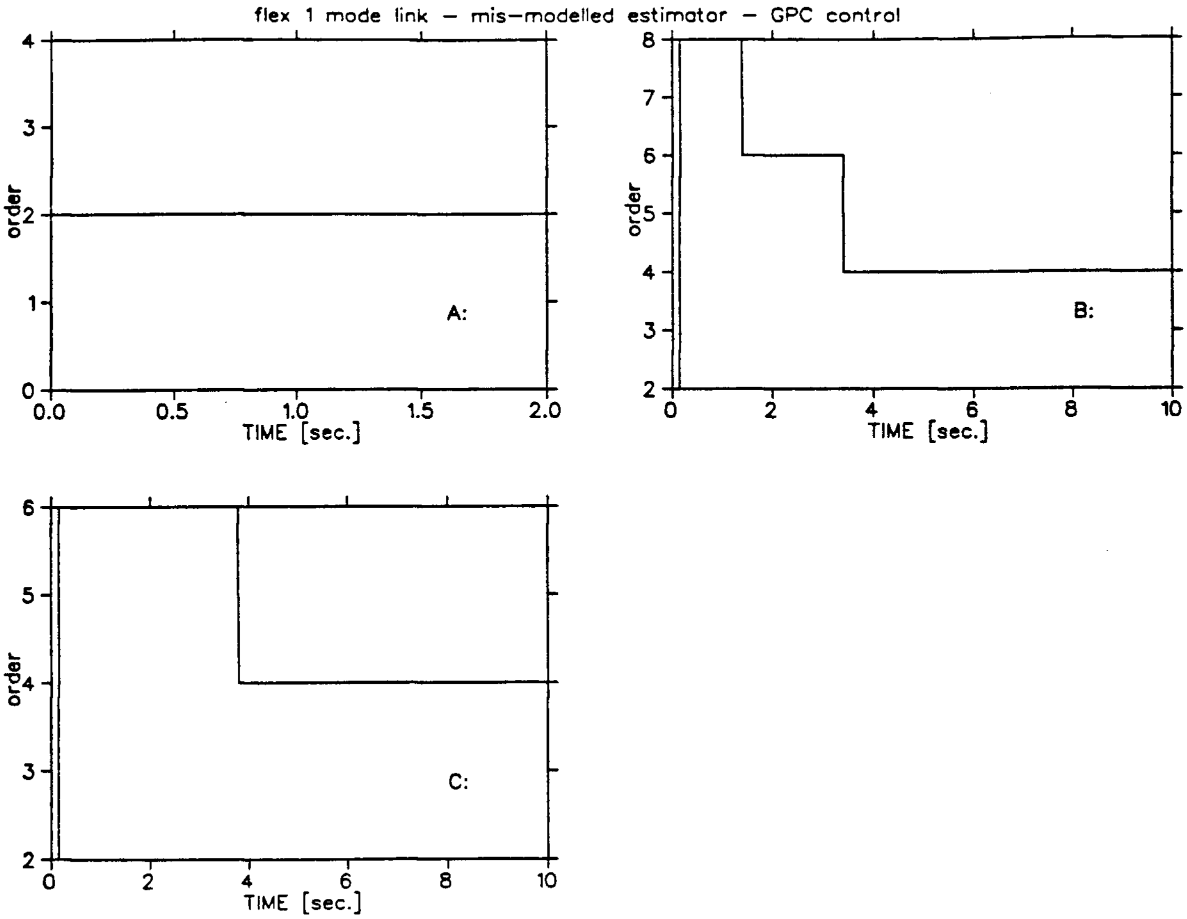

Figure 6.11: Order changes of the estimated model for a one mode flexible link estimated initially with order 2

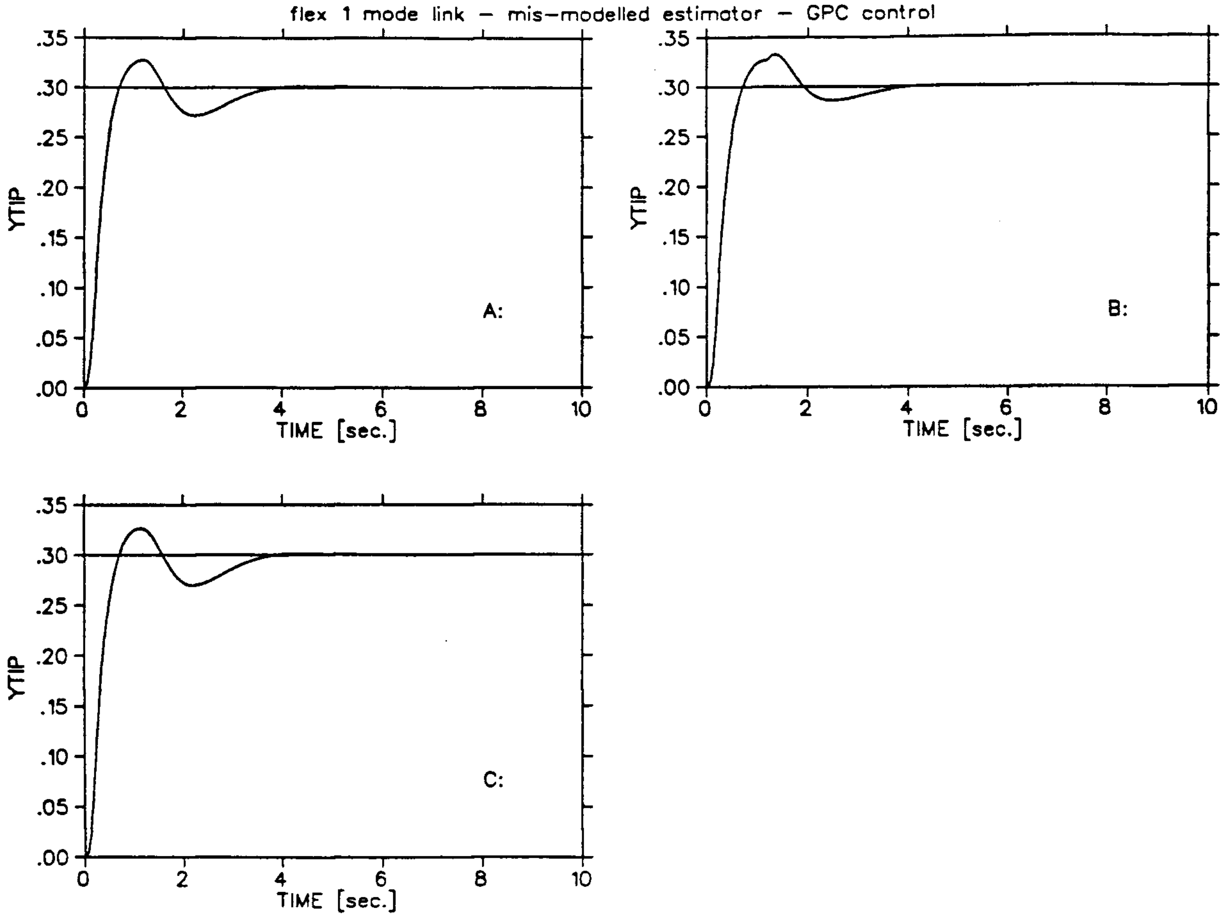

Figure 6.12: The output behavior of a one mode flexible link initially over-modeled

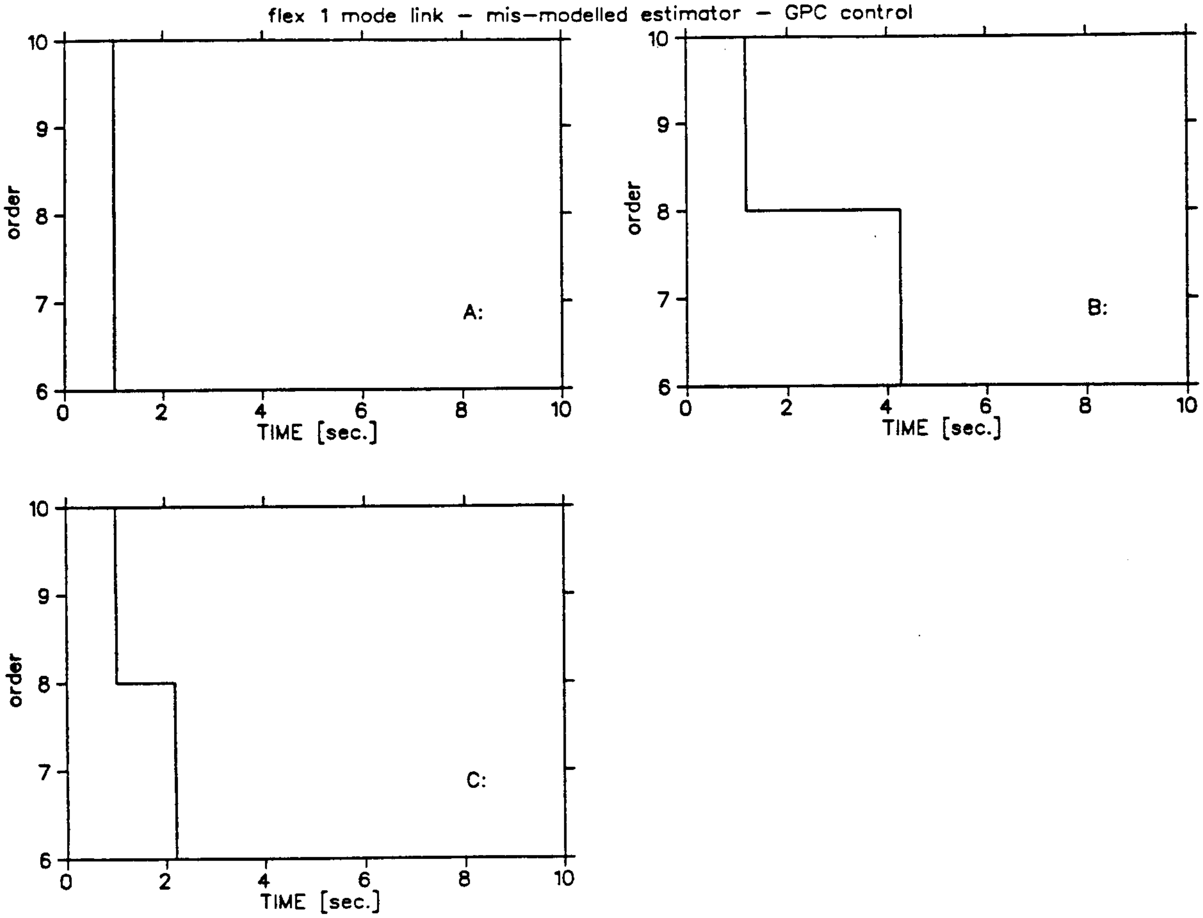

Figure 6.13: Order changes of the estimated model for a one mode flexible link estimated initially with order 10

6.1.2 The Method For The Two Link Manipulator With Hydraulic Actuators

Data for the Two Link Manipulator

The values for the parameters presented in this section are based on the investigation done in Section 5.2 for the behavior of the cost function of the Two Link Manipulator. The values for $T_{C_{min}}$, $T_{C_{max}}$, J_{R_m}, $\dot{J}_{R_m}$, J_{Z_m}, $\dot{J}_{Z_m}$, J_{Z_nl}, $\dot{J}_{Z_nl}$ were determined from correct modeling results shown in Figure 5.17 and Figure 5.18. Values for J_{R_U}, $\dot{J}_{R_U}$ and T_U are from data based on under-modeling, Figure 5.19. Data for the over-modeling case, J_{R_o}, $\dot{J}_{R_o}$, J_{Z_o}, $\dot{J}_{Z_o}$, was obtained from Figure 5.21 and Figure 5.22.

1. Time data

   (a) $T_{C_{min}} = 2.0[sec.]$

   (b) $T_{C_{max}} = 3.5[sec.]$

   (c) $T_U = 1.0[sec.]$

2. Data for correct modeling

   (a) $J_{R_m} = 5 \cdot 10^{-9}$

   (b) $\dot{J}_{R_m} = 5 \cdot 10^{-9}$

   (c) $J_{Z_m} = 5 \cdot 10^{-9}$

   (d) $\dot{J}_{Z_m} = 10^{-14}$

   (e) $J_{Z_nl} = 10^{-8}$

   (f) $\dot{J}_{Z_nl} = 10^{-8}$

3. Data for under-modeling

   (a) $J_{R_U} = 10^{-2}$

(b) $\dot{J}_{R_U} = 10^{-2}$

4. Data for over-modeling

   (a) $J_{R_O} = 10^{-8}$

   (b) $\dot{J}_{R_O} = 10^{-4}$

   (c) $J_{Z_O} = 10^{0}$

   (d) $\dot{J}_{Z_O} = 10^{0}$

5. Data for other parameters

   (a) $DUM = 1$, (2) , (3)

   (b) $DOM = 1$, (2)

Results for the Two Link Manipulator

The data for the hydraulic manipulator when modeled correctly, show, as in the previous application, a typical behavior of the cost function and its slope where the values are relatively small. Here too, the changes are more rapid for the under-modeling case than for the over-modeling case. Figure 6.14, Figure 6.15, Figure 6.16, Figure 6.17 present 3 cases. Case A is the one where the order determination algorithm is not activated. Link 2 is initially estimated with a correct order 3 model and is well controlled to follow a set point. Link 1 is initially under-modeled with a second order model, so that the output does not detect the set point. In cases B and C, the order determination algorithm is activated to detect the correct model and control the system. The difference between the two cases is in the parameters used for the order determination algorithm. In Case B, $DUM = 1$ and $DOM = 1$, so the under-modeling when detected, is corrected to the correct order in the first try. The result is that θ_1 converges to the set point in, about 7 seconds in comparison with 5.5 seconds for the case where both links are structurally correctly modeled, as described schematically in Figure 5.3 . In case C, DOM has changed to the value of 2. When under-modeling is detected the algorithm changes to the order 4, over-modeling at first and then reducing it. The result is even slower than in case B (about 9 seconds to converge), but the set point is tracked, in contrast to case A.

The next three cases present the behavior of the hydraulic system when over-modeled. Figure 6.18, Figure 6.19, present the results. Case A shows the behavior of the system where link 1 is correctly modeled with a third order system and link 2 is over modeled ($order = 5$). θ_1 does not track the set point. In case B, link 2 is initially under-modeled ($order = 2$). When the order detection algorithm is activated, $DOM = 3$ brings the system to over-modeling ($order = 5$), which gradually is brought down to the correct value ($order = 3$). As a result, the output in the first 3 seconds goes in the unstable direction and then stabilizes on the set point. In case C, link 2 in initially over-modeled ($order = 5$), and gradually the order is changed to the correct one. The output stabilizes faster (6.5 seconds, as compared to 9.5 seconds).

The hydraulic actuated manipulator, like the flexible link, when mis-modeled can be brought to the desired results with the order determination algorithm. Again, under-modeling affects the response of the output more than over-modeling and is more difficult to control, but both are solved with the order determination algorithm. Figure 6.20 presents results similar to those presented in case B of Figure 6.18, but on a larger time scale. Once the mis-modeling is detected and corrected, the system will persist with the suitable estimated model structure and will yield the desired response.

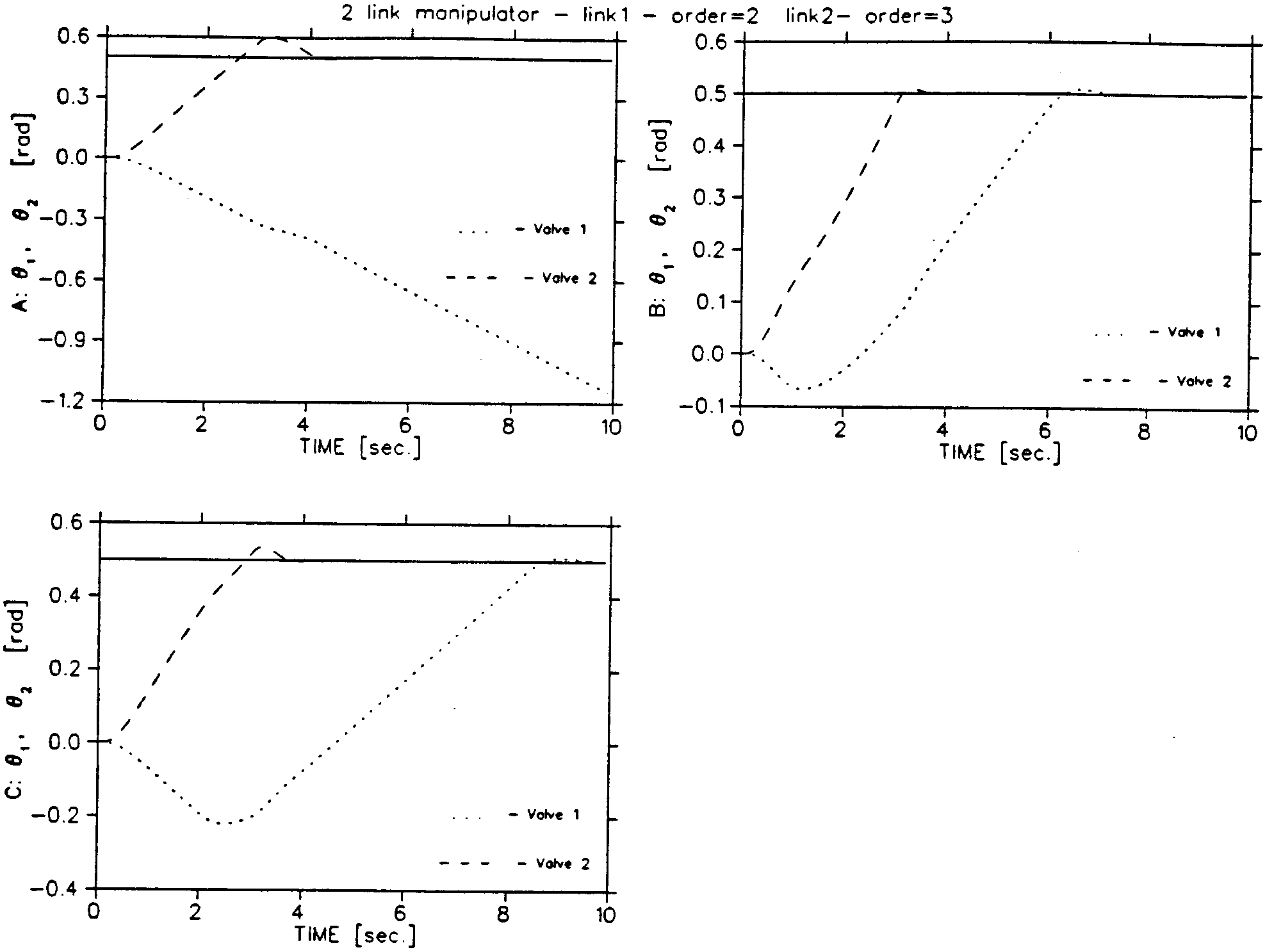

Figure 6.14: The output behavior of a hydraulic actuated two link manipulator initially under-modeled

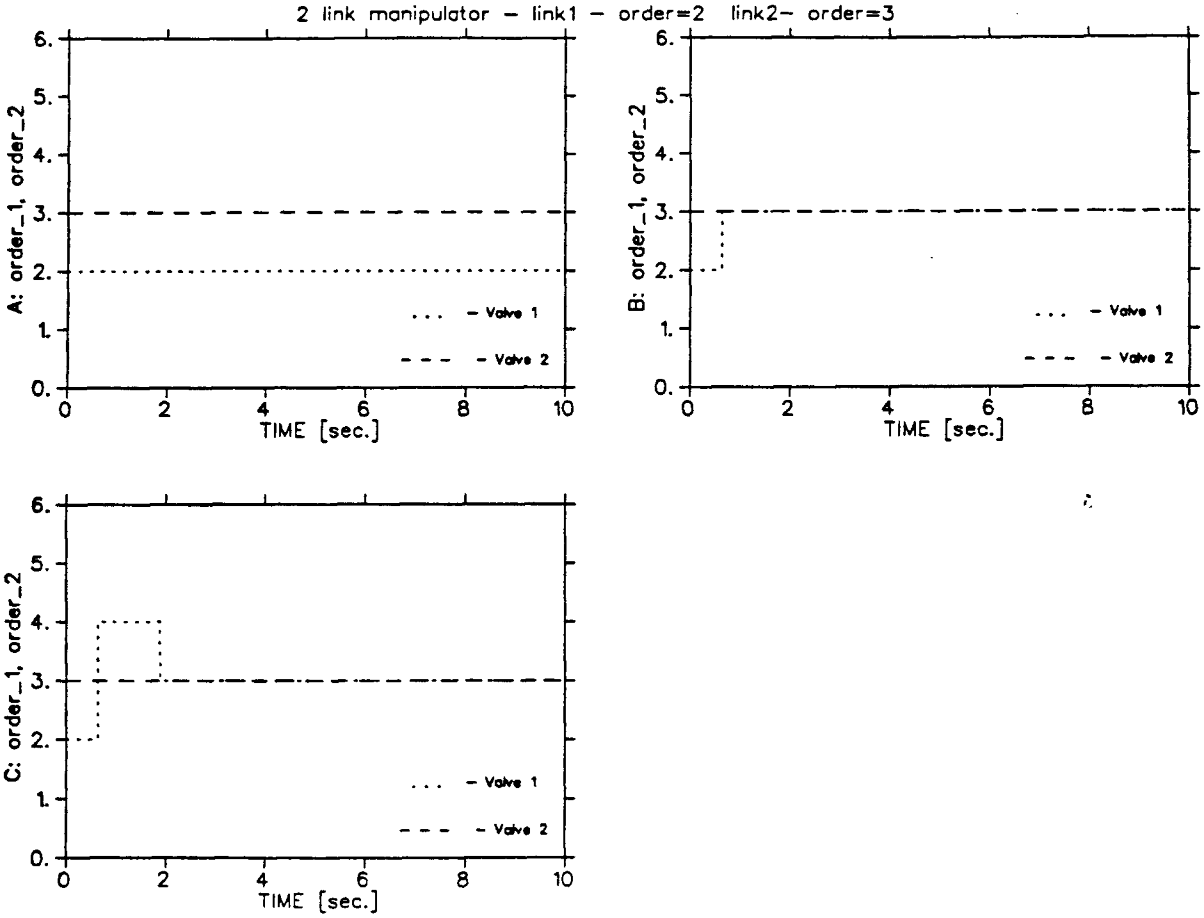

Figure 6.15: Order changes of the estimated model for hydraulic actuated two link manipulator initially under-modeled

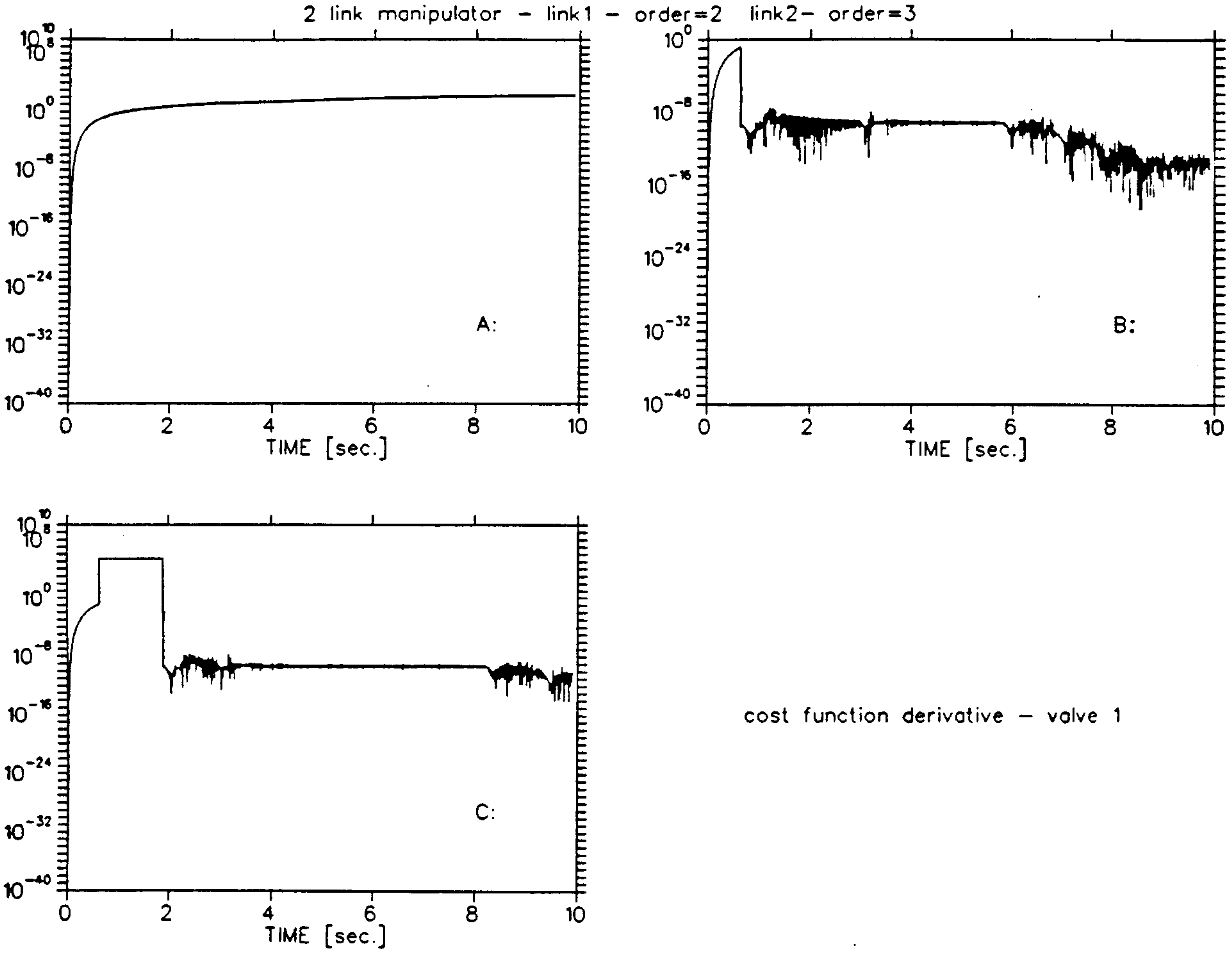

Figure 6.16: The cost function derivative behavior of link1 initially under-modeled

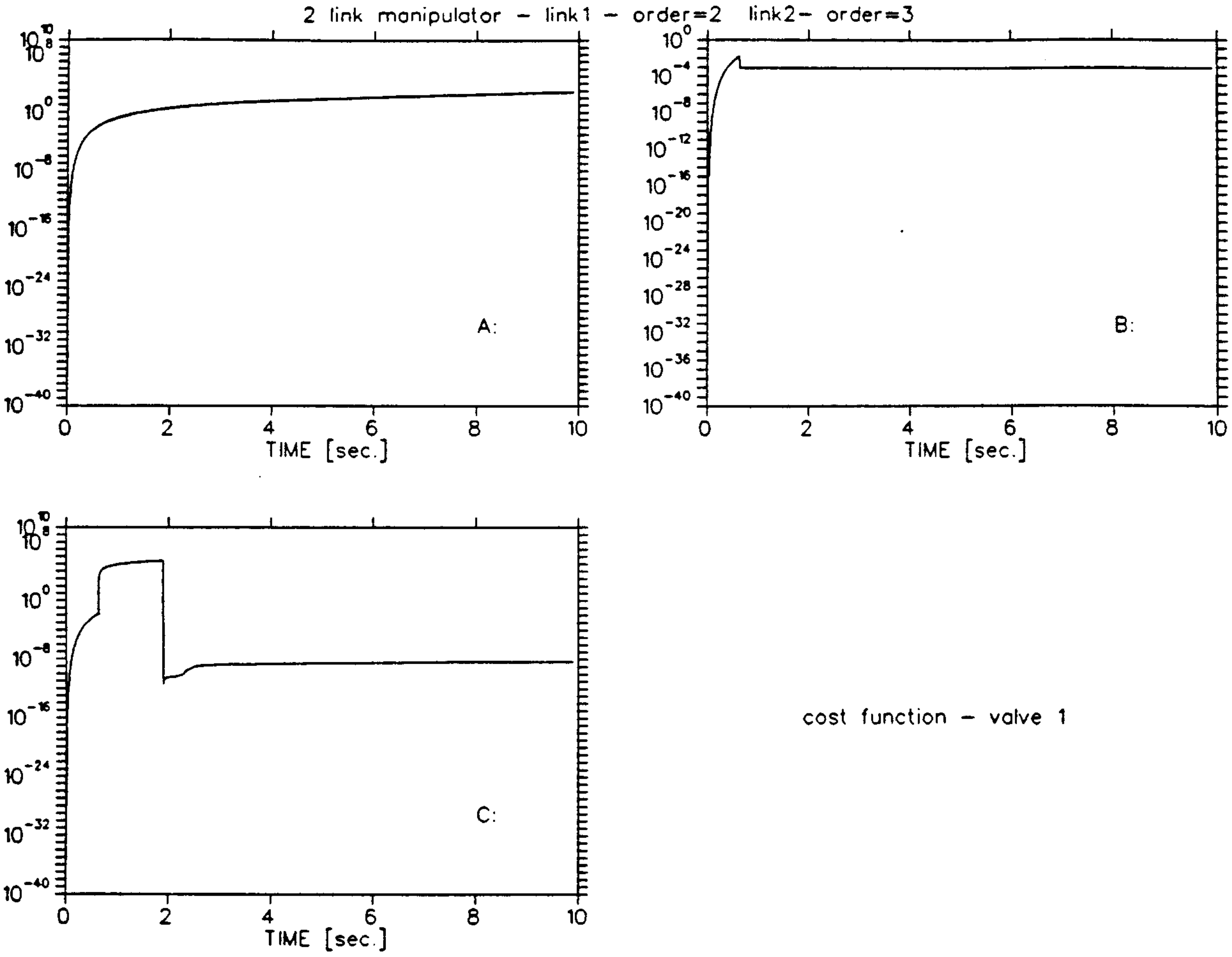

Figure 6.17: The cost function behavior of link1 initially under-modeled

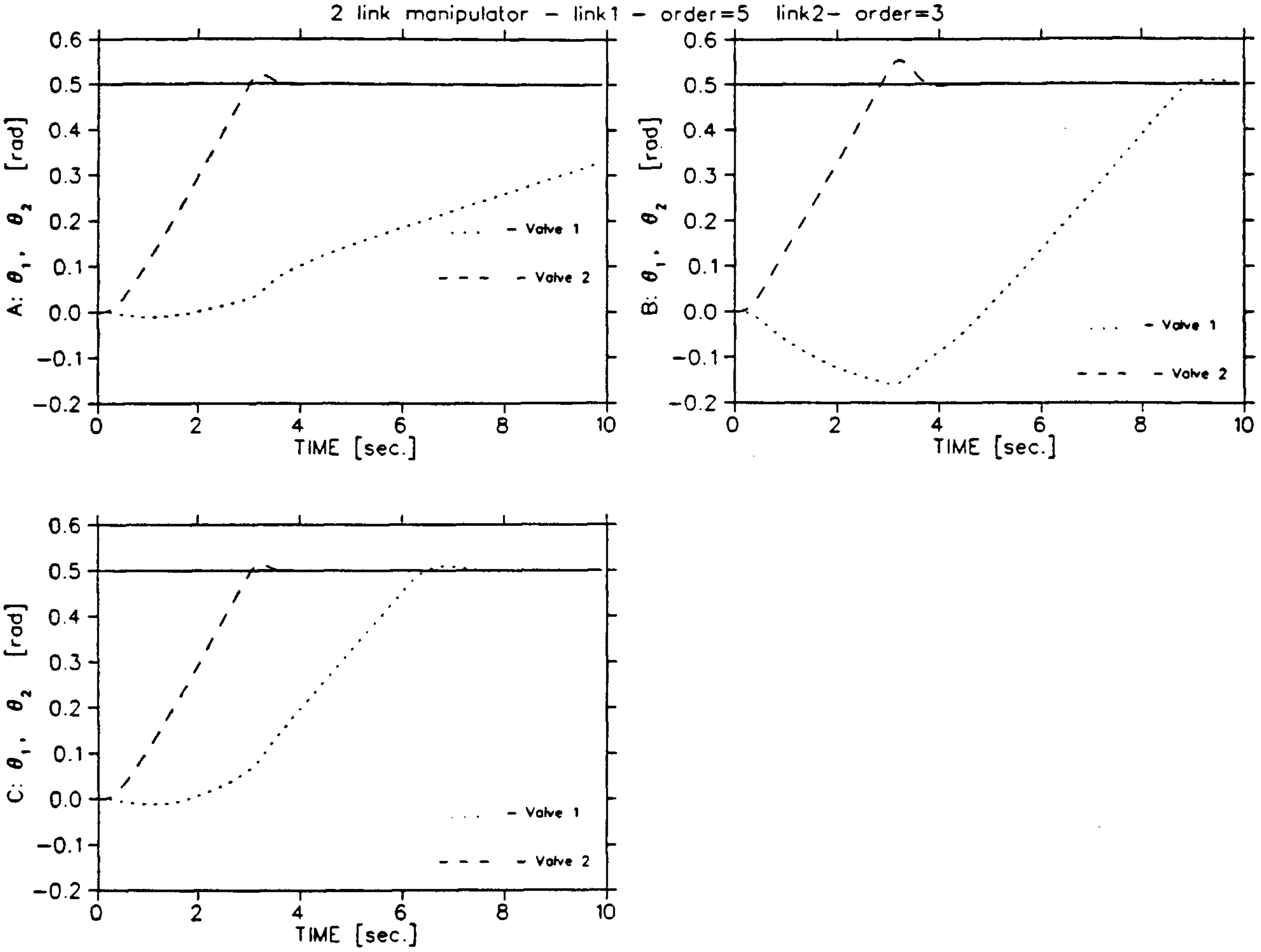

Figure 6.18: The output behavior of a hydraulic actuated two link manipulator Over-modeled

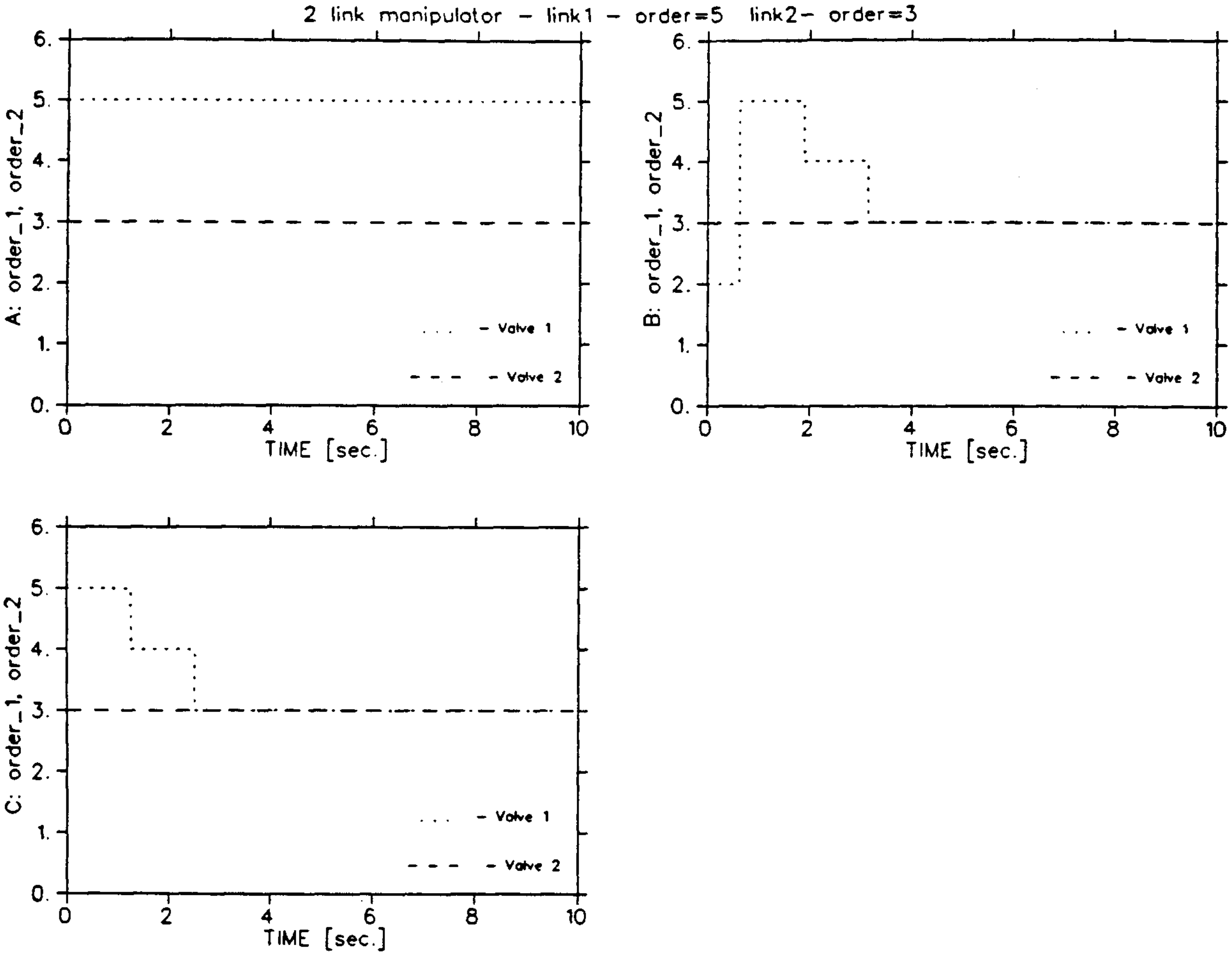

Figure 6.19: Order changes of the estimated model for hydraulic actuated two link manipulator over-modeled

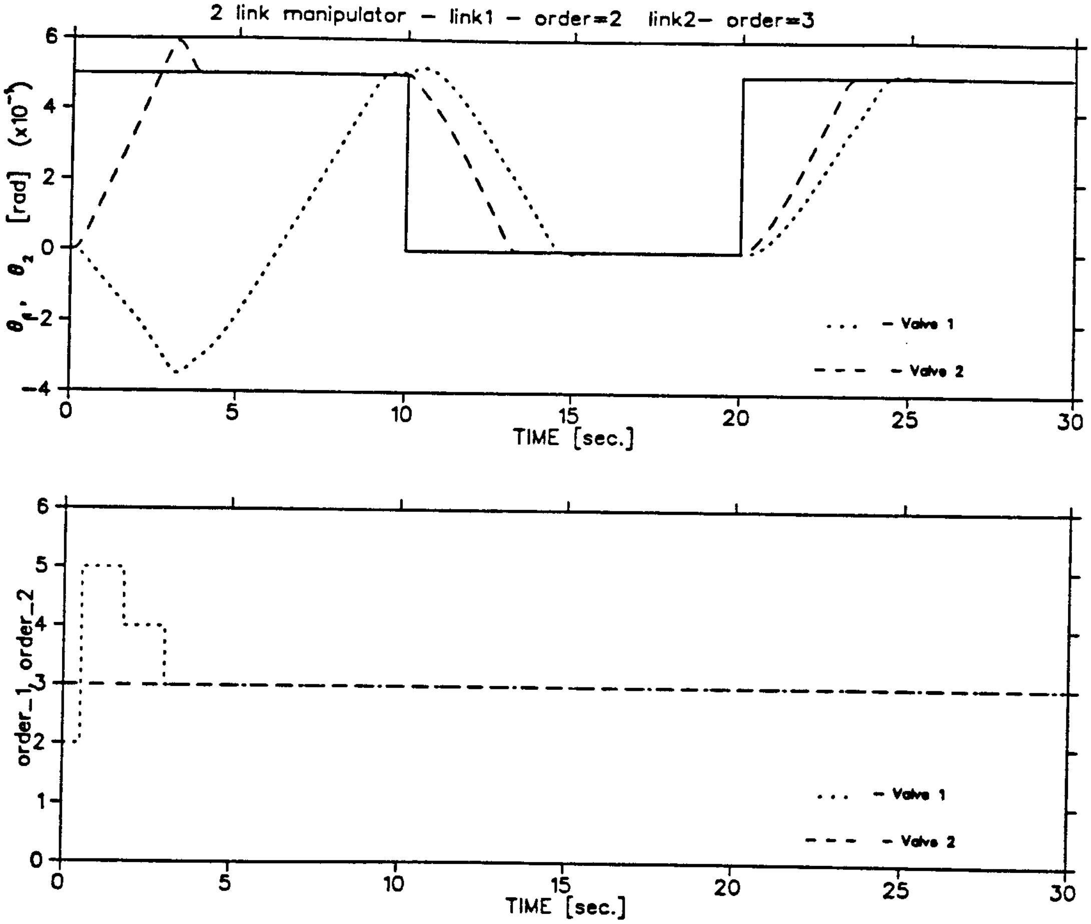

Figure 6.20: Performance of a system initially mis-modeled on a larger time scale

6.2 Comparison of method's Results with Other Work

The method developed in this thesis, detects, determines and executes on-line changes in model order. The MOD algorithm is a gradient algorithm based on the behavior of a chosen cost function and its time derivative. The cost function behavior enables the algorithm to distinguish between types of mis-modeling of the system (under and over-modeling). When mis-modeling creates problems in controlling the system it changes the model's order. Initialization of the MOD algorithm is based either on a priori knowledge of the system and simulation results or on preliminary tests of the system.

The control strategy for a system is designed based on the best knowledge of the system available. A model order change will occur if an operating point change on-line and therefore the conditions change or if the initial identification of the system was not accurate. The MOD algorithm activates a model structure change only when the cost function indicates that mis-modeling is a problem. The computational burden of this algorithm is relatively small, and by using it the desired behavior of the system is achieved.

A discussion of the advantages and disadvantages of existing techniques will now be presented. A more detailed discussion of the methods, has been reviewed in Sections 2.4 and 2.5.

Identifying a system depends strongly on the choice of the model structure. Off-line methods have the advantage of choosing a model structure, identifying its parameters and then validating the model. If the results are not satisfactory, different types of models can be examined to find the best model for the system and the operating conditions. Off-line model validation techniques were published by Akaike [27], Isermann [20], Schwarz [28] and others. These works propose different criteria as a measure of the fit of the model. As well, different methods for state space representation have been developed. Canonical structures were proposed by Guidovzi [33] [34]. Davison [38] presented a method for model order reduction by selecting the dominant eigenvalues and eigenvectors of the system. A minimal description of a system with all significant dynamics is a basis for designing a controller, but the method does not address the question of changing conditions and therefore changes in the number of dominant eigenvalues.

There are also several recursive methods. Overbeek and Ljung [36] suggested the model structure selection (MSS) algorithm in which the structures differ in the parametrization of the model, but there is no recursive order selection that is, the order is chosen a priori. The algorithm calculates at each time the entire set of structures and compares them on-line resulting in a possibility of a high computational burden. Niu, Xiao and Fisher [39] present a simultaneous recursive estimation of parameter and order. The order is found by calculating a cost function for all possible orders up to a known upper bound. The order which corresponds to the minimal value of the cost function is the one that is used. Further work By Niu and Fisher [40] implemented the above algorithm for MIMO sestems. Hemerly [47] presented a method for on line order and parameter identification using the RLS algorithm and the PLS criterion. Mereiros and Hemerly [49] integrated the above method with lattice form filters for a minimum variance controller. This work is the closest in nature to the work presented in this thesis. However, it requires computation of a cost function for all possible orders (up to an upper bound), at each time step. There is no reference or discussion of the type of mis-modeling. Other recursive and off-line order and parameters identification methods were reported (open loop methods), such as the one by Wulich and Kaufman [42] and Katsikas [43]. These methods are based on a priori defined criterion, and calculations of all possible orders and the choice of the one which gives the best performance of the criterion.

There are several advantages to the MOD method presented in this thesis:

1. The method is an on-line method that detects the need to change the model's order and implements it while using well known methods for the indentification and adaptive

control processes. Most of the published works that were presented in Section 2.5 discuss model order determination, either off-line or on-line but open loop, with no control algorithm implemented.

- Off-line methods can estimate a model then check and validate it and if the results do not satisfy, another model structure is chosen, until a good representing model is achieved. If a model is chosen with one of the off-line methods its structure is fixed when used on-line, for control purposes for example. Representing works can be found in Akaike [26], Isermann [20], Schwartz [28], Rissanen [29], Guidovzi [33] [34] and Davison [38].
- Recursive identification methods and structure selection are mainly parameter selection methods where the order is fixed. The possible structures are scanned and the best chosen, Overbeek and Ljung [36], or simultaneous order and parameter estimation with the same principle, a set of possible orders are chosen and a cost function is calculated for all the set each time step. The chosen order is the one that correspond to the minimal value of the cost function. See Niu, Xiao and Fisher [39], Niu and Fisher [40], Wulich and Kaufman [42], Katsikas [43] and Hemerly [47].

2. Cost function behavior indicates the best estimated model order to the MOD algorithm.

   - It was found that the cost function has a different behavior for under and for over-modeling, but similar behavior for the two applications .
   - The cost function indicates the best possible order for the present operating point and does not search for the exact model.
   - An initial order is provided and there is no need to assume on upper bound to the order.

- The cost function behavior indicates indicates the influence of the closed loop, especially in the mis-modeling cases.
- There is no need to examine and scan a set of models for each time step as the cost function is calculated for the present model order.

The work done by Hemerly [49], combines a recursive identification process and recursive order estimation for a model represented by lattice filter form with a minimum variance control algorithm. The upper bound of the order is assumed to be known and the order is estimated by scanning all possible PLS functions and choosing the order that corresponds to the minimal value.

To the best of our knowledge, there is no other method like the one presented in this thesis. Based on the time behavior of a cost function the algorithm detects and executes on-line changes of model order. It was established that both, under and over-modeling can cause poor performance and instability and for both, order correction is done if required. The algorithm detects changes in the cost function behavior which is monitored on line. The computation burden is fairly small and the algorithm has stable characteristics and is able based on some a priori knowledge within a few iterations to maintain desired performance of the system.

6.3 Conclusions

This chapter presented the behavior of two kinds of robotic manipulators controlled with the GPC algorithm in mis-modeling of the estimated model for the control algorithm. The behavior of the cost function and its derivative for the one and two mode flexible link and for the two link hydraulic actuated manipulator showed a pattern of behavior for under, over and correct structure for the estimated model. Based on these results, an order determination algorithm was presented. Depending on its parameters, mis-modeling can be corrected to give an acceptable response from these systems, even when they are initially unstable or have bad performance.

Chapter 7

CONCLUSIONS AND SUMMARY

7.1 Main Results of the Thesis

This work has challenged the concept of using a fixed structure model for a plant controlled by an adaptive control algorithm (GPC). Generally, in order to implement an adaptive algorithm for a system, the plant is modeled by a linear model in which parameters are estimated on-line. This can result in uncertainties in parameter values, especially when the model order is incorrectly chosen.

Two robotic applications chosen for the study were modeled, simulated and controlled. The single flexible link can give rise to modes of oscillations on-line during a working cycle, and therefore have on-line changes of the plant dynamics. The two link manipulator with hydraulic actuators can be mis-modeled, but it was also chosen because of its highly nonlinear nature and its extensive use in industry.

Chapters 3 and 4 present the dynamic modeling of the systems , implementation of the GPC algorithm and the tuning of the control parameters to achieve good performance control. Chapter 5 presents a model order determination (MOD) algorithm for detecting the need to change the model structure, correcting the order and executing it on-line. The study that led to the above method began with the establishment of a cost function as a measure of the error between the plant and model dynamics.

The main results of this study are:

- A cost function for the modeling error was studied and it was found that for correct modeling, the cost function rises initially (when the error goes to zero), and then settles on a constant value. When under-modeled, the cost function's initial rise is steeper, going to much higher values and leading to instabilities. When over-modeled, the behavior is much more moderate, but the performance deteriorates and the system can go unstable. When under-modeling is involved, the regression vector is correlated to the unmodeled dynamics. However, the over-modeling does not include that correlation and therefore the response is more moderate. The excess of dynamics in the model (over-modeling) causes the control algorithm to try and control dynamics that are not there; thus, the control parameters are no longer well tuned.

- Both applications show similar cost function behavior.
- Based on the above results, an on-line model order determination (MOD) algorithm is given to detect the need to change the model order and to correct it on on-line.
- Results from implementing the method on both applications show that for under and over-modeling, instabilities are avoided and desired performance is restored.
- Generalized Predictive Control (GPC)can be applied to heavy duty manipulators which are highly nonlinear systems. The hydraulically actuated heavy duty manipulators are used extensively in large resource based industries, and any improvement in efficiency may result in major financial benefits. Therefore, the results in Chapter 4 that advance the state of the art will be stated next (Kotzev et at [57]):
  - This work examined the effect of nonlinearities in the application of GPC to a wide range of hydraulically actuated manipulators.
  - Special attention is given to the maximum output horizon. The work introduces an on-line automatic change of the maximum output horizon so that the transient

response can be sufficiently fast and undesirable overshoots avoided. The selection of other GPC design parameters is also addressed.

- Experimental results from an open loop experiment on a heavy duty manipulator, a 215B Caterpillar, indicate that the cost function behavior in open loop does not vary strongly enough for mis-modeling to be reliably determined. This has also been verified by numerical simulations with other applications. A closed loop approach is needed.

7.2 Suggestions for Future Work

The goal of this thesis was to study the behavior of a system controlled with an adaptive control algorithm when plant mis-modeling occurs, understand the behavior, and suggest a method to overcome problems that arise, such as poor performance and instability. Such a study was conducted and a method that provides good results is presented. The scope of the investigation can be made broader for further generalization of the results:

- More adaptive algorithms, predictive and non-predictive, should be tried.

- More systems should be checked, not only from the robotic family. This could generalize the conclusions on the behavior of those systems in mis-modeling and may come up with parameters to characterize it. For example the time constant of the system could have an influence on the results.

- A closed loop experiment with the order determination method implemented should be done with a $215B$ Caterpillar or a similar system to find advantages and disadvantages, since the aim is to implement it for industrial use.

- More attention should be given to other possible order determination methods and their performance should be compared with the present one.

- The sensitivity of the identification algorithm and its influence on the cost function should be checked. Othe indentification algorithms should be considered, shuch as the Householder transform ^{64}, which is reported to be numerically stable, and able to determine directly from signals, the correct regression order at a given time instant.

Bibliography

[1] K.J. Astrom and B. Wittenmark. *Adaptive Control.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[2] R.H. Middleton and G.C. Goodwin. *Digital Control and Estimation.* Prentice Hall, New Jersey, 1990.

[3] L. Ljung. *System Identification Theory for the User.* Prantice-Hall, Inc., Englewood Clifs, N.J., 1987.

[4] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control — part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[5] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control — part ii. extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[6] L. Ljung and T. Soderstrom. *Theory and Practical Recursive Identification.* The MIT Press, Cambridge, Massachusetts, 1987.

[7] K.J. Astrom and B. Wittenmark. On self tuning regulators. *Automatica*, 9:185–199, 1973.

[8] K.J. Astrom, U. Borrison, L. Ljung, and B. Wittenmark. Theorey and application of self tuning regulators. *Automatica*, 13:457–476, 1977.

[9] Y.D. Landau. *Adaptive Control - the Model Reference Approach.* Course of Theoretical Physics. Macel Dekker, Inc., 1979.

[10] B. Edgar. *Stability of Adaptive Controller.* Springer-Verlag, Berlin Heidelberg, New-York, 1979.

[11] K.J. Astrom and B. Wittenmark. Theorey and application of self tuning regulators. *Automatica*, 13:457–476, 1977.

[12] D.W. Clarke and P.J. Gawthrop. Self-tuning controller. *IEE Proceedings, Part D*, 122:929–934, 1975.

[13] J.M. Edmunds. *Digital Adaptive Pole Shifting Regulators.* Ph.D. Dissertation Control System Center, University of Manchester, England, 1976.

[14] R.L. Borrison and R. Iserman. Self tuning regulators for a class of multivariable systems. *Automatica*, 15:209–215, 1979.

[15] K.J. Astrom and B. Wittenmark. *Computer Controlled Systems — Theory and Design.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

[16] K.S. Fu, R.C. Gonzolez, and C.S.G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence.* McGraw-Hill Inc., New York, NY, 1987.

[17] M. Tomizuka, R. Horowitz, G. Anwar, and Y.L. Jia. Implementation of adaptive techniques for motion control of robotic manipulators. *asme:jbe*, 110:63, March 1988.

[18] H. Demircioglu and P.J. Gawthrop. Continuous–time generalized predictive control (gpc). *Automatica*, 27(1):55–74, 1991.

[19] D.J. Latornell and D.B. Cherchas. Force and motion control of a single flexible manipulator link. *Robotics and Computer Integrated Manufacturing*, April 1991. Accepted for publication.

[20] R. Isermann. Practical aspects of process identification. *Automatica*, 16:575–587, 1980.

[21] K.J. Astrom and P. Eykhoff. System identification a survey. In *Proceedings IFAC Symp. on Identification and Process Parameter Estimation*, pages 890–896, 1970.

[22] K.J. Astrom. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.

[23] V. Strejc. Least squares parameter estimation. *Automatica*, 16:535–550, 1980.

[24] G.C. Goodwin and R.Z. Payne. *Dynamic System Identification*. Academic-Press, New-York, 1977.

[25] C.E. Rohrs, L. Valavani, M. Athans, and G. Stein. Robustness of continuous time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Transactions On Automatic Control*, AC-30:881–889, 1985.

[26] h. Akaike h. Fitting autoregressive models for prediction. *Ann. Inst. Statist Niath*, 21:243–247, 1969.

[27] h. Akaike. Information theory and extension of the maximum likelihood principle. In *Proceedings Int. Symp. Information Theory, Supp to Ptoblems of Control and Information Theory*, pages 267–281, 1972.

[28] G. Schwarz. Estimating the dimension of a model. *Stat.*, 6:2, 1978.

[29] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[30] h. Akaike. On entropy maximisation principle. *App. of Statistics*, pages 27–41, 1977.

[31] E.J. Hannan and B.G. Quinn. The determination of the order of an autoregresive model. *Jr. Statist. Soc.*, b(41):190–195, 1979.

[32] J. Zhang L. Guo H. Chen. Consistent order estimation of linear stochastic feedback control systems. *Automatica*, 25(1):147–151, 1989.

[33] R.P. Guidovzi. Canonical structures in the identification of multivariable system. *Automatica*, 11:117–133, 1975.

[34] R.P. Guidovzi. Invariant canonical forms for system structural and parametric identification. *Automatica*, 17(1):117–133, 1981.

[35] A. Gauthier and I.D. Landau. On recursive identification of multi output systems. *Automatica*, 14:609–614, 1978.

[36] A.J.M. Overbeek and L. Ljung. On line structure selection for multivariable state-space models. *Automatica*, 14:529–543, 1982.

[37] P.A.J. Nagy and L. Ljung. Computer-aided model structure selection. In *Proceedings of the 1991 IFAC International Symposium on Intelligent Tuning and Adaptive Control*, pages 918–923. IFAC, 1991.

[38] E.J. Davison. A method for simplifying linear dynamic systems. *IEEE Transactions on Automatic Control*, AC-11(1):93–101, January 1966.

[39] S. Niu, D. Xiao, and D.G. Fisher. A recursive algorithm for simultaneous identification of model order and parameters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(5):884–886, May 1990.

[40] S. Niu and D.G. Fisher. Mimo system identification using augmented ud factorization. In *Proceedings of the 1991 IFAC International Symposium on Intelligent Tuning and Adaptive Control*, pages 699–703. IFAC, 1991.

[41] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* Academic Press, 1977.

[42] D. Wulich and Y. Kaufman. Order estimation of linear time invariant system. *Journal of the Franklin Institue*, 322(3):143–150, 1986.

[43] S.K. Katsikas, S.D. Likothanassis, and D.G. Lainiotis. A model identification with unknown process order. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(5):872–876, May 1990.

[44] G.e. Birch, P.d. Lawrence, J.C. Lind, and R.D. Hare. Appliction of prewhitening to a ar spectral estimation of eeg. *IEEE Transactions on Biomedical Engineering*, 35(8):640–645, August 1988.

[45] W. Zhou, D.B. Cherchas, S. Calisal, and G. Rohling. Identification of rudder - yaw - roll steering model by using r.p.e. techniques. *Department of Mechanical Engoneering, U.B.C.(to be published)*, 1989.

[46] C.C. Zervos and G.A. Dumont. Deterministic adaptive control based on laguerre series representation. *Int. J. Control*, 48:2333–2359, 1988.

[47] E.M Hemerly. Order and identification of industrial processes using the pls criterion. In *Proceedings of the 1991 IFAC International Symposium on Intelligent Tuning and Adaptive Control*, pages 293–297. IFAC, 1991.

[48] J. Rissanen. A predictive least squares principle. *IMA J. of Math. Control and Information*, 3:211–222, 1986.

[49] A.A.D Mereiros and E.M Hemerly. Latice form in adaptive control with recursive order estimation. In *Proceedings of the 1991 IFAC International Symposium on Intelligent Tuning and Adaptive Control*, pages 1083–1087. IFAC, 1991.

[50] R.H. Cannon and E. Schmitz. Precise control of flexible manipulators. *Robotics Research*, pages 841–861, 1983.

[51] J.J. Craig. *Introduction to Robotics Mechanics and Control.* Addison-Wesley Publishing Company Inc., Reading Massachusetts, 1986.

[52] H.E. Merritt. *Hydraulic Control Systems.* John Wiley and Sons, Inc., New York, 1966.

[53] P. Vaha. Application of parameter adaptive approach to servo control of a hydraulic manipulator. *Acta Polytechnica Scandinavia, Mathematics and Computer Science*, 51:3–86, 1988.

[54] N. Sepehri, G.A.M Dumont, P.D. Lawrence, and F. Sassani. Cascade control of hydraulic actuated manipulators. *Advanced Systems Institute*, 1990.

[55] C.H. An, C.G. Atkenson, , and J.M. Hollerbach. *Model-Based Control of a Robot Manipulator.* The MIT Press series in artificial intelligence. The MIT Press, Cambridge, Massachusetts, 1988.

[56] H. Asada and j.j.e. Slotine. *Robot Analysis and Control.* John Wiley & Sons, Inc., New-York, 1986.

[57] A. Kotzev, D.B. Cherchas, P.D. Larence, and N. Sepehri. Generalized predictive control of a robotic manipulator with hydraulic actuators. *Robotica - in press.*, 1992.

[58] J. Watton. The dynamic performance of an electro-hydraulic servovalve/motor system with transmission line effects. *ASME Journal of Basic engineering*, 190:14–18, March 1987.

[59] A.A. Bahnasawi and M.S. Mahmoud. *Control of Partially Known Dynamical Systems.* Springer-Verlag, New-York, N.Y., 1989.

[60] P. Eykoff. *System Identification: Parameter and State Estimation.* John Wiley & Sons, Inc., New-York, 1974.

[61] Moog Inc. *Moog Type 30 Flow Control Servo Valves*, volume Catalog 301385. Moog Inc., Aerospace Groop, New York, 1988.

[62] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipies, The Art of Scientific Computing.* cambrige University Press, Cambridge, 1990.

[63] D.J. Latornell. *Force Control for Robotic Manipulators with Structurally Flexible Links.* PhD thesis, Dept. of Mechanical Engrg., University of British Columbia, 1992.

[64] A.O. Steinhardt. Householder transfroms in signal processing. *IEEE ASSP magazine*, 5(3):4–12, July 1988.

[65] N. Sepehri. *Control of a Heavy Duty Hydraulic Machine.* PhD thesis, UBC - The University of British Columbia, The Faculty Of Mechanical Engineering, 1991.

[66] R.L. Bisplinghoff, H. Ashley, and R.L. Halfman. *Aeroelasticity.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1955.

[67] F.S. Tse, I.E. Morse, and R.T. Hinkle. *Mechanical Vibrations Theory and Applications*, volume Second dition. Aiiyn and Bacon Inc., 1980.

Appendix A

Experimental Results for the Hydraulic Actuated Manipulator

A.1 Introduction

At UBC (the University Of British Columbia) an excavator, Caterpillar 215B, which is engaged in a teleoperation project, Sepehri [65] was available to us for some experiments. The goal in the experiment was to identify the dynamic model of the manipulator with open loop, check the behavior of the cost function (Equation 5.1), and compare it with the results of a simulation of a similar manipulator controlled with GPC, as described in Chapter 4 and Chapter 5. The results show that after the cost function becomes flat at high orders, it is easy to choose a mis-matched order for the system. But results in Chapter 5 for the two link manipulator with the hydraulic actuator show that if the order is wrongly chosen, then the instabilities can occur.

A.2 Description of the System

The Caterpillar 215B excavator is a mobile three degree of freedom manipulator. The links are the "Swing", which is the base that rotates. The "Boom" and the "Stick" are two links operated through hydraulic cylinders. The end effector, the "bucket", which is used to dig and carry heavy loads is also operated with a hydraulic actuator. Figure A.1 describes the excavator's structure. The motions of the "Boom" and "Stick" are coupled by cross-over valves, which allow for a faster movement of one link when the other is slower.

The experiment for this work is the identification of the dynamics of the "Boom" and its actuator that were operated. The coupling between the valves was eliminated. The estimation algorithm used was a Recursive Least Squares. The input is the signal from the spool valve, and the output, from the angular position of the link.

A.3 Results

Six runs were made wherein each, the next input characteristics were given:

- S1f - amplitude of 1.0 volt and frequency of 4 seconds.
- S2f - amplitude of 1.5 volt and frequency of 3 seconds.
- S3f - amplitude of 1.5 volt and frequency of 2 seconds.
- S4f - amplitude of 1.3 volt and frequency of 6 seconds.
- R1f - random input, maximum amplitude of 2 volt.
- R2f - random input, maximum amplitude of 3 volt.

Figures A.2 A.3 A.4 A.5 A.6 A.7 present the behavior of the measured output, the estimated model output, and the cost function which is an indication of the error between the two outputs. In all six cases, the drawing of the measured output vs. the model output show very little difference between the two, as does the cost function which grows fast to a value and drifts slowly from it due to the nonlinearities which are not modeled in the linear model for the estimation algorithm. Figure A.8 presents the behavior of the cost function for the four Sif runs. The values of the cost function in all runs was taken after 2000 sampling steps. The results confirm the discussion in Chapter 2 that the cost function for open loop identification will have high values for under-modeling and will reach a plateau for higher

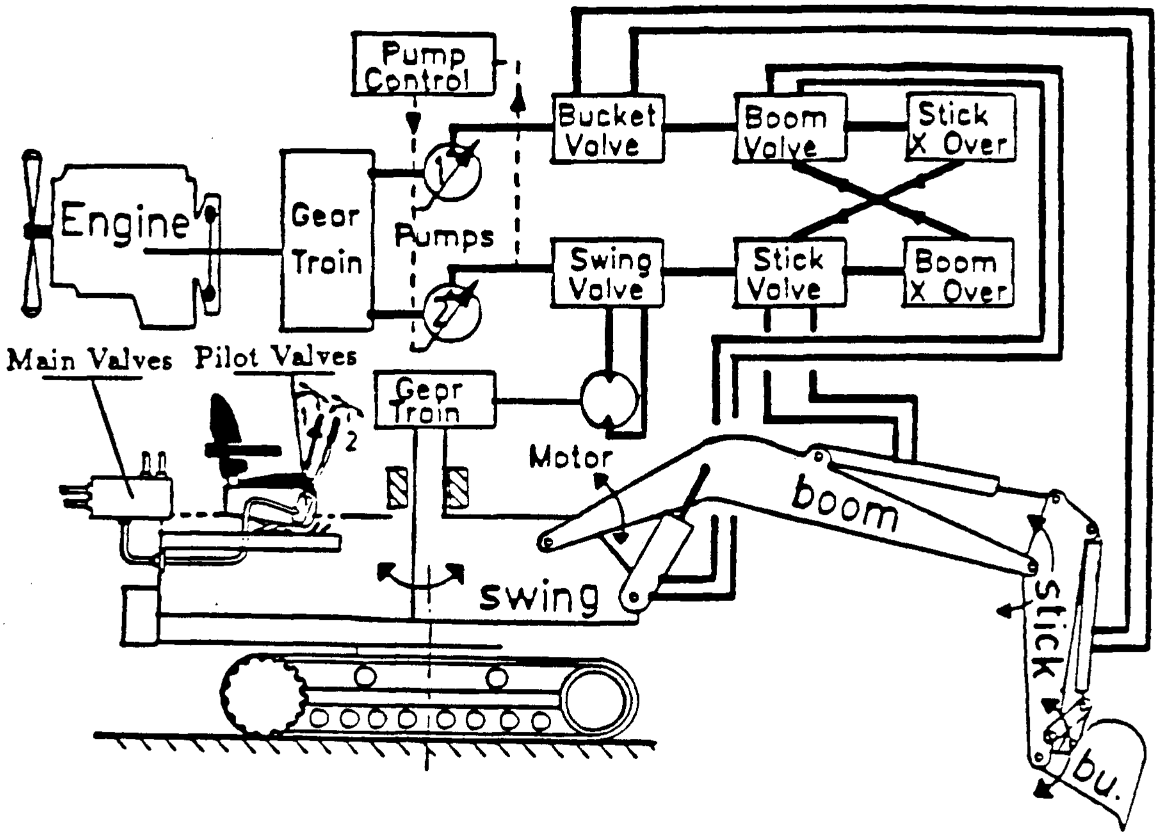

Figure A.1: Caterpillar 215B excavator

orders of the estimated model. The same goes for the Rif experiments, as shown in Figure A.9, the measured and the model output agree with each other quite well, and the cost function, after 2000 sampling steps, again has high values for under-modeling and reaches a plateau for higher order values. In Chapter 5 the cost function for the closed loop algorithm behaves differently. There is a clear difference between the values for under-modeling and correct modeling. For over-modeling, there could also be a clear difference from correct modeling, resulting in instability of the system if left unattended.

A.4 Conclusions

This experiment was done with an excavator which is used extensively in the forest and construction industry. The system is a highly nonlinear one, and it is therefore of interest to check the behavior of the cost function when open loop identification is done. This confirms the fact that even when an identification is done open loop and an over modeled model is chosen the closed loop controlled system may run into performance and stability problems. Thus, there is a need to detect on-line a significant model mis-match as mentioned in Chapter 5. The results show (see Figures A.8 and A.9), that when identification is done in open loop the behavior of the cost function is not such that an error in model structure can be easily determined. Thus a closed loop approach is needed. This has also been shown in simulations in chapter 5.

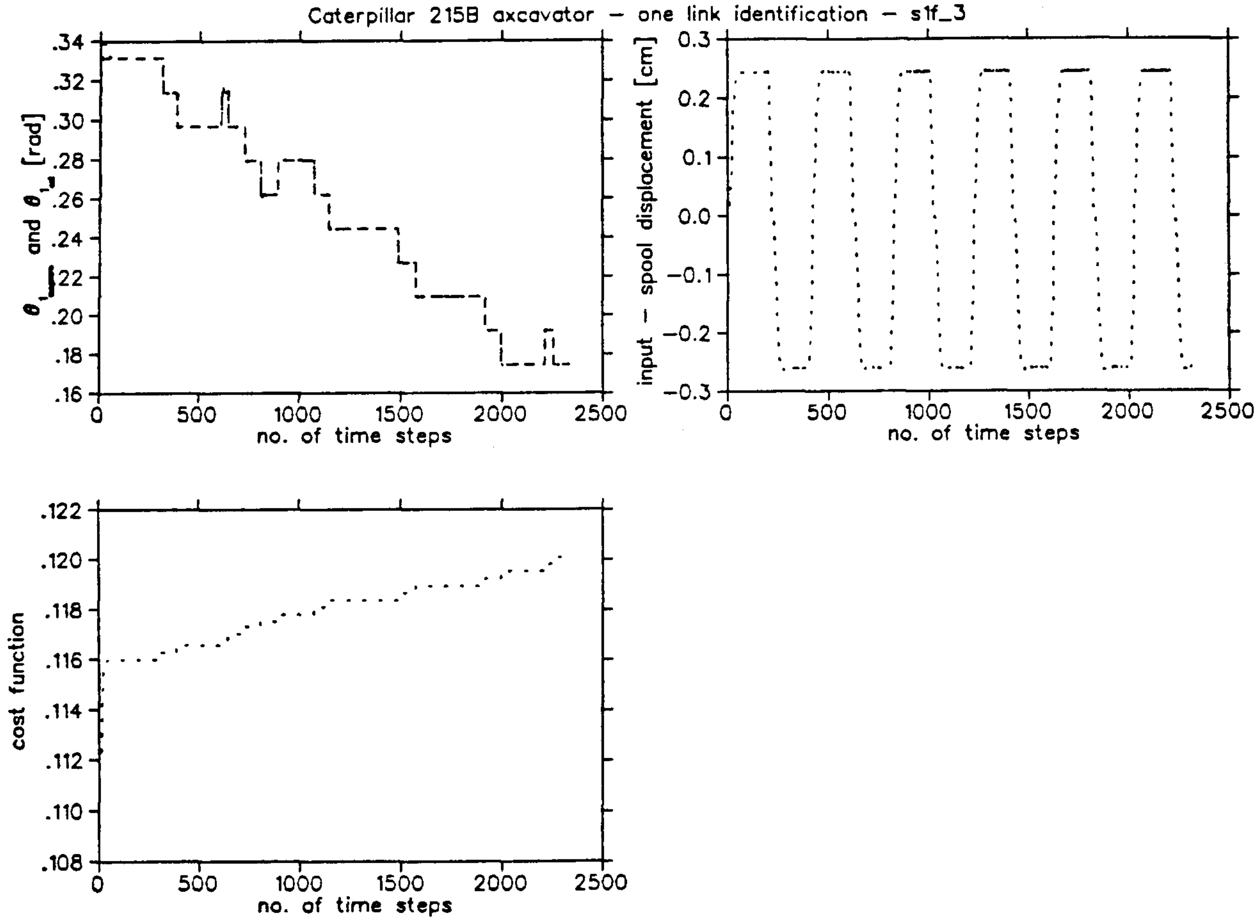

Figure A.2: Input output behavior for S1f

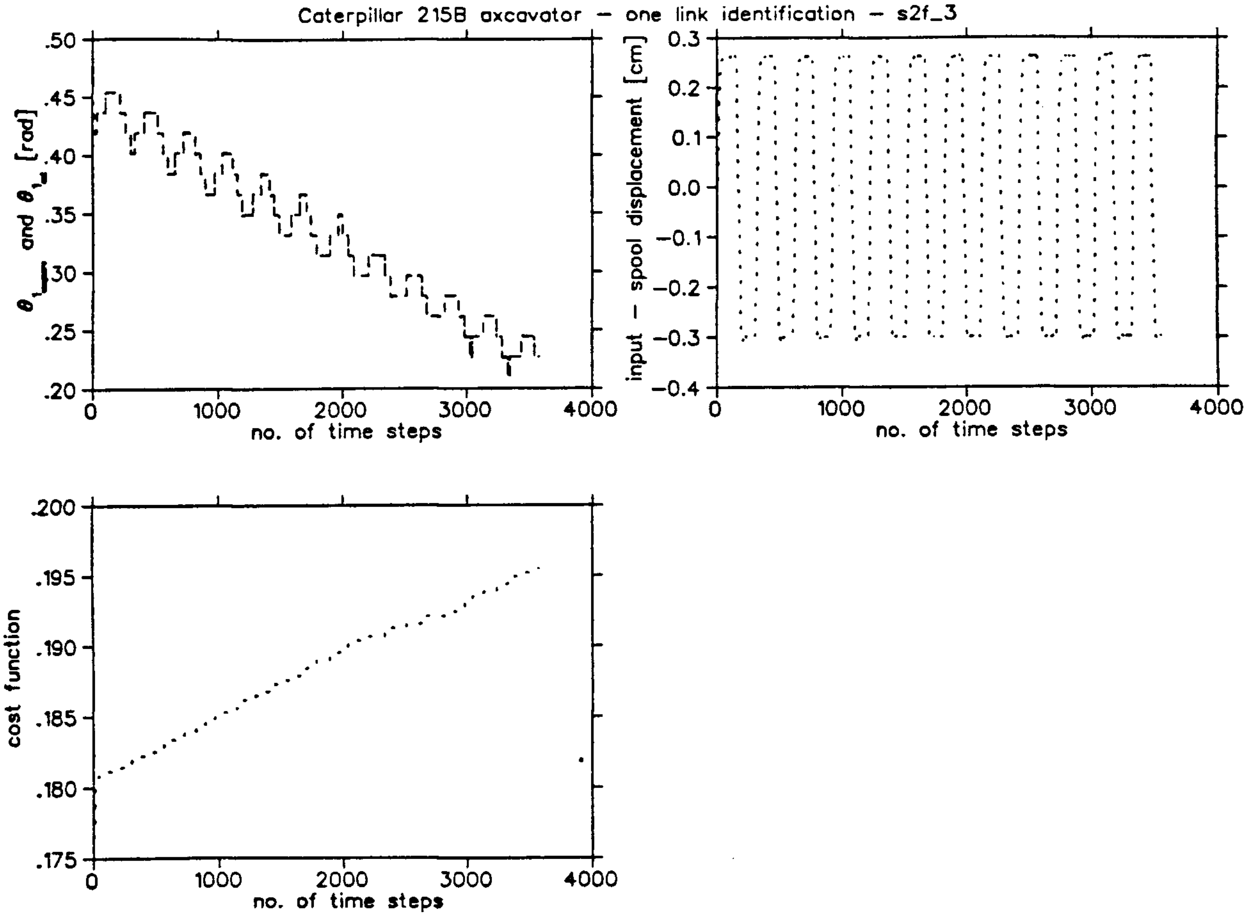

Figure A.3: Input output behavior for S2f

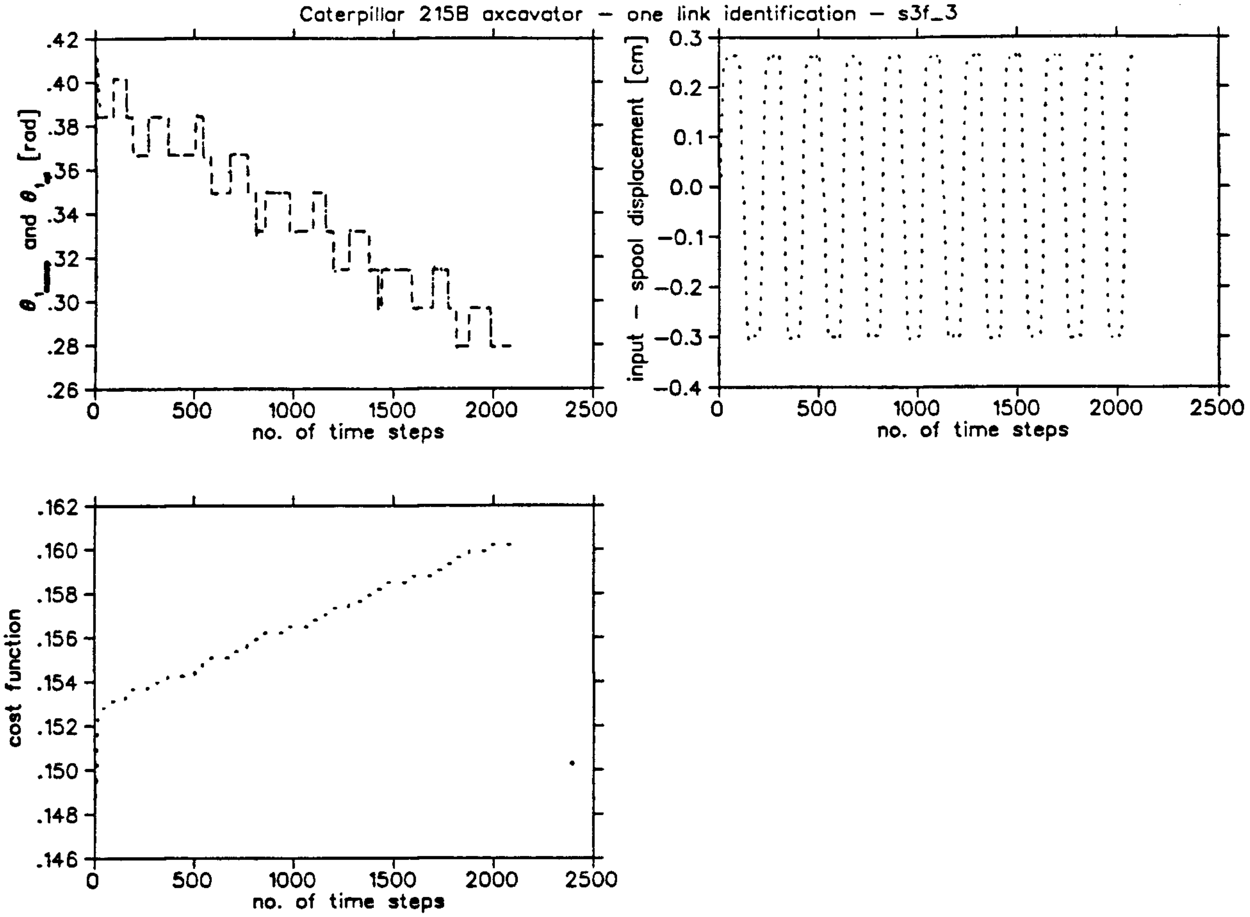

Figure A.4: Input output behavior for S3f

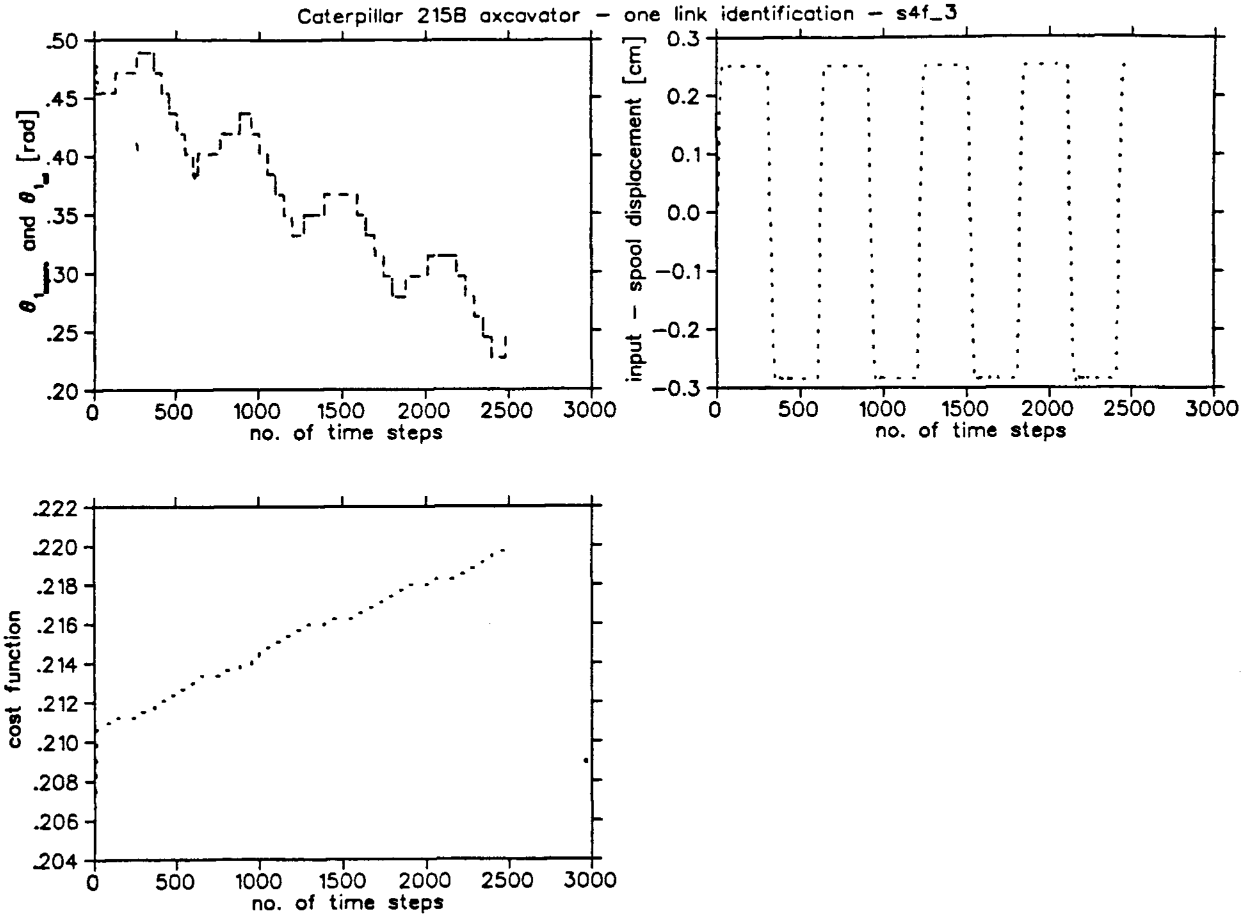

Figure A.5: Input output behavior for S4f

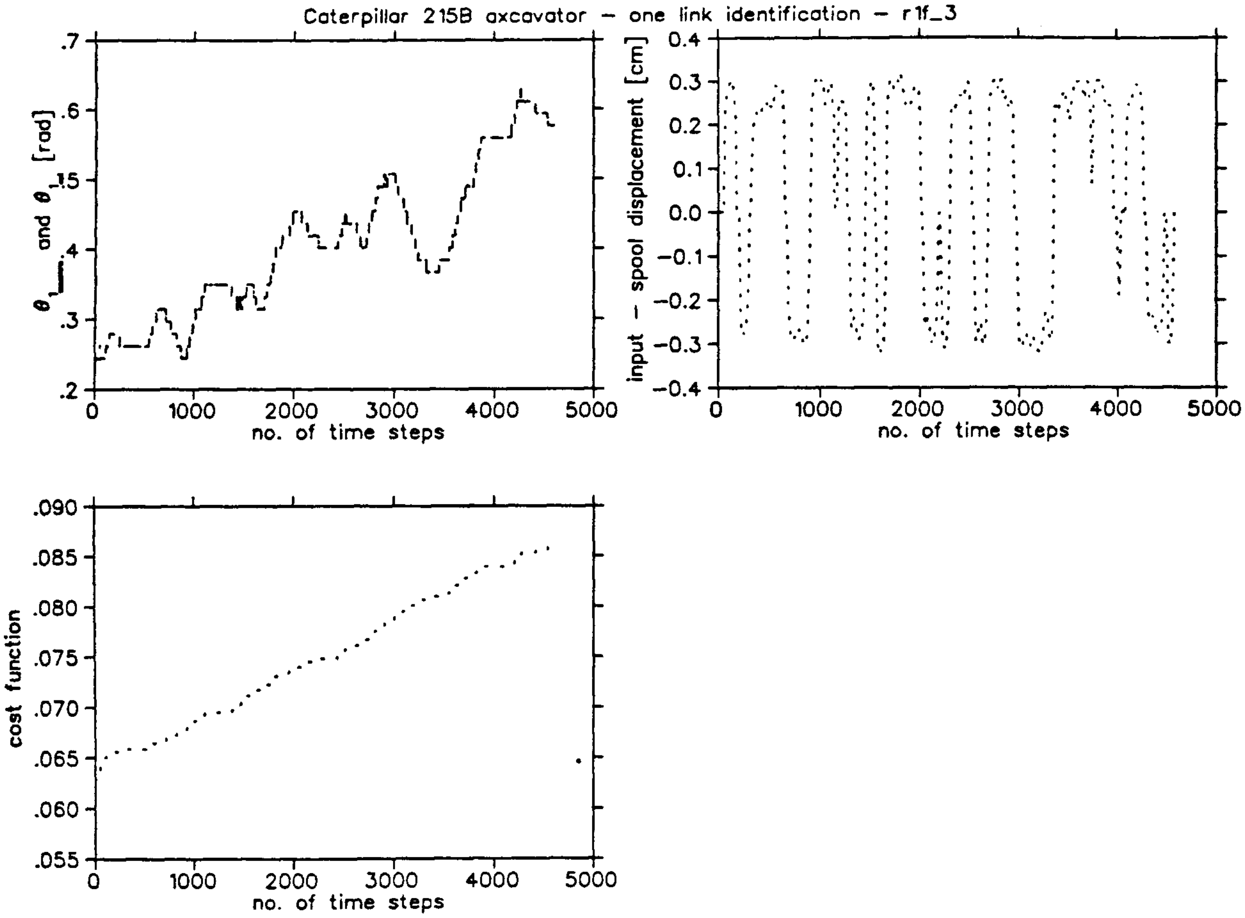

Figure A.6: Input output behavior for Rlf

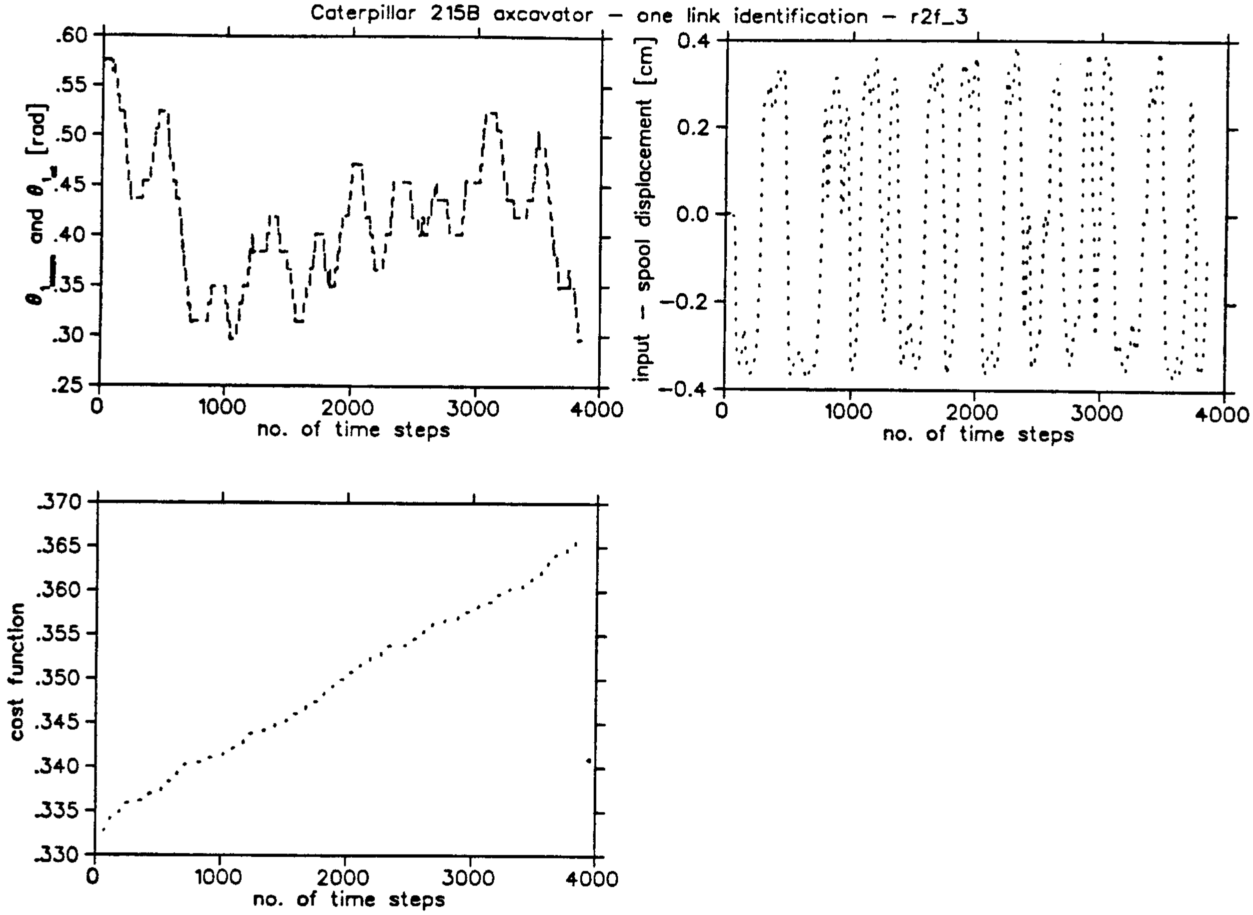

Figure A.7: Input output behavior for R2f

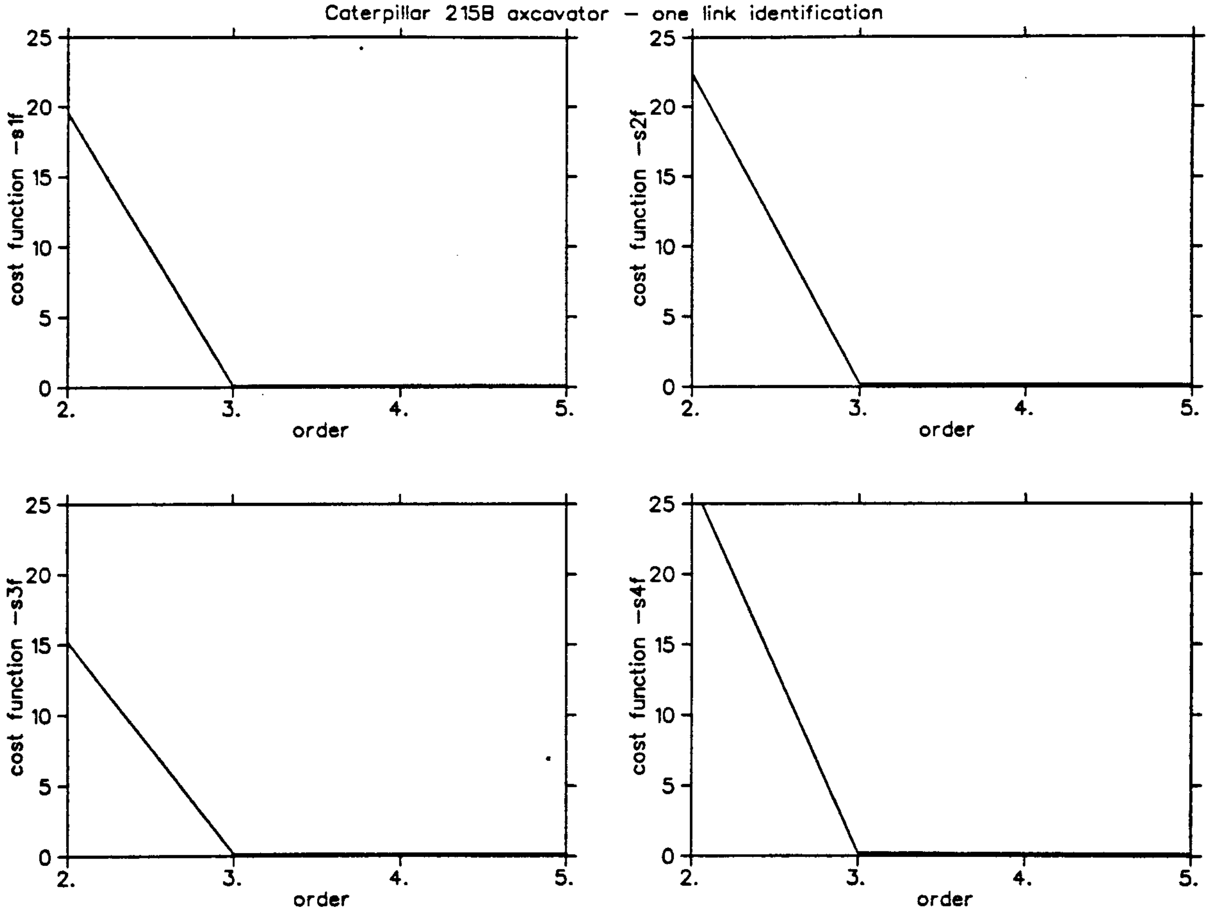

Figure A.8: Cost function behavior for the Sif cases

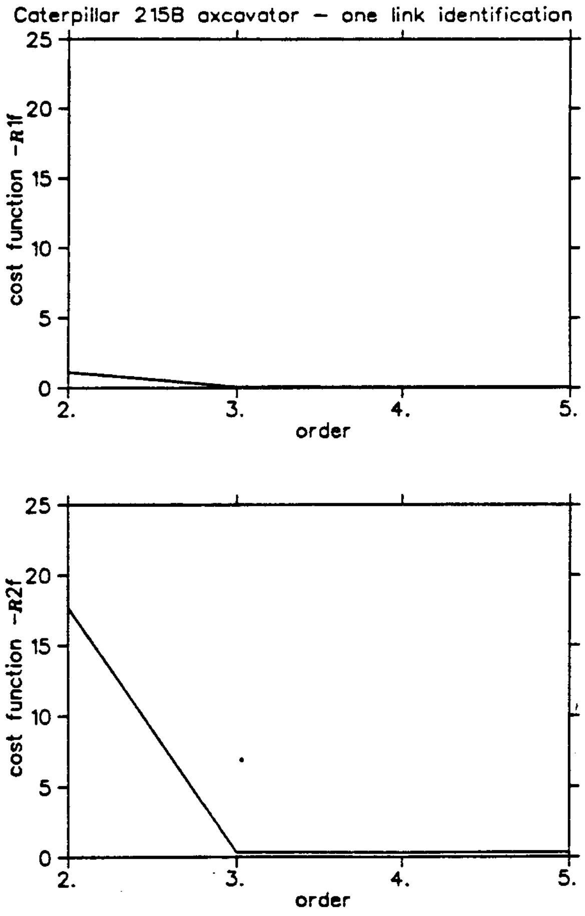

Figure A.9: Cost function behavior for the Rif cases

Appendix B

Modal Analysis for a Cantilever Beam

The analysis is based on Bisplinghoff et. al [66] and Tse et. al [67]. The Partial Differential Equation (PDE) for the deflection $w(x,t)$ in a cantilever beam, considering bending and shearing strains and neglecting shear deformation and rotary inertia effects, is:

$$m\frac{d^2w}{dt^2} + EI\frac{d^4w}{dx^4} = 0 \tag{B.1}$$

Where E is Young's modulus, I is the cross sectional moment of inertia of the arm, and m is the mass per unit length. The above PDE is separable, so let:

$$w(x,t) = \phi(x)q(t) \tag{B.2}$$

$$\frac{d^2q}{dt^2} + \omega^2 q = 0 \tag{B.3}$$

$$\frac{d^4\phi}{dx^4} - b^4\phi = 0 \tag{B.4}$$

Where $b^4 = \frac{\omega m}{EI}$. General solutions for the above equations are:

$$q(t) = Asin(\omega t) + Bcos(\omega t) \tag{B.5}$$

$$\phi(x) = Csinh(bx) + Dcosh(bx) + Esin(bx) + Fcos(bx) \tag{B.6}$$

The boundary conditions for cantilever beam are:

at $x = 0$: $\phi(0) = 0,$ $\frac{d\phi(0)}{dx} = 0$

at $x = L$: $\frac{d^2\phi(L)}{dx^2} = 0, \qquad \frac{d^3\phi(L)}{dx^3} = 0$

Solving the above equations with the boundary conditions result with:

The natural frequencies by:

$$cos(bL)cosh(bL) = 1 \tag{B.7}$$

Where $bL = 0.5969\pi, \quad 1.4942\pi, \quad \frac{5}{2}\pi, \quad \frac{7}{2}\pi, \quad ......$

The mode shapes by:

$$\phi(x) = D[\Delta(sinh(bx) - sin(bx)) + cosh(bx) - cos(bx)] \tag{B.8}$$

Where:

$$\Delta = \frac{sin(bL) - sinh(bL)}{cos(bL) + cosh(bL)} \tag{B.9}$$

D is a normalized coefficient where $\phi(L) = 1$ so:

$$D = \frac{cosh(bL) + cos(bL)}{2sinh(bL)cosh(bL)} \tag{B.10}$$

A dynamic model has three integrals as a function of $\phi(x)$, which are (assuming uniform mass distribution m) :

$$\begin{aligned} I1 &= m \cdot \int_0^L x\phi(x)dx = \frac{2mD}{b^2} \\ I2 &= m \cdot \int_0^L \phi^2(x)dx = mD^2L \\ I3 &= EI\int_0^L [\frac{d^2\phi(x)}{dx^2}]^2 dx = \omega^2 I2 \end{aligned} \tag{B.11}$$

For arm with the next data:

$L = 1$ $[meter]$

$EI = 574.024$ $[N \cdot m^2]$

$I_t = 0.2817$ $[kg \cdot m^2]$

$\phi(L) = 1$

$m = 0.8451$ $[kg/m]$

Table B.1 presents the data for the first five modes. Figure B.1 shows the modal shapes for the cantilever beam.

Mode	*bl (rad)*	*w (rad/dec)*	*Delta*	*D*	I_1	I_2	I_3
1	1.8752	91.644	-0.7266	0.5	0.2404	0.2113	1774.4
2	4.6942	574.294	-0.9805	-0.5	-0.03835	0.2113	69681.4
3	7.854	1606	-0.9999	0.5	0.00138	0.2113	544928
4	10.9956	3151	-0.99998	-0.5	-0.00699	0.2113	2097707
5	14.137	5208.77	-0.9999	0.5	0.00423	0.2113	5732162

Table B.1: Data for the first five modes of a cantilever beam

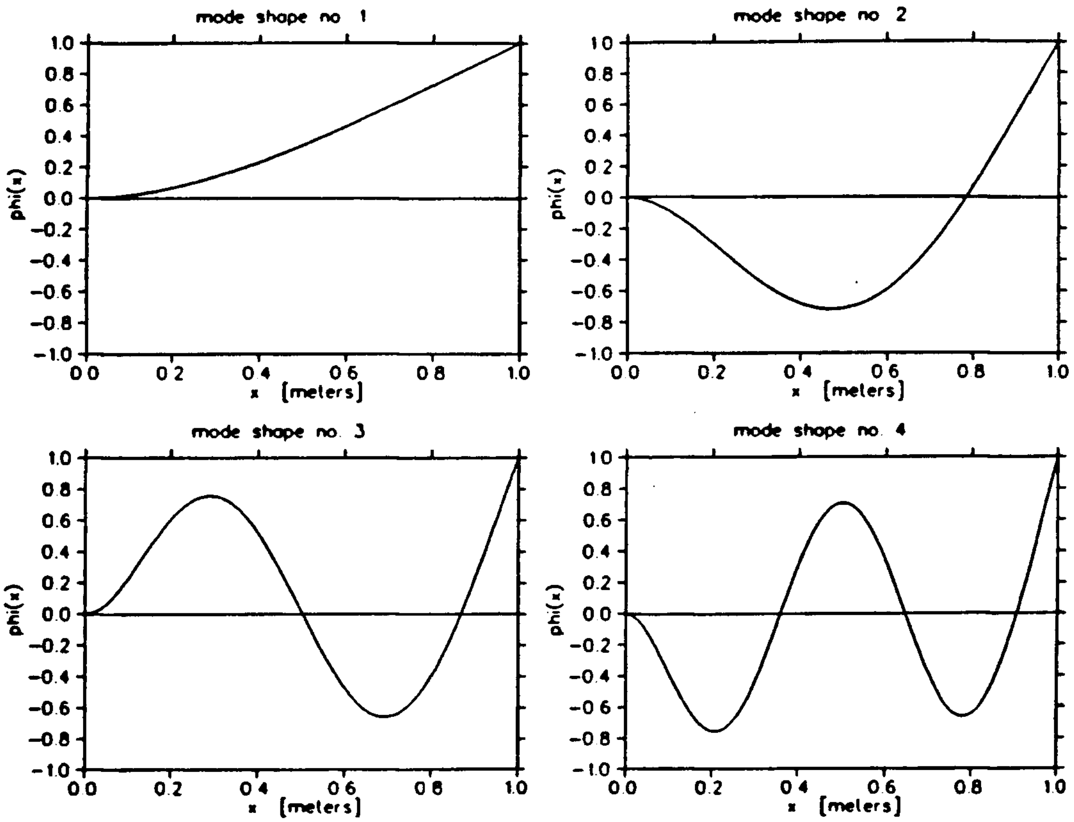

Figure B.1: Modal Shapes for a Cantilever Beam