IMPLEMENTATION OF AN ADAPTIVE CONTROLLER ON A
TI TM 990 MICROCOMPUTER

by
Bahram Tooba
B.Sc. Purdue University, 1979

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the department
of
Electrical Engineering
We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
March 1982
© Bahram Tooba, 1982
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.

Bahram Tooba

Department of Electrical Engineering

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date March 11 1982
ABSTRACT

The thesis tests the implementation of an adaptive identifier and controller on a Texas Instrument TM 990 microcomputer system.

The adaptive algorithm was proposed by Martin-Sanchez [20], and was selected because of the small amount of computation needed in comparison with methods such as recursive least squares estimation. The controller is similar to a simplified minimum variance controller used in self-tuning regulators.

The thesis describes the adaptive algorithm and the hardware and software used in the microcomputer implementation of this method. The results show operation of the algorithms for first-order and second-order processes.
# TABLE OF CONTENTS

**ABSTRACT** ................................................................. i

**TABLE OF CONTENTS** .................................................. ii

**LIST OF TABLES** ....................................................... iv

**LIST OF ILLUSTRATIONS** ........................................... v

**ACKNOWLEDGEMENTS** .................................................. vii

**I. ADAPTIVE CONTROL** ................................................. 1

1. Introduction ......................................................... 1
2. Adaptive Parameter Estimation and Control ...................... 4
3. Computer Simulation ................................................. 9
4. The Outline of This Thesis ........................................ 10

**II. IDENTIFICATION AND CONTROL ALGORITHM** .................... 12

1. Overview .............................................................. 12
2. Discrete-Time Systems .............................................. 12
   3.1. Formulation .................................................... 16
   3.2. Hyperstability of the Control System ....................... 20
   3.3. Relation Between $e(k)$ and $s(k)$ ......................... 23
   3.4. Convergence of the Control Block Parameters .......... 25
   3.5. Conclusions .................................................. 27

**III. HARDWARE AND SOFTWARE** ..................................... 28

1. Overview .............................................................. 28
2. The Hardware ......................................................... 28
   2.1. TM 990/101M-1 Microcomputer ................................ 28
   2.2. RTI-1241-S Board .............................................. 29
3. The Software .......................................................... 30
   3.1. The First-Order Systems ...................................... 30
   3.2. The Second-Order Systems .................................... 34
   3.3. Arithmetic Operations ........................................ 35
   3.4. The Sampling Rate ............................................. 37
   3.5. The Computer Programme ...................................... 39
IV. MICROCOMPUTER EVALUATION OF THE ADAPTIVE IDENTIFIER AND CONTROLLER .................................................. 42

1. Overview .............................................. 42
2. First-Order Processes .................................. 43
   2.1. Identification .................................. 47
   2.2. Control ......................................... 51
3. Second-Order Processes ............................... 58
   3.1. Identification .................................. 58
   3.2. Control ......................................... 62

V. CONCLUSION ............................................. 67

1. Summary ............................................. 67
2. Future Work ......................................... 68

APPENDIX A  THE SYSTEM HARDWARE .................. 70

A.1. TM 990/101M-1 Microcomputer .................... 70
A.2. RTI-1241-S Board .................................. 71
A.3. TI 745 Terminal ................................... 73
A.4. TM 990/510 Card Cage .............................. 73
A.5. The Power Supply .................................. 73

REFERENCES ............................................. 75
Table 3.1 The sampling rates and their corresponding hexadecimal coded data. ..........................38
<table>
<thead>
<tr>
<th>Fig.</th>
<th>Illustration Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The Block Diagram of (a) RLS Adaptive Identifier (b) Self-Tuning Regulator (STR).</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Basic Configuration of a Model Reference Adaptive system (MRAS).</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Configuration of the Adaptive System in: (a) Identification Mode, (b) Control Mode.</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>General Configuration of an Asymptotically Hyperstable Feedback Autonomous System.</td>
<td>19</td>
</tr>
<tr>
<td>2.3a</td>
<td>The Nonlinear Asymptotically Hyperstable Feedback Autonomous Systems,</td>
<td>21</td>
</tr>
<tr>
<td>2.3b</td>
<td>The Equivalent Autonomous System.</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Flowchart for the Identification of a First-Order System.</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Flowchart for the Control of a First-Order System.</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Flowchart for two's Complement Multiplication.</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>The Block Diagram of the Adaptive System in Identification Mode.</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>The Block Diagram of the Adaptive System in the Control Mode.</td>
<td>45</td>
</tr>
<tr>
<td>4.3a</td>
<td>The Estimated Parameters.</td>
<td>48</td>
</tr>
<tr>
<td>4.3b</td>
<td>The Process Output and The Predicted Output.</td>
<td>48</td>
</tr>
<tr>
<td>4.4a</td>
<td>The Estimated Parameters.</td>
<td>49</td>
</tr>
<tr>
<td>4.4b</td>
<td>The Process Output and The Predicted Output.</td>
<td>49</td>
</tr>
<tr>
<td>4.5a</td>
<td>The Estimated Parameters.</td>
<td>50</td>
</tr>
<tr>
<td>4.5b</td>
<td>The Process Output and The Predicted Output.</td>
<td>50</td>
</tr>
<tr>
<td>4.6a</td>
<td>The Estimated Parameters.</td>
<td>52</td>
</tr>
<tr>
<td>4.6b</td>
<td>The Process Output and The Predicted Output.</td>
<td>52</td>
</tr>
<tr>
<td>4.7a</td>
<td>The Estimated Parameters.</td>
<td>53</td>
</tr>
<tr>
<td>4.7b</td>
<td>The Process Output and The Predicted Output.</td>
<td>53</td>
</tr>
<tr>
<td>4.8a</td>
<td>The Process Output for the First-Order System.</td>
<td>54</td>
</tr>
<tr>
<td>4.8b</td>
<td>The Reference Signal.</td>
<td>54</td>
</tr>
<tr>
<td>4.9a</td>
<td>The Process Output.</td>
<td>56</td>
</tr>
</tbody>
</table>
Fig. 4.9b The Control Signal. ........................................56
Fig. 4.10a The Process Output. ......................................57
Fig. 4.10b The Control Signal. ......................................57
Fig. 4.11a The Estimated Parameters. ...........................59
Fig. 4.11b The Process Output and The Predicted Output. ....59
Fig. 4.12a The Estimated Parameters. ...........................60
Fig. 4.12b The Process Output and The Predicted Output. ....60
Fig. 4.13a The Estimated Parameters. ...........................61
Fig. 4.13b The Process Output and The Predicted Output. ....61
Fig. 4.14a The Process Output. ......................................63
Fig. 4.14b The Control Signal. ......................................63
Fig. 4.15a The Process Output for the Second-Order System. ..65
Fig. 4.15b The Reference Signal. ...................................65
Fig. 4.16a The Process Output. ......................................66
Fig. 4.16b The Control Signal. ......................................66
I wish to thank Dr. M.S. Davies for his patience, help, and encouragement during the life of this project.
1. Adaptive Control

1. Introduction

Adaptive control has been a challenge for control engineers for a long time, the first research papers on the subject being published in the early 1960's. The interest in adaptive control arose from the need for improving the performance of increasingly complex engineering systems involving large uncertainties, of systems with unknown and time varying parameters. An adaptive system is one that continually monitors changes in the process behaviour and adjusts its control strategy automatically to maintain good performance. Therefore, these systems are generally nonlinear. In spite of much interest in adaptive control, only modest progress has been made in introducing these controllers into industry. Controllers allowing proportional, integral and derivative action (PID) are extensively used, with almost 90 percent of electronic analog controllers used in the industry being three term controllers of this type [27]. As is evident, most industrial processes can be controlled satisfactorily with PI or PID controllers, and indeed many processes have been designed to be controlled by PID controllers [1]. In control loops that are not critical, these three term controllers will undoubtedly continue to be used extensively in the future.

With an increasing demand for efficiency in the use of
energy and raw materials, however, there is a corresponding increase in the number of control problems which require controllers more sophisticated than the conventional PID type. Although it is fairly easy to tune a PI controller which has only two terms, once operating in an installation where there may be several hundred of these controllers, it is a substantial amount of work to keep all of them well-tuned if for any reason the most desirable settings fluctuate. A PID controller which has three or four parameters, however, is not very easy to tune, particularly if the process dynamics are slow. Therefore the derivative action is often switched off in industrial controllers although it might be used to advantage. More complicated controllers which include feedforward, feedback and observers often have a large number of parameters which can be adjusted, and this can not be done without a systematic adjustment procedure. Adaptive controllers attempt to address this problem.

The difficulty of implementing advanced control procedures is a reason in explaining why modern control theory has not been used more extensively. Some of the algorithms proposed for example, are computationally too difficult to implement on small computers. In designing advanced controllers, it is often necessary to develop a mathematical model for the process and its disturbances, and to derive the controller parameters using these model and disturbance characteristics. The appropriate mathematical model can be obtained from physical consideration or system identification. The drawback with such a procedure is
that it is time consuming, and may need the continuous attention of personnel with skills in modelling, system identification and control design. The adaptive controller can be regarded as a convenient combination of system identification and a control design techniques that can operate without supervision. In addition, if the system parameters and its disturbance statistics are slowly varying with time, a properly designed adaptive controller can continuously tune the system for close to optimal performance.

Recent increased interest in adaptive control has concentrated on procedures that simplify the implementation of adaptive algorithms on minicomputers and microcomputers, thereby reducing the cost, and providing control engineers with a viable alternative to PID controllers. The success of these adaptive controllers has justified the increased interest in them. Reference [14] describes a successful application of an adaptive controller for a paper machine. The design of adaptive autopilots for steering large tankers seems likely to replace conventional autopilots, in recent tests [23], the adaptive autopilot installed on the tanker outperformed the conventional PID autopilot under almost all conditions. This design is based on recursive least squares estimation with a minimum variance controller, the so-called "self-tuning regulator". Other designs which can replace the conventional PID autopilot are discussed in [22], which describes a design based on model reference techniques.
2. Adaptive Parameter Estimation and Control

The two primary approaches to parameter identification that have been developed are:

i) Recursive Least Squares (RLS), and


Both methods have received great attention in the literature. The previous work done on these methods will be briefly described, although it must be realized that there have been hundreds of papers published on these methods during the past 20 years.

The Recursive Least Squares method has been developed and successfully implemented in the industry, when combined with a suitable control algorithm. [2] is a good survey of this method. Fig. 1.1 shows the block diagram of a self-tuning regulator. In this method the parameters of a difference equation model are determined by minimizing the cost:

$$V = \sum_{k=0}^{N} |y(kT) - \hat{y}(kT)|^2$$

with respect to these parameters, here $y(kT)$ is the process output and $\hat{y}(kT)$ is the prediction model output. The recursive algorithm can be found in [2] or [3]. The parameter estimates are updated after each sampling, and used in the tuning of the control algorithm. In the work of Peterka [18], Astrom, and Wittenmark [4], and Wittenmark [19], the recursive least
Fig. 1.1 The Block Diagram of (a) RLS Adaptive Identifier  
(b) Self-Tuning Regulator (STR).
squares parameter estimation is combined with a minimal variance controller. In this case the model structure may be chosen so that the estimated parameters coincide with the controller parameters, so that the controller parameters are calculated directly. In the work of Cegrell and Hedqvist [14], and Borrison and Syding [15] successful application of these 'self-tuning regulators' are reported. An interesting aspect of the second work [15] was that it was performed using telecommunications between a plant in Kiruna in northern Sweden and a computer in Lund in southern Sweden, 1800 km apart. In [10] Kurz, Isermann, and Schumann experimentally compare and evaluate two recursive parameter estimation methods, the (RLS) estimation and Recursive Maximum Likelihood (RML) estimation and 6 different control algorithms, so considering a total of 12 adaptive control schemes. These algorithms were implemented on a process computer and tested on simulated analog processes. It was found that the control algorithms perform better when used with RLS estimation method, and the control algorithm could be applied to different type of processes, e.g. minimum phase and non-minimum phase, stable and unstable processes, and processes with time delay.

The adaptive regulator based on the (RLS) seems to have a broad field of applications, however a proof of the convergence of adaptation algorithm of this type is not available in spite of very good experimental results [3]. Another problem with (RLS) estimation is that not all the parameters of the model can be determined from the input-output observations when
feedback exists in the process. Åström and Wittenmark [4] explain how to overcome this problem and suggest some guidelines for unbiased parameter estimation which allow the estimated parameters to become very close to their optimal values.

The Model Reference Adaptive method (MRAS), seems to have been introduced by Whitaker, Yarmon, and Kezer [28] in 1958. Figure 1.2 shows the basic configuration of standard MRAS. In this method the input $u$ is fed both to the process and to the adjustable model with the difference between the process output and the model output being used to generate an error $e(t)$. This error vector is used by an adaptive mechanism to adjust the model parameters and force $e \to 0$ as $t \to \infty$. Since its introduction, there have been many papers and doctoral dissertations dedicated to the various types of MRAS. Landau [8] provides an excellent survey of MRAS from 1964 to 1973, a recent book [9] describes several different techniques in MRAS and provides good examples of applications. Landau uses a hyperstability theorem proposed by Popov [16] for the proof of the convergence of MRAS. The adaptation laws have been formulated based on different methods, Pearson [13] developed an algorithm based on the gradient method, and this method has been used by others such as Serdyukov [29]. Here again there is no analytical proof of the convergence of the parameters estimates, although the experimental results are good. A new method was proposed by Martin-Sanchez [20], for an adaptive algorithm which is related to Popov's hypertability criterion.
Fig. 1.2 Basic Configuration of a Model Reference Adaptive System (MRAS)
as applied to discrete systems [7], but is significantly different from the MRAS formulation. In this method the adjustment of the identification and control block parameters is carried out based on the information available from the input and output of the process. This method will be described in detail in Chapter II. [24] contains a large number of papers presented in the 1979 Yale University Workshop on the Application of Adaptive Control. It includes good tutorials on both RLS estimation and MRAS. Applications described in this book are based mostly on these two methods.

3. Computer Simulation

In order to assess their suitability for microcomputer implementation, the recursive least squares estimation method proposed by Astrom and Wittenmark [3], and the Martin-Sanchez method were simulated using Amdahl/470 computer. The programmes were written in Fortran and the matrix operations were performed by subroutines. The transfer functions of the processes simulated were identical for both methods, a first-order process and a second-order process. The results were very satisfactory for both methods when the systems were noise free and the estimated parameters reached their correct values. The performance of the two methods deteriorated with the introduction of noise at the input and output of the two processes. The (RLS) estimation was found to be slightly more sensitive to the selection of initial parameters when system identification was performed. The computation time was a little
higher for (RLS) estimation. The number of arithmetic operations, especially multiply and divide, was much more in the case of (RLS) estimation as the order of the processes, and so the number of parameters to be identified increased. In the identification of the second-order process, the RLS method needed about 100 multiply and divide operations compared with only 25 operations for Martin-Sanchez method.

Based on the results from the computer simulation it was decided that the Martin-Sanchez method will be considered in our work mainly because it is computationally simpler than (RLS) for microcomputer implementation and also because it is a method that has not been implemented on a microprocessor before.

4. The Outline of This Thesis

The aim of this thesis is to implement and evaluate the Martin-Sanchez method on a microprocessor. This system has been found to perform the identification and control on line with fairly good results, as will be described.

Chapter II will cover the analysis of the algorithm, the formulation, the proof of hyperstability and the proof of the convergence of the parameters. Chapter III will discuss the implementation of the algorithm on microprocessor and the interfacing of the different parts of the system. It will also analyse the software employed. Chapter IV will cover the testing of the identifier and the controller for the first-order systems (with at most two parameters) and
second-order systems (4 parameters maximum). Chapter V will discuss the results and will evaluate the identifier and the controller.
II. IDENTIFICATION AND CONTROL ALGORITHM

1. Overview

In this chapter, the identification and control algorithm will be described. The mathematical foundation of the algorithm, the proof of the hyperstability of the parameter estimates, and the proof of the convergence of the control block parameters will be outlined.

2. Discrete-Time Systems

A single-input-single-output (SISO) linear time-invariant continuous time process in a noise free environment can be represented by the transfer function

\[ G(s) = \frac{\alpha_n s^n + \alpha_{n-1} s^{n-1} + \cdots + \alpha_1 s + \alpha_0}{s^m + \beta_{m-1} s^{m-1} + \cdots + \beta_1 s + \beta_0} \]  (2.1)

where \( m \geq n \). To control a process such as (2.1) digitally it is convenient to develop a discrete equivalent of \( G(s) \) preceded by a sample and hold block representing the D/A conversion at the output of the digital controller. The discretization of \( G(s) \) will result in a model of the process which can be represented in the z-transform domain or, equivalently, by an input-output difference equation of the form:

\[ y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) \]  (2.2)
We will study the identification and control of systems which can be represented by a difference equation such as (2.2) using the algorithm first proposed by Martin-Sanchez [20]. This algorithm will be implemented on a Texas Instrument TM 990 microcomputer system to be described in detail in Chapter III. This chapter is dedicated to the analysis of the adaptive algorithm.

3. An Algorithm for Model Reference Adaptive Systems

In this algorithm, an adaptive control block is developed which uses only information from the input and output of the process. The controller attempts to behave as the exact inverse of the plant. In this way the control block output, equal to the process input, drives the process output to follow the control block input or reference. The identification can be carried out on line, and unlike the standard MRAS which needs some a'priori knowledge of the parameters of the plant, this method can carry out the identification for any initial parameters. The two modes of the system, identification and control, are shown in Figures 2.1a and 2.1b. Figure 2.1a resembles an MRAS identification block diagram, but the formulation of the algorithm is quite different. In the identification mode, Figure 2.1a, an input $u$ is fed to both the process and the identification block. The comparison between the plant output and the identification block output generates an error $e$, which will be used by the adaptive mechanism to adjust the identification block parameters so as to make $e \to 0$. 
Fig. 2.1 Configuration of the Adaptive System in:
(a) Identification Mode, (b) Control Mode
as $t \to \infty$.

In the control mode, Figure 2.2b, the desired output is the input to the control block, which uses the adaptive mechanism to generate an input to the plant. The error $e$ is a measure of the deviation of the plant output from the desired output. This error is again used to update the controller parameters, so as to make $e \to 0$. It is assumed that the plant is output controllable and can be represented by an input-output equation of the form (2.2).

Let $d(k)$ be the desired output of the plant at sampling instant $k$, and of the same dimension as $y$. An algorithm is required which will generate the control signal $u(k)$ from $d(k+1)$ and the error $e(k) = y(k) - d(k)$. The adaptive mechanism used to adjust the model parameters is the same during both the identification and the control modes of operation. The parameter estimation scheme will be shown to be asymptotically hyperstable by use of the Popov hyperstability criterion [16]. If the controller is designed properly then the error $e \to 0$ as $t \to \infty$, and therefore $y(k) \to d(k)$. The parameters of the controller will remain unchanged when $e=0$, but if the error is non-zero due to process perturbations or the variation of the plant's parameters, then proper control action will be taken to achieve equilibrium. This of course will be done by adjusting the controller parameters.
3.1. Formulation

Consider a process characterized by the difference equation

\[ y(k) = Ay(k-1) + Bu(k-1) \]  \hspace{1cm} (2.3)

where \( y(k) \) and \( u(k) \) are the output and the input vectors of the plant at sampling instant \( k \). Furthermore, if the input vector \( u \) and the output vector \( y \) are assumed of the same dimension, say \((r \times 1)\), then \( A \) and \( B \) are real matrices of dimension \((r \times r)\). In addition, it is assumed that the plant is output controllable. It is desired to find an identifier that will estimate the matrices \( A \) and \( B \) using only the input \( u \) and the output \( y \). This identification procedure will include the reference model:

\[ d(k) = \tilde{A}(k-1)y(k-1) + \tilde{B}(k-1)u(k-1) \]  \hspace{1cm} (2.4)

where \( A(k-1) \) and \( B(k-1) \) are the estimates of the matrices \( A \) and \( B \) at the sampling instant \( k-1 \); \( d(k) \) will therefore be the predicted plant output at sampling instant \( k \), from the information available at instant \( k-1 \).

Equation (2.4) also describes the control block, but now the output sequence is the controller \( u(k) \) obtained from the information available on \( d(k+1) \). Thus equation (2.4) becomes

\[ u(k) = [\tilde{B}(k)]^{-1}d(k+1) - [\tilde{B}(k)]^{-1}\tilde{A}(k)y(k) \]  \hspace{1cm} (2.5)

In the above equation \( d(k+1) \) represents the desired output of
the plant at instant \( k+1 \). \( u(k) \) will be the input to the plant and, since the identification block and the control block are governed by the same set of equations, this input will ideally ensure that the process output is always equal to the desired output.

The error for both the identification block and the control block will be defined as

\[
e(k) = y(k) - d(k) \tag{2.6}
\]

To ensure that the error converges to zero as \( t \to \infty \), an stable adaptive mechanism to adjust \( \hat{A}(k) \) and \( \hat{B}(k) \) is developed. This uses the following parameter update:

\[
\hat{A}(k) = \Delta \hat{A}(k) + \hat{A}(k-1) \tag{2.7}
\]

\[
\hat{B}(k) = \Delta \hat{B}(k) + \hat{B}(k-1) \tag{2.8}
\]

These new estimates of \( A \) and \( B \) can be used to predict the plant output at instant \( k \). Let \( g(k) \) be the predicted output at this instant

\[
g(k) = \hat{A}(k)y(k-1) + \hat{B}(k)u(k-1) \tag{2.9}
\]

This is the a'posteriori model which will generate a new a'posteriori error defined by:

\[
s(k) = y(k) - g(k) \tag{2.10}
\]

The adaptive mechanism proposed by Martin-Sanchez uses the error \( s(k) \) to determine the incremental matrices \( \Delta \hat{A}(k) \) and
ΔB(k) independent of the initial conditions (A(0), B(0), y(0), u(0)). The mechanism is based on the following theorem:

THEOREM 1: The system defined by equations (2.3) to (2.10) and whose general configuration is shown in Figure 2.1 will be asymptotically hyperstable if the matrices \( \tilde{A}(k) \) and \( \tilde{B}(k) \) are generated by:

\[
\begin{align*}
\tilde{A}(k) &= \tilde{A}(k-1) + \alpha s(k)y(k-1) \\
\tilde{B}(k) &= \tilde{B}(k-1) + \beta s(k)u(k-1)
\end{align*}
\]

where \( \alpha \) and \( \beta \) are positive constant coefficients, to be selected depending on the process characteristics.

The concept of hyperstability is a less restrictive form of absolute stability. If the control system can be decomposed into a linear time-invariant part and a nonlinear time-varying part, as shown in Figure 2.2, the Hyperstability Theorem of Popov specifies sufficient conditions which guarantee the stability of the combined system. The main criteria to be satisfied are the inequality

\[
\sum_{k=1}^{k_f} u_i(k)y(k) \geq -a_0^2
\]

where \( a_0^2 \) is a positive constant, and also the input/output relationship is:

\[
\begin{align*}
u_i(k) &= F(y(i), i) \quad \text{for} \quad i \leq k, \quad \text{and} \\
u(k) &= -u_i(k) \quad \text{for} \quad k \geq 0.
\end{align*}
\]
Fig. 2.2 General Configuration of an Asymptotically Hyperstable Feedback Autonomous System.
The first criterion is similar to the familiar constraint on the inequality condition in the conventional Popov criterion. The system is asymptotically hyperstable, i.e. $y(k) \to 0$ as $k \to \infty$, if the linear plant is strictly positive real discrete. It is only hyperstable, i.e. $y(k)$ is bounded as $k \to \infty$, if the linear plant is positive real.

As shown in Figure 2.3, the system equations (2.4) to (2.12) form a nonlinear feedback autonomous system whose state can be defined by the vector $s(k)$, equivalent to the system investigated by Popov. The essence is to convert the adaptive identifier into an equivalent feedback system with a linear block in the forward path and a nonlinear time-varying block in the feedback path. The linear part in this case reduces to a unity gain.

The Popov hyperstability criterion [16] for such an equivalent system in the discrete form [7] states that asymptotic hyperstability is assumed if the following inequality is satisfied for all $k$,

$$
\sum_{k=0}^{K_i} s(k)'s(k) \leq \sigma^2_0, \quad \sigma_0 = \text{constant} \forall k, \quad (2.13c)
$$

where $s(k)'$ is the transpose of vector $s(k)$. If the parameter updating algorithm can be shown to satisfy the above inequality, it will be asymptotically hyperstable.
Fig. 2.3a The nonlinear asymptotically hyperstable feedback autonomous system.

Fig. 2.3b The equivalent autonomous system.
3.2. Hyperstability of the Control System

The proof of hyperstability for the control system with the adaptive mechanism given in equations (2.11) and (2.12) is from [20] by Martin-Sanchez. This is actually a particular case of Popov's theorem for discrete systems investigated by I.D. Landau [7]. The outline proof is as follows:

From (2.3), (2.9), and (2.10)

\[ s(k) = [A - \tilde{A}(k)]y(k-1) + [B - \tilde{B}(k)]u(k-1) \]  \hspace{1cm} (2.14)

If (2.14) is multiplied by $-s(k)'$,

\[ -s(k)'s(k) = s(k)'[\tilde{A}(k)-A]y(k-1) + s(k)'[\tilde{B}(k)-B]u(k-1) \] \hspace{1cm} (2.15)

\[ -\sum_{k=0}^{k'} s(k)'s(k) = \sum_{k=0}^{k'} s(k)'[\tilde{A}(k)-A]y(k-1) + s(k)'[\tilde{B}(k)-B]u(k-1) \] \hspace{1cm} (2.16)

Equation (2.16) can be decomposed to make the following two conditions:

\[ \sum_{k=0}^{k'} s(k)'[A(k)-A]y(k-1) \geq -\lambda_a^2 \forall k_i \] \hspace{1cm} (2.17)

\[ \sum_{k=0}^{k'} s(k)'[B(k)-B]u(k-1) \geq -\lambda_b^2 \forall k_i \] \hspace{1cm} (2.18)

where $\lambda_a$ and $\lambda_b$ are constants.

Conditions (2.17) and (2.18) can be written in scalar form:

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=0}^{k_i} s_i(k)[\tilde{a}_{ij}(k) - a_{ij}]y_j(k-1) \geq -\lambda_a^2 \forall k_i \] \hspace{1cm} (2.19)
\[
\sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} \sum_{k=0}^{n} s_{ij}(k)[b_{pq}(k) - b_{pq}]u_{i}(k-1) \geq -\lambda^2 \bigvee_{k_i} \]  

(2.20)

The above conditions will be satisfied if the parameters \( a(k) \) and \( b(k) \) are formed according to the algorithms

\[
a_{ij}(k) = \sum_{h=0}^{k} s_{ij}(h)y_{ij}(h-1), \quad i=1,n \quad j=1,n \]  

(2.21)

\[
b_{pq}(k) = \sum_{h=0}^{k} s_{pq}(h)u_{pq}(h-1), \quad p=1,n \quad q=1,n \]  

(2.22)

If (2.21) and (2.22) are substituted in (2.19) and (2.20), the resulting equations are a special case of the more general equation

\[
\sum_{k=0}^{n} x(h)[k \sum_{h=0}^{k} x(h)+c] = 1/2[k \sum_{k=0}^{n} x(k)+c]^2 + 1/2 \sum_{k=0}^{n} x(k)^2 \]  

(2.23)

where \(-(c^2/2) \geq -\lambda^2\), \(c=\text{constant} \bigvee_{k_i} \lambda=\text{constant} \)

The left hand side of this equation is always positive; therefore, equations (2.21) and (2.22) can be expressed in a new form.

\[
a_{ij}(k) = s_{ij}(k)y_{ij}(k-1)+a_{ij}(k-1) \quad i=1,n \quad j=1,n \]  

(2.24)

\[
b_{pq}(k) = s_{pq}(k)u_{pq}(k-1)+b_{pq}(k-1) \quad p=1,n \quad q=1,n \]  

(2.25)

If (2.24) and (2.25) are converted into matrix form, equations (2.11) and (2.12) are obtained. The parameter updating of (2.21), (2.22) thus carries with it the important property of hyperstability.
3.3. Relation Between $e(k)$ and $s(k)$

Since the a'posteriori error $s(k)$ cannot be obtained before the parameters are updated and the parameters cannot be updated unless $s(k)$ is available; a relation between $s(k)$ and the a'priori error $e(k)$ must be established. This can be done as follows.

From (2.11), (2.12), and (2.14),

$$s(k) = [A-A(k-1)-cs(k)y(k-1)']y(k-1) + [B-B(k-1)-*s(k)u(k-1)']u(k-1)$$  \hspace{1cm} (2.26)

From equations (2.3), (2.4), (2.6), and (2.26),

$$s(k) = e(k)[ay(k-1)'y(k-1) + pu(k-1)'u(k-1)]$$  \hspace{1cm} (2.27)

solving (2.27) for $s(k)$,

$$s(k) = e(k)/[1+ay(k-1)'y(k-1) + pu(k-1)'u(k-1)]$$  \hspace{1cm} (2.28)

Let $\psi(k)$ be defined as

$$\psi(k) = [1+ay(k-1)'y(k-1) + pu(k-1)'u(k-1)]^{-1}$$  \hspace{1cm} (2.29)

(2.28) can now be written as

$$s(k) = \psi(k) e(k)$$  \hspace{1cm} (2.30)

The adaptive algorithm equations (2.11) and (2.12) can now be rewritten in a form, that avoids the need to calculate $s(k)$ in
order to update the parameters.

\[ \begin{align*}
\tilde{A}(k) &= \alpha \psi(k)e(k)y(k-1)' + \tilde{A}(k-1) \\
\tilde{B}(k) &= \beta \psi(k)e(k)u(k-1)' + \tilde{B}(k-1)
\end{align*} \]

(2.31) (2.32)

3.4. Convergence of the Control Block Parameters

It is important to realize that hyperstability of \( s(k) \) does not assure the convergence of the model parameters to their true values. It is easy to give examples of system for which the control laws based on incorrect parameters can give good output error behaviour. The hyperstability of the control system implies at least a local convergence of the control block parameters. The proof of this parameter convergence is given by Martin-Sanchez based on the work of Ngumo and Noda [21]. The method uses gradient parameters estimation technique and is outlined below:

According to the formulation in Section 3.3, the ith component of the a'posteriori error \( s(k) \) is given by

\[ s_i^e(k) = \psi(k)e_i^e(k) \]  

(2.33)

where \( \psi(k) \) is defined by (2.29) and \( e(k) \) by

\[ e_i^e(k) = y_i^e(k) - \hat{d}_i^e(k) \]  

(2.34)

Let \( \Theta \) and \( \Theta(k-1) \) be

\[ \Theta_i = [a_{i1}, \ldots, a_{in}, b_{i1}, \ldots, b_{in}]' \]  

(2.35)

\[ \tilde{\Theta}_i(k-1) = [\tilde{a}_{i1}(k-1), \ldots, \tilde{a}_{in}(k-1), \tilde{b}_{i1}(k-1), \ldots, \tilde{b}_{in}(k-1)]' \]  

(2.36)
Let \( x(k-1) \) be

\[
x(k-1) = [y_1(k-1), \ldots, y_N(k-1), u_1(k-1), \ldots, u_N(k-1)]' \tag{2.37}
\]

From (2.3), (2.35), and (2.37)

\[
y_{i'}(k) = x(k-1)'\hat{\theta}_i \tag{2.38}
\]

From (2.4), (2.36), and (2.37)

\[
d_i(k) = x(k-1)'\hat{\theta}_i(k-1) \tag{2.39}
\]

The algorithm previously developed for adjusting the identification and control block parameters can now be written in the following form:

\[
\hat{\theta}_i(k) = \hat{\theta}_i(k-1) + \psi(k)e_i(k)x(k-1) \tag{2.40}
\]

The algorithms are equivalent to a gradient parameter estimation method which minimizes the following cost function:

\[
J = [e_i(k)]^2/2 \tag{2.41}
\]

This is a special case of the cost function considered in [11] or the cost function used in the usual minimum variance adaptive controller [1].

From (2.40) and (2.34)

\[
\hat{\theta}_i(k) = \hat{\theta}_i(k-1) + \psi(k)x(k-1)[y_i(k) - d_i(k)] \tag{2.42}
\]
From (2.42), (2.38), and (2.39)

\[ \hat{e}_i(k) = \hat{e}_i(k-1) + \psi(k)x(k-1)x(k-1)'[\hat{e}_i - \hat{e}_i(k-1)] \]  

(2.43)

Let \( \hat{\theta}_i(k) \) be the identification error:

\[ \hat{\theta}_i(k) = \theta_i - \hat{e}_i(k) \]  

(2.44)

From (2.43) and (2.44)

\[ \hat{\theta}_i(k) = [I - \psi(k)x(k-1)x(k-1)'] \hat{\theta}_i(k-1) \]  

(2.45)

This is the case considered in [21], which proves \( \| \hat{\theta}_i(k) \|^2 \) will converge to zero as \( t \to \infty \), unless \( \hat{\theta}_i(k) \) and \( x(k) \) become orthogonal. Such an orthogonality can be avoided by using an input signal with adequate frequency components.

3.5. Conclusions

In this Chapter an identification and control algorithm was introduced, the identification method is similar to a model reference approach but the formulation is different and could be viewed essentially as a simplified form of recursive least squares estimation method. The adaptive control method was shown to establish a hyperstable system according to Popov theorem, this however does not imply that the controller will reduce the noise in the control system drastically or even provide a much better performance than a PID controller. This adaptive controller is similar to a minimum variance controller, used in the self-tuning regulators, with a simplified formulation for the parameter updates.
III. HARDWARE AND SOFTWARE

1. Overview

The identifier and the controller developed in Chapter II are tested by implementing the identification and the control algorithm on a microcomputer system. The system consists of:

(a) Texas Instrument TM 990/101M-1 microcomputer board.
(b) Texas Instrument TM 990/510 card chassis.
(c) Texas Instrument SILENT 700 (745 model) electronic data terminal.
(d) Analog Devices RTI-1241-S (A/D), (D/A) conversion board.
(e) GSC GOF-2A-1T triple output power supply.

The components were selected based on the system requirements, compatibility (simplicity to interface), ease of programming, and finally, the total cost of the system. The first part of this chapter will briefly describe the different parts of the system, and the interfacing. The second part will explain the software production and implementation.

2. The Hardware

2.1. TM 990/101M-1 Microcomputer

The TMS 9900 microprocessor is the heart of the TM990/101M-1. This 16 bit CPU features a memory to memory architecture and the extensive instruction set includes hardware multiply and divide. There are a total of eight addressing modes. The clock rate is 3 MHz. The communication
with the TI 745 terminal is carried out through a modified EIA RS-232-C serial I/O interface. There are three programmable interval timers which are used to set up the interrupt service routine (ISR), and therefore the sampling rate.

The debugging is done by using the TIBUG debug monitor which provides an interactive interface between the user and the TM990/101M-1 microcomputer. The TIBUG program is located on two TI 2708 EPROM's and also uses about 40 words (in this microprocessor every word is two bytes) of RAM. The TIBUG provides seven software routines and sets the Baud Rate automatically everytime the microprocessor is reset. This system is set to operate at 300 Baud. Some of the other important features of this microcomputer board are listed in Appendix A.

2.2. RTI-1241-S Board

To interface the digital identification and control system to the continuous time process simulation, analog to digital and digital to analog conversion is necessary. The analog devices RTI-1241-S board provides both the analog input and output.

The RTI-1241 series has been designed for simple and efficient interfacing of the analog signals to TM 990 microcomputers. The standard board used in the control system is described in Appendix A.
3. The Software

The software implementation for the identification and control of two SISO systems will be described here. These two systems are: i) a first-order system, and ii) a second-order system. The continuous-time transfer function representing these two systems are:

\[ G_1(s) = \frac{K}{s + r} \]  \hspace{1cm} (3.1)

\[ G_2(s) = \frac{K\omega^2}{s^2 + 2\zeta \omega + \omega^2}. \]  \hspace{1cm} (3.2)

Where \( \omega \) is the undamped natural frequency; and \( \zeta \), the damping ratio of the system [26].

The software required for identification and the control of \( G_1(s) \) will be developed and from that the software for the identification and the control of \( G_2(s) \) can be easily derived.

3.1. The First-Order Systems

The transfer function of the first-order system \( G(s) \), in discrete form can be represented by the difference equation

\[ y(k) = a_1 y(k-1) + b_1 u(k-1) \]  \hspace{1cm} (3.3)

where: \( a_1 = \exp(-rT) \), and \( b_1 = (K/r)[1 - \exp(-rT)] \).

To identify the parameters \( a_1 \) and \( b_1 \), the adjustable model

\[ d(k) = A(k-1)y(k-1) + B(k-1)u(k-1) \]  \hspace{1cm} (3.4)
will be used; where \( A(k-1) \) and \( B(k-1) \) are the estimates of \( a \) and \( b \).

From the definitions in Chapter II, the a'priori error is

\[
e(k) = y(k) - d(k) \quad (3.5)
\]

The a'posteriori error is

\[
s(k) = \frac{e(k)}{1 + a y(k-1)' y(k-1) + b u(k-1)' u(k-1)}. \quad (3.6a)
\]

For the scalar systems \( y(k-1)' = y(k-1) \), therefore

\[
s(k) = \frac{e(k)}{1 + a y(k-1)^2 + b u(k-1)^2}. \quad (3.6b)
\]

The parameters are adjusted according to the following algorithm:

\[
A(k) = A(k-1) + \alpha s(k) y(k-1), \quad (3.7)
\]
\[
B(k) = B(k-1) + \beta s(k) u(k-1). \quad (3.8)
\]

To control the process, the a'posteriori error computation and the parameter adjustment will remain the same, but the input to the process will be the controller \( u(k) \), where:

\[
u(k) = [B(k)]^{-1} [d(k+1) - A(k) y(k)]. \quad (3.9)
\]

The flowchart for the two modes of operation for the first order system are shown in Figures 3.1 and 3.2.
START

read A/D channel: Input u(k)
read A/D channel: Output y(k)

compute the adjustable model output
d(k) = A(k-1)y(k-1) + B(k-1)u(k-1)

compute: the error
e(k) = y(k) - d(k)
s(k) = e(k)/(1 + ay(k-1)^2 + bu(k-1)^2)

update the model parameters
A(k) = A(k-1) + cs(k)y(k-1)
B(k) = B(k-1) + ps(k)u(k-1)

update the input and the output
y(k-1) ← y(k) and u(k-1) ← u(k)

Fig. 3.1 Flowchart for the identification of a first order system
read A/D channel: Input $d(k)$
read A/D channel: Output $y(k)$

compute: the error
$e(k) = y(k) - d(k)$
$s(k) = e(k) / \left(1 + ay(k-1)^2 + bu(k-1)^2\right)$

update the model parameters
$A(k) = A(k-1) + os(k)y(k-1)$
$B(k) = B(k-1) + ps(k)u(k-1)$

compute the controller value $u(k)$
$u(k) = \left[B(k)\right]^{-1}[d(k+1) - A(k)y(k)]$

update the input and the output
$y(k-1) \leftarrow y(k)$ and $u(k-1) \leftarrow u(k)$

Fig. 3.2 Flowchart for the control of a first order system.
3.2. The Second-Order Systems

In the case of a second-order process, the procedure for the identification and control of the first-order system can be extended to obtain the equations given below. The transfer function in discrete form is represented by the difference equation

\[ y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) . \]  (3.10)

where:

\[
\begin{align*}
a_1 &= 2\exp(-\zeta \omega T) \cos(\omega T) \\
a_2 &= -\exp(-2\zeta \omega T) \\
b_1 &= K \{1 - \exp(-\zeta \omega T) [\cos(\omega T) + (\zeta/(1-\zeta^2)^{1/2}) \sin(\omega T)]\} \\
b_2 &= K \{\exp(-\zeta \omega T) [\exp(-\zeta \omega T) - \cos(\omega T) + (\zeta/(1-\zeta^2)^{1/2}) \sin(\omega T)]\} .
\end{align*}
\]

The problem is to identify the four parameters \( a_1, a_2, b_1, b_2 \).

- **Identification:** The adjustable model:

\[ d(k) = A_1(k-1)y(k-1) + A_2(k-1)y(k-2) + B_1(k-1)u(k-1) + B_2(k-1)u(k-2) \]  (3.11)

The a'priori error:

\[ e(k) = y(k) - d(k) \]  (3.12)

The a'posteriori error:

\[ s(k) = e(k) / \{1 + \sigma[y(k-1)^2 + y(k-2)^2] + \rho[u(k-1)^2 + u(k-2)^2]\} \]  (3.13)

The parameters are updated by the algorithm given in equations (3.14a-d).
\begin{align*}
A_1(k) &= A_1(k-1) + c(s(k)y(k-1) \quad (3.14a) \\
A_2(k) &= A_2(k-1) + c(s(k)y(k-2) \quad (3.14b) \\
B_1(k) &= B_1(k-1) + b(s(k)u(k-1) \quad (3.14c) \\
B_2(k) &= B_2(k-1) + b(s(k)u(k-2) \quad (3.14d) \\
\end{align*}

- Control: In the control mode the controller \( u(k) \) must be computed. The desired output \( d(k+1) \) is available.

\[ u(k) = [B(k)]^{-1}[d(k+1)-A_1(k)y(k)-A_2(k)y(k-1)-B_2(k)u(k-1)] \quad (3.15) \]

The computation sequence for the second-order system has a flowchart similar to the one for the first-order systems. The only difference is the number of parameters being estimated, and therefore the amount of computation.

3.3. Arithmetic Operations

The implementation of the equations developed in the previous sections on the microprocessor requires the following arithmetic operations:

- **ADD:** The microprocessor performs the add operation in 2's complement form.
- **SUBTRACT:** Same as ADD.
- **MULTIPLY:** The TM 990 microcomputer has hardware multiply; but it is performed in natural binary. To perform the multiplication in 2's complement, a simple subroutine would do the job. The flowchart is shown in Figure 3.3 for two 16 bit
Fig. 3.3 Flowchart for two's Complement Multiplication.
numbers, A, B. The result is a 32 bit number, stored in 2 consecutive words.

- **DIVIDE:** Similar to multiply operation; but here a 32 bit number is divided by a 16 bit number. The same technique can be used to perform signed division.

### 3.4. The Sampling Rate

A very important function of the software is to set the sampling rate. This is done by the implementation of an Interrupt Service Routine (ISR), which takes advantage of the TMS 9901 programmable systems interface as an interval timer. The timer has a resolution of 21.33 microseconds. A 14 bit number will be counted down to zero (every decrement takes 21.33 microseconds), and the clock will be reloaded automatically. A maximum time of 349 milliseconds can be achieved. Table 3.1 lists some of the sampling rates and the corresponding hexadecimal codes. Upon the initialization of the system, the sampling rate is selected by the operator loading the proper hexadecimal code into the designated memory location TCONST. Any sampling rate can be created with a 21.33 microseconds resolution so that, for example, a sampling rate of 3 Hz or \( T = \frac{1}{3} \) seconds, is obtained by taking

\[
K = \frac{0.333}{21.33 \times 10^{-6}} = 15625
\]

The number \( K \) represented in hex, is 3D09. The counter uses bits 1-14. This number is therefore shifted to the left and a '1' is placed in 15th bit to enable the interrupt. Finally the hexadecimal code to be used by the operator would thus be 7A13.
<table>
<thead>
<tr>
<th>SAMPLES PER SECONDS</th>
<th>14 BIT CLOCK REGISTER</th>
<th>16 BIT CLOCK REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3D09</td>
<td>7A13</td>
</tr>
<tr>
<td>5</td>
<td>24A1</td>
<td>4963</td>
</tr>
<tr>
<td>10</td>
<td>1250</td>
<td>24A1</td>
</tr>
<tr>
<td>20</td>
<td>0928</td>
<td>1251</td>
</tr>
<tr>
<td>30</td>
<td>061B</td>
<td>0C37</td>
</tr>
<tr>
<td>40</td>
<td>0494</td>
<td>0929</td>
</tr>
<tr>
<td>50</td>
<td>03A9</td>
<td>0753</td>
</tr>
<tr>
<td>60</td>
<td>030D</td>
<td>061B</td>
</tr>
<tr>
<td>70</td>
<td>029D</td>
<td>053B</td>
</tr>
<tr>
<td>80</td>
<td>024A</td>
<td>0495</td>
</tr>
<tr>
<td>90</td>
<td>0209</td>
<td>0413</td>
</tr>
<tr>
<td>100</td>
<td>01D5</td>
<td>02AB</td>
</tr>
</tbody>
</table>

**TABLE 3.1** The sampling rates and their corresponding hexadecimal coded data.
3.5. The Computer Programme

The identification and control programmes were written in assembly language, and then transformed into machine codes. The code was transferred to the microcomputer on two EPROM's. The EPROM's were loaded using a programme written in Fortran which creates two files, the first file contains bits 0-7 and the second file contains bits 8-15 of every word.

The programmes for both the first-order and the second-order systems consist of five subroutines:
1) The Interrupt Service Routine (ISR): This routine is used to set the sampling rate. The ISR will use the rate provided by the system operator. At this time this rate is any number between 3 to 46875 interrupts/second with a 21.33 microseconds resolution. This rate can be further slowed down by repeating the counter countdown before the interrupt occurs.
2) A subroutine to compute the a'posteriori error $s(k)$. The routine actually computes $1/s(k)$, because this number will always be larger than unity.
3) A subroutine to update the parameters $A(k)$ and $B(k)$. This subroutine computes the increments $\Delta A(k)$ and $\Delta B(k)$ and adds them to the old parameters.

A good choice of the parameters $\sigma$ and $\beta$, used in the equations (2.29), (2.31), and (2.32) in Chapter II, will enhance the performance of the identifier. Computer simulation showed that small values of $\sigma$ and $\beta$ (0.1<$\sigma,\beta$<1.0) will slow down the convergence but also that the identifier performs better for these values than when used with larger values.
([a, b]>1.0). This is especially the case when the input changes are sudden, as in a square wave reference, for example. For simplicity in the computation these parameters are selected as powers of two i.e. 2, 4, ... or 0.5, 0.25, ... so allowing the multiplication to be performed by shifting the registers to the left or right.

4) A subroutine which computes the predicted output d(k). This subroutine will be active only when the system is operating in the identification mode. The first call of the main programme is to this subroutine.

5) A subroutine to compute the controller u(k). This subroutine, called only in the control mode, is the last subroutine call in the main programme. At sampling instant k, this routine will read the (A/D) channel to obtain d(k+1). The value of the controller output u(k) is checked before being sent to the D/A converter and consequently, to the process. The following constraints are applied:

\[
\begin{align*}
&\text{a)} \quad \text{if } -2047 \leq u(k) \leq 2047 \quad u(k) = u(k) \\
&\text{b)} \quad \text{if } u(k) > 2047 \quad u(k) = 2047 \\
&\text{c)} \quad \text{if } u(k) < -2047 \quad u(k) = -2047
\end{align*}
\]

The above saturation constraints are necessary because of the conversion board 12 bit resolution. If the sampling rate is too high, the parameter B1(k) becomes very small and this leads to large controller values, inhibiting the convergence of the system. This is not a problem when identification is being performed because the inverse of B1(k) is not used.
The program to identify and control a first-order system is less than 400 words (800 bytes), and less than 600 words (1200 bytes) for the second-order systems. Intermediate values of the system input, output, predicted output, controller, and the parameters are stored, enabling the operator to inspect the gradual change of these parameters. The execution time for these programs allows a sampling rate of about 100 Hz for the first-order system and of about 50 Hz for the second-order systems.
IV. MICROCOMPUTER EVALUATION OF THE ADAPTIVE IDENTIFIER AND CONTROLLER

1. OVERVIEW

In this Chapter the implementation of the identifier and controller developed in Chapter III will be evaluated on the microcomputer system. The processes to be identified and controlled are simulated by using operational amplifiers as integrators and inverters. The effect of such factors as the sampling rate, and the choice of initial values for the process parameters will be investigated.

Since the identifier is designed to be used with the controller, the performance of the identifier is evaluated for the closed loop processes. The identified parameters may be biased because of the correlation between the input and the output, however, the process output in all cases converges to the input. For first-order processes this was not a significant problem and the estimated parameters are very close to the true parameters. The second-order processes are harder to identify and more sensitive to the selection of sampling rate and the initial values of the parameters. In both cases the estimated parameters do not converge to the correct values but oscillate close to the true values with a small bias.

This Chapter is divided into two parts. The first part evaluates the identification and control of first-order systems, and three different processes will be investigated. The second part evaluates the identification and control of
second-order systems. Figure 4.1 shows the system in the identification mode, and Figure 4.2 shows the system in its control mode.

2. First-Order Processes

A first-order process can be characterized by a difference equation

\[ y(k) = a_1 y(k-1) + b_1 u(k-1) \]  \hspace{1cm} (4.1)

where \( y \) and \( u \) are the output and the input respectively. To identify this process is to estimate the parameters \( a_1 \) and \( b_1 \). In the identification mode the operator must provide the sampling rate and some reasonable initial values for \( a_1 \) and \( b_1 \). The selection of these variables is briefly described here.

The sampling rate should be selected based on the information available from the process and the random input. If the process is unknown this rate can be selected only based on the input to get some knowledge about the process and then choose a new rate if desired. A good choice of sampling rate, although important, is not critical, any value 10 to 100 times the highest frequency of the input signal will produce satisfactory results. A very fast sampling rate, however, is not desirable because at very high rates the parameters \( a \) and \( b \) will reach their limits, \( a \rightarrow 1 \) and \( b \rightarrow 0 \).

The choice of initial parameters is also important. It should be noted that the identification computer program will
Fig. 4.1 The Block Diagram of the Adaptive System in Identification Mode.
4.2 The Block Diagram of the Adaptive System in the Control Mode.
use a set value (hard limit) any time it encounters an unreasonable value. Therefore, at every sample the parameters are compared and altered if they exceed the limit. For the first-order systems the estimated parameter $A(k)$ has an upper limit of 1.0. This is a valid assumption because if $A(k) > 1.0$ the system is unstable. The parameter $B(k)$ does not have a hard limit when the system is in identification mode, but has a lower limit on its magnitude when the system is in the control mode. This limit is a number greater than zero, i.e. $|B_1| > 0.05$.

Numerical analysis and computer simulation of different algorithms have shown that the input signal to the process must have enough information available for the algorithm to identify the process parameters. Such an input signal could be a white noise, a square wave or a sinusoidal inputs can also be used.

To evaluate the identification and control algorithm on the microcomputer system, three processes are simulated, $P_1$, $P_2$, and $P_3$ with the following transfer functions.

- $P_1 \quad G(s) = \frac{10.0}{s+5.0}$
- $P_2 \quad G(s) = \frac{0.5}{s+1.0}$
- $P_2 \quad G(s) = \frac{30.0}{s+20.0}$

These three processes will be identified and the control algorithm will be evaluated for the process $P_1$. 
2.1 Identification

The process P1 can be represented by a difference equation where the coefficients depend on the sampling rate and its parameters. Using the transformation formulas given in Chapter III, process P1 can be represented by the difference equation:

\[ y(k) = 0.37y(k-1) + 1.26u(k-1) \quad \text{at 5 samples/sec.} \]
\[ y(k) = 0.61y(k-1) + 0.78u(k-1) \quad \text{at 10 samples/sec.} \]
\[ y(k) = 0.78y(k-1) + 0.44u(k-1) \quad \text{at 20 samples/sec.} \]

To identify process P1 the sampling rates indicated in the above equations are used. The estimated parameters obtained are shown in Figures 4.3a, 4.4a, and 4.5a. These values are very close to the true parameters of the process. As can be seen, the parameters change with sampling rate. The sampling rate selected should not be too high. Since as the sampling rate is increased even further (>50), then the parameters converge to their limits i.e. \( A(k) \to 1.0 \), \( B(k) \to 0 \). Figures 4.3b, 4.4b, and 4.5b show the process output and the output predicted by the microprocessor. The predicted output closely follows the process output.

The sampling rate selected for P2 was 5 samples/sec., at this rate P2 can be characterized by the difference equation

\[ y(k) = 0.82y(k-1) + 0.09u(k-1) \]
Fig. 4.3a The estimated parameters

5 Samples/second.

Fig. 4.3b The process output and
The Predicted Output.
Fig. 4.4a The Estimated Parameters.

10 Samples/Second.

Fig. 4.4b The Process Output and The Predicted Output.
Fig. 4.5a The Estimated Parameters.

20 Samples/Second.

Fig. 4.5b The Process Output and The Predicted Output.
The estimated parameters are shown in Figure 4.6a, once again they are very close to the true values. In this case a very slow (low frequency) input signal was used. The predicted output shown in Figure 4.6b is noisy but still very close to the process output. A slower sampling rate could perhaps improve the process identification.

The process P3 is a faster process than P1 and P2, therefore, the sampling rate selected was 100 samples/second. At this rate the difference equation is given by:

\[ y(k) = 0.82y(k-1) + 0.27u(k-1) \]

The identification results for the process P2 are shown in Figure 4.7.

2.2. Control

The performance of the adaptive controller is evaluated and process P1 was selected for this test. Figure 4.8 shows the response of this process to a step input, it is desirable that the transient response be faster than the present response. This can be done by introducing the adaptive controller developed in Chapter III into the system. The reference input goes through this controller which in turn generates a control signal to the process, as shown in Figure 4.2. In the control mode higher sampling rates could be used to achieve a very fast response, however, the control signal may be very large and therefore impractical. Since the largest value handled by the
Fig. 4.6a The Estimated Parameters.

5 Samples/Second.

Fig. 4.6b The Process Output and The Predicted Output.
Fig. 4.7a The Estimated Parameters.

100 Samples/Second.

Fig. 4.7b The Process Output and The Predicted Output.
Fig. 4.8a The process Output for the First-Order System.

Fig. 4.8b The Reference Signal.
(A/D) or (D/A) conversion board is ±10 volts, care should be taken to prevent overflow. This may be done by scaling the data. As mentioned in Chapter III any time there is an overflow, the highest value is used for the controller, ±10 volts. In addition, in this mode of operation the parameter B1 has a lower limit to prevent overflow. For the process P1 this limit is set at 0.05. In the control mode, the parameters of the adaptive controller remain constant only if the steady state has been reached. Since the time to reach steady state depends on the control signal, if the control signal is limited then the system response will be slower.

The adaptive control of process P1 was carried out at different sampling rates; all results were satisfactory although slow rates introduced only slight improvements over the system with no controller. Figure 4.9 shows the response of the process with the adaptive controller operating at 20 samples/second. As is shown, the control signal changes dramatically when the reference input goes through a sudden change. To simplify the plotting of the data, the plot of the control signal is limited to ±4 volts, the limit is ±10 volts on the actual system. Figure 4.10 shows the control action at 50 samples/second. The result is still very satisfactory. Note that although the control signal is saturated for a greater number of samples this does not represent a longer time interval than that in Figure 4.9b. A faster rate can be achieved on this system but the saturation of the control level will eventually lead to instability.
Fig. 4.9a The process Output

Fig. 4.9b The Control Signal

20 Samples/Second.
Fig. 4.10a The Process Output.

Fig. 4.10 The Control Signal.

50 Samples/Second
3. Second-Order Processes

The closed loop transfer function of the second-order process simulated for this part is:

\[
\frac{C(s)}{R(s)} = \frac{39.5}{s^2 + 6.5s + 28.0}
\]

where: \( \omega = 5.28 \text{ rad/sec.} \) and \( \zeta = 0.61 \).

3.1. Identification

Using the formulation given in Chapter III, the above process can be represented by a difference equation

\[
y(k) = 0.71y(k-1) - 0.28y(k-2) + 0.49u(k-1) + 0.31u(k-2) \quad \text{at 5 samples/sec.}
\]

\[
y(k) = 1.33y(k-1) - 0.53y(k-2) + 0.16u(k-1) + 0.12u(k-2) \quad \text{at 10 samples/sec.}
\]

\[
y(k) = 1.66y(k-1) - 0.72y(k-2) + 0.05u(k-1) + 0.04u(k-2) \quad \text{at 20 samples/sec.}
\]

In the identification mode the parameters A1 and A2 have upper limits; the limits are: |A1| < 2.0 and |A2| < 1.0. This will ensure that all the roots are inside the unit circle. In addition, the constraint B2 < B1 will result in a minimum phase system. The identification was carried out with the system operating at the same sampling rates indicated in the above equations. The estimated parameters are shown in Figures 4.11a, 4.12a, and
Fig. 4.11a The Estimated Parameters.

5 Samples/Second.

Fig. 4.11b The Process Output and The Predicted Output.
Fig. 4.12a The Estimated Parameters.

Fig. 4.12b The Process Output and The Predicted Output.
Fig. 4.13a The Estimated Parameters.

Fig. 4.13b The process Output and The Predicted Output.
4.13a, along with the process output and the output of the prediction model, Figures 4.11b, 4.12b, and 4.13b. As can be seen, the parameters $B_1$ and $B_2$ become very small as the sampling rate increases. Again the bias in estimates of $B_1$ and $B_2$ does not prevent the reference model from tracking the process output.

3.2. Control

The process simulated for the evaluation of the identifier is used now to evaluate the adaptive controller. The parameters $A_1$ and $A_2$ have upper limits higher than the limits set in the identification mode, but the limits are low enough to prevent overflow in the arithmetic operations. The limit for parameter $B_1$ must be selected with care, a very small value ($0 < B_1 < 0.1$) could cause a very underdamped system with a large overshoot when a sudden change occurs in the reference input. However, a large value for the limit ($B_1 > 1.0$) will cause an overdamped response. For a second-order process the selection of the sampling rate is critical; a fast rate can cause instability and a noisy output. A slow rate, on the other hand, results in a highly underdamped response when $B_1$ has a very small limit, and an overdamped response when the limit is set at a large value. The control action was carried out at 10 samples/second, the parameter $B_1$ was limited to 0.5 from below and $A_1$ and $A_2$ had an upper limit of 2.0, and the reference input is a square wave. The result is shown in Figure 4.14. The adaptive controller has speeded up the response significantly when
Fig. 4.14a The Process Output. \( \alpha = 0.25 \)
\( \beta = 0.50 \)
10 Samples/Second.

Fig. 4.14b The Control Signal.
compared with Figure 4.13b. The simulated process parameters are changed so that it becomes a highly underdamped system. The step response of the process is now shown in Figure 4.15. For this process the adaptive controller can reduce the overshoot significantly. Parameter B1 has a lower limit since the process has a gain less than unity. The result is shown in Figure 4.16. As can be seen, although there is almost no overshoot, the process does not have a smooth transient response. The noise in the response, and slow response are caused by controller limit and quantization noise. In the control of the two processes described the parameters \( \alpha \) and \( \beta \) were selected based on the gain of the process.
Fig. 4.15a The process Output for the Second-order System.

Fig. 4.15b The Reference Signal.
Fig. 4.16a The Process Output.  
\[ \alpha = 0.5 \]
\[ \beta = 1.0 \]

10 Samples/Second.

Fig. 4.16b The Control Signal.
V. CONCLUSIONS

1. Summary

Adaptive control techniques have been reviewed and results in the area have been examined. The self-tuning regulators and model reference adaptive systems have received the greatest attention from researchers and control engineers. A method proposed by Martin-Sanchez was found to be computationally simple to implement on a microprocessor. The identification algorithm is essentially a simple form of the recursive least squares estimation method, and the controller was found to be similar to the minimum variance controller used in the self-tuning regulators. The method has been established as asymptotically hyperstable using the Popov hyperstability theorem. The hardware necessary for the implementation of this method on a TI TM 990 microprocessor was described; The software was written in assembly language and uses integer arithmetic. In Chapter IV the algorithm was implemented on the microcomputer system and evaluated. As was shown, the result for the system identification are good and the system performed reasonably well. The estimated parameters were very close to the true parameters but, at some sample points, the predicted parameters were noisy although convergence had been reached. This effect caused by the use of the integer arithmetic in the system, did not prevent the overall performance of the adaptive identifier being satisfactory.

The performance of the adaptive controller, although
acceptable, was not as good as the adaptive identifier, and to achieve a good performance, some a'priori knowledge about the structure of the process was often needed. The most important shortcoming of this adaptive controller is that the controller always tries to reach steady state, the desired output, in one sample. This action demands very large amplitude control signals which are unrealizable and can cause instability. Another problem with this algorithm lies in the difficulty of selecting the parameters $c$ and $p$ discussed in Chapter II. These parameters were found to be very important when the system is in the control mode. The simplest choice of these parameters is based on the gain of the process. In addition to problems associated with the algorithm, the integer arithmetic operations of the realization also induce errors into the system.

2. Future Work

The algorithm implemented on this microcomputer can be used as a reliable identification method, but the control algorithm is not a robust method and needs modification to provide a reliable controller. An analytical procedure for choosing the $c$ and $p$ parameters would be invaluable. However, the selection of these parameters must be made so as to maintain the economy of computation which is a very important aspect of this method for microcomputer implementation.

The microcomputer system could be combined with a floating point arithmetic operation board such as the Am 9511 arithmetic processing unit to provide more convenient and reliable
implementation. Another alternative would be to write routines which use the hardware multiply and divide capability of the TM 990 microprocessor to perform floating point arithmetic. Such methods, however, may be very slow. The addition of a high level language and bulk storage would add to the convenience of further expansion.

Self-tuning regulators are computationally awkward to implement on microcomputers, but the minimum variance controller is usually very reliable. It is therefore suggested that this microcomputer be used for the implementation of self-tuning regulators to provide a comparison with the present work.
APPENDIX A

The System Hardware

A.1. TM 990/101M-1 Microcomputer

Some of the important features of this microcomputer board are listed below:

- The TMS 9901 programmable system interface is a multifunctional component, which provides interrupt, I/O ports, and an interval timer. This component will be used to set up the ISR to provide a variable sampling rate.
- The TMS 9902 Asynchronous Communications Controller (ACC). This component provides an interface between the microprocessor and a serial asynchronous communication channel. The TMS 9902 ACC accepts EIA RS-232-C protocol, it therefore facilitates the communication between the terminal and the microprocessor serial I/O ports. The TMS 9902 ACC can also be used as an interval timer with lower resolution, 64 microseconds compared with 21.33 microseconds for the TMS 9901 PSI.

The TMS 9901 PSI and TMS 9902 ACC provide some other very useful functions which are described in detail in TI TMS 9900 Microcomputer Manual.

- 17 interrupt input are available on the TM 990/101M. Two interrupts are non-maskable and others are maskable. There are three interrupt sources on board: The two serial I/O ports, and the TMS 9901 PSI interval timer. The 15 maskable interrupts are
also automatically prioritized.

- Hardware multiply and divide is available on the TM 990 family. This feature was one of the reasons that this microcomputer was selected.

- The hardware registers available on the TM 990 boards are the program counter (PC), status register (ST), and the workspace pointer (WP). WP always contains the address of the first word of a set of 16 contiguous words which can be used like an accumulator. This feature is very useful when branching to a subroutine, because it allows the user to use a new set of 16 word workspace. When branching to a subroutine, the last three registers (R13, R14, R15) contain the old PC, WP, and ST register.

- MEMORY- The TM990-101M-1 microcomputer provides 2K words of RAM and 2K words of EPROM. It should be mentioned that the TIBUG monitor software occupies 1K word of EPROM, therefore only 1K word of EPROM is available for programming.

A.2. RTI-1241-S Board

The standard board used in the control system has the following features:

- The resolution is 12 bits. The coding of the data is jumper-selectable and in three forms; i) Natural Binary, ii) Offset Binary, and iii) 2's complement.

- There are 16 single ended (SE) or 8 differential (DIFF) input channels. These can be doubled on board or can be expanded to 256 SE or 128 DIFF input channels using an
expansion board.

- The software-programmable gain permits subranging over the gains of 1, 2, 4, 8.

- Memory-Mapped interface: It was mentioned earlier that this board has been designed to be used with a TM 990 microcomputer board. The host computer when interfaced with the RTI-1241 board, perceives the analog interface board as a block of eight words in memory. The communication is in the same way it stores and retrieves data. The computer is therefore free to do other things while the conversion is being performed.

To perform the (A/D) conversion:

(a) Set the GAIN,

(b) Set the MULTIPLEXER channel,

(c) Send the CONVERSION COMMAND (start the conversion).

There are only three instructions needed to do the (A/D) conversion. It takes about 35 microseconds to execute the above instructions.

To perform the (D/A) conversion:

(a) Send the data to one of the two DAC's.

Only one instruction is required. The DAC output range is jumper-selectable. In this system it is set for ±10 volts operation with 2's complement coding.
A.3. TI 745 Terminal

The communication between the microcomputer and the system operator is done through the TI 745 electronic data terminal. This terminal can be easily interfaced to the TM 990 microcomputer board using the EIA RS-232-C cabling. The 745 terminal comes with a modem which can connect the microcomputer board to another device, such as a large computer. The Baud Rate for this terminal can be either 110 or 300 Baud. The control system is set for the 300 Baud Rate.

A.4. TM 990/510 Card Cage

The TM 990/510 card cage has four slots, two of these slots contain the TM 990 microcomputer board and the RTI-1241 conversion board. The backpanel contains the address bus, data bus, and control lines to permit memory, I/O, and DMA expansion of CPU modules.

A.5. The Power Supply

The DC voltages needed for the system are:

\[ +5 \text{ V} \quad \pm 12 \text{ V} \quad \pm 15 \text{ V} \]

These voltages are provided by a GSC (GOF-2A-1T) triple output power supply. It provides:

\[ +5 \text{ V} \quad \text{at} \quad 5 \text{ Amps} \]
\[ \pm 15 \text{ V} \quad \text{at} \quad 1.5 \text{ Amps} \]
the ±12 V is obtained from ±15 V using two regulators.
REFERENCES


