STUDY OF ESTIMATION AND OPTIMIZATION

TECHNIQUES SUITABLE FOR MICROPROCESSOR ADAPTIVE CONTROLLERS

by

PETER SPASOV

B.A.Sc. (E.E.), University of Toronto, 1977

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

in

THE FACULTY OF GRADUATE STUDIES
(Department of Electrical Engineering)

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
May, 1979

© Peter Spasov, 1979
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 representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

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

Date May 17, 1979
ABSTRACT

Adaptive controllers are controllers that perform optimally in unknown or changing environments. One class of adaptive controllers are conventional controllers that tune themselves. This is done by estimating the plant system parameters and optimizing the controller based on these estimates. It is desired to have algorithms that are short in both program length and execution time so that implementation in a device such as a microprocessor is possible.

Generalized Geometric Programming (GGP) is used to optimize both PID control of a second order system and lead-lag compensation of a servomotor system. These algorithms normally converge in a few iterations. The parameters of a second order plant are estimated by two techniques. One technique involves curve fitting of a step response with cubic splines to find the coefficients of the characteristic equation. The other technique, called Walsh Function Parameter Identification, (WFPI) uses a square wave test input and finds the phase tangents by correlation of the output with Walsh Functions.

In general, each of these algorithms is estimated to require no more than 1000 lines of code with execution times of less than 1 second, once the measured data is available.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>SYMBOLS</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>ABBREVIATIONS</td>
<td>vi</td>
</tr>
<tr>
<td></td>
<td>LIST OF ILLUSTRATIONS AND TABLES</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>ACKNOWLEDGEMENTS</td>
<td>viii</td>
</tr>
<tr>
<td>I.</td>
<td>INTRODUCTION - ADAPTIVE CONTROL</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>OPTIMIZATION USING GENERALIZED GEOMETRIC PROGRAMMING (GGP)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.1 Introduction to GGP</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.2 Condensation of PID Control Problem</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.3 Optimization of the Dual Problem - PID Control</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>2.4 Servomotor Lead-Lag Network Compensation</td>
<td>25</td>
</tr>
<tr>
<td>III.</td>
<td>PARAMETER IDENTIFICATION USING CUBIC SPLINES</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3.1 The Problem of Parameter Identification</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3.2 Cubic Splines</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>3.3 Identifying the Second Order System</td>
<td>39</td>
</tr>
<tr>
<td>IV.</td>
<td>PARAMETER IDENTIFICATION USING WALSH FUNCTIONS</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>4.1 Development of the Method</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>4.2 Testing the Method</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>4.3 Conclusion for the WFPI Technique</td>
<td>61</td>
</tr>
<tr>
<td>V.</td>
<td>IMPLEMENTATION STUDIES</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>5.1 Feasibility of Implementation</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>5.2 Test of uP Programs</td>
<td>67</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>REFERENCES</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A - GGP Algorithms</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>APPENDIX B - Parameter Estimation Algorithms</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>APPENDIX C - Walsh Function Parameter Identification</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>APPENDIX D - PID Control and Lead-Lag Compensation</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>
SYMBOLS

Chapter 2

g, constraint and cost function for GGP
G, ln g, function for z = ln X space
X, primal variable
δ, σ, dual variables
t, time scaling factor
γ_n, ω_n, ξ_n, parameters for 2nd order system
K_c, T_i, T_d, parameters for PID control
K_p, τ_p, parameters for servomotor
K_c, τ_c, a, parameters for lead-lag network

Chapter 3

C_{i,j}, cubic spline coefficient
S(t), spline function
x, x̅, system output
y, first derivative of system output
z, second derivative of system output
a_1, a_2, γ_p, T_D, parameters for 2nd order system
α_k, d_k, D_k, r_k, R_k, s_k, variable for identification algorithm

Chapter 4

x, sine type Walsh function
y, cosine type Walsh function
xc, yc, correlation function
Î_i, integral segment
s_i, sign of integral segment
γ_p, ω_p, ξ_p, parameters for 2nd order system
ω, frequency of square wave test signal
tanθ, phase tangent
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D/A</td>
<td>digital to analog converter</td>
</tr>
<tr>
<td>GCP</td>
<td>generalized geometric programming</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>ISE</td>
<td>integral square error</td>
</tr>
<tr>
<td>μP</td>
<td>microprocessor</td>
</tr>
<tr>
<td>PID</td>
<td>proportional integral derivative controller</td>
</tr>
<tr>
<td>RAM</td>
<td>random access memory</td>
</tr>
<tr>
<td>ROM</td>
<td>read only memory</td>
</tr>
<tr>
<td>WFPI</td>
<td>Walsh function parameter identification</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS AND TABLES

### Figure

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Use of Adaptive Algorithms in a Controller</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Step Response for Optimal PID Control</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Lag Compensation Using GGP</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>Conventional Lag Compensation</td>
<td>33</td>
</tr>
<tr>
<td>2.4</td>
<td>Lead Compensation Using GGP</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Cubic Spline Curve Fitting</td>
<td>47</td>
</tr>
<tr>
<td>3.2</td>
<td>PID Tuning Based on Estimated 2nd Order System</td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>System Output in Response to a Test Signal</td>
<td>55</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>GGP Algorithm Results for PID Control of a 2nd Order System</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Comparison of Spline Approximation to Analytic Function</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Parameter Estimates of a 2nd Order System</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Delay Time Estimation</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Phase Tangent Values Determined from Test Input</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Determining Phase Tangents by Averaging Noisy Data</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>Parameter Estimations</td>
<td>60</td>
</tr>
<tr>
<td>5.1</td>
<td>Performance Estimates</td>
<td>65</td>
</tr>
<tr>
<td>5.2</td>
<td>Phase Tangents Determined with the TMS 9900</td>
<td>69</td>
</tr>
</tbody>
</table>
Many people and events in the Department have been invaluable for the completion of this work. In particular, the author acknowledges the following:

Dr. E. V. Bohn for expert supervision and introduction to the concepts of GCP, Spline Curve fitting, and WFPI without which there would of been no beginning.

Dr. M. Ito and Dr. P. Lawrence for expressing interest and help in the implementation aspects.

I. Stewart for much appreciated help in using the TMS 9900.

V. Wong for instruction and help in using the Tektronix µProcessor support system.
INTRODUCTION - ADAPTIVE CONTROL

Control
"To exercise restraint of direction upon the free action of; to dominate, command..."

Adapt
"Fit, adjust, (a thing to another); make suitable (to or for a purpose); modify, alter.
"Ive. Adapt oneself to conditions..."

The Oxford Dictionary

Adaptive Control Systems are devices that attain a defined, usually optimal, set of system states in an unknown and/or changing environment. There are many complex adaptive schemes. Some of the more sophisticated of these allow the structures of the control system to change. These abstract theoretical methods are generally too complicated to be widely used in applications. On the other hand, satisfactory control has historically been effected with simple, insensitive controllers such as PID controllers and lead-lag network compensators.

Taking advantage of this experience, the controllers themselves can be restricted to these forms, but with their control parameters adaptively changed. Such a hybrid approach can be adaptive in the sense that it is self tuning. If a plant system, i.e., a plant to be controlled, is unknown, an adaptive controller will identify it and, based on this acquired knowledge, set the control parameters to meet some specified optimum. One such scheme is shown in Figure 1.1.

Many identification and optimization algorithms exist with varying degrees of complexity. It is desirable to have simple algorithms, such that the program memory requirement is small enough to fit in a microprocessor and such that the execution time is short enough to allow a
Fig. 1.1 Use of Adaptive Algorithms in a Controller (One Example)
program to run in real time. This thesis concerns itself with the development of identification and optimization algorithms that are sufficiently simple to meet the above objectives. One of the parameter estimation schemes involves correlation of a system output with Walsh functions. Another scheme to be studied is a method using cubic splines. Once the parameters are known, a tuning (optimization) scheme based on Geometric Programming can be used. The actual controller is conventional, such as PID or lead-lag network compensation. The feasibility of implementing the algorithms on a microprocessor such as the TMS 9900 will be investigated. A lead-lag network compensator and part of a Walsh Function Parameter Identification (WFPI) algorithm for the microprocessor will be implemented.
2. OPTIMIZATION USING GENERALIZED GEOMETRIC PROGRAMMING (GGP)

2.1 Introduction To GGP

For those who are not familiar with the technique of GGP, an excellent introduction can be found in reference [17]. Many recent developments can be found in reference [6]. GGP is used to solve a class of problems in which signomials describe the cost function and the constraints. In general, using the Kuhn Tucker conditions to solve this class of problems would be very difficult since it involves solving nonlinear equations. GGP transforms this problem into that of solving a system of linear equations in dual space. Signomials have the following general form,

\[ g = \sum_{i=1}^{n} s_i C_{ij} \prod_{l=1}^{m} x_l^{a_{ij}} \]  

(2.1)

for \( C_i > 0, \quad s_i = \pm 1 \)

As will be shown later, the optimization problem for both PID control and lead-lag network compensation will involve a 2 term cost and a 2 term constraint with 3 independent variables.

That is,

\[ g_0^p = s_1 C_1 x_1 a_{11} x_2 a_{12} x_3 a_{13} + s_2 C_2 x_1 a_{21} x_2 a_{22} x_3 a_{23} \]  

(2.2A)

\[ g_1^p = s_3 C_3 x_1 a_{31} x_2 a_{32} x_3 a_{33} + s_4 C_4 x_1 a_{41} x_2 a_{42} x_3 a_{43} \]  

(2.2B)

where the superscript \( P \) designates primal function as opposed to the dual (i.e. transformed) function. These equations can now be transformed to a dual space in the form,
where the superscript \( D \) designates dual function. The dual variables \( \delta_k \) and \( \sigma_k \) in (2.3) are to be evaluated at a point of condensation, \( \mathbf{x} \), such that equations (2.3A, 2.3B) are linear approximations to \( G_0(z) \) and \( G_1(z) \) where \( z = \ln X \). The dual variables are defined by

\[
\delta_i = \frac{C_i x_1^{a_{11}} x_2^{a_{12}} x_3^{a_{13}}}{g_0(x)} \quad i = 1, 2 \tag{2.4A}
\]

\[
\sigma_{i-2} = \frac{C_i x_1^{a_{11}} x_2^{a_{12}} x_3^{a_{13}}}{g_1(x)} \quad i = 3, 4 \tag{2.4B}
\]

Often, the original primary problem is not in the signomial form of equations (2.2A, 2.2B). In such cases it is necessary to condense the functions into a convenient form such as equations (2.2A, 2.2B). It should be noted that equations (2.2A, 2.2B) are only one convenient form for the specific case of optimizing 3 variables with one constraint.

Condensation is the approximation of a signomial function into a positive monomial function. Details and theorems relating to condensation can be found in references [12] and [7]. Condensation results in the function and derivative values being equal at the point of condensation. It is another form of approximation, analogous to expanding a nonlinear function at a point by a Taylor series expansion.
The primary problem in z space is to solve

\[ \begin{align*}
G_0 \cdot P + \lambda G_1 \cdot P &= 0 \quad \text{(2.5A)} \\
G_1 &\leq 0 \quad \text{(2.5B)}
\end{align*} \]

where \( \lambda \) is the Lagrange multiplier in z space, and \( G = \ln g \). Since the log function is concave, i.e. \( g^D \) and \( \ln g^D \) have the same set of maximizing points, then one can optimize the dual problem by solving

\[ \begin{align*}
G^D_x &= 0 \quad \text{(2.6A)} \\
\text{where} \quad g^D &= g^D_0 \cdot g^D_1 \cdot \lambda 
\end{align*} \]

Taking the derivative of \( g^D \) and setting it to zero,

\[ \frac{\partial G^D}{\partial x_i} = (s_1 a_{11} \delta_1 + s_2 a_{21} \delta_2 + s_3 a_{31} \lambda \sigma_1 + s_4 a_{41} \lambda \sigma_2) \cdot g^D = 0 \quad \text{(2.7)} \]

\[ i = 1, 2, 3 \]

For \( x_1, g \neq 0 \), the following equations are to be solved,

\[ \begin{align*}
&\begin{align*}
s_1 a_{11} \delta_1 + s_2 a_{21} \delta_2 + s_3 a_{31} \lambda \sigma_1 + s_4 a_{41} \lambda \sigma_2 &= 0 \quad \text{(2.8A)} \\
s_1 a_{12} \delta_1 + s_2 a_{22} \delta_2 + s_3 a_{32} \lambda \sigma_1 + s_4 a_{42} \lambda \sigma_2 &= 0 \quad \text{(2.8B)} \\
s_1 a_{13} \delta_1 + s_2 a_{23} \delta_2 + s_3 a_{33} \lambda \sigma_1 + s_4 a_{43} \lambda \sigma_2 &= 0 \quad \text{(2.8C)}
\end{align*} \]

and to satisfy the orthogonality conditions,

\[ \begin{align*}
&\begin{align*}
s_1 \delta_1 + s_2 \delta_2 &= 1 \quad \text{(2.9A)} \\
s_3 \sigma_1 + s_4 \sigma_2 &= 1 \quad \text{(2.9B)}
\end{align*} \]

This gives 5 linear equations with 5 unknowns, \( \delta_1, \delta_2, \sigma_1, \sigma_2 \) and \( \lambda \). This condition of equal number of unknowns and equations is called a zero
degree of difficulty condition in the GGP literature. In general, it is possible to have more unknowns, hence a higher degree of difficulty. This would require using more complex linear programming techniques to find a maximum. The optimum primary variables can be found by a reverse transformation making use of equations (2.4A, 2.4B).

In flowchart form the general strategy for optimization is the following:

\[ \begin{align*}
\text{x}^{(0)} & \quad \rightarrow \quad \text{Condensation} \\
& \quad \downarrow \quad \text{i.e. Find C_i's, s_i's and a_ij's} \\
& \quad \downarrow \quad \text{Solve Dual Problem} \\
& \quad \downarrow \quad \text{i.e. Find } \delta_i's, \sigma_i's \text{ and } \lambda_i's \\
& \quad \downarrow \quad \text{x}^{(i+1)} \\
& \quad \downarrow \quad \text{Solution if } g^P=g^D
\end{align*} \]

This general strategy will now be used to find a suitable algorithm for tuning a controller given the open loop parameters of a plant system.

2.2 Condensation of PID Control Problem

The problem condensed here is that of determining the controller parameters to give optimum response. In this case, the open loop 'plant' is (or is modelled as) a second order system with a transfer function of
\[ G_p(s) = \frac{\omega_n^2 \gamma_n}{s^2 + 2\zeta_m \omega_n s + \omega_n^2} \]  

(2.10)

Since the PID controller is commonly used in industry and it has well-known and stable characteristics, it was the type of controller chosen. As is well known, the transfer function is \([11,20]\)

\[ G_c(s) = K_c \left( 1 + \frac{1}{T_1 s} + T_d s \right) \]  

(2.11)

A common control problem is the servo problem in which the output tracks changes in the input, optimum control would thus minimize deviation of the output from the setpoint. Hence, a performance criterion of minimum square error to a step input is chosen.

In order to simplify the algebra, various definitions to be used in the further mathematical developments, are introduced now. These are:

\[ a_1 = 2\zeta_m \omega_n \]  

(2.12A) \[ b_1 = a_1 t_s \]  

(2.13A)

\[ a_2 = \omega_n^2 \]  

(2.12B) \[ b_2 = a_2 t_s^2 \]  

(2.13B)

\[ a_3 = \gamma_n \omega_n^2 \]  

(2.12C)

\[ \alpha_1 = a_2 + a_3 K_c \]  

(2.14A)

\[ \alpha_2 = a_1 + a_3 K_c T_d \]  

(2.14B)

\[ \alpha_3 = \frac{a_3 K_c}{T_1} \]  

(2.14C)

\[ X_1 = a_2 t_s \]  

(2.15A)

\[ X_2 = a_1 t_s^2 \]  

(2.15B)

\[ X_3 = a_3 t_s^3 \]  

(2.15C)
\[ b_3 = b_1^2 - 2b_2 \quad (2.16A) \]

\[ b_4 = b_1 + \frac{b_2}{2} \quad (2.16B) \]

\[ b_5 = 1 + \frac{b_1}{2} + \frac{b_2}{6} \quad (2.16C) \]

\[ b_6 = \frac{1}{2} + \frac{b_1}{6} \quad (2.16D) \]

t_s is a time scaling factor to set a time constraint and to keep the variables reasonable, numerically.

The technique of using GGP to solve this control problem was investigated by Bohn [12] and Carver[14], each of them handling a time constraint differently. Most of the definitions used here are the same except that Carver didn't use time scaling for the primal variables.

The closed loop error transfer function is given by

\[ E \frac{R}{s} = \frac{s^3 + a_1 s^2 + a_2 s}{s^3 + a_0 s^2 + a_1 s + a_3} \quad (2.17) \]

By use of the frequency translation \( p = s t_s \), (2.17) can be rewritten as

\[ E \frac{R}{p} = \frac{p^3 + b_1 p^2 + b_2 p}{p^3 + x_1 p^2 + x_2 p + x_3} \quad (2.18) \]

From (2.18), some rationale for choosing the previous definitions can be seen. The b's and X's are dimensionless quantities. The X's are the coefficients of the characteristic equation of the transfer function. The b's are related to the zeroes.

Applying the Routh Hurwitz criterion to the characteristic equation of (2.18), it can be seen that the stability requirement (negative roots) is that
\[
\frac{x_3}{x_1x_2} < 1 \quad (2.19)
\]

The integral square error can be found in the frequency domain by using Parseval's theorem [26]

\[
I = \int_0^\infty e^2(t)dt = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} E(s)E(-s)ds \quad (2.20)
\]

Define the cost to be

\[
g_0^P = \frac{2I}{t_s} \quad (2.21)
\]

The superscript P denotes a primal function. This is the function to be minimized using GGP. This definition permits a convenient dimensionless form for \(g_0\) and gives a time scale design parameter \(t_s\).

The expression for \(E(s)\) when \(R = \frac{1}{s}\) (a step input) can be determined by using (2.17). Then substituting \(E(s)\) into (2.20) and using (2.21) yields the cost function

\[
g_0^P \left[ \frac{1}{X_1} + \frac{b_3}{X_1X_2} + \frac{b_2^2}{X_2X_3} \right] \left[ 1 - \frac{X_3}{X_1X_2} \right]^{-1} \quad (2.22)
\]

Since all the variables are positive, it can be seen that in order to have \(g_0^P > 0\), it is necessary to have equation (2.19) satisfied.

If an attempt to find the minimum of \(g_0^P\) is made, a physically unrealizable solution of \(g_0^P = 0\) can be found, which requires some controller parameters to have infinite values. The linear model of the plant system would not then be applicable.

One way of circumventing this problem is to constrain the output to be some value \(C_s\), at some time \(t_s\) during the initial transient period.
The output is

\[ c(t) = 1 - e(t), \quad e(t) = \mathcal{L}^{-1} \mathcal{E}(s) \]  \hspace{1cm} (2.23)

where \( \mathcal{E}(s) \) can be found by dividing the denominator into the numerator of (2.17) for \( R = \frac{1}{s} \). Then, using the definitions in (2.15, 2.16) yields

\[ c(t_s) = -b_1 + X_1 + \frac{(-b_2 + X_2 + b_1 X_1 - X_1^2)}{2} \]

\[ + \frac{(-2 X_1 X_2 - b_1 X_1^2 + b_2 X_2 + b X_1^3)}{6} \]  \hspace{1cm} (2.24)

\[ + \frac{(2 X_3 X_1 - b_1 X_3 - b_2 X_2 + 2 b_1 X_1 X_2 + X_2^2 - 3 X_2 X_1^2 - b_1 X_1^3 + X_1^4)}{24} + \ldots \]

The tuning problem can now be stated as follows:

Find \( X_1, X_2, X_3 \) to minimize \( g_0 \) subject to \( c(t_s) = C_s \). Note that the controller parameters \( K_c, T_1, T_d \) can be found from \( X_1, X_2, X_3 \).

A constraint can now be defined as

\[ g_0 = \frac{C(t_s)}{C_s} = 1 \]  \hspace{1cm} (2.25)

A general technique for finding the optimum is by trying to satisfy the Kuhn Tucker necessary conditions. Even if only the first few terms of \( g_1^p \) are considered, the nonlinear system of equations would be extremely difficult to solve on a microprocessor. This is why a technique such as GGP is attractive.

With the primal problem defined, it is possible to proceed with the condensation. It is desired to have the cost function condensed into the form of (2.2A). The function to be condensed is described by (2.22).
Set
\[ u_1 = 1 - \frac{X_3}{X_1X_2} = u_{11} - u_{12} = \bar{u}_1 = \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} \frac{\delta_{11}}{\delta_{12}} \]

where
\[ \delta_{11} = \frac{u_{11}}{u_1} = \frac{1}{u_1}, \quad \delta_{12} = \frac{u_{12}}{u_1} = \frac{X_3}{X_1X_2} \]

(2.26)

The bar over the \( u \) refers to the condensed version of the original function. The condensed function approximates the original function as described in [12]. It is a tangent hyperplane to the original function in \( z=\eta\mathbf{n}X \) space. Note that condensation is performed in the manner described by Bohn [12].

\[ \bar{u}_1 = C_{11} \frac{b_{11}}{X_1} \frac{b_{12}}{X_2} \frac{b_{13}}{X_3} \]

(2.27)

where
\[ C_{11} = \left( \frac{1}{\delta_{11}} \right) \left( \frac{\delta_{11}}{\delta_{12}} \right) \]

\[ b_{11} = b_{12} = \delta_{12}, \quad b_{13} = -\delta_{12} \]

Continuing on with the condensation, set
\[ u_2 = 1 + \frac{b_3}{x_2} \]

(2.28)

Then similarly obtain
\[ \bar{u}_2 = C_{21} \frac{b_{21}}{X_1} \frac{b_{22}}{X_2} \frac{b_{23}}{X_3} \]

(2.29)

where
\[ C_{21} = \left( \frac{1}{\delta_{21}} \right) \left( \frac{b_3}{\delta_{22}} \right) \]

\[ b_{21} = b_{22} = 0, \quad b_{23} = -\delta_{22} \]

\[ \delta_{21} = \frac{1}{u_2}, \quad \delta_{22} = \frac{b_3}{x_2u_2} \]

with these two parts of the function condensed, then
\[ \bar{g}_0 = \left[ \frac{1}{X_1} \bar{u}_2 + \frac{b_2}{X_2X_3} \right] \frac{1}{u_1} \]

(2.30)
The condensed cost function finally is

\[ g_0 = C_1 X_1 a_{11} X_2 a_{12} X_3 a_{13} + C_2 X_1 a_{21} X_2 a_{22} X_3 a_{23} \]  \hspace{1cm} (2.31)

where

\[ C_1 = \frac{C_{21}}{C_{11}}, \quad a_{11} = -1 - a_{13}, \quad a_{12} = b_{22} - b_{12}, \quad a_{13} = -b_{13} \]  \hspace{1cm} (2.32A)

\[ C_2 = \frac{b_2^2}{C_{11}}, \quad a_{21} = -b_{11}, \quad a_{21} = -1 - b_{12}, \quad a_{23} = -1 - b_{13} \]  \hspace{1cm} (2.32B)

Parts of (2.32A) and (2.32B) can be rewritten using (2.27) and (2.29)

\[ a_{11} = -1 - \delta_{12} \quad a_{21} = -\delta_{12} \]
\[ a_{12} = -\delta_{22} - \delta_{12} \quad a_{22} = -1 - \delta_{12} \]
\[ a_{13} = \delta_{12} \quad a_{23} = -1 + \delta_{12} \]  \hspace{1cm} (2.33)

The next step is to condense the constraint into the form of (2.2B). Using the first 4 terms of (2.24) and (2.25) and the definitions of (2.16), the constraint then becomes

\[ g_1 = \frac{1}{C_s} (\frac{X_3}{3} \frac{X_1^3}{6} + b_5 X_1^2 - b_6 X_1 + X_2) \]  \hspace{1cm} (2.34)

Note that the \( \frac{X_3}{3} \) term was dropped. This is because it was later found out to be insignificant. Also, it is known that the integral action (specified by \( X_3 \)) has little effect on the initial portion of the transient response.

Condensation of the constraint yields

\[ c(t) = u_3 + u_4 \]  \hspace{1cm} (2.35)
where

\[ u_3 = -b_4 + b_5 X_1 - b_6 X_1^2 + \frac{X_1^3}{6} = -u_{31} + u_{32} - u_{33} + u_{34} \]

\[ u_4 = b_6 X_2 - \frac{1}{3} X_1 X_2 = u_{41} - u_{42} \]

It follows that

\[ g_1 = C_3 X_1^{a_{31}} + C_4 X_1^{a_{41}} X_2 \leq 1 \]  

\[ (2.36) \]

where

\[ C_3 = \frac{1}{C_s} \left( \frac{\delta_{31}}{b_4} \right) \delta_{32} \left( \frac{\delta_{33}}{b_6} \right) \delta_{34} \]

\[ C_4 = \frac{1}{C_s} \left( \frac{b_6}{\delta_{41}} \right) \delta_{42} \]  

\[ (2.37A) \]

\[ (2.37B) \]

\[ a_{31} = \delta_{32} - 2 \delta_{33} + 3 \delta_{34} \]

\[ a_{32} = 0 = a_{33} \]

\[ a_{41} = -\delta_{42} \]  

\[ a_{42} = \delta_{41} - \delta_{42} = 1 \]  

These equations are true only for b's > 0. Equation (2.36) is the constraint that will be used to solve the dual problem. Bohn [12] has independently shown that a constraint of the form \( g_1 = C_3 X_1 + C_4 X_2 \), gives an accurate representation of the time domain constraint \( c(t_s) = C_s \). This results in some obvious computational simplifications. \( C_3 \) and \( C_4 \) are found by iteratively solving an equation that is an approximation of the transient response. As will be shown later, the optimum can be found without explicitly using the form of (2.36).

Now that the problem is put into the appropriate condensed form, the step of solving the dual problem can be performed.
2.3 Optimization of the Dual Problem - PID Control

Since the cost and constraint are now in the form of (2.2A), (2.2B), it is just necessary to solve the linear equations of (2.8) and (2.9). Solving the 5 equations, one obtains

\[ \delta_1 = \frac{\gamma}{1 + \gamma}, \quad \delta_2 = \frac{\gamma}{1 + \gamma} \quad (2.39) \]

where

\[ \gamma = \frac{a_{23}}{a_{13}} \]

and

\[ \lambda \sigma_2 = -(a_{12} \delta_1 + a_{22} \delta_2) \quad (2.40A) \]
\[ \lambda \sigma_1 = -1(a_{11} \delta_1 + a_{21} \delta_2 + a_{41} \lambda \sigma_2) \quad (2.40B) \]

\[ a_{31} \]

Using equations (2.33), the dual variables can be simplified to

\[ (\lambda \sigma_2) = 2\delta_{12} + \delta_{22} - \delta_{12} \delta_{22} \quad (2.41) \]
\[ a_{31} = \delta_{32} - 2\delta_{33} + 3\delta_{34} \quad (2.42) \]

\[ \lambda = \frac{1}{a_{31}} + (\lambda \sigma_2) \left(1 + \frac{\delta_{42}}{a_{31}}\right) \quad (2.43) \]

Then for iteration purposes

\[ \sigma_2 = \frac{(\lambda \sigma_2)}{\lambda} \quad (2.44) \]
\[ \sigma_1 = 1 - \sigma_2 \quad (2.45) \]

Knowing the dual variables, it is possible to find the new primary variables. At an optimum, in dual space, the constraint is satisfied by the equality, hence \( g_1^D = 1 \).

Therefore

\[ x_1 = \left(\frac{\sigma_1}{a_{31}}\right) \quad (2.46) \]
\[ X_2 = \frac{\sigma_2}{C_4 a_{41}} \]  

(2.47)

\( X_3 \) can be found by taking the derivative of the primal cost and setting it equal to zero since the constraint is independent of \( X_3 \).

Hence

\[ X_3 = \frac{-b_2 X_1}{X_2 + b_3} (b_2^2 - \sqrt{b_2^2 + X_2^2 + b_3 X_2}) \]  

(2.48)

These values of \( X \) would then become the condensation point for the next iteration. The optimum is found when

\[ g_0^D = g_0^P \]  

\[ g_1^D = g_1^P = 1 \]  

(2.49)

It is desired that any algorithm for calculating control parameters be as simple as possible, but simplification often occurs at the expense of generality. If one is attempting to control a second order plant, then the following simplifications can be performed.

From (2.39), (2.41), (2.42) and (2.43), it can be seen that not all of the \( \delta \)'s are required to be known. By using the definitions of \( \delta_{12} \) and \( \delta_{22} \), one obtains

\[ \delta_{22} = \frac{b_3}{X_2 + b_3} \]  

(2.50)

and

\[ \delta_{12} = \frac{X_3}{X_1 X_2 - X_3} \]  

(2.51)

Then (2.41) can be rewritten as

\[ (\lambda \sigma_2) = \frac{2X_2 X_3 + b_3 X_1 X_2}{(X_1 X_2 - X_3)(X_2 + b_3)} \]  

(2.52)

Similarly by using the definitions of \( \delta_{3j} \)'s
If this type of optimization algorithm were to be done by a μP controller, then other simplifications such as eliminating exponential calculations would be desirable. In general, calculations of the form $X^Y$ are quite long. In a μP controller, these types of calculations could be speeded up with special hardware options. If they were to be done with software only (including software multiply & divide), they could require 0.5 seconds for each calculation. [32] This is a good incentive to eliminate exponential calculations.

These calculations occur during the calculation of dual costs and constants. The purpose of the algorithm is to determine the controller parameters. The calculation of cost is only required in order to determine if the optimum has been found. Various other methods could be used to detect an optimum.

One such method is checking if $X^{(k+1)} - X^{(k)} = 0$. Also, one of the Kuhn Tucker conditions is easy to use as a check for the optimum point.

$$\frac{dg_0}{dx_3} = \left( \frac{g_0}{x_1} - \frac{b_2^2}{x_3^2} \right) \left( 1 - \frac{x_3}{x_1 x_2} \right)^{-1} x_2 = 0 \quad (2.55)$$

Then a condition for an optimum is

$$g_0 = \frac{x_1 b_2^2}{x_3^2} \quad (2.56)$$
The remaining exponential calculations occur during the updating of the primal variables. The constraint constants can be rewritten as

\[ C_3 = \frac{U_3}{X_1} - a_{31} \]  \hspace{1cm} (2.57)

\[ C_4 = \frac{U_4 X_1}{X_2 C_s} - a_{41} = \left( \frac{3b_6 - X_1}{3C_s} \right) X_1^{-a_{41}} \]  \hspace{1cm} (2.58)

Substituting (2.57) into (2.46), \( X_1 \) can then be updated by

\[ X_1^{(k+1)} = \left( \frac{\sigma_1}{U_3} \right)^{a_{31}} X_1^{(k)} \]  \hspace{1cm} (2.59)

The other symbols are for their \( k \)th iteration values. Since \( g_1 = 1 \) at an optimum

\[ C_4 X_1^{(k+1)} \frac{a_{41}}{X_2^{(k)}} = 1 \]  \hspace{1cm} (2.60)

Substituting (2.58) into (2.60) and by assuming \( X_1^{(k+1)} = X_1^{(k)} \),

then

\[ \left( \frac{3b_6 - X_1}{3C_s} \right) X_2 = \sigma_2 \]  \hspace{1cm} (2.61)

Then finally

\[ X_2^{(k+1)} = \frac{3\sigma_2 C_s}{3b_6 - X_1^{(k+1)}} \]  \hspace{1cm} (2.62)

Also at an optimum

\[ \frac{\sigma_1 C_s}{U_3} = 1 \]  \hspace{1cm} (2.63)

Hence

\[ U_3 - \sigma_1 C_s = 0 \]  \hspace{1cm} (2.64)

\( U_3 \) is a cubic equation of \( X_1 \) only. Then, once \( \sigma_1 \) has been found, \( X_1 \) could be determined by solving a cubic equation. Using an iterative
technique for solving non-linear equations \([21]\), (2.64) can be solved. Thus, \(X_1\) is iteratively updated by the following,

\[
\frac{X_1(k+1)}{X_1(k)} = \frac{1}{b_5} \left[ a_1(k) c_s + b_4 + b_6 X_1(k)^2 - \frac{X_1(k)^3}{6} \right] \quad (2.65)
\]

\[X_1(k+1) = \theta(k) \frac{X_1(k)}{X_1(k+1)} + (1 - \theta(k)) X_1(k) \quad (2.66)\]

where

\[\theta(k) = \frac{1}{1 - \frac{\partial X_1}{\partial X_1}} \quad (2.67)\]

which is

\[\theta(k) = \frac{2b_5}{2(b_5 - 2b_6 X_1(k) + X_1(k)^2)} \quad (2.68)\]

If \(X_1(k)\) is close to the optimum, then one iteration will suffice.

A remaining problem is how to start the algorithm. A method for obtaining the initial condensation points will now be shown:

1) Find the minimum ISE with respect to \(X_1\).

\[\frac{\partial e^2}{\partial X_1} = 2e \frac{\partial e}{\partial X_1} \quad (2.69)\]

2) Obtain the first two terms of the error output from (2.24) to get

\[\frac{\partial e}{\partial X_1} = 0 = -1 - \frac{b_1}{2} + X_1 \quad (2.70)\]

Hence set

\[X_1(0) = 1 + \frac{b_1}{2} \quad (2.71)\]

3) Using the first terms of \(c(t)\) in (2.24),

\[X_2(0) = 2(c_s + b_1 - X_1(0)) + b_2 - b_1 X_1(0) + X_1(0)^2 \quad (2.72)\]

\(X_3(0)\) can be obtained from (2.48)
Now the algorithm is relatively easy to implement. It will be shown to be valid for a certain range of open loop parameters. Modifications or generalizations could be used for other ranges.

When using GGP, it is necessary that the primal variables be kept positive. Negative variables also imply negative PID controller settings. Choosing \( C_g = 0.5 \) and using (2.62) to ensure that \( X_1 \) is positive, then

\[
X_1 < 3b_6, \text{ i.e. } X_1 < -\frac{3}{2} + \frac{b_1}{2} \tag{2.73}
\]

There is an upper limit for \( t_s \) because the higher order terms become more significant as \( t_s \) increases and the approximations will become less valid.

By using the first few terms of \( e(t) \) and setting \( \frac{\partial e}{\partial X_2} = 0 \), obtain

\[
3X_1 = -\frac{3}{2} + \frac{b_1}{2} \tag{2.74}
\]

Averaging (2.71) and (2.74),

\[
X_1 = \frac{5 + 2b_1}{4} \tag{2.75}
\]

By substituting (2.75) into (2.73), an approximate bound on \( t_s \) is found to be

\[
t_s < \frac{3}{4a_1} \tag{2.76}
\]

It is necessary that \( a_1, a_2 \) are positive. Using (2.75) as an approximation for \( X_1 \) and substituting into (2.72), it is found that

\[
b_2 < \frac{1 + 16b_1 - 4b_1^2}{16} = b_1 - \left(\frac{b_1}{2}\right)^2 \tag{2.77}
\]

(2.77) includes all overdamped open loop responses and underdamped responses with \( \zeta > 0.6 \).
It is necessary to insure that $X_3$ is positive. If $b_2$ is assumed
to be small, then by using (2.48), $X_3$ can be approximated as

$$X_3 = b_2X_1\sqrt{\frac{X_2}{X_2 + b_3}}$$  \hspace{1cm} (2.79)

Hence require

$$X_2 + b_3 > 0$$  \hspace{1cm} (2.80)

If (2.77) is satisfied then (2.80) is also satisfied, thus insuring
that $X_3$ is positive. Therefore in summary the conditions can be
listed as

$$t_s < \frac{3}{4a_1}$$ \hspace{1cm} (2.81A)

$$a_1 > 0, \ a_2 > 0$$ \hspace{1cm} (2.81B)

$$b_2 < b_1 - \left(\frac{b_1}{2}\right)^2$$ \hspace{1cm} (2.81C)

Using the equations of (2.14) and (2.15), the controller parameters
are found by

$$K_c = \frac{1}{a_3t_s^2} (X_2 - b_2)$$  \hspace{1cm} (2.82A)

$$T_d = \frac{t_s(X_1 - b_1)}{(X_2 - b_2)}$$  \hspace{1cm} (2.82B)

$$T_1 = \frac{t_s(X_2 - b_2)}{X_3}$$  \hspace{1cm} (2.82C)

In order to demonstrate how rapidly the algorithm converges, a
numerical example will be given.

**Example**

Data: $a_1 = 0.25 \quad a_2 = 0.01 \quad \gamma = 1$

Choosing $C_s = 0.5 \quad t_s = 2.5$
yields $x_1(0) = 1.3125$

$x_2(0) = 0.59$

$x_3(0) = 0.0624$

Table 2.1 summarizes the numerical results and Figure 2.1 shows the step response of the optimum system. As can be seen from Table 2.1, convergence occurs very rapidly. Note that the initial condensation points are close to the optimum and would yield a step response similar to the optimum. Other data were tried with similar convergence rates. Summaries of the algorithms are listed in Appendix A.

If the open loop system has a time delay, the same algorithm can be applied assuming no time delay. Once $X_1, X_2$ and $X_3$ have been found, the controller settings may be determined by a pair of non-linear equations. The details of this method are given in [12].
GCP Algorithm Results For PID Control
Of A 2nd Order System

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$g_0$</th>
<th>$g_{oD}$</th>
<th>$g_1$</th>
<th>$g^D$</th>
<th>$g_{op}$</th>
<th>$X_1(\frac{b_2}{X_3})^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.217</td>
<td>0.710</td>
<td>0.0602</td>
<td>1.306</td>
<td>1.305</td>
<td>1.010</td>
<td>1.327</td>
<td>1.312</td>
<td>1.237</td>
</tr>
<tr>
<td>2</td>
<td>1.195</td>
<td>0.746</td>
<td>0.0597</td>
<td>1.309</td>
<td>1.309</td>
<td>1.000</td>
<td>1.310</td>
<td>1.310</td>
<td>1.310</td>
</tr>
<tr>
<td>3</td>
<td>1.192</td>
<td>0.753</td>
<td>0.0596</td>
<td>1.310</td>
<td>1.310</td>
<td>1.000</td>
<td>1.310</td>
<td>1.310</td>
<td>1.310</td>
</tr>
<tr>
<td>4</td>
<td>1.191</td>
<td>0.754</td>
<td>0.0596</td>
<td>1.310</td>
<td>1.310</td>
<td>1.000</td>
<td>1.310</td>
<td>1.310</td>
<td>1.310</td>
</tr>
</tbody>
</table>
Fig. 2.1 Step Response for Optimal PID Control
2.4 Servomotor Lead-Lag Network Compensation

Often, servomotor systems are controlled by lead-lag compensation networks. There are many classical techniques available for designing such systems based on certain specifications. \[26\] It is also possible to design such systems using the GGP techniques described in Section 2.3.

A servomotor is often modelled as

\[ G_p(s) = \frac{K}{s(1 + \tau s)} \quad (2.84) \]

The problem is to design a compensation network of the form

\[ G_c(s) = \frac{K_c(1 + \alpha c s)}{1 + \tau c s} \quad (2.85) \]

The following definitions are made

\[ K = \frac{\Delta K}{p c} \quad (2.86) \]

\[ X_1 \triangleq \frac{(\tau_c + \tau_p) t_s}{\tau_c \tau_p} \quad (2.87A), \quad X_2 \triangleq \frac{(1 + \alpha \tau_c) t_s^2}{\tau_c \tau_p} \quad (2.87B) \]

\[ X_3 \triangleq \frac{K \tau^3_s}{\tau_c \tau_p} \quad (2.87C), \quad X_4 \triangleq \frac{t_s}{\tau_c} \quad (2.87D), \quad a \triangleq \frac{t_s}{\tau_p} \quad (2.87E) \]

\[ g_0 \triangleq \frac{\Delta}{2(ISE)} \quad \tau_s \quad (2.88) \]

From equations (2.84) to (2.88) it can be found that, for a unit step input,

\[ g_0 = \frac{1 + \frac{X_1}{2AX_4} + a^2X_4^2}{X_1X_2X_1X_2X_2X_3} \quad (2.89) \]

\[ 1 - X_3/X_1X_2 \]
Again, a time constraint is necessary. For the transient response, an approximation can be made.

\[ G = \frac{G_c G}{1 + \tau_p s} \]  

\[ \hat{G} = \frac{\alpha K}{s(1 + \tau_p s)} \]  

Then the closed loop error for a step input is

\[ \hat{E}(s) = \frac{s + 2\zeta \omega_n}{s^2 + 2\zeta \omega_n + \omega_n^2} \]  

where \( \omega_n^2 = \frac{\alpha K}{\tau_p} \), \( 2\zeta \omega_n = \frac{1}{\tau_p} \) \( \text{(2.93)} \)

The parameter \( \tau_c \) has virtually no effect on the transient portion of the response. The initial portion of the transient is determined mainly by \( K \alpha \). A time constraint can be set in the following manner:

Define

\[ Y_1 \triangleq 2\zeta \omega_n s \]  
\[ Y_2 \triangleq (\zeta \omega_n s)^2 = \frac{\alpha K \tau_p^2}{s} \]  

The error at time \( t_s \) is approximated by taking the inverse Laplace transform of (2.92) and using the definitions of (2.94) and (2.95).

This yields

\[ e(t_s) = e^{-2}[\cos \theta + \frac{Y_1}{2\theta}] \]  
\[ \theta = \sqrt{Y_2 - \left(\frac{Y_1}{2}\right)^2} \]  

The approximation is close to the actual error during the rise time of the response. Using (2.87) and (2.95), we have
\[ Y_2 = X_2 - aX_4 = X_2 - aX_1 + a^2 \]  

(2.97)

Now a time constraint can be specified.

\[ s_1 = C_3X_1 + C_4X_2 = 1 \]  

(2.98)

where

\[ C_3 = \frac{-a}{Y_2 - a^2} \]  

(2.99)

\[ C_4 = \frac{1}{Y_2 - a^2} \]  

(2.100)

After specifying an \( e(t_s) \), \( Y_2 \) can be found by solving the following equation.

\[
Y_2 = \sigma^2 + \frac{e^{-2\sigma}}{e^{2(t_s)}} \left[ \sqrt{Y_2 - \sigma^2 \cos \sqrt{Y_2 - \sigma^2} + \sigma \sin \sqrt{Y_2 - \sigma^2}} \right]
\]  

(2.101)

\[
\sigma = \frac{Y_1}{a} = \frac{a}{2}
\]  

(2.102)

(2.101) was solved for various values of \( \sigma \) and \( e(t_s) = 0.5 \). It was found that

\[ Y_2 \approx 1 + a/2 \]  

(2.103)

This approximation was found to be within 2\% of the actual value of \( Y_2 \) in the range of \( 1.0 < a < 2.0 \).

Hence, solving for \( Y_2 \) can be simplified by using (2.103) as a starting point. It should be noted that from (2.101), \( Y_2 \) must satisfy the following condition.

\[ \frac{a^2}{4} \leq Y_2 \]  

(2.104)
After condensation, the cost becomes

\[
g = g_0 \lambda = \left( \frac{C_1 X_1 X_2 X_3}{\delta_1} \right)^{\delta_1} \left( \frac{C_2 X_1 X_2 X_3}{\delta_2} \right)^{\delta_2} \left( \frac{C_3 X_1}{\sigma_1} \right)^{\lambda \sigma_1} \left( \frac{C_4 X_2}{\sigma_2} \right)^{\lambda \sigma_2}
\] (2.105)

where direct use was made of the constraint (see (2.87))

\[
X_4 = a - X_1
\] (2.106)

The process of condensation and solving the dual problem is done in a manner similar to that of PID control. The algorithm is listed in Appendix A.3. There are a few differences, that are discussed below.

Once the dual variables have been found, the primal variables can be easily calculated.

\[
X_1 = \frac{\sigma_1}{C_3}
\] (2.107)

\[
X_2 = \frac{\sigma_2}{C_4}
\] (2.108)

\[X_3\] can be found by taking the derivative of the uncondensed function. There are limits for the algorithm to work. Again, the stability criterion requires that

\[
\frac{X_3}{X_1 X_2} < 1
\] (2.109)

Another stability condition can be derived from the Kuhn Tucker conditions. These are

\[g_0 x_1 + \mu g_1 x_1 = 0 \quad (2.110A)\]

\[g_0 x_2 + \mu g_1 x_2 = 0 \quad (2.110B)\]

\[g_0 x_3 = 0 \quad (2.110C), \quad g_1 = 1 \quad (2.110D)\]
We have

\[ g_{0x1} = \frac{-X_2X_3g_0 + (2X_1X_3 - 2aX_3 + 2a^2X_1X_4 + a^2X_4^2)}{h} \]  
(2.111)

\[ g_{0x2} = \frac{-X_1X_3g_0 + X_3}{h} \]  
(2.112)

\[ g_{0x3} = \frac{(2X_3 - X_1X_2)g_0 + (X_2 + X_1^2 - 2aX_4)}{h} \]  
(2.113)

where \( h = X_1X_2X_3 - X_3^2 \)  
(2.114)

and

\[ g_{1x1} = -aC_4 \]  
(2.115)

\[ g_{1x2} = C_4 \]  
(2.116)

Solving the first two Kuhn Tucker equations of (2.110) for \( \mu \), yields

\[ \frac{g_{0x1}}{a} + g_{0x2} = 0 \]  
(2.117)

Substituting in (2.117) by (2.111) and (2.112) and by solving for \( g_0 \) at the optimum, yields

\[ g_0 = \frac{X_2X_3 - 2aX_3 + 2a^2X_1X_4 + a^2X_4^2 + aX_3}{X_2X_3 + aX_1X_3} \]  
(2.118)

Assuming that \( X_3 > 0 \), implies a new stability condition.

\[ X_2 + aX_1 > 0 \]  
(2.119)

Using (2.97) with (2.119), a condition for \( Y_2 \) is,

\[ Y_2 < a(2X_1 - a) \]  
(2.120)

Since \( X_1 > a \), then a definitive condition is

\[ Y_2 < a^2 \]  
(2.121)

Though the general condition is described by (2.120), it is not very useful since \( X_1 \) is unknown initially. (2.121) is more useful in choosing a constraint that is realizable.
Another condition for an optimum can be found by solving (2.110C) for $g_0$.

$$g_0^* = \frac{X_2 + X_1^2 - 2aX_4}{X_1X_2 - 2X_3}$$  \hspace{1cm} (2.122)

This implies another stability condition.

$$X_1X_2 > 2X_3$$  \hspace{1cm} (2.123)

Then from (2.104) and (2.123), the following limit for $Y_2$ is obtained.

$$\frac{a^2}{4} < Y_2 < a^2$$  \hspace{1cm} (2.124)

This implies

$$\frac{1}{4\tau_p} < K_\alpha < \frac{1}{\tau_p}$$  \hspace{1cm} (2.125)

It should be noted that these limits are useful for computational purposes. It is possible to exceed the limits with $Y_2 > a^2$ as long as (2.120) is satisfied.

The algorithm was tested using several examples. One of them was an example taken from reference [26]. The parameters of this servo-system are

$$K_p = 100$$
$$\tau_p = 0.04$$
$$\tau_s = 0.05625 \text{ for } C_s = 0.5000$$

The results of the optimization algorithm are

$$Y_2 = 1.71 \quad a = 1.40625$$

Using $X_1^{(0)} = 1.70$

Obtain $X_1 = 1.6066$, $X_2 = 1.9917$

$\text{ISE} = 0.0430$
\[ K_c = 0.2436, \quad \tau_c = 0.2808, \quad \alpha = 0.8875 \]

The time response is shown in Figure 2.2. Verification of the optimum was determined by substituting the optimum settings into the Kuhn Tucker equations. Figure 2.3 shows the time response obtained from the conventional technique of lag network design. The graph is taken from p. 543 of ref. [26]. A comparison will show that GGP offers improved control. The response in Figure 2.3 has a higher overshoot and a slowly decaying term with a relatively large time constant of 0.46 seconds. Better control could be obtained with conventional design but GGP offers it more readily without going through a trial and error procedure. Note that the same constraint is satisfied by both compensators.

A lead design was obtained with \( Y_2 = 1.3125 \), and \( t_s = 0.05 \) with its response shown in Figure 2.4.
Fig. 2.2 Lag Compensation Using GGP
Fig. 2.3 Conventional Lag Compensation
Fig. 2.4 Lead Compensation Using GGP

\[ G(s) = \frac{K(1 + \alpha \zeta s)}{1 + \zeta s} \]

\[ G(s) = K_\zeta \]

\[ K_\xi = 0.1700 \]
\[ T_\xi = 0.2650 \]
\[ \alpha = 1.2400 \]
\[ \text{ISE} = 0.0432 \]
3. PARAMETER IDENTIFICATION USING CUBIC SPLINES

3.1 The Problem of Parameter Identification

In the previous chapters it has been shown that given the plant system parameters and an appropriate time constraint (specification), suitable settings for a controller can be obtained. To apply such controller design procedures requires a system model. This model must be identified. An excellent overview of the problems in identification and control are given in reference [2].

Often an appropriate 'type' of equation can be used to describe a system based on a priori knowledge of its dynamics, i.e. an appropriate model may be chosen. For example, a chemical process plant could be modelled as a non-linear system or linear approximations could be used. Since it is convenient to use systems described by linear differential equations, the remaining problem is to choose what the order of the system is. Once that is done, it will be necessary to determine the coefficients of the linear equations, i.e. identify the parameters.

In the literature describing state and parameter estimation, there is often some differences as to what is defined as a state and what is defined as a parameter. For the purposes of this thesis, the following definition will be chosen:

Parameter Identification: This involves determining the coefficients of a linear equation describing (modelling) an input/output system of the form

$$ a_0 + a_1 y + a_2 \frac{\partial y}{\partial t} + \ldots a_n \frac{\partial^n y}{\partial t^n} = b_0 + b_1 u + b_2 \frac{\partial u}{\partial t} + \ldots b_m \frac{\partial^m u}{\partial t^m} \quad (3.1) $$
For Adaptive Control, it would be desirable to have some parameter identification scheme that can be carried out in real time. Many algorithms have been proposed, the most typical of these being some form of extended Kalman filtering [27].

Alternate schemes have been proposed that make use of spline curve fitting techniques. This is the type of scheme that will be investigated in this chapter. An output response to some known input is measured. These measurements, typically noisy, are fitted with a smooth curve using splines. Parameter identification is performed by finding the coefficients of a differential equation that will give the best fit to the smoothed curve.

In this chapter, the technique to be studied will be that of identifying the parameters of a second order model with time delay, using cubic splines.

3.2 Cubic Splines

In order to obtain intuitive insight into the nature of splines, a quotation from ref. [18] will be given.

"Draftsmen have long used mechanical splines, which are flexible strips of an elastic material. The mechanical spline is secured by means of weights at the points of interpolation - historically called knots. The spline assumes that shape which minimizes its potential energy, and beam theory states that this energy is proportional to the integral with respect to arc length of the square of the curvature of the spline."

Essentially, instead of fitting all of the data with one analytical curve, segments (intervals) of data are fitted with a cubic spline function for each segment.
When the knots \((x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\) are given, the interpolating spline is a function such that

\[ S(t_i) = x(t_i), \quad i = 1, 2, \ldots, n \] (3.2)

such that

\[ J = \int_{t_1}^{t_n} (S(t))^2 \, dt \] (3.3)

is minimized.

The cubic spline has the additional property that \(s(t), \dot{s}(t)\) and \(\ddot{s}(t)\) are continuous. That is, at the knot points the following is true,

\[ S_{i-1}(t_i) = S_i(t_i) \quad \text{(3.4A)} \]
\[ \dot{S}_{i-1}(t_i) = \dot{S}_i(t_i) \quad \text{(3.4B)} \]
\[ \ddot{S}_{i-1}(t_i) = \ddot{S}_i(t_i) \quad \text{(3.4C)} \]

for \(i = 2, 3, \ldots, n\).

For higher order splines, these continuity conditions apply for their respective higher order derivatives. The \(J\) function also differs.

For the \(i^{th}\) interval, let the spline be represented by,

\[ S_i(t) = C_{0,i} + C_{1,i} \tau + C_{2,i} \tau^2 + C_{3,i} \tau^3 \] (3.5)

\[ \tau = t - t_i, \quad t_i < t < t_{i+1} \]

Given the knot points, there is a sequential algorithm which will determine the coefficients of the spline represented by (3.5). The details of this algorithm can be obtained from ref. [18]. The problem with this approach is that the knot points must be given. Since measurements are typically noisy, the above procedure must be modified to smooth the noisy data.
In ref. [25] this problem was approached by the use of modified cubic splines. A modified spline is found by minimizing

\[ J = \int_{t_0}^{t_1} (S(t))^2 dt + \sum_{i=0}^{L} W_i (S(t_i) - x(t_i))^2 \]  

(3.6)

The weighting factors \( W_i \) are chosen such that \( S(t_i) \) and \( x(t_i) \) do not deviate by more than some chosen confidence limit. This method can be seen to be a hybrid of a conventional spline and a weighted least-squares method.

The problem with weighted spline fitting is that it requires the solution of \( 2n \) equations where \( n \) is the number of knots hence its complexity can increase significantly with \( n \). These equations cannot be solved in an iterative form. Conventional spline fitting can be done iteratively so its complexity is independent of the number of knots. Only the number of iterations changes. Hence, it was decided that the simpler technique of conventional splines would be used.

Various attempts at system identification have been made using cubic splines [8, 9, 10, 25]. Some of these methods are quite complex and general [9, 10]. One is used to improve initial estimates from Kalman filtering [8]. The technique to be used in this thesis will be different.
3.3 Identifying the Second Order System

"It has been found that high order overdamped systems as often encountered in chemical process control can be represented to a fair accuracy by a second-order model containing transport lag." [24]

A second order system with time delay is an adequate model for many physical systems. It gives a fairly accurate description and it is relatively simple to use.

A system to be controlled is modelled by the transfer function

$$H(S) = \frac{\gamma a_2 e^{-TD S}}{S^2 + a_1 S + a_2} = \frac{X(S)}{U(S)}$$  \hspace{1cm} (3.7)

ie., its dynamics can be described by

$$\ddot{x} + a_1 \dot{x} + a_2 x = \gamma a_2 u(t-T_D)$$  \hspace{1cm} (3.8)

$$x(T_D) = 0 = \dot{x}(T_D) , \hspace{0.5cm} \ddot{x}(T_D) = \gamma a_2 \hspace{0.5cm} \text{for} \hspace{0.5cm} x = x(t)$$

The problem is to identify the parameters $\gamma, a_1, a_2, T_D$. Using the closed loop response would be considerably more complicated though it would be more convenient because identification could take place during control. It was decided to use the simpler approach of using an open loop response. The test input was chosen to be a step function.

$$u(t) = u_0$$  \hspace{1cm} (3.9)

Then

$$z + a_1 y + a_2 x = 0 \hspace{0.5cm} \text{for} \hspace{0.5cm} t < T_D$$  \hspace{1cm} (3.10A)

$$z + a_1 y + a_2 x = \gamma a_2 u_0 \hspace{0.5cm} \text{for} \hspace{0.5cm} t \geq T_D$$  \hspace{1cm} (3.10B)

where

$$y \triangleq \dot{x} , \hspace{0.5cm} z \triangleq \ddot{x}$$  \hspace{1cm} (3.11)
γ can be readily identified from the steady state response. From the final value theorem and (3.7) and (3.9)

\[
\lim_{t \to \infty} x(t) = \lim_{s \to 0} X(S) = \gamma y_0
\]

Hence

\[
\gamma = \frac{x(\infty)}{u_0}
\]

As will be shown later, \( T_D \) can be determined after \( a_1, a_2 \) have been found. For an open loop system, the time delay simply delays the output further in time with \( x(t) = x_0 \) for \( t < T_D \). In order to determine when (3.10b) applies under noisy conditions, one must have \( x(t) > TH \) where \( TH \) is a threshold value that is greater than the largest noise magnitude. With \( \gamma \) known, (3.10b) has two unknowns. The derivatives, \( y \) and \( z \) can be determined from the cubic spline curve fitting technique by taking the first and second derivatives of (3.5).

Using two measurements, \( a_1 \) and \( a_2 \) can be determined by

\[
\begin{pmatrix}
y_1 & -x_1 \\
y_{i+1} & -x_{i+1}
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2
\end{pmatrix}
= \begin{pmatrix}
-z_1 \\
-z_{i+1}
\end{pmatrix}
\]

(3.14)

where

\[
\bar{x} \triangleq x - \gamma u_0
\]

Define

\[
a = \frac{a_2}{a_1}
\]

(3.16)

Then, by solving (3.14)

\[
a_i = \frac{z_{i+1}y_i - z_iy_{i+1}}{\bar{x}_i z_{i+1} - \bar{x}_{i+1}z_i}
\]

(3.17)
Using a recursive form to average during many measurements, the \((N+1)\)th iteration is

\[
\alpha_{N+1} = \alpha_N - \frac{1}{R_{N+1}} (r_{N+1} \alpha_N + S_{N+1})
\]  

(3.18)

where

\[
r_{N+1} = \bar{x}_N z_{N+1} - \bar{x}_{N+1} z_N
\]  

(3.19)

\[
S_{N+1} = z_N y_{N+1} - z_{N+1} y_N
\]  

(3.20)

\[
R_{N+1} = \frac{1}{N} \sum_{i=1}^{N} (\bar{x}_i z_{i+1} - \bar{x}_{i+1} z_i)
\]  

(3.21)

\[= r_{N+1} + R_N\]

Using (3.10B) and (3.17)

\[
a_1 = \frac{-z}{\alpha x + y}
\]  

(3.22)

\[
a_2 = a a_1
\]  

(3.23)

For averaging, then

\[
a_{N+1}^1 = a_1 - \frac{1}{D_{N+1}} (d_{N+1} a_N^1 + z_{N+1})
\]  

(3.24)

\[
a_{N+1}^2 = \alpha_{N+1} a_1
\]  

(3.25)

where

\[
d_{N+1} = y_{N+1} + \alpha_{N+1} \bar{x}_{N+1}
\]  

(3.26)

\[
D_{N+1} = \frac{1}{N} \sum_{i=1}^{N} (\bar{x}_i y_{i+1} - \bar{x}_{i+1} y_i) = d_{N+1} + D_N
\]  

(3.27)

With \(\gamma, a_1, a_2\), known, the time delay \(T_D\) can be determined. Using discrete z transform techniques, a step response with arbitrary time delay can be
generated. These generated values can be compared to measured values and the time delay can be determined by their respective times.

The crucial part of the entire estimation scheme is the determination of $a_1$, $a_2$ which depend heavily on knowing $x_i$ and its derivatives $y_i$ and $z_i$.

In order for the equations of (3.14) to be independent, no more than two spline approximations $(x,y,z)$ from the same spline interval should be used. Because $x,y,z$ are continuous at the knot points, then just the knot points can be used. This makes the computation of $x$, $y$, $z$ simpler.

Then from (3.11) and (3.5) for the $i$th knot

\[ x_i = C_0_i \]  
\[ y_i = C_1_i \]  
\[ z_i = 2C_2_i \]

Hence, the spline coefficients can be used directly for the parameter identification equations. Another reason for using only the initial knot points becomes apparent when the following is considered.

Substitute the spline approximation into (3.8)

\[ (2C_2 + a_1C_1 + a_2(C_0-\gamma)) + (6C_3 + 2a_1C_2 + a_2C_1)\tau \\
+ (3a_1C_3 + a_2C_2)\tau^2 + a_2C_3\tau^3 = 0 \]

At $\tau=0$ have

\[ 2C_2 = -a_1C_1 - a_2C_0 + a_2\gamma \]  

Differentiating (3.8) and substituting in the spline approximation
\[(6C_3 + 2a_1C_2 + a_2C_1) + (6a_1C_3 + 2a_2C_2)\tau + 3a_2C_3\tau^2 = 0\]  \hspace{1cm} (3.31)

At \(\tau = 0\) we have

\[6C_3 = -2a_1C_2 - a_2C_1\]  \hspace{1cm} (3.32)

Then the only time when the spline satisfies (3.29) is

\[\tau = -3 \left( \frac{C_2}{3C_3} + \frac{a_1}{a_2} \right)\]  \hspace{1cm} (3.33)

Similarly to satisfy (3.31)

\[\tau = -2 \left( \frac{C_2}{3C_3} + \frac{a_1}{a_2} \right)\]  \hspace{1cm} (3.34)

If (3.8) is differentiated twice and solved for \(\tau = 0\), then

\[\frac{a_1}{a_2} = \frac{-C_2}{3C_3}\]  \hspace{1cm} (3.35)

By substituting (3.35) into (3.33) and (3.34), it can be seen that (3.8) can only be described by a spline at \(\tau = 0\). This does not demonstrate that other \(\tau\)'s can't be used, but if \(\tau = 0\) is chosen as a point for solving for \(a_1, a_2\) then other \(\tau\)'s cannot be used.

Due to noisy data, smoothing must be done before spline curve fitting is used. Since it is desirable to keep the algorithm as simple as possible, a simple averaging technique is used to smooth the data with the following equation,

\[x_k = \frac{1}{4} (x_{mk-1} + 2x_{mk} + x_{mk+1})\]  \hspace{1cm} (3.36)

where \(x_k\) = smoothed data

\(x_{mk}\) = measured data

The estimation algorithm is shown in Appendix B.1.
The question of how accurately a cubic spline can represent a second order response is important. A spline approximation was tried on a simulated system with \( a_1 = 0.25 \), \( a_2 = 0.01 \). The values of the function with its derivatives have been compared with its analytic response, ie.

\[
x = \gamma \left( 1 + \frac{1}{a-b} \left( b e^{-at} - a e^{-bt} \right) \right) \tag{3.37A}
\]

\[
y = \dot{x} = \frac{\gamma_{ab}}{a-b} \left( e^{-bt} - e^{-at} \right) \tag{3.37B}
\]

\[
z = \ddot{x} = \frac{\gamma_{ab}}{a-b} \left( a e^{-at} - b e^{-bt} \right) \tag{3.37C}
\]

where

\[
a, b = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \tag{3.37D}
\]

ie. \( a, b \) are the poles.

A plot of a spline fit to noisy data is shown in Figure 3.1 and numerical results are given in Table 3.1. It can be seen that there are some significant differences for the derivatives at the smaller values of time. The limitation in the accuracy of spline fit during the transient response will limit the accuracy of estimating \( a_1, a_2 \). The spline will fit steady state very well because it is almost a straight line. That is not very useful for estimation purposes because all stable second order systems have the same steady state values of \( y u_0 \), so one could not determine a particular system.

Increased accuracy can be obtained by making the spline intervals narrower. This works if the data is not noisy. If the intervals are narrowed, the spline approximation becomes more sensitive to noise, especially the derivatives.
There is a reason why the accuracy decreases when one approaches $t=0$. Consider the Taylor expansion of $x$ about $t=0$.

$$x(t) = -\frac{\gamma a_2 t^2}{2} - \frac{\gamma a_1 a_2 t^3}{6} + \frac{\gamma a_2 (a_1^2 - a_2^2) t^4}{24} - \ldots$$  \hspace{1cm} (3.38)

An accurate fit using a third order equation would be possible for only a very small range of $t$.

As mentioned before, parameter estimation can only be done during the transient part of the response. At steady state, i.e. $x, y, z = 0$, no further update occurs. In practice, once steady state is approached, it was found that $a_1, a_2$ kept on becoming incrementally larger, probably due to numerical error when using small values.

Since there is a time delay in general, the estimation should occur when it is known that the system is affected by an input. Hence, estimation is restricted between two output thresholds.

Table 3.2 shows the results of parameter estimation for $a_1=0.25$, $a_2=0.01$ with the two thresholds being $0.1\gamma$ and $0.9\gamma$. Similar behaviour was observed for other overdamped systems. It can be seen that the estimates are dependent on the spline interval, starting threshold and the number of times of smoothing chosen.

One convenient method of checking the accuracy of the estimates could be done by checking the ISE of the system. The closed loop error is known in terms of the parameters as in (2.22). It could also be compared with the open loop integral once the time delay is known.

Time delay can be estimated by simulating the response with the estimates of the other parameters and comparing them to measurements. The
algorithm listed in Appendix B.2 uses the knot points in order to find the
time shift of these points relative to the appropriate point of a zero time
delay simulation. If the exact values of $a_1$, $a_2$ were used, the exact time
delay would be found. Table 3.3 shows time delay estimates when the estimates
of $a_1$, $a_2$ are not exact.

The $T_D$ estimates become more inaccurate for larger $t$, hence the estimate should be restricted to a range of outputs, for example $t$ such that $0.1 < x_m < 0.6$. Otherwise, the time delay estimate varies only slightly due to small errors in the estimates of $a_1$, $a_2$.

It would be interesting to see how tuning of a PID controller would be affected by the estimates. The results for this test are shown in figure 3.2. For the response $x_3(t)$, the ISE is less than the case based on the true parameters. Instead $x_1(t_8) < x_3(t_8)$ which is why the ISE is smaller. The converse is true for $x_2(t)$. Note that the same relationship holds true for the overshoot.

In general, cubic splines offer a considerably simpler technique for estimation but the accuracy is limited.
Output x

\[ \ddot{x} + a_1 \dot{x} + a_2 x = u(t-T_p) y a_x \]

\[ a_1 = 0.25 \quad \gamma = 1.0 \]
\[ a_2 = 0.01 \quad T_p = 5.0 \text{ sec.} \]

Fig. 3.1 Cubic Spline Curve Fitting
Comparison of Spline Approximation to Analytic Function
(*refers to the spline approximation)

<table>
<thead>
<tr>
<th>t [sec]</th>
<th>X</th>
<th>$X^*$</th>
<th>y</th>
<th>$y^*$</th>
<th>$z$ [10^{-3}]</th>
<th>$z^*$ [10^{-3}]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>10</td>
<td>1.83</td>
</tr>
<tr>
<td>10</td>
<td>0.2364</td>
<td>0.2378</td>
<td>0.0314</td>
<td>0.0285</td>
<td>-0.217</td>
<td>4.95</td>
</tr>
<tr>
<td>20</td>
<td>0.5156</td>
<td>0.5165</td>
<td>0.0233</td>
<td>0.0242</td>
<td>-0.980</td>
<td>-1.360</td>
</tr>
<tr>
<td>30</td>
<td>0.7033</td>
<td>0.7026</td>
<td>0.147</td>
<td>0.0143</td>
<td>-0.710</td>
<td>-0.615</td>
</tr>
<tr>
<td>40</td>
<td>0.8197</td>
<td>0.8175</td>
<td>0.009</td>
<td>0.0089</td>
<td>-0.447</td>
<td>-0.456</td>
</tr>
</tbody>
</table>

Table 3.1

Parameter Estimates of a Second Order System
$a_1=0.25$, $a_2=0.01$, $TH=0.1\gamma$

<table>
<thead>
<tr>
<th>Noise Variance</th>
<th>$\hat{a}_1$</th>
<th>$\hat{a}_2$</th>
<th>Spline Interval [sec]</th>
<th># of times of smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.2397</td>
<td>0.00927</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.2385</td>
<td>0.00970</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.2078</td>
<td>0.00748</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.2277</td>
<td>0.00998</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.2320</td>
<td>0.00860</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.267</td>
<td>0.0103</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0.0115</td>
<td>0.341</td>
<td>0.0144</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

*1 $TH=0.25\gamma$

*2 $TH=0.25\gamma$, $a_1=0.35$, $a_2=0.015$

Table 3.2
Delay Time Estimation

\( a_1 = 0.25, \ a_2 = 0.01, \ T_D = 10 \text{ sec}, \ \text{Noise Variance} = 0.0115 \)

<table>
<thead>
<tr>
<th>Knot Point</th>
<th>( \hat{a}_1 = 0.23, \ \hat{a}_2 = 0.0095 )</th>
<th>( \hat{a}_1 = 0.27, \ \hat{a}_2 = 0.0105 )</th>
</tr>
</thead>
</table>
| \( X_c \)  | \( T_c \ [\text{sec}] \) | \( X_c \) | \( t \ [\text{sec}] \) | \( \hat{T}_D \ [\text{sec}] \) | \( X_c \) | \( t \ [\text{sec}] \) | \( 
T_D \ [\text{sec}] \) |
| .100       | 15.6 | \( .100 \) | 5.7 | 9.9 | \( .102 \) | 5.6 | 10.0 |
| .407       | 25.6 | \( .410 \) | 15.7 | 9.9 | \( .409 \) | 15.9 | 9.7 |
| .633       | 35.6 | \( .634 \) | 25.1 | 10.5 | \( .634 \) | 26.3 | 9.3 |
| .775       | 45.6 | \( .775 \) | 34.4 | 11.2 | \( .775 \) | 36.7 | 8.9 |
| .862       | 55.6 | \( .862 \) | 43.7 | 11.9 | \( .862 \) | 47.3 | 8.3 |
| .914       | 65.6 | \( .914 \) | 52.8 | 12.8 | \( .914 \) | 57.7 | 7.9 |

Table 3.3
System Response $x(t)$

- $x_1(t)$ - tuning based on $\hat{a}_1 = .25$, $\hat{a}_2 = .01$
  $X_3 = 0.0596$, $X_7 = 1.191$, $X_2 = 0.7538$, ISE = 1.675

- $x_2(t)$ - tuning based on $\hat{a}_1 = .23$, $\hat{a}_2 = .0095$
  $X_3 = 0.0547$, $X_7 = 1.148$, $X_2 = 0.6966$, ISE = 1.791

- $x_3(t)$ - tuning based on $\hat{a}_1 = .27$, $\hat{a}_2 = .0105$
  $X_3 = 0.0640$, $X_7 = 1.235$, $X_2 = 0.8106$, ISE = 1.575

Real System parameters - $a_1 = .25$, $a_2 = .01$

Fig. 3.2 PID Tuning Based on Estimated Parameters, 2nd Order System
4. PARAMETER IDENTIFICATION USING WALSH FUNCTIONS

4.1 Development of the Method

In the spline identification method, the parameters were essentially identified using the system response in the time domain. Since systems have a characteristic frequency spectrum, information about the parameters can also be obtained using a set of orthonormal square wave functions which are called Walsh functions. They have been used in various control applications [15].

Recently some work has been carried out on the use of Walsh functions in system identification[13]. A procedure based on correlation functions will be investigated here. Essentially it involves testing the system with a square wave and so is easily implemented. Identification is done by correlation of the output with various known square wave signals, hence, due to the correlation process, the estimates are less sensitive to noise. Also it has the property of being simple. This method was tested for second order systems.

The basic theoretical background can be found in ref. [13]. The system is given a square wave input with a frequency of $\omega_1$. Time average correlation is performed by using various Walsh functions.

These are

$$\overline{uc} = \lim_{N \to \infty} \frac{1}{2N} \int_{-N}^{N} u(t)c(t)dt$$

(4.1)

where

- $c(t)$ = output
- $u(t)$ = known signal

The known signals are Walsh functions.
\( x_1 \) - square wave with \( \omega = \omega_1 \)

\( x_3 \) - square wave with \( \omega = \omega_3 = 3\omega_1 \)

\( x_5 \) - square wave with \( \omega = \omega_5 = 5\omega_1 \)

where \( x_1 \) is the original generated test input for the system. Other signals used are

\( y_1 \) - 90° phase shifted square wave with \( \omega = \omega_1 \)

\( y_3 \) - 90° phase shifted square wave with \( \omega = \omega_3 = 3\omega_1 \)

\( y_5 \) - 90° phase shifted square wave with \( \omega = \omega_5 = 5\omega_1 \)

The second order system has the following transfer function

\[
G_p(j\omega) = \frac{\gamma\omega^2}{(\omega_p - \omega)^2 + j2\zeta\omega_p\omega} = |G_p|e^{-j\theta} \tag{4.2}
\]

where

\[
tan\theta = \frac{2\zeta\omega_p\omega}{\omega^2 - \omega_p^2} \tag{4.3}
\]

It can be shown that the correlation functions can be approximated in terms of their harmonics of the form [13]

\[
x_{1c} = A_1\cos\theta_1 + \frac{A_3}{3}\cos\theta_3 + \frac{A_5}{5}\cos\theta_5 \tag{4.4A}
\]

\[
x_{3c} = A_3\cos\theta_3 \tag{4.4B}
\]

\[
x_{5c} = A_5\cos\theta_5 \tag{4.4C}
\]

and

\[
y_{1c} = A_1\sin\theta_1 + \frac{A_3}{3}\sin\theta_3 + \frac{A_5}{5}\sin\theta_5 \tag{4.5A}
\]

\[
y_{3c} = A_3\sin\theta_3 \tag{4.5B}
\]

\[
y_{5c} = A_5\sin\theta_5 \tag{4.5C}
\]

Since the signals are just plus or minus levels, the correlation can be easily performed numerically from the data.

Given the correlation functions, then phase information can be determined by the following.
\[
\tan \theta_1 = \frac{y_1c - y_3c/3 - y_5c/5}{x_1c - x_3c/3 - x_5c/5} \quad (4.6A)
\]
\[
\tan \theta_3 = \frac{y_3c}{x_3c} \quad (4.6B)
\]
\[
\tan \theta_5 = \frac{y_5c}{x_5c} \quad (4.6C)
\]

The parameters \( \zeta_p \) and \( \omega_p \) can then be determined by equating (4.6) and (4.3).

This yields
\[
\hat{\omega}_P^2 = \frac{(3 \tan \theta_1 - 9 \tan \theta_3) \omega_1^2}{3 \tan \theta_1 - \tan \theta_3} \quad (4.7A)
\]
\[
\hat{\omega}_P^2 = \frac{(45 \tan \theta_3 - 75 \tan \theta_5) \omega_1^2}{5 \tan \theta_3 - 3 \tan \theta_5} \quad (4.7B)
\]
\[
\hat{\omega}_P^2 = \frac{(25 \tan \theta_5 - 5 \tan \theta_1) \omega_1^2}{\tan \theta_5 - 5 \tan \theta_1} \quad (4.7C)
\]
\[
\hat{\zeta}_P = \frac{(\omega_1^2 - \omega_P^2) \tan \theta_1}{2 \omega_1 \omega_P} \quad (4.8A)
\]
\[
\hat{\zeta}_P = \frac{(9 \omega_1^2 - \omega_P^2) \tan \theta_3}{6 \omega_1 \omega_P} \quad (4.8B)
\]
\[
\hat{\zeta}_P = \frac{(25 \omega_1^2 - \omega_P^2) \tan \theta_5}{10 \omega_1 \omega_P} \quad (4.8C)
\]

The gain \( \gamma_p \), of a system can be determined from the steady state response due to a step input.
4.2 Testing the Method

A typical periodic output response for a square wave input is shown in Figure 4.1. The correlation functions were computed by integrating the data in lengths of 30 equal segments for the half period (T/2).

The value for each is obtained by

\[ u^c = \frac{s_1 I_1 + s_2 I_2 + \cdots + s_{30} I_{30}}{K} \]  

(4.9)

where 

\[ I_i = \text{integral of the } i^{th} \text{ segment} \]

\[ s_i = \pm 1 \text{ which determines the particular correlation function} \]

\[ K = \text{some convenient value} \]

This method is found to be relatively insensitive to noise because correlation is performed with a deterministic known signal. Any inaccuracies that had occurred are due primarily to the fact that only the first 3 harmonics of c(t) are being used. Numerical accuracy in determining the tanδ's would diminish at very high and very low values of \( \omega \) and at \( \omega = \omega_p \). When \( \omega_1 \) is too small, steady state is approached. The transient part of the response is then mainly determined by higher frequency components, i.e. \( \omega > \omega_5 \). Since these are not accounted for, then inaccuracy occurs. Similarly, if \( \omega_1 \) is too large then important lower frequency components are neglected.

The method was tested and some results are shown in Table 4.1. Data was sampled over one half period. The algorithm used is shown in Appendix C. The inaccuracies (>5%) due to the frequency used, are shown circled. For low input test frequencies, the accuracy of the lower harmonics diminish more rapidly. Conversely, for the higher input frequencies, the accuracy of the higher harmonics diminish more rapidly than the lower harmonics. Inaccuracies due to noise can be minimized.
Fig. 4.1 System Output in Response to a Test Signal
by averaging more data points over more periods. Noise plays a significant part in the estimation of tanθ's because the correlation function at lower fundamental frequencies become very small. For \( \omega_1 = 1 \), the correlation functions have the following values in the noiseless case.

\[
\begin{align*}
x_{5c} &= -0.008053 \\
y_{5c} &= -0.001845
\end{align*}
\]

With a noise variance of 0.0115, the errors of these values become significant. The input frequency could be increased to overcome the measurement noise problem.

There is a way of determining the accuracy of the phase tangent estimates based on a property of the WFPI technique. This property occurs when the system is tested 3 times with square waves of frequency \( \omega_1 = \omega_0, 3\omega_0, 5\omega_0 \). The matrix of phase tangents (cotangents) is symmetrical. Using equation (4.3), the matrix is found to be,

\[
\begin{bmatrix}
\tan \theta_1 & \tan \theta_3 & \tan \theta_5 \\
\frac{\omega}{\omega^2 - 1} & \frac{3\omega}{(3\omega)^2 - 1} & \frac{5\omega}{(5\omega)^2 - 1} \\
\frac{3\omega}{(3\omega)^2 - 1} & \frac{9\omega}{(9\omega)^2 - 1} & \frac{15\omega}{(15\omega)^2 - 1} \\
\frac{5\omega}{(5\omega)^2 - 1} & \frac{15\omega}{(15\omega)^2 - 1} & \frac{25\omega}{(25\omega)^2 - 1}
\end{bmatrix}
\]

where

\[
\omega = \frac{\omega_1}{\omega_0}
\]

Hence, the accuracy of the phase tangent estimate can be checked by using three test signals and the symmetry property.

By taking advantage of the periodic waveform, noise can be averaged
out. Estimates of the phase tangents with simulated noisy data is shown in Table 4.2. The noise was simulated by generating random numbers with a uniform probability distribution. Sufficient averaging will diminish the noise errors.

The parameter estimates determined from a test signal of \( \omega_1 = 0.903 \text{ rad./s} \) are shown in Tables 4.3. Errors in estimation are strongly dependent on errors of the phase tangent estimates. A more sophisticated algorithm would use weighting factors. In the example shown, \( \tan \theta_5 \) is subject to the most inaccuracy. Those equations of (4.7) and (4.8) which are more dependent on \( \tan \theta_5 \) should be weighted less. In the example, (4.7B) is the most influenced by \( \tan \theta_5 \). Eliminating this equation improves the accuracy significantly. A detailed investigation of weighting the equations is beyond the scope of this work.
Phase Tangent Values Determined from Test Input

Actual System Parameters: \( \gamma_p = 1, \rho_p = 0.5, \omega_p = 1.0 \)

Test input: \( u = 1, 0 < t < \frac{\pi}{\omega_1} \), half period

| \( \omega_1 \) [rad./sec] | Theoretical | | Computed-No noise | |
|--------------------------|-------------|--------------------------|
|                           | \( \tan \theta_1 \) | \( \tan \theta_3 \) | \( \tan \theta_5 \) | \( \tan \theta_1 \) | \( \tan \theta_3 \) | \( \tan \theta_5 \) |
| 20.0                     | 0.0501      | 0.167              | 0.0100              | 0.0497         | 0.0161         | 0.0096         |
| 15.0                     | 0.0670      | 0.0222             | 0.0133              | 0.0667         | 0.0218         | 0.0129         |
| 10.0                     | 0.1010      | 0.0334             | 0.0200              | 0.1001         | 0.0327         | 0.0194         |
| 4.0                      | 0.2667      | 0.0839             | 0.0501              | 0.2644         | 0.0823         | 0.0490         |
| 3.0                      | 0.3750      | 0.1125             | 0.0667              | 0.3720         | 0.1104         | 0.0651         |
| 2.0                      | 0.6667      | 0.1714             | 0.1010              | 0.6617         | 0.1682         | 0.0991         |
| 1.5                      | 1.200       | 0.2338             | 0.1357              | 1.191          | 0.2295         | 0.1332         |
| 1.25                     | 2.222       | 0.2871             | 0.1642              | 2.205          | 0.2819         | 0.1612         |
| 1.0                      | \( \frac{1}{0} \) | 0.3750             | 0.2083              | indeterminate | \( \frac{1}{0} \) | 0.3682         | 0.2050         |
| .903                     | -4.879      | 0.4275             | 0.2330              | -4.835         | 0.4198         | 0.2283         |
| .75                      | -1.714      | 0.5538             | 0.2871              | -1.677         | 0.5437         | 0.2819         |
| .5                       | -0.6667     | 1.200              | 0.4762              | \( \text{indeterminate} \) | -0.5730         | 1.173          | 0.4675         |
| .25                      | -0.2667     | -1.714             | 2.222               | \( \text{indeterminate} \) | -0.0424         | -1.772         | 2.148          |
| .1                       | -0.1010     | -0.3297            | -0.6667             | -0.036        | -0.2232        | -0.6544        |
| .05                      | -0.0501     | -0.1535            | -0.2667             | -0.0184        | -0.1069        | -0.1713        |

Computed-Noise Variance = 0.0115

Table 4.1
Determining Phase Tangents by Averaging Noisy Data

\[ \omega_p = 1.0, \quad \rho_p = 0.5 \]

\[ \omega_1 = 0.903 \]

<table>
<thead>
<tr>
<th>No. of Samples</th>
<th>( \tan \Theta_1 )</th>
<th>( \tan \Theta_3 )</th>
<th>( \tan \Theta_5 )</th>
<th>Noise Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>-4.835</td>
<td>.4198</td>
<td>.2283</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-4.839</td>
<td>.4125</td>
<td>.2704</td>
<td>.0115</td>
</tr>
<tr>
<td>8</td>
<td>-4.838</td>
<td>.4198</td>
<td>.2426</td>
<td>.0115</td>
</tr>
<tr>
<td>12</td>
<td>-4.837</td>
<td>.4176</td>
<td>.2156</td>
<td>.0115</td>
</tr>
<tr>
<td>12</td>
<td>-4.838</td>
<td>.4154</td>
<td>.2021</td>
<td>.0231</td>
</tr>
<tr>
<td>20</td>
<td>-4.835</td>
<td>.4236</td>
<td>.2389</td>
<td>.0231</td>
</tr>
<tr>
<td>8</td>
<td>-4.842</td>
<td>.4136</td>
<td>.3650</td>
<td>.0346</td>
</tr>
<tr>
<td>15</td>
<td>-4.837</td>
<td>.4121</td>
<td>.2053</td>
<td>.0346</td>
</tr>
<tr>
<td>30</td>
<td>-4.834</td>
<td>.4227</td>
<td>.2246</td>
<td>.0346</td>
</tr>
</tbody>
</table>

Table 4.2
Parameter Estimations

Noise Variance=.0346, 30 samples

Table 4.3A

<table>
<thead>
<tr>
<th>Estimates</th>
<th>Equation Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\omega}^2_P$</td>
<td>$\hat{\rho}_P$ ($\hat{\omega}_P=1.037$)</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>1.000</td>
<td>0.672</td>
</tr>
<tr>
<td>1.232</td>
<td>0.471</td>
</tr>
<tr>
<td>0.995</td>
<td>0.463</td>
</tr>
<tr>
<td>1.075</td>
<td>0.553</td>
</tr>
<tr>
<td>0.999</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Parameter Estimations

No Noise

Table 4.3B

<table>
<thead>
<tr>
<th>Estimates</th>
<th>Equation Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\omega}^2_P$</td>
<td>$\hat{\rho}_P$ ($\hat{\omega}_P=1.001$)</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>0.998</td>
<td>0.496</td>
</tr>
<tr>
<td>1.012</td>
<td>0.491</td>
</tr>
<tr>
<td>0.998</td>
<td>0.490</td>
</tr>
<tr>
<td>1.003</td>
<td>0.492</td>
</tr>
</tbody>
</table>
4.3 Conclusion For The WFPI Technique

Investigation of this type of parameter identification has only been begun recently. The preliminary investigations have been promising. It seems that any frequency that excites the system before steady state is reached, but giving enough time for significant transients to occur is adequate.

An iterative least squares procedure for rational function curve fitting to determine a transfer function from phase frequency response has been proposed[23]. A transfer function can be written in terms of even and odd functions of \( \omega^2 \).

\[
H(j\omega) = \frac{Q(\omega^2) + j\omega P(\omega^2)}{U(\omega^2)} \tag{4.11}
\]

The phase tangent is

\[
\tan \theta(\omega) = \frac{\omega P(\omega^2)}{Q(\omega^2)} = \frac{(p_0 + p_1\omega^2 + \cdots + p_M\omega^{2M})\omega}{q_0 + q_1\omega^2 + \cdots + q_N\omega^{2N}} \tag{4.12}
\]

Given the phase response data \( \tan \theta \) of a linear system, an iterative least squares rational approximation procedure can be used to find \( P(\omega^2)/Q(\omega^2) \). The transfer function can then be obtained from this function. This method is not very applicable for microprocessor control systems because it involves extensive frequency response tests and least squares curve fitting.

A generalization of WFPI to higher order systems has been proposed. It is possible to determine more phase tangents more accurately to identify higher order systems[13]. Instead of using a Fourier series, the output response is expanded into a Walsh function series expansion in the form of
\[ c(t) = a_1 x_1 + a_3 x_3 + a_5 x_5 + \cdots \]
\[ + b_1 y_1 + b_3 y_3 + b_5 y_5 + \cdots \]

where \( x, y \) are the sine and cosine type Walsh functions described in Section 4.1 and \( a, b \) are the expansion coefficients. These coefficients can be determined by correlation functions. The phase tangents can then be determined by using the coefficients. Using the phase tangents, a system of equations that is linear in the parameters can be derived. For example, the parameters of a third order system can be found by solving,

\[
\begin{pmatrix}
\tan \Theta_1 & \omega_1 & -\omega_1^2 \tan \Theta_1 \\
\tan \Theta_3 & \omega_3 & -\omega_3^2 \tan \Theta_3 \\
\tan \Theta_5 & \omega_5 & -\omega_5^2 \tan \Theta_5
\end{pmatrix}
\begin{pmatrix}
a_0 \\ a_1 \\ a_2
\end{pmatrix}
= 
\begin{pmatrix}
\omega_1^3 \\ \omega_3^3 \\ \omega_5^3
\end{pmatrix}
\]  

(4.14)

where

\[
H(s) = \frac{1}{s^3 + a_2 s^2 + a_1 s + a_0}
\]

(4.15)

Proposals for making use of magnitude information to decouple the even and odd parameters of a transfer function and estimating time delay have been also made [13].
5. IMPLEMENTATION STUDIES

The feasibility of implementing the adaptive control algorithms will be shown and tests with a lag compensation program and an integration program for the WFPI technique will be carried out.

5.1 Feasibility of Implementation

A list of applications of adaptive control systems can be found in ref. [30]. Many implementations have been of the type where a conventional controller such as PID, lead-lag compensators or variations of these are self tuning. The effectiveness of conventional controllers and their variants have been investigated by many researchers. A self tuning PID controller was investigated in references [4,5]. Digital implementation of PID does not attain the same quality as analog PID and a variant such as state vector feedback with integral action was proposed and investigated[4]. Extensive work was done on self-tuning controllers based on the minimum variance regulator—a control algorithm developed for setpoint control of discrete-time-randomly-distributed systems [16,19,31].

An adaptive (self-tuning) controller can be implemented based on the strategy outlined in Figure 1.1. The main work covered so far has been the development of the techniques for parameter estimation and control parameter tuning. The controllers themselves are conventional, having their development based on the history of analog control. Despite the fact that digital controllers could be "better" utilized, there is still usefulness in utilizing the conventional control algorithms, mainly because of their insensitivity, simplicity and familiarity.
The effectiveness on conventional controllers has been studied before. [3,5,28] Commercial success to date, with µP control using conventional algorithms has been demonstrated with such systems as the TDC-2000 system [11] and others. Adaptive versions of these have yet to be widely used.

The question may arise, how feasible is it to implement adaptive algorithms on-line, in particular, to implement them with a microprocessor? The answer is hard to predict before an actual implementation has been done. It depends on many factors such as the microprocessor used, the algorithms (and options) used and how they are implemented. Still, it is possible to estimate execution time and program size using certain assumptions.

The target microprocessor is assumed to be a TMS 9900 with a 3 M Hz clock. Table 5.1 shows the performance estimates for the various algorithms. These estimates were derived in the following manner. The number of arithmetic instructions required was totalled from the flow charts. The number of clock cycles was used to obtain a time [32]. These times will not be precise since other operations such as scaling, register transfers, branches and memory access etc. are required. Based on the execution time of the lag compensator program actually written, this factor was estimated to be approximately 5.5. The lines of machine code required was similarly extrapolated from the written compensator program. It should be noted that the estimates do not include data acquisition, operating interfacing and other peripheral operations particular to an entire control system.

Self tuning could be performed in a few seconds after open loop measurements have been obtained. If data were to be stored in RAM, it
## Performance Estimates

<table>
<thead>
<tr>
<th>Algorithm (Refer to Appendix)</th>
<th>Execution Time (General) [ms]</th>
<th>Probable Total Execution Time [ms]</th>
<th>Lines of Machine Code</th>
<th>Comments and Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>4+6.25/iteration</td>
<td>28</td>
<td>1025</td>
<td>3 iterations for convergence</td>
</tr>
<tr>
<td>A.2</td>
<td>6</td>
<td>6</td>
<td>600</td>
<td>One step</td>
</tr>
<tr>
<td>A.3</td>
<td>1.5+6.5/iteration</td>
<td>28</td>
<td>800</td>
<td>4 iterations for convergence</td>
</tr>
<tr>
<td>B.1</td>
<td>4/knot +(0.3/data)N + 2/spline interval</td>
<td>358</td>
<td>625</td>
<td>N=# of times of smoothing 10 knots, 9 spline intervals, 1000 pts. of data, N=1</td>
</tr>
<tr>
<td>B.2</td>
<td>0.25+1.5/data</td>
<td>1,500</td>
<td>175</td>
<td>1000 data pts.</td>
</tr>
<tr>
<td>C (Integral segments)</td>
<td>1+30h(0.15) per half period</td>
<td>N/A</td>
<td>50</td>
<td>h=# of samples per integral segment</td>
</tr>
<tr>
<td>C (continued)</td>
<td>4</td>
<td>12</td>
<td>920</td>
<td>test with 3 signals</td>
</tr>
<tr>
<td>PID D</td>
<td>0.5/data</td>
<td>N/A</td>
<td>50</td>
<td>done every Sample</td>
</tr>
<tr>
<td>Compensator D</td>
<td>0.5/data</td>
<td>N/A</td>
<td>50</td>
<td>done every Sample</td>
</tr>
</tbody>
</table>

Table 5.1
could be sampled in time increments as short as approximately 100 µs. A typical adaptive PID controller may use algorithms A.2, B.1 and B.2 so that the actual control part of a program would be slightly longer than 1 K which can be easily accommodated in a µP.

It is assumed that 16 bit integer data is acceptable. Increasing the precision would significantly increase both execution time and memory requirement. Admittably using a TMS 9900 makes implementation attractive because it is a 16 bit machine with hardware multiply and divide.

To see how this has an effect, a comparison can be made with a PID controller using a Motorola 6800[28]. This is an 8 bit machine. 2's complement arithmetic was performed using 16 bit integers. 12 bits were selected from the computed result to be an output to a D/A. This controller required an average of 7 ms to perform calculations used in the PID algorithm, mainly because multiplication had to be performed in software. When comparing this to the estimated execution time for using the TMS 9900 it can be seen that the tuning algorithms would be very long.

When designing a µP controller, one important question is which one to use. Since controllers operate in real time, they must be capable of doing numerical algorithms and I/O operations efficiently. Many microprocessors are available and many trade offs would be required in order to make a choice [1,29].

The µP chosen was the Texas Instruments TMS 9900. The main factor for this choice was the fact that it was readily available and has a good monitoring system (TIBUG) for debugging and an assembler (Tektronix 8002 µProcessor Lab). Other advantages are its 16 bit wordlength giving
greater accuracy, and the hardware multiply and divide. This reduces software complexity and execution time as mentioned previously.

5.2 Test of μP Programs

Two programs were written and run using off-line data. One program was the lead-lag compensation network mentioned previously. The other was a program to compute the integral segments (I₁ of eqn.(4.9)) for WFPI.

The compensator program was useful in estimating the requirements for implementing the other algorithms. A data acquisition system would be required to test it in real time. An off-line test was performed by running a simulation of a servomotor using optimum settings as determined by the GGP algorithm. The servomotor used has a time constant of \( \tau_p \approx 10 \) ms. The control parameters used were

\[ A_1 = 0.9474, \quad A_2 = 0.7366, \quad A_3 = 0.3821 \]

The servo response was then loaded into ROM. The compensator program reads this data every 1 ms, i.e., a simulated sampling from an A/D. The computed controller outputs were listed in RAM giving close results to the simulation results. Truncation error was responsible for the differences.

The WFPI program was tested in a similar manner. Data was generated by a 2nd order system simulation with noise. The data was converted into scaled hexadecimal integers and loaded into ROM. The integral segments were computed by the μP program. The correlation functions and phase tangents were computed off-line using these values by the method described in Chapter 4. It was tested using 16 and 12 bit (truncated) data since a data acquisition system could be typically 12 bits.
The results are shown in Table 5.2.

Truncation errors due to approximate representation of integrals by sums were found to be significant. A simple technique was used to correct for this. Given that there are $H$ samples to be summed to obtain an integral segment, the truncation error of the sum will be between 0 and $H$ because each sample will have an error between 1 and 0. Since truncation should only occur for rounding off decimal $< 0.5$, the average truncation error of the sum will be $H/2$. Hence, correction can be done by adding $sH/2$ to each segment where $s$ is the sign of the integral segment.

One can conclude that it is feasible to implement the optimizing and identification algorithms in present state of the art microprocessors. (1979)
Phase Tangents Determined with the TMS 9900

Actual System Parameters: $\gamma_p = 1$, $\zeta_p = 0.5$, $\omega_p = 1.0$

Test input: $u = 1$, $0 < t < \frac{\pi}{\omega_1}$

$-1, \quad \frac{\pi}{\omega_1} < t < \frac{2\pi}{\omega_1}$

Noise: Variance of 0.0115

<table>
<thead>
<tr>
<th>Case</th>
<th>$\tan \theta_1$</th>
<th>$\tan \theta_3$</th>
<th>$\tan \theta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical</td>
<td>-4.879</td>
<td>0.428</td>
<td>0.233</td>
</tr>
<tr>
<td>Simulation</td>
<td>-4.850</td>
<td>0.417</td>
<td>0.222</td>
</tr>
<tr>
<td>uP Results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Bit Data, Truncation Correction</td>
<td>-4.864</td>
<td>0.402</td>
<td>0.219</td>
</tr>
<tr>
<td>No Truncation Correction</td>
<td>-4.846</td>
<td>0.535</td>
<td>0.060</td>
</tr>
<tr>
<td>12 Bit Data, Truncation Correction</td>
<td>-4.885</td>
<td>0.400</td>
<td>0.221</td>
</tr>
<tr>
<td>No Truncation Correction</td>
<td>-4.866</td>
<td>0.531</td>
<td>0.065</td>
</tr>
</tbody>
</table>

Table 5.2
CONCLUSIONS

The algorithms for optimization and identification as listed in the appendices are relatively simple though the off-line analysis and program design is somewhat complex. By making use of these algorithms, a conventional controller can be self-tuning, hence adaptive. Generalized Geometric Programming (GGP) converges rapidly when calculating optimum controller settings given knowledge of the plant system parameters. The Walsh Function Parameter Identification (WFPI) method is good for parameter estimation for second order systems. Cubic spline approximations seem to be less accurate for parameter estimation. Implementation of these algorithms seems to be feasible for microprocessor based controllers.

Usually, investigations lead to further investigations and the subject of control algorithms is no exception. Further work can be carried out using GGP. There is the possibility of using different constraints, such as limiting the maximum overshoot. Optimizing other controllers may also be investigated.

Due to minor inaccuracies of cubic spline fit to a 2nd order system, it would be interesting to see if a simple system can be represented accurately by a cubic spline. Quite accurate results were obtained when estimating the time constant for a first order system. The reason for accurate results is that the higher order spline coefficients $C_0$ and $C_1$ (ie $x$ and $y$) are quite accurate but the higher order ones are subject to significant inaccuracy. Higher order systems require these coefficients in order to solve the equations for their coefficients. It would seem likely that more accurate coefficients, hence higher order splines are needed for estimating higher order systems. Unfortunately,
this would result in a more complex algorithm. Other variations such as specifying the endpoint derivatives of cubic splines could be tried.

In general, parameter identification using WFPI seems to be more promising. Extensions and generalizations for higher order systems based on equations (4.13) and (4.14) as mentioned in the conclusion of Chapter 4 could be further investigated. Only phase information was used and no attempt was made to make use of output magnitude as mentioned in Chapter 4. This could also be further investigated as well as looking for ways to determine time delay. Regarding the effectiveness of real life implementation, a definite answer can only be given if it is tried for a real physical process, though no insurmountable problem is foreseen.
REFERENCES


A.1

GGP Algorithm To Determine Controller Settings For PID Control Of
A 2nd Order System

Given \( a_1, a_2, \)

Choose \( t_s, C_s = 0.5 \)

Define

\[
\begin{align*}
    b_1 &= a_1 t_s, \\
    b_2 &= a_2 t_s^2, \\
    b_3 &= b_1^2 - 2b_2 \\
    b_4 &= b_1 + b_2, \\
    b_5 &= 1 + b_1 + b_2, \\
    b_6 &= 1 + b_1
\end{align*}
\]

Starting Values

\[
\begin{align*}
    x_1(0) &= 1 + b_1/2 \\
    x_2(0) &= b_1 + b_2 - b_1^2/4 \\
    x_3(0) &= \frac{-b_2 x_1(0)}{x_2(0) + b_3} \left[ \frac{b_2 - \sqrt{b_2^2 - x_2(0)^2 + b_3 x_2(0)}}{2} \right]
\end{align*}
\]
\text{\textbf{i}th Iteration} \quad i = 1

\begin{align*}
\lambda_2(i) &= \frac{2x_2x_3 + b_3x_1x_2}{(1 + x_1x_2)(x_2 + b_3)} \\
a_{31}(i) &= \frac{b_5x_1 - 2b_6x_1^2 + x_1^{3/2} - b_4 + b_6x_1 - b_6x_1^2 + x_1^{3/6}}{x_1x_2 + 3b_6x_2} \\
\delta_{42}(i) &= \frac{x_1x_2}{x_1x_2 + 3b_6x_2} \\
\sigma_2(i) &= \frac{a_{31}(i)(\lambda_2(i))}{1 + (\lambda_2(i))(a_{31}(i) + \delta_{42}(i))} \\
\bar{x}_1(i) &= \frac{1}{b_5} \left[ 1 - \frac{\sigma_2(i)}{2} + b_4 + b_6x_1^{(i-1)^2} - x_1^{(i-1)^3}/6 \right] \\
\theta(i) &= \frac{2b_5}{2(b_5 - 2b_6x_1^{(i-1)^2}) + x_1^{(i-1)^2}} \\
x_1(i) &= \theta(i)\bar{x}_1(i) + (1 - \theta(i))x_1(i-1) \\
x_2(i) &= \frac{1.5\sigma_2(i)}{3b_6 - x_1(i)} \\
x_3(i) &= \frac{-b_2x_1(i)}{x_2(i) + b_3} \left\{ b_2 - \sqrt{b_2^2 + x_2(i)^2 + b_3x_2(i)} \right\}
\end{align*}

Optimum is found if \( \frac{g_1}{x_1} = \left( \frac{b_2}{x_3} \right)^2, \ g_1 = 1 \) \quad i \leftarrow i + 1
\[ K_c = \frac{1}{a_3 t_s^2} (x_2 - b_2) \]

\[ T_d = \frac{t_s(x_1 - b_1)}{(x_2 - b_2)} \]

\[ T_l = \frac{t_s(x_2 - b_2)}{x_3} \]
A.2

A Quick One Step Algorithm To Determine Controller Settings For PID Control Of A 2nd Order System

If the accuracy obtained from one iteration is acceptable, then the following could be done

Choose \( t_s = \frac{1}{4a_1} \), \( C_s = 0.5 \)

By referring to Appendix A.1, have

\[
X_1(0) = 1
\]

\[
X_2(0) = b_2 + \frac{1}{4a_1^2} \text{ where } b_2 = \frac{a_2}{16a_1^2}
\]

\[
X_3(0) = b_2X_1\left[\frac{X_2}{X_2 + b_3}\right]^{0.5} \text{ where } b_3 = b_1^2 - 2b_2
\]

\[
\lambda \sigma_2 = \frac{2X_2X_3 + b_3X_2}{(1 + X_2)(X_2 + b_3)}
\]

\[
a_{31} = \frac{13 + 4b_2}{12 - 8b_2}
\]

\[
\delta_{42} = \frac{8}{21} , \quad b_6 = \frac{13}{24}
\]

\[
\sigma_2 = \frac{a_{31}(\lambda \sigma_2)}{(1 + (\lambda \sigma_2))(a_{31} + \delta_{42})}
\]

\[
\sigma_1 = 1 - \sigma_2
\]
\[ X_1 = \frac{12(\sigma_1 + b_2) - 9}{3 + 4b_2} \]

\[ X_2 = \frac{12\sigma_1}{13 - X_1} \]

\[ X_3 = b_2 X_1 \left[ \frac{X_2}{X_2 + b_3} \right]^{0.5} \]

Calculate \( K_c, T_d, T_1 \) (Appendix A.1)
A.3

GGP Algorithm For Servomotor

Lead-Lag Network Compensation

For definition of variables, see Section 2.4

\[
g_0 = \frac{1}{X_1} + \frac{1}{X_2X_3} \begin{bmatrix} -2aX_3 + 2a^2X_3 + a^2X_1^2 - 2a^3X_1 + a^4 + X_1X_3 \\ X_1 \\ \end{bmatrix} \begin{bmatrix} 1 - \frac{X_3}{X_1X_2} \\ \end{bmatrix}
\]

Given \( Y_2, t_s \) (\( Y_2 \) already solved for)

\[
C_3 = \frac{-a}{Y_2 - a^2}, \quad C_4 = \frac{1}{Y_2 - a^2}
\]

Choose initial \( X_1 \) (\( a < X_1 < 2a \))
APPENDIX B - Parameter Estimation Algorithms

B.1

Parameter Estimation Algorithm Using Cubic Splines

For $\ddot{x} + a_1 \dot{x} + a_2 x = \gamma a_2$, identify $a_1, a_2$

\[ x(k+1) = \frac{1}{4}(x_m(k) + 2x_m(k+1) + x_m(k+2)) \]

smooth data

cubic spline fit, starting at threshold

[see ref.[18] for algorithm]

ie find $C$ coefficients for

\[ x_k = C_{01} + C_{11} \tau + C_{21} \tau^2 + C_{31} \tau^3 \]

\[ \bar{x}_i = C_{01} - \gamma u_0 \]

\[ y_i = C_{11} \]

\[ z_i = 2C_{21} \]

$R_0 = 0$  $a_0 = 0$  $a_{10} = 0$  $D_0 = 0$
\[ r_{N+1} = \bar{x}_N y_{N+1} - \bar{x}_{N+1} y_N \]

\[ s_{N+1} = z_N y_{N+1} - z_{N+1} y_N \]

\[ d_{N+1} = \bar{x}_N y_{N+1} - \bar{x}_{N+1} y_N \]

\[ R_{N+1} = r_{N+1} + R_N \]

\[ D_{N+1} = d_{N+1} + D_N \]

\[ a_{N+1} = a_N - \frac{1}{R_{N+1}} (r_{N+1} a_N + s_{N+1}) \]

\[ a^1_{N+1} = a_1 N - \frac{1}{D_{N+1}} (d_{N+1} a^1 N + z_{N+1}) \]

\[ a^2_{N+1} = a_{N+1} a^1_{N+1} \]
B.2

Estimation of Time Delay

Have stored data (or just knot) points of spline fit \( (x_c(t_c)) \)
- or new measured data, given \( a_1, a_2 \) (already estimated)

\[ t = kT, \quad T = \text{sample time}, \quad k = 0, 1, 2, 3, \ldots \]

\( x_c(k_c), \quad k_c \) refers to a knot point

**START**

- \( x(k_c), \quad k_c \)

- \( x(k+1) = -A_1x(k) - A_2x(k-1) + (1 + A_1 + A_2)u_0y \)

  - if \( x(k+1) < x_c(k_c) \)

  \[ T_{D}^{N+1} = (k_c - k)T \]

  \[ z = z + T_{D}^{N+1} \]

**END**, \( T_{D} = \frac{z}{N+1} \)

where \( x(-1), x(-2), z_0 = 0 \)

\( A_1 = -(\alpha + \beta) \)

\( A_2 = \alpha \beta \)

\( \alpha = e^{-Ta} \)

\( \beta = e^{-Tb} \)

\( a, b = a_1 \pm \sqrt{a_1^2 - 4a_2}, \) overdamped case
APPENDIX C

Walsh Function Parameter Identification

Algorithm For A 2nd Order System

Input square wave of frequency \( \omega = \omega_1 \), period of \( T \)

Read system output

Integrate over time of \( T/60 \)

Correlation Calculation

\[
\begin{align*}
\overline{x_1}\text{c}_c &= \sum_{i=1}^{30} I_i \\
\overline{x_3}\text{c}_c &= \sum_{i=1}^{10} I_i - \sum_{i=11}^{20} I_i + \sum_{i=21}^{30} I_i \\
\overline{x_5}\text{c}_c &= \sum_{i=1}^{6} I_i - \sum_{i=7}^{12} I_i + \sum_{i=13}^{18} I_i - \sum_{i=19}^{24} I_i + \sum_{i=25}^{30} I_i
\end{align*}
\]
\[ y_{1c} = \sum_{i=1}^{15} I_i - \sum_{i=16}^{30} I_i \]

\[ y_{3c} = \sum_{i=1}^{5} I_i - \sum_{i=16}^{15} I_i + \sum_{i=6}^{25} I_i - \sum_{i=26}^{30} I_i \]

\[ y_{5c} = \sum_{i=1}^{3} I_i - \sum_{i=16}^{9} I_i + \sum_{i=4}^{15} I_i - \sum_{i=10}^{21} I_i + \sum_{i=16}^{27} I_i - \sum_{i=22}^{30} I_i \]

**Phase Tangents**

\[ \tan \theta_1 = \frac{y_{1c} - \frac{y_{3c}}{3} - \frac{y_{5c}}{5}}{x_{1c} - \frac{x_{3c}}{3} - \frac{x_{5c}}{5}} \]

\[ \tan \theta_3 = \frac{y_{3c}}{x_{3c}} \quad \tan \theta_5 = \frac{y_{5c}}{x_{5c}} \]

**Estimates**

\[ \hat{\omega}_p^2 = \frac{i^2 j \tan \theta_1 - i_j^2 \tan \theta_1}{j \tan \theta_1 - i \tan \theta_1} \omega_1^2 \]

\[ \xi_p = \frac{(i^2 \omega_1^2 - \hat{\omega}_p^2)}{2 \omega_1 \omega_1 \hat{\omega}_p} \tan \theta_1 \]

\[ i, j = 1, 3, 5 \quad \text{if} \quad i \neq j \]
APPENDIX D

PID Control and Lead-Lag Compensation

These are conventional control schemes that are insensitive with respect to stability and are simple to use. The digital versions of these algorithms are shown below. [3,11]

**PID Control**

\[ u(k) = -k_p y(k) + \sum_{i=0}^{k} k_i [r(i) - y(i)] - k_d [y(k) - y(k-1)] \]

- \( u = \) controller output
- \( y = \) plant output
- \( r = \) set point

\( k_p, k_i, k_d = \) proportional, integral, derivative parameters respectively

**Lead-Lag Compensation**

\[ u_k = A_1 u_{k-1} + A_2 (w_k - w_{k-1}) + A_3 w_k, \quad w = r - y \]

\[ A_1 = \frac{\tau}{T_S + \tau}, \quad A_2 = \frac{TSK_C}{T_S + \tau}, \quad A_3 = \frac{K}{T_S + \tau} \]

where \( T_S = \) sampling time

and the analog model model is described by

\[ \frac{U(S)}{W(S)} = \frac{K_C (1 + \alpha Ts)}{1 + Ts} \]