National Library
of Canada

Canadian Theses Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Services des thèses canadiennes

CANADIAN THESES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

THÈSES CANADIENNES

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

NL 339 (r. 86/01)

Canada

National Library of Canada
Bibliothèque nationale du Canada

60

Ottawa, Canada
K1A 0N4

TC –

0-315-26793-3

CANADIAN THESES ON MICROFICHE SERVICE – SERVICE DES THÈSES CANADIENNES SUR MICROFICHE

PERMISION TO MICROFILM – AUTORISATION DE MICROFILMER

• Please print or type – Écrire en lettres moulées ou dactylographier

AUTHOR – AUTEUR

Full Name of Author – Nom complet de l'auteur
HEATHER JANE WIEBE REINHOLT

Date of Birth – Date de naissance
19 AUGUST 1959

Canadian Citizen – Citoyen canadien
☑ Yes / Oui ☐ No / Non

Country of Birth – Lieu de naissance
CANADA

Permanent Address – Résidence fixe
825 EXMOUTH STREET
UNIT 2
SARNIA, ONTARIO, CANADA, N7T5R1

THESIS – THÈSE

Title of Thesis – Titre de la thèse
Evaluation of Parameter Estimation Methods for the Self-Tuning Controller

Degree for which thesis was presented
Grade pour lequel cette thèse fut présentée
M. Sc.

Year this degree conferred
Année d'obtention de ce grade
1985

University – Université
University of Alberta

Name of Supervisor – Nom du directeur de thèse
R. K. Wood

AUTHORIZATION – AUTORISATION

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to **microfilm** this thesis and to **lend** or **sell copies of the film.**

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

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de **microfilmer** cette thèse et de **prêter** ou de **vendre des exemplaires du film.**

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

ATTACH FORM TO THESIS – VEUILLEZ JOINDRE CE FORMULAIRE À LA THÈSE

Signature
Reinholt

Date
18 April 85

THE UNIVERSITY OF ALBERTA

EVALUATION OF PARAMETER ESTIMATION METHODS FOR THE SELF-TUNING CONTROLLER

by

(C) HEATHER J.W. REINHOLT

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE
IN
PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA
SPRING 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR HEATHER J.W. REINHOLT

TITLE OF THESIS EVALUATION OF PARAMETER ESTIMATION METHODS FOR THE SELF-TUNING CONTROLLER

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1985

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

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

(SIGNED) H.J.W. Reinholt

PERMANENT ADDRESS:

825 Exmouth Street
Unit 2
Sarnia, Ontario N7T5R1

DATED April 12 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled EVALUATION OF PARAMETER ESTIMATION METHODS FOR THE SELF-TUNING CONTROLLER submitted by HEATHER J.W. REINHOLT in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in PROCESS CONTROL.

....
Supervisor

....

R. E. Rink

Date 19 APRIL 1985

To my husband and my father

Abstract

The purpose of this thesis is to evaluate five recursive parameter estimation techniques used in conjunction with the self-tuning controller. The identification algorithms under study are the recursive least squares estimator, the recursive square root estimator, the recursive upper diagonal factorization estimator, the recursive learning estimator and the recursive maximum likelihood estimator.

The identification methods were first tested for the control of a third order linear system with no time delay. The control performance was quantitatively evaluated using three criteria: the sum of predicted errors, the control effort and the final parameter estimate values. Setpoint tracking tests were performed using a square wave change in setpoint, a sawtooth function change in setpoint and a step change in setpoint. The effects of certain estimation algorithm parameters such as the initial covariance matrix value, the forgetting factor and the initial parameter estimate values on the recursive upper diagonal factorization estimator were also studied during the linear simulations. The results of Q and R weighting of the self-tuning controller and disturbance rejection abilities were also considered in the examination.

A nonlinear model of a binary distillatin column was used to determine if the identification techniques could compensate for nonlinearities.

Experiments using a binary distillation column to separate a methanol and water feed into a 95 mass percent methanol top product and a 5 mass percent methanol bottom product were also performed. Both the simulations using the nonlinear model of the column and the pilot plant binary distillation column experiments evaluated the estimation techniques on the resulting control performance for step changes in the feed rate and square wave changes in setpoint.

The recursive upper diagonal factorization estimator proved to be the most efficient and accurate estimator for all tests.

Acknowledgement

The author would like to thank her thesis supervisor, Dr. R.K. Wood, for the assistance he has provided during the preparation of this thesis. Appreciation is also extended to Bob Barton, Henry Sit, Andrew Smillie and Jim Langman for their valuable assistance.

The support and assistance from the eighth floor graduate students is also greatly appreciated and will never be forgotten. A special thanks to Will Cluett for his never-ending support and encouragement.

Lastly the financial support provided by the Department of Chemical Engineering and the Natural Sciences and Engineering Research Council is gratefully acknowledged.

Table of Contents

Chapter Page

1. Introduction 1

2. Development of the Self-Tuning Controller 3

2.1 Introduction 3

2.2 Explicit and Implicit Self-Tuning Controllers ...4

2.3 Derivation of the Self-Tuning Controller 6

2.3.1 Selection of Model Order 6

2.3.2 Selection of Time Delay 8

2.3.3 Self-Tuning Controller for Systems with Known Parameters 8

2.3.4 Stability of the Closed Loop System 14

2.3.5 Self-Tuning Controller for Systems with Unknown Parameters 15

2.4 Convergence of the Self-Tuning Controller 18

2.5 Q-Weighting 19

2.6 P-Weighting 20

2.7 R-Weighting 21

3. Identification Methods 22

3.1 Introduction 22

3.2 Recursive Least Squares Estimator 23

3.2.1 Least Squares Theory 23

3.2.2 Recursive Least Squares 29

3.2.3 On-Line Recursive Least Squares Estimator 33

3.3 Recursive Square Root Estimator 36

3.4 Recursive Upper-Diagonal Factorization Estimator 44

3.5 Recursive Learning Estimator 50

3.6 Recursive Maximum Likelihood Estimator 55

3.6.1 Maximum Likelihood Estimator 55

3.6.2 Recursive Maximum Likelihood Estimator ...67

4. Linear Simulation Results 72

4.1 Introduction 72

4.2 Setpoint Tracking 74

4.2.1 Recursive Least Squares Estimator 76

4.2.2 Recursive Square Root Estimator 86

4.2.3 Recursive Upper Diagonal Factorization Estimator 91

4.2.4 Recursive Learning Estimator 101

4.2.5 Recursive Maximum Likelihood Estimator ..109

4.3 The Effect of Q and R Weighting 120

4.4 Disturbance Rejection Testing 153

4.5 Parameter Identification Variables 167

4.5.1 Initial Covariance Values 168

4.5.2 Forgetting Factor 183

4.5.3 Initial Parameter Estimates 201

4.6 Convergence Problems of the Estimators 213

4.7 Conclusions 215

5. Nonlinear Simulation Results 217

5.1 Evaluation of the Recursive Least Squares Estimator 220

5.2 Evaluation of the Recursive Square Root Estimator 228

5.3 Evaluation of the Recursive Upper Diagonal Factorization Estimator 234

5.4 Evaluation of the Recursive Learning Estimator 237

5.5 Summary 242

5.6 Setpoint Tracking using RLS and RUD Estimators 245

6. Experimental Results 253

6.1 Recursive Least Squares Estimation Evaluation .256

6.2 Recursive Square Root Estimation Evaluation ...268

6.3 Recursive Upper Diagonal Factorization Estimation Evaluation 280

6.4 Recursive Learning Estimation Evaluation 292

7. Conclusions 296

8. Further Work 300

9. References 301

10. Appendix A 305

11. Appendix B 308

12. Appendix C 347

List of Tables

Table Page

4.1 Summary of Setpoint Tracking Results for Different Identification Methods and Setpoint Changes 75

4.2 Effect of the Choice of the Initial Covariance Matrix using the Recursive Upper Diagonal Factorization Estimator on Performance 184

List of Figures

Figure Page

2.1 Block Diagram for the Explicit Self-Tuning Controller 5

2.2 Block Diagram for the Implicit Self-Tuning Controller 7

4.1 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Least Squares Estimator 77

4.2 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator 78

4.3 Setpoint Tracking a Step Change in Setpoint using the Recursive Least Squares Estimator 79

4.4 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Least Squares Estimator 80

4.5 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator 81

4.6 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Least Squares Estimator 82

4.7 Parameter Estimates for a Step Change in Setpoint using the Recursive Least Squares Estimator 85

4.8 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Square Root Estimator 87

4.9 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator 88

4.10 Setpoint Tracking a Step Change in Setpoint using the Recursive Square Root Estimator 89

Figure Page

4.11 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Square Root Estimator 90

4.12 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator 92

4.13 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Square Root Estimator 93

4.14 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 94

4.15 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 95

4.16 Setpoint Tracking a Step Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 96

4.17 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 98

4.18 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 99

4.19 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator 100

4.20 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Learning Estimator 102

4.21 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Learning Estimator 103

4.22 Setpoint Tracking a Step Change in Setpoint using the Recursive Learning Estimator 104

Figure Page

4.23 Square Wave Setpoint Tracking after 800 Sample Intervals using the Recursive Learning Estimator 105

4.24 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Learning Estimator 106

4.25 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Learning Estimator 107

4.26 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Learning Estimator 108

4.27 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Maximum Likelihood Estimator 112

4.28 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Maximum Likelihood Estimator 113

4.29 Setpoint Tracking a Step Change in Setpoint using the Recursive Maximum Likelihood Estimator 114

4.30 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Maximum Likelihood Estimator 115

4.31 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Maximum Likelihood Estimator 116

4.32 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Maximum Likelihood Estimator 117

4.33 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for the Square Wave Change in Setpoint..122

4.34 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for the Sawtooth Function Change in Setpoint 123

Figure Page

4.35 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for a Step Change in Setpoint 124

4.36 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Square Wave Change in Setpoint 125

4.37 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Sawtooth Function Change in Setpoint 126

4.38 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Step Change in Setpoint 127

4.39 Parameter Estimates for Square Wave Change in Setpoint using the Recursive Least Squares Estimator with Q and R Weighting 128

4.40 Parameter Estimates for Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator with Q and R weighting 129

4.41 Parameter Estimates for Step Change in Setpoint using the Recursive Least Squares Estimator with Q and R weighting 130

4.42 Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Square Root Estimator with Q and R Weighting 132

4.43 Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator with Q and R Weighting 133

4.44 Parameter Estimates for the Step Change in Setpoint using the Recursive Square Root Estimator with Q and R Weighting 134

4.45 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Square Wave Change in Setpoint 136

Figure Page

4.46 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Sawtooth Function Change in Setpoint 137

4.47 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Step Change in Setpoint 138

4.48 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Square Wave Change in Setpoint 140

4.49 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Sawtooth Function Change in Setpoint 141

4.50 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Step Change in Setpoint 142

4.51 Control Response with Q and R Weighting using the Recursive Maximum Likelihood Estimator for the Square Wave Change in Setpoint 144

4.52 Control Response with Q and R Weighting using the Recursive Maximum Likelihood Estimator for the Sawtooth Function Change in Setpoint 145

4.53 Control Response with Q and R Weighting using the Recursive Maximum Likelihood Estimator for the Step Change in Setpoint 147

4.54 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Square Wave Change in Setpoint 148

4.55 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Sawtooth Function Change in Setpoint 150

4.56 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Step Change in Setpoint 151

Figure Page

4.57 Disturbance Rejection using the Recursive Least Squares Estimator 156

4.58 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Least Squares Estimator 157

4.59 Disturbance Rejection using the Recursive Square Root Estimator 159

4.60 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Square Root Estimator 160

4.61 Disturbance Rejection using the Recursive Learning Estimator 162

4.62 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Learning Estimator 163

4.63 Disturbance Rejection using the Recursive Maximum Likelihood Estimator 164

4.64 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Maximum Likelihood Estimator 165

4.65 Control Response for an Initial Covariance Matrix $S(0)=\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator 170

4.66 Parameter Estimates for an Initial Covariance Matrix $S(0)=\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator 171

4.67 Parameter Estimates for an Initial Covariance Matrix $S(0)=10.0\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator 172

4.68 Control Response for an Initial Covariance Matrix $S(0)=10.0\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator 174

Figure Page

4.69 Parameter Estimates for an Initial Covariance Matrix $S(0)=100.0\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 175

4.70 Control Response for an Initial Covariance Matrix $S(0)=100.0\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 177

4.71 Parameter Estimates for an Initial Covariance Matrix $S(0)=10^4\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 178

4.72 Parameter Estimates for an Initial Covariance Matrix $S(0)=10^5\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 179

4.73 Parameter Estimates for an Initial Covariance $S(0)=10^6\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 180

4.74 Control Response for an Initial Covariance Matrix $S(0)=10^4\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator 181

4.75 Control Response for a Forgetting Factor of 0.7 using the Recursive Upper Diagonal Factorization Estimator 185

4.76 Parameter Estimates for a Forgetting Factor of 0.7 using the Recursive Upper Diagonal Factorization Estimator 186

4.77 Control Response for Forgetting Factor of 0.85 using the Recursive Upper Diagonal Factorization Estimator 187

4.78 Parameter Estimates for a Forgetting Factor of 0.85 using the Recursive Upper Diagonal Factorization Estimator 8

4.79 Parameter Estimates for a Forgetting Factor of 0.9 using the Recursive Upper Diagonal Factorization Estimator 190

Figure Page

4.80 Control Response for Forgetting Factor of 0.9 using the Recursive Upper Diagonal Factorization Estimator 191

4.81 Parameter Estimates for a Forgetting Factor of 0.95 using the Recursive Upper Diagonal Factorization Estimator 193

4.82 Control Response for Forgetting Factor of 0.95 using the Recursive Upper Diagonal Factorization Estimator 194

4.83 Parameter Estimates for a Forgetting Factor of 0.975 using the Recursive Upper Diagonal Factorization Estimator 195

4.84 Control Response for Forgetting Factor of 0.975 using the Recursive Upper Diagonal Factorization Estimator 196

4.85 Parameter Estimates for a Forgetting Factor of 1.0 using the Recursive Upper Diagonal Factorization Estimator 198

4.86 Control Response for Forgetting Factor of 1.0 using the Recursive Upper Diagonal Factorization Estimator 199

4.87 Control Response for Initial Parameter Estimates set to 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint 202

4.88 Control Response for Initial Parameter Estimates set to 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint 203

4.89 Parameter Estimates for Initial Parameter Estimates of 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint 205

4.90 Parameter Estimates for Initial Parameter Estimates of 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint 206

Figure Page

4.91 Control Response for Initial Parameter Estimates set to -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint 207

4.92 Control Response for Initial Parameter Estimates set to -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint 208

4.93 Parameter Estimates for Initial Parameter Estimates of -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint 209

4.94 Parameter Estimates for Initial Parameter Estimates of -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint 211

5.1 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation 221

5.2 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation 222

5.3 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation 223

5.4 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation 225

5.5 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation 226

Figure Page

5.6 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation 227

5.7 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation 229

5.8 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation 230

5.9 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation 232

5.10 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation 233

5.11 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation 235

5.12 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation 236

5.13 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation 238

5.14 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation 239

Figure Page

5.15 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation 240

5.16 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation 241

5.17 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation 243

5.18 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation 244

5.19 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Least Squares Estimation 246

5.20 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Least Squares Estimation 247

5.21 Simulated Response of the Binary Distillation Column to a Square Wave Change in Setpoint for Both Loops using Recursive Upper Diagonal Factorization Estimation 249

5.22 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Upper Diagonal Factorization Estimation 250

Figure Page

5.23 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Upper Diagonal Factorization Estimation 251

6.1 Schematic Diagram of the Pilot Plant Distillation Column 255

6.2 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator 257

6.3 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator 258

6.4 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator 259

6.5 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator 260

6.6 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator 262

6.7 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator 265

6.8 Response of the Bottom Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator 266

Figure Page

6.9 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator 267

6.10 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator 269

6.11 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator 271

6.12 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator 272

6.13 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator 273

6.14 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator 275

6.15 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator 276

6.16 Response of the Bottom Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator 278

6.17 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator 279

Figure Page

6.18 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 281

6.19 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 283

6.20 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 284

6.21 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 285

6.22 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 287

6.23 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 288

6.24 Response of the Bottom Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 290

6.25 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator 291

Figure Page

6.26 Control Performance of the Top Loop and the Parameter Estimates for the 25% Step Increase in Feed Rate using the Recursive Learning Estimator 293

6.27 Control Performance of the Bottom Loop and the Parameter Estimates for the 25% Step Increase in Feed Rate using the Recursive Learning Estimator 294

Nomenclature

A	Polynomial of the model equation associated with the output of the system
B	Polynomial of the model equation associated with the input of the system
C	Polynomial of the model equation associated with the noise
D	Square Diagonal Matrix used in recursive upper diagonal factorization
E'	Polynomial introduced when separating noise into past and future terms
e	error
F	Polynomial of the controller equation associated with the output
F'	Polynomial introduced when separating noise into past and future terms
f	Probability density function
FE	Feed flow rate for the distillation column (g/s)
G	Polynomial of the controller equation associated with the input
G'	Intermediate polynomial of the controller equation associated with the input
G*	Polynomial of the controller equation associated with the top and bottom loop interaction on the distillation column
H	Polynomial of the controller equation associated with the noise

$\hat{I}$	Minimization function for recursive least squares estimation
J_1	Costing function for the self-tuning controller
J_2	Costing function for the self-tuning controller
J'	Costing function for the self-tuning controller
Kg	Gain of the estimation method
L	Polynomial of the model equation associated with measured disturbances
l	Loss function used in the recursive maximum likelihood estimator
P	Weighting polynomial associated with the output
P_d	Denominator of the weighting polynomial associated with the output
P_n	Numerator of the weighting polynomial associated with the output
Q	Weighting polynomial associated with the input
Q'	Intermediate weighting polynomial associated with the input
R	Weighting polynomial associated with the setpoint
RE	Reflux flow rate for the distillation column (g/s)
S	Covariance matrix
S'	Square root of the covariance matrix
ST	Steam flow rate for the distillation column (g/s)
T	A discrete transfer function
U	Square unit upper triangular matrix used in the recursive upper diagonal factorization estimator
u	Input

V	Time-varying loss function used in the recursive maximum likelihood estimator
v	Measurable disturbances
W	Setpoint
X	Observation vector
XB	Bottom composition for the distillation column (%MeOH)
XD	Top composition for the distillation column (%MeOH)
Y	Output vector
y^{*}	Predicted output
Z	Corrupted measurement vector
z	Noise corrupted measurements
z^{-1}	Backward shift operator

Greek Letters

α	Forgetting factor
ϵ	Output prediction error
ζ'	Error gradient
Θ	Vector of unknown parameters
Λ	An orthogonal matrix
μ	Mean vector
ν	Measurement error
ξ	Zero-mean uncorrelated random sequence
ρ	Forgetting factor
σ^2	Variance
τ	Transformation parameter used in the recursive square root estimator
Φ	Measured scalar output

Φ^*	Predicted scalar output
χ	Difference between the actual and predicted parameter vectors in the recursive learning estimator
ψ	Square root of the covariance matrix multiplied by the observation vector in recursive square root estimation
Ω	A real upper triangular matrix
ω	Transformation parameter used in the recursive square root estimator

1. Introduction

The three requirements for chemical process control are good long term steady state regulation, good setpoint tracking and disturbance rejection abilities. The focus of this study is to examine the effect of parameter estimation on these requirements.

The ability of the estimation algorithm, used to identify the parameters of a prespecified control law, is evaluated with respect to the control performance using a self-tuning controller, the control effort required, and, for the simulation of the control of a linear system, the estimated parameter values.

The five parameter estimation techniques being studied are the recursive least squares, recursive square root, recursive upper diagonal factorization, recursive learning and recursive maximum likelihood estimators.

The material presented in this thesis has been organized in the following manner. The theory of the single-input single-output self-tuning controller. is presented in Chapter 2 and the theory of the on-line identification methods outlined in Chapter 3. Evaluation of the parameter identification techniques is conducted by simulations and experimental testing. In Chapter 4 the results of use of all five estimation methods for self-tuning control of a linear system are examined. The effects of Q and R weighting, disturbance rejection, initial covariance matrix values, forgetting factors and initial parameter estimate values were also investigated. The results from simulation of the control performance of a distillation column described by a nonlinear model are presented in Chapter 5. These results test the ability of the identification technique to compensate for the nonlinearities present in the model. The results from the experiments performed on the pilot plant binary distillation column are in Chapter 6. Chapter 7 summarizes the conclusions drawn from the results and Chapter 8 suggests topics for further investigation.

2. Development of the Self-Tuning Controller

2.1 Introduction

Astrom and Wittenmark [1] developed a self-tuning regulator in 1973 for the control of systems with constant, but unknown parameters. The algorithm was obtained by combining a recursive least squares estimator with a minimum variance regulator computed from the estimated model. Although the regulator attempts to minimize the system fluctuations when the system is randomly disturbed it does not try to ensure that setpoints are followed optimally nor does it penalize excessive control action. The lack of control costing by the regulator prohibits modification of its asymptotic behavior unless the sample interval is changed and the algorithm is restarted. Furthermore, stable control of a nonminimum phase system requires that the self-tuning regulator to be replaced by an on-line solution of the Riccati equation which is more complex and time consuming. All of the above problems can be overcome in a number of cases by introducing a costing function. This costing function incorporates system input, output, and setpoint variations and deriving a control law which includes a least-squares predictor of a function related to the costing function with the control input chosen to make the prediction zero. The development of this approach was pioneered by Clarke and Gawthrop [2,3] leading to the more general self-tuning controller. This implicit algorithm provides setpoint tracking as well as regulatory control and can be considered as minimizing a combination of control and output variances.

The self-tuning theory has been successfully applied in practice. These applications cover a range of industries. A few of them are: Borisson and Syding [4] used the self-tuning regulator on an ore-cusher and found that the regulator was able to adapt to the characteristics of the ore and crusher. This resulted in a 10% incease in production over that achieved with proportional-integral control.

Dumont and Belanger [5] used the self-tuning regulator for controlling a titanium oxide kiln which reduced the switchover time involved during a grade change from 10-12 hours to 2 hours.

Other applications include cement blending [6], distillation columns [7] and paper making [8].

2.2 Explicit and Implicit Self-Tuning Controllers

The explicit self-tuning controller, as illustrated in Figure 2.1, has three main elements: a standard feedback law, a recursive parameter estimation, and a control design algorithm. The feedback law is in the form of a difference equation which produces the new control action and the control-design algorithm provides a new set of coefficients for the feedback law from the parameter estimates. The algorithm can be simplified by omitting the controller

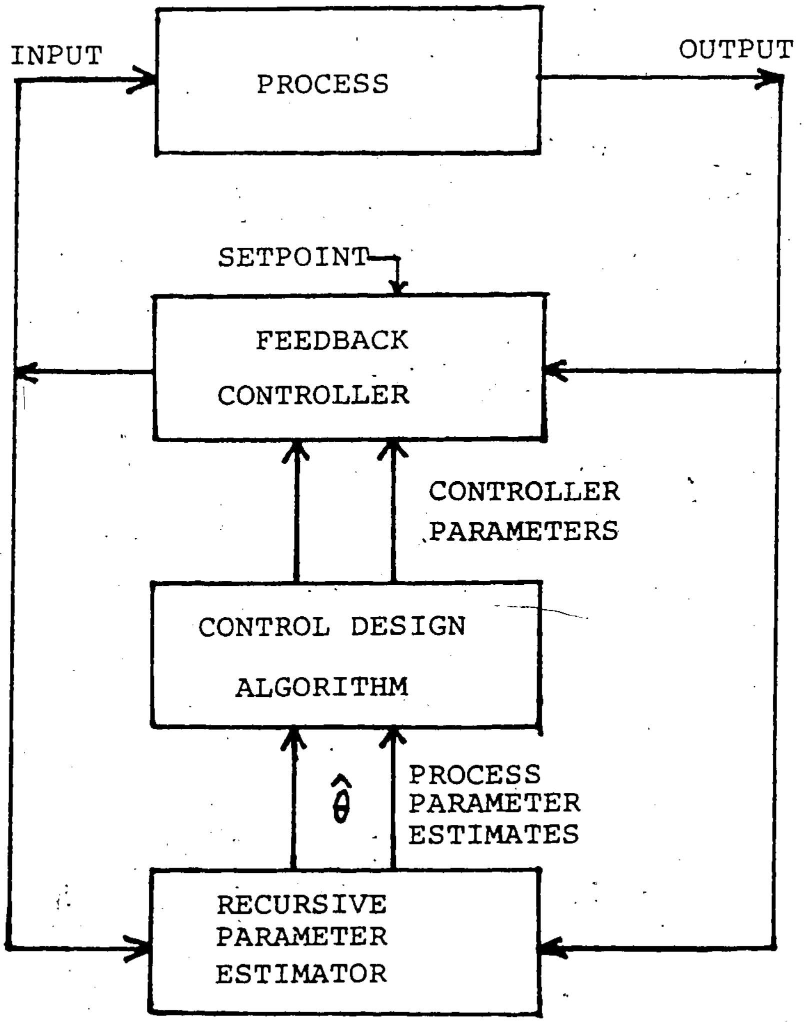


Figure 2.1 Block Diagram for the Explicit Self-Tuning Controller

design stage and have the estimator directly estimate the coefficients of the required control law. This constitutes an implicit or direct self-tuner, as seen in Figure 2.2. The implicit method is based on predictive control theory which depends on the knowledge of a system time delay and the effectiveness of this control depends on the accuracy of the prediction.

In general, an algorithm is self-tuning if, as the number of sample times approaches infinity, the control signal generated becomes that which would be produced by the corresponding feedback law designed on the basis of known process dynamics.

2.3 Derivation of the Self-Tuning Controller

In the derivation of the self-tuning controller, Clarke [9] made the following assumptions about the process and its environment: the process is adequately locally linearizable, the signals are persistently exciting such that the estimator can produce a reasonable model, and the parameters of the self-tuner (model order and time delay) have been correctly chosen.

2.3.1 Selection of Model Order

For the generalized minimum variance self-tuner the model order, n, in practice is often arbitrary and depends on the frequency range of interest and the controller sample time. The number of parameters to be estimated increases

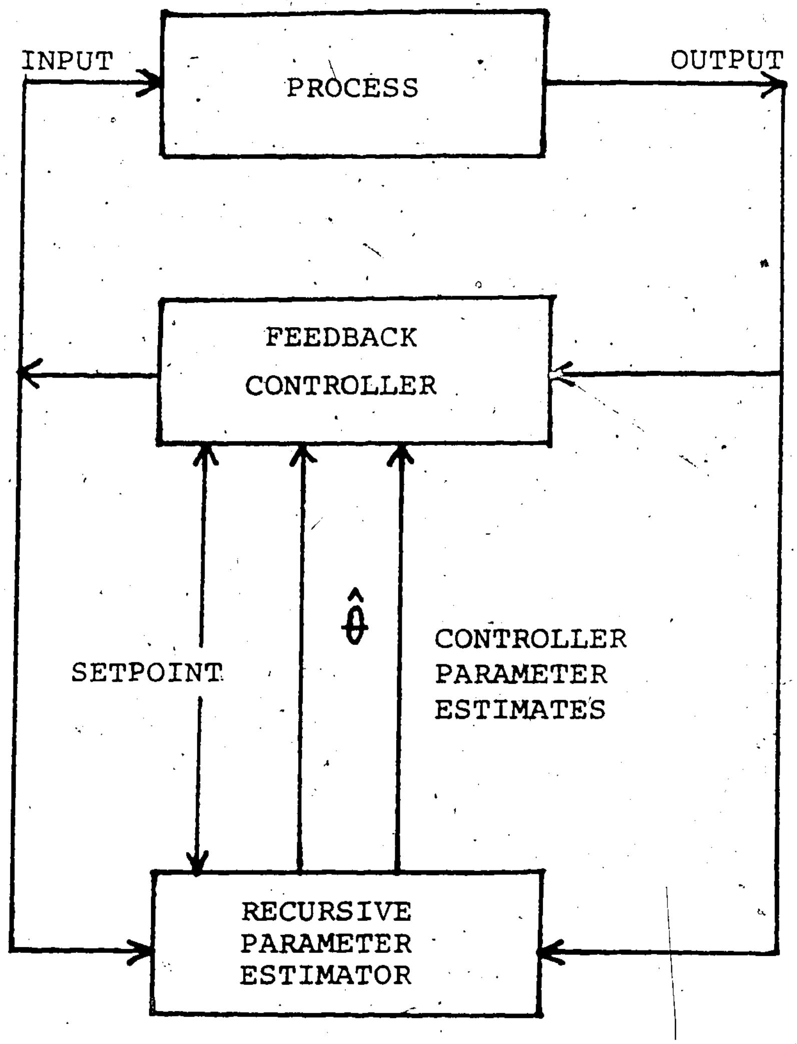


Figure 2.2 Block Diagram for the Implicit Self-Tuning Controller

with n. Therefore if the order is chosen too high it may lead to poor control performance while too low a value of n results in a suboptimal controller.

2.3.2 Selection of Time Delay

The model time delay, d, expressed as a multiple of sampling intervals depends on the sampling interval length and the actual process delay. If the model ~~delay~~ is greater than half the dominant process time constant, the sampling interval length is chosen so that d is equal to 2 or 3 sample intervals since the number of parameters to estimate increases with the value of d. This will avoid the uncontrollable modes mentioned above.

2.3.3 Self-Tuning Controller for Systems with Known Parameters

Clarke and Gawthrop [2] developed the self-tuning controller by first dealing with the design for systems with known parameters and then extending it to those with unknown parameters.

Consider the single input - single output system

$$A(z^{-1})y_k = z^{-d} B(z^{-1})u_k + C(z^{-1})\xi_k + z^{-\ell}L(z^{-1})v_k \qquad (2.1)$$

where A, B, C and L are known polynomials in the backshift operator, z^{-1}, such that $a_0=1$, $b_0\neq0$ and $c_0=1$. C is assumed to have its roots within the z-domain unit circle to maintain prediction stability and ξ_k is a zero-mean uncorrelated random sequence. The addition of v_k allows measurable loop disturbances to be included in a feedforward manner. The system order is n, time delay is d sample intervals and the feedforward delay is l sample intervals.

The design of the the control law is based on the minimization of one of the costing functions

$$J_1 = E\{[Py_{k+d} - Rw_k]^2 + [Q'u_k]^2\} \qquad (2.2)$$

$$J_2 = E\{[Py_{k+d} - Rw_k]^2 + [Q'(u_k - u_{k-1})]^2\} \qquad (2.3)$$

where P, Q and R are polynomial functions in z^{-1} enabling discrete weightings to be placed on both measured and calculated variables if desired, at the expense of an increase in controller order. The weighting polynomials P and R do not affect the controller structure except to increase the order. The cost function, J_1, compromises between a reduction in control variations and increased deviations of the system output. If w_k, the setpoint, has a non-zero mean, the J_1 cost function will not ensure that the output mean is equal to the mean of w_k and this leads to a possibility of the introduction of an offset in the plant output unless Q' is zero. Since the J_2 cost function weights the deviations in controller output, offset present using the J_1 cost function is eliminated by guaranteeing the equality of y_k and w_k. However, the system dynamic performance can be degraded due to the additional integrating effect this has on the control loop.

Th self-tuning controller is a generalization of the theory underlying the minimum variance regulator which predicts the system output d sampling intervals ahead in time, $y^{*}_{k+d/k}$, as a function of control input at time k. The minimum variance regulator is obtained by setting P=1, R=Q'=0 in the costing function.

Consider the controller design for systems with known parameters which minimizes J_1. The self-tuning regulator is then extended to include an offset eliminating control magnitude weighting which is the basis of the following derivation.

Define the optimal least-squares predictor of y_{k+d} using data up to and including time k as

$$y^{*}_{k+d/k} = y_{k+d} - \epsilon_{k+d} \qquad (2.4)$$

where ϵ_{k+d} is the output prediction error which is assumed to be uncorrelated with u_{k-i} and y_{k-i}, $i \geq 0$.

From equation (2.1) the system output d steps ahead in time is given by

$$y_{k+d} = \frac{B}{A}u_k + z^{d}\frac{C}{A}\xi_k + z^{d-\ell}\frac{L}{A}v_k \qquad (2.5)$$

The term $z^{d}(C/A)\xi_k$ can be expanded into the sum of future disturbances, e_{k+d}, and those disturbances up to and including time k, e_k, using the following identity

$$z^{d}\frac{C}{A} = z^{d}E' + \frac{F'}{A} \qquad (2.6)$$

thus

$$z^{d}\frac{C}{A}\xi_k = z^{d}E'\xi_k + \frac{F'}{A}\xi_k = e_{k+d} + e_k \qquad (2.7)$$

where

$$E' = 1 + e'_1 z^{-1} + \ldots + e'_{d-1}z^{-(d-1)} \qquad (2.8)$$

$$F' = f'_0 + f'_1 z^{-1} + \ldots + f'_{ni-1} z^{-(ni-1)} \tag{2.9}$$

such that $ni=\max(n_{P_d} + n_A \;, -d + n_{P_n} + n_C+1)$.

Combining equations (2.5) and (2.7) yields

$$y_{k+d} = e_{k+d} + \left[\frac{B}{A}u_k + \frac{F'}{A}\xi_k + z^{d-\ell}\frac{L}{A}v_k \right] \tag{2.10}$$

then by comparing equations (2.4) and (2.10) it can be shown that

$$y^+_{k+d/k} = \frac{B}{A}u_k + \frac{F'}{A}\xi_k + z^{d-\ell}\frac{L}{A} v_k \tag{2.11}$$

From equation (2.7) it can be seen that

$$z^d E' \xi_k = e_{k+d} \tag{2.12}$$

therefore

$$e_k = E' \xi_k \tag{2.13}$$

and rewriting equation (2.4) for e at time k

$$e_k = y_k - y^+_{k/k-d} \tag{2.14}$$

modifies equation (2.13) into

$$\xi_k = \frac{e_k}{E'} = \frac{(y_k - y^+_{k/k-d})}{E'} \tag{2.15}$$

Substituting equation (2.15) into equation (2.11) gives

$$y^+_{k+d/k} = \frac{B}{A}u_k + \frac{F'}{A}\,\frac{[y_k - y^+_{k/k-d}]}{E'} + z^{d-\ell}\frac{L}{A}v_k \tag{2.16}$$

which can also be written as

$$y^+_{k+d/k} + z^{-d}\frac{F'}{E'A}y^+_{k+d/k} = \frac{B}{A}u_k + \frac{F'}{E'A}y_k + z^{d-\ell}\frac{L}{A}v_k \tag{2.17}$$

By grouping the $y^+_{k+d/k}$ terms on the left hand side of equation (2.17), the identity of equation (2.6) written as

$$\frac{C}{E'A}\, y^{+}_{k+d/k} = y^{+}_{k+d/k} + z^{-d}\frac{F}{E'A}\, y^{+}_{k+d/k} \qquad (2.18)$$

can be used to replace these terms to give

$$\frac{C}{E'A}\, y^{+}_{k+d} = \frac{B}{A} u_k + \frac{F'}{A} y_k + z^{d-\ell}\frac{L}{A} v_k \qquad (2.19)$$

thus

$$y^{+}_{k+d} = \frac{F'}{C} y_k + \frac{E'B}{C} u_k + z^{d-\ell}\frac{E'L}{C} v_k \qquad (2.20)$$

However, if weighted output is required by equation (2.2) applying P weighting to equation (2.20) where $P=P_n/P_d$ yields

$$Py^{+}_{k+d} = E'\xi_{k+d} + \frac{F'}{P_d C}\, y_k + \frac{G'}{C} u_k + z^{d-\ell}\frac{E'L}{C} v_k \qquad (2.21)$$

where $G'=E'B$ and

$$(Py_{k+d/k})^{+} = \frac{F'}{P_d C} y_k + \frac{G'}{C} u_k + z^{d-\ell}\frac{E'L}{C} v_k \qquad (2.22)$$

Now returning to the costing function to be minimized, substitution of equation (2.4) into equation (2.2) gives

$$J_1 = E\{[P(\, y^{+}_{k+d/k} + \epsilon_{k+d}) - Rw_k]^2 + [Q'u_k]^2\} \qquad (2.23)$$

For $P\epsilon_{k+d}$ uncorrelated with u_{k-i}, y_{k-i} and w_{k-i}, $i \geq 0$, letting $\sigma^2 = E\{[P\ \epsilon_{k+d}\ ]^2\}$ allows equation (2.23) to be expressed as

$$J_1 = E\{(Py^{+}_{k+d/k} - Rw_k)^2 + (Q'u_k)^2 + \sigma^2\} \qquad (2.24)$$

which is minimized at each sample instant [9] by choosing u_k such that

$$\frac{\partial J_1}{\partial u_k} = 2[Py^{+}_{k+d/k} - Rw_k]G'(0) + 2q_0'Q'u_k = 0 \qquad (2.25)$$

Using the fact that $G'(0)=b_0$ the control law is given by

$$Py^{+}_{k+d/k} + \frac{q'_0 Q'}{b_0} u_k - Rw_k = 0 \tag{2.26}$$

If a new costing polynomial $Q(z^{-1})$ is defined such that $Q = q'_0 Q'/b_0$, and ϕ^{+} is a predicted scalar output function such that

$$\phi^{+}_{k+d/k} = Py^{+}_{k+d/k} + Qu_k - Rw_k \tag{2.27}$$

The control law is given by setting $\phi^{+}_{k+d/k} = 0$. Since the measured scalar output function is defined by

$$\phi_{k+d} = Py_{k+d} + Qu_k - Rw_k \tag{2.28}$$

recalling that $y_{k+d} = y^{+}_{k+d/k} + \epsilon_{k+d}$ from equation (2.4) where ϵ_{k+d} is uncorrelated with $y^{+}_{k+d/k}$ it follows that

$$\phi_{k+d} = \phi^{+}_{k+d/k} + \epsilon'_{k+d} \tag{2.29}$$

such that $\epsilon'_{k+d} = \Sigma_{i=0}^{d-1} \; p_i \epsilon_{k+d-i}$ is uncorrelated with ϕ^{+}, the least-squares optimal predictor of ϕ.

Defining a costing function J' to be $E\{\phi^{2}_{k+d}\}$ which gives

$$J' = (\phi^{+}_{k+d/k})^{2} + \sigma^{2} \tag{2.30}$$

and minimizing J' or J_1 will generate the same control law.

The control law is obtained by substituting $y^{+}_{k+d/k}$ into equation (2.27) so that

$$\begin{aligned} \phi^{+}_{k+d/k} = \Sigma_{j>o} \; p_i \; [F'_{d-j} y_k + G'_{d-j} u_{k-j} \; + \\ + \, z^{d-\ell} E'_{d-j} L_{d-j} v_k]/C + Qu_k - Rw_k \end{aligned} \tag{2.31}$$

By defining

$$F = \Sigma \; p_j \; F'_{d-j} \tag{2.32}$$

$$G = \Sigma \; p_j \; z^{-j} G'_{d-j} + CQ \tag{2.33}$$

$$H = -CR \tag{2.34}$$

$$D = z^{d-\ell} E'_{d-j} L_{d-j} \tag{2.35}$$

it follows that ϕ^+ can be recursively expressed as

$$C\phi^+_{k+d/k} = \frac{F}{P_d} y_k + Gu_k + Hw_k + Dv_k \tag{2.36}$$

Since the control strategy requires that ϕ^+ be set to zero at each stage of the calculation of the control policy, equation (2.36), the control law, gives the control action by

$$u_k = \frac{[Rw_k - (Py_{k+d/k})^+]}{Q} \tag{2.37}$$

The control algorithm is similar to the basic minimum variance control law in that it minimizes $E\{\phi_k^2\}$ by setting the predicted future value of ϕ equal to zero at each step which means it minimizes the infinite stage variance of the generalized output ϕ_k.

2.3.4 Stability of the Closed Loop System

To determine the stability of the controlled system the closed loop system response must be derived. Since ϕ^+ is set to zero at each stage, under optimal control conditions

$$\phi_k = \phi^+_{k/k-d} + \epsilon_k = \epsilon_k = E\xi_k \tag{2.38}$$

Substituting equation (2.28) into equation (2.38) and solving for y_k gives

$$y_k = \frac{[E\xi_k + z^{-d}Rw_k - z^{-d}Qu_k]}{P} \tag{2.39}$$

then solve the system equation for $z^{-d}u_k$ to obtain

$$z^{-d}u_k = \frac{A}{B} y_k - \frac{C}{B} \xi_k - z^{-\ell} \frac{L}{B} v_k \tag{2.40}$$

Substitute equation (2.40) into equation (2.39) to obtain the closed loop system response or transfer function

$$y_k = \frac{[EB + CQ]\xi_k + z^{-d}BRw_k + z^{-\ell}QLv_k}{[\ PB + QA\ ]} \tag{2.41}$$

Therefore the stability of the optimally controlled system is dependent upon the roots of the characteristic equation

$$PB + QA = 0 \tag{2.42}$$

If P=1 and Q=0 the minimum variance regulator is obtained and the roots depend entirely upon B and hence for nonminimum phase systems the closed loop response is unstable. For $Q \neq 0$ where $Q = q_0' Q'/b_0$, if the system is open loop stable the system will be closed loop stable for large Q. Similarly, if the system is open loop unstable, the system will be stablized if the control weighting Q is small enough.

Thus the control law for systems with known parameters is determined by setting the optimal prediction, $\phi^*_{k+d/k}$, of a function ϕ_{k+d}, closely related to the chosen costing function, to zero at each stage.

2.3.5 Self-Tuning Controller for Systems with Unknown Parameters

For the system with unknown parameters the system equation from the known parameter case and the costing function , equation (2.2), are combined to form an equivalent system given by

$$C\phi^{+}_{k+d/k} = Fy_k + Gu_k + Hw_k + Dv_k \qquad (2.43)$$

$$\phi_{k+d} = \phi^{+}_{k+d/k} + \epsilon_{k+d} \qquad (2.44)$$

with the control strategy setting ϕ^{+} to zero at each stage of the control policy calculation.

If C=1, equation (2.44) can be written in the form

$$\phi_{k+d} = X_k^t\Theta + \epsilon_{k+d} \qquad (2.45)$$

where X_k^t is a column vector containing known functions of time

$$X_k^t = [y_k \; y_{k-1} \ldots \; u_k \; u_{k-1} \ldots \; w_k \; w_{k-1} \ldots \; 1] \qquad (2.46)$$

and Θ is the vector of unknown parameters.

Since the components of X are uncorrelated with ϵ_{k+d} then the best (minimum variance and unbiased) linear estimate $\hat{\Theta}$ of Θ is that given using a least-squares algorithm in its recursive form.

The control law assumes the estimated parameters are the true ones and chooses u_k such that

$$\hat{\phi}^{+}_{k+d/k} = \hat{F}y_k + \hat{G}\,u_k + \hat{H}w_k + \hat{D}v_k = X_k^t\hat{\Theta}_k = 0 \qquad (2.47)$$

Now if C is a polynomial of order n then equation (2.45) becomes

$$\phi_{k+d} = X_k^t\Theta + \epsilon_{k+d} + (1-C)\phi^{+}_{k+d/k} \qquad (2.48)$$

In this situation ϕ^{+} is correlated with X since ϕ^{+} is a function of X so a least squares estimator alone would produce estimates that are biased. But the estimator is used with a control law which sets ϕ^{+} to zero, therefore if $\hat{\Theta}\to\Theta$ then the control law of equation (2.47) tends to the optimal control law, $\phi^{+}=0$, and the objectionable terms in equation (2.48) disappear. Thus $\hat{\Theta}=\Theta$ is a fixed point of the algorithm but not necessarily a stable fixed point [10].

The above argument, however, may not apply in practice as it depends on equation (2.47) being satisfied by the control action u_k. If there are strict control limits the desired u_k may not be used and equation (2.47) no longer holds. Although the recursive least squares estimator should fail to give unbiased parameter estimates, it is found that provided u_k is not always clipped the algorithm is still effective [10]. A method which has been shown to converge for all useful C is the recursive maximum likelihood estimator.

Ljung [11] has shown for a self-tuning regulator that if the system input and output are bounded, which is required for stable closed loop control, the stability of the fixed point of the algorithm is related to the stability of a set of associated ordinary differential equations.

The conditions for obtaining systems which produce unstable fixed points and where parameter estimates oscillate about the point are closely related to the parameter sensitivity criteria for a minimum variance control law. This indicated that a small error in the parameter vector is propagated into poor control in these stable cases resulting in further bias in future parameter estimates. This excess sensitivity may be reduced by an appropriate choice of Q so convergence is possible in a wider range of circumstances.

2.4 Convergence of the Self-Tuning Controller

The parameter estimates in equation (2.47) are not unique as the control performance would not be affected by multiplying the equation by an arbitrary constant. This could lead to numerical problems if the estimates become excessively large or small. This can be overcome by fixing one parameter. If w_k is non-zero the leading coefficient, h_0, is independent of the system parameters and hence is known a priori. Thus it is the most convenient to set to a fixed value as it does not exhibit convergence problems.

If $w_k=0$, then $H(z^{-1})$ is not included in the control law and g_0 can never be zero provided that d, the time delay, is properly chosen. However convergence problems may arise. If g_0 is much smaller than its actual value, the estimates may diverge and if g_0 is taken too large convergence tends to be slow. The rate of convergence depends on the ratio of $g_0/\hat{g}_0$.

If $C\neq1$ the convergence of the algorithm is hindered by the inclusion of early data of ϕ_k which contains components due to non-zero $\phi^*_{k+d/k}$ since the control law, equation (2.47) , is away from the optimum [2]. The standard recursive least squares estimator, however, is derived ignoring these non-zero ϕ^* terms so the norms of the covariance and gain vectors may approach zero more rapidly than ϕ^* and the estimates, $\hat{\theta}_k$, may only be changing slowly while still far from the true values. This is overcome by the addition of the forgetting factor introduced in the recursive least squares estimator.

2.5 Q-Weighting

If the control signal oscillates between the upper and lower limits it may produce stability problems. As well, operation in an industrial situation would not allow such changes to occur as it would be abusive and thus reduce the life of the equipment. Introduction of a Q-weighting polynomial can improve control performance and closed loop stability by penalizing the control action.

Closed loop dynamics can be modified by the choice of the Q polynomial. The closed loop poles can be manipulated by choosing a scalar Q. However, this usually results in a steady state offset as the measured output function given by equation (2.28) becomes

$$y_k = w_{k-d} + \lambda u_k \qquad (P=R=1) \qquad (2.49)$$

This offset can be eliminated by choosing an appropriate Q polynomial. A pure integrator can be introduced by setting $Q=\lambda(1-z^{-1})$ but this may render the system unstable and cause the deterioration of the transient response. A better choice would be to set the inverse of the Q polynomial to a form of a discrete PID compensator

$$Q^{-1}(z^{-1}) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \qquad (2.50)$$

The robustness of PID control assures this method of good self tuning properties and avoids the steady state offset [12]. However the coefficients must be tuned before starting the control action.

2.6 P-Weighting

In the self-tuning regulator theory the control law tries to make the process equal to the d-step-ahead reference value in a single step, that is, deadbeat control. This may cause large excursions in the process variables if a change in setpoint occurs. This is particularly true of the initial period after the change.

This can be improved by designing a reference model which generates the optimal trajectory for the setpoint change [12]. If Q weighting is not considered the output of the system follows the output of the reference model $R(z^{-1})/P(z^{-1})$ whose input is w_{k-d}, the delayed setpoint given by

$$y_k = \frac{1}{P(z^{-1})}[Rw_{k-d} + \epsilon_k] \tag{2.51}$$

From equation (2.51) it can be seen that the unmeasureable system noise ϵ_k is also filtered by the inverse of the P polynomial. If this noise is differentiated it may lead to an erratic response and degradation of the parameter estimation.

The order of the P polynomial should be less than or equal to the order of the plant with unity gain and an open loop response that is faster than that of the process[13].

2.7 R-Weighting

The use of P-weighting is dependent on the closed loop stability requirements. The R polymonial is another way of modifying setpoint changes to improve transient response but it does not alter the closed loop stability of the system. It can also be used in conjunction with the P polynomial.

The parameter degradation observed with the P-weighting modifying both the setpoint and stochastic noise terms can be avoided by setting P=1. Thus the R-weighting dictates the way in which a setpoint is made.

3. Identification Methods

3.1 Introduction

For control engineering the purpose of identification is to design a control strategy and analyze the properties of a system. As the performance of the self-tuning controller depends on the effectiveness of the parameter estimator it is necessary to obtain an on-line estimator which quickly and accurately determines the parameter estimates of the system model. The identification of the process parameters is done by using measured data from the system to estimate the best values of the unknown parameters of the chosen model. From the adaptive control point of view it is also necessary to identify the process recursively in order to track time-varying parameters.

The estimator algorithm chosen for identification must be computationally efficient and easy to implement on a computing device for it to be useful for the self-tuning controller. The estimator algorithm must also be able to retain numerical stability and significance over very long periods of constant setpoint operation prior to disturbances affecting the system.

Identification methods have been extensively surveyed by Astrom and Eykhoff [14]. Books dealing with identification methods have also been published by Graupe [15], Eykhoff [16] and Hsia [17].

Regression analysis has been used to develop identification methods which are applicable to both linear and nonlinear processes.

There are many ways of obtaining recursive estimator algorithms. In the following sections, five recursive identification algorithms will be derived for use in the self-tuning controller. All the estimators are essentially variations or extensions of the least squares method which is presented first.

3.2 Recursive Least Squares Estimator

3.2.1 Least Squares Theory

The least squares (LS) method was developed in 1795 when Karl Gauss formulated its basic concept and used it for astronomical computations. The method is based on the idea that the most appropriate values for the unknown but desired parameters are the "most probable" values. These "most probable" values were defined as the values for which the sum of the squares of the differences between actual observed and computed values, multiplied by numbers that measure the degree of precision, is a minimum.

Parameter estimation of controlled systems to determine a mathematical model is a vital part of adaptive control schemes. Given measured data from a plant the "best" guesses for the parameters provide the means for initial control system analysis and design as well as the ability to predict the future behavior of the system. Least squares theory is a major tool in parameter estimation from experimental data as the estimates obtained can have optimal properties, that is the estimates are consistent, unbiased and efficient.

Given the system where y is related linearly to a set of n variables X_i,

$$y = x_1\theta_1 + x_2\theta_2 + \ldots + x_n\theta_n \tag{3.1}$$

where $\Theta = \{\theta_1\theta_2 \ldots \theta_n\}^t$ is a set of constant, unknown parameters. If a sequence of observations of the outputs and inputs is made at regular intervals of time, these measurements can be represented in matrix form at the sampling interval, k

$$Y_k = \underline{X}_k^t\hat{\Theta} \tag{3.2}$$

where

$$Y_k^t = [\, y_1\ y_2\ \ldots\ y_k\,] \tag{3.3}$$

$$\underline{X}_k^t = \begin{bmatrix} x_1(1) & x_2(1) & \ldots & x_n(1) \\ x_1(2) & x_2(2) & \ldots & x_n(2) \\ \vdots & \vdots & & \vdots \\ x_1(k-1) & x_2(k-1) & \ldots & x_n(k-1) \\ x_1(k) & x_2(k) & \ldots & x_n(k) \end{bmatrix} \tag{3.4}$$

To estimate n coefficients of the vector Θ the number of equations, k, is greater than the number of coefficients being estimated, $k \geq n$. If $k = n$ then equation (3.2) can be solved uniquely to obtain the estimate of Θ, $\hat{\Theta}$

$$\hat{\Theta} = \underline{X}_k^{-1}\, Y_k \tag{3.5}$$

provided that $\underline{X}_k^{-1}$, the inverse of the square matrix X_k, exists.

If $k > n$ and data is corrupted by process or measurement noise or modelling error, or both it is generally impossible to determine Θ to exactly satisfy all k equations so Θ is determined on the basis of a least squared error.

Define an error vector

$$E(k,\Theta) = [e(1)\ e(2)\ .\ .\ .\ e(k)]^t \tag{3.6}$$

so that

$$Y_k = \underline{X}_k^t\Theta + E(k,\Theta) \tag{3.7}$$

The parameter estimates are specified to minimize the sum of the square of the error

$$\hat{I}(\Theta) = \Sigma_{i=1}^{k}\ e^2(i,\Theta) \tag{3.8}$$

$$\hat{I}(\Theta) = E^t(k,\Theta)\ E(k,\ \Theta) \tag{3.9}$$

$$\hat{I}(\Theta) = (Y_k - \underline{X}_k\Theta)^t(Y_k - \underline{X}_k\Theta) \tag{3.10}$$

$$\hat{I}(\Theta) = Y_k{}^tY_k\ \underline{X}^t\Theta^tY_k - Y_k{}^t\ \underline{X}_k\Theta + \Theta^t\underline{X}^t{}_kX_k\Theta \tag{3.11}$$

By differentiating $\hat{I}$ with respect to Θ, setting it to zero and solving for $\hat{\Theta}$, that is

$$\left.\frac{d\hat{I}}{d\Theta}\right|_{\Theta=\hat{\Theta}} = \left[\frac{d\hat{I}}{d\theta_1}\frac{d\hat{I}}{d\theta_2}\ .\ .\ .\ \frac{d\hat{I}}{d\theta_n}\right] \tag{3.12}$$

$$\left.\frac{d\hat{I}}{d\Theta}\right|_{\Theta=\hat{\Theta}} = -2Y_k^t\underline{X}_k + 2\Theta^t\underline{X}_k^t\ X \tag{3.13}$$

it follows that

$$\hat{\Theta}_{LS} = [\underline{X}_k^t\underline{X}_k]^{-1}\underline{X}_k^tY_k \tag{3.14}$$

is the least squares estimator of Θ. In the nonrecursive case, for $\hat{\Theta}_{LS}$ to exist $[\underline{X}_k^t\underline{X}_k]^{-1}$ must be a nonsingular k x k matrix and the rows of $\underline{X}_k$ must have rank k which implies the input is persistently exciting.

Since the statistical properties of the LS estimator are based on the properties of the noise corrupting the system, the system equation is now given by

$$Y_k = \underline{X}_k^t \Theta + \epsilon_k \tag{3.15}$$

It is assumed that ϵ_k is a white noise with zero mean value so

$$E\{\epsilon_k\} = 0 \tag{3.16}$$

where E{} is the statistical expectation. It is also assumed that ϵ_k is uncorrelated with Y and $\underline{X}$. By substituting equation (3.7) into equation (3.14)

$$\hat{\Theta} = \Theta + [\underline{X}_k^t \underline{X}_k]^{-1} \underline{X}_k^t \epsilon_k \tag{3.17}$$

Using the assumptions that ϵ_k is a white noise and is uncorrelated with Y and $\underline{X}$ and taking the expectation of both sides of equation (3.17), that is

$$E\{\hat{\Theta}\} = E\{\Theta\} + E\{ [\underline{X}_k^t \underline{X}_k]^{-1} \underline{X}_k^t\} \; E\{\epsilon_k\} = \Theta \tag{3.18}$$

so it follows that since $E\{\Theta\} = \Theta$, the estimator is unbiased.

The covariance matrix corresponding to the estimated error is defined as

$$\underline{S}_{\Theta} = E\{(\hat{\Theta} - \Theta)\ (\hat{\Theta}^t - \Theta^t)\} \tag{3.19}$$

Substituting equation (3.17) into equation (3.19) gives

$$\underline{S}_{\Theta} = E\{(\Theta + [\underline{X}^t \underline{X}]^{-1} \underline{X}^t \epsilon - \Theta)(\Theta^t + [\underline{X}^t \underline{X}]^{-1} \underline{X} \epsilon^t - \Theta^t)\} \tag{3.20}$$

$$\underline{S}_{\Theta} = E\{([\underline{X}^t \underline{X}]^{-1} \underline{X}^t \epsilon)([\underline{X}^t \underline{X}]^{-1} \underline{X} \epsilon^t)\} \tag{3.21}$$

$$\underline{S}_{\Theta} = [\underline{X}^t \underline{X}]^{-1} X^t \; E\{\epsilon \epsilon^t\} \; \underline{X}[\underline{X}^t \underline{X}]^{-1} \tag{3.22}$$

so if the covariance matrix of the error vector ϵ is defined to be

$$\underline{S} = E\{\epsilon\epsilon'\} \tag{3.23}$$

then

$$\underline{S}_{\theta} = [\underline{X}'\underline{X}]^{-1}\underline{X}'\underline{S}\underline{X}[\underline{X}'\underline{X}]^{-1} \tag{3.24}$$

Now if the noise is identically distributed and independent with zero mean and variance σ^2 then

$$\underline{S} = E\{\epsilon\epsilon'\} = \sigma^2\underline{\underline{I}} \tag{3.25}$$

and the covariance of the estimated error is

$$\underline{S}_{\hat{\theta}} = \sigma^2[\underline{X}'\ \underline{X}]^{-1} \tag{3.26}$$

which implies that the corresponding least squares estimation of $\hat{\theta}$ is a minimum variance estimator. The diagonal elements of $\underline{S}_{\hat{\theta}}$ are equal to the estimates of the variances of the parameter estimates.

All statistical properties of the LS estimator depend on the assumption that ϵ is uncorrelated but this generally is not valid which can be easily demonstrated by considering a noisy system.

Following the presentation given by Hsia [17], consider the noisy model to be

$$A(z^{-1})y_k = B(z^{-1})\ u_k + \epsilon_k \tag{3.27}$$

where y and u are the measured outputs and inputs to the process and ϵ is the residual or equation error. The noise-disturbed measurements are then

$$z_k = y_k + e_k \tag{3.28}$$

where e_k accounts for random disturbances to the system and measurement noise. Assuming that e_k is a stationary random process with zero mean and autocorrelation function $S_{ee}(\tau)$, uncorrelated with u or y, it follows that the residual term,

ϵ_k, for the autocorrelated random process is

$$\epsilon_k = A(z^{-1})e_k \tag{3.29}$$

It should be noted that ϵ_k is now designated as the correlated residual although previously it was the uncorrelated residual.

On the basis of the representation given by equation (3.28) $\underline{X}$ can be partitioned into two parts

$$\underline{X}_{z,u} = \underline{X}_{y,u} + \underline{X}_{e,o} \tag{3.30}$$

where

$$\underline{X}_{e,o} = \left[\begin{array}{ccc|c} -e(n) & \dots & -e(1) & \\ \cdot & & \cdot & \\ \cdot & & \cdot & 0 \\ -e(n+k-1) & \dots & -e(k) & \end{array}\right] \tag{3.31}$$

Now Θ is biased if $E\{(\underline{X}'\underline{X})^{-1}\underline{X}'\epsilon\} \neq 0$ or $E\{\underline{X}'\epsilon\} \neq 0$ and since $E\{\underline{X}'_{y,u}\epsilon\} = 0$ it must be proven that $E\{\underline{X}'_{e,o}\} \neq 0$. The elements in $E\{\underline{X}'_{e,o}\,\epsilon\}$ are

$$E\{e(i)\epsilon(j)\} = S_{e\epsilon}(\tau) \qquad ;\ \tau = 1,2,\dots,n \tag{3.32}$$

and from equation (3.29)

$$e(k) = \epsilon(k) + \Sigma_{i=1}^{n}\, a_i\,\epsilon(k-i) \tag{3.33}$$

therefore

$$S_{e\epsilon} = S_{ee}(\tau) + \Sigma_{i=1}^{n}\, a_i\, S_{ee}(\tau-i) \tag{3.34}$$

But it was assumed earlier that $e(k)$ is autocorrelated and $S_{ee}(\tau) \neq 0$ for all τ hence not all $S_{e\epsilon} = 0$ and $E\{\underline{X}'\epsilon\} \neq 0$ and $\hat{\Theta}$ is biased. The only exception to this result is for the case that S_{ee} satisfies the condition

$$S_{ee}(\tau) + \Sigma_{i=1}^{n}\, a_i\, S_{ee}(\tau-i) = 0 \qquad ;\tau=1,2,\dots,n \tag{3.35}$$

The existance of bias has now been demonstrated so obviously a modification of the basic least squares method is required to obtain better parameter estimates. In 1967 Clarke [18] developed this modification, the method of generalized least squares (GLS), to remove this bias by introducing a whitening filter to convert the correlated residual into a white residual.

If the process is governed by equation (3.27) and the correlations of the residuals are known, then $\epsilon(k)$ can be represented by

$$\epsilon(k) = T(z^{-1})\, e(k) \tag{3.36}$$

where $\{e(k)\}$ is a sequence of uncorrelated random variables and $T(z^{-1})$ is a discrete transfer function. Then the process model can be written as

$$A(z^{-1})\tilde{y}(k) = B(z^{-1})\, \tilde{u}(k) + e(k) \tag{3.37}$$

where

$$\tilde{y}(k) = T^{-1}(z^{-1})y(k) \tag{3.38}$$

$$\tilde{u}(k) = T^{-1}(z^{-1})u(k) \tag{3.39}$$

Hence if u and y are considered inputs and outputs, respectively, and since e(k) is an uncorrelated random sequence the problem results in an unbiased least squares estimation.

3.2.2 Recursive Least Squares

The LS estimator is a batch calculation which implies that all the data between times n and k must be obtained before the estimates are calculated. In real-time systems, however, new experimental data is continuously being supplied. Thus the need for a recursive solution arises to avoid the repetitious calculations of the estimates, with successively supplemented data resulting from these new measurements. The advantages of the recursive least squares estimator as cited by Hasting-James and Sage [19] are

i. the reduction in storage of the large quantities of data and the calculations performed.

ii. the estimates can be updated without a matrix inversion which further reduces computation time.

iii. the effect of each new observation is immediately reflected on the estimated parameter.

iv. the recursive algorithm is well suited for adaptive control of a nonstationary system.

The following derivation of the recursive least squares follows the development given by Strejc [16]. Given the updated output vector y, and the corresponding observation matrix $\underline{\underline{X}}$, where

$$Y_{k+1} = \begin{bmatrix} Y_k \\ y_{k+1} \end{bmatrix}, \quad \underline{\underline{X}}_{k+1} = \begin{bmatrix} \underline{\underline{X}}_k \\ X_{k+1} \end{bmatrix} \tag{3.40}$$

where

$$Y^t_k = [y_1 \; y_2 \; . \; . \; . \; . \; y_k \; y_{k+1}] \tag{3.41}$$

and the last row of $\underline{\underline{X}}_{k+1}$ is

$$\underline{\underline{X}}^t_{k+1} = [u(k+1-n) \; y(k+1-n) \ldots u(k) \; y(k) \; u(k+1)] \tag{3.42}$$

The least squares estimate of the parameters at time k+1 from equation (3.14) is

$$\hat{\theta}_{k+1} = (\underline{\underline{X}}^t_{k+1}\underline{\underline{X}}_{k+1})^{-1}\underline{\underline{X}}^t_{k+1}y_{k+1} \tag{3.43}$$

$$\hat{\theta}_{k+1} = (\underline{X}_k'\underline{X}_k + \underline{X}_{k+1}\underline{X}_{k+1}')^{-1}(\underline{X}_k' y_k + \underline{X}_{k+1}' y_{k+1}) \qquad (3.44)$$

To avoid the matrix inversion in updating the following matrix inversion lemma is utilized.

Lemma: Let A, C and A-BCD be nonsingular square matrices, then

$$(A - BCD)^{-1} = A^{-1} - A^{-1}B\,(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad (3.45)$$

Proof: Multiply both sides of equation (3.45) by (A - BCD)

$$(A-BCD)(A-BCD)^{-1} \Rightarrow (A-BCD)(A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1})$$

$$\underline{\underline{I}} = \underline{\underline{I}} - BCDA^{-1} + BCDA^{-1}$$

$$\underline{\underline{I}} = \underline{\underline{I}} \qquad (3.46)$$

Therefore

$$(\underline{X}_k'\underline{X}_k + \underline{X}_{k+1}\underline{X}_{k+1}')^{-1} = (\underline{X}_k'\underline{X}_k)^{-1} - (\underline{X}_k'\underline{X}_k)^{-1}X_{k+1} *$$

$$[1 + \underline{X}_{k+1}'(\underline{X}_k'\underline{X}_k)^{-1}\underline{X}_{k+1}]^{-1}\underline{X}_{k+1}'(\underline{X}_k'\underline{X}_k)^{-1} \qquad (3.47)$$

Defining

$$Kg_{k+1} = (\underline{X}_k'\underline{X}_k)^{-1}\underline{X}_{k+1}[1 + \underline{X}_{k+1}'(\underline{X}_k'\underline{X}_k)^{-1}\underline{X}_{k+1}]^{-1} \qquad (3.48)$$

it follows from equation (3.44) that

$$\hat{\theta}_{k+1} = \hat{\theta}_k + Kg_{k+1}\,(Y_{k+1} - \underline{X}_{k+1}'\hat{\theta}_k) \qquad (3.49)$$

where $(y_{k+1} - \underline{X}_{k+1}'\hat{\theta}_k)$ is the equation error at time (k+1)

From equation (3.49) it is obvious that the updated estimate is equal to the previous estimate $\hat{\theta}_k$, corrected by a term proportional to the equation error, the difference between the actual and predicted output based on the estimate of the parameters $\hat{\theta}_k$, and the set of measurements $\underline{X}_{k+1}'$. The predicted output is equal to the actual output only if the exact system model has the correct parameters and no noise is present, so the correction term in equation (3.49) is zero.

The elements of the Kg_{k+1} matrix, usually called the gain of the estimation, are weighting coefficients. To calculate the gain matrix in a recursive manner let

$$S_k = \alpha(\underline{X}'_k\ \underline{X}_k)^{-1} \qquad (3.50)$$

where α is a positive constant called a forgetting factor and S_k is called the covariance matrix of the estimates. Then

$$Kg_{k+1} = S_k X_{k+1}[\alpha + X_{k+1}^{t} S_k X_{k+1}]^{-1} \qquad (3.51)$$

Substituting equation (3.50) and equation (3.51) into equation (3.47) yields

$$S_{k+1} = S_k - S_k X_{k+1}[\alpha + X_{k+1}^{t} S_k X_{k+1}]^{-1} X_{k+1}^{t} S_k$$

$$S_{k+1} = S_k - Kg_{k+1} X_{k+1}^{t} S_k$$

$$S_{k+1} = S_k[\underline{\underline{I}} - Kg_{k+1} X_{k+1}^{t}] \qquad (3.52)$$

so the recursive least squares algorithm can be summarized by the following three equations

$$Kg_{k+1} = S_k X_{k+1}[\alpha + X_{k+1}^{t} S_k X_{k+1}]^{-1} \qquad (3.51)$$

$$S_{k+1} = S_k[\underline{\underline{I}} - Kg_{k+1} X_{k+1}^{t}] \qquad (3.52)$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + Kg_{k+1}[y_{k+1} - X_{k+1}^{t}\hat{\theta}_k] \qquad (3.49)$$

The forgetting factor α may be selected in the interval $0 < \alpha \leq 1$. For $\alpha = 1$ all sampled data pairs are weighted equally (ordinary least squares) but for $\alpha < 1$ a heavier weighting is placed on the recently acquired data with the weighting decreasing as α approaches zero. Thus it is necessary to compromise between fast adaptive capability and loss of accuracy due to data sequence truncation.

3.2.3 On-Line Recursive Least Squares Estimator

The development of a recursive least squares (RLS) algorithm enables parameter identification to be done on-line. In real-time systems, however, the estimated parameters may converge to the appropriate values very slowly. The magnitude of the correction of the estimates as shown in equation (3.49) depends on the value of the entries in the gain matrix. The norm of the covariance S_{k+1} and hence Kg_{k+1} will tend towards very small values faster than the parameters converge to their true values. To prevent this situation, which may be a problem, a weighting factor is introduced in order to place more emphasis on the recent information so this data dominates the error function. This is done by writing the minimization function as an exponentially weighted error function

$$I = \Sigma_{i=1}^{k} \rho^{k-i} e_k^{2} \qquad 0 < \rho \leq 1 \tag{3.53}$$

where ρ is an exponential forgetting factor which enables past data to be weighted out to improve the convergence rate. However, this faster convergence due to data sequence truncation leads to an increase in the variance of the parameter estimates.

The forgetting factor is also beneficial to time-varying systems as it enhances the adaptive capabilities of the estimation method by enabling it to track the slowly time-varying parameters. Therefore the covariance matrix update equation is now .

$$S_{k+1} = S_k\frac{1}{\rho}[\underline{\underline{I}} - Kg_{k+1}X_{k+1}^{t}] \tag{3.54}$$

where $S_k = \alpha(X_k^tX_k)^{-1}$. S_k is inflated by the factor ρ^{-1} at every iteration to prevent it from becoming decreasingly small.

The distinction between the two forgetting factors is made as the function of each is different. [illegible]orgetting factor α which weights the sampled data is a[illegible]qual to ρ in the recursive least squares approach but [illegible]an be chosen to be equal to unity or σ^2 as in the Kalman filter approach. The forgetting factor ρ affects the covariance matrix and is equal to unity during constant setpoint operation but is set to less than unity when any setpoint changes occur for parameter tracking purposes.

Thus the on-line recursive least squares estimator is represented by equations (3.49), (3.51) and (3.54).

In spite of the applicability of this method to parameter estimation, the covariance update of the recursive least squares estimator, Equation (3.54), is numerically unstable. The main reason for this instability is that the covariance matrix S_{k+1} is computed as a difference of two positive semi-definite matrices. Due to numerical round-off errors there is the possibility of even obtaining a negative definite covariance matrix. Another cause of this instability is that as the estimation time increases, when the forgetting factor is less than unity, problems are encountered. If the system is not sufficiently excited or disturbed the weighted RLS estimator will gradually lose dynamic information which leads to a progressive loss of confidence in the estimates as the values become biased. This is due to the covariance matrix continuing to grow by the continual division by the forgetting factor [38]. The parameter covariance matrix and the corresponding gain values become numerically large causing large erratic parameter and process behavior when the system is eventually disturbed and severe numerical problems eventually arise. The worst case results when the covariance matrix loses its positive definiteness when the diagonal elements become negative, due to the rounding error caused by the finite word length of the computing device, and consequently its numerical significance.

If the estimate update equation is rewritten as

$$\hat{\theta}_{k+1} = \hat{\theta}_k \, [\underline{\underline{I}} - X_{k+1} \, Kg_{k+1}] + Kg_{k+1} y_{k+1} \qquad (3.55)$$

this difference equation is unstable if $|\det(\underline{I} - X_{k+1} Kg_{k+1})| > 1$ as the parameters will continue to grow. This may occur when some components of the observation vector X_k are linearly dependent which is the case when the self-tuning controller converges to a constant control law.

There are two alternatives for preventing the numerical instability of the covariance matrix. An adaptive forgetting factor can be used so that as less dynamic information is obtained and satisfactory estimates are achieved the forgetting factor is set to unity to preserve the numerical significance of the estimator. Alternatively the covariance matrix can be forced to maintain positive definiteness by factorization. The square root or upper triangular factorization methods which are discussed in Sections 3.3 and 3.4 , respectively, provide algorithms which maintain covariance positive definiteness.

3.3 Recursive Square Root Estimator

The recursive square root estimator was first developed by Potter [20] to avoid the loss of positive definiteness of the covariance matrix which can occur using recursive least squares. The square root algorithm requires the factorization of the covariance matrix. Since this factorization is not unique, several square root estimators were developed; a survey of these techniques is given by Kaminski et al [21]. In 1974, Peterka [22] developed a recursive square root estimator for multivariate regression which is closely related to the square root filters summarized by Kaminski and co-workers. but is in a more compact form. It is Peterka's estimator which will be developed for this study.

The fundamental difference between the recursive square root estimator and recursive least squares is the method of propagating the covariance matrix. The recursive equation for the inverse of $\underline{S}$ is obtained by generalizing equation (3.50) .

$$\underline{S}_k = \alpha\ (\underline{X}_k^t \underline{X}_k)^{-1} \tag{3.50}$$

to yield

$$\underline{S}_{k+1}^{-1} = \rho \underline{S}_k^{-1} + X_{k+1} X_{k+1}^t \tag{3.56}$$

where $\underline{X}_{k+1}$ is the vector containing the most recent observations and ρ is an exponential forgetting factor. Applying the matrix inversion lemma, given by equation (3.45), to equation (3.56) gives

$$\underline{S}_{k+1} = \underline{S}\ \frac{1}{\rho} \left[\underline{I} - \frac{X_{k+1} \underline{X}_{k+1}^t \underline{S}_k}{\kappa_{k+1}^2} \right] \tag{3.57}$$

where

$$\kappa_{k+1}^2 = \alpha + X_{k+1} S_k \underline{X}_{k+1}^t \tag{3.58}$$

Multiplying both sides by X_{k+1}

$$\underline{S}_{k+1} X_{k+1} = \frac{S_k}{\rho} \left[1 - \frac{\kappa_{k+1}^2 - \alpha}{\kappa_{k+1}^2} \right] X_{k+1}$$

$$\underline{S}_{k+1} X_{k+1} = \frac{\underline{S}_k X_{k+1} \alpha}{\rho \kappa_{k+1}^2} \tag{3.59}$$

and solving for $\underline{S}_{k+1}^{-1}$

$$\underline{S}_{k+1}^{-1} = \left[\frac{\underline{S}_k \alpha}{\kappa_{k+1}^2 \rho} \right]^{-1} \tag{3.60}$$

This covariance matrix update equation can be used to derive the same estimate update equation as the recursive least squares estimator [22], that is

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \frac{\underline{S}_k X_{k+1}}{\kappa_{k+1}^2} [y_{k+1} - X_{k+1}^t \hat{\theta}_k^t] \tag{3.61}$$

Comparison of this equation with the previous recursive least squares estimate update equation, equation (3.49) shows that by setting

$$Kg_{k+1} = \frac{\underline{S}_k X_{k+1}}{\alpha + X_{k+1}\underline{S}_k X^t_{k+1}} \qquad (3.51)$$

that equation (3.61) and equation (3.49) are identical.

In order to update the positive definite covariance matrix, it is factored into the product

$$\underline{S} = \underline{S}^{s}\, \underline{S}^{st} \qquad (3.62$$

where $\underline{S}^{s}$ is a square matrix, always positive definite, and the square root of $\underline{S}$. This factorization is equivalent to the Cholesky decomposition [23].

Using the factorization of equation (3.62) the covariance matrix recursion equation, equation (3.57) can be rewritten as

$$\underline{S}^{s}{}_{k+1}\underline{S}^{s}{}^{t}_{k+1} = \frac{\underline{S}^{s}_{k}}{\rho}\left[\underline{\underline{I}} - \frac{\psi_{k+1}\psi^{t}_{k+1}}{\kappa^{2}_{k+1}}\right]\underline{S}^{st}_{k} \qquad (3.63)$$

where

$$\psi_{k+1} = S^{st}_{k}\underline{X}_{k+1} \qquad (3.64)$$

Equation (3.63) is now rearranged symmetrically to form the basis of the square root procedure

$$\underline{S}^{s}_{k+1}\underline{S}^{s}{}^{t}_{k+1} = \frac{S_k}{\sqrt{\rho}}\left[\underline{\underline{I}},\ \frac{j\psi_{k+1}}{\kappa_{k+1}}\right]\underline{\Delta\Delta}^{t}\left[\begin{array}{c}\underline{\underline{I}}\\ \dfrac{j\psi_{k+1}}{\kappa_{k+1}}\end{array}\right]\frac{S^{t}_{k}}{\sqrt{\rho}} \qquad (3.65)$$

where $j=\sqrt{-1}$ and $\underline{\Delta}$ is any orthogonal matrix, therefore

$$\underline{\Delta\Delta}^{t} = \underline{\underline{I}} \qquad (3.66)$$

Thus, the recursive update equation for $\underline{S}^{s}_{k+1}$ can be determined if the orthogonal matrix $\underline{\Delta}$ is found such that the imaginary vector $j\psi_{k+1}/\kappa_{k+1}$ is zeroed, that is

$$\left[\underline{\underline{I}} \; , \; \frac{j\psi_{k+1}}{\kappa_{k+1}} \right] \underline{\Delta} = [\; \underline{\Omega} \; , \; 0 \;] \tag{3.67}$$

where $\underline{\Omega}$ is a real upper triangular matrix and 0 is a vector whose elements are all equal to zero.

From equations (3.65) and (3.67) it can be seen that

$$\underline{S}^{s}_{k+1} = \frac{\underline{S}^{s}_{k}\underline{\Omega}}{\sqrt{\rho}} \tag{3.68}$$

where $\underline{S}^{s}_{k+1}$ is an upper triangular matrix since it is a product of two upper triangular matrices.

The transformations of ψ_{k+1}/κ_{k+1} to $\underline{\Omega}$, which satisfies equation (3.67) can be achieved without explicit calculation of the orthogonal matrix $\underline{\Delta}$ and using only real numbers.

For equation (3.67) to be true, it is obvious that the orthogonal matrix $\underline{\Delta}$ cannot be real. The matrix $\underline{\Delta}$ is of dimension (n+1) x (n+1) where n is the dimension of the vector X_{k+1} and ψ_{k+1}. The orthogonal matrix $\underline{\Delta}$ can be expressed as a product of the generalized elementary matrices of rotation [22]

$$\underline{\Delta}=\underline{\Delta}^{(n)}\underline{\Delta}^{(n-1)}\ldots\underline{\Delta}^{(i)}\ldots\underline{\Delta}^{(1)} \tag{3.69}$$

where

$$\underline{\Delta}^{(j)} = \begin{array}{c} \left[\begin{array}{cccccccc} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & \cdot & & & & \\ & & & & \omega_i & \cdot\ \cdot & \cdot\, j\tau_i & \\ & & & & \cdot & 1 & \cdot & \\ & & & & \cdot & \cdot & 1 & \\ & & & & -j\tau_i & \cdot\ \cdot & \cdot\,\omega_i & \end{array}\right] \begin{array}{l} \\ \\ \\ \\ \text{---}i \\ \\ \\ \text{---}n+1 \end{array} \\ \begin{array}{ccc} | & \qquad & | \\ i & & n+1 \end{array} \end{array} \qquad (3.70)$$

The only difference between an ordinary elementary matrix of rotation and the generalized matrix is that the off-diagonal elements of equation (3.70) are imaginary.

The condition of orthogonality of $\underline{\Delta}$ requires that

$$\omega_i^2 - \tau_i^2 = 1 \qquad (3.71)$$

To develop the equations to determine ω_i and τ_i consider the first product of the left-hand side of equation (3.67). The matrix multiplication results only the last two columns of the factors being changed, therefore it is sufficient to consider the transformation as

$$\begin{bmatrix} \underline{I}, & \dfrac{j\psi_{k+1}}{\kappa_{k+1}} \end{bmatrix} \underline{\Delta}^{(n)} = \begin{bmatrix} 0 & \\ 0 & \\ \cdot & \dfrac{j\psi^{(n)}}{\kappa_n} \\ \cdot & \\ \cdot & \\ 0 & \\ 1 & \end{bmatrix} \begin{bmatrix} \omega_n & j\tau_n \\ -j\tau_n & \omega_n \end{bmatrix} \qquad (3.72)$$

where

$$\psi^{(n)} = \psi = \underline{S}^{*t}_{k} X_{k+1} \qquad (3.73)$$

$$\kappa_n = \sqrt{\alpha + \psi^{(n)t}\psi^{(n)}} = \sqrt{\alpha + \psi^{t}} \qquad (3.74)$$

The two parameters of the transformation, ω_n and τ_n, are bounded by the orthogonality condition of equation

(3.71) but must also be chosen to zero the column $j\psi^{(n)}/\kappa_n$, obtained by meeting the requirement

$$\tau_n + \frac{\omega_n \psi_n^{(n)}}{\kappa_n} = 0 \tag{3.75}$$

These two conditions are met when equations (3.71) and (3.75) are solved simultaneously for τ_n and ω_n giving

$$\omega_n = \frac{\kappa_n}{\sqrt{\kappa^2 - (\psi_n^{(n)})^2}} = \frac{\kappa_n}{\kappa_{n-1}} \tag{3.76}$$

$$\tau_n = \frac{-\psi_n^{(n)}\omega_n}{\kappa_n} = \frac{-\psi_n^{(n)}}{\kappa_{n-1}} \tag{3.77}$$

where

$$\kappa_{n-1} = \sqrt{\kappa_n^2 - (\psi_n^{(n)})^2} = \sqrt{\alpha + \Sigma_{k=1}^{n-1}\psi_k^2} \tag{3.78}$$

The notation for κ can be generalized as

$$\kappa_i = \sqrt{\alpha + \Sigma_{k=1}^{i}\psi_k^2} \tag{3.79}$$

so the recursive equation for κ_i is

$$\kappa_i = \sqrt{\kappa_{i-1}^2 + \psi_i^2}\,. \tag{3.80}$$

where

$$\kappa_0 = \sqrt{\alpha} \tag{3.81}$$

The transformation of equation (3.72) using the parameters ω_n and τ_n from equations (3.76) and (3.77), respectively, gives the result

$$\begin{bmatrix} 0 & \\ 0 & \\ \cdot & \dfrac{j\psi^{(n)}}{\kappa} \\ \cdot & \\ 0 & \\ 1 & \end{bmatrix} \begin{bmatrix} \omega_n & j\tau_n \\ -j\tau_n & \omega_n \end{bmatrix} = \begin{bmatrix} \Omega_{1n} & \\ \Omega_{2n} & \dfrac{j\psi^{(n-1)}}{\kappa_{n1}} \\ \cdot & \\ \cdot & \\ \Omega_{n-1\,n} & \\ \Omega_{nn} & 0 \end{bmatrix} \tag{3.82}$$

where

$$\psi_k^{(n-1)} = \psi_k^{(n)} = \psi_k \qquad \text{for } k < n \tag{3.83}$$

$$\Omega_{kn} = \frac{\Omega^{(n)}\tau_n}{\kappa_n} = \frac{-\psi_k\psi_n}{\kappa_{n-1}} \qquad \text{for } k<n \tag{3.84}$$

$$\Omega_{nn} = \frac{\omega_n + \psi_n^{(n)}\tau_n}{\kappa_n} = \frac{\kappa_{n-1}}{\kappa_n} \tag{3.85}$$

The elementary matrix $\underline{\Delta}^{(n-1)}$ is used to zero the last component of the vector $j\psi_{n-1}^{n-1}/\kappa_{n-1}$ and simultaneously the next column of the upper triangular matrix $\underline{\Omega}$ is obtained. The entire $\underline{\Omega}$ matrix can be generalized in this manner operating only with the norms of equation (3.80).

The general i th step of this procedure is given by

$$\Omega_{ki} = \frac{\psi_k\psi_i}{\kappa_i\kappa_{i-1}} \qquad \text{for } i<k \tag{3.86}$$

$$\Omega_{ii} = \frac{\kappa_{i-1}}{\kappa_i} \tag{3.87}$$

equations (3.86) and (3.87) enable direct calculation of the matrix product of equation (3.68)

$$\underline{S}_{k+1}^{*} = \frac{\underline{S}_k^{*}\underline{\Omega}}{\sqrt{\rho}} \tag{3.68}$$

$$\underline{S}_{k+1}^{*} = \frac{\kappa_{j-1}}{\sqrt{\rho}\,\kappa_j}\left[\underline{S}_k^{*} - \frac{\psi_{k+1}\underline{S}_k^{*}\psi_{k+1}}{\kappa_{j-1}^{2}}\right] \tag{3.88}$$

By defining the vector Kg as

$$Kg_{k+1} = \frac{\underline{S}_k^{*}\psi_{k+1}}{\kappa_{j-1}^{2}}$$

$$= \frac{\underline{S}_k^s \underline{S}_k^{s\,t} X_{k+1}}{\alpha + X_{k+1} \underline{S}_k^s X_{k+1}^t} \tag{3.89}$$

which is just the recursive least squares estimator gain, the final recursion formula for the square root of the covariance matrix is given by

$$\underline{S}_{k-1}^s = \frac{\kappa_{j-1}}{\sqrt{\rho}\kappa_j}[\underline{S}_k^s - \psi_{k+1} Kg_{k+1}] \tag{3.90}$$

Thus, the entire recursive square root estimator can be summarized as follows

$$\psi_{k+1} = \underline{S}^{s\,t}_k X_{k+1} \tag{3.64}$$

$$\kappa_0 = \sqrt{\alpha} \tag{3.81}$$

$$\kappa_j = \sqrt{\kappa_{j-1}^2 + \psi_{k+1}^2} \tag{3.80}$$

$$Kg_{k+1} = \frac{\underline{S}_k^s \psi_{k+1}}{\kappa_{j-1}^2} \tag{3.89}$$

$$\underline{S}_{k-1}^s = \frac{\kappa_{j-1}}{\sqrt{\rho}\kappa_j}[\underline{S}_k^s - \psi_{k+1} Kg_{k+1}] \tag{3.90}$$

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + Kg_{k+1}[Y_{k+1} - X_{k+1}^t \hat{\Theta}_k] \tag{3.49}$$

The main advantages of this estimator are that it always ensures a positive covariance matrix. It also essentially doubles the precision of the calculations involved thus reducing the effect of the rounding error of various calculations in the recursive least squares equations (e.g. in the computation of the covariance matrix update).

One disadvantage of this estimator is that numeric degeneracy is harder to detect as the covariance matrix is never negative. Therefore off-line tests using single and double precision arithmetic are used to determine accuracy degradation [24]. However, the most serious disadvantage of this estimator is the computation time required for calculating the square root. This may become a problem if there are a large number of system parameters to be estimated, particularly if the sampling interval is short, as possibly the estimation would not be completed before the next observation was taken.

3.4 Recursive Upper-Diagonal Factorization Estimator

To further improve computational efficiency of the covariance update algorithm, while maintaining the numerical stability of the square-root estimator, Bierman (1976) introduced the upper-diagonal (U-D) factorization method. This method guarantees the non-negativity of the computed covariance matrix without involving square roots thus reducing the computation time and the storage requirements[25].

In U-D factorization the covariance matrix follows from the Cholesky factorization of a positive definite matrix and is constructed as

$$\underline{S} = \underline{U}\,\underline{D}\,\underline{U}' \qquad (3.91)$$

where $\underline{U}$ is a square unit upper triangular matrix and $\underline{D}$ is a square diagonal matrix. These triangular matrices reduce the number of arithmetic operations.

The formulation of the U-D factorization method begins by considering the classic Kalman update algorithm which combines an a priori estimate and error covariance Θ and S, respectively, with a scalar observation z, to construct an updated estimate and error covariance Θ and S where

$$z = X^t \Theta + \nu \tag{3.92}$$

The updated factors act as a priori estimates for the next measurement thus the estimation method is not self-starting.

The Kalman update formula are

$$Kg_{k+1} = \frac{S_k X_{k+1}}{\alpha} \tag{3.93}$$

$$\alpha = X_{k+1}^t \, S_k \, X_{k+1} + r \tag{3.94}$$

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + Kg_{k+1}(z - X_{k+1}^t \, \hat{\Theta}_k) \tag{3.95}$$

$$\hat{S}_{k+1} = S_k \, Kg_{k+1} \, X_{k+1} \, S_k \tag{3.96}$$

where Kg is the gain, and r is the measurement error variance, $r = E(\nu^2)$ and $E(\nu) = 0$.

The covariance U-D factors update algorithm development as given by Bierman [23] begins by factoring $\underline{S}$ in Equation (3.96) using Equation (3.91) to yield

$$\hat{\underline{S}} = \hat{\underline{U}} \, \hat{\underline{D}} \, \hat{\underline{U}}^t = \underline{U}[\underline{D} - \alpha^{-1}(\underline{D}\underline{U}^t X)(\underline{D}\underline{U}^t X)^t]\underline{U}^t \tag{3.97}$$

If the vectors f and v are defined by

$$f = \underline{U}^t X \tag{3.98}$$

$$v = \underline{D} f \qquad (v_i = d_i \, f_i) \tag{3.99}$$

and for $\bar{\underline{U}}$ and $\bar{\underline{D}}$ the U-D factors of $\underline{D} - vv^t/\alpha$, it follows that

$$\bar{\underline{U}}\ \bar{\underline{D}}\ \bar{\underline{U}}^t = \underline{D} - \frac{vv^t}{\alpha} \qquad (3.100)$$

so substituting equation (3.98) and equation (3.99) into equation (3.100) gives

$$\bar{\underline{U}}\ \hat{\underline{D}}\ \bar{\underline{U}}^t = \underline{D} - \alpha^{-1}(\underline{D}\underline{U}^t X)\ (\underline{D}\underline{U}^t X)^t \qquad (3.101)$$

Now from equation (3.97) since

$$\hat{\underline{U}}\ \hat{\underline{D}}\ \hat{\underline{U}}^t = (\underline{U}\bar{\underline{U}})\ \bar{\underline{D}}\ (\underline{U}\bar{\underline{U}})^t \qquad (3.102)$$

and since both $\bar{\underline{U}}$ and $\underline{U}$ are unit upper triangular, that is

$$\hat{\underline{U}} = \underline{U}\ \bar{\underline{U}} \text{ and } \hat{\underline{D}} = \bar{\underline{D}} \qquad (3.103)$$

it follows that the updated covariance matrix U-D factors can be determined in terms of the simpler factorization of

$$\bar{\underline{U}}\ \hat{\underline{D}}\ \bar{\underline{U}} = \underline{D} - \frac{vv^t}{\alpha} \qquad (3.104)$$

Application of the Agee-Turner positive definite factorization update theorem (see Appendix A) to equation (3.104) gives the following representation for $\bar{\underline{U}}$ and $\hat{\underline{D}}$

$$\hat{d}_j = d_j + c_j\ v_j^2 \qquad (3.105)$$

$$c_{j-1} = \frac{c_j\ d_j}{\hat{d}_j} \qquad (3.106)$$

$$\bar{U}_{ij} = \frac{c_j v_j v_i}{\hat{d}_j} \qquad i=1,2,\ldots,j-1 \qquad (3.107)$$

Equations (3.105) through (3.107) are backwards recursive for j=n to j+1, where n is the number of parameters to be identified and $c_n = -\alpha^{-1}$.

However since all the d's are theoretically positive it follows from equation (3.106) that all the c_j's have the same sign and as c_n is negative the c_j's will all be negative. This indicates that the updated diagonals will be computed as differences which can lead to loss of accuracy due to cancellation and may even result in calculation of negative d elements. To ensure that this will not happen a different representation for $\hat{d}_j$ is needed.

Rearranging equations (3.105) and (3.106) gives

$$\hat{d}_j = d_j\,(1 + c_j d_j f_j^2) \qquad (3.108)$$

$$c_{j-1}^{-1} = c_j^{-1} + d_j f_j^2 \qquad (3.109)$$

but since

$$c_n^{-1} = -\alpha_n = -(r + \Sigma_{i=1}^{n}\, d_i f_i^2) \qquad (3.110)$$

it follows that

$$c_j^{-1} = -\alpha_j = -(r + \Sigma_{i=1}^{j}\, v_i f_i) \qquad (3.111)$$

Substitution of equation (3.111) into equation (3.108) yields

$$\hat{d}_j = \frac{d_j\,(\alpha_j - d_j f_j^2)}{\alpha_j} \qquad (3.112)$$

then using equation (3.111) with equation (3.109) and rearranging gives

$$\alpha_j = \alpha_{j-1} + d_j f_j^2 \qquad (3.113)$$

which when substituted into the numerator of equation (3.112) results in the updated diagonal elements

$$\hat{d}_j = \frac{d_j(\alpha_{j-1} + d_j f_j^2 - d_j f_j^2)}{\alpha_j}$$

$$\hat{d}_j = \frac{d_j \alpha_{j-1}}{\alpha_j} \qquad (3.114)$$

Since the diagonal elements have been calculated the algorithm now focuses on the updated elements of U which must now be calculated. Recalling from equation (3.103) that $\hat{U} = U\bar{U}$ and using the results of equation (3.111) in equation (3.107) gives

$$\bar{U}_{ij} = \frac{-v_j v_i}{\alpha_j d_j} \tag{3.115}$$

Then, setting

$$\lambda_j = \frac{-v_j}{\alpha_j \hat{d}_j} = \frac{-f_j}{\alpha_{j-1}} \tag{3.116}$$

allows $\bar{U}$ to be written as

$$\bar{U} = \underline{\underline{I}} + [\ 0 \ \ \lambda_2 v^{(1)} \ \ \lambda_3 v^{(2)} \ \ldots \ \lambda_n v^{(n-1)}] \tag{3.117}$$

where

$$(v^{(j)})^t = (v_1 \ \ldots \ v_j \ 0 \ \ldots \ 0) \tag{3.118}$$

Since $\bar{U}$ and U are triangular matrices of the form of equation (3.117), matrix products are easily and efficiently evaluated. A direct computation of $U\bar{U}$ gives

$$\hat{U} = U + [\ 0 \ \ \lambda_2 Uv^{(1)} \ \ \lambda_3 Uv^{(2)} \ \ldots \ \lambda_n Uv^{(n)}] \tag{3.119}$$

such that

$$\hat{U} = [\hat{u}_1 \ \ldots \ \hat{u}_n] \text{ and } U = [u_1 \ \ldots \ u_n]$$

where $\hat{u}_i$ and u_i are the i th columns of the $\hat{U}$ and U matrices, respectively.

Now that the updated U-D factors have been obtained the updated gain must be computed. Factoring S of equation (3.93) into its U-D factors and substituting the f and v vectors for $U^t X$ and $\underline{D}$ f, respectively, gives

$$K = \frac{\underline{U}\underline{D}\underline{U}^t X}{\alpha} = \frac{\underline{U}v}{\alpha_n} \tag{3.120}$$

The labelling of the columns of $\hat{U}$ and U along with the special structure of v and the inherent recursion of $K_j = Uv^{(j)}$ leads to the construction of the expression for the numerator of equation (3.120) to be stated as

$$K_{j+1} = Uv^{(j+1)} = K_j + v_j u_j \qquad (3.121)$$

so the updated estimator gain for n parameters is

$$Kg = \frac{K_{n+1}}{\alpha_n} \qquad (3.122)$$

The updating algorithm of the U-D factorization estimator can now be summarized as follows, for j=1

$$f = \underline{U}^t X \qquad (3.98)$$

$$v = \underline{D}f \quad ; \ v_i = d_i f_i \qquad (3.99)$$

$$\alpha_1 = r + v_1 f_1 \qquad (3.111)$$

$$\hat{d}_1 = \frac{d_1 r}{\alpha_1} \qquad (3.114)$$

and then for j = 2, . . . , n, recursively cycle through the following four equations

$$\alpha_j = \alpha_{j-1} + v_j f_j \qquad (3.113)$$

$$\hat{d}_j = \frac{d_j \alpha_{j-1}}{\alpha_j} \qquad (3.114)$$

$$\hat{u}_j = u_j + \lambda_j K_j \quad ; \ \lambda_j = -f_j/\alpha_{j-1} \qquad (3.119)$$

$$K_{j+1} = K_j + v_j u_j \qquad (3.121)$$

and finally

$$Kg = \frac{K_{n+1}}{\alpha_n} \qquad (3.122)$$

so the estimated parameters can now be calculated by equation (3.95).

This U-D algorithm is ideally suited to computer implementation because of its recursive structure. The $\hat{\underline{U}}$ computation employed in this algorithm is advantageous because the need for storing the $\bar{\underline{U}}$ matrix is circumvented and the number of calculations is reduced by a factor of n as the computation of $\bar{\underline{U}}$ and $\hat{\underline{U}}$ required a total of $n(n-1)/2$ additions and multiplications while direct computation of $\underline{U}\bar{\underline{U}}$ requires $n(n-1)(n-2)/2$ such operations. [23]

From equation (3.114) it can be seen the the computation of $\underline{D}$ avoids differencing and also negative values so long as the measurement error variance is positive. These features result in a numerically stable algorithm. Furthermore, it can also be observed that the elements of $\underline{D}$ can become very small without affecting the stability of the algorithm.

3.5 Recursive Learning Estimator

Nagumo and Noda [26] proposed a method for linear system identification which is based on the error-correcting training procedure in learning machines and referred to it as learning identification. The method does not provide a lea[illegible]ares estimate of the parameters and is slower in co[illegible]. The main advantage of the technique is the sim[illegible]mulation of the identification algorithm.

The systems are considered to be described by a representation and the measurements are assumed to be noise-free. Given the system

$$y_j = \theta_1 x_{j-1} + \theta_2 x_{j-2} + \; . \; . \; . \; + \theta_m x_{j-m} = \Sigma_{i=1}^{m} \; \theta_i x_{j-i} \quad (3.123)$$

where θ_i are the system parameters and x_{j-i} are the corresponding observations. Let $\theta_i^{(j)}$ (i=1,2,...,m; j=m+1,m+2,...) be the set of estimates of θ_i at iteration step j by a recursive procedure, calculated by an equation that will be derived, then the output predicted by the identifier y^{+}_{j}, expressed as

$$y_j^{+} = \Sigma_{i=1}^{m} \hat{\theta}_i^{(j)} x_{j-i} \quad (3.124)$$

approaches y_j at iteration step j.

Define Θ, $\hat{\Theta}_j$ and X_j to be m dimensional vectors containing.

$$\Theta = [\; \theta_1 \; \theta_2 \; . \; . \; . \; \theta_m]^t \quad (3.125)$$

$$\hat{\Theta}_j = [\; \theta_1^{(j)} \; \theta_2^{(j)} \; . \; . \; . \; \theta_m^{(j)}]^t \quad (3.126)$$

$$X_j = [X_{j-1} \; X_{j-2} \; . \; . \; . \; X_{j-m}]^t \quad (3.127)$$

The vector X_j is a modified observation vector obtained by collecting m terms of the original observations

$$_2, \; x_3, \; . \; . \quad (3.128)$$

hence

$$y_j = (\Theta, X_j) \quad (3.129)$$

$$y_j^{+} = (\hat{\Theta}_j, X_j) \quad (3.130)$$

The estimates, y_j^{+}, are updated by adding the identification error between the output and estimator predicted output proportional to the magnitude of the corresponding element of the observation vector X_j, according to

$$y_{j+1}^{+} = y_j^{+} + \frac{(y_j - y_j^{+}) \; X_j}{||X_j||^2} \quad (3.131)$$

where

$$||X_j||^2 = \Sigma^m_{i=1}(x_{j-i})^2 \qquad (3.132)$$

The procedure is error-correcting in the sense that

$$\begin{aligned}(\hat{\Theta}_{j+1}, X_j) &= (\hat{\Theta}_j + \Delta\hat{\Theta}_j, X_j) \\ &= (\hat{\Theta}_j, X_j) + (\Delta\hat{\Theta}_j, X_j) \\ &= y_j^* + (y_j - y_j^*) = y_j \qquad (3.133)\end{aligned}$$

The updating procedure can be enhanced [26] by the introduction of an error-correcting coefficient to equation (3.131) to give

$$y^*_{j+1} = y^*_j + \frac{\alpha_1(y_j - y^*_j)X_j}{||X_j||^2} \qquad (3.134)$$

where $0 < \alpha_1 < 2$.

The convergence of the method is shown, based on a geometrical interpretation of the updating procedure, by Nagumo and Noda [26]. Convergence of $\hat{\Theta}_i$ to Θ_i is attained for $||\Theta - \hat{\Theta}_j|| \to 0$ when $j \to \infty$. The following is the proof of the necessary and sufficient condition based on the development given by Nagumo and Noda [26] and Graupe [15].

Expressing equation (3.134) in the form

$$y^*_{j+1} = y^*_j + \frac{\alpha_1(\Theta - \hat{\Theta}_j, X_j)X_j}{||X_j||^2}$$

$$y^*_{j+1} = y^*_j + \frac{\alpha_1(\Theta - \hat{\Theta}_j)X_jX_j^*}{||X_j||^2} \qquad (3.135)$$

and letting $\chi_j = \Theta - \hat{\Theta}_j$ allows equation (3.135) to be written as

$$\chi_{j+1} = K_j\chi_j \quad (j \geq m+1) \qquad (3.136)$$

where

$$K_j = \underline{\underline{I}} - \frac{\alpha_1 X_j X_j^\dagger}{||X_j||^2} \qquad (3.137)$$

and noting that $X_j X_j^\dagger = ||X_j||^2$, it follows that

$$K_j^2 = \underline{\underline{I}} - \frac{2\alpha_1 X_j X_j^\dagger}{||X_j||^2} + \frac{\alpha_1^2 X_j X_j^\dagger X_j X_j^\dagger}{||X_j||^4}$$

$$K_j^2 = \underline{\underline{I}} - \frac{\alpha_1(2 - \alpha_1) X_j X_j^\dagger}{||X_j||^2} \qquad (3.138)$$

therefore convergence, that is $\widehat{\Theta}_j = \Theta$ occurs when $||x_j|| \to 0$ as $j \to \infty$.

The convergence criteria is proven from equation (3.136)

$$||x_{j+1}||^2 = (K_j x_j, K_j x_j) = (x_j\ , K_j^2 x_j) \qquad (3.139)$$

$$= \underline{\underline{I}} - \frac{\alpha_1(2 - \alpha_1) X_j X_j^\dagger}{||X_j||^2} x_j x_j^\dagger) \qquad (3.140)$$

Multiplying the right-hand-side of equation (3.140) by $||x_j||^2/||x_j||^2$

$$||x_{j+1}||^2 = (1 - \Phi_j)||x_j||^2 \qquad (3.141)$$

where

$$\Phi_j = \alpha_1(2 - \alpha_1)\left[\frac{||X_j||}{||X_j||\ ||x_j||}\right]^2 \qquad (3.142)$$

and

$$||x_{j+1}||^2 = ||x_{m+1}||^2 \prod_{k=m+1}^{j} (1 - \Phi_k) \qquad (3.143)$$

Since $0<\alpha_1<2$ then $0<\Phi_k\le 1$ and the necessary and sufficient condition for $||x_j|| \to 0$ as $j \to \infty$ is given by

$$\Sigma_{j=m+1}^{\infty}\ \Phi_j = \infty \qquad (3.144)$$

If the input sequence is random then the observation vector

X_j is random and therefore the sequence

$$\Phi(m+1), \Phi(m+2), \Phi(m+3), . . . \tag{3.145}$$

is random where Φ_j is dependent on $\Phi_{m+1}, \Phi_{m+2}, . . ., \Phi_{j-1}$.

By defining

$$\eta_j = E\{\Phi_j | \Phi_{m+1}, \Phi_{m+2}, . . . , \Phi_{j-1}\} \tag{3.146}$$

and assuming that

$$\Sigma_{j=m+1}^{\infty} \eta_j = +\infty \tag{3.147}$$

the corollary suggested by Doob [27] from the Borel-Cantelli lemma shows that equation (3.147) is necessary and sufficient for the convergence of

$$||X_j|| \to 0 \ (j \to \infty) \tag{3.148}$$

The corollary states that given y_1, y_2, ... as a sequence of uniformly bounded nonnegative random variables , and letting $P_j = E\{y_j | y_1 . . . y_{j-1}\}$, the series $\Sigma_1^{\infty} y_j(\omega)$ converges for almost all ω for which $\Sigma_1 P_j(\omega)$ converges. Therefore if in a sequence of random variables $\Phi_{m+1}, \Phi_{m+2}, . . .$ there occurs an infinity of Φ_j's such that $\eta_k > \delta$, where δ is a positive constants, since $\Sigma\eta_j = +\infty$ then $\Sigma\Phi_j = \infty$ which would satisfy equation (3.148).

The recursive learning method has been shown, theorectically and experimentally , to converge for a wide class of input signals. The estimator algorithm is given by

$$Kg_{k+1} = \frac{\alpha X_{k+1}^t}{X_{k+1} X_{k+1}^t + \rho_1} \tag{3.149}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + Kg_{k+1}[y_{k+1} - X_{k+1}^t \hat{\theta}_k] \tag{3.150}$$

where ρ_1, with a value between 0 and 1 not in the original algorithm is added to ensure division by zero never occurs.

Kg is the gain of the estimator, and α is the scalar convergence factor $(0<\alpha<2)$.

The recursive learning estimator calculates the parameter estimates without the parameters covariances used in the previous methods. The covariance matrix is replaced by the error-correcting coefficient, α, thus the computation time is considerably less than the other methods. Unfortunately with the lack of the covariance matrix there is no measure of the accuracy of the parameter estimates and the algorithm is suboptimal in the least squares sense. To obtain the optimal parameters the initial guesses must be close to the actual values.

3.6 Recursive Maximum Likelihood Estimator

3.6.1 Maximum Likelihood Estimator

The maximum likelihood method was developed by Fisher in 1912 [16] although the basic ideas were known to Gauss in 1809 [16]. The method involves the construction of a real function of the unknown parameters and process data and the parameter estimate is obtained by determining the parameter values which maximize the function. This real function is called the likelihood function and is essentially the probability density function of the observation. To determine this function it is necessary to have presumptive knowledge in order to write the conditional probability density function of the observed output. For sequential observations the conditional distribution of the output at time $k+1$ based on data obtained up to time k must be determined. This is a prediction problem so the likelihood function can be expressed as a product of the conditional densities of the prediciton errors.

The derivation of the maximum likelihood estimator as given by Franklin and Powell [28] begins by introducing a probability density function for the random variable involved. The probability density function is not restricted to any one function, however the normal or Gaussian distribution is generally used.

For a set of n random variables, x, with a joint normal distribution the mean vector μ, and the non-singular covariance matrix, S, can be defined as

$$E\{X\} = \mu \qquad (3.151)$$

$$E\{(X-\mu)(X-\mu)'\}=S \qquad (3.152)$$

Then the probability density function can be written as

$$f(\xi) = \frac{1}{[(2\pi)^n \det S]^{1/2}} \exp\left[\frac{-1}{2}(\xi-\mu)' S^{-1}(\xi-\mu)\right] \qquad (3.153)$$

If the elements of X are mutually uncorrelated and have identical means, μ, and variances, σ^2, then $S=\sigma^2\underline{I}$ and $\det S=(\sigma^2)^n$ and equation (3.153) can be written as

$$f(\xi) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(\xi_i-\mu)^2\right] \qquad (3.154)$$

The maximum-likelihood estimate is calculated on the basis of an assumed structure for the probability density function of the available observations. The density function can be written as a function of Θ, the parameters, and is considered to be a measure of the "likelihood" for a particular value since the probability that a particular x_i is in the range $r \leq x_i \leq s$ is given by

$$\Pr\{r \leq x_i \leq s\} = \int_r^s f_{x_i}(\xi_i|\Theta)d\xi_i \qquad (3.155)$$

where

$$f_{x_i}(\xi|\Theta) = (2\pi\sigma^2)^{-n/2}\exp\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{(\xi_i-\Theta)^2}{\sigma^2}\right] \qquad (3.156)$$

when f is a function of the parameters, Θ, then f is the likelihood function. The maximum-likelihood estimate $\hat{\Theta}_{ML}$, is the estimate of the parameters which maximizes $f_X(X|\Theta)$, the probability density function of the actual data.

Suppose the data consists of a set of observations having a density given by equation (3.154) but an unknown mean/ the maximum-likelihood estimate of the mean μ is calculated by setting the derivative of $f(X|\Theta)$ with respect to Θ to zero. Since the logarithm of f is a simpler function than f, it is convenient to compute

$$\frac{d(\log f)}{d\Theta} = \frac{1}{f}\frac{df}{d\Theta} = 0 \qquad (3.157)$$

Since the value of f cannot be zero in the neighborhood of its maximum therefore $df/d\Theta = 0$ for the derivative to be zero.

Define

$$l(X|\theta) = -\log f_X(X|\theta) \qquad (3.158)$$

as the likelihood function for μ and if f is normally distributed then

$$l(X|\theta) = \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i-\theta)^2}{\sigma^2} \qquad (3.159)$$

and

$$\frac{\partial l}{\partial\theta} = \frac{-1}{2\sigma^2}\sum_{i=1}^{n} 2(x_i-\theta) = \frac{-1}{\sigma^2}\{\sum_{i=1}^{n} x_i - n\theta\} \qquad (3.160)$$

Setting the deriv[illegible] zero to get the maximum-likelihood estimate gives

$$\sum_{i=1}^{n} x_i - n\hat{\theta}_{ML} = 0 \qquad (3.161)$$

$$\hat{\theta}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (3.162)$$

Therefore for an unknown mean of normal distribution the maximum-likelihood estimate is the sample mean which is also a least squares estimate, unbiased and consistent (that is, the estimates converge to the "true" parameters as the number of samples goes to infinity) [28].

The principles of maximum likelihood estimation can be applied further to dynamic system identification. Consider a model with simple white-noise disturbances

$$y_k = f_1 y_{k-1} + \ldots + f_n y_{k-n} + g_1 u_{k-1} + \ldots + g_n u_{k-n} + v_k \qquad (3.163)$$

Assume the distribution of $V=[v_n \ldots v]^t$ is normal with zero mean and variance $\sigma^2\underline{\underline{I}}$, $N(0,\sigma^2\underline{\underline{I}})$, thus

$$E\{v_k v_j\} = 0 \quad (k\neq j) \qquad (3.164)$$

In order to estimate f_i, g_i and σ^2 by a maximum likelihood method, from a sequence of observed y_k, it is necessary to know the probability density function of the observed y_k for known values of the parameters. Assuming f_i, g_i, y and u are known, v_k can be computed from the observed y, u and assumed true f_i and g_i, and the distribution of y is immediately determined by the distribution of v.

For the true parameters, Θ°,

$$Y(N) = \underline{X}\Theta^\circ + V(N) \tag{3.165}$$

where

$$Y(N) = [y_n \ldots y]^t \tag{3.166}$$

$$\Theta^\circ = [f_1 \ldots f_n \; g_1 \ldots g_n]^t \tag{3.167}$$

To obtain the probability density function $f(Y|\Theta^\circ)$ for the maximum likelihood method, V(N) must be computed from Y(N) using the probability density function of V(N). Therefore in order to compute V(N) from Y(N) the inverse of the model of the process is required The noise sequence, from equation (3.165) is

$$V(N) = Y(N) - X\Theta^\circ \tag{3.168}$$

Now, since the density function of V is $N(0,\sigma^2\underline{I})$ it follows that

$$f(Y|\Theta^\circ) = (2\pi\sigma^2)^{-m/2}\exp\left[\frac{-1}{2}\,\frac{(Y-X\Theta^\circ)^t(Y-X\Theta^\circ)}{\sigma^2}\right] \tag{3.169}$$

where m=N-n+1 (the number of samples in Y). The likelihood function is by definition $f(Y|\Theta)$ which is given by equation (3.169) with Θ° replaced by the general vector Θ, and taking the logarithm of f gives

$$l(Y|\Theta)=\log(2\pi\sigma^2)^{-m/2}+\log\left\{\exp\left[\frac{-1}{2}-\frac{(Y-X\Theta)'(Y-X\Theta)}{\sigma^2}\right]\right\}$$

or

$$l(Y|\Theta) = \frac{-m}{2}\log 2\pi - \frac{m}{2}\log\sigma^2 - \frac{1}{2}\frac{(Y-X\Theta)'(Y-X\Theta)}{\sigma^2} \qquad (3.170)$$

The estimates $\hat{\Theta}_{ML}$ and $\hat{\sigma}^2_{ML}$ are the values of Θ and σ^2 which minimize $l(Y|\Theta)$. These estimates are found by setting the partial derivative of l with respect to Θ and σ^2 to zero which gives

$$\frac{1}{\hat{\sigma}^2_{ML}}\left[X'X\hat{\Theta}_{ML} - X'Y\right] = 0 \qquad (3.171)$$

$$\frac{-m}{2\hat{\sigma}^2_{ML}} + \frac{(Y-X\hat{\Theta}_{ML})'(Y-X\hat{\Theta}_{ML})}{2\hat{\sigma}^4_{ML}} = 0 \qquad (3.172)$$

Since equation (3.171) is the identical "normal" equation of the least squares method, equation (3.13), then for this case $\hat{\Theta}_{ML} = \hat{\Theta}_{LS}$ and thus $\hat{\Theta}_{ML}$ is also asymptotically unbiased and consistent. The estimate of $\hat{\sigma}^2_{ML}$ can be determined by decoupling $\hat{\sigma}^2_{ML}$ and $\hat{\Theta}_{ML}$ and solving equation (3.172) to give

$$\hat{\sigma}^2_{ML} = \frac{1}{m}(Y-X\hat{\Theta})'(Y-X\hat{\Theta})$$

$$\hat{\sigma}^2_{ML} = \frac{1}{N-n+1}\sum_{k=n}^{N} e^2(k|\hat{\Theta}_{ML}) \qquad (3.173)$$

To obtain the $\hat{\Theta}_{ML}$ estimate the general model is modified so that past values of the noise are weighted by the h_i parameters

$$y_k = \Sigma_{i=1}^{n} f_i y_{k-i} + \Sigma_{i=1}^{n} g_i u_{k-i} + \Sigma_{i=1}^{n} h_i v_{k-i} + v_k \qquad (3.174)$$

where the distribution of V is still $N(0,\sigma^2\underline{\underline{I}})$. The Least-squares estimate is now biased if h_i are non-zero.

To simplify the derivation, assume h_1 is the only non-zero coefficient and write equation (3.174) with the noise terms on one side and define z_k as

$$z_k = v_k + h_1 v_{k-1}$$

$$z_k = y_k - \Sigma_{i=1}^{n} f_i y_{k-i} - \Sigma_{i=1}^{n} g_i u_{k-i} \qquad (3.175)$$

Define a reduced parameter vector $\bar{\Theta}=[f_1 \ldots f_n \; g_1 \ldots g_n]^t$ which does not include the h_i parameters, thus

$$Z(N) = Y(N) - X\bar{\Theta}^{\circ} \qquad (3.176)$$

where

$$Z(N) = [z_n \; \ldots \; z_N]^t \qquad (3.177)$$

But z_k is the sum of two random variables, v_k and v_{k-1}, each having a normal distribution , therefore Z is also normally distributed. The mean and covariance of z_k is calculated by

$$E\{z_k\} = E\{v_k + h_1 v_{k-1}\} = 0 \quad \text{(for all k)} \qquad (3.178)$$

$$\left.\begin{aligned} E\{z_k z_j\} &= E\{(v_k + h_1 v_{k-1})(v_j + h_1 v_{j-1})\} \\ &= \sigma^2(1+h_1^2) \qquad (k=j) \\ &= \sigma^2 h_1 \qquad (k=j+1) \\ &= \sigma^2 h_1 \qquad (k=j-1) \\ &= 0 \text{(elsewhere)} \end{aligned}\right\} \qquad (3.179)$$

Thus the covariance of Z(N) is

$$\underline{\underline{S}} = E\{Z(N)Z^t(N)\} \qquad (3.180)$$

$$\underline{\underline{S}} = \begin{bmatrix} 1+h_1^2 & h_1 & 0 & \cdot & \cdot & 0 \\ h_1 & 1+h_1^2 & h_1 & \cdot & & \cdot \\ 0 & h_1 & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & h_1 \\ 0 & \cdot & \cdot & \cdot & h_1 & 1+h_1^2 \end{bmatrix} \sigma^2 \qquad (3.181)$$

Now that the mean and covariance are known the probability density function of Z(N) is

$$g(Z(N)|\Theta^{\circ}) = [(2\pi)^m \det S]^{-1/2}\exp\left[\frac{-1}{2}Z'S^{-1}Z\right] \qquad (3.182)$$

Substituting $(Y-X\bar{\Theta})$ for Z and $\bar{\Theta}$ for Θ° from equation (3.176) into equation (3.182) gives

$$f(Y|\Theta)= \frac{1}{[(2\pi)^m \det S]^{1/2}}\exp\left[\frac{-1}{2}(Y-X\bar{\Theta})'S^{-1}(Y-X\bar{\Theta})\right] \qquad (3.183)$$

and taking the logarithm of f yields

$$l(Y|\Theta) = \frac{m}{2}\log 2\pi + \frac{1}{2}\log(\det S) + \frac{1}{2}(Y-X\bar{\Theta})'(Y-X\bar{\Theta}) \qquad (3.184)$$

By inspecting equation (3.184) it can be seen that $l(Y|\Theta)$ is dependent on the f_i's and g_i's from $\bar{\Theta}$ and is thus quadratic in these parameters but it depends on the h_i's through S and detS which is definitely not quadratic. Therefore an explicit formula for the maximum-likelihood estimate is not possible so a numerical algorithm must be found to obtain $\hat{\Theta}_{ML}$.

To obtain an algorithm to be used for the numerical search for $\hat{\Theta}_{ML}$, equation (3.175) is first rewritten to form the inverse system

$$v_k = y_k - \Sigma_{i=1}^{n} f_i y_{k-i} - \Sigma_{i=1}^{n} g_i u_{k-i} - \Sigma_{i=1}^{n} h_i v_{k-i} \qquad (3.185)$$

where v_k has a normal distribution with zero mean and unknown scalar variance S_v. The vector V(N), defined previously, has a normal distribution with zero mean and covariance

$$S = S_\nu \underline{\underline{I}}_m = E\{V\ V'\} \tag{3.186}$$

where $\underline{\underline{I}}_m$ is an m x m identity matrix and m=N-n+1 (the number of elements in V). The density function in terms of the true parameters, Θ°, can be written as

$$f(V|\Theta) = [(2\pi)^m S_\nu^{\ m}]^{-1/2} \exp\left[\frac{-1}{2}\sum_{k=n}^{N}\frac{v_k^2}{S_\nu}\right] \tag{3.187}$$

The likelihood function is found by substituting arbitrary parameters Θ in $f(V|\Theta)$ and using e_k as the output of equation (3.185) when $\Theta \neq \Theta^\circ$ to give

$$l(E|\Theta) = \frac{m}{2}\log 2\pi + \frac{m}{2}\log\hat{S}_\nu + \frac{1}{2\hat{S}_\nu}\sum_{k=n}^{N} e^2{}_k \tag{3.188}$$

As before, $\hat{S}_\nu$ is calculated by setting $\partial l/\partial\hat{S}_\nu = 0$ which gives

$$\hat{S}_\nu = \frac{1}{N-n+1}\sum_{k=n}^{N} e_k^2 \tag{3.189}$$

The problem again, is to find estimates for f_i, g_i and h_i it is necessary to construct a numerical algorithm for minimizing $l(E|\Theta)$. There are many such algorthims but the algorithm based on a method by Newton is presented here. The basic concept of the algorithm is: given the k th estimate of $\hat{\Theta}$, find a (k+1) st estimate which makes

$$l(E|\hat{\Theta}_{k+1}) < l(E|\hat{\Theta}_k) \tag{3.190}$$

This is done by expanding l about Θ_k and choosing Θ_{k+1} so the quadratic terms of l are minimized. Let

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \delta\hat{\Theta} \tag{3.191}$$

then

$$l(E|\hat{\Theta}_k) = l(E|\hat{\Theta}_k + \delta\hat{\Theta})$$

$$= h + \phi'\delta\hat{\Theta} + \frac{1}{2}\delta\hat{\Theta}Q\delta\hat{\Theta} + \ldots \qquad (3.192)$$

where

$$h = l(E|X_k) \qquad (3.193)$$

$$\phi' = \left.\frac{\partial l}{\partial \Theta}\right|_{\Theta=\hat{\Theta}_k} \qquad (3.194)$$

$$Q_i = \left.\frac{\partial^2 l}{\partial\Theta\partial\Theta}\right|_{\Theta=\hat{\Theta}_k} \qquad (3.195)$$

Select δΘ so that the quadratic approximation of l is minimized, that is $\partial l/\partial(\delta\Theta)=0$. Differentiation of equation (3.192), ignoring the higher order terms in δΘ and setting the derivative equal to zero gives

$$\frac{\partial l}{\partial(\delta\Theta)} = \phi' + \delta\Theta' Q_i = 0 \qquad (3.196)$$

Solving equation (3.196) for δΘ yields

$$\delta\Theta = -Q_i^{-1}\phi \qquad (3.197)$$

then using this [illegible] to compute $\hat{\Theta}_{k+1}$ gives

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - Q_i \text{[illegible]}$$

or

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \left[\frac{\partial^2 l}{\partial\Theta\partial\Theta}\right]^{-1}\left[\frac{\partial l}{\partial\Theta}\right] \qquad (3.198)$$

It is now necessary to express the partial derivatives in terms of the observed signals y and u. Since it was shown, from equation (3.189) that $\hat{S}_v$ is a constant, equation (3.188) can be recalled to calculate the partial derivatives in the following manner

$$\frac{\partial l(E|\Theta)}{\partial\Theta} = \frac{\partial}{\partial\Theta}\left[\frac{m}{2}\log 2\pi + \frac{m}{2}\log\hat{S}_{\upsilon} + \frac{1}{2\hat{S}_{\upsilon}}\sum_{k=n}^{N} e_k^2\right]$$

$$\frac{\partial l(E|\Theta)}{\partial\Theta} = \frac{1}{\hat{S}_{\upsilon}}\sum_{k=n}^{N} e_k\frac{\partial e_k}{\partial\Theta} \tag{3.199}$$

and therefore

$$\frac{\partial^2 l(E|\Theta)}{\partial\Theta\partial\Theta} = \frac{\partial}{\partial\Theta}\left[\frac{1}{\hat{S}_{\upsilon}}\sum_{k=n}^{N} e_k\frac{\partial e_k}{\partial\Theta}\right]$$

$$\frac{\partial^2 l(E|\Theta)}{\partial\Theta\partial\Theta} = \frac{1}{\hat{S}_{\upsilon}}\sum_{k=n}^{N}\left[\frac{\partial e}{\partial\Theta}\right]'\frac{\partial e}{\partial\Theta} + \frac{1}{\hat{S}_{\upsilon}}\sum_{k=n}^{N}\frac{\partial^2 e}{\partial\Theta\partial\Theta}e_k \tag{3.200}$$

Since the algorithm is only expected to produce an improvement in Θ and since the first derivative terms will dominate near the minimum, the algorithm can be simplified by including only the first term of equation (3.200). Now the partial derivatives from equations (3.199) and (3.200) depend only on the derivative of e with respect to Θ. Thus only $\partial e/\partial\Theta$, the sensitivities of e with respect to Θ must be computed.

Substituting e_k for v_k in equation (3.185) gives

$$\partial e_k/\partial f_i = -y_{k-i} - \Sigma_{j=1}^{n} h_j(\partial e_{k-j}/\partial f_i) \tag{3.201}$$

$$\partial e_k/\partial g_i = -u_{k-i} - \Sigma_{j=1}^{n} h_j(\partial e_{k-j}/\partial g_i) \tag{3.202}$$

$$\partial e_k/\partial h_i = -e_{k-i} - \Sigma_{j=1}^{n} h_j(\partial e_{k-j}/\partial h_i) \tag{3.203}$$

For a given value of "i", equation (3.201) is a constant coefficient difference equation in the variable $\partial e_k/\partial f_i$ with

y_{k-i} as a forcing function. Taking the z-transform of equation (3.201) and defining

$$e_{f_i} = \partial e/\partial f_i \tag{3.204}$$

allows equation (3.201) to be represented by

$$E_{f_i}(z) = -z^{-i}Y(z) - \Sigma_{j=1}^{n} h_j z^{-j} E_{f_i}(z) \tag{3.205}$$

or

$$E_{f_i}(z) = \frac{z^{-i}}{1 + \Sigma_{j=1}^{n} h_j z^{-j}} Y(z) \tag{3.206}$$

Therefore the derivative of e with respect to a_2 is z^{-1} times the partial derivative of e with respect to a_1, and so forth.

The sensitivities of e with respect to g_i and h_i can be developed in the same fashion.

Now the maximum likelihood algorithm can be constructed to compute the updated parameters Θ. First, select the initial parameter estimates and then for n observations and N total parameters, the following equations apply

$$e_k = y_k - \Sigma_{i=1}^{n} f_i y_{k-i} - \Sigma_{i=1}^{n} g_i u_{k-i} - \Sigma_{i=1}^{n} h_i v_{k-i} \tag{3.207}$$

$$\partial e_k/\partial f_i = -y_{k-i} - \Sigma_{j=1}^{n} h_j (\partial e_{k-j}/\partial f_i) \tag{3.201}$$

$$\partial e_k/\partial g_i = -u_{k-i} - \Sigma_{j=1}^{n} h_j (\partial e_{k-i}/\partial g_i) \tag{3.202}$$

$$\partial e_k/\partial h_i = -e_{k-i} - \Sigma_{j=1}^{n} h_j (\partial e_{k-i}/\partial h_i) \tag{3.203}$$

$$\hat{S}_v \frac{\partial l}{\partial \Theta} = \sum_{k=n}^{N} e_k \frac{\partial e_k}{\partial \Theta} \tag{3.208}$$

$$\hat{S}_v \frac{\partial^2 l}{\partial \Theta \partial \Theta} = \sum_{k=n}^{N} \left[\frac{\partial e_k}{\partial \Theta}\right]^t \left[\frac{\partial e_k}{\partial \Theta}\right] \tag{3.209}$$

$$\hat{S}_{\upsilon} = \frac{1}{N-n+1} \sum_{k=n}^{N} e_k^2 \qquad (3.189)$$

$$\phi = \frac{1}{\hat{S}_{\upsilon}} \left[\hat{S}_{\upsilon} \frac{\partial l}{\partial \Theta} \right] \qquad (3.210)$$

$$Q_l = \frac{1}{\hat{S}_{\upsilon}} \left[\hat{S}_{\upsilon} \frac{\partial^2 l}{\partial \Theta \partial \Theta} \right] \qquad (3.211)$$

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - Q_l^{-1} \phi \qquad (3.188)$$

These equations are calculated for successive groups of n observations until the significance of the reduction, calculated by

$$[\hat{S}_{\upsilon k+1} - \hat{S}_{\upsilon k}] / \hat{S}_{\upsilon k} \qquad (3.212)$$

reaches a user specified limit.

Although this method gives unbiased, consistent estimates it is an off-line, batch method and therefore does not utilize the new information provided immediately. It also requires a matrix inversion calculation which is time-consuming. Therefore an on-line maximum likelihood method would be useful in dynamic system identification.

3.6.2 Recursive Maximum Likelihood Estimator

A recursive algorithm for maximum likelihood estimation of parameters in a linear dynamic system was proposed by Soderstrom [29] for on-line identification.

The algorithm is derived by a recursive minimization of a time-varying loss function $V_n(\Theta)$ where

$$V_n(\hat{\theta}) = \frac{1}{2} \sum_{k=1}^{m} e^2(k|\hat{\theta}) \qquad (3.213)$$

and m is the number of samples, e is the residual at time k and $\hat{\theta}$ is the estimate of the parameter vector whose elements are the model parameters.

The variance of the residuals is estimated by

$$\sigma^2 = \frac{2}{m} \min_{\hat{\theta}} V_n(\hat{\theta}) \qquad (3.1214)$$

Let $\hat{\theta}_n$ be the vector which minimizes $V_n(\hat{\theta})$. The estimate $\hat{\theta}_{m+1}$ is calculated from a Taylor expansion of $V_{m+1}(\hat{\theta})$ around $\hat{\theta}_n$. The expansion including second order terms is assumed accurate enough for the calculation of $\hat{\theta}_{n+1}$

$$V_{n+1}(\hat{\theta}) = V_{n+1}(\hat{\theta}_n) + V'_{n+1}(\hat{\theta}_n)[\hat{\theta} - \hat{\theta}_n] +$$
$$1/2\ [\hat{\theta} - \hat{\theta}_n]^t V''_{n+1}(\hat{\theta}_n)[\hat{\theta} - \hat{\theta}_n] \qquad (3.215)$$

Using the method of iteration [30] for $V'(\hat{\theta}_n)=0$ to produce the recursive equation for $\hat{\theta}_{n+1}$ gives

$$\hat{\theta}_{n+1} = \hat{\theta}_n - V''_{n+1}(\hat{\theta}_n)^{-1} V'_{n+1}(\hat{\theta}_n)^t \qquad (3.216)$$

which is the same form as the first iteration of a Newton-Raphson algorithm applied to the equation $V'_{n+1}(\hat{\theta})=0$.

Substituting $\hat{\theta}_{n+1}$ of equation (3.206) for $\hat{\theta}$ in equation (3.205) allows the estimated minimum value of $V_{n+1}(\hat{\theta})$ to be expressed as

$$V_{n+1}(\hat{\theta}) = V_{n+1}(\hat{\theta}_n) - 1/2 V'_{n+1}(\hat{\theta}) -$$
$$V''_{n+1}(\hat{\theta})^{-1} V'_{n+1}(\hat{\theta}_n)^t \qquad (3.217)$$

In order to form a recursive estimator, the relationship between $V_{n+1}(\hat{\theta})$ and $V_n(\hat{\theta})$ must be used. Since by definition

$$V_{n+1}(\hat{\theta}) = V_n(\hat{\theta}) + 1/2e^2(n+1|\hat{\theta}) \qquad (3.218)$$

taking the first and second derivatives

$$V'_{n+1}(\hat{\theta}) = V_n(\hat{\theta}) + e(n+1|\hat{\theta})\, e'(n+1|\hat{\theta}) \qquad (3.219)$$

$$V''_{n+1}(\hat{\theta}) = V''_n(\hat{\theta}) + e'(n+1|\hat{\theta})'e'(n+1|\hat{\theta}) + e(n+1|\hat{\theta})e''(n+1|\hat{\theta}) \qquad (3.220)$$

gives recursive equations for the terms of equation (3.207). To simplify these recursive equations the following three assumptions are made

i. $V'_n(\hat{\theta}_n) = 0$ (3.221)

since $\hat{\theta}_n$ is assumed to minimize $V_n(\hat{\theta}_n)$.

ii. $e(n+1|\hat{\theta}_n)e''(n+1|\hat{\theta}_n) = 0$ (3.222)

as it is assumed for off-line maximum likelihood estimation the term $\Sigma^n_{k=1}\{e(k|\hat{\theta})e''(n+1|\hat{\theta})\}$ of equation (3.200) has little influence on the minimization and can be set to be zero.

iii.

$V''_n(\hat{\theta}_n) = V''_n(\hat{\theta}_{n-1})$ (3.223)

which holds approximately if $\hat{\theta}_n$ is close to $\hat{\theta}_{n+1}$.

Using the approximations of equations (3.221) to (3.223), the recursive equations become

$$V_{n+1}(\hat{\theta}_{n+1}) = V_n(\hat{\theta}_n) + \frac{1}{2} e^2(N+1|\hat{\theta}_n) - \frac{1}{2} V'_{n+1}(\hat{\theta}_n)V'_{n+1}(\hat{\theta}_n)'V''_{n+1}(\hat{\theta}_n)^{-1} \qquad (3.224)$$

$$V'_{n+1}(\hat{\theta}_{n+1}) = e(n+1|\hat{\theta}_n)e'(n+1|\hat{\theta}_n) \qquad (3.225)$$

$$V''_{n+1}(\hat{\theta}_{n+1}) = V''_n(\hat{\theta}_{n-1}) + e'(n+1|\hat{\theta}_n)'e(n+1|\hat{\theta}_n) \qquad (3.226)$$

Introducing the following notation

$$Kg_n = V''_n(\hat{\theta}_{n-1})^{-1} \qquad (3.227)$$

$$\zeta_n = e'(n|\hat{\theta}_{n-1})^t \tag{3.228}$$

$$e_n = e(n|\hat{\theta}_{n-1}) \tag{3.229}$$

$$\gamma_{n+1} = 1 + \zeta_{n+1}^t Kg_n \zeta_{n+1} \tag{3.230}$$

where Kg_n is called the gain, ζ_n is called the gradient and e_n the error, allows the minimization equation, equation (3.216), to be written as

$$\hat{\theta}_{n+1} = \hat{\theta}_n - Kg_{n+1}\zeta_{n+1}e_{n+1} \tag{3.231}$$

Applying the matrix inversion lemma used previously (cf Section 3.2.2, equation (3.45)) to equation (3.226) gives

$$Kg_{n+1} = Kg_n - \frac{Kg_n \zeta_{n+1} \zeta_{n+1}{}^t Kg_n}{\gamma_{k+1}} \tag{3.232}$$

Finally equation (3.224) can be written as

$$V_{n+1}(\hat{\theta}_n) = V_n(\hat{\theta}_n) + \frac{1}{2} \frac{1}{\gamma_{n+1}} e_{n+1}^2 \tag{3.233}$$

As in the off-line maximum likelihood derivation all that remains is the development of recursive equations for e_n and ζ_n. These recursive equations will be the same equations that apply for the off-line method with one exception. The model equation will now be

$$\hat{F}(z^{-1})y_k = \hat{G}(z^{-1})u_k + \hat{H}(z^{-1})e(k|\hat{\theta}) \tag{3.234}$$

where $\hat{F}(z^{-1})y_k$ is on the opposite side of the model equation so the gradient of f_i will be of the opposite sign to that in equation (3.201).

The recursive maximum likelihood algorithm can now be summarized. For initial values of e(0)=0 and e'(0,θ)=0 the algorithm is

$$\zeta_{k+1/f_i} = \Sigma_{j=1}^{n} h_j \zeta_{k/f_i} + y_k \tag{3.235}$$

$$\zeta_{k+1/g_i} = \Sigma^{n}_{j=1} h_j \zeta_{k/g_i} - u_k \qquad (3.326)$$

$$\zeta_{k+1/h_i} = \Sigma^{n}_{j=1} h_j \zeta_{k/h_i} - e_k \qquad (3.327)$$

$$\gamma_{k+1} = 1 + \zeta_{k+1}{}^{t} Kg_k \zeta_{k+1} \qquad (3.230)$$

$$Kg_{n+1} = Kg_n - \frac{Kg_n \zeta_{n+1} \zeta_{n+1}{}^{t} Kg_n}{\gamma_{k+1}} \qquad (3.232)$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k - Kg_{k+1} \zeta_{k+1} e_{k+1} \qquad (3.231)$$

where n_f, n_g, and n_h are the number of f, g and h parameters, respectively. The gain, Kg, may be chosen as simply a positive scalar constant or selected as the recursive least squares gain. This gain is obtained by formally considering

$$\hat{\theta}_k = \hat{\theta}_{k-1} - Kg_k e_k \zeta_k \qquad (3.238)$$

as the parameter updating equation in the standard recursive least squares algorithm.

4. Linear Simulation Results

4.1 Introduction

The objective of this chapter is to evaluate different parameter estimation techniques when used to estimate controller parameters for the self-tuning controller. It is desirable to have an estimator that produces parameters which give good control with minimum control action.

The five estimation methods were tested to determine which technique estimated the parameters most accurately and efficiently. The estimator performance was evaluated by observing the control performance of the self-tuning controller with respect to its ability to track setpoints, reject load disturbances, and to observe the effects of changing certain estimator variables such as the initial covariance matrix, forgetting factors, initial parameter estimates and choice of initial g_0 value. Cost function weighting was also used in order to obtain better control with smoother control action.

The estimation algorithms were quantitatively evaluated using three criteria: the sum of the prediction errors, the control effort and the accuracy of the final parameter estimates. The sum of the prediction errors was calculated by adding the absolute value of the difference between actual output and the setpoint at each sample instant. The control effort is the sum of the absolute values of the difference between the actual control action and the steady state input which is calculated by multiplying the steady state gain by the current setpoint value.

The results will be presented with certain trends discussed and characteristics indicated for the specific cases but the overall assessment of the results will be given at the end of each section.

The estimation algorithms were first tested for the control of a third-order, linear system without any time delay as used by Morris et al [31] in previous studies. The system in discrete form is expressed as

$$y_k=1.69y_{k-1}-1.08y_{k-2}+.33y_{k-3}+.34u_{k-1}$$
$$-.07u_{k-2}-.22u_{k-3}+\xi_k-.79\xi_{k-1}+.37\xi_{k-2} \quad (4.1)$$

In order to obtain a unique solution for the controller parameters it is assumed that $h_0 = -1.0$ thus it is not included in the parameters to be identified so the identification algorithm must estimate eight parameters for this third order model. The correct values for these parameters are $f_0=0.9$, $f_1=-0.71$, $f_2=0.33$, $g_0=0.34$, $g_1=-0.07$, $g_2=-0.22$, $h_1=0.79$ and $h_2=-0.37$.

The control action amplitude was limited to ±25 to keep the input within reasonable values or limited to 100% of the previous input value to keep the variations of the input moderate. This is to simulate actual conditions where there would be certain physical restrictions on the manipulated variable.

4.2 Setpoint Tracking

Three setpoint waves were used to evaluate the tracking ability of the self-tuning controller with each of the estimation methods, a square wave, a sawtooth wave and a step change in setpoint. The square wave was used to observe the response to a succession of abrupt changes, the sawtooth wave to determine the ability to adapt to rapidly changing setpoint values, and the single step change in setpoint to determine the controller's ability to track for a nonexciting setpoint.

Table 4.1 summarizes the linear simulation results for the five estimation methods used for the minimum variance controller, $P=R=1$, $Q=0$. In all simulations the initial covariance matrix, $S(0)$, forgetting factor, ρ, initial parameter estimates, $\hat{\theta}(0)$ and variance of the noise added, σ^2, were unchanged. The effects of changing these parameters will be dealt with later. The initial covariance matrix value is $3000\underline{\underline{I}}$, the forgetting factor is 0.995, the variance of the noise is 0.04. The initial parameter estimates except for g_0 are set to 0.0, g_0 is given a non-zero value of 0.3. The initial guess for g_0 was chosen close to the actual value to ensure parameter convergence within a reasonable amount of time.

The square wave has a period of 50 sample instants, the setpoint is 5.0 for the first 25 sample instants and is 0.0 for the following 25 instants and then the pattern is repeated. The sawtooth function has a period of 100 sample

Table 4.1 Summary of Setpoint Tracking Results for Different Identification Methods and Setpoint Changes

ESTIMATION METHOD	SETPOINT TYPE	CONTROL EFFORT	SUM OF PREDICTED ERRORS	
RLS	SQW	580.34	96.21	Figure 4.1
RLS	SAW	319.29	48.38	Figure 4.2
RLS	STS	342.27	59.49	Figure 4.3
RSR	SQW	565.10	95.19	Figure 4.8
RSR	SAW	317.37	48.38	Figure 4.9
RSR	STS	324.29	58.82	Figure 4.10
RUD	SQW	565.10	95.19	Figure 4.14
RUD	SAW	317.36	48.38	Figure 4.15
RUD	STS	324.29	58.82	Figure 4.16
RL	SQW	606.00	166.06	Figure 4.20
RL	SAW	136.61	43.94	Figure 4.21
RL	STS	144.33	63.87	Figure 4.22
RML	SQW	603.86	184.86	Figure 4.27
RML	SAW	185.44	144.44	Figure 4.28
RML	STS	546.84	392.67	Figure 4.29

RLS - Recursive Least Squares
RSR - Recursive Square Root
RUD - Recursive Upper Diagonal Factorization
RL - Recursive Learning
RML - Recursive Maximum Likelihood

SQW - Square Wave Setpoint
SAW - Sawtooth Function Setpoint
STS - Step Change in Setpoint

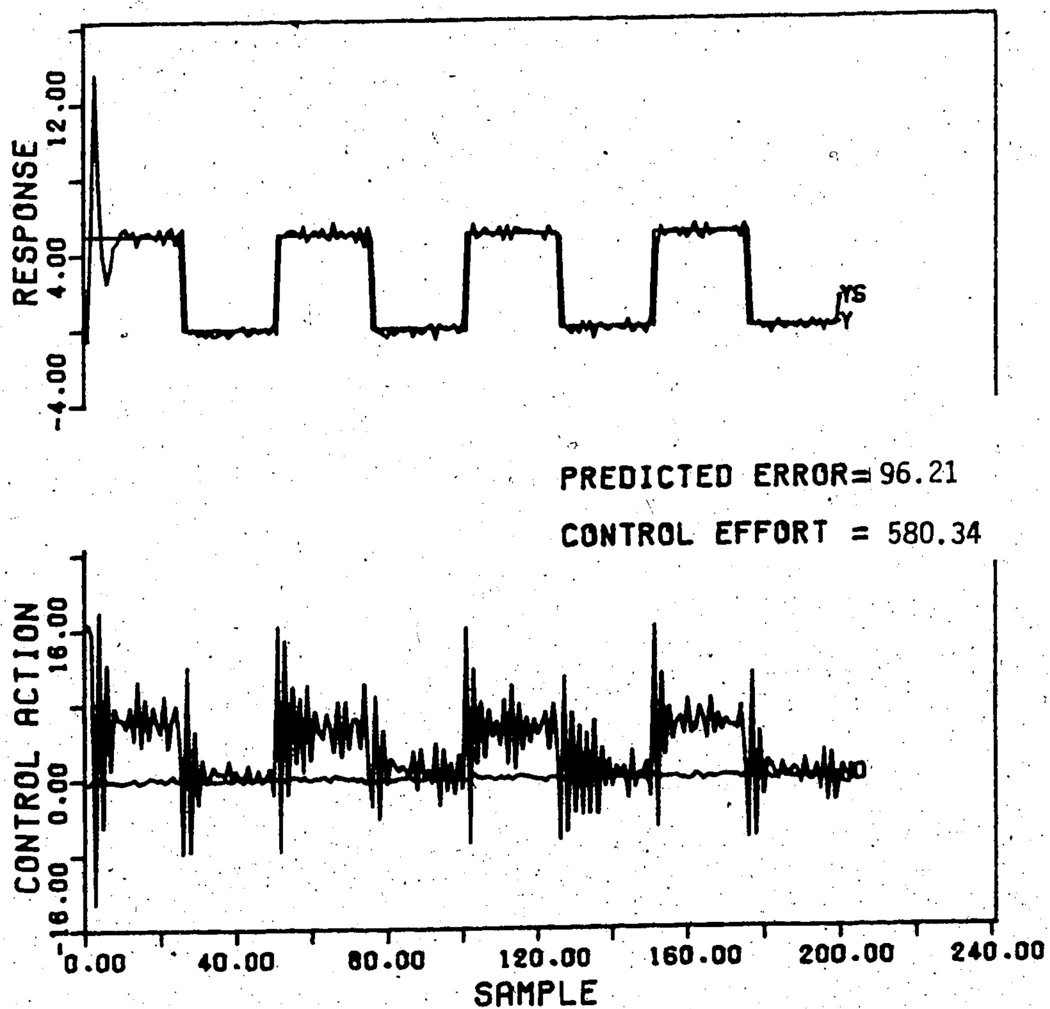
instants. At the start of the period the setpoint is 0.0 and the setpoint is increased by 0.1 at each instant until a setpoint of 5.0 is attained and then the setpoint is decreased by 0.1 until a setpoint of 0.1 is reached. For both the square wave and sawtooth function simulations the setpoint is specified by repeating the periods over the duration of the simulation. The step change in setpoint is a change from 0.0 to 5.0 which is fixed for the entire simulation. In each of the plots the setpoint will be indicated by the symbol YS and NO is the random noise added to the system.

The input to the system or control action is the manipulated variable and is given an initial value of 0.0. The output is the controlled variable and will be required to follow a desired setpoint.

4.2.1 Recursive Least Squares Estimator

From Figures 4.1, 4.2 and 4.3 it can be seen that for minimum variance control, the recursive least squares estimator tracks well for all three setpoint types after 10 sampling instants. The greatest control effort and sum of prediction errors is obtained with the square wave setpoint change which is expected as the setpoint changes are larger than for the other two cases.

Figures 4.4, 4.5 and 4.6 illustrate the behavior of the estimates for the three setpoint waves. The F and G estimates make the biggest initial changes for the square



PREDICTED ERROR= 96.21

CONTROL EFFORT = 580.34

Figure 4.1 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Least Squares Estimator

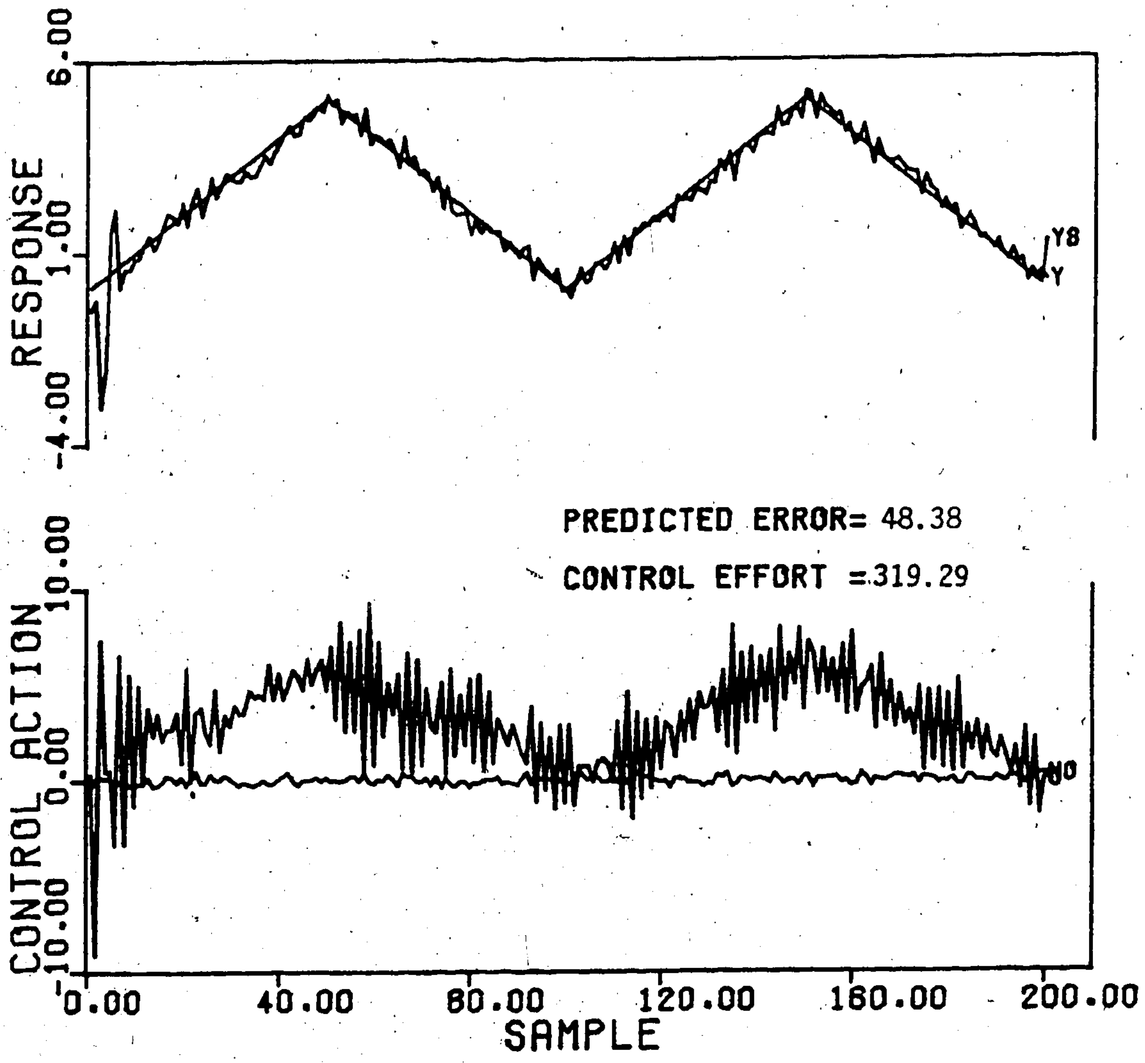


Figure 4.2 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator

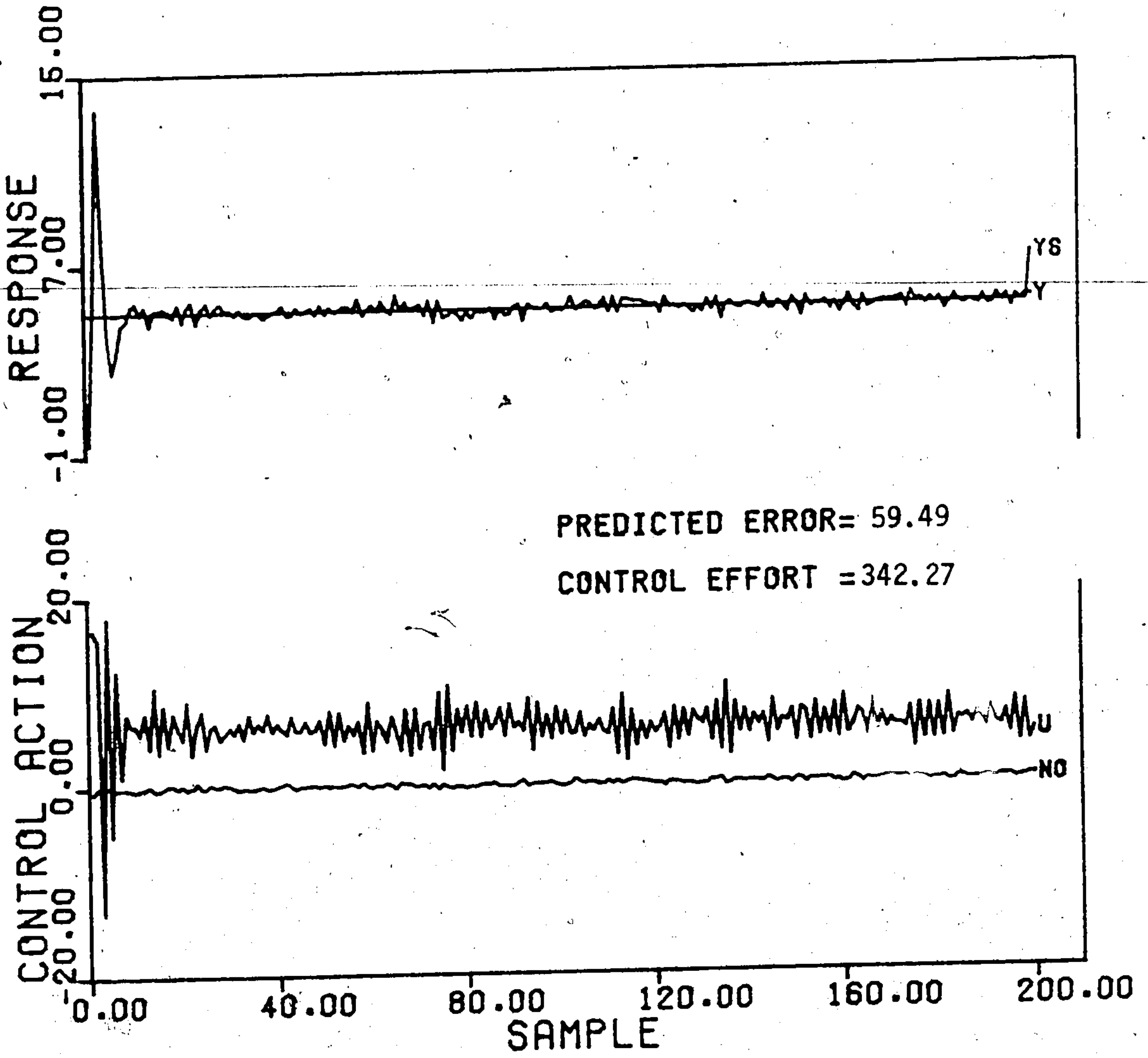


Figure 4.3 Setpoint Tracking a Step Change in Setpoint using the Recursive Least Squares Estimator

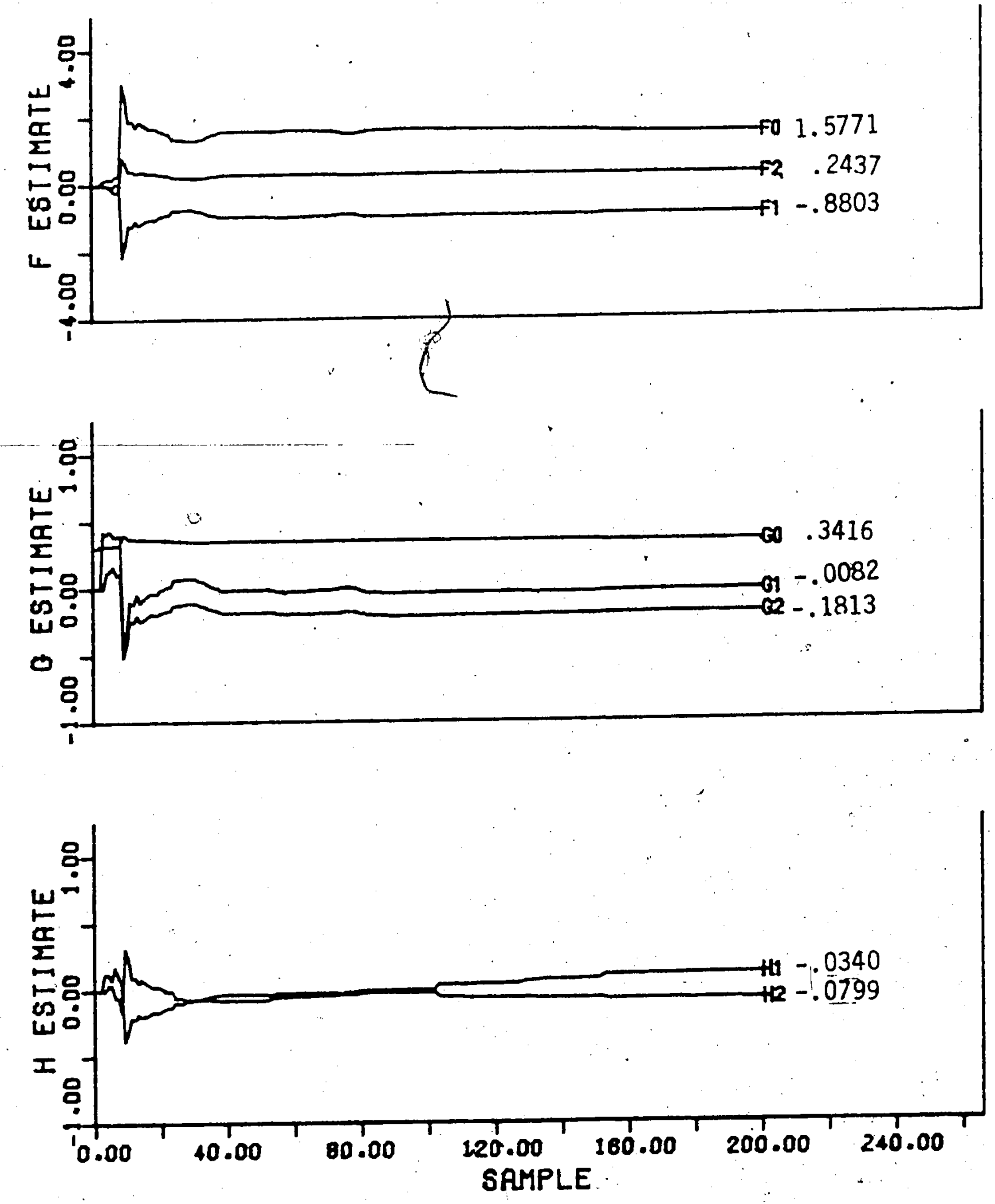


Figure 4.4 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Least Squares Estimator

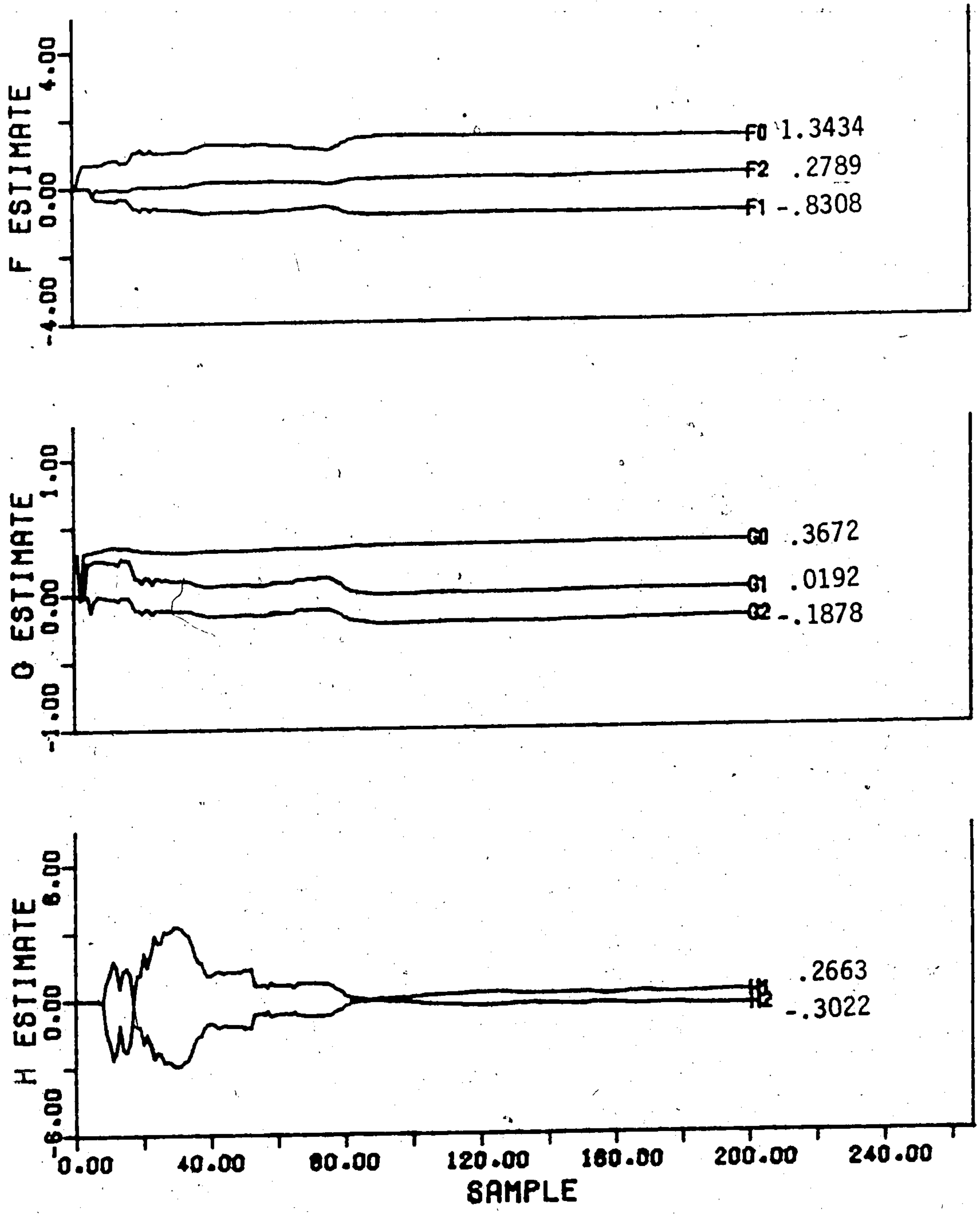


Figure 4.5 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator

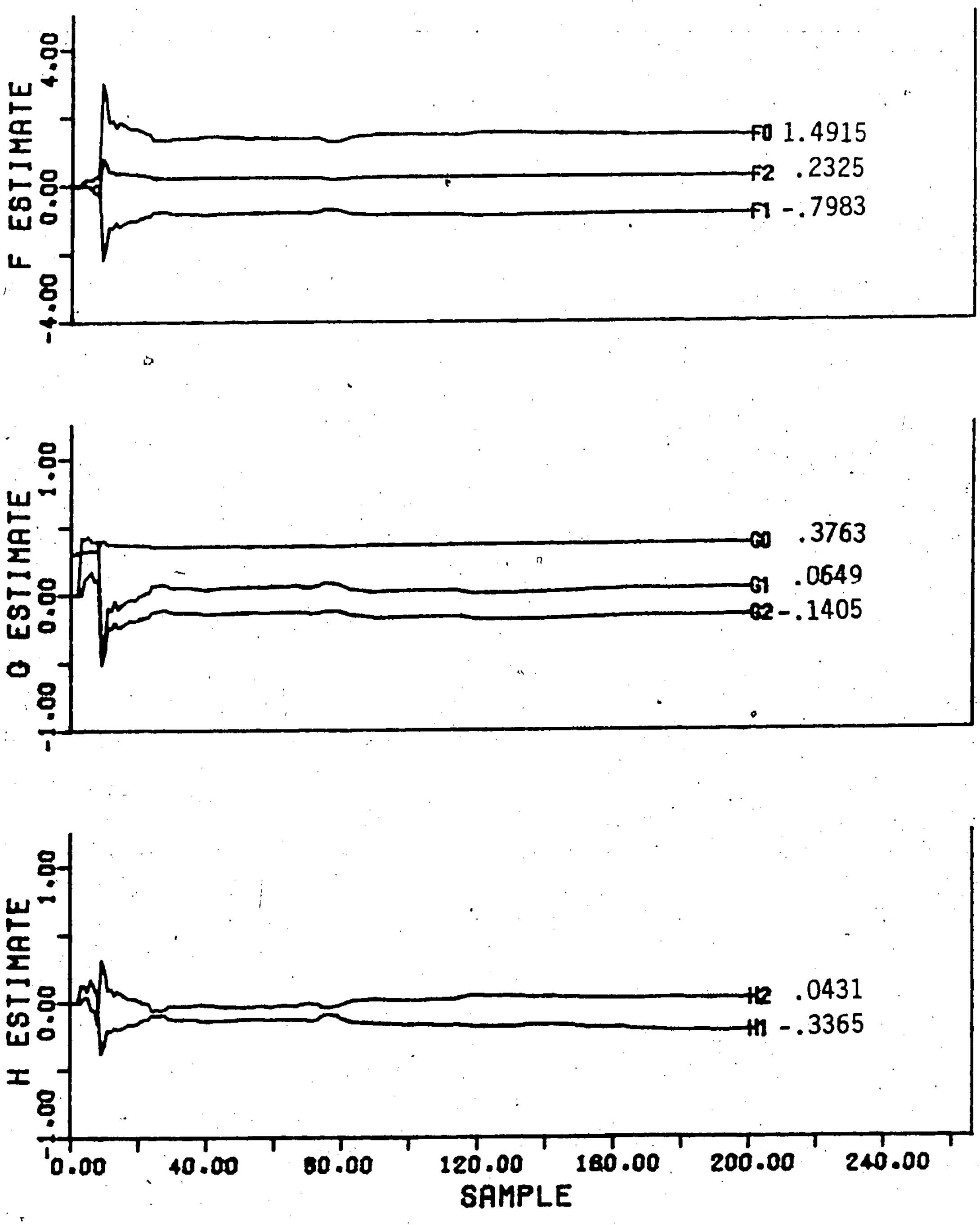


Figure 4.6 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Least Squares Estimator

wave change in setpoint and step change in setpoint which is caused by the large initial jump in the setpoint from 0.0 to 5.0 For all cases the F and G estimates are constant and converged to a nearby value by the 100th sample instant. The G parameter adaption pattern for the sawtooth change in setpoint is initially erratic but the estimates settle out by the 80th sample instant. The H parameters remain close to zero [illegible] or the sawtooth change the estimates increase drastically at the 10th sample instant, then change sign and increase to an even larger magnitude before changing sign again and converging toward the true parameter values. The H parameter estimates for the square and sawtooth waves are mirror images of each other but for the step change in setpoint the estimates are diverging from their true values as both estimates are opposite in sign to the true values. The H estimates begin to grow further from the true values by the 500th sample instant and have not converged even by the 3000th sample instant.

As it can be seen, after 200 sample instants none of the parameter estimates have reached the true values however, by the 1000th instant the estimates are within on average of absolute errors of ±4.1% of their true values for the sawtooth wave. For the square wave setpoint change the estimates are only within an average of absolute errors of ±22% of the true values with the f_0, f_1, h_1 and h_2 parameters at least ±25% from the true values. The f_2, g_0, g_1 and g_2 estimates are within ±10% of the true values. For the constant setpoint the parameter estimates continue to increase significantly and eventually will diverge or blow up as seen in Figure 4.7 and control deteriorates by the 2970th sample instant. The estimation becomes ill-conditioned because the forgetting factor inflates the covariance matrix which in turn causes the parameters to grow. A simulation with the same conditions as above was done except that the forgetting factor was 1.0. After 3000 sample instants there was no parameter blow up the the parameter estimates were increasing.

The accuracy of the estimates using the recursive least squares estimator is impaired by the order of the C polynomial in the system equation. When the coefficients other than c_0 of the C polynomial in equation 2.1 are not zero the recursive least squares estimator will give biased estimates since ϵ_k, given by

$$\epsilon_k = \xi_k + \Sigma_{i=1}^{n} c_i \xi_{k-i} \tag{4.2}$$

is correlated with y_k. This bias is substantial when the signal-to-noise ratio is low (<10/1) but is not a factor for high signal-to-noise ratios.[10] The signal-to-noise ratio in the simulations, calculated as the ratio of the absolute magnitude of the input signal to the absolute magnitude of the noise (equal to 1.), ranges from 5 to 20. Therefore the bias will affect some of the estimates obtained using the recursive least squares estimator.

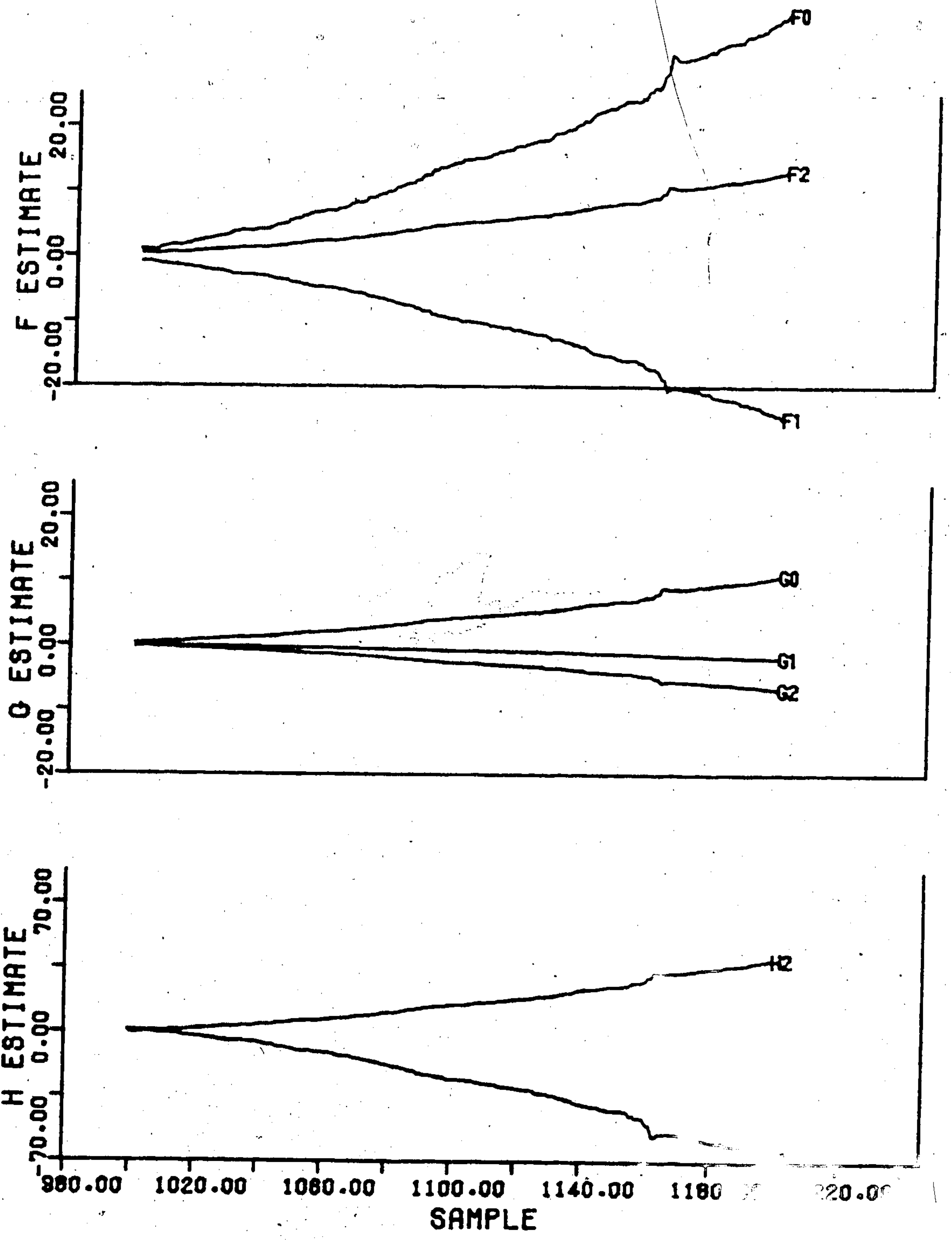


Figure 4.7 Parameter Estimates for a Step Change in Setpoint using the Recursive Least Squares Estimator

4.2.2 Recursive Square Root Estimator

The control performance displayed in Figures 4.8, 4.9 and 4.10 demonstrate that the parameter estimates obtained using the recursive square root estimator also provide good setpoint tracking for the three different types of setpoint changes. The performance is almost identical to that achieved using the recursive least squares estimator, with the trend of the sum of the prediction errors and control effort nearly the same but slightly lower values for the recursive square root tests indicating slight improvement in the control performance.

The parameter estimates for the square wave setpoint change, plotted in Figure 4.11, particularly the F and G estimates are initially the same as the values from the recursive least squares test but by the 200th sample instant the recursive square root parameters overall are within ±6% of the true values compared with the ±22% for the recursive least squares estimates. Comparison of Figure 4.4 with Figure 4.11 reveals that the h_1 and h_2 estimates are converging toward the true values more quickly than resulted using the recursive least squares identification technique. The initial h estimate behavior is only the same as for recursive least squares identification until the 20th sample instant.

For the sawtooth function change in setpoint all three sets of parameter estimates behave exactly as for the recursive least squares simulation ~~as~~ can be seen from the

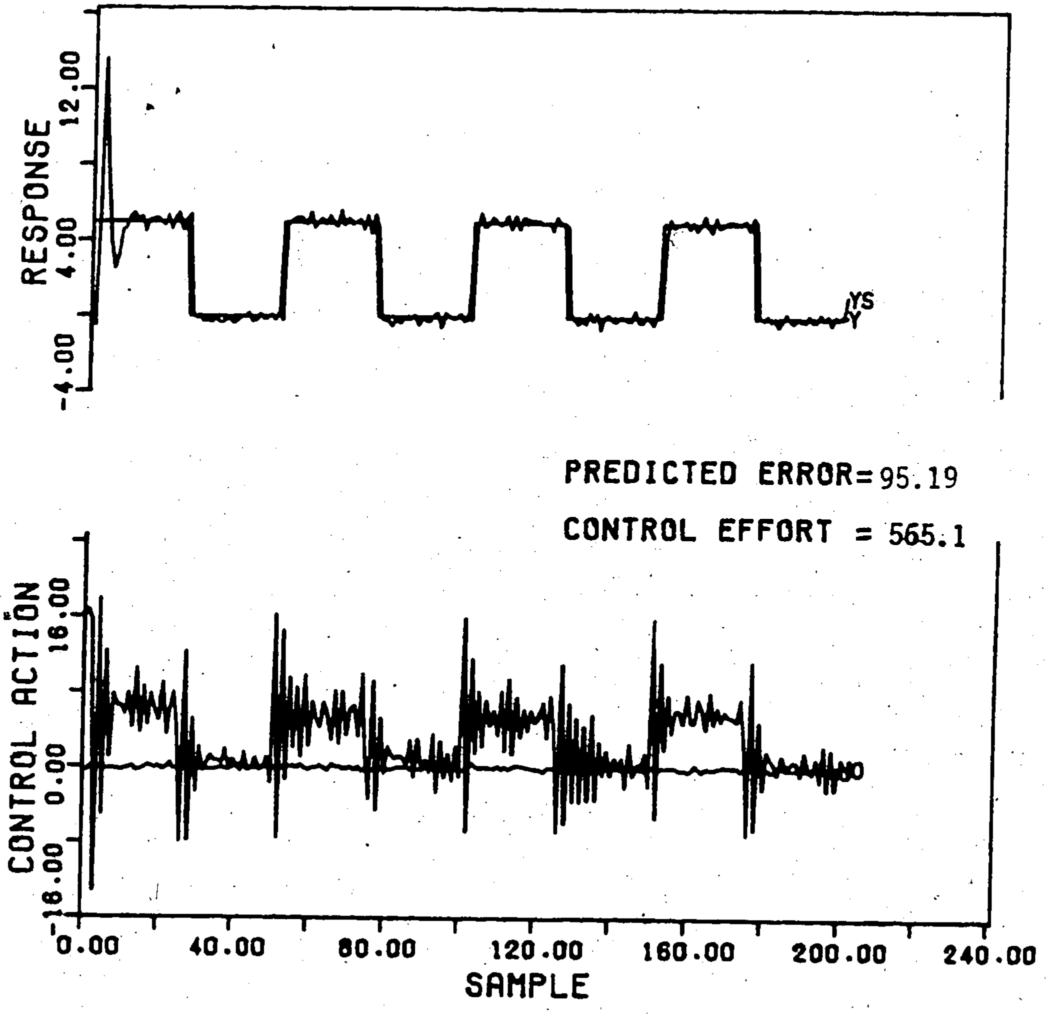


Figure 4.8 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Square Root Estimator

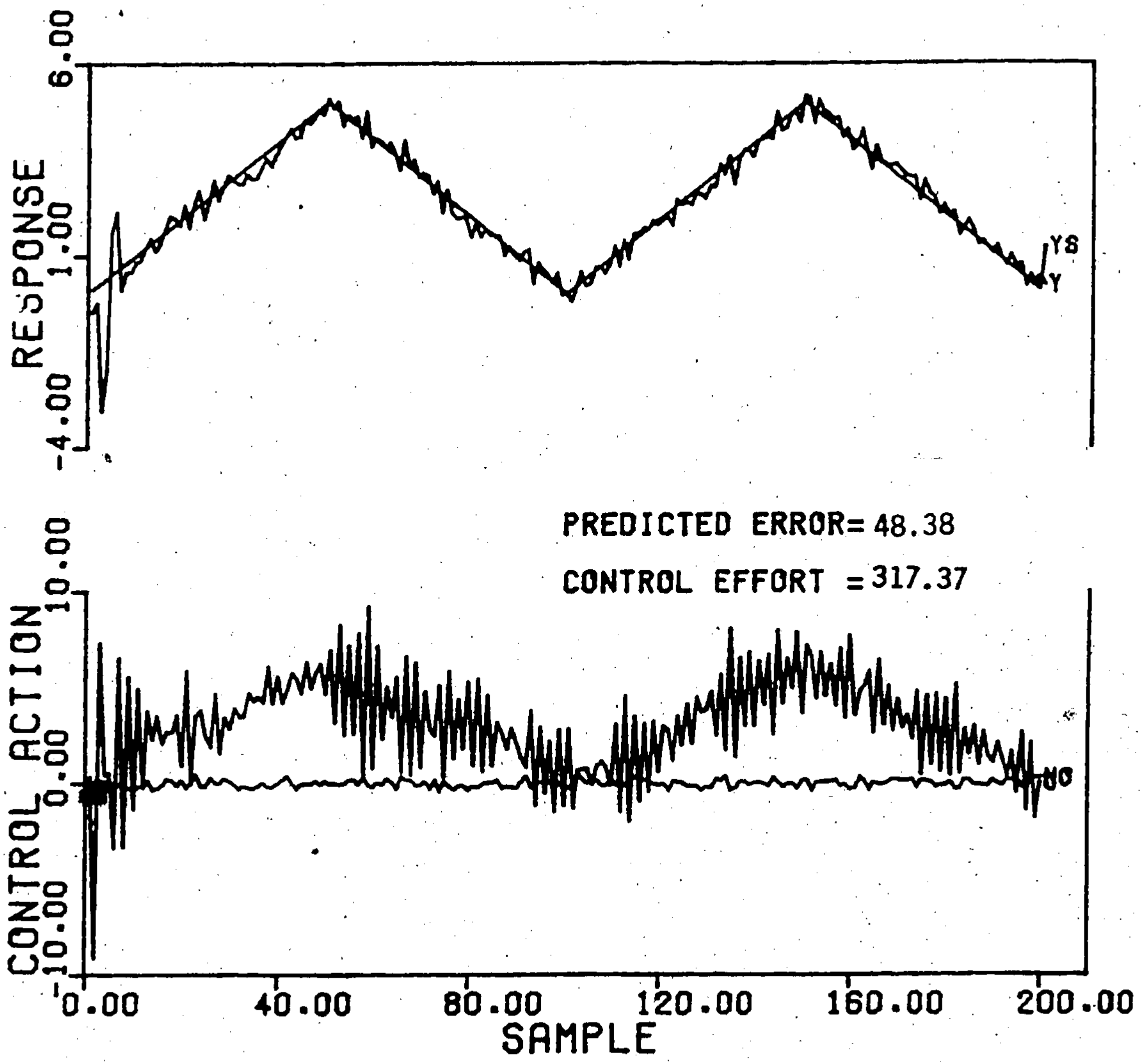


Figure 4.9 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator

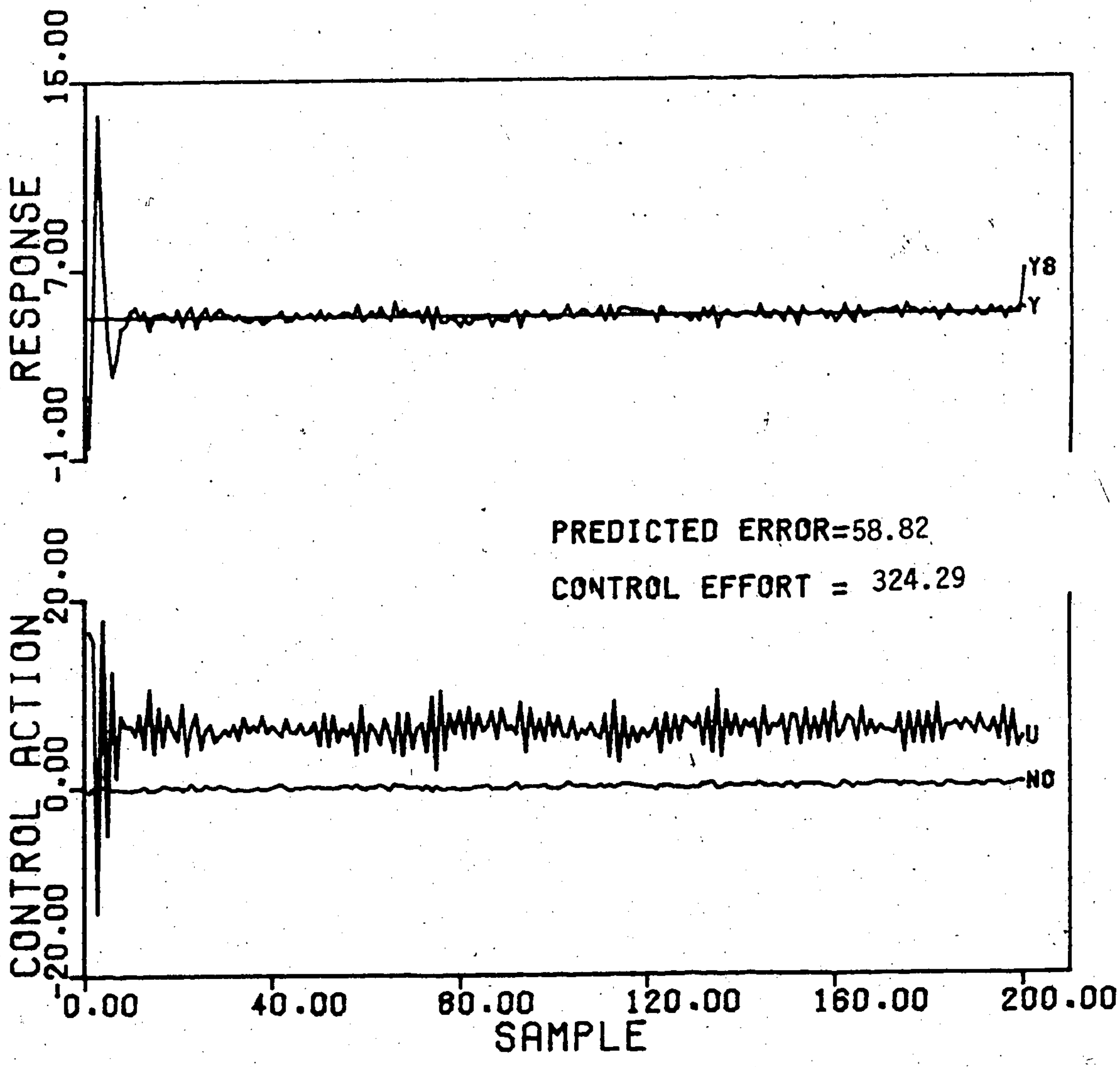


Figure 4.10 Setpoint Tracking a Step Change in Setpoint using the Recursive Square Root Estimator

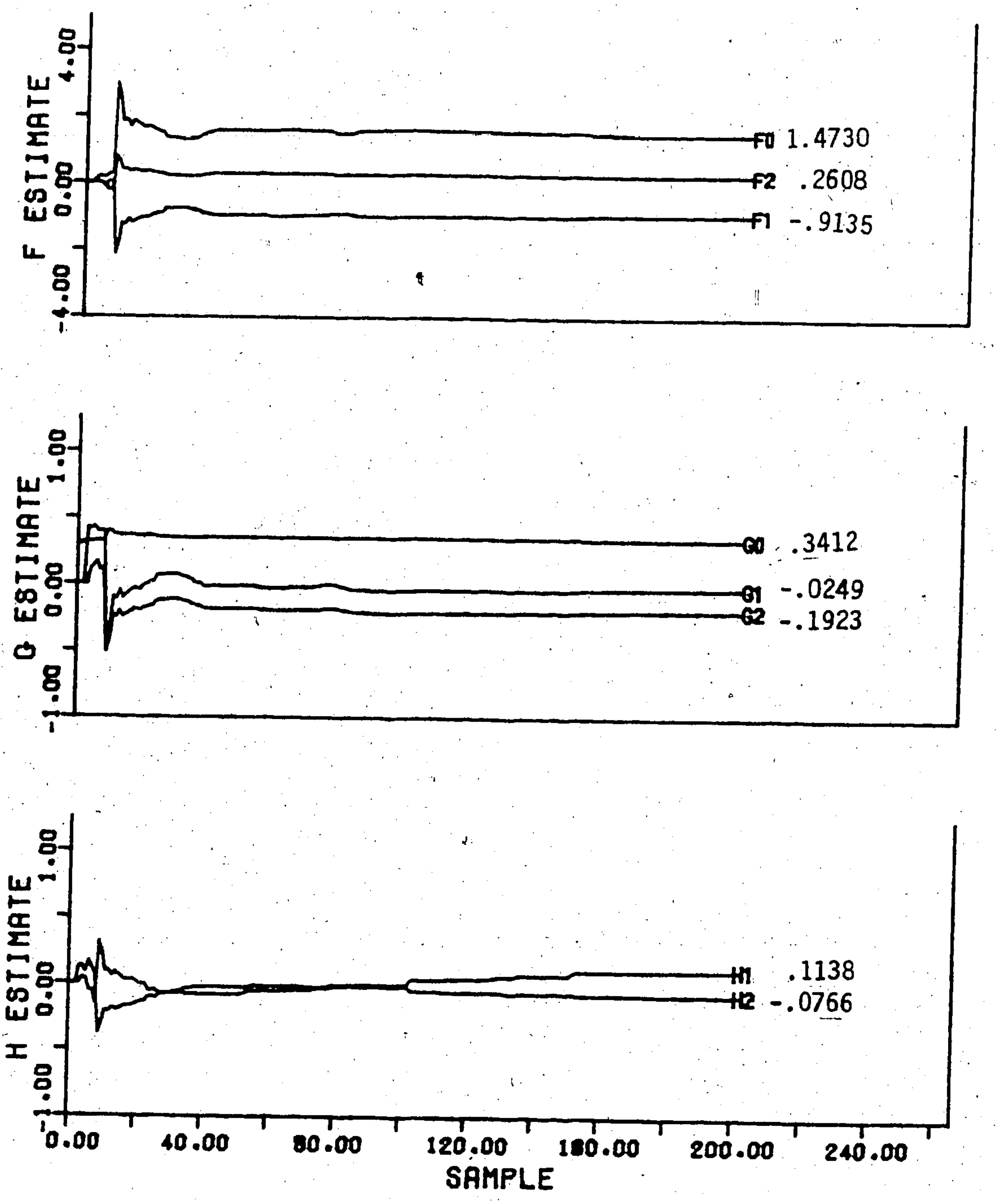


Figure 4.1.1 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Square Root Estimator

results in Figure 4.12. For the step change in setpoint simulation, the F and G parameter estimates as can be seen from Figure 4.13 are the same as for recursive least squares case but the h_1 and h_2 parameter estimates are different. The estimates are not diverging as fast as the recursive least squares case and the h_2 estimate is beginning to tend towards its correct sign. The H parameter estimates are also more symmetrical to each other using recursive square root identification than when using the recursive least squares technique.

By the 1000th sample instant the recursive square root parameter estimates are within 5-10% of the true values for the sawtooth function change and the step change in setpoint, while only the G parameter estimates were within 5% of the correct values for the square wave setpoint change. The larger errors for the square wave change occur for the H parameter estimates which cause the F parameter estimates to have large errors. (cf Section 4.6) No reason for the unsatisfactory parameter estimates was determined.

4.2.3 Recursive Upper Diagonal Factorization Estimator

The simulation results showing the control performance using the recursive upper diagonal factorization estimator, for the three setpoint changes are given in Figures 4.14, 4.15 and 4.16. The responses are almost identical to those obtained using the recursive square root identification method with the difference in the sum of the prediction

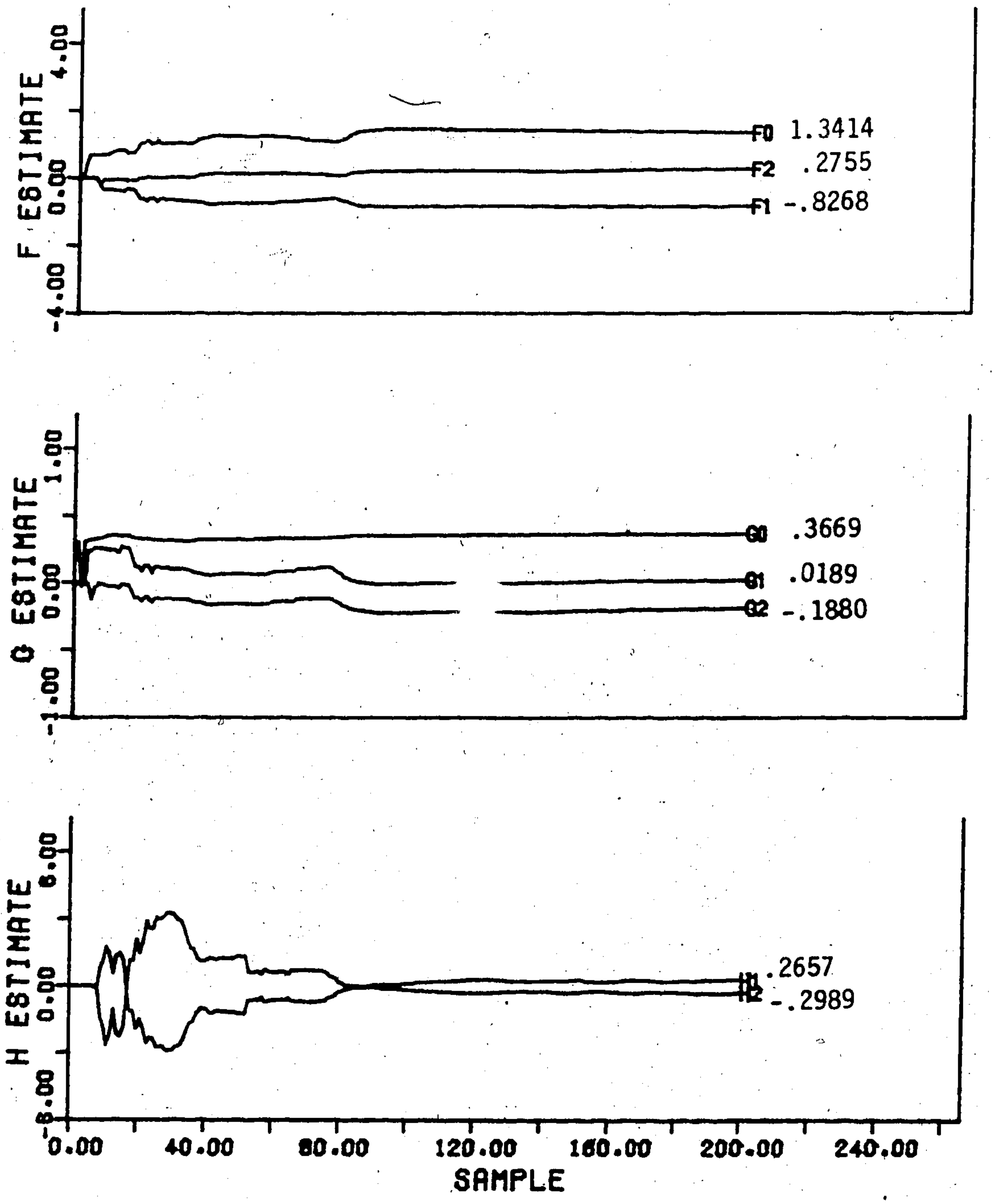


Figure 4.12 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator

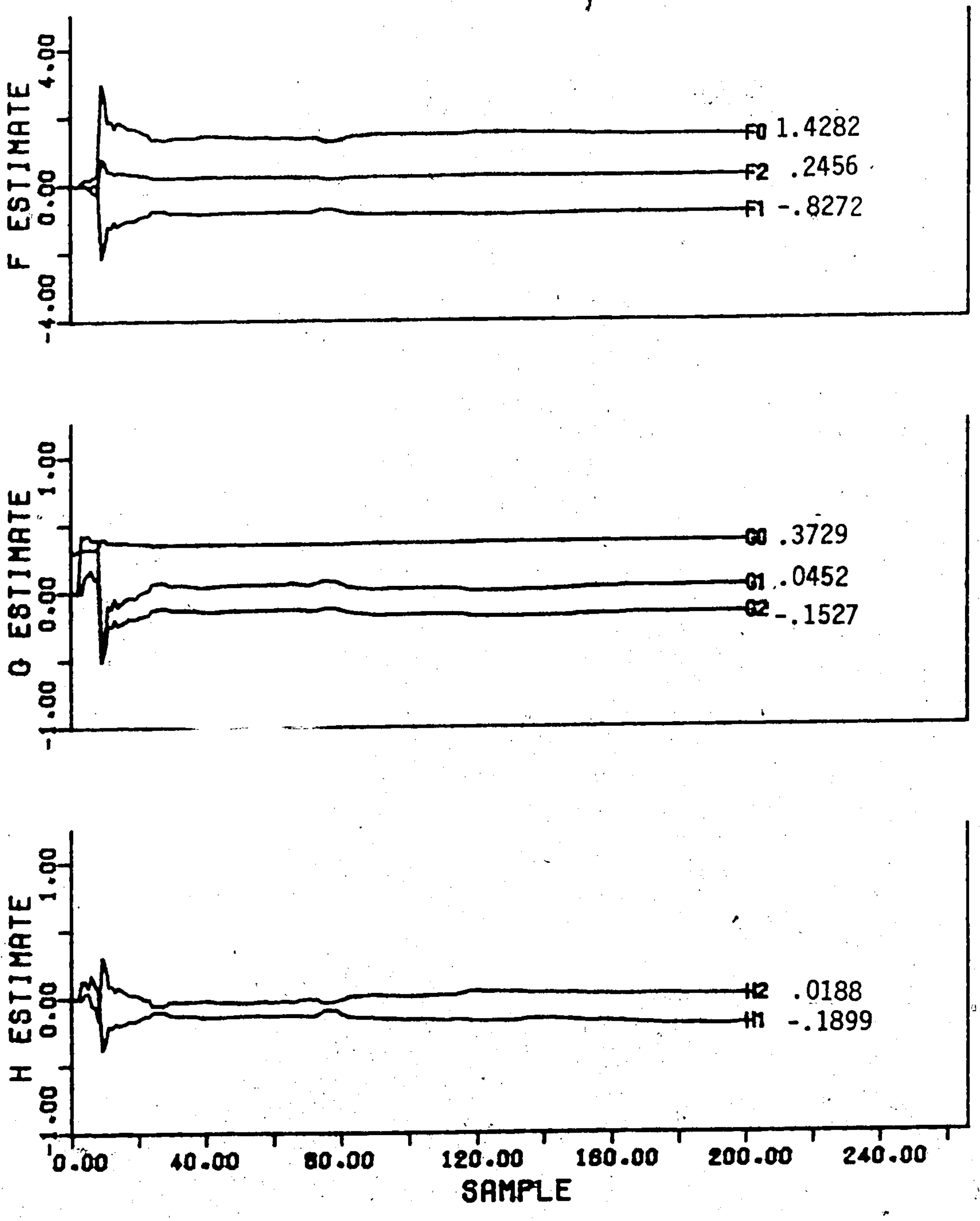


Figure 4.13 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Square Root Estimator

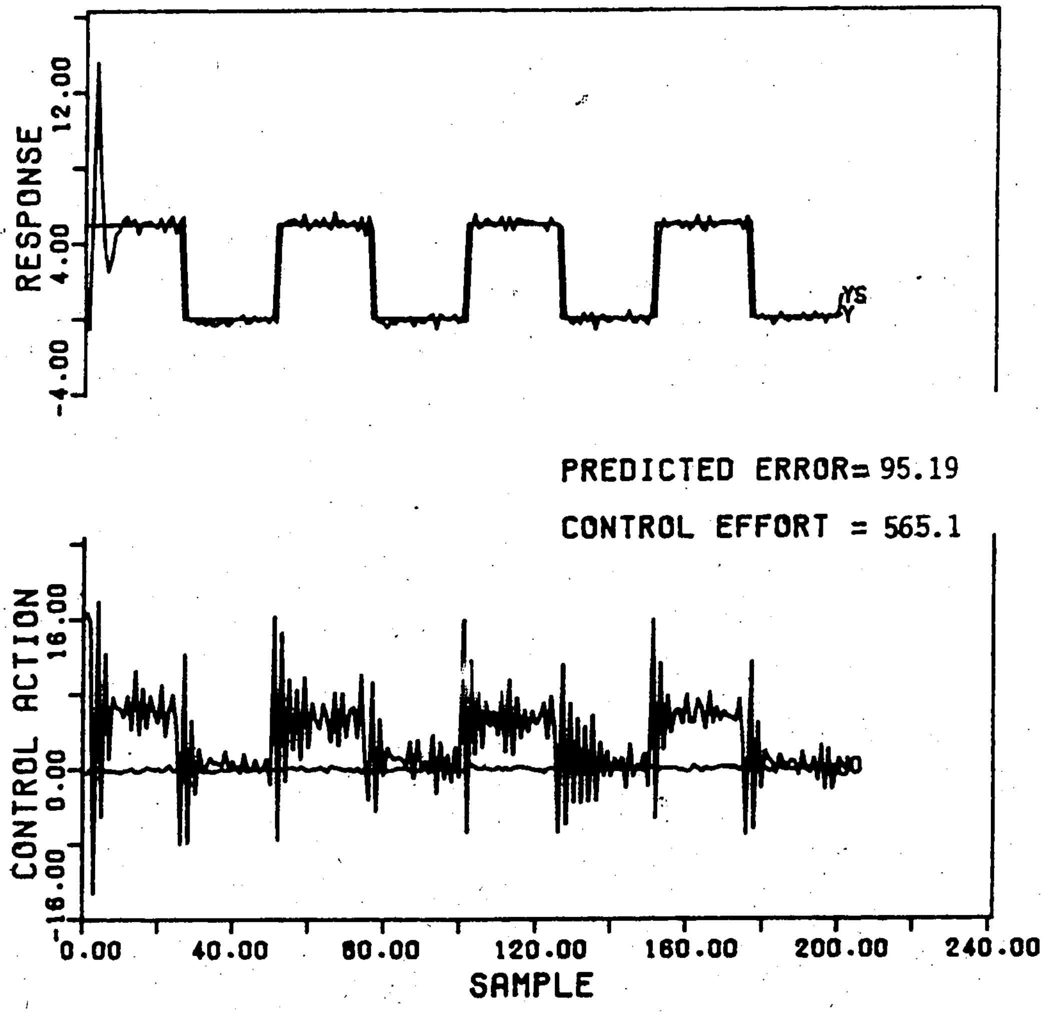


Figure 4.14 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

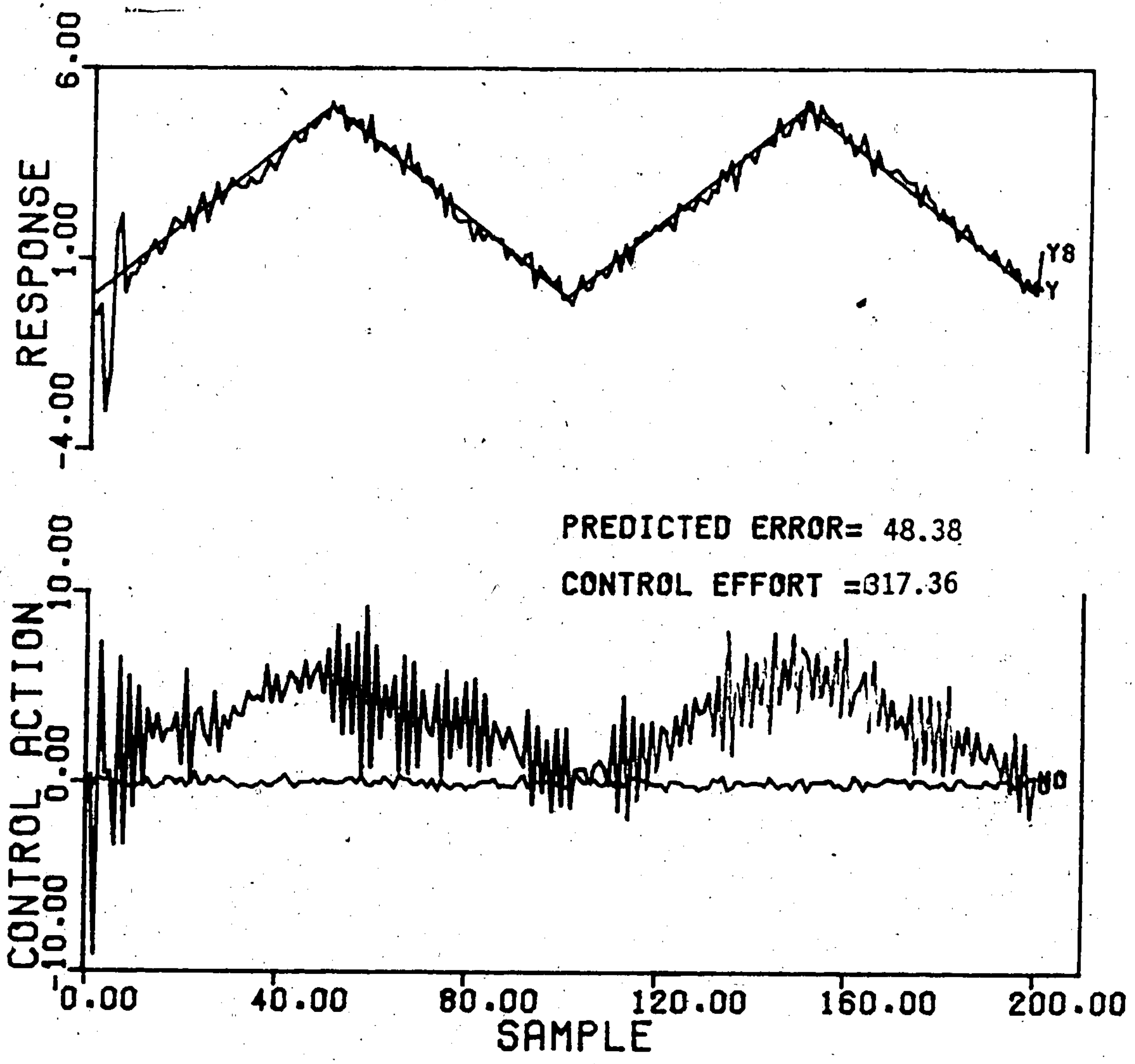


Figure 4.15 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

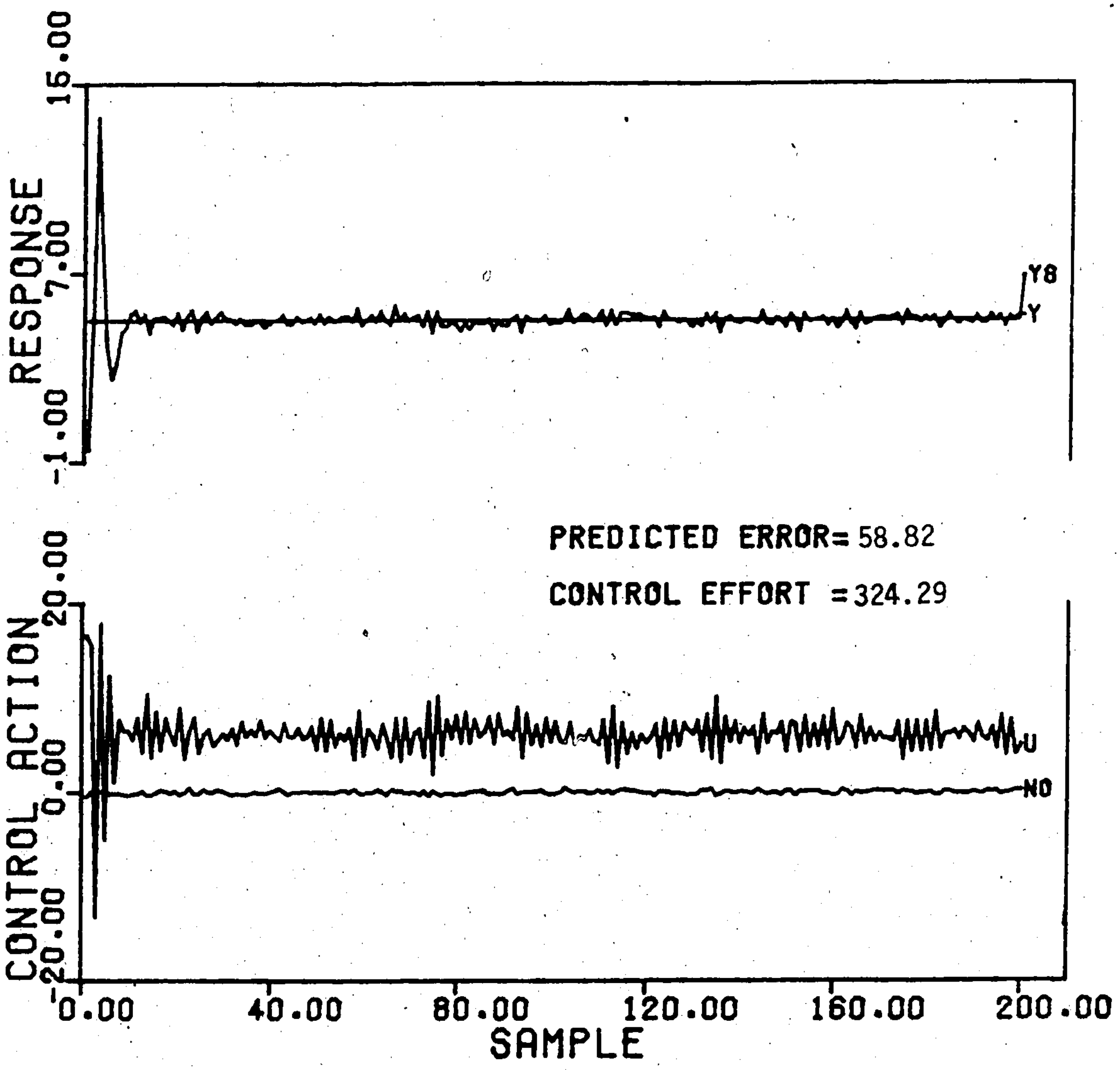


Figure 4.16 Setpoint Tracking a Step Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

errors and control effort values insignificant with respect to the magnitude of the values themselves.

Furthermore, comparison of the parameter adaption behavior displayed in Figures 4.17, 4.18 and 4.19 with that in Figures 4.11, 4.12 and 4.13 shows that the behavior is observed using the same as the recursive square root identification technique. After 200 sample instants the two estimation algorithms yield parameter estimates that are within 10^{-3} of each other and by the 1000th sample instant the estimates from the sawtooth function and square wave setpoint changes tests are within 10^{-4}, but the estimates for the step change in setpoint have errors of ±0.3% more than the parameter estimates yielded by the recursive square root simulations which are ±5% from the true values.

The recursive square root and recursive upper diagonal factorization estimators give similar results because the recursions derived for U and D are equivalent to the matrix square root, $S^{1/2}=UD^{1/2}$. The algorithms are almost identical except that the recursive square root estimator requires n scalar square root and n extra divisions, where n is the number of parameters to be identified for each observation to be processed.[23] The differences between the estimates is the result of the method in which the noise is processed by the estimator.

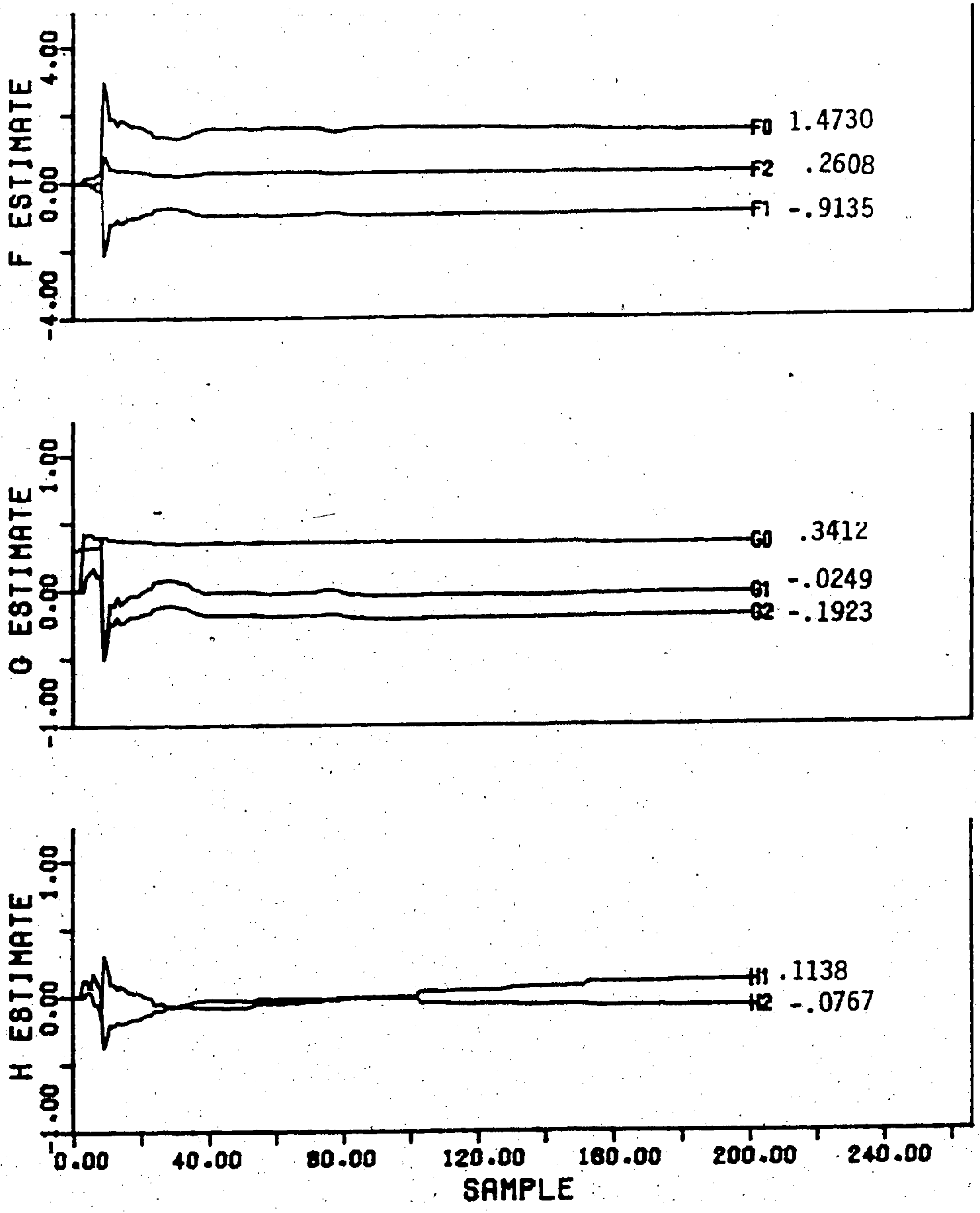


Figure 4.17 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

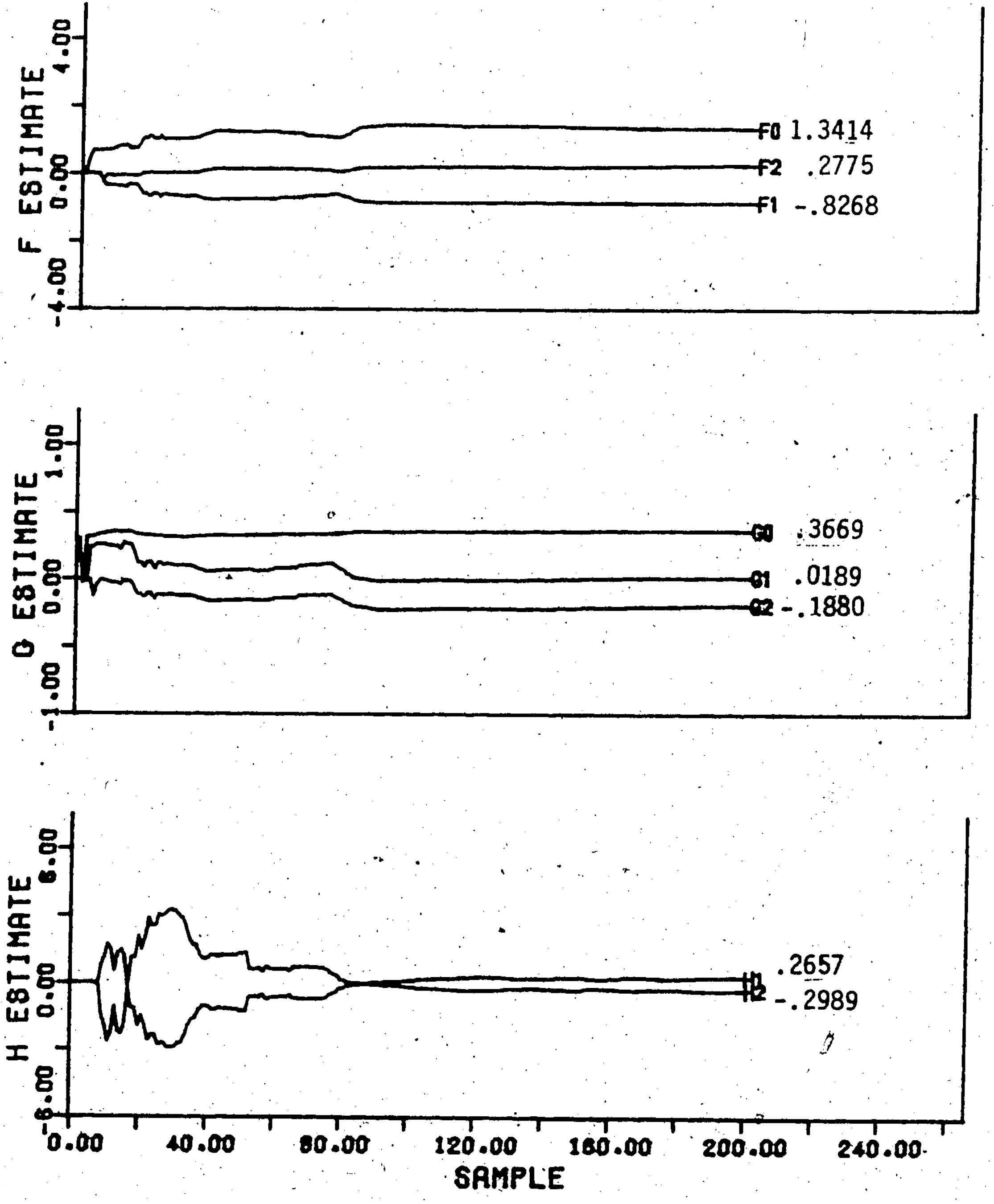


Figure 4.18 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

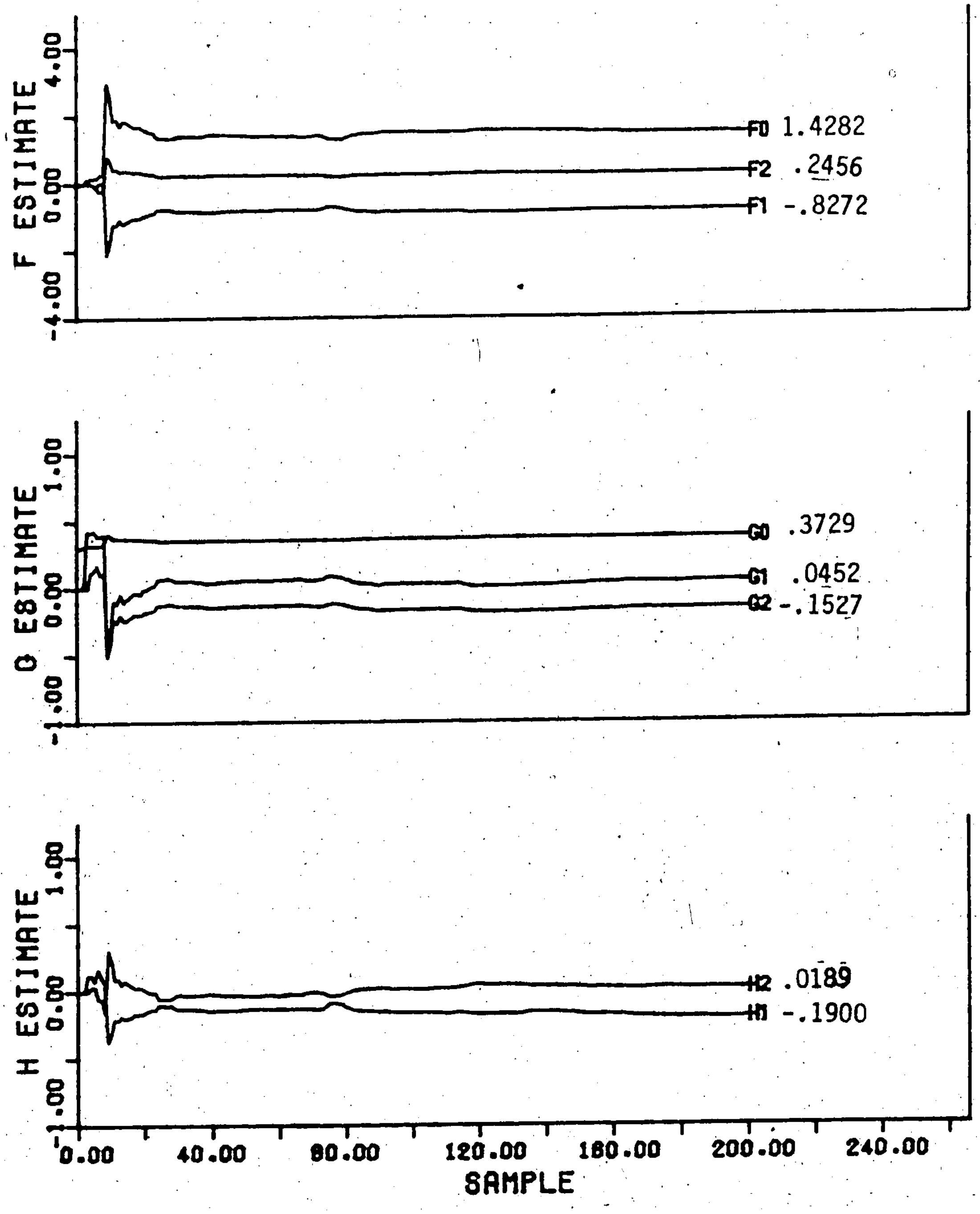


Figure 4.19 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Upper Diagonal Factorization Estimator

4.2.4 Recursive Learning Estimator

The system response when the recursive learning estimator is used is shown in Figures 4.20, 4.21 and 4.22. The sawtooth wave function and the step change in setpoint results show good tracking ability and the control effort value is considerably lower than the recursive square root and recursive upper diagonal factorization results, but only the sawtooth wave function had a smaller sum of predicted errors. However, for the square wave change in setpoint, the control performance is characterized by very poor tracking ability which is reflected in the larger sum of prediction errors and control effort values than were obtained using the previous estimators. The setpoint tracking performance for the square wave setpoint does not improve even after 800 sample instants as can be seen from the results in Figure 4.23.

The performance of the estimation algorithm in terms of the behavior of the parameter estimates for all three setpoint changes can be seen in Figures 4.24, 4.25 and 4.26. For the square wave setpoint change the F and H parameters are constant but the estimate values are only within ±90% of the true values while the G parameters show erratic changes with no signs of becoming constant or converging. The F and H estimates for the sawtooth function change in setpoint have smaller values than the parameter estimates obtained for the square wave setpoint change while the G estimates are larger than for the square wave setpoint. The G

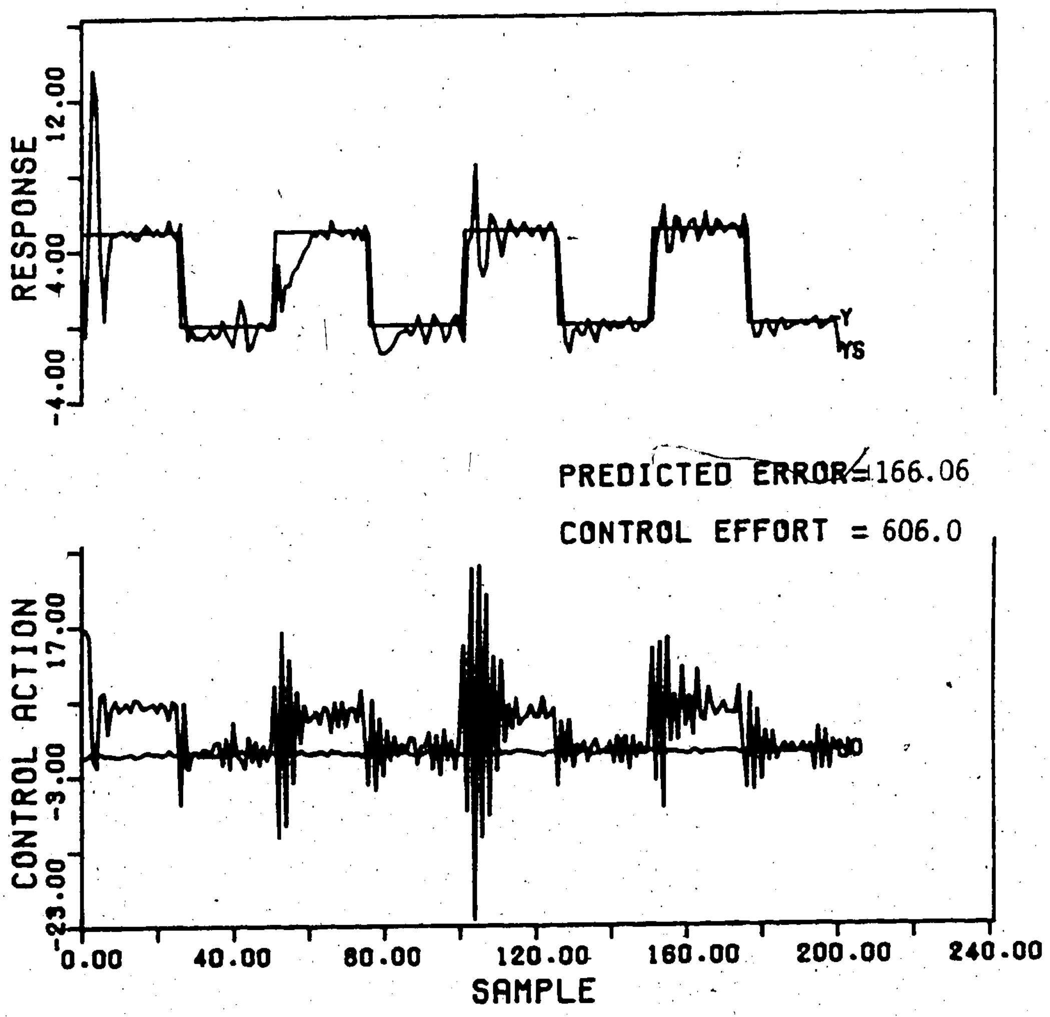


Figure 4.20 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Learning Estimator

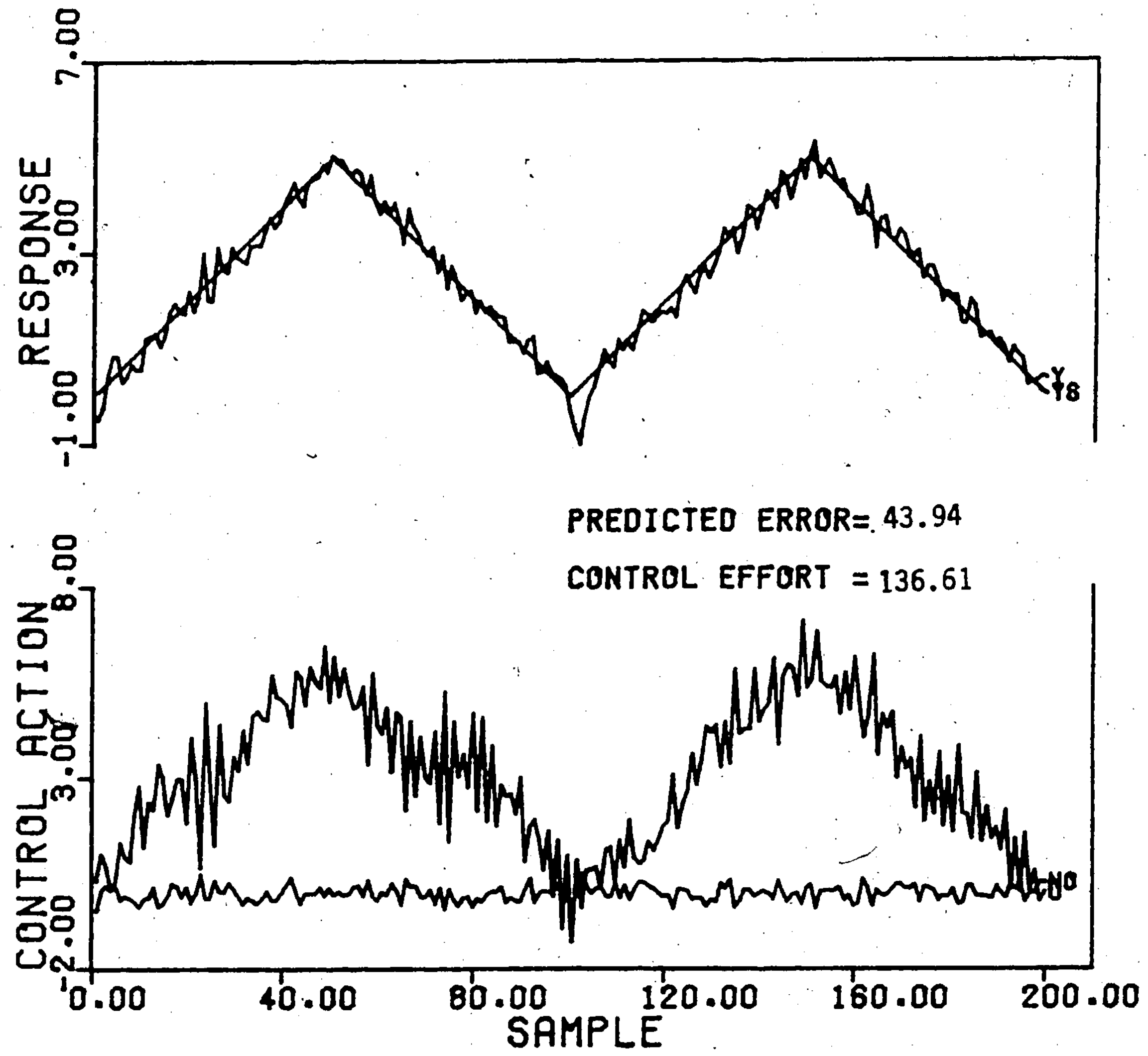


Figure 4.21 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Learning Estimator

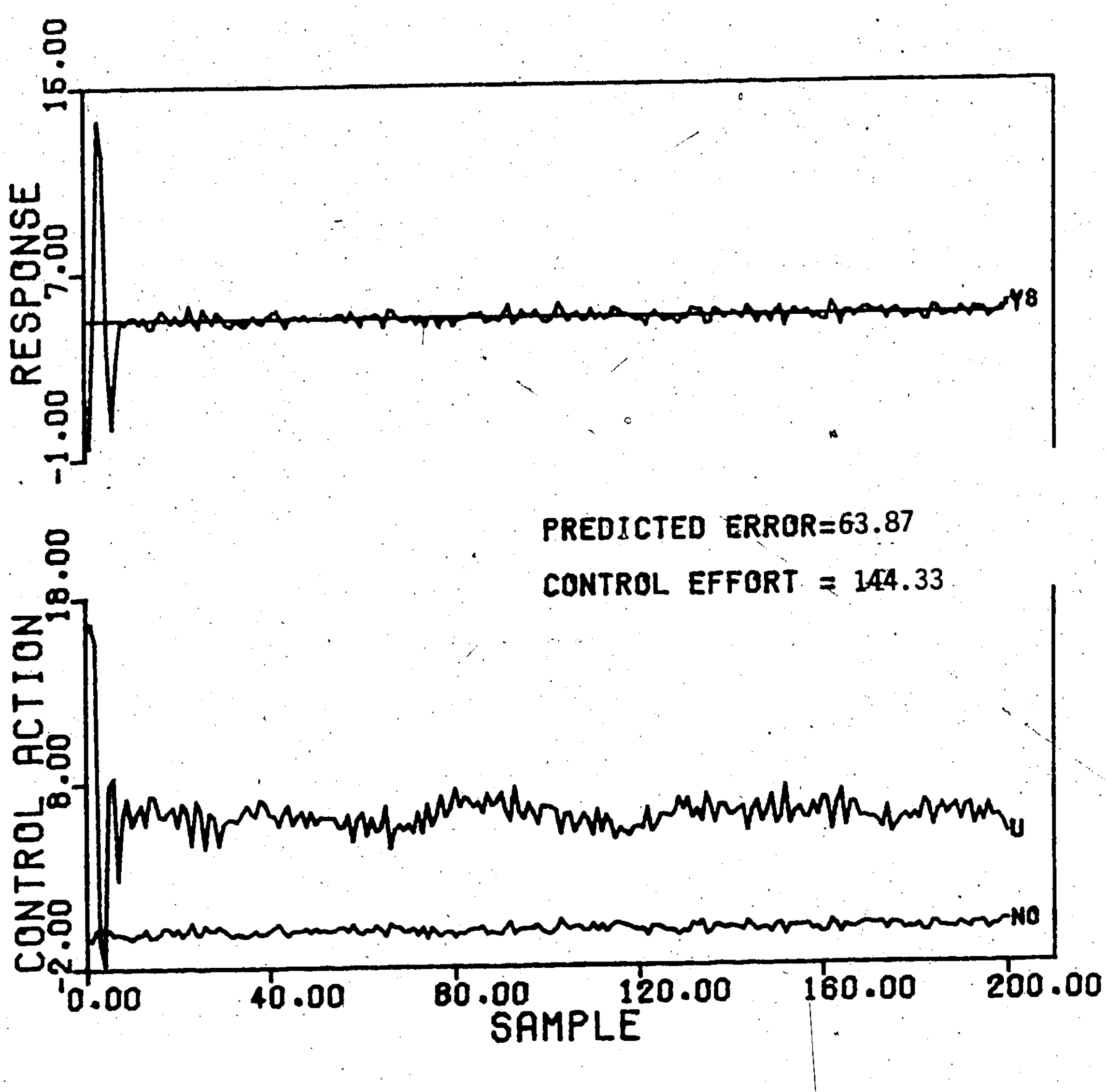


Figure 4.22 Setpoint Tracking a Step Change in Setpoint using the Recursive Learning Estimator

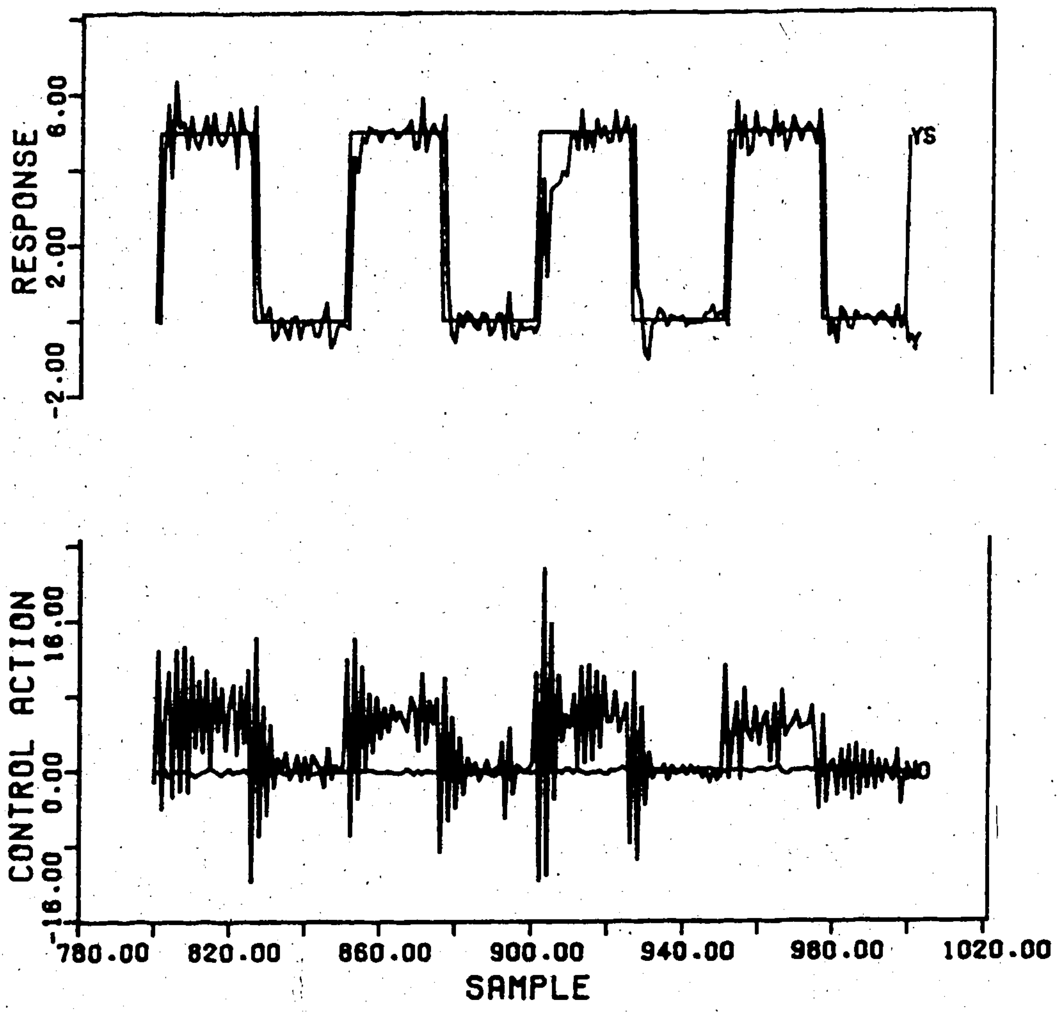


Figure 4.23 Square Wave Setpoint Tracking after 800 Sample Intervals using the Recursive Learning Estimator

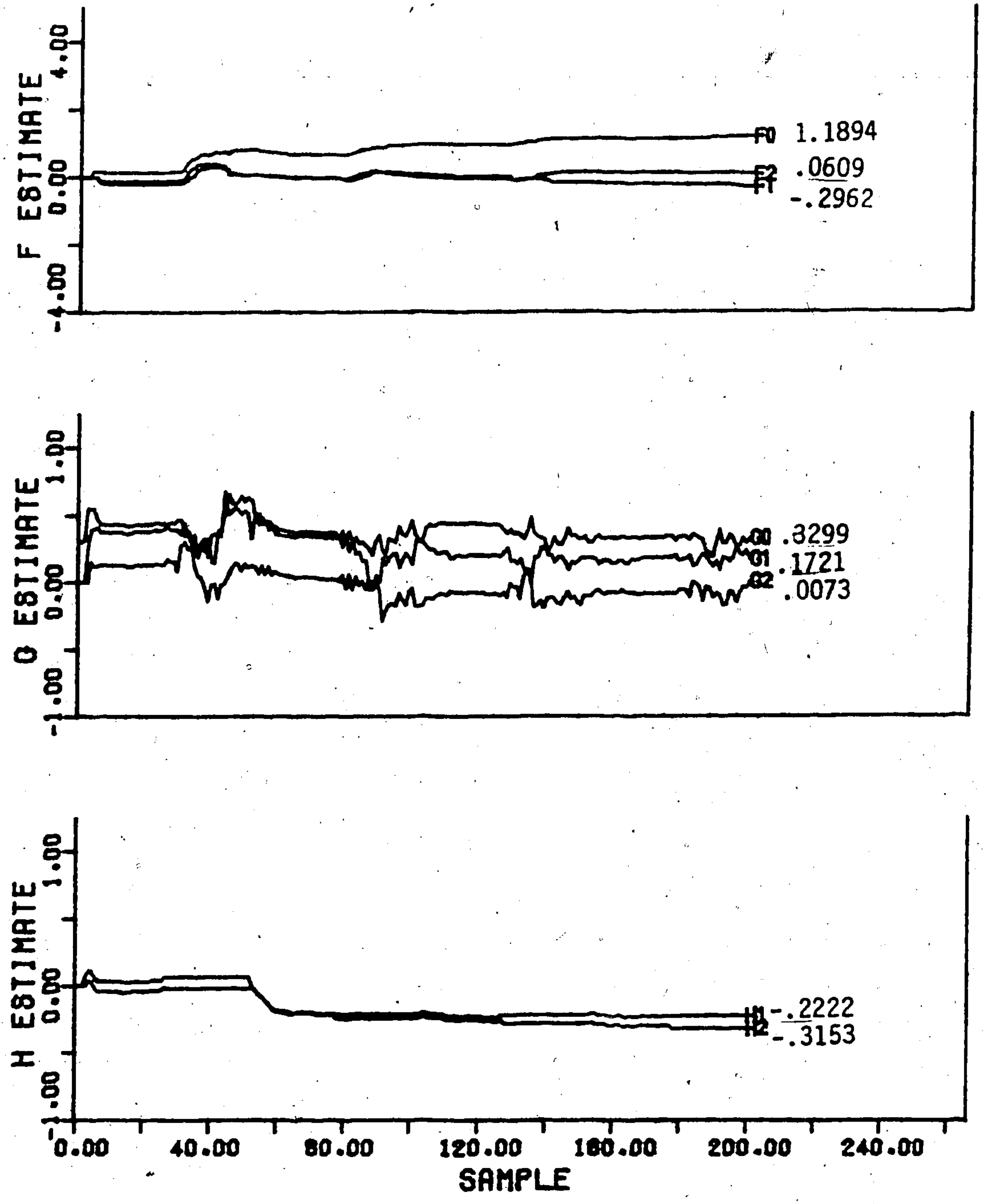


Figure 24 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Learning Estimator

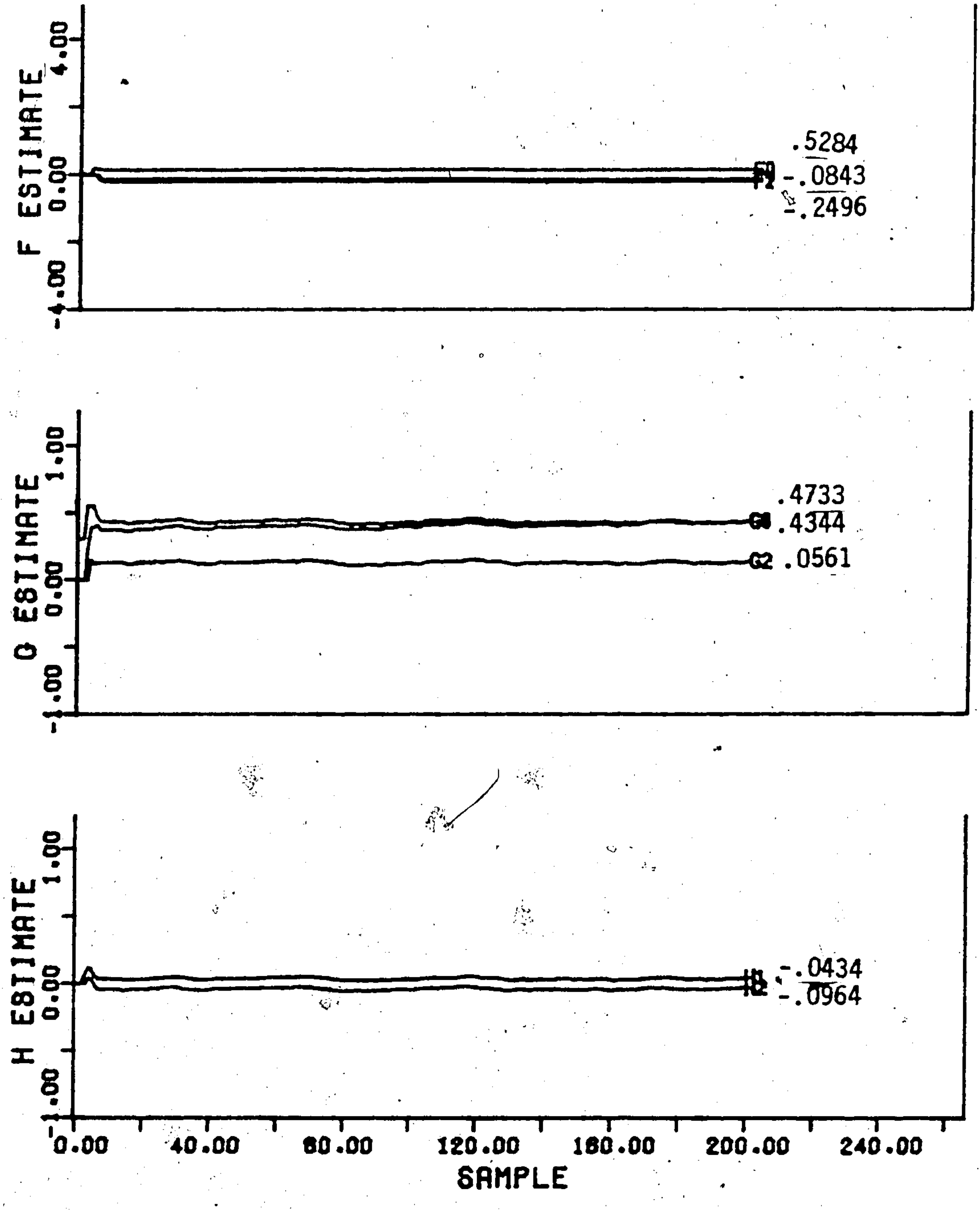


Figure 4.25 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Learning Estimator

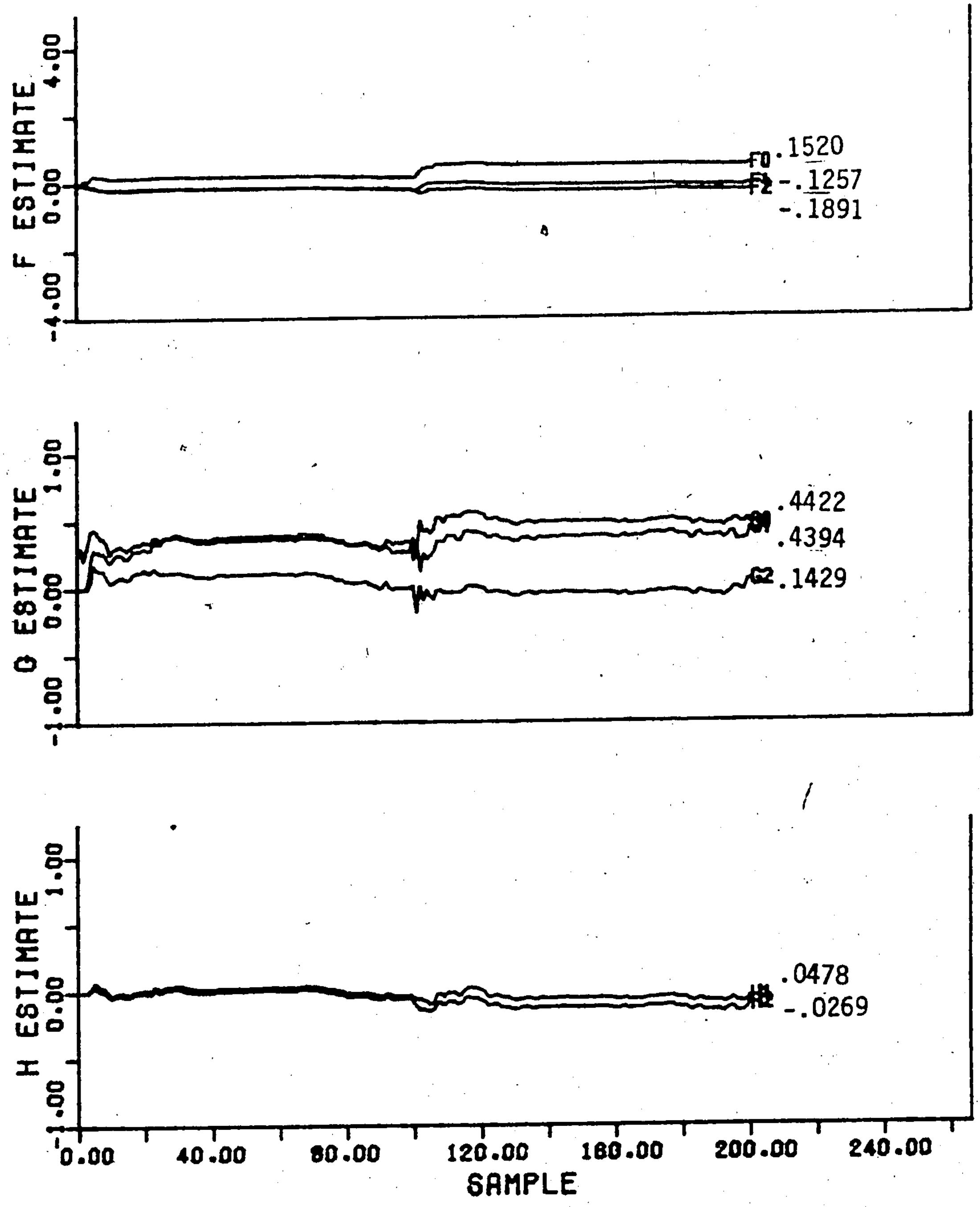


Figure 4.26 Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Learning Estimator

parameter behavior is less erratic and the H parameters display fewer changes and the estimates are only within ±70% of the true values. The estimates for the step change in setpoint simulation are relatively constant with no indication of convergence to the true values. As can be observed, after 200 sample instants the estimates are not reliable. Simulations up to the 1000th sample instant reveal that the estimates still do not converge to the true value.

In an attempt to improve the estimates obtained by the recursive learning estimator the recursive least squares estimator was used for the first 25 sample instants. It was anticipated that the recursive least squares algorithm would identify parameter estimates close to their true values in this time so the recursive learning estimator could then adapt the parameters to the true values. This change had little effect on the estimates although some parameters were closer to the true value.

4.2.5 Recursive Maximum Likelihood Estimator

The recursive maximum likelihood estimator does not use a covariance matrix instead it has a gain which is chosen before identification begins.

The parameter estimates resulting from use of the recursive maximum likelihood estimator were extremely sensitive to the initial gain values used. The parameter estimates determined for initial gains of 1.0 to 150 were only within ±50% of the true values after 200 sample instants. However, as the initial gain values were increased from 150 to 400 the estimates began to increase to values that were four and five times larger than the true values and remained large even after 1000 sample instants. If the initial gain is in the range of 401 and 999 the parameter estimates converge to values similar to those for the gains between 1.0 and 150. The most interesting observation occurs for gain values of 999, 1000 and 1001. As stated above, a gain of 999 yields acceptable parameter estimates, and a gain of 1001 also give estimates (excluding g_1) within ±50% of the true values. However an initial gain of 1000 caused all parameter estimates except the H estimates to blow up. The reason for this occurrence was not determined.

This sensitivity to initial gain values was not observed with the other estimators when the initial covariance matrix was altered. The best results were obtained for an initial gain value of 1.0 so this value was used for all evaluations of the recursive maximum likelihood estimator. Since the model was a third order, nine parameters should be estimated but in practice, one parameter is fixed, that is $h_0 = -1.0$ leaving only eight parameters to be estimated. Although the recursive maximum likelihood algorithm is structured so that $n_f = n_g = n_h$ by setting $n_h = 2$ the correct model can be recovered [29].

The results from tracking simulations for the three different setpoint changes using the recursive maximum likelihood estimator are presented in Figures 4.27, 4.28 and

4.29. For all three setpoint changes the output never reaches the desired maximum setpoint value even after 200 sample instants. The output signal for the sawtooth change finally started to track after 400 sample instants. The poor control performance is accompanied by a fairly small change in the parameter estimates as can be seen from Figures 4.30, 4.31 and 4.32.

Analysis of the adaption that takes place reveals that the initial jumps in the F estimates evident in the results using the other identification schemes for a square wave change in setpoint no longer exist. The G estimates show very slow adaption and the initial changes are not severe. In contrast, both the h_1 and h_2 estimates are of the incorrect sign and are still increasing by the 500th sample instant with the trend to values even further away from their true value.

At the 500th sample instant, the only parameter estimate that is close to the true value is the g_0 coefficient. All the remaining parameter estimates except g_2 have grown larger than the values at the 200th sample instant. The f_0, f_1, h_1 and h_2 are moving away from the true values while the remaining parameter estimates are approaching the true values.

For the sawtooth function setpoint change the F and G parameter estimates are similar to those found for the square wave setpoint change but there are more noticeable initial oscillations in the G estimates. The h_1 parameter

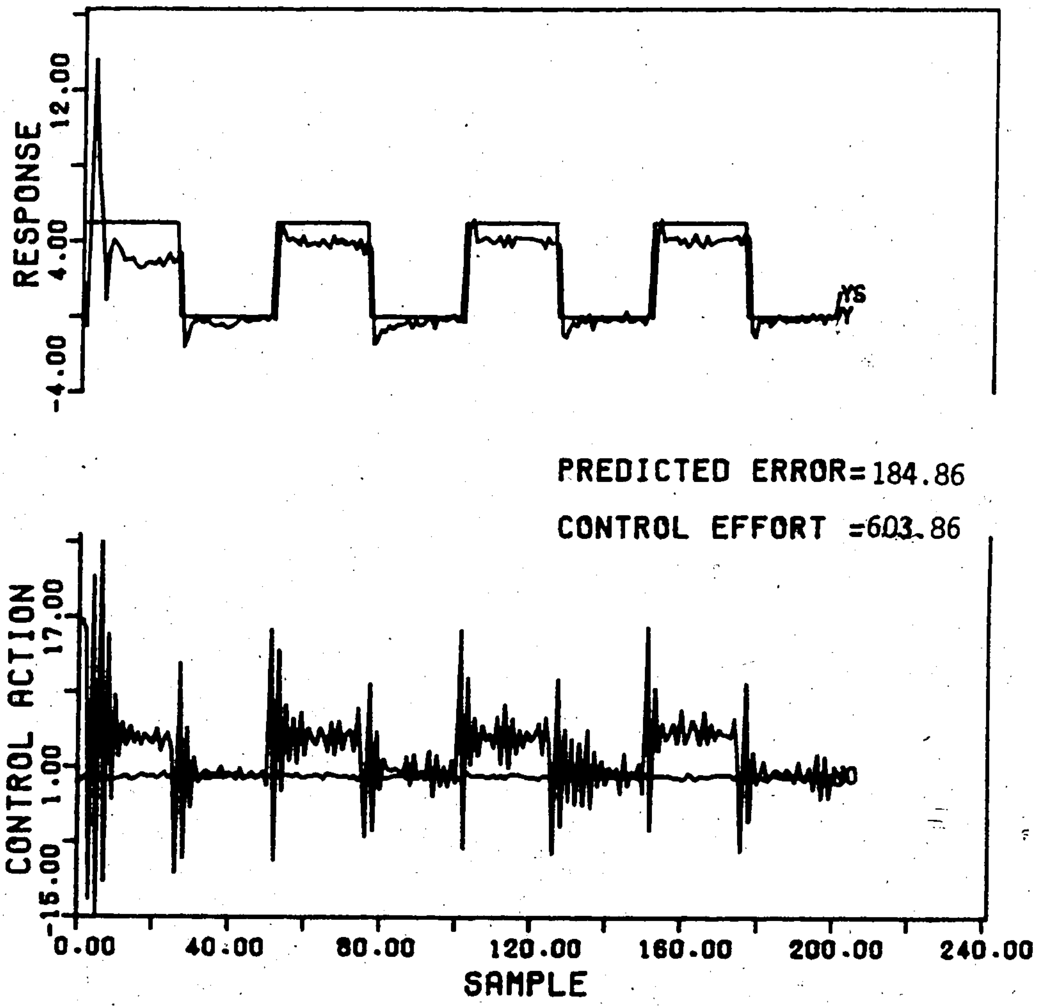


Figure 4.27 Setpoint Tracking for a Square Wave Change in Setpoint using the Recursive Maximum Likelihood Estimator

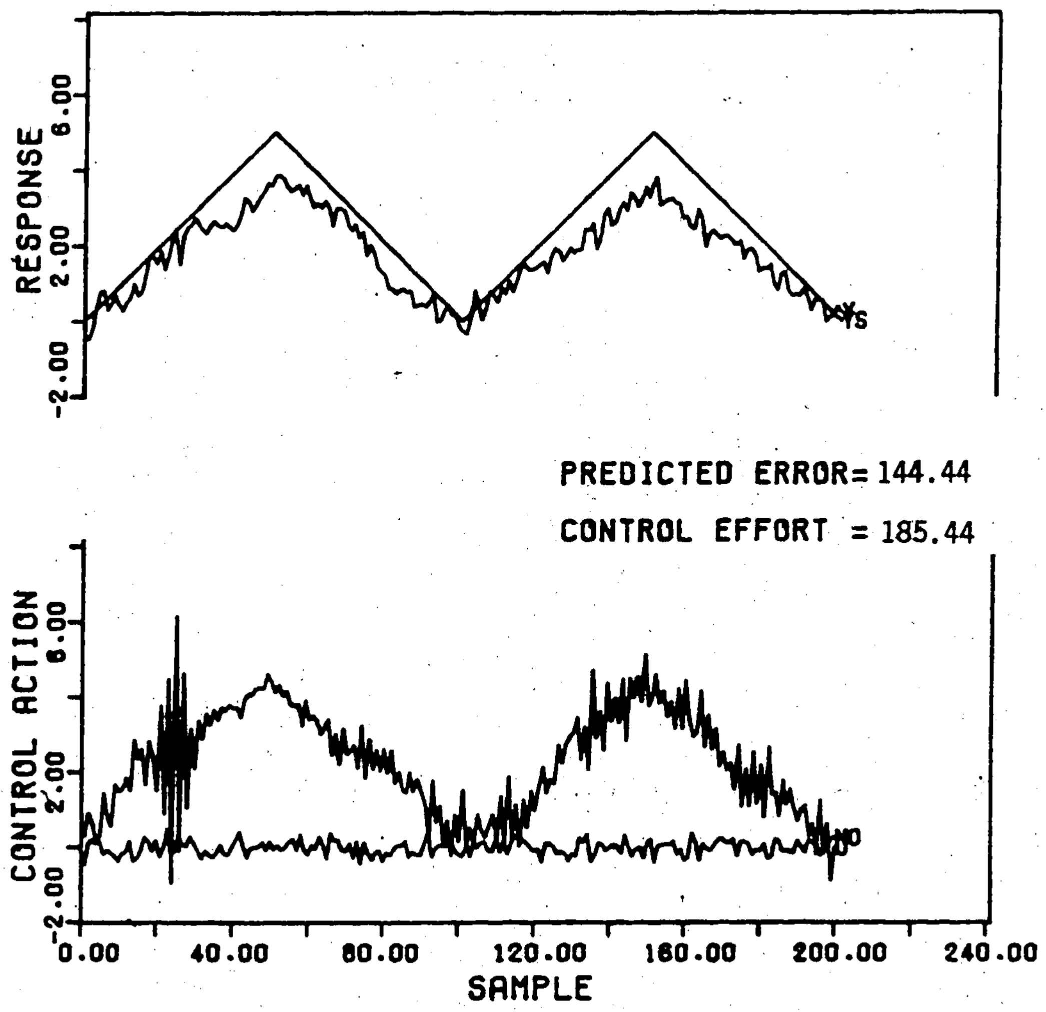


Figure 4.28 Setpoint Tracking for a Sawtooth Function Change in Setpoint using the Recursive Maximum Likelihood Estimator

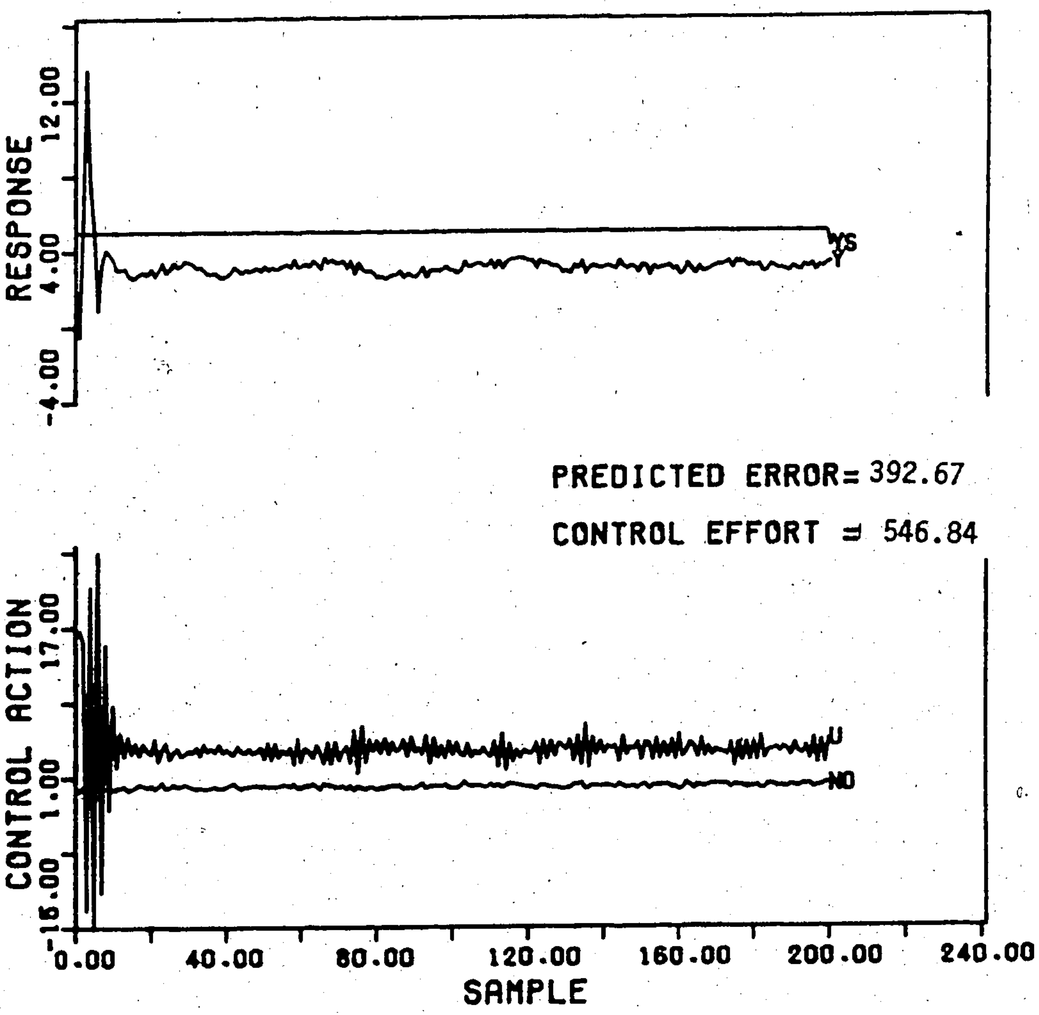


Figure 4.29 Setpoint Tracking a Step Change in Setpoint using the Recursive Maximum Likelihood Estimator

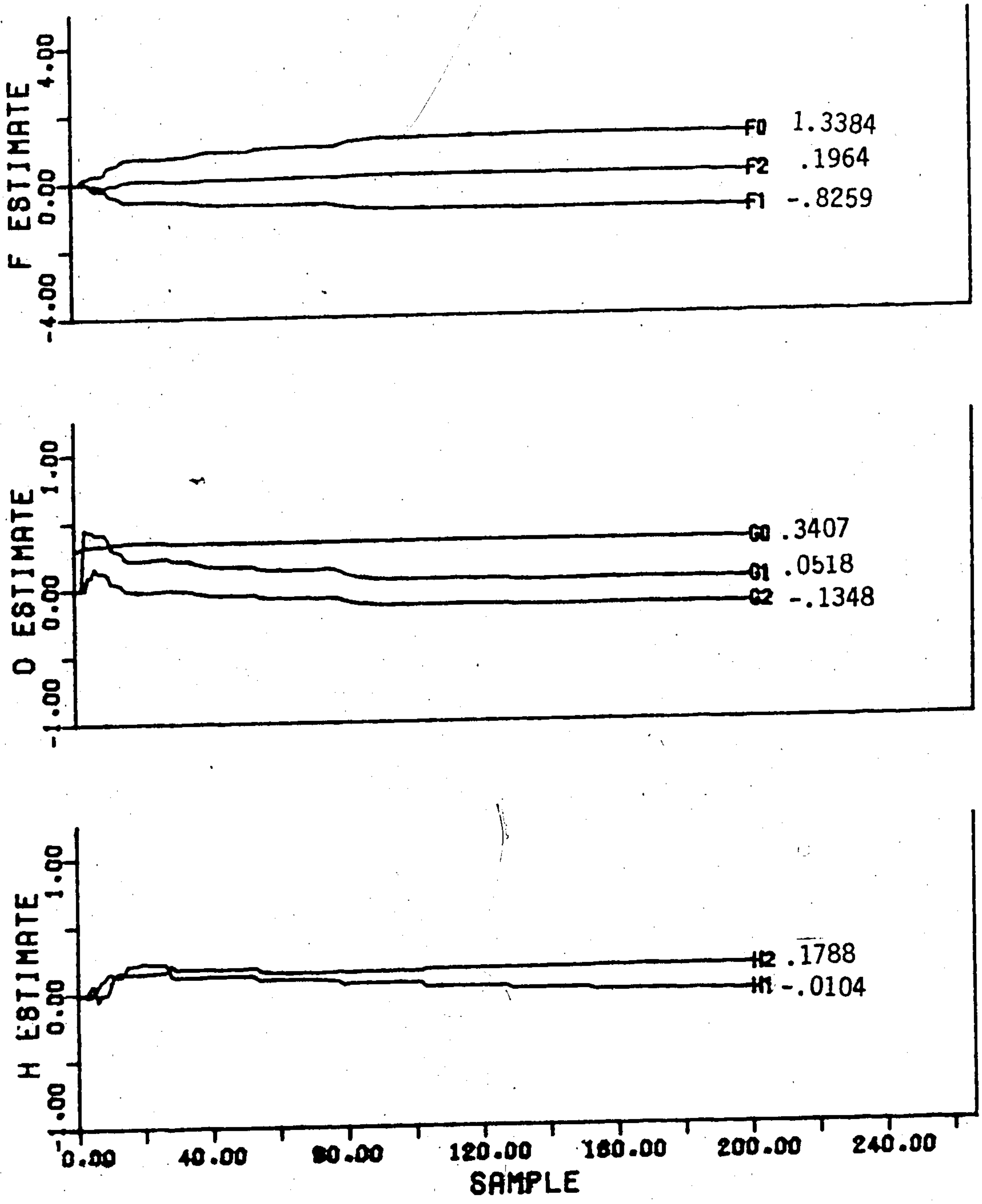


Figure 4.30 Controller Parameter Estimates for the Square Wave Change in Setpoint using the Recursive Maximum Likelihood Estimator

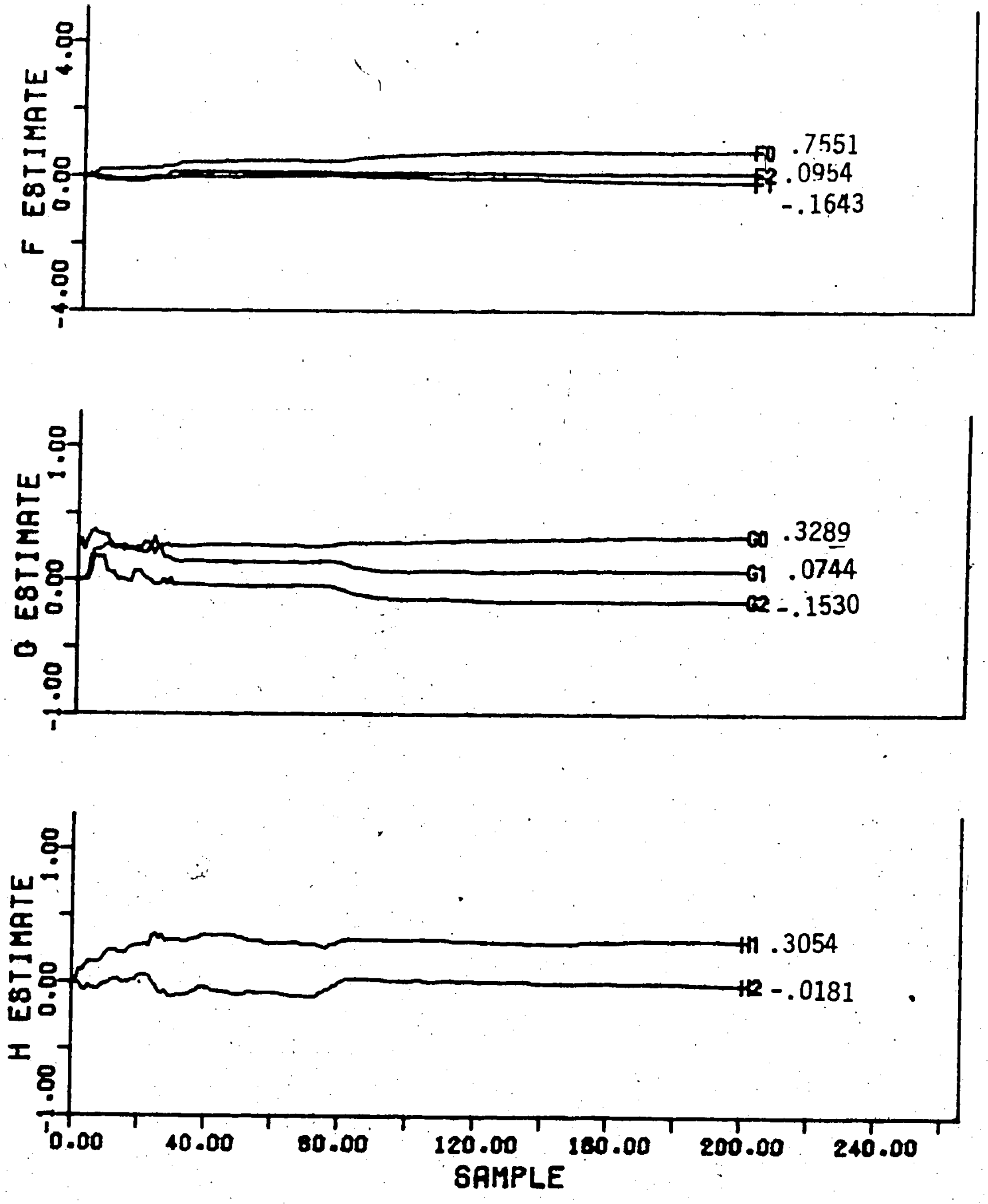


Figure 4.31 Controller Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Maximum Likelihood Estimator

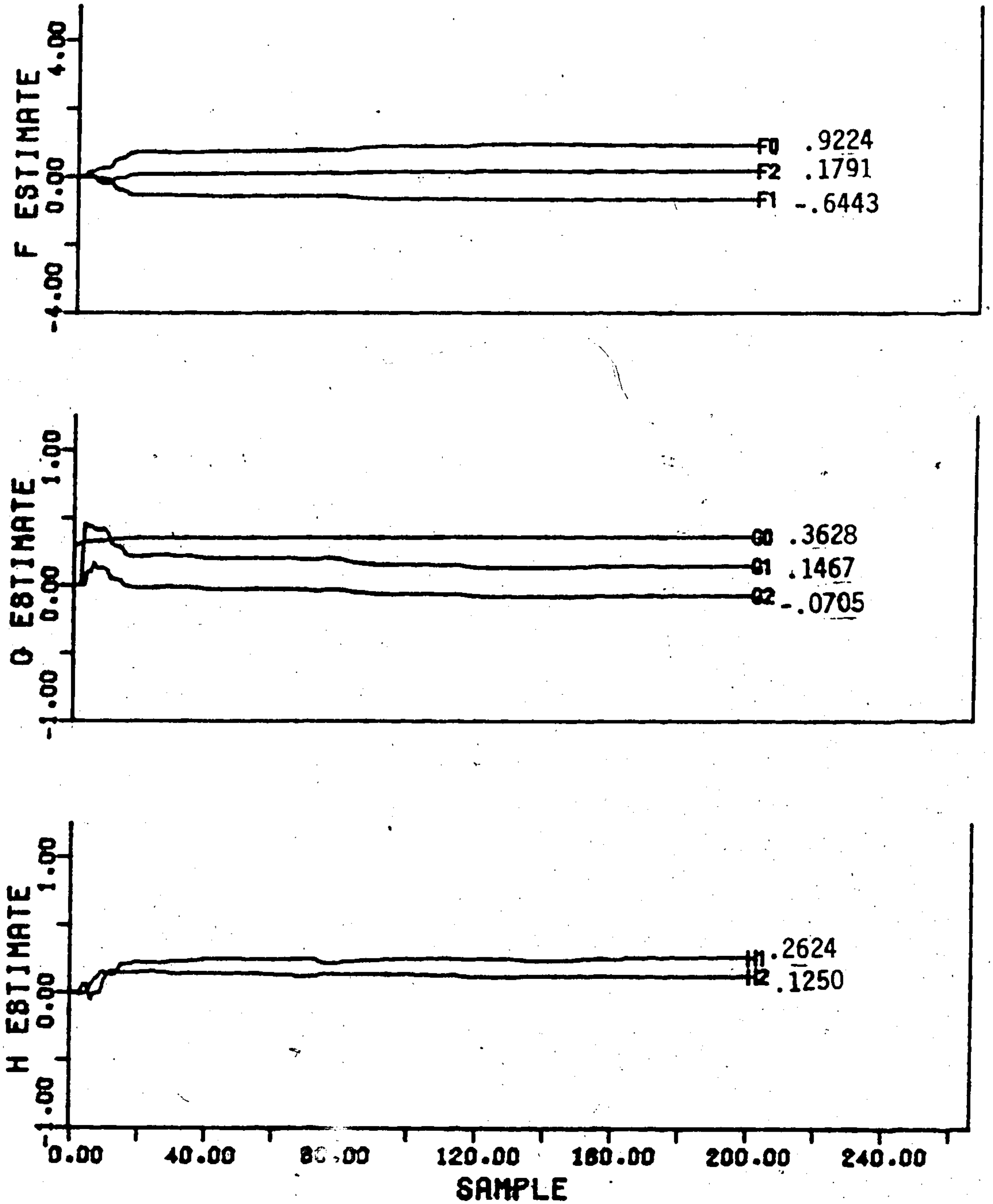


Figure 4.3. Controller Parameter Estimates for the Step Change in Setpoint using the Recursive Maximum Likelihood Estimator

estimate increases to a greater value initially than for the square wave setpoint change but does not attain its true value. The h_2 estimate appears to be adapting to its true value but by the 500th sample instant it is still not near that value. After 500 sample instants the f_0, g_0 and g_2 estimates are approaching the true values but have not converged and continue to increase.

For the step change in setpoint the parameter estimates are not noticeably increasing after 40 sample instants but the parameter estimates after 500 sample instants show that there is no convergence. The H estimates are both positive and appear to remain that way. Although not shown, the estimate values at the 500th sample instant are only with $\pm 70\%$ of the true values while the g_1 parameter estimate is in error by a factor of 2 (0.073 vs -0.07) and g_2 is 39% below its true value. The g_0 value has grown larger than its actual value but is still within 8% of its true value. The f_1 parameter estimate is only 1% (-0.7133 vs -0.71) in error while the f_0 and f_2 are over 20% from the true values.

As identification using the recursive maximum likelihood estimator continues the gain may become small which causes the estimates of the H parameters to be far from the true values. Therefore a restart of the vector may prove useful. A convergence test suggested by Soderstrom [29] was used to try to improve the estimates.

The modification was implemented in the following way. The algorithm was applied in a straight forward manner for n1 steps, where n1 is an arbitrary number of steps, then a convergence test was performed. If convergence was attained the algorithm would continue. If no convergence had occurred a restart would be made by holding the parameter estimate vector constant and reinitializing all other variable to their starting values. The parameter estimate vector is constrained to be constant for the next n2 steps, where n2 is also arbitrary. After another n1 steps the convergence test is made again and continues until convergence is attained.

The convergence test uses a loss function

$$W(\Theta_k;\Theta_{k-n1-n2}) = \frac{1}{\sigma^2} E\epsilon_k^2 \tag{4.3}$$

where

$$\sigma^2 = \sqrt{\frac{2}{k} V_k^2(\Theta_k)} \tag{4.4}$$

If the value of W is within 5% of 1.0, convergence may be considered to have occurred and no more restarts are made.

Since the recursive maximum likelihood simulations exhibited such poor control performance for all three changes, the same simulations were repeated for $n_h = 3$ to determine if the reduced number of estimates was the cause. It was found that the control was worse, proven by the increase in control effort and sum of predicted error values for all three cases. The h_3 estimate, which is equal to 0.0, never approached zero so another explanation for the unsatisfactory control must be found.

According to Ljung et al [32], parameter identifiability may be lost if the feedback is too simple and if there are no additional input signals. The input signal may not excite the system properly especially when the system noise is very small as was the case for all simulations. Simulations were performed with system noise variances of 0.1 and 1.0. There was no significant improvement in the parameter estimates nor was there any improvement in the control performance compared too the results for the noise-free simulations. Although the recursive maximum likelihood identification technique is known to provide reliable parameter estimates for uncontrolled systems, it has not been shown to perform well with controlled systems.

4.3 The Effect of Q and R Weighting

The control effort of the previous simulations was reduced by using Q and R weighting while continuing to track the setpoint. Although a scalar Q weighting would have produced offset in the output this was avoided by choosing the R-weighting to compensate for this offset. It was found by trial and error that the best choice for R was 1.0 plus the value of the Q weighting.

The values chosen for Q and R were 0.2 and 1.2 respectively. These values gave the best control performance for the least control effort. It should be noted that although this weighting gave good results for the following simulations it should not be used as a general rule for all systems.

The setpoint tracking results using these R and Q weighting values can be found in Figures 4.33, 4.34 and 4.35 for the square wave, sawtooth function and step changes in setpoints, respectively. These results were obtained using the recursive least squares estimator. Figures 4.36, 4.37 and 4.38 show the results obtained using the recursive square root estimator. The differences in control effort and sum of predicted errors between the recursive square root and recursive upper diagonal factorization estimators were negligible so the performance of both estimators can be discussed together.

The following observations hold true for the three algorithms as all three estimators gave similar results.

When compared to the minimum variance control the control effort and sum of predicted errors were reduced an average of 25% for the square wave and the sawtooth function setpoint changes while the sum of predicted errors was reduced 54.5% for the constant setpoint case. The initial overshoot exhibited in the square wave and constant setpoints is reduced by 28% when the weighting is used and most importantly, the initial tracking error for the sawtooth setpoint change are reduced significantly.

The parameter estimates yielded by the three recursive least squares simulations can be found in Figures 4.39, 4.40 and 4.41. For the square wave setpoint the F and G estimates

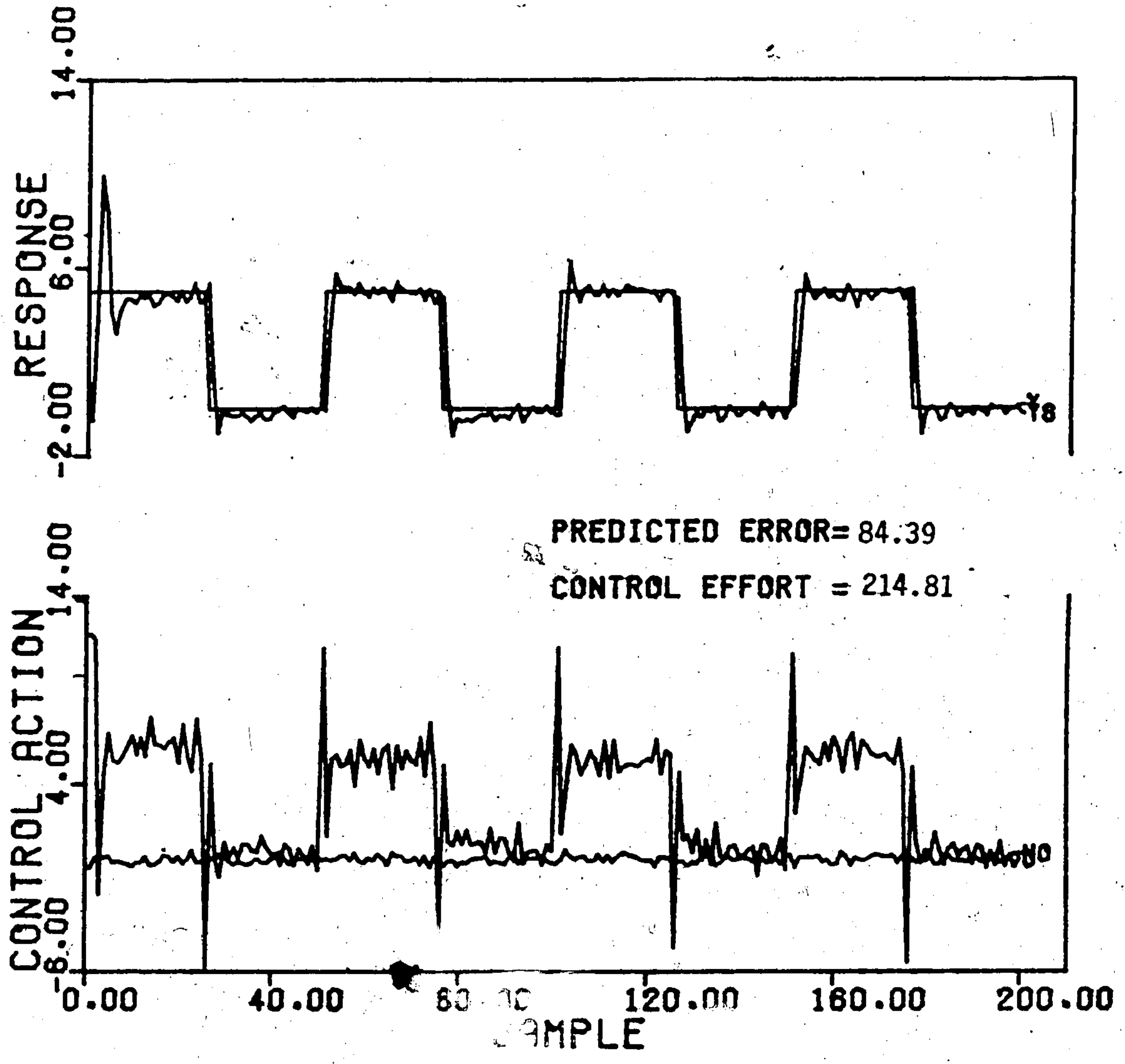


Figure 4.33 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for the Square Wave Change in Setpoint

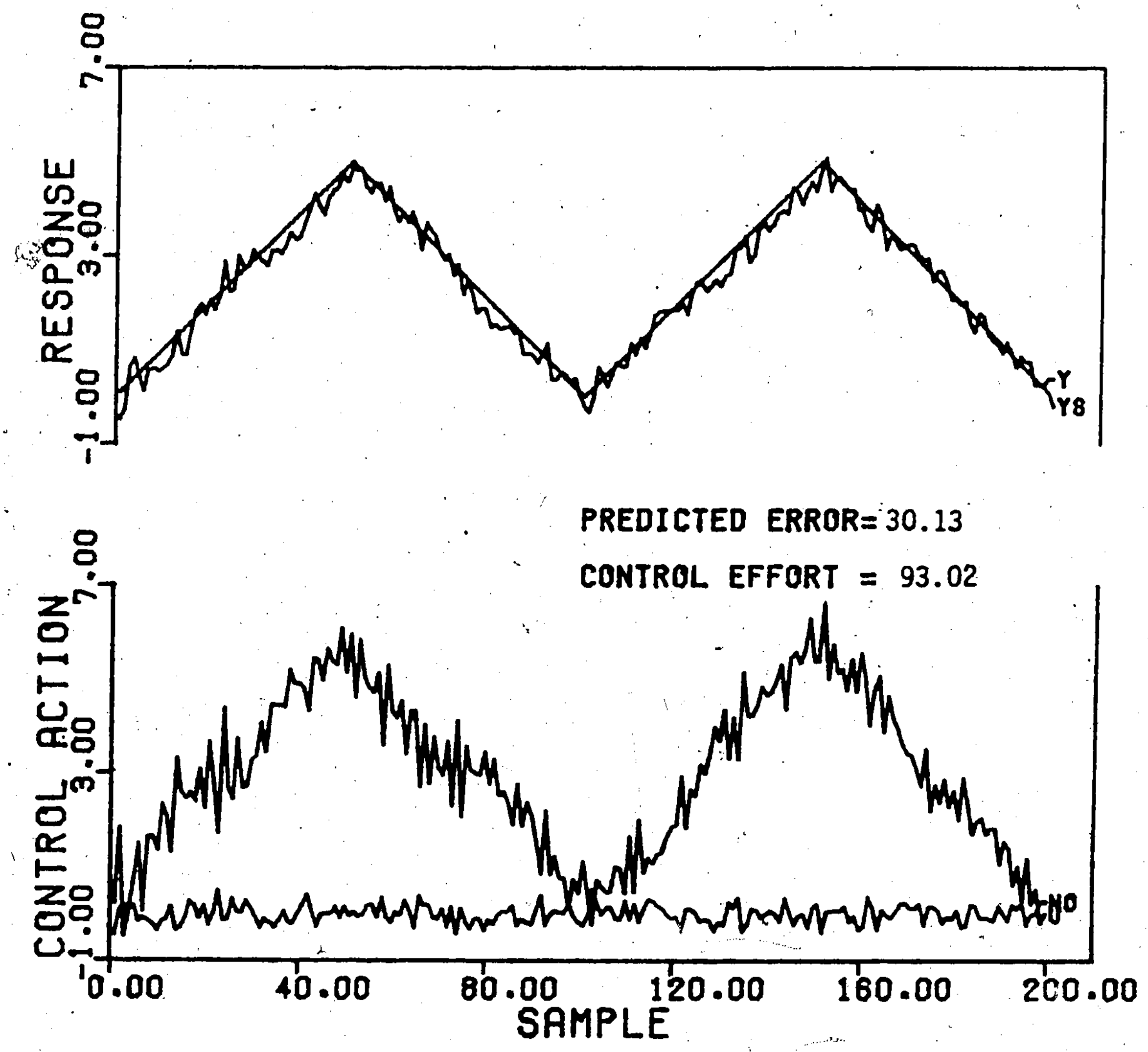


Figure 4.34 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for the Sawtooth Function Change in Setpoint

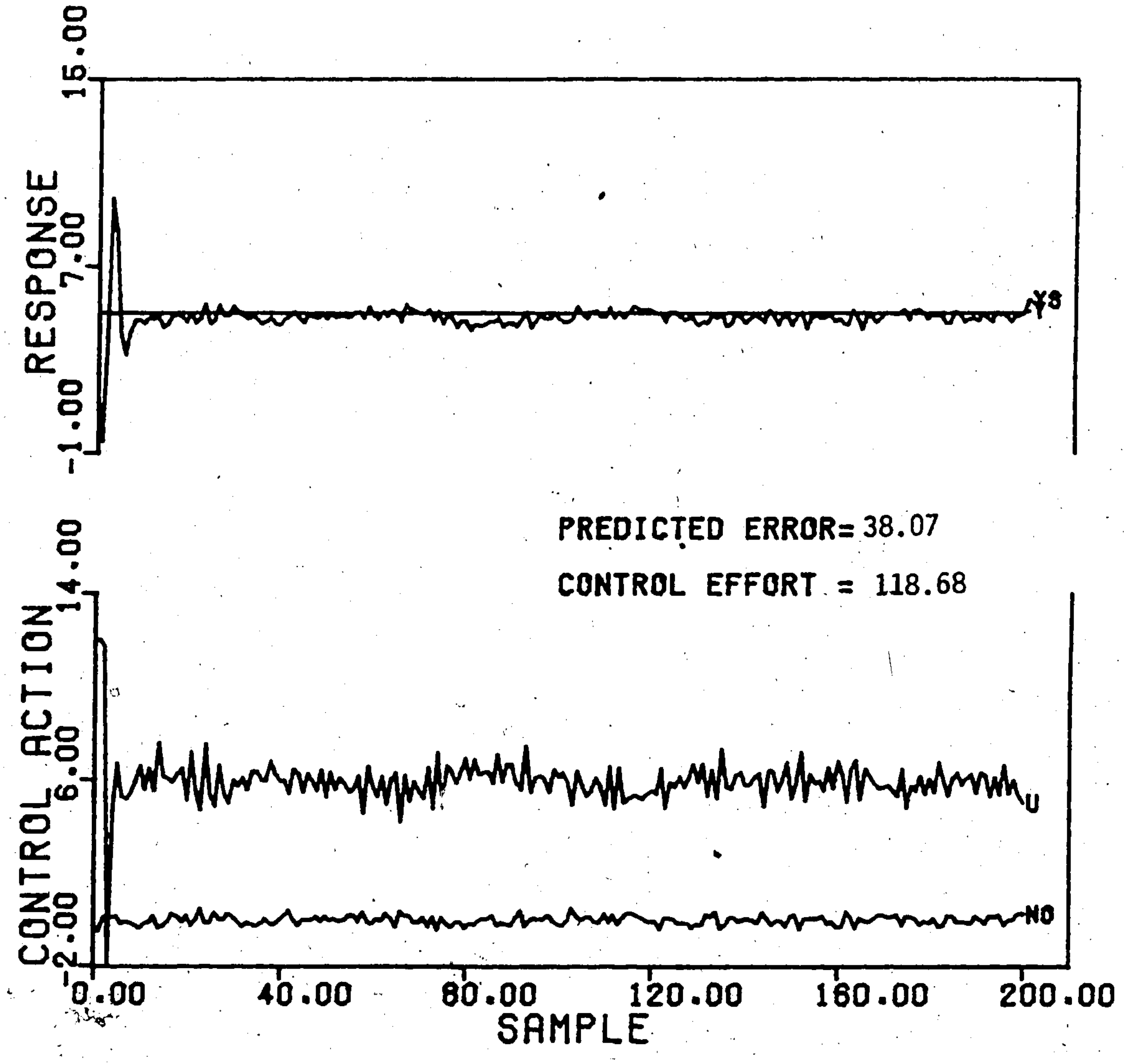


Figure 4.35 Control Response with Q and R Weighting using the Recursive Least Squares Estimator for a Step Change in Setpoint

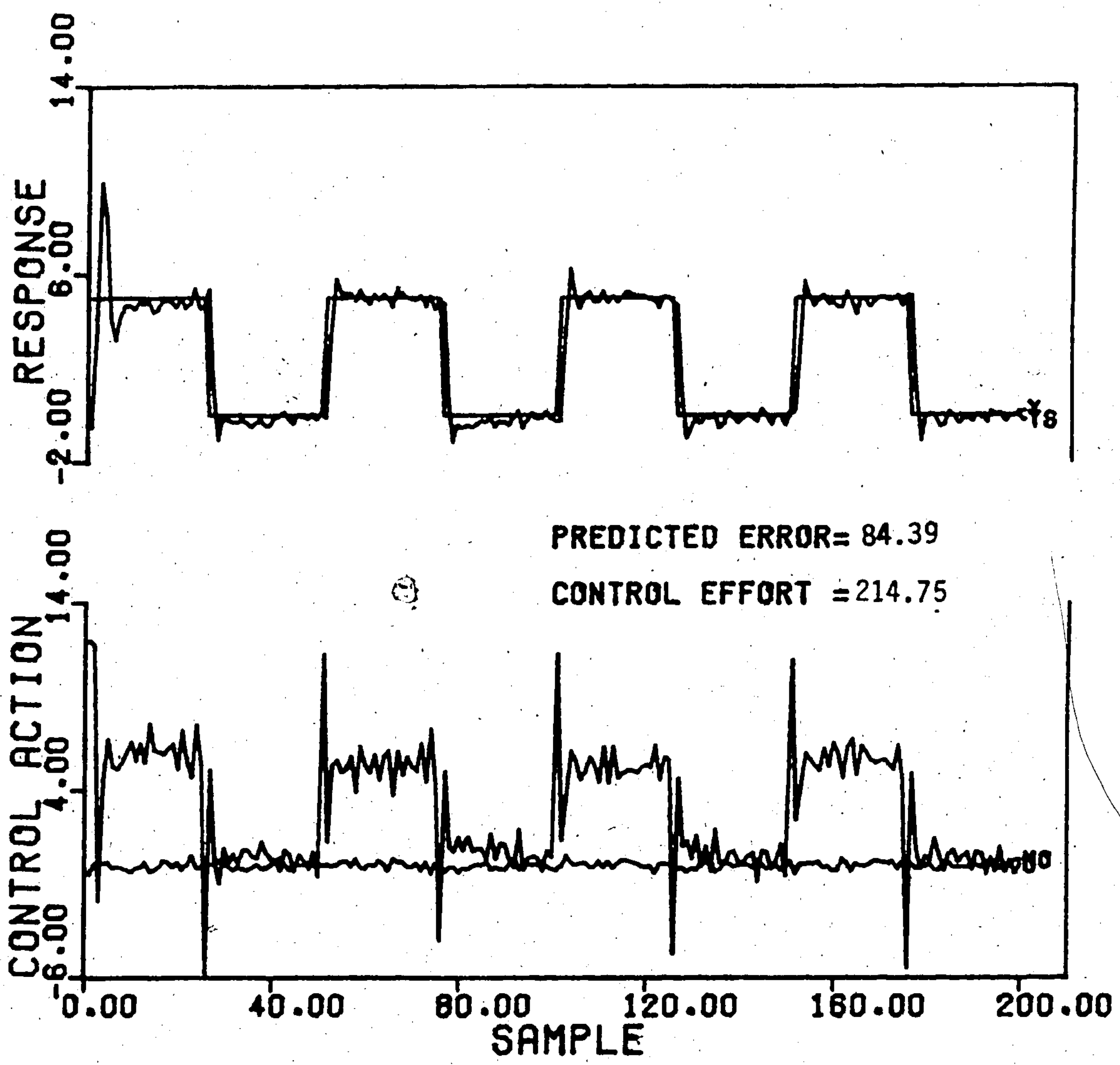


Figure 4.36 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Square Wave Change in Setpoint

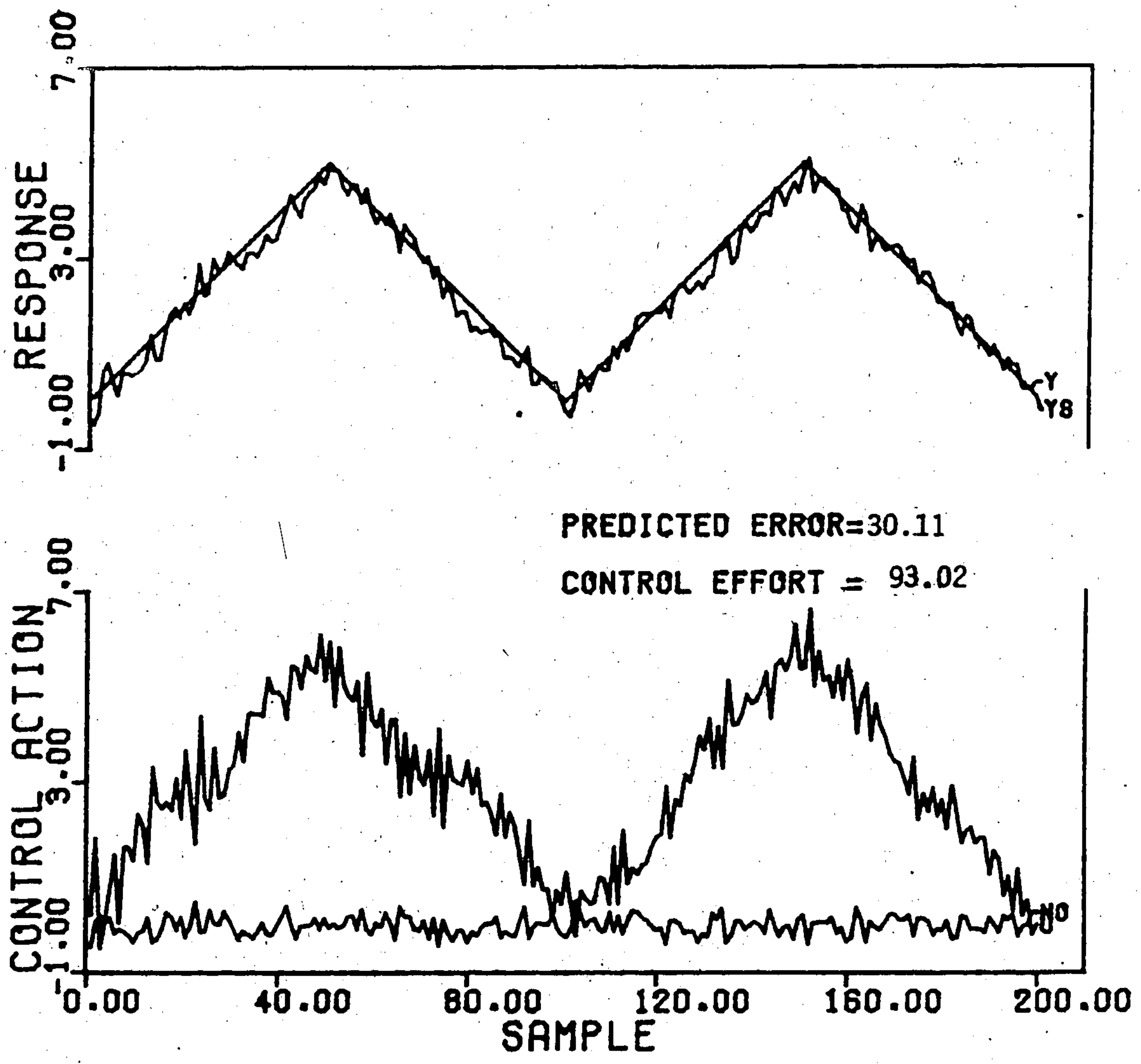


Figure 4.37 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Sawtooth Function Change in Setpoint

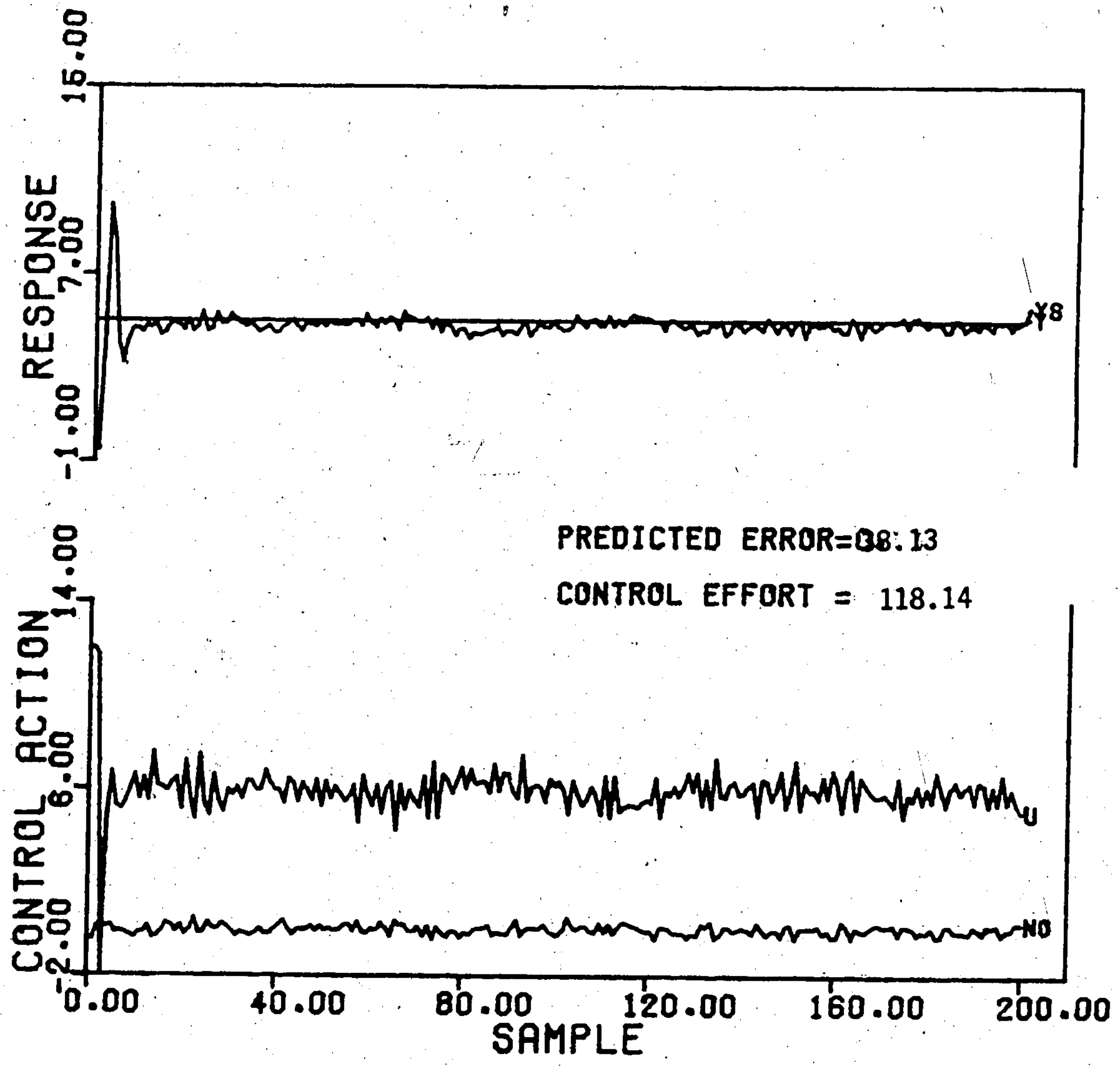


Figure 4.38 Control Response with Q and R Weighing using the Recursive Square Root Estimator for the Step Change in Setpoint

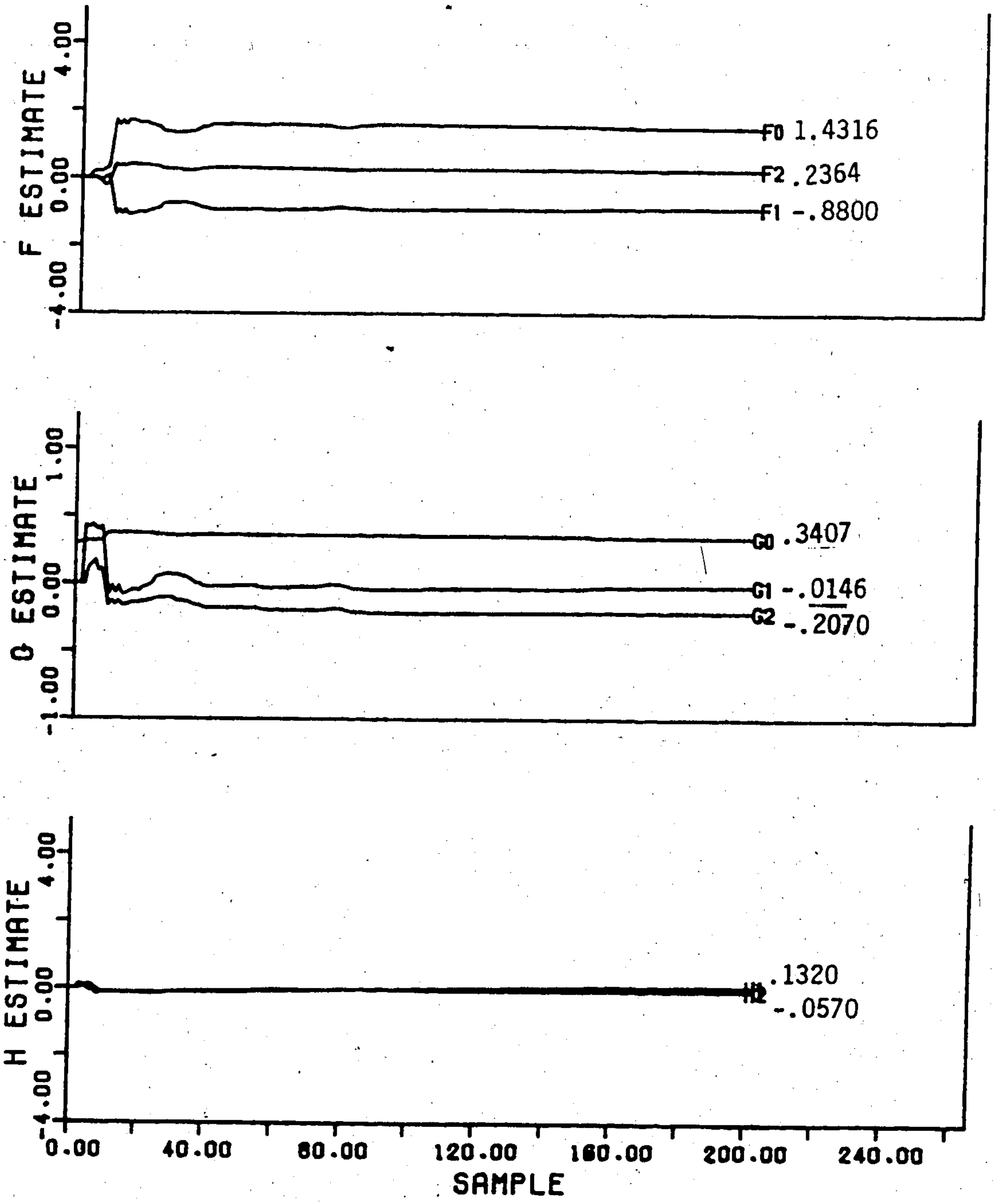


Figure 4.39 Parameter Estimates for Square Wave Change in Setpoint using the Recursive Least Squares Estimator with Q and R Weighting

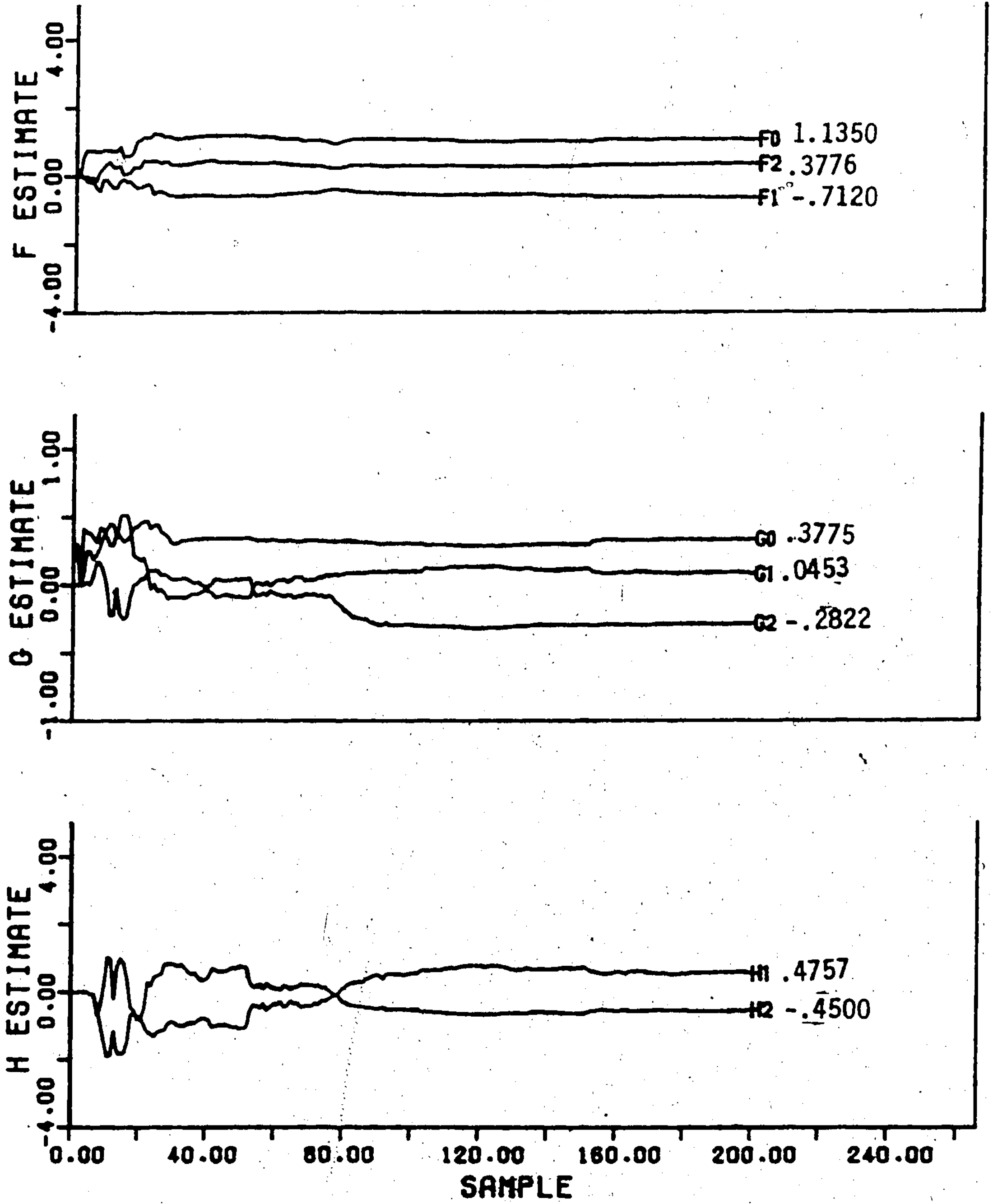


Figure 4.40 Parameter Estimates for Sawtooth Function Change in Setpoint using the Recursive Least Squares Estimator with Q and R weighting

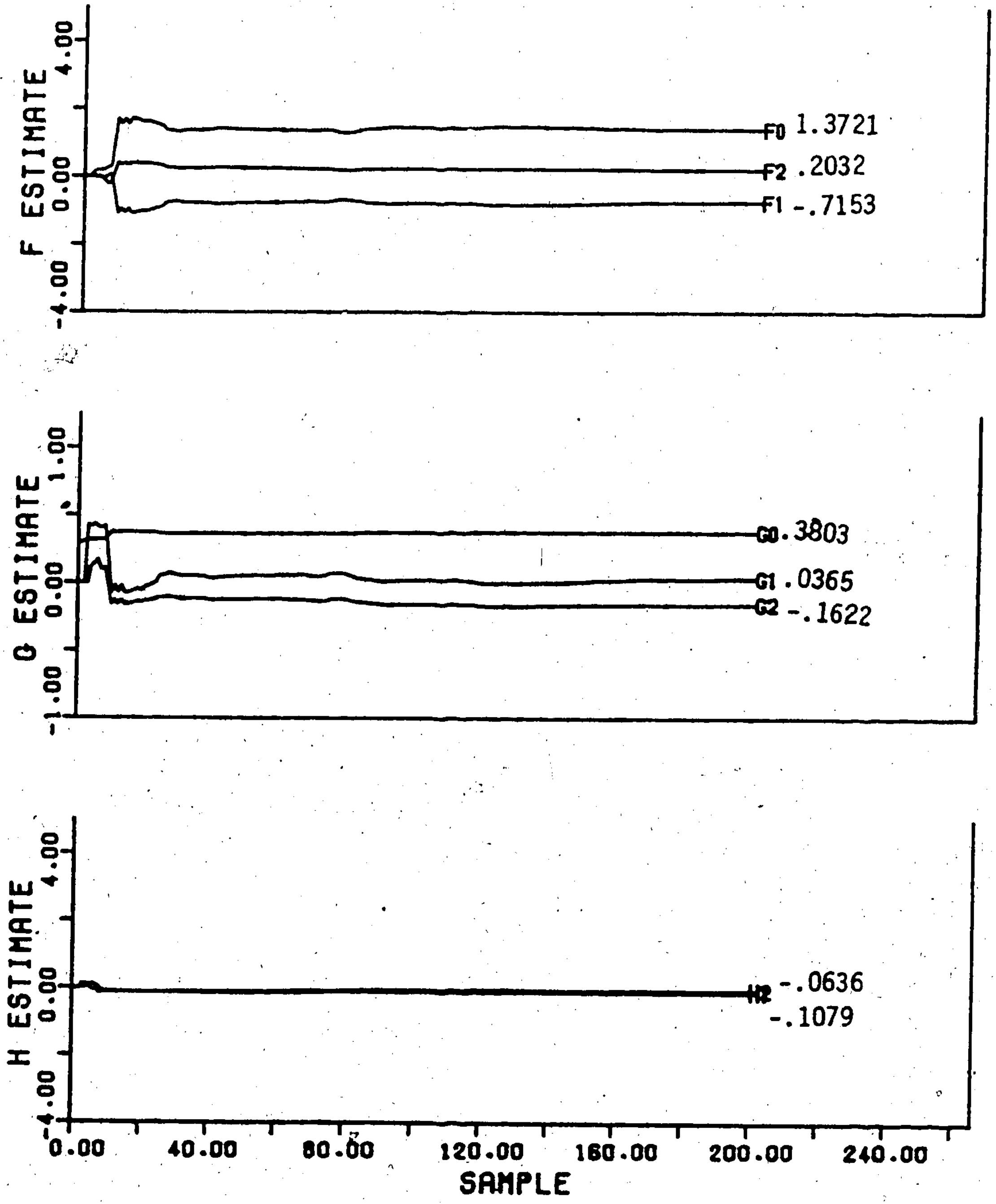


Figure 4.41 Parameter Estimates for Step Change in Setpoint using the Recursive Least Squares Estimator with Q and R weighting

have converged but the H estimates appear to be changing slightly toward the true values. All parameter estimates have converged for the sawtooth function case although the estimates were more erratic for the first 100 sample instants than for the square wave setpoint change. The G and H estimates oscillate substantially at the start of the test before converging. The H estimates are mirror images of each other which is not evident for the square wave change in setpoint. The parameter estimate behavior for the step change in setpoint is almost identical to that for the square wave change in setpoint, but the estimates are both negative which is worse than the estimates for the square wave change.

The initial changes in the parameter estimate for the square wave and constant changes in setpoints are reduced by 33% for the F estimates while the negative changes observed at the 10th sample instant for the g_1, g_2 and h_1 estimates are reduced by 50% when compared with the minimum variance control simulation results. For the sawtooth setpoint change the Q and R weighting reduced the fluctuations in the H estimates by 50% which resulted in better parameter estimates. The parameter estimates using the recursive square root and recursive upper diagonal factorization algorithms were virtually identical. The estimates shown in Figures 4.42, 4.43 and 4.44 obtained using the recursive square root algorithm display the same general trends as the recursive least squares simulation results except the

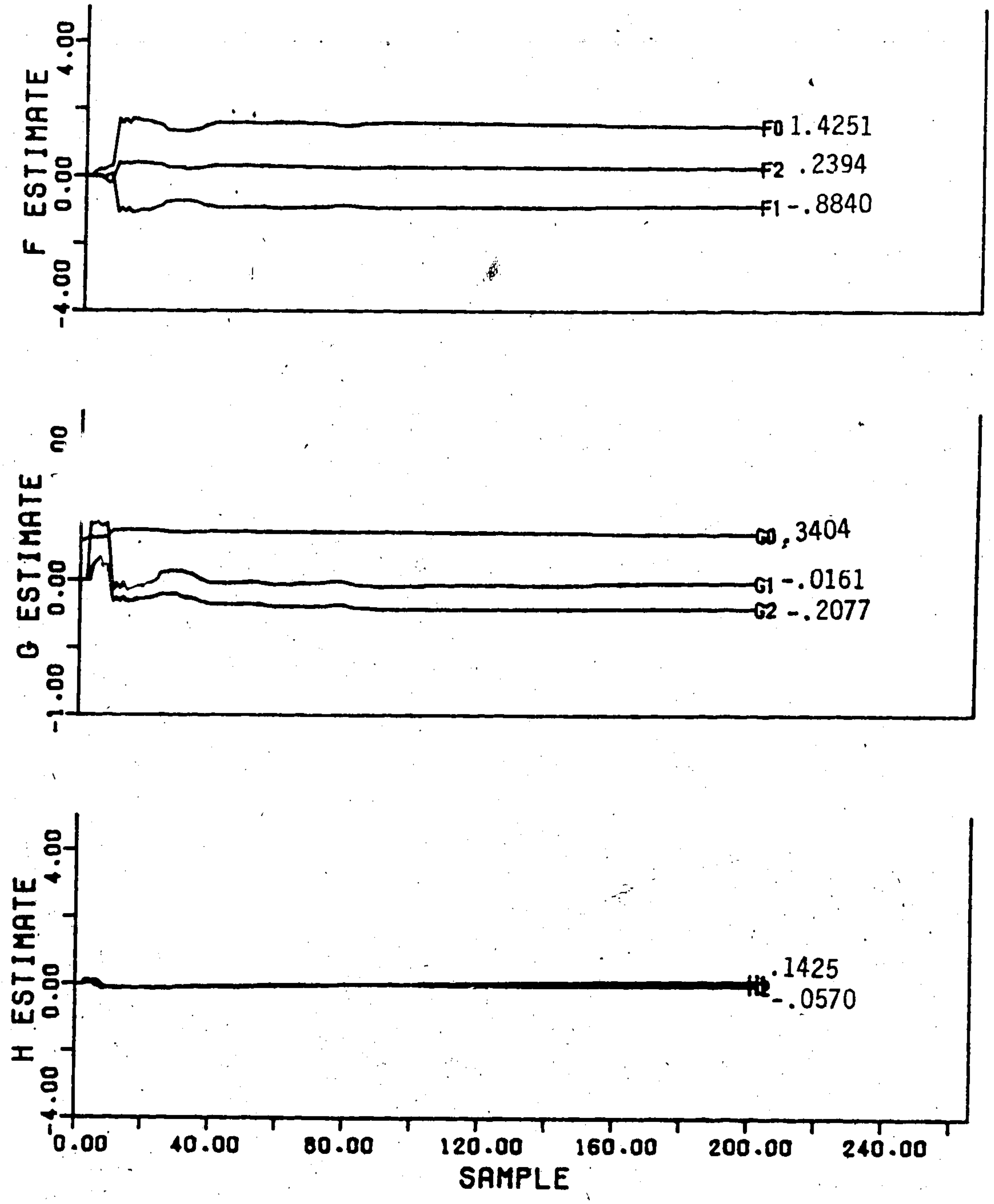


Figure 4.42 Parameter Estimates for the Square Wave Change in Setpoint using [illegible] Recursive Square Root Estimator with Q and R Weighting

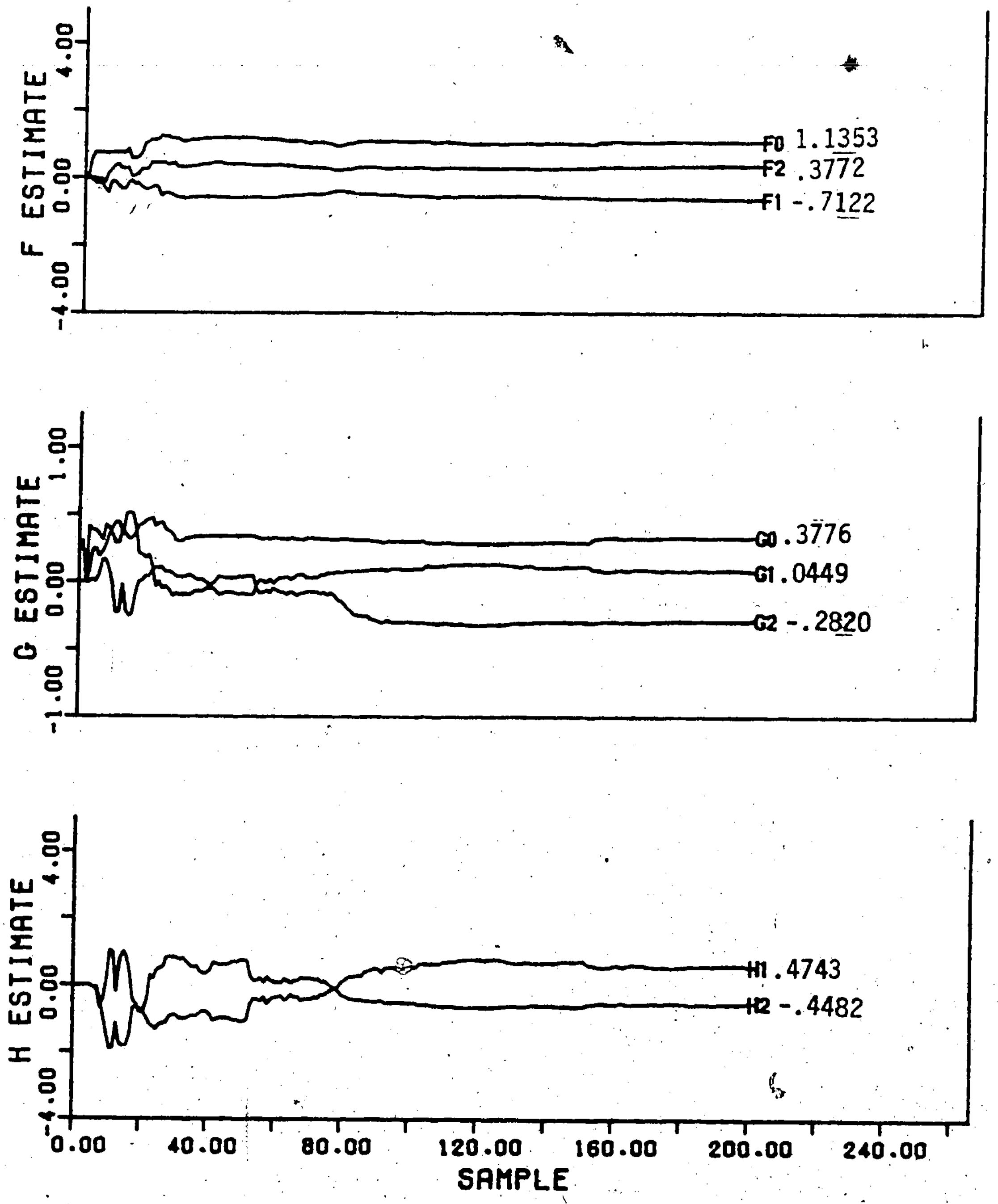


Figure 4.43 Parameter Estimates for the Sawtooth Function Change in Setpoint using the Recursive Square Root Estimator with Q and R Weighting

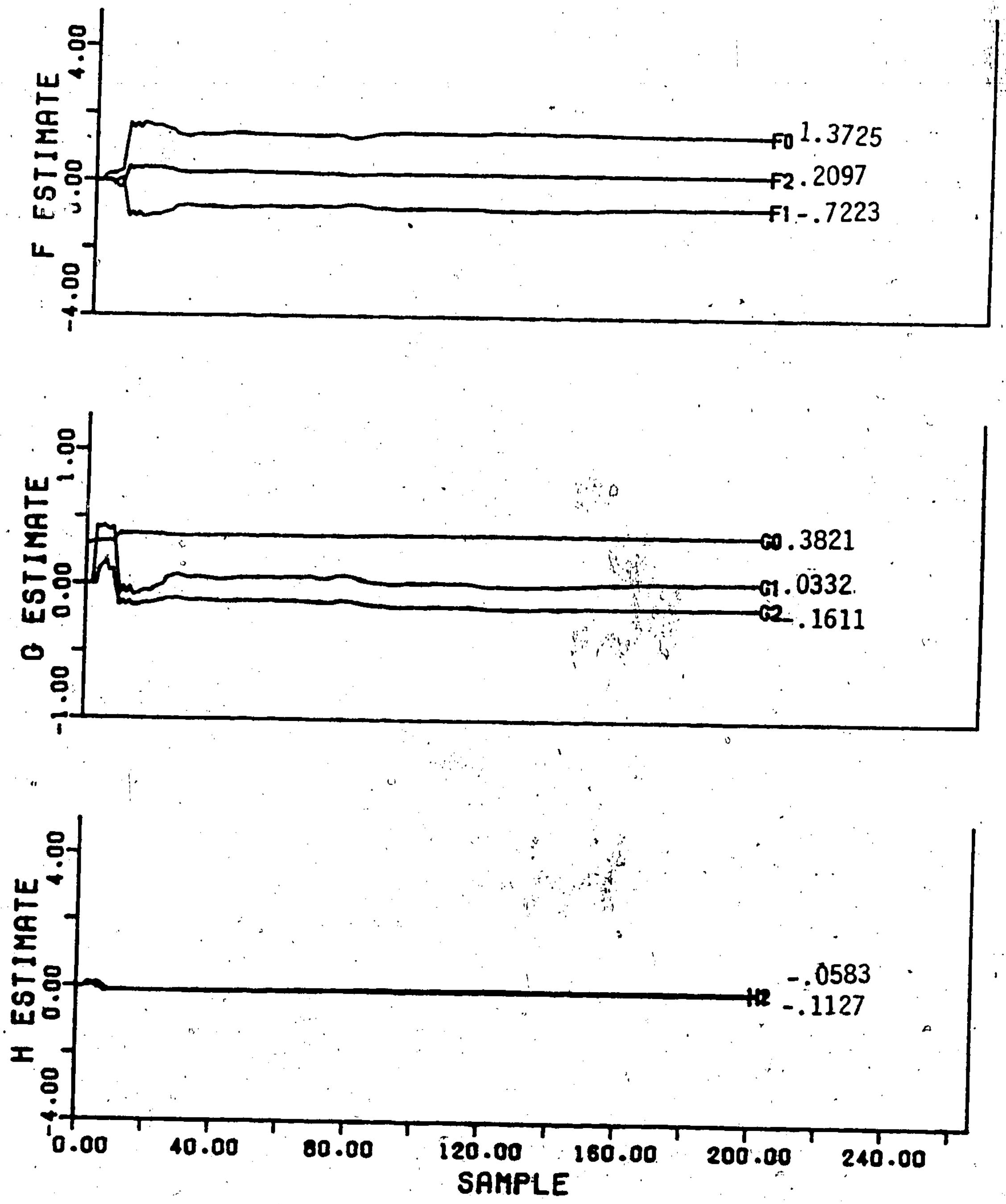


Figure 4.44 Parameter Estimates for the Step Change in Setpoint using the Recursive Square Root Estimator with Q and R Weighting

estimated values are slightly different.

The R and Q weighting were also used with the recursive learning estimator giving rise to the control performance shown in Figures 4.45, 4.46 and 4.47. As it can be seen, the setpoint tracking is very poor for the square wave setpoint change especially at the beginning of the change although it has improved before the next change in setpoint occurs. The control action is erratic with large fluctuations. The tracking for the sawtooth function setpoint change is acceptable, however, on the increasing portion of the change the output tends to be below the setpoint. The setpoint following performance for a step change in setpoint is good but the output remains slightly below the desired setpoint. The control action for both the sawtooth function change and step change in setpoint shows more fluctuation but the changes are not as large as observed for minimum variance control.

The addition of R and Q weighting reduced the control effort by an average of 50% and reduced the sum of predicted errors and average of 28% from that obtained from the minimum variance control tests. The control effort reduction using the recursive learning estimator was more significant than for recursive least squares, recursive square root and recursive upper diagonal factorization. The square wave output had a reduction of 18.5% in the overshoot observed at the start of the change in setpoint thus reducing the initial output error and consequently the sum of predicted

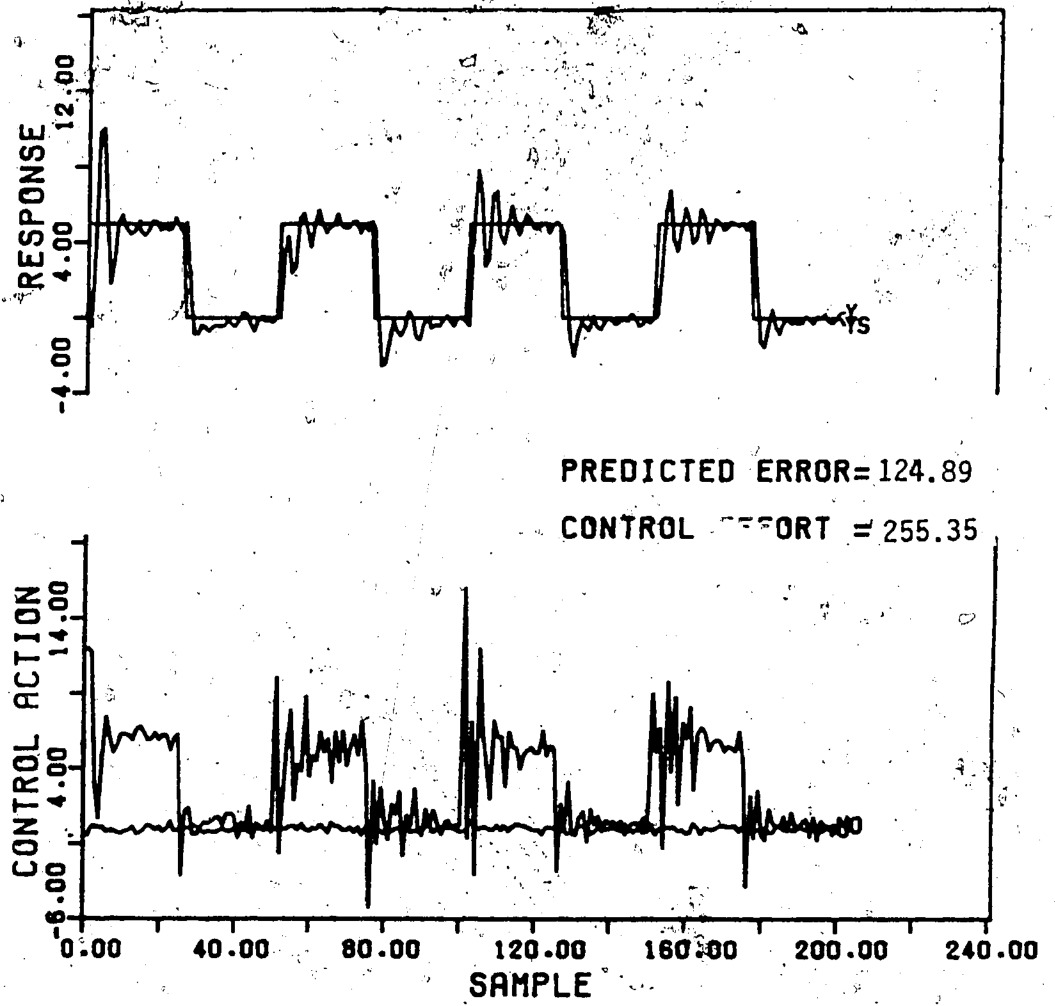


Figure 4.45 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Square Wave Change in Setpoint

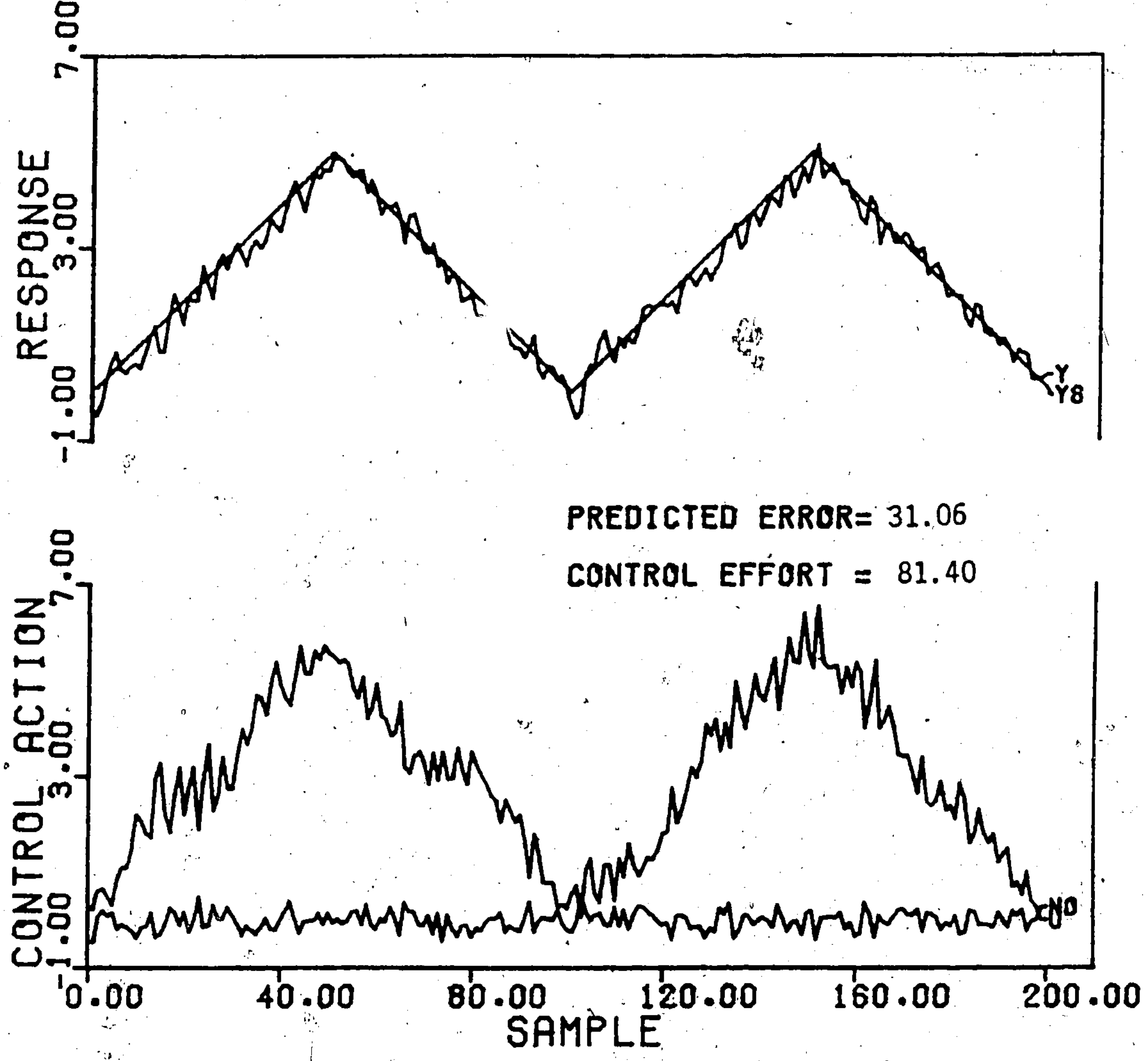


Figure 4.46 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Sawtooth Function Change in Setpoint

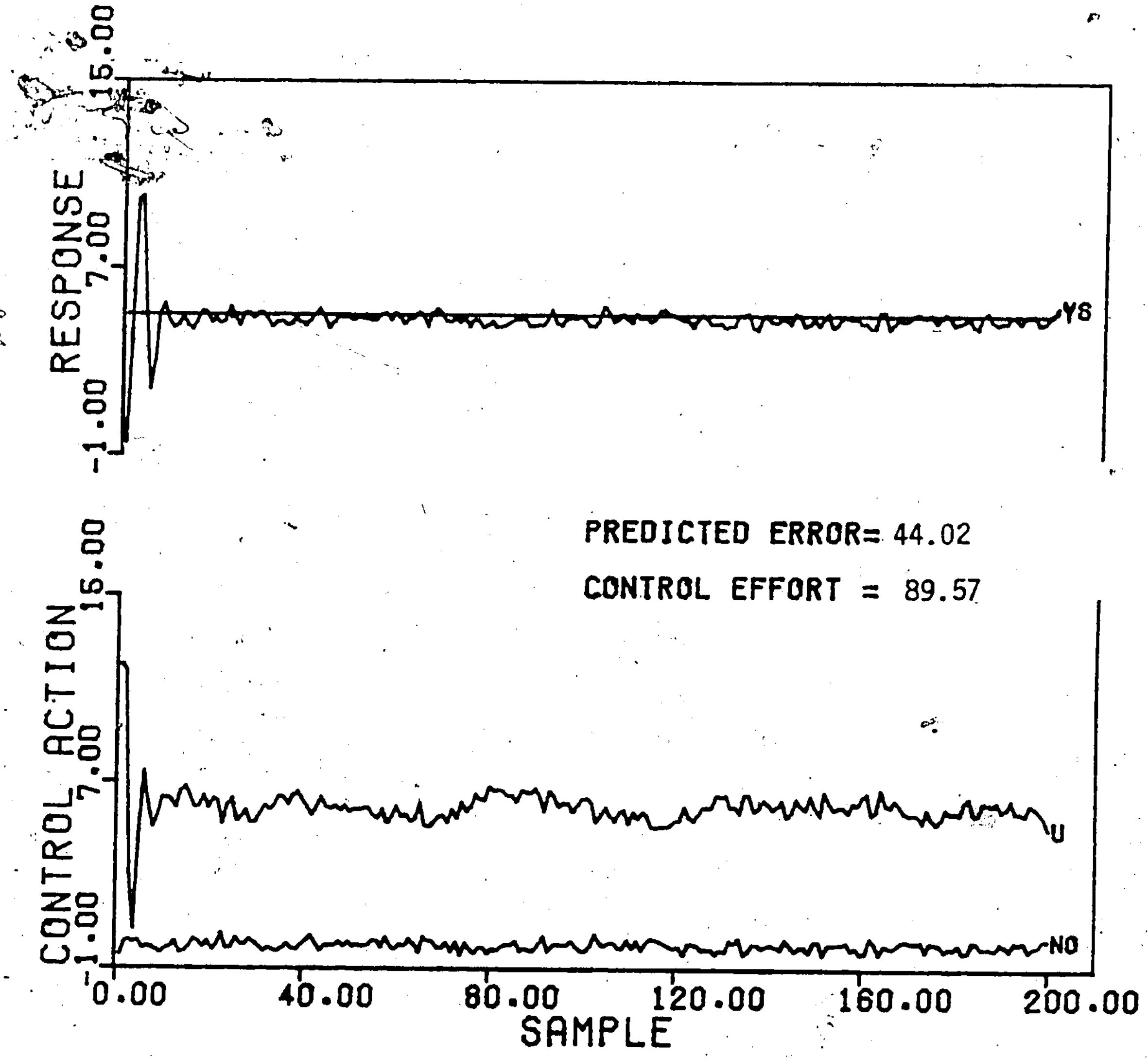


Figure 4.47 Control Response with Q and R Weighting using the Recursive Learning Estimator for the Step Change in Setpoint

errors. The setpoint tracking is also improved, however, it is still not acceptable. The setpoint tracking performance deteriorated slightly with weighting for the sawtooth function change in setpoint as the sum of predicted errors increased 23%. The step change in setpoint also showed a 21% reduction in the initial output overshoot with Q and R weighting.

The parameter estimates obtained for the three different setpoint changes using the recursive learning identification technique are given in Figures 4.48, 4.49 and 4.50. The F and G estimates for the square wave change setpoint have not converged and the G estimates behave quite erratically. The estimates for h_1 and h_2 are both negative when only h_2 should be negative.

In the case of the sawtooth function change in setpoint less erratic parameter estimates resulted than for the square wave change but the G estimates are still erratic and the H estimates are near zero and are not approaching the true values. The F estimates although showing convergence, have not converged to the true values.

The parameter estimates for the step change in setpoint are relatively constant although the G estimates still have slight oscillations with the h_1 and h_2 estimates of zero as for the sawtooth function setpoint change.

Compared to minimum variance control the parameter estimates for the square wave change in setpoint behave similarly but for the sawtooth function changes and the step

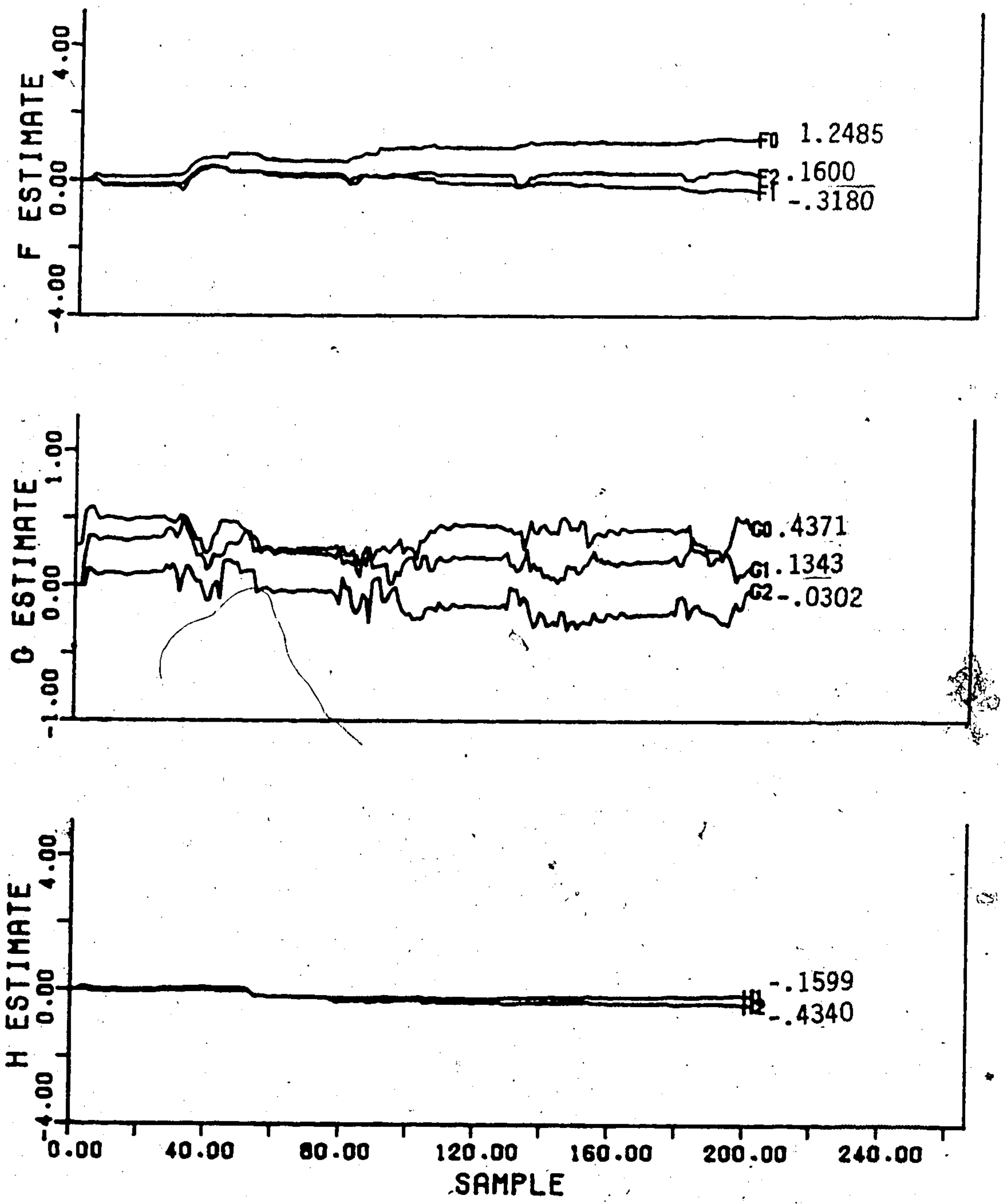


Figure 4.48 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Square Wave Change in Setpoint

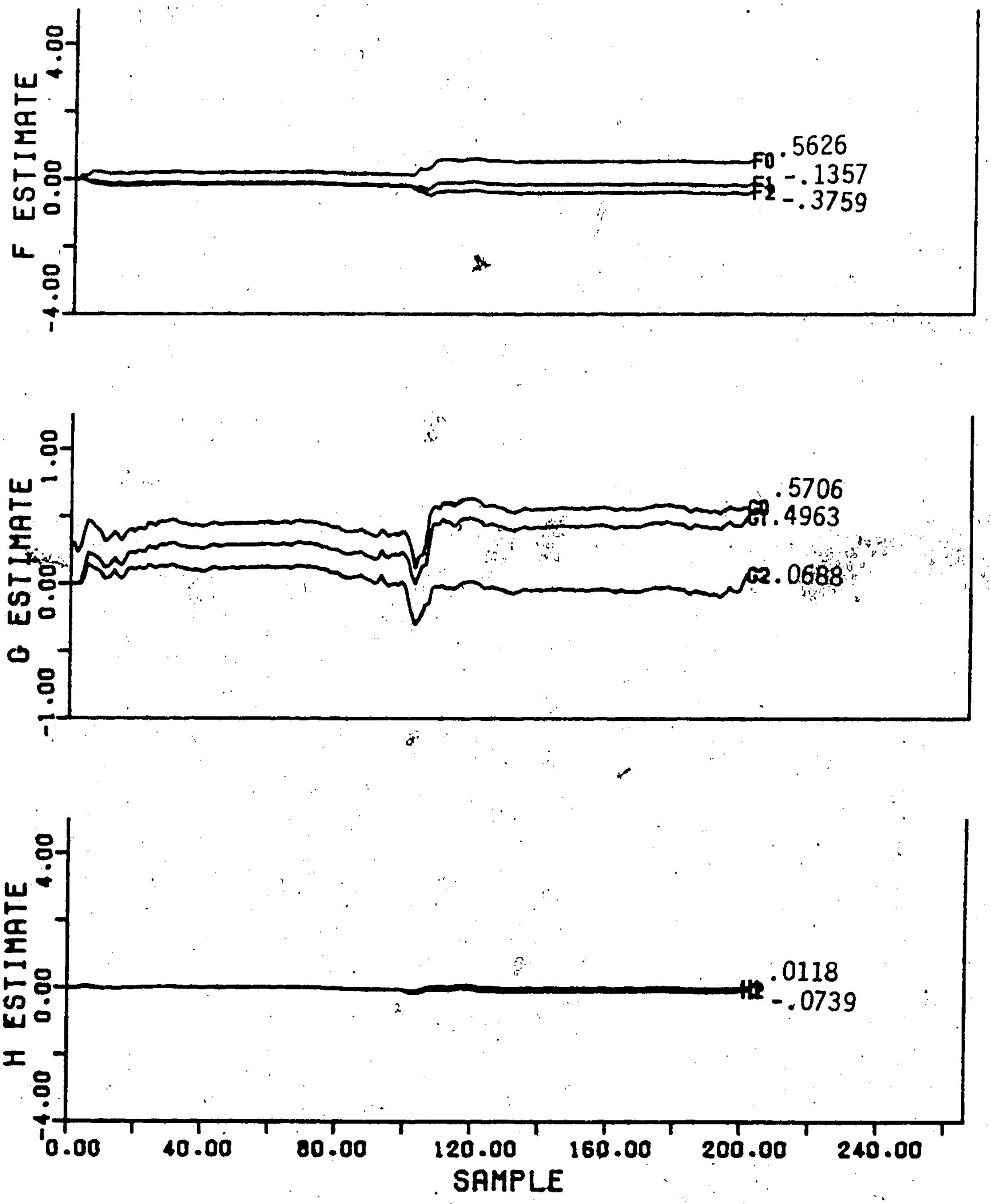


Figure 4.49 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Sawtooth Function Change in Setpoint

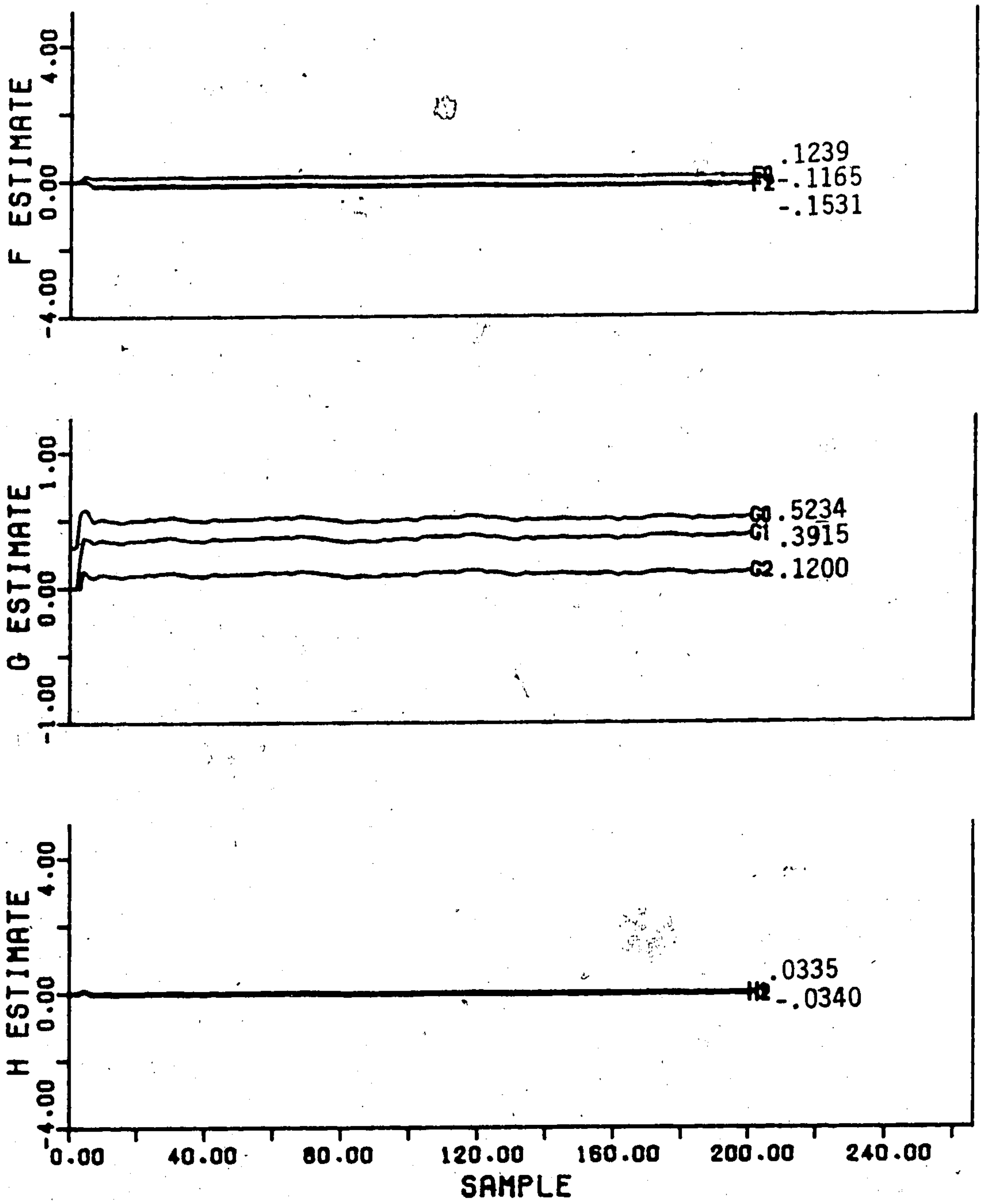


Figure 4.50 Parameter Estimates for the Recursive Learning Estimator with Q and R Weighting for the Step Change in Setpoint

setpoint changes some differences are evident. For the sawtooth setpoint the F estimates have larger final values with the R and Q weighting as well the G estimates oscillate more especially at the 100th sample instant where the setpoint has started to increase. For the step change in setpoint the only noticeable difference is that the parameter estimates are more constant with Q and R weighting.

The cost function weighting was used with the recursive maximum likelihood estimator to determine if the control performance could be improved from the minimum variance control simulation. The control performance for the square wave change in setpoint, as can be seen in Figure 4.51 shows that the tracking is poor as the output does not stay at the maximum setpoint value and cannot maintain a zero setpoint.

The control performance with the weighting is a slight improvement over the minimum variance control after the first 25 sample instants causing a 1.4% decrease in the sum of predicted errors. The output is closer at the maximum value of the setpoint but not at the zero value of the setpoint. There is also a reduction of 100% in the initial output overshoot for the weighted case than achieved under minimum variance control.

The control performance for a sawtooth function change in setpoint shown in Figure 4.52, shows poor setpoint tracking for the entire simulation. The cost function weighting did not significantly change the output

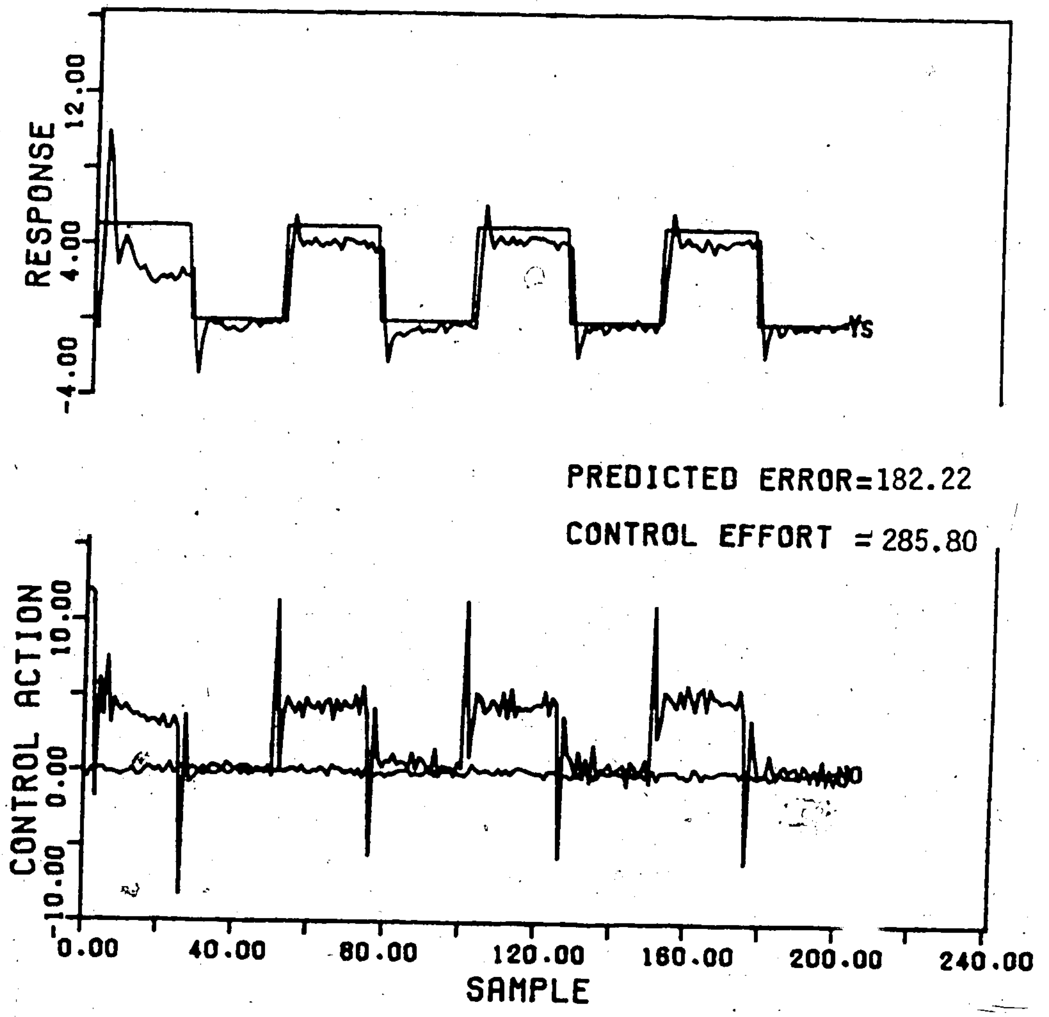


Figure 4.51 Control Response with P and R Weighting using the Recursive Maximum Likelihood Estimator for the Square Wave Change in Setpoint

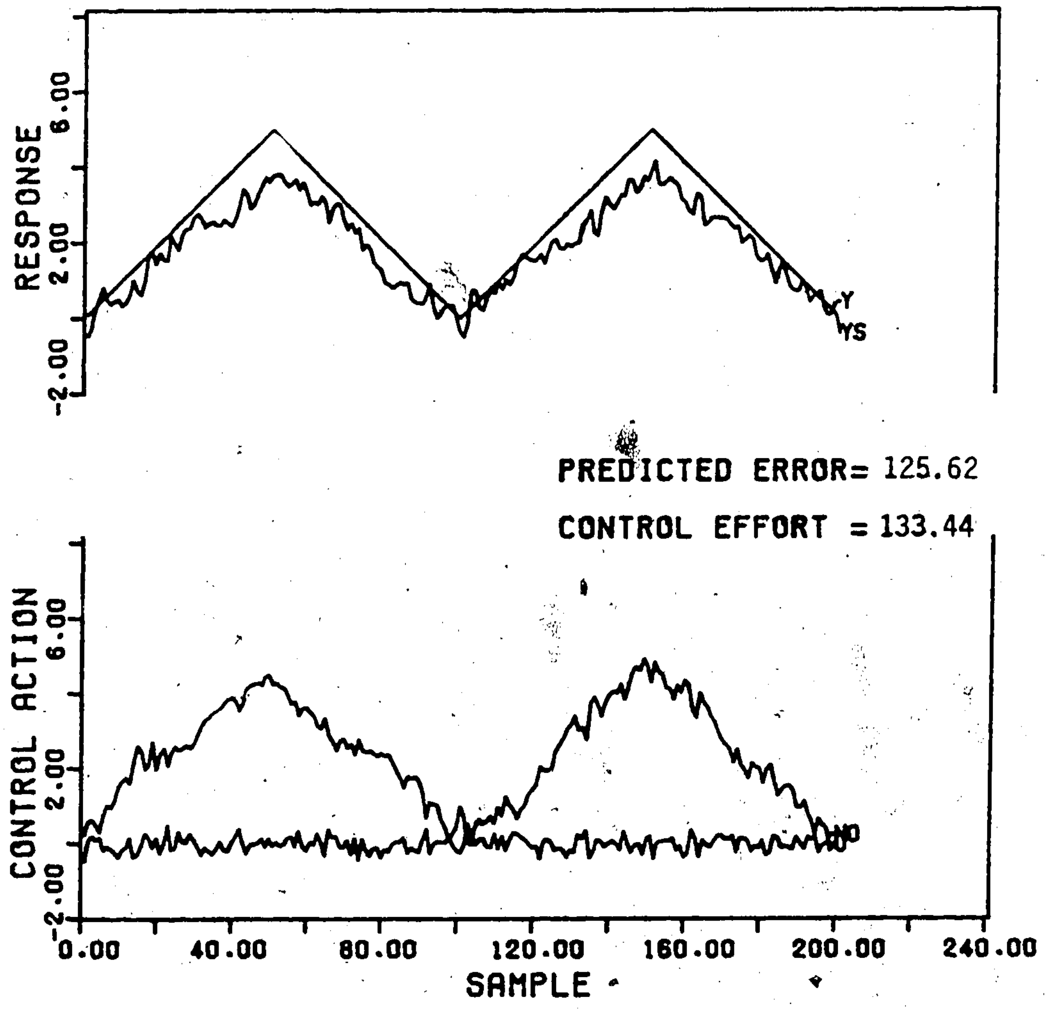


Figure 4.52 Control Response with Q and R Weighting using the Recursive Maximum Likelihood Estimator for the Sawtooth Function Change in Setpoint

performance reducing the sum of predicted errors for this setpoint by only 15% except that the control action is now reduced 28% compared to the minimum variance control action.

The results for the step change in setpoint as seen in Figure 4.53 show the output unable to attain the setpoint after the initial output overshoot. The addition of weighting for this step change in setpoint resulted in poorer control performance as the output is now 50% further from the setpoint than for the minimum variance case. The only benefit of the weighting was the reduction of 46% in the initial output overshoot.

For the square wave setpoint change the control effort was reduced 68% with the Q and R weighting while the control effort was reduced by 15% for the sawtooth function change in setpoint but for the step change in setpoint, an increase of about 40% in both control effort and sum of predicted errors which is expected due to the deterioration in setpoint tracking peformance.

The parameter estimates for the square wave change in setpoint using Q and R weighting shown in Figure 4.54 are constant and appeared to have converged. The F and H parameter estimates increased very slowly while the g_1 and g_2 show large initial changes. The g_0 estimate does not change noticeably and the h_1 and h_2 estimates remain near zero with incorrect signs. Compared to minimum variance contol the weighting only caused a change in the H estimate behavior from the 5th to the 30th sample instant. With Q and

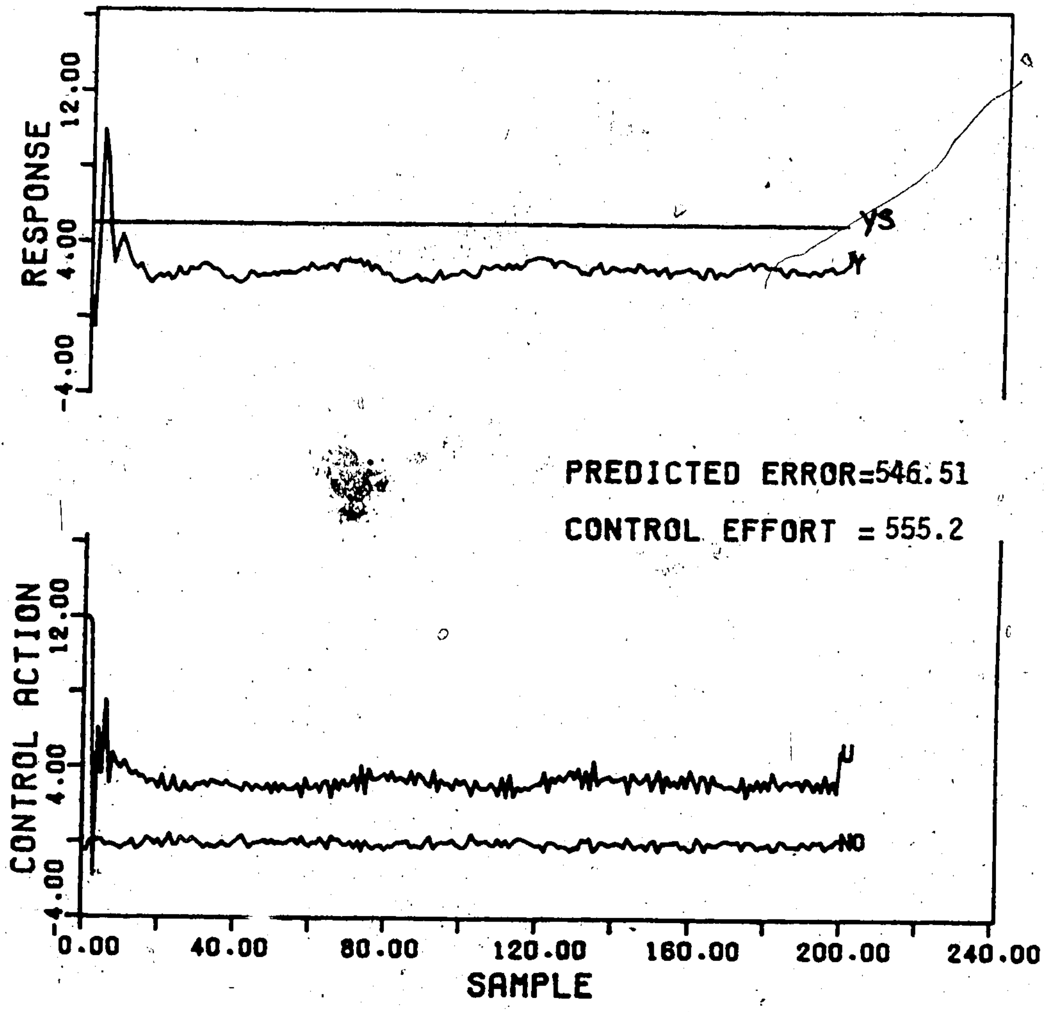


Figure 4.53 Control Response with Q and R Weighting using the Recursive Maximum Likelihood Estimator for the Step Change in Setpoint

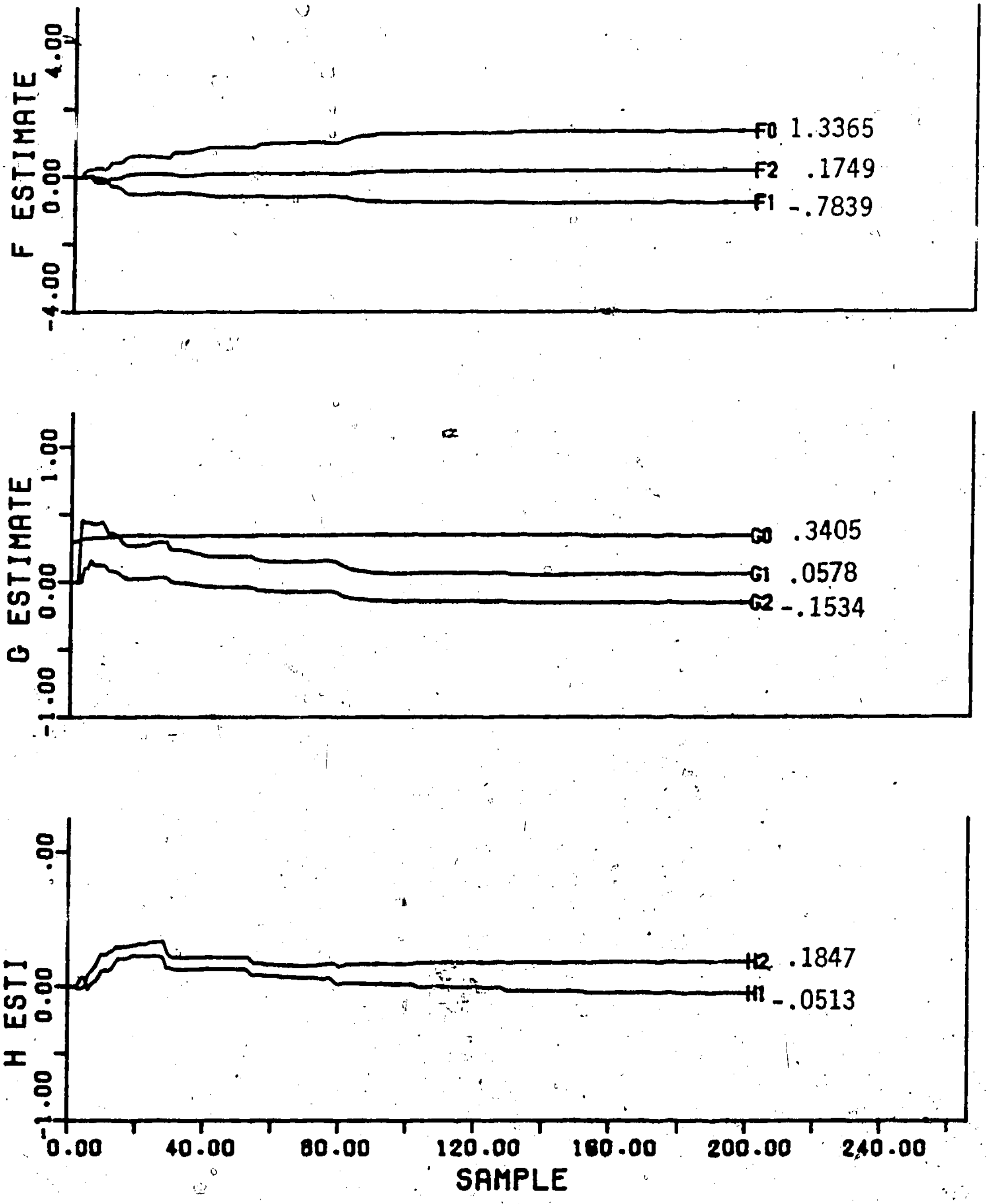


Figure 4.54 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Square Wave Change in Setpoint

R weighting the h_1 and h_2 parameter estimates grow larger during that interval and never become equal as is observed in Figure 4.30 for the minimum variance control. The estimate values yielded after 200 sample instants for both minimum variance control and Q and R weighting simulations are within ±10% of each other.

The parameter estimates for the R and Q weighted sawtooth function change in setpoint are given in Figure 4.55. It can be seen that the F estimates increase gradually but never approach the true values. The f_1 and f_2 estimates remain close to zero and f_2 converges to a value with the incorrect sign. The G estimates have converged but the initial behavior is erratic. The H estimates have also converged but h_2 is still close to zero. This parameter estimate behavior using weighting is similar to that observed for the minimum variance control case. The only differences being that the f_2 estimate has changed sign fo the simulation with Q and R weighting, there are larger changes in the G estimates resulting in larger estimates after 200 sample instants. The F and H estimates display similar adaption behavior for both minimum variance control and Q and R weighting.

The step change in setpoint resulted in constant parameter estimates. By comparing Figures 4.56 and 4.54 it can be seen that the F parameter estimates follow a similar adaption pattern as the square wave change in setpoint except that the parameter estimate changes for the step

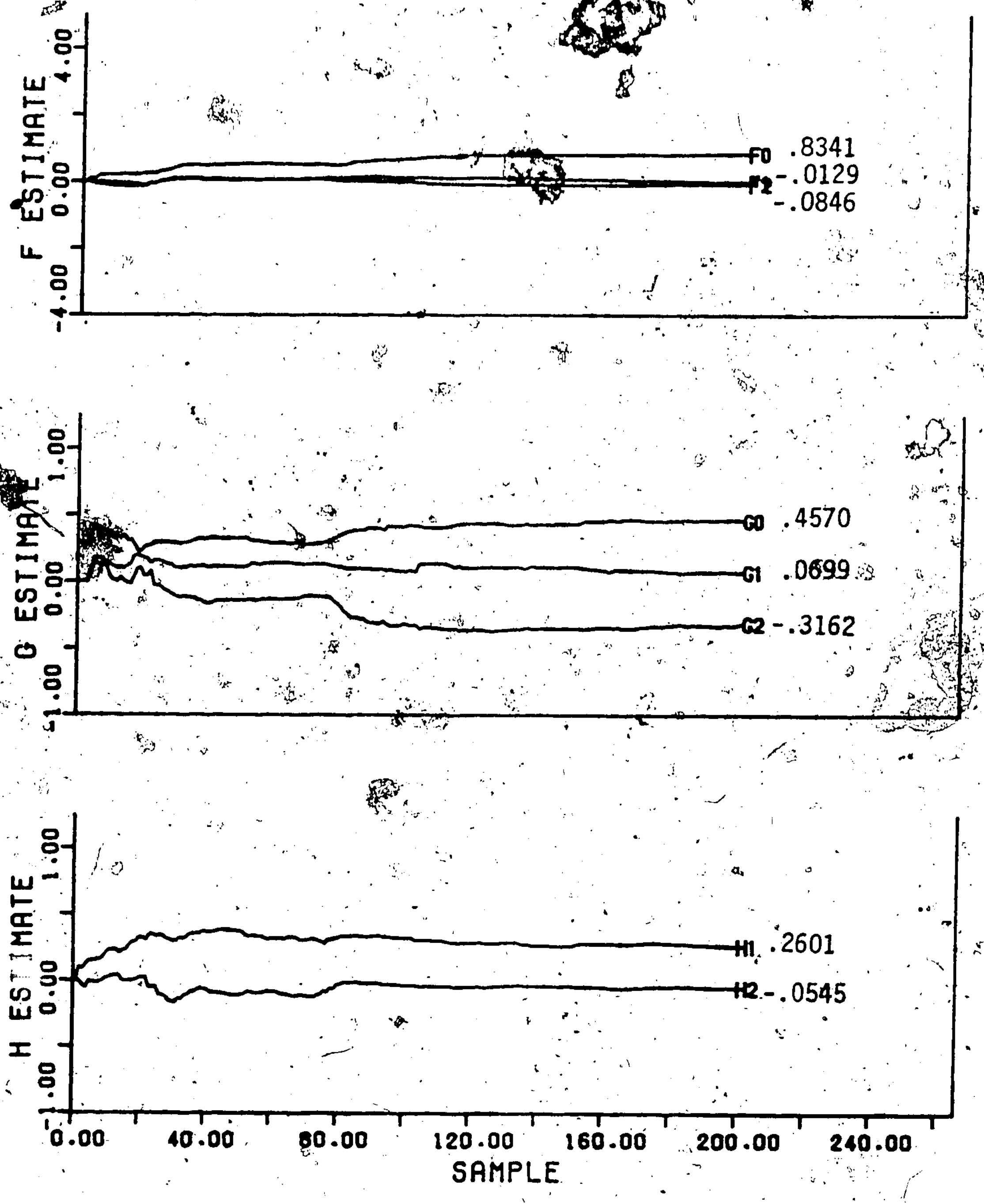


Figure 4.55 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Sawtooth Function Change in Setpoint

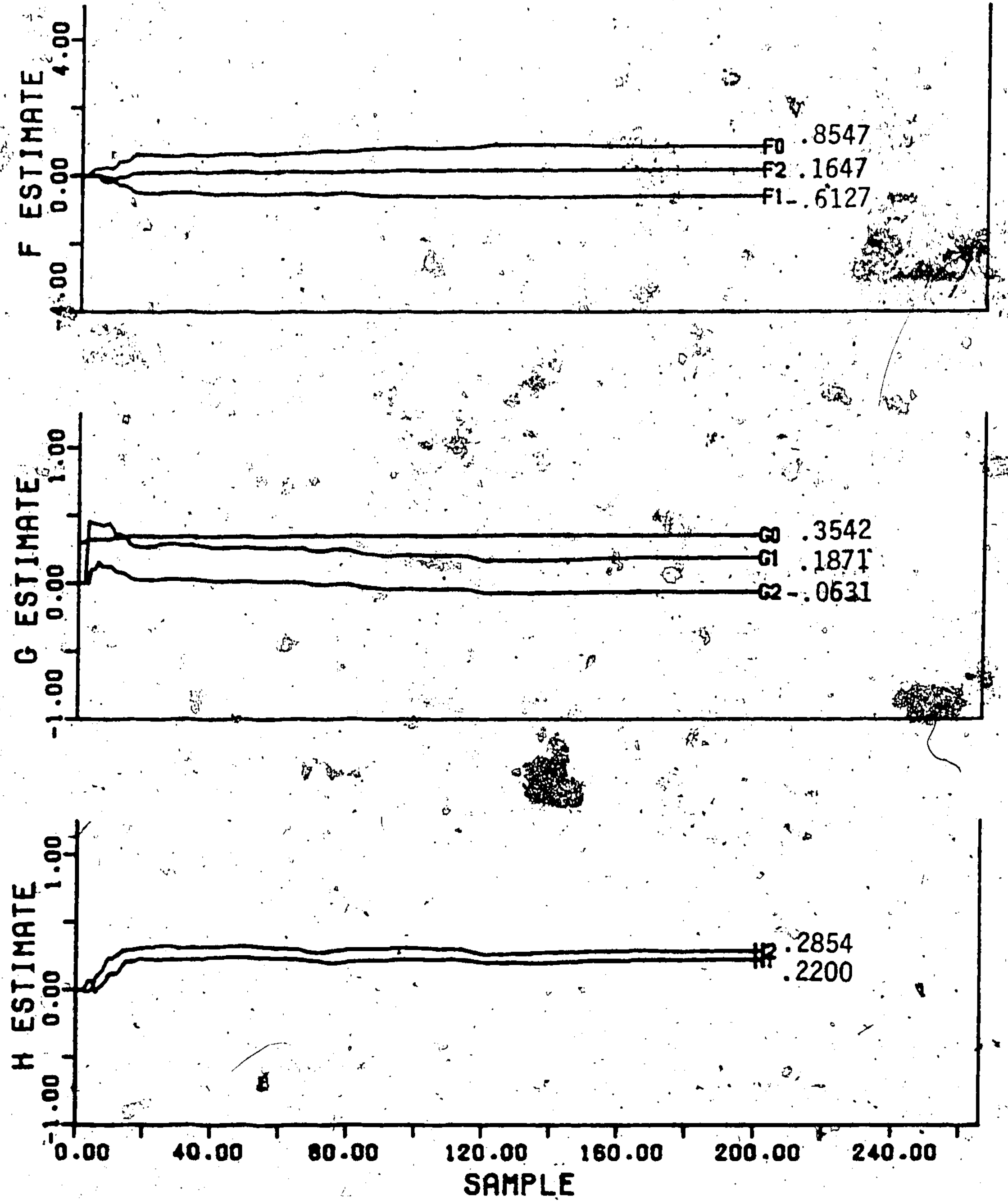


Figure 4.56 Parameter Estimates for the Recursive Maximum Likelihood Estimator with Q and R Weighting for the Step Change in Setpoint

change in setpoint are not as noticeable. However, the f_0 and f_1 estimate values after [illegible] sample instants are 20 to 40% lower for the step change. The G estimates have converged but only g_0 is within ±4% of the true value, the g_1 and g_2 estimates are over ±70% from the true values and g_1 has the incorrect sign. The h_1 and h_2 estimates have converged but h_2 has the incorrect sign while h_1 is 72% from the true value.

The resulting parameter estimates for a step change from the Q and R weighting test are similar to those yielded by minimum variance control. The adaption patterns were similar with only an average of ±18% difference between the parameter estimates obtained after 200 sample instants. Use of the recursive maximum likelihood estimator did not yield suitable estimates with or without weighting to satisfactorily track the setpoint.

By comparing the results from the simulations for all estimators using Q and R weighting with the minimum variance control simulations the following conclusions were made. The use of changes in R and Q weighting does reduce the control effort for all estimators, however, the control performance is not improved for the step change in setpoint using the recursive maximum likelihood estimator and setpoint tracking for the other setpoint changes is not satisfactory. Based on the reduction of the sum of predicted errors and control effort it can be stated that the recursive least squares, recursive square root and recursive upper diagonal factorization and recursive learning estimators all exhibit better control performance than just using minimum variance control for less control action for the three different types of changes. However, the sum of predicted errors using the recursive learning estimator is still 15% larger for the square wave setpoint change and 47% larger for the step change in setpoint than the recursive least squares, recursive square root and recursive upper diagonal factorization estimators.

The Q and R weighting reduced the initial output overshoot for the square wave and step change in setpoint cases and reduced the initial tracking error for the sawtooth function setpoint change. The parameter estimates obtained by all five estimators are more constant for the F and H estimates but caused the G estimates to oscillate for the recursive square root, recursive upper diagonal factorization and recursive learning estimator more than in the minimum variance control case.

4.4 Disturbance Rejection Testing

If a system load disturbance is not measurable the effects of the disturbance will be propagated through the system in the same manner as for a random disturbance ξ_k except that the future mean value of v_k can not be assumed to be zero. Therefore if the controller parameters are fixed at the correct values an offset due to the load disturbance will occur unless P, Q, and R are chosen to remove the offset. If the parameters are allowed to adapt to the changing system dynamics the load disturbance effects are eliminated when the controller parameter estimates change to satisfy the linear regression relationship

$$\phi_{k+d} = X_k^{t}\Theta_k + (1-C)\phi^{*}_{k+d/k} + \epsilon_{k+d} + \frac{EL}{C} z^{d-\ell} v_k \qquad (4.5)$$

At steady state, with $\phi^{*}_{k+d/k}$ set to zero the estimates, Θ, will be biased since the estimator excitation group, $(\epsilon_{k+d} + EL/C*z^{d-\ell}\, v_k)$ is correlated with the observation vector, X_k. Parameter estimation is therefore hindered as variations in v_k, l, L or C will be reflected in the controller parameters, F, G and H. There is also a possibility that these controller parameters will have system eigenvalues that give rise to unsatisfactory control performance or even instability.

In order to study the effect of unmeasurable load disturbances, an unknown load disturbance, magnitude $v_k = 5.0$ filtered through

$$L(z^{-1}) = .1z^{-1} + 0.0905z^{-2} + 0.0082z^{-3} \qquad (4.6)$$

was added to the process with the setpoint maintained constant. The disturbance was added at the 1500th sample instant so that the identification algorithm should have identified suitable parameter estimates before the load is added.

This series of simulations used conditions identical to those used for the minimum variance control tests except the value of the forgetting factor was changed. Tests were first done using a forgetting factor of 0.995, however this caused the estimates to blow up with the recursive least squares estimator even before the disturbance was added. It also resulted in the H estimates from the recursive square root and recursive upper diagonal factorization estimators to begin to rapidly increase to values above 10 by the 3000th sample instant as well as causing the F and G estimates to begin to increase but not at as fast a rate as the H estimates.

Although t[illegible]ive square root and recursive upper diagonal fact[illegible]on estimators guarantee numerical stability the parameter estimates are still inflated due to the continued data truncation caused by the forgetting factor not equal to 1.0 while little new dynamic information is made available to the estimation algorithm.

Consequently, a forgetting factor of 1.0 was used to compensate for the non-exciting constant setpoint so that there was no estimate blow up for any of the estimation algorithms.

The control performance and parameter estimates, respectively, for load disturbance rejection using the recursive least squares estimator can be seen in Figure 4.57 and 4.58, respectively. The output returns to the setpoint approximately 60 sample instants after the disturbance is added although the deviations about the setpoint are larger than before the disturbance was added and continue to grow. Although not shown, by the 2000th sample instant the control

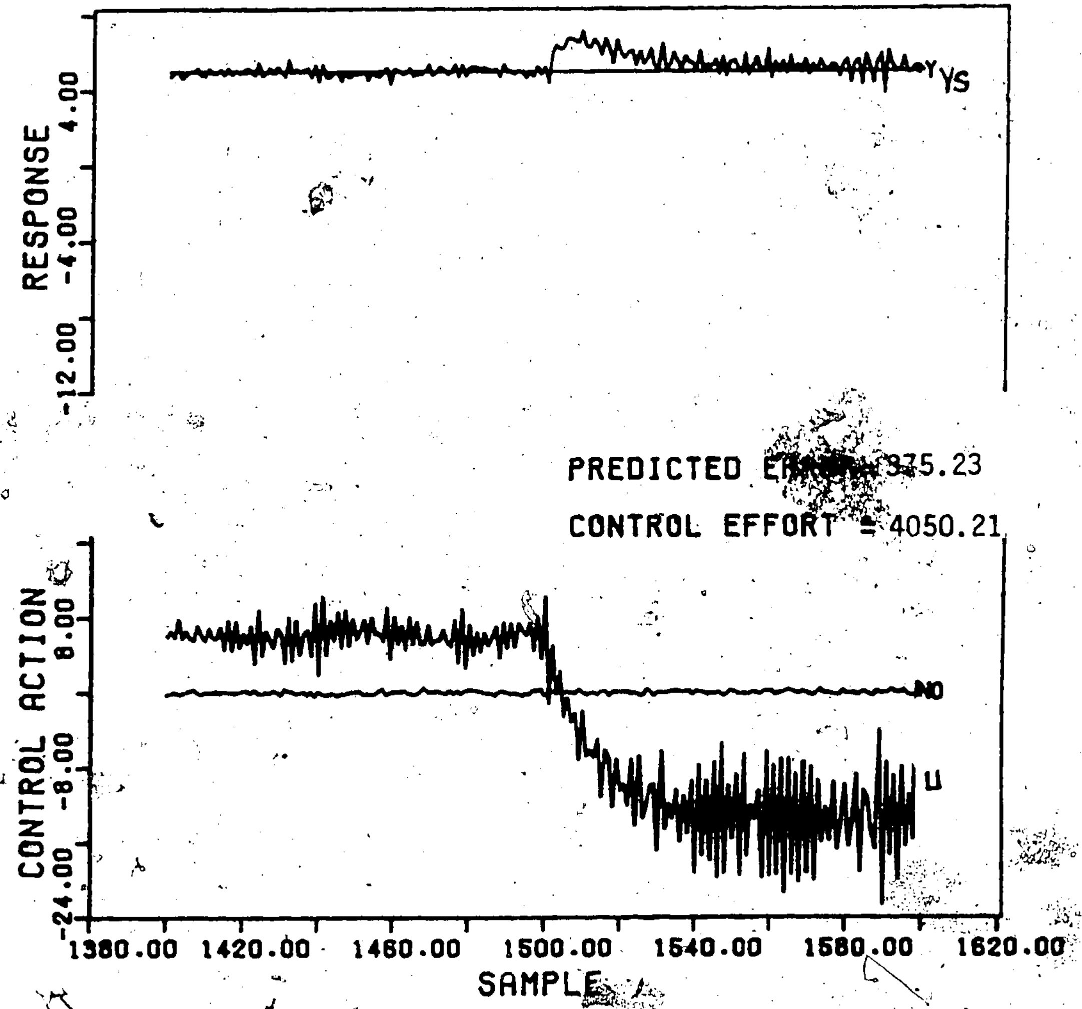


Figure 4.57 Disturbance Rejection using the Recursive Least Squares Estimator

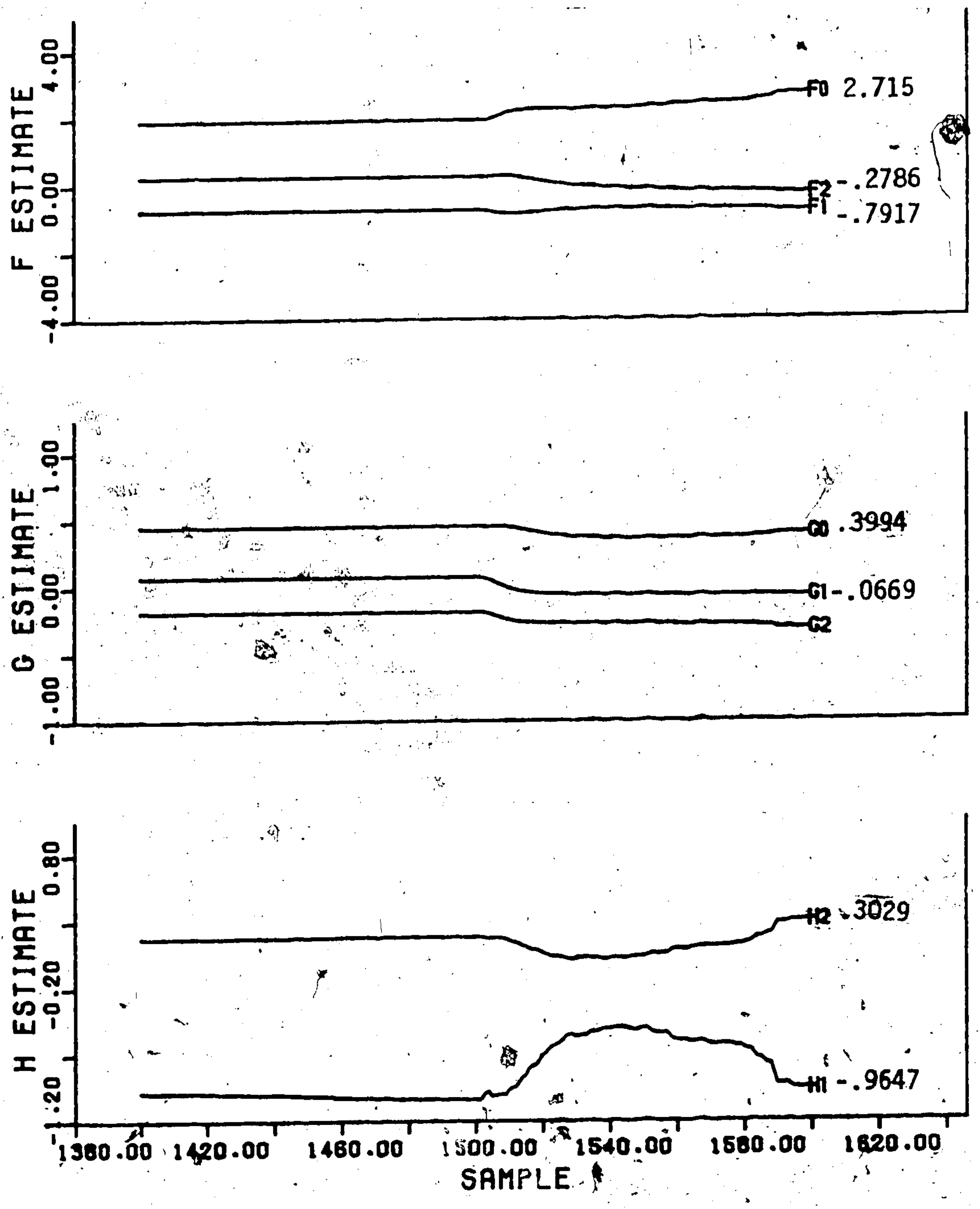


Figure 4.58 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Least Squares Estimator

action is alternating between the upper and lower limits and the output is oscillating around the setpoint. The F and G parameter estimates display a slight change at the 1500th sample instant but converge to new values by the 1600th sample instant. The H estimates display larger changes than the other parameter estimates but return to values within 10% of the values prior to the disturbance.

The recursive square root and recursive upper diagonal estimators produce almost identical results when a load disturbance is introduced to the system. The results for the recursive square root estimator given in Figure 4.59 and 4.60 also illustrate the control performance and parameter estimate behavior displayed when the recursive upper diagonal factorization estimator is used. The output shows smaller deviations from the setpoint at the sample instant when the disturbance is added than was the case for the recursive least squares estimator and the output returned to the the setpoint in only 15 sample instants. There was a reduction in control effort and in the sum of the prediction errors indicating an improvement in control performance compared to the recursive least squares simulation. The F and H parameter estimates show smaller changes than was the case for the estimates yielded by the recursive least squares estimator but the G estimates displayed the same behavior.

The recursive learning estimator did not obtain satisfactory estimates to calculate the control action that

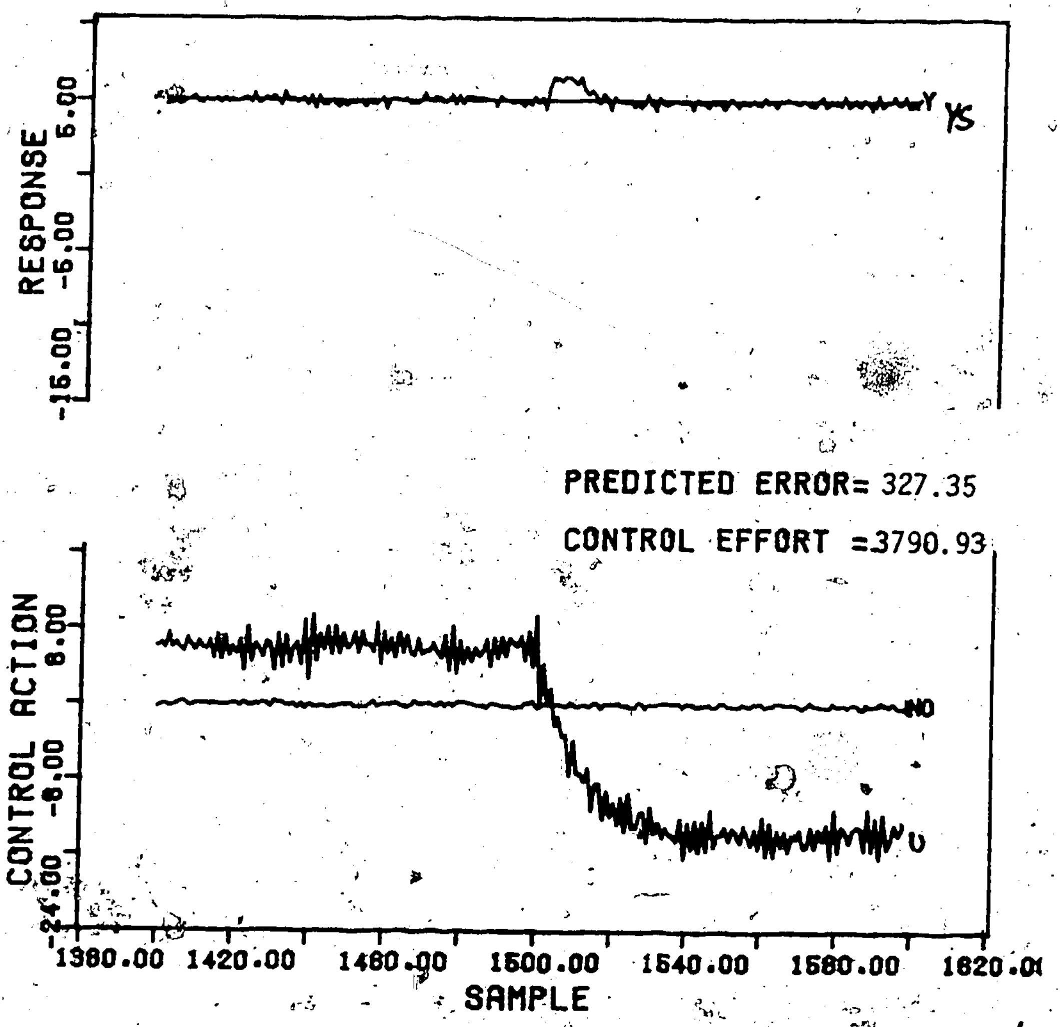


Figure 4.59 Disturbance Rejection using the Recursive Square Root Estimator

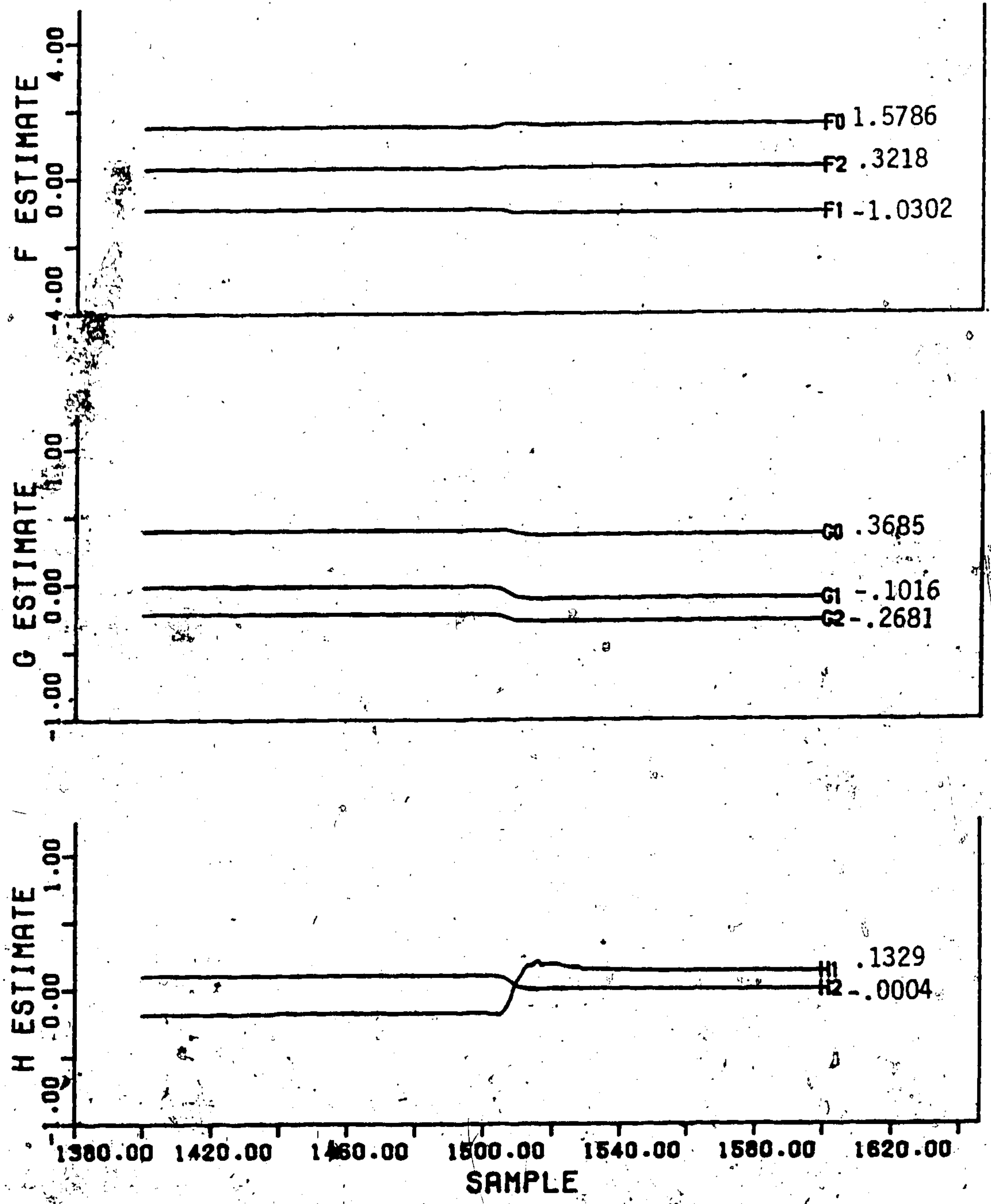


Figure 4.60 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Square Root Estimator

would result in good control performance as seen in Figure 4.61. Fluctuations around the setpoint are observed. The fluctuations continue to grow and control performance deteriorates almost to the point of instability. The G parameter estimates in Figure 4.62 are beginning to oscillate. The F and H estimates are constant and have converged to new values however, slight oscillations in the H estimates are observed at the 1590th sample instant.

The control performance to a load disturbance added to a system using the recursive maximum likeihood estimator can be seen in Figure 4.63. As seen before in the minimum variance control simulations the output is not tracking the setpoint before the disturbance is added and after the addition the output is above the setpoint. The control performance only appears to be improved because the output is closer to the setpoint but is most likely an offset caused by the load disturbances. The parameter estimates obtained by [illegible] recursive maximum likelihood estimator are displayed in [illegible] 4.64. All parameter estimates are constant before and after the disturbance. There was no noticeable change in the f_2 and h_2 values while the remaining estimates gradually converged to new values. The magnitude of all estimates decreased except for f_0 after the load disturbance was added.

The disturbance was then added for the square wave and sawtooth setpoint changes. Basically the same results as the step change in setpoint were observed, the output returned

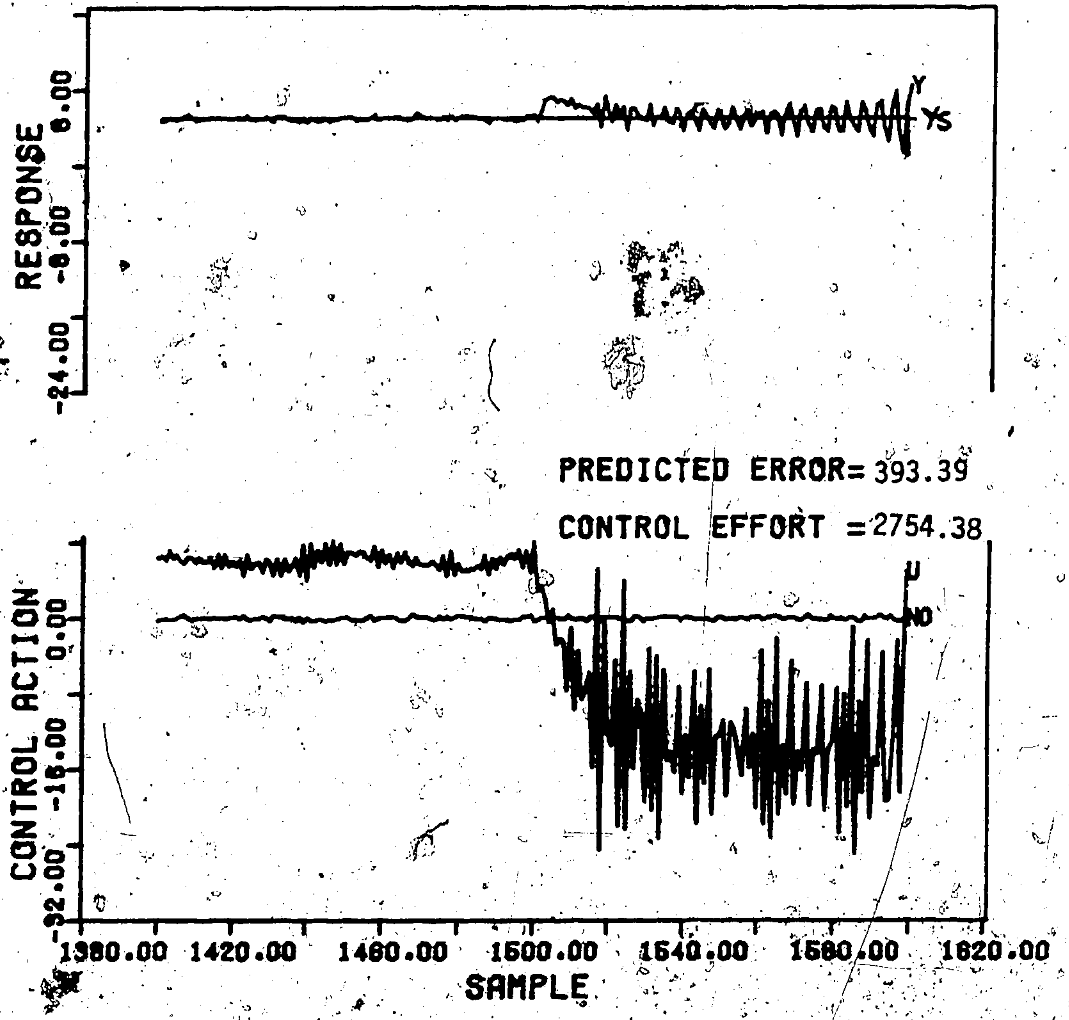


Figure 4.61 Disturbance Rejection using the Recursive Learning Estimator

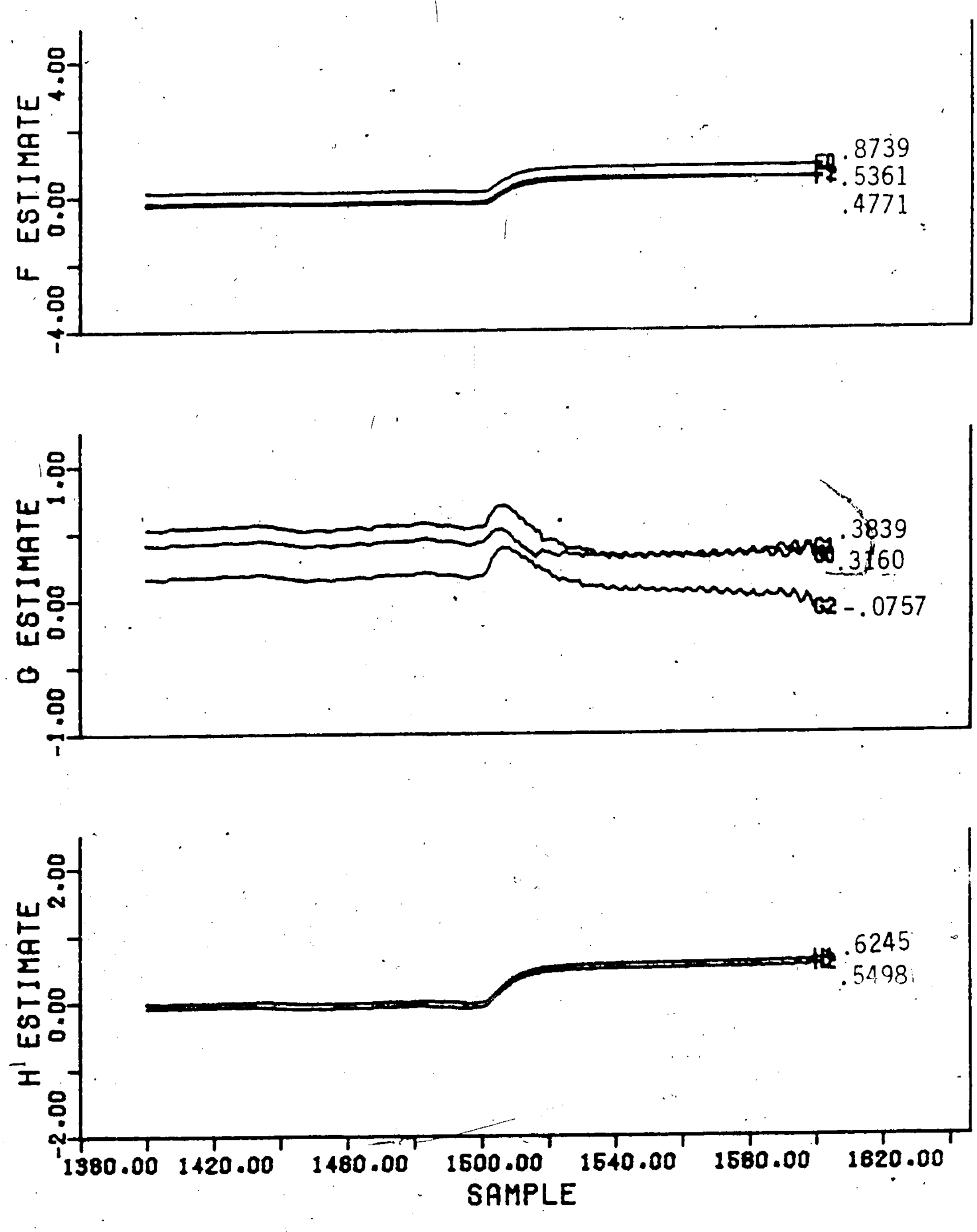


Figure 4.62 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Learning Estimator

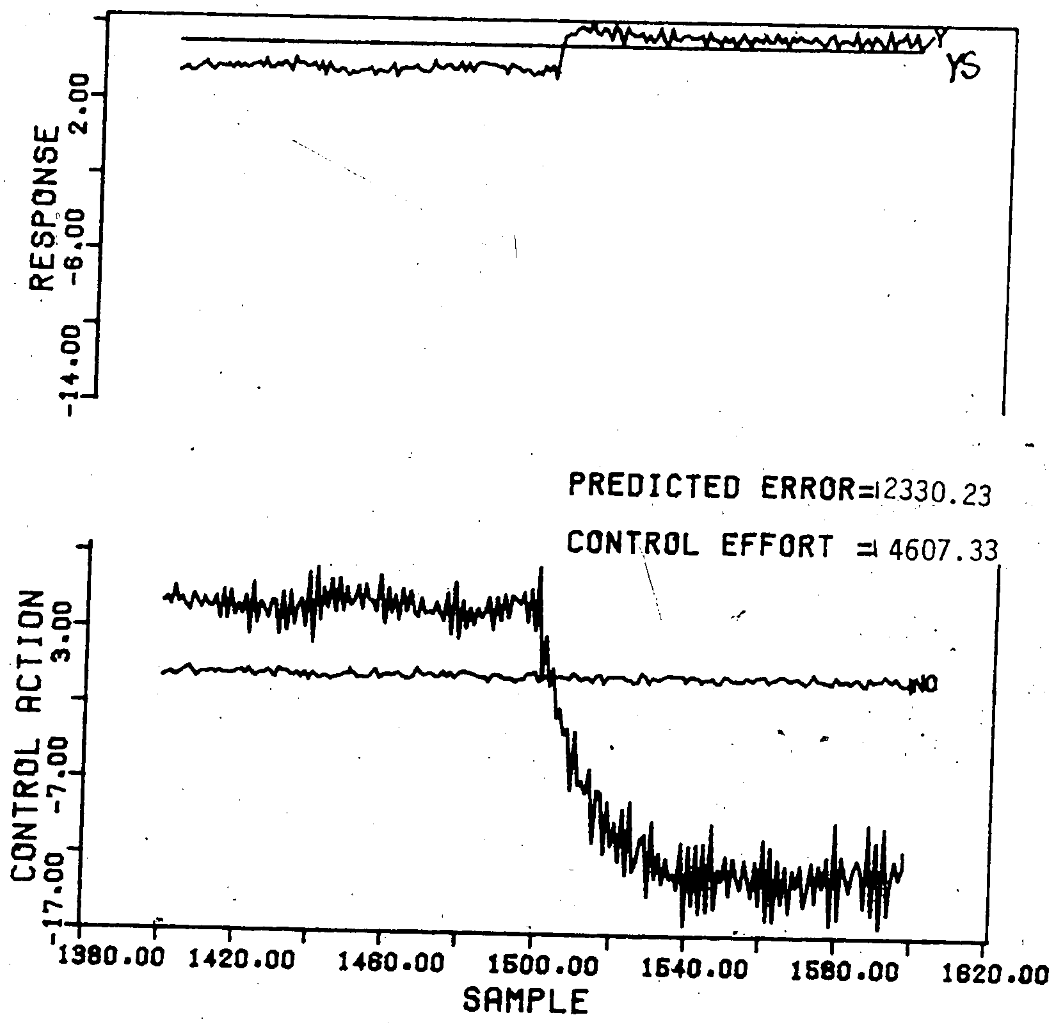


Figure 4.63 Disturbance Rejection using the Recursive Maximum Likelihood Estimator

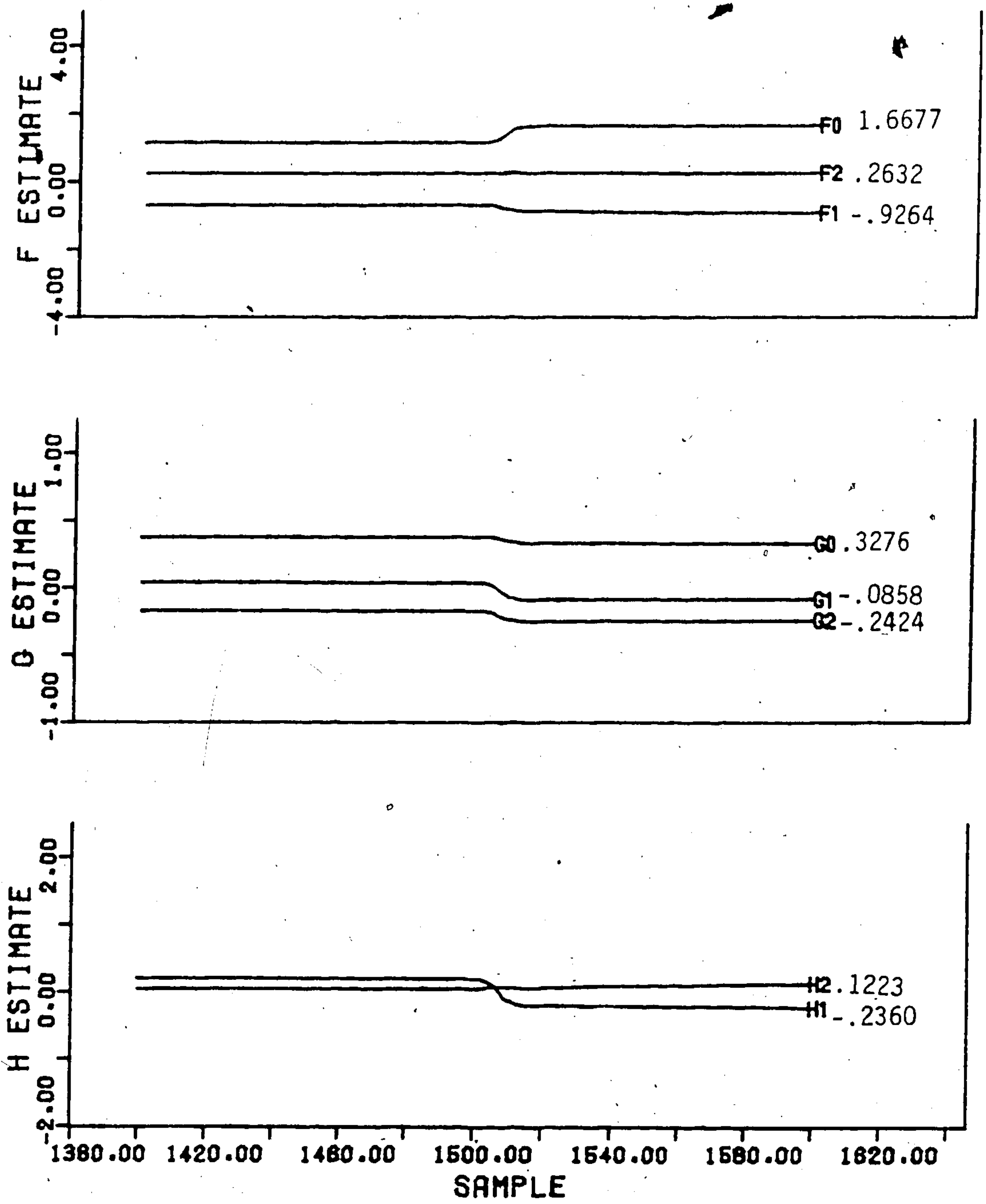


Figure 4.64 Parameter Estimates Resulting from Disturbance Rejection using the Recursive Maximum Likelihood Estimator

to the setpoint quickly and again the deviations from the setpoint were larger after the load disturbance was added. The parameter estimates were observed to have slightly bigger changes in their values at the time the disturbance was added than for the step change in setpoint but again converged by the 1600th sample instant.

The effect of Q and R weighting when dealing with a load disturbance was also studied. Using the values of Q=0.2 and R=1.2 tests were done using the same load disturbance described in the previous section using the recursive least squares and the recursive upper diagonal factorization estimators. It was found that an offset was observed for all setpoint changes and parameter identification methods tested. This is because the load disturbance will affect the k-step-ahead system output via the function $z^{k-\ell}LQ$ unless Q=0 or if Q is chosen such that its inverse is made to exhibit integral action at low frequencies.

In addition to the removal of parameter estimate blow up, a forgetting factor of 1.0 reduced the change displayed by the estimates at the time the load disturbance is added for all estimators except the recursive maximum likelihood estimator as it does not have a forgetting factor. The parameter estimates also exhibited different behavior for the two different forgetting factors. The f_0 and f_1 estimates increased when the load disturbance is added for ρ=1.0 but decreased for ρ=0.995. The absolute values of the H estimates increased for ρ=0.995 but for ρ=1.0 the magnitude of h_2 decreased while h_1 increased.

The only disadvantage to the use of the unity forgetting factor is that the output takes about 5 to 10 sample instants longer to return to the setpoint and begin tracking again than for $\rho=0.995$.

4.5 Parameter Identification Variables

As as result of the setpoint tracking, cost function weighting and disturbance rejection tests, two parameter identification techniques have proven to be superior. In all tests the recursive square root and recursive upper diagonal factorization estimators provided the lowest sums of predicted errors and control effort showing these schemes to be the most accurate estimation methods when used with the self-tuning controller.

Although the recursive square root estimator performed as well as the recursive upper diagonal factorization estimator it has the disadvantage of the necessity of calculating the square root which increases the amount of time required to obtain the estimates. Therefore, if identification is needed for a system with a large number of parameters to be estimated or the sample interval is very short a problem may arise because the next sample of data will be taken before the previous sample data has been processed. In addition, the long term use of the recursive square root estimator would consume valuable computer time that could be used for other functions.

Therefore the most accurate and the most efficient parameter identification algorithm for the control of linear systems with no time delay using the self-tuning controller is the recursive upper diagonal factorization estimator.

The recursive upper diagonal factorization estimator will now be examined in more detail to evaluate its identification capabilities when the various estimator variables are altered. The effect of the initial covariance value, forgetting factor and initial parameter estimate values will be studied. The simulation conditions for each test are identical to the setpoint tracking tests except for the estimator variable being examined. The base case used to compare certain simulations is the simulation with the original conditions specified for the setpoint tracking, that is $S(0)=3000\underline{\underline{I}}$, $\rho=0.995$, $\sigma^2=0.04$ and $\theta(0)=0.0$.

The tests of the effect of the choice of the initial covariance matrix and initial parameter estimates will be conducted using the square wave and sawtooth function change in setpoints and the forgetting factor tests will employ the square wave change in setpoint.

4.5.1 Initial Covariance Values

The magnitude of the entries of the initial covariance matrix is an indication of the accuracy of the initial parameter estimate guesses. The closer the initial estimates are to the true values the smaller the initial covariance value, S(0), to be selected. If the true parameter values are not known a relatively large $S(0)$ value of $1000\underline{\underline{I}}$ to $3000\underline{\underline{I}}$ should be used to ensure the parameter estimates will converge to the true values as quickly as possible.

The covariance matrix is initialized by setting the diagonal elements to the desired value and all off-diagonal elements are zero.

If $S(0)$ is selected as the identity matrix, that is $S(0)=\underline{\underline{I}}$ the control performance and parameter estimates that result are shown in Figures 4.65 and 4.66. Since the initial covariance value is small thus implying the estimate guesses are close to the true parameters, there are no large fluctuations in any of the parameter estimates. The F estimates change very slowly. After 100 sample instants f_0 and f_2 are within 16% of the true value but f_1 is not. The G estimates change frequently at first but by the 25th sample instant the parameter estimates are constant and the estimates are converging to the true values. The H estimates tend to follow each other until the 55th sample instant and then become equal in magnitude but opposite in sign. The control performance is very good even for the first 10 sample instants but deteriorates from the 20th to the 30th sample instant. The control effort and sum of prediction errors values after 200 sample instants are 36% and 14%, respectively, lower than for the base case.

Parameter estimate behavior when $S(0)=10\underline{\underline{I}}$ is illustrated in Figure 4.67. The F estimates grow faster than for $S(0)=\underline{\underline{I}}$ at the start of the simulation. The f_0 and f_1

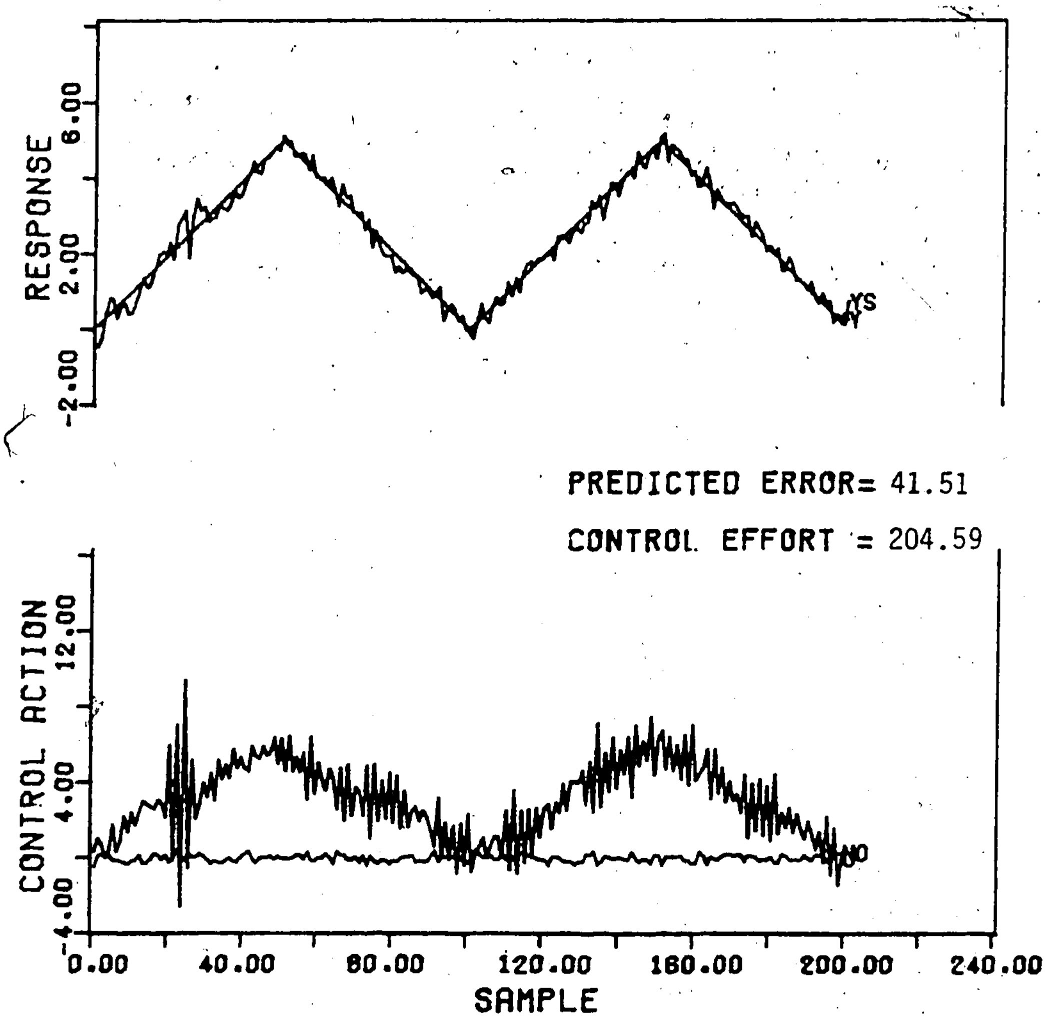


Figure 4.65 Control Response for an Initial Covariance Matrix $S(0)=\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

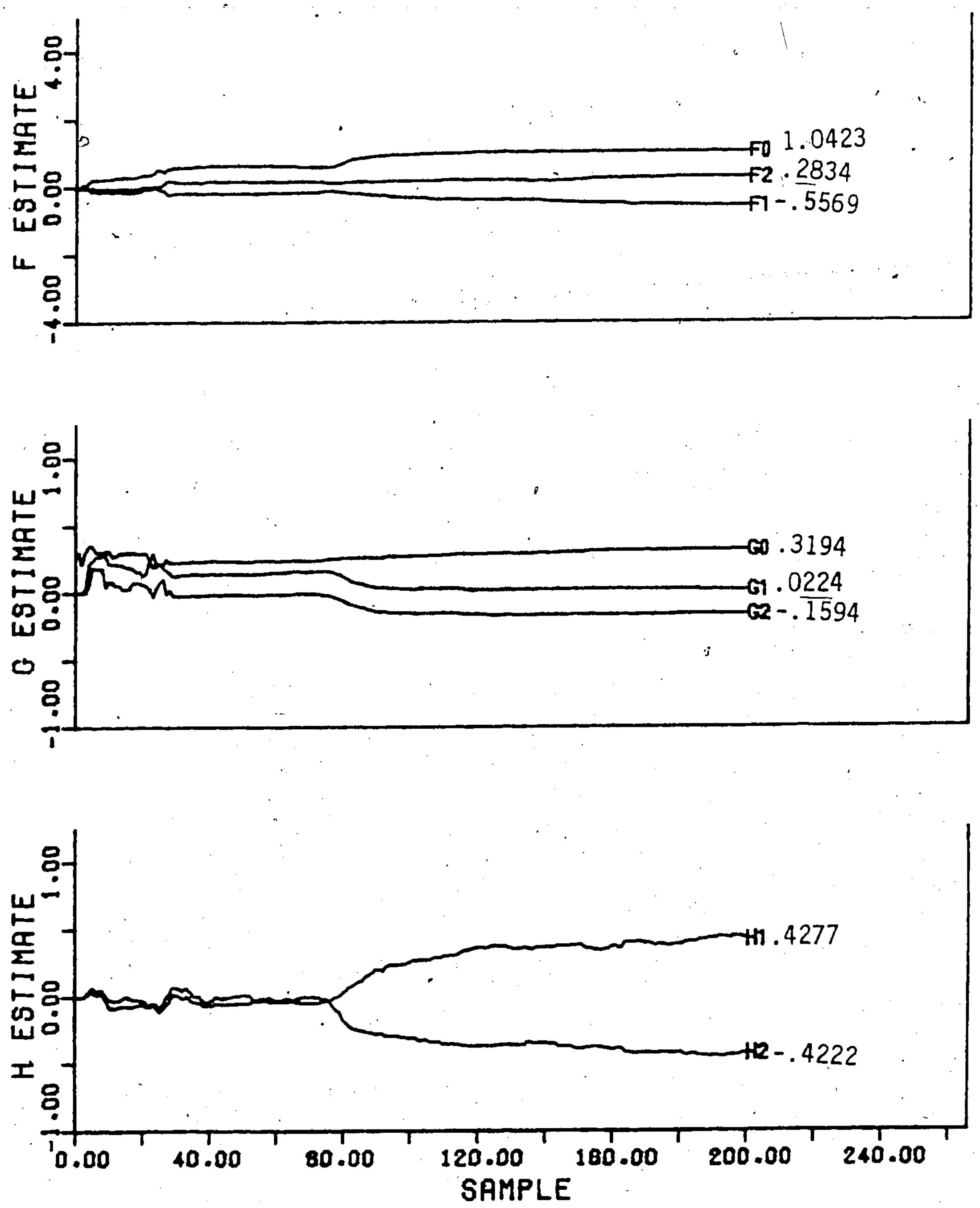


Figure 4.66 Parameter Estimates for an Initial Covariance Matrix $S(0)=\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

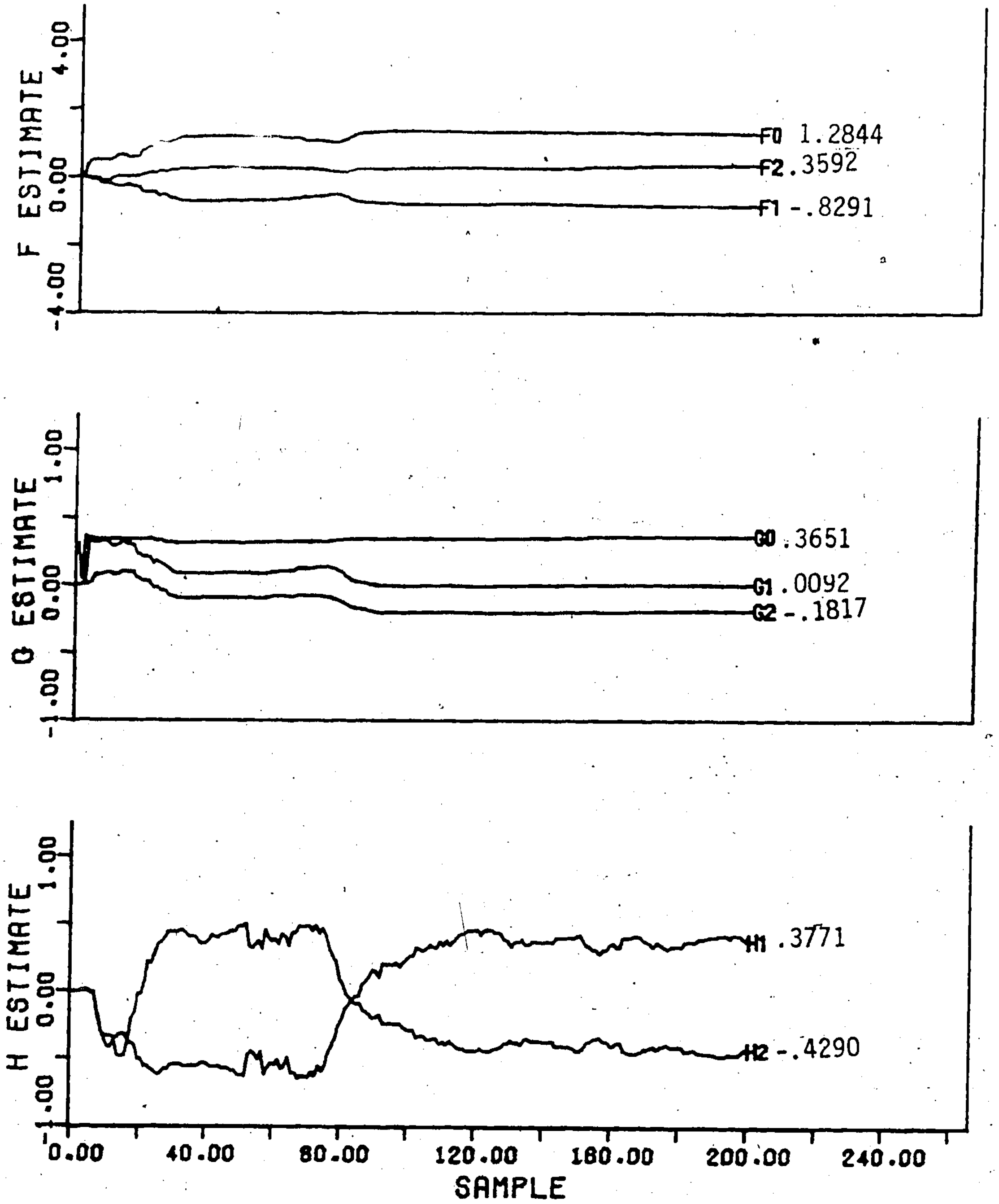


Figure 4.67 Parameter Estimates for an Initial Covariance Matrix $S(0)=10.0\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

estimates are significantly larger than when starting the $S(0)=\underline{\underline{I}}$ case but the general trends of the parameter estimates are similar. The G estimates adapt quickly at startup and g_1 and g_2 are 10% closer to the true values by the 200th sample instant than the parameter estimates when starting with $S(0)=\underline{\underline{I}}$ The H esimates are approximately equal until the 10th sample instant and then the estimates become mirror images of each other by the 40th sample instant. The H estimates are within 12% of the parameter estimates that resulted using $S(0)=\underline{\underline{I}}$ but are not converging to the true values. Figure 4.68 illustrates how the control performance has deteriorated for the first 15 sample instants causing a 26.5% increase in control effort compared to the simulation results for $S(0)=1\underline{\underline{I}}$. However, overall there was a decrease of 3.7% in sum of prediction errors as the output tracks the setpoint quite well after the 20th sample instant.

For $S(0)=100\underline{\underline{I}}$ Figure 4.69 shows that the F estimates grow larger than for $S(0)=10\underline{\underline{I}}$ in the initial sample instants of the simulation. The f_0 and f_1 estimates have grown larger than their true values by the 200th sample instant. The g_1 and g_2 estimates do not increase at the 4th and 5th sample instants as much as was the case starting with $S(0)=10\underline{\underline{I}}$ and furthermore, the final values of g_0 and g_2 at the 200th sample instant are also smaller. The H estimates grow about three times larger than the true values. The H estimates are more symmetrical to each other starting with $S(0)=100\underline{\underline{I}}$ than for $S(0)=10\underline{\underline{I}}$ and the parameter estimate values at the 200th

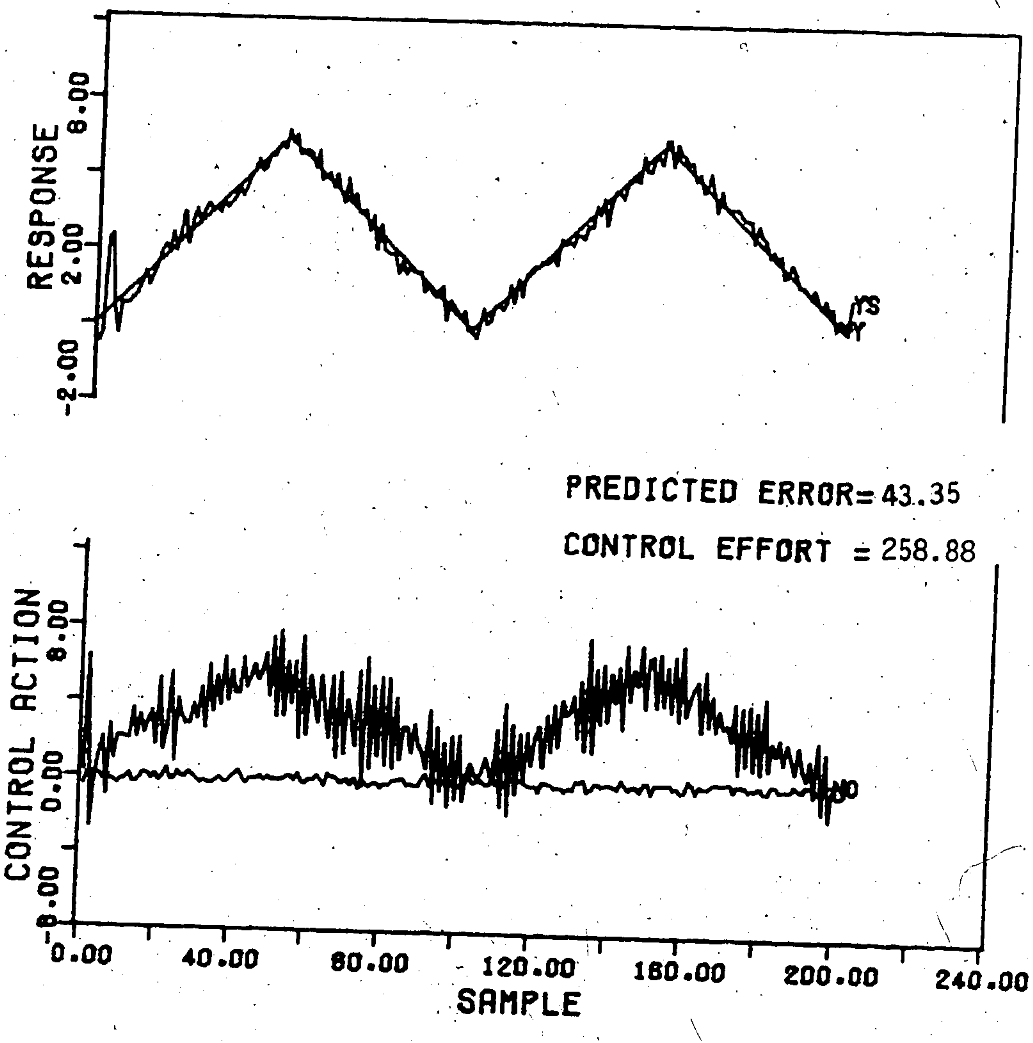


Figure 4.68 Control Response for an Initial Covariance Matrix $S(0)=10.0\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

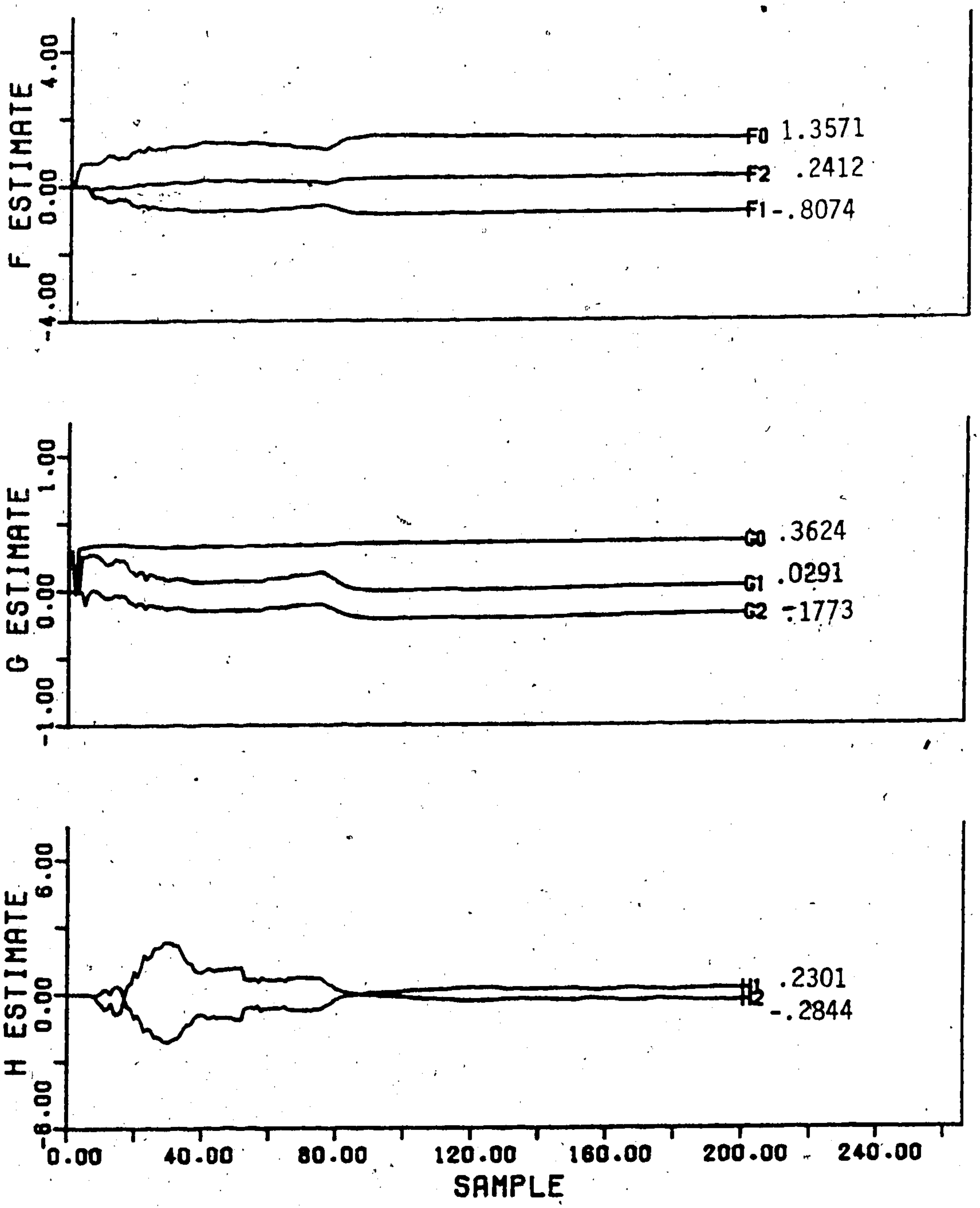


Figure 4.69 Parameter Estimates for an Initial Covariance Matrix $S(0)=100.0\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator

sample instant are about 40% lower than the values obtained starting with $S(0)=\underline{\underline{I}}$. The initial control performance as seen in Figure 4.70 is significantly worse than for the tests using the smaller initial covariance values and the large increases in control effort of up to 63% and sum of prediction errors of 18% provide substantiation. The output does not track the setpoint well until the 20th sample instant.

Comparison of the adaption pattern of the F and G estimates for $S(0)=100\underline{\underline{I}}$ exhibit similar trends as observed for the base case $S(0)=3000\underline{\underline{I}}$. As can be seen from Figure 4.18 the f_1 and g_1 estimates, by the 200th sample instant, are 27% and lower while the remaining parameter estimates are within ±5.2% of the estimates obtained for $S(0)=100\underline{\underline{I}}$. The H estimates are still mirror images of each other. The initial large fluctuations have increased about 40% and the final estimates have also increased compared to the values estimated starting with $S(0)=100\underline{\underline{I}}$. The control performance as shown in Figure 4.15 which is seen to be marginally better because of a 4.5% reduction in control effort and 4.3% decrease in the sum of prediction error value. However, control performance is still not as desirable as that observed for $S(0)=\underline{\underline{I}}$ and $S(0)=10\underline{\underline{I}}$. The parameter estimates for $S(0)=10^4\underline{\underline{I}}$, $10^5\underline{\underline{I}}$ and $10^6\underline{\underline{I}}$, can be seen in Figures 4.71, 4.72 and 4.73, respectively, and Figure 4.74 contains the control performance for $S(0)=10^4\underline{\underline{I}}$. In all three cases the parameter estimates are almost equivalent, differences of

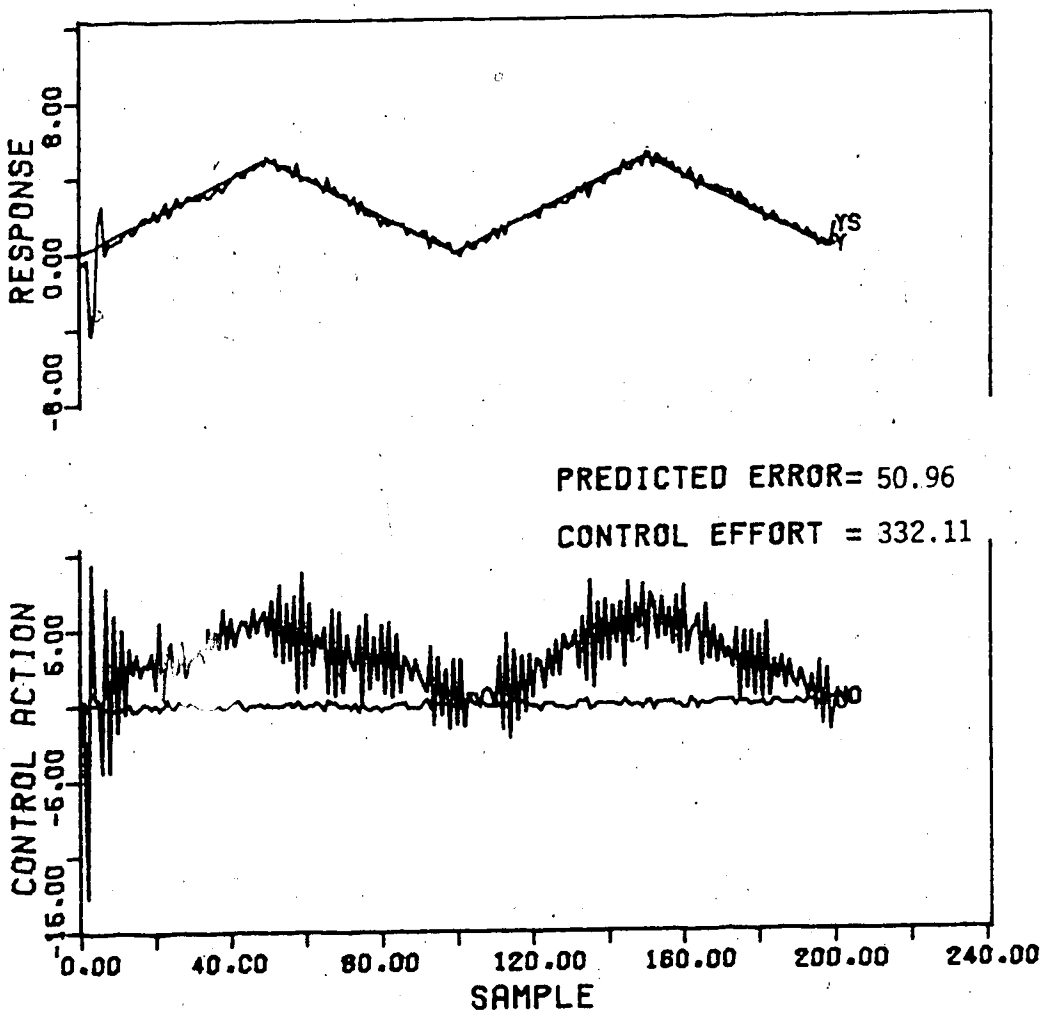


Figure 4.70 Control Response for an Initial Covariance Matrix $S(0)=100.0\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator

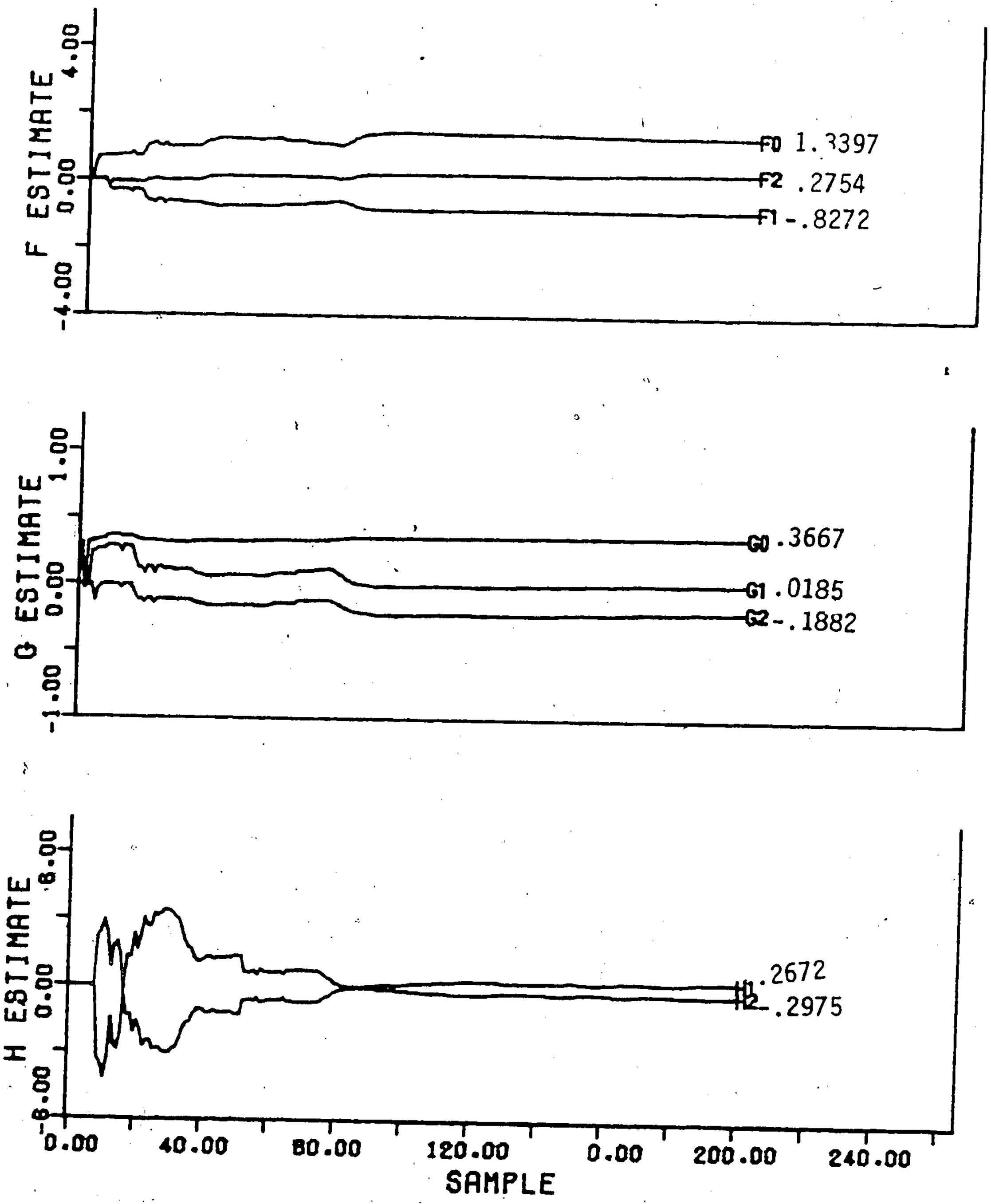


Figure 4.71 Parameter Estimates for an Initial Covariance Matrix $S(0)=10^{4}\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

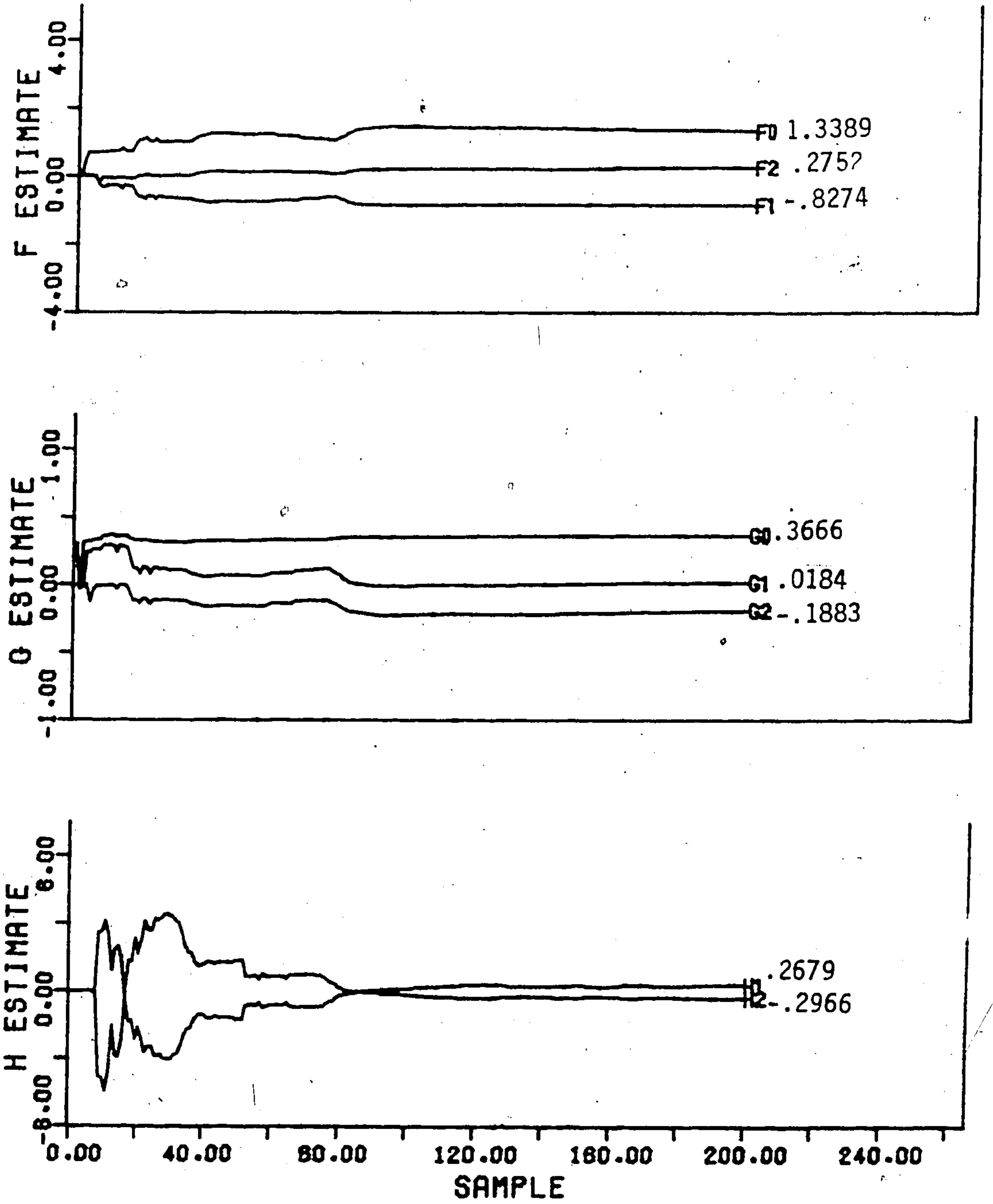


Figure 4.72 Parameter Estimates for an Initial Covariance Matrix $S(0)=10^5\underline{\underline{I}}$ using the Recursive Upper Diagonal Factorization Estimator

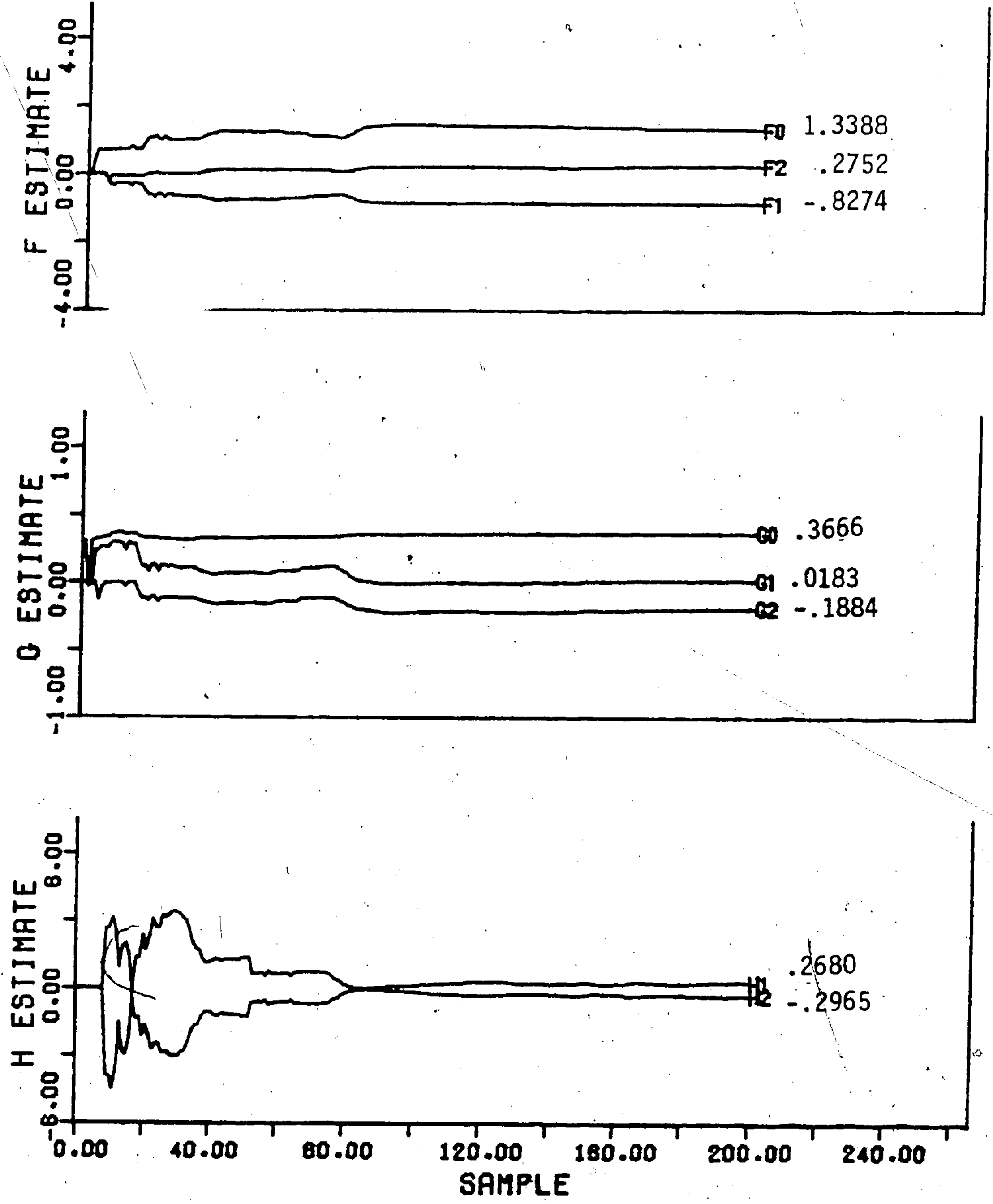


Figure 4.73 Parameter Estimates for an Initial Covariance $S(0)=10^4\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator

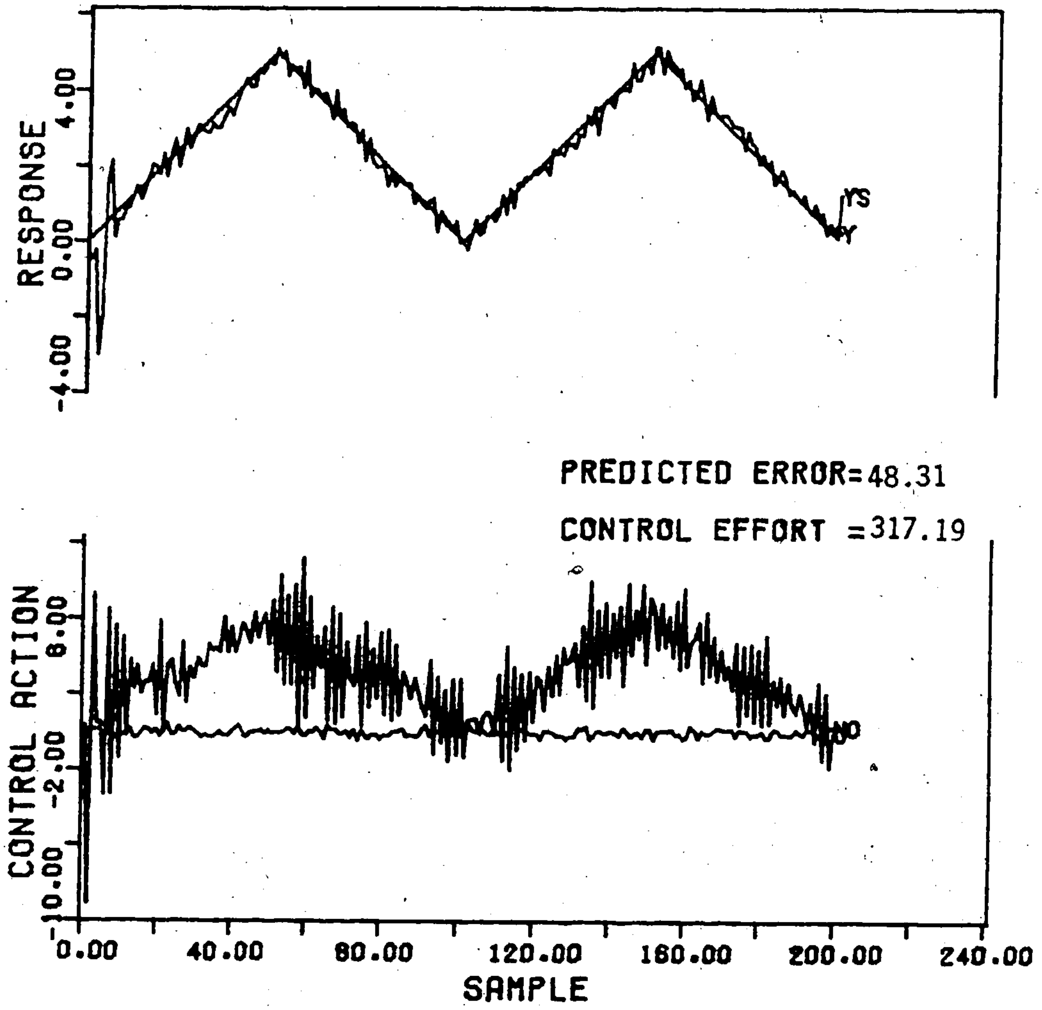


Figure 4.74 Control Response for an Initial Covariance Matrix $S(0)=10^4\underline{I}$ using the Recursive Upper Diagonal Factorization Estimator

less than 1% are observed in the parameter estimates that resulted from using these three initial covariance matrices. The control performance for all three initial covariance values simulations was found to be identical as the control effort values were equal to within less than 1% and the sum of prediction errors values were within .06% of each other. Compared to the control performance when $S(0)=3000\underline{I}$ the control effort and sum of predicted errors values are within .1% of each other and the parameter estimates are also within 1% of the estimates yielded when $S(0)=3000\underline{I}$. Therefore any increase in the initial covariance matrix, S(0) beyond $S(0)=3000\underline{I}$ does not improve control effort, sum of prediction errors or final parameter estimate values. However, the choice of initial covariance matrix does influence the parameter estimate behavior. As S(0) is increased the F estimates show more initial oscillations but by the 100th sample instant the estimates are no longer affected by S(0). For $S(0)=\underline{I}$ the G estimates displayed initial oscillations but for the larger values the initial parameter estimate changes were less erratic. The H estimate behavior is more sensitive to the intitial covariance choice than the F and G estimates. The lower the S(0) value the longer the two estimates remained equal to each other. As the value of S(0) increased so did the magnitude of the H parameter fluctuations observed between the [illegible]th and 40th sample instant. Although the lower the initial covariance values gave the lower control effort and sum of prediction errors because the initial setpoint tracking was very good, this was only due to the small setpoint changes of the sawtooth function. When the same S(0) values were used with a square wave setpoint change the minimum control effort and sum of prediction errors resulted by starting with $S(0)=100\underline{\underline{I}}$ as shown by the summary in Table 4.2. Therefore for larger abrupt setpoint changes the estimates converge more quickly providing better control performance. Consequently a compromise between the requirement for good tracking and faster parameter convergence must be made when choosing an initial covariance matrix.

4.5.2 Forgetting Factor

The effect of changing the forgetting factor, ρ, in controller parameter identification was also studied. All simulations used the recursive upper diagonal factorization method with P=R=1 and Q=0 and only the forgetting factor was altered. The results from all tests, performed for a square wave change in setpoint, showed no difference in output until the first setpoint change so the forgetting factor value ρ does not affect initial parameter estimates. This is because the observation vector is not yet complete so data truncation is not a factor. However, by the 200th sample instant the differences are evident.

The control performance and parameter estimate results for $\rho=0.7$ and $\rho=0.85$, are shown in Figures 4.75, 4.76, 4.77 and 4.78, respectively. Even after 200 sample instants the

Table 4.2 Effect of the Choice of the Initial Covariance Matrix using the Recursive Upper Diagonal Factorization Estimator on Performance

INITIAL COVARIANCE VALUE	SETPOINT TYPE	CONTROL EFFORT	SUM OF PREDICTED ERRORS	
1.0	SQW	571.77	99.07	
	SAW	204.59	40.57	Figure 4.65
10.0	SQW	565.78	85.38	
	SAW	258.88	43.39	Figure 4.68
100.0	SQW	565.51	94.76	
	SAW	332.11	50.49	Figure 4.70
3000.0	SQW	565.10	95.19	Figure 4.14
	SAW	317.36	48.38	Figure 4.15
$1.x10^4$	SQW	638.53	125.77	
	SAW	317.19	48.41	Figure 4.74
$1.x10^5$	SQW	684.26	98.18	
	SAW	309.06	48.60	
$1.x10^6$	SQW	704.55	117.38	
	SAW	275.70	49.69	

SQW - Square Wave Setpoint Change
SAW - Sawtooth Function Change in Setpoint

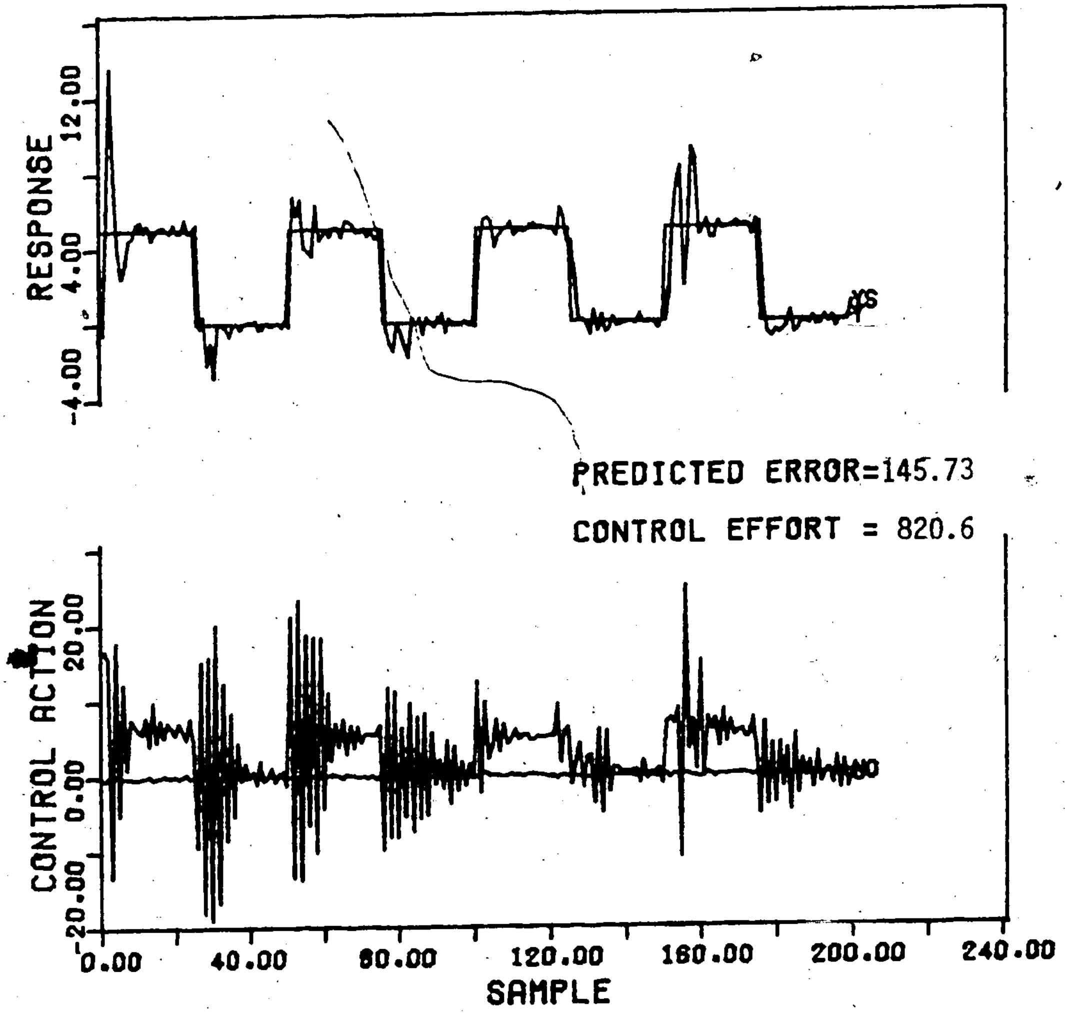


Figure 4.75 Control Response for a Forgetting Factor of 0.7 using the Recursive Upper Diagonal Factorization Estimator

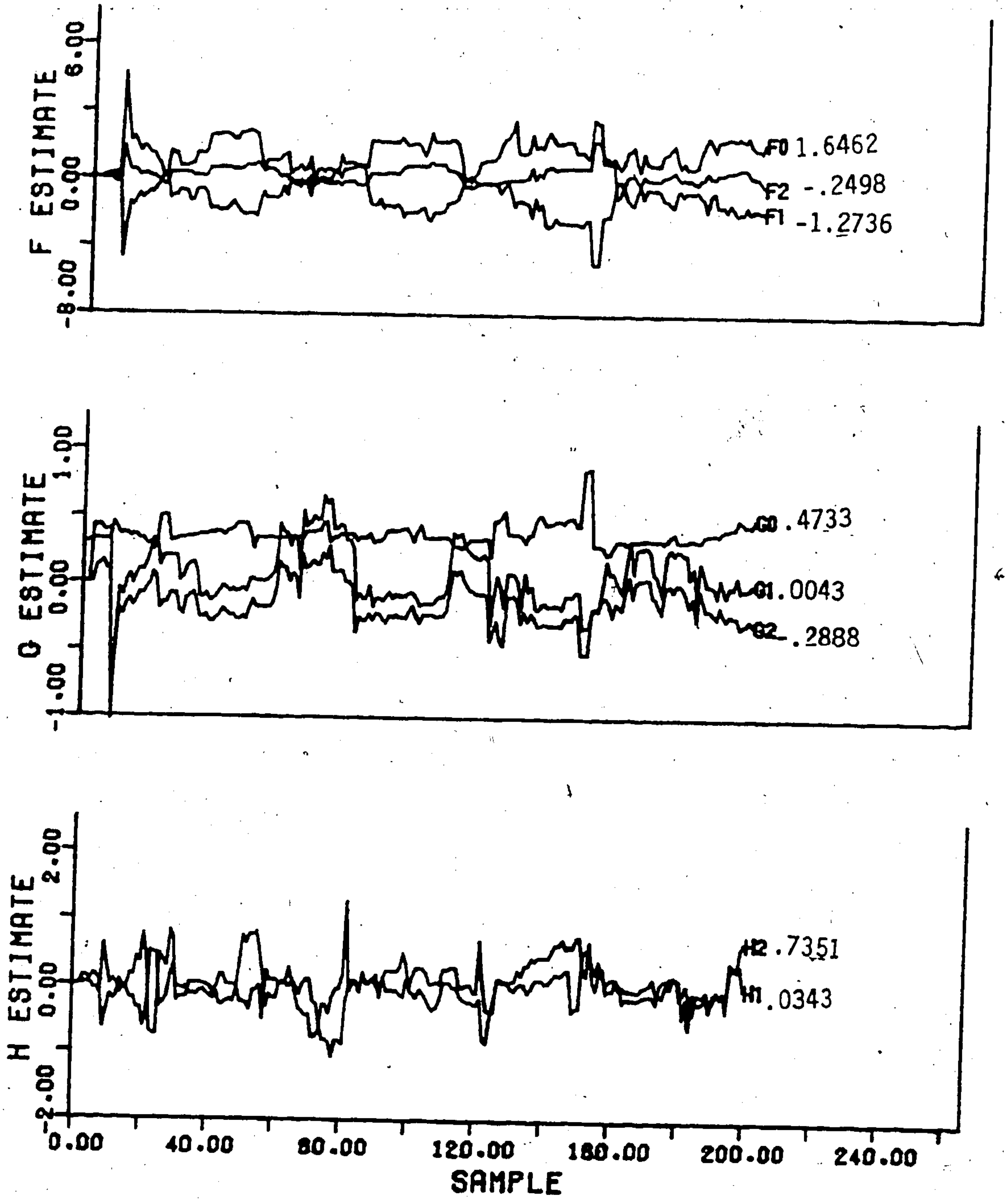


Figure 4.76 Parameter Estimates for a Forgetting Factor of 0.7 using the Recursive Upper Diagonal Factorization Estimator

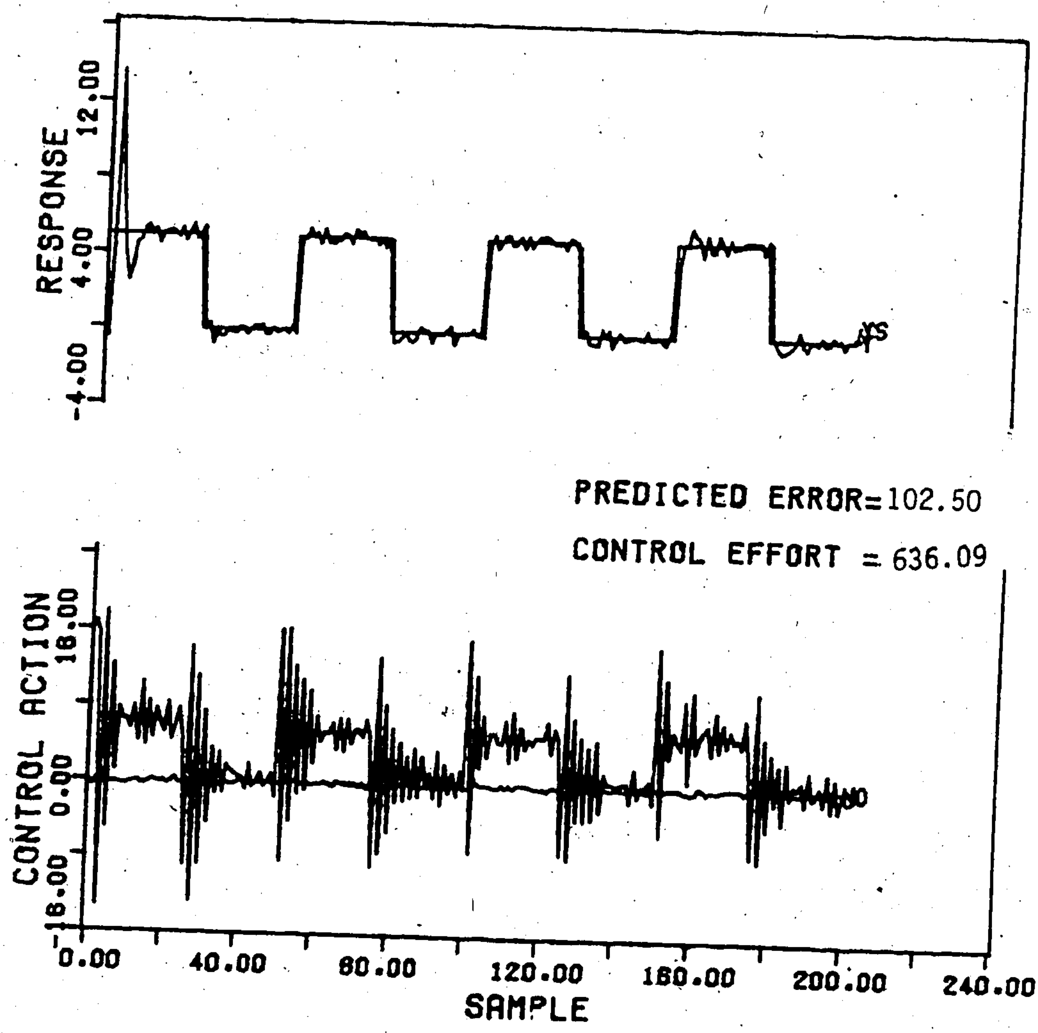


Figure 4.77 Control Response for Forgetting Factor of 0.85 using the Recursive Upper Diagonal Factorization Estimator

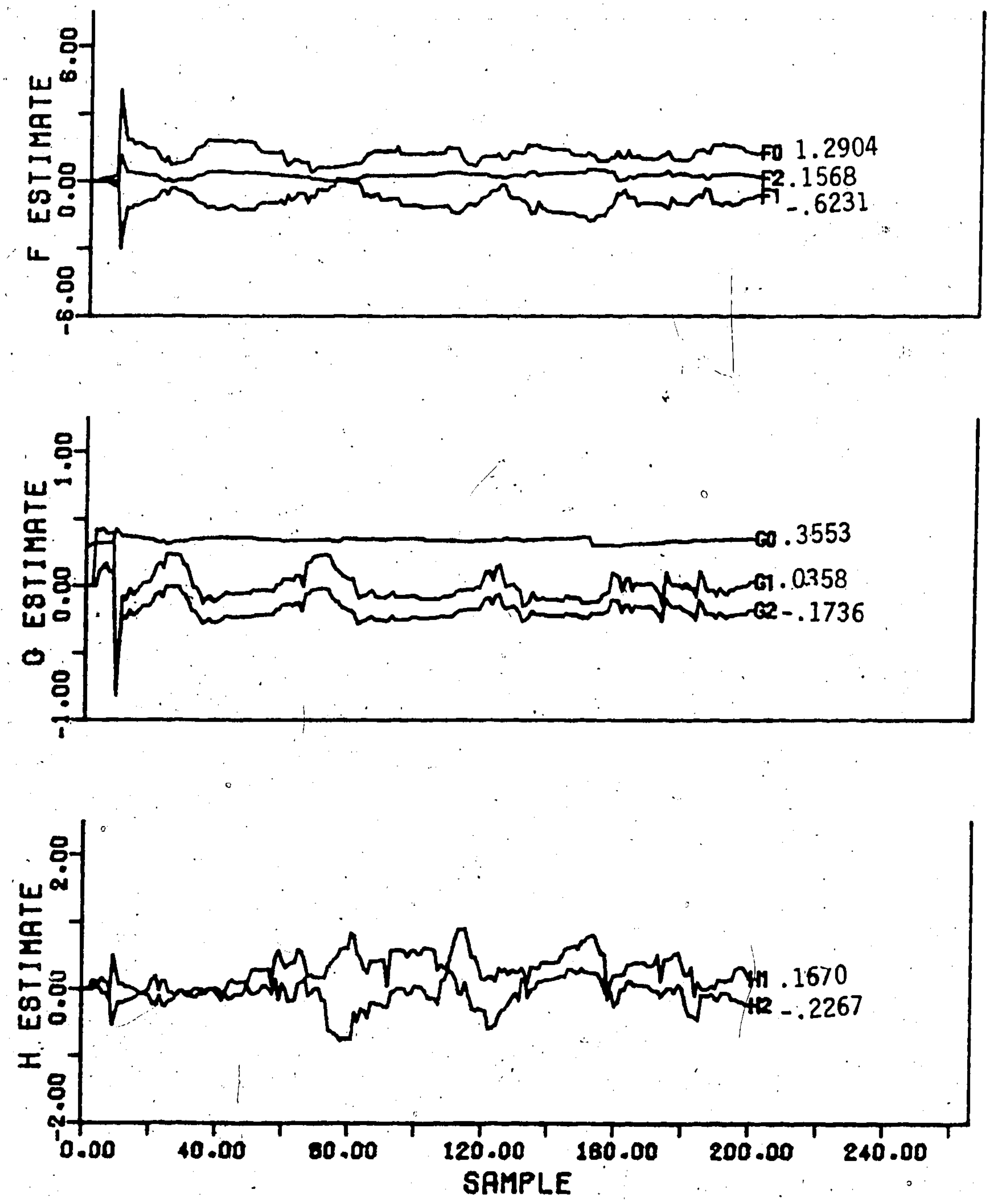


Figure 4.78 Parameter Estimates for a Forgetting Factor of 0.85 using the Recursive Upper Diagonal Factorization Estimator

parameter estimates have not converged and are very erratic for the entire simulation. For ρ=0.7 the control performance is not satisfactory since when the setpoint changes occur rather large oscillations occur. The control performance for ρ=0.85 is considerably better as reflected by a reduction of 23% in the control effort value and 10% in the sum of prediction errors, but the initial performance after a setpoint change is still not acceptable as the setpoint tracking is poor.

Simulations using smaller value of the forgetting factor than 0.7 were not performed since values of ρ less than 0.7 would result in unsatisfactory estimates and possibly parameter blow up which would lead to unstable control performance.

It can be seen from Figure 4.79 that increasing the forgetting factor to 0.9 reduces the erratic behavior of the parameter estimates and the F and G estimates seem to be approaching values within a constant range but do not appear to be converging to any specific value. In contrast, the H estimate adaption pattern is very erratic and the estimates do not appear to be converging. The control performance shown in Figure 4.80 is acceptable but fluctuations around the setpoint are evident between the 75th and 100th sample instants and the output does not initially reach 5.0 at the step change at the 150th sample instant. It would appear that a forgetting factor even larger than 0.9 is necessary to obtain good parameter estimates and therefore

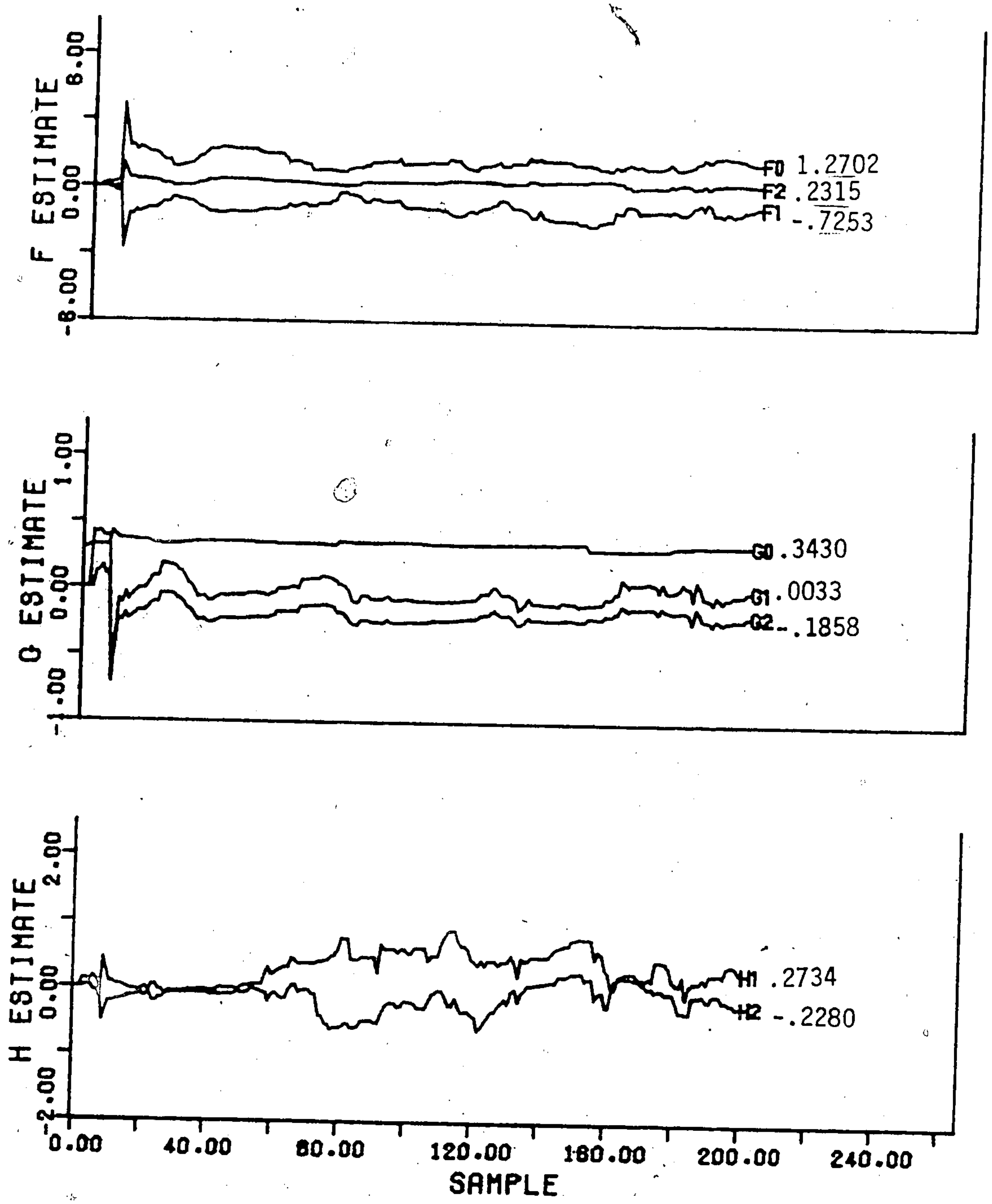


Figure 4.79 Parameter Estimates for a Forgetting Factor of 0.9 using the Recursive Upper Diagonal Factorization Estimator

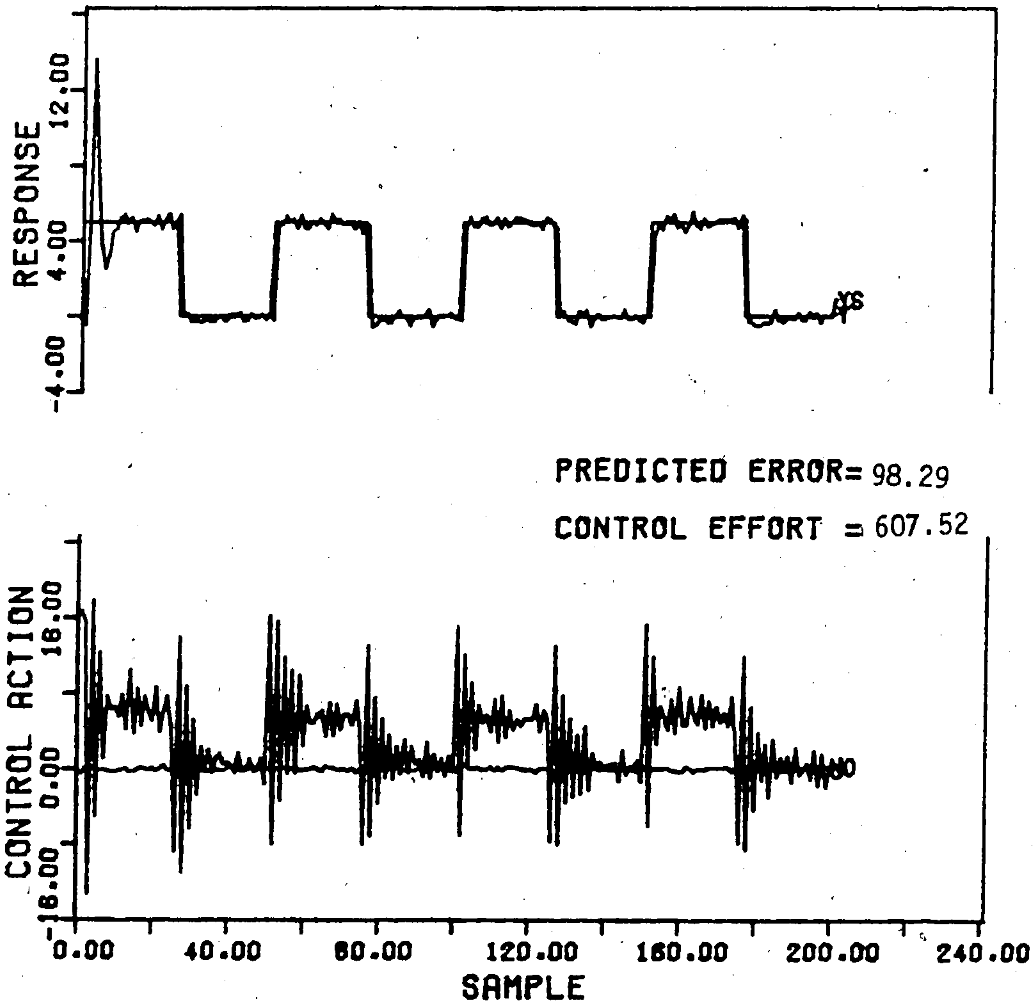


Figure 4.80 Control Response for Forgetting Factor of 0.9 using the Recursive Upper Diagonal Factorization Estimator

satisfactory control performance so simulations with larger values were performed. For a forgetting factor of 0.95 the F and G estimates have still not converged to a constant value as shown by the results displayed in Figure 4.81 although the behavior is less erratic than for $\rho=0.9$. The H parameter estimates change more than the F and G estimates and the adaption pattern represents a considerable improvement compared to that for $\rho=0.90$. As can be seen in Figure 4.82, an increase in ρ from 0.9 to 0.95 improves the control performance however, the output at the setpoint change at the 150th sample instant still does not reach the new setpoint. On the basis of these results, it is reasonable to consider that an acceptable lower bound for the forgetting factor is 0.95 as the control performance is marginally acceptable even if the parameter estimates are still adapting.

The parameter estimates and control performance for $\rho=0.975$ is given in Figures 4.83 and 4.84. The F and G estimates are now constant and all parameter estimates except for the h_2 estimate are within ±24% of the true values. The H estimates are fairly stable and both appear to be approaching the true values of $h_1=0.79$ and $h_2=-0.37$ although h_2 is adapting slower than h_1. The control performance shown in Figure 4.84 is only slightly better than achieved using a forgetting factor of $\rho=0.95$. The new setpoint is reached at the 150th sample instant but reductions of less than 1.0% in the control effort and sum

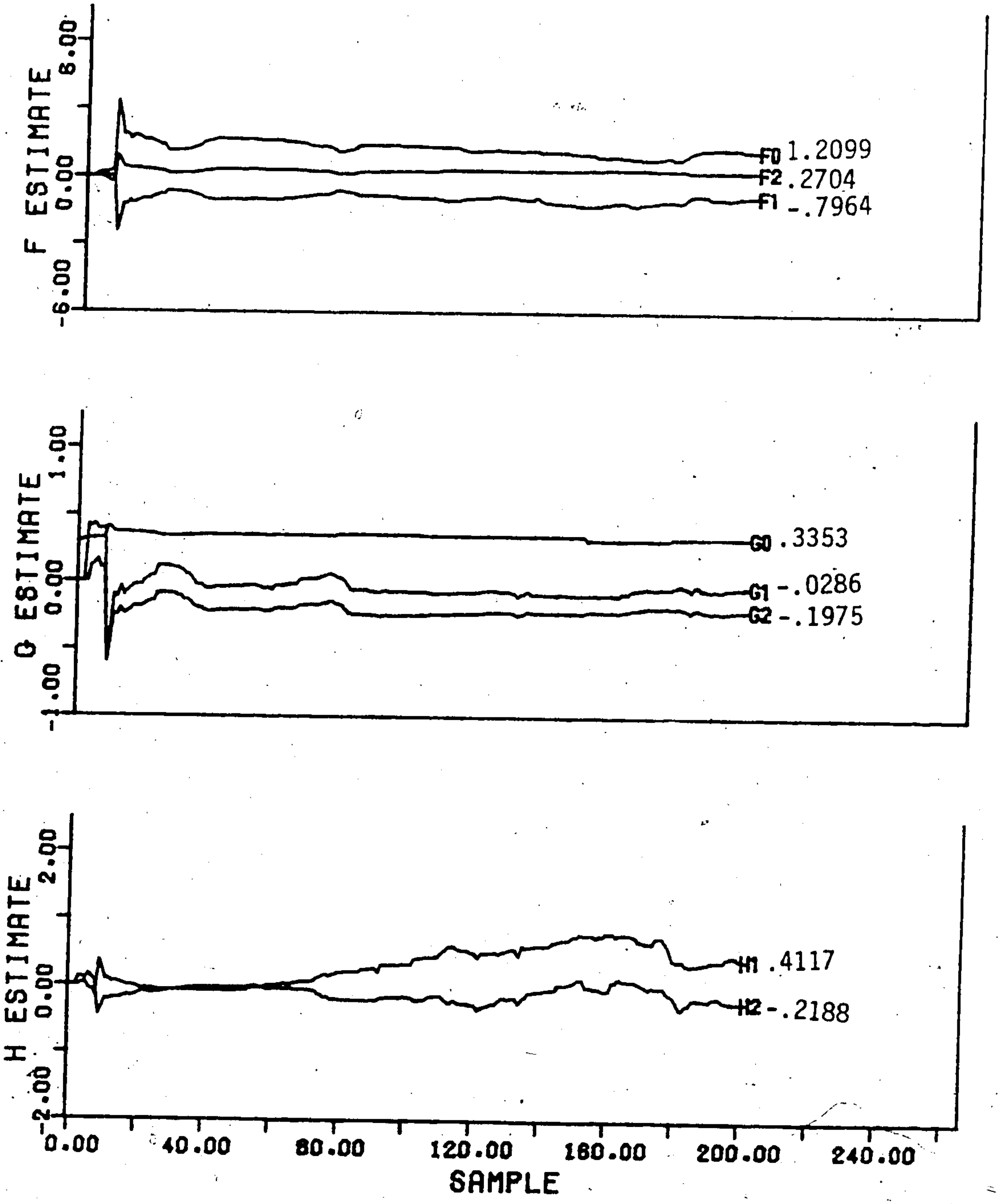


Figure 4.81 Parameter Estimates for a Forgetting Factor of 0.95 using the Recursive Upper Diagonal Factorization Estimator

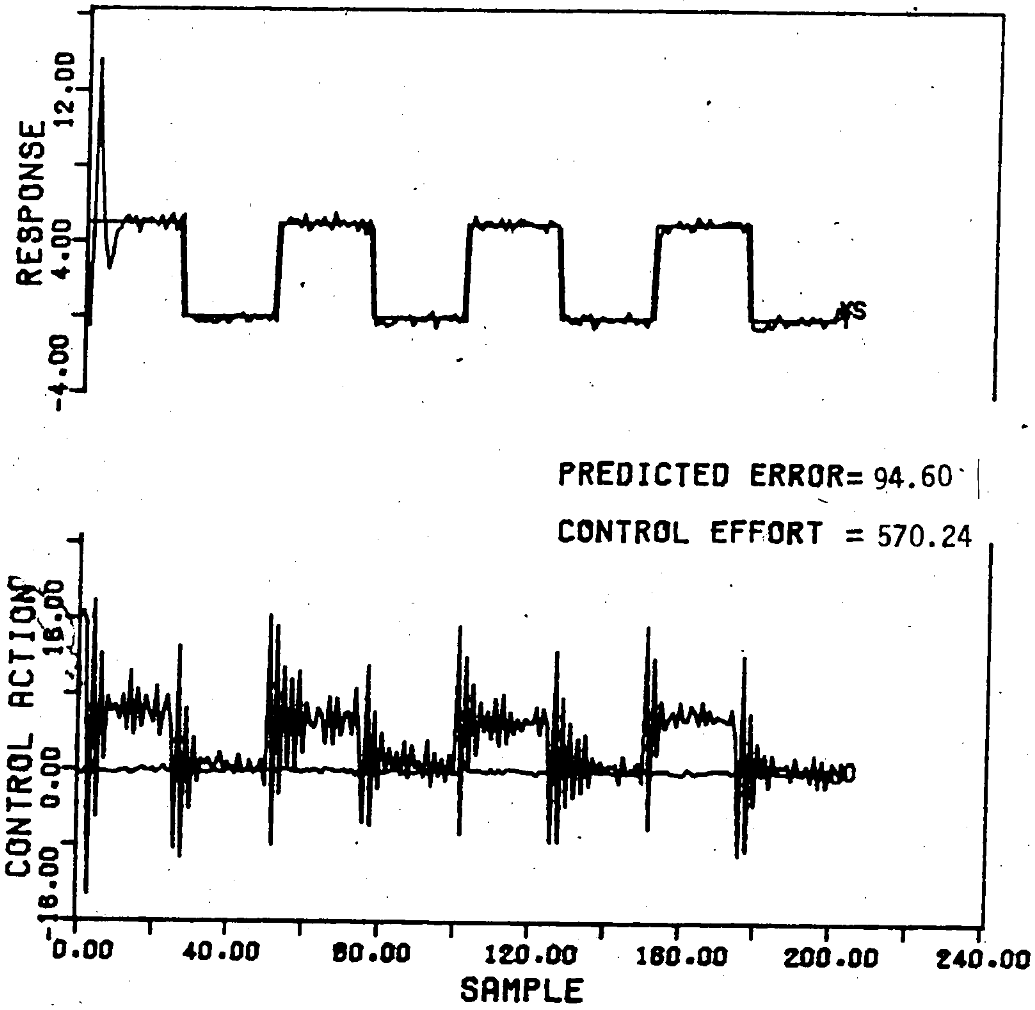


Figure 4.82 Control Response for Forgetting Factor of 0.95 using the Recursive Upper Diagonal Factorization Estimator

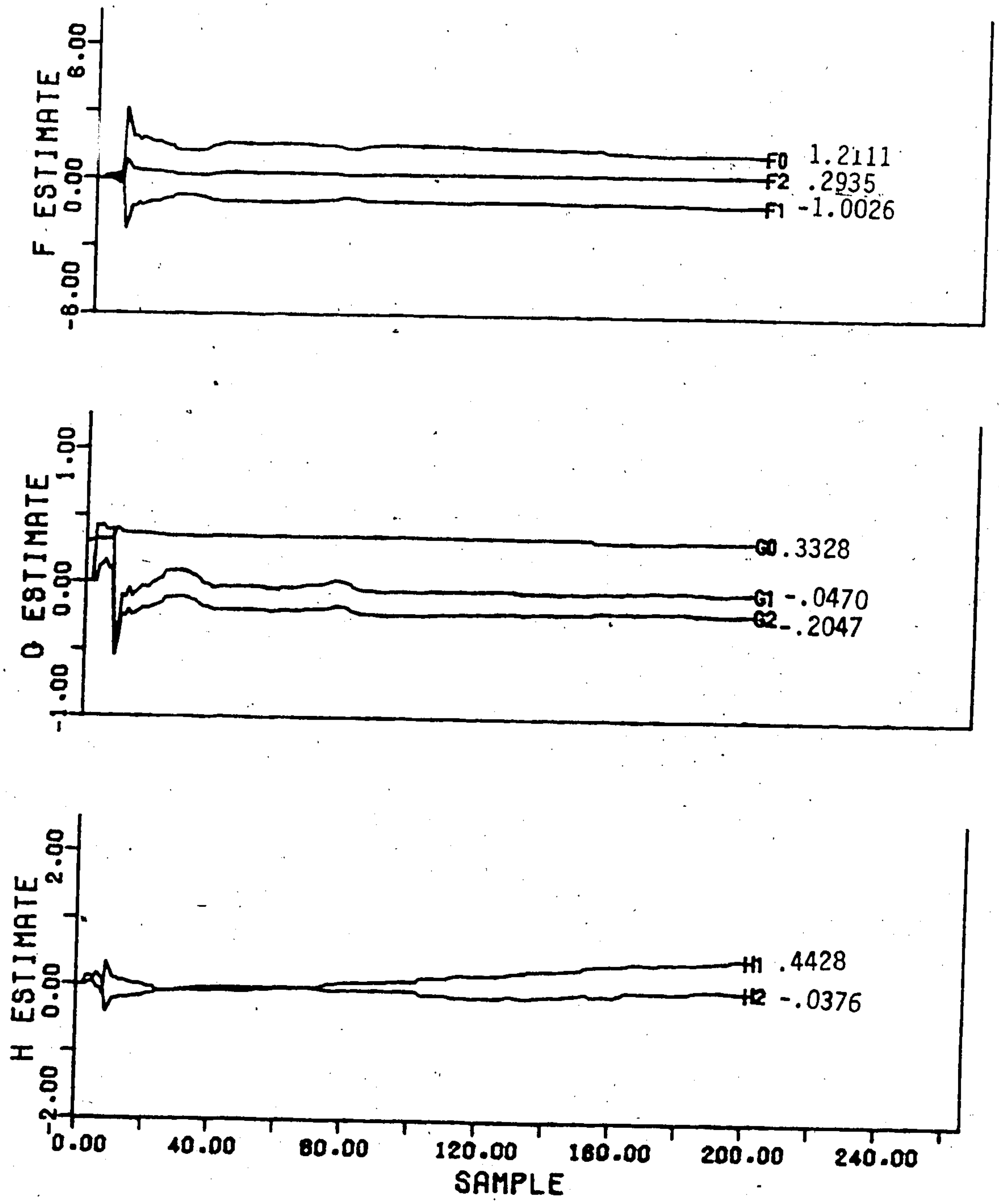


Figure 4.83 Parameter Estimates for a Forgetting Factor of 0.975 using the Recursive Upper Diagonal Factorization Estimator

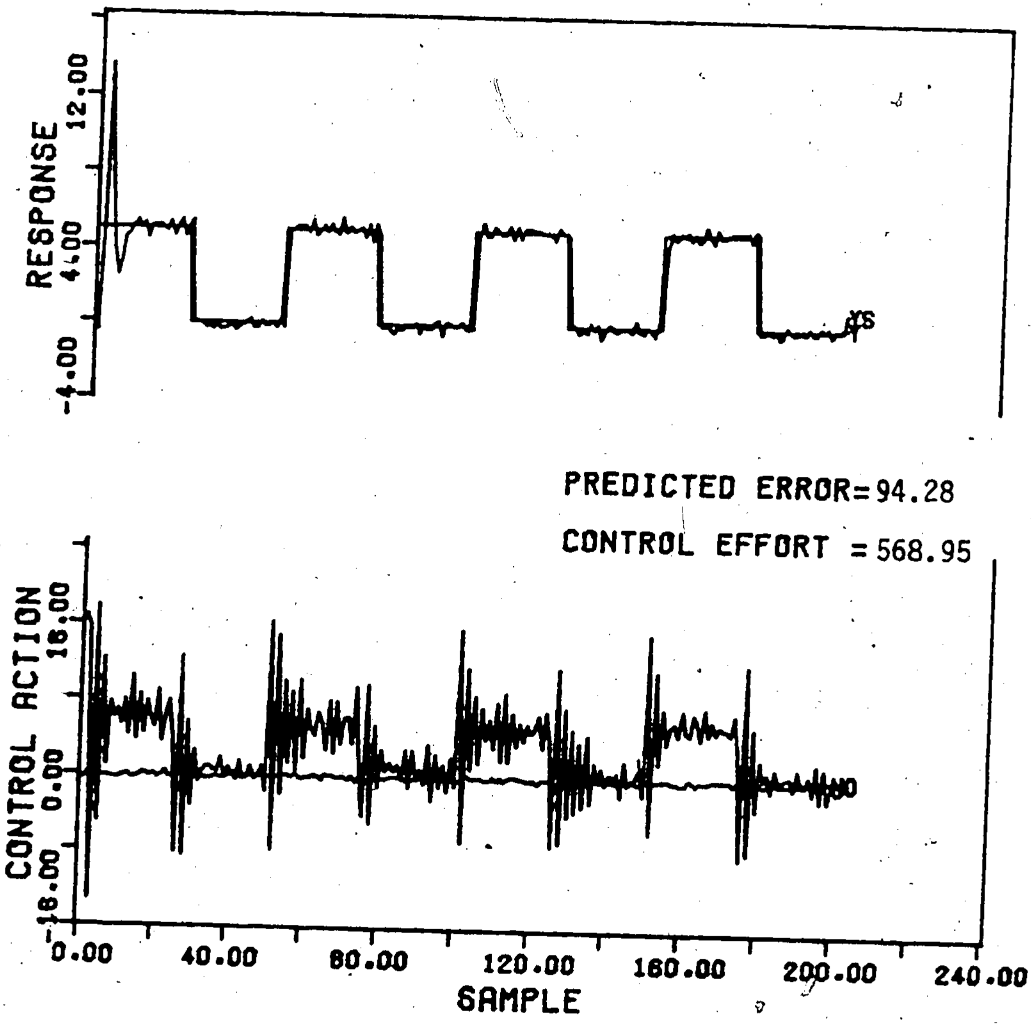


Figure 4.84 Control Response for Forgetting Factor of 0.975 using the Recursive Upper Diagonal Factorization Estimator

of prediction errors values is not considered significant. Comparison of the base case results in Figure 4.17 for $\rho=0.995$ shows that the F and G estimates are very similar to those in Figure 4.83 but despite the fact that the H estimates have become constant, the estimates are still only within 80% of the true values. The control performance shown for $\rho=0.995$ shown in Figure 4.14 when compared with that in Figure 4.84 is very similar with the decrease of the control effort for $\rho=0.995$ but accompanying this desirable change there is a 1% increase in the magnitude of the sum of prediction error value.

If no data truncation is desired then the forgetting factor is set to=1.0. The parameter estimates and control performance that results for $\rho=1.0$ is presented in Figures 4.85 and 4.86. The parameter estimates are constant and have converged, and all estimates except the g_0 and g_2 parameters are between 20% and 90% from the true values. The control performance is acceptable with an increase of less than .4% in the value of the sum of the prediction errors while control effort value is reduced by .9% from that using $\rho=0.995$.

From the simulation results using forgetting factors from 0.7 to 1.0 it has been found that for a forgetting factor below 0.95 the parameter estimates change rapidly and no convergence is observed even after 200 sample instants. The H parameter estimates are the most sensitive to the forgetting factor as the F and G estimates are fairly

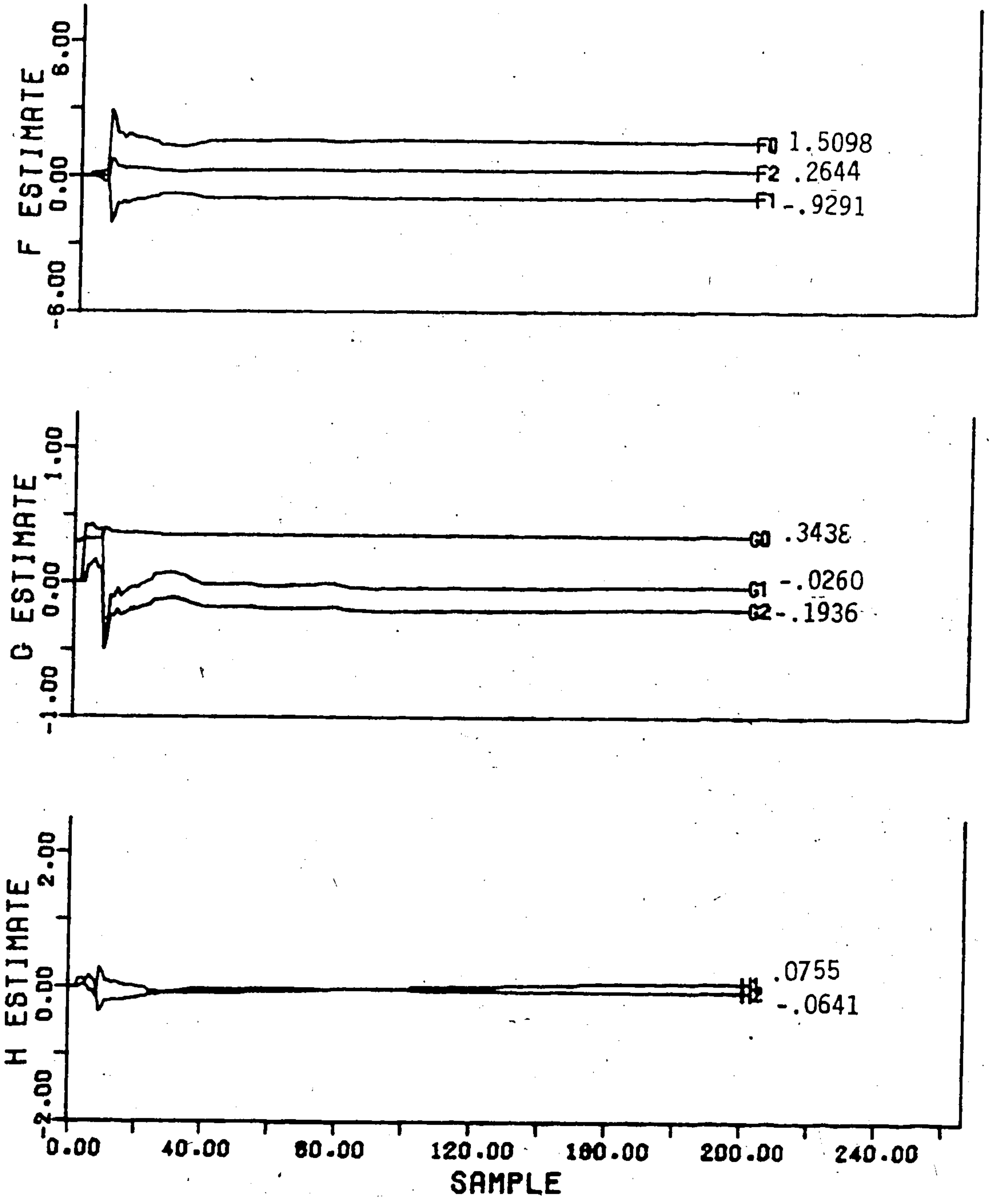


Figure 4.85 Parameter Estimates for a Forgetting Factor of 1.0 using the Recursive Upper Diagonal Factorization Estimator

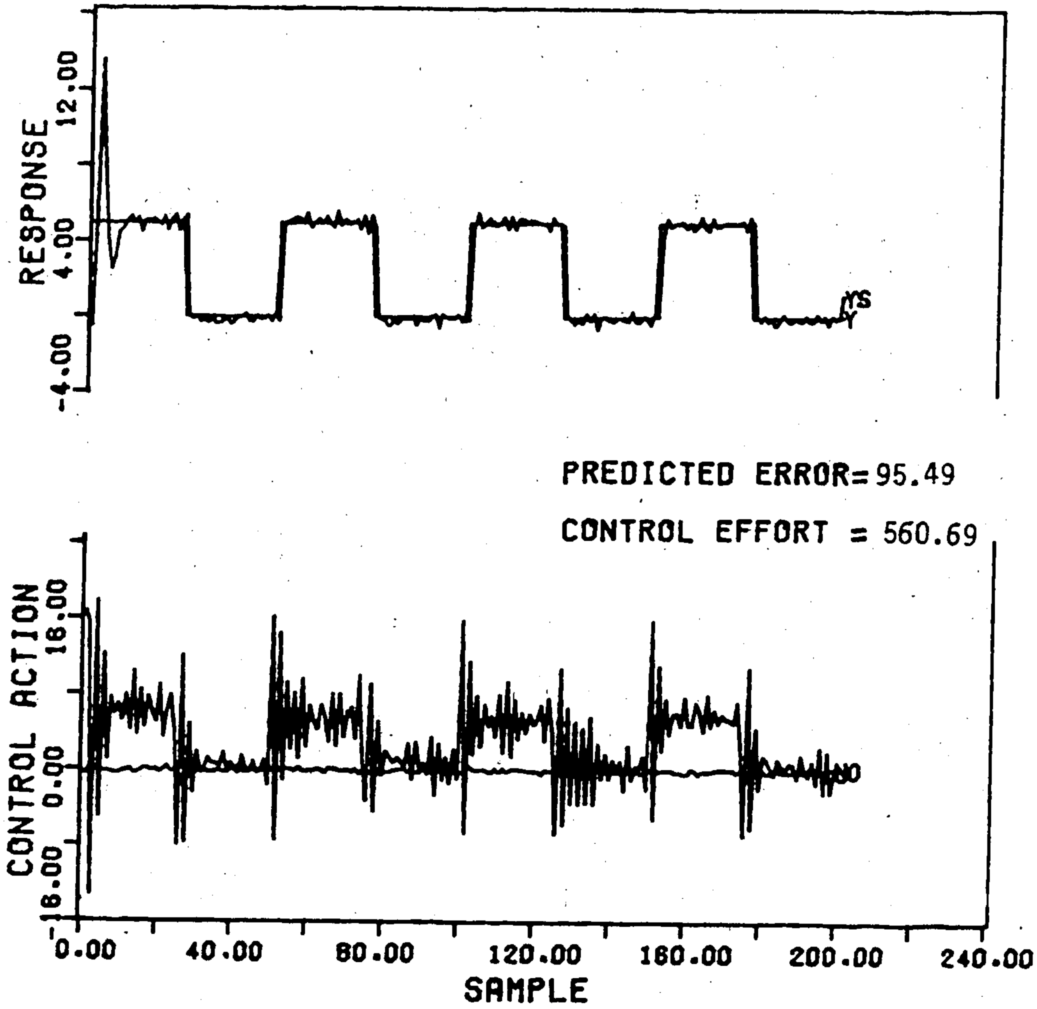


Figure 4.86 Control Response for Forgetting Factor of 1.0 using the Recursive Upper Diagonal Factorization Estimator

constant for $\rho=0.975$ and 0.995 but the H estimates only exhibited constant behavior for $\rho=0.995$. Although the simulation results for a forgetting factor of 0.975 exhibited the smallest sum of predicted errors for a square wave change in setpoint, a sawtooth function change in setpoint caused the F and G parameter estimates to increase to values approximately double the true value and the sum of predicted errors became larger than the total obtained for $\rho=0.995$. The F and G parameter estimates were even greater for a step change in setpoint than for the sawtooth function change in setpoint and caused parameter blowup in the H estimates. So on this basis a forgetting factor of 0.995 is considered to be appropriate for the system under study as it gives the most satisfactory results for all three types of setpoint changes.

In general, the forgetting factor should be chosen on the basis of the expected variations in system parameters. If the parameters are slowly changing, values near 1.0 should be used and for faster variations a forgetting factor of 0.95 would be more appropriate.

In systems where there is low excitation or operation, such as a constant setpoint, the forgetting factor should be set to 1.0 and only changed when plant conditions change.

For a nonlinear system, frequent setpoint changes lead to rapid changes in the assumed linear model so a forgetting factor of 0.95 should be used initially and once parameter tuning is complete a value near 1.0 should be implemented to allow for the slow parameter changes [10].

4.5.3 Initial Parameter Estimates

The choice of initial parameter values is important, since it is these values that determine the trajectory of the parameter estimates and thus the final stationary points [33]. The values also influence convergence time and in some cases the closed loop stability of the system.

If the true values of the parameters are not known the initial estimates are usually set to zero except for g_0 which must be non-zero and positive to avoid dividing by zero in the control action calculation. The g_0 should be chosen for fast convergence as discussed in the self-tuning theory section. (cf. Section 2.4)

The control behavior for the square wave and sawtooth wave setpoint changes for initial parameter estimates, $\theta(0)$, set to 1.0 are shown in Figures 4.87 and 4.88 , respectively. The initial control performance for both cases is poor as it takes more than 20 sample instants to start tracking the setpoint. The sum of predicted errors for the square wave change in setpoint is 37% larger than for the base case. However, the sum of predicted errors for the sawtooth function change in setpoint is 2% less than the base case showing a slight improvement in control performance.

For the square wave change in setpoint the parameter estimates are constant by the 120th sample instant as shown

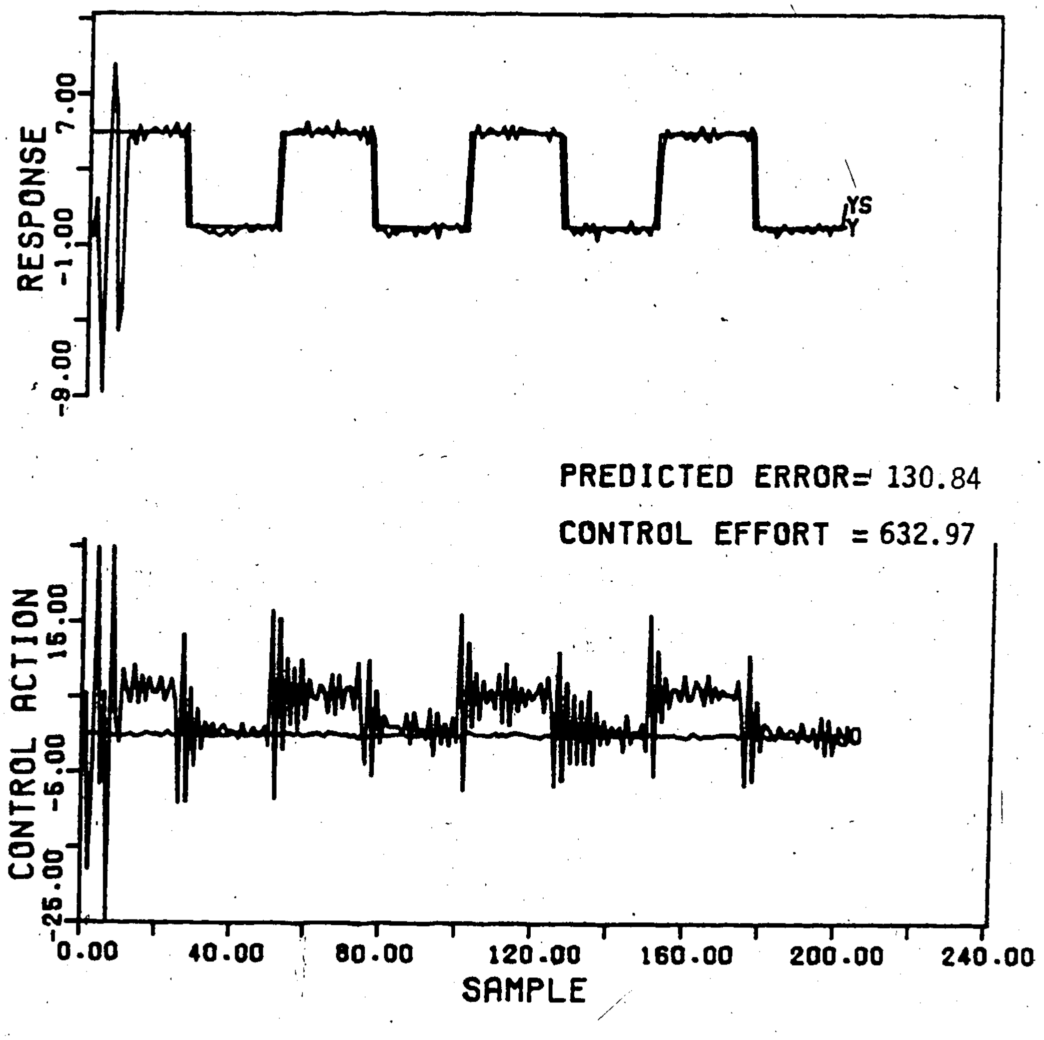


Figure 4.87 Control Response for Initial Parameter Estimates set to 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint

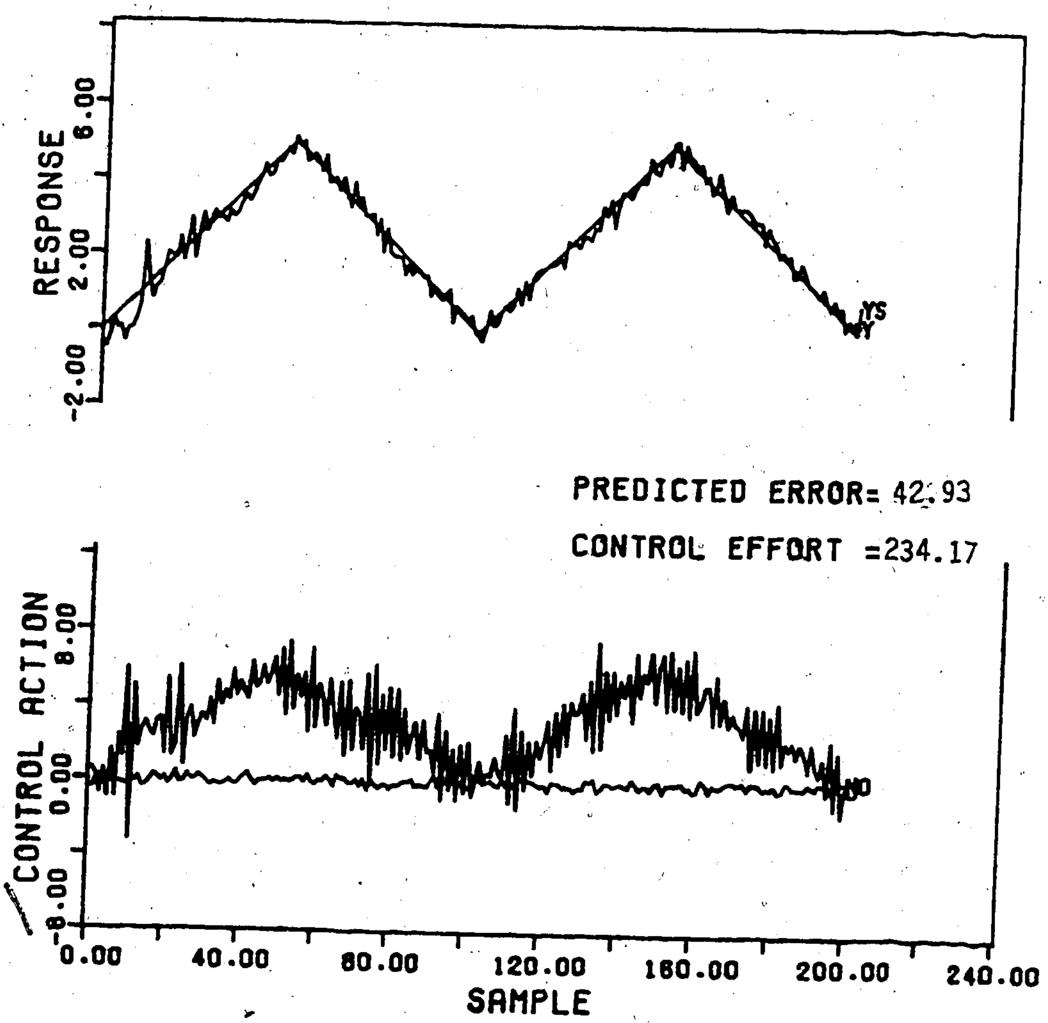


Figure 4.88 Control Response for Initial Parameter Estimates set to 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint

by the parameter estimates in Figure 4.89. However, by the 200th sample instant only the g_0 estimate is within 2% of the true value. For the sawtooth wave change in setpoint the pattern of parameter adaption shown in Figure 4.90 displays the large fluctuations in H estimates at the start of the adaption but the parameter estimates become constant by the 120th sample instant. The f_2 and g_0 estimates are within 4% of the true values by the 200th sample instant but the remaining parameter estimates have errors of over 20%. The H estimates are mirror images of each other and are not converging to the true values even after 200 sample instants.

The control performance that resulted for the square wave and sawtooth function changes in setpoint is presented for initial estimates of -0.5 is presented in Figures 4.91 and 4.92. As can be seen, the initial setpoint tracking is not satisfactory for either setpoint change. The control effort value for the square wave setpoint is 7% lower and the sum of predicted errors is 5.8% lower for initial estimates of 1.0 while for the sawtooth change in setpoint the control effort and sum of predicted error values were amost twice as large.

For the square wave change in setpoint, it can be seen from the parameter estimates plotted in Figure 4.93 that the F estimates are constant after 200 sample instants. In addition the values are within 6% of those for $\Theta(0)=1.0$ which were far from the true values. The G estimates

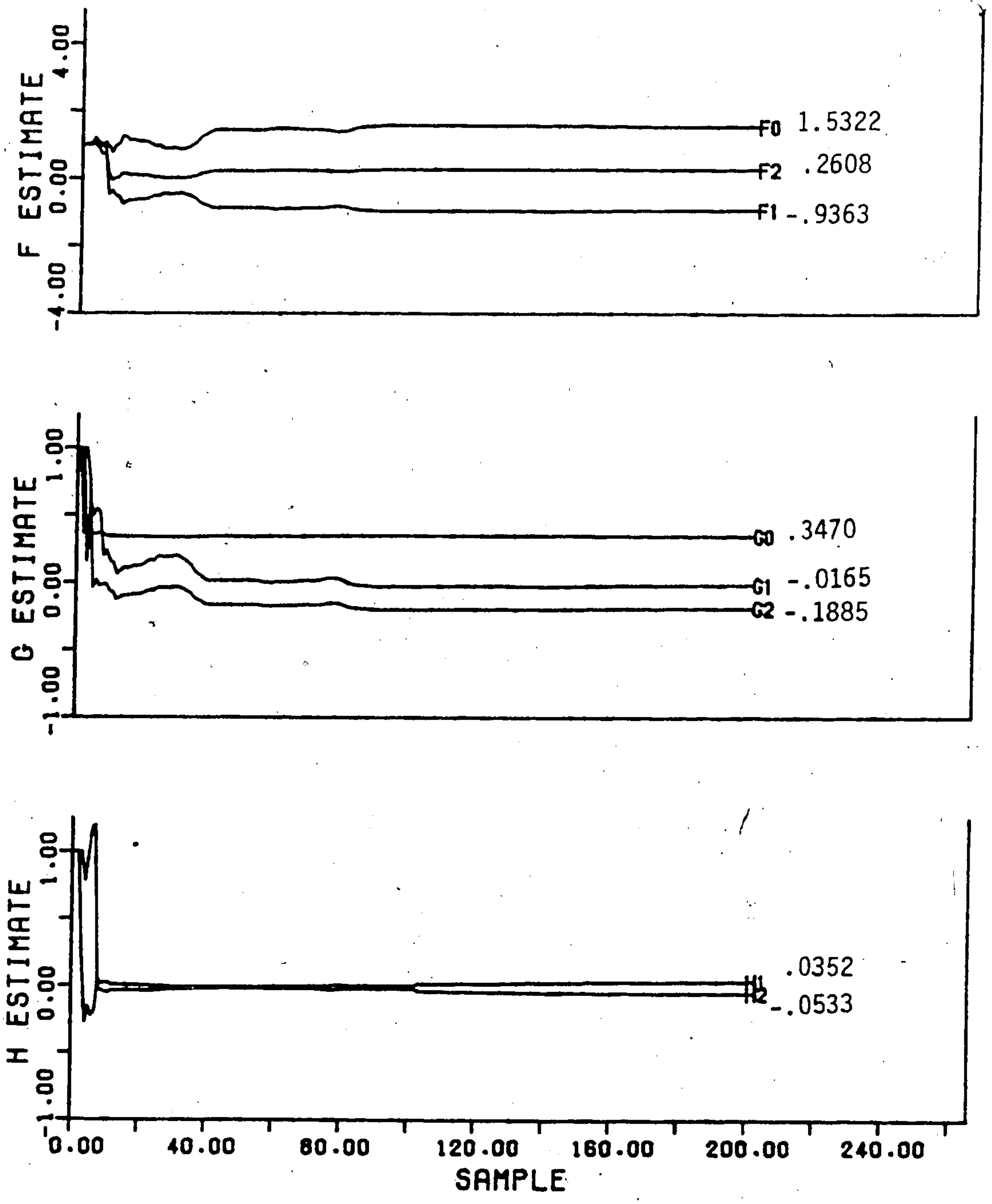


Figure 4.89 Parameter Estimates for Initial Parameter Estimates of 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint

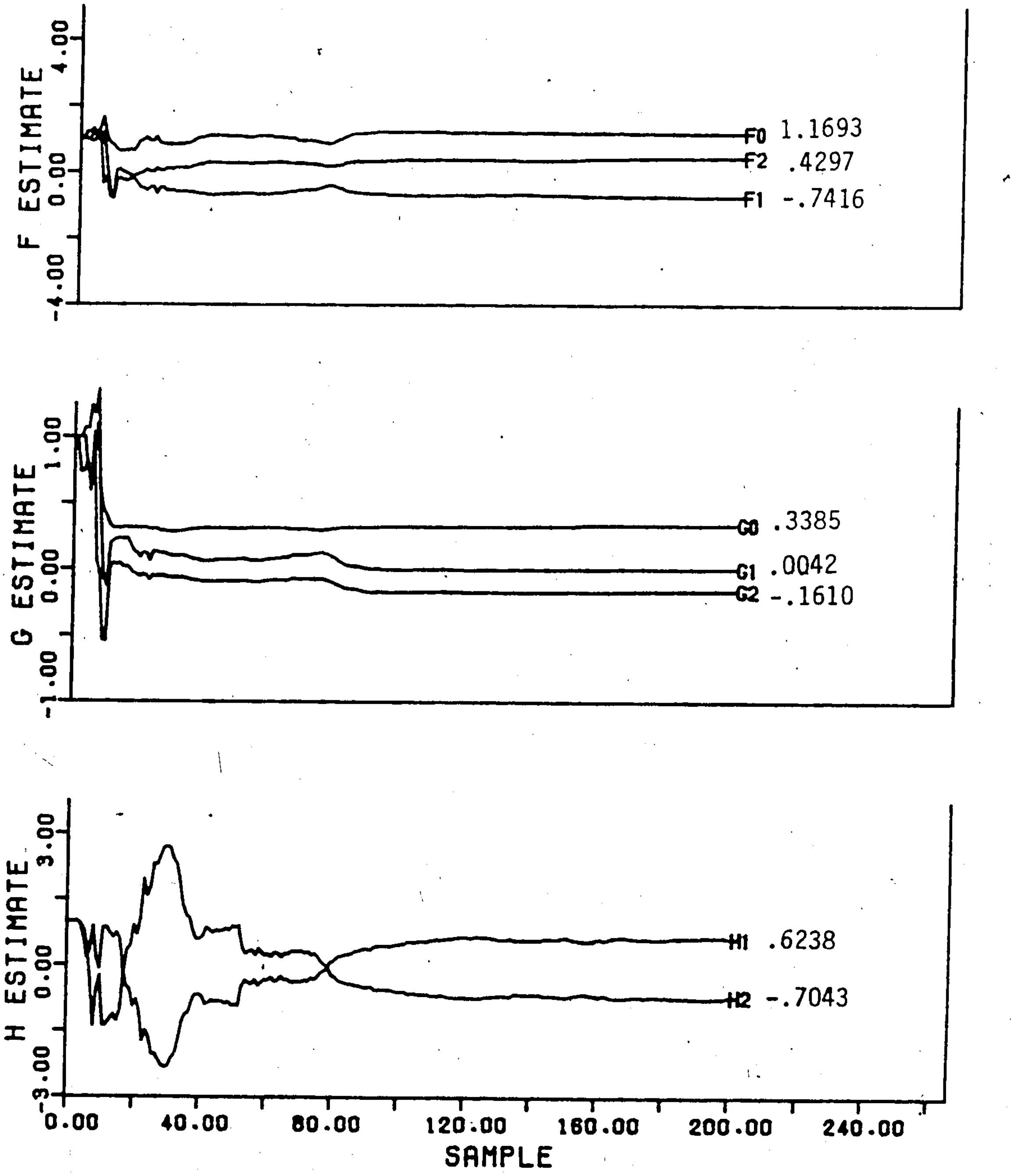


Figure 4.90 Parameter Estimates for Initial Parameter Estimates of 1.0 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint

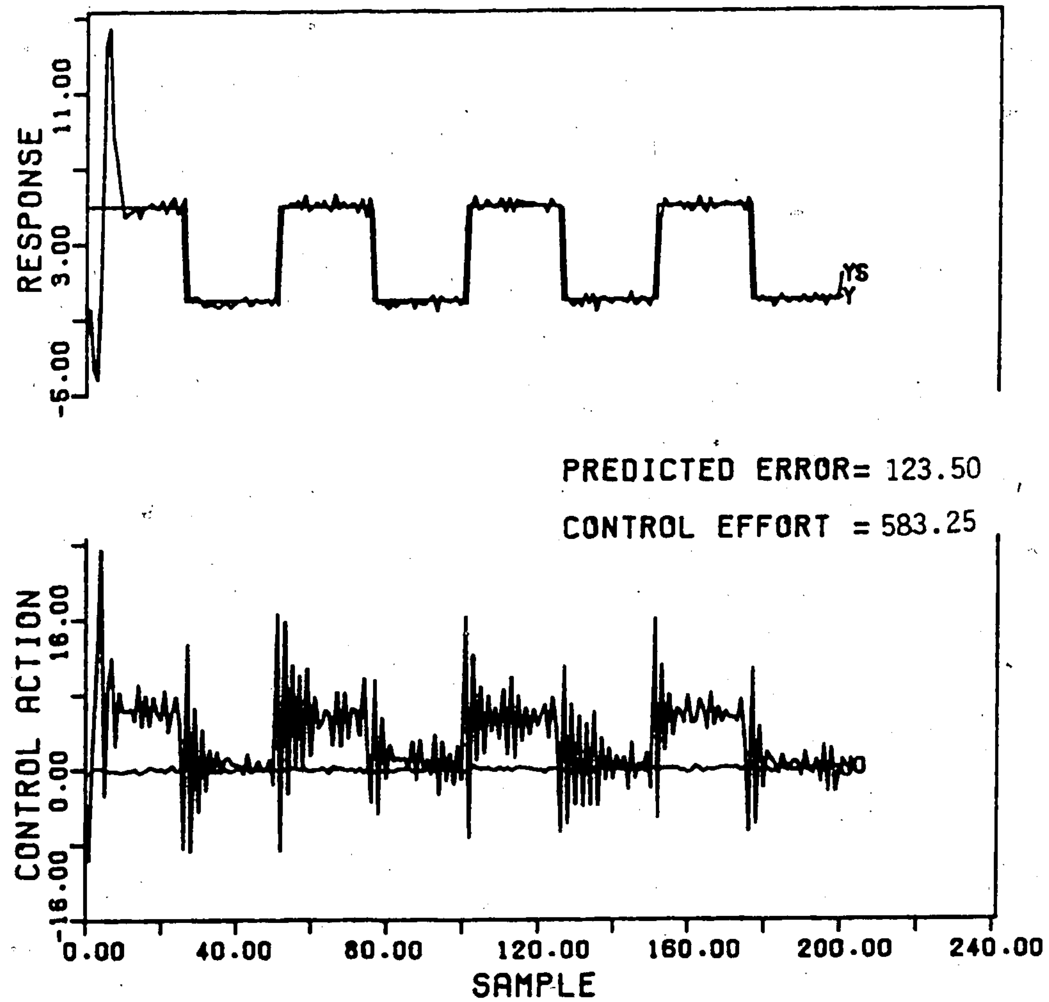


Figure 4.91 Control Response for Initial Parameter Estimates set to -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint

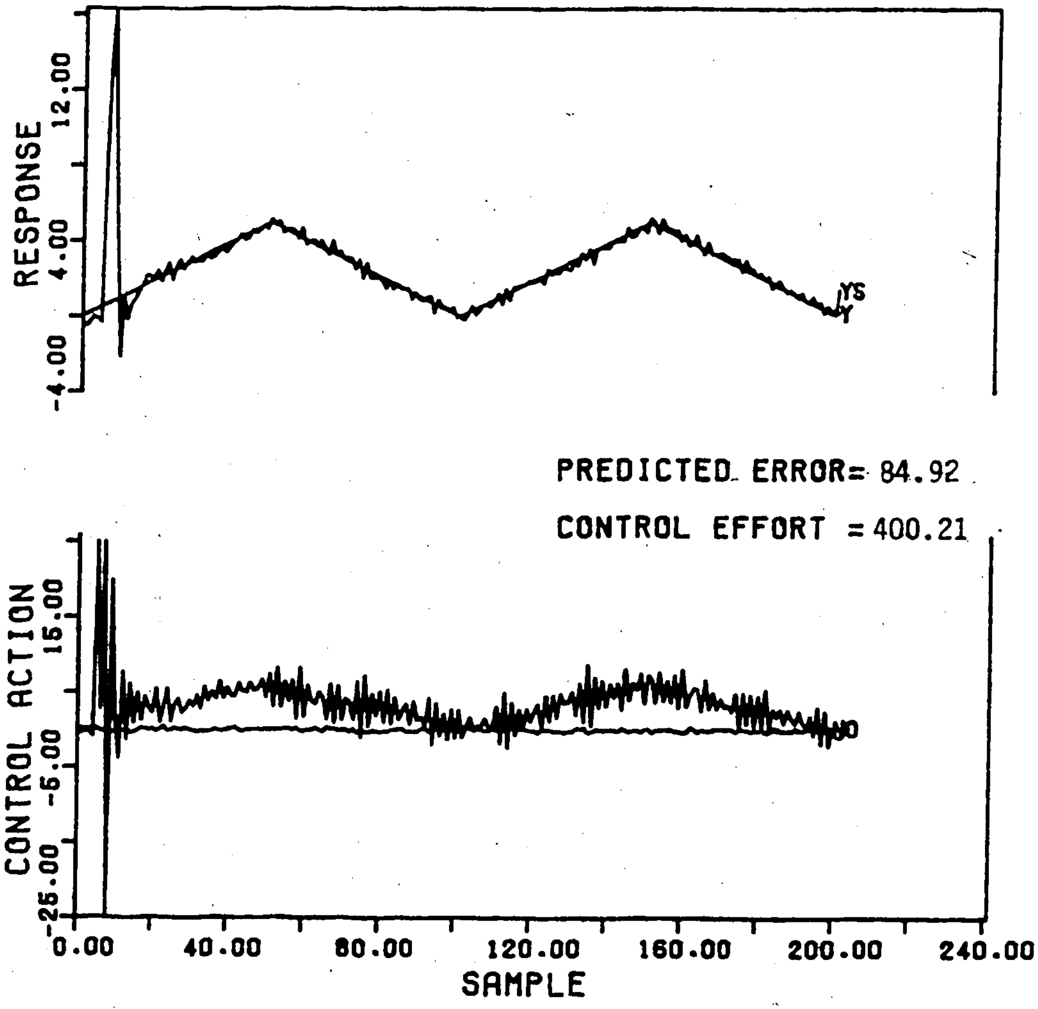


Figure 4.92 Control Response for Initial Parameter Estimates set to -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint

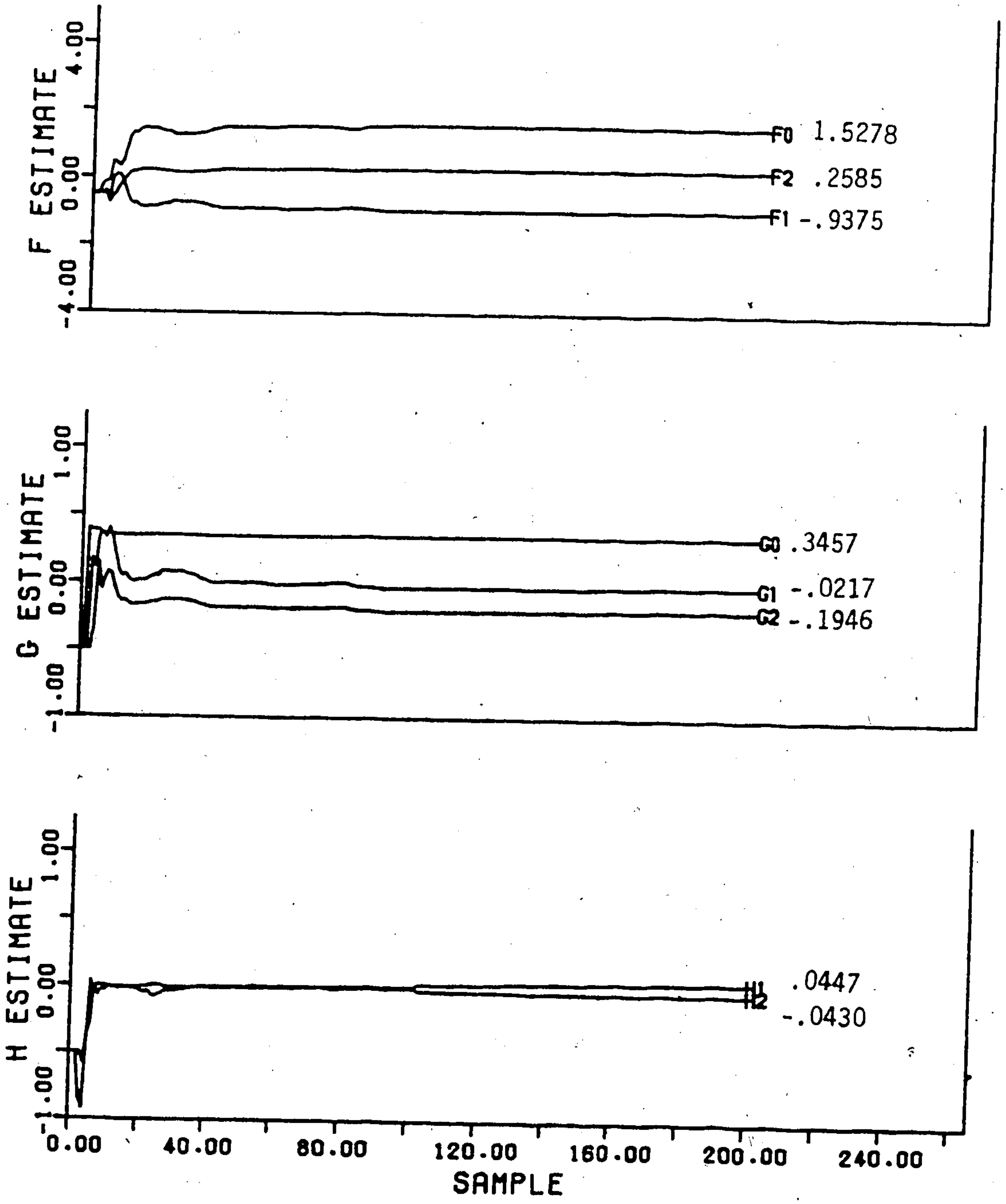


Figure 4.93 Parameter Estimates for Initial Parameter Estimates of -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Square Wave Change in Setpoint

displayed a similar adaption pattern to that when starting with the initial estimates of 1.0 and became constant by the 110th sample instant. The g_0 estimate after 200 sample instants is within 2% of the true value and it can be seen that the h_1 and h_2 estimates did not change noticeably after the 10th sample interval and do not appear to be approaching the true values of 0.79 and -0.37, respectively.

For the sawtooth wave change in setpoint the F estimates, plotted in Figure 4.94, increase slowly for the first 20 sample instants but the f_0 and f_1 estimates at 200th sample instant are larger than the true values including those obtained when $\theta(0)=1.0$. Although g_0 is 1% higher than the true value, g_1 is 33% and g_2 is 25% higher than the true value. The H estimates display greater changes than for the square wave change in setpoint and for a sawtooth change in setpoint when $\theta(0)=1.0$ but still do not approach the correct values.

Comparison of the base case results, starting with $\theta(0)=0.0$ and $g_0=0.3$, still in Figures 4.17 and 4.18 with the results when $\theta(0)=1.0$ it can be seen that the parameter estimates for the square wave and sawtooth function setpoint changes are similar after the first 10 sample instants. The average of errors of the parameter estimates after 200 sample instants for the sawtooth function setpoint change when $\theta(0)=0.0$ and $g_0=0.3$ were 37.5% which was 13.3% better than the estimates for $\theta(0)=-0.5$ and only 1% higher than when $\theta(0)=1.0$. The parameter estimates obtained by the

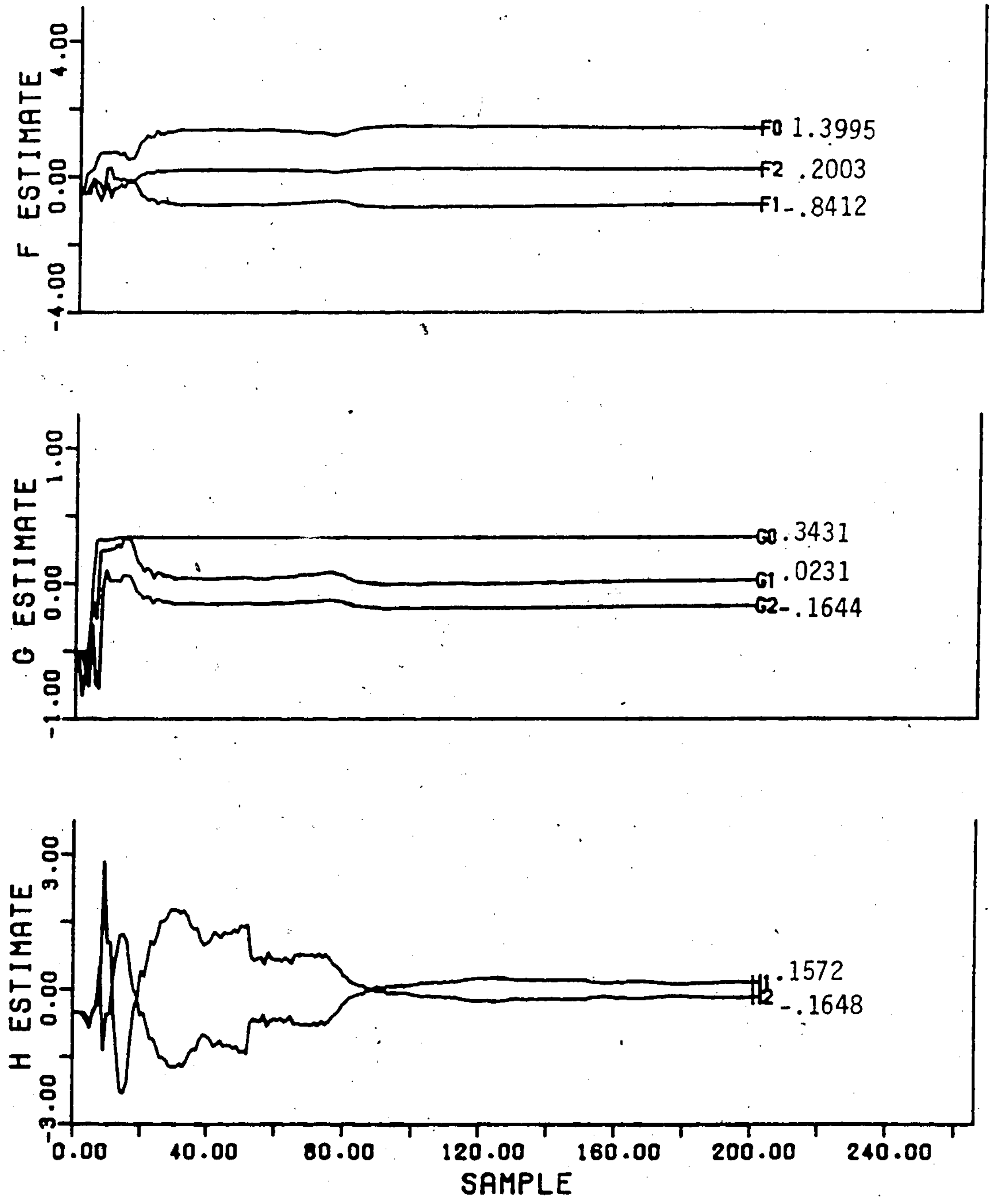


Figure 4.94 Parameter Estimates for Initial Parameter Estimates of -0.5 using the Recursive Upper Diagonal Factorization Estimator for a Sawtooth Function Change in Setpoint

square wave change in setpoint differed on average 44.4% from the true values which slightly better than those for the other two initial parameter estimate values. The best control performance for the least control effort is obtained for the square wave change in setpoint for $\Theta=0.0$ and $g_0=0.3$ but the sawtooth change in setpoint showed a 11.2% increase in the sum of predicted errors compared to the $\Theta=1.0$ case.

For the two nonzero initial parameter estimate tests, g_0 was initially set to 0.3 to determine if the poor initial g_0 estimate was the cause of the control performance deterioration. The results for $\Theta(0)=1.0$ and $g_0=0.3$ and for $\Theta(0)=1.0$ and $g_0=0.3$ when a sawtooth change in setpoint was used showed a decrease in control effort and sum of predicted errors when compared to the base case where $\Theta(0)=0.0$ and $g_0=0.3$. However, the results for the square wave change in setpoint showed marked increases of 25 to 45% in control effort and sum of predicted error values from the base case due to increased initial output errors. Although the sum of predicted errors for the sawtooth function setpoint when $\Theta(0)=0.0$ and g_0 was larger than for $\Theta=1.0$ the overall performance of these initial parameter estimates indicate that if only limited information about the true values of the parameter estimates is known the initial estimates should be set to 0.0 and the g_0 estimate should be set close to the true value to ensure fast convergence. The 0.0 value is ideal as it does not cause the initial calculated estimates to be of the incorrect sign and therefore begin the estimate trajectory in the opposite direction of the true value.

4.6 Convergence Problems of the Estimators

The simulation tests performed for all five parameter estimation methods for linear system studied show that the recursive upper diagonal factorization method is the most accurate and efficient algorithm. However, not all parameter identification techniques converged to parameters that yield good control performance. The unsatisfactory performance is largely due to the incorrect parameter estimates which resulted for various reasons. The convergence problems of the estimators is now discussed.

The incorrect H estimates obtained by the parameter estimation algorithms observed primarily for the first 1000 sample instants directly influences the F estimates, thus causing them to be incorrect as well. This can be appreciated from equation (2.6) of Section 2.3.3, which shows that the F' and E' polynomials are determined by

$$z^{d}\frac{C}{A} = z^{d}E' + \frac{F'}{A} \qquad (2.6)$$

Since H = -C, incorrect H estimates will give incorrect F estimates. Clarke [10], has shown that if C=1 the recursive least squares estimator does not yield biased estimates although the initial tuning may be unsatisfactory and the dynamics of C^{-1} and the convergence rate are related. This can be applied to the recursive square root, recursive upper diagonal factorization and recursive learning estimators as well since these algorithms are also based on the assumption that there is no correlation between the noise and the input and output. However, by the 3000th sample instant the recursive square root and recursive upper diagonal factorization estimators yield parameter estimates that are an average of ±10.6% from the true values for the square wave change in setpoint.

Another cause of the inaccuracy of the estimates can be attributed to the computer used for the estimation algorithm calculation. Bierman and Thornton [34] showed that the accuracy of the covariance algorithms deteriorates rapidly as the computer word length decreases. Therefore the estimates obtained by the recursive least squares estimator could be improved by using double precision arithmetic however, this would not be necessary for the recursive square root and recursive upper diagonal factorization methods. In fact it was found that the stability of the factorization algorithms was shown by their lack of sensitivity to the choice of a priori variance and process noise levels. In all simulations in this study, single precision arithmetic was used to keep the computation time and storage at a minimum. As expected the factorization techniques provided parameter estimates that could adapt to the true parameter values even with single precision arithmetic while the remaining methods did not perform satisfactorily.

The lack of a covariance matrix in the recursive learning estimator does not enable the measure of parameter confidence to be determined and thus parameter estimates are incorrect and the adaptive properties of the estimator are unsatisfactory.

4.7 Conclusions

Based on the extensive evaluation of the five parameter identification techniques the following conclusions can be made. From the setpoint tracking simulations, the recursive upper diagonal factorization estimator was shown to be the most accurate and efficient algorithm. The recursive square root estimator was equally as accurate as RUD however it was not as efficient as it required the calculation of square root which is more time consuming than arithmetic operations.

From the Q and R weighting tests it was observed that there was a reduction in the control effort for all the estimators. The control peformance for all but the recursive maximum likelihood estimator was improved when compared to the minimum variance control however, the RL estimator performance was not comparable to that of RLS, RSR and RUD.

When the system was subjected to an unmeasurable load disturbance the RLS, RSR and RUD estimator were able to estimate parameters that compensated for the disturbance. The use of recursive learning estimation resulted in the system becoming unstable while the recursive maximum likelihood estimation resulted in offset.

The recursive maximum likelihood estimator has proven that it is unable to estimate parameters to provide satisfactory control performance in closed loop feedback control.

Identification capabilities of the recursive upper diagonal factorization estimator were examined when various estimator variables were altered. The initial covariance matrix value should be chosen to compromise between good setpoint tracking and fast parameter convergence. The forgetting factor should be chosen based on the variation of parameters with time. The faster the parameters vary the closer the forgetting factor should be to 0.95 and the slower varying parameters would only need a forgetting factor close to 1.0. The initial estimate values should be selected as close to the true values as possible but if only limited information is available setting all estimates except $g_0(0)$ to 0.0 convergence of the parameters will occur within an acceptable amount of time. The choice of g_0 is very important as it determines the speed at which the parameters converge. A value as close as possible to the true value is the best initial estimate.

5. Nonlinear Simulation Results

A nonlinear simulation of a binary distillation column was used to further test the parameter identification techniques. The distillation column simulation program based on the nonlinear material and energy balance differential equations that was developed by Kan [35], based on previous work of Simonsmeier [36] and Bilec [37] and modified by Nazer [38] to study self-tuning control was used to further evaluate the parameter identification methods. This simulation allows an assesment of the techniques for their performance in the estimation of parameters that can compensate for the nonlinearities and interaction between the loops in order to achieve satisfactory control performance. In the simulation program, the control algorithm is based on an assumed column model given by

$$\begin{bmatrix} XD* \\ XB* \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \end{bmatrix} \begin{bmatrix} RE* \\ ST* \\ FE* \end{bmatrix} \qquad (5.1)$$

where XD* and XB* represent the deviation of the top and bottom product composition from the steady state operating conditions, respectively. RE*, ST* and FE* represent the deviation of the reflux, steam and feed flow rates from steady state operating conditions. The transfer functions, G_{ij} are of the following form

$$G_{ij}(s) = \frac{K_{P(ij)}\exp(-\tau_{d(ij)}s)}{(\tau_{P(ij)}s + 1)} \qquad (5.2)$$

where K_P is the gain of the process and τ_P is the time constant of the process and τ_d is the time delay of the process.

The parameter identification algorithms tested were the recursive least squares (RLS), recursive square root (RSR), recursive upper diagonal factorization (RUD) and recursive learning (RL) estimators. The recursive maximum likelihood estimator was not tested because the unsatisfactory results from the simulation study of the linear systems indicated that this method did not warrant further study.

Every nonlinear simulation was identical changing only the parameter identification technique. Each estimation method was tested for a feed flow rate disturbance of ±20% from the steady state value. The minimum variance self-tuning controller was introduced five sample intervals after the start of the simulation. The disturbance was implemented 10 sample intervals after the simulation began. The sampling rate for both loops was 3 minutes.

The starting values for the feed, reflux and steam rates were 18.2 g/s, 18.0 g/s and 14.7 g/s respectively. The top composition was set at a concentration of 96 mass percent methanol (MeOH) and the bottom composition 4.996% MeOH.

The results of this series of simulations are illustrated with plots of the feed, reflux and steam flow rates and the parameter estimates for both top and bottom loops. In each of the plots the flow rates will be indicated by ST for the steam, RE for the reflux and FE for feed with the top composition shown as XD and the bottom composition as XB.

The number of parameters for the two loops was different; the top loop had two F parameter estimates and three G, D* and G* parameter estimates whereas the bottom loop had three F parameter estimates and four G, D* and G* parameter estimates. The orders of the controller polynomials were selected to obtain the best control performance.

Just as for the study of the control of linear systems, the F and G parameter estimates are associated with the outputs and inputs, respectively. The G* parameter estimates correspond to the loop interactions and the D* estimates to the measurable disturbances or feed flow rate. All parameter estimates were initially set to zero with only the G0 estimate set to 0.5 to avoid division by zero in the control law.

The steam flow rate was the measurement used for the loop interaction of the top loop while the reflux flow rate was used by the bottom loop to determine the loop interaction.

The flow rates and compositions are plotted only until the compositions return to their steady state value or desired values in the case of a setpoint change. However, the parameter estimates are plotted for the entire simulation which was 600 minutes.

The true parameter values were not determined so the performance evaluation of the estimation methods is based on the control performance. The return of the top and bottom compositions to the setpoints with the minimum time away from the desired setpoint is taken as the criterion for good control performance.

5.1 Evaluation of the Recursive Least Squares Estimator

The results using the recursive least squares estimator for a 20% increase in feed flow rate are given in Figures 5.1, 5.2 and 5.3. The bottom composition required 80 minutes for return to the setpoint while the top composition required only 60 minutes. The greater deviations are observed for the bottom composition because the bottom loop is more sensitive to any feed disturbances. Both the reflux and steam flow rates did not fluctuate once steady state was reached and the composition setpoints were attained.

The parameter estimates for the bottom loop have converged by 200 minutes while the top loop converged after 180 minutes. There was more initial oscillation for the bottom loop parameter estimates, as shown by the results in Figure 5.3, than for the top loop as can be seen from Figure 5.2. This is no doubt due to the larger number of parameters that must be estimated, which requires more time for parameter convergence.

When the feed flow rate is abruptly decreased by 20% from the steady state value, it can be observed from the

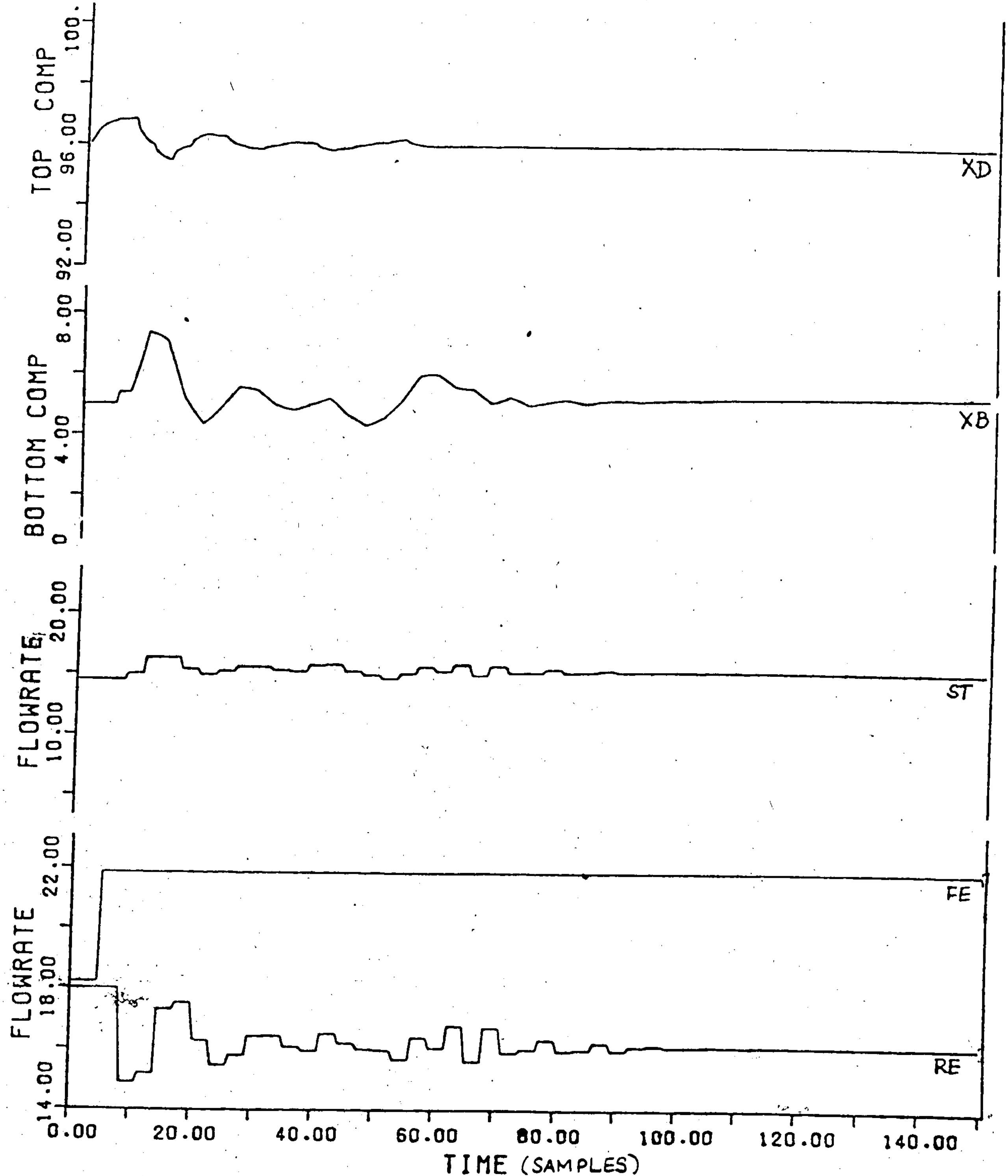


Figure 5.1 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation

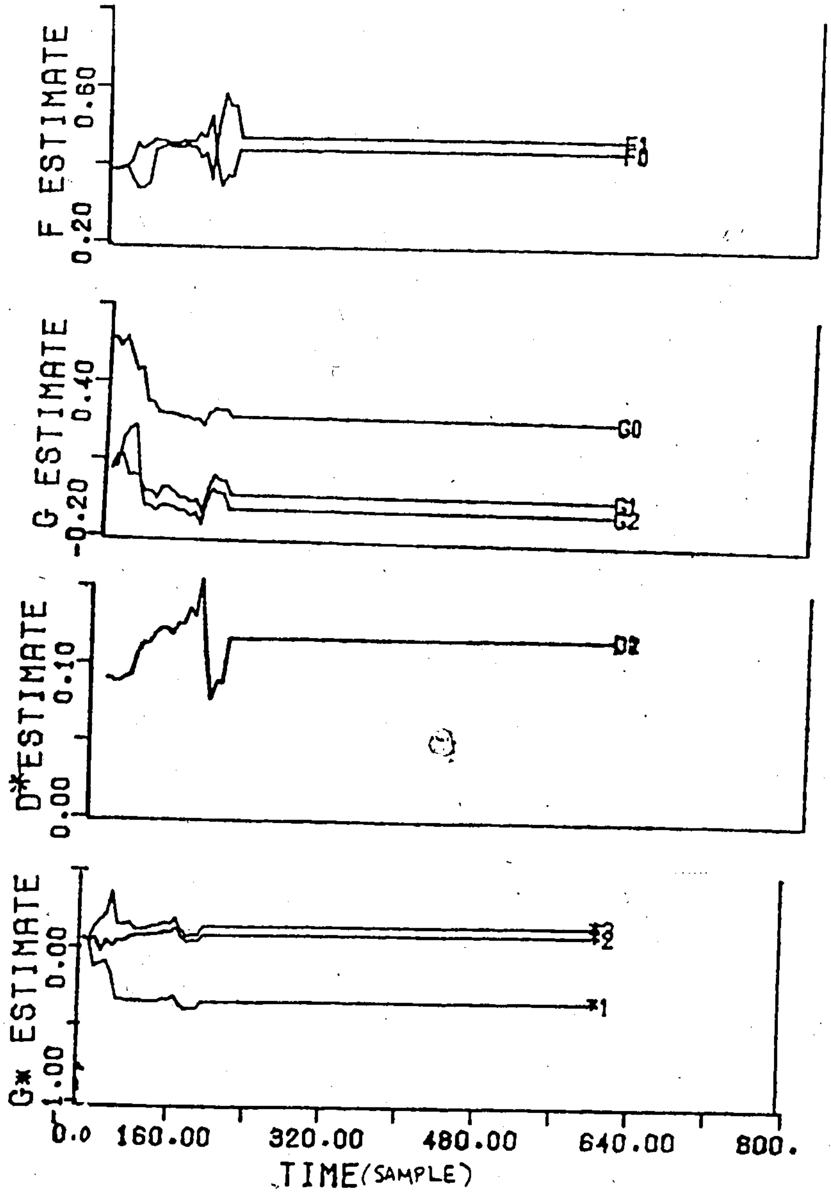


Figure 5.2 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation

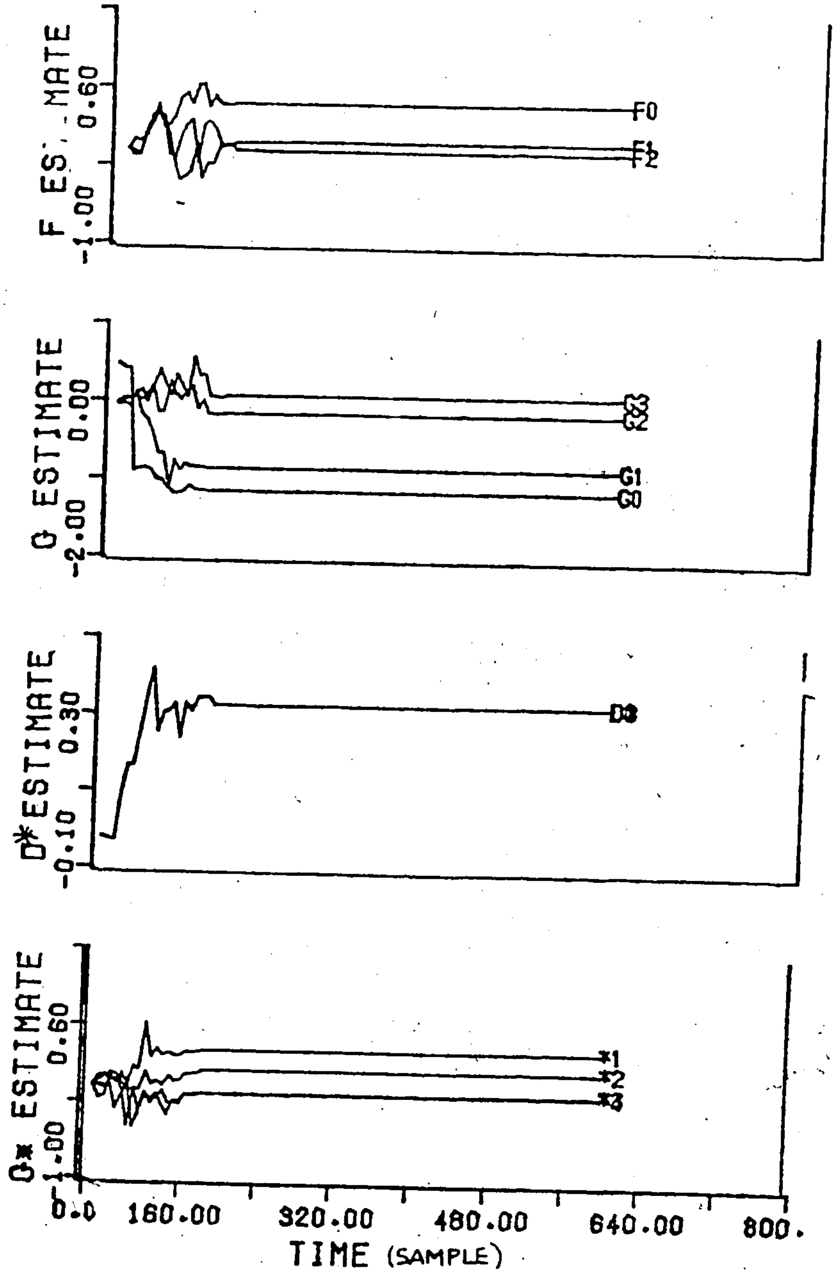


Figure 5.3 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Least Squares Estimation

results shown in Figure 5.4, that the top composition returned to the setpoint after 110 minutes while the bottom composition took 130 minutes. The increase in the amount of time to return to the setpoint indicates that a reduction in feed flow rate causes a greater upset in the composition control than an increase in feed. The deviations from the setpoint of both compositions were also considerably larger which accounts for the increased duration in time the compositions deviate from the setpoint. The reflux and steam flow rates showed larger fluctuations than for the 20% step increase in feed flow rate as a result of the large deviations in the setpoints.

The top and bottom loop parameter estimates presented in Figures 5.5 and 5.6, respectively, show greater initial changes than for a 20% increase in feed flow rate. However, the estimates have converged in the same amount of time as for the positive step increase in feed rate. The bottom loop parameter estimates again show larger initial changes than the top loop estimates.

Therefore the control performance observed using the recursive least squares estimator is acceptable for an increase and decrease in feed flow rate as the maximum deviations from the desired compositions were 1% for the top loop and 2.5% for the bottom loop. The output also returned to the setpoint after only 80 minutes.

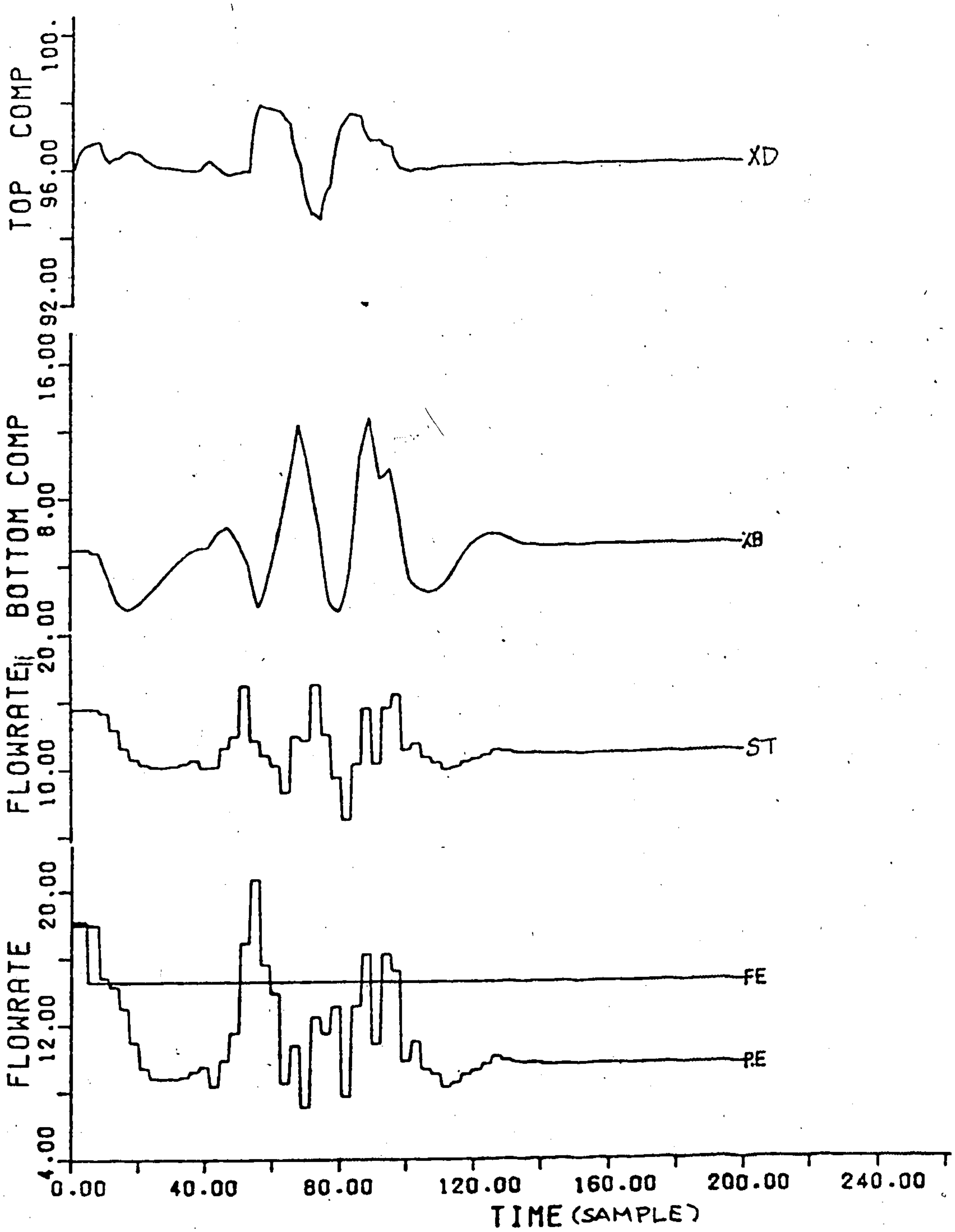


Figure 5.4 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation

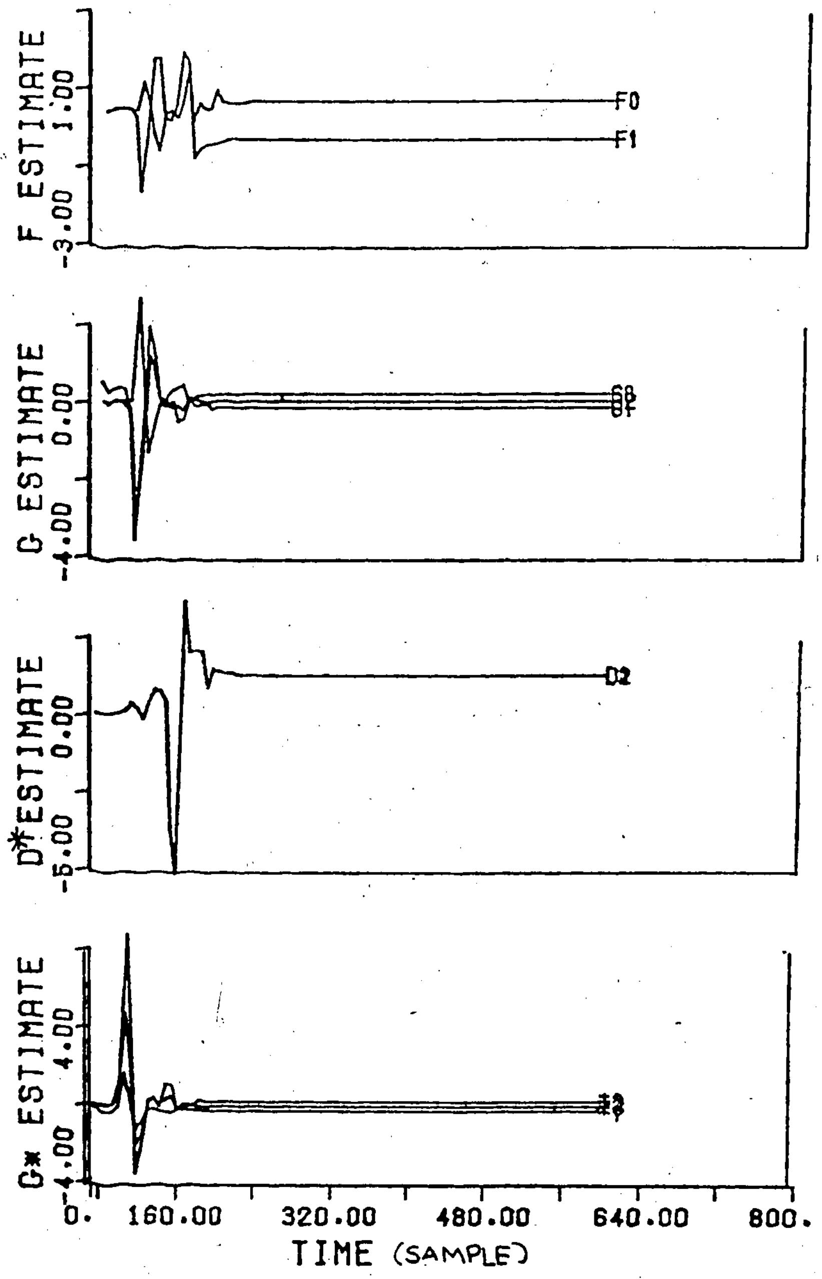


Figure 5.5 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation

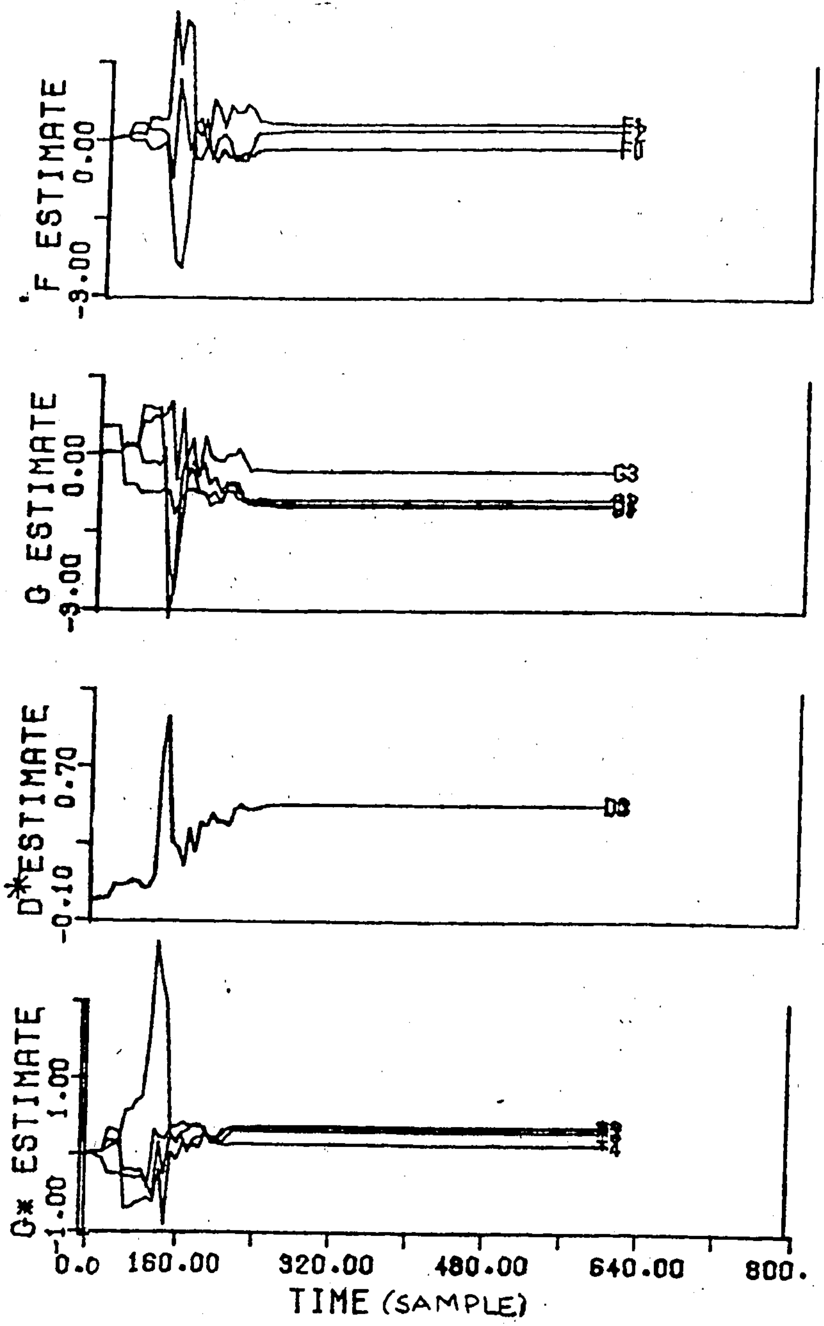


Figure 5.6 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Least Squares Estimation

5.2 Evaluation of the Recursive Square Root Estimator

The performance of the recursive square root estimator was also studied for 20% changes in feed flow rate. The flow rates and top and bottom composition behavior for a 20% step increase in feed rate is given in Figure 5.[illegible] can be seen, the bottom composition returned to th[illegible]int after 80 minutes but the top composition required [illegible]0 minutes of elapsed time. The difference in the performance of the RSR estimator compared to that using RLS was the smoother return to the desired setpoint exhibited for the RSR estimator as seen in Figure 5.7. This behavior is caused by the smaller changes in reflux and steam flow rates also shown in Figure 5.7.

The parameter estimates for the top loop, presented in Figure 5.8, show the same initial adaption pattern for the first 20 minutes, for the F, G and G* parameter estimates as for the 20% increase in feed flow rate using the RLS estimator. However, when the parameter estimates have converged the F0 and F1 estimates have grown in the opposite directions to the values obtained with the RLS estimator. The G1 and G2 parameter estimates also illustrate this same behavior. When the RSR estimator is used the D* parameter estimates no longer all have the same values and the adaption patterns are different, with the final D* estimate values larger than the values obtained using the RLS estimator. The G* parameter estimates also converge faster using the RSR estimator than the RLS method.

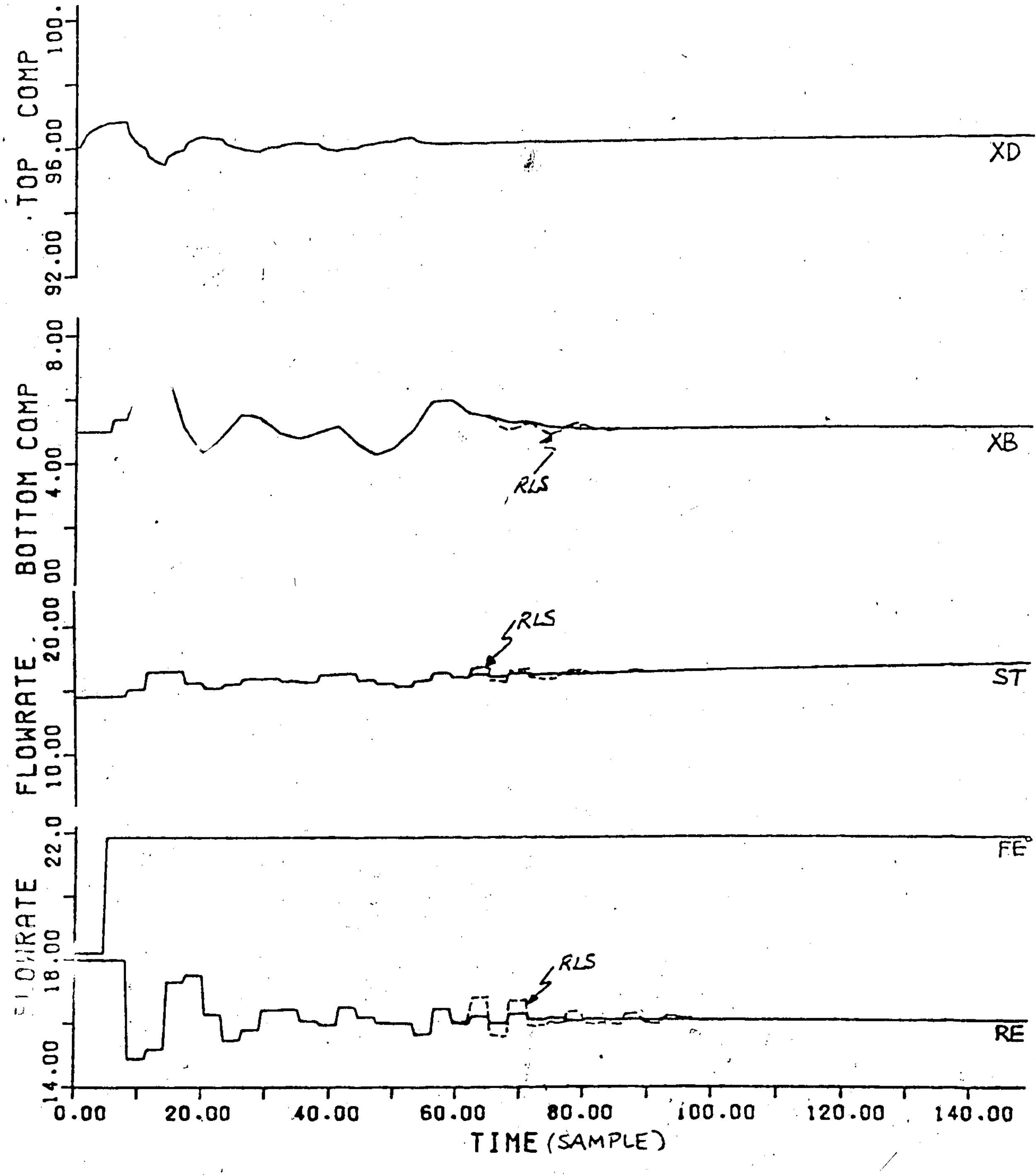


Figure 5.7 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation

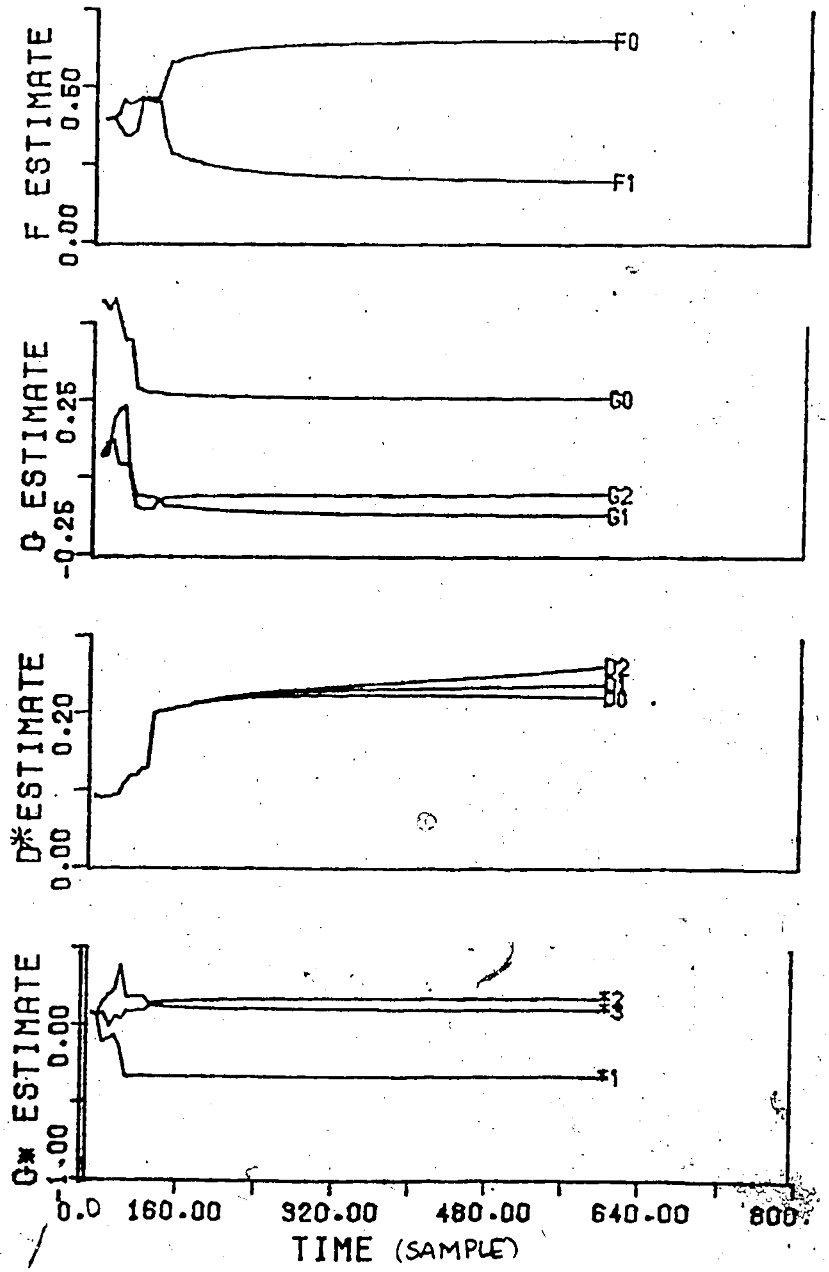


Figure 5.8 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation

The bottom loop parameter estimates found in Figure 5.9 exhibit a different adaption pattern than resulted using RLS estimation. Comparison of the final estimate values with the values obtained using the RLS estimator given in Figure 5.3 shows that the F1 and F2 as well as the G2 and G3 parameter estimate values have changed with the F2 and G2 now having the larger values. The D* parameter estimates are no longer equal as seen for the top loop parameter estimates and the G3* and G4* estimates are not equal as is the case when the RLS estimator is used.

A step decrease of 20% in feed flow rate caused the top composition to deviate from the setpoint for 80 minutes and the bottom composition for 120 minutes as can be observed from Figure 5.10. These times represent a reduction in the time required for the composition to return to the setpoint compared to the performance using RLS identification. The top composition showed smaller deviations from the setpoint than with the RLS estimation and both top and bottom composition response curves had one less oscillation using the RSR estimator. As seen in Figure 5.10, the steam and reflux flow rates reached steady state values in 100 minutes. Comparison of the flow rates using the RSR estimator with those using the RLS, reveals that only for the first 50 minutes are the flow changes similar. It is to be noted that the steam flow rate did not reach as high a value during the elapsed times using the RSR estimation as was the case for RLS estimation. The flow rates reached new

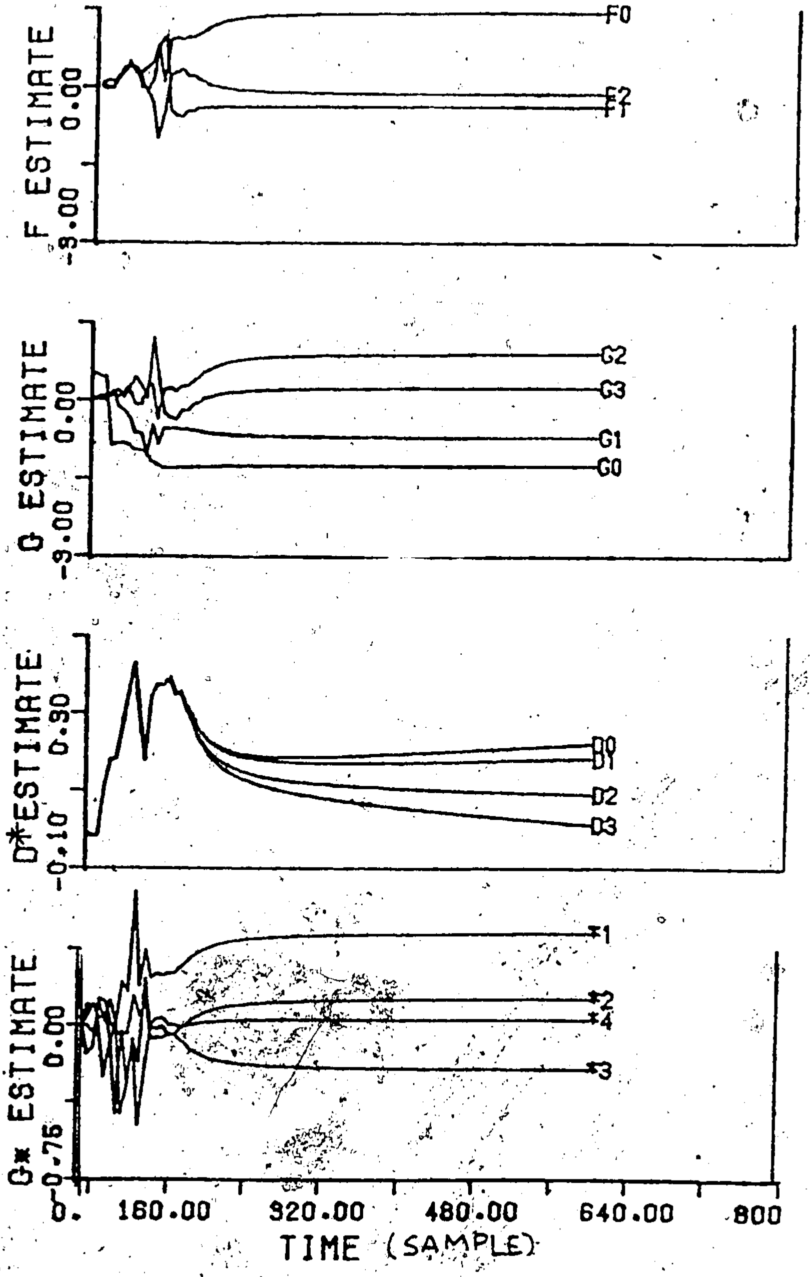


Figure 5.9 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Square Root Estimation

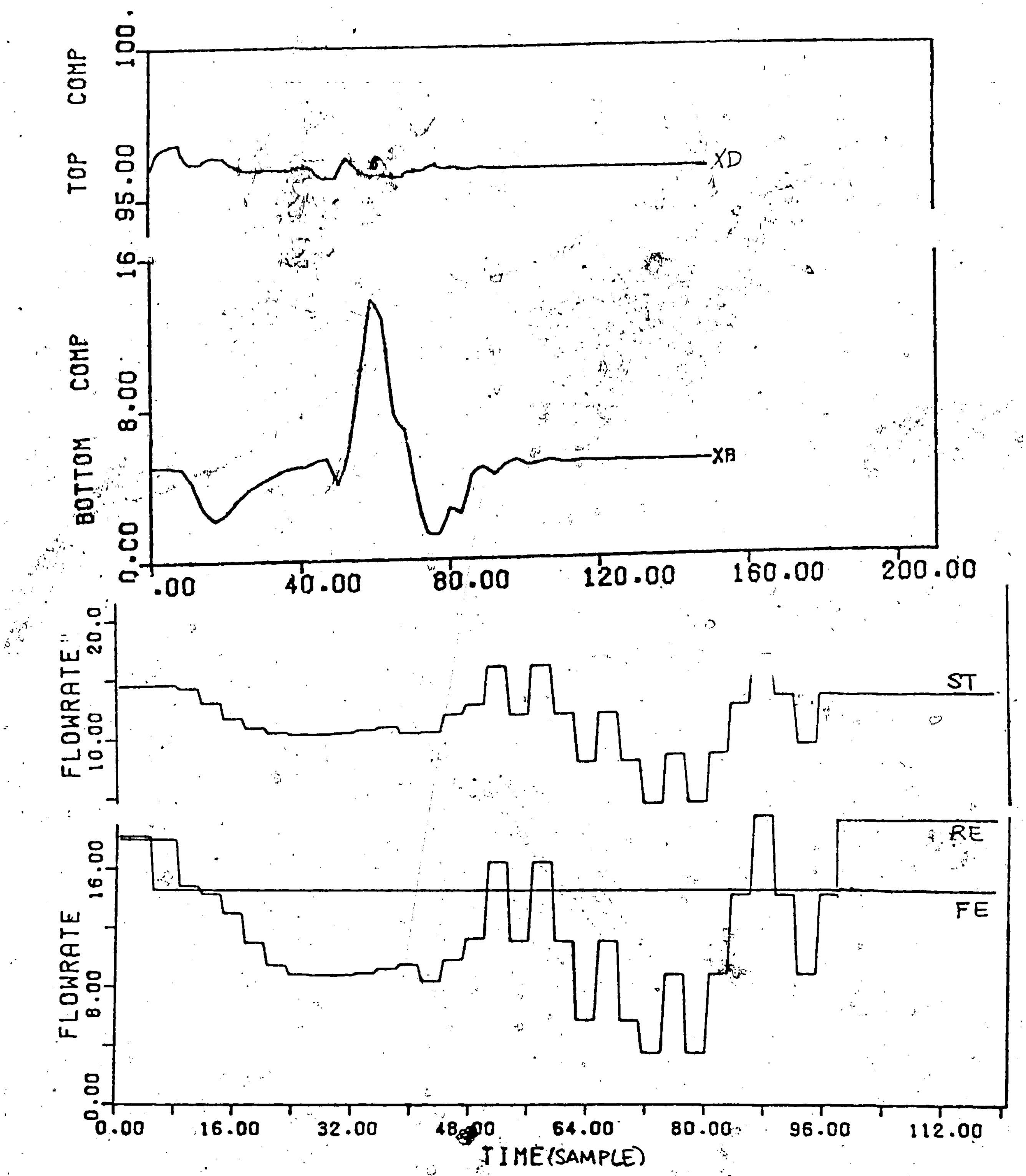


Figure 5.10 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation

steady state values in only 100 minutes, 25 minutes sooner than was the case using RLS estimation.

The top loop parameter estimates found in Figure 5.11 show a substantial reduction in initial estimate value changes for the G, D* and G* estimates as well as different adaption patterns compared to those in Figure 5.5. All top loop parameter estimates have converged by 200 minutes and furthermore the D* values are all equal to each other.

The parameter estimates for the bottom loop seen in Figure 5.12 show that unlike the top loop parameter estimates the initial changes for the RSR estimates are larger for all but the D* estimates than those using the RLS estimator. The parameter estimates using both estimators, however, do converge after 200 minutes.

Use of the RSR estimator rather than the RLS estimator does not speed up the parameter convergence but results in improved control performance.

5.3 Evaluation of the Recursive Upper Diagonal Factorization Estimator

The results obtained in studying the performance of the recursive upper diagonal factorization estimator for control of the column were found to be virtually identical to those obtained using the RSR estimator for control of the linear systems. Since this behavior was observed for both the increase and decrease in feed flow rate, no results are shown.

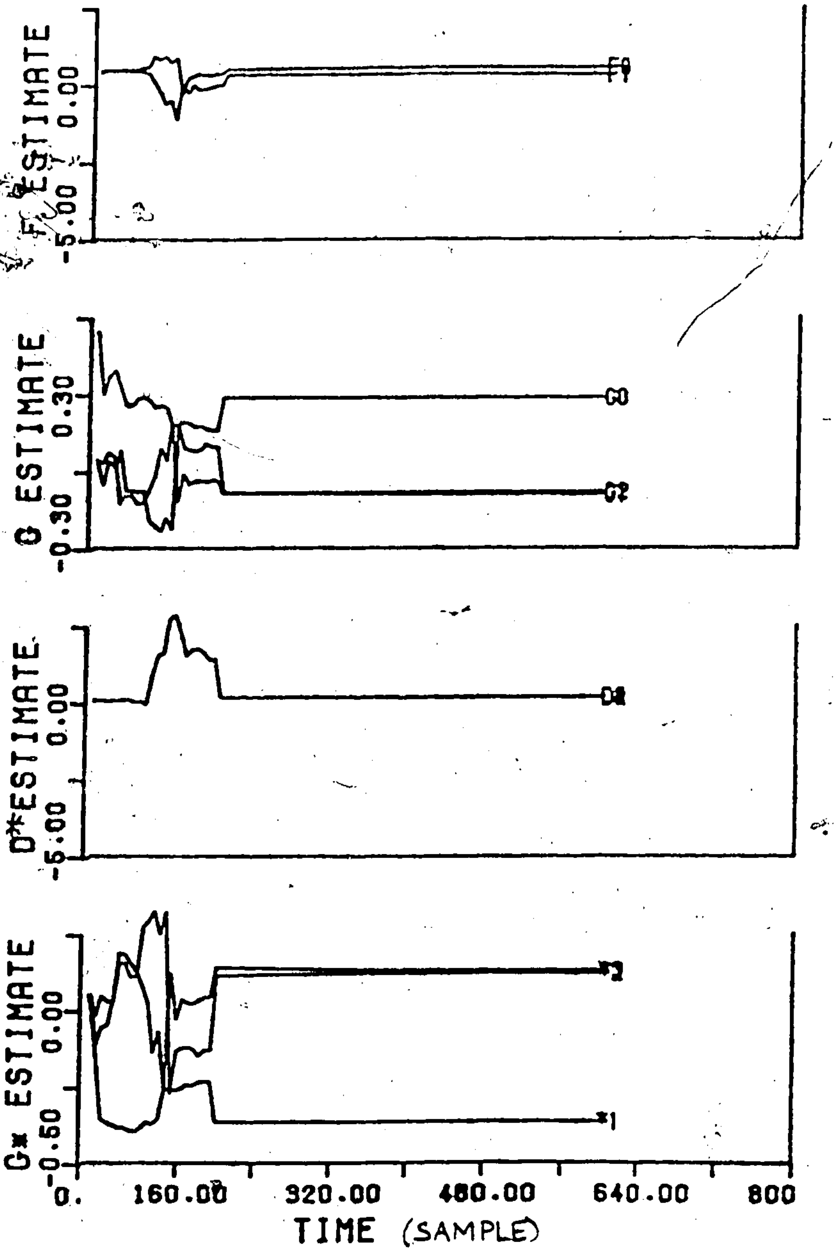


Figure 5.11 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation

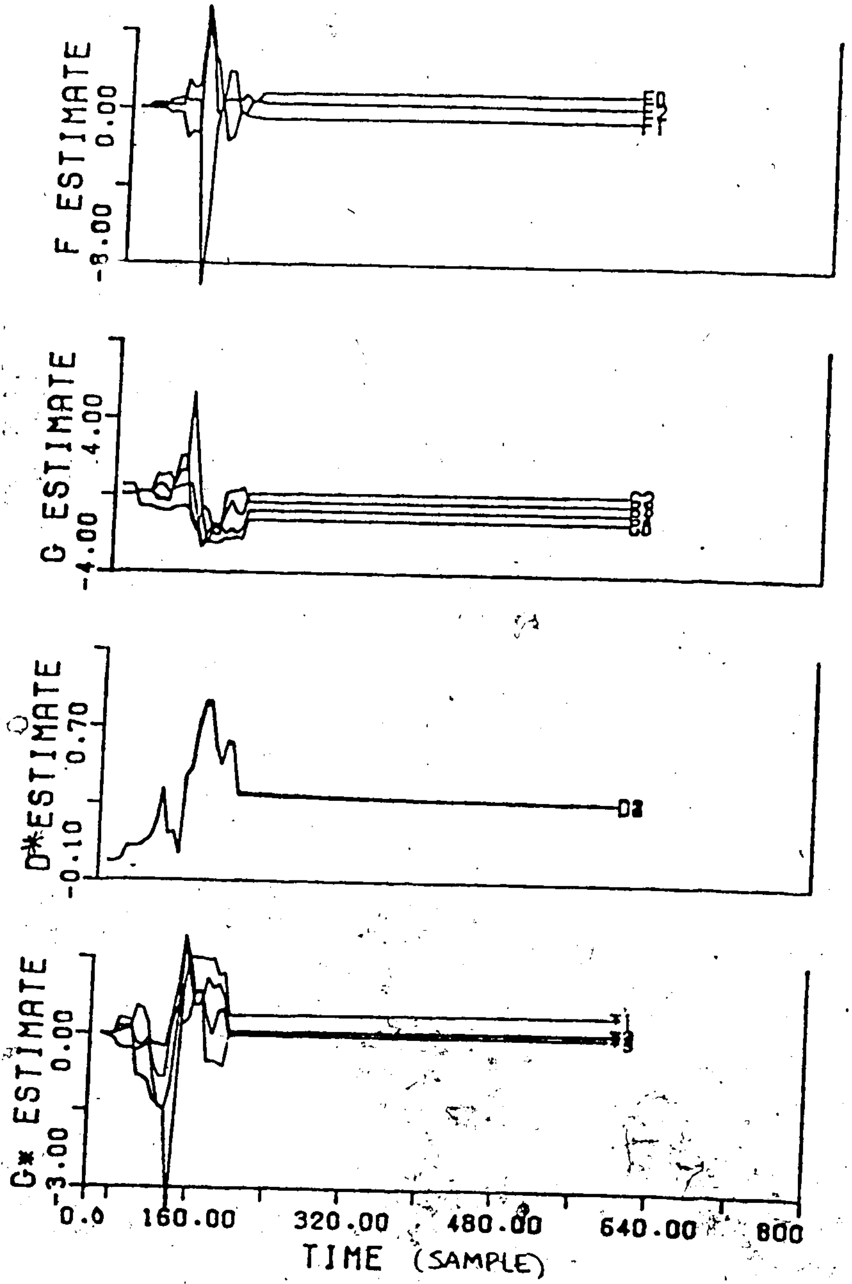


Figure 5.12 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Square Root Estimation

5.4 Evaluation of the Recursive Learning Estimator

The final identification technique tested was the recursive learning estimator. As can be seen from the results in Figure 5.13 neither the top or bottom composition returned to the desired setpoint after the feed flow rate was increased by 20%. The top composition was calculated to be at the limit of 100% MeOH and the bottom composition to 70% MeOH with the steam and reflux flow rates reaching the upper limits and remaining at those values.

The top loop parameter estimates are presented in Figure 5.14. It can be seen that there is virtually no change in the parameters from the initial value and all F parameter estimates are equal to each other as are all D* estimates. Only the G1 and G2 estimates remained equal to each other, the G0 estimate was a different value as the initial value was not zero which was the case for the G1 and G2 estimates.

The parameter estimates for the bottom loop, except the D* estimates, do not remain equal to each other as seen in Figure 5.15. All parameter estimates show an increase in value and converge in 100 minutes.

A 20% step reduction in feed flow rate using the recursive learning estimator resulted in the same final top and bottom compositions as for the 20% feed flow rate increase and the same final reflux and steam flow rates resulted although the overall trends were different as can be seen from the results in Figure 5.16.

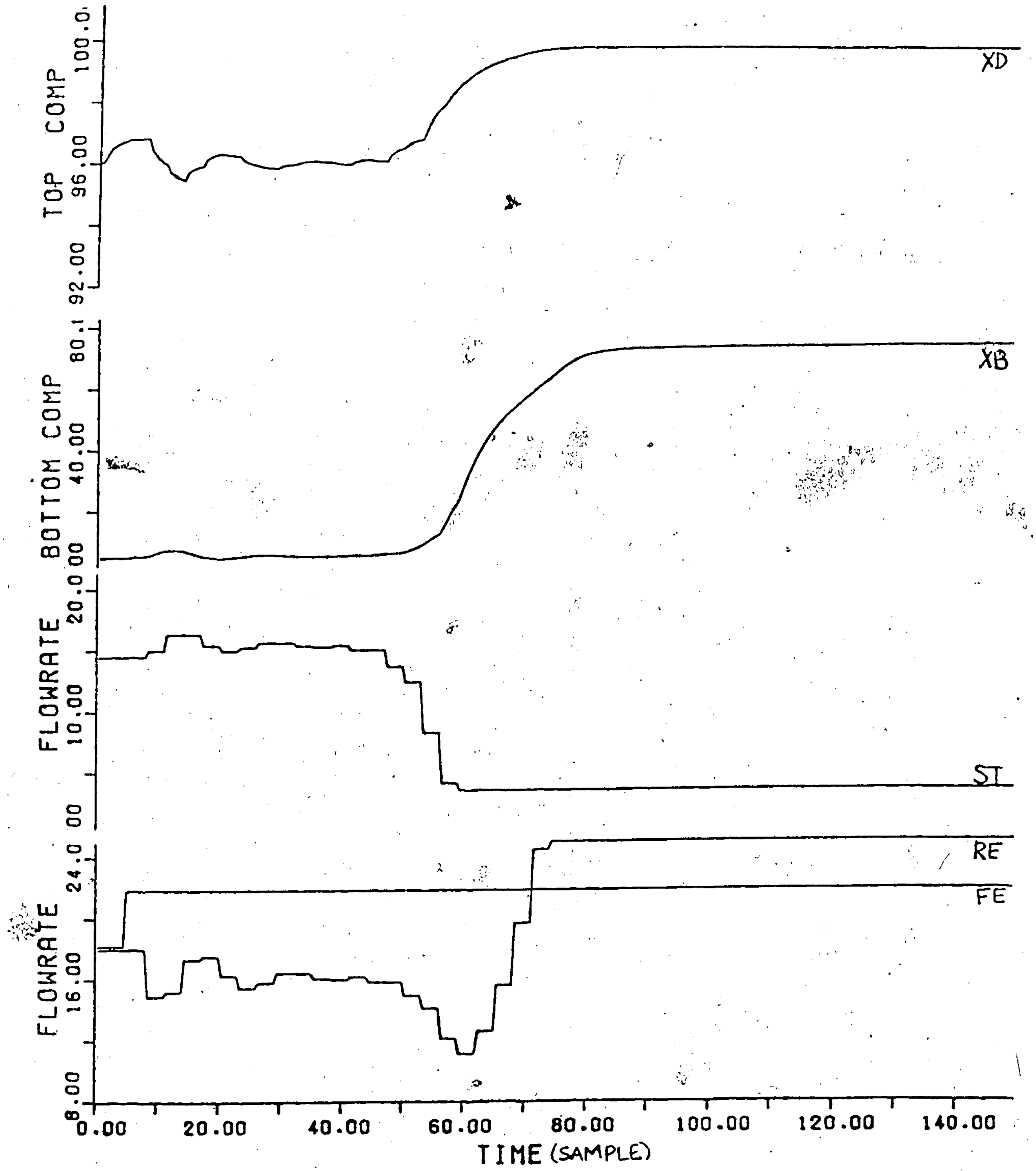


Figure 5.13 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation

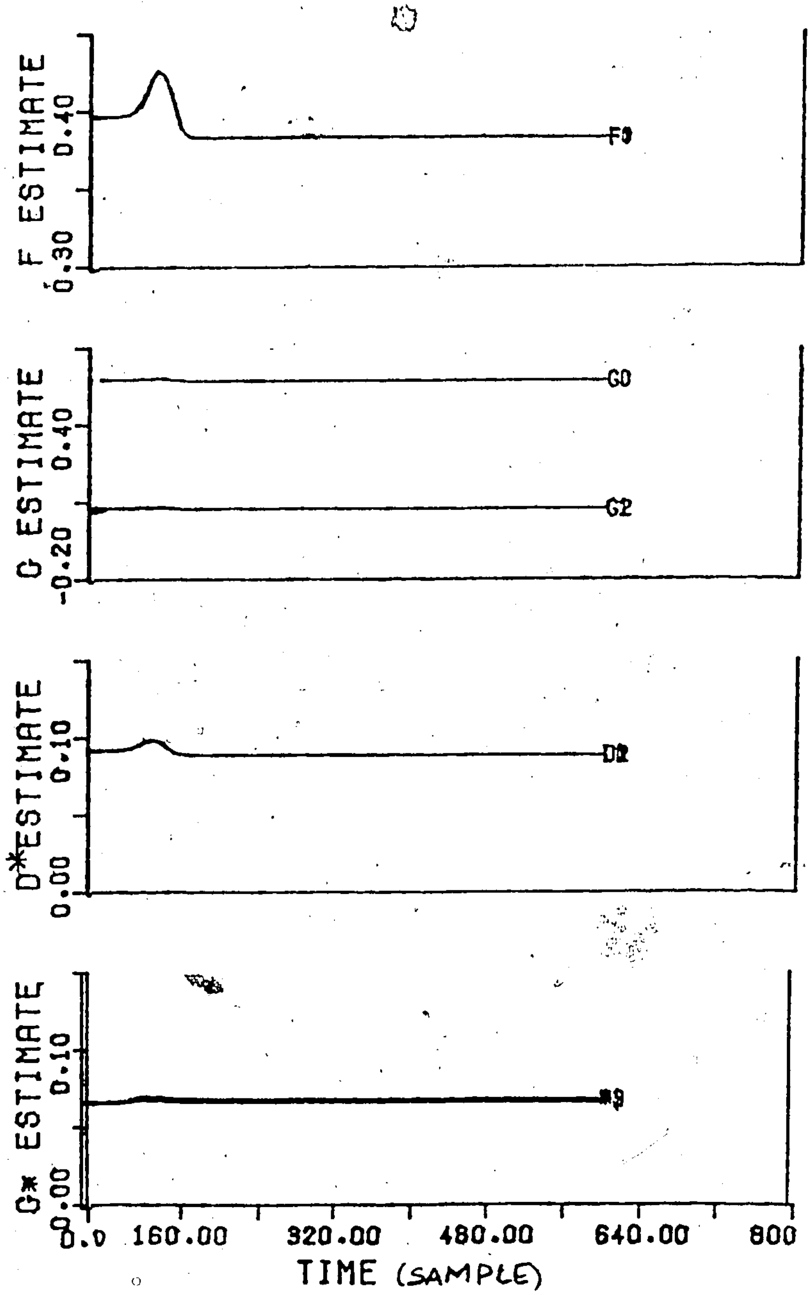


Figure 5.14 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation

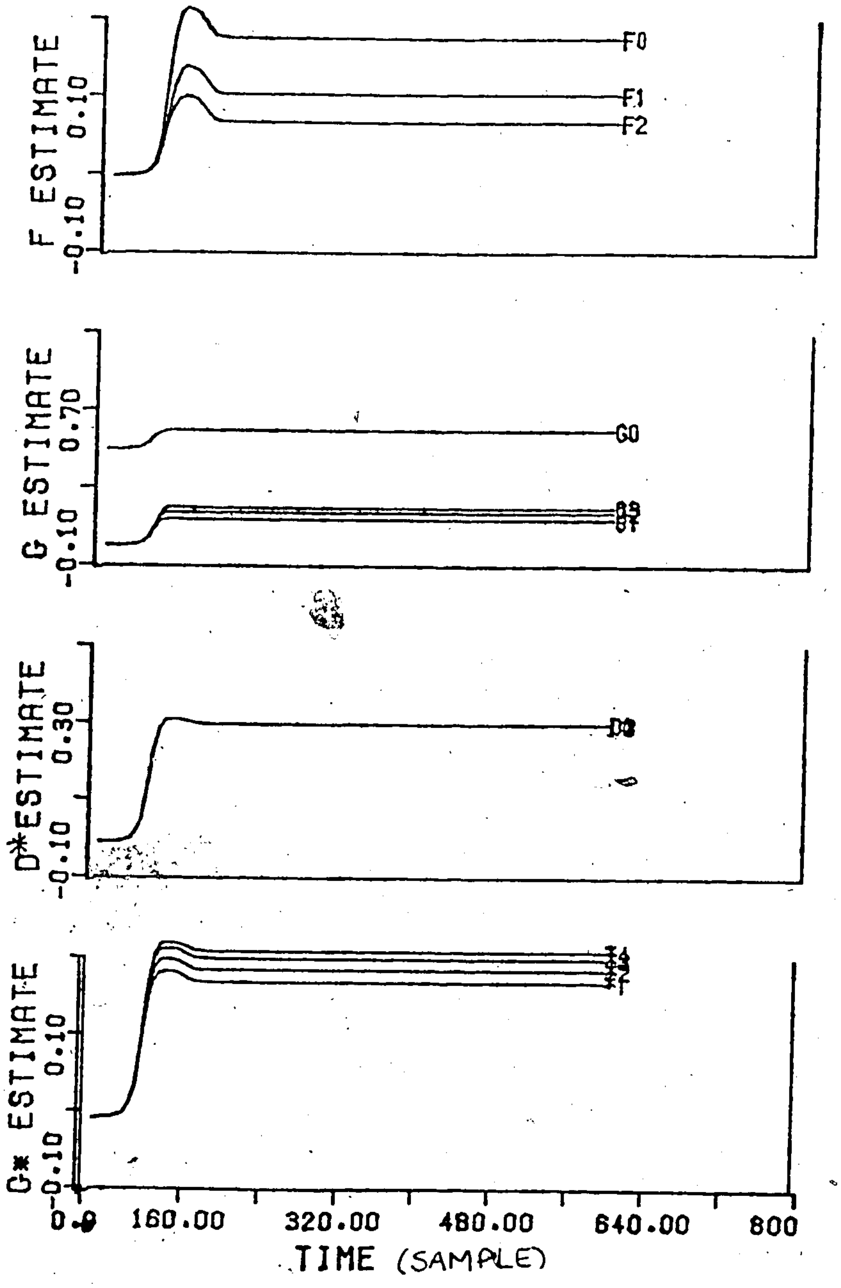


Figure 5.15 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Increase in Feed Flow Rate using Recursive Learning Estimation

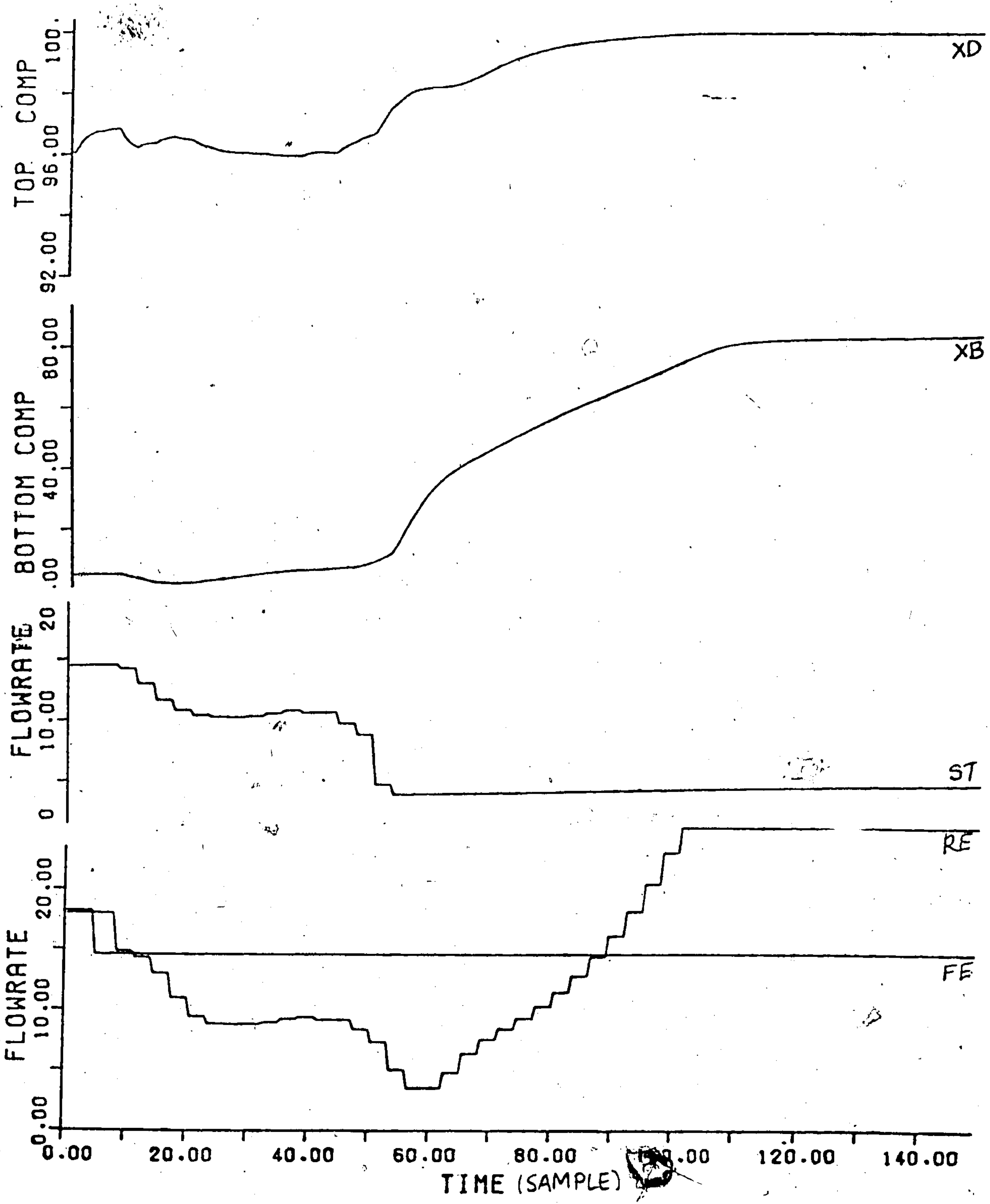


Figure 5.16 Simulated Response to the Binary Distillation Column Subjected to a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation

The parameter estimates for both the top and bottom loop found in Figures 5.17 and 5.18, respectively, show the same adaption patterns as for the 20% increase in feed flow rate. The only noticeable difference was the smaller initial changes exhibited by the bottom loop estimates before converging.

5.5 Summary

In summary, the results for column control show that the RSR and RUD estimators are the most accurate identification techniques. However, the RUD method would be chosen over the RSR estimator as no square root calculation is necessary which makes the algorithm more efficient. These results are in agreement with those from simulation of the control of linear systems. Further simulations using the RUD estimator were performed for feed disturbances of only ±10% in magnitude. As expected, the deviations from the desired setpoints were smaller and therefore the control performance was better than for the larger disturbances as the compositions returned to the desired values in a shorter elapsed time.

For the 10% increase in feed flow rate the initial adaption pattern for all but the G* top loop parameter estimates were the same as those for the 20% increase in feed flow rate as well as the final values. The G* estimates have different initial adaption patterns and the G2 and G3 estimates converge to practically equal values. The bottom

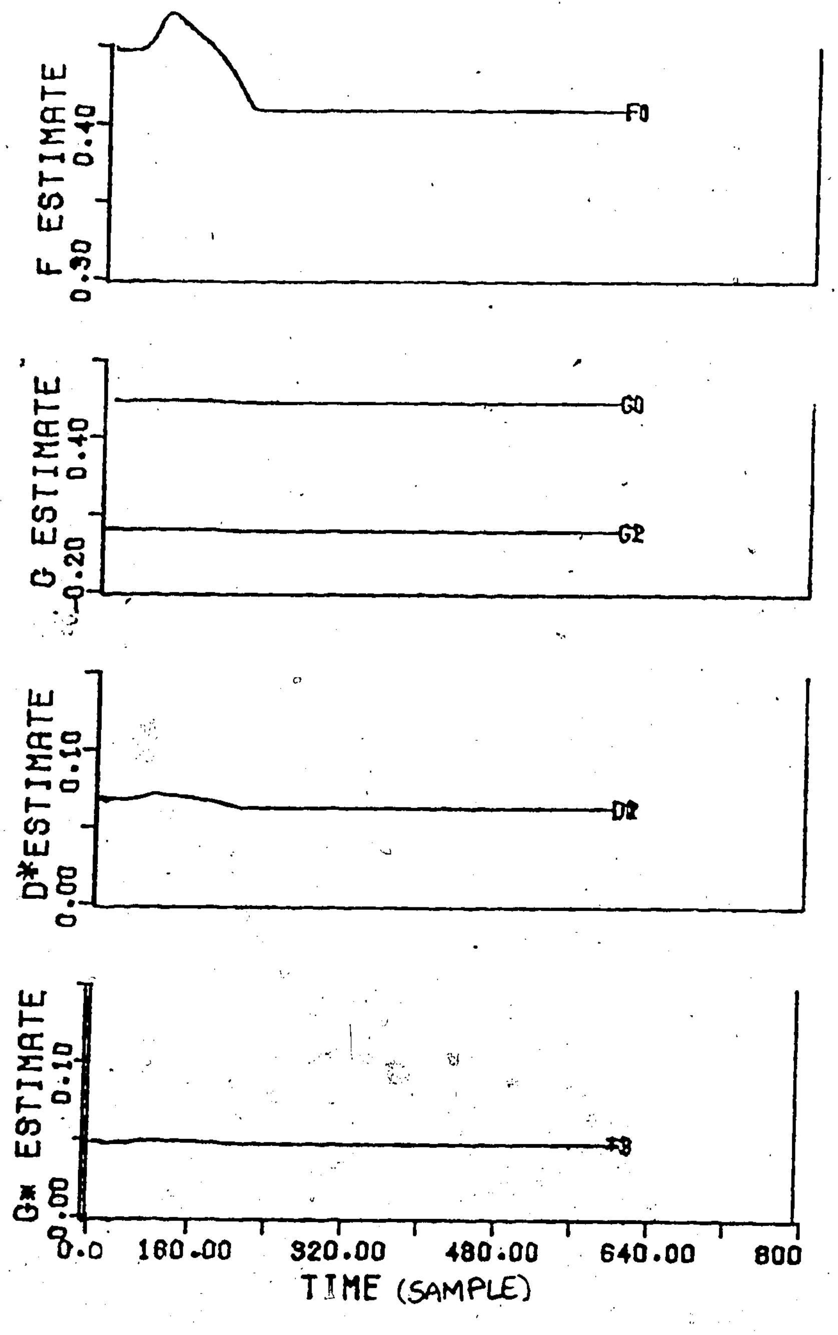


Figure 5.17 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation

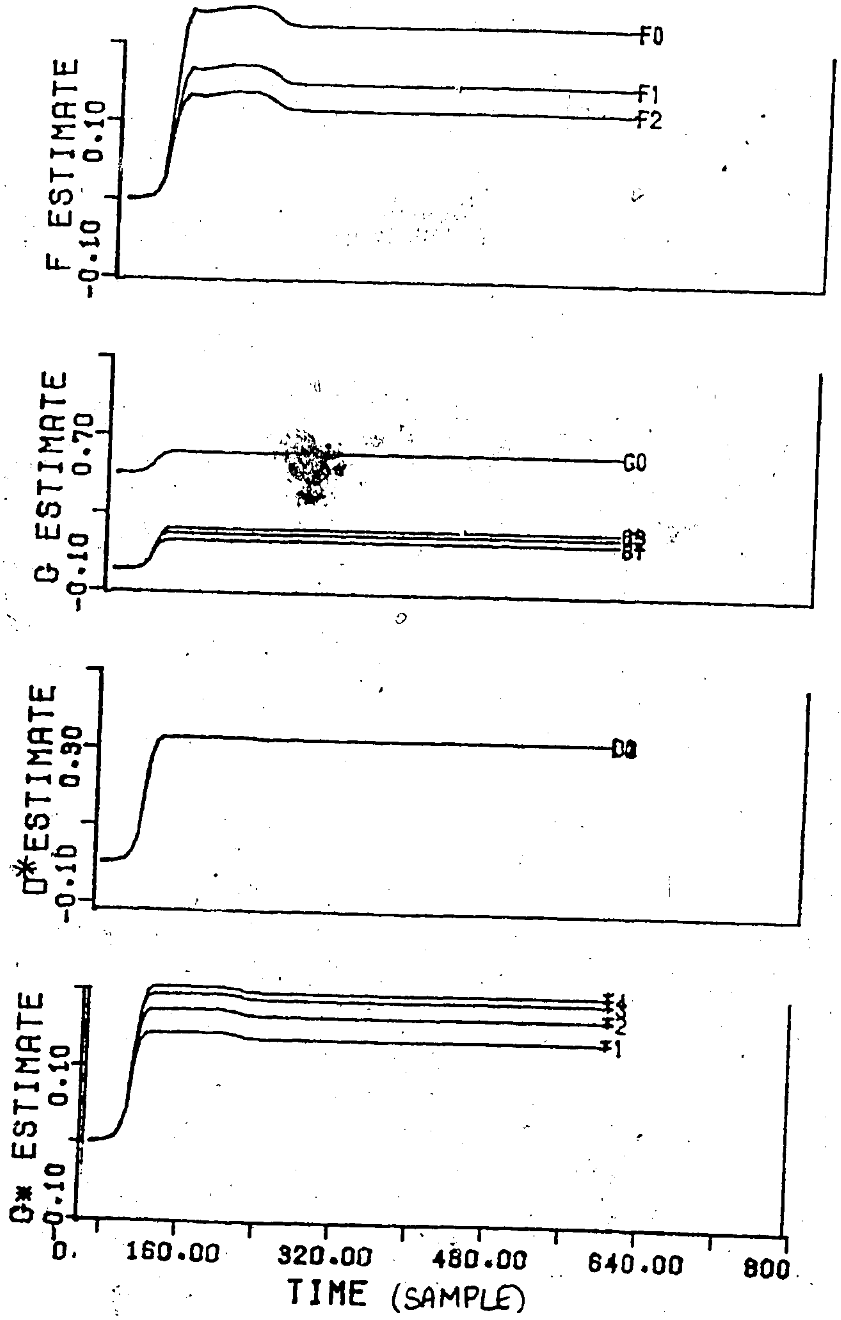


Figure 5.18 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a 20% Step Decrease in Feed Flow Rate using Recursive Learning Estimation

loop parameter estimates showed different convergence patterns for all except the G0 parameter estimate.

A 10% decrease in feed flow rate reduced the larger error observed in the bottom composition from the 20% feed decrease simulation with both the top and bottom loop parameter estimates exhibiting different adaption patterns than for the 20% feed decrease. The parameter estimate values are also smaller for the smaller feed flow change so it can be reasoned that the nonlinearities of the controlled system caused some of the parameter estimates to converge to different values for the various disturbance magnitudes.

5.6 Setpoint Tracking using RLS and RUD Estimators

Setpoint tracking was also tested with RUD and RLS estimation. The RSR method was not tried because results would be expected to be identical to those using the RUD estimation.

A square wave change in setpoint similar to that used in the study of the linear systems was employed. The top composition had setpoint limits of 96% and 98% MeOH while the bottom composition limits were 4.996% and 6% MeOH. The period of the wave was 100 minutes with each setpoint value desired for 50 minutes.

The top loop parameter estimates using the RLS estimator exhibited parameter blow-up by the 200th minute as shown in Figure 5.19, yet as can be seen from Figure 5.20, there is no parameter blow-up for the bottom loop parameter

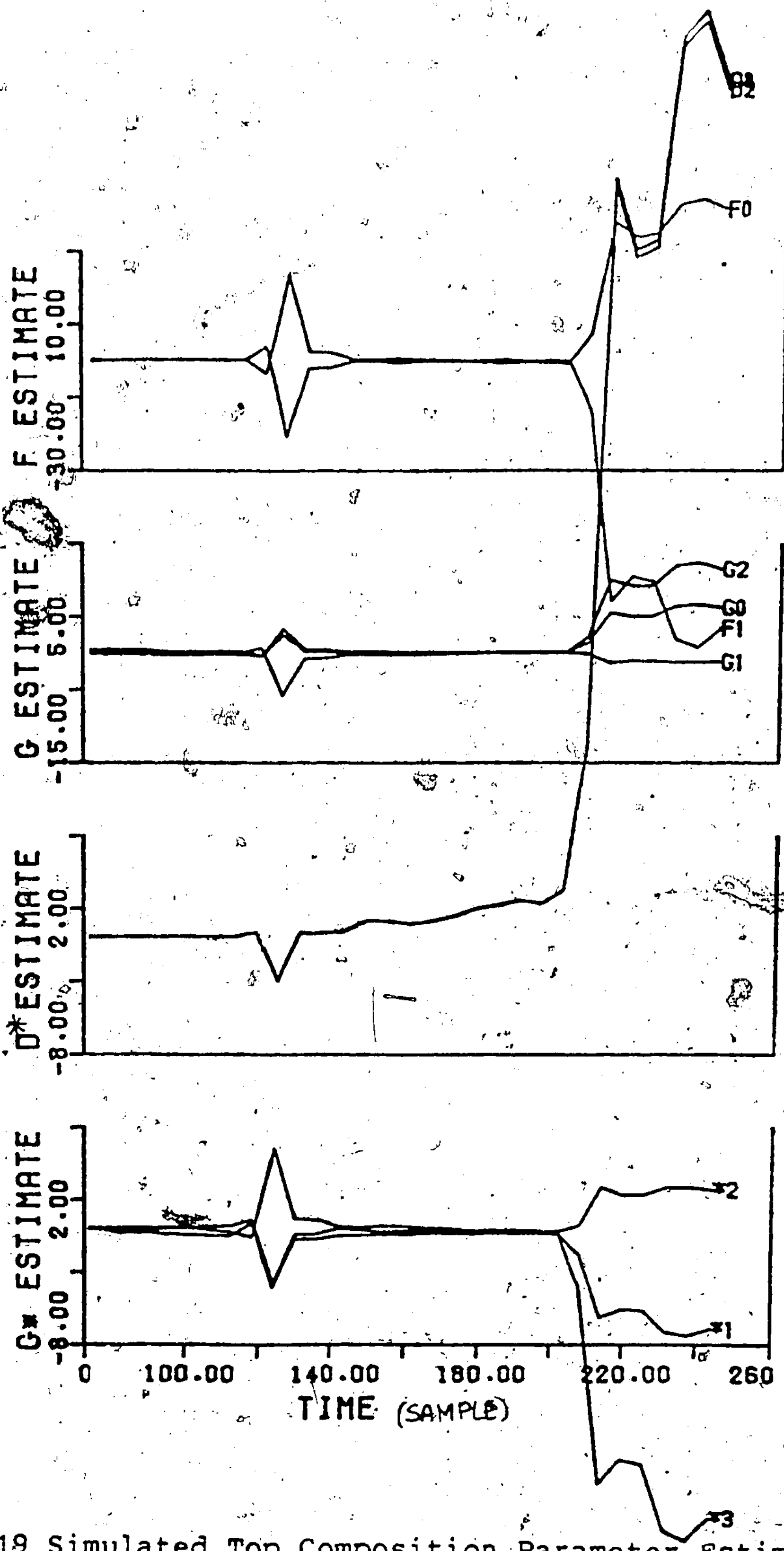


Figure 5.19 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Least Squares Estimation

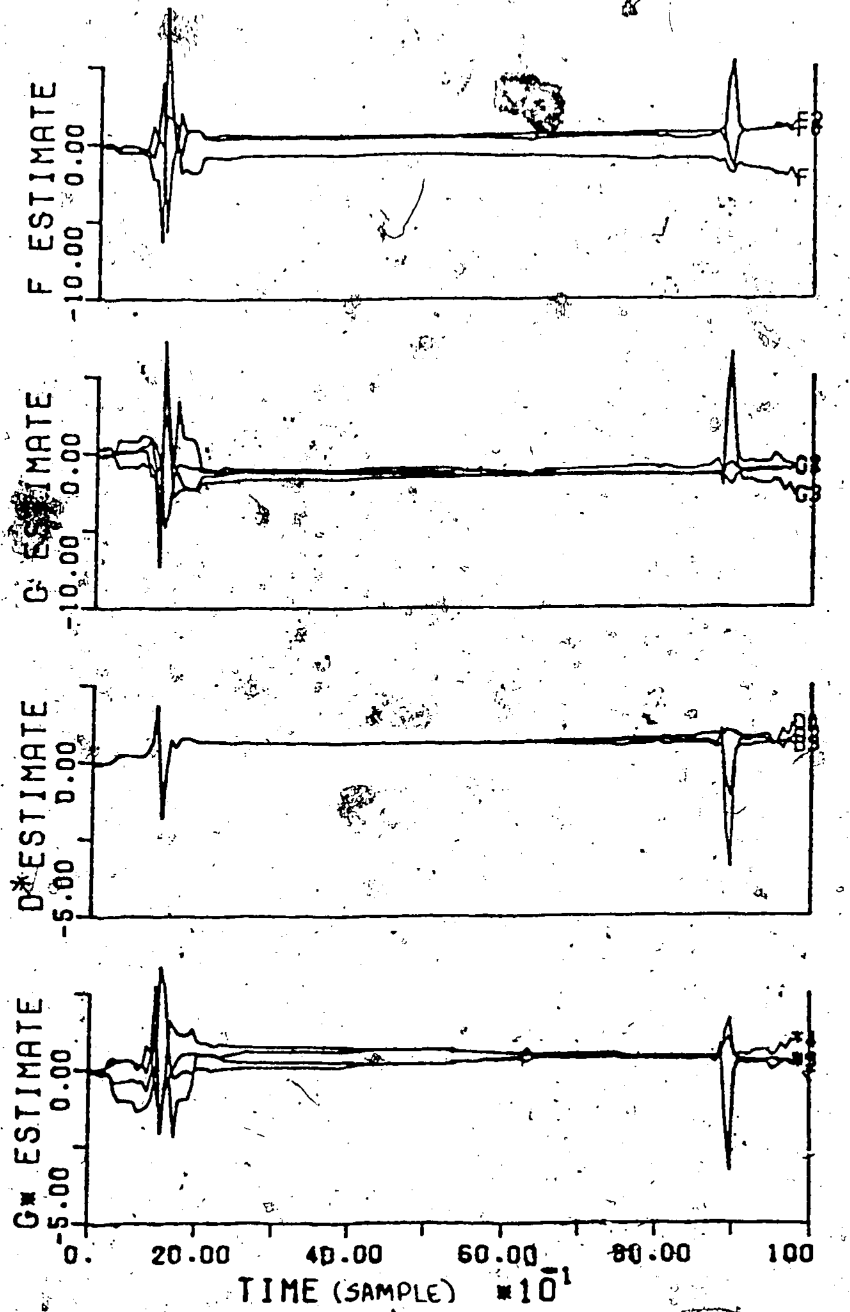


Figure 5.20 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Least Squares Estimation

estimates with only large fluctuations occurring at the time the top loop estimates blow-up. The reflux and steam flow rates fluctuated as shown in Figure 5.21 with the resulting compositions not following the desired setpoint in any manner.

Simulation using the RUD estimator for a square wave change in setpoint were also performed. It can be observed from the results in Figure 5.22 that initially the control performance is unsatisfactory however after 200 minutes the top composition follows the setpoint within acceptable limits of ±0.1% MeOH with the better control performance observed for a setpoint of 96% MeOH. The bottom composition exhibited the same trends as the top composition after 200 minutes however the deviations from the setpoint were greater.

The top loop parameter estimates displayed in Figure 5.23 show that the F, G and G* parameter estimates have converged after 400 minutes however the D* estimates continue to make incremental changes every 50 minutes.

The parameter estimates for the bottom loop exhibit large initial changes in the first 100 minutes as seen in Figure 5.24. The F and G estimates have converged after 400 minutes, however the D* estimates continue to adapt for the entire simulation. The G* estimates converged after 110 minutes of elapsed time.

These servo system simulations further confirm that use of the recursive upper diagonal factorization estimator

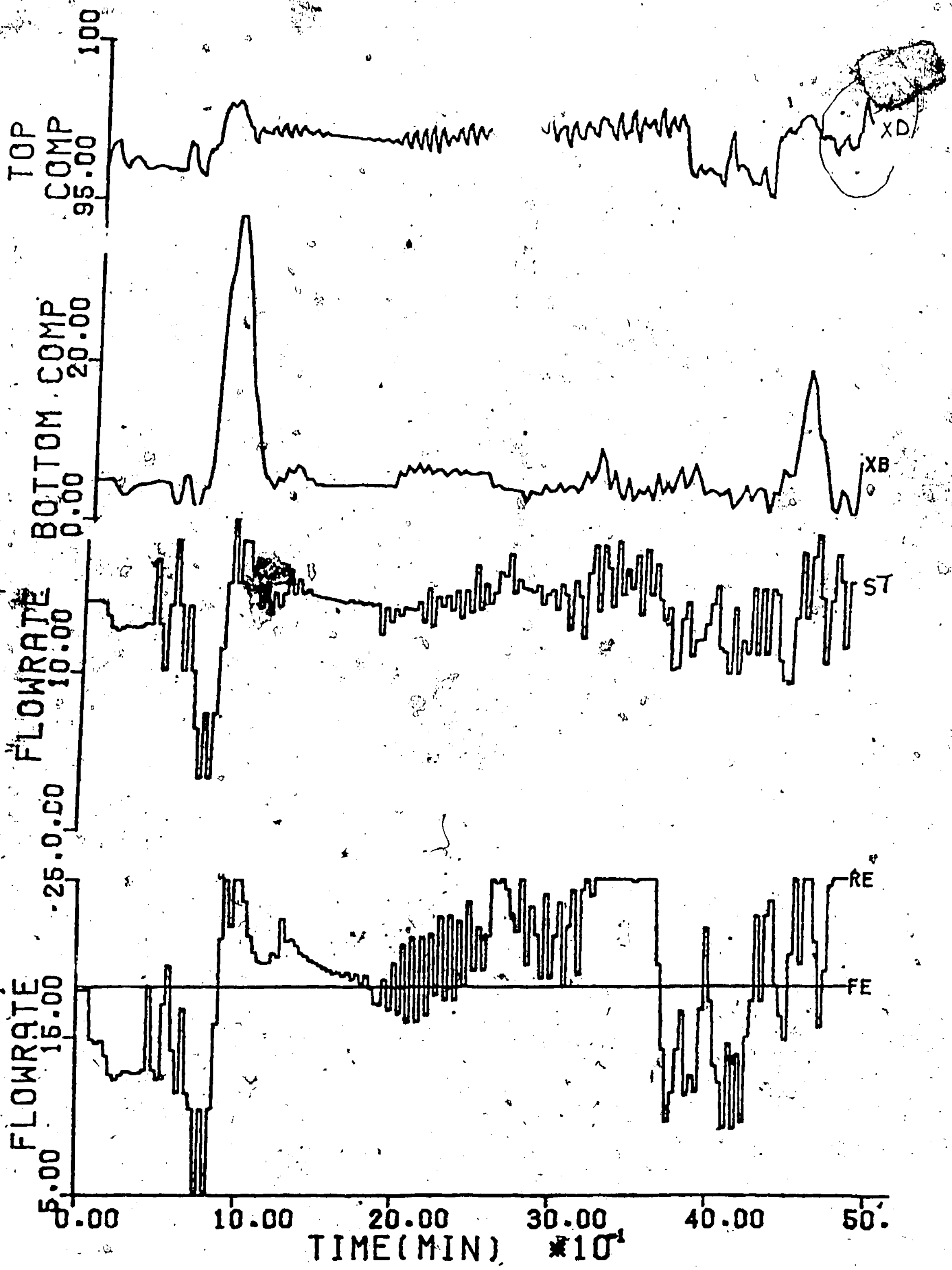


Figure 5.21 Simulated Response of the Binary Distillation Column to a Square Wave Change in Setpoint for Both Loops using Recursive Upper Diagonal Factorization Estimation

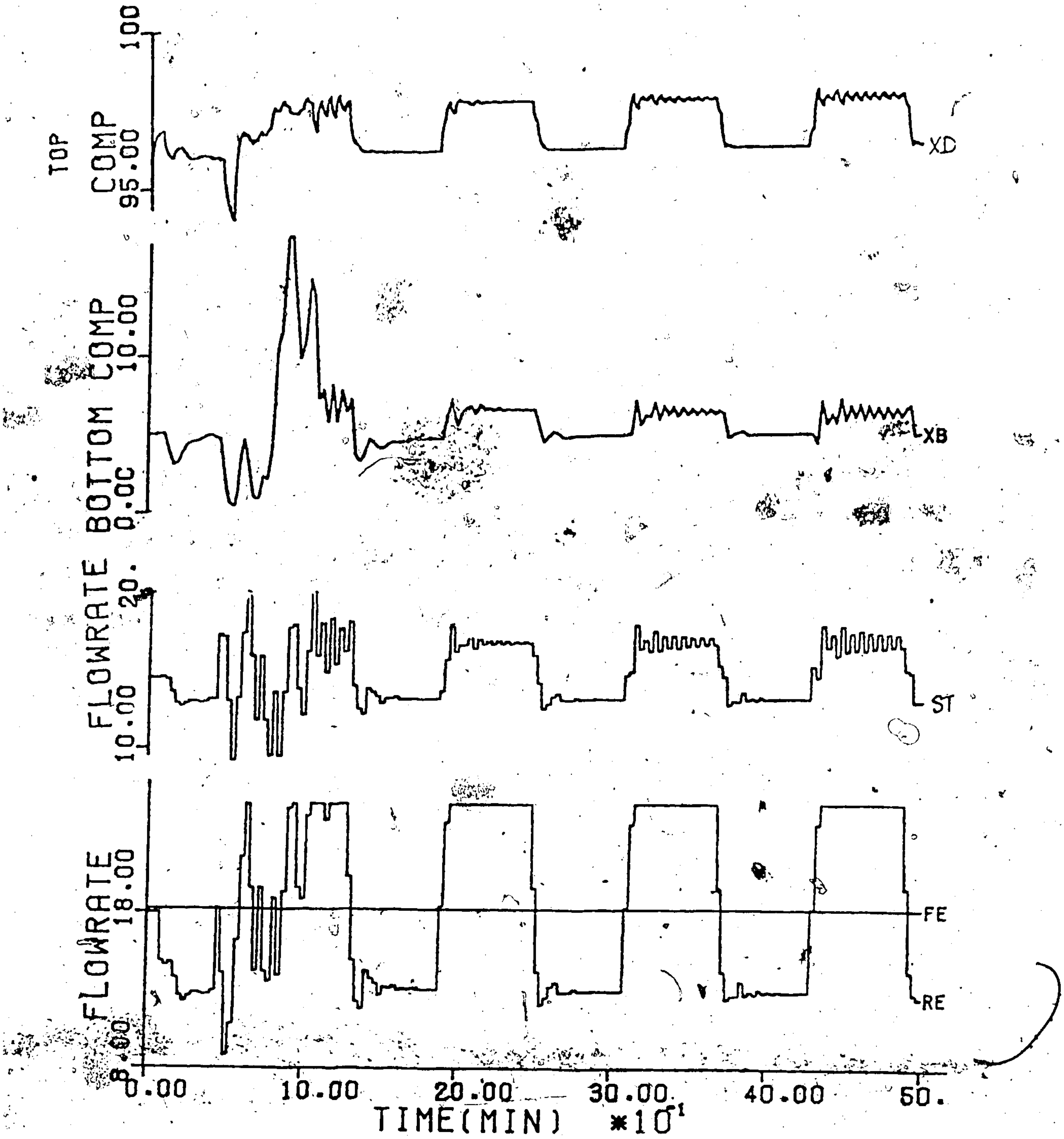


Figure 5.22 Simulated Top Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Upper Diagonal Factorization Estimation

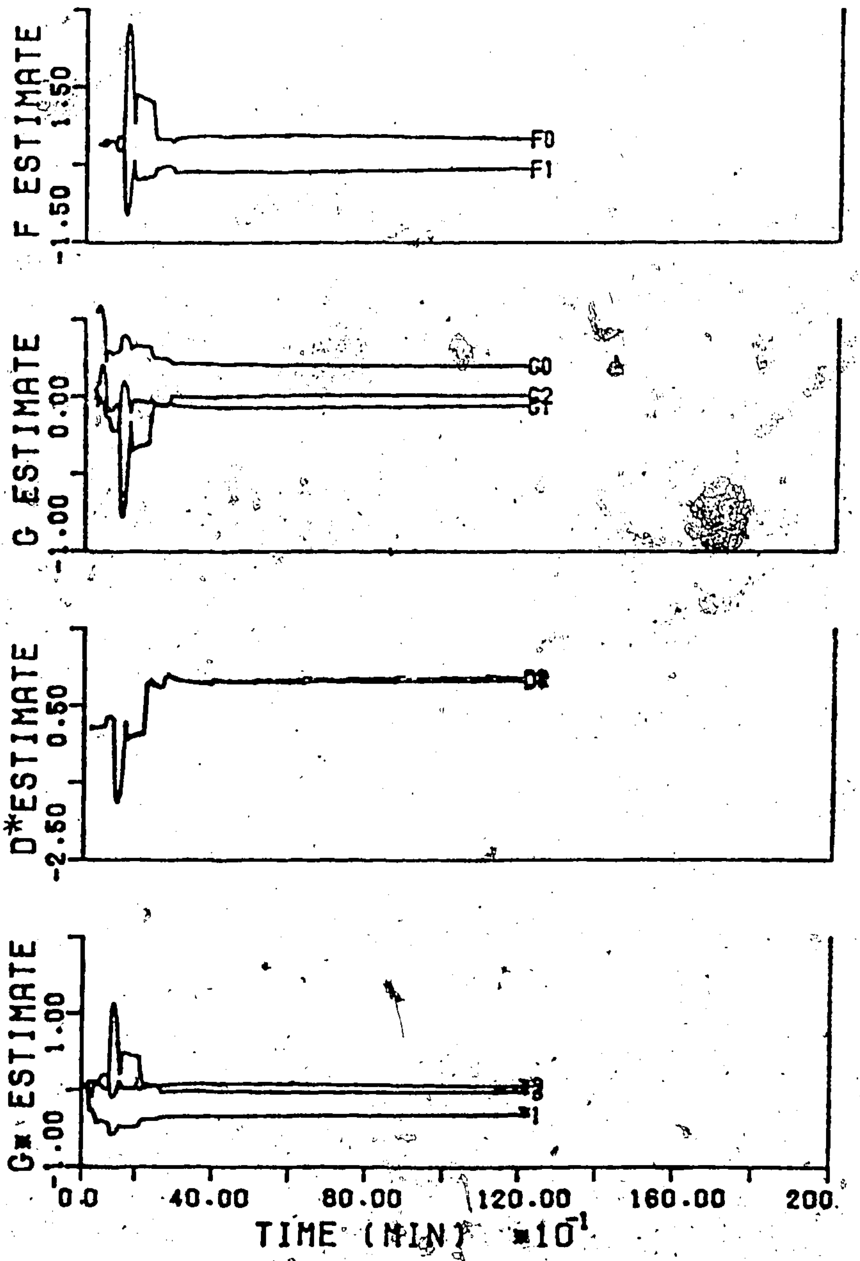


Figure 5.23 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Upper Diagonal Factorization Estimation

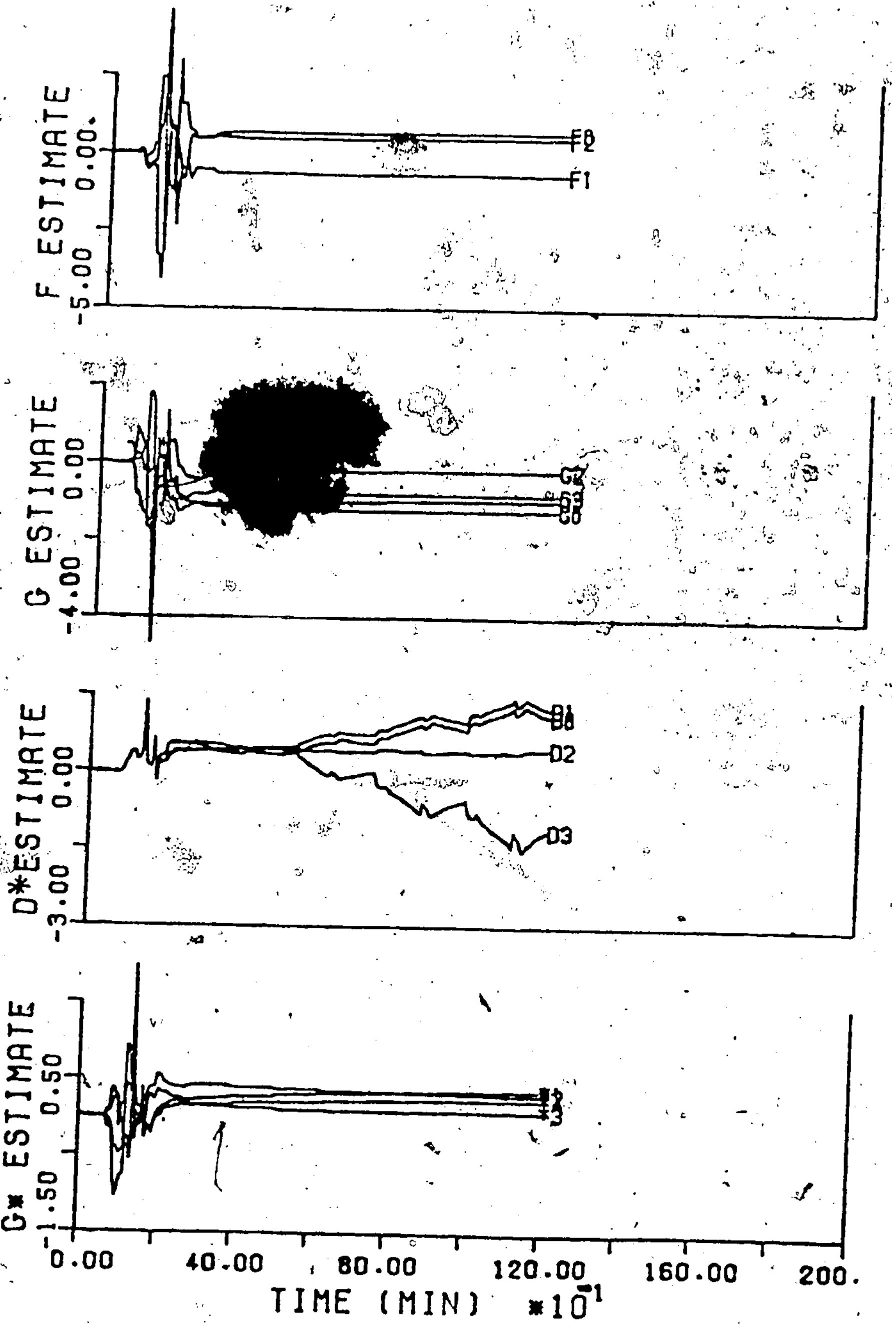


Figure 5.24 Simulated Bottom Composition Parameter Estimates of the Binary Distillation Column for a Square Wave Change in Setpoint using Recursive Upper Diagonal Factorization Estimation.

provides the best control performance and is the most efficient estimation method of all the identification techniques tested.

6. Experimental Results

Evaluation of the parameter identification techniques also involved experimental testing using the pilot plant distillation column. Self-tuning control was applied to control both composition loops using the four estimation methods tested with the simulations performed with the column model, the recursive least squares, recursive square root, recursive upper diagonal factorization and recursive learning estimators.

Separation of a methanol and water feed by the pilot plant distillation column is interfaced to an LSI 11/03 based microcomputer system. The glass-walled column has a diameter of 22.86 cm and contains eight trays, spaced 30.48 cm apart, fitted with four bubble caps per tray. A total condenser and a thermosiphon reboiler complete the column. Levels in both the condenser and reboiler are maintained by adjusting the top and bottom product flow rates, respectively. Reflux and steam flow rates are manipulated to adjust the top and bottom compositions, respectively, to attain the desired setpoints. Changes in feed flow rate were the load disturbances introduced to the column.

Tower pressure was regulated using the cooling water flow rate. The inlet feed and reflux temperatures were also controlled. Top product composition is measured continually by an on-line capacitance probe and the bottom product composition measure by an on-line HP5702A gas chromatograph equipped for liquid sampling. The sampling time for the top composition was arbitrarily selected to be 60 seconds while the bottom product was sampled every 180 seconds. A schematic diagram of the pilot plant distillation column is given as Figure 6.1.

The same experimental conditions were maintained for all four estimation methods with a "test" lasting for a total of 600 minutes with the feed flow rate changed every 150 minutes. The first step change was a 25% increase from the steady state value of 18.06 g/s; the second change in the same magnitude step decrease to the steady state value. A 25% decrease in steady state feed flow rate is the third step with the fourth and final step change the return of the feed to the steady state value of 18.06 g/s.

Setpoint of the top composition was 95.0% mass percent MeOH and the bottom composition setpoint 5% MeOH.

From the experience of previous tests [39] the orders of the controller polynomials were selected to be as follows: For the top composition loop there were five G parameters, five G* parameters and three F parameters with the F parameters associated with the outputs, top composition, the G parameters with the input, reflux flow rate, and the G* parameters with the loop interaction, steam flow rate. The bottom composition self-tuning controller employed three F parameters (bottom composition), five G parameters (steam flow rate) and six G* parameters (reflux flow rate). Unlike the simulations performed with the column model the feed flow rate disturbance was not measured and

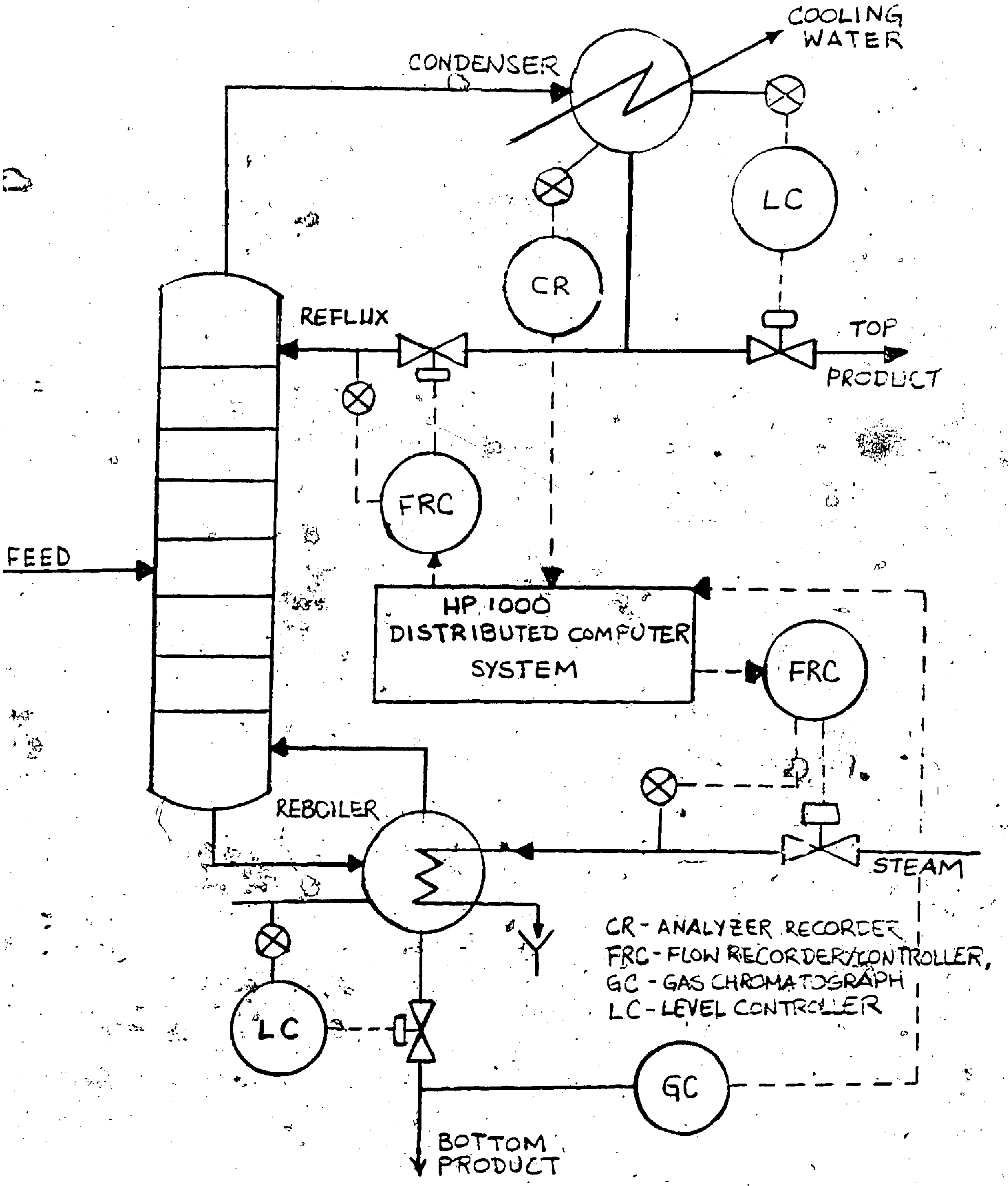


Figure 6.1 Schematic Diagram of the Pilot Plant Distillation Column

therefore there were no D* parameters.

For all experiments except the test using the recursive upper diagonal factorization estimator the column was operated under proportional-integral-derivative (PID) control for 200 minutes and the estimation algorithm allowed to identify parameters before the self-tuning control was turned "on" and the PID controller turned "off". This was done to provide the self-tuning controller with initial parameter estimates other than zero so that the column operation did not become unstable.

As with the results from the column model simulations the results of the experiments are illustrated with plots of the feed, reflux and steam flow rates and both top and bottom compositions. The parameter estimates are also plotted for both loops.

6.1 Recursive Least Squares Estimation Evaluation

Results for the 25% step increase in feed flow rate and the subsequent return to the steady state value using the recursive least squares estimator are presented in Figures 6.2, 6.3, 6.4 and 6.5. The increase in feed flow rate caused deviation in the top composition up to +0.4% which was followed by the output cycling about the setpoint (±0.2%) that continued until the step decrease in flow rate to the steady state value was introduced. The initial error may have been caused by the incorrect parameter estimates obtained during PID control and preliminary identification.

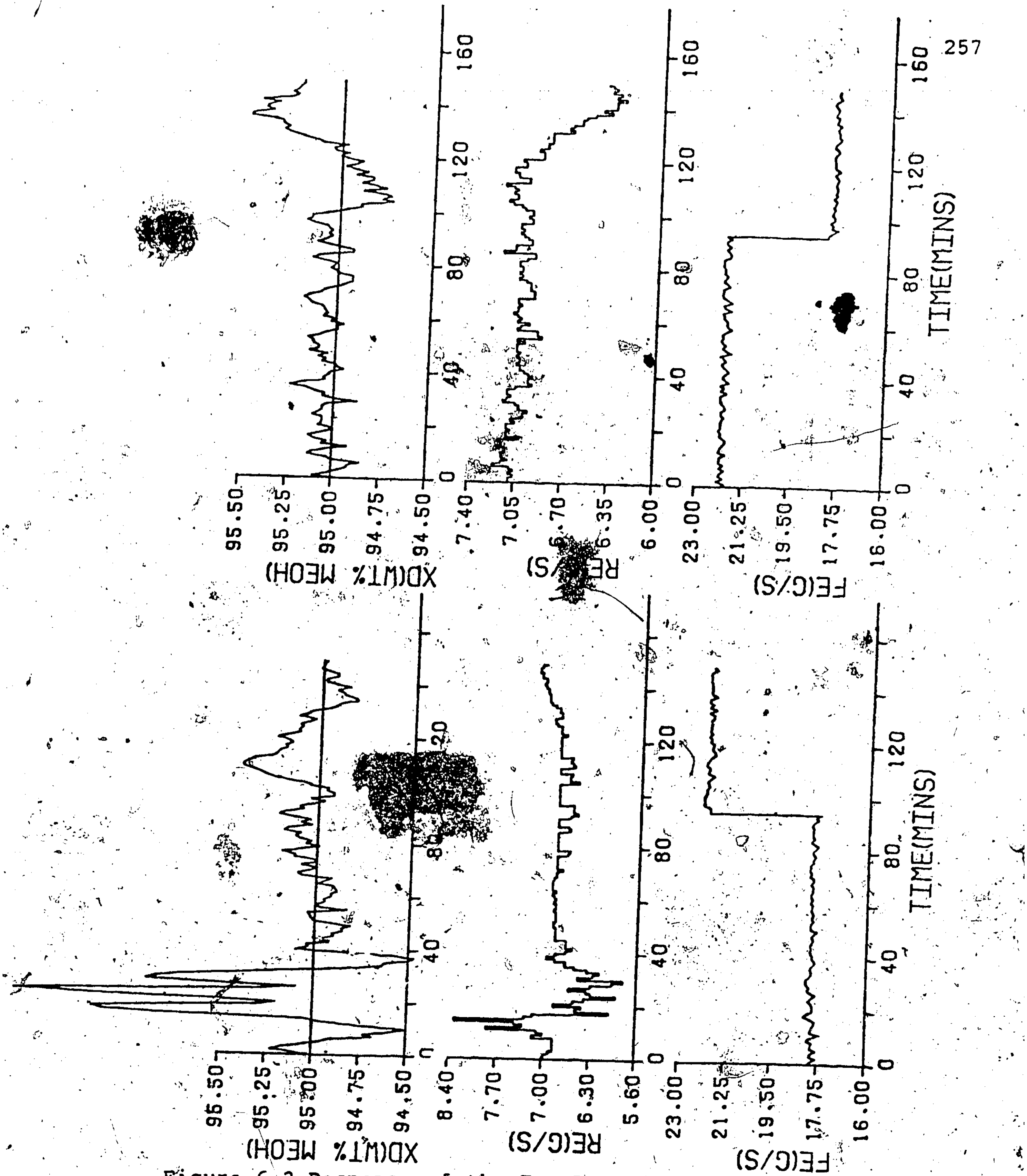


Figure 6.2 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator

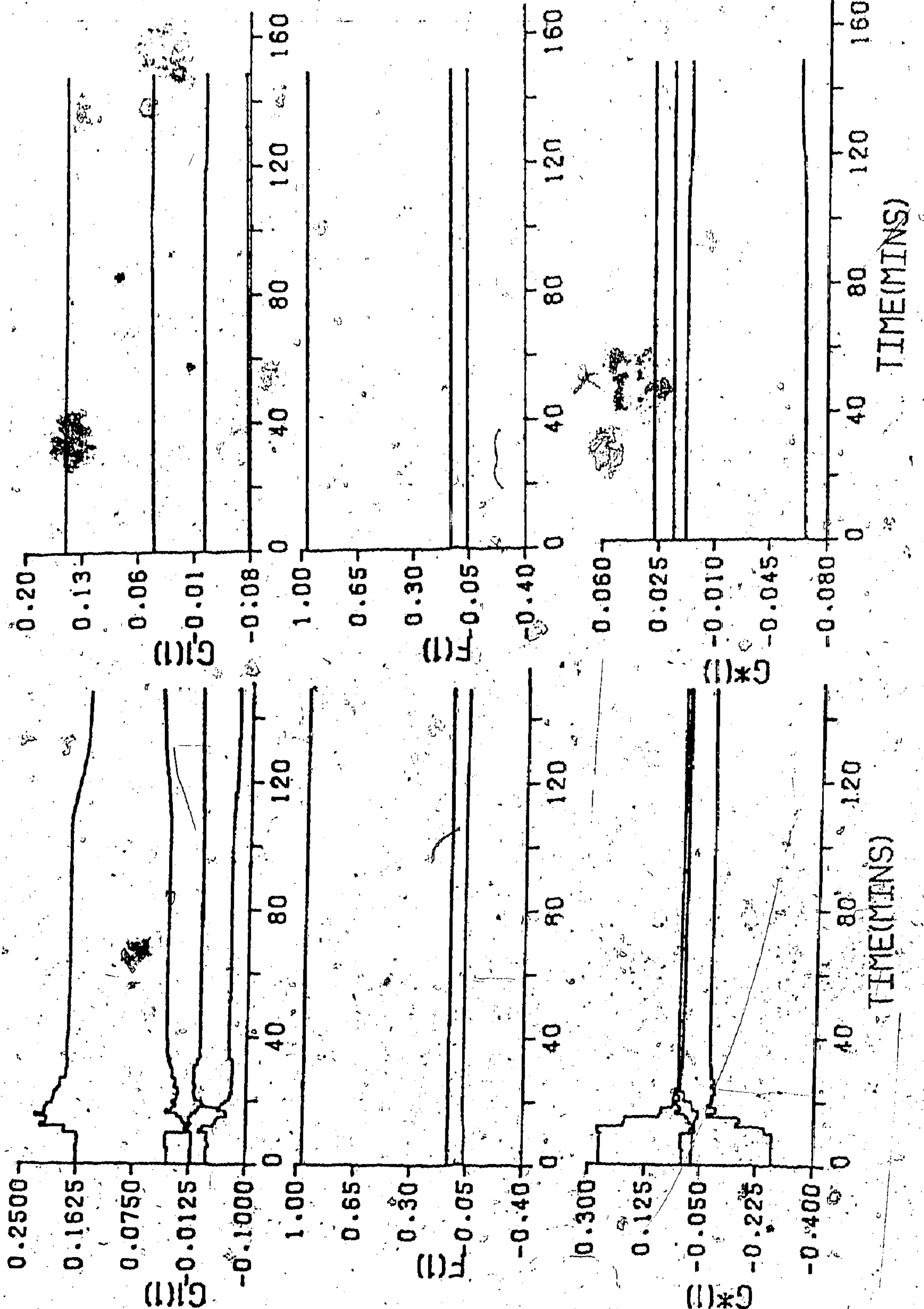


Figure 6.3 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator

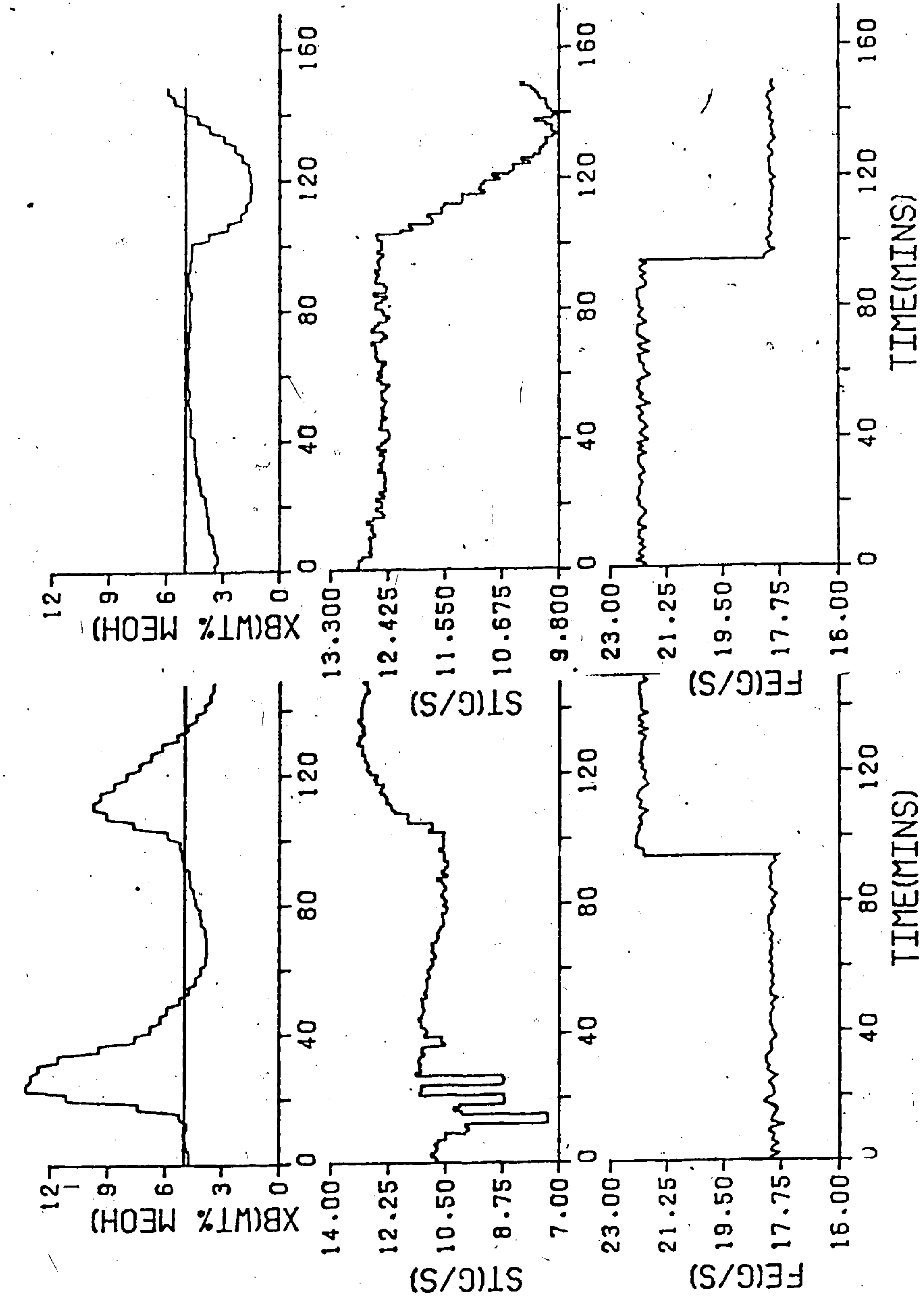


Figure 6.4 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator

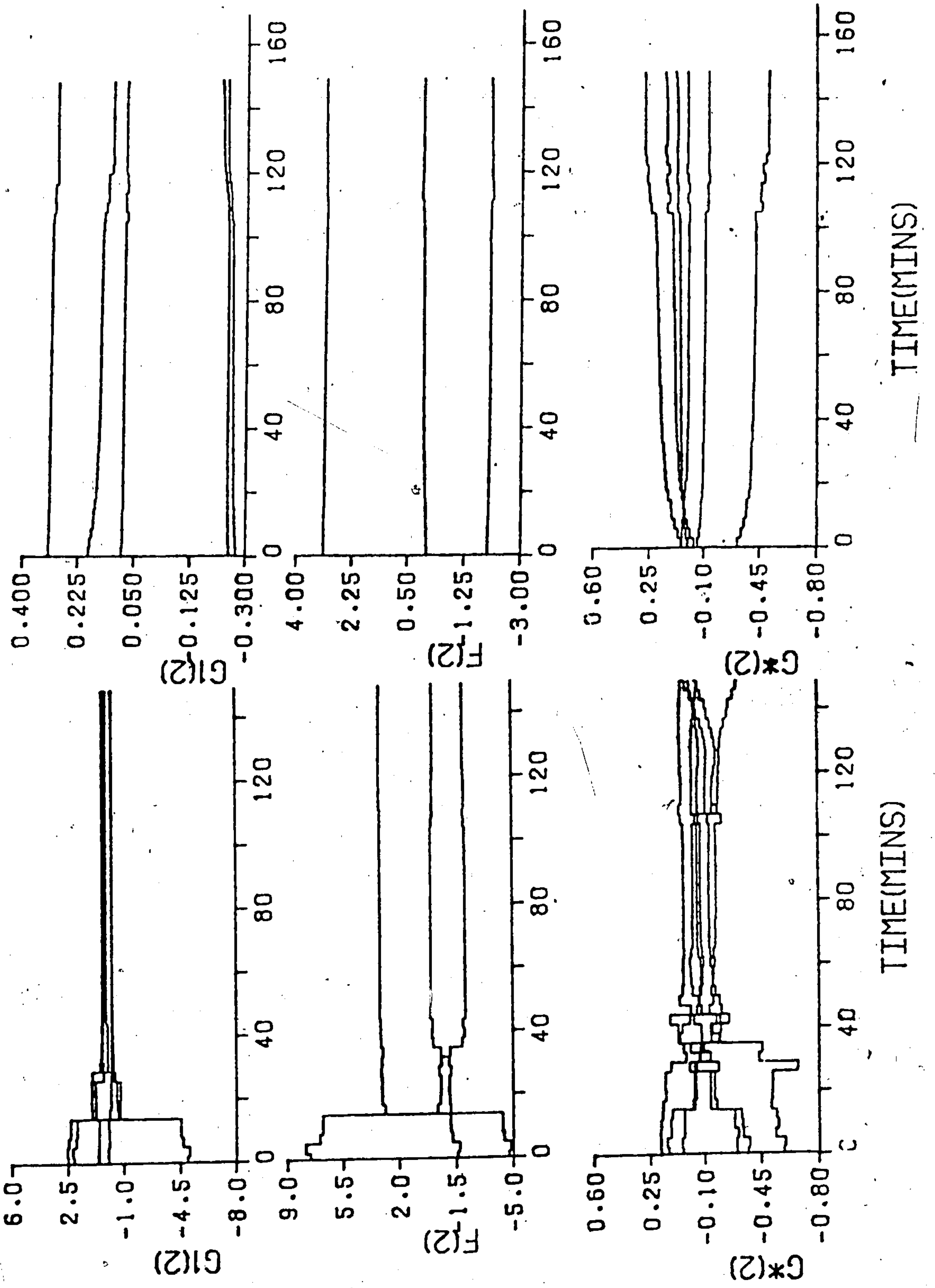


Figure 6.5 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Least Squares Estimator

This can be concluded by noting that the G* and G parameter estimates in Figure 6.3 begin to adapt once the self-tuner is turned "on". When the feed flow rate decreased the top composition decreased from the setpoint to 94.75% MeOH and then increased to 95.5% MeOH before returning to the setpoint 70 minutes after the step change. (This final part of the response can be seen in Figure 6.6).

The reflux flow rate only showed large fluctuations at the start of the experiments as the flow changes at the first [illegible] feed step changes were gradual.

G and G* top loop parameter estimates showed initial adaption when the self-tuning controller was implemented. Convergence of the G* estimates occurred in 35 minutes while the G parameter estimates began to adapt again when the step change down to the steady state feed value was introduced. The F parameter estimates showed no changes from the initial values obtained using PID control and recursive least squares identification. There were no significant changes in any of the estimate values for the step down to the steady state feed value.

The bottom composition showed a deviation of 9% when the self-tuning controller was first applied as can be seen from the results in Figure 6.4. When the feed flow rate was increased by 25% there was a 4% positive deviation before the bottom composition went below the setpoint and only approached the desired setpoint after 195 minutes of operation. As can be seen, the bottom composition never

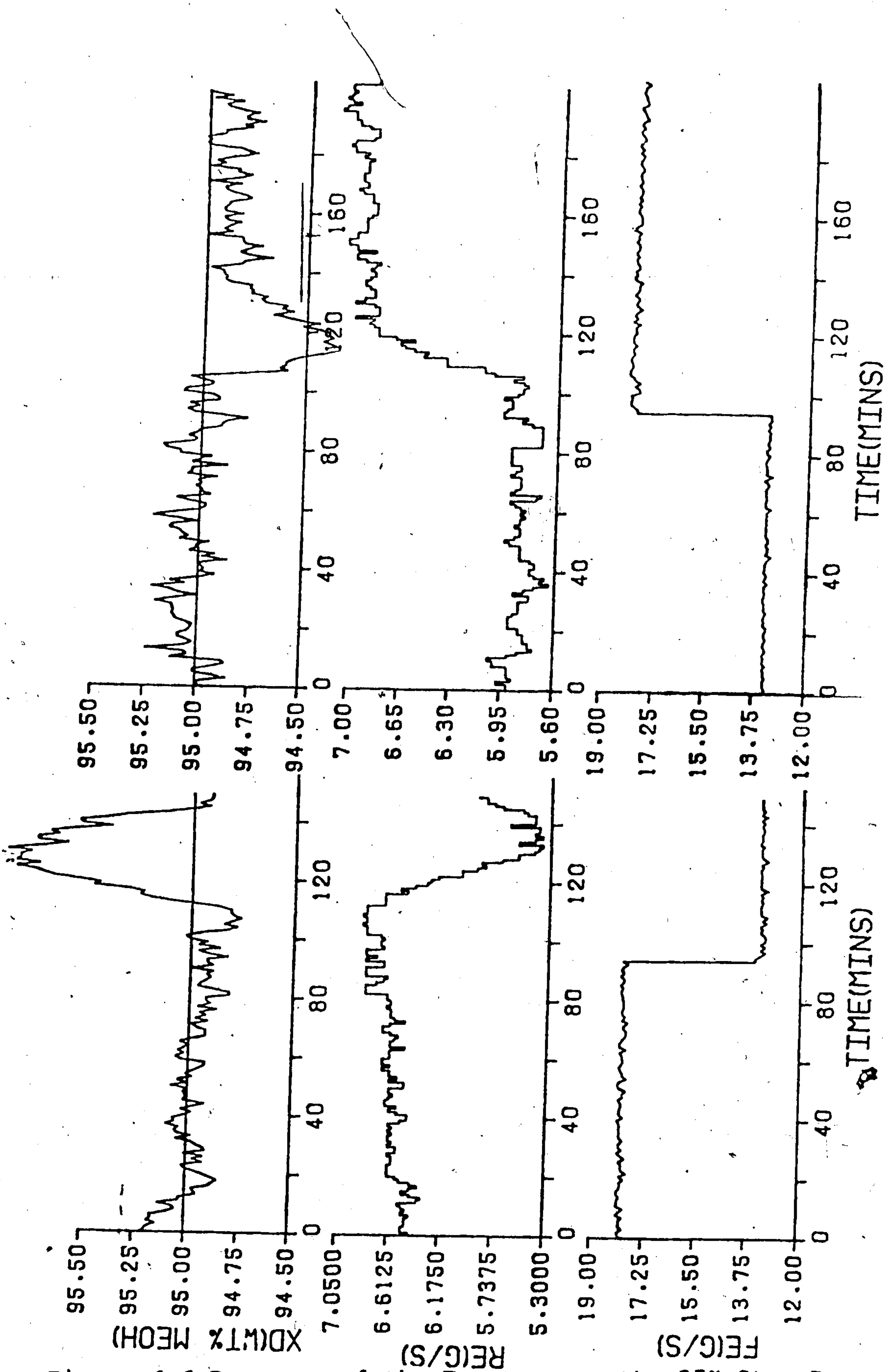


Figure 6.6 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator

reached the setpoint, exhibiting and offset of about 0.2%.

The decrease in feed rate to the original steady state value caused a deviation of -3.5% in bottom composition as can be seen in Figure 6.4, but the bottom composition returned to the desired setpoint in 70 minutes and control performance was satisfactory.

Initially the steam flow rate exhibited large fluctuations but after 80 minutes changes were gradual except when the step changes in feed flow were introduced.

A comparison of the results in Figure 6.3 with those in Figure 6.5 reveals that the bottom loop parameter estimates had larger initial values than the top loop estimates. The F and G parameter estimates converged after 50 minutes however the G* estimates continued to adapt even after the first step change in feed rate. The F and G parameter estimates showed minor changes only after the second step change while the G* estimates continued to adapt.

The remaining 300 minutes of the experiment involved a 25% step decrease in feed flow rate from its steady state value and then return of the rate to the steady state value after 150 minutes. As can be seen in Figure 6.6 the reduction in feed flow rate resulted in a slight decrease in the top composition which was followed by a large overshoot of 0.9%. The composition returned to the desired setpoint in 40 minutes and as with the top composition behavior for the 25% increase in feed flow rate, cycled around the setpoint until the next feed disturbance.

When the feed flow rate was increased to the steady state value the top composition deviated from the setpoint by 0.8%. The top composition continued to remain below the desired setpoint indicating that offset was present.

The top loop G and G* parameter estimates showed gradual changes in value for the 25% decrease in feed flow rate shown in Figure 6.7. Further adaption was observed when the feed flow rate returned to the steady state value but both step changes did not cause significant parameter estimate value changes.

Comparing Figure 6.4 with 6.8 it can be seen that the step change down from the steady state feed flow rate caused the same -3.5% deviation in bottom composition as the step change down when the flow was returned to its steady state value. The bottom composition returned to the desired setpoint in 60 minutes. The step change increasing the feed flow rate to the steady state value resulted in a deviation of 0.6% however, the composition reached the setpoint after 90 minutes. The last step change produced large changes, of 1 to 1.5 g/s in the steam flow rate, not observed for any of the other feed changes.

The bottom loop parameter estimates responded to the decrease in the steady state feed flow rate value by adapting to new values as illustrated by the results in Figure 6.9 with the step change causing only minimal changes in the parameter estimates.

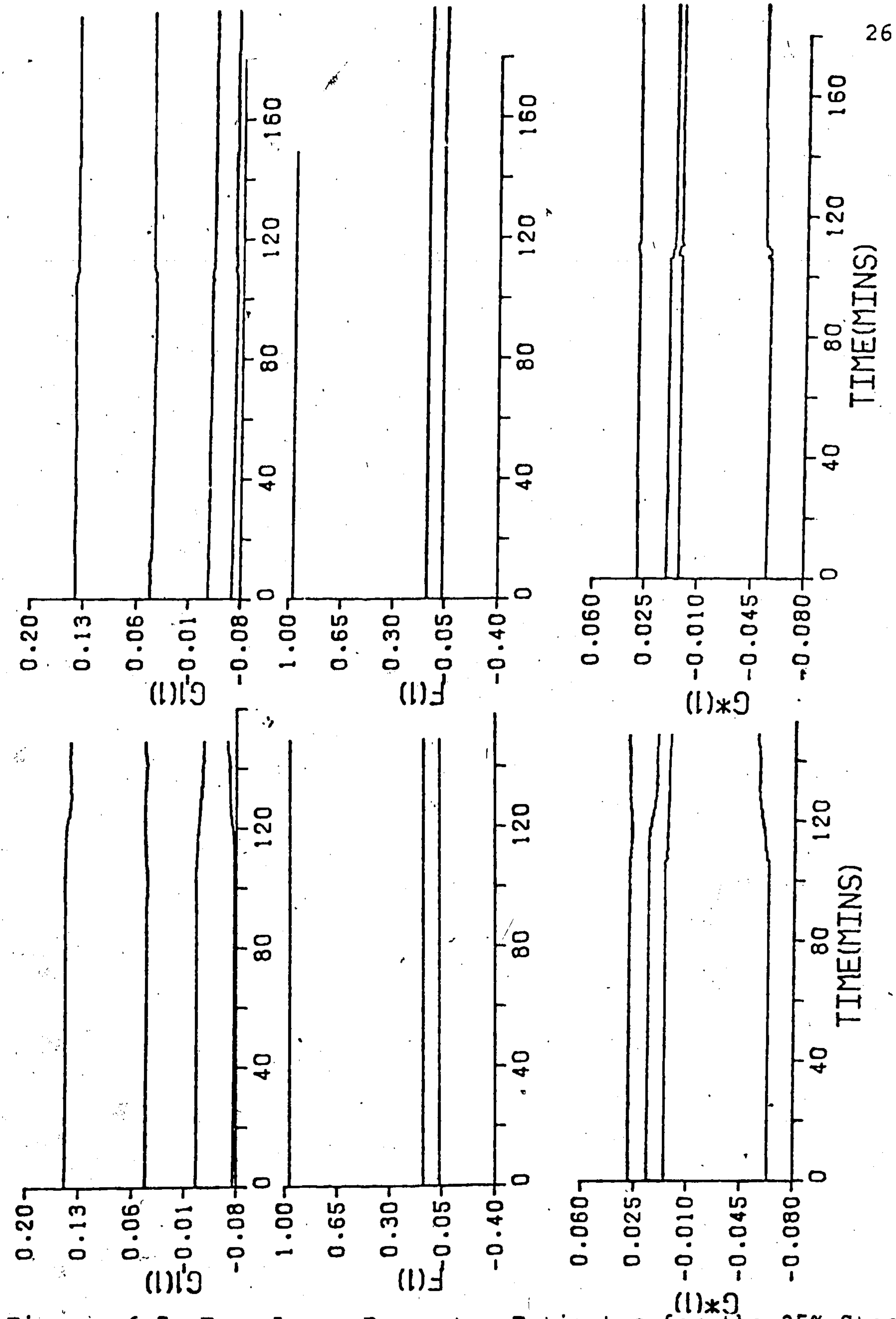


Figure 6.7 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator

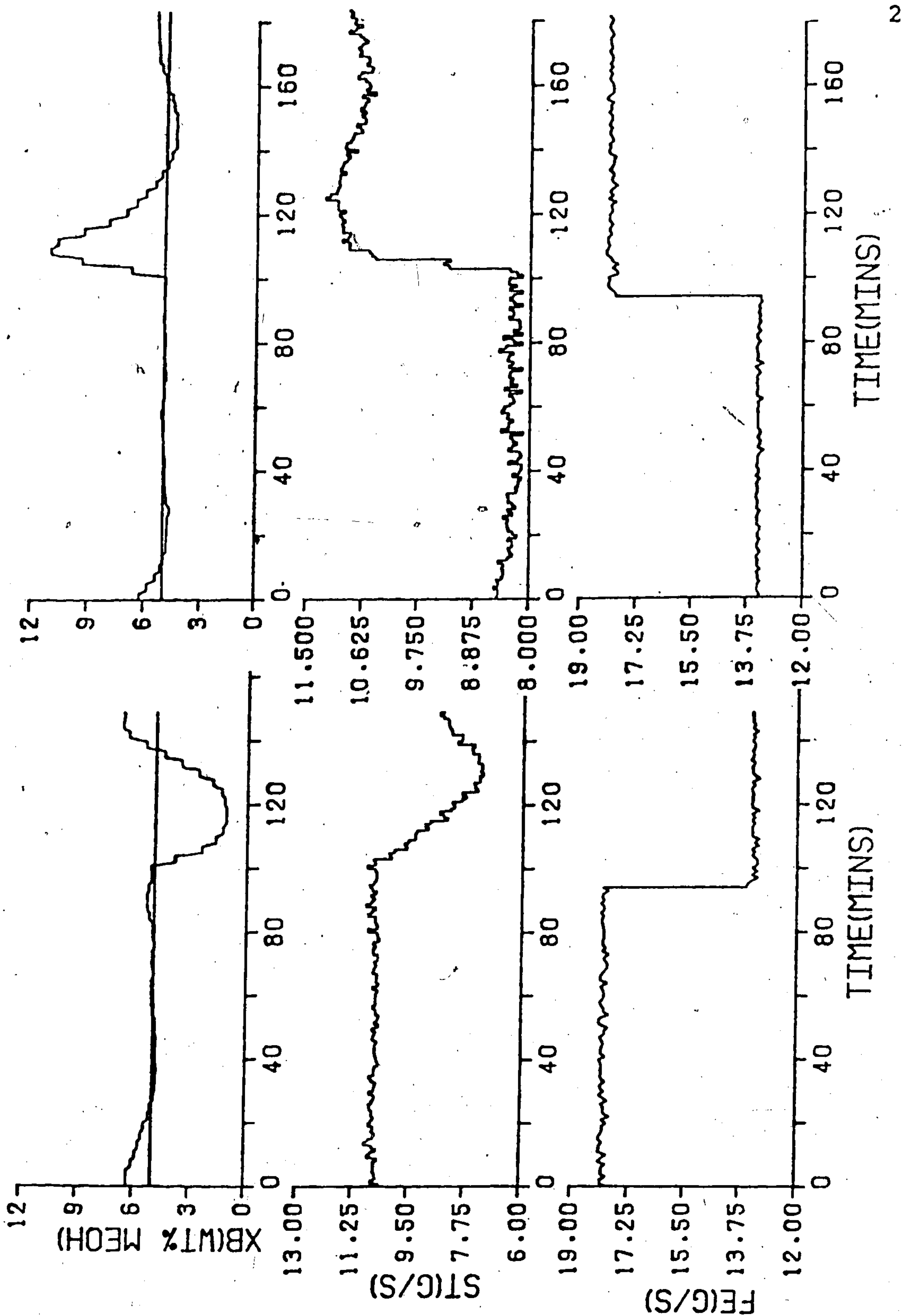


Figure 6.8 Response of the Bottom Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator

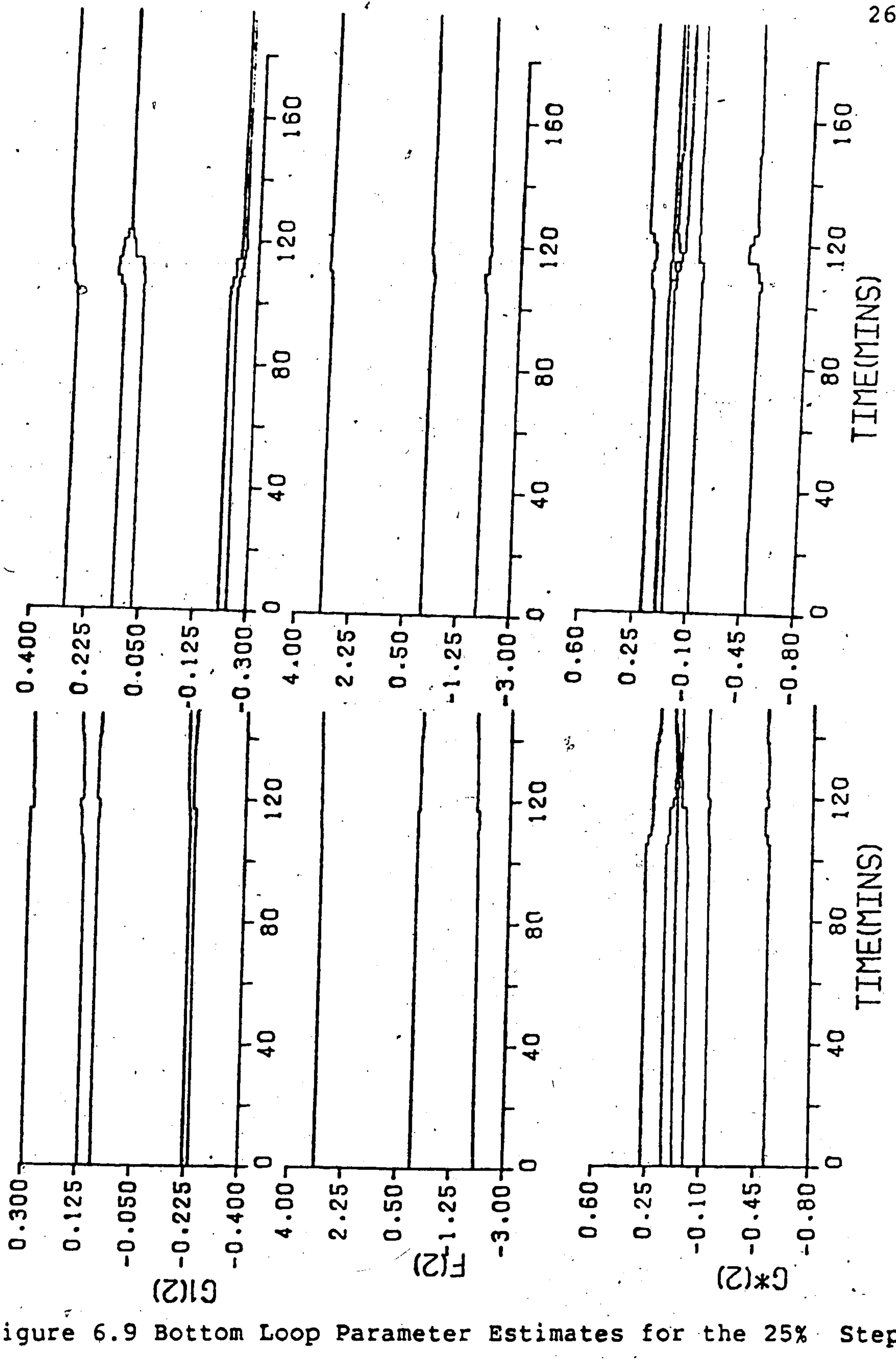


Figure 6.9 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Least Squares Estimator

Examining the results obtained for the feed disturbances using the recursive least squares estimation the overall conclusion is that this estimation method does not give particularly satisfactory control performance. There were large deviations from the desired setpoints for both top and bottom loops and these effects continued for more than 60 minutes. Cycling about the setpoint was also observed. The initial behavior of the column was not satisfactory even after 200 minutes of PID control while the estimator was identifying nor when the transfer between PID and self-tuning control was executed.

6.2 Recursive Square Root Estimation Evaluation

Testing of the recursive square root method was implemented in the same manner as the RLS estimator with 200 minutes of PID control while the identification was being performed to avoid unstable operation.

From the results in Figure 6.10 it can be seen that the initial step increase in feed flow rate caused a positive 0.6% deviation followed by a negative 0.4% drop in top composition below the setpoint. However, the composition did return to the desired setpoint in about 80 minutes but the top composition continued to cycle about the setpoint (±0.15%). Return of the feed flow rate to the steady state value resulted in an initial 0.3% increase in top composition followed by a 0.4% decrease and then a 0.5% increase. The top composition returned to the setpoint in 60

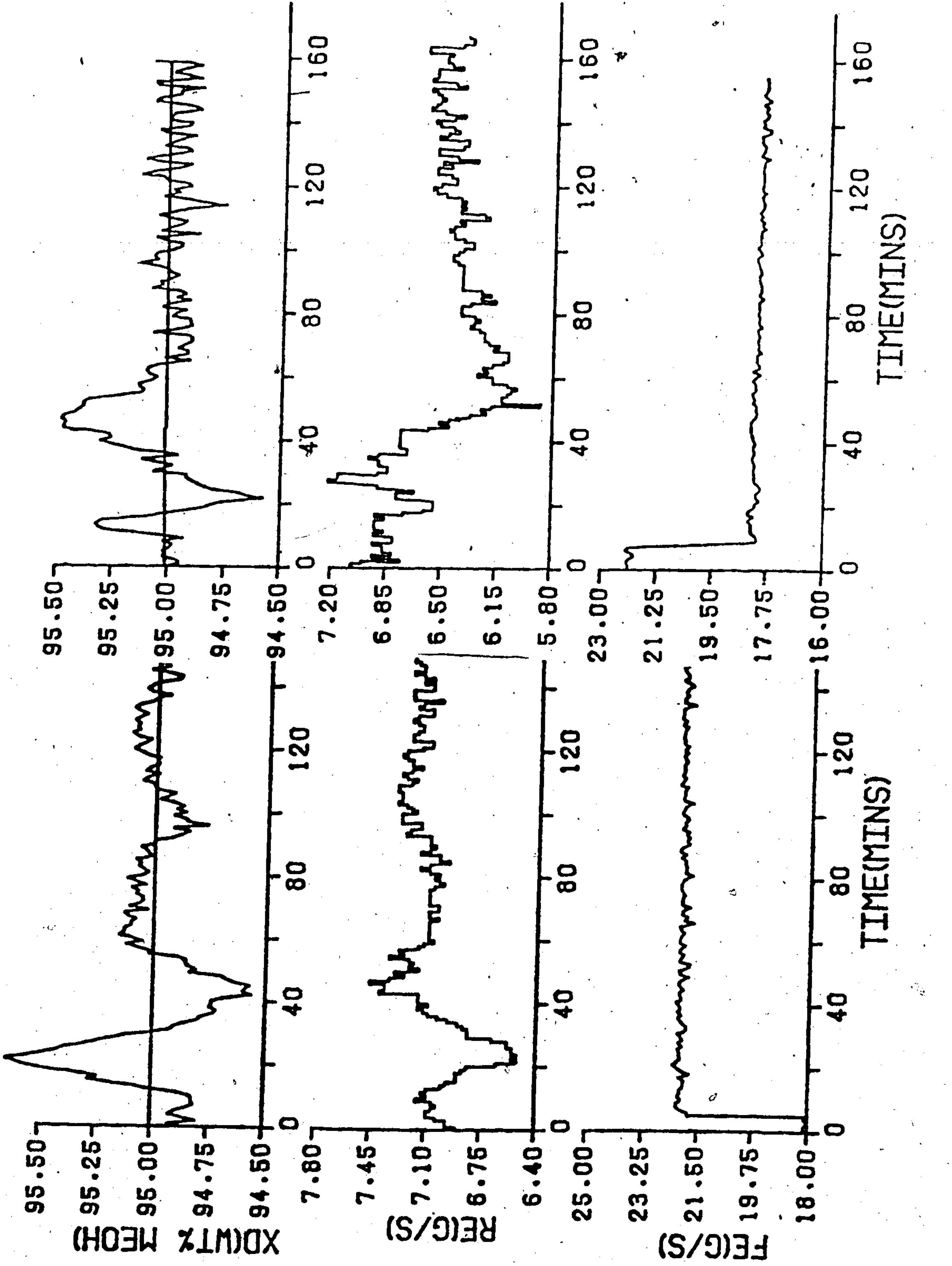


Figure 6.10 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator

minutes and fluctuated about the desired setpoint within acceptable limits.

The top loop parameter estimate behavior is shown in Figure 6.11. As can be seen, no perceptible adaption had occurred for the F parameter estimates, the F0 estimate remained near 1.0 while the balance of the F estimates were at 0.0. The G and G* parameter estimates exhibited changes at the initial step change and became constant by the 100th minute. However, when the step change decrease in flow occurred, the G and G* parameter estimates continue to adapt until the next step change was made.

Comparing Figure 6.10 with Figure 6.12 it can be seen that the bottom composition exhibited the same initial behavior pattern as the top composition to the same step change with a 4% departure from the setpoint following the step increase in feed flow rate followed by a -1.5% deviation from the desired setpoint. The composition returned to the desired setpoint in 90 minutes. Return of the feed flow rate to the steady state value caused only minor deviations in bottom composition. Initially the composition decreased to 3% below the desired setpoint but was within 0.5% of the setpoint after 60 minutes.

The bottom loop parameter estimates, presented in Figure 6.13 exhibited very gradual changes at the time of the +25% step change in flow rate. All parameter estimates converged after 90 minutes to values similar to those during the initial identification phase with the column operating

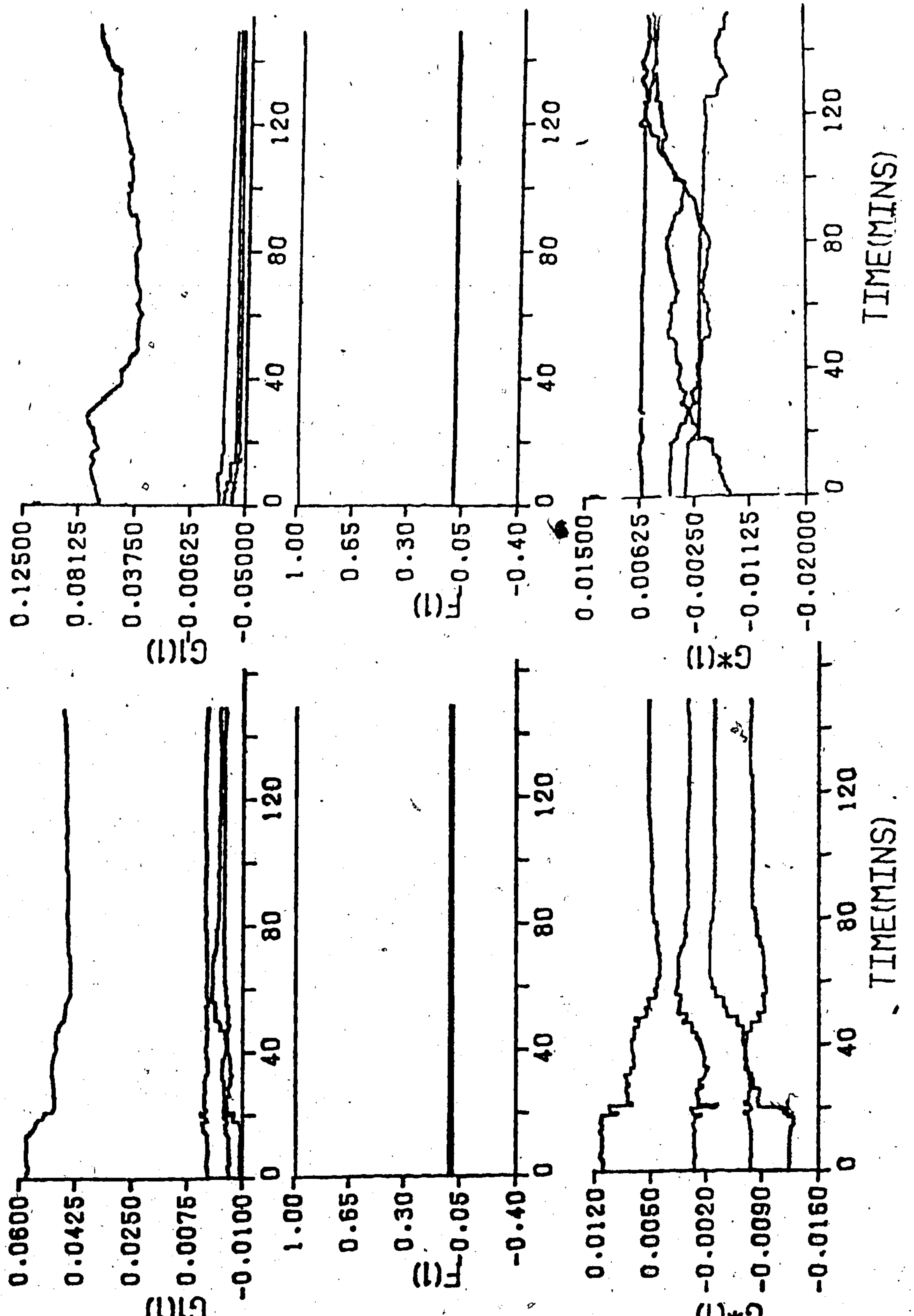


Figure 6.11 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator

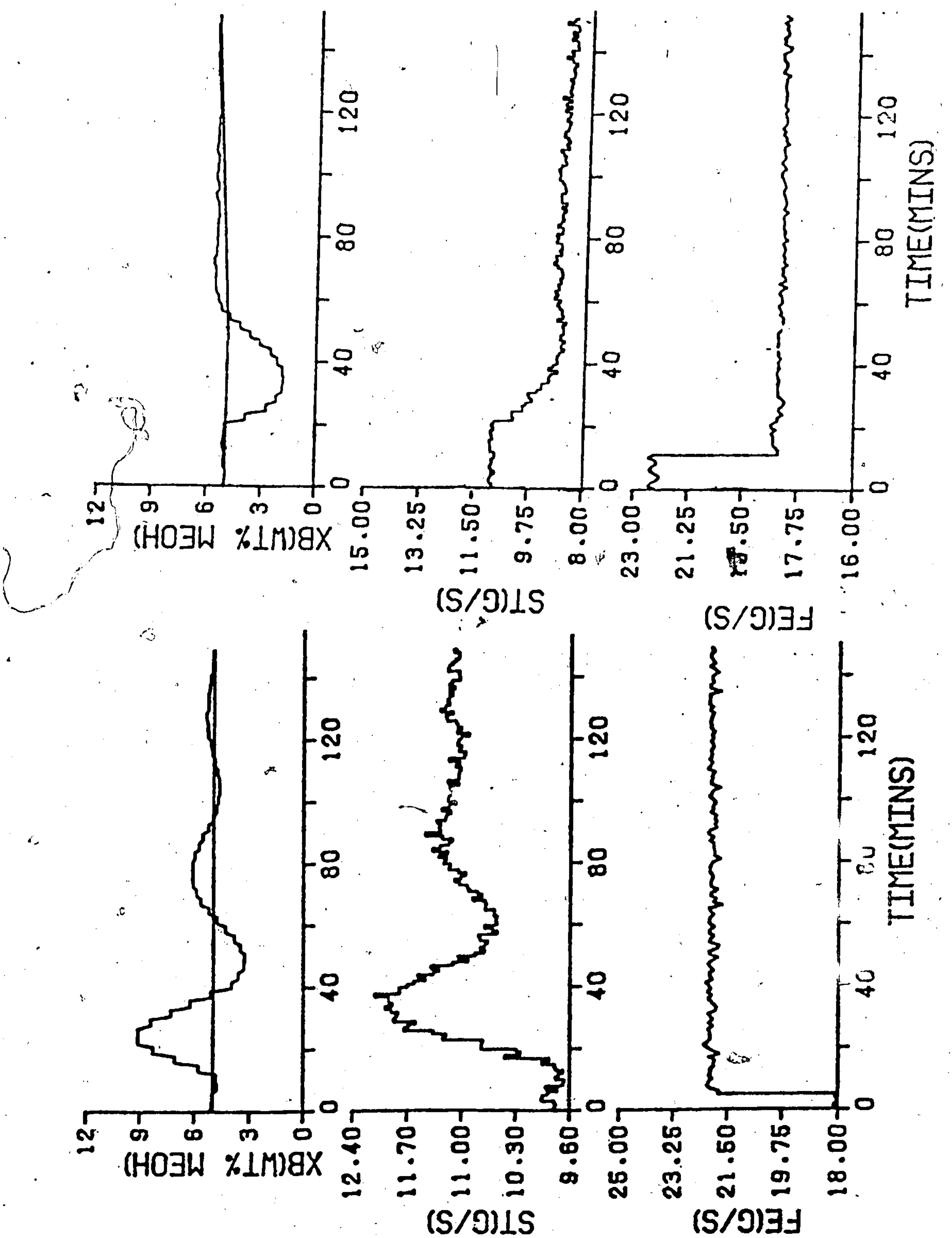


Figure 6.12 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator

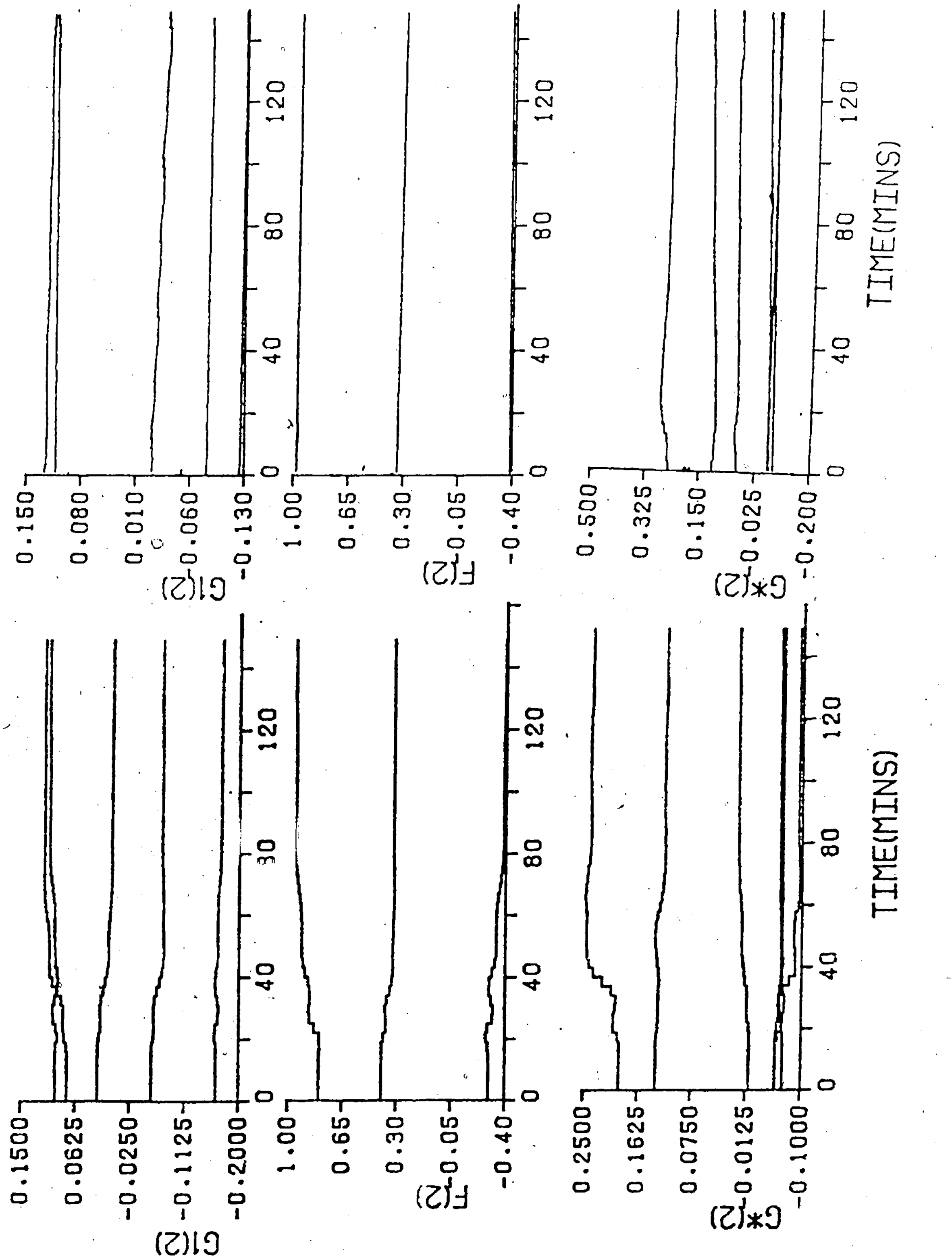


Figure 6.13 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Square Root Estimator

under PID control. When the feed flow rate was decrease to the steady state value, all of the bottom loop parameter estimates showed very small or no changes with the values of the previously converged values or new values adapted to within 30 minutes.

The step change which decreased the feed flow to a value below the steady state value caused and 0.8% increase in top composition from the desired setpoint as can be seen from the results in Figure 6.14. The composition then decreased and reached a value 0.55% below the desired setpoint and did not return to the setpoint for 100 minutes. The return of the feed flow rate to its steady state value brought about a 1% decrease in top composition. After 100 minutes the composition became relatively constant however the top composition exhibited some offset as it remained slightly below the desired setpoint so the control performance for these two feed disturbances was not satisfactory.

The G and G* top loop parameter estimates, shown in Figure 6.15 exhibited significant changes when the feed flow rate was reduced 25% from the steady state values while the F parameters did not change. The G* parameter estimates converged 100 minutes after the step change but the G parameter estimates were still changing when the next step change was made. When the feed flow rate was returned to the steady state value the G and G* parameter estimates started adaption again, however, the changes observed for the G

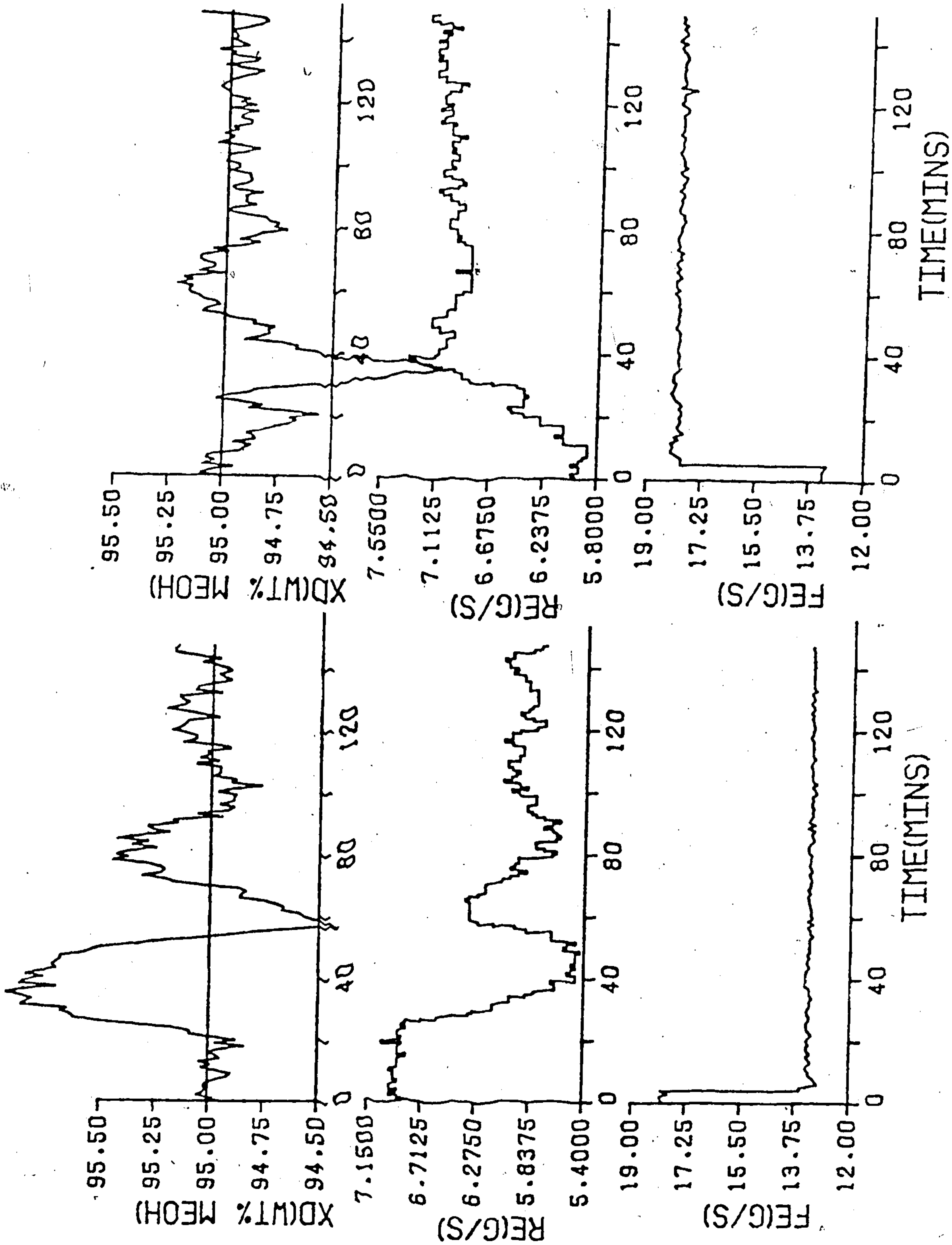


Figure 6.14 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator

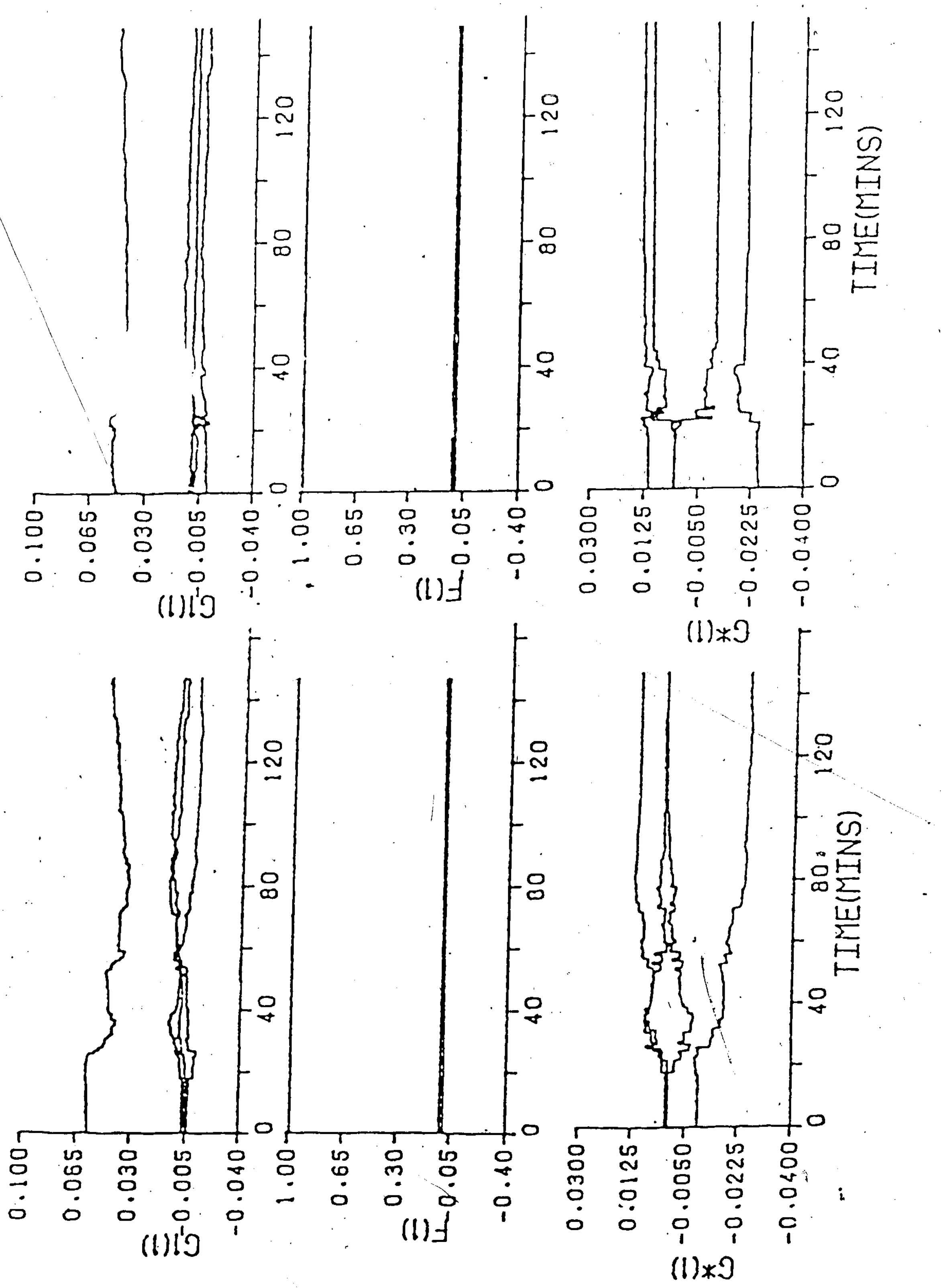


Figure 6.15 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator

estimates were not as large as those observed for the previous step change. Again the G parameter estimates did not converge while the G* estimates converged in only 60 minutes.

The bottom composition was severely affected by the 25% decrease in feed flow rate as can be seen from the results in Figure 6.16. The concentration of methanol was as low as 1.5% and then became as high as 8.0% before returning to the desired setpoint 110 minutes later. For the return of the feed rate to the steady state value there was a marked improvement in the control performance as even though the bottom composition showed a deviation of 5.5% it returned quickly to the setpoint, within 80 minutes.

The behavior of the estimates of the parameters for the bottom loop presented in Figure 6.17, exhibited only minor adaption for both step changes. The parameter estimates converged after 110 minutes for the decrease in feed flow rate from the steady state value but only took 60 minutes to converge when the feed flow was increased to its steady state value.

Therefore the control performance observed when the recursive square root estimator is used is not acceptable for all changes as cycling about the desired top composition setpoint and offset occurred for the 25% step increase in feed rate from the steady state feed rate and the subsequent return to the steady state value, respectively. It is to be noted, by reference to Figures 6.13 and 6.17, that the

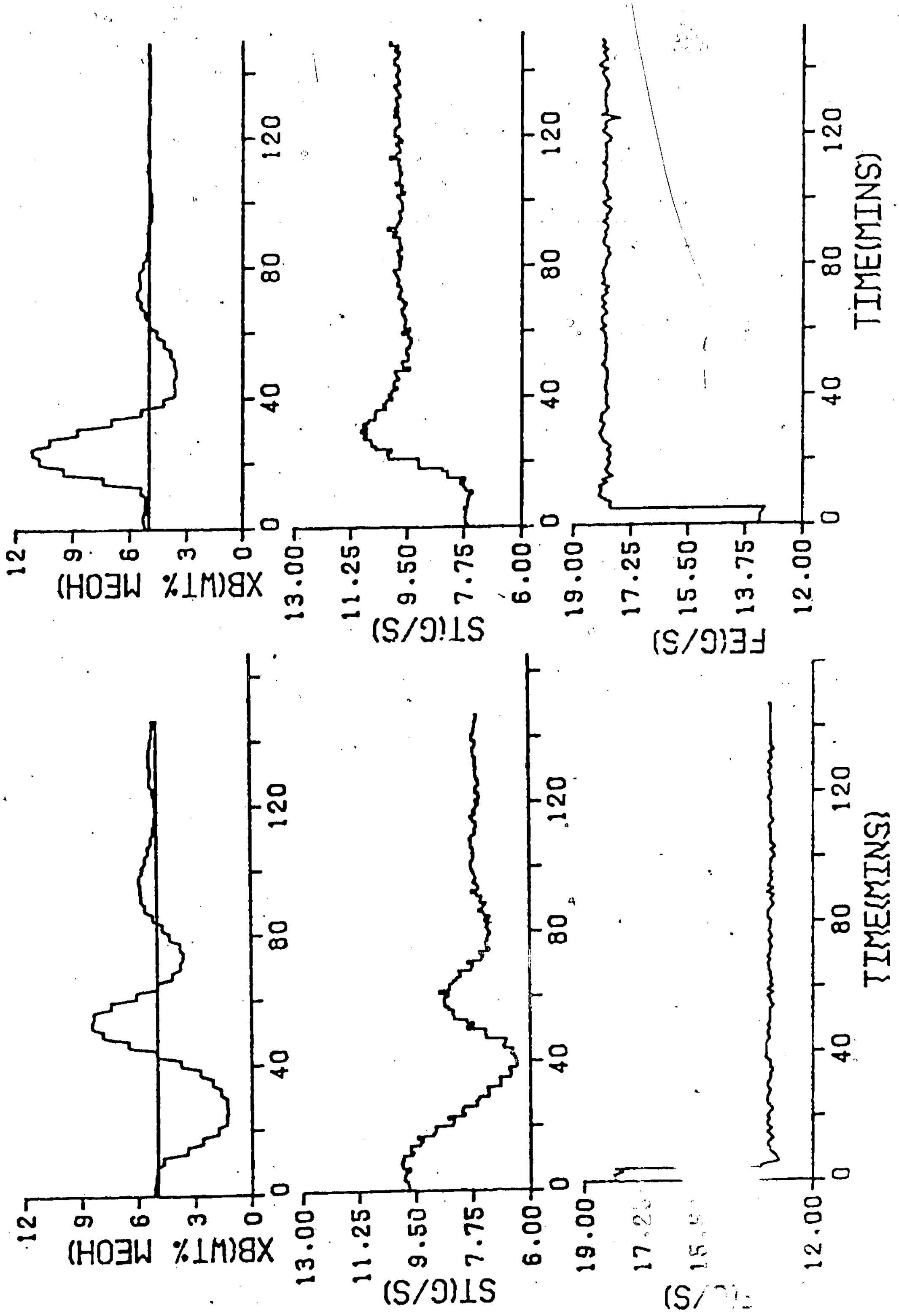


Figure 6.16 Response of the Bottom Loop to [illegible] 5% Step Decrease in Feed Flow Rate and the Subsequent 2[illegible] Step Increase in Feed Rate using the Recursive Square Root Estimator

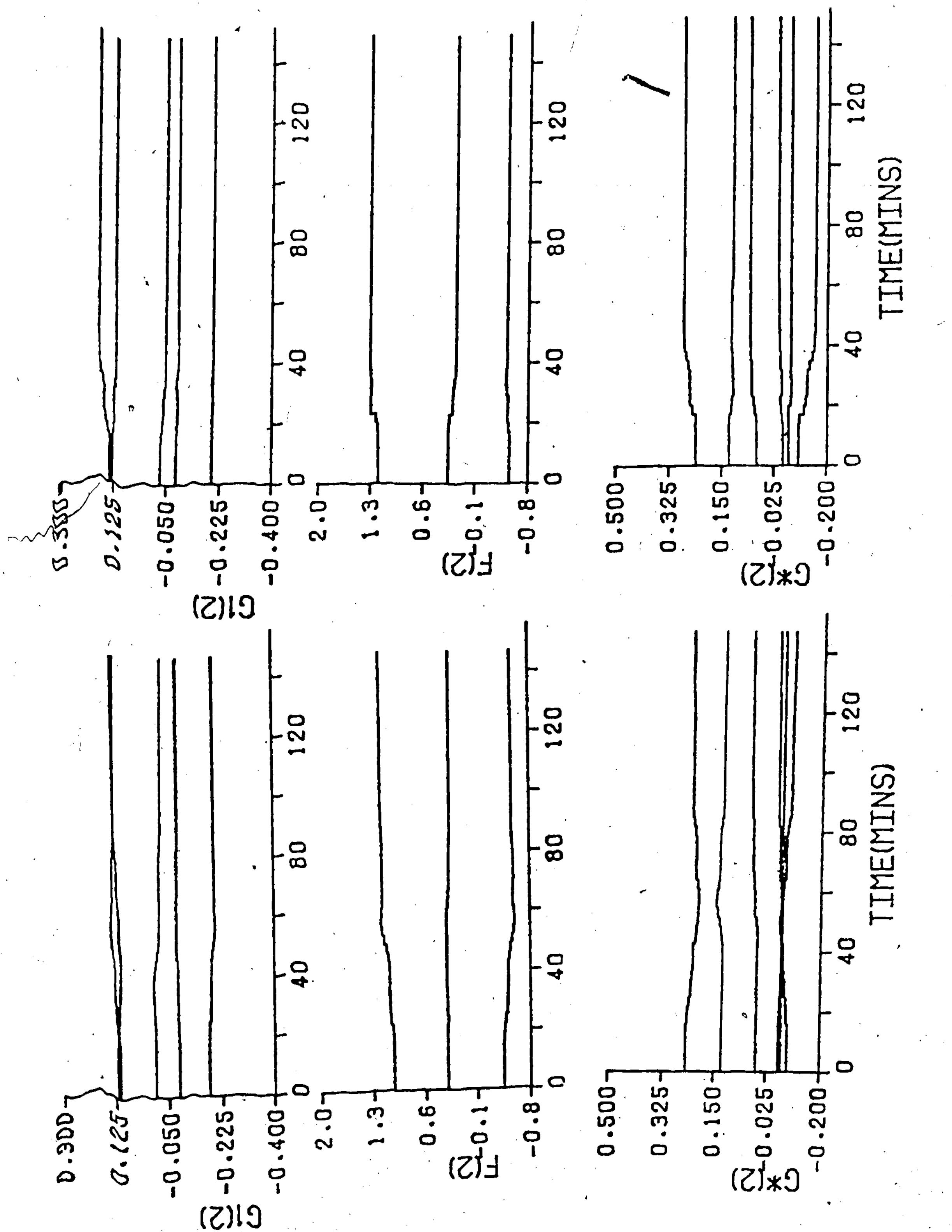


Figure 6.17 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Square Root Estimator

bottom parameter estimates for the entire experiment do not change substantially which may be the reason for the unsatisfactory control performance.

6.3 Recursive Upper Diagonal Factorization Estimation Evaluation

Testing of the recursive upper diagonal factorization estimator did not require the initial adaption period of 200 minutes with the column operating under PID control to establish suitable initial parameter estimates. The self-tuning controller was implemented immediately without adverse effects. As expected this caused the results to be different than those using the RSR estimator since the initial estimate values were different.

The top composition reached the desired setpoint 30 minutes after the self-tuning controller was implemented. As shown in Figure 6.18 the 25% step increase in feed flow rate caused the top composition to exceed the setpoint by 0.5% and then fall to a composition 0.45% below the setpoint before returning, in about 60 minutes, to the desired setpoint composition where it fluctuated until the change in flow rate. As can be seen from the results, this change in feed flow rate to the steady state value resulted in top composition deviations of about -0.25% and +0.25% before returning to the desired setpoint 60 minutes after the step change.

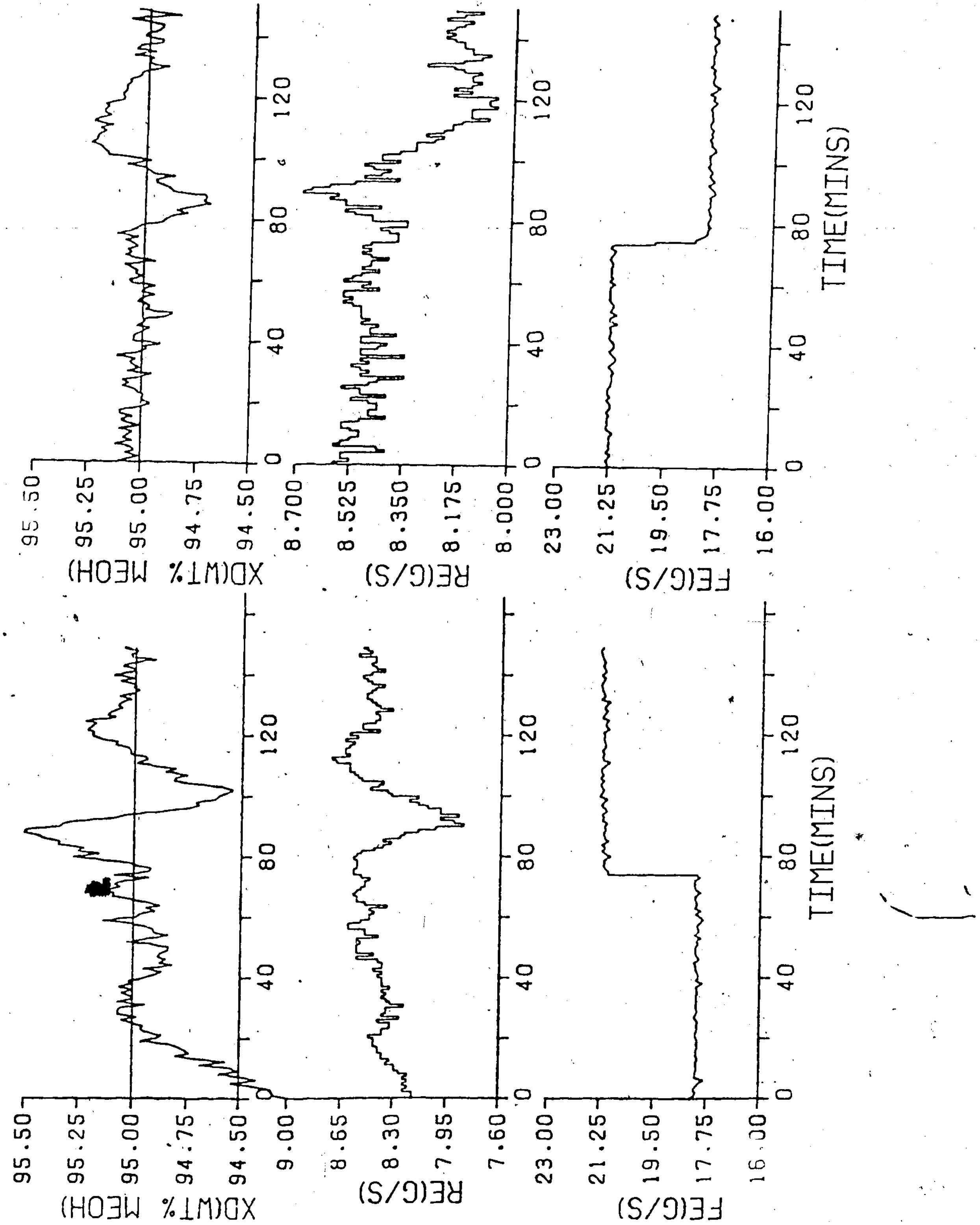


Figure 6.18 Response of the Top Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

The F and G parameter estimates for the top loop exhibited very little adaption as can be seen from the results presented in Figure 6.19. The G* parameter estimates showed more initial changes than the F and G estimates. At the introduction of the 25% step change in feed rate, the F parameter estimates made small changes and the G parameter estimates had slightly larger changes while the G* estimates exhibited the largest changes of all the parameter estimates. The adaption pattern continued when the feed flow rate returned to the steady state value.

Comparison of the results in Figure 6.18 with those in Figure 6.20 for the first 150 minutes reveals that the bottom composition exhibits the same trends as the top composition, however, the initial deviation from the desired setpoint for the bottom compositions was greater. There was a 1% deviation initially and a 3% error after the 25% step increase in feed rate. Return of the flow rate to the steady state value caused the bottom composition to fall 3% below the desired setpoint and then return to the desired setpoint in about 60 minutes as was the case for the increase in feed rate.

The bottom loop parameter estimates, given in Figure 6.21, showed no significant changes when the self-tuning controller was first implemented. When the feed flow rate was increased by 25% from the steady state value the G* parameter estimates adapted for 60 minutes before converging while the F and G parameter estimates converged 30 minutes

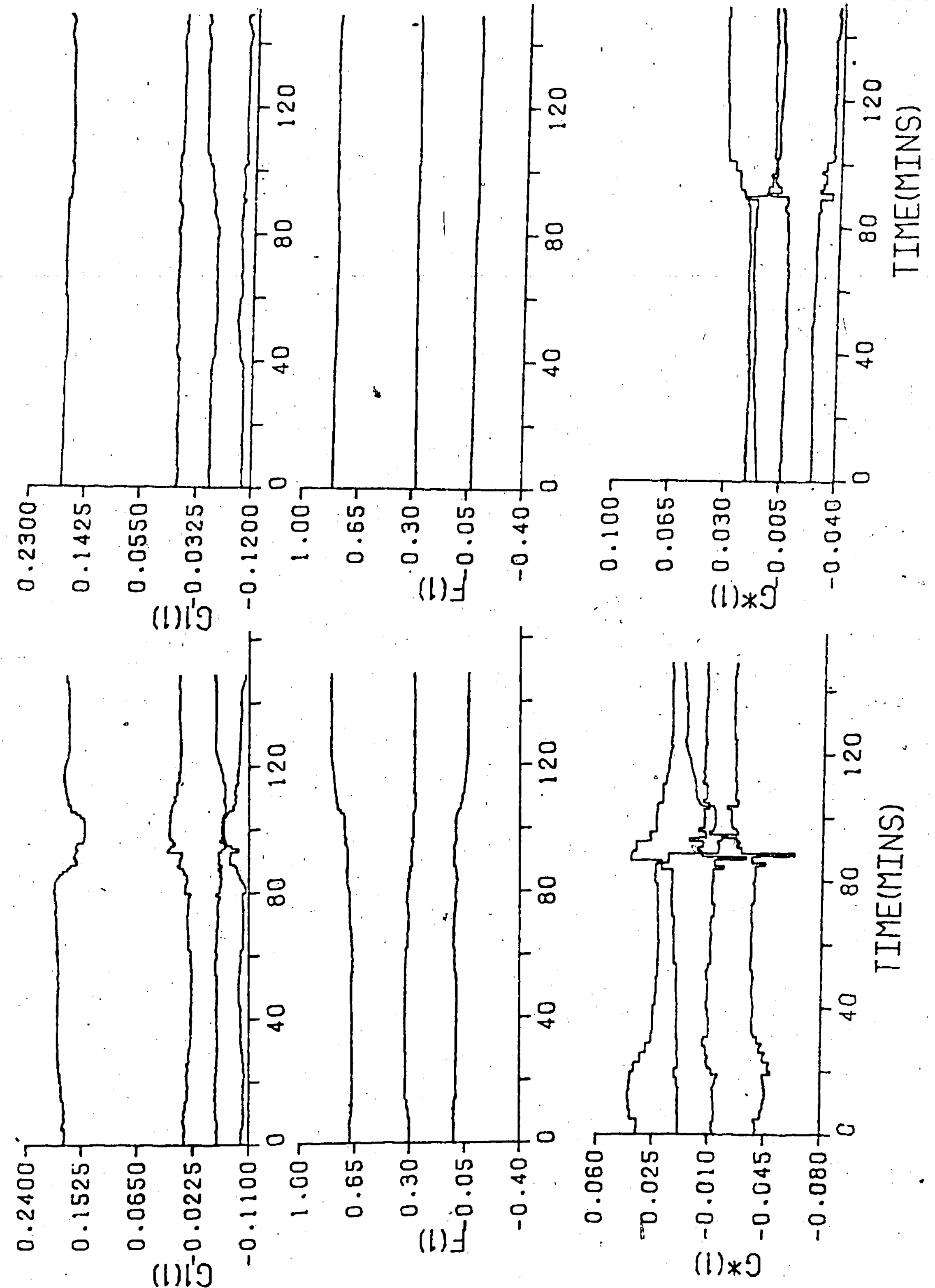


Figure 6.19 Top Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

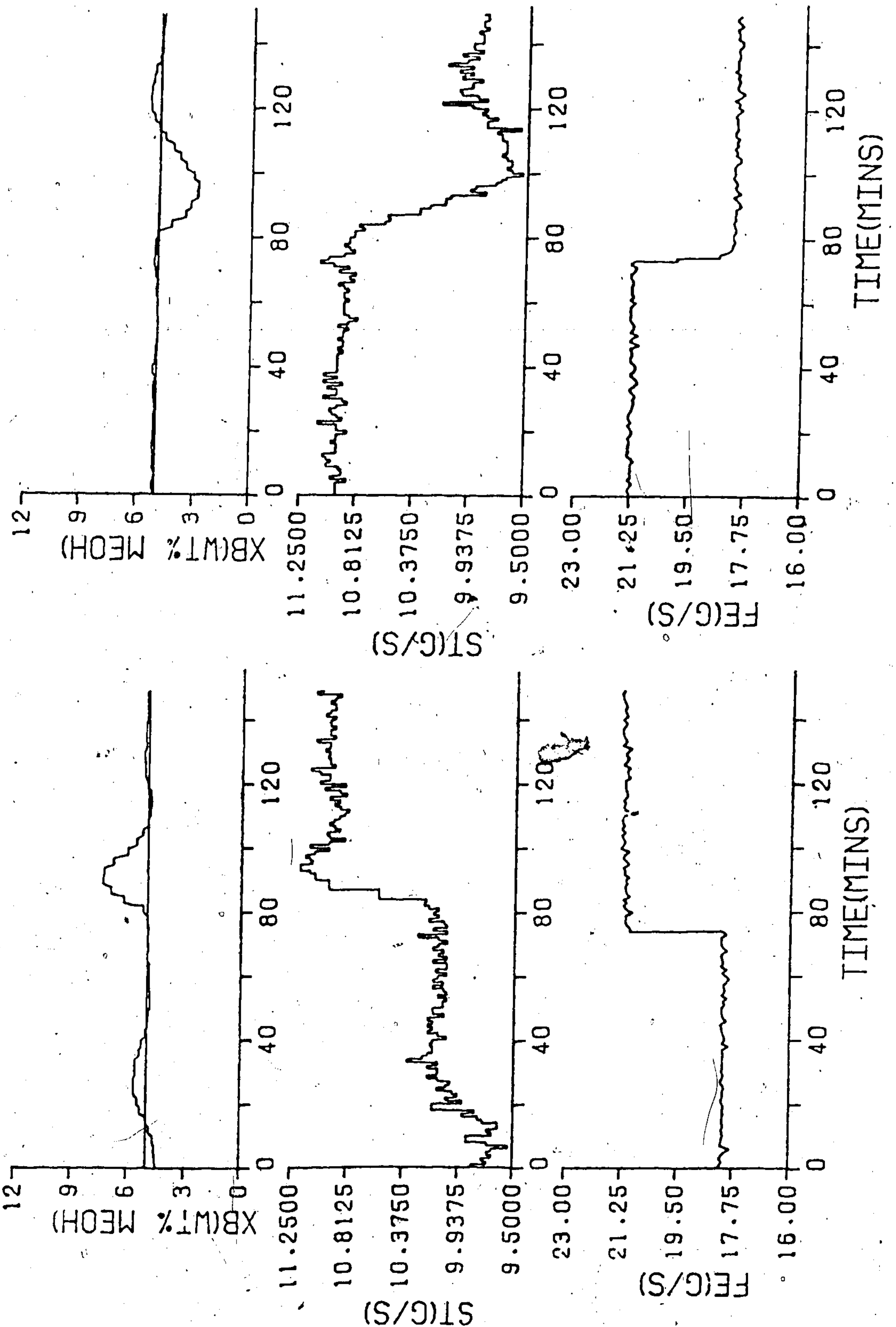


Figure 6.20 Response of the Bottom Loop to the 25% Step Increase in Feed Flow Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

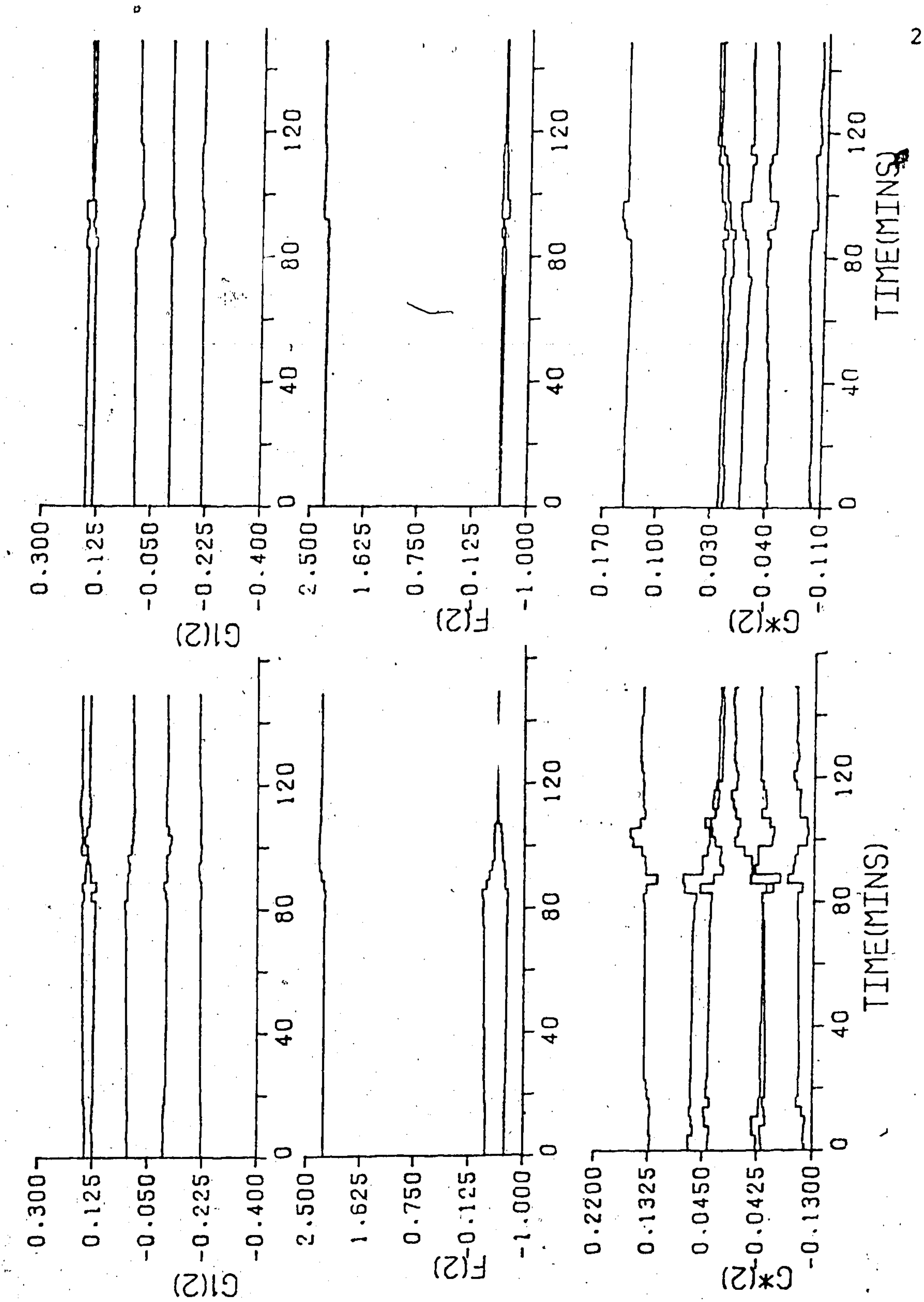


Figure 6.21 Bottom Loop Parameter Estimates for the 25% Step Increase in Feed Rate and the Subsequent 25% Step Decrease in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

after the step change. From the 150th minute of the experiment until the next step change at the 210th minute the F and G parameter estimates converged while the G* estimates showed minimal changes. After the step change which returned the feed to the steady state value, the F, G and G* parameter estimates only changed slightly and converged in 60 minutes.

The step change which decreased the feed flow rate by 25% from the steady state value caused the top composition to increase 0.8% above the desired setpoint as seen in Figure 6.22. This increase was followed by a composition 0.2% below the setpoint with composition returning to the desired setpoint after 80 minutes. Return of the feed flow to the steady state value caused a major decrease of 1% in the top composition but the composition returned to the desired setpoint only 40 minutes after the feed disturbance.

The parameter estimates for the top loop exhibited changes, as illustrated in Figure 6.23, when the feed flow rate was decreased from the steady state value. None of the parameter estimates converged until the feed rate was returned to the steady state value even though the adaption that occurred at the step change was not significant. As the results show, the increase in the feed flow rate caused larger changes in the G and G* parameter estimates than in the F estimates however all parameter estimates had converged after 40 minutes.

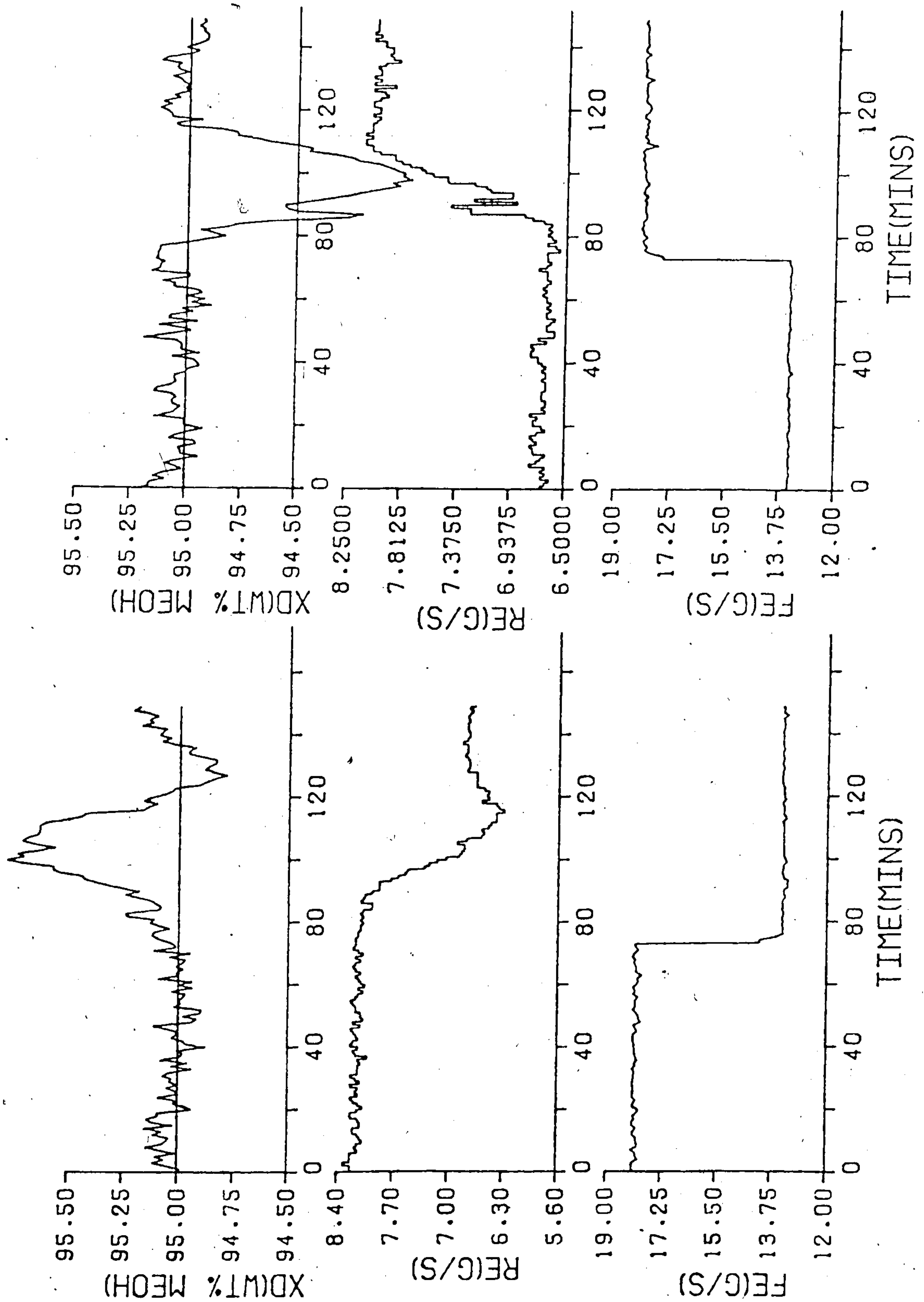


Figure 6.22 Response of the Top Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

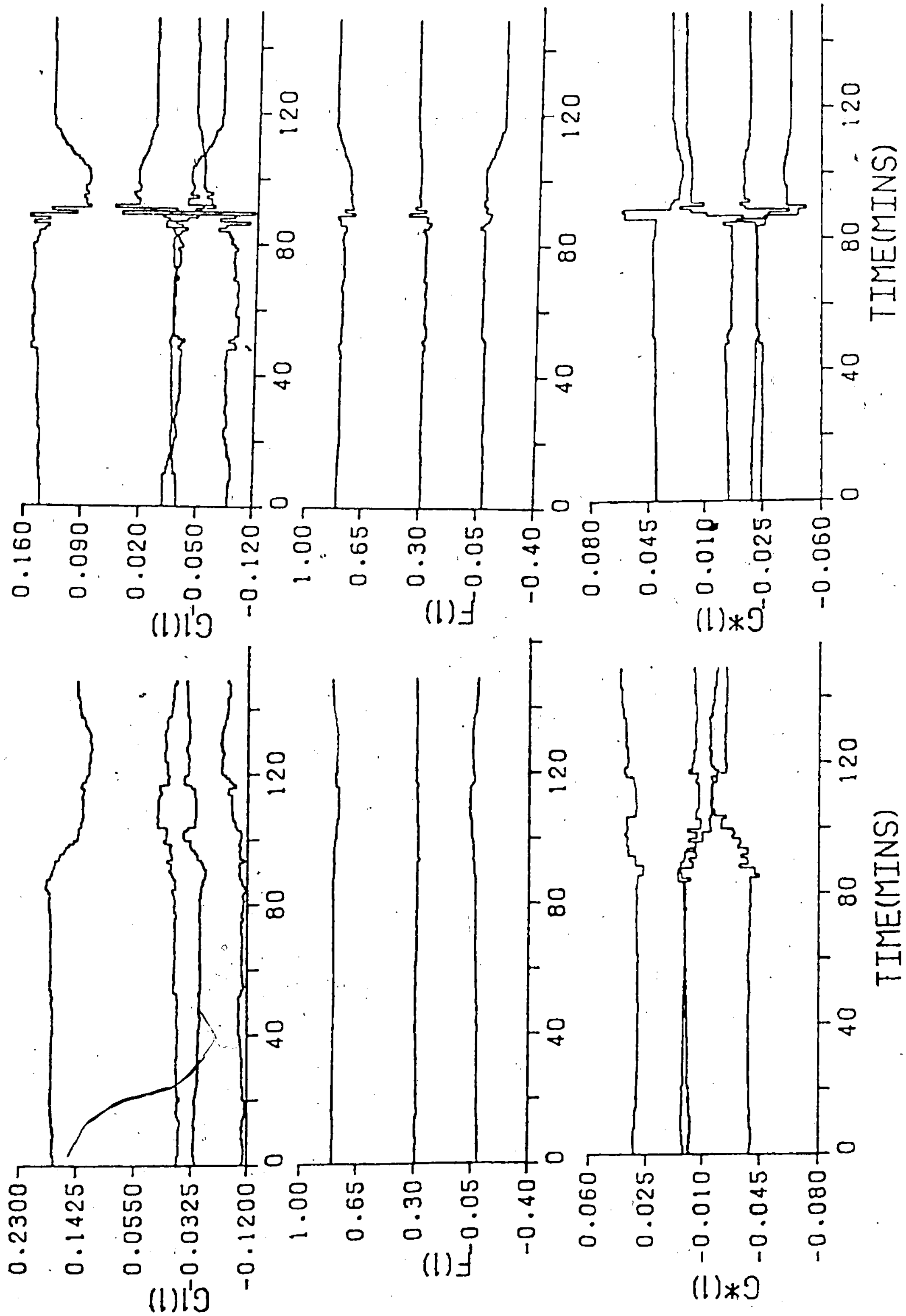


Figure 6.23 Top Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

Comparison of the bottom composition response in Figure 6.24 with that in Figure 6.20, for the 25% decrease in feed rate displayed the same trend as the response when the feed flow was returned to the steady state value from the 25% increase. Again the bottom composition deviated from the desired setpoint however the maximum error was larger with the composition falling to 1.5% mass percent methanol instead of only 3% methanol. When the feed rate increased to the steady state value the initial deviation was 5% above the desired bottom composition but, the composition returned to the setpoint in only 40 minutes.

The bottom loop parameter estimates, presented in Figure 6.25, showed that the parameter values changed for both step changes in feed rate. The F and G parameter estimates for the decrease in rate converged before the next step change yet, the G* estimates were still adapting. The same F and G parameter estimate behavior occurred when the feed rate was returned to the steady state value while the G* estimates had begun to converge unlike the response to the previous step change.

From the evaluation of the results using the RUD estimator it can be concluded that the control performance is better than that observed for the RLS and RSR estimators. There was no offset or cycling about the desired setpoint after the step changes in feed flow rate and the length of time that the top and bottom composition deviated from the setpoint was less than for the other two estimators. The

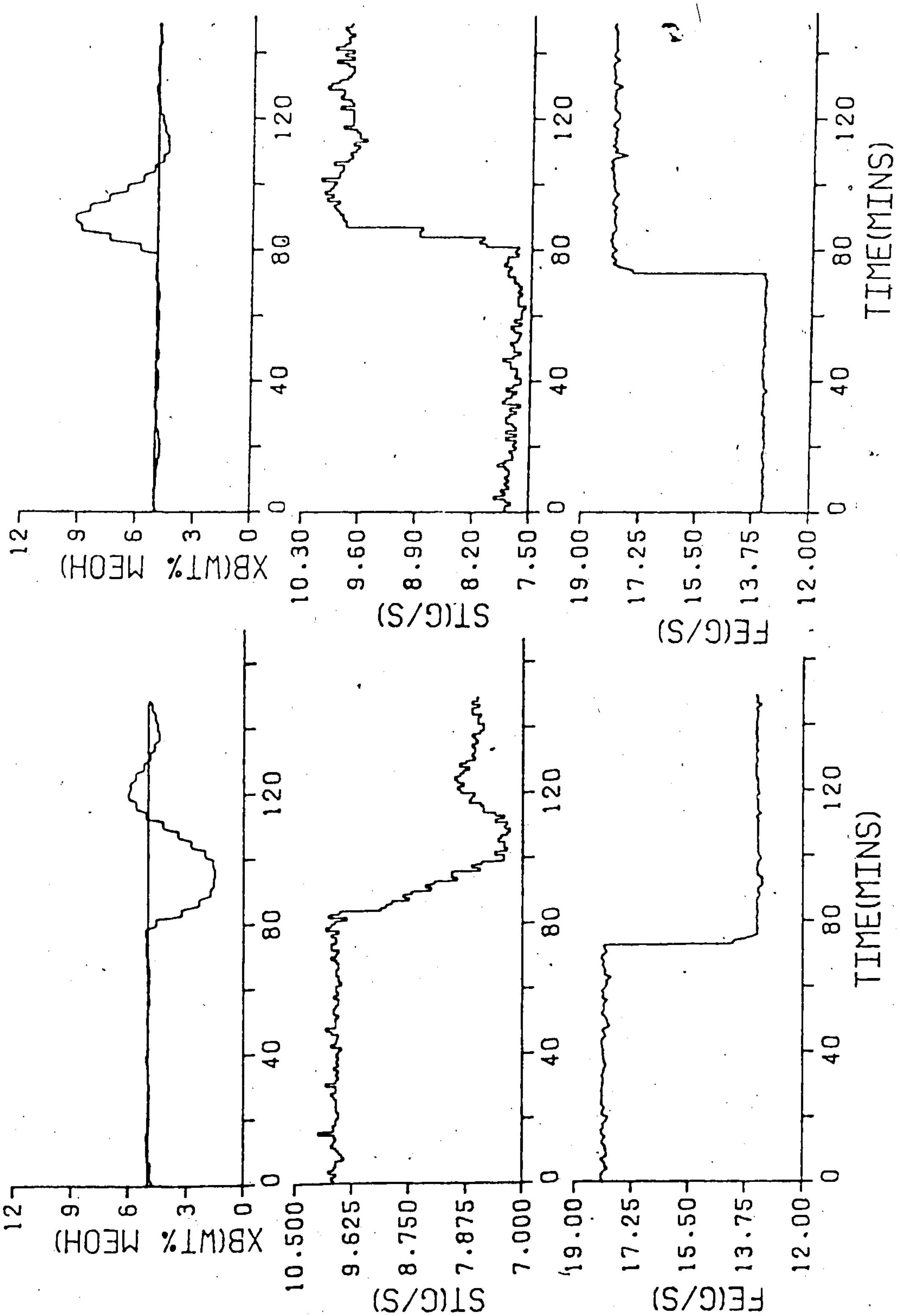


Figure 6.24 Response of the Bottom Loop to the 25% Step Decrease in Feed Flow Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

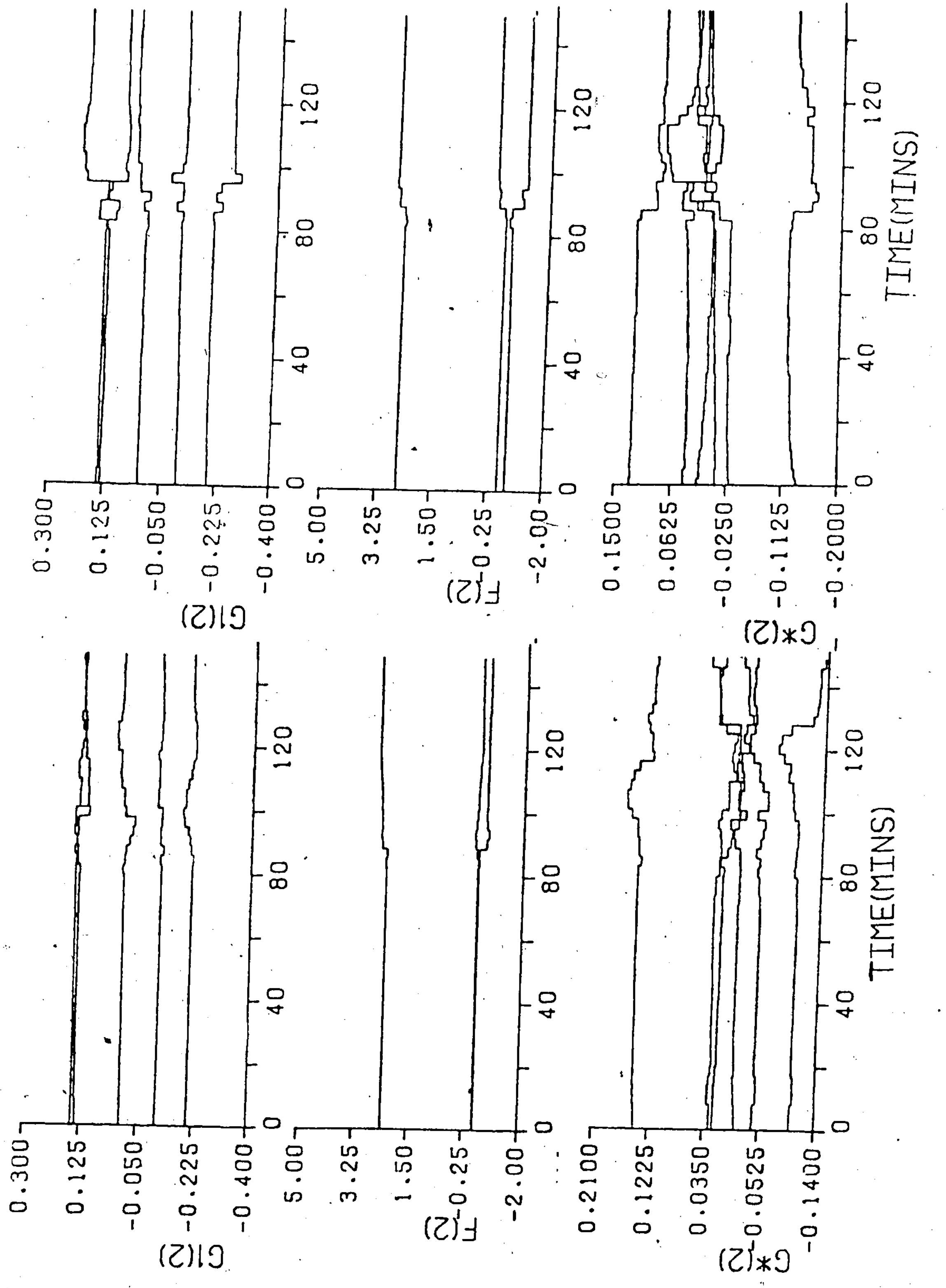


Figure 6.25 Bottom Loop Parameter Estimates for the 25% Step Decrease in Feed Rate and the Subsequent 25% Step Increase in Feed Rate using the Recursive Upper Diagonal Factorization Estimator

ability of the self-tuning controller with RUD identification to maintain satisfactory control performance without the first 200 minutes of PID control and identification to obtain good initial parameter estimates is also an improvement over the two identification techniques.

6.4 Recursive Learning Estimation Evaluation

The final parameter estimation method tested was the recursive learning method. As with the RLS and RSR estimators PID control with recursive learning estimation was implemented for 200 minutes before the self-tuning controller was used. The control performance, as shown in Figure 6.26 and 6.27, before the step change in feed is satisfactory. However, when the flow rate was increased the column became unstable and the top and bottom compositions oscillated unbounded. As can be observed, the F and G parameter estimates for the top loop showed no adaption and the remaining parameter estimates for both loop exhibited erratic, oscillating adaption patterns.

The results would imply that the recursive learning estimator does not identify parameters which the self-tuning controller can use as a minimum variance controller to provide satisfactory control performance. In fact, the estimation method may not be capable of adapting fast enough to compensate for the nonlinearities of the distillation column. Furthermore as shown in Figure 4.21 and 4.26, good initial parameter estimates and stable operation are

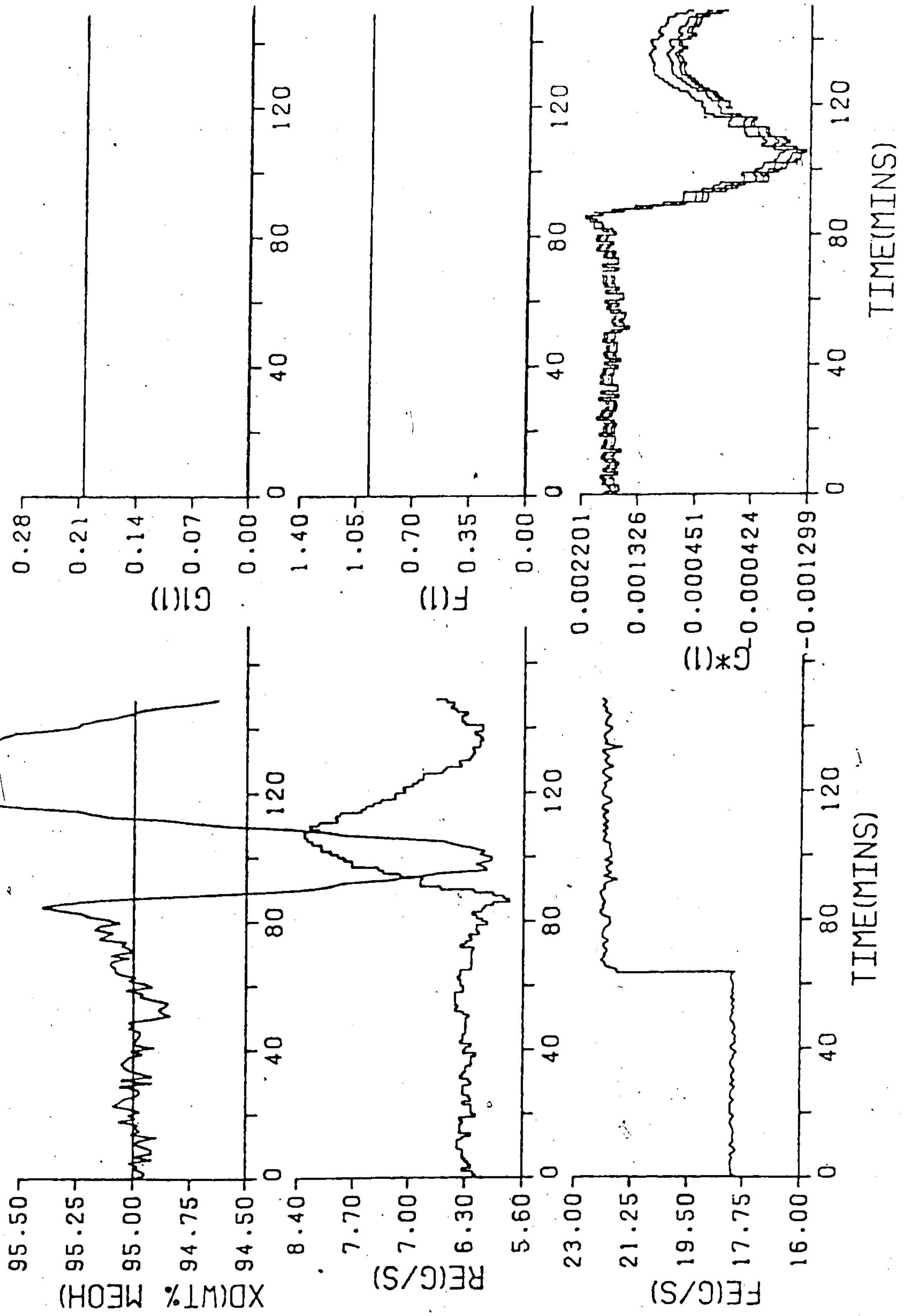


Figure 6.26 Control Performance of the Top Loop and the Parameter Estimates for the 25% Step Increase in Feed Rate using the Recursive Learning Estimator

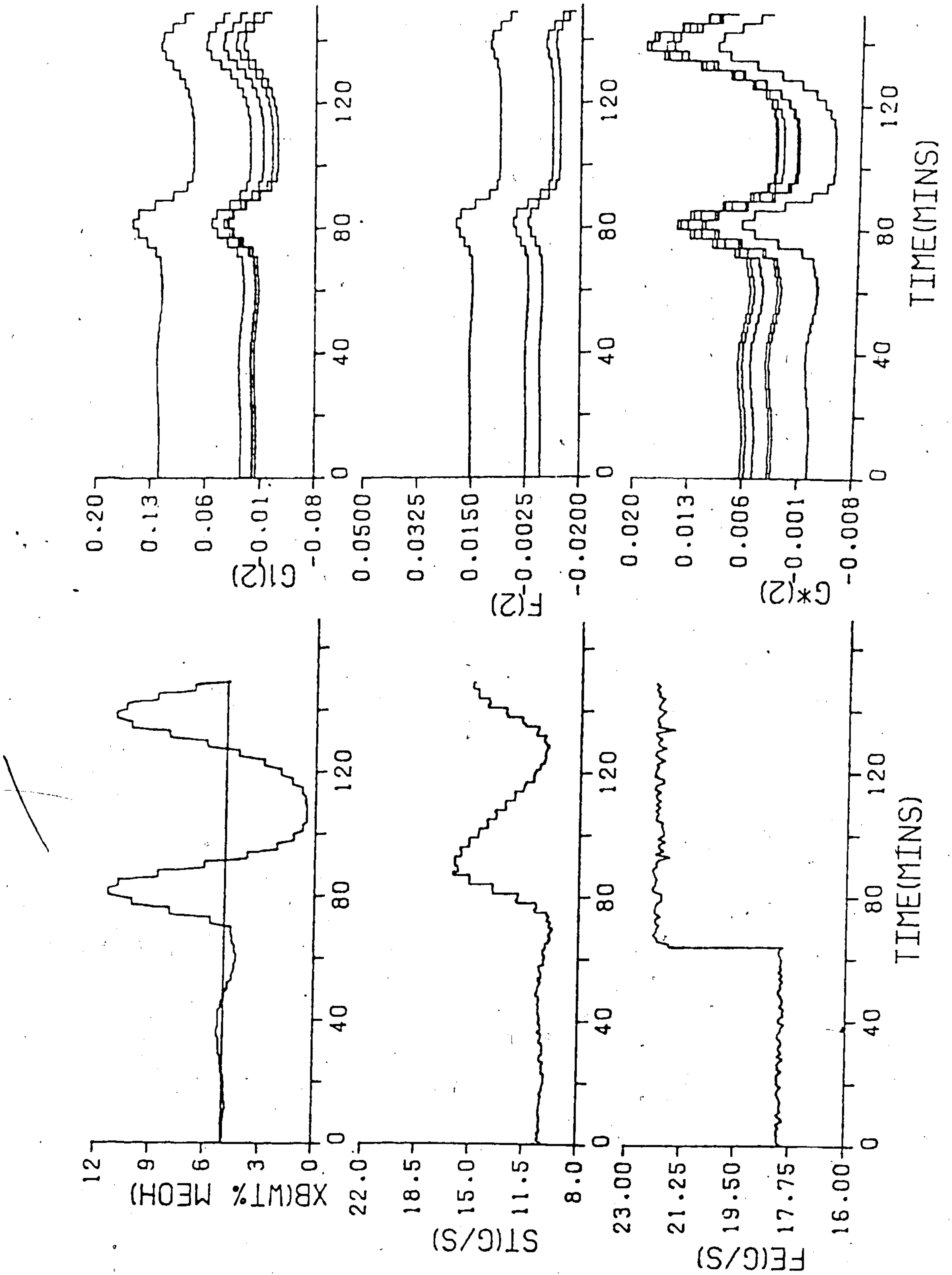


Figure 6.27 Control Performance of the Bottom Loop and the Parameter Estimates for the 25% Step Increase in Feed Rate using the Recursive Learning Estimator

required to use the recursive learning estimator with the self-tuning controller. P, Q and R weighting may also be required for the self-tuning controller with RL estimation to be used for distillation column control.

As was the case for both the linear and distillation column model simulations the performance of the recursive upper diagonal factorization estimator has provided the best control performance. Since these test were performed only for a minimum variance self-tuning controller, the addition of any weighting should enhance the performance of the recursive upper diagonal factorization identification technique.

7. Conclusions

Utilization of the recursive least squares, recursive square root, recursive upper diagonal factorization, recursive learning and recursive maximum likelihood estimators for use with the self-tuning controller has been studied. Simulations have been performed using a linear model to investigate the performance of the different algorithms for efficiency and accuracy and to analyze the effects of the various estimator parameters such as initial covariance matrix values, forgetting factors and initial estimate values on the parameter estimates. Similar simulations using a nonlinear model of the distillation column were undertaken and a further evaluation conducted by performing experiments with the pilot scale distillation column. On the basis of the simulation results and the experimental data the following conclusions can be stated:

1. The recursive upper diagonal factorization (RUD) estimator proved to be the most accurate and efficient identification technique not only for the linear and nonlinear simulations but also during experimental testing. The simulation results showed that the use of the recursive square root (RSR) estimator instead of the RUD method gave rise to an identical control performance but since the algorithm required the calculation of square roots, it was more time consuming than simple arithmetic operations associated with the RUD. The experimental performance favored the RUD identification

as the RSR estimator required that column to be operated under PID control before the self-tuning controller was implemented in order to provide the self-tuning controller with initial parameter estimate other than zero so that the column operation did not become unstable. The RUD estimator, on the other hand, was able to identify parameter estimates that provided good control performance without the column operating under PID control and is therefore the most desirable estimation algorithm as it required the least effort to implement.

2. Simulations performed using the recursive least squares (RLS) estimator for control of linear and nonlinear systems, with little excitation, showed that the parameter estimates would usually exhibit estimate blowup due to the inherent numerical deterioration of the estimator covariance matrix.
3. Control performance tests with the recursive learning (RL) method of estimation was characterized by poor setpoint tracking ability. Furthermore the lack of a covariance matrix resulted in slow parameter estimate adaption which caused the unsatisfactory performance.
4. The recursive maximum likelihood (RML) estimator was unable to identify parameter estimates that would provide satisfactory control performance even for the simulation studies so no further testing could be performed. No specific cause for this behavior was

determined.

5. The addition of Q and R weighting reduced the control effort in the linear simulations for all estimation algorithms and improved the control performance only for the recursive least squares, recursive square root, recursive upper diagonal factorization and recursive learning estimators. Simulation results obtained using the recursive learning estimator with Q and R weighting in the linear system were inferior to the other three identification methods.
6. Use of the recursive least squares, recursive square root and recursive upper diagonal factorization estimators provided acceptable control performance in the linear system simulations for the disturbance rejection tests but the recursive learning and the recursive maximum likelihood estimators could not compensate for the disturbance. The simulation of a linear system and the introduction of a load disturbance with the recursive learning estimator caused the output to go unstable while the recursive maximum likelihood estimator results showed offset.
7. The initial covariance matrix values of the RLS, RSR and RUD estimators should be chose to compromise between the requirement for good setpoint tracking and fast parameter estimate convergence.
8. The forgetting factor, used for all of the estimation routines except for the recursive maximum likelihood

estimator should be chosen on the basis of expected variations in the system parameters. For slowly changing parameters the forgetting factor should be near 1.0 while it would be more appropriate to use 0.95 for rapidly varying parameters.

In conclusion, the recursive upper diagonal factorization estimator is the preferred choice for a stable and efficient parameter estimate identification technique.

8. Further Work

In an effort to more completely understand the effects of the parameter estimation techniques on the self-tuning controller performance, further investigations might deal with the following:

1. Compare the performance of the different parameter estimation techniques when on-line tuning of the P, Q and R weighting is used to obtain optimal weighting.
2. Examine by the use of simple systems the reason that the recursive maximum likelihood estimator is unable to provide satisfactory closed loop identification.
3. Study the performance of different estimation techniques when measured load disturbances are used for feedforward control on the distillation column.

Finally, the use of the recursive upper diagonal factorization estimator with the self-tuning controller on a practical application such as a full-scale distillation tower in industry has great potential and further study would prove most beneficial.

9. References

1. Astrom, K.J., and Wittenmark, B., On Self-Tuning Regulators, Automatica, Volume 19, 1973, pp 185-199.

2. Clarke, D.W., and Gawthrop, P.J., Self-Tuning Controller, IEE Proc., Volume 122, Number 9, 1975, pp 929-934.

3. Clarke, D.W., and Gawthrop, P.J., Self-Tuning Control, IEE Proc., Volume 126, Number 6, 1979, pp 635-640.

4. Borisson, U. and Syding, R., Self-Tuning Control of an Ore Crusher, Automatica, Volume 12, 1976, pp 1-7.

5. Dumont, G.A., and Belanger, P.R., Self-Tuning Control of a Titanium-Dioxide Kiln, IEEE Trans. Aut. Contr., Volume AC-23, Number 4, 1978, pp 588-592.

6. Keviczky, L., Hetthessy, J., Hilger, M., and Kolostori, J., Self-Tuning Adaptive Control of Cement Raw Material Blending, Automatica, Volume 14, 1978, pp 525-532.

7. Sastry, V.A., Seborg, D.E. and Wood, R.K., A Self-Tuning Regulator Applied to a Binary Distillation Column, Automatica, Volume 13, 1977, pp 417-424.

8. Cegrell, T., and Hedqvist, T., Successful Adaptive Control of Paper Machines, Automatica, Volume 11, 1975, pp 53-59.

9. MacGregor, J.F., and Tidwell, P.W., Discrete Stochastic Control with Input Constraints, IEE Proc., Volume 124, Number 8, 1977, pp 732-734.

10. Clarke, D.W., Self-Tuning and Adaptive Control, edited by C.J. Harris and S.A. Billings, Peter Peregrinus Ltd., New York, 1981, pp 50, 54, 144-160.

11. Ljung, L., and Wittenmark, B., Analysis of a Class of Adaptive Regulators, Proceedings of the IFAC Stochastic

Control Symposium, Budapest, 1974, pp 431-437.

12. Song, H.K., Derivation and Experimental Evaluation of a Stable Adaptive Controller, PhD Thesis, Department of Chemical Engineering, The University of Alberta, Edmonton, Alberta, 1983.

13. Gawthrop, P.J., Some Interpretations of the Self-Tuning Controller, IEE Proc., Volume 124, Number 10, October 1977, pp 889-894.

14. Astrom, K.J., and Eykhoff, P., Identification Methods - A Survey, Automatica, Volume 7, 1971, pp 123-162.

15. Graupe, D., Identification of Systems, Van Nostrand and Reinhold Company, New York, N.Y., 1972, pp 135-140.

16. Eykhoff, P., Trends and Progress in System Identification, Pergamon Press, New York, N.Y., 1981.

17. Hsia, T.C., System Identification, Lexington Books, D.C. Heath and Company, Toronto, Ontario, 1977, pp 97-100.

18. Clarke, D.W., Generalized Least Squares Estimation of the Parameters of a Dynamic Model, IFAC Symposium on Identification in Automatic Control Systems, Prague, Paper 3.17, 1967.

19. Hasting-James, R. and Sage, M.W., Recursive Generalized Least Squares for On-Line Identification of Process Parameters, IEE Proc., Volume 116, Number 12, 1969, pp 2057-2062.

20. Battin, R.H., Astronautical Guidance, McGraw-Hill, New York, 1964, pp 338-339.

21. Kaminski, P.G., Bryson, A.E., and Schmidt, S.F., Discrete Square Root Filtering: A Survey of Current Techniques, IEEE Trans. Aut. Contr., Volume AC-16, Number 6, 1971, pp 727-735.

22. Peterka, V., A Square Root Filter for Real Time Multivariate Regression, Kybernetika, Volume 11, Number

1, 1975, pp 53-67.

23. Bierman, G.J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977, p40.

24. Wampler, R.H., A Report on the Accuracy of Some Widely Used Least Squares Computer Programs, Journal of the American Statistics Association, Volume 65, Number 330, 1970, pp 549-565.

25. Bierman, G.J., Measurement Updating Using the U-D Factorization, Automatica, Volume 12, Number 4, 1976, pp 375-382.

26. Nagumo, J., and Noda, A., A Learning Method of System Identification, IEEE Trans. Aut. Contr., Volume AC-12, Number 3, 1967, pp 282-287.

27. Doob, J.L., Stochastic Processes, John Wiley & Sons, New York, N.Y., 1953, p323.

28. Franklin, G.F., and Powell, J.D., Digital Control of Dynamic Systems, Addison-Wesley Publishing Company, Don Mills, Ontario, 1981, pp 234-243.

29. Soderstrom, T., An On-Line Algorithm for Approximate Maximum Likelihood Identification of Linear Dynamic Systems, Report 7308, Lund Institute of Technology, Division of Automatic Control, 1973, pp 2-10.

30. Gerald, C.F., Applied Numerical Analysis, Second Edition, Addison-Wesley Publishing Company, Don Mills, Ontario, 1980, p21.

31. Morris, A.J., Nazer, Y. and Chisholm, K., A Comparison of Identification Techniques for Robust Self-Tuning Control (unpublished paper).

32. Ljung, L., Gustavsson, I., and Soderstrom, T., Identification of Linear Multivariable Systems Under Linear Feedback Control, IEEE Trans. Aut. Contr., Volume AC-19, 1974, pp 836-841.

33. Ljung, L., Analysis of Recursive Stochastic Algorithms, IEEE Trans. Aut. Contr., Volume AC-12, Number 4, 1977, pp 551-575.

34. Bierman, G.J. and Thornton, C.L., Numerical Comparisons of Kalman Filter Algorithms: Orbit Determination Case Study, Automatica, Volume 13, 1977, p23-35.

35. Kan, H.W., Binary Distillation Column Control: Evaluation of Digital Control Algorithms, M.Sc. Thesis, The University of Alberta, Edmonton, Alberta, 1982.

36. Simonsmeier, U.F. Nonlinear Binary Distillation Column Models, M.Sc. Thesis, Department of Chemical Engineering, The University of Alberta, Edmonton, Alberta, 1977.

37. Bilec, R.J., Modelling and Dual Control of a Binary Distillation Column, M.Sc. Thesis, The University of Alberta, Edmonton, Alberta, 1980.

38. Nazer, Y., Single and Multivariable Self-Tuning Controllers, PhD Thesis, Department of Chemical Engineering, University of Newcastle upon Tyne, England, 1981.

39. Vagi, F.J., M.Sc. Thesis, Department of Chemical Engineering, The University of Alberta, Edmonton, Alberta, to be published.

10. Appendix A

The following is the Agee Turner positive definite factorization theorem. Let

$$P = U D U^t = U D U^t + c\,a\,a^t \tag{A.1}$$

where c is a scalar, a is an n-vector, U is unit upper triangular, $D = \mathrm{diag}(d_1,\dots,d_n)$ and $n = \dim P$.

If P is PD (positive definite) then the factors U and D can be calculated by:

For $j = n, n-1, \dots, 2$ evaluate equation (A.2) through equation (A.5) recursively

$$d_j = d_j + c_j a_j^2 \tag{A.2}$$

$$a_k := a_k - a_j U_{kj} \qquad k = 1,\dots,j-1 \tag{A.3}$$

$$U_{kj} = U_{kj} + \frac{c_j a_j a_k}{d_j} \qquad k = 1,\dots,j-1 \tag{A.4}$$

$$c_{j-1} = \frac{c_j d_j}{d_j} \tag{A.5}$$

and then

$$d_1 = d_1 + c_1 a_1^2 \tag{A.6}$$

PROOF:

Consider the associated quadratic form $x^t P x$

$$x^t P x = \Sigma_1^n\, d_j v_j^2 + c(a^t x)^2 \tag{A.7}$$

where $v = U^t x$

$$\begin{aligned} x^t P x = {} & \Sigma_1^{n-1}\, d_j v_j + (d_n + c a_n^2) x_n^2 + \\ & 2x_n \Sigma_1^{n-1} (d_n U_{jn} + c a_n a_j) x_j + \\ & d_n [\Sigma_1^{n-1} U_{jn} x_j]^2 + c[\Sigma_n^{-1}\, a_j x_j]^2 \end{aligned} \tag{A.8}$$

Let

$$w^t = [\, U_{1n}, \; . \; . \; . \;, U_{n-1\,n} \,]$$

$$a^t = [\ a_1,\ .\ .\ .\ ,\ a_{n-1}\ ] \qquad (A.9)$$

$$x^t = [\ x_1,\ .\ .\ .\ ,\ x_{n-1}\ ]$$

$$d_n = d_n + ca_n^2$$

Then equation (A.8) can be written as

$$x^tPx = \Sigma_n^{-1}d_jv_j^2 + d_n[x_n+(d_nw+ca_na)x/d_n]^2 + d_n[w^tx]^2 + c[a^tx]^2 - [(d_nw-ca_na^t)^tx]^2/d_n \qquad (A.10)$$

By setting

$$y_n = x_n + \frac{1}{d_n}(d_nw + ca_na)^tx \qquad (A.11)$$

and combine the quadratic expression from $d_n[w^tx]^2 + c[a^tx]^2$ we get

$$x^tPx = d_ny_n^2 + \Sigma_1^{n-1}d_jv_j^2 + \frac{cd_n}{d_n}[(a-a_nw)^tx]^2 \qquad (A.12)$$

Note that the bracketed term in equation (A.12) is of the same form as equation (A.7) except only (n-1) variables are involved. Thus the inductive reduction that follows

$$x^tPx = d_ny_n^2 + d_{n-1}y_{n-1}^2 + \Sigma_1^{n-2}d_jv_j^2 + \frac{cd_n}{d_n}[(a'-a_nw)^tx]^2$$

$$= d_ny_n^2 + d_{n-1}y_{n-1}^2+d_{n-2}y_{n-2}^2 + \Sigma_1^{n-3}d_jv_j^2 + \frac{cd_n}{d_n}[(a''-a_nw)^tx]^2 \qquad (A.13)$$

where each successive a, w and x is one element shorter and after n-1 steps

$$x^tPx = \Sigma_2^nd_jy_j^2 + d_1y_1^2 + \frac{cd_n}{d_n}[(a_1-a_1)^tx_1]^2 \qquad (A.14)$$

but $a_1 - a_1 = 0$ therefore

$$x^tPx = \Sigma_2^nd_jy_j^2 + d_1y_1^2 \qquad (A.15)$$

Now since $y = U^t x$ where U is unit upper triangular

$U_1{}^t = [\ 1\ 0\ 0\ 0\ .\ .\ .\ 0]$ (dim $U_1 = n$)

therefore

$y_1 = x_1$

and

$$x^t P x = \Sigma_2^n\ d_j y_j^2 + d_1 x_1^2$$

$$= \Sigma_1^n\ d_j y_j^2 \qquad (A.16)$$

From the definition of $y=U^t x$ and D

$$x^t P x = x^t\ U\ D\ U^t\ x \qquad (A.17)$$

11. Appendix B

```
C
C THIS PROGRAM AND ITS SUBROUTINES WILL IMPLEMENT
SELF-TUNING
C CONTROL ON THE GIVEN PROCESS. ESTIMATION OF PARAMETERS
WILL
C BE DONE BY 1 OF 5 METHODS SELECTED BY THE USER.
SELF-TUNING
C ALGORITHM USED FROM CLARKE AND GAWTHROP 1979. SUBROUTINE
C TO CONVERT A CONTINUOUS TRANSFER FUNCTION INTO DISCRETE
FORM
C WAS TAKEN FROM THE UNIVERSITY OF NEWCASTLE UPON TYNE
PACKAGE.
C
C HEATHER J. W. REINHOLT SEPTEMBER 1, 1983
DIMENSION P(2,465),G(2,30),C(5),T(3),U(30),Y(30),V(30)
DIMENSION A(2,30),X(10,30),PHI(30),AA(20),B(20),RHO(2,3)
DIMENSION PN(3),PD(3),QN(3),RN(3),PDINV(10),RSTPT(3000)
DIMENSION QD(3),RD(3)
INTEGER STOP,ORDER,SET,PTEST,WAVE,FLAG1,STOP1
DATA P/930*0.0/,G/60*0.0/,C/5*0.0/,RHO/6*0.995/
DATA
A/60*0.0/,X/300*0.0/,PHI/30*0.0/,Y/30*0.0/,QD/1.0,2*0.0/
DATA AA/20 * 0.0/, B/20 * 0.0/,U/30*0.0/,V/30*0.0/
DATA PN/3*1.0/,PD/3*1.0/,QN/3*0.0/,RN/3*1.0/,PDINV/3*1.0/
DATA
NPN/1/,NPD/1/,NQD/1/,NQN/1/,NRN/1/,RSTPT/1000*0.0,2000*5.0/
```

```
DATA QD/3*1.0/, RD/3*1.0/, NRD/1/
C
C GET THE Z TRANSFORM OF THE SYSTEM
C
WRITE(6,70)
READ(5,71) IANS1
WRITE(6,74)
READ(5,71) IANS2
IF (IANS2 .EQ. 2) GOTO 77
CALL WEIGHT(PN,PD,PDINV,QN,RN,NPN,NPD,NQN,NRN,NF,
1 NRD,NQD,QD,RD)
77 IF (IANS1 .EQ. 2) GOTO 78
WRITE(6,72)
READ(2,73) AA(2),AA(3),AA(4),AA(5),B(1),B(2),B(3),B(4),B(5)
AA(1) = 1.0
IF (IANS1 .EQ. 1) GOTO 76
78 WRITE(6,1)
READ(7,2) ITYPE,TS,GAIN,T(1),T(2),T(3)
CALL ZTRAN(ITYPE,TS,GAIN,T,AA,B,NA,NB)
76 WRITE(6,88) AA(1),AA(2),AA(3),AA(4),AA(5),B(1),B(2),B(3),
1 B(4),B(5)
WRITE(6,6)
READ(8,7) (C(I),I=1,5)
WRITE(6,5)
C
C OBTAIN SYSTEM INFORMATION
C
```

```
C NF NUMBER OF F PARAMETERS
C NG NUMBER OF G PARAMETERS
C NH NUMBER OF H PARAMETERS
C NL NUMBER OF L PARAMETERS
C METHOD IDENTIFICATION METHOD
C STOP NUMBER OF SAMPLES TO BE TAKEN
C STPT VALUE OF SETPOINT
C VAR VARIANCE OF THE NOISE TO BE ADDED
C SLT MAGNITUDE OF THE NUMBERS GENERATED FOR NOISE
C PINIT INITIAL COVARIANCE MATRIX VALUE
C ITD TIME DELAY OF THE SYSTEM
C ORDER ORDER OF THE SYSTEM
C EMEAN MEAN OF THE NOISE
C G0 INTIAL G0 VALUE
C PTEST ITERATION TO PRINT COVARIANCE MATRIX
C
WRITE(6,3)
READ(9,4)NF,NG,NH,NL,METHOD,STOP,STPT,VAR,SLT,
1 PINIT,ITD,ORDER
1 ,EMEAN,G0,PTEST,RHOI,DIST
N = NF + NG + NH + NL
A(1,NF+1) = G0
DO 60 I=1,3
RHO(1,I) = RHOI
RHO(2,I) = RHO1
60 CONTINUE
WRITE(6,9)
```

```
READ(5,16) SET
C
C
C INITIALIZE SET POINT TRACKING
C
IF(SET .NE. 3) GOTO 17
C
C SAWTOOTH FUNCTION SET POINT
C
C STPT - - -
C / \ / \ / \
C 0/ \/ \/ \
C 0 100
C
K = 1
RSTPT(1) = 0.10
DO 11 I=1,10
IF(I .GT. 1) K = K + 1
IF(I .GT. 1) RSTPT(K)=RSTPT(K-1) + 1.0/10.
DO 12 J = 2, 50
K = K + 1
RSTPT(K) = RSTPT(K-1) + 1.0/10.
12 CONTINUE
DO 13 J = 1, 50
K = K + 1
RSTPT(K) = RSTPT(K-1) - 1.0/10.
13 CONTINUE
```

```
11 CONTINUE
C
17 IF(SET .NE. 2) GOTO 18
C
C TRIANGLE SETPOINT
C
WAVE = STOP / 200
STOP1 = 200
K = 2
FLAG1 = 0
DO 55 I=1,WAVE
RSTPT(K-1) = 0.00
IF(I .NE. 1) STOP1 = 201
IF ( I .EQ. 1) K = 1
IF(I .EQ. 1) RSTPT(1) = 0.025
DO 21 J=2,STOP1
IF (FLAG1 .NE. 1) K = K + 1
FLAG1 = 0
RSTPT(K) = RSTPT(K-1) + 0.025
21 CONTINUE
K = K + 1
FLAG1 = 1
55 CONTINUE
DO 28 I=200,STOP,200
RSTPT(I) = 5.0
28 CONTINUE
18 IF(SET .NE. 1) GOTO 63
```

```
C
C SQUARE WAVE FUNCTION SETPOINT
C
C STPT ----- ----- -----
C | | | | | |
C 0----+ +------+ +------+ +----
C 0 50 100
K=1
INT = STOP / 50.
DO 40 I=1,INT
DO 41 J=1,25
RSTPT(K) = STPT
K=K+1
41 CONTINUE
DO 42 J=1,25
RSTPT(K) = 0.0
K=K+1
42 CONTINUE
40 CONTINUE
C
C CONSTANT SETPOINT
C
63 IF(SET .NE. 4) GOTO 19
DO 61 K=1,STOP
RSTPT(K) = STPT
61 CONTINUE
C INITIALIZE THE COVARIANCE MATRIX
```

```
C
C COVARIANCE MATRIX FOR RLS AND RSR
C
C _ _
C | 1 0 0 0 0 0 |
C | 2 3 0 0 0 0 |
C P = | 4 5 6 0 0 0 |
C | 7 8 9 10 0 0 |
C |11 12 13 14 15 0 |
C |16 17 18 19 20 21 |
C -- --
C
19 J = 0
IF (METHOD .EQ. 3) GOTO 10
C IF RSR TAKE SQUARE ROOT OF THE COVARIANCE MATRIX
IF (METHOD.NE.2) GOTO 22
PINIT = SQRT(PINIT)
RHO(1,1) = SQRT(RHO(1,1))
RHO(1,2) = SQRT(RHO(1,2))
RHO(1,3) = SQRT(RHO(1,3))
22 DO 15 I=1,N
J = J + I
P(1,J) = PINIT
15 CONTINUE
GO TO 20
C
C COVARIANCE MATRIX FOR UD FACTORIZATION
```

```
C
C  _ _
C | 1 2 3 4 5 6 |
C | 0 7 8 9 10 11 |
C P = | 0 0 12 13 14 15 |
C | 0 0 0 16 17 18 |
C | 0 0 0 0 19 20 |
C | 0 0 0 0 0 21 |
C -- --
C
10 INC = 1
NK=N
DO 14 L = 1, N
P(1,INC) = PINIT
INC = NK +1
NK = INC + N - L - 1
14 CONTINUE
20 DO 100 KK = 1,STOP
C
C OBTAIN PROCESS OUTPUT
C
CALL DATA(AA,B,NF,NG,NH,NL,Y,[illegible]NEW,ITD,ORDER,C,VAR,SLT,
1 KK,RHO,DIST)
C
C START CONTROL
C
CALL STC(N,NF,NG,NH,NL,P,X,A,V,U,Y,ITD,PN,PD,QN,RN,PDINV,
1 NRN,NQD,NQN,NPD,NPN,KK,YNEW,VAR,METHOD,RSTPT,UD,PHI,RHO,
2 PERER,QD)
C
IF( METHOD.NE.5)WRITE(4,25) KK, (A(1,I),I=1,N)
WRITE(3,30) KK,YNEW,U(1),RSTPT(KK),V(1)
IF((KK.EQ.PTEST).AND.(METHOD.EQ.3))                WRITE(6,8)
(P(1,J),J=1,36)
IF((KK.EQ.PTEST).AND.(METHOD.LE.2))               WRITE(6,66)
(P(1,J),J=1,36)
100 CONTINUE
C
C IF(METHOD.EQ.3)WRITE(6,8) (P(1,J),J=1,36)
WRITE(6,26) (A(1,I),I=1,N)
C
1 FORMAT('ENTER TRANSFER FUNCTION TYPE',/,
1 'TYPE 1 - 1./(TS + 1)',/,
2 'TYPE 2 - 1./S * (S + 1)',/,
3 'TYPE 3 - 1./(ST1 + 1)(ST2 + 1)',/,
4 'TYPE 4 - 1./(TS + 1)(TS + 1)',/,
5 'TYPE 5 - 1./(T1*S*S + T2*S + 1)',/,
6 'ENTER SAMPLE TIME, GAIN, TIME CONSTANTS',/,
7 'ENTER AS TYPE,TS,GAIN,T(1),T(2),T(3)')
2 FORMAT(I2,5F7.2)
3   FORMAT('ENTER   NUMBER   OF   F   PARAMETERS   TO   BE
ESTIMATED',/,6X,
1 ' NUMBER OF G PARAMETERS TO BE ESTIMATED',/,6X,
2 ' NUMBER OF H PARAMETERS TO BE ESTIMATED',/,6X,
2 ' NUMBER OF L PARAMETERS TO BE ESTIMATED',/,6X,
3 ' ESTIMATION METHOD',/,6X,
4 ' NUMBER OF OBSERVATIONS (MULTIPLE O[illegible]0)',/,6X,
5 ' SET POINT',/,6X,
6 ' VARIANCE OF THE DISTURBANCE',/,6X,
7 ' MAGNITUDE OF THE NUMBERS GEN FOR NOISE',/,6X,
8 ' INITIAL VALUE FOR COVARIANCE MATRIX DIAGONAL',/,6X,
9 ' TIME DELAY OF THE SYSTEM',/,6X,
1 ' ORDER OF THE SYSTEM',/,6X,
2 ' MEAN OF THE NOISE',/,6X,
3 ' INITIAL VALUE OF G0')
4 FORMAT(6I5,4F15.5,2I2,2F5.2,I5,2F10.5)
5 FORMAT(' ESTIMATION  METHODS:',/,' 1  =  RECURSIVE  LEAST
SQUARES',/,
1 ' 2 = RECURSIVE SQUARE ROOT METHOD',/,
2 ' 3 = U-D FACTORIZATION',/,' 4 = RECURSIVE LEARNING',/,
3 ' 5 = RECURSIVE MAXIMUM LIKELIHOOD')
6 FORMAT('ENTER THE TRUE NOISE PARAMETERS, C PARAMETERS',/,
1 ' C(1), C(2), C(3), C(4), C(5)')
7 FORMAT(5F10.5)
8 FORMAT(8F8.2,/,8X,7F8.2,/,16X,6F8.2,/,
1 24X,5F8.2,/,32X,4F8.2,/,
1 40X,3F8.2,/,48X,2F8.2,/,56X,F8.2)
9 FORMAT(' ENTER TYPE OF SETPOINT',/,
1 ' 1. SQUARE WAVE',/,
2 ' 2. RAMP ',/,
3 ' 3. SAWTOOTH (1000 ITER), CONSTANT (2000 ITER)',/,
4 ' 4. CONSTANT ')
66 FORMAT(F8.2,/,2F8.2,/,3F8.2,/,4F8.2,/,5F8.2,/,6F8.2,/,
1 7F8.2,/,8F8.2)
16 FORMAT(I1)
25 FORMAT(I5,15F10.4)
26 FORMAT(15F8.4)
30 FORMAT(I5,8F10.5)
70 FORMAT(' DO YOU WISH TO ENTER A AND B POLYNOMIALS?',/,
1 ' YES = 1 NO = 2')
71 FORMAT(I1)
72 FORMAT(' ENTER A2 - A5 AND B1 - B5 A1 = 1.0')
73 FORMAT(9F10.4)
74 FORMAT(' DO YOU WANT P, Q, OR R WEIGHTING?',/,
1 ' IF NO, P = R = 1.0 Q = 0',/,
2 ' YES = 1 NO = 2')
88 FORMAT(1X,10F6.3)
STOP
END
C
C
SUBROUTINE
DATA(AA,B,NF,NG,NH,NL,Y,U,V,YNEW,ITD,ORDER,C,VAR,SLT
1 ,KK,RHO,DIST)
DIMENSION
AA(20),B(20),Y(30),U(30),V(30),C(5),DIS(5),RHO(2,3)
DATA DIS/5*5.0/
C
```

```
C +-----------+
C | |
C | DATA GENERATION SUBROUTINE |
C | |
C +-----------+
C
INTEGER ORDER
V(1) = RNOISE (VAR,SLT,EMEAN)
YNEW = 0.0
DO 10 I= 1, ORDER
YNEW = YNEW - Y(I+1)*AA(I+1) + U(I+ITD-1)*B(I) + V(I)*C(I)
10 CONTINUE
C IF((KK.GT.700).AND.(KK.LT.1200))
C 1 DISTUR=.1*DIS(1)+.0905*DIS(2)+.0082*DIS(3)
C IF((KK.GT.700).AND.(KK.LT.1200))YNEW=YNEW+DISTUR
C IF((KK.LT.700).OR.(KK.GT.1200)) GOTO 88
C DIS(3)=DIS(2)
C DIS(2)=DIS(1)
C DISTUR=DIS(1)
88 RETURN
END
C
C
FUNCTION RNOISE(VAR,SLT,EMEAN)
DIMENSION IR(12,2)
C
C +---------+
C | |
C | NOISE FUNCTION |
C | |
C +---------+
C
DATA IR/24*0/
DATA IR(11,1),IR(11,2)/2*1/
RNOISE = 0.0
IF(VAR.EQ.0.0) GOTO 40
SUM = 0.0
DO 45 I=1,12
CALL RAND(RESULT,IR)
SUM = SUM + RESULT
45 CONTINUE
RNOISE = (SUM -6.0) * SQRT(VAR)/1.62 + EMEAN
RETURN
40 CONTINUE
CALL RAND(RESULT,IR)
RNOISE = (RESULT-0.5) * 2.0 * SLT + EMEAN
RETURN
END
C
C
SUBROUTINE RAND(RESULT,IR)
DIMENSION IR(12,2)
C
C +----------+
C | |
C | RANDOM NUMBER GENERATOR |
C | |
C +----------+
C
INEW = IR(12,1)+IR(6,1)+IR(4,1)+IR(1,1)+1
ISUM = 1
IF((INEW/2)*2.EQ.INEW) ISUM = 0
J = 12
INEW = 0
DO 10 I=1,11
IR(J,1) = IR(J-1,1)
INEW = INEW + IR(J,1)*2**(J-1)
J = 12 - I
10 CONTINUE
IR(1,1) = ISUM
INEW = INEW + ISUM
RESULT = FLOAT(INEW)/4096.
RETURN
END
C
C
SUBROUTINE STC(N,NF,NG,NH,NL,P,X,A,V,U,Y,ITD,PN,
1 PD,QN,RN,PDINV,
1 NRN,NQD,NQN,NPD,NPN,KK,YNEW,VAR,METHOD,
2 RSTPT,UD,PHI,RHO,
2 PERER,QD)
```

```
DIMENSION X(10,30),A(2,30),RHO(2,3),PHI(30),V(30),U(30)
DIMENSION
P(2,465),Y(30),PHIOUT(3),XX(2,30),G(2,30),UD(15,15)
C
C +-+
C | |
C | SELF TUNING CONTROL SUBROUTINE |
C | |
C +-+
C
DIMENSION PN(3),PD(3),QN(3),RN(3),PDINV(10),RSTPT(3000)
DIMENSION PREV(3),QD(3)
DATA PHIOUT/3*0.0/,COEFF/0.0/,ACLOSS/0.0/,PREV/3*0.0/
DATA XX/60*0.0/
C
C CCALCULATE THE PRESENT AUXILIARY OUTPUT PHIY(T) = PY(T)
C
Y(1) = YNEW
PHIOUT(1) = 0.0
DO 10 I=1,NPN
PHIOUT(1) = PHIOUT(1) + PN(I)*Y(I)
10 CONTINUE
C
DO 15 I=2,NPD
PHIOUT(1) = PHIOUT(1) - PD(I)*PHIOUT(I)
15 CONTINUE
PHIOUT(1) = PHIOUT(1)/PD(1)
```

```
C
C BUILD THE OBSERVATION MATRIX
C PROCESS OUTPUTS (NF = ORDER OF A POLYNOMIAL + ORDER OF  PD
POLYNOMIAL
C
DO 20 I=1,NF
X(1,I) = Y(I) * PDINV(I)
20 CONTINUE
C
C PROCESS INPUTS
C
IF (NG.LT.2) GOTO 26
DO 25 I=2,NG
X(1,NF+I) = U(I-1)
25 CONTINUE
26 CONTINUE
C
C AUXILIARY OUTPUTS
C
IF (NH.EQ.0) GOTO 31
DO 30 I=1,NH
X(1,NF+NG+I) = PHI(I+1)
30 CONTINUE
31 CONTINUE
C
DO 32 I=1,NL
X(1,NF+NG+NH+I) = 1.0
```

```
32 CONTINUE
IS = 1
DO 35 I=1,N
XX(IS,I) = X(ITD+1,I)
35 CONTINUE
C
C
C CALL THE IDENTIFICATION ROUTINE TO GET TH[illegible]ED'
C
IF(METHOD.EQ.1) CALL IDENT1(YNEW,N,IS,PERER,[illegible]P,G,RHO)
IF(METHOD.EQ.2) CALL IDENT2(YNEW,N,IS,PERER,XX,A,P,G,RHO)
IF(METHOD.EQ.3) CALL IDENT3(YNEW,N,IS,PERER,XX,A,P,G,RHO,
1 KK,VAR)
IF(METHOD.EQ.4) CALL IDENT4(YNEW,N,IS,PERER,XX,A,P,G,RHO)
IF(METHOD.EQ.5)                                        CALL
IDENT5(YNEW,N,NF,NG,NH,KK,XX,A,P,G,RHO,IS)
91 CONTINUE
38 IF (METHOD .NE. 5 ) GOTO 51
DO 54 J=1, NF
A(IS,J) = (-1.0) * A(IS,J)
54 CONTINUE
C WRITE(6,88) (A(1,I),I=1,N)
WRITE(4,81)KK,(A(1,J),J=1,N)
81 FORMAT(I5,15F8.4)
88 FORMAT(1X,8F15.5)
C
C CALCULATE NEW PREDICTION FOR PHI(T+K/T) = A(T) * X(T)
```

```
C
51 NF1 = NF + 1
PHIY = 0.0
DO 40 I = 1,N
IF(I.EQ.NF1) GOTO 40
PHIY = PHIY + X(1,I)*A(1,I)
40 CONTINUE
C
C CALCULATE CONTROL ACTION
C
RW = 0
DO 45 [illegible]NRN
RW = RW + RN(I) * RSTPT(KK)
45 CONTINUE
C
PREV(1) = RW - PHIY
C
C ADD EFFECT OF Q COSTING
C
C CONT=(PREV(1)*QD(1)+PREV(2)*QD(2)+PREV(3)*QD(3)
C 1 - U(1)*(A(1,NF1)*QD(2)+QN(2)) - U(2)*(A(1,NF1)*
C 2 QD(3)+QN(2))) / (A(1,NF1)*QD(1) + QN(1))
CONT = PREV(1) / (A(1,NF1)+QN(1))
DO 8 II=1,2
I = 4 - II
PREV(I)=PREV(I-1)
8 CONTINUE
```

```
APHI = PHIOUT(1) - RW + CONT
DO 52 II=1,29
I = 31 - II
U(I) = U(I-1)
52 CONTINUE
C
PHI(1) = (PHIY + CONT * A(1,NF1))
ACLOSS = ACLOSS + ABS(YNEW-PHI(1))
RLIM = 50.
UTOP = U(1) + RLIM
UBOT = U(1) - RLIM
IF (CONT .GT. UTOP) CONT = UTOP
IF (CONT .LT. UBOT) CONT = UBOT
IF (CONT .GT. 25.0) CONT = 25.
IF (CONT .LT. -25.) CONT = -25.
X(1,NF1) = CONT
COEFF = COEFF + ABS(RSTPT(KK)*1.2 - CONT)
U(1) = CONT
C
WRITE(1,77) COEFF,ACLOSS,YNEW,PHI(1)
77 FORMAT(1X,4F15.2)
DO 55 I=1,2
II = 4-I
PHIOUT(II) = PHIOUT(II-1)
55 CONTINUE
C
DO 60 I=1, ITD
DO 65 J=1, N
X(ITD+2-I,J) = X(ITD+1-I,J)
65 CONTINUE
60 CONTINUE
C
C UPDATE THE VECTORS
C
DO 50 II = 1,29
I = 31 - II
V(I) = V(I-1)
PHI(I) = PHI(I-1)
Y(I) = Y(I-1)
50 CONTINUE
IF (METHOD .NE. 5) GOTO 41
DO 42 J=1, NF
A(IS,J) = (-1.0) * A(IS,J)
42 CONTINUE
41 RETURN
END
C
C
SUBROUTINE ZTRAN(ITYPE,TS,GAIN,T,AA,B,NA,NB)
DIMENSION T(3), AA(20), B(20)
C
C +------------+
C | |
C | Z TRANSFORMATION OF TRANSFER FUNCTION |
C | |
C +----------+
C
C
NA = 1
NB = 1
AA(1) = 1.0
B(1) = 0.0
IF(ITYPE.LE.0.OR.ITYPE.GT.5) GOTO 40
GOTO (5, 10, 15, 20, 25), ITYPE
C
C
C FIRST ORDER - 1. / (T*S   10
C
5 NB = 1
NA = 2
B(1) = GAIN * (1.-EXP(-TS/T(1)))
AA(2) = - EXP(-TS/T(1))
GOTO 40
C
C SECOND ORDER - 1. / S * (S + 1)
C
10 NB = 2
NA = 3
B(1) = GAIN*(TS-T(1)*(1.-EXP(-TS/T(1))))
B(2) = GAIN*(-EXP(-TS/T(1))*TS+T(1)*(1.-EXP(-TS/T(1))))
AA(2) = -(1.0 + EXP(-TS/T(1)))
AA(3) = -EXP(-TS/T(1))
GOTO 40
C
C SECOND ORDER - 1. / (S*T1 + 1)(S*T2 + 1)
C
15 NB = 2
NA = 3
AAA = EXP(-TS/T(1))
BB = EXP(-TS/T(2))
B(1)
=GAIN*((BB/T(2)-AAA/T(1))/(1./T(2)-1./T(1))-(AAA+BB)+1.)
B(2) = GAIN*(AAA*BB-(BB/T(2)-AAA/T(1))/(1./T(2)-1./T(1)))
AA(2) = -(BB + AAA)
AA(3) = AAA * BB
GOTO 40
C
C SECOND ORDER - 1. / (S*T + 1) (S*T + 1)
C
20 NB = 2
NA = 3
AAA = EXP(-TS/T(1))
B(1) = GAIN * (-(1.+TS/T(1)) * AAA +1.0)
B(2) = GAIN * (AAA*AAA-(1.-TS/T(1))*AAA)
AA(2) = -2. * AAA
AA(3) = AAA * AAA
GOTO 40
C
```

```
C SECOND ORDER - 1. / (T1 * S*S + T2 * S + 1)
C
25 AAA = T(2)/(2.*T(1))
BB = 1./T(1)-(T(2)*T(2))/(T(1)*2.)**2
IF (BB .LT. 0.0) GOTO 30
BB = SQRT(BB)
B(1) =EXP(-AAA*TS)*(COS(BB*TS)-AAA/BB*SIN(BB*TS))+1.0
B(2)=EXP(-2.*AAA*TS)-EXP(-AAA*TS)*(COS(BB*TS)-AAA/BB*
1 SIN(BB*TS))
AA(1) =1.0
AA(2) = -2.*EXP(-AAA*TS) + COS(BB*TS)
AA(3) = EXP(-2.*AAA*TS)
B(1) = GAIN*B(1)/(T(1)*(BB*BB+AAA*AAA))
B(2) = GAIN*B(2)/(T(1)*(BB*BB+AAA*AAA))
NB = 2
NA = 3
GOTO 40
30 WRITE(6,3)
3 FORMAT(' NEGATIVE BB VALUE, UNABLE TO TAKE SQUARE ROOT')
40 RETURN
END
SUBROUTINE IDENT1(YNEW,N,IS,PERER,XX,A,P,G,RHO)
DIMENSION XX(2,30),A(2,30),P(2,465),G(2,30),RHO(2,3)
C
C +-----------+
C | |
C | RECURSIVE LEAST SQUARES |
C | |
C +-----------+
C
PERER = YNEW
XPX = 0.0
C
DO 15 I=1, N
PERER = PERER - XX(IS,I) * A(IS,I)
G(IS,I) = 0.0
C
DO 10 J=1, N
IJ = J * (J - 1) / 2 + I
IF (J .LT. I) IJ = I * (I - 1) / 2 + J
G(IS,I) = G(IS,I) + XX(IS,J) * P(IS,IJ)
10 CONTINUE
C
XPX = XPX + XX(IS,I) * G(IS,I)
15 CONTINUE
C
XPX = XPX + RHO(IS,1)
IJ = 0
C
DO 20 I=1, N
DO 20 J=1,I
IJ = IJ + 1
P(IS,IJ) = (P(IS,IJ) - G(IS,I)*G(IS,J)/XPX) / RHO(IS,2)
20 CONTINUE
```

```
C
DO 25 I=1, N
G(IS,I) = G(IS,I) / XPX
A(IS,I) = A(IS,I) + G(IS,I) * PERER
25 CONTINUE
C
RETURN
END
C
C
SUBROUTINE IDENT2(YNEW,N,IS,PERER,XX,A,P,G,RHO)
DIMENSION XX(2,30),A(2,30),P(2,465),G(2,30),RHO(2,3)
C
C +---------+
C | |
C | RECURSIVE SQUARE ROOT METHOD |
C | |
C +---------+
C
PERER = YNEW
C
DO 10 I=1, N
PERER = PERER - XX(IS,I) * A(IS,I)
10 CONTINUE
C
GAMMA = RHO(IS,1)
GAMMA2 = RHO(IS,1) * RHO(IS,1)
```

```
IJ = 0
JI = 0
C
DO 30 J=1, N
PX = 0.0
J1 = J - 1
C
DO 15 I=1, J
JI = JI + 1
PX = PX + P(IS,JI) * XX(IS,I)
15 CONTINUE
C
ALPHA = GAMMA / RHO(IS,2)
BETA = PX / GAMMA2
GAMMA2 = GAMMA2 + PX * PX
GAMMA = SQRT(GAMMA2)
ALPHA = ALPHA / GAMMA
G(IS,I) = P(IS,JI) * PX
P(IS,JI) = ALPHA * P(IS,JI)
IF (J1 .EQ. 0) GO TO 25
C
DO 20 I=1, J1
IJ = IJ + 1
PQP = P(IS,IJ)
P(IS,IJ) = ALPHA * (PQP - BETA*G(IS,I))
G(IS,I) = G(IS,I) + PQP * PX
20 CONTINUE
```

```
C
25 CONTINUE
IJ = IJ + 1
30 CONTINUE
C
DO 35 I=1, N
G(IS,I) = G(IS,I) / GAMMA2
A(IS,I) = A(IS,I) + G(IS,I) * PERER
35 CONTINUE
C
RETURN
END
C
C
SUBROUTINE IDENT3(YNEW,N,IS,PERER,XX,A,P,G,RHO,UD,KK,VAR)
DIMENSION
A(2,30),XX(2,30),P(2,465),G(2,30),RHO(2,3),UD(15,15)
C
C +---+
C | |
C | U-D FACTORIZATION |
C | |
C +---+
C
DIMENSION V(30)
C
PERER = YNEW
```

```
C
C
C
C INITIALIZATION
C
CC
DO 15 L=2,N
J = N - L + 2
PERER = PERER - A(IS,J) * XX(IS,J)
JM1 = J - 1
IJ = J
DO 10 K = 1, JM1
XX(IS,J) = XX(IS,J) + P(IS,IJ) * XX(IS,K)
IJ = IJ + N - K
10 CONTINUE
V(J) = P(IS,IJ) * XX(IS,J)
15 CONTINUE
PERER = PERER - A(IS,1) * XX(IS,1)
V(1) = P(IS,1) * XX(IS,1)
ALPHA = RHO(IS,1)/RHO(IS,2) + V(1) * XX(IS,1)
GAMMA = 1. / ALPHA
P(IS,1) = P(IS,1)*GAMMA*RHO(IS,1)/(RHO(IS,2)*RHO(IS,2))
G(IS,1) = V(1)
IJ = N + 1
DO 20 J = 2, N
BETA = ALPHA
ALPHA = ALPHA + V(J) * XX(IS,J)
```

```
PQ = -XX(IS,J) * GAMMA
GAMMA = 1. / ALPHA
P(IS,IJ) = P(IS,IJ)*BETA*GAMMA/RHO(IS,2)
IJ=IJ + N -J + 1
G(IS,J) = V(J)
JM1 = J-1
JI = J
DO 20 I=1,JM1
BETA=P(IS,JI)
P(IS,JI)=BETA+G(IS,I)*PQ
JI=JI+N-I
G(IS,I) = G(IS,I) + V(J) * BETA
20 CONTINUE
DO 40 J=1,N
G(IS,J) = G(IS,J) /ALPHA
A(IS,J) = A(IS,J) + G(IS,J) * PERER
40 CONTINUE
RETURN
END
C
C
SUBROUTINE IDENT4(YNEW,N,IS,PERER,XX,A,P,G,RHO)
DIMENSION
XX(2,30),A(2,30),P(2,465),G(2,30),RHO(2,3),IFP(2,3)
DATA IFP/6*0/
C
C +--+
C | |
C | RECURSIVE LEARNING METHOD |
C | |
C +-+
C
XXX = 0.0
PERER = YNEW
C
DO 10 I=1,N
PERER = PERER - XX(IS,I) * A(IS,I)
XXX = XXX + XX(IS,I) * XX(IS,I)
10 CONTINUE
C
XPX = XXX + RHO(IS,1)
C
DO 15 I=1, N
G(IS,I) = XX(IS,I) / XPX
15 CONTINUE
C
IF1 = IFP(IS,1)
IF (IF1 .GT. 0) G(IS,IF1) = 0.0
IF1 = IFP(IS,2)
IF (IF1 .GT. 0) G(IS,IF1) = 0.0
C
DO 20 I=1, N
A(IS,I) = A(IS,I) + G(IS,I) * PERER
20 CONTINUE
```

```
C
RETURN
END
C
C
SUBROUTINE WEIGHT(PN,PD,PDINV,QN,RN,NPN,NPD,NQN,NRN,NF,
1 NRD,NQD,QD,RD)
DIMENSION PN(3),PD(3),QN(3),RN(3),PDINV(10),QD(3),RD(3)
C
C +------------+
C | |
C | P, Q, AND R WEIGHTING ENTRY SUBROUTINE |
C | |
C +------------+
C
WRITE(6,10)
READ(13,11) NPN,NPD,NQN,NQD,NRN,NRD
WRITE(6,12)
READ(13,13)(PN(I),I=1,3),(PD(J),J=1,3)
WRITE(6,14)
READ(13,13)(QN(I),I=1,3),(QD(J),J=1,3)
WRITE(6,15)
READ(13,13)(RN(I),I=1,3),(RD(I),I=1,3)
CALL INVERS(PD,PDINV,NPD,NF)
C
10 FORMAT(' ENTER NO OF TERMS IN PN, PD, QN, QD, RN, RD')
11 FORMAT(6I2)
```

```
12 FORMAT(' ENTER PN1 - PN3 AND PD1 - PD3')
13 FORMAT(6F8.4)
14 FORMAT(' ENTER QN1 - QN3 AND QD1 - QD3')
15 FORMAT(' ENTER RN1 -RN3 AND RD1 - RD3')
C
RETURN
END
C
C
SUBROUTINE INVERS(PD,PDINV,NPD,NF)
DIMENSION PD(5),PDINV(10),WKN(10),WKD(10)
C
C +-----------+
C | |
C | PD INVERSE CALCULATION |
C | |
C +-----------+
C
DO 3 I=1,10
WKN(I) = 0.0
WKD(I) = 0.0
3 CONTINUE
C
WKN(1) = 1.0
C
DO 10 I=1,NPD
WKD(I) = PD(I)
```

```
10 CONTINUE
C
DO 30 J=1,NF
PDINV(J) = WKN(1) / WKD(1)
DO 20 I=1,9
WKN(I) = WKN(I+1) - WKD(I+1) * PDINV(J)
20 CONTINUE
WKN(10) = 0.0
30 CONTINUE
RETURN
END
C
SUBROUTINE IDENT5(YNEW[illegible]NF,NG,NH,KK,XX,A,P,G,RHO,IS)
DIMENSION XX(2,30),A(2,30),P(2,465),G(2,30),RHO(2,3)
C
C +------+
C |  |
C | RECURSIVE MAXIMUM LIKELIHOOD METHOD |
C |  |
C +------+
C
DIMENSION GRAD(10),GR(10),V(10),C(10),GAIN(10,10),GRTG(10)
      ION TEMP(10),GGR(10),E(3000),AO(30)
       10*0.0/, N1/35/, N2/10/, IFLAG/0/
   ER ORDER
NDIM = N
C
```

```
IF (KK.GT.1) GOTO 99
WRITE(6,61)
READ(5,51) PP
C
DO 10 I=1,3000
10 E(I) = 0.0
98 DO 15 I=1, N
GR(I) = 0.0
V(I) = 0.0
GRAD(I) = 0.0
GRTG(I) = 0.0
GGR(I) = 0.0
TEMP(I) = 0.0
DO 15 J=1, N
GAIN(I,J) = 0.0
GAIN(I,I) = PP
15 CONTINUE
C
99 ICOUNT = ICOUNT +1
KOUNT = 0
C IF(KK.EQ.N1) CALL CHECK(N1,A,AO,N,GAIN,KK,IFLAG,ICOUNT,N2)
IF((IFLAG.EQ.1) .AND. (ICOUNT.EQ.0)) GOTO 98
NS = NG + NF + 1
C
DO 20 I = NS, NDIM
K = I - NF - NG
C(K) = A(IS,I)
```

```
20 CONTINUE
C
DO 25 I=2, NDIM
GR(I) = GRAD(I-1)
25 CONTINUE

KT = 1
SUM = 0.0
DO 30 I=1, NF
KOUNT = KOUNT + 1
SUM = SUM - C(I) * GRAD(KOUNT)
V(KOUNT) = -1.0 * XX(IS,I)
30 CONTINUE
C
GR(KT) = SUM + XX(IS,1)
KT = KT + NF
SUM = 0.0
C
DO 35 I=1, NG
KOUNT = KOUNT + 1
SUM = SUM - C(I) * GRAD(KOUNT)
ITEMP = NF + I
IF(ITEMP .LT. 1) GOTO 35
V(KOUNT) = XX(IS,ITEMP)
35 CONTINUE
C
GR(KT) = SUM - XX(IS,NF+1)
```

```
KT = KT + NG
SUM = 0.0
C
DO 40 I = 1, NH
KOUNT = KOUNT + 1
SUM = SUM - C(I) * GRAD(KOUNT)
ITEMP = KK-I
IF (ITEMP .LT. 1) GOTO 40
V(KOUNT) = E(ITEMP)
[illegible] CONTINUE

GR(KT) = SUM - E(KK-1)
SUM = 0.0
C
DO 45 J= 1, NDIM
SUM = A(IS,J) * V(J) + SUM
45 CONTINUE
C
E(KK) = YNEW - SUM
DO 5  I = 1,NDIM
GRTG(I) = 0.
GGR(I) = 0.0
DO 50 J=1,NDIM
GRTG(I) = GR(J) * GAIN(I,J) + GRTG(I)
GGR(I) = GAIN(J,I) * GR(J) + GGR(I)
50 CONTINUE
C
```

```
SUM = 0.0
DO 55 I = 1, NDIM
SUM = GR(I) * GRTG(I) + SUM
55 CONTINUE
C
DENOM = 1.0 + SUM
DO 60 I=1,NDIM
DO 60 J=1,NDIM
GAIN(I,J) = GAIN(I,J) - (GGR(I)*GRTG(J)/DENOM)
60 CONTINUE
C
DO 65 I =1,NDIM
TEMP(I) = 0.0
DO 65 J=1,NDIM
TEMP(I) = GAIN(I,J) * GR(J) * E(KK) + TEMP(I)
65 CONTINUE
C
DO 75 J=1,NDIM
GRAD(J) = GR(J)
75 CONTINUE
C IF (KK.EQ.299) WRITE(6,77)((GAIN(I,J),J=1,N),I=1,N)
77 FORMAT(9(1X,9F8.5,/))
IF((IFLAG.EQ.1) .AND. (ICOUNT.LT.N2)) GOTO 83
DO 70 J=1,NDIM
A(IS,J) = A(IS,J) - TEMP(J)
70 CONTINUE
83 IF(KK.NE.5) GOTO 80
```

```
DO 81 I=1,N
AO(I) = A(1,I)
81 CONTINUE
C WRITE(6,88) GRAD(1),GRAD(2) GRAD(3)
88 FORMAT(1X,3F10.5)
C
C
61 FORMAT(' INPUT THE INITIAL VALUE OF THE GAIN')
51 FORMAT(F10.5)
80 RETURN
END
C
C
SUBROUTINE CHECK(N1,A,AO,N,GAIN,KK,IFLAG,ICOUNT,N2)
DIMENSION A(2,30),AO(30),GAIN(10,10),DIF(30)
C
C +----+
C | |
C | "CONVERGENCE" TEST SUBROUTINE |
C | |
C +----+
C
REAL LOSS
VTEST = 1.0005
TEMP = 0.0
DO 10 I=1,N
DIF(I) = A(1,I) - AO(I)
```

```
10 CONTINUE
C
DO 20 I=1,N
DO 20 J=1,N
TEMP = TEMP + DIF(I) * GAIN(I,J) * DIF(J)
20 CONTINUE
C
LOSS = TEMP / KK + 1.0
IFLAG = 1
IF(VTEST .GT. ABS(LOSS)) IFLAG = 0
WRITE(6,40) KK,LOSS,IFLAG,TEMP,A(1,1),AO(1)
DO 30 I=1,N
AO(I) = A(1,I)
30 CONTINUE
C
ICOUNT = 0
N1 = N1 + N2 + 10
40 FORMAT(1X,I5,F10.5,I3,3F10.5)
RETURN
END
```

12. Appendix C

The simulation of nonlinear model of the binary distillation column was run with the following parameters. Any parameter values not listed use the default values given in the program.

M1:

T-SA = 1.000

M2:

T-SA = 1.000

S1:

NG1 = 3 NF = 2 ND = 3

NH = 0 NG2=3

G1 = 0.5000

TYPE OF IDENT = (1=RLS 2=RSR 3=RUD 4=RL)

RHO-1 = 0.995 (FORGETTING FACTOR)

RHO-2 = 0.995

P-INT = 1000.0

S2:

NG1 = 4 NF = 3 ND = 4

NH = 0 NG2=4

G1 = 0.5000

TYPE OF IDENT = (1=RLS 2=RSR 3=RUD 4=RL)

RHO-1 = 0.995 (FORGETTING FACTOR)

RHO-2 = 0.995

P-INT = 1000.0

```
C1:

    TYPE OF CONTROLLER = 1 ST

    -U = 5.000 +U = 30.000 %U = 100.0

    QD(Z) = 1.000

    QN(Z) = 0.000

    PN(Z) = 1.000

    PD(Z) = 1.000

    RN(Z) = 1.000

    RD(Z) = 1.000

C2:

    TYPE OF CONT[illegible]1 ST

    -U = 5.000 +U[illegible].000 %U = 100.0

    QD(Z) = 1.000

    QN(Z) = 0.000

    PN(Z) = 1.000

    PD(Z) = 1.000

    RN(Z) = 1.000

    RD(Z) = 1.000
```