UNIVERSITY OF ALBERTA

Studies on Gain Scheduled Controllers and Gray Predictive PI

BY

Jianmin Chen ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EDMONTON, ALBERTA

Spring, 1999

0-612-40037-9

UNIVERSITY OF ALBERTA

LIBRARY RELEASE FORM

NAME OF AUTHOR:	**Jianmin Chen**
TITLE OF THESIS:	**Studies on** **Gain Scheduled Controllers** **and Gray Predictive PI**
DEGREE:	**Master of Science**
YEAR THIS THESIS GRANTED:	**1999**

Jianmin Chen

5540 – Sherbrooke Street

Vancouver, BC

Canada

November 25, 1998

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 **Studies on Gain Scheduled Controllers and Gray Predictive PI** submitted by Jianmin Chen in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. Max Q-H Meng (Supervisor)

R. E. Rink

Dr. Ray E. Rink

Dr. Sirish Shah

November 19, 1998

ABSTRACT

The practice of process control has demonstrated that gain scheduled PID has many good features such as simple concept. easy design, good control quality, strong robustness and fast adaptive ability. Gray predictive control is attracting attention from the control engineering and theory community because of its special prediction mechanism. A boiler-turbine-generator (BTG) system is the major system in power generation. It is subject to nonlinear, time-varying and disturbance effects. It requires not only good qualities for setpoint tracking and disturbance rejection. but al.'o good stability. This thesis, first, focuses on the study on three types of gain scheduled PID, and gives design suggestions. Next is presented a gray predictive PI controller. and the relevant issues such as consistency, suppression of sensitivity to noise etc are discussed. Finally gray predictive PI combined with gain scheduled PI is applied to a simulated BTG system, to demonstrate improved control qualities and stability as well as almost the same sensitivity of gray predictive PI to noise as PI.

Acknowledgements

I would like to express my heartful gratitute to Dr. Ray E. Rink who has given me kindly help and financial support through out this research work. Without his help and support, it is impossible to finish my research.

I also would like to give my thanks to Dr. Max Q-H Meng and Dr. Sirish Shah for their time to review the thesis, as well as their precious comments, and to the Department of Electrical and Computer Engineering for providing the opportunity and all kinds of supports.

Finally a well-deserved expression of appreciation should go to my wife, my parents for their invaluable care and encouragement.

Contents

Chapter 1 Introduction 1

Chapter 2 Gain Scheduled Control 6

2.1 Introduction to Gain Scheduled Control Based On Local Linearization 8

2.2 Stability Issues 11

2.3 Study On Three Types of Gain Scheduled PID 13

2.3.1 Three Types of Implicit Model-Based Gain Scheduled PID 13

2.3.2 Design Suggestion for Gain Scheduled PID Algorithm 17

2.3.3 Comparison among Three Types of Continuous-Linearly -Interpolated Gain Scheduled PID 25

2.3.4 Conclusions for Continuous-Linearly-Interpolated Gain Scheduled PID 50

Chapter 3 Gray Predictive PI Controller 52

3.1 Gray Prediction 52

3.1.1 Basic Concept 53

3.1.2 Gray Model — GM(1,1) Model 58

3.1.3 Gray Prediction 63

3.1.4 Discussion On Linear System 70

3.1.5 Simulation and Analysis 74

3.1.6 Noise Suppression 98

3.2 Error-Prediction-Based Gray Predictive PI 104

3.2.1. Consistency of Gray Predictive PI 106

Chapter 4 Application of Gain Scheduled PI and Gray Predictive PI 111

4.1 Basic Control Schemes For BTG System 113

4.1.1 Turbine Following 113

4.1.2 Boiler Following 113

4.1.3 Coordinated Control 114

4.2 Application of MFGS PI and Gray Predictive PI 115

Chapter 5 Conclusion 174

Bibliography 179

Figures

Figure 2.1 Setpoint Step Responses for Process Gain Varies 23
Figure 2.2 Setpoint Step Responses for Process Time Delay Varies 24
Figure 2.3 Variation of Kc as Process Gain Varies 30
Figure 2.4 Responses as Process Gain Varies 31
Figure 2.5 Kc, Ti and Td as Process Time Constant Varies 36
Figure 2.6 Responses as Process Time Constant Varies 37
Figure 2.7 Kc, Ti and Td as Process Time Constant Varies 38
Figure 2.8 Responses as Process Time Constant Varies 39
Figure 2.9 Kc, Ti and Td as Process Time Constant Varies 40
Figure 2.10 Kc, Ti and Td as Process Time Delay Varies 45
Figure 2.11 Responses as Process Time Delay Varies 46
Figure 2.12 Kc, Ti and Td as Process Time Delay Varies 47
Figure 2.13 Responses as Process Time Delay Varies 48
Figure 2.14 Kc, Ti and Td as Process Time Delay Varies 49

Figure 3.1 Comparison between the Original Sequence
and Its AGO Sequence 59
Figure 3.2 Gray Prediction --- Part 1 68
Figure 3.3 Gray Prediction --- Part 2 69
Figure 3.4 Gray Prediction --- The First Order Process (Con't) 77
Figure 3.4 Gray Prediction --- The First Order Process 78
Figure 3.5 Gray Prediction --- The First Order Process (Con't) 79
Figure 3.5 Gray Prediction --- The First Order Process 80
Figure 3.6 Gray Prediction --- The First Order Process (Con't) 81
Figure 3.6 Gray Prediction --- The First Order Process 82
Figure 3.7 Gray Prediction --- Process with Two
Different Negative Real Eigenvalues (Con't) 84

Figure 3.7 Gray Prediction --- Process with Two
Different Negative Real Eigenvalues 85
Figure 3.8 Gray Prediction --- Process with Two
Stable Complex Eigenvalues (Con't) 86
Figure 3.8 Gray Prediction --- Process with Two
Stable Complex Eigenvalues 87
Figure 3.9 Sensitivity to Noises of Gray Predictor 99
Figure 3.10 Large Data Interval --- Noise Suppression 102
Figure 3.11 Predictive Effect of LDISPI Gray Predictor
--- Noise Suppression 103
Figure 3.12 Basic Structure of Gray Predictive Control 105
Figure 3.13 Initial Structure of Suggested Gray Predictive Control 105
Figure 3.14 Final Structure of Suggested Gray Predictive Control 105

Figure 4.1 Basic Structure of Boiler Following
Variant Pressure Operation 117
Figure 4.2 Step Responses of BTG System at 70 MW
--- Electric Power (Continued) 120
Figure 4.2 Step Responses of BTG System at 70 MW
--- Electric Power 121
Figure 4.3 Step Responses of BTG System at 70 MW
--- Throttle Pressure (Continued) 122
Figure 4.3 Step Responses of BTG System at 70 MW
--- Throttle Pressure 123
Figure 4.4 Turbine Controller Gains from MFGS, FGS1 and FGS2 125
Figure 4.5 Step Responses of Fuel Controllers at N = 58 MW 130
Figure 4.6 Step Responses of Fuel Controllers at N = 76 MW 131
Figure 4.7 Step Responses of Fuel Controllers at N = 101 MW 132
Figure 4.8 Load Demand Rises from 76 MW to 101 MW
at The Rate of 9 MW/minute 133
Figure 4.9 Control Outputs of Fuel Controllers 134

Figure 4.10 Load Demand Lowers from 101 MW
to 76 MW at the rate of 9 MW/minute 135
Figure 4.11 Control Outputs of Fuel Controllers 136
Figure 4.12 Load Demand Lowers from 76 MW
to 58 MW at The Rate of 9 MW/minute 137
Figure 4.13 Control Outputs of Fuel Controllers 138
Figure 4.14 Load Demand Rises from 58 MW to 76 MW
at The Rate of 9 MW/minute 139
Figure 4.15 Control Outputs of Fuel Controllers 140
Figure 4.16 Load Demand Lowers from 101 MW to 76 MW
at The Rate of 17 MW/minute 141
Figure 4.17 Control Outputs of Fuel Controllers and Turbine Controllers 142
Figure 4.18 Load Demand Lowers from 76 MW to 58 MW
at The Rate of 25 MW/minute 143
Figure 4.19 Control Outputs of Fuel Controllers and Turbine Controllers 144
Figure 4.20 Load Demand Rises from 58 MW to 76 MW
at The Rate of 22 MW/minute 145
Figure 4.21 Control Outputs of Fuel Controllers and Turbine Controllers 146
Figure 4.22 Load Demand Rises from 76 MW to 101 MW
at The Rate of 17 MW/minute 147
Figure 4.23 Control Outputs of Fuel Controllers and Turbine Controllers 148
Figure 4.24 Load Demand Lowers from 101 MW to 76 MW
at the rate of 34 MW/mininute 149
Figure 4.25 Control Outputs of Fuel Controllers and Turbine Controllers 150
Figure 4.26 Load Demand Lowers from 75 MW
to 56 MW at the rate of 50 MW/minute 151
Figure 4.27 Control Outputs of Fuel Controllers and Turbine Controllers 152
Figure 4.28 Disturbance Rejection 154
Figure 4.29 Control Outputs to Fuel Controllers 155
Figure 4.30 Disturbance Rejection 156
Figure 4.31 Control Outputs of Fuel Controllers 157

Figure 4.32 Disturbance Rejection 158
Figure 4.33 Control Outputs of Fuel Controllers 159
Figure 4.34 Sensitivity to White Noises --- Responses to Disturbance 164
Figure 4.35 Sensitivity to White Noises ---
Control Outputs of Fuel Controllers 165
Figure 4.36 Sensitivity to White Noises --- Responses to Disturbance 166
Figure 4.37 Sensitivity to White Noises ---
Control Outputs of Fuel Controllers 167
Figure 4.38 Sensitivity to White Noises --- Responses to Disturbance 168
Figure 4.39 Sensitivity to White Noises ---
Control Outputs of Fuel Controllers 169
Figure 4.40 Sensitivity to Colored Noises --- Responses to Disturbance 170
Figure 4.41 Sensitivity to Colored Noises ---
Control Output of Fuel Controller 171
Figure 4.42 Sensitivity to Colored Noises --- Responses to Disturbance 172
Figure 4.43 Sensitivity to Colored Noises ---
Control Outputs of Fuel Controllers 173

Abbreviation and Notation

FGS1	The first type continuous-linearly-interpolated gain scheduled PID described by (2.6)
FGS1	The second type continuous-linearly-interpolated gain scheduled PID described by (2.7)
MFGS	The model-based continuous-linearly-interpolated gain scheduled PID described by (2.9)
AGO	Accumulated generating operation, or discrete integral
I-AGO	Inverse accumulated generating operation, or discrete derivative, or difference
e'(t)	de(t)/dt
sqrt(x)	Square root of x
x	Absolute value of x
LDISPI	Large data interval and small prediction interval
BTG	Boiler-Turbine-Generator
<<	Much lower than

Chapter 1

Introduction

In the real world, almost all actual processes have nonlinear features and indirectly time-varying parameters which vary with the changes of operating region. The linear model description of a system as well as linear design of control system with fixed parameters is often valid only in the neighborhood of the design operating point (equilibrium point). When nonlinear features of a system are strong, or a system involves motions with wide range and high speed, the results obtained from linear methods are often unsatisfactory.

At present, there is no general method for designing nonlinear controllers. What we have is a rich collection of alternative and complementary techniques [1], each best applicable to particular classes of nonlinear control problems.

This brief introductory chapter includes some general comments on these techniques, and concludes by outlining the rest of the thesis.

Trial-and-Error: The idea here is to use the analysis tools to guide the search for a controller which can then be justified by analysis and simulations. The phase plane method, the describing function method and Lyapunov analysis can all be used for the design. Experience and intuition are critical in this method. However, for complex systems, trial-and-error methods often fail.

Feedback Linearization [1][2]: The basic idea is to first transform a nonlinear system into a (fully or partially) linear system, and then apply the well-known and powerful linear design techniques to practical nonlinear control problems. It is applied to an important class of nonlinear systems (called input-state linearizable or asymptotically minimum-phase systems), and typically requires full state measurement. However, for most process control systems, it is difficult or impossible to obtain nonlinear state models, accurately measure and estimate all states, and derive the needed linearizing transformations. Also, major questions remain concerning the robustness of the approach. While it may, in principle, be successful, perfect accuracy of model and state estimating may be needed. This is impractical.

Robust Control [3]: In pure model-based nonlinear control, the control law is designed based on a nominal model of the physical system. In H^{∞}-based methods for the design of robust nonlinear control, the controller is designed based on the consideration of both the nominal model and some characterization of the model uncertainties. This latter approach has been proven to be effective in a variety of practical control problems. It usually requires state measurement and its design techniques are relatively complicated. Also, the enhancement of robustness is always at the cost of control quality, and robust control design is difficult for processes with large pure time-delay uncertainty.

Adaptive Control or Self-tuning Controller [4]: Adaptive and self-tuning control techniques can be used to treat an important class of nonlinear systems with measurable states or linearly parametrizable dynamics. For self-tuning controllers, because system models are described with differential equations, if systems have large pure time-delays or time constants, identification of a suitable model may become difficult or time-consuming. Usually they are suitable for slow time-varying systems.

Gain Scheduled Controller [5]][6][7][8]: Another important nonlinear control technique is the gain scheduled controller proposed by Stein [5] in 1980. Its main advantages are that linear design techniques can be applied to linearized system at each operating point. It can be applied to a class of implicitly time-varying nonlinear systems, i.e. the systems whose models are relative to their operating conditions. The major question in gain scheduling design is the selection of the gain scheduling procedures, i.e. the functions, which force control parameters to change with the scheduling variables. Another shortcoming of gain scheduled controllers is that the overall performance of control system must be checked by extensive trials, since no proof of robust stability is available as of this writing.

PI/PID controllers are still widely used in process control, since PI/PID controller is strongly robust, simple in concept, and easily tuned. The PID controller, if it is used properly, usually can achieve good control quality due to the predictive effect of the D mode. The significant shortcoming of D mode is that it may be too sensitive to noise. This feature often limits its application in process control.

Gray prediction [17][18] attempts to predict the future behavior of system by using current behavior data but ignoring what causes the current behavior. That is, it doesn't attempt to build a relationship (model) between system inputs (causes) and outputs (effects). Instead, it directly uses the outputs (effects) to build a model, then uses it to predict the future outputs. If the information about outputs is sufficient, it can effectively reduce the standard deviation of system so that it can be insensitive to noise. If it is used to form a gray predictive controller, this type of predictive controller can have stronger robustness than predictive controllers such as Smith Predictor [9], Dynamic Matrix Control (DMC)[10] and Generalized Predictive Control (GPC)[11], since it does not use any internal model.

A unit system of boiler-turbine-generator (BTG) is the major system in power generation. Two of most important controlled variables are fuel flow and turbine governor valve, while important outputs are steam pressure and electric power. If all other boiler control systems work properly, usually we can treat it as a 2-input, 2-output nonlinear time-varying, noisy, highly coupled system. Two major goals of BTG control systems are to achieve fast load following, and maintain throttle pressure (steam pressure) and load (electric power) stable. Both goals require that BTG control systems not only have good abilities for set-point tracking and disturbance rejection, but also have strong stability. As we know, BTG systems are noisy, i.e. the measured variables exhibit random fluctuations. Whether a controller is good for BTG systems is not only determined by the features that we just mentioned above, but more often is determined by whether it is sensitive to noise. If a controller is too sensitive to noise, it may cause the actuators to be worn out quickly. An industrial actuator usually is quite expensive, thus a good controller must be insensitive to noise.

Following on these brief comments, the outline of the rest of the thesis is as follows.

Chapter 2: Briefly introduce the general theory of gain scheduled controllers, then focus on the detailed studies on gain scheduled PID.

Chapter 3: Briefly introduce the theory of gray prediction, then develop a gray predictive PI controller, and discuss the relevant issues such as consistency, noise suppression, and approximate to high order processes etc.

Chapter 4: Apply gain scheduled PI and gray predictive PI to BTG control, using some simulation examples to show that control quality and stability of BTG system can be improved considerably, and also a properly designed gray predictive PI has almost the same sensitivity to noise as PI.

Chapter 5: Draw some conclusions for this thesis, and suggestions for further investigations.

Chapter 2

Gain Scheduled Control

In many situations, the relation between plant dynamic properties and operating conditions are known, i.e. changes of plant dynamic properties are caused by certain known time-varying linear or nonlinear properties. Hence, we can modify control parameters on-line in terms of operating conditions. This method is called gain scheduled control [4], for initially it was used to adapt system gains [5]. The gain scheduled controller is a special type of nonlinear control, and it consists of a feedback linear controller and a feed-forward compensator. The controller parameters change automatically as a function of operating conditions. That is, its feedback gain is modified by its feed-forward compensator. The different ways of designing feed-forward compensators can lead to the following two types of gain scheduled controllers.

Local Linearization: The first step is to linearize the model about one or more operating points (equilibrium points or operating conditions), then linear design methods are applied to the linearized model to obtain a satisfactory performance at each operating point. The second step is to interpolate the linear control law at intermediate operating conditions. That is, a feed-forward compensator is devised to change (schedule) the control law according to the scheduling variables.

Nonlinear Transformation -- Global Linearization: The first step is to transform the nonlinear system into a linear system independent of operating conditions by nonlinear transformation, then apply linear design methods to this linear system to get a satisfactory performance (global). The final step is to transform this linear controller back into nonlinear controller. Its feed-forward compensator consists of two nonlinear transformations [4]. An early application can be found in the literature [6]. A major problem for this method is that the nonlinear transformation usually can not be realized. In this thesis, only local linearization method will be discussed.

2.1 Introduction To Gain Scheduled Control Based On Local Linearization

Gain scheduled controllers based on local linearization can fall into two categories:

Explicit Model-based Gain Scheduled Controller: One of the typical methods has been analyzed in the literature [7]. In this method, the design for the feed-forward compensator of gain scheduling explicitly depends on the system model. Hence, the linearized model at each operating point must be obtained first.

Implicit Model-based Gain Scheduled Controller: One of the interesting methods can be found in the literature [8]. This method gets "optimal" functional relations between optimal controller parameters and nonlinear time-varying process parameters from some optimal control theory. When designing the feed-forward compensator of gain scheduling for a practical process, only these "optimal" functions are used, and there is no need to know the accurate relationship between scheduling parameters and process parameters at intermediate points. In this thesis, besides the implicit model-based gain scheduled PID proposed in the literature [8], two other types of popular implicit model-based gain scheduled PID also will be discussed.

Here, we briefly introduce an explicit model-based gain scheduled controller [7], and give some conclusions about stability issues. The detailed comparison studies between three implicit mode-based gains scheduled PIDs will be given later.

Consider a process described by

$$X'(t)=F(X(t),U(t),W(t))$$
$$Y(t)=H(X(t),U(t),W(t)), \quad t \geq 0, \tag{2.1}$$

where X(t) is the nx1 state, U(t) is the mx1 process input(or control output), Y(t) is the px1 process output, and W(t) is the qx1 vector of exogenous scheduling variables.

The feedback control law for the gain scheduling is selected as

$$U(t)=K(X(t),W(t),Z(t)) \tag{2.2a}$$
$$Z'(t)=Y(t) - Y(W(t)) \tag{2.2b}$$

where Y(W(t)) is the desired process output at each constant value of W(t) = W, and Z(t) is the px1 state of an integral-error compensator. Thus, at an operating point where Y(t) = Y(W) = constant, Z(W) is a constant vector that fixes the feedback (static) relationship between X(W) and U(W). During transients, Z(t) varies so as to force the error Y(t) - Y(W) to zero, provided that the system is stable.

At each operating point W(t)=W, the corresponding linearized closed-loop system can be written in the form

$$X_d'(t) = A(W)X_d(t) + B(W)U_d(t) + E(W)W_d(t) \tag{2.3a}$$
$$Z_d'(t) = C(W)X_d(t) + D(W)U_d(t) + F(W)W_d(t) \tag{2.3b}$$
$$U_d(t) = K_1(W)X_d(t) + K_2(W)W_d(t) + K_3(W)Z_d(t) \tag{2.3c}$$

in which $X_d(t)$ etc. are deviation variables and

$$K_1(W) = \partial K/\partial X(X(W),W,Z(W)),$$

$$K_2(W) = \partial K/\partial W(X(W),W,Z(W)),$$

$$K_3(W) = \partial K/\partial Z(X(W),W,Z(W)). \qquad (2.4)$$

where $K_1(W)$ and $K_3(W)$ are linear feedback control gains at each W, and $K_2(W)$ is a feed-forward gain on the exogenous scheduling variable deviation.

From (2.3) and (2.4), the eigenvalue matrix of the closed-loop linearized system is

$$\begin{bmatrix} A(W)+B(W)K_1(W) & B(W)K_3(W) \\ C(W)+D(W)K_1(W) & D(W)K_3(w) \end{bmatrix}, \qquad (2.5)$$

where A, B, C and D are known from the process model (2.1) and selected feedback control law(2.2). We can use many linear design techniques to choose $K_1(W)$ and $K_3(W)$ at each operating point W so that the eigenvalues of (2.5) are stable. The feed-forward compensator of gain scheduling, from (2.2a) and (2.4), should satisfy

$$K_2(W) = \partial U(W)/\partial W - K_1(W)\partial X(W)/\partial W - K_3(W)\partial Z(W)/\partial W.$$

At each operating point W, we can choose an arbitrary smooth function Z(W), then $K_2(W)$ can be determined from the above expression, and the linearized control law (2.3c) implies the gain scheduled control law

$$U(t) = K_1(W)X(t) + K_3(W)Z(t) + [\ U(W) - K_1(W)X(W) - K_3(W)Z(W)\ ],$$

the term in brackets being a slowly varying bias item, and with $Z'(t) = Y(t) - Y(W)$.

2.2 Stability Issues

From the literature [6][7][13][14][15], a few conclusions about stability of gain scheduling on slow variables can be drawn as:

1. If the eigenvalues of closed-loop system (2.5) at each operating point W have real parts less than some $\varepsilon < 0$, then given positive constants p and T, there exist positive constant $\delta_1(p)$ and $\delta_2(p, T)$ for which the following holds. If

$$\| X(0) - X(W(0)) \| < \delta_1,$$

$$(1/T)\int_{t}^{t+T} \| W'(q) \| \, dq < \delta_2, \qquad t \geq 0,$$

then

$$\| X(t) - X(W(t)) \| < p, \text{ and}$$

$$\|Y(t) - Y(W(t)) \| < p, \text{ for } t \geq 0.$$

The proof of this theorem is given in reference [6], and cited in [7].

2. Stability requires that the scheduling variables do not excite any unmodeled dynamics. In fact, since the scheduling variables are fed forward to the plant, it is unlikely that any scheduling strategy can escape this restriction.

3. If the scheduling variables vary too fast, the system may become unstable. One example can be found in literature [15]. The question of how fast is "too fast" has no simple answers in general. Conclusion 1 above only asserts that some δ_2 exists, not how large it may be, so the global stability of gain scheduled controller must be checked by extensive trials.

2.3 Study On Three Types of Gain Scheduled PID

2.3.1 Three Types of Implicit Model-Based Gain Scheduled PID

Suppose a PID controller has two sets of gain scheduled parameters, (K_{c1}, T_{i1}, T_{d1}) and (K_{c2}, T_{i2}, T_{d2}), at W_1 and W_2 $(W_1<W_2)$. We can have the control law

$$U(W, t) = K_c(W)[\ e(t) + \int(\ e(t)/T_i(W)\ )dt + T_d(W)(\ de(t)/dt\ )\ ].$$

Given a fixed threshold value W^* for the scheduling variable with $W_1 < W^* < W_2$,

```
if $W>W^*$ then
    $K_c = K_{c1}$,   $T_i = T_{i1}$,   $T_d = T_{d1}$;
else
    $K_c = K_{c2}$,   $T_i = T_{i2}$,   $T_d = T_{d2}$;
```

Usually we can not get good performance at intermediate operation points with this method.

If a continuous-linear interpolation is used, we can obtain the first implicit model-based gain scheduled PID (called FGS1 in this thesis)

$$U(W, t) = K_c(W)[\ e(t) + \int(\ e(t)/T_i(W)\ )dt + T_d(W)(\ de(t)/dt\ )\ ], \qquad (2.6a)$$

where K_c, T_i and T_d are scheduled as

$$K_c(W) = q_1K_{c1} + q_2K_{c2}\,, \qquad (2.6b)$$

$$T_i(W)^{-1} = K_c(W)^{-1}[q_1(K_{c1}/T_{i1}) + q_2(K_{c2}/T_{i2})], \qquad (2.6c)$$

$$T_d(W) = K_c(W)^{-1}[q_1(K_{c1}T_{d1}) + q_2(K_{c2}T_{d2})], \qquad (2.6d)$$

$$q_2 = (W - W_1)/(W_2 - W_1), \quad q_1=1-q_2, \qquad (2.6e)$$

where W is any intermediate value of scheduling variable between W_1 and W_2. We can see that the continuous-linearly-interpolated gain scheduled PID results in smooth transitions of controller parameters at any intermediate operating point.

The second continuous-linearly-interpolated gain scheduled PID is designed as(called FGS2 in this thesis)

$$U(W,t) = K_c(W)[\ e(t) + \int(\ e(t)/T_i(W)\ )dt + T_d(W)\ (\ de(t)/dt\ )\ ], \qquad (2.7a)$$

where K_c, T_i and T_d are scheduled as

$$K_c(W) = q_1K_{c1} + q_2K_{c2}, \qquad (2.7b)$$

$$T_i(W) = q_1T_{i1} + q_2T_{i2}, \qquad (2.7c)$$

$$T_d(W) = q_1T_{d1} + q_2T_{d2}, \qquad (2.7d)$$

$$q_2 = (W - W_1)/(W_2 - W_1), \quad q_1=1-q_2, \qquad (2.7e)$$

where W is any intermediate value of scheduling variable between W_1 and W_2. This differs from the first continuous-linearly-interpolated gain scheduled PID in the way that T_i and T_d are interpolated relative to K_c.

The analysis of the third type implicit model-based gain scheduled PID [8] is given in the following.

Consider an ITAE criterion-based PID for the 1st order system [16]

$$U(W, t) = K_c(W)[\, e(t) + \int(e(t)/T_i(W))dt + T_d(W) (de(t)/dt)] \qquad (2.8a)$$

$$K_c(t) = [A/K_p(t)][\tau/T]^{-B} \qquad (2.8b)$$

$$T_i(t) = C\tau^{D}T^{1-D} \qquad (2.8c)$$

$$T_d(t) = E\tau^{F}T^{1-F} \qquad (2.8d)$$

in which $K_p(t)$, $\tau(t)$ and $T(t)$ are the gain, time-delay and time constant of the process, respectively. A, B, C, D, E and F have the following values

$$A = 1.357, B = 0.947, C = 1.176, D = 0.738, E = 0.381, F = 0.995.$$

A common practice in selecting the controller gain is to maintain the closed-loop gain $K_c(t)K_p(t)$ at a constant value [16], if all other elements in a loop have constant values. The controller gain should be a strictly decreasing function of the process gain with an reciprocal relationship. It is nonlinear relationship! We have

$$\partial K_c(t)/\partial K_p(t) = -(A/K_p^2)(\tau(t)/T(t))^{-B} < 0,$$

therefore the smaller K_p, the faster K_c changes.

From (2.8c), we can see that $T_i(t)$ varies, to a degree, directly with $T(t)$.

$$\partial T_i(t)/\partial T(t) = C(1 - D)(\tau(t)/T(t))^{D} > 0,$$

therefore $T_i(t)$ increases as $T(t)$ increases. The smaller $T(t)$, the faster $T_i(t)$ changes.

From (2.8d), we have

$$T_d(t) = 0.381\tau^{0.995}T^{0.005},$$

$$\partial T_d(t)/\partial T(t) = 0.381*0.005(\tau/T)^{0.995},$$

therefore ITAE rule suggests that the derivative time is a very weak function of the process time constant. However, a rule of thumb states that the ratio between $T(t)$ and $T_d(t)$ should be kept at a constant value [16].

From the above analysis, the authors suggested that for the controller gain, its reciprocal should be used in controller parameter interpolation [8]. Now we can get an implicit model-based continuous-linearly-interpolated gain scheduled PID controller (called MFGS in this thesis):

$$U(W, t) = K_c(W)[\ e(t) + \int(\ e(t)/T_i(W)\ )dt + T_d(W)\ (\ de(t)/dt\ )\ ] \tag{2.9a}$$

$$K_c(W)^{-1} = q_1K_{c1}^{-1} + q_2K_{c2}^{-1}, \tag{2.9b}$$

$$T_i(W) = q_1T_{i1} + q_2T_{i2}, \tag{2.9c}$$

$$T_d(W) = q_1T_{d1} + q_2T_{d2}, \tag{2.9d}$$

$$q_2 = (W - W_1)/(W_2 - W_1), \quad q_1=1-q_2, \tag{2.9e}$$

where W is any intermediate value of scheduling variable between W_1 and W_2.

Comments: The authors suggested that the reciprocal of process gain should be used in the interpolation of controller gain K_c. However, the authors did not consider a relation between scheduling variable W and controller gain K_c as in (2.9b). In fact, it is better to use the reciprocal of the controller gain K_c in gain scheduled PID like (2.9b) only when the process gain can be approximated better by the first order polynomial of the scheduling variable W than by the first order polynomial of W^{-1}. If the process gain can be approximated better by the first order polynomial of W^{-1} than by the first order polynomial of W, the interpolation method for controller gain K_c in (2.6b) and (2.7b) is a reasonable selection. For example, suppose the process gain $K_p(W(t)) = W(t)$, the process time constant and time delay are both equal to 1. From (2.8b), we have

$$K_c(W) = AK_p(W)^{-1} = AW(t)^{-1}.$$

Obviously, the controller gain $K_c(W)$ is proportional to the reciprocal of W(t), and the interpolation method (2.9b) is better than (2.6b) and (2.7b). Instead if the process gain $K_p(W(t)) = W(t)^{-1}$, we have

$$K_c(W) = AK_p(W)^{-1} = AW(t).$$

The controller gain $K_c(W)$ is proportional to W(t) rather than the reciprocal of W(t). The interpolation method (2.6b) or (2.7b) is better than (2.9b). This basic principle is also suitable for the selection of the interpolation method for the controller integral time T_i and derivative time T_d.

2.3.2 Design Suggestion for Gain Scheduled PID Algorithm

For continuous-linearly-interpolated gain scheduled PID, the controller parameters are not constant. The designing of PID algorithm will greatly influence the performance of gain scheduled PID.

In discrete time-domain and continuous time-domain, the integral mode of PID can be designed with the following three methods.

Method 1: $I(k) = I(k-1) + [K_c(k)/T_i(k)]e(k)$, or

$$I(t) = \int [K_c(t)/T_i(t)]e(t)dt. \tag{2.10}$$

Method 2: $I_1(k) = I_1(k-1) + e(k)$, $I(k) = [K_c(k)/T_i(k)]I_1(k)$,

$I_1(k-1) = [K_c(k-1)/T_i(k-1)]^{-1} I(k-1)$ or

$$I(t) = [K_c(t)/T_i(t)]\int e(t)dt. \tag{2.11}$$

Method 3: $I_1(k) = I_1(k-1) + e(k)/T_i(k)$, $I(k) = K_c(k)I_1(k)$,

$I_1(k-1) = K_c(k-1)^{-1}I(k-1)$, or

$$I(t) = K_c(t)\int [e(t)/T_i(t)]dt. \tag{2.12}$$

If $K_c(t)$, $T_i(t)$ and $T_d(t)$ are all constant, there is no difference among these three PIDs. However, for time-varying $K_c(t)$, $T_i(t)$ and $T_d(t)$, the outputs of three PIDs are obviously different. The following example will show this conclusion.

Denote the outputs for Method 1, Method 2 and Method 3 as U_1, U_2 and U_3, respectively. For simplicity, only I-mode is used for control. Suppose at time k, these three I-mode controllers have the same outputs and controller parameters, e.g.

$U_1(k) = U_2(k) = U_3(k) = 60.0$, and

$K_c(k) = 4.5$, $T_i(k) = 15.0$.

At time k+1, given $e(k+1) = 1.0$, $K_c(k+1) = 5.0$ and $T_i(k+1) = 10.0$, we have

$U_1(k+1) = 60.0 + (5.0/10.0) * 1.0 = 60.5$,

$U_2(k+1) = (5.0/10.0)*(180.0 + 1.0) = 90.5$, and

$U_3(k+1) = 5.0*((60.0/4.5) + 1.0/10.0) = 67.15.$

Hence, the following conclusions can be made.

1. If scheduling variables are setpoints which may have step change, it is suggested to use the integral mode (2.10), since we have no reasons to abruptly change the historic output (integral output) in a gain scheduled PID. However, it also should be noted that it is usually not good to choose a setpoint as scheduling variable for continuous-linearly-interpolated gain scheduled controller although we sometimes do. It is obvious that for a large disturbance, the system may become unstable, since, if the setpoint remains unchanged, the gain scheduled controller is merely a controller with fixed parameters.

2. If the scheduling variables are continuous states and outputs of a process, or continuous exogenous variables, it is suggested to use the integral mode (2.11) and (2.12) for the process gain changes. A gain scheduled PID with the integral mode (2.10) can result in a large over-shot or over-damped step response for time-varying or nonlinear process gain K_p, since it can not quickly offset the change of output caused by the gain change. In contrast, the PIDs with the integral mode (2.11) and (2.12) may quickly offset the change of output, caused by the time-varying gain, via changing the historic outputs of PIDs.

3. If only process time constant or time delay changes, it is suggested to use the integral mode (2.10). A gain scheduled PID with the integral mode (2.11) and (2.12) may result in much over-shot or over damped step responses, since the change of time constant or time delay influences neither the historic value of process output nor the future steady-state value of output. It is unreasonable to change the historic output of PID. For example, after the process output reaches its steady state, let the time constant or time delay change. For this case, the process output remains unchanged. If the integral mode (2.11) or (2.12) is used, the output of controller will change due to the changes of K_p and T_i. It is an unexpected disturbance of controller output.

Simulation 2.1: Process Gain Varies

The process is described by

$$20dx(t)/dt + x(t) = K_p(y)u(t),$$
$$y(t) = x(t-4),$$

and

$$K_p(y) = 0.2, \text{ for } y \le 0.76,$$
$$K_p(y) = \text{sqrt}(y - 0.72), \text{ for } 0.76 < y < 7,$$
$$K_p(y) = 2.5, \text{ for } y \ge 7.$$

where $K_p(y)$ is a continuous process gain. The parameters of PID are set to be

$$K_c(y) = 1.6069/K_p(y), \quad T_i = 7.1713 \text{ and } T_d = 1.5363.$$

Figure 2.1 shows the set-point step responses. From Figure 2.1 (a) we can see the overshoot of PID with the integral mode (2.10) is unacceptable compared with its response at $y < 0.76$ (overshoot is less than 20%), since PID with the integral mode (2.10) can not quickly offset the influence of $K_p(y)$ to the output y as $K_p(y)$ changes. In contrast, PIDs with the integral mode (2.11) and (2.12) can achieve this goal by changing the historic output of PID (integral output). This simulation shows the case where $K_p(y)$ increases as y increases. If we have $K_p(y)$ increase as y decreases, PID with the integral mode (2.10) will have a greatly over damped step response.

Simulation 2.2: Process Time Delay Varies

The process is described by

$$20dx(t)/dt + x(t) = 0.2u(t),$$
$$y(t) = x(t-\tau(y(t-D))),$$

where D is a small time delay, and

$$\tau(y(t-D)) = 4 \text{ sec}, \text{ if } y(t-D) \le 0.76,$$
$$\tau(y(t-D)) = 7.8446*\text{sqrt}(y - 0.5) \text{ sec}, \text{ if } 0.76 < y(t-D) < 7,$$
$$\tau(y(t-D)) = 20 \text{ sec}, \text{ if } y(t-D) \ge 7.$$

The small time delay D is used only to make the system realizable by Simulink 4.2c. The parameters of PID are set to be

$$K_c = 5.9723\tau(y(t-D))^{-0.947},$$
$$T_i = 2.578\tau(y(t-D))^{0.738}, \quad \text{and}$$
$$T_d = 0.3867\tau(y(t-D))^{0.995}.$$

From Figure 2.2 (a) and (b) we can see that PID with the integral mode (2.11) and (2.12) have much over-damped step responses at both directions of y. As the process time delay varies, we only need to change the controller parameters, and should not change the historic output of PID i.e. integral output, for the variation of process time delay only influences the rising time and settling time of the process but changes neither the historic value of process output nor the future steady-state output of the process. This simulation shows the case in which τ(y) increases as y increases. If we have τ(y) decrease as y increases, PID with the integral mode (2.11) and (2.12) will have a step response with large over-shoot. The similar responses can be observed for the variation of process time constant T(y).

For a practical process, usually there exist the combined variations of K_p, T and τ. Which integral mode should be used can be determined by trials perhaps, though it may be difficult with arbitrary combined variations. Generally, the integral mode (2.12) is the best trade-off solution for general purpose, because

1. For a time-varying process gain, the change of controller gain K_c in (2.12), which results in the variation of historic output of controller, can quickly offset the change of process output caused by the variation of process gain. As a result, better control quality can be expected. The PID with the integral mode (2.12) can achieve the same control quality as the PID with the integral mode (2.11), whose control quality for time-varying gain is better than the PID with the integral mode (2.10).

2. A time-varying process time constant or time delay does not influence the steady-state values of the process. Instead, it only influences the time for the system to reach the steady-state. To achieve a good control quality, what we need is only to change the current controller parameters, therefore the PID with the integral mode (2.10) is the most reasonable. The PID with the integral mode (2.11) may result in the worst control quality, since both controller parameters K_c and T_i can change the historic output (integral output) of controller. The historic output (integral output) of controller may be partly changed by the process gain K_p in the integral mode (2.12), which will result in a trade-off control quality between PID with the integral mode (2.10) and (2.11).

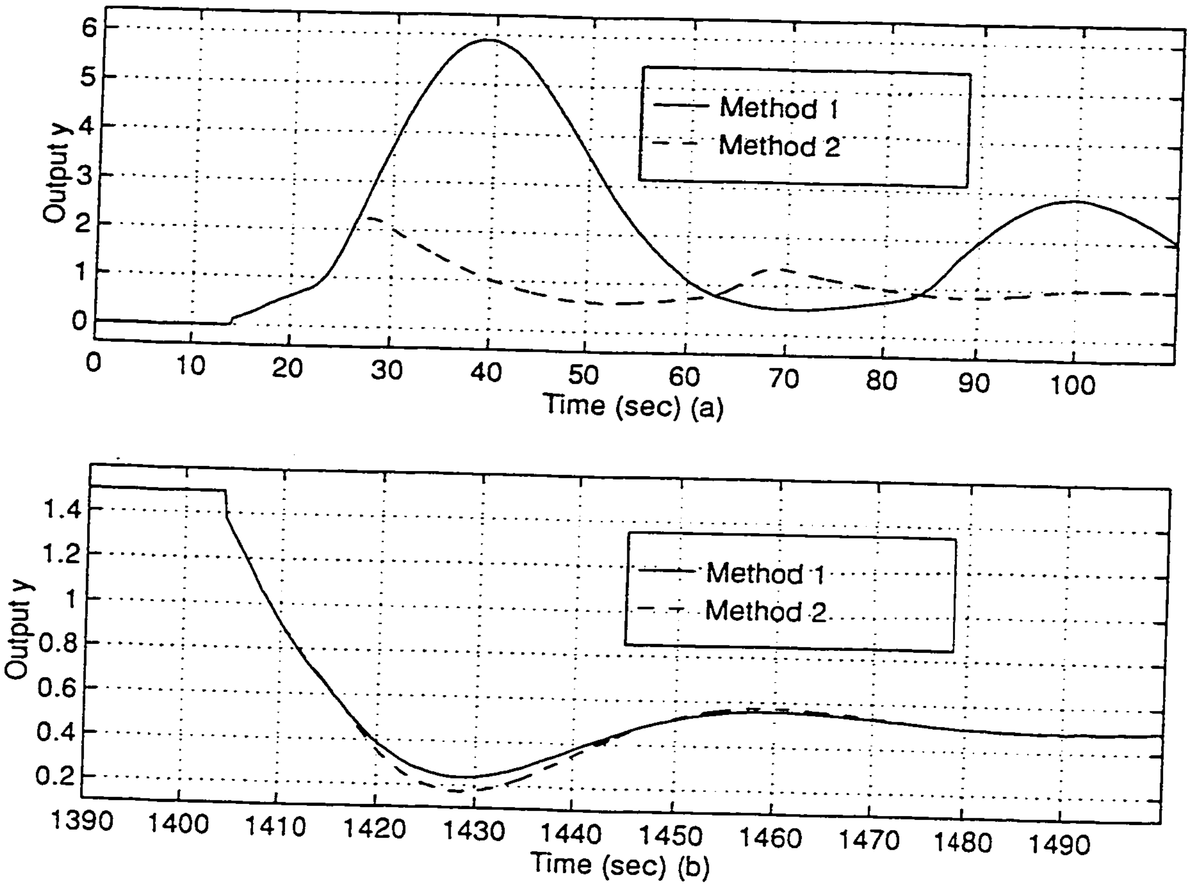


Figure 2.1 Setpoint Step Responses for Process Gain Varies

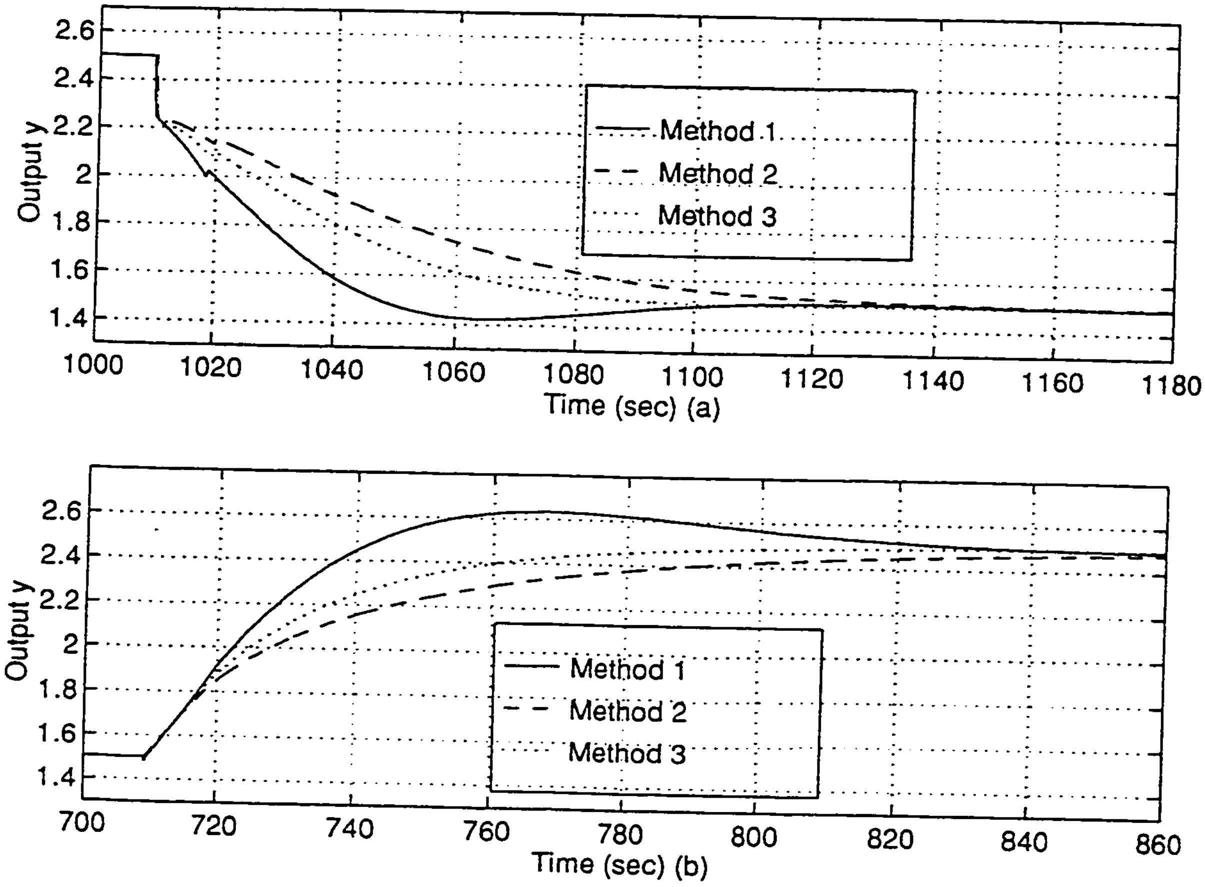


Figure 2.2 Setpoint Step Responses for Process Time Delay Varies

2.3.3 Comparison among Three Types of Continuous-Linearly-Interpolated Gain Scheduled PID

Three types of continuous-linearly-interpolated PID are given in (2.6), (2.7) and (2.9), and called FGS1, FGS2 and MFGS, respectively. The ITAE criterion based PID will be a reference.

A class of important implicit time-varying processes will be used for simulation. That is, the single-input single-output process whose process parameters, gain, time constant and time delay, are linear or non-linear functions of the scheduling variable W, and can be approximated better by the first order polynomial of scheduling variable W than the first order polynomial of W^{-1}. The process is given by

$$T(y)dx(t)/dt + x(t) = K_p(y)u(t) + K_p(y)d(t), \qquad (2.13a)$$

$$y(t) = x(\ t - \tau(y(t-D))\ ), \qquad (2.13b)$$

where D is a small time delay. It is used only to make the process realizable by the Simulink 4.2c. Simply we suppose that the scheduling variable W is equal to y.

No matter how the process parameters vary from v_1 to v_n, we can obtain 3 basic types of parameter variations by properly dividing $[v_1, v_n]$ into $[v_1, v_2]$, $[v_2, v_3]$, ... , $[v_{n-1}, v_n]$:

Situation 1: Linear variation.

Situation 2: Non-linear variation with each intermediate value, $v \in (v_i, v_{i+1})$, greater than the corresponding linear value.

Situation 3: Non-linear variation with each intermediate value, $v \in (v_i, v_{i+1})$, lower than the corresponding linear value.

Similar to the above, as y varies from y_1 to y_m, the relation between the process output y and process parameter, gain K_p, time constant T, or time delay τ, also consists of 3 basic forms by properly dividing $[y_1, y_m]$ into $[y_1, y_2]$, $[y_2, y_3]$, ..., $[y_{m-1}, y_m]$.

Situation 4: y varies and the process parameter remains unchanged.
Situation 5: y increases and the process parameter increases, or y decreases and the process parameter decreases.
Situation 6: y increases and the process parameter decreases, or y decreases and the process parameter increases.

Situation 4 can be treated as a constant process, and will not be studied in this thesis. The following simulation and analysis are based on the remaining 5 basic situations above.

Simulation 2.3: Process Gain $K_p(y)$ Varies

The time constant and time delay of the process (2.13) are

$$T(y) = 20 \text{ sec}, \quad \tau(y(t-D)) = 4 \text{ sec},$$

where D is the small time delay. The parameters of PIDs are set to be

ITAE: $K_c(y) = 0.2961/K_p(y)$,
MFGS, FGS1 and FGS2: $K_c(0.76) = 1.4805$, $K_c(7) = 0.1185$,

where $K_c(y)$ for all controllers are de-tuned for the sake of comparison. The integral mode (2.12) is used in this simulation.

CASE 1: Situation 2 combined with Situation 5

In this case, the process gain $K_p(y)$ is non-linear, and $K_p(y)$ increases as y increases.

$K_p(y) = 0.2$, for $y(t) \leq 0.76$,
$K_p(y) = \text{sqrt}(y(t) - 0.72)$, for $0.76 < y(t) < 7$,
$K_p(y) = 2.5$, for $y(t) \geq 7$.

Controller gain $K_c(y)$ obtained from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.3(a). $K_c(y)$ in MFGS is very close to, but somewhat higher than, $K_c(y)$ in ITAE. The controller gain $K_c(y)$ for FGS1 and FGS2 are the same but too large, since $K_c(y)$ in ITAE is proportional to 1/sqrt(y(t)-0.72), reciprocal of nonlinear $K_p(y)$, $K_c(y)$ in MFGS is proportional to 1/(0.3686y-0.0801), reciprocal of linear $K_p(y)$, and $K_c(y)$ in FGS1 and FGS2 are proportional to linear $K_p(y)$, 0.386y-0.0801. Figure 2.4 shows the responses. Figure 2.4(a) indicates that FGS1 and FGS2 have unacceptable over-shoots although $K_c(y)$ in this simulation is greatly de-tuned. Figure 2.4(b) and (c) indicate that the abilities of FGS1 and FGS2 to reject the step disturbance d(t), positive and negative, shown in (2.13a), are also worse than those of MFGS.

CASE 2: Situation 1 combined with Situation 5

In this case, the process gain $K_p(y)$ is linear, and $K_p(y)$ increases as y increases.

$K_p(y) = 0.2$, for $y(t) \leq 0.76$,
$K_p(y) = 0.3686y(t) - 0.0801$, for $0.76 < y(t) < 7$,
$K_p(y) = 2.5$, for $y(t) \geq 7$.

$K_c(y)$ obtained from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.3(b). $K_c(y)$ in

MFGS is exactly the same as $K_c(y)$ in ITAE. $K_c(y)$ in FGS1 and FGS2 are still unacceptable, although they are slightly closer to $K_c(y)$ in ITAE, since both $K_c(y)$ from ITAE and MFGS are proportional to the reciprocal of linear $K_p(y)$, 1/(0.3686y-0.0801), and $K_c(y)$ in FGS1 and FGS2 are proportional to linear $K_p(y)$, 0.386y-0.0801 instead.

CASE 3: Situation 3 combined with Situation 5

In this case, the process gain $K_p(y)$ is non-linear, and $K_p(y)$ increases as y increases.

$$K_p(y) = 0.2, \ \text{if } y(t) \le 0.76,$$
$$K_p(y) = 0.033(y(t) + 1.701)^2, \ \text{if } 0.76 < y(t) < 7,$$
$$K_p(y) = 2.5, \ \text{if } y(t) \ge 7.$$

$K_c(y)$ obtained from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.3(c). $K_c(y)$ in MFGS is slightly lower than $K_c(y)$ in ITAE. $K_c(y)$ in FGS1 and FGS2 are still unacceptable, although they are further closer to $K_c(y)$ in ITAE. The reason is similar to CASE 1 and CASE 2.

Now, let's briefly discuss the case in which the process gain decreases as y decreases. If we agree that the controller gain K_c from ITAE is the best, then, from Figure 2.3 (a), (b) and (c), we can see that K_c from MFGS is the closest to K_c in ITAE, while the controller gains from FGS1 and FGS2 are too large. Hence the control quality of MFGS for the setpoint tracking, again, can be expected to be the best. Certainly, as the process gain decreases, the control system tends to be more robust. Hence, the control qualities of setpoint tracking for MFGS, FGS1 and FGS2 have much less difference compared with the cases in which the process gain increases as the process output y increases.

For the case in which the process gain increases as the process output y decreases, the controller gains for FGS1 and FGS2 are too high compared with the controller gains for

ITAE and MFGS, which are shown in Figure 2.3 (a), (b) and (c). Hence the integral outputs for FGS1 and FGS2 in (2.12) may decrease too slowly. This may make the control outputs of FGS1 and FGS2 decrease slower, or much slower than the control outputs of ITAE and MFGS. As a result, the control outputs of FGS1 and FGS2 may not be able to quickly offset the effect of the process gain increasing, finally the set-point step responses for FGS1 and FGS2 will be more, or much more, damped than MFGS. The similar results also can be expected for the cases when y increases the process gain decreases. The only difference is that as the process gain decreases, the control system tends to be less sensitive to the variations of controller parameters. Hence, the control qualities of MFGS, FGS1 and FGS2 for set-point tracking are less, or much less different compared with the cases in which the process gain increases as the process output y decreases.

From the simulation and discussion above, we can conclude that it is difficult to use FGS1 and FGS2 to achieve good control qualities for both setpoint tracking and disturbance rejection for the time varying K_p, which can be approximated better by the first order polynomial of the scheduling variable W than the first order polynomial of W^{-1}.

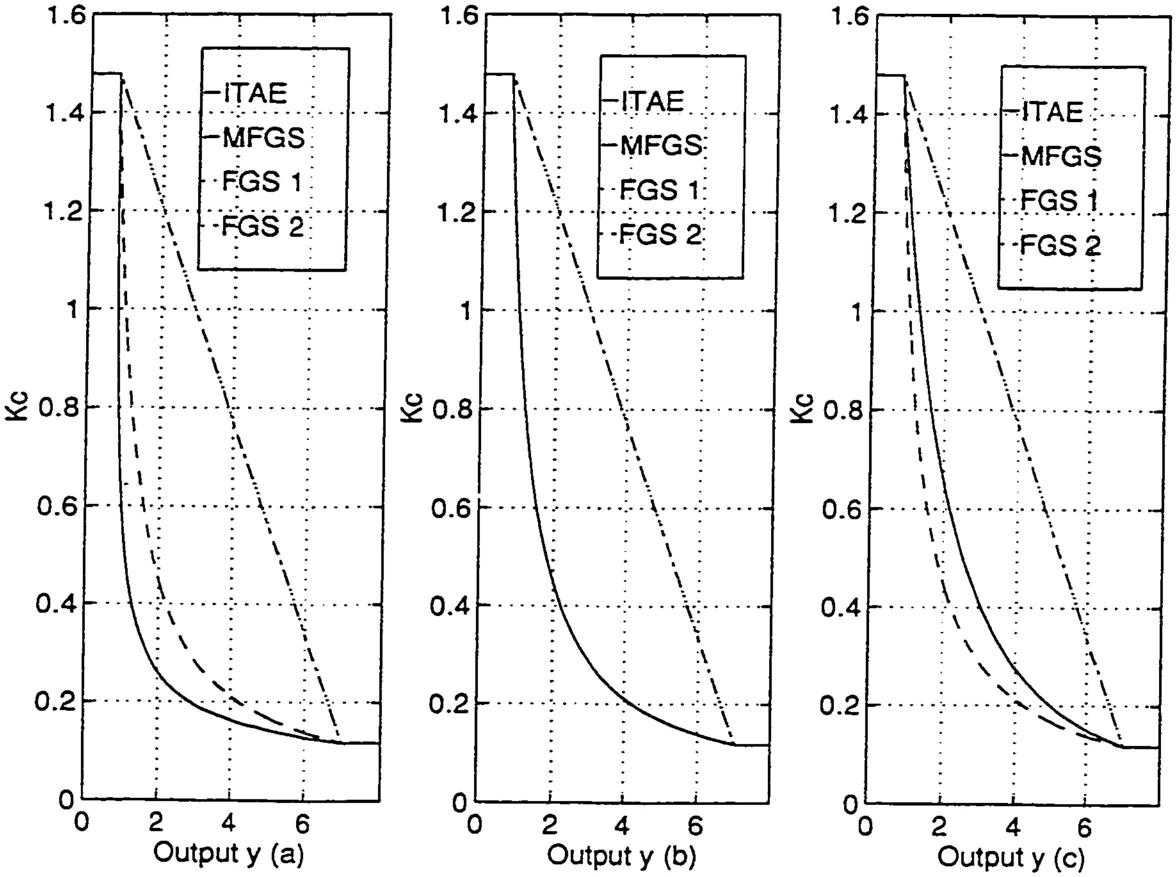


Figure 2.3 Variation of Kc as Process Gain Varies

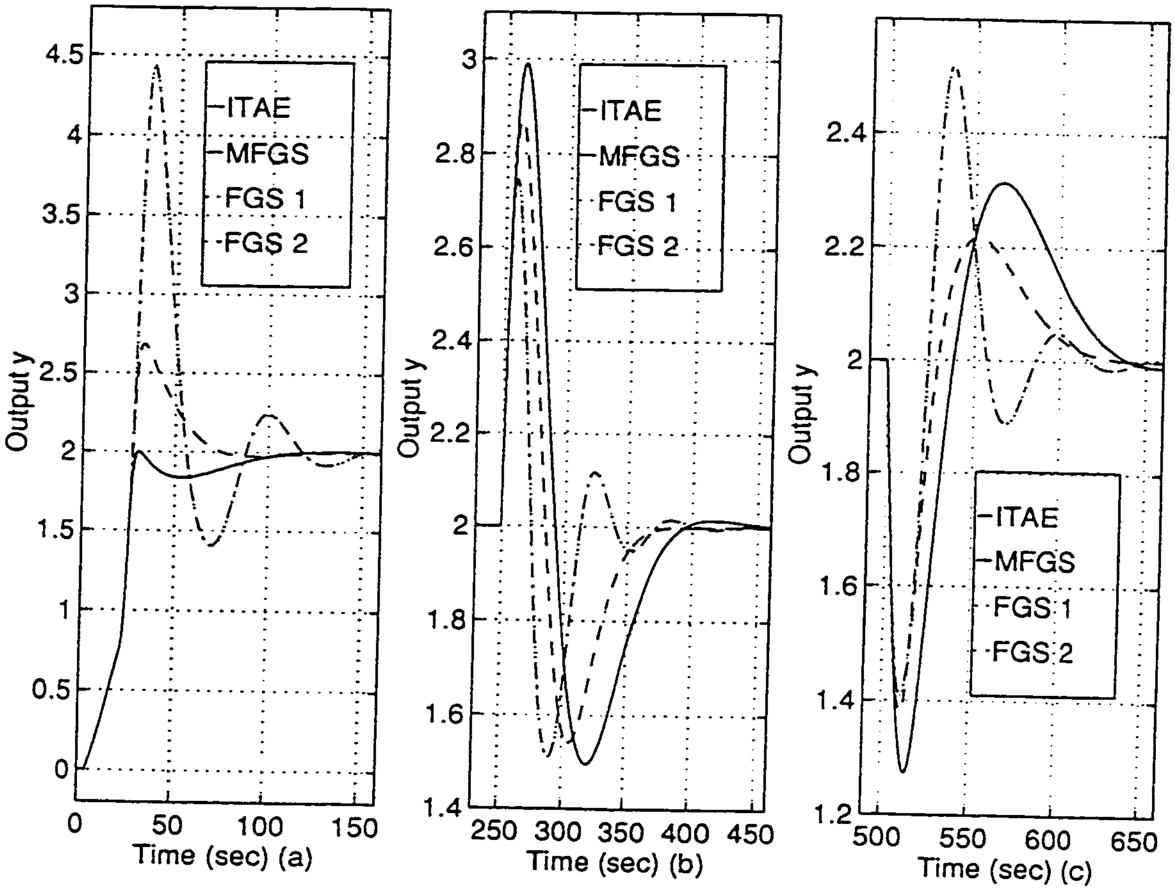


Figure 2.4 Responses as Process Gain Varies

(a) Setpoint Step Responses

(b) Responses for Positive Step Disturbance

(c) Responses for Negative Step Disturbance

Simulation 2.4: Process Time Constant T(y) Varies

The gain and time delay of the process (2.13) are

$K_p(y) = 0.2, \ \tau(y(t\text{-}D)) = 4$ sec,

where D is the small time delay. The parameters of PIDs are set to be

ITAE:

$K_c(y) = 0.4709T(y)^{0.947}$,

$T_i(y) = 3.2717T(y)^{0.262}$,

$T_d(y) = 1.5135T(y)^{0.005}$.

MFGS, FGS1 and FGS2:

$K_c(0.76) = 8.0346, \quad K_c(7) = 1.75,$

$T_i(0.76) = 7.1713, \quad T_i(7) = 4.704,$

$T_d(0.76) = 1.5363, \quad T_d(7) = 1.524.$

The integral mode (2.10) is used in this simulation.

CASE 1: Situation 2 combined with Situation 6

In this case, the process time constant T(y) is non-linear, described by Situation 2, and T(y) decreases as y increases.

$T(y) = 20$ sec, for $y(t) \leq 0.76$,

$T(y) = 7.8446 * \text{sqrt}(\ 7.26 - y(t)\ )$ sec, for $0.76 < y < 7$,

$T(y) = 4$ sec, for $y(t) \geq 7$.

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.5(a), (b) and (c), respectively. Because $K_c(y)$ in ITAE is proportional to $T(y)^{0.947}$, (7.8446*sqrt(7.26-y))$^{0.947}$, while $K_c(y)$ in MFGS is proportional to $(-2.5641y+21.9487)^{-1}$, and Kc(y) in FGS1 and FGS2 is proportional to (-2.5641y+21.9487), therefore, from Figure 2.5(a), we can see that MFGS has too low $K_c(y)$ compared with ITAE. Instead, FGS1 and FGS2 have reasonable $K_c(y)$ which are slightly lower than $K_c(y)$ in ITAE. From 2.5(b) and (c), MFGS and FGS2 have the same $T_i(y)$ and $T_d(y)$ which are reasonable but somewhat lower than ITAE, and FGS1 has almost the same $T_i(y)$ and $T_d(y)$ as ITAE for this case. Figure 2.6 shows the step responses and disturbance rejection. FGS1 and FGS2 are better selection, but MFGS is worse, with greatly over-damped step response and comparatively worse disturbance rejection.

CASE 2: Situation 3 combined with Situation 6

In this case, the process time constant T(y) is non-linear, described by Situation 2, and T(y) decreases as y increases.

$$T(y) = 20 \text{ sec}, \quad \text{for } y \le 0.76,$$
$$T(y) = 0.157 * (y - 12.0483)^2 \text{ sec}, \quad \text{for } 0.76 < y < 7,$$
$$T(y) = 4 \text{ sec}, \quad \text{for } y \ge 7.$$

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.7(a), (b) and (c), respectively. From the similar reasons in CASE 1, in Figure 2.7(a), MFGS still has considerable lower $K_c(y)$ than ITAE. FGS1 and FGS2 have reasonable $K_c(y)$ which is slightly higher than $K_c(y)$ in ITAE. From 2.7(b) and (c), MFGS and FGS2 have the same $T_i(y)$ and $T_d(y)$ which are very close to those from ITAE, and FGS1 has somewhat higher $T_i(y)$ and $T_d(y)$ than ITAE. Figure 2.8 shows the step responses and disturbance rejection. again FGS1 and FGS2 are better selection, but MFGS is worse, with over-damped step response, although it is better than in CASE 1. Certainly the disturbance rejection of MFGS in this case is not bad compared with ITAE, FGS1 and FGS2.

CASE 3: Situation 1 combined with Situation 6

In this case, the process time constant T(y) is linear, and T(y) decreases as y increases.

$T(y) = 20$ sec, for $y \leq 0.76$,

$T(y) = -2.5641y + 21.9487$ sec, for $0.76 < y < 7$,

$T(y) = 4$ sec, for $y \geq 7$.

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.9(a), (b) and (c), respectively. For this case, $K_c(y)$ for MFGS is lower than CASE 1 but higher than CASE 2. FGS1 and FGS2 have the best $K_c(y)$ (almost the same as $K_c(y)$ as ITAE) in this case compared with the corresponding values in CASE 1, which are too low, and the corresponding values in CASE 2, which are too high. From 2.9(b) and (c), MFGS and FGS2 have the same $T_i(y)$ and $T_d(y)$ which are somewhat lower than those from ITAE, and FGS1 has somewhat higher $T_i(y)$ and $T_d(y)$ than ITAE. We can again expect that FGS1 and FGS2 can have better control quality than MFGS for setpoint tracking, and slightly better disturbance rejection than MFGS.

Now, let's briefly discuss the case in which the process time constant increases as the process output y decreases. From Figure 2.5(a), 2.7(a) and 2.9(a), if we agree that the controller gain K_c from ITAE is the best, we can see that K_c from FGS1 and FGS2 is the closest to K_c from ITAE, while the controller gains K_c from MFGS is too small, comparatively. From 2.5(b),(c), 2.7(b),(c) and 2.9(b),(c), we can see that T_i and T_d from FGS1 is the closest to those from ITAE, so that the control quality of FGS1 for the setpoint tracking, again, can be expected to be the best. Certainly, as the process time constant increases, the control system tends to be less sensitive to the variations of the controller parameters. Hence, the control qualities of setpoint tracking for MFGS, FGS1 and FGS2 have much less difference compared with the cases in which the process time constant decreases as the process output y increases. For the case in which the process time constant decreases as the process output y decreases, the controller gains for MFGS is too small compared to the controller gains from ITAE, FGS1 and FGS2, which are shown in Figure 2.5 (a), 2.7(a) and 2.9(a). Hence the integral output for MFGS in (2.10) may decrease too slowly, and it may make the control output of MFGS to decrease slower, or much slower than that of ITAE. Consequently the setpoint step response for MFGS will be more, or much more, damped than ITAE. Comparatively FGS1 and FGS2 are better. The similar results also can be expected for the cases in which the process time constant increases as the process output y increases. The only difference is that as the process time constant increases, the control system tends to be less sensitive to the variations of controller parameters. As a result, the control qualities of MFGS, FGS1 and FGS2 for set-point tracking are less, or much less different compared with the cases in which the process time constant increases as the process output y decreases.

From CASE 1 to CASE 3, Simulation 2.4, we can see that FGS1 and FGS2 can achieve better control qualities for setpoint tracking, and slightly better disturbance rejection than MFGS for the time varying process time constant T, which can be approximated better by the first order polynomial of the scheduling variable W than the first order polynomial of W^{-1}, since $K_c(y)$ in MFGS is too small. MFGS may have more, or much more, damped set-point step responses than FGS1 and FGS2 while the process is running at intermediate values of the scheduling variable. Certainly, the variations of process time constants usually do influence control quality much less than the variation of process gains and time delays. This also can be observed typically in predictive controllers such as MAC and GPC, etc. It should be pointed out that the optimal formula of ITAE is derived for the optimal set-point tracking, it may not be guaranteed that the disturbance rejection is also optimal.

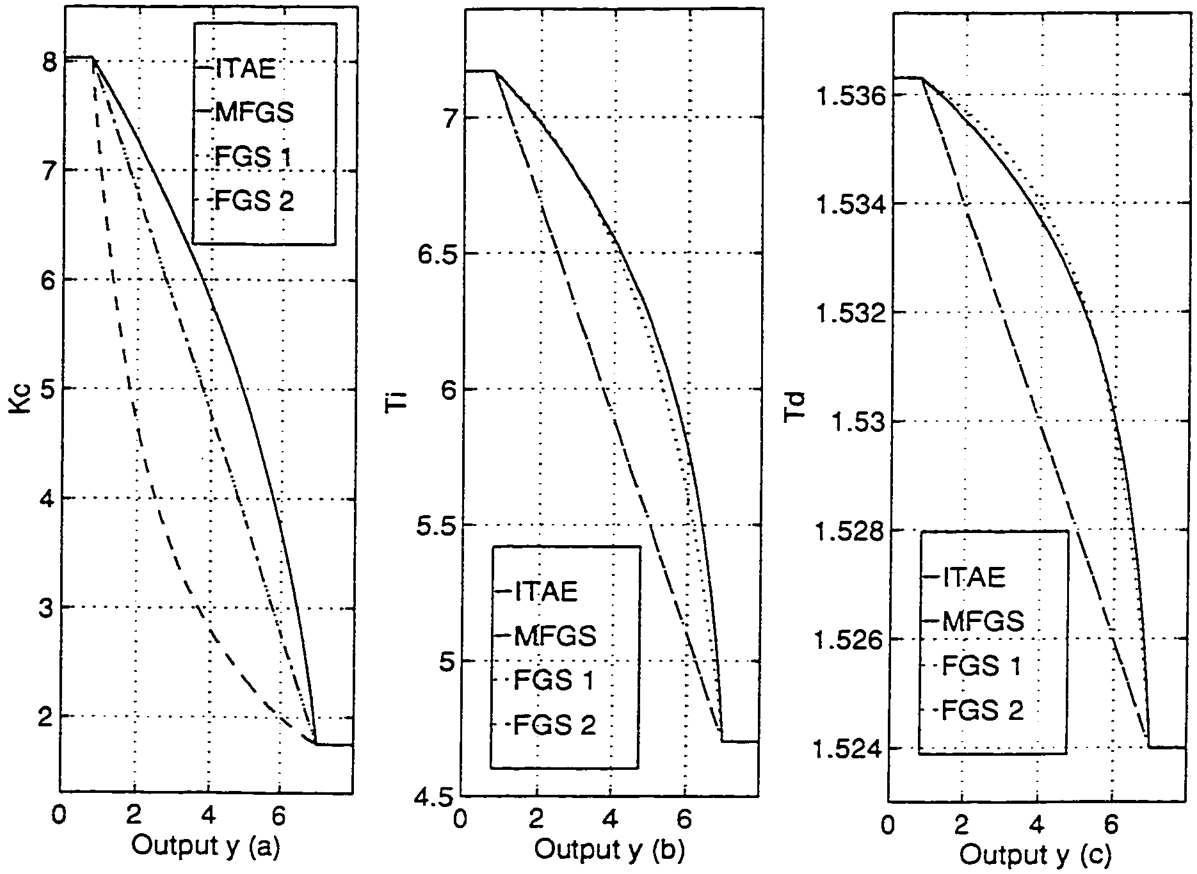


Figure 2.5 Variation of Kc as Process Time Constant Varies

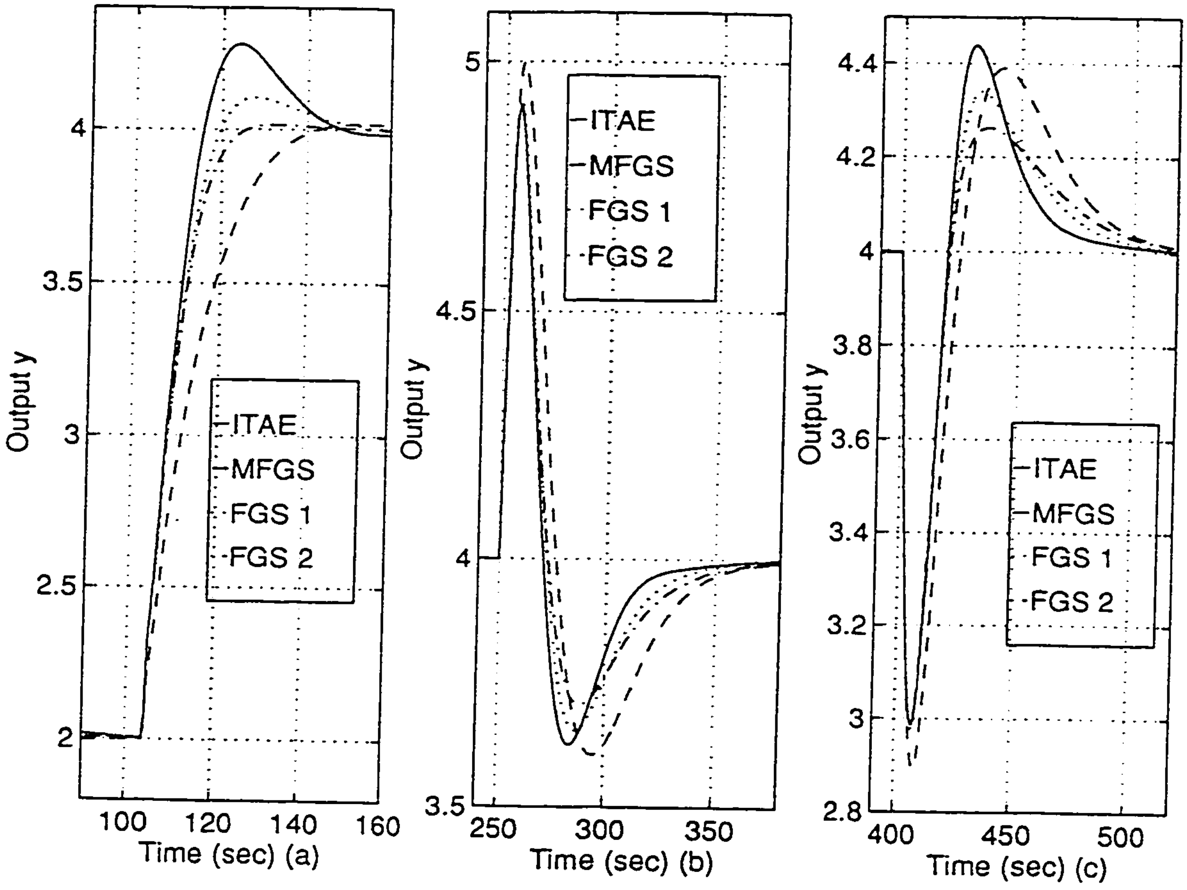


Figure 2.6 Responses as Process Time Constant Varies

(a) Setpoint Step Responses

(b) Responses for Positive Step Disturbance

(c) Responses for Negative Step Disturbance

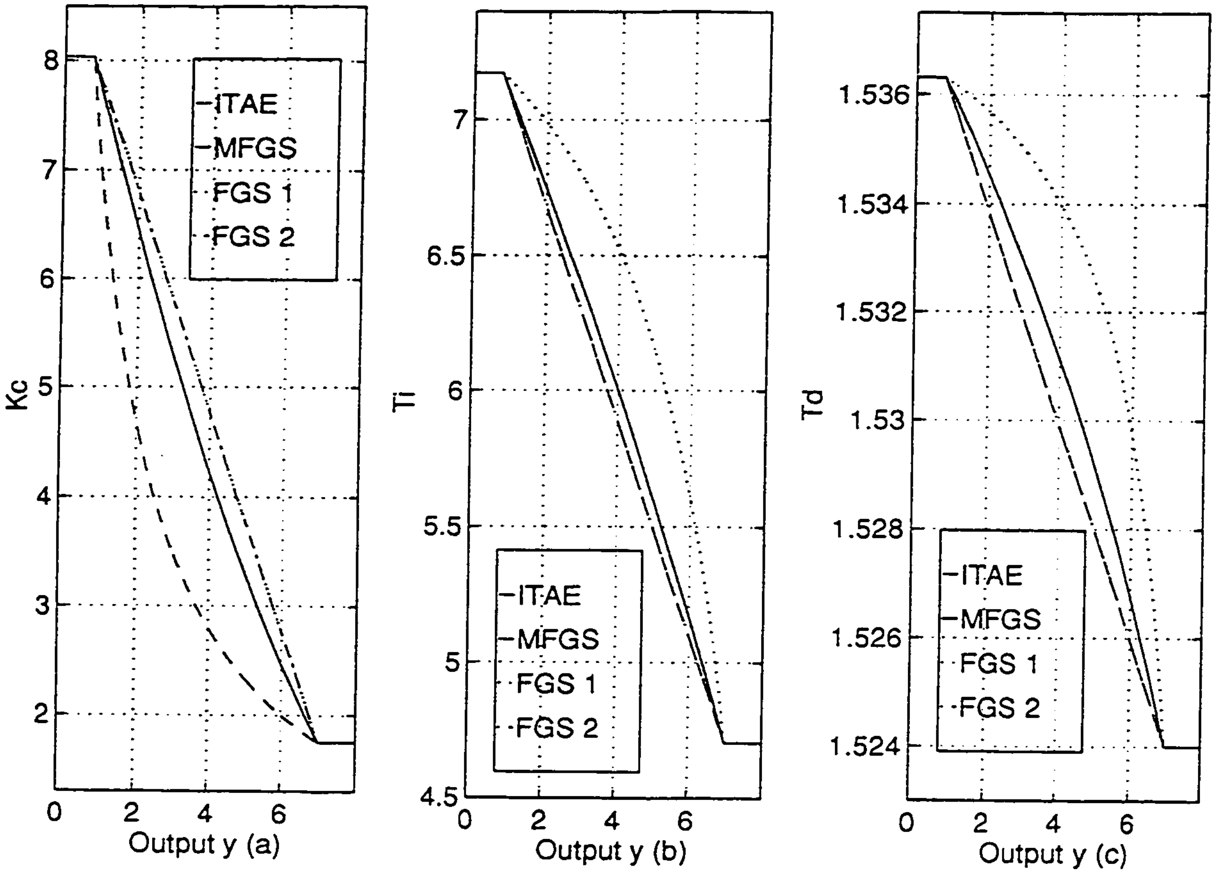


Figure 2.7 Variation of Kc as Process Time Constant Varies

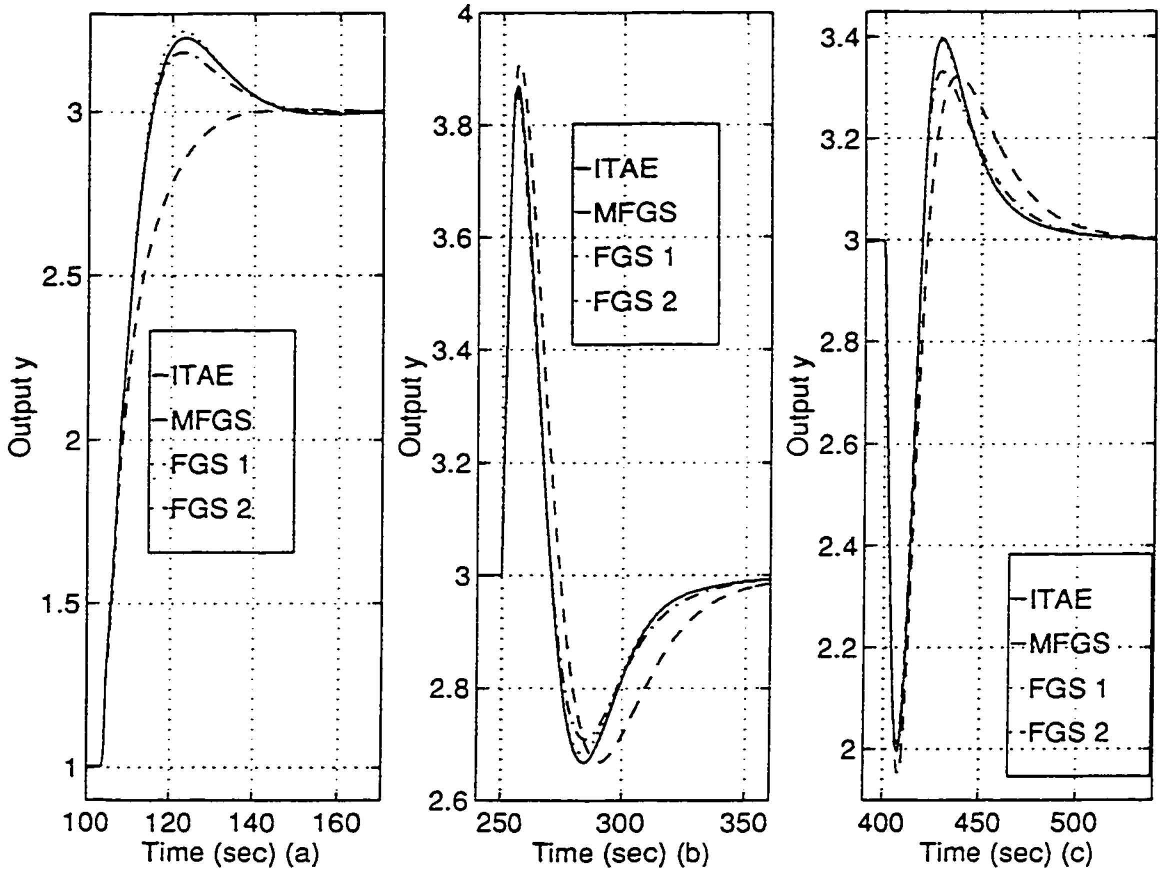


Figure 2.8 Responses as Process Time Constant Varies

(a) Setpoint Step Responses

(b) Responses for Positive Step Disturbance

(c) Responses for Negative Step Disturbance

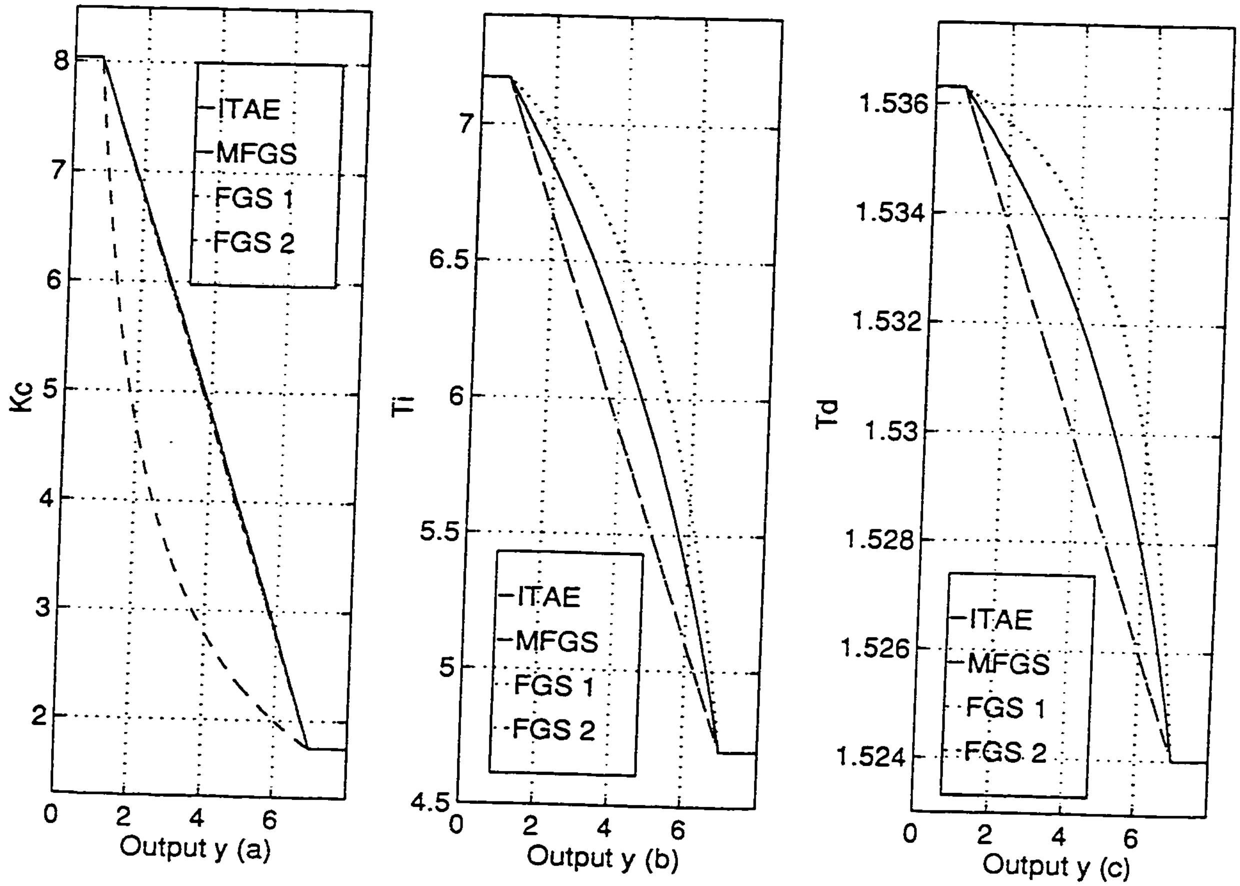


Figure 2.9 Variation of Kc as Process Time Constant Varies

Simulation 2.5: Process Time Delay $\tau(y)$ Varies

The gain and time constant of the process (2.13) are

$K_p(y) = 0.2$, $T(y) = 20$ sec,

The parameters of PIDs are set to be

ITAE:

$K_c(y) = 29.8616\tau(y)^{-0.947}$,

$T_i(y) = 2.578\tau(y)^{0.738}$,

$T_d(y) = 0.3867\tau(y)^{0.995}$.

MFGS, FGS1 and FGS2:

$K_c(0.76) = 8.0346$, $K_c(7) = 1.75$,

$T_i(0.76) = 7.1713$, $T_i(7) = 23.52$,

$T_d(0.76) = 1.5363$, $T_d(7) = 7.62$.

The integral mode (2.12) is used in this simulation.

CASE 1: Situation 2 combined with Situation 5

In this case, the process time delay $\tau(y)$ is non-linear, described by Situation 2, and $\tau(y)$ increases as y increases.

$\tau(y(t\text{-}D)) = 4$ sec, for $y(t\text{-}D) \leq 0.76$,

$\tau(y(t\text{-}D)) = 7.8446 * \text{sqrt}(\ y(t\text{-}D) - 0.5\ )$ sec, for $0.76 < y(t\text{-}D) < 7$,

$\tau(y(t\text{-}D)) = 20$ sec, for $y(t\text{-}D) \geq 7$.

where D is the small time delay.

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.10(a), (b) and (c), respectively. $K_c(y)$ in ITAE is proportional to $\tau(y)^{-0.947}$, and Kc(y) in MFGS is proportional to $(2.5641y + 2.0513)^{-1}$, while $K_c(y)$ in FGS1 or FGS2 is proportional to 2.5641y + 2.0513. Similar to Simulation 2.3 "Process Gain Varies", $K_c(y)$ in MFGS is very close to, but somewhat higher than, $K_c(y)$ in ITAE. $K_c(y)$ from FGS1 and FGS2 are too large. MFGS and FGS2 have the same $T_i(y)$ and $T_d(y)$ which are reasonable close to ITAE. FGS1 has too small $T_i(y)$. In ITAE, K_c/T_i is proportional to $\tau^{-1.685}$, but in FGS1, the linear interpolation is used to calculate the intermediate values of K_c/T_i, in which it is assumed that K_c/T_i is proportional to the time delay τ. It is very unreasonable. Finally, from Figure 2.11(a), we can see that for the set-point step response, FGS1 has the largest over-shoot and settling time, FGS2 has larger ones, and MFGS has the least. In Figure 2.11(b), for the positive step disturbance d(t) shown in (2.13a), although it is difficult to judge, to some extent, which control quality is the best, at least, in the author's opinion, the control quality of ITAE is the most reasonable. Based on this point, the disturbance rejection of MFGS is the best compared with FGS1 and FGS2. Therefore MFGS is the best, FGS2 is worse and FGS1 is the worst for this simulation case.

CASE 2: Situation 3 combined with Situation 5

In this case, the process time delay $\tau(y)$ is non-linear, described by Situation 3, and $\tau(y)$ increases as y increases.

$\tau(y(t\text{-}D)) = 4$ sec, if $y(t\text{-}D) \leq 0.76$,
$\tau(y(t\text{-}D)) = 0.157 * \text{sqrt}(\ y(t\text{-}D) + 4.2883\ )2$ sec, if $0.76 < y(t\text{-}D) < 7$,
$\tau(y(t\text{-}D)) = 20$ sec, if $y(t\text{-}D) \geq 7$.

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.12(a), (b) and (c), respectively. The step responses and disturbance rejection are shown in Figure 2.13. From similar reasons as in CASE 1, we can observe the similar responses to CASE 1. Again MFGS is the best, FGS2 is worse and FGS1 is the worst due to too large $K_c(y)$ and too small $T_i(y)$.

CASE 3: Situation 1 combined with Situation 5

In this case, the process time delay $\tau(y)$ is linear, and $\tau(y)$ increases as y increases.

$$\tau(y(t\text{-}D)) = 4 \text{ sec}, \ \text{if } y(t\text{-}D) \leq 0.76,$$
$$\tau(y(t\text{-}D)) = 2.5641y + 2.0513 \text{ sec}, \ \text{if } 0.76 < y(t\text{-}D) < 7,$$
$$\tau(y(t\text{-}D)) = 20 \text{ sec}, \ \text{if } y(t\text{-}D) \geq 7.$$

$K_c(y)$, $T_i(y)$ and $T_d(y)$ from ITAE, MFGS, FGS1 and FGS2 are shown in Figure 2.14(a), (b) and (c), respectively. Comparing Figure 2.14 with Figure 2.10 in Case 1 and Figure 2.12 in Case 2, we can expect that the control qualities for MFGS, FGS1 and FGS2 should be between the corresponding control qualities in CASE 1 and CASE 2. Still MFGS is expected to be the best, FGS2 is worse and FGS1 is the worst due to too large $K_c(y)$ and too small $T_i(y)$.

Now, let's briefly discuss the case in which the process time delay decreases as the process output y decreases. From Figure 2.10(a), 2.12(a) and 2.14(a), if we agree that the controller gain K_c from ITAE is the best, we can see that K_c from MFGS is the closest to K_c from ITAE, while the controller gains from FGS1 and FGS2 is too large comparatively. From 2.10(b), (c), 2.12(b), (c) and 2.14(b), (c), we can see that MFGS and FGS2 have the same T_i and T_d, which are much closer to those from ITAE compared with the T_i and T_d from FGS1. Therefore the control quality of MFGS for the set-point tracking can be expected to be the best again. Certainly, as the process time delay decreases, the control system tends to be less sensitive to the variations of the controller parameters. Hence, the control qualities of setpoint tracking for MFGS, FGS1 and FGS2 have much less difference compared with the cases in which the process time delay increases as the process output y increases.

For the case in which the process time delay increases as the process output y decreases, the controller gains from FGS1 and FGS2 are too high compared with the controller gains from ITAE and MFGS, which are shown in Figure 2.10(a), 2.12(a) and 2.14(a). As a result, the integral outputs for FGS1 and FGS2, from (2.12), may decrease too quickly, and they may make the control outputs of FGS1 and FGS2 decrease faster, or much faster than the control outputs of ITAE and MFGS. Finally the setpoint step responses for FGS1 and FGS2 will be more, or much more, over-shot than MFGS. The similar results also can be expected for the cases in which the process time delay decreases as the process output y increases. The only difference is that as the process time delay decreases, the control system tends to be less sensitive to the variations of controller parameters. Hence, the control qualities of MFGS, FGS1 and FGS2 for set-point tracking are less, or much less different compared with the cases in which the process time delay increases as the process output y decreases.

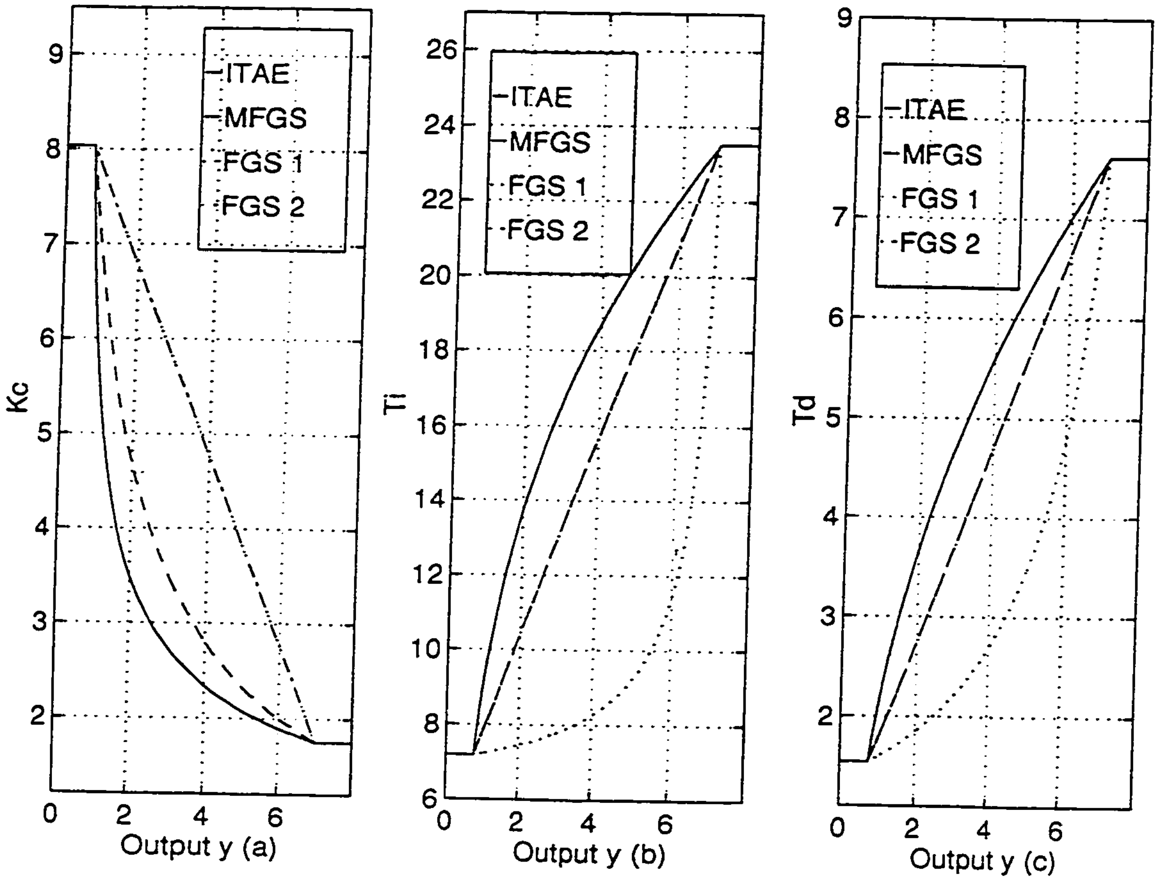


Figure 2.10 Variation of Kc as Process Time Delay Varies

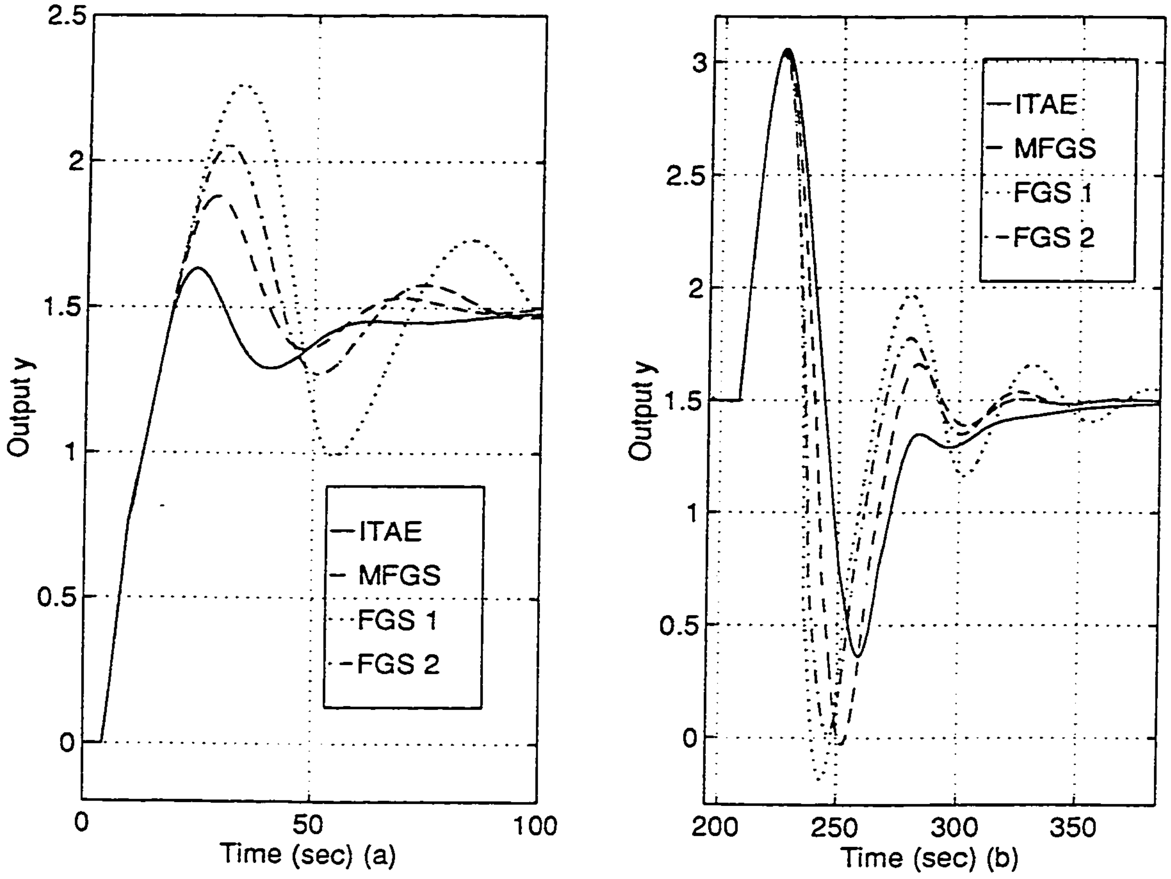


Figure 2.11 Responses as Process Time Delay Varies

(a) Setpoint Step Responses

(b) Responses for Positive Step Disturbance

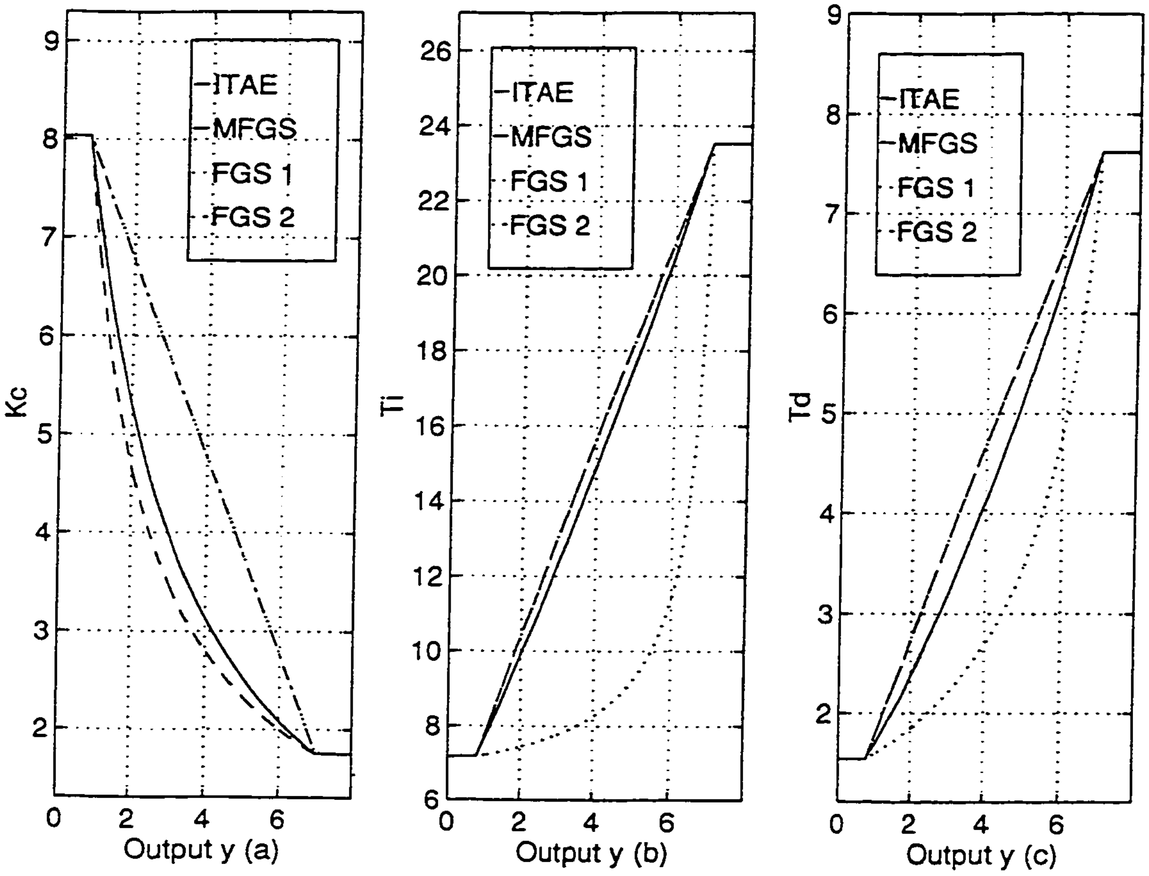


Figure 2.12 Variation of Kc as Process Time Delay Varies

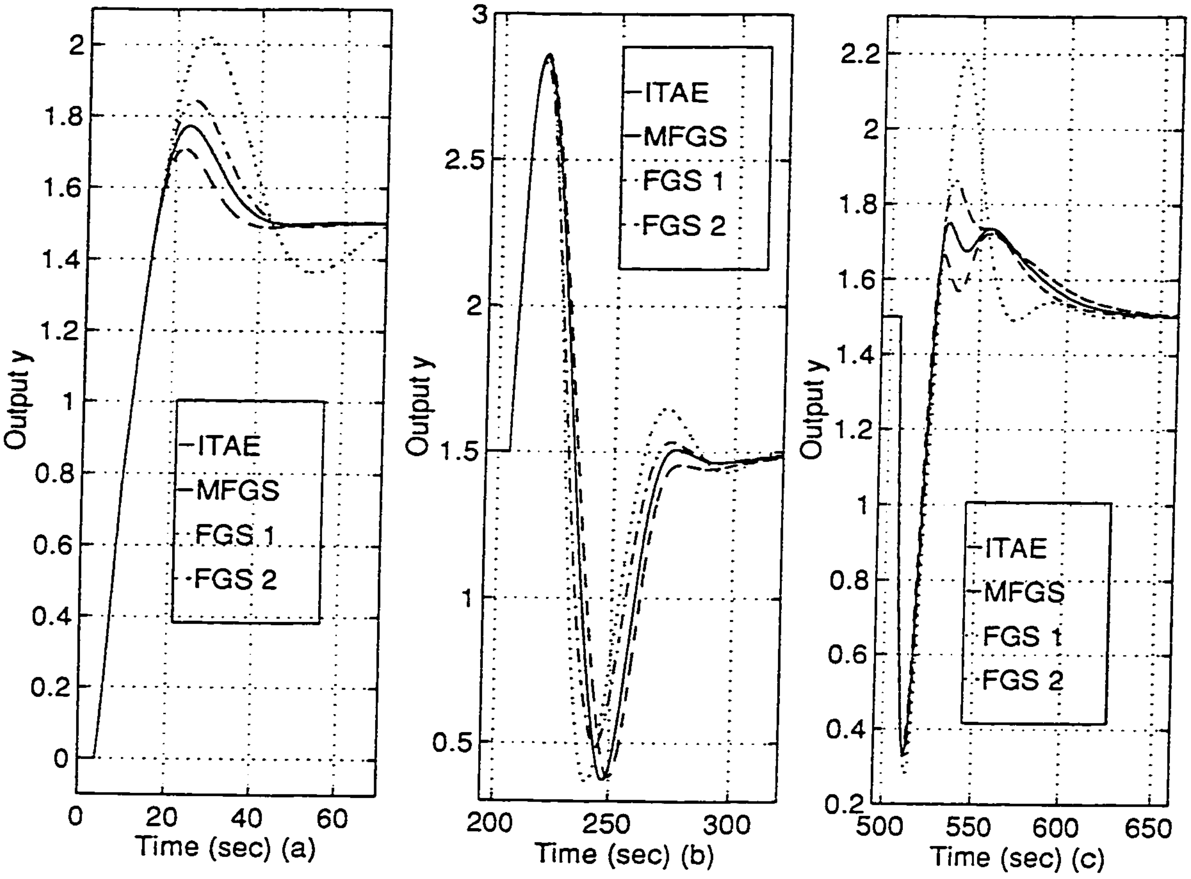


Figure 2.13 Responses as Process Time Delay Varies

(a) Setpoint Step Responses

(b) Responses for Positive Step Disturbance

(c) Responses for Negative Step Disturbance

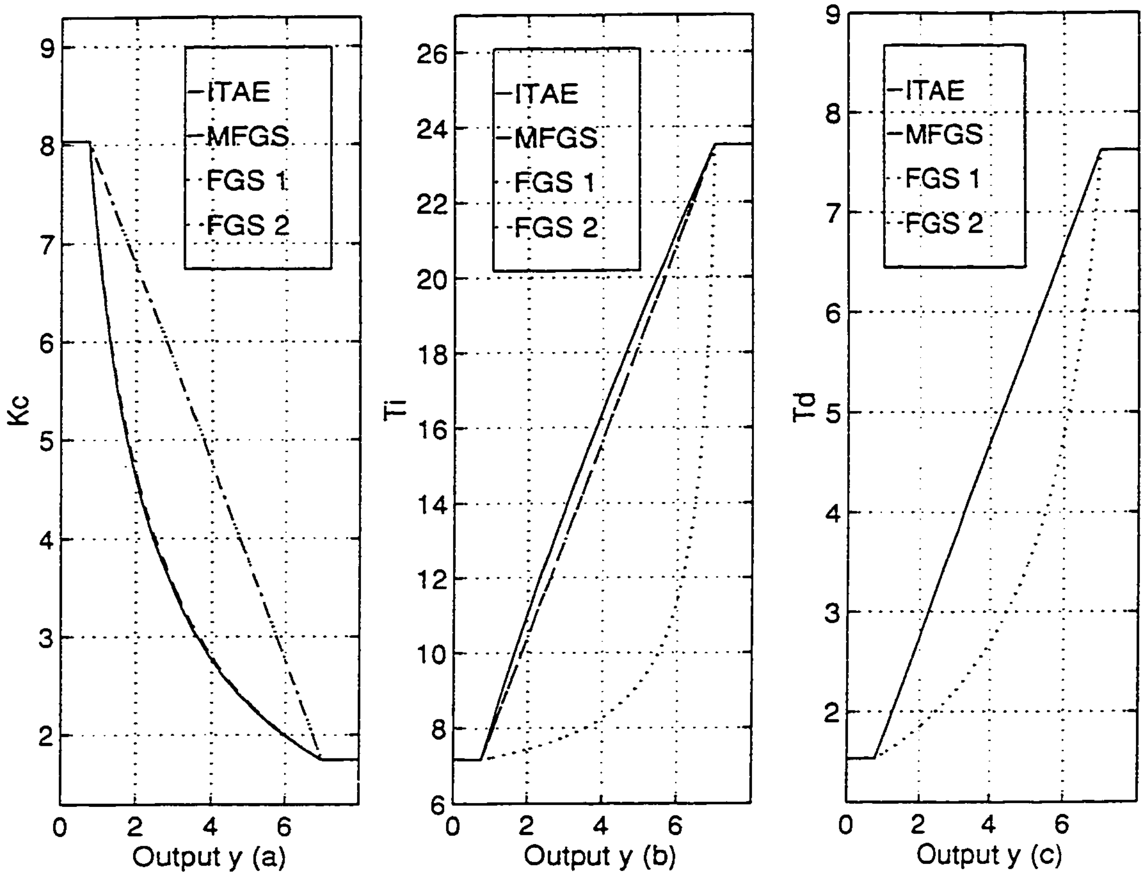


Figure 2.14 Variation of Kc as Process Time Delay Varies

2.3.4 Conclusions for Continuous-Linearly-Interpolated Gain Scheduled PID

From Simulation 2.3 to 2.5, we can make some conclusions for the single-input single-output process, whose process parameters, gain, time constant and time delay, are the linear or non-linear functions of the scheduling variable W, and can be better approximated by the first order polynomial of the scheduling variable W than the first order polynomial of W^{-1}.

1. For the variation of process gain K_p, MFGS is the definite selection. FGS1 and FGS2 are much worse, they are more likely to cause the closed-loop system to become unstable or much more damped than MFGS when the process is at the intermediate values of the scheduling variables. To achieve the desired control quality, more operating points are needed for FGS1 and FGS2.

2. For the variation of process time constant T, we can conclude that FGS1 and FGS2 can achieve better control qualities for set-point tracking, and slightly better disturbance rejection than MFGS, since $K_c(y)$ in MFGS is too small! MFGS has more damped set-point step responses than FGS1 and FGS2 when the process is running at intermediate values of the scheduling variable. Certainly, the variations of process time constants usually do influence control quality much less than the variations of process gains and time delay.

3. For the variation of process time delay τ, MFGS is the best. FGS2 is worse. FGS1 is the worst -- too large over shoot for step responses and disturbance rejection when the process is running at intermediate values of the scheduling variables.

The other issue which should be noted is that the optimal formula of ITAE is derived for optimal set-point tracking, it may not be guaranteed that the disturbance rejection is also optimal.

Chapter 3

Gray Predictive PI Controller

3.1 Gray Prediction

In order to give the readers basic concept on gray prediction, some relevant results from the references [17][18] are summarized in 3.1.1 to 3.1.5. Details and proofs may be omitted, and terminologies are directly taken from the references [17][18].

3.1.1 Basic Concept [17]

Gray System

A system is called a gray system if its information is incomplete. "Incomplete" usually means

1. The elements of system are not completely known.
2. The relationship among elements is not completely known.
3. The structure of system is not completely known.
4. The action principles of system are not completely known.

A process including "disturbances" and "noises" is a gray system, since

1. The elements (sources) of disturbances and noises usually can not be clearly known.
2. The amplitude and time of disturbances and noises are unknown.
3. The relationship between disturbances/noises and major variables can not be known completely.

From the above, we can say that a gray system is a stochastic system, and both terms are interchangeable. The gray theory deals with stochastic systems using different approaches from the stochastic theory. This can be seen in the references [17][18][23][24]. For example, the gray prediction tries to predict the future output values only using current and past output data instead of using both input and output data. We will see this in the following sections.

Gray Prediction and Black-Box Prediction

The principle of black-box prediction is to find out the "best" quantitative relation (mathematical model) between system inputs (causes) and outputs (effects) but ignore the physical structure and action principle (see system as a black box), then use this relation to predict the future outputs of the system. Therefore two data sequences are needed: inputs and outputs.

The principle of gray prediction is only to use the current and past output (effect) data to build prediction models, then use these models to predict the future output values.

Data Generating Methods of Gray System

In gray prediction, two of the most important data generating methods are Accumulated Generating Operation (AGO) and Inverse Accumulated Generating Operation (I-AGO).

AGO: Assume $e^{(0)}$ is an original positive discrete data sequence

$$e^{(0)} = (e^{(0)}(1), e^{(0)}(2), \dots , e^{(0)}(n)),\ \ e^{(0)}(k) > 0, \text{ for } k =1,2,\dots,n. \qquad (3.1a)$$

If another discrete data sequence $e^{(1)}$

$$e^{(1)} = (e^{(1)}(1), e^{(1)}(2), \dots , e^{(1)}(n)) \qquad (3.1b)$$

satisfies

$$e^{(1)}(k) = \sum_{i=1}^{k} e^{(0)}(i), \qquad \text{for } k = 1,2,\dots n, \qquad (3.2a)$$

then $e^{(1)}$ is called the First-time Accumulated Generating Operation of $e^{(0)}$, denoted as

$$e^{(1)} = \mathbf{1AGO}(e^{(0)}) \quad \text{or} \quad e^{(1)} = \mathbf{AGO}(e^{(0)}).$$

From the definition, AGO is a form of discrete integral.

I-AGO: Assume $e^{(1)}$ is an original discrete data sequence

$$e^{(1)} = (e^{(1)}(1), e^{(1)}(2), \dots , e^{(1)}(n)).$$

If another discrete data sequence $e^{(0)}$

$$e^{(0)} = (e^{(0)}(1), e^{(0)}(2), \dots , e^{(0)}(n))$$

satisfies

$$e^{(0)}(k) = e^{(1)}(k) - e^{(1)}(k-1), \quad k \geq 2, \tag{3.2b}$$

then $e^{(0)}$ is called the First-time Inverse Accumulated Generating Operation of $e^{(1)}$, denoted as

$$e^{(0)} = \mathbf{1I\text{-}AGO}(e^{(1)}) \quad \text{or} \quad e^{(0)} = \mathbf{I\text{-}AGO}(e^{(1)}).$$

From the definition, I-AGO is a form of discrete derivative or difference.

Proposition 3.1[17][18]: Assume that e is a discrete data sequence

$$e = (e(1), e(2), \dots , e(n)), \tag{3.3}$$

then the data sequence e can be described with exponential form

$$e(k) = c_1 * \exp[c_3(k-1)] + c_2. \tag{3.4a}$$

if and only if the ratio r(k),

$$r(k) = (e(k) - c_2)/(e(k+1)-c_2), \quad k \geq 1, \tag{3.4b}$$

is a constant.

From Equation (3.4a) and (3.4b), it follows that

$$r(k) = \exp(-c_3).$$

Gray Exponential Form

Given a discrete data sequence e shown in (3.3). If r(k) defined in (3.4) satisfies

$$r(k) \subseteq [\alpha, \beta] \subset (0, 1], \quad \text{for any } k = 1,2, \ldots ,n-1,$$

then we say that e has a positive gray exponential form. If β-α = q, where q is a positive real number, then we say that e has q-positive gray exponential form. If

$$r(k) \subseteq [\alpha, \beta] \subset R, \quad \alpha>1, \quad R \text{ is the real number set,}$$

then we say that e has negative gray exponential form.

Smooth Discrete Data Sequence

A discrete data sequence obtained by sampling (discretizing) a smooth function in continuous domain is called a smooth discrete data sequence.

Necessary Conditions for A Smooth Discrete Data Sequence

A necessary condition for a smooth discrete date sequence is that there exist a K > 0, such as for any k > K, we have

$$e^{(0)}(k) < \sum_{i=1}^{k-1} e^{(0)}(i) = e^{(1)}(k-1).$$

From this necessary condition for a smooth discrete data sequence, we can have the following theorem.

Note: In remaining of the thesis, it is always assumed that all given $e^{(0)}$ are non-negative smooth discrete sequence. That is, all given sequences are supposed to be discretized from smooth function in continuous time domain, and satisfy the necessary conditions for a smooth discrete data sequence presented above.

Theorem 3.1[17]**:** Given a non-negative smooth discrete data sequence $e^{(0)}$

$$e^{(0)} = (e^{(0)}(1), e^{(0)}(2), \dots , e^{(0)}(n)).$$

Its 1AGO sequence $e^{(1)}$ has 0.5-positive gray exponential form, i.e.

$$r(k) = e^{(1)}(k)/e^{(1)}(k+1) = e^{(1)}(k)/[e^{(1)}(k)+e^{(0)}(k+1)] \in [0.5\ 1].$$

Theorem 3.1 is the foundation of gray prediction. It indicates that discrete smooth sequences can be approximated by the first order exponential equations, and standard deviations of the corresponding AGO sequences $e^{(1)}$ may be reduced.

Example 3.1

$e^{(0)}$ is given as

$$e^{(0)} = (590.227, 615.369, 591.63, 651.777, 677.115,$$
$$699.870, 704.298, 695{,}565, 749.658, 816.966, 788.533, 799.5).$$

We have

$$e^{(1)} = (590.277, 1205.6, 1797.3, 2449.1, 3126.2,$$
$$3826.1, 4530.4, 5225.9, 5975.6, 6972.6, 7761.1, 8560.6).$$

$e^{(0)}$ is shown in Figure 3.1(a), and $e^{(1)}$ is shown in Figure 3.1(b). Obviously, compared with $e^{(0)}$, the standard deviation of $e^{(1)}$ is reduced.

3.1.2 Gray Model --- GM(1,1) Model

GM(1,1) Model Description

Given two stochastic discrete data sequences $e^{(0)}$ and $e^{(1)}$ shown in (3.1a) and (3.1b), and $e^{(1)} = \text{AGO}(e^{(0)})$, if $e^{(1)}$ satisfies the conditions for gray exponential form in Theorem 3.1, then it can be approximated by the 1st-order constant coefficient gray difference equation, i.e. GM(1,1) model.

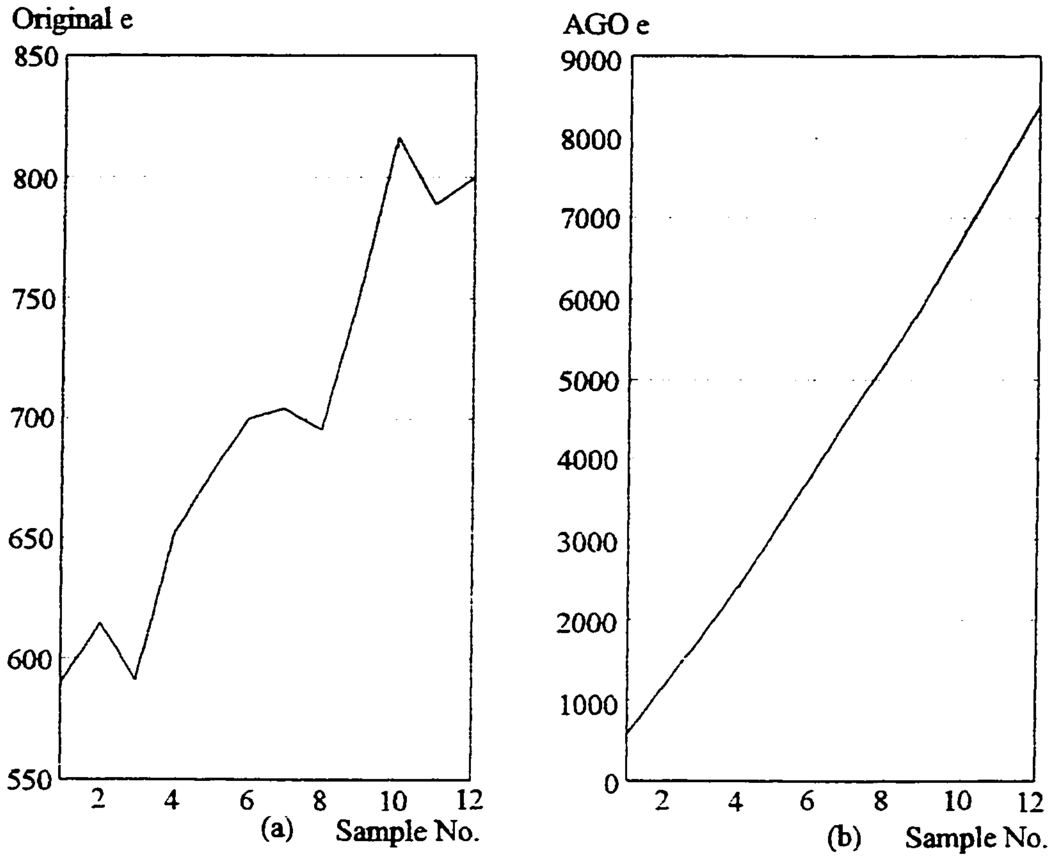


Figure 3.1 Comparison between the Original Sequence and Its AGO Sequence

$$e^{(0)}(k) + a*Z^{(1)}(k) = b, \qquad k = 2,3, \dots, n. \tag{3.5a}$$

where

$$\begin{aligned} &e^{(1)}(1) = e^{(0)}(1), \\ &e^{(1)}(k) = e^{(0)}(k) - e^{(1)}(k-1), \ k=2,3,\dots,n, \\ &Z^{(1)}(k) = 0.5(\, e^{(1)}(k) + e^{(1)}(k-1)\,), \ \ k=2,3,\dots,n. \end{aligned} \tag{3.5b}$$

a and b are the model parameters to be identified.

In the literature [17][18], although the author had not given any explicit mathematical derivation of GM(1,1) model, from Theorem 3.1, we can think that if we use the least-square method to obtain a constant C which is an optimal approximate of $r(k) = e^{(1)}(k)/e^{(1)}(k+1)$, $k=2,\dots,n-1$, then from Proposition 3.1, $e^{(1)}(k)$ can be approximated by

$$e^{(1)}(k) = c_1*\exp[c_3(k-1)] + c_2.$$

which is of the form of the discrete general solution of the first order constant coefficient differential equation

$$de^{(1)}(t)/dt + a_1*e^{(1)}(t) = b_1. \tag{3.5c}$$

where

$$e^{(1)}(t) = \int e^{(0)}(t)dt$$

If discretizing it with the forward rectangular transform, from the definition of AGO and IAGO, we have

$$e^{(0)}(k) + a_2 * e^{(1)}(k) = b_2, \quad k = 2, 3, \dots . \qquad (3.5d)$$

Instead, if Equation (3.5c) is discretized with the bilinear transform, we obtain (3.5a), i.e.

$$e^{(0)}(k) + a * Z^{(1)}(k) = b, \qquad k = 2,3, \dots, n.$$

In this thesis, we will use the form (3.5a) as GM(1,1) model.

Dynamic GM(1,1) Model Description

Let $\{1_i, 2_i, \dots, n_i\}$ denote the sample times of ith time window, $e_i^{(0)}$ the ith original sample data sequence

$$e_i^{(0)} = (e^{(0)}(1_i), e^{(0)}(2_i), \dots, e^{(0)}(n_i)).$$

For any $i \geq 1$, if we have

$$\{1_i, 2_i, \dots, n_i\} \cap \{1_{i+1}, 2_{i+1}, \dots, n_{i+1}\}$$
$$= \{2_i, 3_i, \dots, n_i\} = \{1_{i+1}, 2_{i+1}, \dots, (n-1)_{i+1}\},$$

i.e.

$$2_i = 1_{i+1}, 3_i = 2_{i+1}, \dots, n_i = (n-1)_{i+1},$$

then $e_i^{(0)}$ is called a n-dimension sequence or moving window sequence. If $e_i^{(1)}$=AGO($e_i^{(0)}$) satisfies the conditions for gray exponential form in Theorem 3.1, then it can be approximated by the dynamic GM(1,1) model

$$e^{(0)}(k_i) + a(i) * Z^{(1)}(k_i) = b(i), \quad k = 2,3, \dots, n, \qquad (3.5e)$$

where

$$e^{(1)}(1_i) = e^{(0)}(1_i),$$
$$e^{(1)}(k_i) = e^{(0)}(k_i) + e^{(1)}(k_i\text{-}1), \ \ k=2,3,...n,$$
$$Z^{(1)}(k_i) = 0.5(e^{(1)}(k_i) + e^{(1)}((k\text{-}1)_i), \ \ k=2,3,...,n. \qquad (3.5f)$$

a(i) and b(i) are the identified model parameters, and k_i and $(k\text{-}1)_i$ denote the time k and k-1 at the ith time window, respectively.

Theorem 3.2: The least-square solution of the dynamic GM(1,1) model (3.5e) exists for the ith time window. The solution is

$$[a(i) \ \ b(i)]^T = (B(i)^T B(i))^{-1} B(i)^T X(i)$$

where

$$B(i) = \begin{vmatrix} -Z^{(1)}(2_i) & 1 \\ -Z^{(1)}(3_i) & 1 \\ . & . \\ . & . \\ & . \\ -Z^{(1)}(n_i) & 1 \end{vmatrix}$$

$$X(i) \ = [e^{(0)}(2_i) \ \ e^{(0)}(3_i) \ \dots \ e^{(0)}(n_i)]^T.$$

Extending the above solution and rearranging the items, we have the following equivalent solution

$$a(i) = A(i)/C(i), \ \ b(i) = B(i)/C(i), \qquad (3.6a)$$

in which

$$A(i) = \sum_{k=2}^{n} Z^{(1)}(k_i) * \sum_{k=2}^{n} e^{(0)}(k_i) - (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i),$$

$$B(i) = \sum_{k=2}^{n} (Z^{(1)}(k_i))^2 \sum_{k=2}^{n} e^{(0)}(k_i) - \sum_{k=2}^{n} Z^{(1)}(k_i) \sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i)$$

$$C(i) = (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)^2 - (\sum_{k=2}^{n} Z^{(1)}(k_i))^2, \qquad (3.6b)$$

Remark: From this theorem, we know that the gray prediction uses a time-varying model or time-varying parameters a(i) and b(i) to approximate a process, no matter whether the process is time-varying or not.

3.1.3 Gray Prediction

From the dynamic GM(1,1) model, described by (3.5e) and (3.5f), we have

$$e^{(1)}(k_i) - e^{(1)}((k-1)_i) + a(i) * 0.5 * (e^{(1)}(k_i) + e^{(1)}((k-1)_i)) = b(i),$$
$$k = 2,3, ..., n.$$

It can be further written as

$$(1 + 0.5a(i)) * e^{(1)}(k_i) - (1 - 0.5a(i)) * e^{(1)}((k-1)_i) = b(i),$$
$$k = 2,3, ..., n, ... , \qquad (3.6c)$$

with solution of general form

$$e^{(1)}(k_i) = c_1(i)r(i)^{(ki-1)} + c_2(i). \tag{3.6d}$$

Solving Equation (3.6c) and (3.6d), we get

$$c_2 = b(i)/a(i), \qquad \text{and}$$
$$r(i) = (\ 1 - 0.5a(i)\ )/(\ 1 + 0.5a(i)\ ),$$

then applying the initial condition $e^{(1)}(1_i) = e^{(0)}(1_i)$, we have

$$c_1 = (\ e^{(0)}(1_i) - b(i)/a(i)\ ).$$

Thus, we can write the solution of GM(1,1) model (3.5e) and (3.5f) at the ith window as

$$e^{(1)}(k_i) = \ (\ e^{(0)}(1_i) - b(i)/a(i)\ ) * r(i)^{\ (ki-1)} + b(i)/a(i),\ \ k = 2,3, \ldots, n, \ldots .$$
$$r(i) = (\ 1 - 0.5a(i)\ )/(\ 1 + 0.5a(i)\ ), \qquad |\,a(i)\,| < 2, \tag{3.7a}$$

where $|\ a(i)\ |$ is the absolute value of a(i). Using the inverse accumulated generating operation I-AGO, the predictive algorithm of $e_i^{(0)}$ has the form

$$e^{(0)}(k_i) = e^{(1)}(k_i) - e^{(1)}((k-1)_i),$$
$$= [\ (\ b(i) - a(i) * e^{(0)}(1_i)\ )/(1 - 0.5a(i)\ )\ ] * r(i)^{\ (ki-1)},\ \ k \geq n + 1. \tag{3.7b}$$

in which r(i) is shown in (3.7a). Certainly, if the sample interval is small enough, i.e. $|a(i)k_i|$ << 1, the predictive algorithm also can be well approximated by the discretized solution of the first order constant differential equation. That is

$$e^{(0)}(k_i) = (\ b(i) - a(i) * e^{(0)}(1_i)\ ) * (1 - \exp(\ a(i)\ ) * \exp(-a(i) * (k_i\text{-}1)\ ),$$
$$k \geq n + 1. \tag{3.7c}$$

Remark 1: The gray predictive algorithm (3.7b) can be used only for $| a(i) | < 2$.

Remark 2: When $k = 2,3,...,n$, what we get from (3.7a) and (3.7b) are estimates of the original sequence $e^{(0)}$ in the ith time window. If we want to predict $e^{(0)}$, use $k \geq n+1$.

Remark 3: In dynamic GM(1,1) model, first we calculate the predictive values of AGO sequence $e_i^{(1)}$ using a(i) and b(i), then use the predictive values $e_i^{(1)}$ and I-AGO to obtain the predictive values of the original sequence $e_i^{(0)}$. This is the special feature of gray prediction.

Remark 4: To build the dynamic GM(1,1) model, only 4 original data are needed. Using a few data to build predictive model is one of important feature of gray prediction. For a gray predictive control system, it is suggest to use 4 to 6 data [18].

Simulation 3.2: Use the dynamic GM(1,1) model (3.5e) and (3.5f) to predict the future values of the given original discrete data sequence $e^{(0)}$.

$$\begin{aligned} e^{(0)} = (& e^{(0)}(1), e^{(0)}(2), e^{(0)}(3), e^{(0)}(4), e^{(0)}(5), e^{(0)}(6), e^{(0)}(7), \\ & e^{(0)}(8), e^{(0)}(9), e^{(0)}(10), e^{(0)}(11), e^{(0)}(12)) \\ = (& 590.227, 615.369, 591.63, 651.777, 677.115, 699.87, 704.298, \\ & 695.565, 749.685, 816.966, 788.533, 799.5). \end{aligned}$$

CASE 1: Use the 1-st time window with $n = 7$ and $e^{(0)}(1_1) = 590.227$, i.e.

$$\begin{aligned} e_1^{(0)} &= (e^{(0)}(1_1), e^{(0)}(2_1), e^{(0)}(3_1), e^{(0)}(4_1), e^{(0)}(5_1), e^{(0)}(6_1), e^{(0)}(7_1)) \\ &= (e^{(0)}(1), e^{(0)}(2), e^{(0)}(3), e^{(0)}(4), e^{(0)}(5), e^{(0)}(6), e^{(0)}(7)) \\ &= (590.227, 615.369, 591.63, 651.777, 677.115, 699.87, 704.298), \end{aligned}$$

to predict $e^{(0)}(8)$, $e^{(0)}(9)$, $e^{(0)}(10)$, $e^{(0)}(11)$ and $e^{(0)}(12)$.

We can easily get $e_1^{(1)}$ and $Z_1^{(1)}$ from the definition of AGO and (3.5f).

$$\begin{aligned} e^{(1)} &= (e^{(1)} (1_1), e^{(1)} (2_1), e^{(1)} (3_1), e^{(1)} (4_1), e^{(1)} (5_1), e^{(1)} (6_1), e^{(1)} (7_1)) \\ &= (590.277,\ 1205.596,\ 1797.226,\ 2449.003,\ 3126.118,\ 3825.988,\ 4530.286), \end{aligned}$$

$$\begin{aligned} Z_1^{(1)} &= (Z^{(1)} (1_1), Z^{(1)} (2_1), Z^{(1)} (3_1), Z^{(1)} (4_1), Z^{(1)} (5_1), Z^{(1)} (6_1), Z^{(1)} (7_1)) \\ &= (590.277,\ 897.912,\ 1501.411, 2123.115, 2787.561, 3476.053, 4178.137). \end{aligned}$$

By (3.6) we have

$$a(1) = -0.0345, \qquad b(1) = 570.5639,$$

so that we can get the estimated values of $e^{(0)} (k_1)$, $k \le 7$, and predictive values of $e^{(0)} (k_1)$, $k = 8,9...,12$, in the 1-st time window using (3.7a) and (3.7b),.

$$\begin{aligned} e^{(0)\wedge} = (&590.227, 615.369, 622.3441, 644.1897, 666.8020, 690.2081, \\ &714.4358, 739.5140, 765.471, 792.3421, 820.1549, 848.944). \end{aligned}$$

12 actual values of the given original discrete data sequence $e^{(0)}$, and its estimated or predictive values in the 1st time window are shown in Figure 3.2. We can see that the gray prediction gives much more smooth estimated and predictive values of $e^{(0)}$.

CASE 2: Use the 2nd time window with $n = 7$ and $e_2^{(0)}(1_2) = 615.369$, i.e.

$$\begin{aligned} &(e^{(0)} (1_2), e^{(0)} (2_2), e^{(0)} (3_2), e^{(0)} (4_2), e^{(0)} (5_2), e^{(0)} (6_2), e^{(0)} (7_2)) \\ &= (e^{(0)} (2), e^{(0)} (3), e^{(0)} (4), e^{(0)} (5), e^{(0)} (6), e^{(0)} (7), e^{(0)} (8)) \\ &= (615.369, 591.63, 651.777, 677.115, 699.87, 704.298, 695.565), \end{aligned}$$

to predict $e^{(0)}(9)$, $e^{(0)}(10)$, $e^{(0)}(11)$ and $e^{(0)}(12)$.

We can easily get $e_1^{(1)}$ and $Z_1^{(1)}$ from the definition of AGO and (3.5d).

$$e^{(1)} = (e^{(1)}(1_2), e^{(1)}(2_2), e^{(1)}(3_2), e^{(1)}(4_2), e^{(1)}(5_2), e^{(1)}(6_2), e^{(1)}(7_2))$$
$$= (615.369, 1206.999, 1858.776, 2535.891, 3235.761, 3940.059, 4635.624),$$

$$Z_1^{(1)} = (Z^{(1)}(1_2), Z^{(1)}(2_2), Z^{(1)}(3_2), Z^{(1)}(4_2), Z^{(1)}(5_2), Z^{(1)}(6_2), Z^{(1)}(7_2))$$
$$= (615.369, 911.184, 1532.888, 2197.334, 2885.826, 3587.91, 4287.842).$$

Replacing them in the solution (3.6), we have

$$a(2) = -0.0291, \qquad b(2) = 595.2835.$$

Figure 3.3 shows the result. Again, the gray prediction gives much more smooth estimated and predictive values.

It also can be seen, from both cases, that the gray prediction uses time-varying GM(1,1) model to approximate stochastic systems at the neighborhood of equilibrium points.

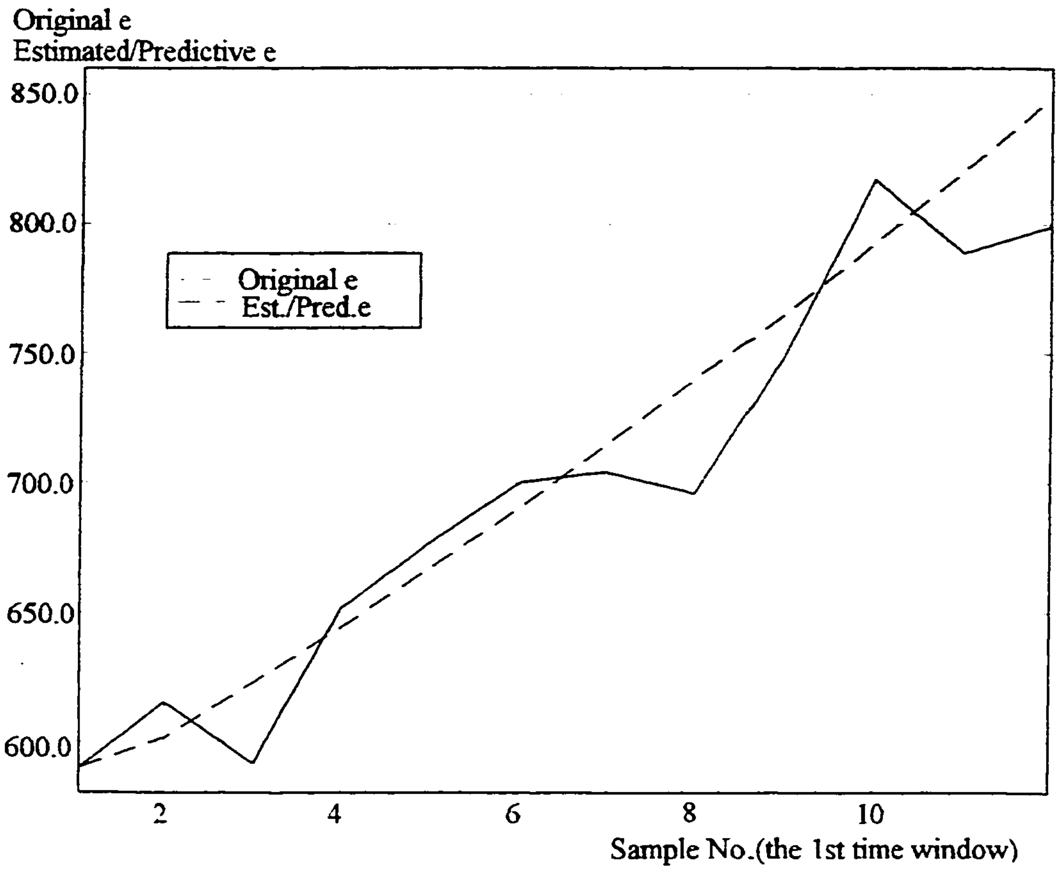


Figure 3.2 Gray Prediction --- Part 1

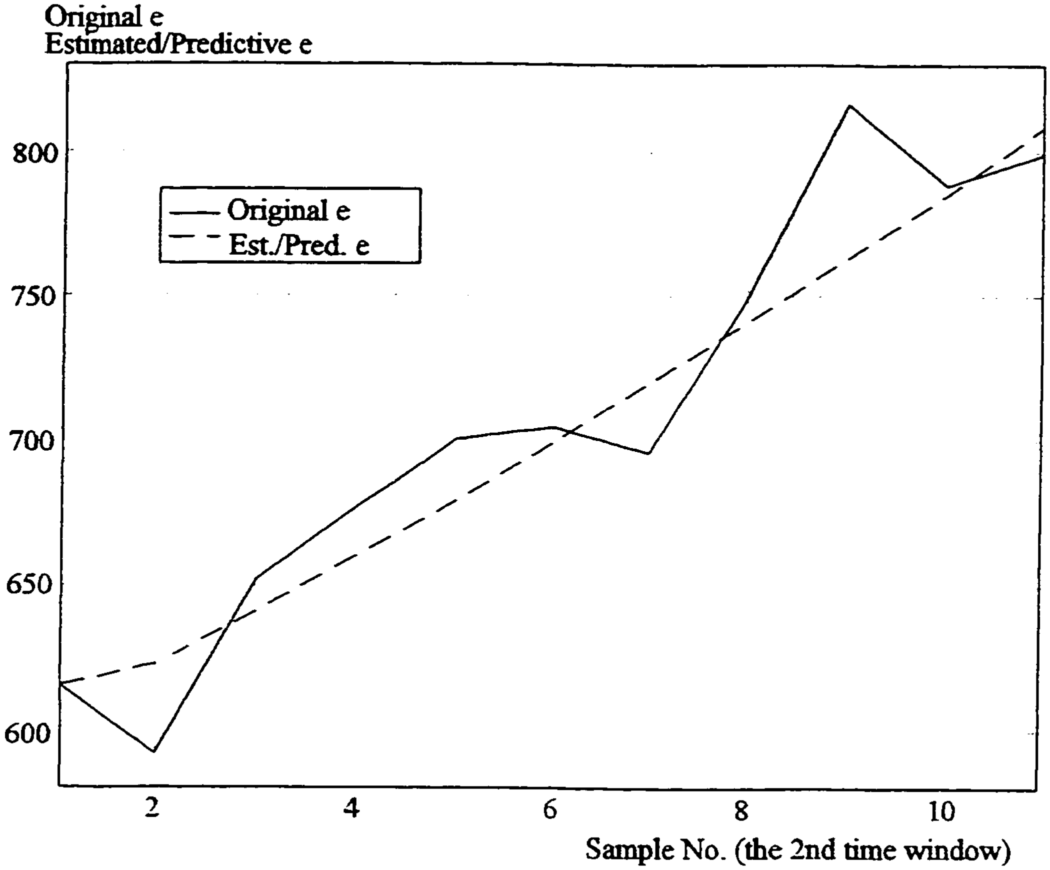


Figure 3.3 Gray Prediction --- Part 2

3.1.4 Discussion on Linear System [18]

In this section, we will discuss some issues about gray prediction and linear systems.

As indicated above, the GM(1,1) model is the first order difference equation which is used to approximate AGO, or discrete integral, sequence $e^{(1)}$ of the original sequence $e^{(0)}$. Is it reasonable for linear systems? We will discuss this from the point view of differential equations.

Given a first order constant coefficient equation

$$de(t)/dt + pe(t) = q, \tag{3.8a}$$

its general solution for $t_0 = 0$ can be written as

$$\begin{aligned} &e(t) = C_1\exp(-pt) + C_2, \quad \text{and} \\ &C_1 = - e'(0)/p, \\ &C_2 = e(0) + e'(0)/p. \end{aligned} \tag{3.8b}$$

Extending it into Taylor series, we have

$$e(t) = C_1[1 - pt + (2!)^{-1}(-pt)^2 + ...] + C_2.$$

For $t \in [0, \delta]$, if δ is small enough, the series above can be approximated by

$$e(t) = C_1(1 - pt) + C_2 = C_3[1 - (C_1p/C_3)t], \tag{3.9}$$

where

$$C_3 = C_1 + C_2.$$

Similarly, we can write the approximate solution of GM(1,1) model, (3.7c), in continuous time domain as

$$e^{(0)}(t) = (\ e^{(0)}(0) - b(i)/a(i)\ ) * (\ 1 - \exp(a_i\Delta t)\ ) * \exp(-a(i) * t\ ). \qquad (3.10)$$

Denote solution (3.10) as

$$e^{(0)}(t) = C_3\exp(-a(i)*t).$$
$$C_3 = (\ e^{(0)}(0) - b(i)/a(i)\ )*[1 - \exp(a(i)\Delta t)],$$
$$a(i) = C_1 p/C_3, \qquad (3.11)$$

It is obvious that Equation (3.11) can be approximated by Equation (3.9) for any $t \in [0, \delta]$, where δ is small enough. That is, any given first order differential equation (3.8a), which has solution (3.8b), can be approximated in the neighborhood of equilibrium points by the GM(1,1) model with solution (3.7b) or (3.10).

Remark: the GM(1,1) model approximates the first order differential equations. It does not reproduce the first order differential equations even in continuous time domain.

Now let's consider a high order constant coefficient equation

$$a_n d^n e(t)/dt^n + a_{n-1} d^{n-1} e(t)/dt^{n-1} + \dots + a_0 e(t) = 0. \qquad (3.12a)$$

with the characteristic equation

$$(s+s_1)^2(s_2+2\zeta\omega s+\omega^2)(s+s_4)...(s+s_n) = 0. \qquad (3.12b)$$

Its general solution is

$$e(t) =(A+Bt)\exp(-s_1t) + \exp(-\zeta\omega t)[C\sin(Q\omega t) + D\cos(Q\omega t)]$$
$$+E_4\exp(-s_4t) +...+E_n\exp(-s_nt), \qquad (3.13a)$$

where

$$Q = \mathrm{sqrt}(1-\zeta^2), \qquad (3.13b)$$

A, B, C, D, E_4,...E_n are constants determined by given initial conditions.

If e(t) in (3.13) can be approximated by (3.9) for t ∈ [0, δ], where δ is small enough, then processes with the form (3.14) can be approximated by the first order process shown in (3.15) in the neighborhood of equilibrium points. Further more, the high order processes with characteristic equation (3.12b) can be approximated by the GM(1,1) model in the neighborhood of equilibrium points.

$$a_nd^ne(t)/dt^n + a_{n-1}d^{n-1}e(t)/dt^{n-1} + ... + a_0e(t) = b_0u(t). \qquad (3.14)$$
$$de(t)/dt + pe(t) = b_1u(t). \qquad (3.15)$$

Theorem 3.3[18]: Given any t ∈ [0, δ], where δ satisfies $|s_1\delta| << 1$, $|\zeta\omega\delta| << 1$, $|Q\omega\delta| << 1$, $|s_4\delta| << 1$, ..., and $|s_n\delta| << 1$, the solution (3.13) can be approximated by

$$e(t) = [A+D+(E_4+...+E_n)] - [As_1+(E_4s_4+...+E_ns_n)-B-CQ\omega + \zeta\omega D]t.$$

i.e. the process described by (3.14) can be approximated by the first order process described by (3.15).

Proof [18]: Equation (3.13a) can be extended into

$$\begin{aligned}
e(t) = \; & A[1 + (-s_1t) + (2!)^{-1}(-s_1t)^2 + \dots] \\
& + Bt[1 + (-s_1t) + (2!)^{-1}(-s_1t)^2 + \dots] \\
& + C\sin(Q\omega t)[1 + (-\zeta\omega t) + (2!)^{-1}(-\zeta\omega t)^2 + \dots] \\
& + D\cos(Q\omega t)[1 + (-\zeta\omega t) + (2!)^{-1}(-\zeta\omega t)^2 + \dots] \\
& + E_4[1 + (-s_4t) + (2!)^{-1}(-s_4t)^2 + \dots] + \dots \\
& + E_n[1 + (-s_nt) + (2!)^{-1}(-s_nt)^2 + \dots],
\end{aligned}$$

in which Q is shown (3.13b) and $t \in [0, \delta]$. For a small enough δ, which satisfies $|s_1\delta| << 1$, $|\zeta\omega\delta| << 1$, $|Q\omega\delta| << 1$, $|s_4\delta| << 1$, ..., and $|s_n\delta| << 1$, the above equation can be approximated by

$$\begin{aligned}
e(t) &= A(1-s_1t) + Bt + CQ\omega t + D(1-\zeta\omega t) + E_4(1-s_4t) + \dots + E_n(1-s_nt) \\
&= A + D + E_4 + \dots + E_n - [As_1 + (E_4s_4 + \dots + E_ns_n) - B - CQ\omega + \zeta\omega D]t.
\end{aligned}$$

Now from Theorem 3.3, if we denote

$$\begin{aligned}
C_3 &= A + D + E_4 + \dots + E_n, \\
p &= As_1 + (E_4s_4 + \dots + E_ns_n) - B - CQ\omega + \zeta\omega D,
\end{aligned}$$

then for $t \in [0, \delta]$, the solution (3.13a) for a high order process can also be approximated by Equation (3.9), therefore any high order process (3.12a) with characteristic equation (3.12b) can be approximated by the GM(1,1) model in the neighborhood of equilibrium points.

3.1.5 Simulation and Analysis

In the following simulations, we will use the dynamic GM(1,1) model shown in (3.5e) and (3.5f) to approximate first order systems and second order systems with stable eigenvalues, then some important features and analysis are given.

Set sample interval $T_s = 1$ sec, and window dimension $n = 4$, i.e. in each time window, i.e. that 4 sample data are used to identify GM(1,1) model. Prediction horizon 20 sec, 30 sec and 40 sec are used to predict process outputs, i.e. we will predict the process output 20 sec, 30 sec and 40 sec ahead using the predictive algorithm (3.7b) of GM(1,1) model.

First the general procedure for all simulations is given in the following.

Suppose in the ith time window, we have 4 most recent process output data, denoted as

$$y_i^{(0)} = (\ y^{(0)}(1_i),\ y^{(0)}(2_i),\ y^{(0)}(3_i),\ y^{(0)}(4_i)\ ).$$

where $y^{(0)}(4_i)$ is the current process output. From $y_i^{(0)}$, we can calculate $y_i^{(1)} = \text{AGO}(\ y_i^{(0)}\ )$ with

$$y_i^{(1)} = (\ y^{(1)}(1_i),\ y^{(1)}(2_i),\ y^{(1)}(3_i),\ y^{(1)}(4_i)\ ),$$

where

$$y^{(1)}(1_i) = y^{(0)}(1_i),$$
$$y^{(1)}(2_i) = (\ y^{(0)}(2_i) + y^{(1)}(1_i)\ ),$$

$$y^{(1)}(3_i) = (y^{(0)}(3_i) + y^{(1)}(2_i)),$$
$$y^{(1)}(4_i) = (y^{(0)}(4_i) + y^{(1)}(3_i)).$$

then the GM(1,1) model can be built by substituting e with y, and n with 4 in (3.5e) and (3.5f).

$$y^{(0)}(k_i) + a_i*Z^{(1)}(k_i) = b_i, \quad k = 2,3,4,$$

in which

$$Z^{(1)}(2_i) = 0.5*(y^{(1)}(2_i) + y^{(1)}(1_i)),$$
$$Z^{(1)}(3_i) = 0.5*(y^{(1)}(3_i) + y^{(1)}(2_i)),$$
$$Z^{(1)}(4_i) = 0.5*(y^{(1)}(4_i) + y^{(1)}(3_i)).$$

Using (3.6a) and (3.6b), and substituting e with y, and n with 4, we can get parameters of the GM(1,1) model, a(i) and b(i), for the ith time window, then obtain predictive algorithms for 20 sec, 30 sec and 50 sec prediction horizon by (3.7b). For example, if prediction horizon is 20 sec, predictive algorithm for the ith time window, denoted as $y^{(0)}(24_i)$, is

$$y^{(0)}(24_i) = y^{(1)}(24_i) - y^{(1)}(23_i)$$
$$= [(b(i) - a(i) * e^{(0)}(1_i))/(1 - 0.5a(i))] * r(i)^{23}, \quad k \geq n + 1,$$

where r(i) is given in (3.7a). Note the time index 4_i represents the current time and the sample time is 1 sec, so that $y^{(1)}(23_i)$ and $y^{(1)}(24_i)$ are the 19th and 20th predictive data of $y_i^{(1)}$, and $y^{(0)}(24_i)$ is the 20th predictive value of $y_i^{(0)}$. At each time window, only one predictive value is used, therefore, the predictive curves in all simulations are drawn by using $y^{(0)}(24_1)$, $y^{(0)}(24_2)$, ..., $y^{(0)}(24_n)$, ...

Simulation 3.3: Gray Prediction For The 1st Order Processes

Process transfer function is

$$Y(s)/U(s) = G_1(s) = 1/(40s + 1).$$

U(s) is the transform of an unit step input signal u(t).

$$u(t) = 2*1(t) + 1(t-250) -1(t-470), \quad t \geq 0.$$

CASE 1: Prediction horizon 20 sec.

CASE 2: Prediction horizon 30 sec.

CASE 3: Prediction horizon 40 sec.

Figure 3.4(a), 3.5(a) and 3.6(a) show the predictive outputs compared with the actual process outputs for CASE 1, CASE 2 and CASE 3, respectively. Figure 3.4(b) and 3.4(c), 3.5(b) and 3.5(c), and 3.6(b) and 3.6(c) show the dynamic a(i) and b(i) from the gray predictors, respectively. a(i) for each case is not equal to the reciprocal of process time constant.

All three cases demonstrate other two common features.

1. The time that predictive outputs are ahead of process outputs is not equal to the prediction horizon.

2. Prediction is more aggressive when the process outputs increase than when they decrease. Predictive effects at two directions are different.

For example, in Figure 3.4, at about t = 253 sec, as the process output increases, the gray predictive output is about 34 sec ahead of the process output, and at about t = 473 sec, as the process output decreases, the gray predictive output is about 26 sec ahead of the process output, while both prediction horizons are 20 sec. In Figure 3.6, there exists large over-shoot

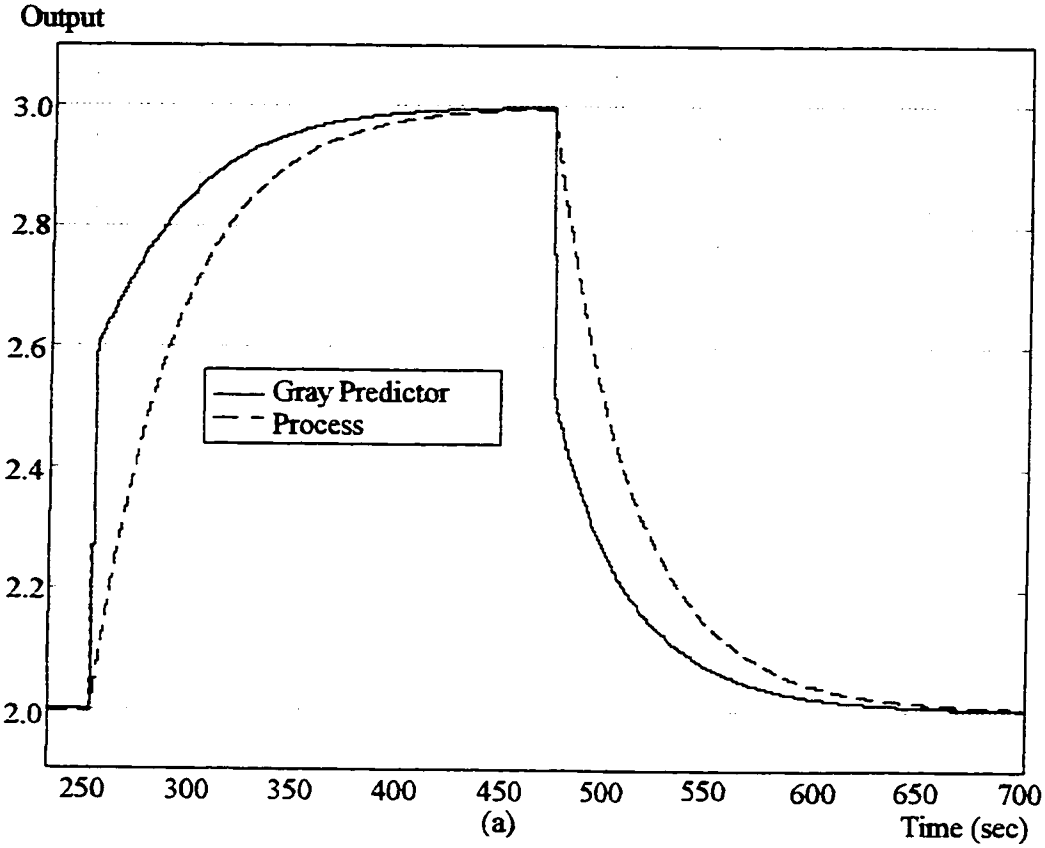


Figure 3.4 Gray Prediction --- The First Order Process (Con't)

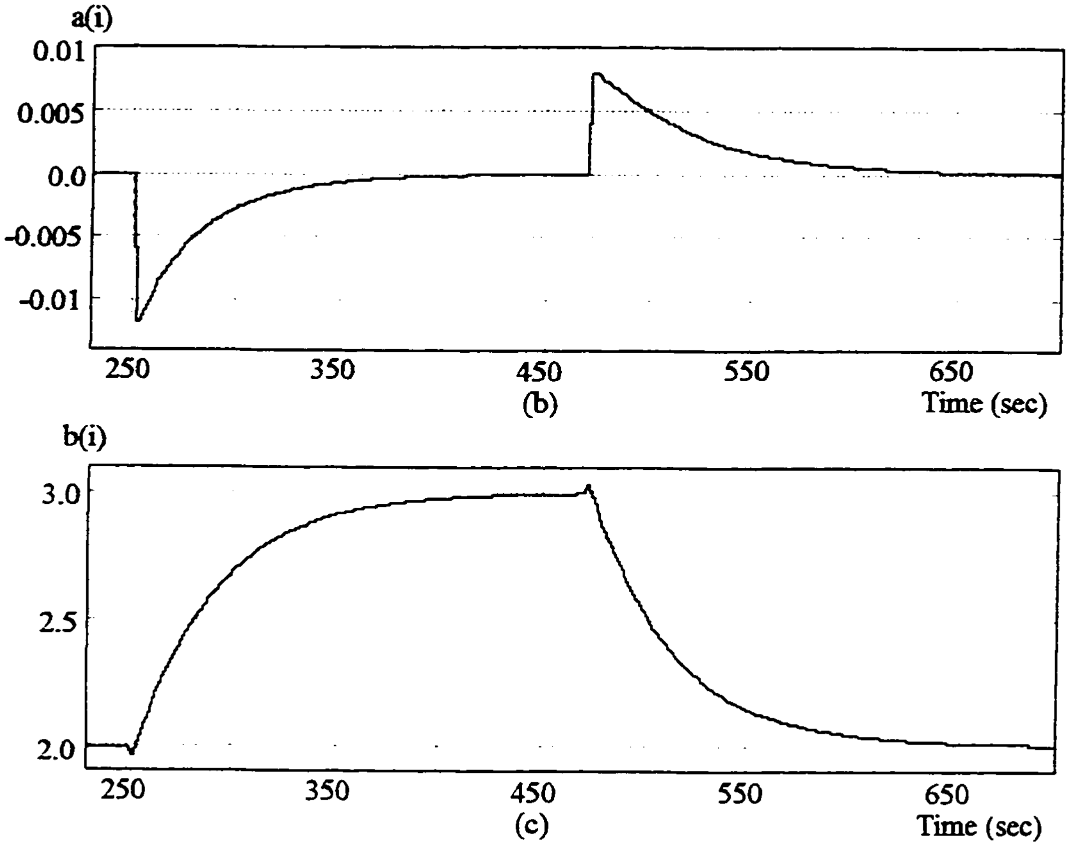


Figure 3.4 Gray Prediction --- The First Order Process

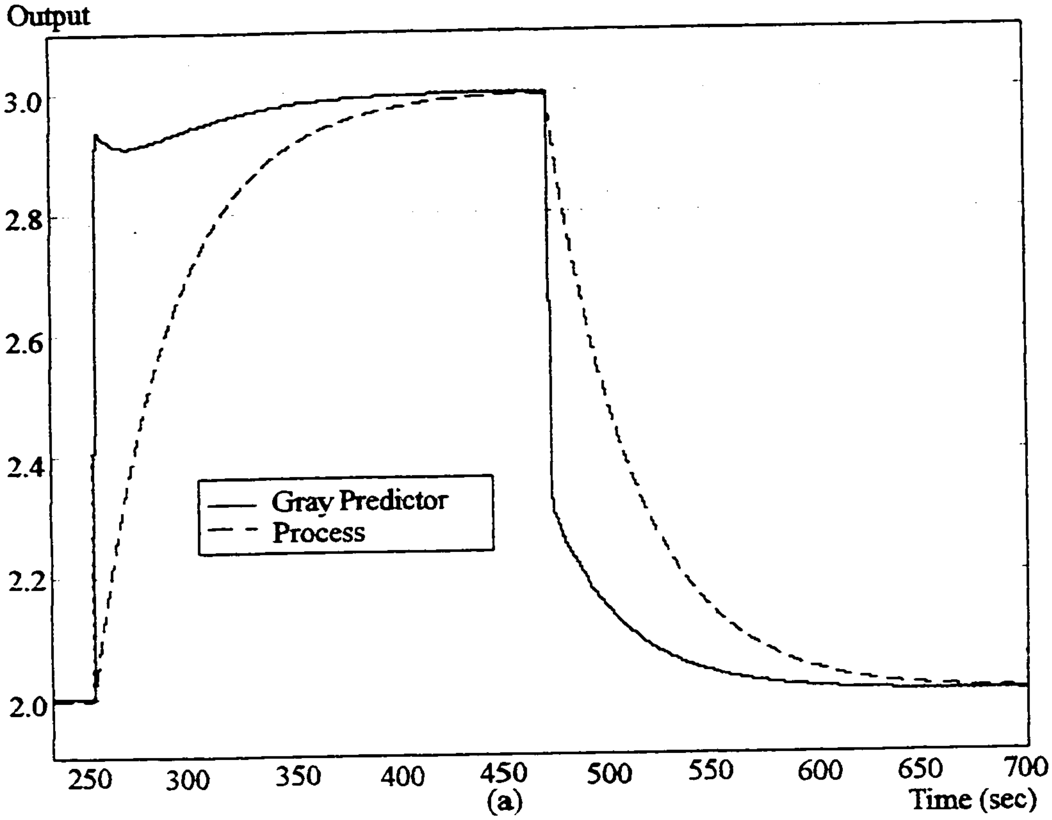


Figure 3.5 Gray Prediction — The First Order Process (Con't)

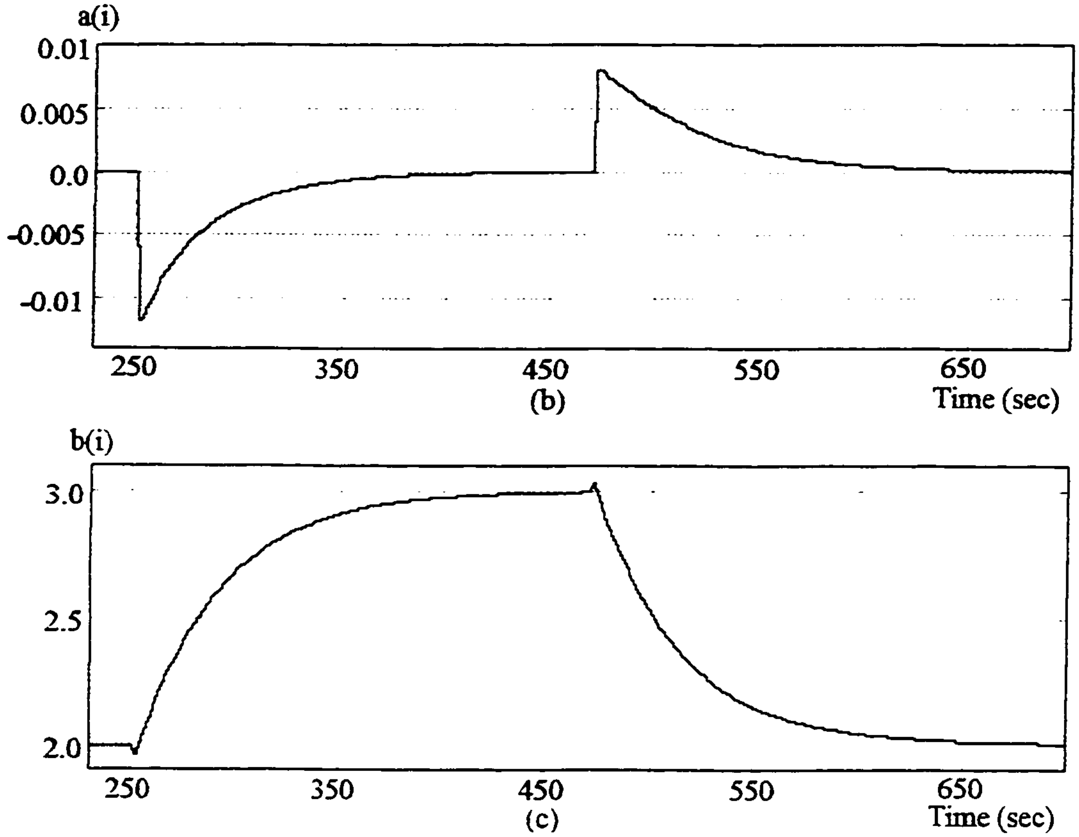


Figure 3.5 Gray Prediction — The First Order Process

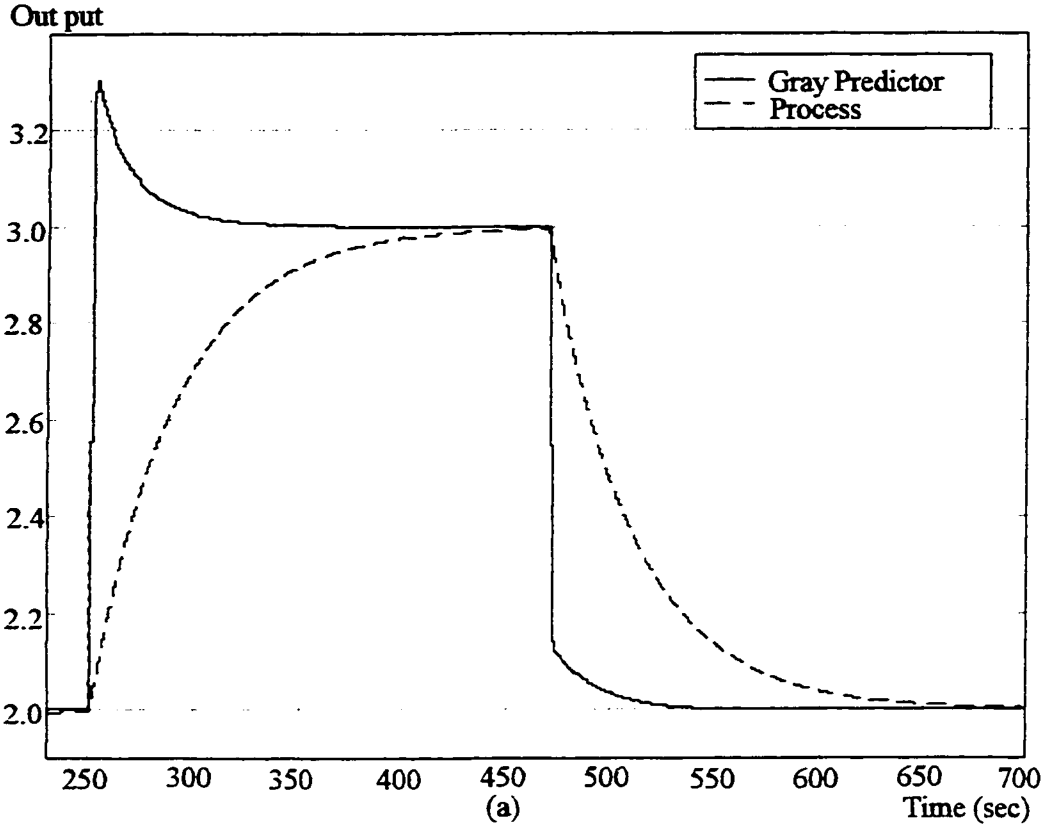
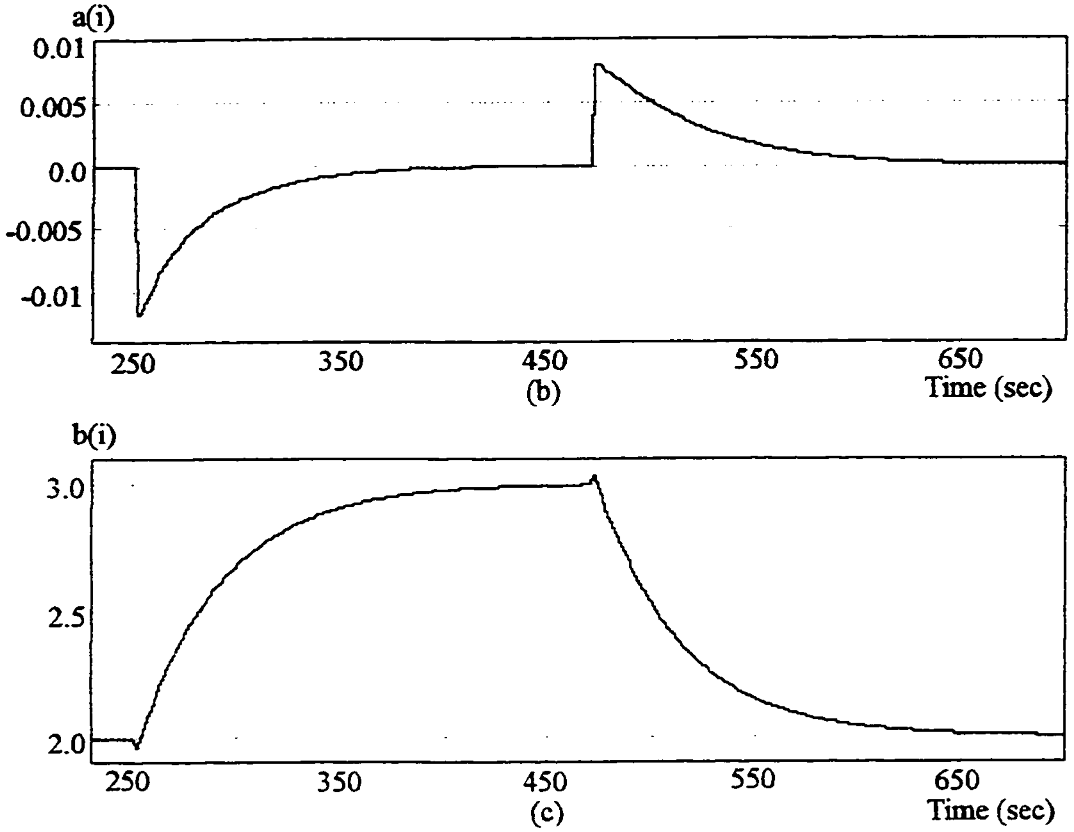


Figure 3.6 Gray Prediction --- The First Order Process (Con't)



(b)

(c)

Figure 3.6 Gray Prediction --- The First Order Process

as the process output increases, while no over-shoot as the process decreases. We will discuss both features later.

Simulation 3.4: Gray Prediction For The 2nd Order Processes

CASE 1: System with 2 different negative real eigenvalues and prediction horizon 40 sec.

Process transfer function is

$$Y(s)/U(s) = G(s) = 1/(1+40s)(1+20s).$$

U(s) is the transform of an unit step input signal u(t).

$$u(t) = 2*1(t) + 1(t-320) - 1(t-540), \quad t \geq 0.$$

Figure 3.7(a) shows the predictive result comparing to the process output. Figure 3.7(b) and 3.7(c) show the dynamic a(i) and b(i) from the gray predictor, respectively.

CASE 2: System with 2 stable complex eigenvalues and prediction horizon 30 sec

$$Y(s)/U(s) = G(s) = 1/(1400s^2 + 30s+1).$$

U(s) is the transfer function of an unit step input signal u(t).

$$u(t) = 2*1(t) + 1(t-600) - 1(t-1000), \quad t \geq 0.$$

Figure 3.8 shows the predictive result compared with the process output. Figure 3.8(b) and 3.8(c) show the dynamic a(i) and b(i) from the gray predictor, respectively.

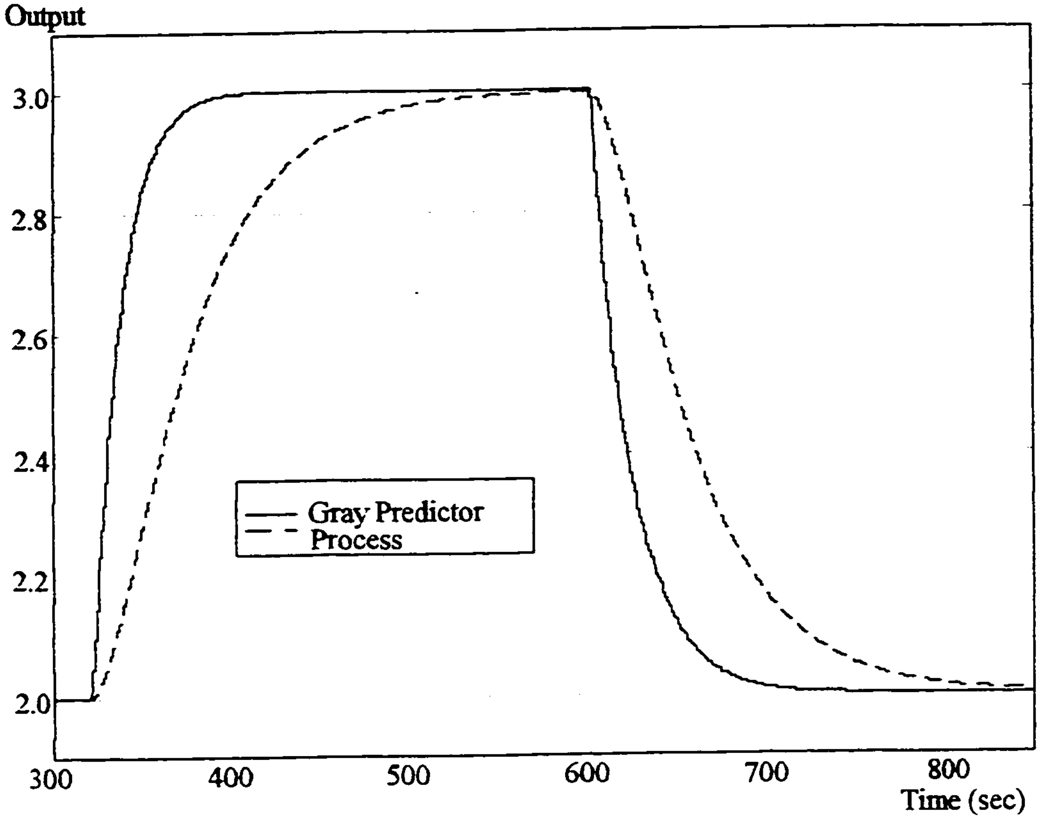


Figure 3.7 Gray Prediction --- Process with
Two Different Negative Real Eigenvalues (Con't)

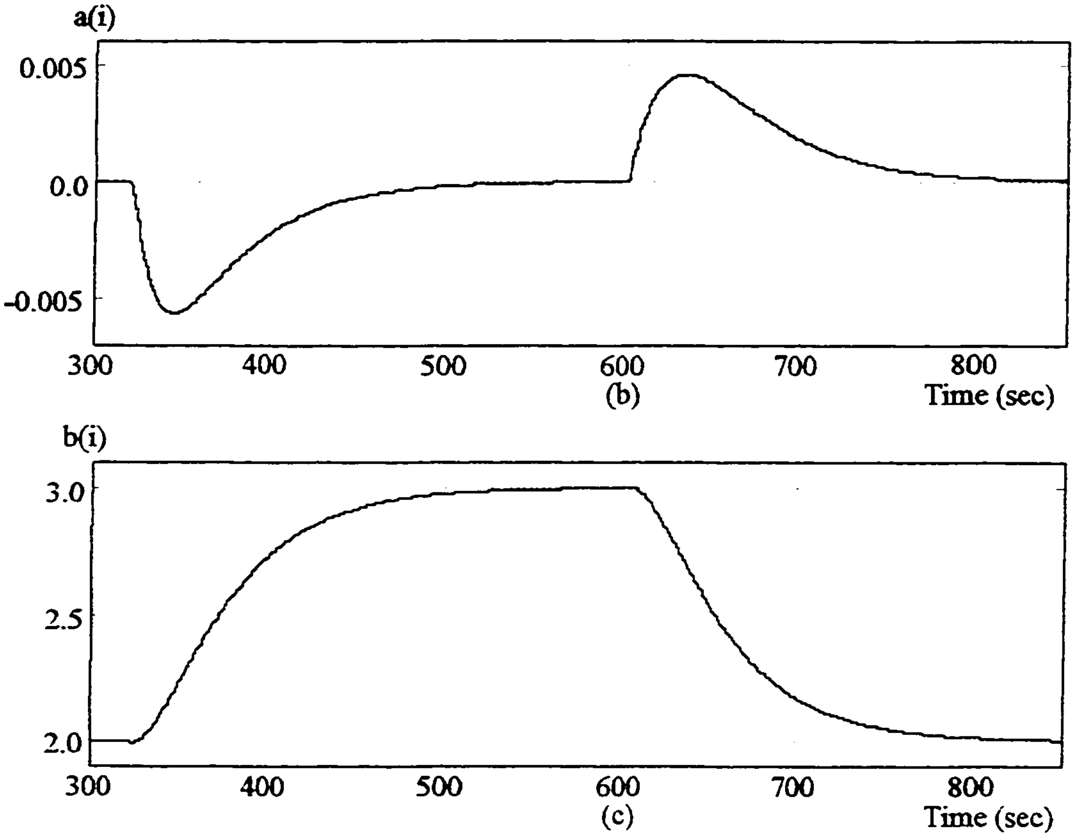


Figure 3.7 Gray Prediction — Process with Two Different Negative Real Eigenvalues

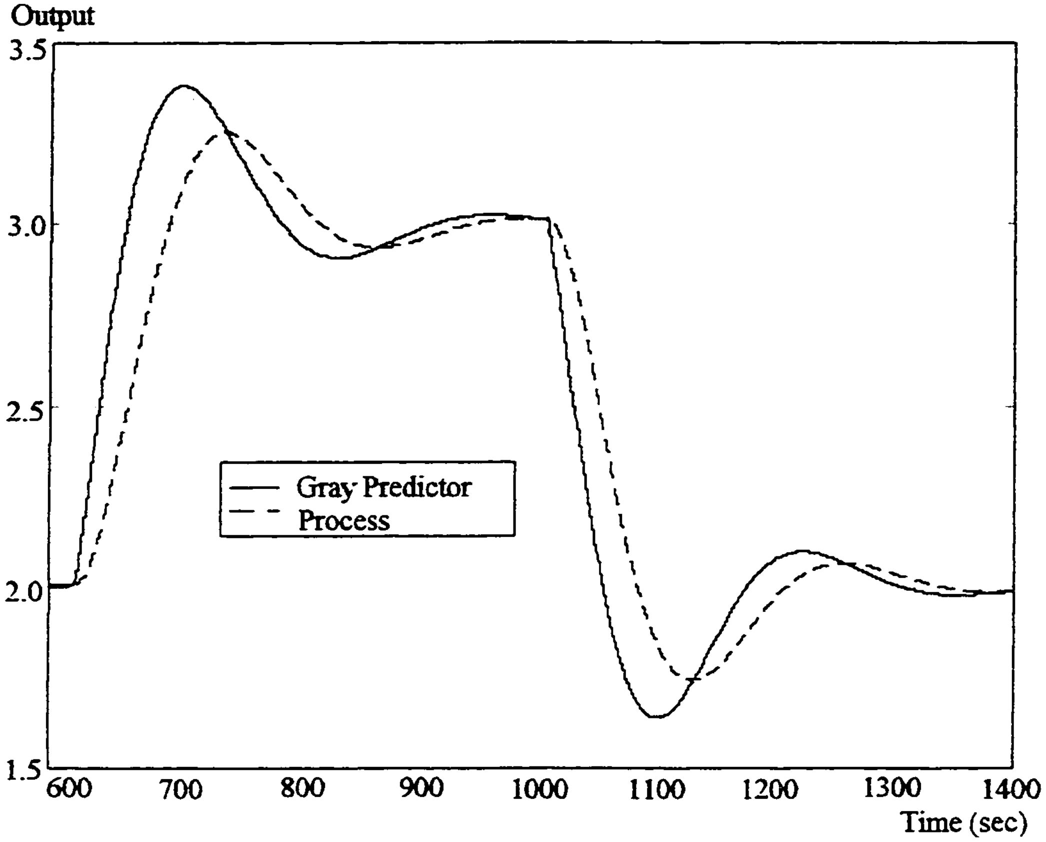


Figure 3.8 Gray Prediction --- Process
with Two Stable Complex Eigenvalues (Con't)

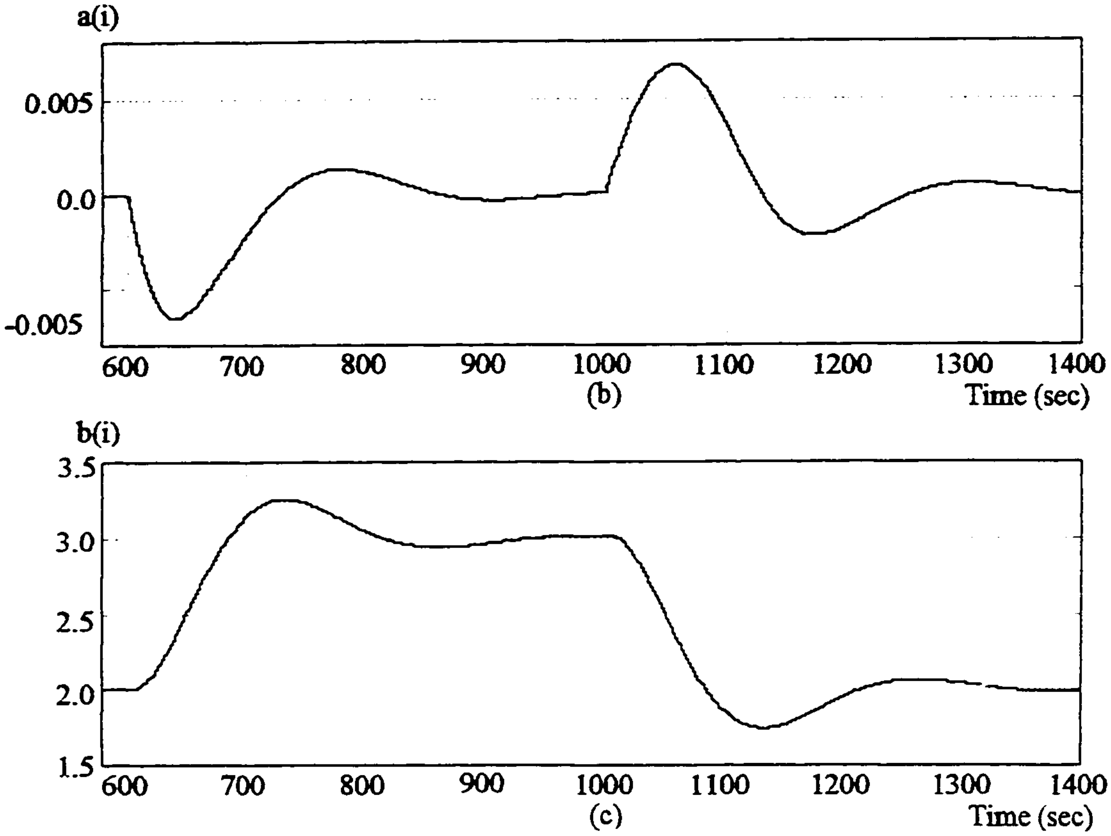


Figure 3.8 Gray Prediction --- Process
with Two Stable Complex Eigenvalues

Similar to Simulation 3.3, both cases in this simulation also show some common features.

1. The time that predictive outputs are ahead of process outputs is not equal to the prediction horizon.
2. Prediction is more aggressive as process outputs increase than decrease. Predictive effects at two directions are different.

As we know, the gray prediction does not directly build the model using original data sequence. Instead, it uses two-step procedure. First, the GM(1,1) model (3.5e) and (3.5f), the first order difference equation, is used to approximate the AGO sequence or discrete integral sequence of the original data, then the predictive values for AGO sequence is calculated. Second, I-AGO or difference, as well as the predictive values of AGO sequence, are used to calculate the predictive values of the original process output sequence. Even if the GM(1,1) model is used to predict the first order processes, the GM(1,1) model does not exactly reproduce the original processes. This has been discussed in Section 3.1.4. Hence, the time that predictive outputs are ahead of process outputs is not equal to the prediction horizon.

Before going to address the second feature above, we introduce two theorems as well as some propositions from the reference [18], then deduce two corollaries.

Proposition 3.2[18]: Given discrete sequences $e^{(0)}$, $e^{(1)}$ and $Z^{(1)}$ which satisfy

$$e^{(0)} = (\ e^{(0)}(1_i), e^{(0)}(2_i), \dots , e^{(0)}(n_i)\ ), \tag{3.16a}$$

$$e^{(1)} = \mathrm{AGO}(\ e^{(0)}\ ) = (\ e^{(1)}(1_i), e^{(1)}(2_i), \dots , e^{(1)}(n_i)\ ), \tag{3.16b}$$

$$e^{(1)}(1_i) = e^{(0)}(1_i), \tag{3.16c}$$

$$e^{(1)}(k_i) = e^{(0)}(k_i) + e^{(1)}(k_i\text{-}1),\ \ k=2,3,...n, \tag{3.16d}$$

$$Z^{(1)}(1_i) = e^{(1)}(1_i), \tag{3.16e}$$

$$Z^{(1)}(k_i) = 0.5(e^{(1)}(k_i) + e^{(1)}((k\text{-}1)_i),\ \ k=2,3,...,n, \tag{3.16f}$$

then

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i) = 2 * [\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(1)}(k_i) - \sum_{k=2}^{n} Z^{(1)}(k_i)^2]. \tag{3.17}$$

Proof: From (3.16d) and (3.16f), we have

$$2*Z^{(1)}(k_i) = e^{(1)}(k_i) + e^{(1)}((k-1)_i), \qquad k = 2,3,...,n, \text{ and}$$
$$e^{(0)}(k_i) = e^{(1)}(k_i) - e^{(1)}((k-1)_i), \qquad k = 2,3,...,n.$$

These two equations directly lead to

$$e^{(0)}(k_i) = 2*[e^{(1)}(k_i) - Z^{(1)}(k_i)], \qquad k = 2,3,\ldots,n,$$

so that we can further have

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i) = \sum_{k=2}^{n} Z^{(1)}(k_i)*2*[e^{(1)}(k) - Z^{(1)}(k_i)].$$
$$= 2 * [\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(1)}(k_i) - \sum_{k=2}^{n} Z^{(1)}(k_i)^2].$$

Proposition 3.3[18]**:** Given discrete sequences $e^{(0)}$, $e^{(1)}$ and $Z^{(1)}$ which satisfy (3.16a) to (3.16f), then

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i) = 0.5*[e^{(1)}(n_i)^2 - e^{(1)}(1_i)^2]. \tag{3.18}$$

Proof: From (3.16f), it is easy to have

$$0.5e^{(1)}((k-1)_i) = Z^{(1)}(k_i) - 0.5e^{(1)}(k_i),$$

therefore

$$\sum_{k=2}^{n} [\, 0.5e^{(1)}((k-1)_i)\,]^2 = \sum_{k=2}^{n} [\, Z^{(1)}(k_i) - 0.5e^{(1)}(k_i)\,]^2$$

$$= \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - \sum_{k=2}^{n} Z^{(1)}(k_i)e^{(1)}(k_i) + \sum_{k=2}^{n} 0.25e^{(1)}(k_i)^2$$

Rearranging items in the equation above, we have

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(1)}(k_i) = \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - \sum_{k=2}^{n} 0.25e^{(1)}((k-1)_i)^2 + \sum_{k=2}^{n} 0.25e^{(1)}(k_i)^2. \qquad (3.19)$$

Substitute (3.19) in (3.17), then it follows that

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i) = 2 * [\, \sum_{k=2}^{n} 0.25e^{(1)}(k_i)^2 - \sum_{k=2}^{n} 0.25e^{(1)}((k-1)_i)^2\,]$$

$$= 0.5 * \{\, [\, \sum_{k=1}^{n} e^{(1)}(k_i)^2 - e^{(1)}(1_i)^2\,] - [\, \sum_{k=1}^{n} e^{(1)}(k_i)^2 - e^{(1)}(n_i)^2\,]\, \}$$

$$= 0.5 * [\, e^{(1)}(n_i)^2 - e^{(1)}(1_i)^2\,].$$

Theorem 3.4[18]: The parameter a(i), shown in (3.6a) and (3.6b), for GM(1,1) model (3.5e) can be written as

$$a(i) = A(i)/C(i), \tag{3.20a}$$

in which

$$A(i) = [\ e^{(1)}(n_i) - e^{(1)}(1_i)\ ] * Q(i), \tag{3.20b}$$

$$Q(i) = \sum_{k=2}^{n} Z^{(1)}(k_i) - 0.5 * (n-1) * [e^{(1)}(n_i) + e^{(1)}(1_i)\ ], \tag{3.20c}$$

$$C(i) = (n-1) \sum_{k=2}^{n} (\ Z^{(1)}(k_i) - Z_{ave}(i)\ )^2 \tag{3.20d}$$

$$Z_{ave}(i) = (n-1)^{-1} \sum_{k=2}^{n} Z^{(1)}(k_i) \tag{3.20e}$$

Proof: First we prove that C(i) can be written as (3.20d) and (3.20e). From (3.6b), we have

$$\begin{aligned} C(i) &= (n-1) \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - [\sum_{k=2}^{n} Z^{(1)}(k_i)\ ]^2, \\ &= (n-1) \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - (n-1)^2\ [\ (n-1)^{-2} \sum_{k=2}^{n} Z^{(1)}(k_i)\ ]^2, \\ &= (n-1) \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - (n-1)^2 * Z_{ave}(i)^2, \end{aligned}$$

where $Z_{ave}(i)$ is given in (3.20e). Furthermore C(i) can be written as

$$\begin{aligned} C(i) &= (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)^2 - 2*(n-1)^2 * Z_{ave}(i)^2 \;+\; (n-1)^2 * Z_{ave}(i)^2, \\ &= (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)^2 - 2*(n-1)^2\,[\,(n-1)^{-1}\sum_{k=2}^{n} Z^{(1)}(k_i)\,]* Z_{ave}(i) + (n-1)^2 * Z_{ave}(i)^2. \end{aligned} \tag{3.21}$$

Notice that

$$(n-1)^2 * Z_{ave}(i)^2 = (n-1)*\sum_{k=2}^{n} Z_{ave}(i)^2,$$

so we can write (3.21) as

$$\begin{aligned} C(i) &= (n-1)\{\sum_{k=2}^{n} Z^{(1)}(k_i)^2 - 2 * [\,\sum_{k=2}^{n} Z^{(1)}(k_i)\,] * Z_{ave}(i) + \sum_{k=2}^{n} Z_{ave}(i)^2\}, \\ &= (n-1)\sum_{k=2}^{n} [\,Z^{(1)}(k_i)^2 - 2 * Z^{(1)}(k_i) * Z_{ave}(i) + Z_{ave}(i)^2\,], \\ &= (n-1)\sum_{k=2}^{n} [\,Z^{(1)}(k_i) - Z_{ave}(i)\,]^2. \end{aligned}$$

Thus (3.20d) and (3.20e) are proven. In order to prove (3.20b) and (3.20c), let's rewrite A(i) in (3.6a) and (3.6b), and transform it to

$$A(i) = \sum_{k=2}^{n} Z^{(1)}(k_i) * \sum_{k=2}^{n} e^{(0)}(k_i) - (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i).$$

$$= \sum_{k=2}^{n} Z^{(1)}(k_i) * [\, e^{(1)}(n_i) - e^{(1)}(1_i) \,] - (n-1)\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i).$$

Substituting (3.18) in the equation above, we can obtain

$$A(i) = \sum_{k=2}^{n} Z^{(1)}(k_i) * [\, e^{(1)}(n_i) - e^{(1)}(1_i) \,] - (n-1) * 0.5 * [\, e^{(1)}(n_i)^2 - e^{(1)}(1_i)^2 \,].$$

$$= \ [\, e^{(1)}(n_i) - e^{(1)}(1_i) \,] * \{ \sum_{k=2}^{n} Z^{(1)}(k_i) - 0.5 * (n-1) * [e^{(1)}(n_i) + e^{(1)}(1_i) \,] \}.$$

Denote

$$Q(i) = \sum_{k=2}^{n} Z^{(1)}(k_i) - 0.5 * (n-1) * [e^{(1)}(n_i) + e^{(1)}(1_i) \,],$$

then (3.20b) and (3.20c) are proven.

Finally an equation for B(i) in (3.6b) from the reference[18] is given below without proof

.

$$B(i) = (\, e^{(1)}(n_i) - e^{(1)}(1_i) \,) \sum_{k=2}^{n} [\, Z^{(1)}(k_i) - Z_{ave}(i) \,]^2 + Z_{ave}(i)A(i), \qquad (3.22)$$

in which $Z_{ave}(i)$ is given in (3.20e).

Corollary 3.1: For any non-negative smooth discrete sequence and the corresponding GM(1,1) model shown in (3.5e) with the window dimension n = 4, then the signs of a(i) and A(i) are determined and only determined by $e^{(0)}(2_i) - e^{(0)}(4_i)$ in the ith time window.

This corollary can be verified simply by noticing that Q(i) in (3.20c) is equivalent to

$$Q(i) = 0.5e^{(1)}(1_i) + e^{(1)}(2_i) + e^{(1)}(3_i) + 0.5e^{(1)}(4_i) - 1.5[\, e^{(1)}(4_i) + e^{(1)}(1_i)\,]$$
$$= e^{(1)}(2_i) + e^{(1)}(3_i) - e^{(1)}(4_i) - e^{(1)}(1_i)$$
$$= e^{(0)}(2_i) - e^{(0)}(4_i), \qquad (3.23)$$

as well as that $[\, e^{(1)}(n_i) - e^{(1)}(1_i)\,]$ in (3.20b), and $C(i)$ in (3.20d) are always positive.

Corollary 3.2: Given a discrete non-negative smooth monotonic-increase sequence

$$e_i^{(0)} = (\, e^{(0)}(1_i), e^{(0)}(2_i), \ldots, e^{(0)}(n_i)\,),$$

then the gray predictive algorithm (3.7b) for GM(1,1) model (3.5e) with $n = 4$ has the following form provided $e^{(0)}(1_i)$ is large enough.

$$e^{(0)}(k_i) = c_1(i) * p_1(i)^{(ki-1)}, \quad k \geq n + 1. \qquad (3.24a)$$
$$c_1(i) = (\, b(i) - a(i) * e^{(0)}(1_i)\,)/(1 - 0.5a(i)\,), \qquad (3.24b)$$
$$p_1(i) = (\, 1 - 0.5a(i)\,)/(\, 1 + 0.5a(i)\,), \quad |\, a(i)\, | < 2, \qquad (3.24c)$$

where $c_1(i) > 0$ and $p_1(i) > 1$. For a discrete smooth monotonic-decrease sequence $e_i^{(0)}$, the gray predictive algorithm (3.7b) for GM(1,1) model (3.5e) with $n = 4$ has form

$$e^{(0)}(k_i) = c_2(i) * p_2(i)^{(ki-1)}, \quad k \geq n + 1, \qquad (3.24d)$$
$$c_2(i) = (\, b(i) - a(i) * e^{(0)}(1_i)\,)/(1 - 0.5a(i)\,), \qquad (3.24e)$$
$$p_2(i) = (\, 1 - 0.5a(i)\,)/(\, 1 + 0.5a(i)\,), \quad |\, a(i)\, | < 2. \qquad (3.24f)$$

where $c_2(i) > 0$ and $0 < p_2(i) < 1$.

Proof: First let's rewrite the gray predictive algorithm (3.7b) below

$$e^{(0)}(k_i) = c(i) * r(i)^{(ki-1)}, \quad k \geq n+1. \tag{3.25a}$$

$$c(i) = (\, b(i) - a(i) * e^{(0)}(1_i)\,)/(1 - 0.5a(i)\,), \tag{3.25b}$$

$$r(i) = (\, 1 - 0.5a(i)\,)/(\, 1 + 0.5a(i)\,), \quad |\, a(i)\,| < 2. \tag{3.25c}$$

From (3.20b), (3.23) and n = 4 we have

$$A(i) = (\, e^{(1)}(4_i) - e^{(1)}(1_i)\,) * (\, e^{(0)}(2_i) - e^{(0)}(4_i)\,). \tag{3.26a}$$

Combining with (3.6a), (3.22) and (3.26a), we can write (3.25b) as

$$\begin{aligned} c(i) &= C(i)^{-1}(\, B(i) - A(i) * e^{(0)}(1_i)\,) / (1 - 0.5a(i)\,), \\ &= C(i)^{-1}[\, E(i) + A(i)(\, Z_{ave}(i) - e^{(0)}(1_i)\,)\,] / (1 - 0.5a(i)\,), \end{aligned} \tag{3.26b}$$

in which

$$E(i) = (\, e^{(1)}(4_i) - e^{(1)}(1_i)\,) * \sum_{k=2}^{4} [\, Z^{(1)}(k_i) - Z_{ave}(i)\,]^2. \tag{3.26c}$$

$$Z_{ave}(i) = (1/3) \sum_{k=2}^{4} Z^{(1)}(k_i). \tag{3.26d}$$

$C(i) > 0$ and $Z_{ave}(i) > 0$.

For smooth monotonic-decrease sequence $e_i^{(0)}$, $e^{(0)}(2_i) - e^{(0)}(4_i) < 0$. From Corollary 3.1, a(i) > 0 and A(i) > 0 so that r(i) in (3.25c) is smaller than 1. Also c(i) shown in (3.26b) is positive, since $E(i) > 0$, $C(i) > 0$, $Z_{ave}(i) - e^{(0)}(1_i) > 0$ and $1 - 0.5a(i) > 0$. Denote c(i) as $c_2(i)$ and r(i) as $p_2(i)$, then (3.24d) to (3.24f) are proven.

If sequence $e_i^{(0)}$ is smooth monotonic-increase sequence, then $e^{(0)}(2_i) - e^{(0)}(4_i) < 0$. From Corollary 3.1, $a(i) < 0$ and $A(i) < 0$ so that r(i) in (3.25c) is larger than 1. Extending (3.26d) and $Z^{(1)}(k_i)$, we have

$$Z_{ave}(i) = (1/3)(1.5e^{(0)}(1_i) + 2.5\ e^{(0)}(2_i) + 1.5\ e^{(0)}(3_i) + 0.5\ e^{(0)}(4_i)), \tag{3.26e}$$

$$Z^{(1)}(2_i) = 0.5e^{(0)}(1_i) + 0.5e^{(0)}(2_i), \tag{3.26f}$$

$$Z^{(1)}(3_i) = 0.5e^{(0)}(1_i) + e^{(0)}(2_i) + 0.5\ e^{(0)}(3_i), \tag{3.26g}$$

$$Z^{(1)}(4_i) = 0.5e^{(0)}(1_i) + e^{(0)}(2_i) + e^{(0)}(3_i) + 0.5e^{(0)}(4_i). \tag{3.26h}$$

Furthermore we have

$$(Z^{(1)}(2_i) - Z_{ave}(i))^2 = [(1/3)(-e^{(0)}(2_i) - 1.5\ e^{(0)}(3_i) - 0.5\ e^{(0)}(4_i))]^2 > e^{(0)}(1_i)^2,$$

$$(Z^{(1)}(3_i) - Z_{ave}(i))^2 = [(1/3)(0.5e^{(0)}(2_i) - 0.5\ e^{(0)}(4_i))]^2 > 0,$$

$$(Z^{(1)}(2_i) - Z_{ave}(i))^2 = [(1/3)(0.5e^{(0)}(2_i) + 1.5\ e^{(0)}(3_i) + e^{(0)}(4_i))]^2 > e^{(0)}(1_i)^2,$$

so that E(i) in (3.26c) satisfies

$$\begin{aligned} E(i) = &(e^{(1)}(4_i) - e^{(1)}(1_i)) * \\ &[(Z^{(1)}(2_i) - Z_{ave}(i))^2 + (Z^{(1)}(3_i) - Z_{ave}(i))^2 + (Z^{(1)}(2_i) - Z_{ave}(i))^2 \\ &> (e^{(1)}(4_i) - e^{(1)}(1_i)) * 2e^{(0)}(1_i)^2. \end{aligned} \tag{3.26i}$$

Combining (3.20b) and (3.26e) with Corollary 3.1, we have

$$\begin{aligned} &A(i)(Z_{ave}(i) - e^{(0)}(1_i)) \\ &= (1/3)(e^{(1)}(4_i) - e^{(1)}(1_i)) * (e^{(0)}(2_i) - e^{(0)}(4_i)) * \\ &\quad [(1.5e^{(0)}(1_i) + 2.5e^{(0)}(2_i) + 1.5e^{(0)}(3_i) + 0.5e^{(0)}(4_i) - 3\ e^{(0)}(1_i)] \\ &> - (e^{(1)}(4_i) - e^{(1)}(1_i)) * | (e^{(0)}(2_i) - e^{(0)}(4_i)) | * 2e^{(0)}(4_i), \end{aligned}$$

where $|\ (e^{(0)}(2_i) - e^{(0)}(4_i))\ |$ is the absolute value of $e^{(0)}(2_i) - e^{(0)}(4_i)$. Since $e^{(0)}$ is smooth discrete sequence, $(e^{(0)}(1_i) - e^{(0)}(4_i))$ is bounded. Assume $|\ (e^{(0)}(1_i) - e^{(0)}(4_i))\ | < \delta$, then

$$A(i)(\ Z_{ave}(i) - e^{(0)}(1_i)\ ) > -(\ e^{(1)}(4_i) - e^{(1)}(1_i)\ ) * 2\delta e^{(0)}(4_i)$$
$$> -(\ e^{(1)}(4_i) - e^{(1)}(1_i)\ ) * (\ 2\delta e^{(0)}(1_i) + 2\delta^2\ ). \qquad (3.26j)$$

By (3.26i) and (3.26j), finally we have

$$E(i) + A(i)(\ Z_{ave}(i) - e^{(0)}(1_i)\ )$$
$$> (\ e^{(1)}(4_i) - e^{(1)}(1_i)\ ) * [\ 2e^{(0)}(1_i)^2 - (\ 2\delta e^{(0)}(1_i) + 2\delta^2\ )\ ].$$

From (3.26b), we know that c(i) is positive if $E(i) + A(i)(\ Z_{ave}(i) - e^{(0)}(1_i)\ ) > 0$, since $C(i) > 0$ and $1 - 0.5a(i) > 0$. This condition can be met by choosing

$$e^{(0)}(1_i) > 2\delta.$$

So far we have proven c(i) in (3.25b) or (3.26b) is positive and r(i) in (3.25c) is larger than 1, simply denote c(i) as $c_1(i)$, and r(i) as $p_1(i)$, then (3.24a) to (3.24c) are proven.

We now consider the second feature shared by Simulation 3.3 and 3.4. That is, prediction is more aggressive as process outputs increase than decrease, and predictive effects at two directions are different.

For all cases in Simulation 3.3 and 3.4, as the process outputs increase, from Corollary 3.2, we use gray predictive algorithm (3.24a) to (3.24c) with $p_1(i) > 1$ to predict the future output values, while as the process outputs decrease, we use (3.24d) to (3.24f) with $0 < p_2(i) < 1$ to predict the future output values. Hence, predictive effects at two directions are different. Also, when prediction horizon tends to infinite, i.e. k_i tends to infinite, predictive value $e^{(0)}(k_i)$ in (3.24a) will tend to positive infinite since $p_1(i)^{(ki-1)}$ tends to positive infinite, but $e^{(0)}(k_i)$ in (3.24d) is bounded by zero. Therefore, prediction is more aggressive as process outputs increase than decrease.

3.1.6 Noise Suppression

As mentioned in 3.1.1, the gray prediction uses AGO or discrete integral to pre-process the original data, which can reduce standard deviation of the original data. On the other hand, the gray prediction suggests to build GM(1,1) model with 4-6 data[18], so GM(1,1) model may be still sensitive to noise if the gray predictor can not get enough information about noise due to small n and sample interval. The following simulation illustrates this.

Simulation 3.5: Sensitivity to noise

Process is described by

$$Y(s)/U(s) = G(s) = 1/(1+30s)(1+5s) \tag{3.27a}$$

The process input u(t) is a step signal with

$$u(t) = 1(t - 320). \tag{3.27b}$$

A white noise ("Band Limited White Noise" in Simulink 4.2c) through a filter 1/(1+5s) is added to the process output from t = 300 sec.

White Noise:
Power = 0.002, Sample Time = 0.1 sec, and
Seed: 23341. (3.27c)

In Figure 3.9, we set the sample interval of gray predictor to be 1 sec. The prediction horizon is 20 sec. Compared with the process output, we can see that gray predictor improperly

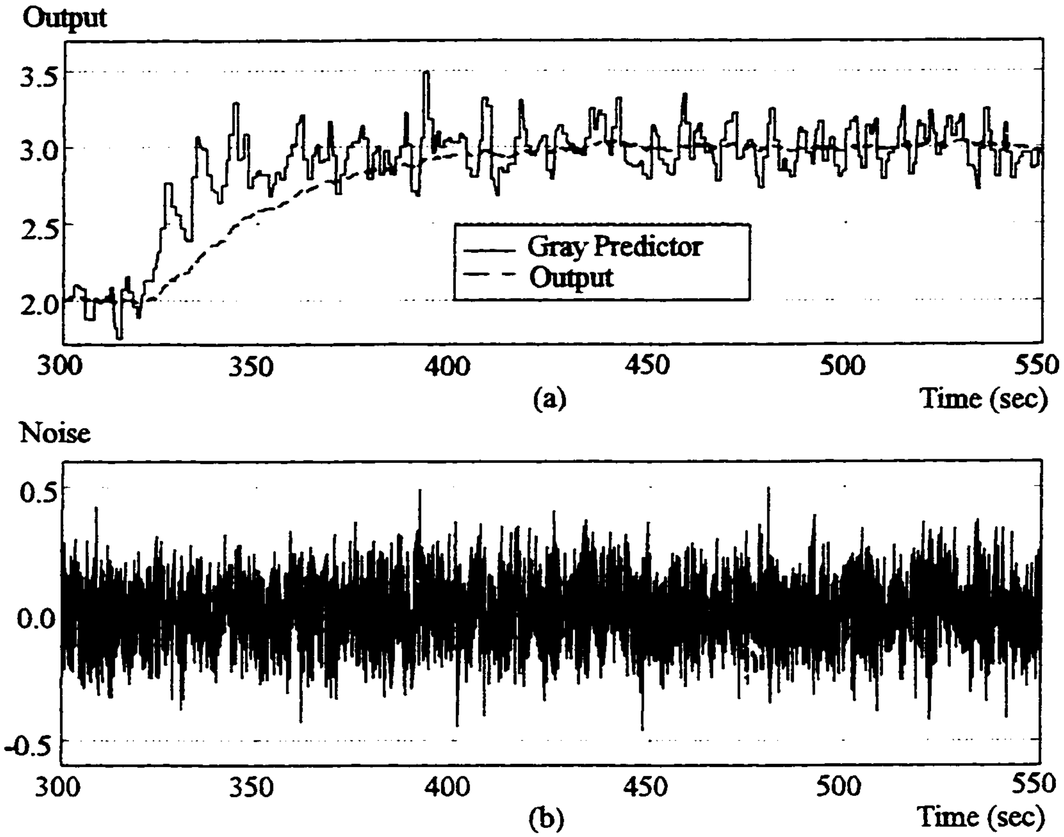


Figure 3.9 Sensitivity to Noises of Gray Predictor

amplifies the influence of noise too much, since the gray predictor can not get enough information about the noise due to small n and sample interval. If the GM(1,1) model is sensitive to noise, when used in noise environment, it may cause excessive actuators wear. In view of this, a "large sample interval and small predictive interval" method is presented in the following.

Methods To Reduce Sensitivity To Noises

Method 1: We can increase n = 4 to n = 15, then we can have satisfactory prediction. However, one of most important advantages for gray prediction is to model with fewer data. Otherwise, the computing burden will increase quickly. This advantage will be lost.

Method 2: Increase the sample interval of gray prediction.

Simulation 3.6: Large data interval --- noise suppression

Process is the same as Simulation 3.4. The transfer function, input and noise are described in (3.27a), (3.27b) and (3.27c).

In Fig 3.10, the sample interval of gray predictor is set to be 5sec and n is still 4. The predictive horizon is 20 sec. Comparing with the process output and Figure 3.9, we see that the sensitivity to the noise is reduced. However, it may worsen control quality to increase the sample interval too much.

Method 3: Instead of Method 1 and Method 2, we present a trade-off method called "large data interval and small predictive interval" method.

At each time window i, keep (n-1)*m+1 data

$$
\begin{aligned}
e_i^{(0)} = (&e^{(0)}(1_i), e^{(0)}(2_i), \ldots , e^{(0)}(m_i),\\
&e^{(0)}(\ (m+1)_i\ ), e^{(0)}(\ (m+2)_i\ ), \ldots , e^{(0)}(\ (2*m)_i\ ),\\
&\ldots , e^{(0)}(\ ((n-1)*m+1)_i\ ),
\end{aligned}
$$

where $e^{(0)}(\ ((n-1)*m+1)_i\ )$ is the current datum. We only use

$$e^{(0)}(1_i), e^{(0)}(\ (m+1)_i\ ), \ldots, e^{(0)}(\ ((n-1)*m+1)_i\ )$$

to build the dynamic GM(1,1) model, and predict the future values. By this method, we do not increase computing burden and worsen control quality too much, but effectively reduce the sensitivity of gray predictor to noise. The following simulation will illustrate this method.

Simulation 3.7: Reduction of sensitivity to noises

Process is the same as Simulation 3.4. Its transfer function. input and noise are described in (3.27a), (3.27b) and (3.27c).

We set the sample interval of gray predictor to be 1 sec and n=4. m=5. In figure 3.11, Method 3 is labeled as "LDISPI". Compared with the normal gray predictor. the LDSP gray predictor greatly reduces the sensitivity to noise at the cost of somewhat lower predictive speed.

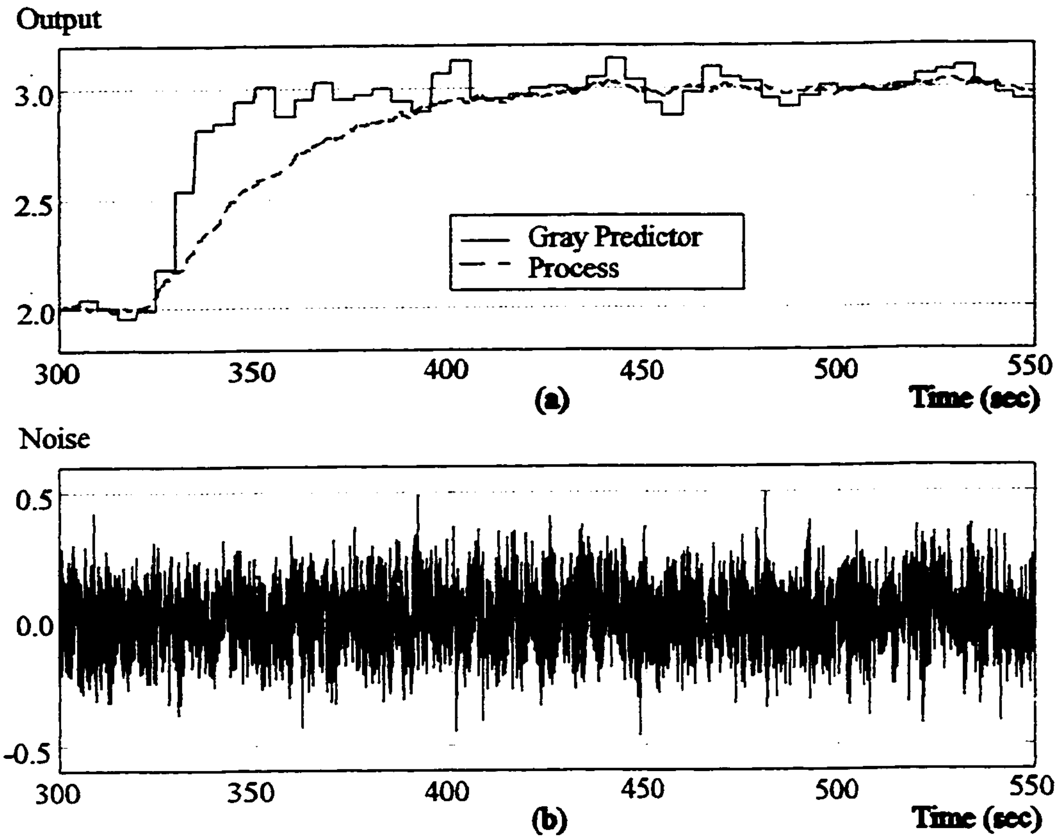


Figure 3.10 Large Data Interval — Noise Suppression

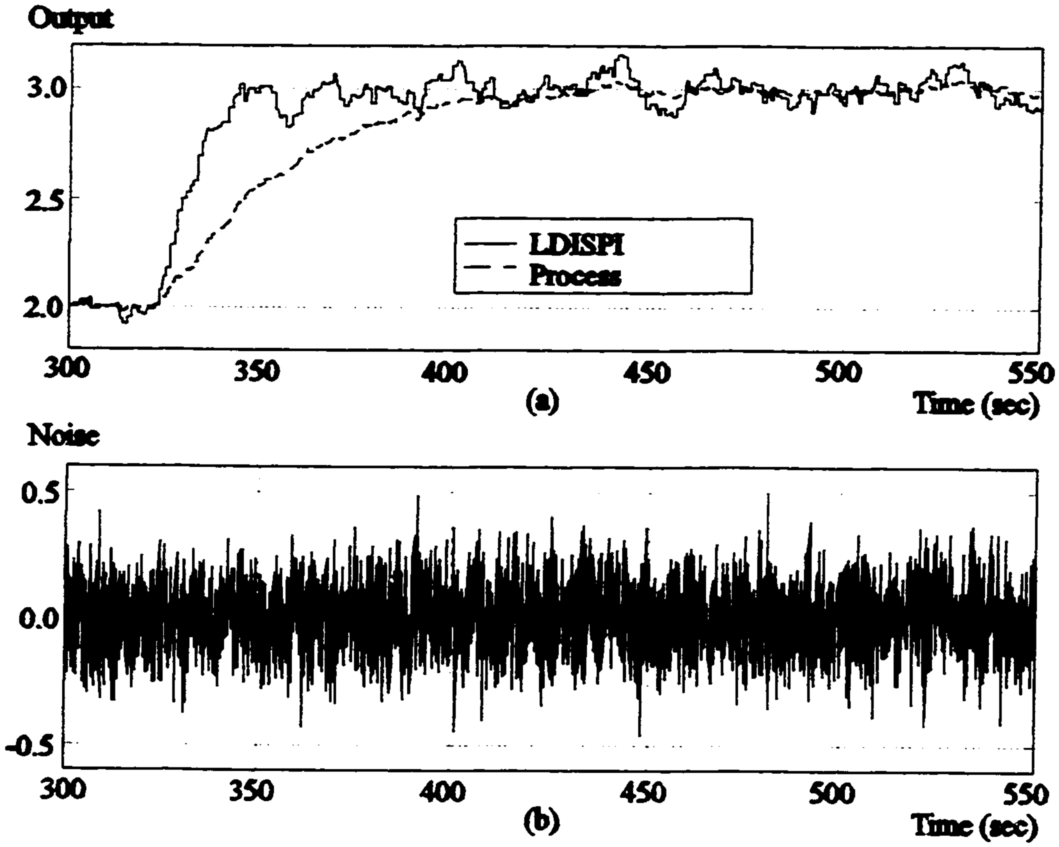


Figure 3.11 Predictive Effect of LDISPI Gray Predictor --- Noise Suppression

3.2 Error-Prediction-Based Gray Predictive PI

Figure 3.12 shows a basic structure of gray predictive controller presented in the literature [18], in which G(s) is the process, $G_c(s)$ is the process controller. The GM(1,1) model is used to predict the process output y to improve control quality for processes with large time constant.

Some advantages of this scheme are

1. For setpoint tracking, usually no overshoot exists.
2. For most processes, y has a fixed sign and usually y is not equal to zero, the sufficient conditions for gray exponential form in Theorem 3.1 are automatically satisfied or can be satisfied simply changing the signs of process outputs if the process outputs are negative.

The shortcoming for this scheme is that tracking speed for setpoint may be too slow if overshoot is allowed.

For boiler-turbine-generator control, which will be discussed later, setpoint tracking is a very important criterion. Considering this, we suggest to use error-prediction-based controller structure shown in Figure 3.13. In this structure, we predict e instead of y. A problem encountered is that error e has no fixed signs, so the sufficient conditions for gray exponential form in Theorem 3.1 can not be held. To solve this problem, a pair of bias signals, B and -B, $B > 0$, will be added into Figure 3.13. First, a large enough positive B is

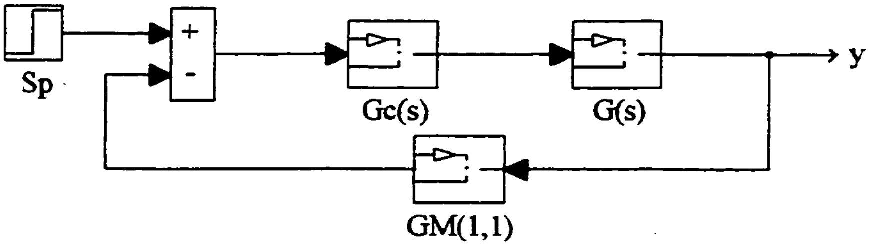


Figure 3.12 Basic Structure of Gray Predictive Control

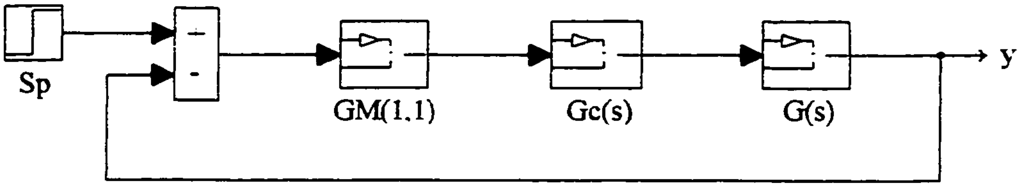


Figure 3.13 Initial Structure of Suggested Gray Predictive Control

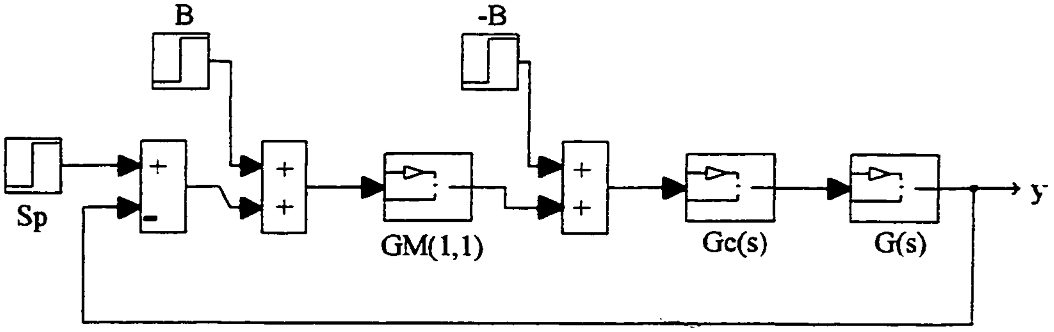


Figure 3.14 Final Structure of Suggested Gray Predictive Control

added to error e so that B+e is always positive and large enough. After using GM(1,1) model to get predictive values of B+e, we subtract B from it to get predictive error e. The final structure for this error-prediction-based gray predictive controller is shown in Figure 3.14. If the controller in Figure 3.14 is PI, then it is an error-prediction-based gray predictive PI. For simplicity, we will still call it gray predictive PI in the following sections.

3.2.1. Consistency of Gray Predictive PI

Proposition 3.4: Given a non-negative smooth discrete data sequence $e_i^{(0)}$

$$e_i^{(0)} = (e^{(0)}(1_i), e^{(0)}(2_i). \dots ,e^{(0)}(n_i)).$$

If $e^{(0)}(k_i)$ ---> $E_\infty \neq 0$, constant, for any k=1,2,...,n, as i ---> ∞, then we have

$$a(i) \text{ ---> } 0,$$

$$b(i) \text{ ---> } E_\infty, \text{ constant} \quad \text{as } i \text{ ---> } \infty,$$

where a(i) and b(i) are the least square solution of GM(1,1) model parameters shown in (3.6).

Proof:

Since $e^{(0)}(k_i)$ ---> E_∞, as t ---> ∞,

thus $Z^{(1)}(k_i) = 0.5(e^{(1)}(k_i) + e^{(1)}((k-1)_i)) \text{ ---> } [(2k-1)*E_\infty]/2,$

$$\sum_{k=2}^{n} Z^{(1)}(k_i) \text{ ---> } (n+1)(n-1)*E_\infty/2, \tag{3.28a}$$

$$\sum_{k=2}^{n} e^{(0)}(k_i) \dashrightarrow (n-1)*E_\infty, \qquad (3.28b)$$

$$\sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i) \dashrightarrow (n+1)(n-1)*E_\infty^2/2, \qquad \text{as } t \dashrightarrow \infty. \qquad (3.28c)$$

From (3.6b), it follows that

$$A(i) \dashrightarrow [(n+1)(n-1)*E_\infty/2][(n-1)*E_\infty] - (n-1)(n+1)(n-1)*E_\infty^2/2 = 0, \text{ as } i \dashrightarrow \infty.$$

From (3.20d) in Theorem 3.4, C(i) does not tend to zero as i ---> ∞, thus

$$a(i) = A(i)/C(i) \dashrightarrow 0, \qquad \text{as } i \dashrightarrow \infty.$$

Also from (3.6b) we know B(i) tends to constant as i tends to infinite.

$$B(i) = \sum_{k=2}^{n} (Z^{(1)}(k_i))^2 \sum_{k=2}^{n} e^{(0)}(k_i) - \sum_{k=2}^{n} Z^{(1)}(k_i) \sum_{k=2}^{n} Z^{(1)}(k_i)e^{(0)}(k_i)$$

$$\dashrightarrow [\ (n-1) \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - (\sum_{k=2}^{n} Z^{(1)}(k_i)\ )^2\ ] * E_\infty,$$

Compared with C(i), which tends to constant as i tends to infinite,

$$C(i) = (n-1) \sum_{k=2}^{n} Z^{(1)}(k_i)^2 - (\sum_{k=2}^{n} Z^{(1)}(k_i)\ )^2,$$

it follows that

$$b(i) = B(i)/C(i) \dashrightarrow E_\infty,\ \text{constant} \qquad \text{as } i \dashrightarrow \infty.$$

Now that we have proven Proposition 3.4, let's recall Figure 3.14. In the ith time window, denote

$$e(k_i) = S_p - y(k_i), \qquad k=1,2,...,n, \qquad (3.29a)$$

$$e_b(k_i) = e(k_i) + B, \qquad k=1,2,...,n. \qquad (3.29b)$$

Replacing $e^{(0)}$ with e_b, and $e^{(1)}$ with $e_b^{(1)} = AGO(e_b)$ in (3.7a) and (3.7b), keep in mind that when $k_i \geq n+1$, what we have are predictive values, so we can write the following predictive algorithm of $e_b(k_i)$

$$e_b\hat{}(k_i) = [\ (\ b(i) - a(i) * e_b(1_i)\ )/(1 - 0.5a(i)\ )\ ] * r(i)^{(ki-1)},\ \ k_i \geq n+1,$$

$$r(i) = (\ 1 - 0.5a(i)\ )/(\ 1 + 0.5a(i)\ ), \qquad |\, a(i)\, | < 2,$$

$$e\hat{}(k_i) = e_b\hat{}(k_i) - B,$$

or

$$e_b\hat{}(n_i+p_h) = [\ (\ b(i) - a(i) * e_b^{)}(1_i)\ )/(1 - 0.5a(i)\ )\ ] * r(i)^{(ni+ph-1)},\ \ p_h \geq 1, \qquad (3.29c)$$

$$e\hat{}(n_i+p_h) = e_b\hat{}(n_i+p_h) - B. \qquad (3.29d)$$

in which e is the actual error, e_b is the conditioned error, e_b^ is the gray predictive value of e_b, and e^ is the gray predictive value of e. $B > 0$, and p_h is the prediction horizon of gray predictive PI.

Proposition 3.5: Given a non-negative smooth discrete data sequence $e_i = (e(1_i), e(2_i), \dots,$

$e(n_i)$), if $e(k_i) \longrightarrow E_\infty$, constant, as $i \longrightarrow \infty$, then for any $p_h \geq 1$, $e^\wedge(n_i+p_h) \longrightarrow E_\infty$, as $i \longrightarrow \infty$.

Proof:

Since $e(k_i) \longrightarrow E_\infty$, as $i \longrightarrow \infty$, for any $k=1,2,\ldots,n$,

thus $e_b(k_i) \longrightarrow B + E_\infty$, as $i \longrightarrow \infty$, for any $k=1,2,\ldots,n$,

then it follows, from Proposition 3.4, that the parameters $a(i)$ and $b(i)$ for the GM(1,1) model in Figure 3.14 satisfy

$$a(i) \longrightarrow 0, \qquad \text{and}$$

$$b(i) \longrightarrow B+E_\infty, \qquad \text{as } i \longrightarrow \infty.$$

To prove this proposition, what we need is to show that

$$e_b{}^\wedge(n_i+p_h) \longrightarrow B + E_\infty, \qquad \text{as } i \longrightarrow \infty, \qquad \text{for any } p_h \geq 1.$$

From (3.29c), we know that

$$e_b{}^\wedge(n_i+p_h) \longrightarrow b(i) \longrightarrow B + E_\infty, \text{ as } i \longrightarrow \infty.$$

Hence from (3.29d), it follows that

$$e^\wedge(n_i+p_h) \longrightarrow E_\infty, \text{ as } i \longrightarrow \infty$$

Theorem 3.5: Given a discrete process

$$G(z) = (a_1 + a_2z^{-1} + \ldots + a_nz^{-n}) / (b_1 + b_2z^{-1} + \ldots + b_mz^{-m}),$$

$$\sum_{i=1}^{m} bi \neq 0,$$

and a discrete controller

$$G_c(z) = K_c[\ 1 + Tz/(T_i(z-1))\ ]$$

where T is the sample interval. Suppose B in Figure 3.14 is so large that the discrete sequence $e_b(k_i)$ is non-negative smooth discrete sequence. If the closed loop system shown in Figure 3.14 is asymptotically stable under the control of gray predictive PI, i.e. the gray predictive error $e^\wedge \text{--->} E_\infty$, constant, as $i \text{--->} \infty (t \text{--->} \infty)$, then y will asymptotically converge to S_p, as $i \text{--->} \infty$ $(t \text{--->} \infty)$, i.e. $E_\infty \text{--->} 0$, as $i \text{--->} \infty (t \text{--->} \infty)$.

It is easy to understand this theorem, since, from Proposition 3.5, we know that the GM(1,1) model in Figure 3.14 is a linear component whose discrete transfer function GM(z) satisfies

$$GM(z) \text{--->} 1, \quad \text{as } z \text{--->} 1,$$

so the gray predictive PI control systems structured by Figure 3.14 have similar consistency as conventional PI control systems.

In the following chapter we will apply the error-prediction-based gray predictive PI controller and continuous-linearly-interpolated gain scheduled PI controller to a boiler-turbine-generator unit, and show the control quality can be improved.

Chapter 4

Application of Gain Scheduled PI and Gray Predictive PI

A unit system of boiler-turbine-generator (BTG) is a non-linear-time-varying, highly-coupled, multivariable system. If all other boiler control systems are put into automation, we can view it as a 2-input and 2-output multivariable system. The inputs are fuel flow and opening of govemor valve; the outputs are throttle pressure (steam pressure) and electric power (load).

Two major goals of BTG control are:

1.To achieve stable and fast load following.
2.To ensure the safety of a BTG system. For example, a 300MW BTG system can have 16.67 MPa pressure and 555°C steam temperature. Under so high temperature, any further increase of pressure may damage the system.

These two goals require that the BTG control system not only have good capability for set-point tracking and disturbance rejection, but also have strong robustness. Further more, as we know, if a controller is too sensitive to noise, it may cause the actuator to be worn out quickly. An industrial actuator usually is quite expensive, therefore, besides the above features, BTG controllers must be insensitive to noise.

4.1 Basic Control Schemes For BTG System [12]

There are three basic control schemes for BTG systems.

4.1.1 Turbine Following

The principle for this mode is to use the error signal of electric power(and frequency) as the input signal(s) of fuel controller to control fuel flow, and use the error signal of throttle pressure as the input signal of turbine controller to control governor valve. Oncε the load demand changes, the fuel controller changes fuel flow first which causes the throttle pressure to change. Then the turbine controller opens/closes the governor valve to change electric power and meet the load demand. This scheme can make boiler systems stable, steam pressure not to vary excessively, but the load following is not very good, since it doesn't use the stored heat of boiler.

4.1.2 Boiler Following

The principle for this mode is to use the error signal of throttle pressure as the input signal of fuel controller to control fuel, and use the error signal of electric power (and frequency) as the input signal(s) of turbine controller to control governor valve. Once the load demand changes, first the turbine controller changes the opening of governor, which causes the electric power and the throttle pressure to change, then the fuel flow controller changes fuel to maintain throttle pressure and achieve energy balance. This scheme can have good load following, but throttle pressure may vary excessively. It is unsuitable for once-through boilers.

4.1.3 Coordinated Control

Based on boiler following mode, a feed-forward power error signal, which can quickly reflect the change of load demand, is sent to the fuel controller. This signal can be the function of P_1/P_t, where P_1 is the first stage pressure of turbine, and P_t is the throttle pressure. On the other hand, the opening of governor valve is limited if the error of throttle pressure exceeds the prescribed limits. Therefore when the load demand changes, this scheme not only can use the stored heat of boiler effectively, and achieve good load following, but also can maintain the throttle pressure reasonably stable.

4.2 Application of MFGS PI and Gray Predictive PI

So far we have discussed MFGS PI and gray predictive PI. Now we will apply them to the BTG control system, and. through some simulation cases, we will show that they can achieve the control goals described in the beginning of this chapter.

We will use the model developed by Bell and Astrom [22] for a 160 MW fossil-fueled power generation unit. It is described by

$$
\begin{aligned}
&dx_1/dt = -0.0018x_1^{9/8} + 0.9f_b - 0.15v_f,\\
&dx_2/dt = (0.073v_t - 0.016)x_1^{9/8} - 0.1x_2,\\
&dx_3/dt = (141v_f - (1.1v_t - 0.19)x_1)/85,\\
&P_d = x_1,\\
&N\ = x_2,\\
&L_d = 0.05(0.13073x_3 + 100a_c + q_c/9 - 67.975,\\
&a_c = \frac{(1 - 0.001583x_3)(0.8x_1 - 25.6)}{x_3(1.0394 - 0.0012304x_1)},\\
&q_c = (0.854v_t - 0.147)x_1 + 45.59f_b - 2.514v_f - 2.096,
\end{aligned}
\tag{4.2a}
$$

where the state variables x_1, x_2 and x_3 denote the drum steam pressure (kgf/cm^2). the electric power(MW) and the mean system fluid density (kg/m^3); P_d, N and L_d are the drum steam pressure (kgf/cm^2), the electric power(MW) and the drum level(mm): f_b, v_t and v_f denote the fuel flow, governor opening and feed-water actuator position. f_b, v_t and v_f are all scaled to the range of 0% - 100%, and their rates are limited to

$$0\% \le f_b \le 100\%, \qquad |f_b| \le 0.7\%/\text{sec},$$
$$0\% \le v_t \le 100\%, \qquad |v_t| \le 2\%/\text{sec},$$
$$0\% \le v_f \le 100\%, \qquad |v_f| \le 5\%/\text{sec}. \qquad (4.2b)$$

The BTG system will operate under boiler following, variant-pressure operation mode. This is a special case of boiler-following control strategy discussed in the previous section. Variant-pressure operation means that when the load demand changes, we change the steam pressure and keep the opening of the governor valve unchanged at steady states. This operation mode can effectively reduce pressure loss, and speed up load tracking. To ensure a good load tracking, we use the boiler following mode, i.e. use the error signal of throttle pressure as the input signal of fuel controller to control fuel, and use the error signal of electric power as the input signal of turbine controller to control governor valve. Once the load demand changes, the turbine controller temporarily changes the opening of governor valve so that the electric power can track the load demand quickly, then the fuel controller changes fuel to achieve energy balance. If the feed-forward load/throttle-pressure-set-point function is properly tuned, the governor opening should be almost unchanged at final steady state. Figure 4.1 shows the basic control scheme for boiler-following mode with variant-pressure operation. The function of "ramp" limits the rate of load demand, and the function of "Fcn Gen" generates the corresponding pressure setpoint in terms of load demand.

Because of the limitation of Model (4.2), the drum steam pressure P_d will be used as the throttle pressure P_t in the following simulations. If the governor opening is kept unchanged. the drum steam pressure P_d and throttle pressure P_t approximately satisfy [20]

$$P_t(s)/P_d(s) = f(F_s),$$

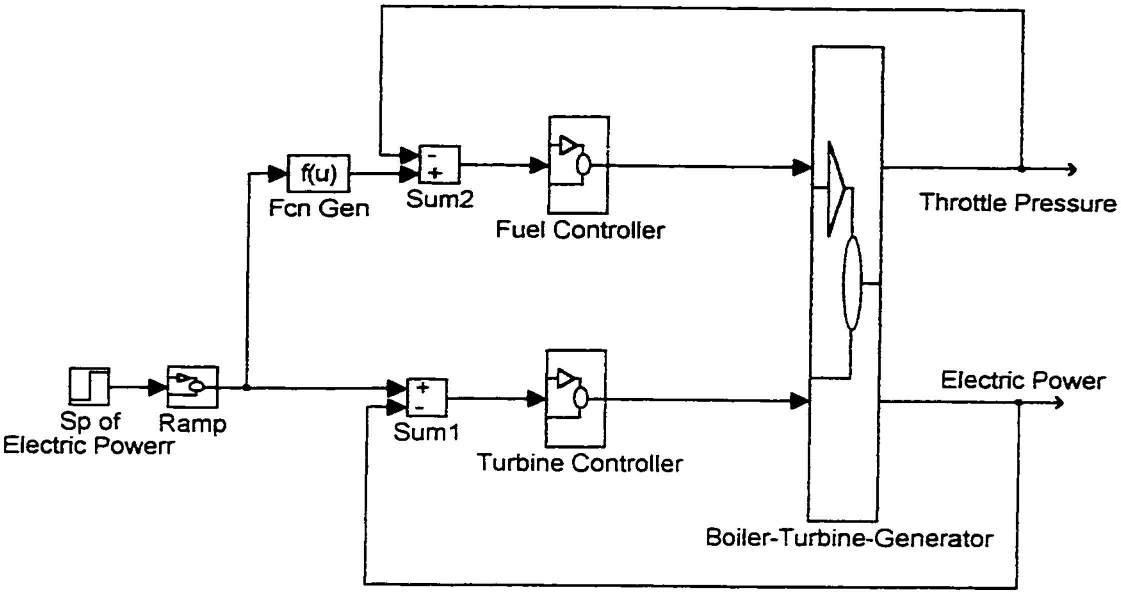


Figure 4.1 Basic Structure of Boiler Following Variant Pressure Operation

where $f(F_s)$ is a non-linear function of steam flow F_s, thus for the real P_t control, if changing the fuel controller gain K_c to $K_c/f(F_s)$, we can get almost the same results.

Suppose that the following relation between the electric power N and throttle pressure P_t will be maintained, which will keeps the governor opening at about 75%.

N	58 MW	76 MW	101 MW
Pt	86 kgf//cm^2	109 kgf//cm^2	140 kgf//cm^2

For the intermediate values of N, we use "Fcn Gen" -- linear interpolation to get P_t.

"Single-element" scheme for the drum level control is used. That is, only the drum level with no pressure correction is used to control the feed-water valves or/and the feed-water pumps, for we only investigate here whether the gray predictive PI can improve the control quality, and enhance the stability of the BTG system. The control criterion for the drum level is to maintain the error within ±50 mm in the static state of load demand, and allow short time over ±50 mm but within ±100 mm during transient state.

The drum level controller used here is the PI controller with fixed parameters shown below

$$v_f(t) = K_c e(t) + (1/T_i)\int e(t)dt,$$
$$K_c = 80,\ T_i = 10,$$

in which $v_f(t)$ is the feed-water actuator position and e(t) drum level error. The drum level is maintained near its setpoint, -100 mm, which is below the middle level of the drum 100 mm. Usually the 0 level is defined as the middle level of a drum in power plants.

The MFGS PI controller shown in (2.9) is used for the turbine control, while the PI controller with fixed parameters shown below is used for the fuel control.

$$v_b(t) = K_c\, e(t) + (1/\,T_i\,)\int e(t)dt,$$

where $v_b(t)$ is the scaled fuel flow and e(t) is the error of throttle pressure. The control criteria are 10% overshoots for the unit-step responses of electric power and throttle pressure, respectively.

At N = 76 MW the parameters of the turbine controller and throttle pressure controller are

$$\begin{array}{lll} \text{Fuel Controller:} & K_c = 8.246, & T_i = 102.394, \text{ and} \\ \text{Turbine Controller:} & K_c = 4.585, & T_i = 16.383. \end{array} \tag{4.3}$$

Figure 4.2 (a), (b), (c) and (d) show the BTG responses for the step change of electric power at both directions, and Figure 4.3 (a), (b), (c) and (d) show the BTG responses for the step change of throttle pressure also at both directions.

Similarly we have the following parameters for the turbine controller and throttle pressure controller at N = 58 MW and N = 101 MW.

$$\begin{array}{lll} \text{N = 58 MW:} & & \\ \text{Fuel Controller:} & K_c = 8.246, & T_i = 102.394, \\ \text{Turbine Controller:} & K_c = 5.707, & T_i = 16.383, \end{array} \tag{4.4}$$

$$\begin{array}{lll} \text{N = 101 MW:} & & \\ \text{Fuel Controller:} & K_c = 8.246, & T_i = 102.394, \\ \text{Turbine Controller:} & K_c = 3.581, & T_i = 16.383, \end{array} \tag{4.5}$$

then we get the almost same step responses as Figure 4.2 and 4.3 at N = 58MW and N = 101MW.

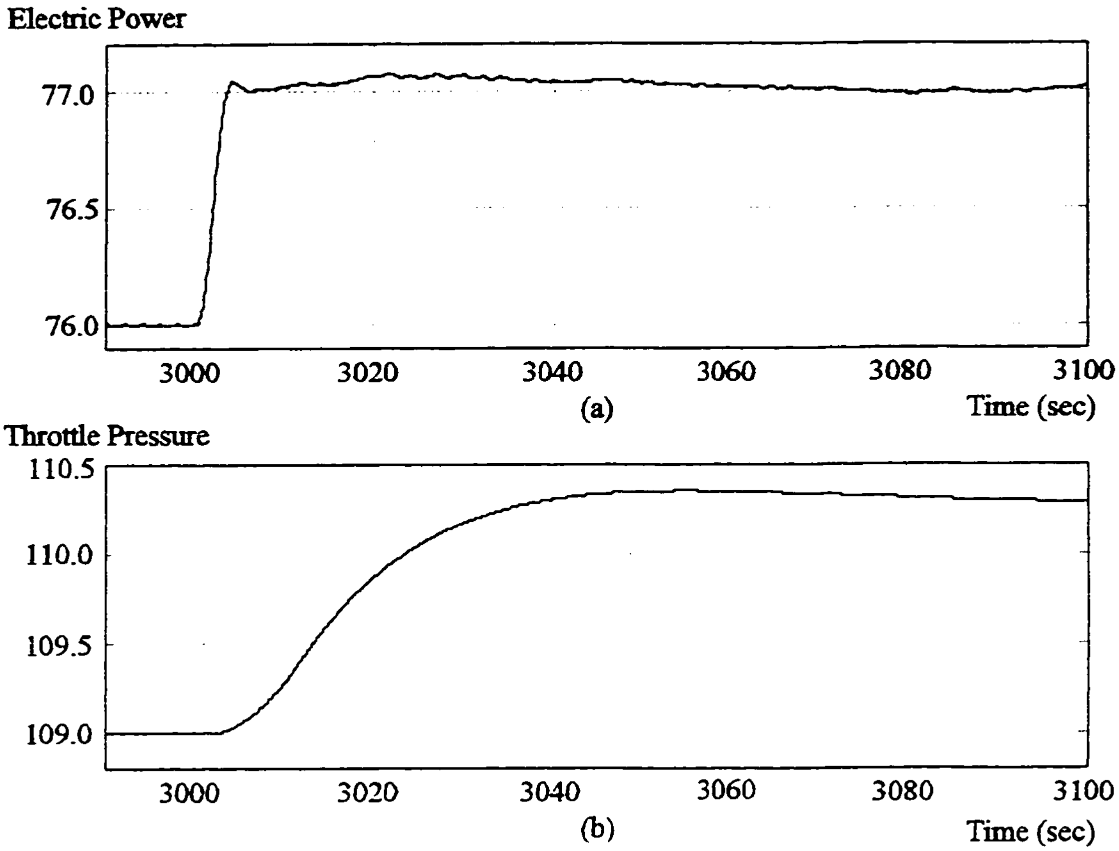


Figure 4.2 Step Responses of BTG System at 70 MW --- Electric Power (Con't)

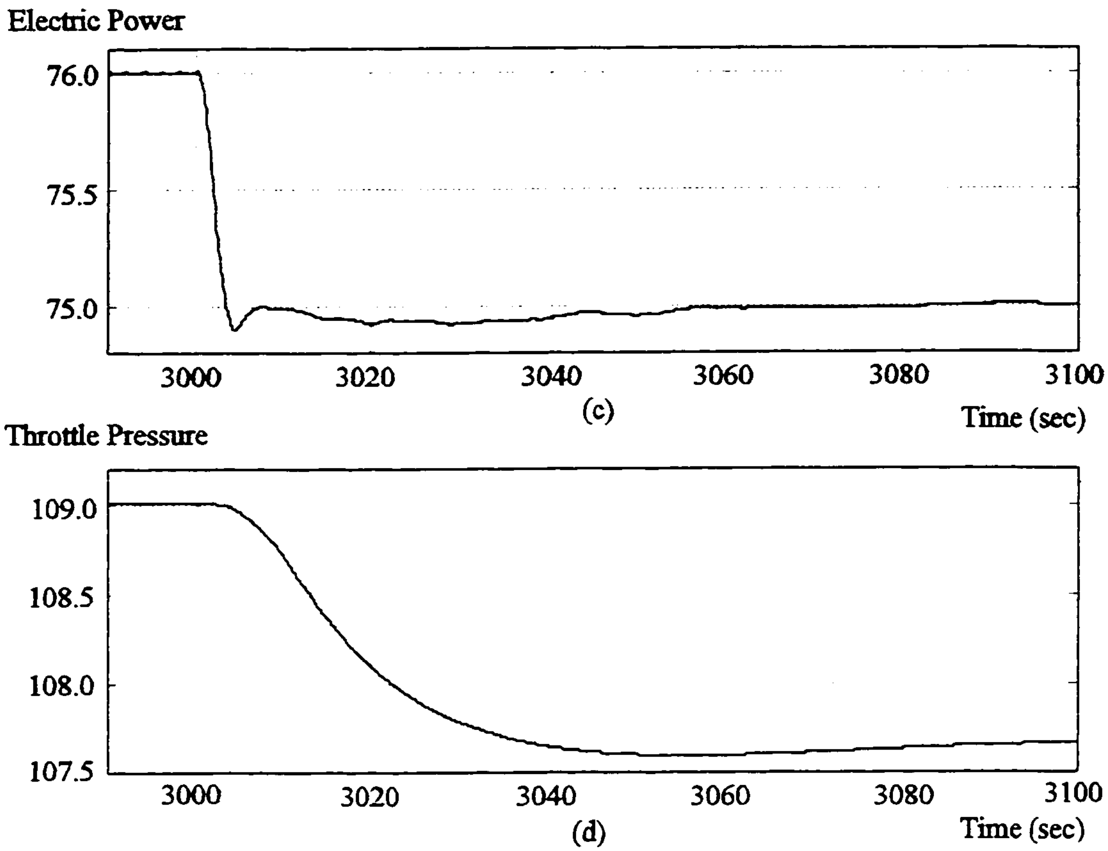
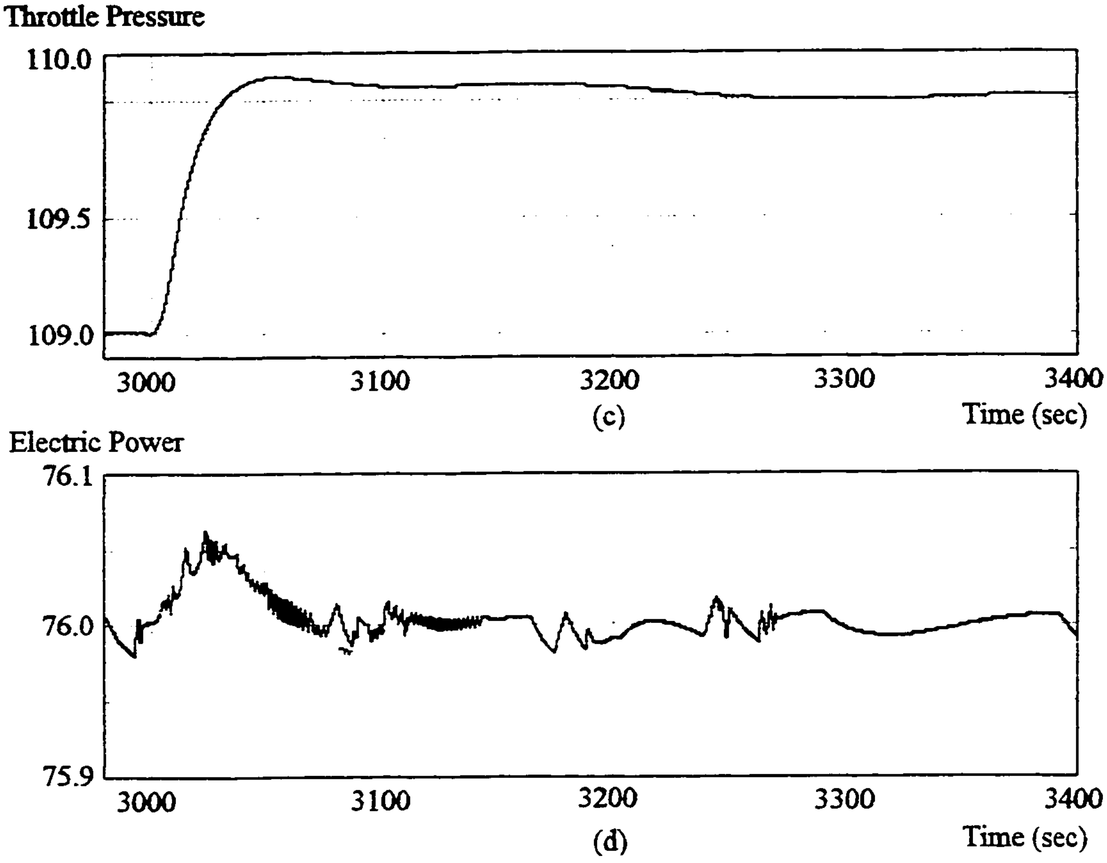


Figure 4.2 Step Responses of BTG System at 70 MW — Electric Power



(c)

(d)

Figure 4.3 Step Responses of BTG System at 70 MW — Throttle Pressure (Con't)

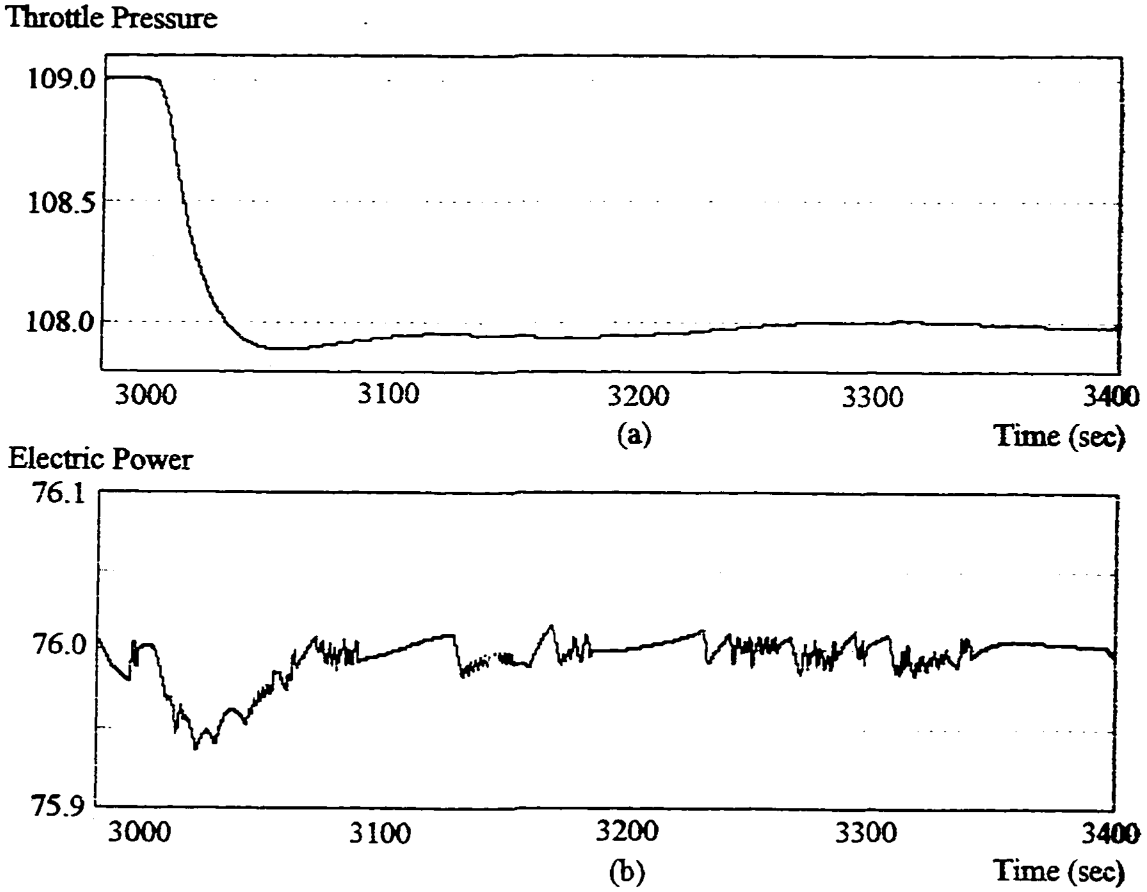


Figure 4.3 Step Responses of BTG
System at 70 MW --- Throttle Pressure

It can be seen that, from (4.3) to (4.5), the controller gain of turbine controller is not constant at different operating points of electric power. From the BTG model (4.2a) we know that if we use governor valve to control electric power, the process gain is $0.073X_1^{9/8}$ or $0.073P_t^{9/8}$. It is a non-linear function of P_t. If using MFGS shown in (2.9) and K_c at N = 58MW and N = 101MW to schedule K_c at N = 76MW, we have

$$K_c(P_t{=}109\text{kgf/cm}^2)^{-1} = K_{c2}^{-1} + (K_{c1}^{-1} - K_{c2}^{-1})(P_t - 86)/54 \qquad (4.6)$$

in which $K_{c1} = 3.581$, $K_{c2} = 5.707$, and P_t is the scheduling variable, so we have

$$K_c(P_t{=}109\text{kgf/cm}^2) = 4.555,$$

which is slightly lower than $K_c = 4.585$ at 76MW. If using FGS1 and FGS2 in (2.6) and (2.7), we have

$$K_c(P_t{=}109\text{kgf/cm}^2) = K_{c2} + (K_{c1} - K_{c2})(P_t - 86)/54 = 4.802.$$

It is slightly larger than $K_c = 4.585$ at 76MW. Although, for this BTG model, the scheduled K_c in MFGS is not significantly different from the scheduled K_c in FGS1 or FGS2 as shown in Figure 4.4, we still suggest to use MFGS PI as the turbine controller in the coordinated control scheme shown in Figure 4.1, for FGS1 PI and FGS2 PI, after all, may lead to higher over-shoot, and tend to cause the system unstable for time-varying process gains.

Since the BTG is operated under variant-pressure mode, we can treat the opening of governor valve V_t in the model (4.2a) as a constant, so the PI with fixed parameters will be used as fuel controller.

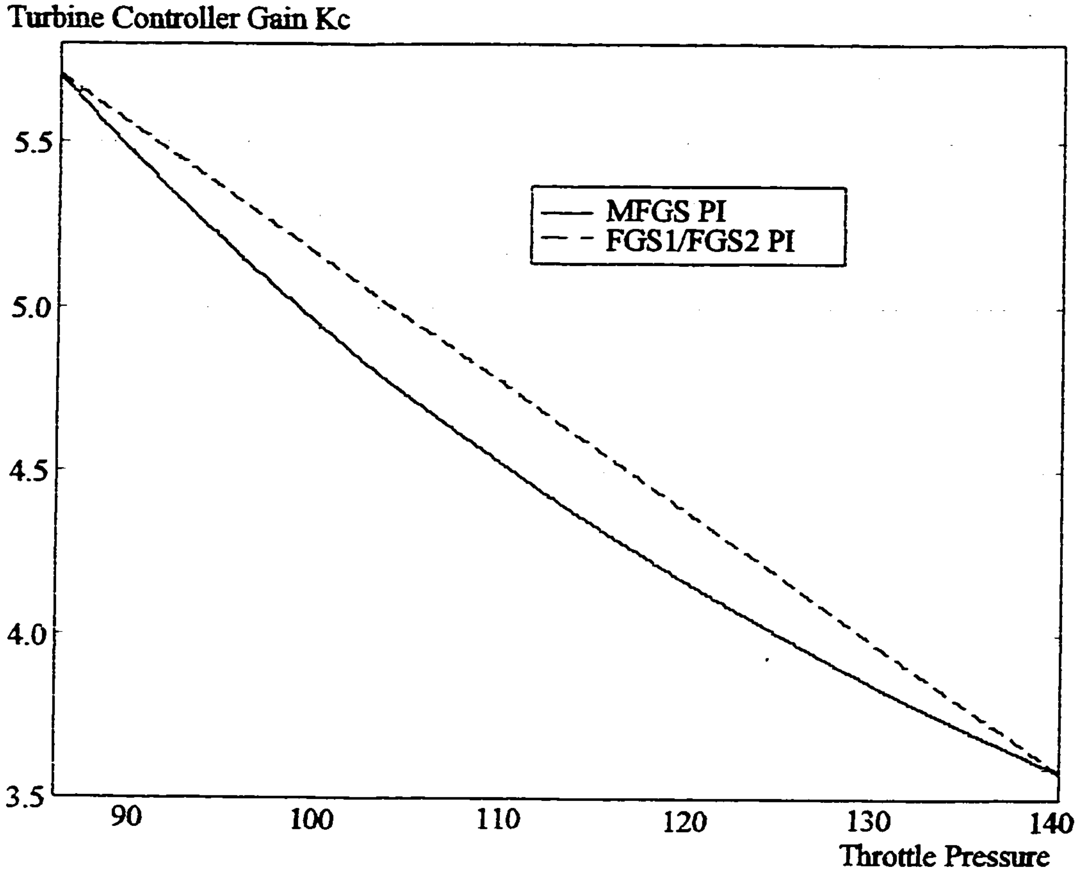


Figure 4.4 Turbine Controller Gains from MFGS, FGS1 and FGS2

So far we have discussed the control scheme and controller parameters for the BTG control, now we will add gray predictive algorithm in the fuel controller, which has fixed controller parameters, to see whether the gray predictive PI can improve the control quality of BTG system and enhance the stability compared with PI with fixed controller parameters. Also we want to see whether the gray predictive PI has almost the same sensitivity to noise as PI. The gain scheduled PI with scheduling algorithm (4.6) will be used as the turbine controllers.

In Simulation 4.1 and 4.3, we will show that although the gray predictive PI and PI have almost the same step responses at three operating points, the gray predictive PI does have better stability for load demand following, and ability for disturbance rejection than PI. In simulation 4.4, also we will demonstrate that the properly designed gray predictive PI has almost the same sensitivity to noise as PI.

In all simulation cases below the predictive horizon of gray predictive PI is set to be 25 sec as the parameter a(i) of GM(1,1) is larger than 0.035. If the parameters a(i) is not larger than 0.035, instead of prediction, we will estimate the current output value. The bias B in the gray predictive PI, shown in Figure 3.14, is set to be 50 kgf/cm^2.

Remark: Similar to "D" mode in PID, the gray prediction should be used to compensate process lag instead of pure time delay, since a controller can not foresee when the process output will change before it changes. This feature is different from some predictive controllers such as Dynamic Matrix Controller and Generalized Predictive Controller etc. in which input signals and internal models are used. This view is supported by a rule of thumb, which states that for first order processes the ratio between the process time constant T(t) and controller derivative time $T_d(t)$ should be kept at a constant value [16].

Simulation 4.1: Similar unit step responses

Figures 4.5 to 4.7 show the step responses of the gray predictive PI and PI, which are used as fuel controllers, at electric power N = 58 MW, 76 MW and 101 MW, respectively. Both controllers have almost the same overshoots, rise time and settling time. The differences in step responses for two controllers at N = 101 MW is somewhat larger than N = 58 MW and N = 76 MW, but is still not significant.

Simulation 4.2: Load Demand Following

Similar to Simulation 4.1, the predictive horizon of gray predictive PI is set to be 25 sec. A colored noise ("Band Limited White Noise" in Simulink 4.2c through a filter) is added into the throttle pressure:

> White Noise:
> Power = 0.2, Sample Interval = 2 sec, and Seed = 23341,
> Filter: 1/(5s+1).

CASE 1: The load demand increases from 76 MW to 101MW at the rate of 9MW/min

Figure 4.8 shows the responses for the electric power and throttle pressure. Figure 4.9 shows the control outputs of the fuel controllers.

CASE 2: The load demand decreases from 101 MW to 76MW at the rate of 9MW/min

Figure 4.10 shows the responses for the electric power and throttle pressure. Figure 4.11 shows the control outputs of the fuel controllers.

CASE 3: The load demand decreases from 76 MW to 58 MW at the rate of 9MW/min

Figure 4.12 shows the responses for the electric power and throttle pressure. Figure 4.13

shows the control outputs of the fuel controllers.

CASE 4: The load demand increases from 58 MW to 76MW at the rate of 9MW/min

Figure 4.14 shows the responses for the electric power and throttle pressure. Figure 4.15 shows the control outputs of the fuel controllers.

Case 1 to Case 4, no matter whether the load demand increases or decreases, demonstrate two common features. One is that the overshoots for both systems under the gray predictive PI control and PI control are almost the same, and the other is that the gray predictive PI only has slightly higher output than PI for a brief period of time. Its sensitivity to noise is almost the same as PI's.

CASE 5: The load demand decreases from 101 MW to 76MW at the rate of 17 MW/min

Figure 4.16 shows the responses for the electric power and throttle pressure. Figure 4.17 shows the outputs of the fuel controllers and turbine controllers.

CASE 6: The load demand decreases from 76 MW to 58 MW at the rate of 25MW/min

Figure 4.18 shows the responses for the electric power and throttle pressure. Figure 4.19 shows the outputs of the fuel controllers and turbine controllers.

CASE 7: The load demand increases from 58 MW to 76 MW at the rate of 22MW/min

Figure 4.20 shows the responses for the electric power and throttle pressure. Figure 4.21 shows the outputs of the fuel controllers and turbine controllers.

CASE 8: The load demand increases from 76 MW to 101 MW at the rate of 17MW/min

Figure 4.22 shows the responses for the electric power and throttle pressure. Figure 4.23 shows the outputs of the fuel controllers and turbine controllers.

Case 5 to Case 8 shows that if load demand increases or decreases at higher rates, the BTG system under PI control becomes unstable, it is still stable under the control of gray predictive PI although the error of the throttle pressure is large, about 8-10kgf//cm^2. We can say that the gray predictive PI is more stable than PI for the different rates of load demand because of its predictive effect.

CASE 9: The load demand decreases from 101 MW to 76MW at the rate of 34MW/min

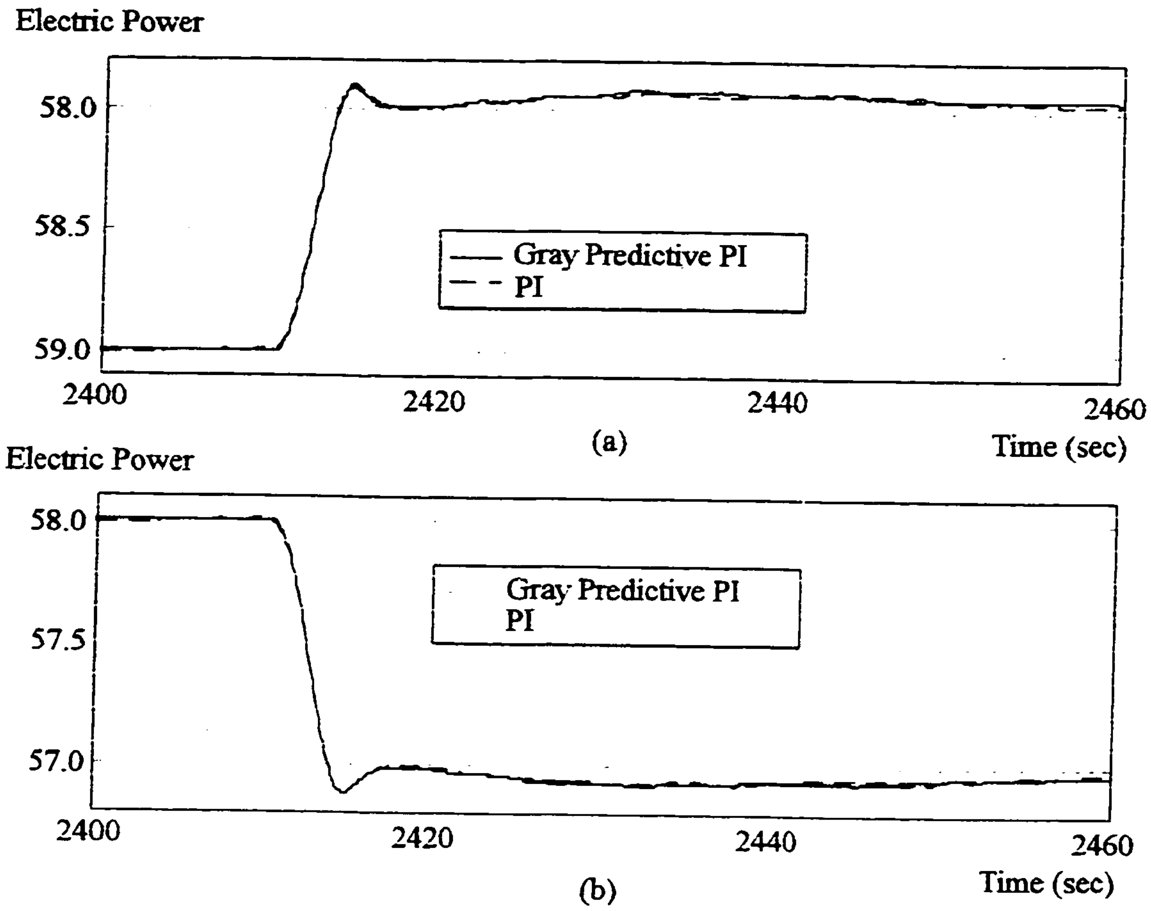
Similar to Case 5, but the rate of load demand is further increased from 17MW/min to 34MW/min. Figure 4.24 shows the responses for the electric power and throttle pressure. Figure 4.25 shows the outputs of the fuel controllers and turbine controllers.

CASE 10: The load demand decreases from 76 MW to 58 MW at the rate of 50MW/min

Similar to Case 5, but the rate of load demand is changed from 25MW/min to 50MW/min. Figure 4.26 shows the responses for the electric power and throttle pressure. Figure 4.27 shows the outputs of the fuel controllers and turbine controllers.

From Case 9 and Case 10, we can see that the gray predictive PI not only is more stable than PI for different rates of the load demand, but also can keep stable for wide ranges of load demand rates.

This simulation indicates that the gray predictive PI can improve the control quality for load following, and enhance the stability of BTG system. Also we can see that a properly designed gray predictive PI has almost the same sensitivity to noise as PI.



(a)

(b)

Figure 4.5 Step Responses of Fuel Controllers at N = 58 MW

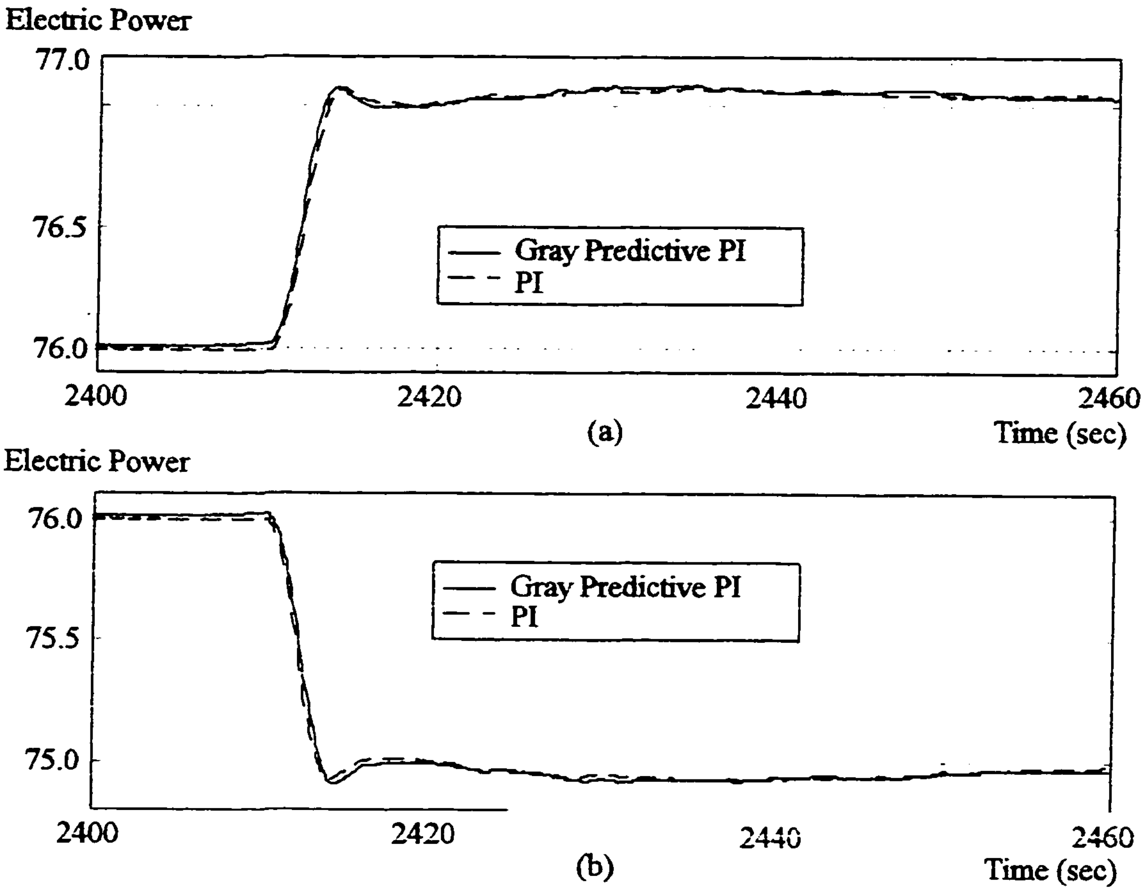


Figure 4.6 Step Responses of Fuel Controllers at N = 76 MW

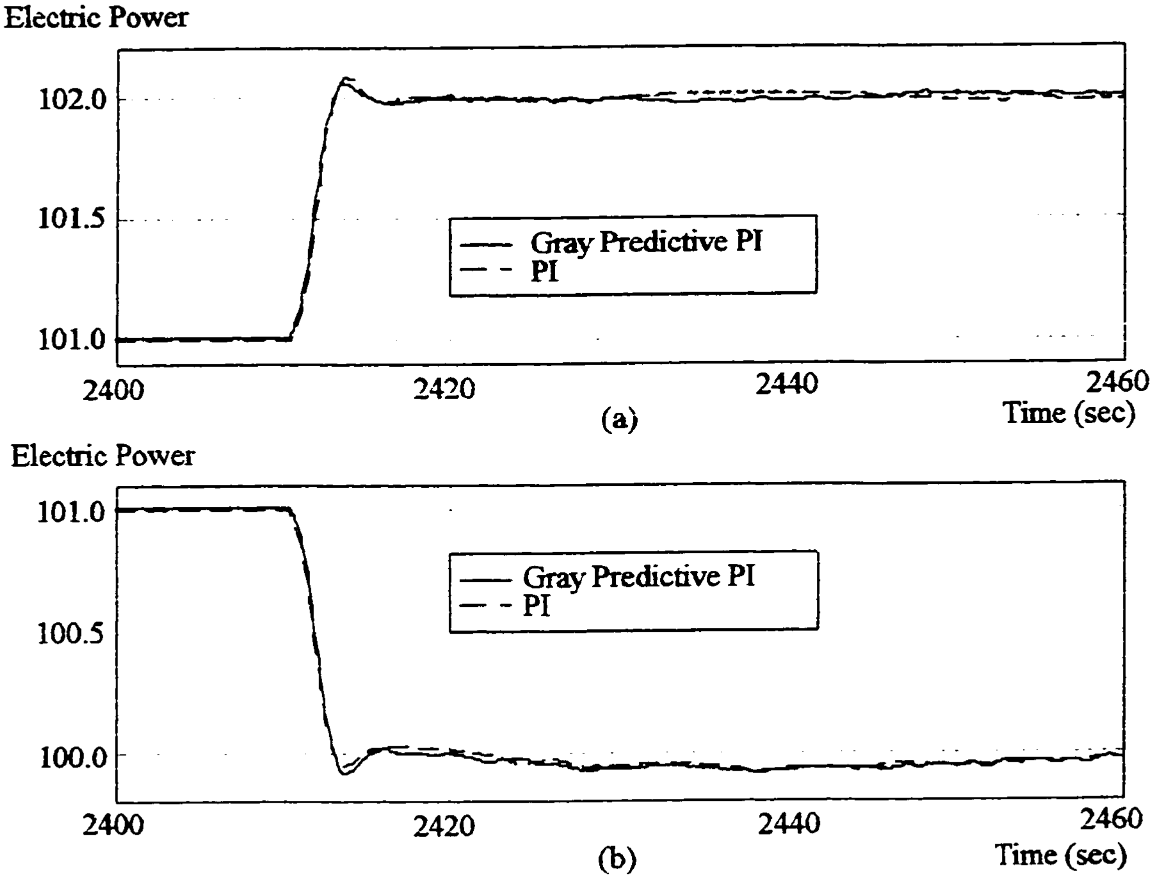


Figure 4.7 Step Responses of Fuel Controllers at N = 101 MW

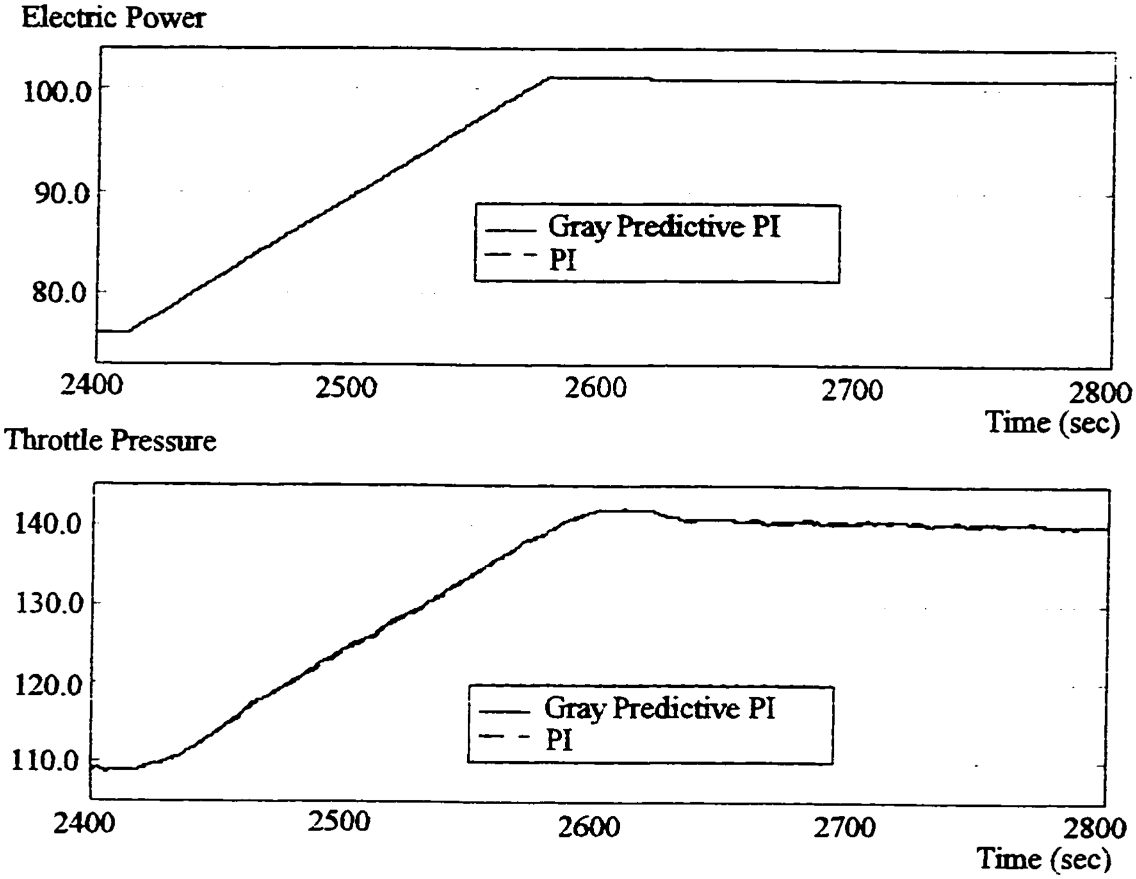


Figure 4.8 Load Demand Increases from 76 MW to 101 MW at the rate of 9 MW/minute

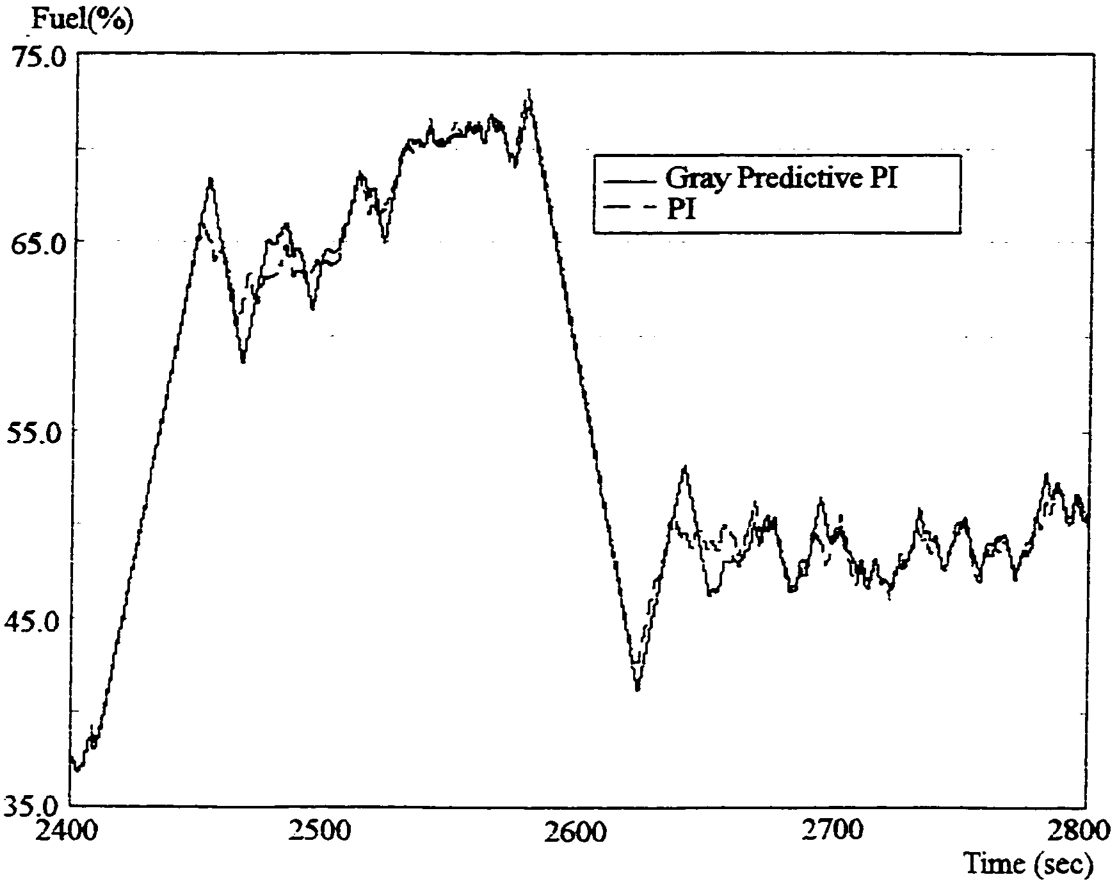


Figure 4.9 Control Outputs of Fuel Controllers

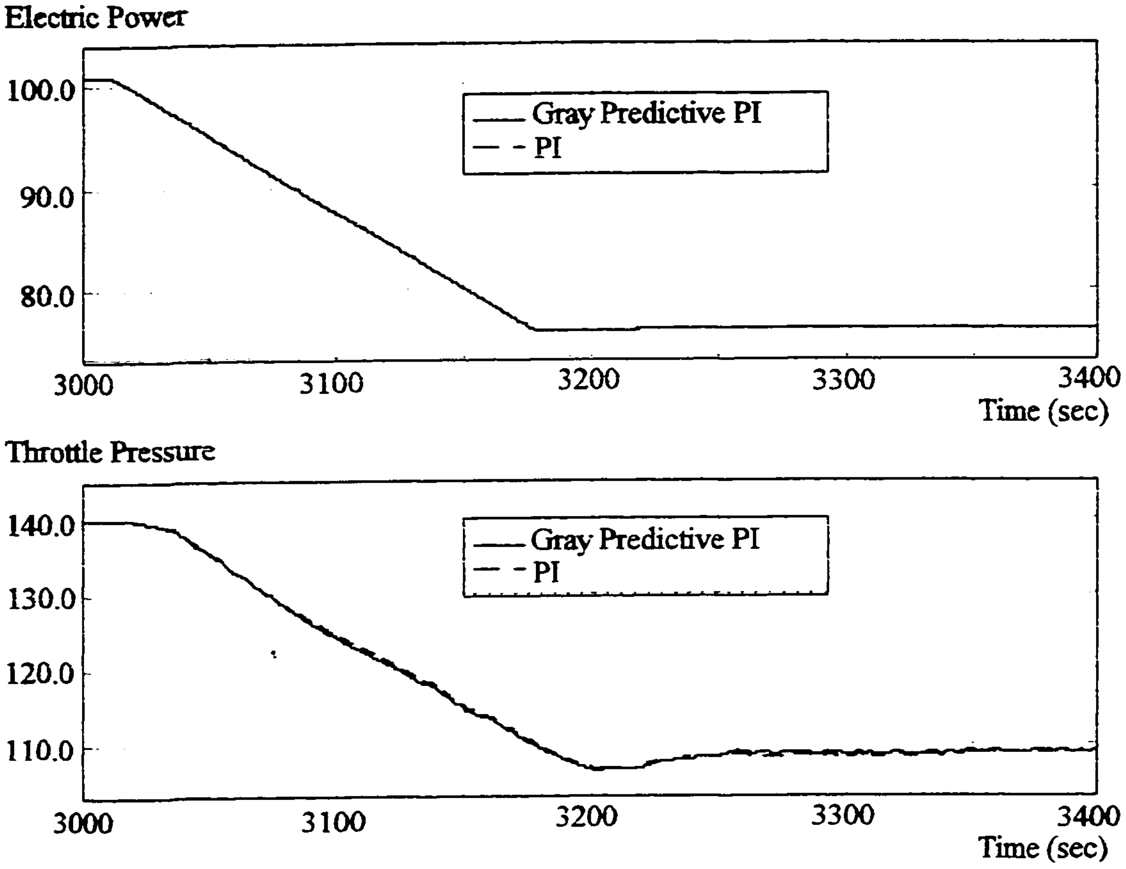


Figure 4.10 Load Demand Decreases from 101 MW to 76 MW at the rate of 9 MW/minute

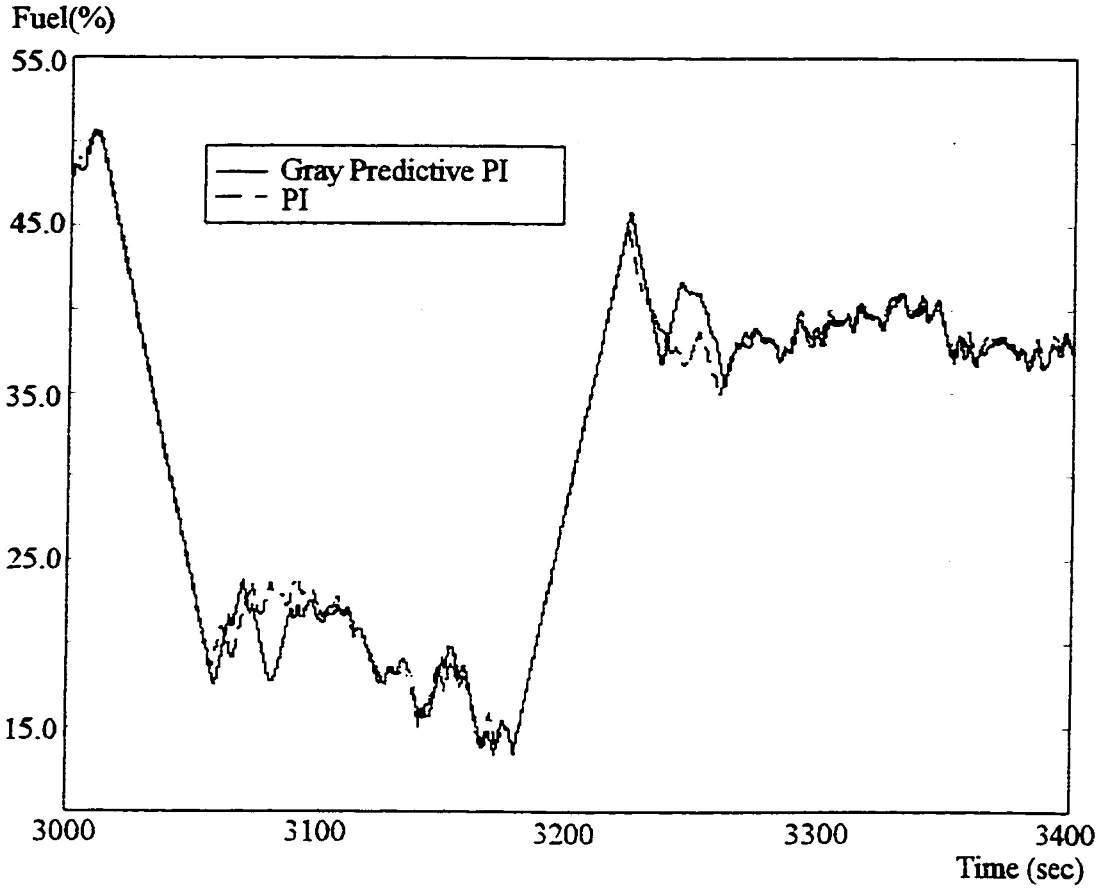


Figure 4.11 Control Outputs of Fuel Controllers

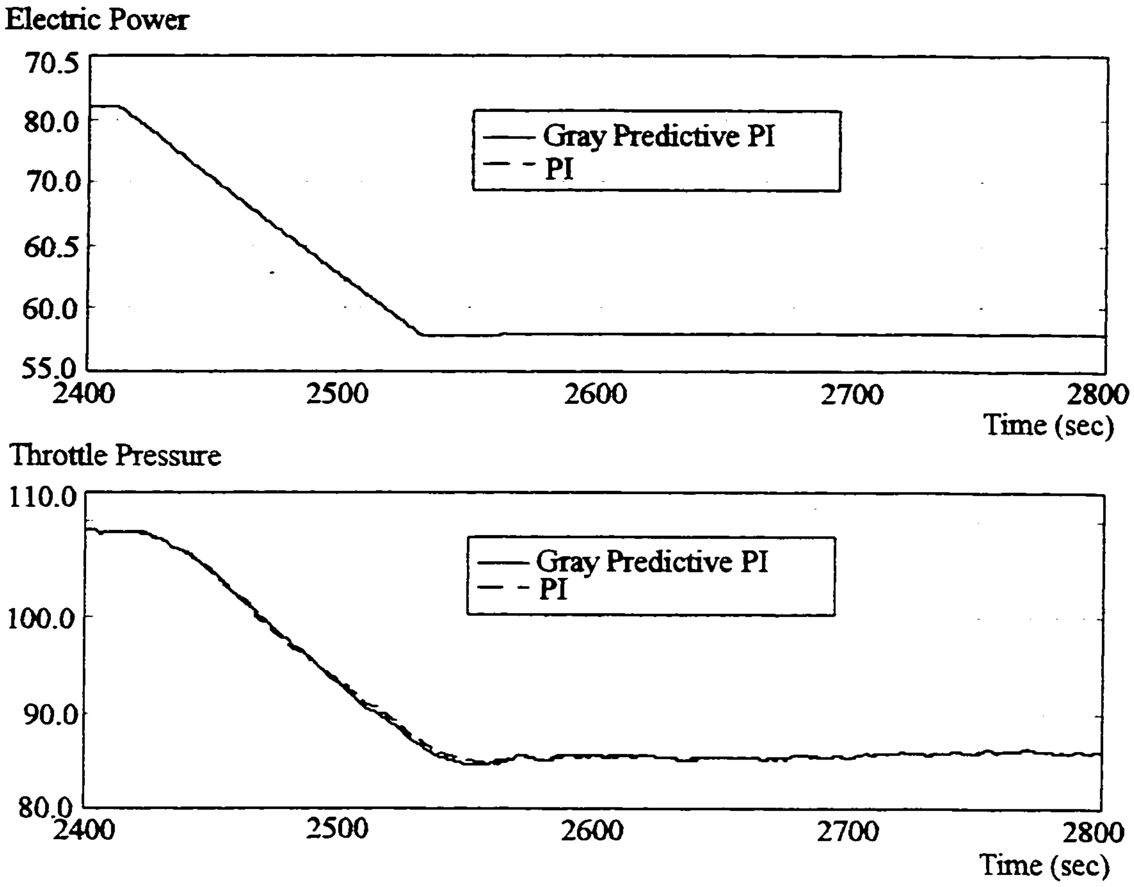


Figure 4.12 Load Demand Decreases from 76 MW
to 58 MW at the rate of 9 MW/minute

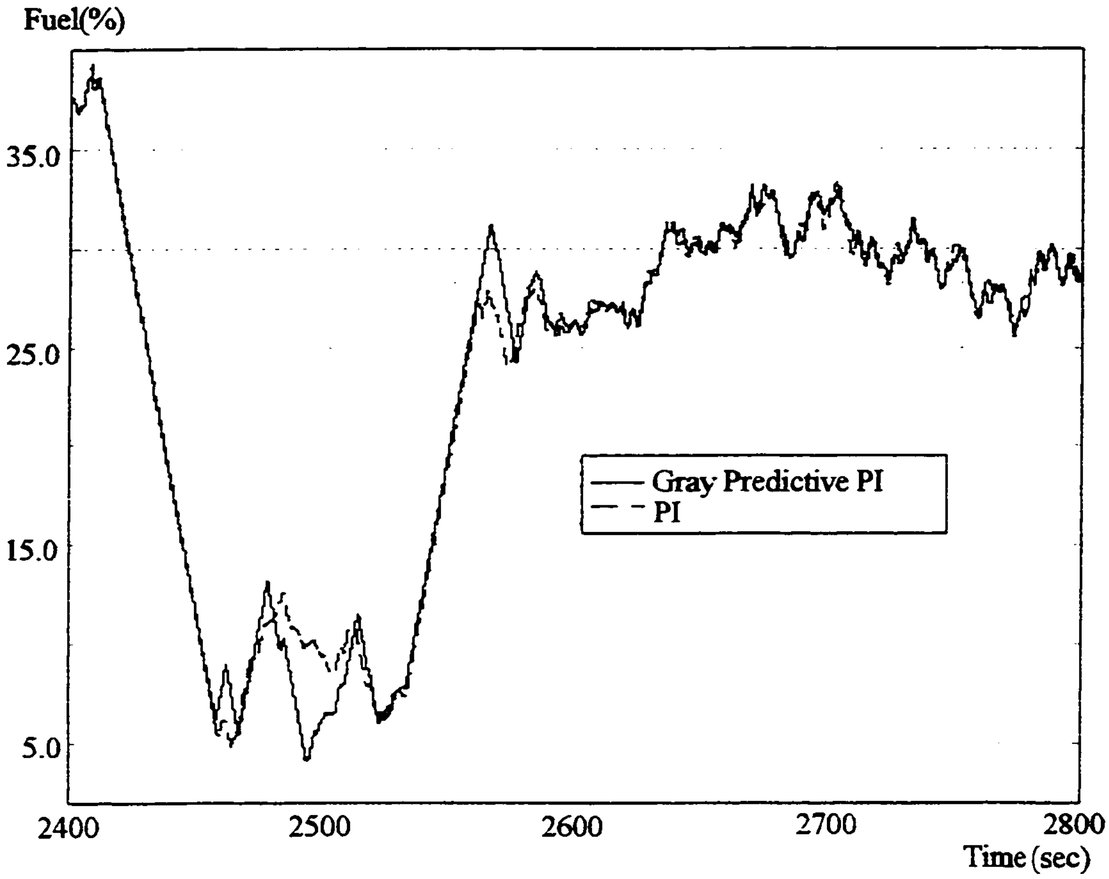


Figure 4.13 Control Outputs of Fuel Controllers

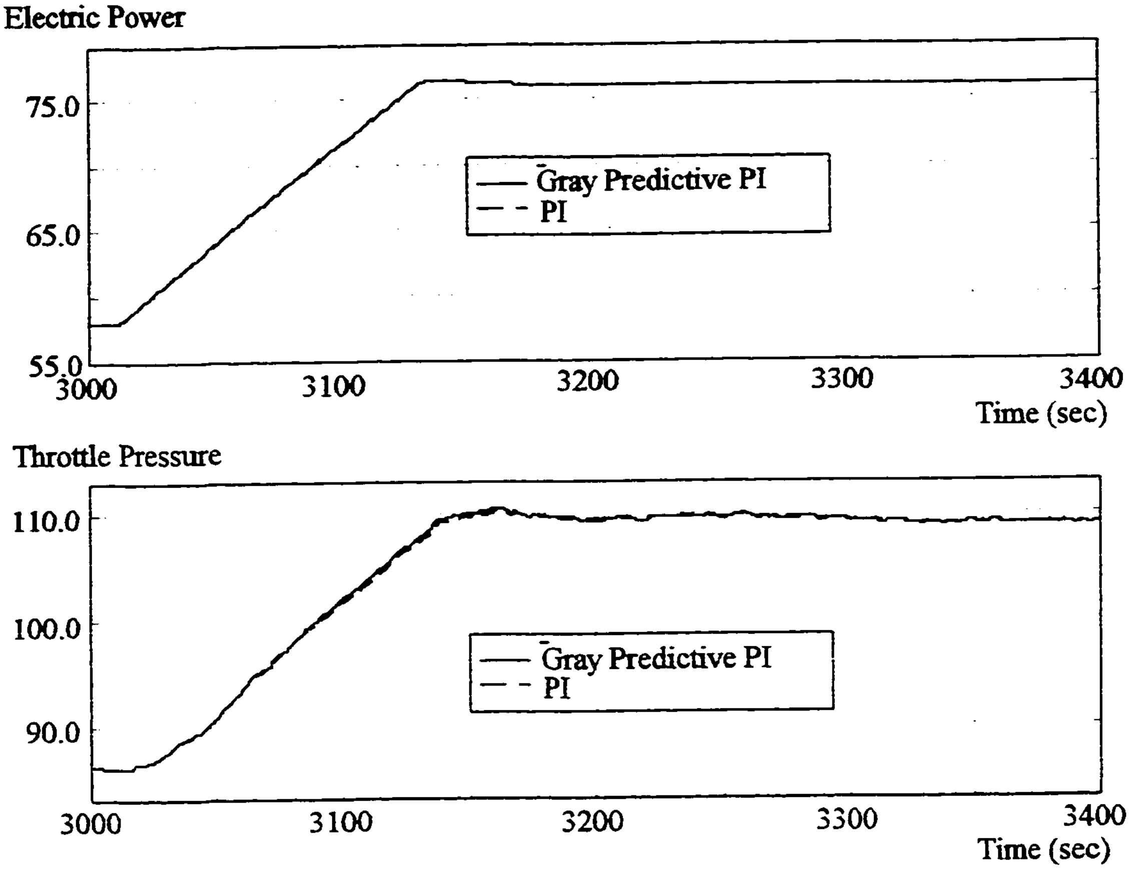


Figure 4.14 Load Demand Increases from 58 MW
to 76 MW at the rate of 9 MW/minute

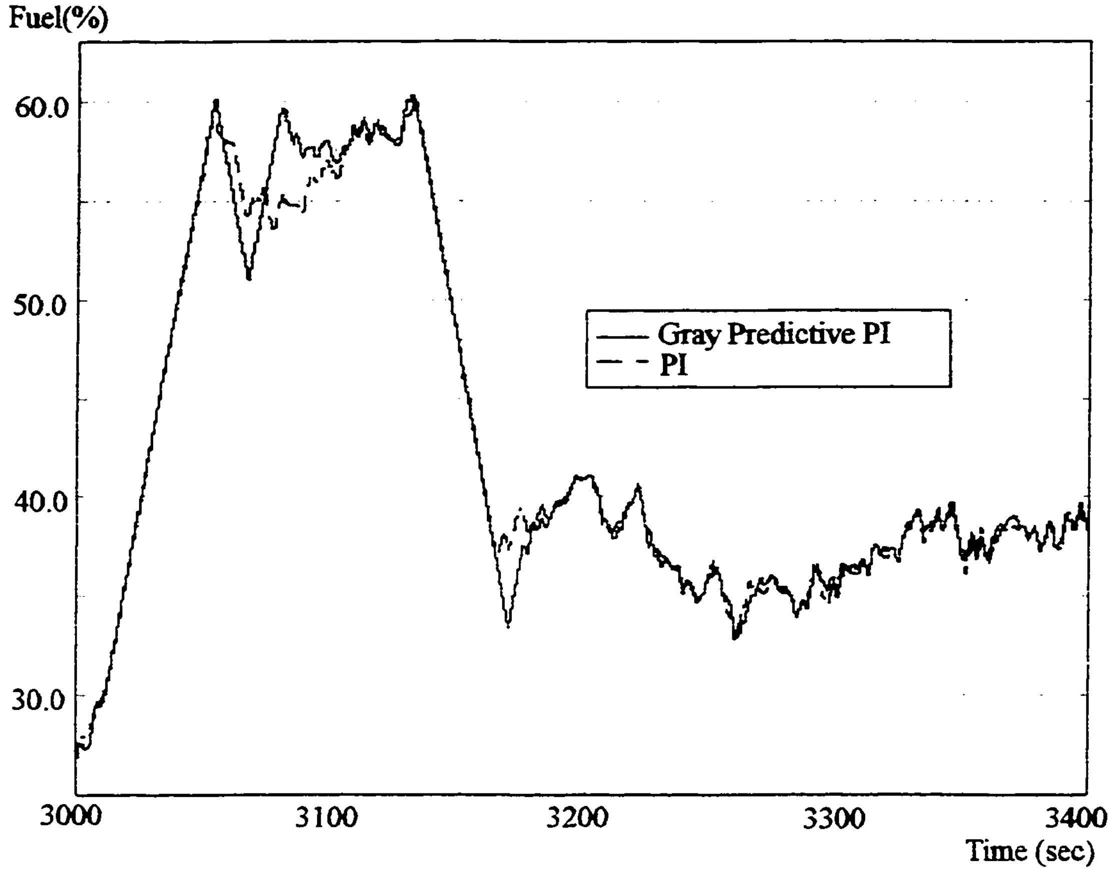


Figure 4.15 Control Outputs of Fuel Controllers

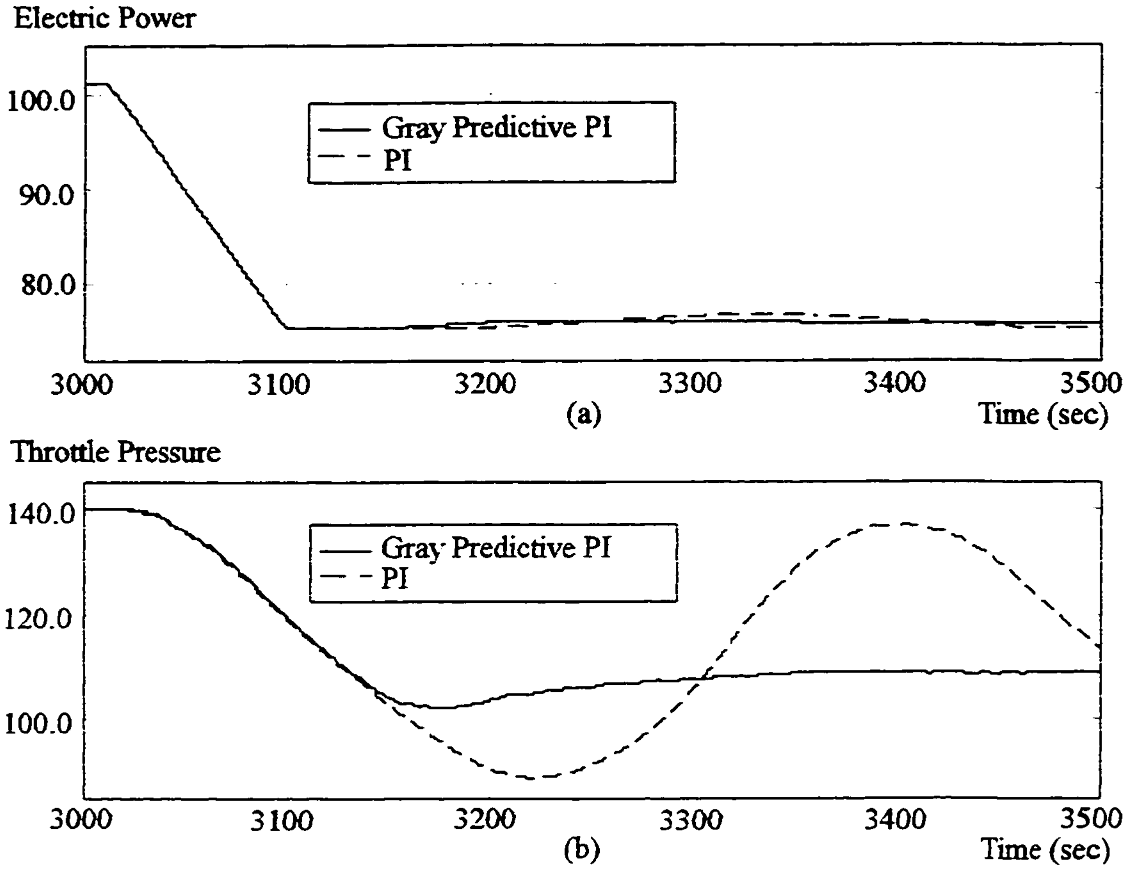


Figure 4.16 Load Demand Decreases from 101 MW to 76 MW at the rate of 17 MW/minute

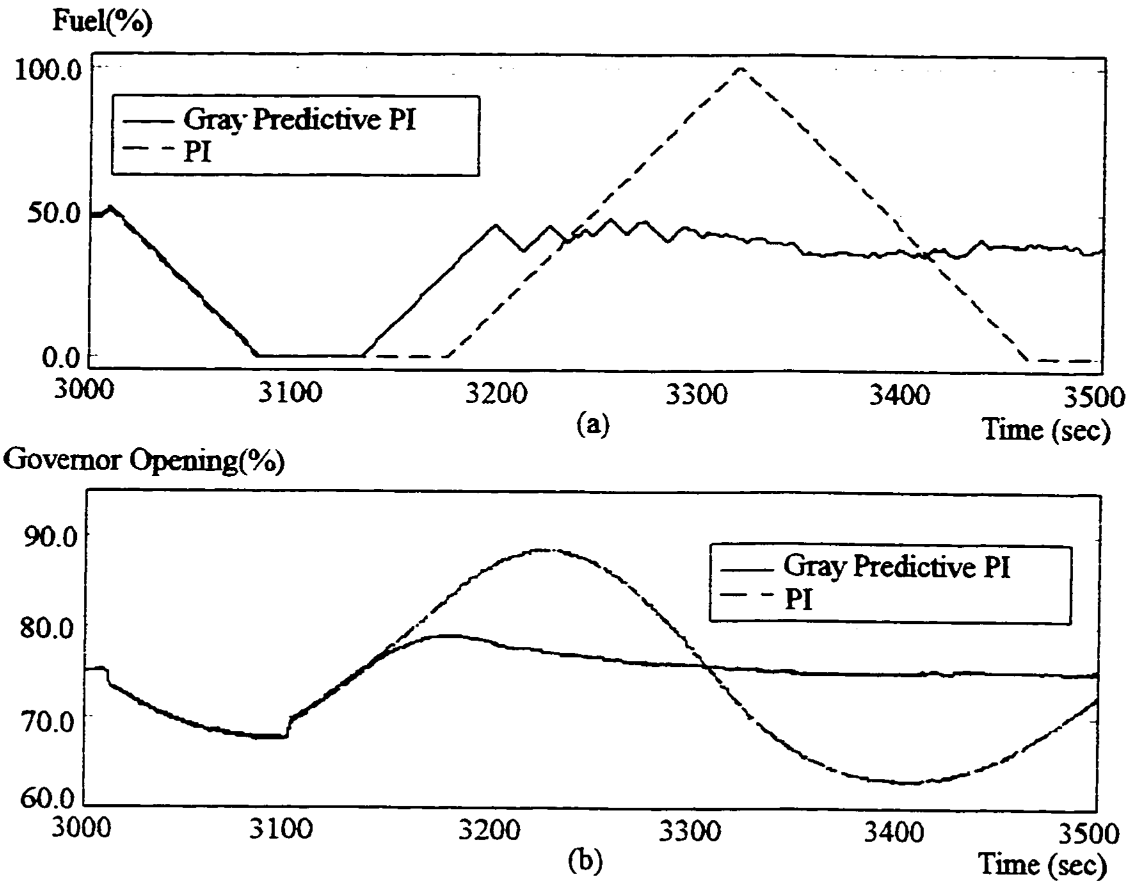


Figure 4.17 Control Outputs of Fuel Controllers and Turbine Controllers

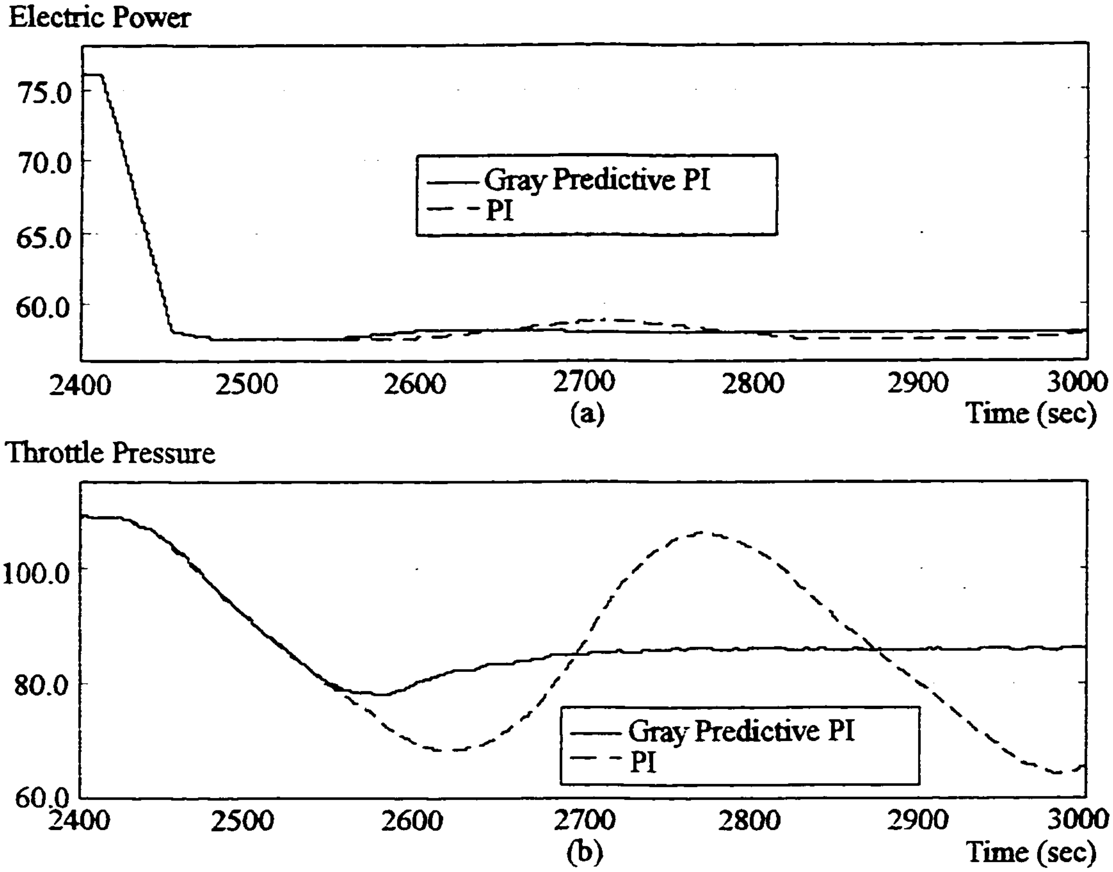


Figure 4.18 Load Demand Decreases from 76 MW to 58 MW at the rate of 25 MW/minute

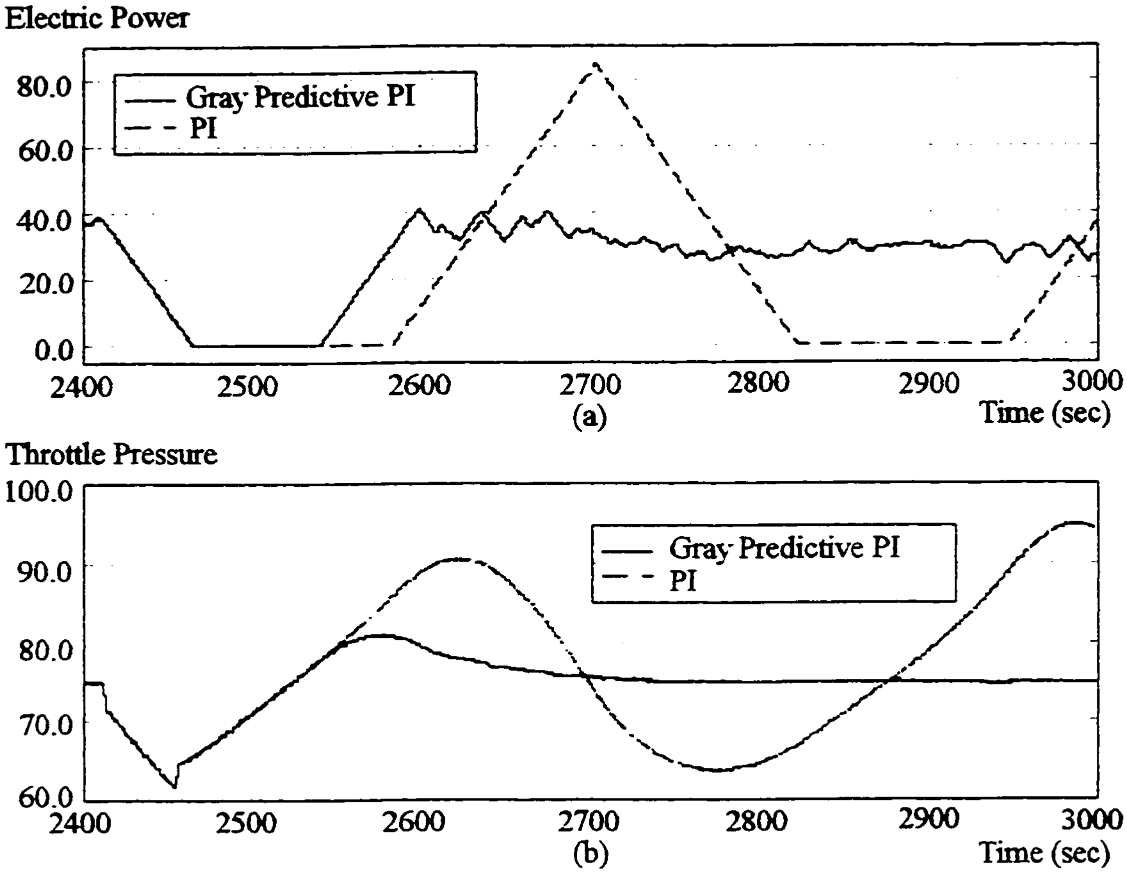


Figure 4.19 Control Outputs of Fuel Controllers and Turbine Controllers

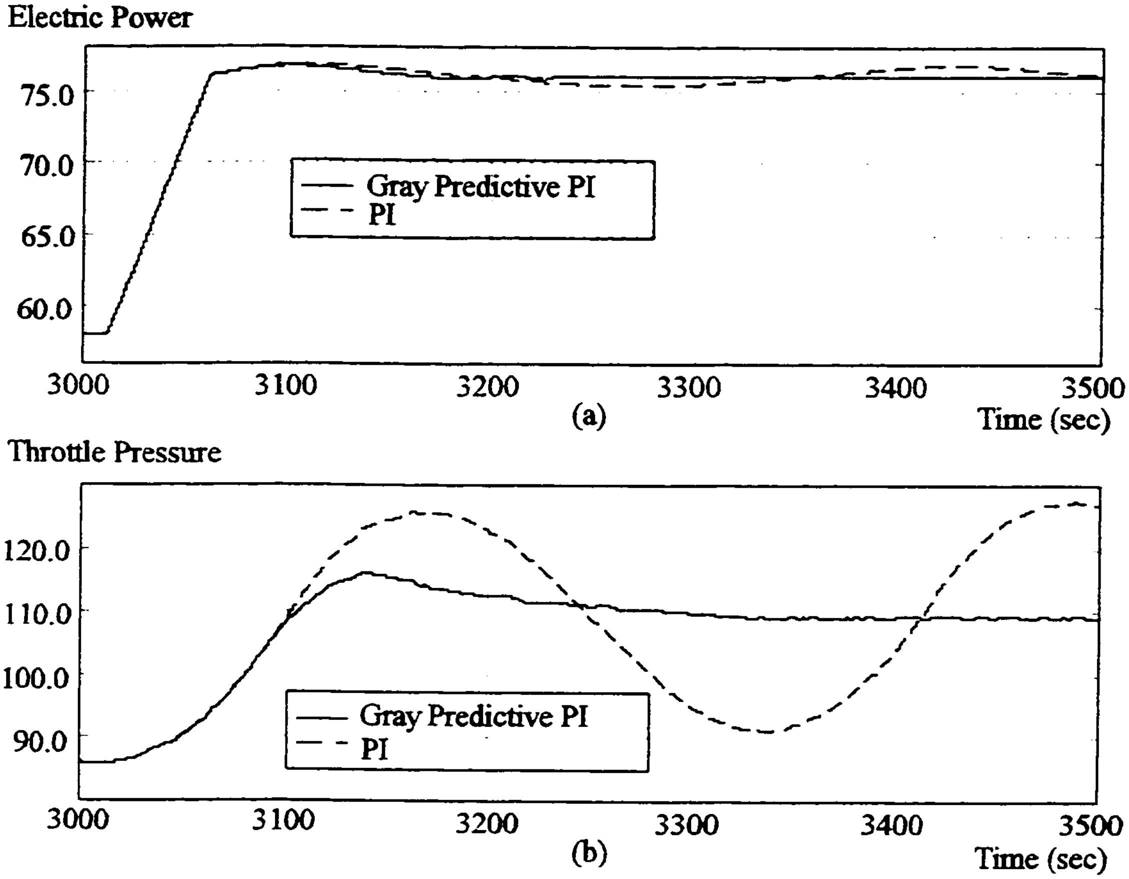


Figure 4.20 Load Demand Increases from 58 MW
to 76 MW at the rate of 22 MW/minute

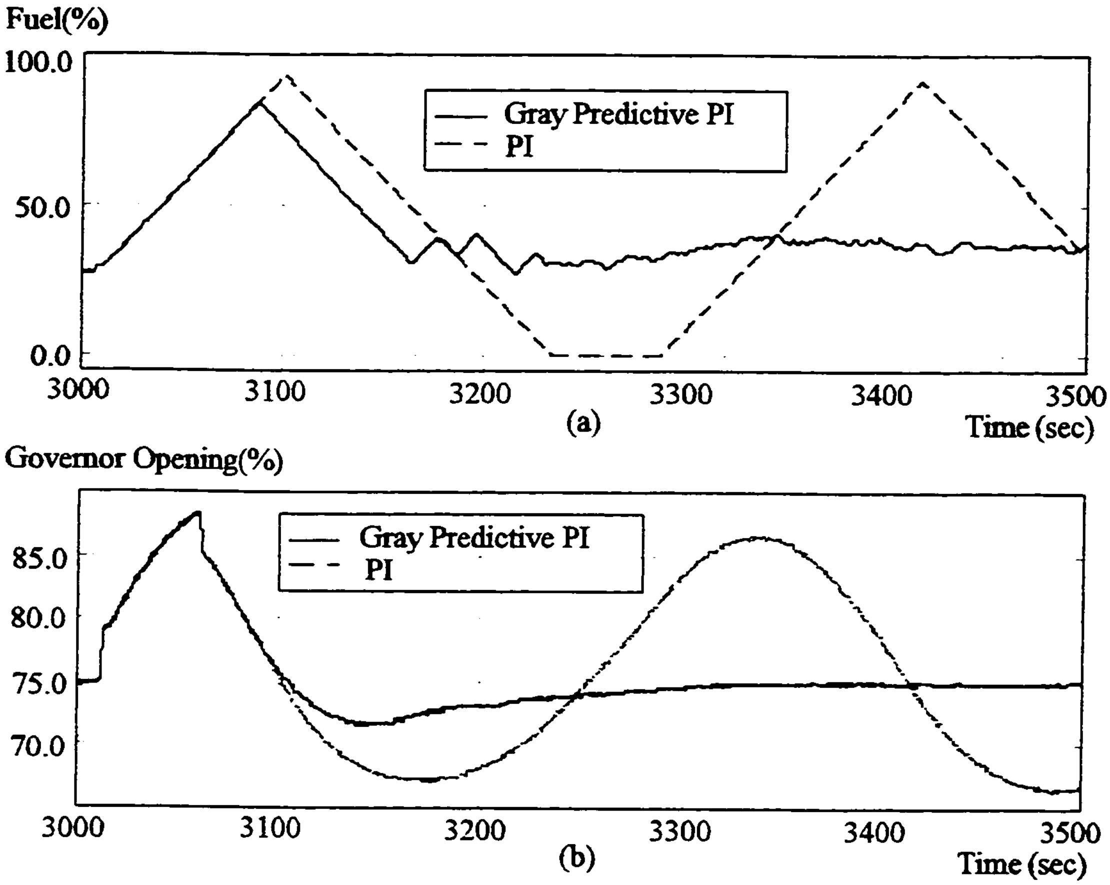


Figure 4.21 Control Outputs of Fuel Controllers and Turbine Controllers

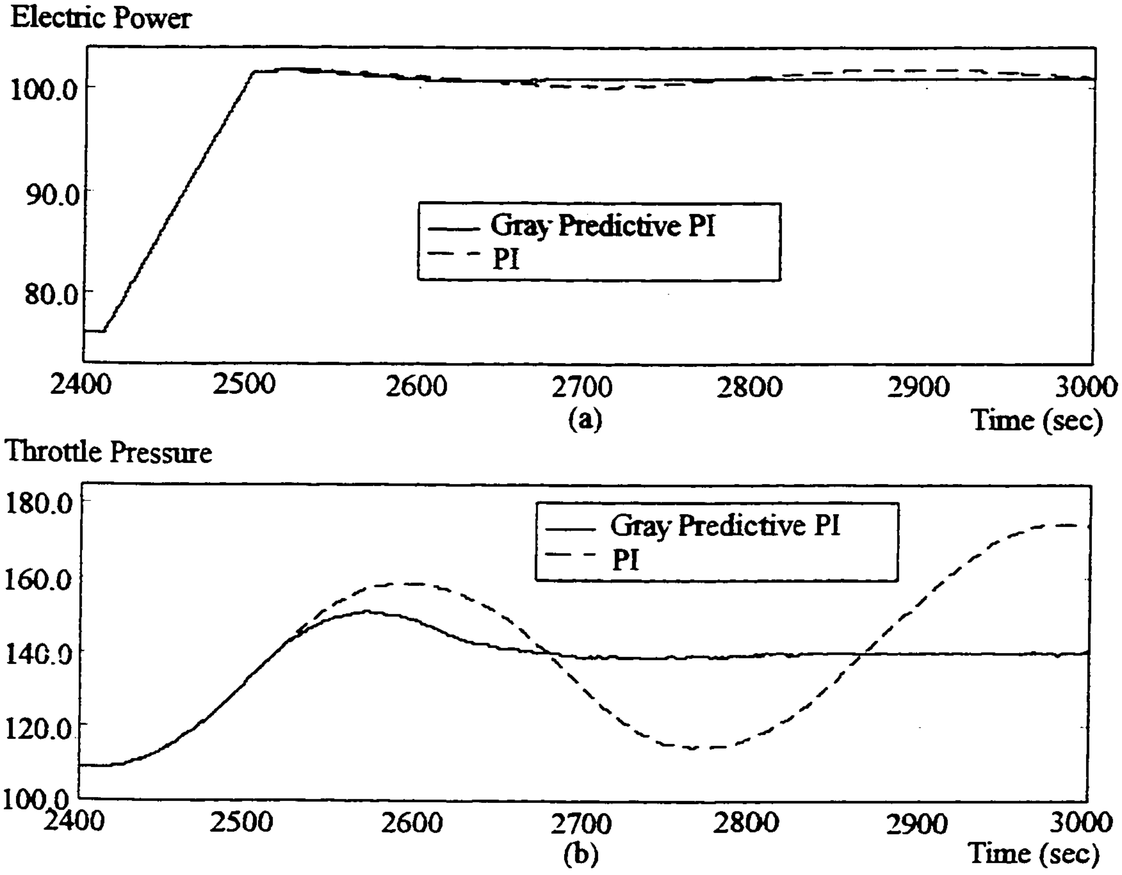


Figure 4.22 Load Demand Increases from 76 MW
to 101 MW at the rate of 17 MW/minute

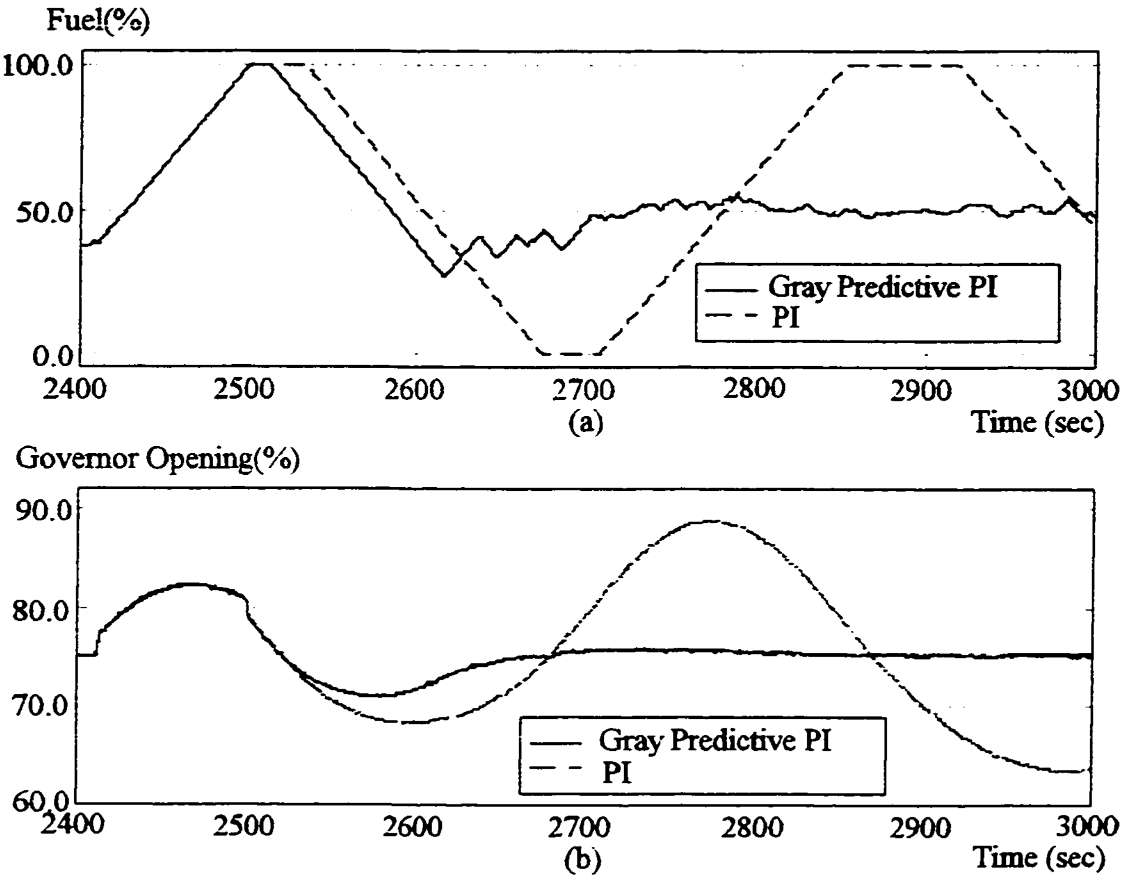


Figure 4.23 Control Outputs of Fuel Controllers and Turbine Controllers

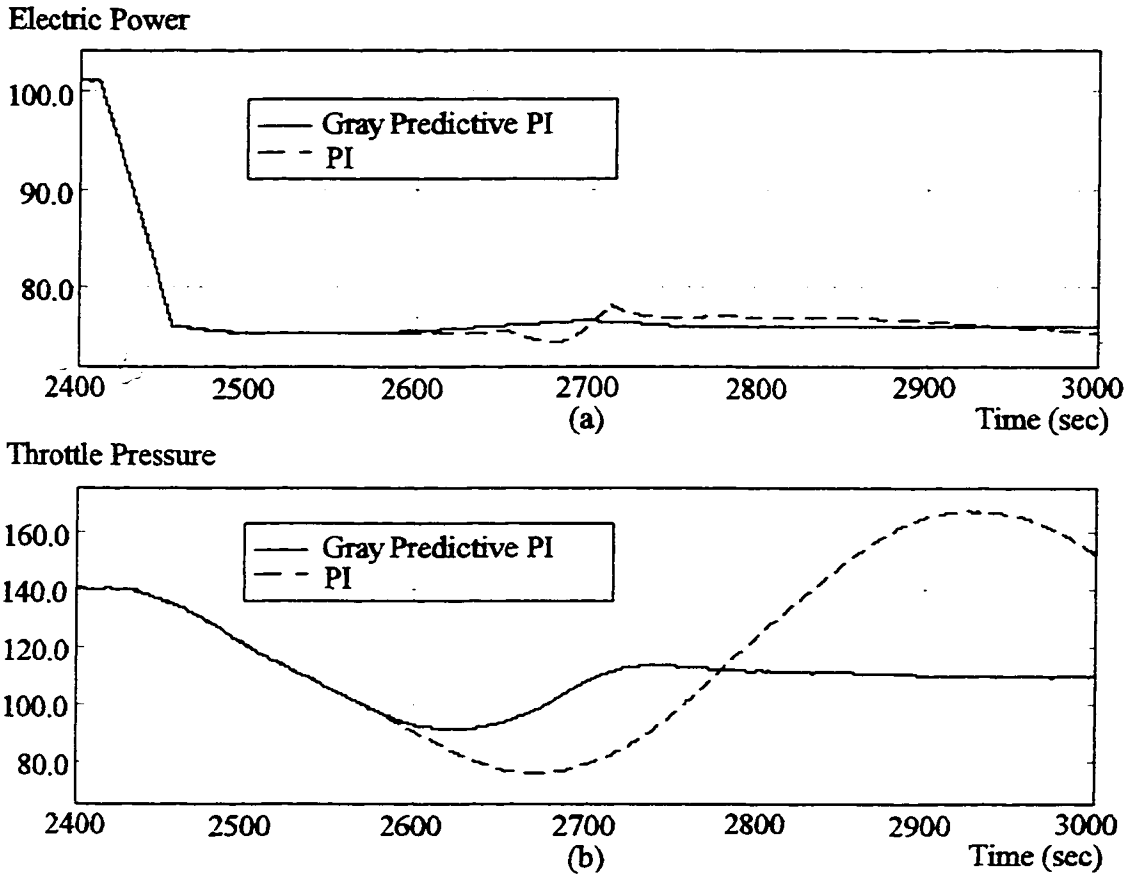


Figure 4.24 Load Demand Decreases from 101 MW
to 76 MW at the rate of 34 MW/minute

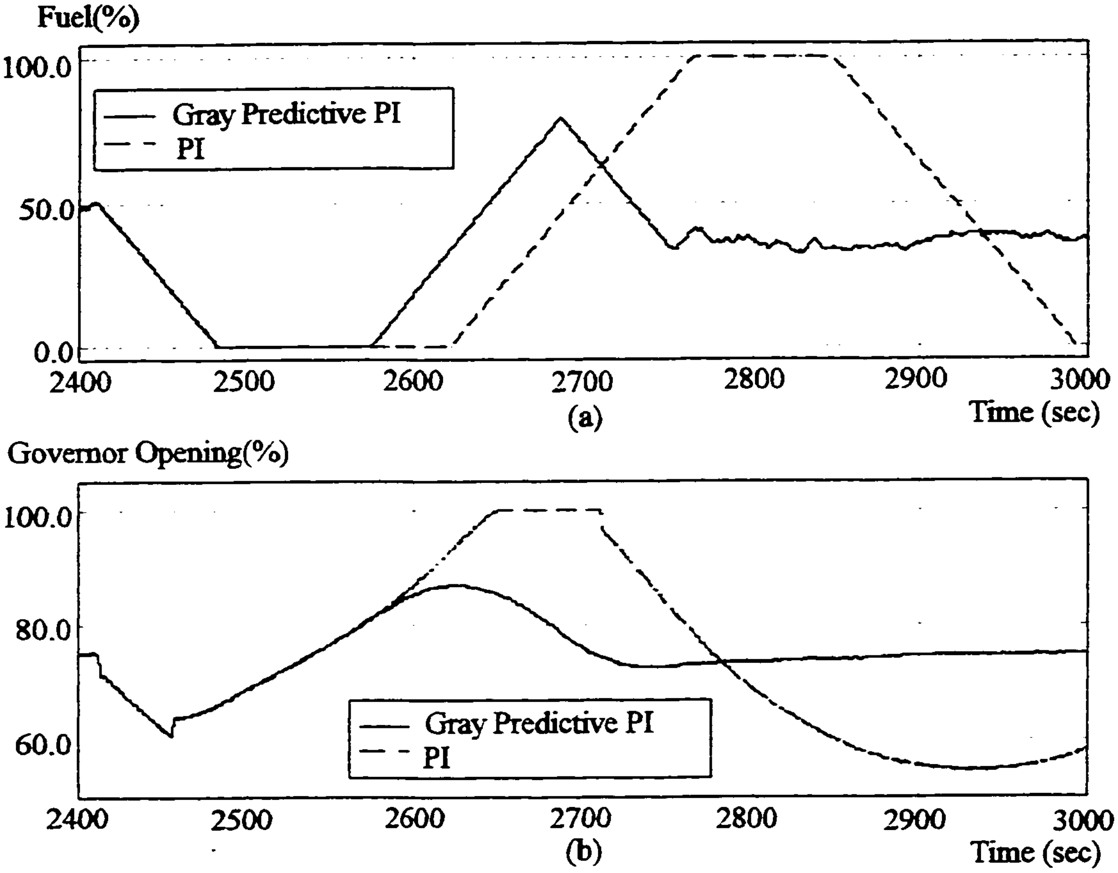


Figure 4.25 Control Outputs of Fuel Controllers and Turbine Controllers

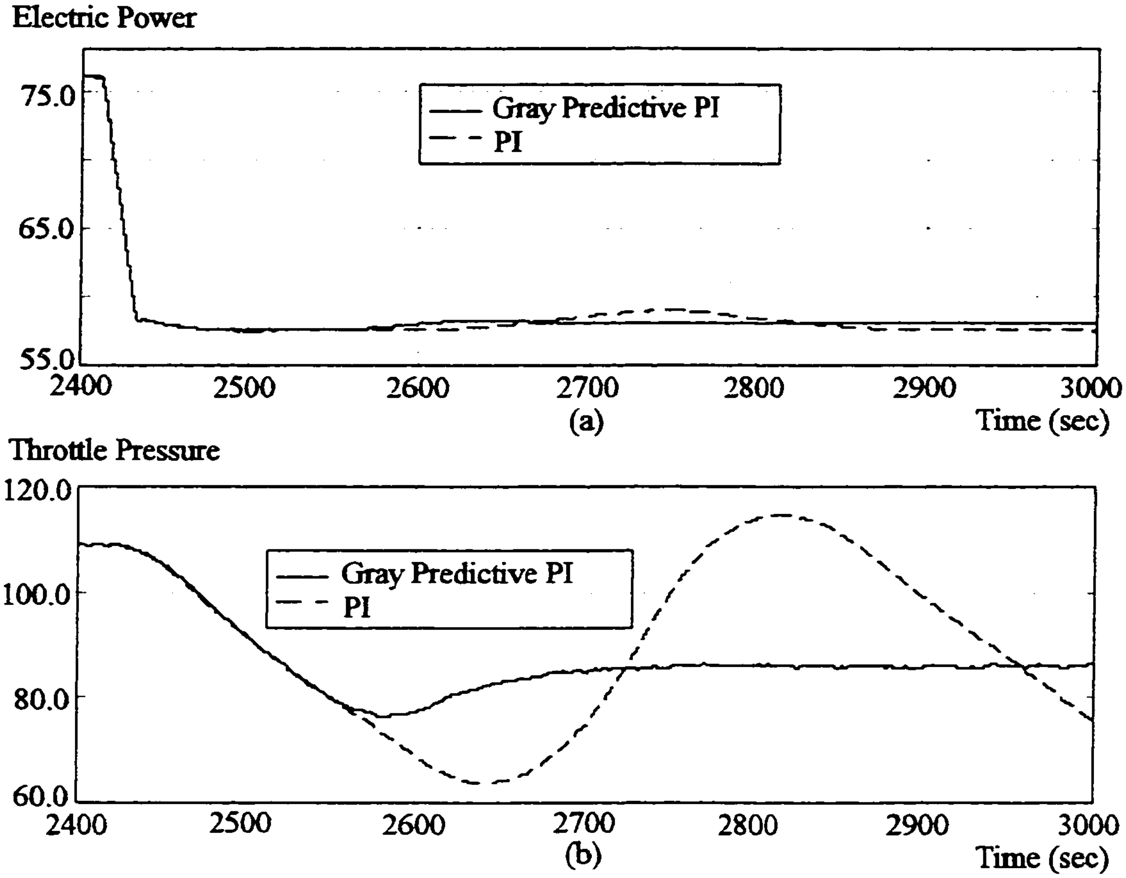


Figure 4.26 Load Demand Decreases from 75 MW
to 56 MW at the rate of 50 MW/minute

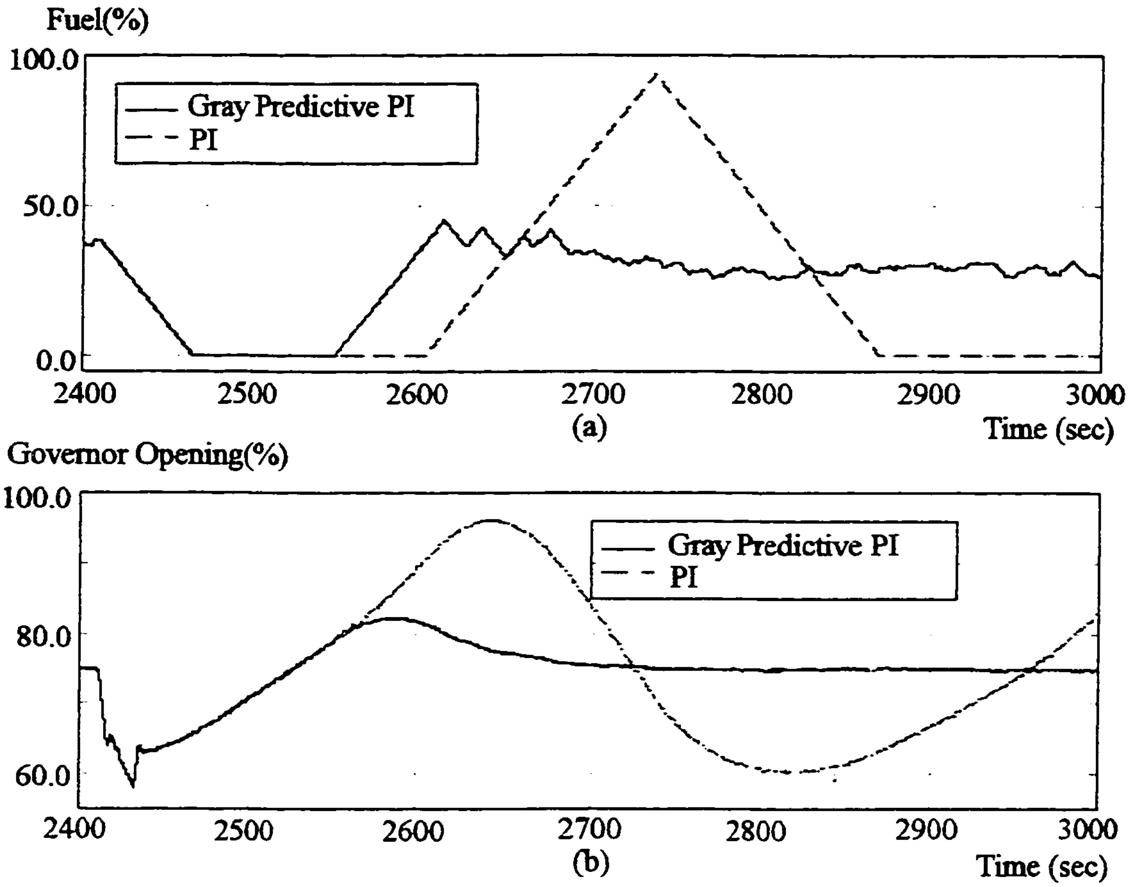


Figure 4.27 Control Outputs of Fuel Controllers and Turbine Controllers

Simulation 4.3: Disturbance Rejection

A pulse disturbance, generated by "Pulse Generator" in Simulink 4.2c, is added to the output of fuel controller, and a colored noise, generated by "Band Limited White Noise" in Simulink 4.2c through a filter, is added to the throttle pressure.

White Noise:
Power = 0.2, Sample Interval = 0.1 sec, and Seed = 23341,
Filter: 1/(5s+1).

CASE 1: Pulse period: 600 sec; pulse width: 300 sec, and pulse height: 26% of maximum fuel flow

Figure 4.28 shows the responses of BTG system. Errors for both throttle pressure and electric power under the control of the gray predictive PI are about 1/3 less than under the control of PI. Figure 4.29 shows the control outputs of fuel controllers. The gray predictive PI needs much less output than PI. It means that the gray predictive PI may still keep good disturbance rejection at high load or low load but PI may not because of reset windup of actuators.

CASE 2: Decrease the period and width of pulse disturbance in CASE 1 to pulse period 400 sec, and pulse width 200 sec

Figure 4.30 shows the responses of BTG system, and Figure 4.31 shows the control outputs of fuel controllers. The disturbance rejection of gray predictive PI is still much better than that of PI.

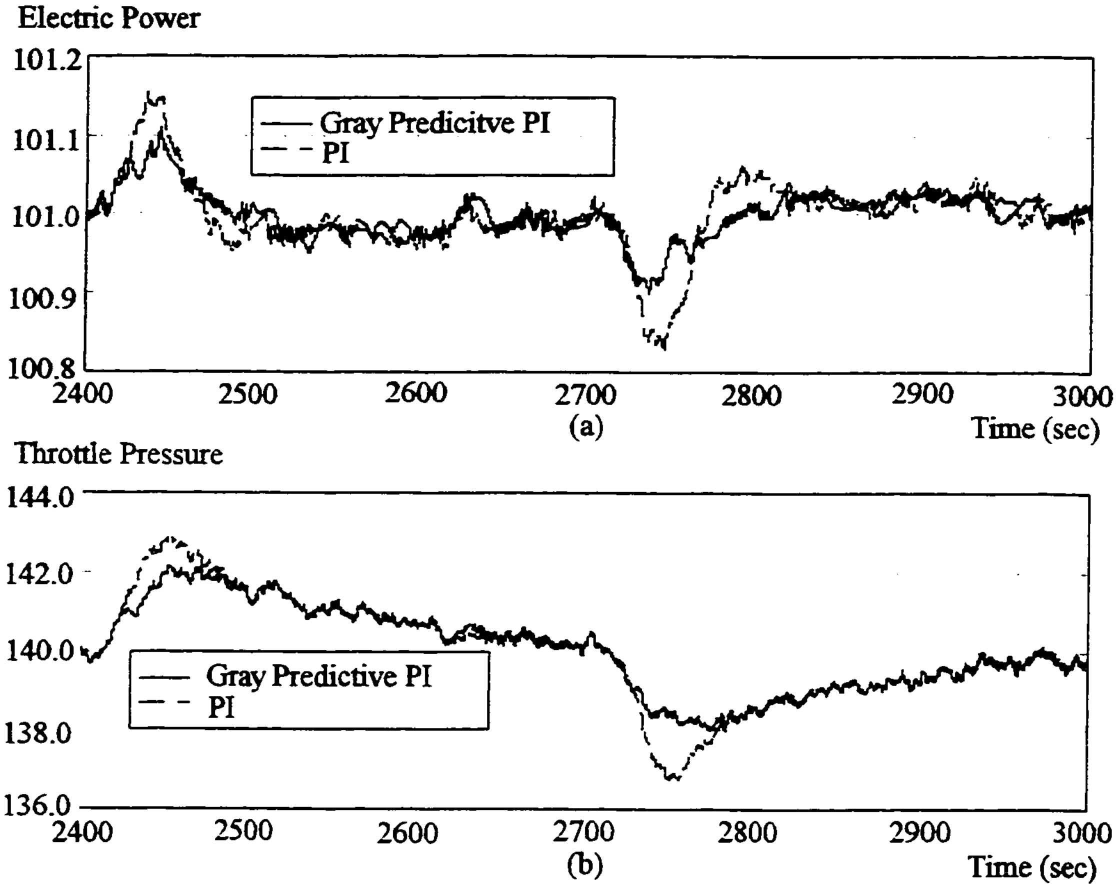


Figure 4.28 Disturbance Rejection

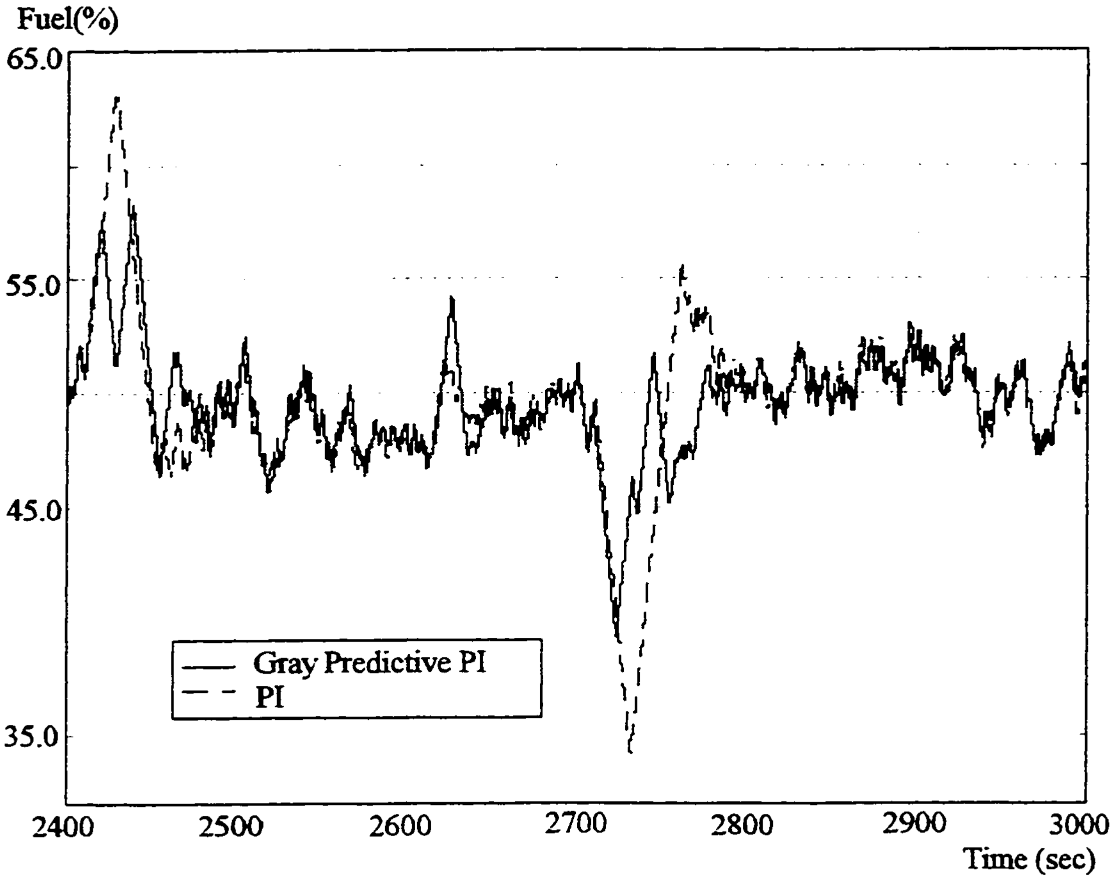


Figure 4.29 Control Outputs to Fuel Controllers

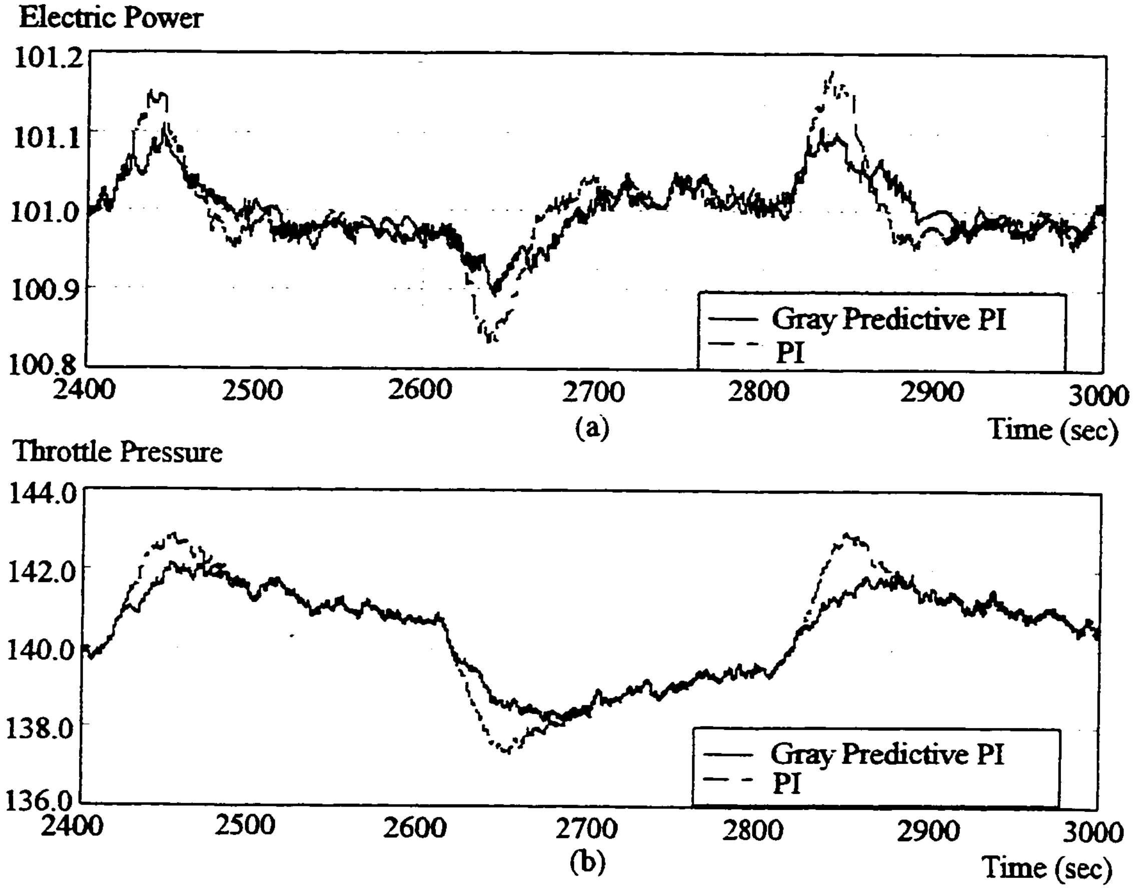


Figure 4.30 Disturbance Rejection

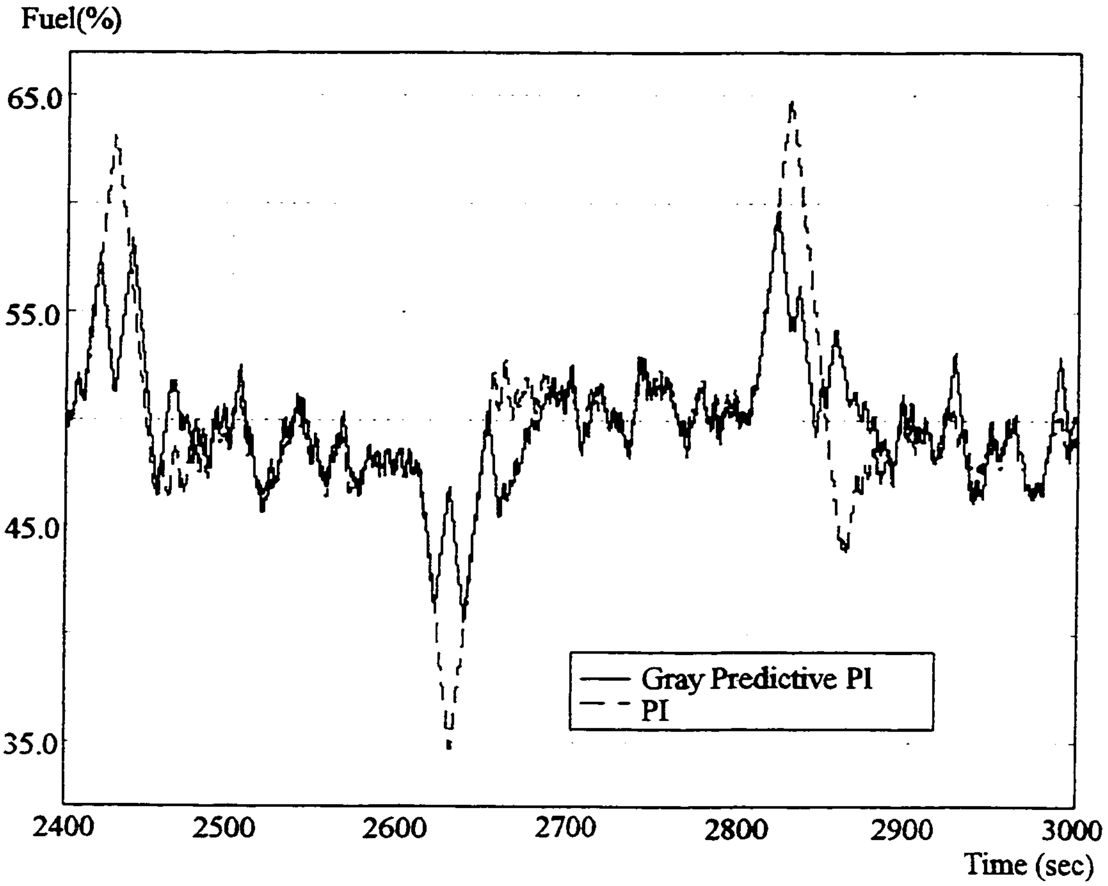


Figure 4.31 Control Outputs of Fuel Controllers

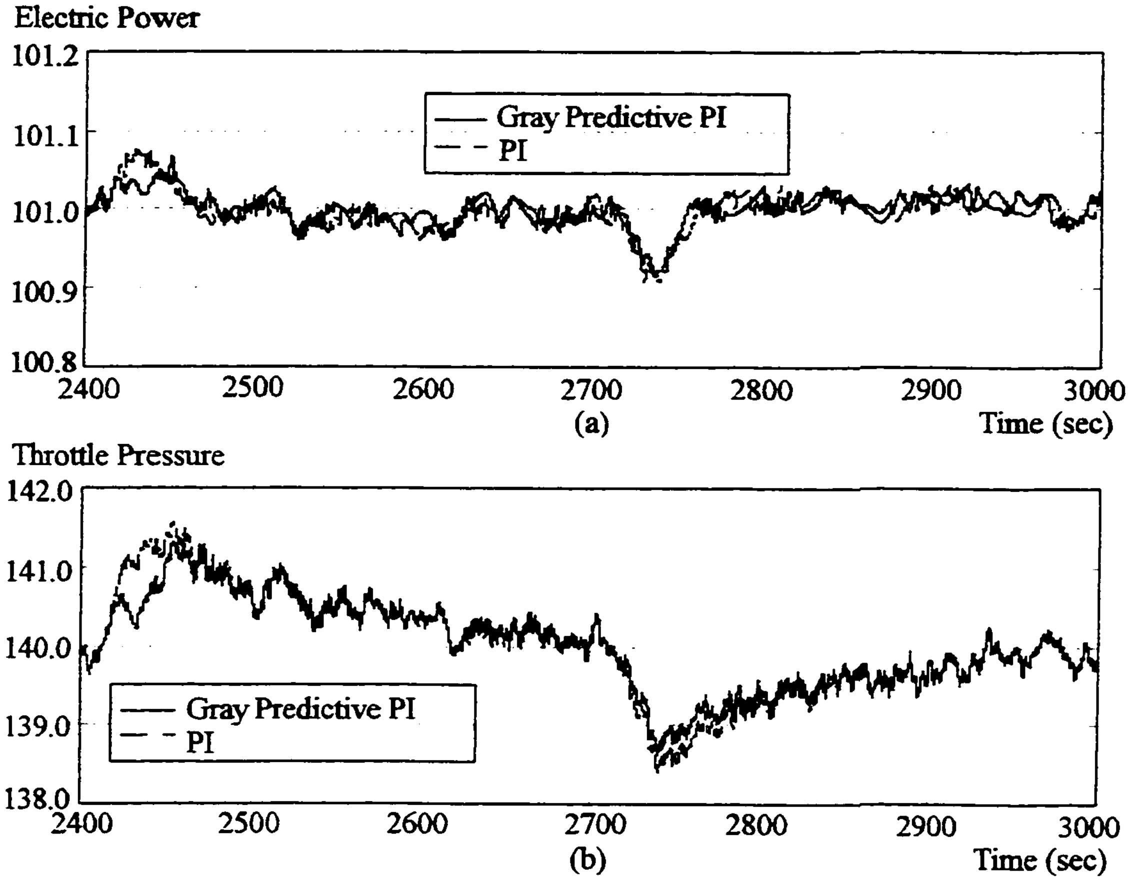


Figure 4.32 Disturbance Rejection

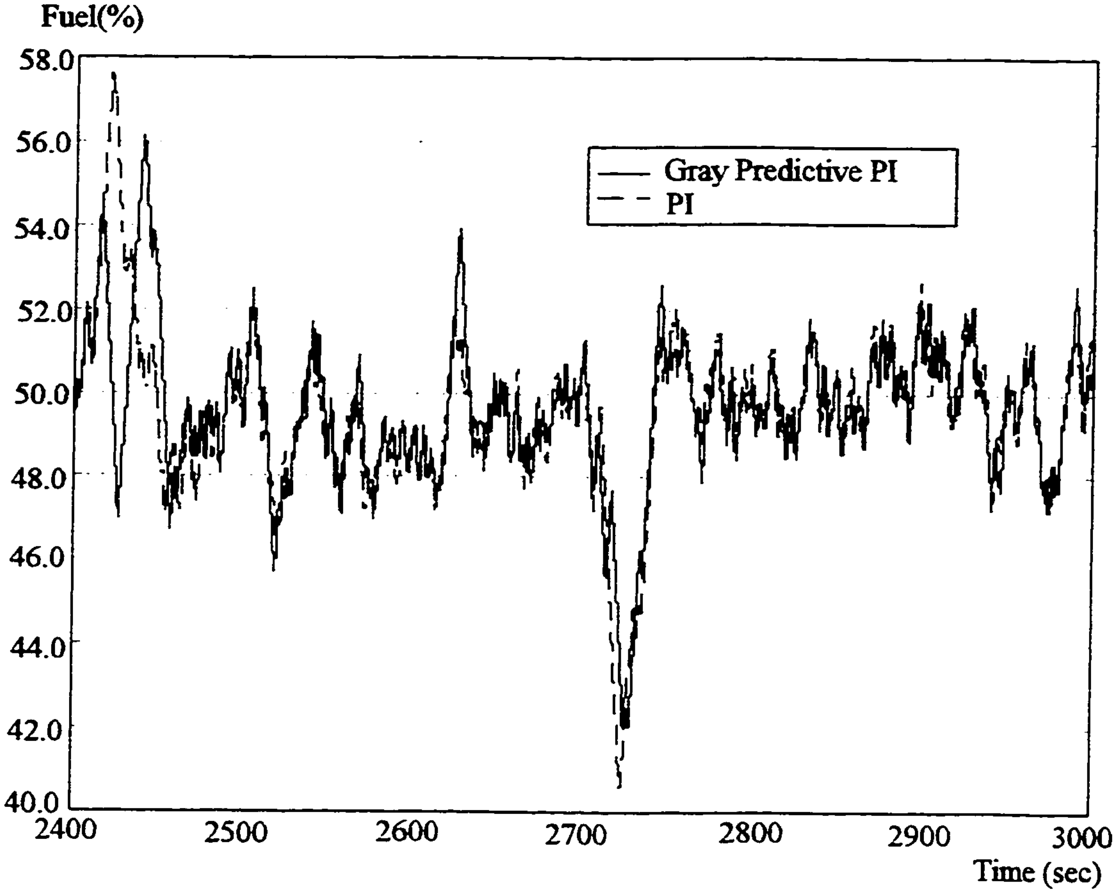


Figure 4.33 Control Outputs of Fuel Controllers

CASE 3: Decrease the height of pulse disturbance in CASE 1 to 13% of maximum fuel flow

Figure 4.32 shows the responses of BTG system, and Figure 4.33 shows the control outputs of fuel controllers. The responses of BTG systems and the control outputs tend to be almost the same. This is an expected result, since we do not desire to improve the disturbance rejection for a small disturbance at the cost of quickly worn out actuator.

Simulation 4.4: Sensitivity to Noise

In Simulation 4.3, we have seen that a properly designed gray predictive PI has almost the same sensitivity to noise as PI. Here, more simulations will be taken to further support this conclusion.

A pulse disturbance is added to the output of fuel controller with Pulse Period 600 sec, Pulse Width 300 sec, and Pulse Height: 26% of maximum fuel flow.

CASE 1: White noise with intermediate frequency

A white noise signal with

Power = 0.2,
Sample Time = 2 sec, and Seed = 23341,

is added to the pressure signal.

Figure 4.34 shows the responses of BTG system, and Figure 4.35 shows the control outputs of fuel controllers.

CASE 2: White noise with low frequency

A white noise signal with

Power = 0.2,
Sample Time = 6 sec, and Seed = 23341,

is added to the pressure signal.

Figure 4.36 shows the responses of BTG system, and Figure 4.37 shows the control outputs of fuel controllers.

CASE 3: Colored noise with low frequency

A colored noise signal

White Noise: Power = 0.2,
Sample Time = 0.1 sec.
Filter = 1/(1+5s).

is added to the pressure signal.

Figure 4.38 shows the responses of BTG system, and Figure 4.39 shows the control outputs of fuel controllers.

CASE 4: Colored noise with low frequency

A colored noise signal

White Noise: Power = 0.4,
Sample Time = 2 sec.
Filter = 1/(1+5s).

is added to the pressure signal.

Figure 4.40 shows the responses of BTG system, and Figure 4.41 shows the control outputs of fuel controllers.

CASE 5: Colored noise with low frequency

A colored noise signal

White Noise: Power = 0.4,
Sample Time = 6 sec.
Filter = 1/(1+5s).

is added to the pressure signal.

Figure 4.42 shows the responses of BTG system, and Figure 4.43 shows the control outputs of fuel controllers.

From CASE 1 to CASE 6, under the influences of different noises, white or colored, high frequency to low frequency, the gray predictive PI not only demonstrates almost the same sensitivity to noises as PI, but also has much better disturbance rejection and more reasonable control outputs than PI.

We can therefore conclude that gray predictive PI can significantly improve the control quality (load following and disturbance rejection) and stability of BTG system. A properly designed gray predictive PI not only can achieve much better control quality than PI, but also has almost the same sensitivity to noise as PI.

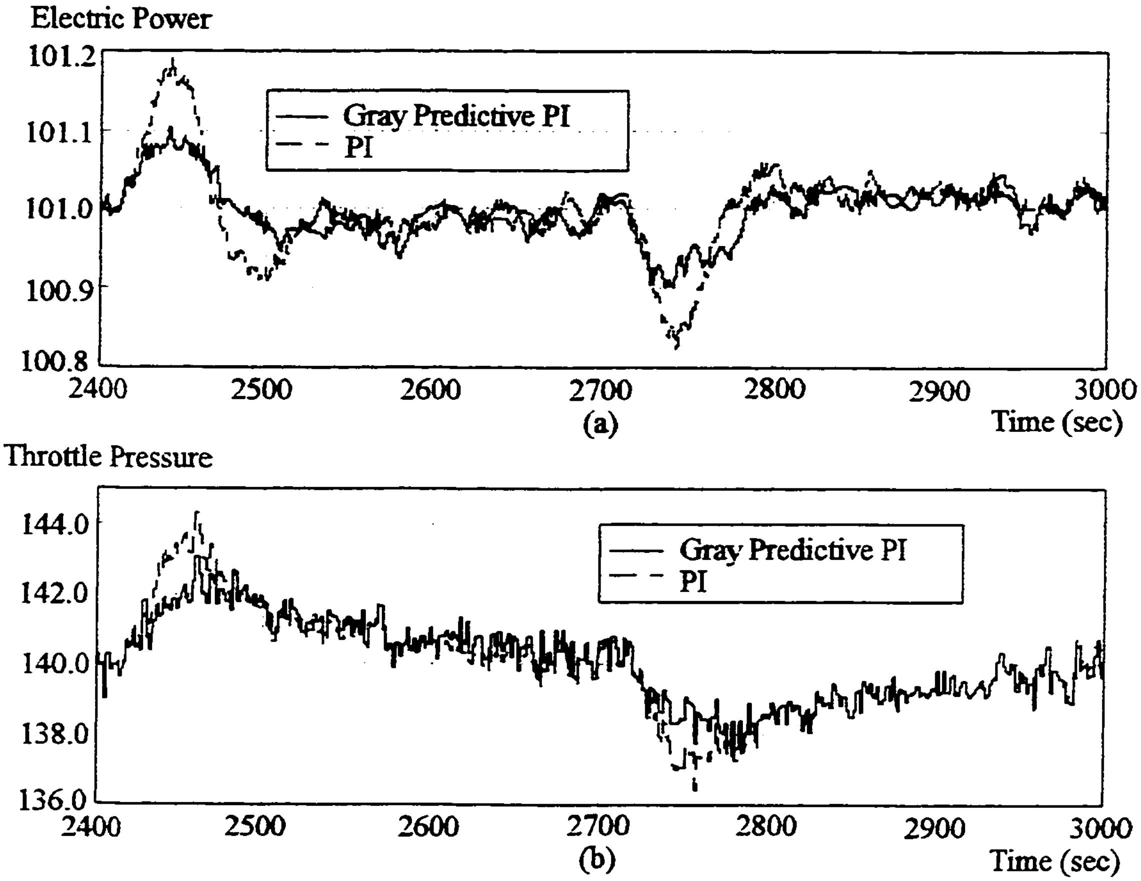


Figure 4.34 Sensitivity to White Noises — Responses to Disturbance

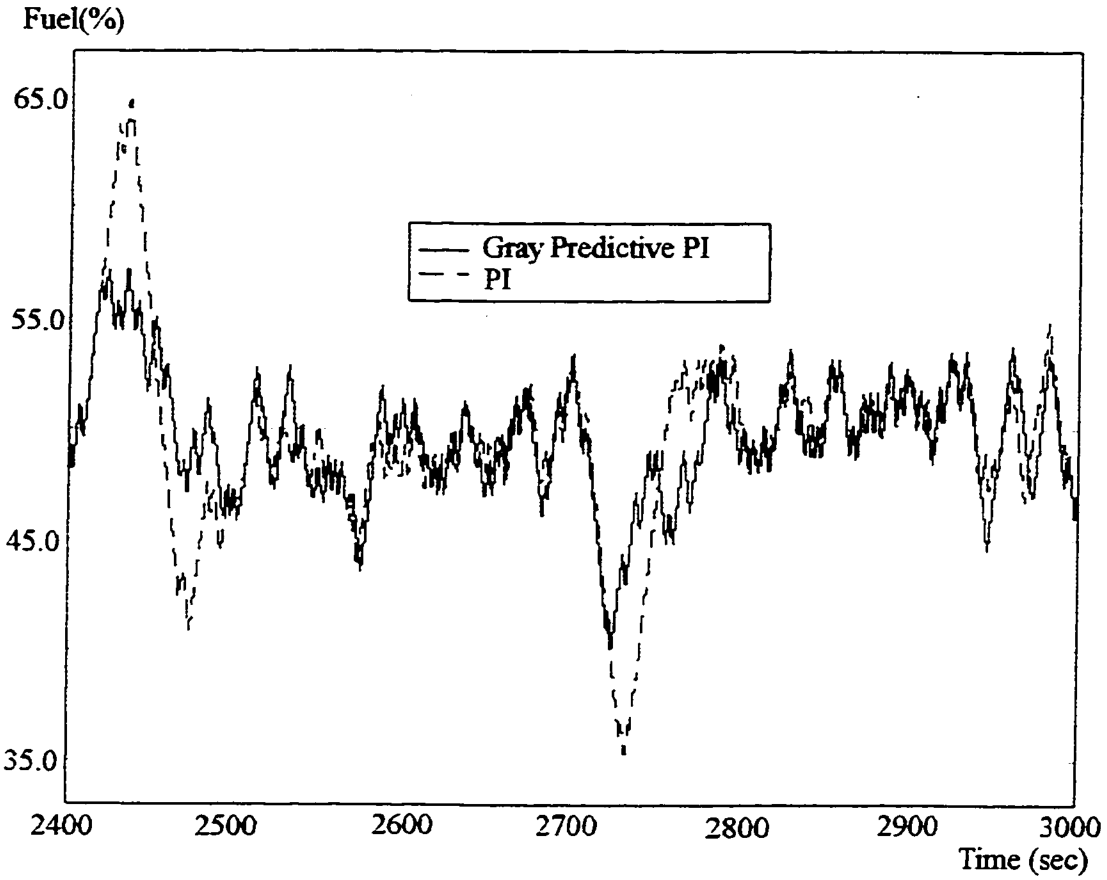


Figure 4.35 Sensitivity to White Noises --- Control Outputs of Fuel Controllers

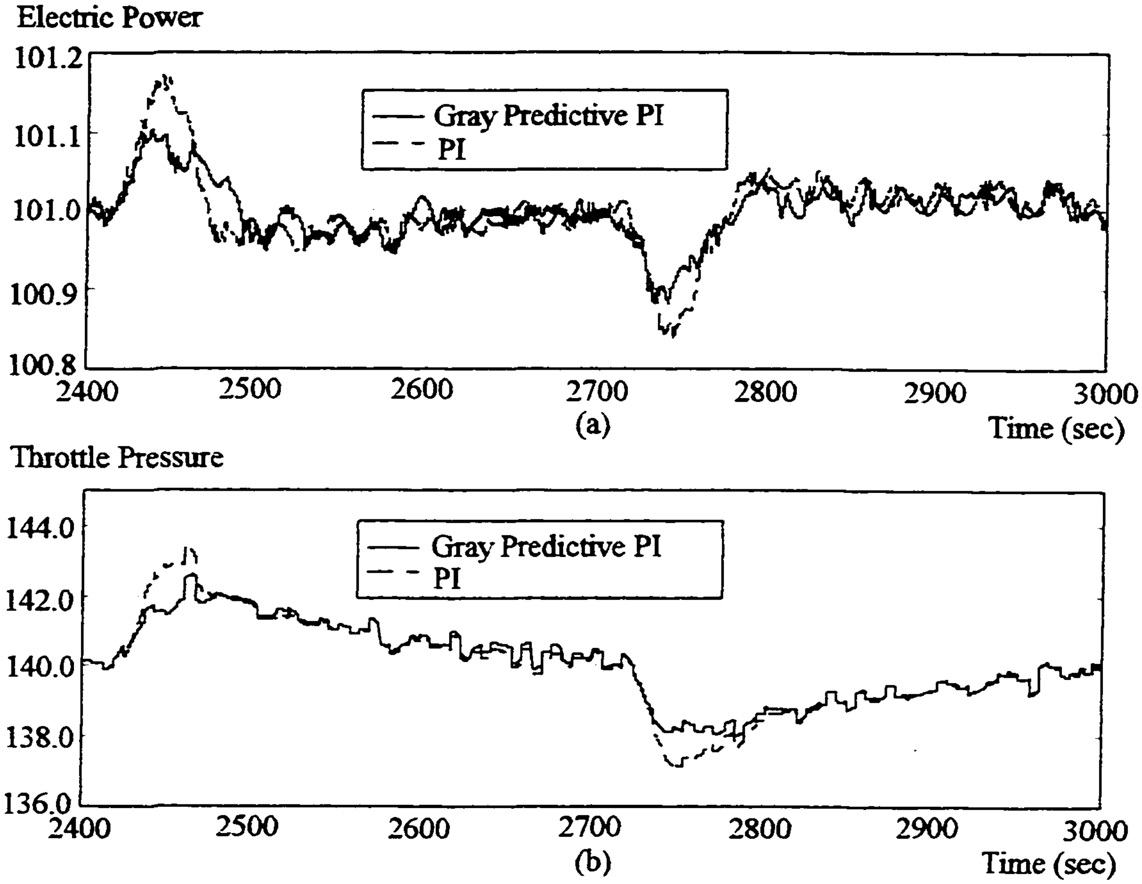


Figure 4.36 Sensitivity to White Noises --- Responses to Disturbance

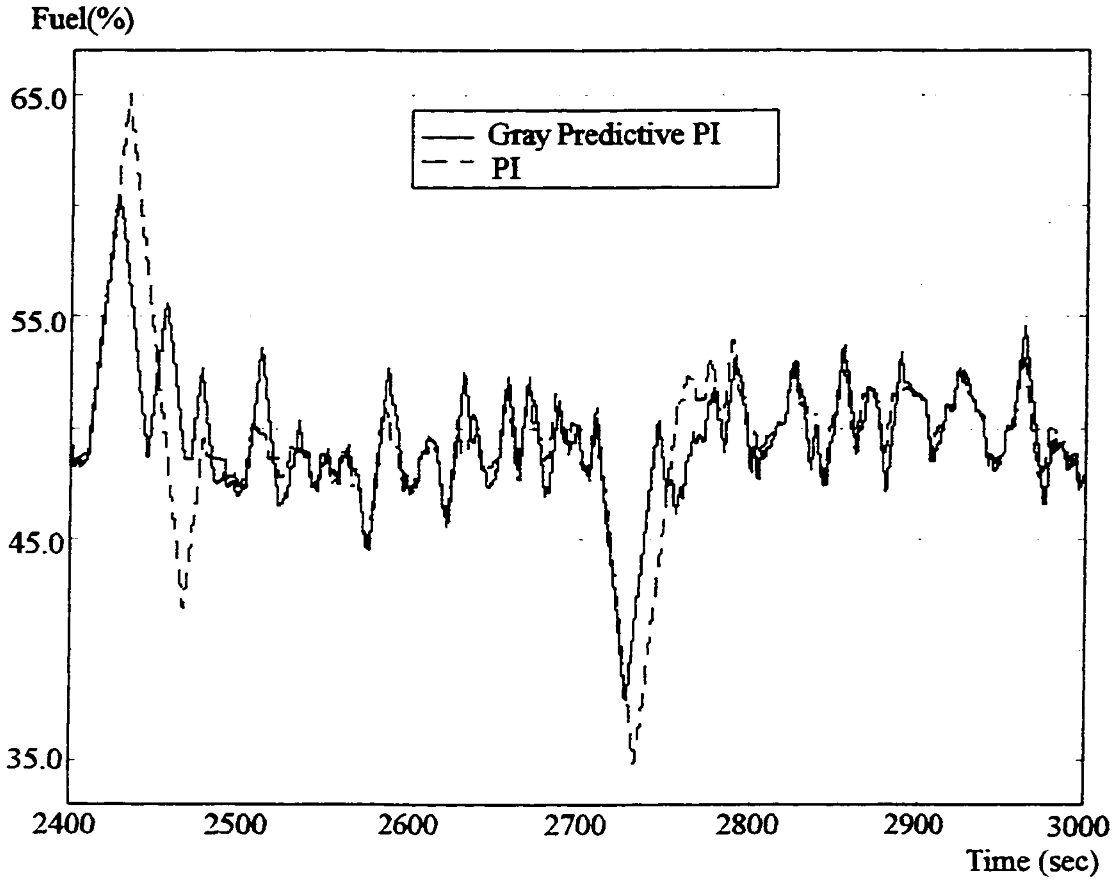


Figure 4.37 Sensitivity to White Noises --- Control Outputs of Fuel Controllers

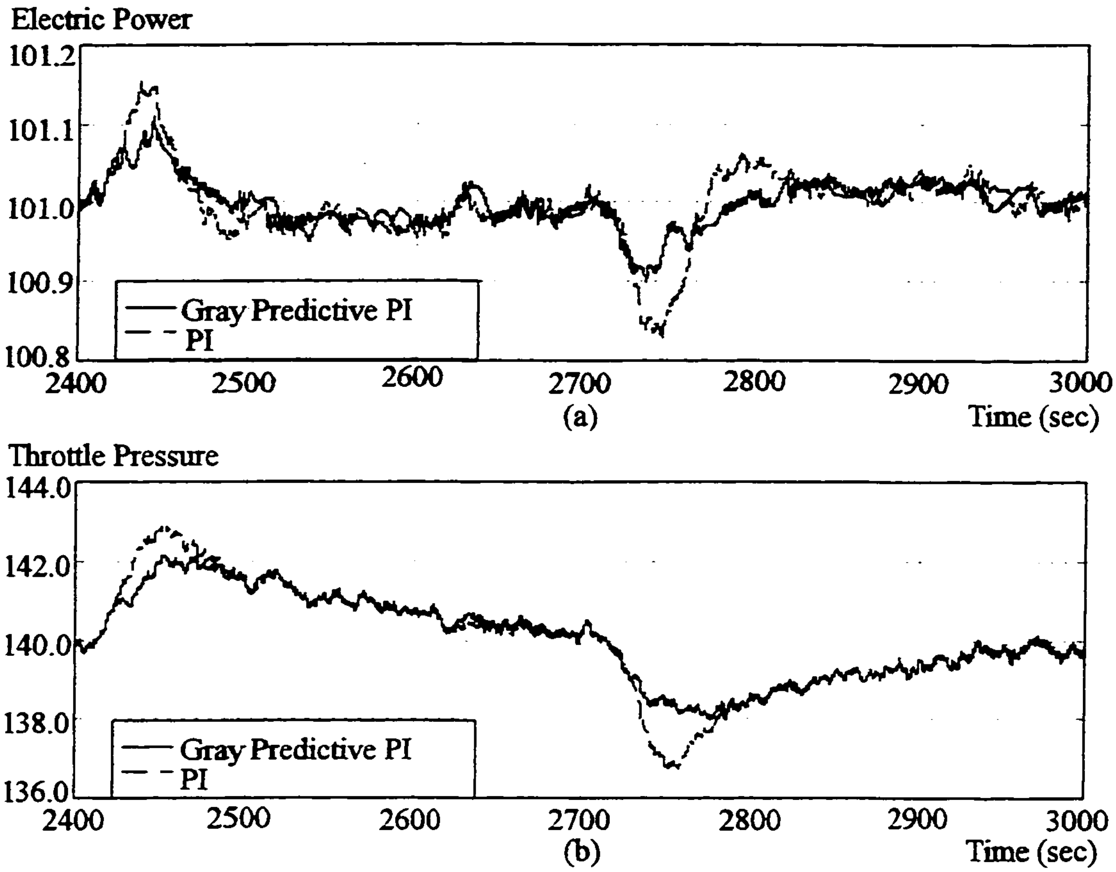


Figure 4.38 Sensitivity to Colored Noises --- Responses to Disturbance

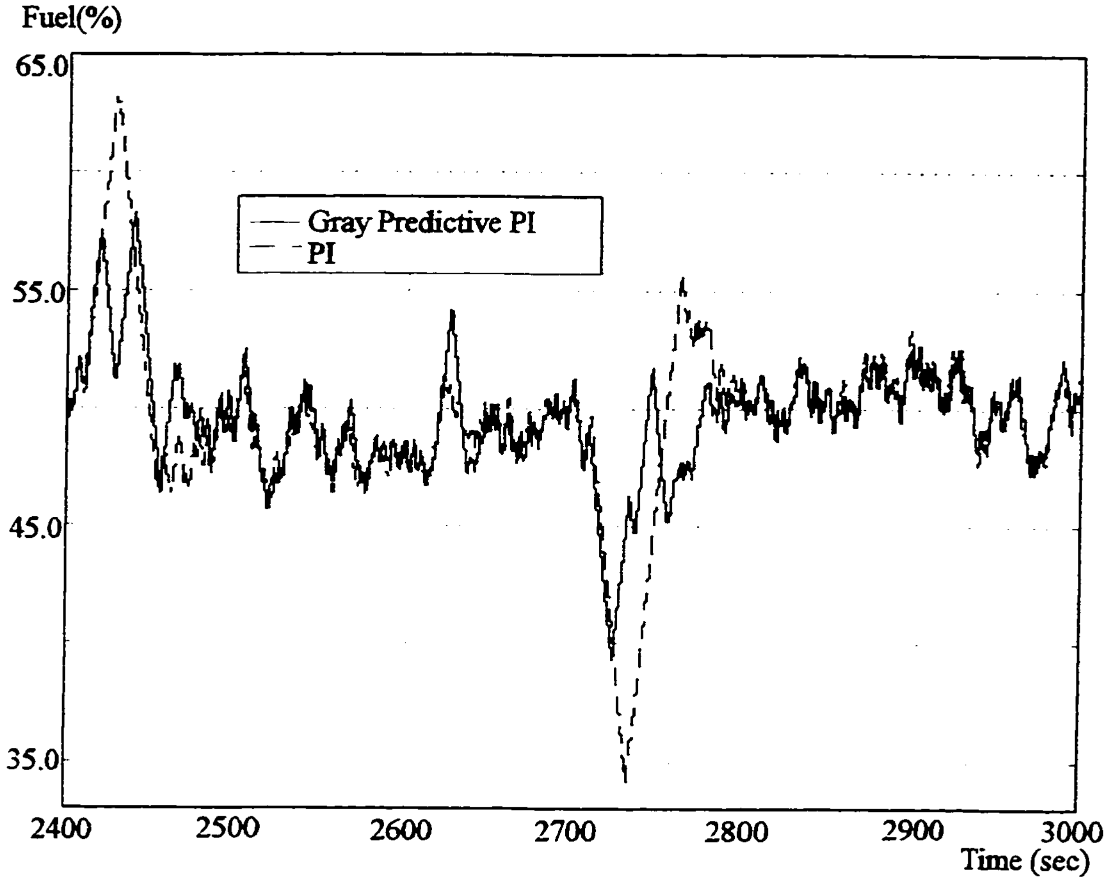


Figure 4.39 Sensitivity to Colored Noises --- Control Output of Fuel Controller

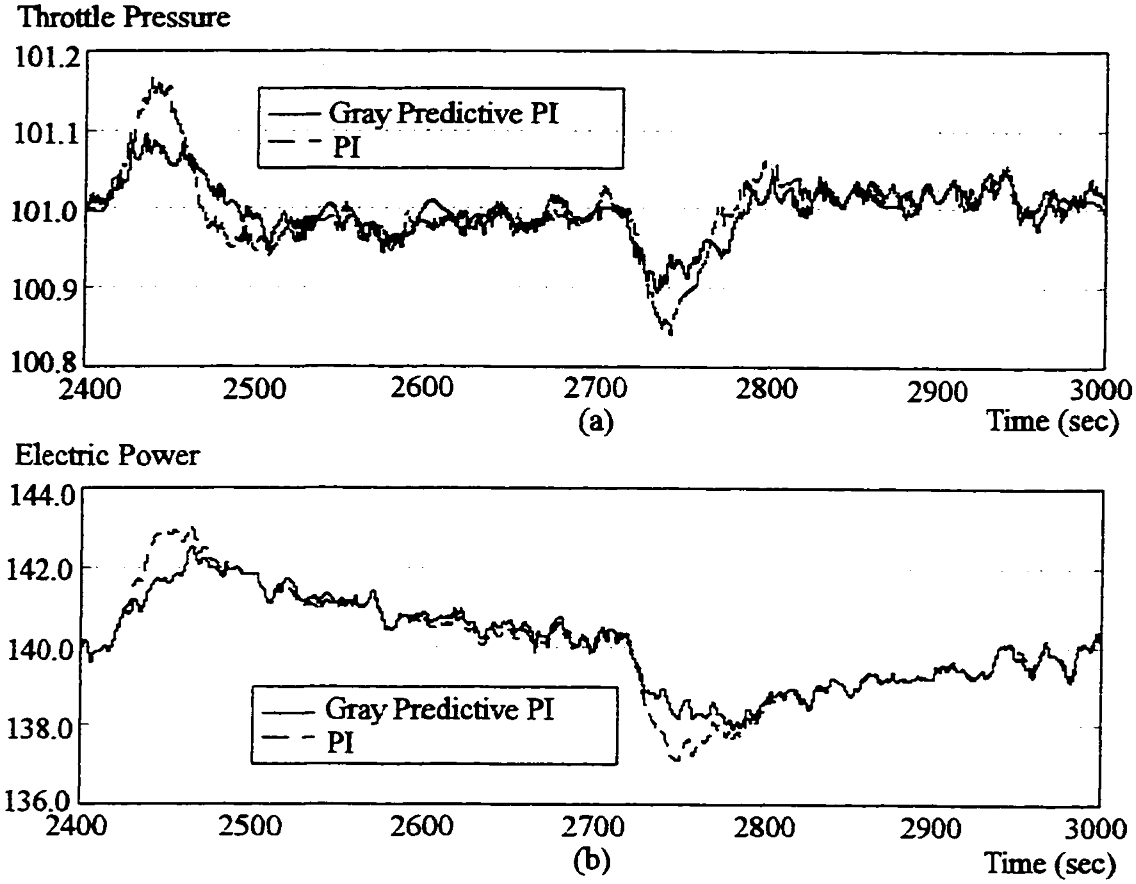


Figure 4.40 Sensitivity to Colored Noises --- Responses to Disturbance

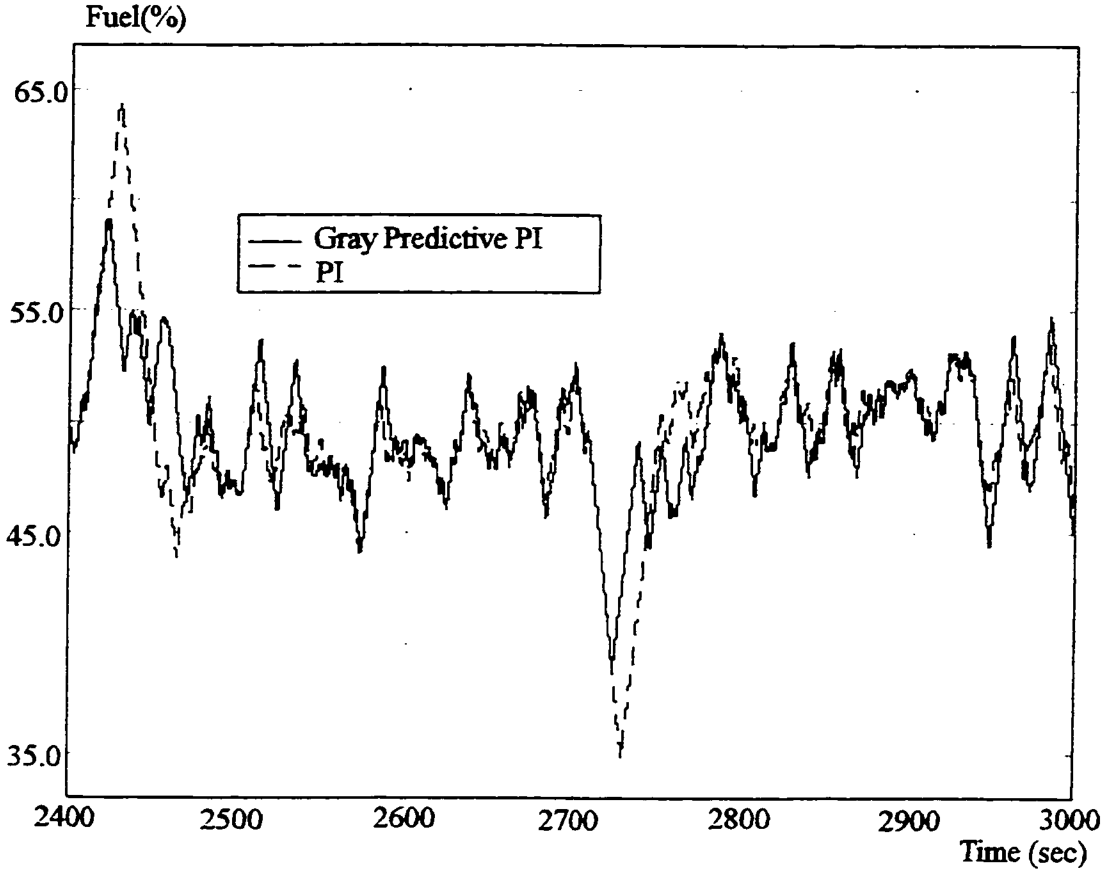


Figure 4.41 Sensitivity to Colored Noises --- Control Outputs of Fuel Controllers

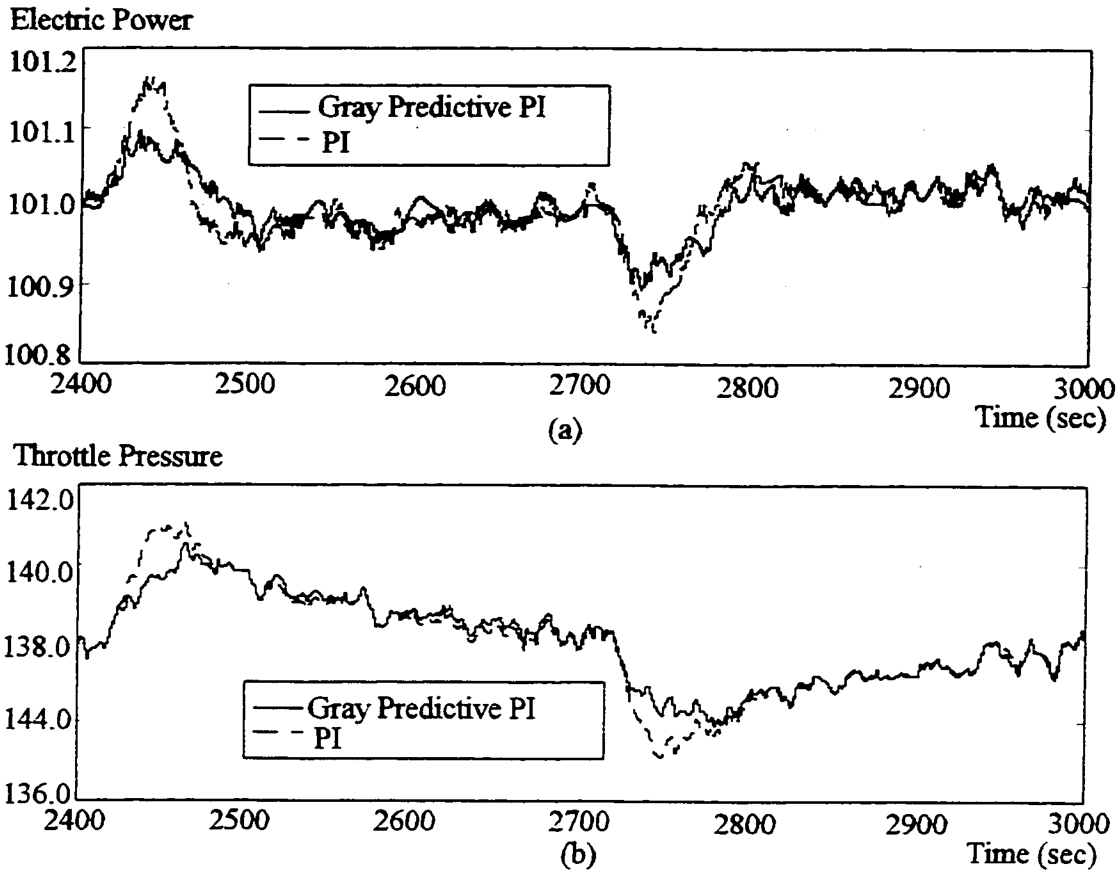


Figure 4.42 Sensitivity to Colored Noises --- Responses to Disturbance

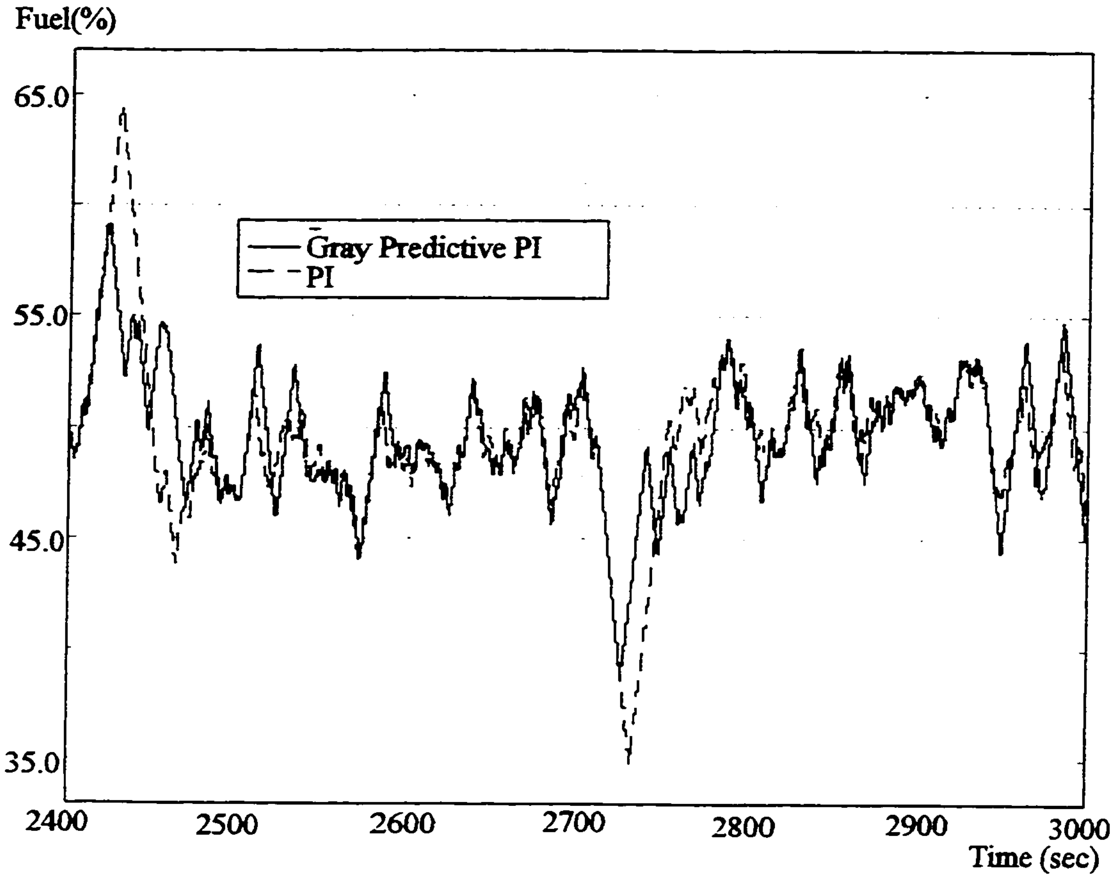


Figure 4.43 Sensitivity to Colored Noises --- Control Outputs of Fuel Controllers

Chapter 5

Conclusion

Through the above various simulations and theoretical analysis on continuous-linearly-interpolated gain scheduled PID, gray predictive PI and their application in a BTG control system of power generation, we have the following conclusions:

1. The design of the integral mode of gain scheduled PID can significantly influence the performance of gain scheduled PID. If the scheduling variables are setpoints and may have step change, it is reasonable to use the integral mode (2.10), i.e.

Method 1: $I(k) = I(k-1) + [K_c(k)/T_i(k)]e(k)$, or

$I(t) = \int [K_c(t)/T_i(t)]e(t)dt.$

If there exist large disturbances in the process, the control system designed with this method is more likely to become unstable than PID with the integral mode (2.11) or (2.12), since it reduces to PID with fixed controller parameters for disturbances.

If the scheduling variables are the states or outputs of a process, for the variation of process gain K_p, it is better to use the integral mode (2.11) or (2.12) instead of (2.10). That is, we should use

Method 2: $I_1(k) = I_1(k-1) + e(k)$, $I(k) = [K_c(k)/T_i(k)]I_1(k)$, or

$I(t) = [K_c(t)/T_i(t)] \int e(t)dt$; or

Method 3: $I_1(k) = I_1(k-1) + e(k)/T_i(k)$, $I(k) = K_c(k)I_1(k)$, or

$I(t) = K_c(t) \int [e(t)/T_i(t)]dt.$

However, for the variation of process time constant T and time delay τ, the integral mode (2.10) is the best selection.

Usually a practical process involves combined variation of K_p, T and τ. Which integral mode is the best can be determined by trials, but the integral mode (2.12) is a reasonable trade-off for general purposes.

2. We should choose gain scheduling approaches based on the relationship between gain scheduling variables and process parameters. For the single-input single-output processs whose process parameters, gain, time constant and time delay, are linear or non-linear functions of the scheduling variable W, and can be better approximated by the first order

polynomial of the scheduling variable W than W^{-1}, we have the conclusions below.

For the variation of process gain K_p, MFGS PID with the integral mode (2.12) is the definite selection. FGS1 and FGS2 are much worse, and they are more likely to cause the closed-loop system to be unstable or much more damped than MFGS when the process is operated at the intermediate values of the scheduling variables. To achieve the desired control quality, more operating points are needed for FGS1 and FGS2.

For the variation of process time constant T, FGS1 and FGS2 PID with the integral mode (2.10) can achieve better control qualities for setpoint tracking, and slightly better disturbance rejection than MFGS, since $K_c(y)$ in MFGS is too small. MFGS has more damped set-point step responses than FGS1 and FGS2 when the process is running at intermediate values of the scheduling variable. Certainly, the variation of process time constants usually does influence control quality much less than the variation of process gains and time delay.

For the variation of process time delay τ, MFGS PID with the integral mode (2.10) is the best, FGS2 is worse, and FGS1 is the worst -- too large over shoot for step responses and disturbance rejection when the process is running at intermediate values of the scheduling variables.

A practical process, even if it can be approximated by the first order process, usually is complicated, i.e. it may involves combined variation of K_p, T and τ. To choose a good gain scheduled PID depends on not only how much we know about gain scheduled PID but also how much we know about the process.

3. From Simulation 4.3 and 4.4, we see that error-prediction based gray predictive PI suggested in Figure 3.14, combined with LDISPI method of noise suppression presented by the thesis in Section 3.1.6, not only can achieve much better control quality and stability than

PI, but also has almost the same noise sensitivity as PI. It uses quite less control output than PI to achieve much better disturbance rejection. This means that if the speeds of actuators are low, or if the actuators are near saturation limits, gray predictive PI can still achieve good control quality, while PI may not.

The theory of gray system and gray prediction originated from the prediction of social systems. Gray predictive controllers can be used in any process with gray exponential form. Certainly for a process with complex eigenvalues, the effect of prediction may not be good. A promising improvement may be found from the literature [29]. In the literature [18], the author points out that gray predictive control can be used in multivariable systems, but gave no further information. Some interesting information may be found in the literature [30]. The gray predictive controllers based on the scheme shown in Figure 3.12 have been successfully used in the control of liquid level, temperature, DC electric machine, and refinery tower etc [18]. Also the gray phase-plane controller has been used in hydraulic servo system [18]. Unfortunately, the author of the literature [18] had not given enough references for these applications.

Finally I would like to point out that gray prediction should be used to compensate the process time lag instead of pure time-delay, thus the prediction horizon shall be determined by the compensation of time lag. As to LDISPI method of noise suppression presented in Section 3.1.6, the sample interval and predictive interval should be determined not only by statistic features of noises but also by the requirement for control quality, actuator feature, and CPU load. Another issue deserving of mention is that, for the error-prediction-based gray predictive PI shown in Figure 3.14, the bias B should be large enough to reduce the different predictive effects for the process output increase and decrease unless we intend to use this unsymmetry.

In conclusion, gain scheduled controllers have been successfully used in industrial control. Gray prediction is becoming more and more attractive to control engineers. In power generation, the further study in multivariable gray predictive BTG control system promises to be interesting and valuable.

Bibliography

[1] J-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, Englewood Cliffs, NJ, 1991.

[2] H. Nijmeijer and A. J. Vander Schaft, *Nonlinear Dynamical Control Systems.* Springer-Verlag New York Inc., 1990.

[3] P. Dorats, L. Fortuna, and G. Muslato, *Robust Control For Unstructured P Perturbations -- An Introduction.* Springer-Verlag New York Inc., 1992.

[4] K. J. Astrom and B. Wittenmark, *Adaptive Control.* Addsion-Wesley Publishing Company.

[5] G. Stein, "Adaptive flight control -- a pragmatic view", *Applications Of Adaptive Control.* Academic Press New York, 1980.

[6] G. Meyer, R. Su, and L. R. Hunt, "Application of nonlinear transformations to automatic flight control", *Automatica*, vol. 20, 1984.

[7] W. J. Rugh, "Analytical framework for gain scheduling", *IEEE Trans. On Control Systems*, No. 1, pp.79-84, 1991.

[8] L. Cheng and T. F. Edgar, "A new fuzzy gain scheduling algorithm for process control", *ACC*, Vol. 3, pp.2284-2290, 1992.

[9] O. J. H. Smith, "A controller to overcome dead time", *ISA Journal*, No. 2, 1958.

[10] C. R. Culter and B. L. Ramaker, "Dynamic matrix control -- a computer control algorithm", *JACC*, 1980.

[11] D .W. Clark and C. Mohtadi, "Generalized predictive control", *Automatica*, Vol. 23, No. 2, pp.137-160, 1989.

[12] Y. H. Jing and C. Z. Huang, *Process Control*. Qinghua University Publishing Company, Beijing, 1993.

[13] J. S. Shamma and M. Athans, "Guaranteed properties for nonlinear gain scheduled control system". *27th IEEE Conference On Decision & Control*, Austin, TX, Vol.3, 1988.

[14] J .S. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants", *IEEE Trans. On Automatic Control*, Vol.35, No.8, pp.898-907,1990.

[15] J. S. Shamma and M. Athans, "Gain scheduling: potential hazards and possible remedies". *IEEE Trans. On Control Systems*, No.6, pp.101-107, 1992.

[16] D. E. Seborg, T. F. Edgar and D. A. Mellichamp, *Process Dynamics and Control*. Wiley, New York, 1989.

[17] J. R. Deng, *A Course On Gray System Theory*. Publishing House Of Huazhong Polytechnology University, P. R. China, 1990.

[18] J. R. Deng, *Gray Control System*. Publishing House Of Huazhong Polytechnology University, P. R. China, 1993.

[19] W. B. Zhang, *Sequence Control And Thermal System Protection*. Publishing House Of Electric Power Ministry, P. R. China, 1991.

[20] Y. Z. Zhang and M. J. Wang, *Thermal Control Systems*, Publishing House Of Electric Power Industry, Beijing, P. R. China, 1993.

[21] C. H. Liu, *Decoupling Theory For Multivariable Process Control Systems*, Publishing House Of Electric Power Industry, Beijing, P. R. China, 1984.

[22] K. J. Astrom and R. D. Bell, "A low-order dynamic model for drum boiler-turbine-alternator units", Report TFRT-7162, Department of Automation Control, Lund Institute of Technology, Sweden, 1987.

The following are some papers on gray control which may be interesting to the readers:

[23] C. S. Zhou and J. R. Deng, "Stability analysis of gray discrete-time systems", *IEEE Trans. On Automatic Control*, No.2, pp.173-174, 1989.

[24] C. S. Zhou and J. R. Deng, "The stability of gray linear systems", *International Journal Of Control*, No.1, pp.313-320, 1986.

[25] J. R. Deng and C. S. Zhou, "Sufficient conditions for the stability of a class of inter-connected dynamic systems", System & Control Letters, No.2, pp.105-108, 1986.

[26] J. R. Deng, "Control problems of unknown systems", *Proceeding Of The Bilateral Meeting On Control Systems*, Shanghai, P.R.China, pp.156-171, 1981.

[27]J. R. Deng, "Essential models for gray forecasting control", *The Journal Of Gray System*, No.1, pp.1-10, 1990.

[28] J. R. Deng, "Models for gray series", *The Journal Of Gray System*, No.3, pp.217-232, 1990.

[29] J. R. Deng, "Method to forecast the turning point of metabolized GM(1,1) model", *The Journal Of Gray System*, No.4, pp.219-225, 1992.

[30] J. R. Deng, "The properties of multivariable gray model GM(1,N)", *The Journal Of Gray System*, No.1, pp.25-42, 1989.