Ang, Kiam Heong (2005) *Evolutionary learning and global search for multi-optimal PID tuning rules.* PhD thesis.

http://theses.gla.ac.uk/2093/

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Glasgow Theses Service
http://theses.gla.ac.uk/
theses@gla.ac.uk

EVOLUTIONARY LEARNING AND GLOBAL SEARCH FOR MULTI-OPTIMAL PID TUNING RULES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING
OF UNIVERSITY OF GLASGOW
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Kiam Heong Ang
January 2005

© Copyright 2005 by Kiam Heong Ang
All Rights Reserved

Abstract

With the advances in microprocessor technology, control systems are widely seen not only in industry but now also in household appliances and consumer electronics. Among all control schemes developed so far, Proportional plus Integral plus Derivative (PID) control is the most widely adopted in practice. Today, more than 90% of industrial controllers have a built-in PID function.

The simple three-term functionalities of PID control offer the most direct and efficient solution to many real-world control problems. Their wide applications have stimulated and sustained the research and development of PID tuning techniques, patents, software packages and hardware modules. Due to parameter interaction and format variation, tuning a PID controller is not as straightforward as one would have anticipated. Therefore, designing speedy tuning rules should greatly reduce the burden on new installation and 'time-to-market' and should also enhance the competitive advantages of the PID system under offer.

In order to achieve this objective, it is important that optimal and effective structures and tuning rules were globally search for under practical constraints. This is also a multi-criteria learning and design problem. Only by taking into account all necessary objectives for practical applications, will it eventually result in a tuning rule that can perform optimally across a wide application range and meet practical requirements. Conflicting objectives between tracking performance and load disturbance rejection are now perhaps the only major problem remaining in PID control, which have haunted the control community. Researchers in PID control, including Karl J. Åström, have thus resorted to modifying the PID structure beyond the traditional unity negative feedback control framework. For example, Åström and Hägglund (1995) proposed an alternative structure that places the derivative action on the plant output, instead of on the error signal, so as to cope with changes in tracking command. This has complicated the whole process and leads to an extra learning curve in tuning PID controllers.

Now with the advances in evolutionary computation, these problems can be addressed systematically and intelligently. In particular, a multi-objective evolutionary algorithm

(MOEA) would be an ideal candidate to conduct the learning and search for multi-objective PID tuning rules. A simple to implement MOEA, termed s-MOEA, is devised and compared with MOEAs developed elsewhere. Extensive study and analysis are performed on metrics for evaluating MOEA performance, so as to help with this comparison and development. As a result, a novel visualisation technique, termed "Distance and Distribution (DD)" chart, is developed to overcome some of the limitations of existing metrics and visualisation techniques. The DD chart allows a user to view the comparison of multiple sets of high order non-dominated solutions in a two-dimensional space. The capability of DD chart is shown in the comparison process and it shows to be a useful tool for gathering more in-depth information of an MOEA which is not possible in existing empirical studies.

Truly multi-objective global PID tuning rules are then evolved as a result of interfacing the s-MOEA with closed-loop simulations under practical constraints. It takes into account multiple, and often conflicting, objectives such as steady-state accuracy and transient responsiveness against stability and overshoots, as well as tracking performance against load disturbance rejection. These evolved rules are compared against other tuning rules both offline on a set of well-recognised PID benchmark test systems and online on three laboratory systems of different dynamics and transport delays. The results show that the rules significantly outperform all existing tuning rules, with multi-criterion optimality. This is made possible as the evolved rules can cover a delay to time constant ratio from zero to infinity based on first-order plus delay plant models. For second-order plus delay plant models, they can also cover all possible dynamics found in practice.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Yun Li, for his guidance, encouragement, inspiration and patience throughout the course of my Ph.D. research. In particular, his vision in, and commitments to, the research and his sharing of knowledge and time that has made the research a success.

Special acknowledgement and appreciation to Mr. Tom O'Hara and Ms. Vi Romanes for their enthusiastic assistance in laboratory and administrative work, respectively.

I would like to express my appreciation to my family for their support. Especially my wife, Fiona Leow, my parents, Ah Peng Ang and Ah Ying, my siblings, Kiam Chong, Pek Yong and Pek Boon and not forgetting my adorable little daughter, Eilidh.

I would also like to thank all my friends in Intelligent Systems Group for their friendliness, help and companionship. In particular, Gregory Chong, Dr. Wen Yuan Feng, Yin Jin, Dr. Lipton Chan and Dr. Cindy Goh. There are also friends from other group and organisations, in particular, Jian Zhang, Dr. Kay Chen Tan (National University of Singapore), Dr. Jin Sheng Gu (Yokogawa Engineering Asia Pte Ltd, Singapore), Terashima Nobuhiko (Yokogawa Electric Corporation, Japan), Kogami Masanobu (Yokogawa Electric Corporation, Japan), Philip Siow and Joanne Teal.

I wish to acknowledge Universities UK and University of Glasgow for the financial support to this research.

Writing a thesis would be impossible without the help of many people who have been generous with their time, energy, enthusiasm and specialist knowledge. Merely saying "thank you" seems an inadequate acknowledgement of their contributions; I can only hope that they understand how much I value them.

List of Publications

Journals

Ang, K.H., Chong, G., and Li, Y. (2004a). PID control system technology. *IEEE Transactions on Control Systems Technology*. (Accepted).

Li, Y., Ang, K.H., Chong, G., Feng, W., Tan, K.C., and Kashiwagi, H. (2004). CAutoCSD – evolutionary search and optimisation enabled computer automated control system design. *International Journal of Automation and Computing*, vol. 1, pp. 83-95.

Book Chapter

Ang, K.H., Chong, G., and Li, Y. (2004b). Visualization technique for analyzing non-dominant Pareto optimality. In: K.C. Tan *et al.* (Eds), *Recent Advances in Simulated Evolution and Learning*, World Scientific Series on Advanced in Natural Computation, Chapter 18.

Conference Proceedings

Ang, K.H. and Li, Y. (2001). Multi-objective benchmark studies for evolutionary computation. In *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, Workshop Program, San Francisco, U.S.A., pp. 393-396.

Ang, K.H., Li, Y., and Tan, K.C. (2001). Multi-objective benchmark functions and benchmark studies for evolutionary computation. In *Proceedings of the 2001 International Conference on Computational Intelligence for Modelling, Control and Automation*, Las Vegas, U.S.A., pp. 132-139.

Ang, K.H. and Li, Y. (2002a). An overview of benchmarking techniques for multi-objective evolutionary algorithms. In: R. Roy *et al.* (Eds), *Soft Computing and Industry: Recent Applications*, Springer, pp. 337-348.

Ang, K.H., Chong, G., and Li, Y. (2002a). Preliminary statement on the current progress of multi-objective evolutionary algorithm performance measurement. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Hawaii, U.S.A., pp.1139-1144.

Ang, K.H. and Li, Y. (2002b). Survey on PID control system design and tuning software tools. In *Proceedings of the 5th Asia-Pacific Conference on Control & Measurement*, Dali, P.R. China, pp. 12-17.

Ang, K.H. and Li, Y. (2002c). Survey on PID auto-tuning modules. In *Proceedings of the 5th Asia-Pacific Conference on Control & Measurement*, Dali, P.R. China, pp. 148-153.

Ang, K.H., Chong, G., and Li, Y. (2002b). Visualization technique for analyzing non-dominated set comparison. In *Proceedings of the 4th Asia-Pacific Conference on Simulation Evolution & Learning*, Singapore, pp. 36-40.

Chong, G., Li, Y., and Ang. K.H. (2002). The implementation of error-power (e^n) in feedback control and its design via evolutionary computation. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution & Learning*, Singapore, pp. 840-844.

Ang, K.H., Chong, G., and Li, Y. (2003). Multi-objective PID control and its automated tuning. In *Proceedings of the 16th International Conference on System Engineering*, Coventry, U.K., pp. 41-46.

Chong, G., Ang, K.H., and Li, Y. (2003). High-performance low-chattering weightless optimal control. In *Proceedings of the 16th International Conference on System Engineering*, Coventry, U.K., pp. 121-126.

Ang, K.H. and Li, Y. (2004c). Modern PID patents, software and hardware: their tuning features and ways forward. In *Proceedings of the 10th Annual Conference of Chinese Automation and Computing Society in the UK*, Liverpool, U.K. (To Appear).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter objectives

This chapter presents the motivation behind this work, statement of the problem, proposed approach, main contributions from this work and organisation of this thesis.

1.1 Motivation

Designing and tuning a Proportional plus Integral plus Derivative (PID) controller appears to be conceptually intuitive. However, it can be hard in practice, if multiple (and often conflicting) objectives such as transient behaviour and high stability have to be achieved. Usually, initial designs obtained by all means need to be adjusted repeatedly through computer simulations until the closed-loop system performs or compromises as desired. This stimulates the development of 'intelligent' software tools that can assist engineers to achieve the best overall PID control for the entire operating envelope. Since the invention of PID, numerous tuning rules have been developed, which differ in complexity, flexibility and amount of process knowledge. One important point that seems to miss out during the development of the tuning rule is – in practical process control industrial environments it is obvious that there is a need to achieve satisfactory control performances without adopting complex control architectures, in order to guarantee the best cost/benefit ratio.

Many tuning rules that are accepted by industry are now incorporated into the hardware modules. However, due to the fact that modelling errors, process variations and human errors exist, user intervention in tuning of PID controllers cannot be avoided. Thus, user needs a tuning rule that is simple to understand and quick to apply. Needless to say, we cannot assume that all users are highly educated in control theory. This simply explains why the classical Ziegler-Nichols tuning rule (Ziegler and Nichols, 1942) is still commonly used. This point is further re-enforce in the presence of increasing numbers of available PID tuning software packages, as discussed in Chapter 3.

Most of the available tuning rules either work for a particular type of process or are too complicated to use. Based on the analysis of patented and industrial PID shown in Chapter 3, it is apparent that most of the tuning rules are tuned towards a certain process. As for the more general tuning rules, their designs do not cater for the multi-objective requirements.

The challenge is to develop a simple tuning rule that takes into consideration of the multi-requirements so that most users are comfortable with it. By providing a simple and well-understood PID structure, coupled with an optimal tuning rule with wide applicability range, industry will definitely benefit with a short 'time-to-market', reducing the burden and mistakes of tuning PID controllers. This improves the control system stability, which in turn maximises company's profitability.

1.2 Statement of the Problem

Currently, there exist deficiencies in PID tuning rules. Firstly, most of the tuning rules are designed for the ideal PID structure to deliver a critically- or over-damped closed-loop response. Thus, there are rarely any studies on analysing the performance of other PID structures. Due to the flexibility of the presence of digital controller, the PID structures used are not typical and vary from different process control manufacturers. This may cause most of the tuning rules, especially those proposed in academia, to perform badly. Secondly, even though it is a multi-objective problem, most of the tuning rule designs are based on a single or composite objective. Thus most of the tuning rules will not yield global or multi-objective optimal performance, hence limiting their applicability.

This research aims to devise a universal and practical rule. This is designed using a truly multi-objective technique. The methodology will also cover both under-and over-damped plant responses. It also aims to study the tuning methodology on various PID structures, analyses the possibility of achieving reasonable performance over a wide range of PID structures and find out if there is any industrial practical PID structure that is suitable for those tuning rules that are designed based on the ideal PID structure.

1.3 Proposed Approach

There are a number of objectives in PID tuning rules, namely stability, regulating performance, tracking performance, robustness and noise attenuation. Not all of these objectives are commensurable or consistent. By using classical search and optimisation methods, it is almost impossible to achieve optimal performance. Since this is a multi-objective problem, a set of non-dominated solutions is expected. This should prove useful for the user to understand the trade-offs and to assist in the selection of the final solution. However this is not possible or easily achieved when using classical methods. As a result, this work seeks to explore some of the potential of artificial evolutionary computation techniques, in particular multi-objective evolutionary algorithms. Coupled with evolutionary computation technique, this work attempts to devise a multi-objective PID tuning rule.

There are also a wide variety of evolutionary computation techniques available. In order to understand their behaviour and usage, first an in-depth analysis on the metrics for measuring evolutionary computation performance is conducted. Through this analysis, the weaknesses of the available metrics are identified and remedies are proposed. In addition, a novel visualisation technique is also proposed to assist in the evaluation process. Next, analyses of the commonly cited evolutionary algorithm features are studied and as a result an easy to implement evolutionary computation technique is developed. Its performance is compared against other commonly cited evolutionary algorithms based on a wide range of test problems.

1.4 Main Contributions

The main contributions of this thesis are:

- This research has devised an effective and efficient PID tuning rule that is based on multi-criteria using multi-objective evolutionary algorithms technique. This rule significantly outperforms all other rules due to its simple generic applicability and wide operating range while still achieving global optimal performance based on multi-criteria. As most of the PID tuning rules is developed based on single or composite objective, thus it is not likely to achieve global multi-objective optimal performance. Some other designs are using multi-objective technique, but they are mainly for ad hoc process.
- A modern study and analysis of patented PID tuning rules and their application to practical software packages and industrial controller modules is conducted. The end result provides a comprehensive source of information for readers. It also highlights the differences in academic and industrial practice and thus provides a very good lead for future improvements of PID. As most of the available literature survey on PID control is focused on academic research, although there are a couple of scattered reports on practical PID software packages and industrial controllers. There is however no study conducted on PID patents.
- A comprehensive study and discussion of performance metrics in single- and multi-objective evolutionary algorithms is conducted. It is extremely difficult to analyse

the performance of multi-objective evolutionary algorithms using existing performance metrics. Hence, this study identifies the weaknesses of each metrics and proposes ways to overcome them.

- A novel visualisation technique that enables the viewing of multi-dimensional data in order to assist in the evaluation of multi-objective evolutionary algorithms is proposed. The current visualisation technique for viewing non-dominated solution set is mainly limited to two objectives. For those higher dimensional data, visualisation is very difficult and normally limited to a set of data. Hence these are not commonly found in multi-objective evolutionary algorithm studies.
- An extensive in-depth study is conducted on the evaluation of various multi-objective evolutionary algorithms on a range of test problems using performance metrics and the proposed visualisation technique. Through the use of the proposed visualisation technique, it further reveals each algorithm's behaviour on various problems, which is not commonly seen in any existing empirical studies. This crucial information is very important for the selection of a suitable algorithm and possible enhancements to an algorithm.
- The development of a low-cost Java-based educational PID tuning software tool. This is greatly motivated by the extensive analysis on practical PID software packages, where it is found that there is no tool available for comparing and testing different tuning rules. This tool enables easy comparison and analysis of different available tuning rules together with user-defined settings. Most importantly, it can be deployed on any operating platform with no additional costs.

1.5 Organisation of the Thesis

This thesis is divided into seven chapters, beginning with this Introduction. Chapter 2 begins with preliminary information on the PID controller, which covers its basic functionality, caveats and remedies, and the tuning objectives. This is followed by a brief overview of the evolutionary computation methodology that will be employed in this research. A description of the classical method used to handle multiple objective problems is used to lead onto the description of evolutionary algorithms and its extension to handle multiple objective problems.

Chapter 3 aims to identify the trend and direction of practical PID development. It begins with a study on the patents on PID controller tuning methodologies, in order to present an overview of the methods that are being patented. Then a study is conducted on the tuning methods found in practical software packages. This also serves as a one-stop information for anyone looking for tuning tools in assisting their work on PID controller. Lastly, a study on four process control companies hardware modules are given, with details on their own incorporated tuning methodology.

Chapter 4 presents the analysis of the proposed evolutionary algorithm methodology performance. The proposed methodology is presented before proceeding to study and analyse the performance metrics used to evaluate their capability in handling single- and multi-objective problems. It starts by looking into single-objective case, where the performance metrics are simple and straightforward. Then it shows the difficulty in multi-objective case and discusses the merits and demerits of the available performance metrics. A novel and effective visualisation technique is proposed, in an attempt to assist users in making decision or selection in a multi-dimensional case, whereby the performance metrics fails. This chapter concludes by performing empirical studies on various multi-objective evolutionary algorithms and discusses the results using performance metrics and the proposed visualisation technique.

Chapter 5 presents the detailed development on the search for globally optimal multi-objective PID tuning rules, PIDeasy. First, it looks into the plant modelling technique where critical plant information is extracted in order to perform the tuning operation. Next, it describes the employed PID structure and its consideration. Followed by the objectives that are considered in the search and learning of multi-optimal PID tuning rules. Subsequently, the process and result of the search and learning are explained in details. This chapter concludes by presenting a software tool that is developed for computing PID parameters and comparison of different tuning rules.

Chapter 6 evaluates the performance of the proposed PIDeasy tuning method against a set of well-recognised PID benchmark test systems and three laboratory systems. First, it compares the performance of PIDeasy with other selected tuning rules over a range of 28 processes. This is a offline computer simulation test and the performances are compared and discussed based on the gain and phase margins and a set of performance indices. Next, PIDeasy is being tested online on three laboratory systems. Each of the online systems exhibits a different behaviour which may used to represent real industrial plants. The first is a DC motor where there is negligible transport delay and very fast response. The second one is a heating system which has a transport delay and is susceptible to atmosphere interferences. The last one is a coupled tanks system where it has minimal transport delay but has a very slow response.

Chapter 7 concludes this thesis by summarising the work that is being done and results achieved. Next, a section on further work first identified the limitations of the results and then suggests where further developments are needed.

Chapter 2

PID Controller and Evolutionary Learning Methods

Chapter objectives

This chapter presents the functionality, design and tuning of a PID controller, and the evolutionary learning methods that are used in the development of a PID tuning rule.

2.1 Introduction

This chapter provides an overview on PID technology. It outlines existing problems and difficulties in understanding and tuning a PID controller. Next, it looks at a common traditional attempt in the search for a multi-optimal PID tuning rule. From there, the problems can be easily highlighted on the methodologies and it can show why evolutionary algorithms are more suitable in handling this type of problems.

2.2 Three-Term Functionality, Design and Tuning of PID Control

With its three-term functionality covering treatment to both transient and steady-state responses, PID control offers the simplest and yet most efficient solution to many real-world control problems. Since the invention of PID control in 1910 (largely owing to Elmer Sperry's ship autopilot), and the Ziegler-Nichols' (ZN) straightforward tuning methods in 1942 (Ziegler and Nichols, 1942), the popularity of PID control has grown tremendously. With advances in digital technology, the science of automatic control now offers a wide spectrum of choices for control schemes. However, more than 90% of industrial controllers are still implemented according to the PID algorithm particularly at the lowest level (Åström and Hägglund, 1996). As no other controllers match the simplicity, clear functionality, applicability and ease of use offered by the PID controller (Wang *et al.*, 1995). Its wide application has stimulated and sustained the development of various PID tuning techniques, sophisticated software packages and hardware modules (Ang *et al.*, 2004a).

The success and longevity of PID controllers were characterised in a recent IFAC workshop, where over 90 papers dedicated to PID research were presented (Quevedo and Escobet, 2000). With much of academic research in this area maturing and entering the region of 'diminishing returns', the trend in present research and development (R&D) of PID technology appears to be focused on the integration of available methods in the form of software so as to get the best out of PID control (IEE Digest, 1996; Quevedo and Escobet, 2000). A number of software-based techniques have also been realised in hardware modules to perform 'on-demand tuning', whilst the search still goes on to find the next key technology for PID tuning (Marsh, 1998).

2.2.1 Three-Term Functionality and the Parallel Structure

A PID controller may be considered as an extreme form of a phase lead-lag compensator with one pole at the origin and the other at infinity. Similarly, its cousins, the PI and PD controllers, can also be regarded as extreme forms of phase-lag and phase-lead compensators, respectively. The PID controller is also known as the 'Three-Term' controller, whose transfer function in 'ideal form' is (Figure 2.1):

$$G(s) = K_P\left(1 + \frac{1}{T_I s} + T_D s\right) \tag{2.1}$$

where K_P is the proportional gain, T_I the integral time constant and T_D the derivative time constant. This can also be transformed to another form commonly known as the 'parallel form' (Figure 2.2):

$$G(s) = K_P + \frac{K_I}{\mathrm{s}} + K_D s \tag{2.2}$$

where K_I is the integral gain and K_D the derivative gain. The 'Three-Term' functionalities are highlighted by:

- The proportional term – providing an overall control action proportional to the error signal through an all-pass gain factor;
- The integral term – reducing steady-state errors through low-frequency compensation by an integrator;
- The derivative term – improving transient response through high-frequency compensation by a differentiator.

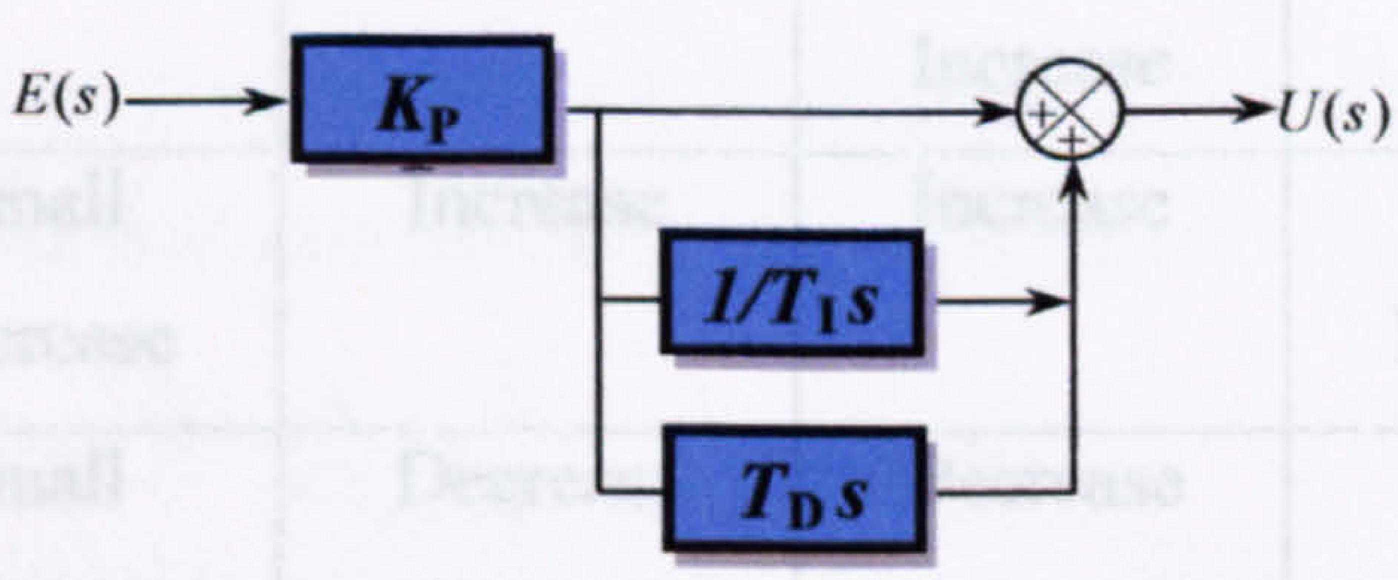

Figure 2.1 PID Structure – Ideal Form

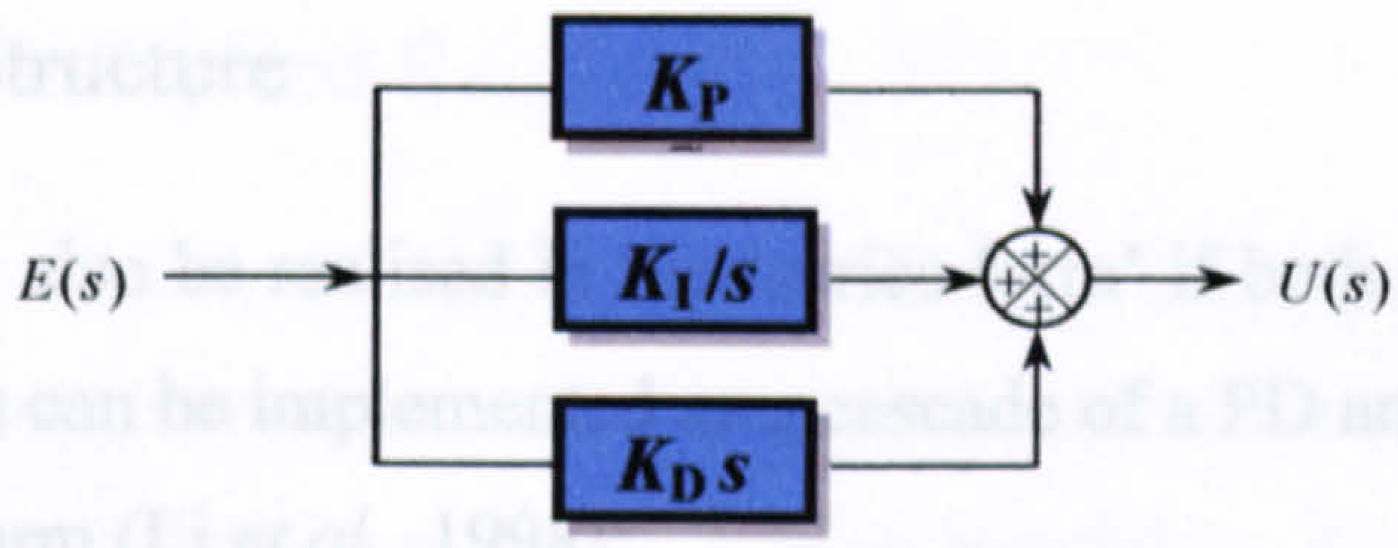

Figure 2.2 PID Structure – Parallel Form

The individual effects of these three terms on a closed-loop system are summarised in Table 2.1. Note that this table serves as a first guide for stable open-loop plants only. For optimum performance, K_P, K_I (or T_I) and K_D (or T_D) are mutually dependent in tuning.

Regarding the message that increasing the derivative gain, K_D, will lead to improved stability is commonly conveyed from academia to industry. However, practitioners have often found that the derivative term can behave against such anticipation particularly when there exists a transport delay (Li *et al.*, 1998; Quevedo and Escobet, 2000). Frustration in tuning K_D has hence made many practitioners switch off or even exclude the derivative term. This matter has now reached the point that requires clarification, which will be discussed in Section 2.2.5.

Table 2.1 Characteristics of P, I, and D Controllers

Closed-Loop Response	**Rise Time**	**Overshoot**	**Settling Time**	**Steady-State Error**	**Stability**
Increase K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increase K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increase K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

2.2.2 The Series Structure

A PID controller may also be realised in the 'series form' if both zeros are real, i.e. $T_I \geq 4T_D$. In this case, (2.1) can be implemented as a cascade of a PD and PI controller, shown in Figure 2.3, in the form (Li *et al.*, 1998):

$$G(s) = K_P(\alpha + T_D s)\left(1 + \frac{1}{\alpha T_I s}\right) \tag{2.3}$$

where

$$\alpha = \frac{1 \pm \sqrt{1 - 4T_D / T_I}}{2} > 0 \tag{2.4}$$

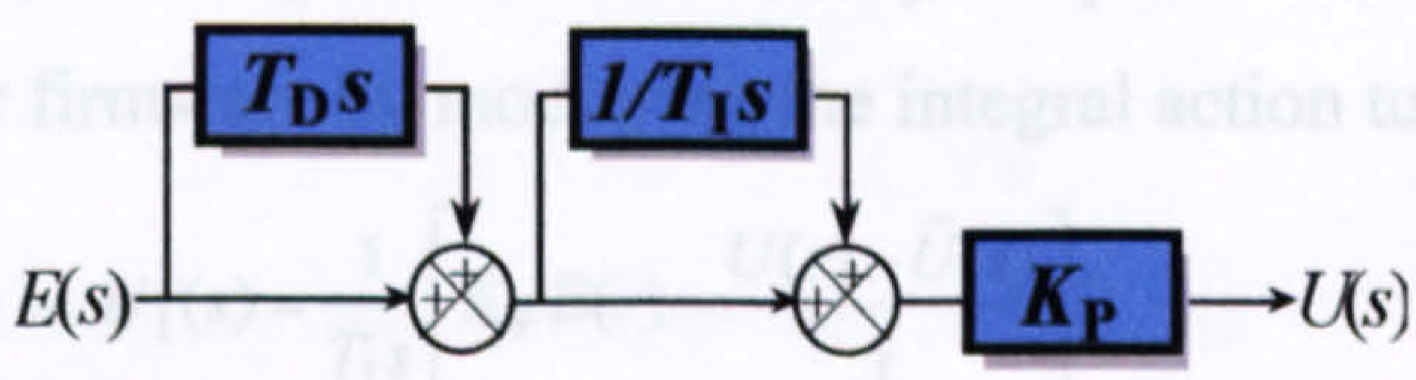

Figure 2.3 PID Structure – Series Form

Another similar form, which is known as the 'classical form', is more commonly used in the industry (O'Dwyer, 2003):

$$G(s) = K_P\left(1 + \frac{1}{T_I s}\right)\left(\frac{1 + T_D s}{1 + \frac{T_D}{\beta} s}\right) \tag{2.5}$$

2.2.3 Effect of the Integral Term on Stability

Refer to (2.1) or (2.3) for $T_I \neq 0$ and $T_D = 0$. It can be seen that, adding an integral term to a pure proportional term will increase the gain by a factor of:

$$\left|1 + \frac{1}{j\omega T_I}\right| = \sqrt{1 + \frac{1}{\omega^2 T_I^2}} > 1, \ \forall \omega \tag{2.6}$$

and will increase the phase-lag at the same time since:

$$\angle\left(1 + \frac{1}{j\omega T_I}\right) = \tan^{-1}\left(\frac{-1/\omega T_I}{1}\right) < 0, \ \forall \omega \tag{2.7}$$

Hence, both stability gain and phase margins will be reduced, i.e., the closed-loop system will become more oscillatory or potentially unstable.

2.2.4 Integrator Windup and Remedies

If an actuator realises the control action has an effective range limit, then the integrator may saturate and future correction will be ignored until the saturation is offset. This causes low-frequency oscillations and may lead to instability. A usual measure taken to counteract this effect is 'anti-windup' (Shinskey, 1994; Åström and Hägglund, 1995; Bohn and Atherton, 1995; Peng *et al.*, 1996). Nearly all software packages and hardware modules have implemented some form of integrator windup protection.

As most modern PID controllers are implemented in digital processors, they can accommodate more mathematical functions and modifications to the standard three terms shown in (2.1) to (2.3). A simple and most widely adopted anti-windup scheme can be realised in software or firmware by modifying the integral action to:

$$U_I'(s) = \frac{1}{T_I s}\left[K_P E(s) - \frac{U(s) - \tilde{U}(s)}{\gamma}\right] \tag{2.8}$$

where $\tilde{U}(s)$ represents the saturated control action and γ is a correcting factor. It is found that the range of [0.1,1.0] for γ results in extremely good performance with any reasonably tuned PID parameters (Li *et al.*, 1998).

It is also reported that, in the 'series form', the PI part may be implemented to counter actuator saturation without the need for a separate anti-windup action, as shown in Figure 2.4 (Shinskey, 1994; Åström and Hägglund, 1995). When there is no saturation, the feedforward-path transfer is unity and the overall transfer from $U_{PD}(s)$ to $U(s)$ is the same as the last factor in (2.3).

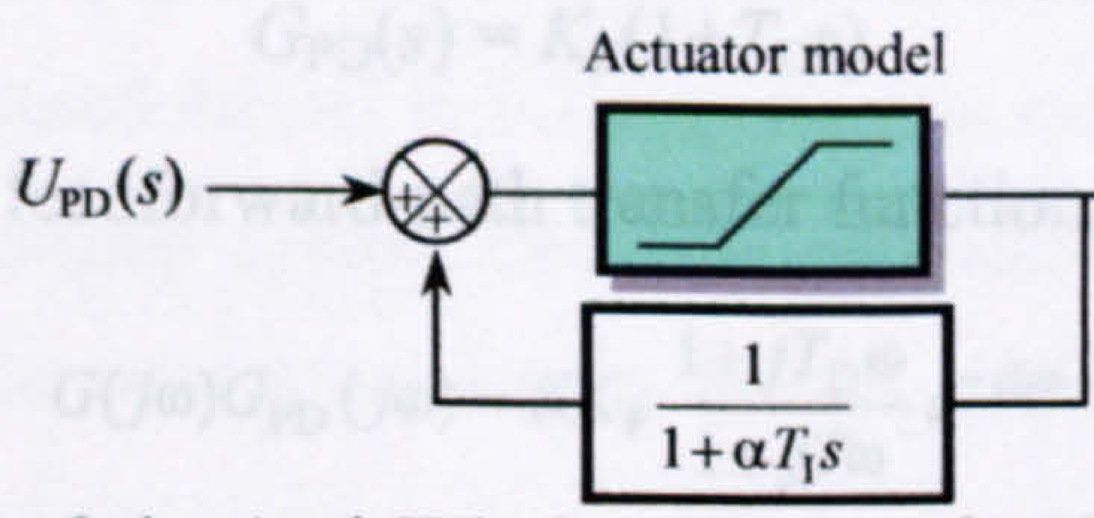

Figure 2.4 Anti-Windup PI Part of a 'Series Form'

2.2.5 Effect of the Derivative Term on Stability

Generally, derivative action is valuable as it provides useful phase lead to offset phase lag caused by integration. It is also particularly helpful in shortening the period of the loop and thereby hastening its recovery from disturbances. It can have a more dramatic effect on the behaviour of second-order plants that have no significant dead-time than first-order plants (Shinskey, 1994).

However, the derivative term is often misunderstood and misused especially if delay exists. For example, it has been widely perceived in the control community that adding a derivative term will improve stability. It will be shown here that this perception is not always valid. In general, adding a derivative term to a pure proportional term will reduce phase lags by:

$$\angle(1+j\omega T_D) = \tan^{-1}\frac{\omega T_D}{1} \in [0, \pi/2], \ \forall\omega \tag{2.9}$$

which alone tends to increase the phase margin. In the meantime, however, the gain will be increased by a factor of:

$$|1+j\omega T_D| = \sqrt{1+\omega^2 T_D^2} > 1, \ \forall\omega \tag{2.10}$$

and hence the overall stability may be improved or degraded.

To prove that adding a differentiator could actually destabilise the closed-loop system, consider without loss of generality a common first-order lag plus delay plant as described by:

$$G(s) = \frac{K}{1+Ts}e^{-Ls} \tag{2.11}$$

where K is the process gain; T is the process time-constant; and L is the process dead-time or transport delay. Suppose that it is controlled by a proportional controller with gain K_P and now a derivative term is added. This results in a combined PD controller as given by:

$$G_{PD}(s) = K_P(1+T_D s) \tag{2.12}$$

The overall open-loop feedforward-path transfer function becomes:

$$G(j\omega)G_{PD}(j\omega) = KK_P\frac{1+jT_D\omega}{1+jT\omega}e^{-jL\omega} \tag{2.13}$$

with gain becoming:

$$\begin{aligned} |G(j\omega)G_{PD}(j\omega)| &= KK_P\sqrt{\frac{1+T_D^2\omega^2}{1+T^2\omega^2}} \\ &\geq KK_P \min\left(1, \frac{T_D}{T}\right) \end{aligned} \tag{2.14}$$

where the inequality has been obtained because $\sqrt{\frac{1+T_D^2\omega^2}{1+T^2\omega^2}}$ is monotonic with ω. This implies that the gain is not less than 0 dB if $T_D \geq T$ and $KK_P \geq 1$ or $T_D \leq T$ and

$$T_D \geq \frac{T}{KK_P} \tag{2.15}$$

In these cases, the 0 dB gain crossover frequency, ω_c, is at infinite, where the phase

$$\begin{aligned}\angle G(j\omega_c)G_{PD}(j\omega_c) &= \tan^{-1}\frac{T_D\omega_c}{1} - \tan^{-1}\frac{T\omega_c}{1} - L\omega_c \\ &= \pi/2 - \pi/2 - \infty \qquad < -\pi\end{aligned} \tag{2.16}$$

Hence, by Bode or Nyquist criterion, there exist no stability margins and the closed-loop system will be unstable. This shows an example that adding a derivative term will not always improve stability, contrary to the general perception that derivative term will improve stability.

This phenomenon could have contributed to the difficulties in the design of a full PID controller and also to the reason that 80% of PID controllers in use have the derivative part omitted or switched off (IEE Digest, 1996). This means that the functionality and potential of a PID controller is not fully exploited. Nonetheless, it is shown that the use of a derivative term can increase stability robustness and help maximise the integral gain so as to achieve the best performance (Åström and Hägglund, 2001). However, care must be taken, as it is difficult to tune the differentiator properly. An example is given in Figure 2.5 and 2.6 for plant (2.11) with $K = 10$, $T = 1$ sec. and $L = 0.1$ sec., which is initially controlled by a PI controller with $K_P = 0.644$ and $T_I = 1.03$ sec. It can be seen that if a differentiator is added with $T_D = 0.0303$ sec., both the gain and phase margins will be maximised whilst the transient response improves to the best. However, if T_D is increased further to 0.1 sec, the gain margin and transient response will deteriorate. The closed-loop system can even be destabilised if the derivative gain is increased to 20% of the proportional gain. Hence, the derivative term should be tuned and used properly.

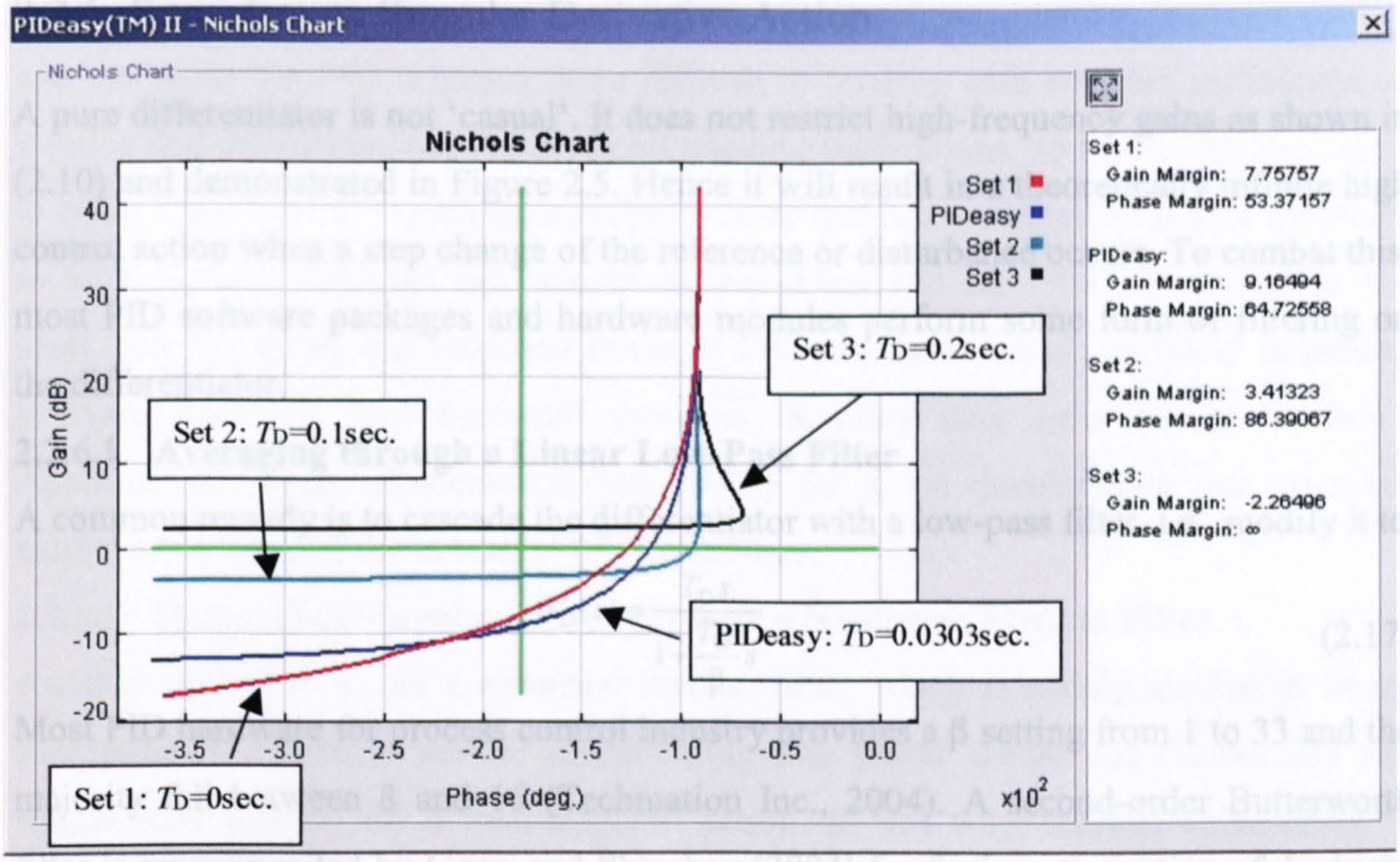

Figure 2.5 Increasing Derivative Gain could Decrease Stability Margins and Destabilise the Closed-Loop System

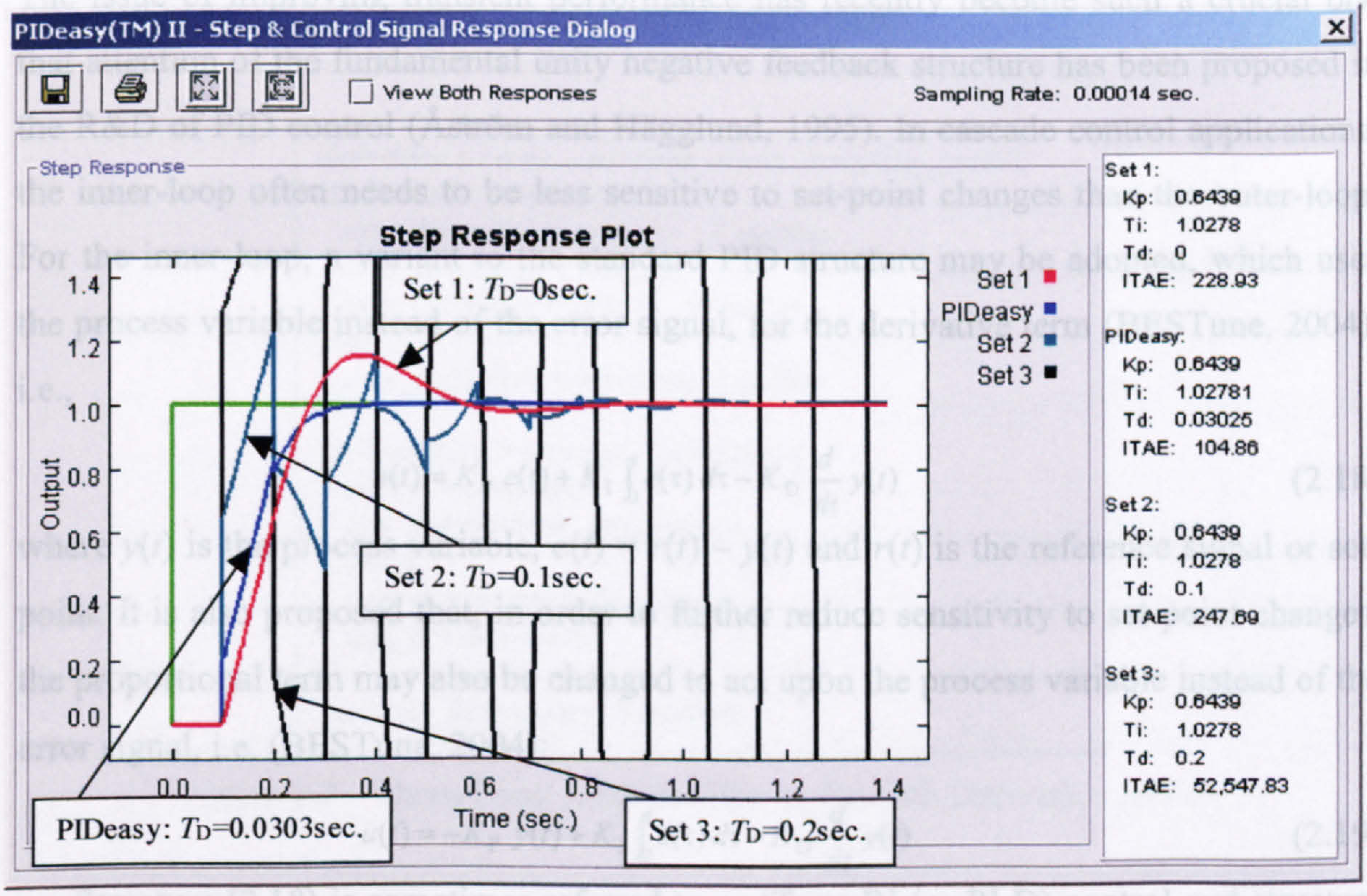

Figure 2.6 Effect of the Closed-Loop System on Increasing Derivative Gain in Time-Domain

2.2.6 Remedies on Singular Derivative Action

A pure differentiator is not 'casual'. It does not restrict high-frequency gains as shown in (2.10) and demonstrated in Figure 2.5. Hence it will result in a theoretically infinite high control action when a step change of the reference or disturbance occurs. To combat this, most PID software packages and hardware modules perform some form of filtering on the differentiator.

2.2.6.1 Averaging through a Linear Low-Pass Filter

A common remedy is to cascade the differentiator with a low-pass filter, i.e., modify it to

$$G'_D(s) = \frac{T_D s}{1 + \frac{T_D}{\beta} s} \tag{2.17}$$

Most PID hardware for process control industry provides a β setting from 1 to 33 and the majority fall between 8 and 16 (Techmation Inc., 2004). A second-order Butterworth filter is recommended by Gerry and Shinskey (2004) for further attenuation of the high-frequency gains.

2.2.6.2 Modified Structure

The issue of improving transient performance has recently become such a crucial one that attention of the fundamental unity negative feedback structure has been proposed in the R&D of PID control (Åström and Hägglund, 1995). In cascade control applications, the inner-loop often needs to be less sensitive to set-point changes than the outer-loop. For the inner-loop, a variant to the standard PID structure may be adopted, which uses the process variable instead of the error signal, for the derivative term (BESTune, 2004), i.e.,

$$u(t) = K_P\, e(t) + K_I \int_0^t e(\tau)\, d\tau - K_D \frac{d}{dt} y(t) \tag{2.18}$$

where $y(t)$ is the process variable, $e(t) = r(t) - y(t)$ and $r(t)$ is the reference signal or set-point. It is also proposed that, in order to further reduce sensitivity to set-point changes, the proportional term may also be changed to act upon the process variable instead of the error signal, i.e. (BESTune, 2004):

$$u(t) = -K_P\, y(t) + K_I \int_0^t e(\tau)\, d\tau - K_D \frac{d}{dt} y(t) \tag{2.19}$$

Structure (2.18) is sometimes referred to as 'Type B' (or PI-D) control and structure (2.19) as 'Type C' (or I-PD) control, whilst structures (2.1) to (2.3) as 'Type A' PID control. Note that, Types B and C alter the foundations of conventional feedback control and can make the PID schemes more difficult to analyse with standard techniques on stability and robustness, etc. For set-point tracking applications, however, one alternative to using Type B or C is perhaps a set-point filter that has a critically-damped dynamics so as to achieve soft-start and smooth control (Feng and Li, 1999). Nevertheless, the ideal, parallel, series and modified forms of PID structures can all be found in present software packages and hardware modules. Readers may refer to Techmation's Applications Manual (Techmation Inc., 2004) for a list documenting the structures employed in some of the industrial PID controllers.

2.2.6.3 Removal of Singular Action through a Nonlinear Median Filter

Another method is to use a nonlinear median filter, which is widely applied in image processing. It compares several neighbouring data points around the current one and selects their median for a 'non-singular' action. In this way, unusual spike noise or unwanted action resulting from a step command or disturbance, for example, will be filtered out completely. Pseudo code of a three-point median filter is illustrated in Figure 2.7 (Li *et al.*, 1998). The main benefit of this method is no extra parameter is needed, though it is not suitable for use in under-damped processes.

```
derivative = (error – previous_error) / sampling_period;
if (derivative > max_d)
  new_derivative = max_d;    // median
else if (derivative < min_d)
  new_derivative = min_d;  // median
else
  new_derivative = derivative;// median

if (derivative > previous_derivative) {
  max_d = derivative;
  min_d = previous_derivative;
} else {
  max_d = previous_derivative;
  min_d = derivative;
}
previous_derivative = derivative;
```

Figure 2.7 Three-Point Median Filter to Smooth Derivative Action

2.2.7 Tuning Objectives and Existing Methods

Pre-selection of a controller structure can pose a challenge in applying PID control. As vendors often recommend their own designs of controller structures, their tuning rules for a specific controller structure do not necessarily perform well with other structures. One solution seen is to provide support for individual structures in software. Readers may refer to (Gerry, 1987; Kaya and Scheib, 1988; Luyben, 2001; Eder, 2003) for detailed discussion on the use of various PID structures. Nonetheless, controller parameters are tuned such that the closed-loop control system would be stable and would meet given objectives associated with:

- Stability robustness;
- Set-point following and tracking performance at transients, including rise-time, overshoot and settling time;
- Regulation performance at steady-state, including load disturbance rejection;
- Robustness against plant modelling uncertainty;
- Noise attenuation and robustness against environmental uncertainty.

With the given objectives, tuning methods for PID controllers can be grouped according to their nature and usage, as follows (Åström and Hägglund, 1995; Li *et al.*, 1998; Feng and Li, 1999):

- Analytical methods – PID parameters are calculated from analytical or algebraic relations between a plant model and an objective (such as Internal Model Control or Lambda tuning). These can lead to an easy-to-use formula and can be suitable for use with on-line tuning, but the objective needs to be an analytical form and the model must be accurate;
- Heuristic methods – These are evolved from practical experience in manual tuning (such as Ziegler-Nichols tuning rule) and from artificial intelligence (including expert systems, fuzzy logic and neural networks). Again, these can serve in the form of a formula or a rule-based for on-line use, often with trade-off design objectives;
- Frequency response methods – Frequency characteristics of the controlled process are used to tune the PID controller (such as loop-shaping). These are often off-line and academic methods, where the main concern of design is stability robustness;

- Optimisation methods – These can be regarded as a special type of optimal control, where PID parameters are obtained ad hoc using an off-line numerical optimisation method for a single composite objective or using computerised heuristics or an evolutionary algorithm for multiple design objectives. These are often time-domain methods and mostly applied off-line;
- Adaptive tuning methods – These are for automated on-line tuning, using one or a combination of the above methods based on real-time identification.

The above classification does not set an artificial boundary and some methods applied in practice may belong to more than one category. An excellent summary on PID tuning methods can be found in Åström and Hägglund (1995), Gorez (1997), Quevedo and Escobet (2000) and O'Dwyer (2003). However, no tuning method so far can replace the simple ZN method in terms of familiarity and ease of use to start with. Further, there exists a lack of generic methods that can be quickly applied to the design of on-board or on-chip controllers for a wide range of consumer electronics, domestic appliances, mechatronic systems and micro-electro mechanical systems (MEMS) where PID controller are easily integrated. Over the past half century, search goes on to find the next key technology for PID tuning and modular realisation (Marsh, 1998).

2.3 Evolutionary Computation Methodology

Real world problems often entail simultaneous optimisation of multiple, possibly conflicting, objectives. Although many of these objectives can be represented sufficiently to allow quantitative analysis, incorporating them into a decision making process requires a multi-objective treatment and optimisation. In a single optimisation problem, the notion of optimality is the best (the minimum or the maximum) value of the objective function. In a multi-objective optimisation problem, however, the notion of optimality is hard to define. Thus, the concept of Pareto optimality is often used. In general, no single solution is considered the best with respect to all the objectives simultaneously. There exists, however, a set of 'best compromised' solutions that are strictly better than the remaining ones in the search space. This set of solutions is known as the Pareto optimal set or set of the non-dominated solutions. It describes the trade-offs in the problem, and helps user to understand the options available, therefore enabling the selection of a final solution.

Multi-objective optimisation is no doubt a very important research topic both for scientists and engineers, because of the multi-objective nature of most real-world problems. Evolutionary algorithms (EAs) seem particularly suitable to solve multi-objective optimisation problems because they deal simultaneously with a set of possible solutions (the so-called population). They facilitates the findings of an entire set of Pareto optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional search and optimisation methods. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front, whereas these two issues are a real concern for traditional search and optimisation methods.

2.3.1 Basic Definitions

Consider a test problem for testing an optimisation, learning or search algorithm. Suppose that its objective function (cost function, performance index or fitness function) is $f(x)$: $X \rightarrow F$, which may be evaluated via analytical calculations or numerical simulations. Here $X \subseteq R^n$ spans the entire search or possible solution space in n dimensions, $x \in X$ represents the n collective variables or parameters to be optimised, $F \subseteq R^M$ represents the M dimensional space of all possible objective values, and $f \in F$ represents the collection of m objective elements. For simplicity, we have used the real space and enclosed X as a genotype, whose phenotype correspondents may take the form of integer or logic values for non-numerical search and machine learning problems.

Denote the theoretical objective vector that may be ultimately reached as:

$$f_0 = obj\ \{f(x)\} \in F \tag{2.20}$$

Note that elements in f_0 can have separate objectives, i.e., some for maximisation and some for minimisation. A $x_0 \in X$ that satisfies:

$$f(x_0) = f_0 \tag{2.21}$$

is said to be a corresponding theoretical solution to the optimisation problem.

Also note that, for a non-dominant or non-commensurate multi-objective optimisation problem, f_0 represents a collection of individual theoretical objectives that may only be reached separately by different solutions. In this case, there exists not a single or dominant solution and hence a quasi-theoretical solution needs to be defined.

A general multi-objective problem consists of a number of objectives to be optimised simultaneously and is associated with a number of inequality and equality constraints (for theoretical background, refer to Cohon (1978)). Such a problem can be stated as follows:

$$\textit{Minimise or Maximise} \quad f_i(x),\ i=1,\ldots,M \tag{2.22}$$

$$\textit{subject to} \begin{cases} g_j(x)=0 & j=1,\ldots,J \\ h_k(x)\le 0 & k=1,\ldots,K \end{cases} \tag{2.23}$$

where f_i is an element of the objective function f, M is the number of objectives, J is the number of equality constraints and K is the number of inequality constraints.

Without loss of generality and considering a minimisation objective, a vector x_1 is said to be partially less than another vector x_2 when:

$$f_i(x_1) \le f_i(x_2),\ \forall i \tag{2.24}$$

and there exists at least one i such that $f_i(x_1) < f_i(x_2)$. We then say that solution x_1 dominates solution x_2. For example, in the case of minimisation of two objectives,

$$\begin{gathered} \textit{Minimise}\ f(x) = [\, f_1(x)\ f_2(x)\,]^T \\ \textit{such that}\ \ x \in X \end{gathered} \tag{2.25}$$

A potential solution x_1 is said to dominate solution x_2 iff:

$$f_1(x_1) < f_1(x_2) \cap f_2(x_1) \le f_2(x_2) \cup f_1(x_1) \le f_1(x_2) \cap f_2(x_1) < f_2(x_2) \tag{2.26}$$

In words, this definition says that x_1 is Pareto optimal if there exists no feasible $x_2 \in X$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Hence, this concept usually gives not a single solution, but rather a set of solutions called the Pareto optimal set. The vector x_1 corresponding to the solutions included in the Pareto optimal set is termed non-dominated. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front.

The above domination principle is based on the assumption that there are no constraints. However, it can be easily extended to handle constraints by slight modification to the principle. Solution x_1 is said to constraint-dominate solution x_2, if any of the following conditions is true:

1. Solution x_1 is feasible and solution x_2 is not.
2. Solutions x_1 and x_2 are both infeasible, but solution x_1 has a smaller overall constraint violation.
3. Solutions x_1 and x_2 are feasible and solution x_1 dominates solution x_2 based on (2.26).

2.3.2 Classical Methodology

Classical ways of handling multi-objective problems usually require aggregating the objectives into a single parameterised objective function. One of the most commonly used is the weighted sum method. This method adds all the objective functions together using different weighting coefficients for each of them. This transforms a multi-objective problem into a scalar optimisation problem of the form:

$$\min \sum_{i=1}^{M} w_i f_i(x) \tag{2.27}$$

where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the M objective functions. It is usually assumed that:

$$\sum_{i=1}^{M} w_i = 1 \tag{2.28}$$

This has the advantage of obtaining a single compromised solution. However, the single compromised solution may not satisfy the decision makers, and thus the importance of each objective function must be known prior to setting the proper weights for each objective function. This method's main drawback is that it cannot generate proper members of the Pareto optimal set when the Pareto front is non-convex regardless of the weights used (Das and Dennis, 1997).

2.3.3 Evolutionary Algorithms

Evolution is a ubiquitous natural force that has shaped all life on Earth for approximately 3.2 billion years. For several thousand years, humanity has also utilised artificial selection to shape domesticated plant and animal species. In the past few decades, however, science has learned that the general principles at work in natural evolution can also be applied to a completely artificial environment. In particular, within Computer Science, the field of automated machine learning has adopted algorithms based on the mechanisms exploited by natural evolution.

Darwin (1859) first proposed that there are four essential requirements for the process of evolution to occur:

1. Reproduction of some individuals within a population.
2. A degree of variation that affects probability of survival.
3. Heritable characteristics, that is, similar individuals arise from similar parents.
4. Finite resources, which drive competition and fitness selection.

The consequence of these processes is the gradual adaptation of the individuals in a population to the specific ecological niche they occupy. This can therefore be viewed as a form of long-term learning by a population, on the characteristics suited to their particular environment.

The term evolutionary computing (EC) refers to the study of the foundations and applications of certain heuristic techniques based on the principles of natural evolution. In spite of this fact, these techniques are traditionally classified into three main categories, namely, genetic algorithms (GAs), evolution strategies (ESs) and evolutionary programming (EP). This classification is based on some details and historical development facts rather than major functioning differences. In fact, their biological basis is essentially the same.

It is particularly useful to consider the history of evolution within computing as it covers much of the timeframe of computing itself. Some of the earliest work can be traced back to Friedberg (1958), who introduced the idea of an evolutionary algorithm approach for automatic programming. Later significant developments included the creation of EP by Fogel *et al.* (1966). Holland (1975) founded the initial work on GAs at the University of Michigan. Parallel work was also initiated by Bienert *et al.* (1966) in ESs. However, the major barrier to the early adoption of evolutionary algorithm in the computing domain came from opposition within the computer science community itself. This was often based on the mistaken belief that such algorithms, with probabilistic processes as a core mechanism, would not be amenable to produce functional code. The second barrier to evolutionary algorithm development was the problem that contemporary computing technology in software, and particularly hardware, in the early

1970s was barely capable of generating useful results in acceptable time scales (i.e., less than a few weeks). This problem added to the belief that such methods, while theoretically interesting, would never be capable for useful applications.

Evolutionary algorithm is now frequently used as a generic term which incorporates GA, ES, EP and their variants. The origin of EA was an attempt to mimic some of the processes taking place in natural evolution. Although the details of biological evolution are not completely comprehended (even nowadays), there exist some points supported by strong experimental evidences:

- Evolution is a process operating over chromosomes rather than over organisms. The former are organic tools encoding the structure of a living being, i.e., a creature is 'built' decoding a set of chromosomes.
- Natural selection is the mechanism that relates chromosomes with the efficiency of the entity they represent, thus allowing those efficient organisms which are well-adapted to the environment to reproduce more often than those which are not.
- The evolutionary process takes place during the reproduction stage. There exists a large number of reproductive mechanisms in Nature. Most common ones are mutation (that causes the chromosomes of offspring to be different to those of the parents) and recombination (that combines the chromosomes of the parents to produce the offspring).

All EAs have two prominent features which distinguish themselves from other search algorithms. Firstly, they are all population-based. Secondly, there are communication and information exchanges among individuals in a population. Such communication and information exchanges are the result of selection and/or recombination in EAs. A general framework of EAs can be summarised in Figure 2.8. The interested reader may refer to Bäck *et al.* (1997) for a detailed description and discussion on evolutionary computation.

1. Set t = 0
2. Generate initial population P(t) at random
3. Evaluate the fitness of each individual in P(t)
4. REPEAT
   (a) Select parents from P(t)
   (b) Apply recombination and/or mutation to the parents and produce children
   (c) Evaluate the fitness of children
   (d) Select individual from the children or parents and children for next generation P(t+1)
5. UNTIL terminating criteria met

Figure 2.8 A General Framework of Evolutionary Algorithm

2.3.4 Multi-Objective Evolutionary Algorithms

A difference between a classical search and optimisation method and an EA is that in the latter, a population of solutions is processed in every iteration (or generation). This feature alone gives an EA a tremendous advantage for use in solving multi-objective optimisation problems. After the pioneering work by Schaffer (1987) in the area of evolutionary algorithms for multi-objective optimisation, development of multi-objective evolutionary algorithms (MOEAs) has taken multiple directions. For a thorough discussion of evolutionary algorithms for multi-objective optimisation, the interested reader may refer to Fonseca and Fleming (1995a), Van Veldhuizen and Lamont (1998), Coello Coello (1999), Deb (2001), Coello Coello *et al.* (2002), Jones *et al.* (2002) and Sarker *et al.* (2002).

There are numerous variations of MOEAs proposed and the main differences are the way they maintain the solutions' diversity and fitness assignment and selection of solutions for next iteration (or generation). Preservation of the solutions' diversity is crucial, not only to avoid premature convergence, but also not to lose any potentially efficient solution. The aim of an MOEA is to find a set of well-distributed solutions close to the true Pareto-optimal front.

Here, some of the most representative MOEAs will be looked into and to decide which methodology to adopt for this research.

2.3.4.1 Fitness Assignment and Selection

There are generally two methods used to assist in the selection, namely, the Goldberg method (Goldberg, 1989) and Fonseca and Fleming method (Fonseca and Fleming, 1993). The similarity is that both sort and rank the solutions according to their degree of non-dominance. The difference is in how they rank the solutions.

Goldberg's method sorts the solutions according to the level of non-dominance. Each solution must be compared with every other solution to find if it is dominated. This process is continued to find the members of the first non-dominated class for all solutions and they are assigned as rank 1. At this stage, all solutions in the first non-dominated front are found. In order to find solutions belonging to the next front, the solutions of the first front are temporarily discounted and the above procedure is performed again. The procedure is repeated to find subsequent fronts until all solutions are ranked.

Fonseca and Fleming have proposed a slightly different scheme, whereby an individual rank corresponds to the number of solutions by which it is dominated. Non-dominated solutions are, therefore, all assigned the same rank, while dominated ones are penalised according to the density of the corresponding region of trade-off surface.

In this way, solutions can be differentiated according to non-dominance instead of weighted sum method, for example.

2.3.4.2 Diversity Preservation

Diversity preservation is used in almost every known MOEAs in order to maintain uniform distribution among solutions. It is mainly used when the amount of non-dominated solutions exceed a user-defined amount. Then in this case, this operation will ensure those similar solutions (or solutions that are very close together) are removed while still maintaining the uniform distribution of the solutions. There are currently a few diversity preservation techniques available in the context of MOEAs, namely, hyper-grid, clustering and crowding methods.

The hyper-grid method deterministically divides the entire objective space in d^n subspaces (where d is the user-defined depth parameter and n is the number of decision variables) and by updating the subspaces dynamically.

The clustering method initially forms N clusters for each of the solution in the population. Thereafter, the distance between all pairs of clusters is calculated by first finding the centroid (the point with minimal average distance to all other points in the cluster) of each cluster and then calculating the Euclidean distance between the centroids. Two clusters having the minimum distance are merged together into a bigger cluster. This procedure is continued until the desired number of clusters is identified. Finally, with the remaining clusters, the solution closest to the centroid of the cluster is retained and all other solutions from each cluster are deleted.

The crowding method requires sorting of the population according to each objective function value in their ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute difference in the function values of two adjacent solutions. This process is continued with other objective functions. The overall crowding distance is calculated as the sum of individual values corresponding to each objective.

2.4 Summary

PID, a structurally simple and generally applicable control technique, stems it success largely from the fact that it just works very well with a simple and easy to understand structure. While a vast amount of research results are published in the literature, there exists a lack of information exchange and this can lead to some misunderstanding between academia and process control industry. For example, there exists no standardisation of a generic PID structure. This is particularly evident with analogue PID controllers being replaced by digital ones, where flexibility in software permits ad hoc patches for some local optimality. It has led to unnecessary complication and extra learning curve in tuning PID controllers. This problem becomes severe when there are multiple control loops and different brands or models of PID controllers in one application. These may explain why the argument exists that academically proposed tuning rules do not work well on industrial PID controllers, while it is desired that years of research results help industrial practice more for improved quality and profitability.

Thus, the next chapter will analyse and discuss the PID tuning rules that are being patented and, employed in practical PID software packages and hardware modules. Nevertheless, a good PID tuning rule should be globally optimal with consideration to all the objectives listed in Section 2.2.7. In this aspect, the design will be a typical multi-objective problem. However, the area of multi-objective optimisation problems is extremely complex and mathematically difficult. As we gain more knowledge of complex systems, in all walks of life and areas of research, we can see the need to understand how objectives co-evolve dynamically, and to determine the attractor structure (alternative optima) of such multi-dimensional state spaces. Outside the mathematical arena, our human systems relate strongly to such nonlinear interrelating values. Here we have the further complication of multiple levels, for example, the environment, human physiology, psychology and sociology, where objectives often have inter-level effects as well as intra-level interdependencies. This escalation in complexity is currently beyond our abilities to model computationally in any detail.

Many real world problems generally do not have accurate measurement of its variables and required multiple incommensurable and competing objectives to be met before any solution is considered adequate. By the nature of EA, it can handle this inaccuracy more effectively than any other classical search algorithms. Through EA extension to MOEA, it makes this technique a more suitable tool for user. Not only can it generally provide users the trade-off for each individual problem, it even let users have the capability to evaluate and determine the final suitable solution.

Chapter 3

Trend and Direction of Practical PID Development

Chapter objectives

This chapter analyses and discusses the PID tuning methods found in patents, practical software packages and hardware modules. It aims to identify the trend and direction of practical PID development.

3.1 Introduction

In this chapter, an investigation is conducted on the patented PID tuning rules, practical PID software packages and hardware modules. There are already numerous academic literatures on the survey of PID tuning rules but there exists no literature on the industry side. Thus the aim of this chapter is to provide a more in-depth study and analysis on the trend and direction of practical PID development.

3.2 Patents

3.2.1 Patents Filed

This section focused on the current patented tuning methods that are often adopted in the industries for PID design tools and hardware modules. A range of patents on PID tuning are studied and analysed, they are chronologically listed in Table 3.1. There are altogether 64 such patents filed in the United States, 11 in Japan, 2 in Korea and 2 by the World Intellectual Property Organization. Note that the patent by Yu *et al.* (1994) is not included in the following analysis as it is not available in English.

Table 3.1 Patents on PID Tuning Filed by USPTO, JPO, KPO, and WIPO

Year	Assignee	Identification Method	Tuning Method
1970	International Business Machines Corporation (Armonk, NY) (Dahlin, 1970)	Excitation-based	Formula-based
1973	Phillips Petroleum Company (Bartlesville, Okla.) (Pemberton, 1973)	Excitation-based	Formula-based
1974	The Foxboro Company (Foxboro, MA) (Bristol II, 1974)	Non-Excitation based	Rule-based
1974	Phillips Petroleum Company (Bartlesville, Okla.) (Pemberton, 1974)	Non-Excitation based	Rule-based
1980	K.R. Jones (Liverpool, England) (Barlow and Jones, 1980)	Excitation-based	Optimisation-based
1982	Phillips Petroleum Company (Bartlesville, OK) (Rutledge, 1982)	Excitation-based	Formula-based
1983	Leeds & Northrup Company (North Wales, PA) (Arcara and Anderson, 1983)	Non-Excitation based	Formula-based
1984	Toyo Systems Ltd. (Tokyo, JP) (Hayashibe, 1984)	Excitation-based	Formula-based

Year	Assignee	Identification Method	Tuning Method
1984	Tokyo Shibaura Denki Kabushiki Kaisha (Kawasaki, JP) (Shigemasa, 1984)	Excitation-based	Formula-based
1984	Tokyo Shibaura Denki Kabushiki Kaisha (Kawasaki, JP) (Shigemasa and Takagi, 1984)	Non-Excitation based	Formula-based
1985	Tokyo Shibaura Denki Kabushiki Kaisha (Kawasaki, JP) (Shigemasa and Ichikawa, 1985)	Excitation-based	Formula-based
1985	NAF Controls AB (Solna, SE) (Hägglund and Åström, 1985)	Excitation-based	Formula-based
1986	Tokyo Shibaura Denki Kabushiki Kaisha (Kawasaki, JP) (Mori and Shigemasa, 1986)	Excitation-based	Formula-based
1986, 1990	The Foxboro Company (Foxboro, MA) (Kraus, 1986, 1990)	Non-Excitation based	Rule-based
1987	Eurotherm Corporation (Reston, VA) (Pettit and Carr, 1987)	Excitation-based	Formula-based
1988	Yamatake-Honeywell Co. Ltd. (Tokyo, JP) (Suzuki, 1988)	Excitation-based	Formula-based
1988	Hightech Network AB (Malmo, SE) (Åström and Hägglund, 1988)	Excitation-based	Formula-based
1988	The Babcock & Wilcox Company (New Orleans, LA) (Lane and Scheib, 1988)	Non-Excitation based	Formula-based
1989	Fischer & Porter Company (Warminster, PA) (Fukumoto, 1989)	Non-Excitation based	Formula-based
1989	Yamatake-Honeywell Company Limited (Tokyo, JP) (Murate and Suzuki, 1989)	Excitation-based	Formula-based
1989	Mitsubishi Denki Kabushiki Kaisha (Tokyo, JP) (Nomoto *et al.*, 1989)	Non-Excitation based	Rule-based
1989	Yokogawa Electric Corporation (Tokyo, JP) (Sakai *et al.*, 1989)	Non-Excitation based	Formula-based
1989	Kabushiki Kaisha Toshiba (Kawasaki, JP) (Iino and Shigemasa, 1989)	Excitation-based	Formula-based
1990	Hitachi Ltd. (Tokyo, JP) (Saito *et al.*, 1990)	Non-Excitation based	Rule-based
1991	Hitachi Ltd. (Tokyo, JP) (Takahashi *et al.*, 1991)	Non-Excitation based	Rule-based
1992	Charles A. White III (Stamford CT) (White III, 1992)	Non-Excitation based	Others (self-learning memory unit)
1992	Hitachi Ltd. (Tokyo, JP) (Saito *et al.*, 1992)	Non-Excitation based	Rule-based
1992	Rockwell International Corporation (Seal Beach, CA) (Chand, 1992)	Non-Excitation based	Rule-based
1992	Yokogawa Electric Corporation (Tokyo, JP) (Takatsu and Kitano, 1992)	Excitation-based	Formula-based
1992	Honeywell Inc. (Minneapolis, MN) (Sklaroff, 1992)	Excitation-based	Formula-based

Year	Assignee	Identification Method	Tuning Method
1993	Allen-Bradley Company Inc. (Milwaukee, WI) (Svarovsky *et al.*, 1993)	Excitation-based	Formula-based
1993	Industrial Technology Research Institute (Chutung, TW) (Chu *et al.*, 1993)	Excitation-based	Formula-based
1993	Hitachi Ltd. (Tokyo, JP) (Miyagaki *et al.*, 1993)	Non-Excitation based	Formula-based
1993	Nippon Denki Garasu Kabushiki Kaisha (Shiga, JP) (Aoki, 1993)	Non-Excitation based	Rule-based
1994	Fisher-Rosemount Systems, Inc. (Austin, TX) (Lloyd, 1994)	Excitation-based	Formula-based
1994	Sanyo Electric Co. Ltd. (Osaka, JP) (Katayama and Kajitani, 1994)	Non-Excitation based	Rule-based
1994	Hitachi Ltd. (Tokyo, JP) (Nomura *et al.*, 1994)	Non-Excitation based	Others (neural network)
1994	Omron Corporation (Kyoto, JP) (Ueda *et al.*, 1994)	Excitation-based	Formula-based
1994	Universal Dynamics Limited (CA) (Gough Jr. and Lyon, 1994)	Non-Excitation based	Formula-based
1994	Johnson Service Company (Milwaukee, WI) (Seem and Haugstad, 1994)	Non-Excitation based	Formula-based
1995	The Foxboro Company (Foxboro, MA) (Hansen, 1995a)	Excitation-based	Formula-based
1995	The Foxboro Company (Foxboro, MA) (Hansen, 1995b)	Non-Excitation based	Rule-based
1995	Fisher Controls International, Inc. (Clayton, MO) (Wojsznis and Blevins, 1995)	Excitation-based	Formula-based
1996	Kabushiki Kaisha Toshiba (Kawasaki, JP) (Hiroi, 1996)	Excitation-based	Formula-based
1996	Johnson Service Company (Milwaukee, WI) (Seem and Decious, 1996)	Excitation-based	Formula-based
1996	The Foxboro Company (Foxboro, MA) (Hansen and Bristol, 1996)	Non-Excitation based	Rule-based
1997	A.K. Mathur and T. Samad (Minneapolis, MN) (Mathur and Samad, 1997)	Excitation-based	Others (neural network)
1997	Motorola Inc. (Schaumburg, IL) (Teng and Wang, 1997)	Non-Excitation based	Optimisation-based
1997	Fanuc Ltd. (Yamanashi, JP) (Kato *et al.*, 1997)	Non-Excitation based	Formula-based
1997	Rosemount Inc. (Eden Prairie, MN) (Zou *et al.*, 1997)	Excitation-based	Formula-based
1998	National Science Council (Taipei, TW) (Yu, 1998)	Excitation-based	Formula-based
1998	Hartmann & Braun A.G. (Frankfurt, DE) (Bunzemeier, 1998)	Non-Excitation based	Formula-based
1998	Motorola Inc. (Schaumburg, IL) (Teng and Wang, 1998)	Non-Excitation based	Optimisation-based
1998	Rosemount Inc. (Eden Prairie, MN) (Zou and Brigham, 1998)	Excitation-based	Formula-based

Year	Assignee	Identification Method	Tuning Method
1998	Honeywell Inc. (Minneapolis, MN) (Samad, 1998)	Non-Excitation based	Others (neural network)
1999	Samsung Electronics Co. Ltd. (Seoul, KR) (Kim, 1999)	Non-Excitation based	Others (genetic algorithm)
1999	Ralph E. Rose (San Jose, CA) (Rose, 1999)	Non-Excitation based	Optimisation-based
2000	National University of Singapore (SG) (Wang and Hang, 2000)	Excitation-based	Formula-based
2000	National Instruments Corporation (Austin, TX) (Luo *et al.*, 2000)	Excitation-based	Formula-based
2000	Fisher Controls International Inc. (Clayton, MO) (Junk, 2000)	Excitation-based	Optimisation-based
2001	Honeywell International Inc. (Morristown, NJ) (Lu, 2001)	Excitation-based	Optimisation-based
2002	Siemens Aktiengesellschaft (Munich, DE) (Weinzierl, 2002)	Excitation-based	Others (neural network)
2002	National University of Singapore (SG) (Wang *et al.*, 2002)	Excitation-based	Formula-based
1984	Fuji Denki Seizo KK (JP) (Takigawa *et al.*, 1984a)	Excitation-based	Formula-based
1984	Fuji Denki Seizo KK (JP) (Takigawa *et al.*, 1984b)	Excitation-based	Formula-based
1991	Yokogawa Electric Corp (JP) (Yamamoto, 1991)	Non-Excitation based	Others (neural network)
1991	Yokogawa Electric Corp (JP) (Otani, 1991)	Non-Excitation based	Others (Auto Regressive Moving Average (ARMA) model with neural network)
1992	Sanyo Electric Co. Ltd. (JP) (Katayama, 1992)	Non-Excitation based	Rule-based
1992	Hitachi Ltd (JP) (Tadokoro *et al.*, 1992)	Excitation-based	Formula-based
1993	Hitachi Ltd (JP) (Tadokoro *et al.*, 1993)	Excitation-based	Formula-based
1994	Hitachi Ltd (JP) (Kobari *et al.*, 1994)	Excitation-based	Formula-based
1995	Matsushita Electric Works Ltd (JP) (Mitsuo, 1995)	Excitation-based	Formula-based
1998	Toshiba Corp (JP) (Hagiwara, 1998)	Non-Excitation based	Formula-based
1999	Yaskawa Electric Corp (JP) (Takeguchi, 1999)	Non-Excitation based	Rule-based
1994	Korea Electronics Telecomm (KR) (Yu *et al.*, 1994)	—	—
1997	Samsung Aerospace Ltd. (KR) (Kim, 1997)	Excitation-based	Rule-based
1998	The University of Newcastle Research Associates Limited (AU) (Goodwin and Crisafulli, 1998)	Excitation-based	Formula-based
2001	Fisher Rosemount Systems, Inc. (US) (Wojsznis and Blevins, 2001)	Non-Excitation based	Formula-based

3.2.2 Identification Methods for Tuning

Most of the tuning methods patented rely on an identification of plant dynamics, using an excitation or non-excitation type of method. The excitation type can be break down further into time- or frequency-domain methods.

Excitation is often used during plant set-up and commissioning in order to set initial PID parameters. Time-domain excitations are usually a step or Pseudo Random Binary Signal (PRBS) applied in an open-loop fashion. This is a classical and the most widely practised method. It is often adopted for model-based tuning methods. Frequency-domain excitations usually use a relay-like method, where the plant will undergo a controlled self-oscillation. This type of identification does not normally require a parametric model in tuning a PID controller, which is the main advantage over time-domain based identification. However, it does not provide insight into which process or control system characteristics could be modified to improve the feedback controller performance (Corripio, 2001).

Generally, non-excitation type of identification is preferred by industry due to safety reasons, particularly during normal operations, as this does not upset the plant. An increasing number of patents are now filed on non-excitation identification, as seen in Figure 3.1.

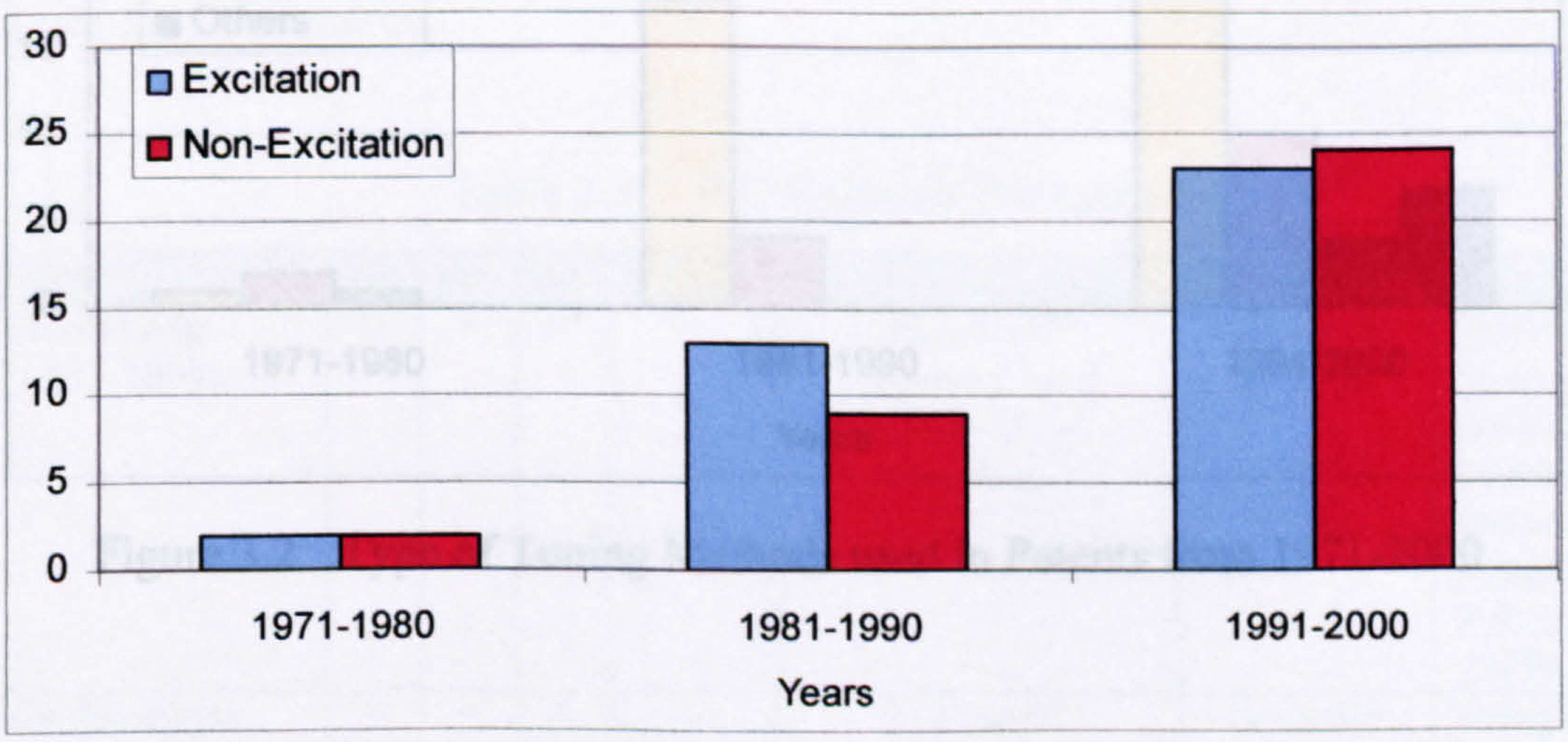

Figure 3.1 Type of Identification used in Patents from 1971-2000

3.2.3 Tuning Methods Patented

Most of the identification and tuning methods patented are process engineering oriented and appear rather ad hoc. Shown in Table 3.1, patented tuning methods are mostly formula-based, rule-based and optimisation-based. Formula-based methods first identified the characteristics of the plant and then perform a mapping (similar to the ZN method). These are often used in on-demand tuning for responsiveness. Rule-based methods are often used in adaptive control, but can be quite complex and ad hoc. These can be expert systems, including simple heuristics and fuzzy logic rules. Optimisation-based methods are often applied off-line or on very slow processes, using a conventional (such as least mean squares) or an unconventional (such as genetic algorithms) search method.

Figure 3.2 shows that formula-based tuning methods are still the most actively developed, whilst other methods have also received an increased attention. However, most do not yield global or multi-objective optimal performance and their applicability are hence often limited.

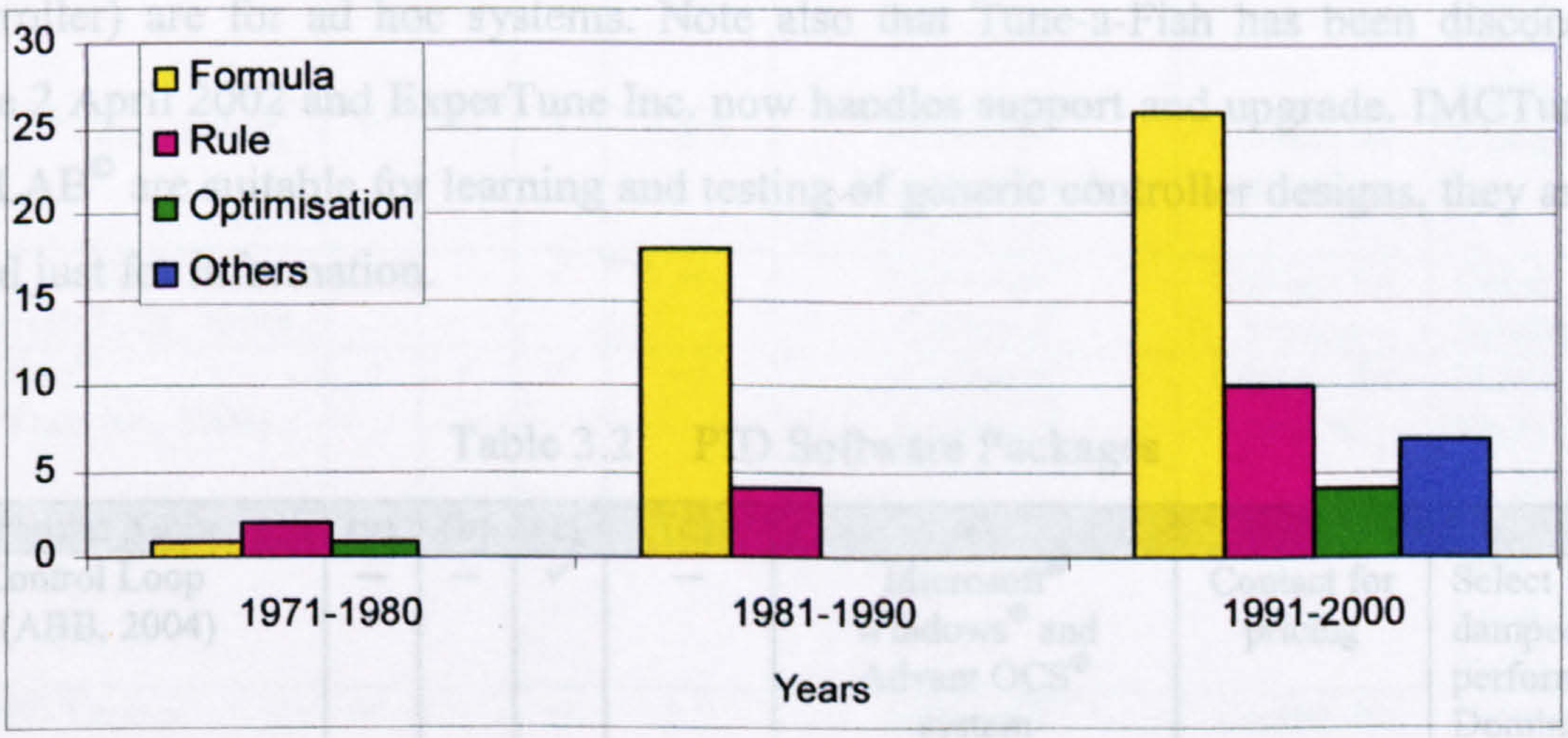

Figure 3.2 Type of Tuning Methods used in Patents from 1971-2000

3.3 PID Software Packages

3.3.1 Software Packages

Due to the lack of a simple and widely applicable tuning method, a need for the development of easy to use PID tuning software has therefore arisen. This allows a practitioner with some control knowledge or plant information to be able to tune a PID controller efficiently and optimally for various applications. It is hoped that such software tools will increase the practising company's system performance and hence production quality and efficiency without needing to invest a vast amount of time and manpower in testing and adjusting control loops.

Table 3.2 analyses and summarises currently available practical PID software packages, grouped by the methods of their tuning engines whenever known. Note that AdvaControl Loop Tuner (Advant OCS® system), DeltaV Tune (DeltaV workstation), Intelligent Tuner (Fisher-Rosemount PROVOX® controller), OvationTune (Westinghouse DCS), Profit PID (Honeywell TPS/TDC system), PID Self-Tuner (Siemens SIMATIC® S7/C7) and Tune-a-Fish (Fisher-Rosemount PROVOX® controller) are for ad hoc systems. Note also that Tune-a-Fish has been discontinued since 2 April 2002 and ExperTune Inc. now handles support and upgrade. IMCTune and CtrlLAB© are suitable for learning and testing of generic controller designs, they are also listed just for information.

Table 3.2 PID Software Packages

	Product Name	(a)	(b)	(c)	(d)	(e)	(f)	Notes
Analytical Methods	AdvaControl Loop Tuner (ABB, 2004)	—	—	✓	—	Microsoft® Windows® and Advant OCS® system	Contact for pricing	Select fast, normal or damped closed-loop performance using Dominant Pole Placement method extended with Robustness Criteria (DPPM-RC)
	IMCTune (Brosilow, 2002)	×	×	×	—	Microsoft® Windows® and MATLAB®	Freeware	Using IMC tuning
	Model ID & PID Tuning Software (Control Arts Inc, 2004)	✓	✓	—	3.5	Microsoft® Windows®	US$ 699 for single user license	Using IMC tuning

	Product Name	(a)	(b)	(c)	(d)	(e)	(f)	Notes
	Robust PID Tuning (Control & Optimization Specialists, 2004)	?	—	×	—	Microsoft® Windows®	Contact for pricing	Select modified IMC/Lambda tuning or ratio of closed-loop to open-loop response time for non-integral process and closed-loop response time for integral process
	INTUNE™ (Control Soft Inc, 2004)	✓	✓	✓	4.12	Microsoft® Windows®	Contact for pricing	Using advanced IMC based tuning
	Control Station® (Cooper, 2004)	✓	×	×	3.0.1	Microsoft® Windows®	US$ 895 per year for single user yearly maintenance license	Select regulating or tracking performance using Lambda tuning correlations
	DeltaV Tune (DeltaV, 2004)	✓	—	✓	5.1	DeltaV workstation and DeltaV controller running control software	Contact for pricing	Select performance ranging from no overshoot to very aggressive using either modified ZN rules for PI, phase and gain margin rules for PID, Lambda tuning rules for PI, Lambda-Averaging Level for PI, Lambda-Smith Predictor or IMC tuning rule
	pIDtune™ (EngineSoft, 2003)	✓	—	×	1.0.5	Microsoft® Windows® and MATLAB®	Contact for pricing	Using IMC tuning
	EnTech Toolkit Tuner Module (EnTech, 2004)	✓	—	✓	—	Microsoft® Windows®	Contact for pricing	Using advanced Lambda tuning
	ExperTune® (ExperTune Inc, 2004)	✓	✓	✓	—	Microsoft® Windows®	Contact for pricing	Select regulating or tracking performance, quarter amplitude damping, 10% overshoot and Lambda (standard or level)
	Easy PID Tuning© (Ingénierie Pour SIgnaux et Systèmes, 2004)	✓	—	—	2.0	Microsoft® Windows® and MATLAB®	Contact for pricing	Using pole placement method
	Tune Plus (Innovention Industries Inc, 2004)	✓	—	✓	—	Microsoft® Windows®	Contact for pricing	Using Lambda/IMC tuning
	Control Loop Assistant (Lambda Controls, 2004)	✓	×	×	1.0c	Microsoft® Windows®	Contact for pricing	Using Lambda tuning
	EZYtune™ (Matrikon Inc, 2004)	✓	✓	×	1.1.02	Microsoft® Windows®	US$ 199 per copy	Select performance based on closed-loop time constant and 10%-90% rise time

	Product Name	(a)	(b)	(c)	(d)	(e)	(f)	Notes
	TuneUp (Metso Automation Inc, 2004)	✓	–	✓	–	Microsoft® Windows® and MATLAB®(optional)	Contact for pricing	Using Optimisation/Lambda tuning
	TuneWizard™ (Plant Automation Services Inc, 2004)	✓	✓	✓	2.5.2	Microsoft® Windows®	Contact for pricing	Select either regulating or tracking performance or IMC (Lambda) tuning or surge tank application
	RSTune™ (Rockwell Automation, 2004)	✓	✓	✓	–	Microsoft® Windows® and Allen-Bradley® PLC-5®, SLC 500™ or ControlLogix PLCs	Contact for pricing	Using ExperTune®
	ProTuner™ 32 (Techmation Inc, 2004)	✓	×	✓	6.04.01	Microsoft® Windows®	Contact for pricing	Select fast, medium or slow response to either regulating or tracking performance using pole cancellation with gain and phase margin and closed loop damping factor
	Tune-a-Fish (TiPS Inc, 2002)	✓	✓	✓	–	Microsoft® Windows® and Fisher-Rosemount PROVOX® Controllers	Contact for pricing	Using ExperTune®
Optimisation Methods	GRAPHIDOR© (Communications & Systems, 2004)	✓	×	×	–	Microsoft® Windows®	Contact for pricing	Generate 3-D plot using P, I and error with objective to search for minimum error
Optimisation Methods	Profit PID (Honeywell International Inc, 2004)	✓	–	✓	–	Honeywell TPS/TDC	Contact for pricing	Using proprietary min-max algorithm
Optimisation Methods	PIDeasy™ (Li *et al.*, 1998)	✓	×	✓	1.0	Microsoft® Windows®	Contact for pricing	Using proprietary algorithm
Optimisation Methods	Simple Analytical Tuning of Digital PI/PID Control for Fluid & Motion Systems (SpecializedControl, 2002)	✓	✓	–	–	Microsoft® Windows®	Contact for pricing	Using proprietary algorithm
Optimisation Methods	VisSim/OptimizePRO™ (Visual Solutions Inc, 2004)	–	–	✓	4.0	Microsoft® Windows® and Professional VisSim 4.0	Contact for pricing	Using generalised, reduced gradient algorithm (GRG2)
Unknown Methods	TOPAS (ACT, 2004)	✓	✓	×	1.2	Microsoft® Windows®	€2000 for single user	Select regulating or tracking performance and tight and average level control
Unknown Methods	WinREG-PID (ADAPTECH, 2004)	✓	✓	✓	–	Microsoft® Windows® and WinREG	Contact for pricing	–

Product Name	(a)	(b)	(c)	(d)	(e)	(f)	Notes
SimAxiom™ (Off-line tuning) (Algosys Inc, 2004)	✓	✓	×	–	Microsoft® Windows®	Contact for pricing	Select desired closed-loop response time
DynAxiom™ (On-line tuning) (Algosys Inc, 2004)	✓	?	✓	–	–	Contact for pricing	–
PITOPS™ (Artcon Inc, 2004)	✓	✓	×	–	Microsoft® Windows®	Contact for pricing	Select regulating or tracking performance
BESTune (BESTune, 2004)	✓	✓	×	4.4	Microsoft® Windows® and MATLAB®	US$ 500 per copy	Select controller tightness
CADET V12 (CP Engineering Systems Ltd, 2004)	–	–	✓	–	Microsoft® Windows®	Contact for pricing	–
Universal Process Identification for Advanced Process Control (UPID™) (Cutler Johnston Corporation, 2004)	✓	–	–	–	Microsoft® Windows®	Contact for pricing	–
PEWIN Pro (Delta Tau Data Systems Inc, 2004)	✓	–	✓	2.0	Microsoft® Windows®	Contact for pricing	–
RaPID (IPCOS, 2004)	✓	✓	✓	1.2	Microsoft® Windows® and MATLAB®	€3300 for single user	Select regulating or tracking performance or both
Commander Supervisory Software (ISE Inc, 2004)	–	✓	✓	4.1.41	Microsoft® Windows®	Contact for pricing	–
Control System Tuning Package (CSTP) (Israel Electric Corporation, 2004)	✓	–	–	3.0	Microsoft® Windows® and MATLAB®	Contact for pricing	–
JC Systems Toolbox (JC Systems Inc, 2004)	–	–	–	–	Microsoft® Windows® and LabVIEW™	US$ 495 per copy	–
LabVIEW™ PID Control Toolset for Windows (National Instruments, 2004)	–	–	✓	–	Microsoft® Windows® and LabVIEW™	Contact for pricing	–
Intelligent Tuner (PROVOX, 2004)	✓	–	✓	–	DEC OpenVMS VAX or OpenVMS AXP series and OpenVMS version 6.1 or later operating software; PROVOX® 10-series, 20-series, 20-series SR90 controllers, or SRx controllers	Contact for pricing	–
PIDS™ (Raczynski, 2004)	×	×	×	–	Microsoft® Windows®	US$18 per copy	Select performance based on ITAE, ITSE, ISE or IAE

Product Name	(a)	(b)	(c)	(d)	(e)	(f)	Notes
PID Self-Tuner (Siemens, 2004)	—	—	✓	5.0	Microsoft® Windows® and S7-300/400 station; STEP 7 (≥ V3.2) and Standard PID Control V5 installed on programming device	Contact for pricing	—
Controller Tuning 101 (Straight-Line Control Co Inc, 2004)	✓	×	×	3.0	Microsoft® Windows®	US$ 11 - Base price	—
OvationTune (Westinghouse Process Control, 2004)	—	—	✓	—	Westinghouse Process Control DCS	Contact for pricing	—
GeneX (Xiera Technologies Inc, 2004)	—	—	—	2.0	Microsoft® Windows® and MATLAB®	Contact for pricing	—
CtrlLAB© (Xue, 2004)	×	×	×	3.0	Microsoft® Windows® and MATLAB®	Freeware	Select performance based on ISE, ISTE, IST^2E or Gain/Phase margins

Remarks:

(a) Model-based tuning. Indicate software that matches the open/closed loop plant response data to a specific model.

(b) Support vendor specific PID structures. Indicate software that explicitly supports vendor specific PID structures and not those that support the generic PID structures.

(c) Support online operation. Indicate software that supports online operation such as sampling of data, online tuning etc.

(d) Software version reviewed.

(e) Operating Systems and Hardware/Software Dependent.

(f) Prices. Please contact the manufacturer for updated prices on their products.

Legend:

✓ Support

× Does not support

? Probably support

— Information not available

3.3.2 Tuning Methods Adopted

Within the 'Analytical Methods' group, it is seen from the 'Notes' column that the Internal Model Control (IMC) or Lambda Tuning method is the most widely adopted tuning method in practical software packages. Almost all these packages require a time-domain model before the controller parameters can be set. The adopted model is the one given by (2.11). pIDtune™ by EngineSoft® is the only one that uses an ARX (Auto Regressive with eXternal input) model instead of the model given by (2.11). On design, 'Type C' (or I-PD) structure is strongly recommended in BESTune (BESTune, 2004). Note that ExperTune® is embedded in RSTune™ and Tune-a-Fish.

It is almost impossible to name a software package to be the best as there is no generic method to set the PID controller optimally to satisfy all design criteria and needs. However, most of the software packages studied in Table 3.2 provide a tuneable parameter set for the user to determine an overall performance that is best suited to an ad hoc application.

3.3.3 Operating Systems and Online Operation

Based on the information summarised in Table 3.2, Microsoft Windows® is currently the most supported platform. Meanwhile, MATLAB® is a popular software environment used in off-line analysis.

Quite a few software packages in Table 3.2 do not support online operations, such as, real-time sampling of data, on-line tuning, etc. The common non-vendor specific interfaces supported for on-line operations are Microsoft Windows® Dynamic Data Exchange (DDE) and OLE for Process Control (OPC®) (OPC Foundation, 2004) based on Microsoft Object Linking and Embedding (OLE), Component Object Model (COM) and Distributed Component Object Model (DCOM) technologies.

OPC® is an industry standard created with the collaboration of a number of leading worldwide automation and hardware/software suppliers working in cooperation with Microsoft Inc. The standard defines a method for exchanging real-time automation data among PC-based clients using Microsoft operating systems. Thus the aim of OPC® is to realise possible interoperability between automation and control applications, field systems and devices, and business and office applications. There are currently hundreds of OPC Data Access servers and clients available.

3.3.4 Modern Features

Remedial features such as differentiator filtering and integrator anti-windup are now mostly accommodated in a PID software package. Now the trend is to provide some additional features, such as diagnostic analysis, which proves to be very helpful in practice. An example is highlighted by ExperTune®, which includes a wide range of fault diagnosis features, such as valve wear analysis, robustness analysis, automatic loop report generation, multi-variable loop analysis, power spectral density plot, auto and cross-correlations plot, and shrink-swell (inverse response) process optimisation, etc. Other additional features seen in practical PID packages include user-friendly interfaces, support of a variety of controller structures and allowing more user-defined settings in determining PID parameters when necessary.

3.4 PID Hardware Modules

3.4.1 Hardware and Tuning

Many PID software features are now incorporated in hardware modules, particularly those used in process control. A range of these is available from the four dominant vendors, namely, ABB, Foxboro, Honeywell and Yokogawa, as listed in Table 3.3. Hardware brands from Elsag Bailey, Kent-Taylor Instruments, Hartmann & Braun and Alfa Laval have been acquired by ABB. The following brands have been acquired under Emerson Process Management Group, namely, Brooks Instrument, Daniel®, DeltaV, Fisher®, Intellution®, Micro Motion®, PROVOX®, Rosemount®, RS3 and Westinghouse Process Control. Invensys Production Management Division consists of APV, Avantis, Esscor, Eurotherm, Foxboro, Pacific Simulation, Triconex and Wonderware. Readers may refer to Versteeg *et al.* (1986), Minter and Fisher (1988), Cao and McAvoy (1990), Hägglund and Åström (1991), Hang and Sin (1991), Åström *et al.* (1993) and Åström and Hägglund (1995) for more information on commercial PID controllers.

Based on a survey carried out by Control Engineering (1998), single-loop models account for 64% of the controllers, while multi-loop models constitute 36%. It also reveals that 85% of the loop controllers are used for feedback control, 6% for feedforward control and 9% for cascade control. The most important features that are expected from a loop controller are, in order of importance, PID function, start-up self-tuning, online self-tuning, adaptive control and fuzzy logic.

Many PID controller manufacturers provide various facilities in their products that allow easy tuning of the controller. As seen in PID patents and software packages, most of the hardware systems also adopt a time-domain tuning method, whilst a minority relies on open-loop relay experiments. Some modules offer gain-scheduling capabilities and hence can cover a large operation envelope. Some are more adaptive, using online model identification or rules inferred from on-line responses.

Automated tuning is mainly implemented through either 'tuning on demand' with upset or 'adaptive tuning'. Some manufacturers refer 'tuning on demand' with upset as 'self-tune', 'auto-tune' or 'pre-tune', whilst 'adaptive tuning' is sometimes known as 'self-tune', 'auto-tune' or 'adaptive tune'. There exists no standardisation in the terminology.

'Tuning on demand' with upset typically determines the PID parameters by inducing a controlled upset in the process. This allows measurements of the process response so as to calculate the appropriate controller parameters. 'Adaptive tuning' aims to set the PID parameters without inducing upsets. When a controller is utilising this function, it constantly monitors the process variable for any oscillation around the set-point and hence closed-loop identification can be as effective as in 'tuning on demand'. This type of tuning is ideal for processes where load characteristics change drastically while the process is running. If there is any oscillation, the controller adjusts the PID parameters in an attempt to eliminate them. It cannot be used effectively, however, if the process has externally induced upsets for which the control could not possibly tune out.

Table 3.3 Commercial PID Controller Hardware Modules

Manufacturer	Product Model	(a)	(b)	(c)	(d)	(e)	Description
ABB	Bitric P	✓	×	×	×	2000	Compact Single Loop Controller
	Digitric 100	✓	×	×	×	2001	Versatile Single Loop Controller
	COMMANDER 100	✓	×	×	×	1999	1/8 DIN Universal Process Controller
	COMMANDER 250	✓	×	×	×	1999	1/4 DIN Compact Process Controller
	COMMANDER 310	✓	×	×	×	1999	Wall/Pipe-mount Universal Process Controller
	COMMANDER 351	✓	✓	×	×	2001	1/4 DIN Universal Process Controller
	COMMANDER 355	✓	✓	×	✓	2001	1/4 DIN Advanced Process Controller
	COMMANDER 505	✓	✓	×	✓	2000	6x3 format Advanced Process Controller
	COMMANDER V100	×	×	×	×	1999	1/8 DIN Motorized Valve Controller
	COMMANDER V250	×	×	×	×	1998	1/4 DIN Motorized Valve Controller

Manufacturer	Product Model	(a)	(b)	(c)	(d)	(e)	Description
	ECA06	✓	×	×	×	2000	ECA Series – General Purpose Process Controller
	ECA60	✓	✓	×	✓	2000	ECA Series – General Purpose Process Controller
	ECA600	✓	✓	✓	✓	2000	ECA Series – General Purpose Process Controller
	MODCELL™ 2050R	✓	×	×	×	2001	Single Loop Controller
	53SL6000	✓	×	×	×	2001	Micro-DCI™ Instrumentation Single Loop Controller
Foxboro	716C	✓	×	✓	×	1996	1/16 DIN Temperature Controller
	718PL, 718PR	✓	×	✓	×	1996	1/8 DIN Process Controller with Local Set Point (PL) and Remote Set Point (PR)
	718TC, 718TS	✓	×	✓	×	1996	1/8 DIN Temperature Controller with mA Output (TC) and Servo Output (TS)
	731C	✓	×	✓	×	1996	1/4 DIN Digital Process Controller
	743C	✓	×	✓	×	1994	Field Station MICRO® Controller
	760C	✓	×	✓	×	1985	Single Station MICRO® Controller
	761C	✓	×	✓	×	1987	Single Station MICRO® Plus Controller
	762C	✓	×	✓	×	1996	Single Station MICRO® Controller
	T630C	✓	×	✓	×	2000	Process Controller
Honeywell	UDC100	×	×	×	×	1999	1/4 DIN Universal Digital Temperature Controller
	UDC700	✓	×	✓	×	1996	1/32 DIN Universal Digital Controller and Indicator
	UDC900	✓	×	✓	×	1997	1/16 DIN Universal Digital Temperature Controller
	UDC1000, UDC1500	✓	×	✓	×	2001	Micro-Pro Series – Universal Digital Controllers
	UDC2300	✓	×	✓	×	1999	1/4 DIN Universal Digital Controller
	UDC3300	✓	✓	✓	×	1999	1/4 DIN Universal Digital Controller
	UDC5000	✓	×	✓	×	1994	Ultra-Pro Universal Digital Controller
	UDC6300	✓	✓	✓	✓	1997	Stand-Alone Process Controller and Process Indicator
Yokogawa	US1000	✓	✓	×	✓	1998	Process Controllers
	UT320, UT350, UT420, UT450, UT520, UT550, UT750	✓	×	×	×	2000	Enhanced Green Series Temperature Controllers
	UP350, UP550, UP750	✓	×	×	×	2000	Enhanced Green Series Programmable Controllers
	YS150	✓	×	✓	✓	1991	High-Level Process Controllers

Manufacturer	Product Model	(a)	(b)	(c)	(d)	(e)	Description
	YS170	✓	✓	✓	✓	1991	High-Level Process Controllers

Remarks:

(a) On-Demand Auto Tune

(b) Gain-Scheduling

(c) Adaptive Control

(d) Feedforward Control

(e) Year of release

Legend:

✓ Support

× Does not support

3.4.2 ABB Controllers

ABB controllers offer two auto-tuning options, namely, quarter-wave and minimal overshoot. They also come with a manual fine-tuning option called Control Efficiency Monitor (CEM). As shown in Figure 3.3, six 'key-performance' parameters labelled are measured and displayed, allowing the user to vary the PID settings to match the process needs and to fine-tune manually.

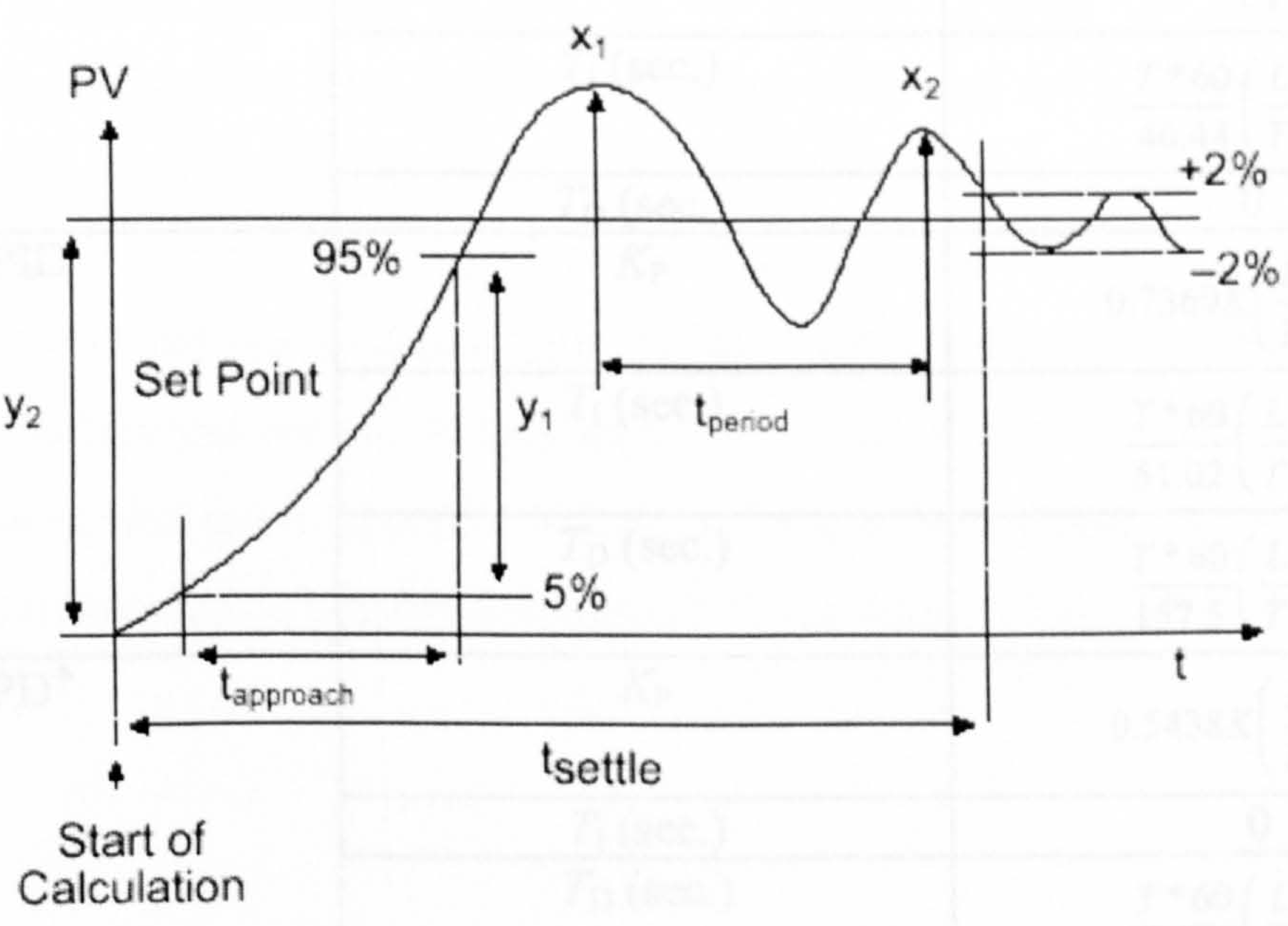

Figure 3.3 ABB – Control Efficiency Monitor (CEM) Measurements (ABB, 2001b)

ABB also offers another tuning algorithm for its Micro-DCI™ series, the Easy-Tune™. The Easy-Tune™ algorithm approximates a process by a first-order plus delay model, as shown in (2.11). It uses a typical graphical method, where the step changes are applied so as to measure the gain, delay and rise-time and hence the time-constant. These are then used to map the controller parameters through formulae shown in Table 3.4 (ABB, 2001a), which are optimised for the integral of time-weighted absolute error (ITAE) performance index.

It is unclear, unfortunately, whether the three plant parameters are continuously identified so as to vary the PID parameters. If they are, however, Micro-DCI™ series should be very powerful in dealing with changing plant dynamics through continuously scheduled optimal PID settings.

Table 3.4 ABB – ITAE Equations

Mode	Action	Equation
P	K_P	$2.04K\left(\frac{L}{T}\right)^{1.084}$
	T_I (sec.)	0
	T_D (sec.)	0
PI	K_P	$1.164K\left(\frac{L}{T}\right)^{0.977}$
	T_I (sec.)	$\frac{T*60}{40.44}\left(\frac{L}{T}\right)^{0.68}$
	T_D (sec.)	0
PID	K_P	$0.7369K\left(\frac{L}{T}\right)^{0.947}$
	T_I (sec.)	$\frac{T*60}{51.02}\left(\frac{L}{T}\right)^{0.738}$
	T_D (sec.)	$\frac{T*60}{157.5}\left(\frac{L}{T}\right)^{0.995}$
PD*	K_P	$0.5438K\left(\frac{L}{T}\right)^{0.947}$
	T_I (sec.)	0
	T_D (sec.)	$\frac{T*60}{157.5}\left(\frac{L}{T}\right)^{0.995}$

* Empirical estimates not based on ITAE method

3.4.3 Foxboro Series

Foxboro 716C, 718 and 731C series use a proprietary self-tuning algorithm, SMART. During start-up and control, SMART continuously monitors the process variable and automatically adjusts the PID parameters according to the response of the process variable, as shown in Figure 3.4. The advantage of SMART is its ability to operate without injecting any artificial change into the system.

Foxboro 743C, 760C, 761C, 762C and T630C controllers use another patented self-tuning algorithm, Expert Adaptive Controller Tuning (EXACT). EXACT does not use a parametric model, but adjusts the controller based on pattern recognition results of the actual current process. When it senses a process upset, it immediately takes corrective action for the pattern recognition. The user can choose the threshold levels of desired damping and overshoot-to-load changes, as shown in Figure 3.5. EXACT needs to have a good initial PID parameter set to start with, in order to achieve satisfactory performance. Thus the initial PID parameters are determined by introducing a small perturbation to the process and use the resulting process reaction curve to do the calculations. To start up the control system, engineers must determine an anticipated noise-band and maximum wait-time of the process. The noise-band is a value representing expected amplitude of noise on the feedback signal. The maximum wait-time is the maximum time that EXACT algorithm will wait for a second peak in the feedback signal after detecting a first peak. These two settings are crucial in order for the EXACT algorithm to have optimal performance but can be quite tricky to determine.

All Foxboro's controllers studied here are rule-based, instead of model-based but do not support feedforward control. If they support gain scheduling, however, they will be very effective for the entire operating envelope, as gain-scheduling can be more useful than continuous adaptation in most situations (Åström *et al.*, 1993).

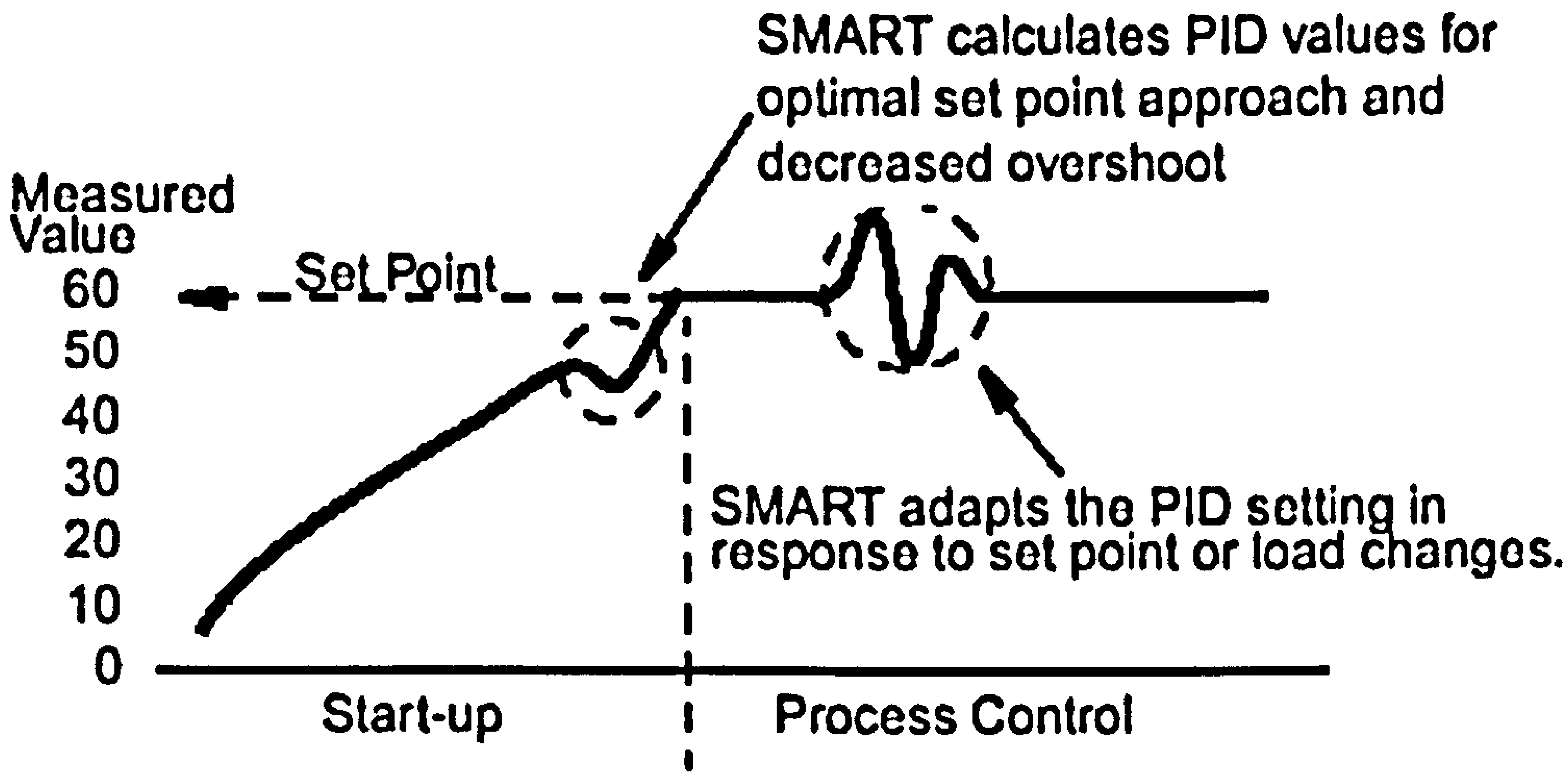

Figure 3.4 Foxboro – SMART Adaptive Self-Tuning (Foxboro, 1996)

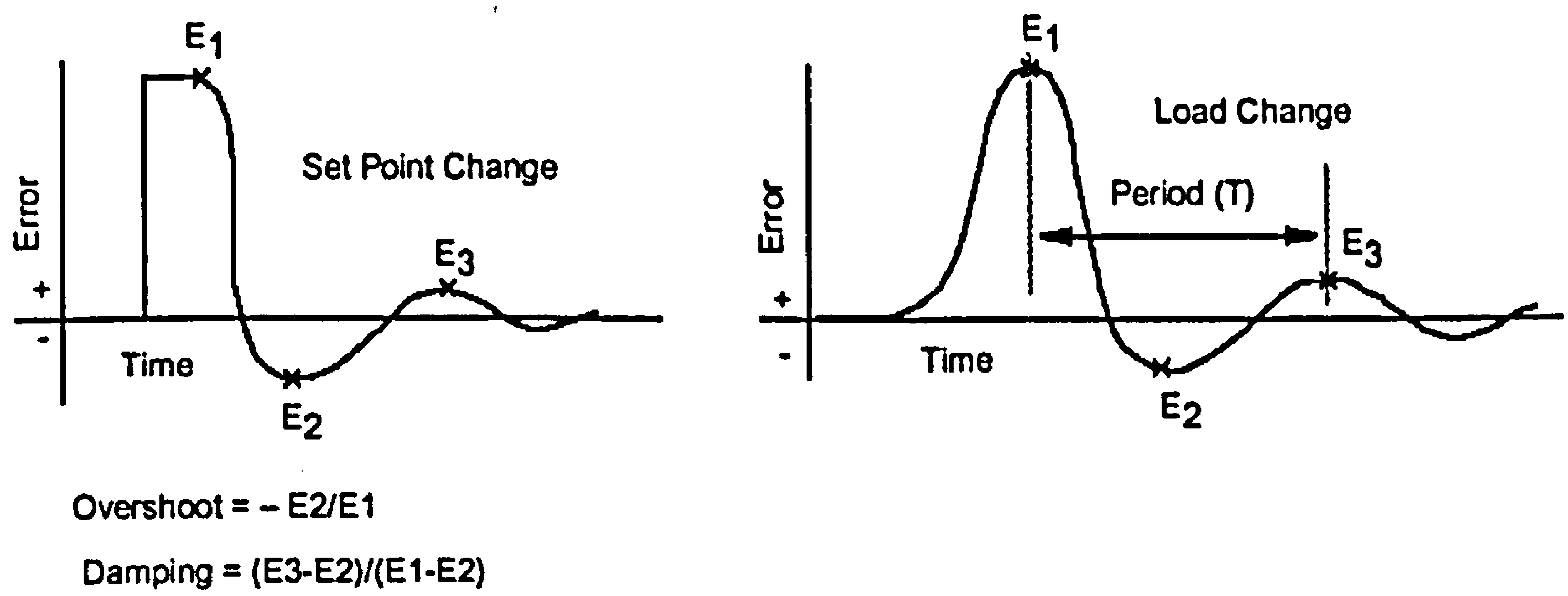

Figure 3.5 Foxboro – Pattern Recognition Characteristics (Foxboro, 1995)

3.4.4 Honeywell Tuners

Honeywell offers a 'tuning on demand' controller, Autotune™, which is not adaptive or continuous. They also offer an adaptive tuner, Accutune™, which uses a combination of frequency and time response analysis plus rule-based expert system techniques to identify the process continually. An enhanced version of this is, Accutune II™, which incorporates a fuzzy logic overshoot suppression mechanism. It provides a 'plug-and-play' tuning algorithm, which will starts at the touch of a button or through an input response data set to identify and tune for any processes including integrating processes and those with a dead-time. This speeds up and simplifies the start-up process and allows retuning at any set-point in an 'automatic mode'. The fuzzy logic overshoot suppression function operates independently from Accutune™ tuning as an add-on. It does not change the PID parameters, but temporarily modifies the control action to suppress overshoot. Although this makes the control system more complex and difficult to analyse, it allows more aggressive action to co-exist with smooth process output. It can be disabled, depending on the application or user requirements, and should be unnecessary if the PID controller is set adaptively optimally.

3.4.5 Yokogawa Modules

Yokogawa first introduced its SUPER CONTROL™ module over a decade ago. Similar to Honeywell's Accutune II™, it also uses a fuzzy logic based algorithm to eliminate overshoots, mimicking control expertise of an experienced operator. It consists of two main parts, namely, the set-point modifier and the set-point selector.

The set-point modifier models the process and functions as an 'expert operator' by first considering that a PID controller is difficult to tune to deliver both a short rise-time and a low over-shoot. It thus seeks a knowledge base about the process, its dynamics, and any nonlinearity of the process (including load changes). Then it leads the system into performing perfectly by feeding artificial target set-points into the PID block through the set-point selector.

In particular, SUPER CONTROL™ operates on three modes. Mode 1 is designed for overshoot suppression by observing the rate of change when the process output approaches a new target set-point. It installs 'sub set-points' as the process output approaches set-point to insure overshoot does not occur. Mode 2 is for ensuring high stability at the set-point while sacrificing some response time to a set-point change. Mode 3 is for a faster response than Mode 2 to a set-point or load change with some compromise in stability when a new set-point is entered and as the process output approaches that change. The process block is simply the first-order lag time with gain model and it simulates the process variable, PV, without any inherent dead time. A functional block diagram for Mode 2 and 3 is shown in Figure 3.6. If Mode 2 or 3 observe any phase shift that has changed from normal operating conditions, it uses the process model to compute a calculated process variable, CPV, and attempts to suppress PV from hunting. The compensation model switches between the measured PV and CPV while the control function block performs the normal PID computation. It is unclear how the three modes are switched between, but it would be advantageous if this is scheduled automatically.

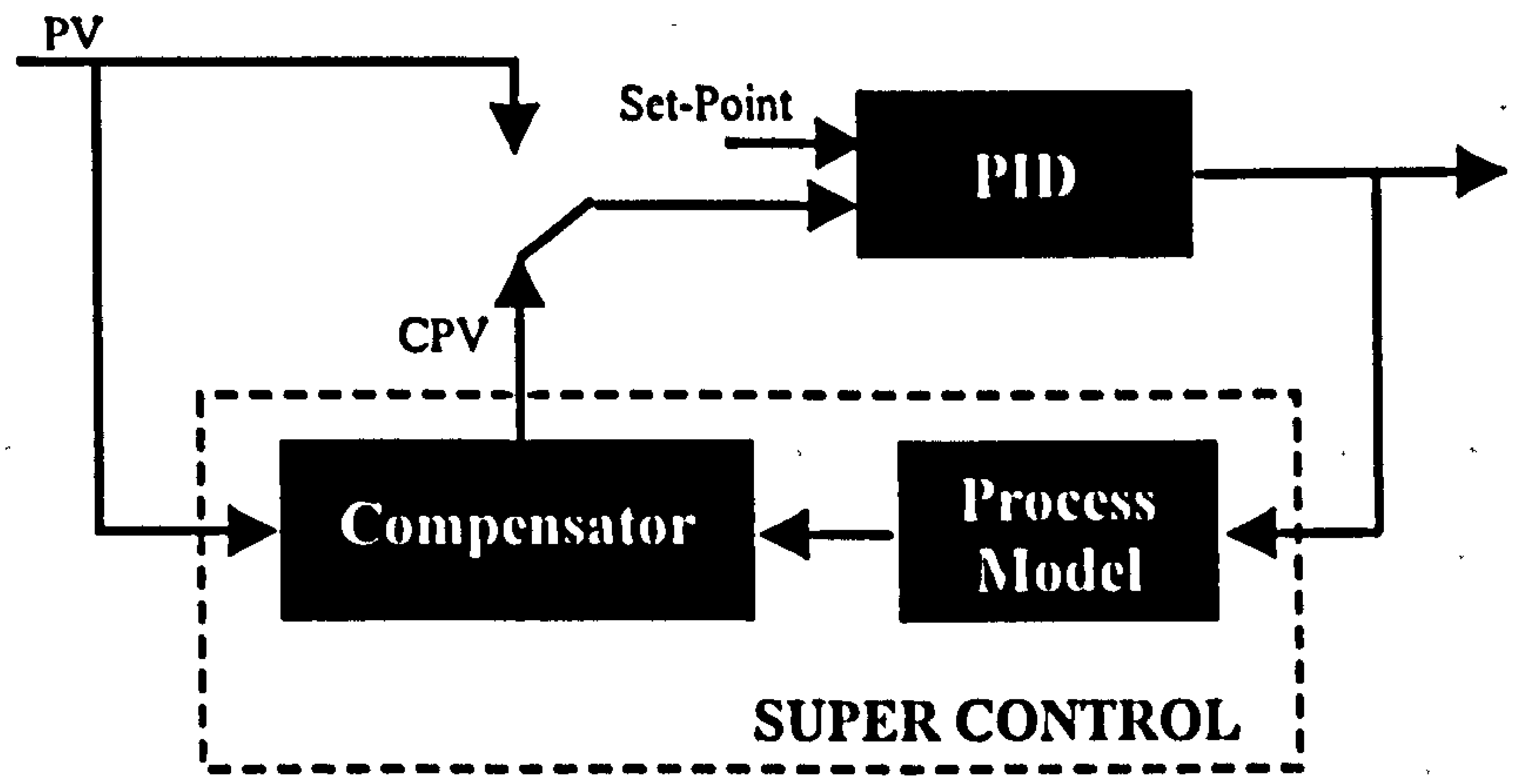

Figure 3.6 Functional Block Diagram of Yokogawa SUPER CONTROL™ Modes 2 and 3 (Wilson and Callen, 2004)

3.4.6 Remarks

Many PID hardware vendors have made a lot of effort to provide a built-in tuning facility. Owing to their vast experience on PID control, most manufacturers have incorporated their knowledge base into their algorithms. Current PID control modules provide 'tuning on demand' with upset or 'adaptive tuning' or both, depending on the model and user settings. Either technique has its advantages and disadvantages. For example, if using 'tuning on demand' only, the controller needs to be retuned periodically and whenever changes occur in the process dynamics. This is tedious and sometimes poor-performance can be noticed too late. Therefore, 'tuning on demand' coupled with 'gain-scheduling' could provide an advantage.

If relying on an 'adaptive tuner' only, the range of changes that can be covered is rather limited and a classical step-response model is still needed for determining initial PID settings. Before normal operations may begin, these systems generally require a carefully supervised start-up and testing period. Furthermore, the more controller parameters the operator needs to select, the more difficult it is to adjust for optimal performance and the longer it takes to prepare for the operation. Nevertheless, once the controller is correctly configured, it can constantly monitor the process and automatically adjust the controller parameters to adapt to the changes in the process.

The second effort made by many PID hardware vendors appears to be incorporating an overshoot suppression function in their on-board algorithms. In order to meet multiple objectives highlighted in Section 2.2.7, they have also added other functions to a standard PID algorithm or allowed the user to switch between modes. However, these features are not commonly seen in practical software packages (see Table 3.2).

3.5 Summary

Many PID patents filed so far focus on automatic tuning for process control. This starts from conventional or 'intelligent' system identification and is more related to hardware modules. Software packages are mainly focused on off-line simulation and have thus a different objective. While automatic tuning is offered in many commercial PID products for multiple optimality, timeliness continues to pose a challenge. The major difficulty appears in delivering an optimal transient response, due to difficulties in setting an optimal derivative term. Hence, modifications to the easy-to-understand PID structure have been made through the use of artificial intelligence so as to suppress overshoots. In order to meet multiple objectives, switching between different functional modes has been offered in PID hardware modules. This further enforces the need for a tuning rule that is designed for handling the multiple objectives.

The present trend in tackling PID tuning problem is to be able to use the standard PID structure to meet multiple design objectives over a reasonable range of operations and systems. Standardisation or modularisation around this structure should also help improve the cost-effectiveness of PID control and its maintenance. This way, a robustly optimal tuning method can be developed. With the inclusion of system identification techniques, the entire PID design and tuning process can be automated and modular building blocks can be made available for timely on-line application and adaptation. This would be particularly suited to 'system-on-board' or 'system-on-chip' integration for future consumer electronics and MEMS.

Therefore, equipped with all this information, the proposed MOEA described in Chapter 4 will be used in the search for multi-objective PID tuning rules detailed in Chapter 5.

Chapter 4

Multi-Objective Evolutionary Algorithms: Analysis and Visualisation

Chapter objectives

This chapter introduces the proposed MOEA methodology, investigates the performance assessment of different MOEAs and visualisation of their solutions in high order dimensions.

4.1 Introduction

Significant progress has been made on the development of MOEA techniques; however the existence of ability to evaluate that progress quantitatively is very small. Due to various experimental methods and performance measures used by researchers nowadays, a thorough comparison is difficult. This is because no one offers a simple-to-use or widely accepted method for evaluating the performance of MOEAs.

At present, ways of comparing non-dominated set of solutions are through visual comparison in the objective space. This method is simple and straightforward. The criterion is to have solutions close to the true Pareto front and must be well distributed over the Pareto frontier. On the other hand, this kind of visualisation is limited to a maximum of three objectives. There are also some other visualisation techniques for viewing high order dimensions, for e.g. scatter-plot matrix, value path, bar chart, start coordinate, etc. as reported by Deb (2001). However, these are not commonly used in MOEA studies, as they are only suitable for displaying a set of non-dominated solutions. Since a MOEA is a stochastic method, multiple runs are required in order to have any statistical significance. Therefore, it is very difficult to view all the runs together in a single plot using those techniques.

Hence, various quantitative and qualitative metrics have been proposed as discussed in the Section 4.4. They are developed to measure MOEA performance more accurately than just visual comparison. Some of them are designed upon the basic criteria of a good MOEA, namely, closeness to the optimal solutions in the objective space and coverage of a wide range of diverse solutions.

Conversely, all the proposed metrics have their limitations. The main problem is the lack of decision maker preferences in the comparison, thereby causing difficulties in certain comparisons. Hansen and Jaszkiewicz (1998) have proposed a formal framework for evaluating the quality of a non-dominated set. However, the proposed metrics only cover the distance between competing non-dominated sets or the distance between a competing non-dominated set and a reference set. Moreover, there are a few intricate settings that the users need to determine, for e.g., the choice of the set of utility functions, the choice of probability distribution of the utility functions and utility functions scaling. Hence, it is neither easy nor straightforward to use.

Recently, Zitzler *et al.* (2003) classified the available metrics into unary and binary types. They have shown that all unary metrics fail to provide reliable performance indication based on dominance relations. However, Bosman and Thierens (2003) have stated that most of the latest MOEAs results would most probably be classified as incomparable using dominance relations of Zitzler *et al.* (2003). In addition, when two sets of non-dominated solutions are incomparable, one of the sets must be more preferable. Thus, unary metrics are still very useful. Farhang-Mehr and Azarm (2003) proposed a conceptual framework based on excellence relations, which attempt to address all the desired aspects of a quality non-dominated solution set. However, to find or design a suitable metric for their framework is not a trivial task.

Indeed, knowledge of the goodness of an observed Pareto solution set should enable the designer to monitor and potentially improve the performance of an MOEA. It should also help the designer to compare and contrast the quality of observed Pareto solution sets as reported by different MOEAs. The goodness of an observed Pareto solution set, as analysed and discussed in this chapter, can be evaluated by performance metrics.

This chapter begins by looking at the approach by those commonly cited MOEAs. By simple examination of their significant features, an easy-to-understand and -implement algorithm is proposed. This is followed by a study on the performance metrics found in single- and multi-objective EAs. Through the studies and analysis of the performance metrics, problems and limitations are highlighted. This leads to the proposal of a novel visualisation technique that aims to alleviate the problems and limitations of the performance metrics. This chapter concludes by an extensive empirical assessment of the proposed methodology with existing MOEAs on a wide range of test problems. The results are analysed using both the performance metrics and visualisation technique.

4.2 Proposed MOEA Methodology

Evolutionary based techniques for multi-objective optimisation can be generally classified into three approaches, namely, aggregating, non-Pareto and Pareto-based. Over the numerous years of studies, Pareto-based evolutionary approaches are well known to out-perform the other approaches (Zitzler and Thiele, 1999; Zitzler *et al.*, 2000; Tan *et al.*, 2001a; Tan *et al.*, 2001b). Hence, the proposed MOEA will adopt Pareto-based approach with elitism.

Some of the most prominent and commonly cited MOEAs are Nondominated Sorting Genetic Algorithm (NSGA-II) (Deb *et al.*, 2000), Pareto Archived Evolution Strategy (PAES) (Knowles and Corne, 2000) and Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele, 1999).

The Nondominated Sorting Genetic Algorithm (NSGA-II) sorts the solutions into different levels of non-domination according to the concept of Pareto dominance. Within each level, a specific crowding measure which represents the sum of distances to the two closest solutions along each objective is used to define an order among the solutions.

The Pareto Archived Evolution Strategy (PAES) approach uses a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) together with a historical archive that records all the non-dominated solutions previously found. It uses a novel approach to keep diversity, which consists of a crowding procedure that divides the objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives. A map of such grid is maintained, indicating the amount of solutions that reside in each grid location.

The Strength Pareto Evolutionary Algorithm (SPEA) approach uses an archive containing non-dominated previously found (the so-called external non-dominated set). At each generation, non-dominated solutions are copied to the external non-dominated set. For each solution in this external set, a strength value is computed. This strength is proportional to the number of solutions to which it dominates. The fitness of each solution in the current population is computed according to the strengths of all the external non-dominated solutions that dominate it. Additionally, a clustering technique is used to maintain diversity.

After studying most of the available MOEAs structures, I decided to use the simplest mechanism, nearest neighbourhood method, to maintain the diversity and the rest are standard MOEA structure. Hereby, this algorithm will be termed as s-MOEA (simple-MOEA) for identity sake and its pseudo code is shown in Figure 4.1.

1. Set t = 0
2. Generate initial population P(t) at random
3. Evaluate the fitness of each individual in P(t)
4. REPEAT
   (a) Select parents from P(t)
   (b) Apply recombination and/or mutation to the parents and produce children
   (c) Evaluate the fitness of children
   (d) Select individual from the children or parents and children for next generation P(t+1)
      (d1) Pareto ranking on the combined child and parent
      (d2) If number of rank 1's solutions exceeds population size then apply nearest neighbourhood method; else check if the next rank will exceed the population size, if yes, apply the nearest neighbourhood method again; else proceed on to next rank etc. until the population for next generation is filled up
5. UNTIL terminating criteria met

Figure 4.1 Pseudo Code of s-MOEA

4.3 Single-Objective Performance Comparison Techniques

Generally, there are two approaches to study the performance of EAs. The first is the analytical approach where it is to "prove theorems about algorithms" based upon a mathematical model of computation. The second is the empirical approach where it draws conclusions about algorithms by looking at computational experiments. The analytical approach can yield significant insights into a number of algorithms and problems, and have the appeal of mathematical certainty. However, its analytical difficulty makes it hard to obtain results for most realistic problems and algorithms. This in turn severely limits their range of applications. In addition, a worst-case result which is by definition pathological, may not give meaningful information on how an algorithm will perform on more representative instances. As a consequence of these difficulties, most of the many algorithms developed for large optimisation problems are evaluated empirically – by applying the procedures to a collection of test problems and comparing the observed solution quality and computational burden. Hence, empirical approach is more commonly adopted by researchers for performance comparison. It is still comparatively rare, given the large volume of literature on evolutionary algorithms, to encounter well-designed computational experiments that produce real insights for researchers.

Before proceeding to multi-objective metrics, a brief overview of single objective metrics is given here in order to gain more understanding behind the needs and development of multi-objective metrics. Performance metrics are less commonly used in single objective optimisation problems, as the objective value is often sufficient for the comparison between algorithms under study.

4.3.1 De Jong's Proposed Metrics

De Jong proposed two metrics in his thesis (De Jong, 1975). One is to gauge the convergence and the other is the ongoing performance, referred to as off-line (convergence) and on-line (ongoing) performance respectively. In his study, De Jong defined the on-line performance $x_e(s)$ of strategy s on environment e as follows:

$$x_e(s) = \frac{1}{T}\sum_{1}^{T} f_e(t) \tag{4.1}$$

where $f_e(t)$ is the objective function value for environment e on trial t. In other words, the on-line performance is measured by the average of all function evaluations up to and including the current trial. While De Jong presented a more general version of this criterion, which permitted non-uniform weighting of trials, conversely a uniform weighting was adopted throughout his study.

The off-line performance, $x^*_e(s)$, of strategy s on environment e is defined as follows:

$$x^*_e(s) = \frac{1}{T}\sum_{1}^{T} f^*_e(t) \tag{4.2}$$

where $f^*_e(t)$ = best$\{f_e(1), f_e(2), \ldots, f_e(t)\}$. In other words, the off-line performance is measured by a running average of the best performance values to a particular time. Once more, a non-uniformly weighted version of this criterion was also proposed although uniform trial weighting was used throughout.

4.3.2 Schwefel's Progress Metric

In order to assess the convergence speed of EAs, a metric is needed independent of the respective starting values on a relative, rather than the absolute, improvement. Schwefel (1988) proposed the progress metric of a single run as:

$$P = \ln\sqrt{\frac{f(1)}{f(T)}} \tag{4.3}$$

where $f(1)$ and $f(T)$ are the best objective function values at the first generation and T generation respectively.

To obtain statistically significant data, a sufficiently large number N of independent runs must be performed. This is often based on the hypothesis that the different progress values P_i ($i \in \{1,\ldots,N\}$) are normally distributed with expectation estimated by the average:

$$\overline{P} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{4.4}$$

and standard deviation estimated by the empirical standard deviation:

$$V(P_i) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(P_i - \overline{P}\right)^2} \tag{4.5}$$

4.3.3 Other Metrics Proposed

Some other metrics, namely, Optimality, Accuracy, Sensitivity and Convergence, have been proposed to systemise the tests of EAs (Feng *et al.*, 1998). This metrics are proposed for use when the convergence of an EA is hard to assess through theoretical proofs.

Optimality represents the relative closeness (or, inversely, distance) of an objective found, $\hat{f}_0$, to the theoretical objective, f_0. It is defined as:

$$Optimality|_a = 1 - \frac{\left\| f_0 - \hat{f}_0 \right\|_a}{\left\| \overline{f} - \underline{f} \right\|_a} \in [0,1] \tag{4.6}$$

where $\underline{f}$ and $\overline{f}$ are the lower and upper bounds of f respectively. Any popular norm used in optimisation or engineering studies may apply to (4.6). In engineering, the 2-norm (Euclidean metric) is most commonly adopted for such a metric and thus the optimality defined in Euclidean space can be termed as 'Euclidean optimality'. A random guess in a search will result in a random optimality value within [0,1].

Accuracy represents the relative closeness of a solution found, $\hat{x}_0$, to the theoretical solution, x_0. This may be particularly useful if the solution space is noisy, or there exist multiple optima, or 'niching' is used, which is defined as:

$$Accuracy = 1 - \frac{\|x_0 - \hat{x}_0\|}{\|\bar{x} - \underline{x}\|} \in [0,1] \tag{4.7}$$

where $\underline{x}$ and $\bar{x}$ are the lower and upper bounds of x respectively, representing the search range.

When the values of optimal parameters found are perturbed (or manufacturing tolerance in accuracy is taken into account), the actual optimality may well change. This affects the robustness of an engineering design. To measure how much a 'small' relative change in the designed parameters (or solutions found) will lead to relative changes in the quality (the objective value found), sensitivity is defined as the ratio between these changes, i.e.:

$$\begin{aligned} Sensitivity &= \lim_{\|\Delta x\|\to 0} \left. \frac{\|\Delta f\| / \|\bar{f} - \underline{f}\|}{\|\Delta x\| / \|\bar{x} - \underline{x}\|} \right|_{x=\hat{x}_0} \\ &= \lim_{\|\Delta x\|\to 0} \left. \frac{\|\Delta f\|}{\|\Delta x\|} \right|_{x=\hat{x}_0} \frac{\|\bar{x} - \underline{x}\|}{\|\bar{f} - \underline{f}\|} && (4.8) \\ &\approx \frac{1 - Optimality}{1 - Accuracy} && (4.9) \end{aligned}$$

Note that the trend of sensitivity is rather dependent on the nature of the problem and the objective function, and not mainly on the algorithm. Sensitivity would be a more useful indicator in a practical design than in an EA performance assessment test. If the sensitivity (and thus design robustness) can be calculated during function evaluations or simulations, it could be used as an additional objective in the design.

In GA, the average fitness of the entire population is often used to assess the convergence trend qualitatively as the mutation rate in a GA is relatively very low. This fitness is, however, often oscillatory when the evolution reaches a 'steady-state' or a relatively high mutation rate is used in the case of EP or ES. Therefore, the fitness differs from the concept of 'convergence' adopted in conventional optimisation paradigms and can hardly fulfil the role as a quantitative indicator or performance metric of convergence. Hence, the following traces are used to indicate the generational convergence:

- The highest 'optimality' or fitness in every generation;

- The highest 'accuracy' or the parameter values of the individual solution that has the highest fitness in every generation.

In order to quantify the convergence metric with respect to an EA, define

$$\mathrm{Re}\,ach-time|_b = C^b \tag{4.10}$$

to represent the total number of 'function evaluations' conducted by which the optimality of the best individual first reaches $b \in [0,1]$. This also means that the relative distance to the theoretical objective first drops to 1-b by the 'reach-time'. For example, the following two reach-times may be useful indicators:

- $C^{0.999}$
- $C^{0.632}$

The former would perhaps be the most significant indicator, in which the optimality is regarded as 100%. The latter means a convergence 'time-constant' by which an optimality of 63.2% is first reached analogous to a first-order dynamic system.

The capability of an EA is that it reduces exponential computational time needed by an exhaustive search algorithm to a non-deterministic polynomial (NP) computational time. To estimate the order of the polynomial, $C^{0.999}$ may be plotted against the number of parameters being optimised, n, as shown below:

$$NP-time(n) = C^{0.999}(n) \tag{4.11}$$

During the entire optimisation process, the optimality of 99.9% may not be reached by certain algorithms under test. The total number of evaluations is the number of function evaluations, search trials or simulations performed in the entire optimisation process until termination. This should be kept the same for all the algorithms compared in a performance test, such as $400mn^2$. It may be more informatively defined as:

$$N = \min\{C^{0.999}, 400mn^2\} \tag{4.12}$$

which implies that a performance test should terminate either when the goal has been reached or $20n$ generations of a size of $20n$ x m has evolved. This also means that we have faith that EAs should not perform worse than an $O(n^2)$ algorithm in terms of computational time.

Lastly, in addition to the 'total number of evaluations', the 'total CPU time' may be used in a performance test. Optimiser overhead would be useful in indicating how long an optimisation or simulated evolution process would take in real world and to indicate the amount of program overhead as a result of the optimisation manipulations such as those by EA operators. More quantitatively, the optimiser overhead may be calculated as:

$$\text{Optimiser Overhead} = \frac{\text{Total time taken} - T_{PFE}}{T_{PFE}} \tag{4.13}$$

where T_{PFE} is the time taken for pure function evaluations.

4.4 Multi-Objective Performance Comparison Techniques

Based on the metrics presented in previous section, single objective comparison is very easy and straightforward and the results are clear-cut. However, this is not the case for multi-objective problems since the result is not a single optimal solutions but a set of non-dominated solutions. Hence, there are numerous studies conducted on the development and survey of metrics on measuring MOEA performance (Deb, 2001; Ang and Li, 2002a; Ang *et al.*, 2002a; Knowles and Corne, 2002; Sarker and Coello Coello, 2002; Zitzler *et al.*, 2003).

In this section, some of the commonly cited and used metrics will be analysed. The available metrics can be generally classified into unary and binary type. Unary metrics' assign a number to a set of non-dominated solution found by an algorithm that reflects a certain quality aspect. Binary metrics' assign a number to pairs of non-dominated solution set.

4.4.1 Unary Type of Metrics

Error ratio (Van Veldhuizen, 1999): This metric shows the ratio of the solutions found by a MOEA that does not belong to the true Pareto front.

$$E = \frac{\sum_{i=1}^{n} e_i}{n} \tag{4.14}$$

where n is the number of solutions found by a MOEA and $e_i = 0$ if solution i is a member of the true Pareto front and 1 otherwise. The drawback of this metric is the requirement of the true Pareto front. Furthermore, it might not be informative as it does not really indicate how well a MOEA performs. For example in the case of two competing MOEAs, if both have all their solutions, except one, that are members of the true Pareto front, then this metric will show that both MOEAs have the same result. This will be misleading as one of the error solutions could be "very far away" from the true Pareto front compared to the other MOEA.

Generational Distance (Van Veldhuizen, 1999): This metric shows the average distance from the solutions found to the true Pareto front.

$$G = \frac{\left(\sum_{i=1}^{n} d_i^p\right)^{1/p}}{n} \tag{4.15}$$

where n is the number of solutions found by a MOEA, d_i is the distance (in objective space) between each solution and the nearest Pareto-optimal solution and p=2 for Euclidean distance. A value of zero indicates those solutions found are indeed the true Pareto front and any value above zero indicates the solutions found deviate from the true Pareto front. This metric is useful as it shows the closeness of the solutions found with respect to the true Pareto front. The weakness is that it is required that the true Pareto front and it might be misleading if used alone.

Maximum Pareto Front Error (Van Veldhuizen, 1999): This metric shows the largest minimum distance between those solutions found and the corresponding closest true Pareto front.

$$ME = \max_j\left(\min_i \left|f_1^i(x) - f_1^j(x)\right|^p + \left|f_2^i(x) - f_2^j(x)\right|^p\right)^{1/p} \tag{4.16}$$

where i and j are the index solutions of the solutions found by a MOEA and the true Pareto front respectively. The weakness is the requirement of a true Pareto front. This metric will be useful if it is used together with error ratio metric.

Size of Space Covered (Zitzler and Thiele, 1998): This metric shows the size of the objective value space that is covered by a set of non-dominated solutions. In the two-dimensional case, each non-dominated solution covers an area — a rectangle defined by the points (0,0) and $(f_1(x),f_2(x))$. The union of all rectangles covered by the set of non-dominated solutions constitutes the total space covered. This metric may be canonically extended to multiple dimensions. An advantage of this measure is that each MOEA can be evaluated independent of the other MOEAs. However, convex regions may be preferred to concave regions, possibly leading to overrating of certain solutions. This metric attempts to combine all three criteria together, namely, distance, distribution, and extent. However, this metric might not be indicative as solutions differing in more than one criterion may not be distinguished. Nevertheless, it does not require true Pareto front. This metric is termed hyperarea in Van Veldhuizen (1999).

Hyperarea Ratio (Van Veldhuizen, 1999): Hyperarea ratio is a ratio of the hyperarea of the solution found and the true Pareto front.

$$HR = \frac{H_{Found}}{H_{True}} \tag{4.17}$$

where H_{Found} is the hyperarea of the solution found and H_{True} is the hyperarea of the true Pareto front. This metric attempt to solve the problem of size of space covered metric when the true Pareto front is non-convex.

Overall Nondominated Vector Generation and Ratio (Van Veldhuizen, 1999): Overall Nondominated Vector Generation (ONVG) metric shows the total number of non-dominated solutions found during MOEA execution. Overall Nondominated Vector Generation Ratio (ONVGR) metric shows the ratio of ONVG and the true Pareto front. ONVG metric if used alone, cannot reflect if the non-dominated solutions are 'close' to the true Pareto front. ONVGR metric attempt to solve this problem of ONVG, but it requires the true Pareto front.

Progress Metric (Van Veldhuizen, 1999): This metric is being modified from Schwefel (1988) which was used to assess single-objective EA convergence velocity that quantifies relative rather than absolute convergence improvement.

$$RP = \ln \sqrt{\frac{G_1}{G_T}} \tag{4.18}$$

where G_1 and G_T are the generational distance at the first and T generations respectively. This metric is informative as it reports on a MOEA convergence improvement rate. The problem is it depends on generational distance metric and hence it requires the true Pareto front.

Generational Nondominated Vector Generation (GNVG) (Van Veldhuizen, 1999): This metric shows how many non-dominated solutions are produced in each MOEA generation. This can be quite misleading as the so-called non-dominated solutions produced by an MOEA might not be global non-dominated.

Nondominated Vector Addition (NVA) (Van Veldhuizen, 1999): This metric indicates how many non-dominated solutions are added at each MOEA generation. It is simply the difference between the number of non-dominated solutions found in the present generation and the previous generation. This is quite similar to the previous GNVG metric and therefore it has the same problem. In addition, this metric might be misleading if a single solution added to the current population may dominate and thus remove several others. The size of the non-dominated solutions found may also remain constant for several successive generations even if GNVG $\neq 0$.

Spacing (Schott, 1995): This metric shows the spread (distribution) of the solutions found by a MOEA.

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{d}-d_i\right)^2} \tag{4.19}$$

where n is the number of solutions found by a MOEA, $d_i = min_{i\in n \cap i\neq j}\sum_{m=1}^{M}\left|f_m^i - f_m^j\right|$ and $\bar{d}$ is the arithmetic mean of all d_i. A value of zero indicates all the solutions found are equidistantly spaced. This metric might be misleading if an algorithm has all the solutions crowded together, occupying a small area of the Pareto front and another algorithm has its solutions well spread over the Pareto front. It will favour the algorithm with all the solutions jam-packed together. One merit of this metric is it does not require the true Pareto front.

Chi-Square Distribution (Deb, 1989): This metric serves the same purposes as spacing metric.

$$\iota = \sqrt{\sum_{i=1}^{q+1}\left(\frac{n_i - \overline{n_i}}{\sigma_i}\right)^2} \tag{4.20}$$

where q is the number of desired optimal points and the $(q+1)$-th sub-region is the dominated region, n_i is the actual number of individuals serving i-th sub-region (niche) of the non-dominated region, $\overline{n_i}$ is the expected number of individuals serving i-th sub-region of the non-dominated region. Using probability theory, it was estimated that

$$\sigma_i^2 = \overline{n_i}\left(1 - \frac{\overline{n_i}}{P}\right) \qquad i = 1,2,\ldots,q \tag{4.21}$$

$$\sigma_{q+1}^2 = \sum_{i=1}^{q} \sigma_i^2 \tag{4.22}$$

where P is the population size. Since it is not desirable to have any individual in the dominated region (i.e., the (q+1)-th sub-region), $\overline{n_{q+1}} = 0$. If the distribution of points is ideal with $\overline{n_i}$ number of points in the i-th sub-region, the measure $\iota = 0$. Therefore, an algorithm with a good distribution capability is characterised by a low deviation measure. The major difficulty with this metric is to determine the sub-region size, as the size will greatly influence the result. Another problem is several parameters need to be estimated before using this metric.

Diversity (Deb, 2001): This metric is commonly being used in place of chi-square distribution metric.

$$Diversity = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{n} \left| d_i - \bar{d} \right|}{\sum_{m=1}^{M} d_m^e + n\bar{d}} \tag{4.23}$$

where d_i can be any distance measure between neighbouring solutions, d_m^e is the distance between the extreme solutions of the obtained non-dominated set and the true Pareto front, n is the number of solutions found, $\bar{d}$ is the average of all distances d_i. This metric is very similar to spacing metric. It also shows the spread (distribution) of the solutions found by a MOEA, however it is much more comprehensive as it also take care of the extreme ends. Hence, it solves the deficiency of spacing metric as it penalises those solutions that are packed together and not well distributed over the Pareto front. However, it also requires the true Pareto front.

4.4.2 Binary Type of Metrics

Coverage of two sets (Zitzler and Thiele, 1998): This metric compares the domination of two sets of non-dominated solutions in a pair-wise manner, i.e., how good each solution from each set dominates each other. Let $X', X'' \subseteq X$ be two sets of decision vectors. The function C maps the ordered pair (X', X'') to the interval [0,1]:

$$C(X', X'') = \frac{\left| \{a'' \in X''; \exists a' \in X' : a' \text{ dominate or nondominate } a''\} \right|}{|X''|} \tag{4.24}$$

The value $C(X',X'') = 1$ means that all solutions in X'' are dominated by or equal to solutions in X'. The opposite, $C(X',X'') = 0$ represents the situation when none of the solutions in X'' are covered by X'. However, note that both $C(X',X'')$ and $C(X'',X')$ have to be considered, since $C(X',X'')$ is not necessarily equal to $1- C(X'',X')$. This metric can be quite troublesome and repetitive efforts are needed whenever a user wants to compare existing MOEAs with any new emerging MOEA.

Attainment Surface (Fonseca, 1995): This metric relies on the notion that the non-dominated solutions from any approximation to a true Pareto front define a surface (called the attainment surface), which divides the objective space into a region that is dominated by the discovered non-dominated solutions, and a region that is not dominated by them. Over multiple runs, an MOEA will generate multiple different attainment surfaces. By looking at the superposition of all the attainment surfaces, a quantitative notion of 'typical' performance can be built. In particular, one may want to identify the family of objective vectors likely to be attained, each on its own, in exactly 50% of the runs (also known as the 50%-attainment surface of the MOEA). This 50% attainment surface can be estimated by using arbitrary auxiliary straight lines and sampling their intersections with the set of attainment surfaces. Estimates for the 25% and 75% attainment surfaces could be produced exactly in the same way by estimating the lower and upper quartiles instead of the median. As a result, the samples represented by, for example, the 50% attainment surface can be relatively assessed by means of non-parametric statistical tests and therefore allow comparison of the performance of those competing MOEAs. The merit of this metric is it does not require any knowledge of the true Pareto front. One drawback is that the non-parametric statistical test cannot show the degree to which one MOEA outperforms another. Another difficulty is in determining how many auxiliary lines are sufficient and the auxiliary lines can distort the proportion of the space they cover, yielding unreliable information. Lastly, this metric is computationally intensive as compared with the other metrics studied here.

Attainment Surface Sampling (Knowles and Corne, 2000): This is an extension to the attainment surface metric. The main difference is the way the sampling lines are drawn. Under their proposal, the sampling lines always start from (0,0) for the case of two objectives as compared with Fonseca (1995) where the sampling lines can start from any points. A non-parametric statistical test based on the Mann-Whitney U-test

(Mendenhall and Beaver, 1994) is used to determine which algorithm performs best on the sampled part of the objective space, at a given confidence level. The result of this analysis yields two numbers, a percentage of the surface that an algorithm is unbeaten and that the algorithm defeats others. This metric is a simpler version of attainment surface metric. The difficulty in determining how many sampling lines are sufficient still exists and the sampling lines can distort the proportion of the space they cover, leading to unreliable information.

4.5 Visualisation

Based on previous sections, it is obvious that performance assessment in MOEAs is very difficult when compared with the single objective cases. Due to the difficulty and nature of the problems, metrics that are designed for multiple objectives assessment are either simple but lack accuracy or too complicated and difficult to understand and implement. Hence, it is still common at present to use visual comparison method in some performance assessment studies. This method simply plots the non-dominated solutions found upon termination of a MOEA. The results found by different MOEAs are usually plotted together onto the same plot in order to visually assess which algorithm is better. Even though this visual comparison method is well recognised to be inadequate and inaccurate to critically assess the performance of MOEAs, the demand for it is still there due to its simplicity.

Facing the situation where there is no simple-to-use and widely accepted metric, visualisation somehow is still necessary. Visualising the non-dominated solutions in objective space is limited to a maximum of three objectives. The motivation of this work is to find an easy way to visualise multi-dimensional objective data and the purpose is to gain insight rather than quantitative analysis. It is expected that users are likely to tolerate loss of information in the initial process of evaluating solutions data. Then, through dimensionality reduction and the use of visuals to represent data, they can numerically support the knowledge that they have extracted through performance metrics. This will be more effective when visualisation is used together with available metrics, so as to further validate the results indicated by the metrics.

Instead of plotting the non-dominated solutions in the objective space (which is only limited to three objectives), we propose to plot the non-dominated solutions against their performance indicated by unary metrics. To begin, we start with the most basic technique, that is plotting the non-dominated solutions against their distance to the approximate or true Pareto front and their distance between each other, which we term as the "Distance and Distribution (DD)" chart. The DD chart consists of three elements, namely, approximate or true Pareto front (or sometimes known as reference set), distance metric and distribution metric.

The approximate Pareto front, P^{*}, can be easily generated using either of the two methods. The first method is to have an archive to store all the best-found non-dominated solutions for a particular problem and the second is to gather all the non-dominated solutions found by the competing algorithms and use it as an approximate Pareto front.

The distance metric is simply the Euclidean distance of each solution to the nearest approximate Pareto front solution. This metric is similar to the generational distance metric (Van Veldhuizen, 1999) except that it is used for measuring the individual distance rather than the overall average distance. A zero value indicates that the solution is Pareto-optimal and any values above zero indicate that the solution deviates from the approximate Pareto front. This is denoted as:

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{(k)}\right)^2} \tag{4.25}$$

where i is the i-th solution of the non-dominated solution set and $f_m^{(k)}$ is the m-th objective function value of the k-th member of the approximate Pareto front.

The distribution metric is simply the Euclidean distance between each solution and taking into consideration the distance between the boundary solutions and the approximate Pareto front. This metric is similar to the diversity metric (Deb, 2001) except that it is used for measuring the individual gap distance rather than the overall average gap distance. Thus, a low performance metric characterises an algorithm with a good distribution capability. This is denoted as:

$$g_i = \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{(j)}\right)^2} \tag{4.26}$$

where i and j are the solutions of the non-dominated solution set.

The computation for distance metric is straightforward. As for the distribution metric, it will get complicated when the number of objectives is more than two. In this case, Deb (2001) proposed to use the non-dominated solutions to construct a higher-dimensional surface by employing the so-called triangularisation method. As several distance metrics can be associated with such a triangularised surface, the average distance of all edges can be used as the gap distance. Note that this method is extremely computationally expensive.

Hence, we proposed another method to compute the distribution metric that is applicable to any number of objectives. This method is not accurate, but it serves as a useful estimation for the distribution metric. First, the non-dominated solutions found must be sorted. It is recommended to sort based on the first objective. For example, if the first objective is to minimise then the solutions should be sorted in ascending order, based on the first objective value. Now, regardless of the objectives, the two-boundary gap distance calculations are simply the Euclidean distance between the first and last non-dominated solution and the first and last solution of the approximate Pareto front respectively. For example, the two-boundary gap distances ($g1$ and $g4$) can be calculated based on the distance between the first solution found and the first solution of the approximate Pareto front as shown in Figure 4.2, where f1 and f2 are the two objectives to be minimised. The circles represent non-dominated solutions found, squares represent an approximate Pareto front, $d1$ to $d3$ represent the distance metrics and $g1$ to $g4$ represent the distribution metrics.

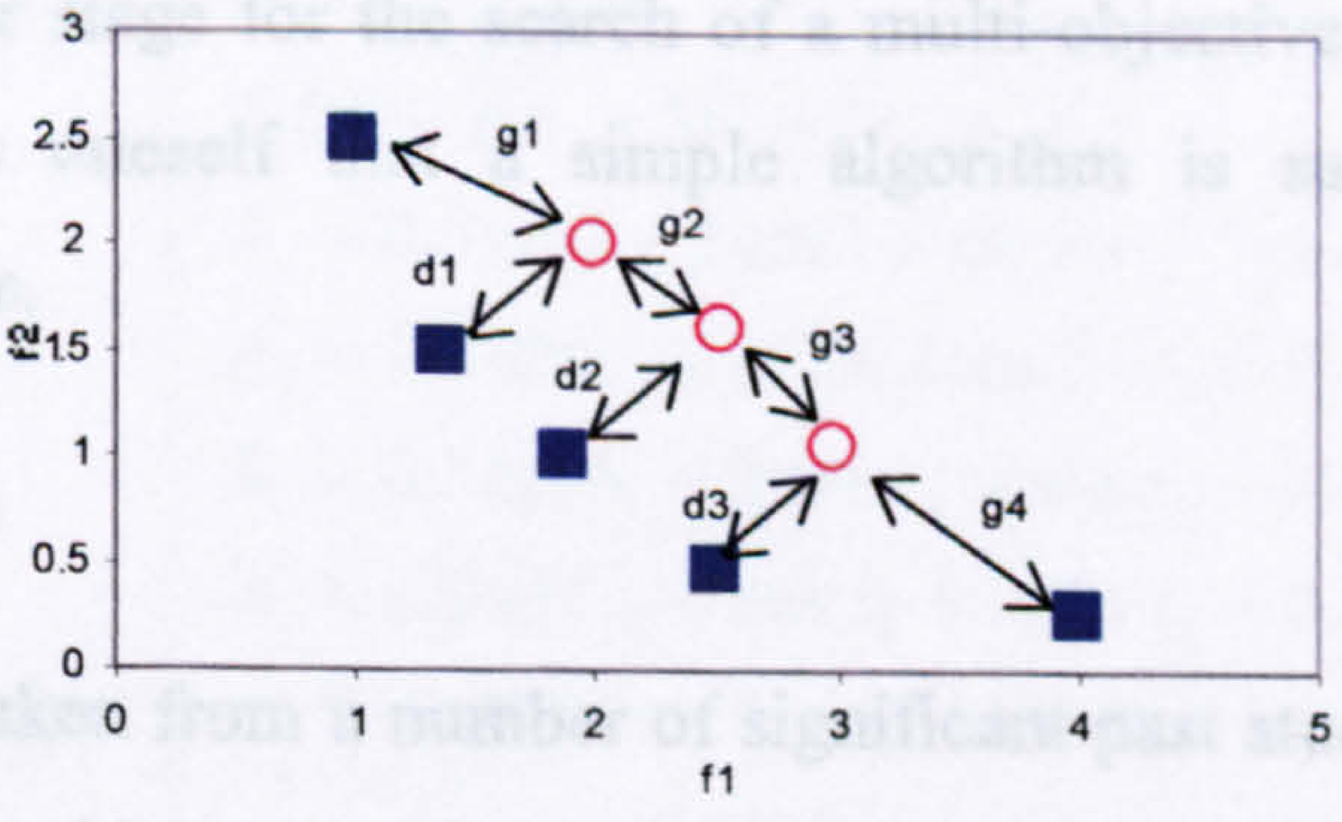

Figure 4.2 An Example Plot

The number of non-dominated solutions required for the DD chart is about 10 to 100. Although, the amount of the competing non-dominated solutions does not need to be the same, their differences should not be more than 50%. Otherwise, it will be difficult to analyse and deduce any conclusive results graphically.

The proposed method is to view the distance and distribution metric of each non-dominated solution found by an algorithm, and to use one simple line chart to plot the non-dominated solutions against its distance metrics and another line chart to plot the non-dominated solutions against its distribution metrics. The distance chart will not only provide information on the overall distance of the solution to the approximate front, but also reveal the maximum Pareto front error. As for the distribution chart, it reveals the coverage of the non-dominated solutions in the objective space.

4.6 Empirical Assessment

The performance of the proposed MOEA, s-MOEA, will be compared with (1+1)-PAES, NSGA-II and SPEA on a set of test problems. The algorithms are implemented according to their descriptions in the literatures. The concerns in the main feature are the fitness assignment and the selection processes; the proposed implementation only differs in these aspects, where the other operators (crossover and mutation) remain identical. For each algorithm, identical population and archive sizes are used. The archive is used to store and update all the best solutions found during each generation.

Please note that this section is not meant to provide detailed analysis into the available MOEAs or to show the superiority of s-MOEA. It is, however, used to verify if the Java library implemented for those MOEAs is running correctly. It is necessary, as it will be used in the later stage for the search of a multi-objective PID tuning rule. It is also used to convince oneself that a simple algorithm is sufficient to provide a satisfactory performance.

4.6.1 Test Problems

The test problems are taken from a number of significant past studies in this area. From these studies, fourteen problems are chosen and they are labelled as FON (from Fonseca and Fleming's study (1995b)), KUR (from Kursawe's study (1990)), POL (from Poloni's study (1995)), SCH (from Schaffer's study (1987)), ZDT1-4 and ZDT6 (from Zitzler *et al.*'s study (2000)), VFM3 (from Viennet *et al.*'s study (1996)) and DTLZ1-4 (from Deb *et al.*'s study (2002)). None of these test problems has any constraint and they are described in details in Table 4.1. As the constraint handling can be done with ease based on the design as shown in Section 2.3.1.

All the approaches are run for a maximum of 300 generations, with the population and archive size set to 100. The crossover probability is fixed at 0.9 and mutation probability is set to $1/n$ (where n is the number of decision variables for real-coded MOEAs). The SBX-20 operator is used for crossover and a polynomial distribution for mutation (Deb and Agrawal, 1995). The archive obtained at the end of the 300 generations is used to calculate a couple of performance metrics (which will be discussed in the next sub-section). For (1+1)-PAES, the depth value is set at 5.

Table 4.1 Unconstrained Test Problems with All Objective Functions to be Minimised

Problem	n	**Variable bounds**	**Objective functions**	**Comments**
FON	3	[-4,4]	$f_1(x)=1-\exp\left(-\sum_{i=1}^{3}\left(x_i-\frac{1}{\sqrt{3}}\right)^2\right)$ $f_2(x)=1-\exp\left(-\sum_{i=1}^{3}\left(x_i+\frac{1}{\sqrt{3}}\right)^2\right)$	Non-convex
KUR	3	[-5,5]	$f_1(x)=\sum_{i=1}^{n-1}\left(-10\exp\left(-0.2\sqrt{x_i^2+x_{i+1}^2}\right)\right)$ $f_2(x)=\sum_{i=1}^{n}\left(\lvert x_i\rvert^{0.8}+5\sin x_i^3\right)$	Non-convex
POL	2	[-π,π]	$f_1(x)=\left[1+(A_1-B_1)^2+(A_2-B_2)^2\right]$ $f_2(x)=\left[(x_1+3)^2+(x_2+1)^2\right]$ $A_1=0.5\sin 1-2\cos 1+\sin 2-1.5\cos 2$ $A_2=1.5\sin 1-\cos 1+2\sin 2-0.5\cos 2$ $B_1=0.5\sin x_1-2\cos x_1+\sin x_2-1.5\cos x_2$ $B_2=1.5\sin x_1-\cos x_1+2\sin x_2-0.5\cos x_2$	Non-convex, disconnected
SCH	1	$[-10^3,10^3]$	$f_1(x)=x^2$ $f_2(x)=(x-2)^2$	Convex
VFM3	2	[-3,3]	$f_1(x)=0.5(x_1^2+x_2^2)+\sin(x_1^2+x_2^2)$ $f_2(x)=\frac{(3x_1-2x_2+4)^2}{8}+\frac{(x_1-x_2+1)^2}{27}+15$ $f_3(x)=\frac{1}{(x_1^2+x_2^2+1)}-1.1e^{(-x_1^2-x_2^2)}$	Continuous

ZDT1	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$	Convex
ZDT2	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-(x_1/g(x))^2\right]$ $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$	Non-convex
ZDT3	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}-\frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$	Convex, disconnected
ZDT4	10	$x_1\in[0,1]$ $x_i\in[-5,5]$, $i=2,\ldots,n$	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x)=1+10(n-1)+\sum_{i=2}^{n}(x_i^2-10\cos(4\pi x_i))$	Non-convex
ZDT6	10	[0,1]	$f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x)=g(x)\left[1-(f_1(x)/g(x))^2\right]$ $g(x)=1+9\left[(\sum_{i=2}^{n}x_i)/(n-1)\right]^{0.25}$	Non-convex, non-uniformly spaced
DTLZ1	7	[0,1]	$f_1(x)=0.5x_1x_2(1+g(x_M))$ $f_2(x)=0.5x_1(1-x_2)(1+g(x_M))$ $f_3(x)=0.5(1-x_1)(1+g(x_M))$ $g(x_M)=100\left[\lvert x_M\rvert+\sum_{x_i\in x_M}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right]$	3-D
DTLZ2	12	[0,1]	$f_1(x)=(1+g(x_M))\cos(x_1\pi/2)\cos(x_2\pi/2)$ $f_2(x)=(1+g(x_M))\cos(x_1\pi/2)\sin(x_2\pi/2)$ $f_3(x)=(1+g(x_M))\sin(x_1\pi/2)$ $g(x_M)=\sum_{x_i\in x_M}(x_i-0.5)^2$	3-D
DTLZ3	12	[0,1]	$f_1(x)=(1+g(x_M))\cos(x_1\pi/2)\cos(x_2\pi/2)$ $f_2(x)=(1+g(x_M))\cos(x_1\pi/2)\sin(x_2\pi/2)$ $f_3(x)=(1+g(x_M))\sin(x_1\pi/2)$ $g(x_M)=100\left[\lvert x_M\rvert+\sum_{x_i\in x_M}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right]$	3-D

DTLZ4	12	[0,1]	$f_1(x) = (1+g(x_M))\cos(x_1^\alpha \pi/2)\cos(x_2^\alpha \pi/2)$ $f_2(x) = (1+g(x_M))\cos(x_1^\alpha \pi/2)\sin(x_2^\alpha \pi/2)$ $f_3(x) = (1+g(x_M))\sin(x_1^\alpha \pi/2)$ $g(x_M) = \sum_{x_i \in x_M}(x_i - 0.5)^2, \quad \alpha = 100$	3-D

4.6.2 Performance Metrics

Unlike single-objective optimisation problem, multi-objective optimisation has two main functional goals. They are, namely, convergence to the Pareto-optimal set and maintenance of diversity in the solution set. It is obvious that these two goals cannot be measured with one performance metric adequately even though attempts have been made. A variety of performance metrics discussed in Section 4.4 will be used to evaluate the performance the MOEAs on the test problems shown in Section 4.6.1. Together with the visualisation technique proposed in Section 4.5, one can roughly deduce if any of the available metrics can provide accurate estimates in evaluating the performance of a MOEA based on the simulation.

A recent study by Zitlzer *et al.* (2003) has shown that for an M-objective optimisation problem, at least M performance metrics must be used. Although a number of different performance metrics have been suggested, many are only applicable to two-objective problems. Most importantly it is not obvious which of these performance metrics to be use in practice. It is intuitive that the use of a set of metrics less than the number of objectives would mean a loss of a dimension and would immediately make the approach theoretically inaccurate. However, one of the ways to overcome the dimensionality problem practically is to use a set of variables that are functionally independent (Goldberg, 1993). Hence, effort can be made in devising metrics based on the two main functional goals of MOEAs. Such metrics will enable performance comparison in terms of their functional requirements.

The selected metrics for this study are *generational distance, diversity, attainment surface sampling* and *optimiser overhead.* They are chosen based on the close similarity to the functional goals of MOEAs and ease of implementation. This is very important in getting user to deploy them without much effort. *Generational distance* and *diversity* metrics are chosen based on the two main functional goals of achieving convergence to

Pareto optimal while maintaining a well diverse set of solutions. However, they are very time-consuming in preparing the data for their computation. *Attainment surface sampling* metric is based on relative comparison between algorithms. *Optimiser overhead* metric indicates the efficiency of an MOEA.

4.6.3 Results and Discussion

Each algorithm and problem was run 30 times with different random seeds. Tables 4.2 to 4.5 show the results of the simulation. For *generational distance*, *diversity* and *optimiser overhead* metrics, lower values indicate better performance while for *attainment surface sampling* metric, higher values suggest better performance.

The problems that some of the algorithms encountered should be highlighted before proceeding to discuss on the results, this is necessary to prevent any misleading information. The following is a list of problems faced by some algorithms:

- All the four algorithms mostly trapped in the local optima on test problem ZDT2;
- SPEA failed to converge anywhere near to the approximate Pareto front on test problem ZDT4;
- s-MOEA and NSGA-II consistently produce the required archive size, while (1+1)-PAES and SPEA mostly did not manage to.

Results from Tables 4.2 to 4.4 will be discussed first. Table 4.5 results will be discussed at a later stage. The discussion here focused mainly on whether all the different metrics provide the same conclusion on a MOEA performance. In the event where there is a difference, the visualisation technique proposed in Section 4.5 will be applied to assist user in making the final conclusive statement on the performance. The context of the best result in this discussion was concluded based on the mean value.

Table 4.2 Mean (shaded rows) and Standard Deviation (unshaded rows) of Generational Distance Metric. Best result is highlighted in red colour.

	s-MOEA	NSGA-II	(1+1)-PAES	SPEA
FON	***0.0001903***	0.0002174	0.0347059	0.0002051
	0.0000311	***0.0000222***	0.1315655	0.0000373
KUR	***0.0012892***	0.0023592	0.2929613	0.0015505
	0.0003388	***0.0001908***	0.7110772	0.0002053

POL	***0.0015356***	0.0099166	0.0579082	0.0018632
	0.0004839	0.0198373	0.1424077	0.0006583
SCH	0.0003838	0.0005188	0.0039502	***0.0003178***
	0.0004189	0.0006623	0.0047169	***0.0000597***
ZDT1	0.0001333	0.0002577	0.0595984	***0.0000981***
	0.0000069	0.0002931	0.0981884	0.0000567
ZDT2	***0.0000016***	0.0000415	0.3314526	0.0000200
	0.0000063	0.0001064	0.4209612	0.0000250
ZDT3	0.0003034	0.0006006	0.0373176	***0.0000998***
	0.0011195	0.0005821	0.1433344	***0.0000256***
ZDT4	***0.0015720***	0.0191595	4.4857448	3.7476752
	0.0064027	0.0667826	3.1162872	2.1833285
ZDT6	***0.0004238***	0.0017923	1.1037344	0.0012434
	0.0000629	0.0010129	1.0264312	0.0002024
VFM3	***0.0009617***	0.0010601	0.3133951	0.0011430
	0.0000805	0.0001428	0.4386621	0.0001460
DTLZ1	0.1813761	***0.0137991***	7.9121307	0.0982418
	0.3207528	***0.0125578***	9.1171815	0.2297886
DTLZ2	***0.0023042***	0.0089405	0.0370805	0.0026277
	0.0015277	0.0024720	0.0514647	***0.0005350***
DTLZ3	0.2219235	***0.1413578***	34.8376217	1.0300805
	0.2784402	***0.1242747***	19.8317261	0.5563667
DTLZ4	***0.0012416***	0.0042002	0.0327058	0.0022949
	0.0005347	0.0018487	0.0654013	***0.0002832***

Table 4.3 Mean (shaded rows) and Standard Deviation (unshaded rows) of Diversity Metric. Best result is highlighted in red colour.

	s-MOEA	**NSGA-II**	**(1+1)-PAES**	**SPEA**
FON	***0.2838104***	0.4002939	0.7437806	0.4402646
	0.0194277	0.0336108	0.1005961	0.0442050
KUR	***0.4027692***	0.4825404	1.0090556	0.4851105
	0.0221449	0.0267995	0.1302290	0.0309282
POL	***0.9459074***	0.9723746	1.1528469	1.0008331
	0.0072669	0.0237324	0.1832135	0.0250407
SCH	***0.3954401***	0.6534920	0.7294428	0.4234878
	0.1598232	0.0592891	0.0602275	***0.0328012***

ZDT1	***0.6751957***	0.7322062	1.0658203	0.7677823
	0.0240484	***0.0203130***	0.1130696	0.0265218
ZDT2	1.0241906	0.9550145	1.0346505	***0.8672508***
	0.1244910	***0.1026438***	0.1881823	0.1175782
ZDT3	***0.7492130***	0.7860603	1.1680104	0.8160597
	0.0205706	***0.0174065***	0.1683934	0.0174521
ZDT4	***0.9455648***	0.9597987	1.3341933	0.9992860
	0.2170644	0.2210939	0.1656440	***0.0039110***
ZDT6	***0.3097098***	0.5031162	1.1791632	0.5528837
	0.0220861	0.0544424	0.1983589	0.0550438
VFM3	0.5959206	***0.4718184***	1.2795245	0.6394972
	0.0358221	0.1072206	0.2003192	0.0396001
DTLZ1	0.8217155	0.8087159	1.3019718	***0.8013833***
	0.2456055	***0.0607149***	0.1814566	0.3077037
DTLZ2	0.6166570	0.7181158	1.1080408	***0.5600703***
	0.0331700	0.0513769	0.1545037	0.0446429
DTLZ3	***0.9240015***	1.0659115	1.1164251	1.1816620
	0.1964611	0.1975325	0.1572957	***0.1002395***
DTLZ4	0.6025389	0.6755746	1.0886586	***0.5669722***
	0.0336717	0.0487950	0.1042259	0.0400486

Table 4.4 Percentage of space unbeaten (shaded rows) and Percentage of space defeats other (unshaded rows) of Attainment Surface Sampling Metric. Best result is highlighted in red colour.

	s-MOEA	NSGA-II	(1+1)-PAES	SPEA
FON	***89.9***	72.1	54.1	46.8
	7.8	1.9	5.4	0
KUR	***70.4***	45.2	0	25.4
	38.5	29.5	0	0
POL	***93.2***	34.3	16	19.8
	54.6	4.1	1.1	0.4
SCH	***89.7***	71	0.1	71.6
	10.8	5.6	0	0.5
ZDT1	***83.7***	52.7	0	9.6
	47.3	10.6	0	0

ZDT2	40.7	***59.3***	11.3	57.7
	40.7	1.5	0	0
ZDT3	59.7	***62.7***	0.1	3.8
	36.7	***40***	0.1	0
ZDT4	3.4	***53.8***	26.8	48
	0.1	***50.4***	0	19.3
ZDT6	***100***	60	0	0
	40	0	0	0
VFM3	37.1191	***74.6999***	0	23.361
	11.1727	***62.0499***	0	0.646353
DTLZ1	98.2456	***100***	25.1154	0
	0	***1.75439***	0	0
DTLZ2	***77.3777***	27.2392	5.26316	3.41644
	65.5586	21.7913	0	0
DTLZ3	98.2456	***100***	0	***100***
	0	***0***	0	***0***
DTLZ4	***95.1062***	41.4589	26.7775	2.12373
	55.8633	1.75439	0.369344	0

For test problems, FON, KUR and POL, all the metrics indicate that s-MOEA is the best among all. As for test problem SCH, it can be quite confusing as *attainment surface sampling* and *diversity* metrics reported that s-MOEA is the best but generational distance metric shows SPEA is the best. In this type of situation, to introduce another metric will not be beneficial, as it will most probably get more confusing. Thus, the DD chart will be used to provide more insight information of the non-dominated solutions of s-MOEA and SPEA. This type of presentation is much neater as compared to displaying all the 30 sets of solutions for each algorithm in the objective space. Figure 4.3 and 4.4 show the DD chart for s-MOEA and SPEA respectively. The way to interpret those DD charts is to look at the overall general trend. Based on the DD chart results, it shows that SPEA outperforms s-MOEA at the expense of smaller archive size.

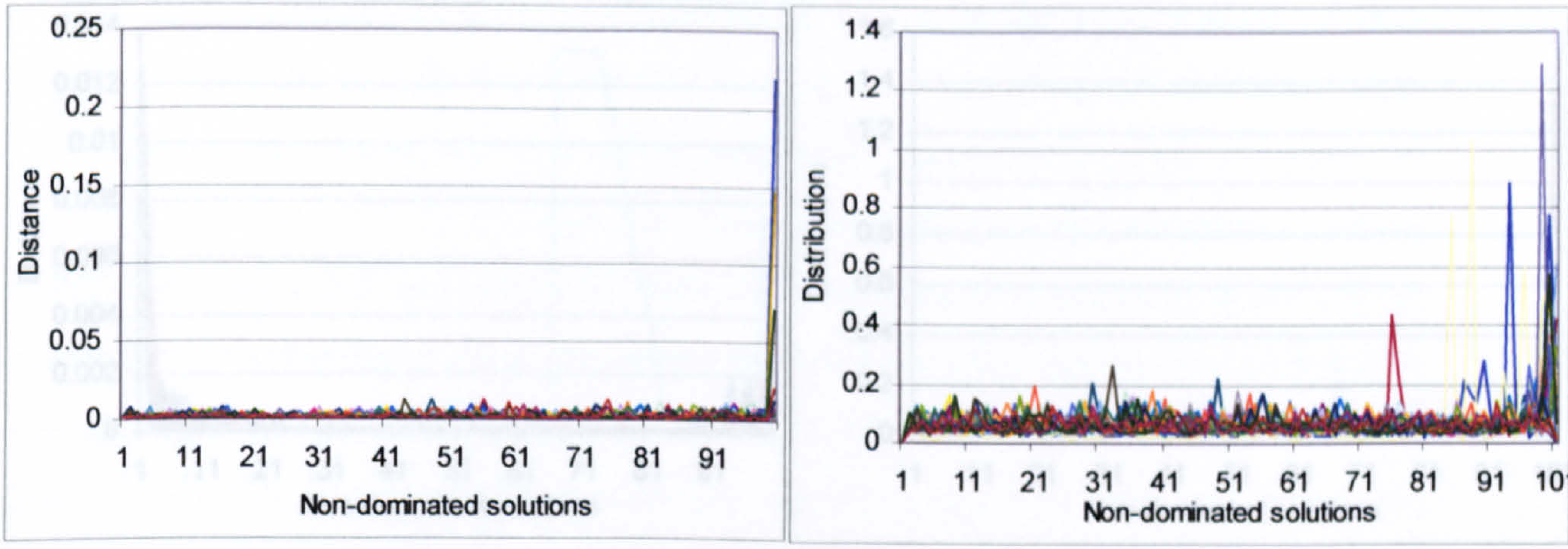

Figure 4.3 DD Chart of s-MOEA on Test Problem SCH

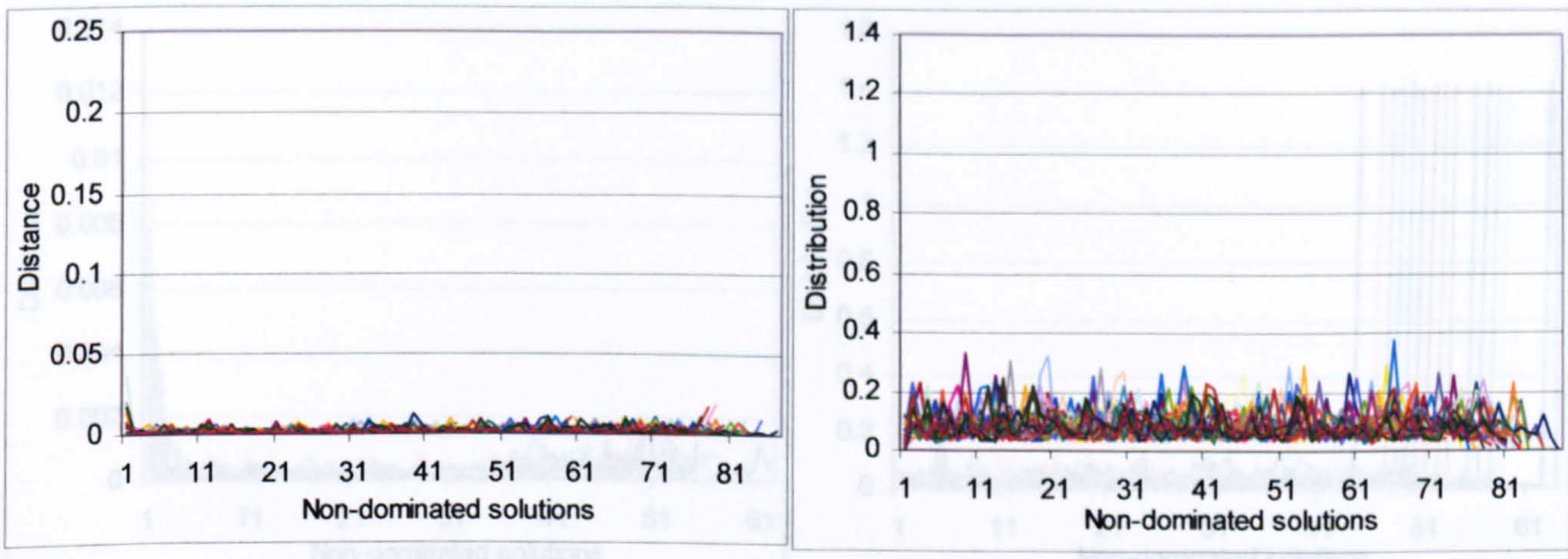

Figure 4.4 DD Chart of SPEA on Test Problem SCH

The same situation, as in test problem SCH for s-MOEA and SPEA, appears again in test problem ZDT1. Thus, the DD chart will be used again. Figure 4.5 and 4.6 show the DD chart for s-MOEA and SPEA respectively. The phenomenon of the DD chart in Figure 4.6 is due to the fact that SPEA has unequal archive size for every simulation runs. From this example, it is obvious that in such a case, different performance metrics will give different results due to their design and thus causing confusion to the user. Yet again, the DD chart shows that it can assist user in making a more subjective selection based on individual preferences.

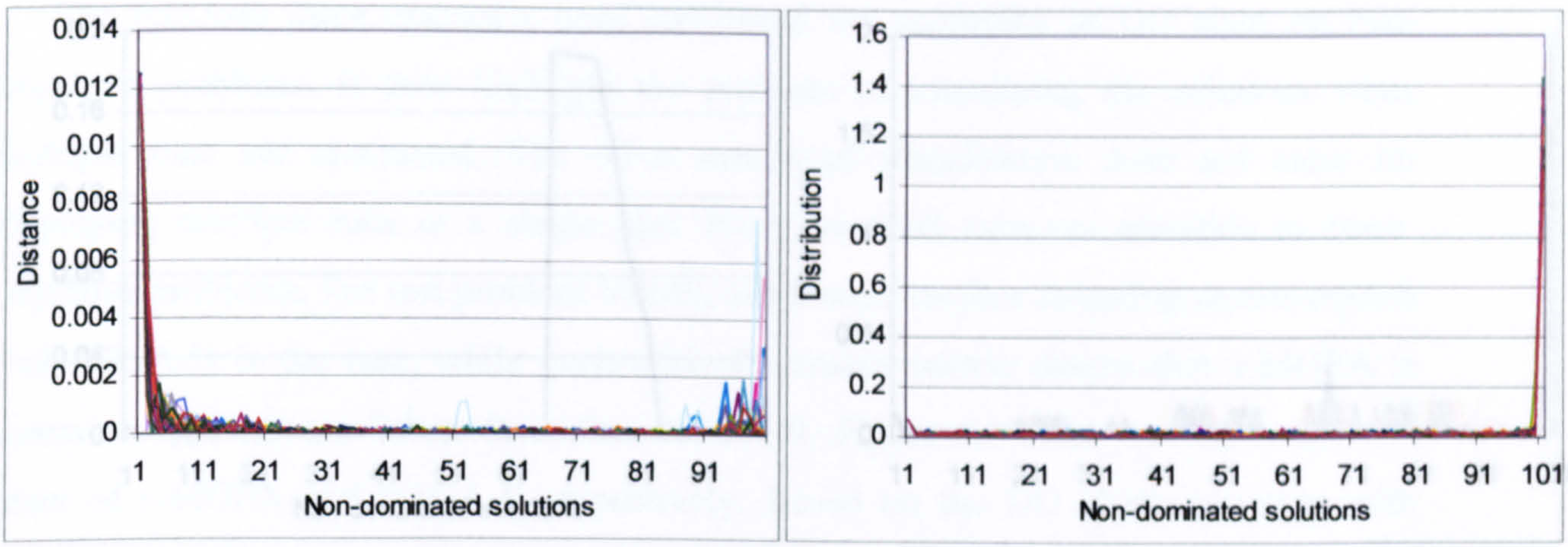

Figure 4.5 DD Chart of s-MOEA on Test Problem ZDT1

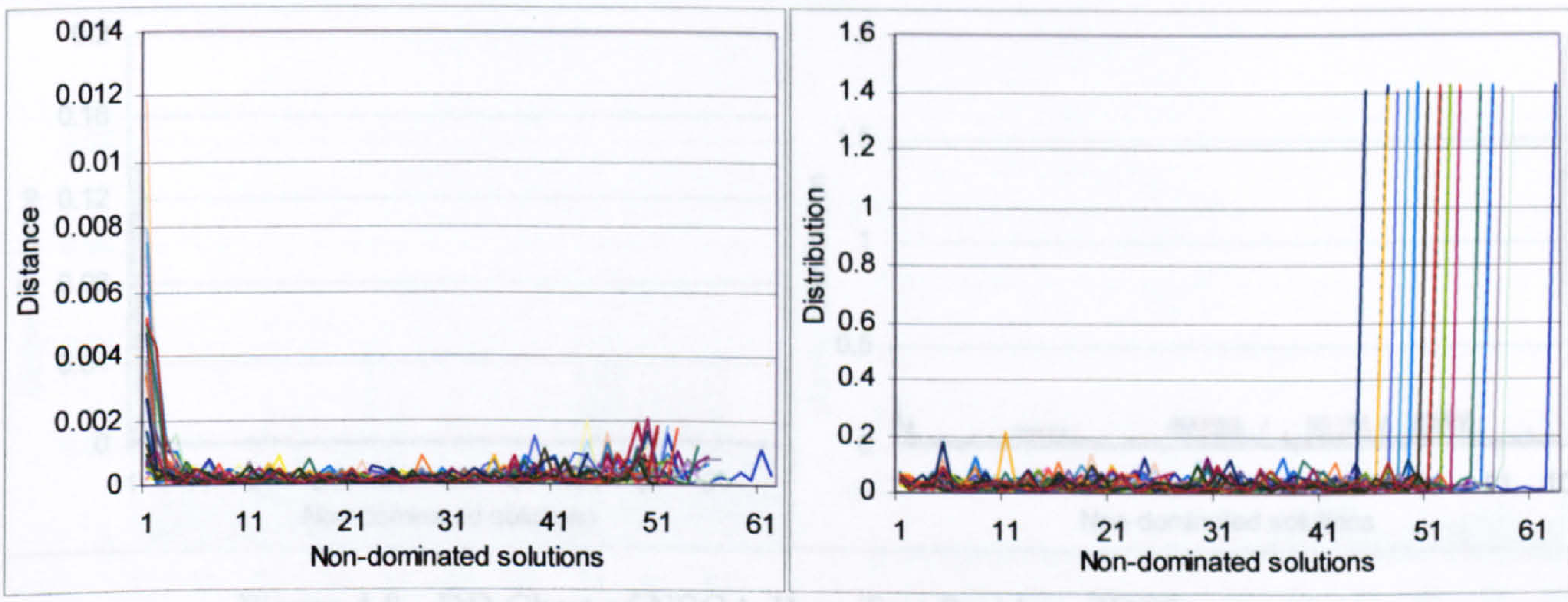

Figure 4.6 DD Chart of SPEA on Test Problem ZDT1

We have a much more interesting situation for test problem ZDT3, where all the three metrics reported different results. In terms of distance to the approximate Pareto front, it seems that SPEA is better but s-MOEA has a better diversity. However, *attainment surface sampling* metric shows that NSGA-II is the best. Figures 4.7 to 4.9 show the DD chart for s-MOEA, NSGA-II and SPEA respectively. The DD chart of the three algorithms shows that s-MOEA might be slightly better as only one of the simulations have some problems converging to the approximate Pareto front. In terms of diversity, s-MOEA is definitely better than the other two. In this case, the DD chart shows that it can provide more insight information than those metrics.

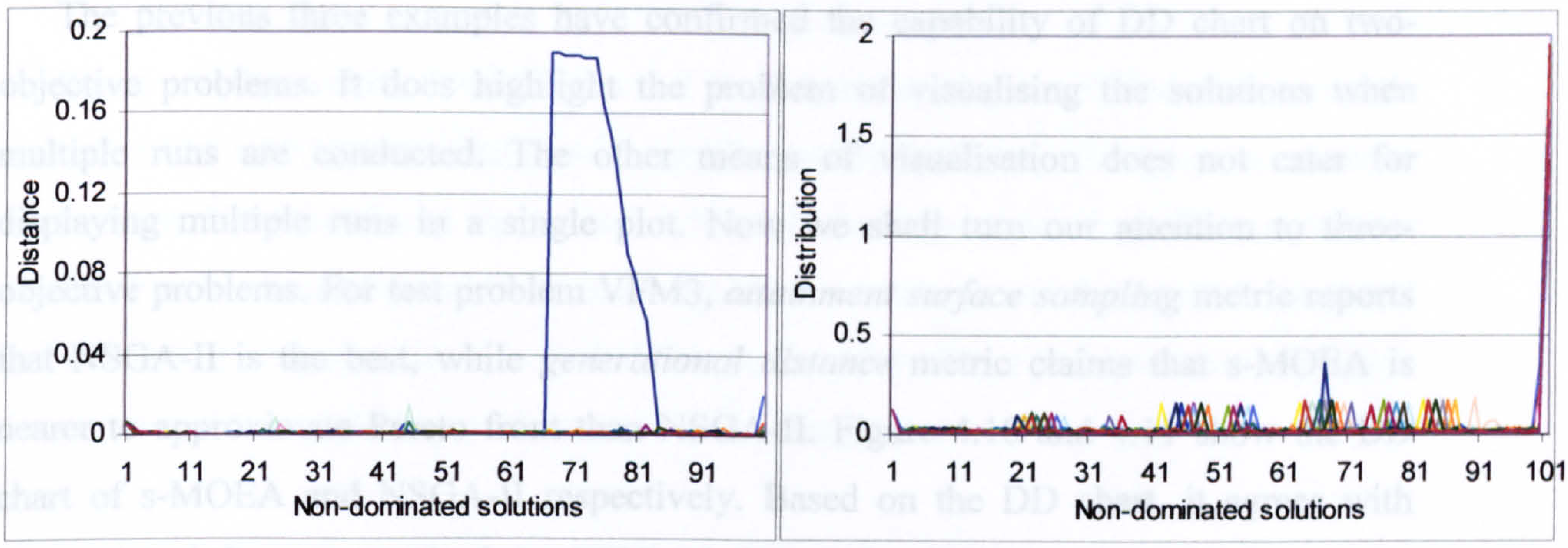

Figure 4.7 DD Chart of s-MOEA on Test Problem ZDT3

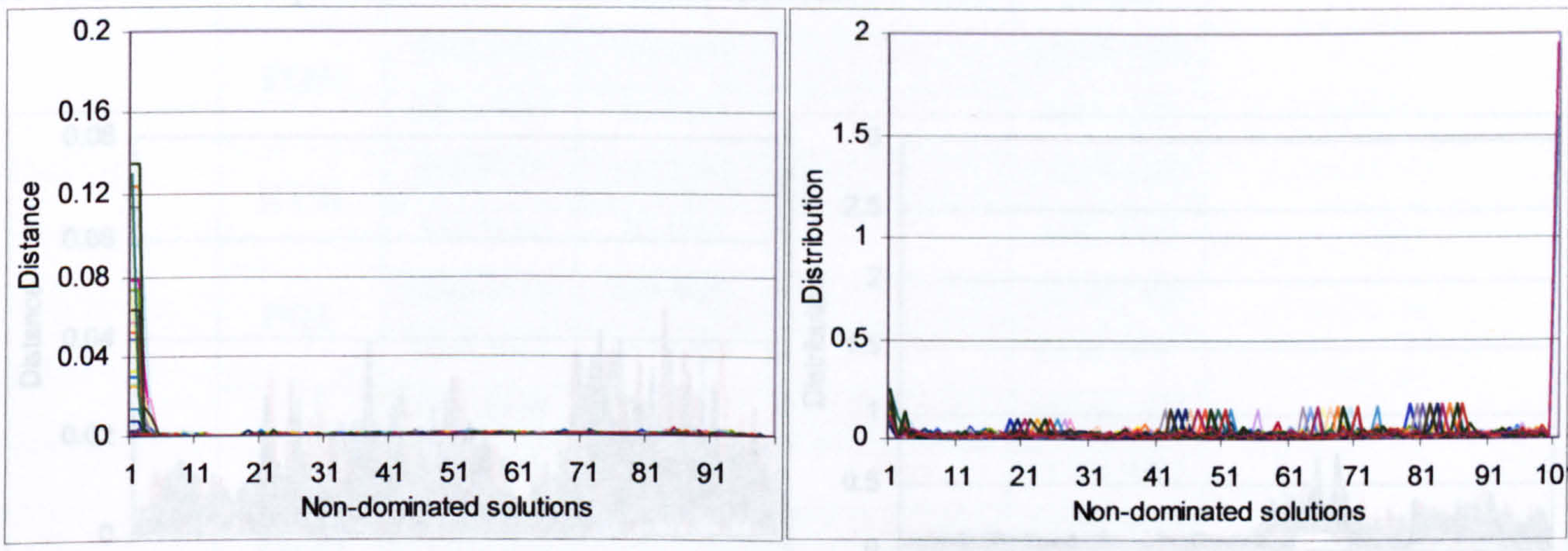

Figure 4.8 DD Chart of NSGA-II on Test Problem ZDT3

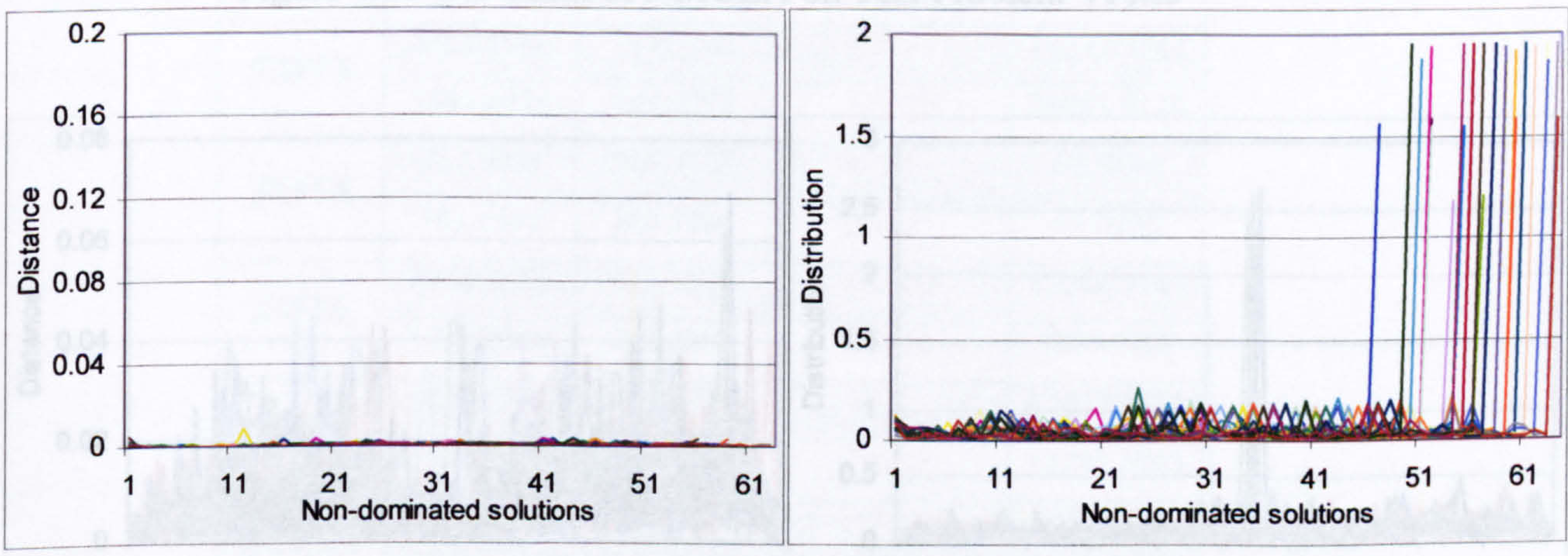

Figure 4.9 DD Chart of SPEA on Test Problem ZDT3

The previous three examples have confirmed the capability of DD chart on two-objective problems. It does highlight the problem of visualising the solutions when multiple runs are conducted. The other means of visualisation does not cater for displaying multiple runs in a single plot. Now we shall turn our attention to three-objective problems. For test problem VFM3, *attainment surface sampling* metric reports that NSGA-II is the best, while *generational distance* metric claims that s-MOEA is nearer to approximate Pareto front than NSGA-II. Figure 4.10 and 4.11 show the DD chart of s-MOEA and NSGA-II respectively. Based on the DD chart, it agrees with *generational distance* metric that s-MOEA is better than NSGA-II. Yet, it can be a bit difficult to decide which algorithm has better solutions diversity. However, DD chart can reveals more in-depth information than those metrics.

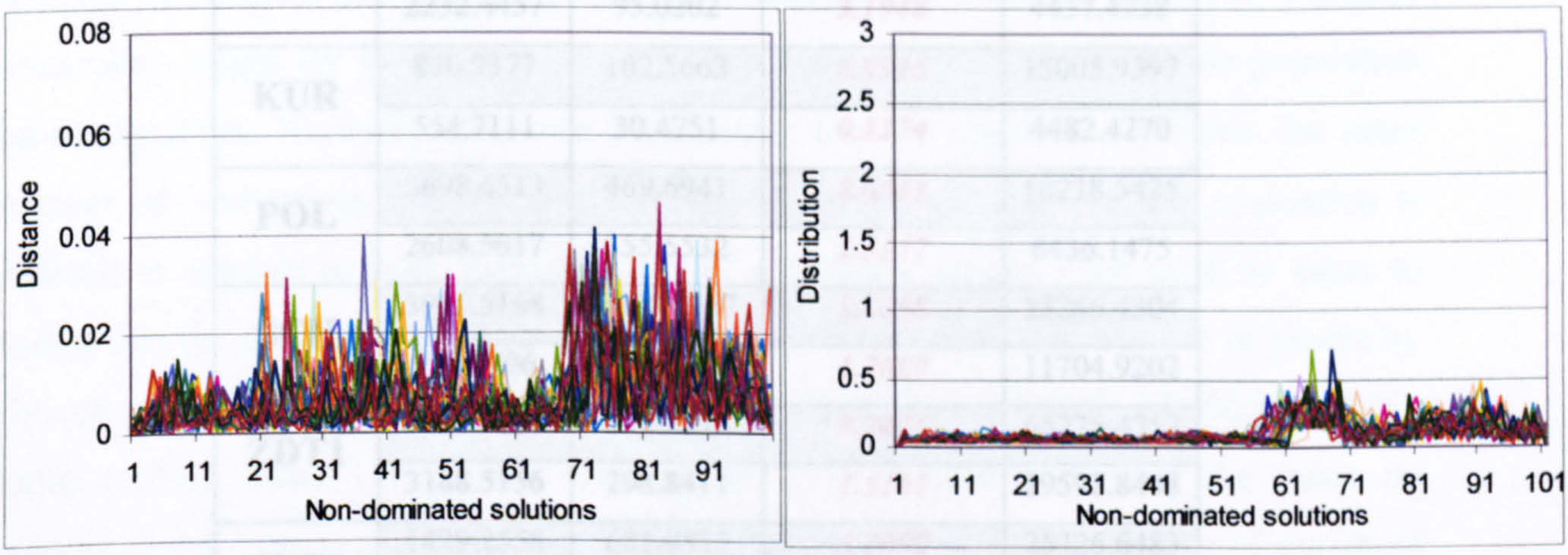

Figure 4.10 DD Chart of s-MOEA on Test Problem VFM3

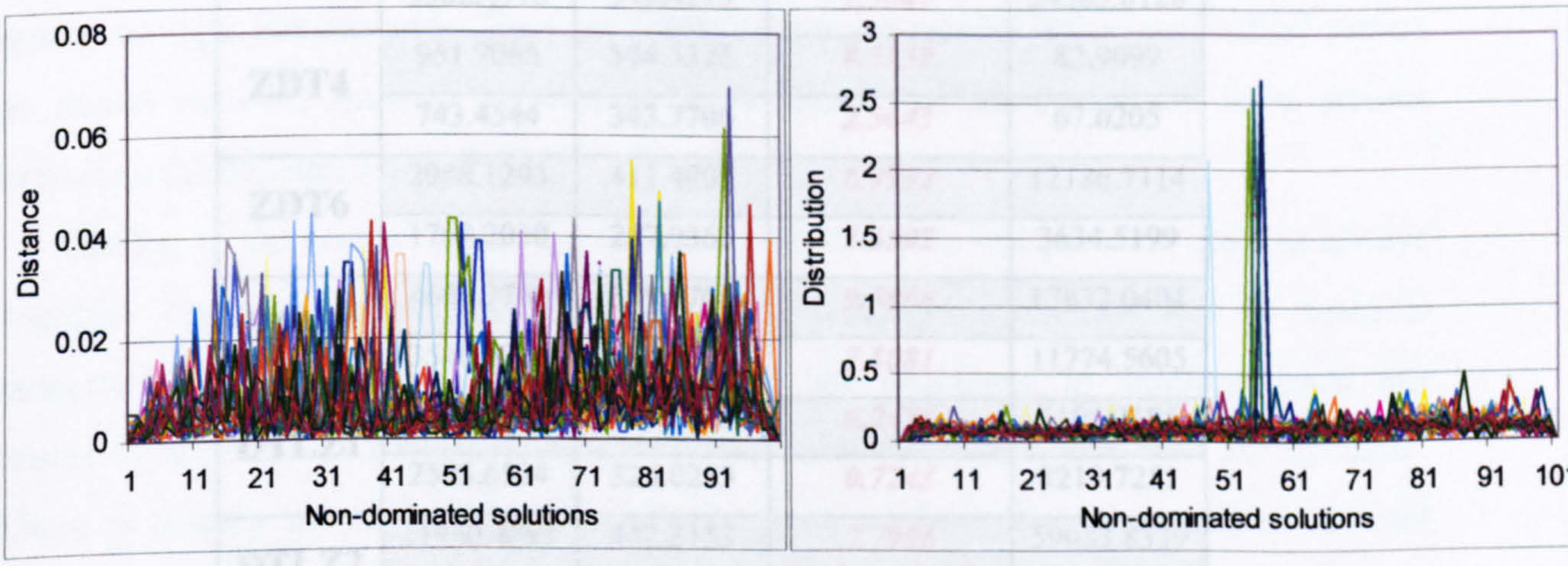

Figure 4.11 DD Chart of NSGA-II on Test Problem VFM3

Based on the results from Tables 4.2 to 4.4, it can be concluded that a simple algorithm like s-MOEA can still provide satisfactory performance. However, to conclude the result without looking at their overheads might be very misleading. Thus, optimiser overhead metric is used to measure each algorithm overheads and the result is tabulated in Table 4.5. It is obvious that (1+1)-PAES is the best for all test problems even though it does not provide satisfactory performance in the comparison tests.

Table 4.5 Mean (shaded rows) and Standard Deviation (unshaded rows) of Optimiser Overhead Metric. Best result is highlighted in red colour.

	s-MOEA	NSGA-II	(1+1)-PAES	SPEA
FON	4385.3072	391.2178	***9.7041***	17347.7573
	2232.4457	95.0202	***3.1948***	4437.4238
KUR	870.9377	102.5663	***6.9916***	15005.9397
	554.7111	30.4751	***0.8174***	4482.4270
POL	3698.6513	469.6941	***8.6031***	16218.5425
	2608.5017	355.5532	***1.3177***	6436.1475
SCH	3905.5108	1070.8267	***5.8638***	33266.4304
	958.0796	83.5866	***1.2609***	11704.9202
ZDT1	7497.2387	697.0142	***8.9476***	65725.4752
	3188.5156	298.8411	***1.5161***	29592.8448
ZDT2	1429.2558	681.9572	***8.0690***	25326.6483
	2463.1927	193.9280	***1.8917***	21379.5246
ZDT3	5428.9882	715.7142	***9.1948***	43279.7448
	3582.4976	349.4273	***1.7687***	29365.8128
ZDT4	951.9065	544.5376	***8.5132***	82.9099
	743.4544	343.7706	***2.5643***	67.0205
ZDT6	2988.1293	411.4909	***6.9932***	12186.7114
	1769.2030	257.0365	***1.5592***	3634.5199
VFM3	4604.7700	539.3706	***9.5864***	17832.0404
	3540.4028	431.9305	***7.5081***	11774.5605
DTLZ1	4594.5892	678.6545	***6.2454***	6105.7185
	2593.6524	325.0213	***0.7243***	3219.7251
DTLZ2	11940.4898	482.3153	***7.7906***	59033.8339
	7442.1533	295.5181	***1.3931***	32102.5625

DTLZ3	569.2981	215.3618	*6.0391*	105.7147
	479.7936	104.3715	*1.2626*	25.8734
DTLZ4	4984.4755	225.0213	*14.3220*	22078.7274
	4527.7621	70.7008	*22.4473*	10136.3161

Based on the results shown here, it is clear that some problems still exist that need to be resolved in existing metrics for evaluating MOEAs. The starting point for this discussion is that a compound problem exists in relation to MOEA comparison, i.e.

- There is no proper and formal way to evaluate a MOEA;
- The question of suitability of the available test problems suite; and
- Performance metrics.

In relation to the first issue, one of the most obvious questions is: how does one compare techniques that are designed in a fundamentally different way? For example, researchers might try to compare a population-based algorithm with a non-population based algorithm. Normally to ensure consistency, the two algorithms perform the same number of evaluations with similar settings. However, comparison and evaluation is difficult to conduct as both algorithms are of a different nature. Care must be taken to ensure that the evaluation process does not favour one algorithm. A number of papers by Greenberg (1990), Barr *et al.* (1995), Hooker (1995), L'Ecuyer (1996), McGeoch (1996), Orlin (1996), Shier (1996) and Gent *et al.* (1997) have addressed the issue on computational experimentation. That would require use of more formal tests of statistical significance as a way of introducing more scientific precision into empirical investigations of algorithms. Evidence to date shows that good evaluations are not done nearly enough. For example, Prechelt (1996) surveyed nearly 200 experimental papers on neural network learning algorithms. It is found that most of them have serious experimental deficiencies.

Turning to the second issue, the suitability of available test problem suite is always doubtful. There is always the possibility that a particular technique will be tuned to outperform the other techniques on a few ad hoc problems. In addition, most test problems are static and non-changing, whereas most real world problems are dynamic. There is usually no explanation or analysis given on how those researchers conduct testing on test problems relating to their performance in real world applications. One way of handling this issue is to have a test-problem generator that produces random problems of different definable characteristics similar in nature with those proposed by De Jong *et al.* (1997), Kennedy and Spears (1998), Grefenstette (1999), Morrison and De Jong (1999), Michalewicz *et al.* (2000) and Schmidt and Michalewicz (2000). This may reduce the possibility of biased comparison since the test problems are random. However, it is not an easy task to produce a multi-objective test problem generator.

Finally, another problem surrounding MOEA performance evaluation is credibility. Since there is no widely accepted performance metric, it is very difficult to convince anyone on the evaluation results yielded. This has severely hindered the progress of MOEA technique development. The following highlight problems surrounding existing performance metrics:

- They are dependent on true Pareto front which is almost impossible to generate in some cases;
- Misleading information is common as the result does not tally with what is shown visually;
- Results returned from the performance metrics are not informative enough;
- Performance metrics do not reveal any other information about the technique other than showing whether the technique has won or not, which is not helpful;
- The methodology of the performance metric is over-complicated; and
- It is computationally intensive.

Based on the above listed problems of performance metrics, suggestions have been set out below for consideration on those problems.

1. Performance metrics which are dependent on true Pareto front does not appear to be a popular choice by researchers due to the vast amount of effort needed to use them. Some possible solutions are to have a repository where the true Pareto front for all available test problems can be easily accessed by researchers. Clearly this is only a tentative solution, as true Pareto front is almost impossible for evaluation in real world applications. Alternatively, researchers can use approximate Pareto front (Ang *et al.*, 2001) instead of the true Pareto front. This approximate Pareto front can be obtained by extracting all the non-dominated solutions from the competing algorithms. Of course, one can avoid those metrics altogether and use those that are not dependent on true Pareto front. *Size of space covered* metric is suggested in this

case. Another advantage of using the ***size of space covered*** metric is that the result can be used to visualise an algorithm convergence velocity that quantifies relative or absolute convergence improvement. The metric can even be integrated into a MOEA technique as a termination criterion. However, it is not indicative enough as it incorporates two criteria: distance and distribution. Therefore, solutions differing in one criterion may not be distinguished from the other. Hence, it is recommended to use this together with measures like ***coverage of two sets*** or ***attainment surface sampling***, which can roughly reveals the non-dominance coverage of the competing solutions.

2. The problem of misleading information might be very difficult to resolve. As the chosen algorithms might produce very different results, it is important to have information about the decision makers' preferences. Such objective information (as stated in Hansen and Jaszkiewicz (1998)) in relation to the human decision-making would enable one to evaluate the best by incorporating some weak assumptions about the decision makers' preferences. Another possible solution is to use the DD chart. Its advantages have been repetitively shown in the comparison tests.

Hence the focus of this chapter has been placed on performance metrics. Clearly, an MOEA is a technique that is supposed to be customised for each application in order to achieve optimal performance and not a technique that will provide optimal performance for all types of applications based on No Free Lunch (NFL) theorems (Wolpert and Macready, 1995; Wolpert and Macready, 1997). This simulation study shows that each algorithm has their own unique attribute and being able to appreciate their attribute and make good use of them is the main point. For example, we could use (1+1)-PAES to generate the initial best population, as this algorithm is very fast.

4.7 Summary

Based on the performance metrics and visualisation results, it shows that s-MOEA performance is comparable with those commonly cited evolutionary algorithms. Although the result of this comparison is sometimes a multi-objective problem situation, however s-MOEA still proves it worth in term of it capability of producing well-diverse and nearly optimal Pareto set, based on its simple architecture. This should prove useful in the search and optimisation of a multi-objective PID tuning rule.

It is evident that more work is required on improving the existing performance metrics for MOEAs. A set of widely accepted performance metrics for use in MOEA evaluation is not so simple and straightforward. Whether it is possible to have such a set of performance metrics that would produce an accurate or distinguishable evaluation of a MOEA, is still unknown. Hence, the visualisation technique proposed in Section 4.5 aims to alleviate the problems of inaccurate evaluation and assist in the final selection of a suitable MOEA. It is important to note that without proper and widely accepted performance metrics, it is very difficult to quantify test results or to guide the development of smart MOEAs. It is therefore in the spirit of both scientific inquiry and pragmatic investigation that this chapter aims to call upon, for a more detail studies on developing and improving the performance metrics for MOEA performance evaluation. To conclude, we would like to quote an interesting note by Hooker (1995):

"*It asks that experimental results be evaluated on the basis of whether they contribute to our understanding, rather than whether they show that the author's algorithm can win a race with the state of the art. It asks scholarly journals to publish studies of algorithms that are miserable failures when their failure enlightens us.*"

Chapter 5

Search for Globally Optimal Multi-Objective PID Tuning Rules

Chapter objectives

This chapter introduces the methodology behind the search for globally optimal multi-objective PID tuning rules.

5.1 Introduction

Based on the introduction and analysis of Chapter 2 and 3, it shows that there exist a number of PID control and tuning techniques, which differ in complexity, flexibility and amount of process knowledge required. Optimal performance, traditionally defined against one objective, is obtained through using an optimiser in the design process. However, there is still a lack of simple, easy to use and intuitive design method that does not require the availability of comprehensive process information and yet delivers highly satisfactory performance. In addition, the performance that these design and tuning techniques target to offer is restricted to one major objective, such as either set-point tracking or load disturbance rejection, but not both. Worse still, present design techniques can hardly incorporate an independent penalty upon control energy requirements or the rate of change of the control signal. Hence, in delivering such performance, the controller obtained can often lead to excess chattering and wearing of actuators.

Therefore, there is a need to consider multiple objectives as listed in Section 2.2.7. One solution taking into account of multiple considerations in design is the use of a weighting factor, as found in conventional optimal control. Whilst this appears effective in some cases, the selection of a weighting factor proves to be a non-trivial or even intractable exercise in practice. During the past decade, research into evolutionary computing, most visibly represented by genetic algorithms, and its application in systems and control engineering have made remarkable progress. This has enabled design automation and structural model fitting for control systems (Li *et al.*, 1996; Tan and Li, 1997).

In this chapter, the search for truly multi-objective PID tuning rules through the use of evolutionary computing technique are detailed.

5.2 Development of PIDeasy Tuning Method

Based on the studies of Chapter 2 and 3, it is apparent that a prime problem to PID technology is their modified structure that has been complicated beyond their original beauty. The main difference is mostly on the derivative part and some minor tweaks to the PID algorithm. This really has complicated the whole process of tuning a PID controller. Therefore, the aim of this development is to search for a multi-optimal tuning rule that can work across a range of PID structures using evolutionary computation technique. The success of this development can greatly reduce the user's burden and enhance efficiency.

Initial attempt to solve this problem started in 1995, where the genetic programming approach is being considered (Ang, 1996). However, due to the enormous search space and limited time, satisfactory performance are found for a limited operating range. This finding in turn encourages and leads to the development of the first generation PIDeasy (Li *et al.*, 1998). PIDeasy is however limited to critically- or over-damped process behaviour. The modelling technique is based on a first-order plus delay plant model, which limits its operating range. Thus, this research attempts to use a general second-order plus delay plant model where it can capture any type of behaviour from under-damped to over-damped process. This in turn can make the tuning rules very general and applicable to a wide operating range.

The extensive analysis given in Chapter 3 indicated the complexity of the problem. It is sometimes impossible to combine all the objectives together as they are often incommensurable and competing against each other. Hence, it is apparent that classical methods are not suitable and optimal for this type of problem. Therefore, MOEA seems to be a natural choice. It is used to perform the search process of achieving optimal compromised performance for tuning a PID controller over a wide operating range.

The relationships given between the optimal controller settings and the process parameters are not based on theoretical considerations such as pole cancellation or model matching. They are empirical rules developed as the result of hundreds of simulations; the concept employed here is a balance of optimal performance between servo (set-point tracking) and regulator (load disturbance rejection) problems without having an excessive high gains. This can provide a reasonable initial tuning for user and therefore guarantees robustness without sacrificing performance. The results obtained for a particular class of process and type of controller, were plotted against the process parameters, and curves were fitted to connect the data points. Polynomial equation is used to fit the data. Rather than presenting the data in tables, the fitted equations are shown, as this would be the most useful form to represent such a large compilation of data.

5.2.1 Modelling

Processes with time delay may be modelled in various ways. The modelling strategy used will influence the value of the model parameters, which will in turn affect the controller values determined from the tuning rules. From Chapter 3, it is clear that regardless of how intelligent the PID tuning rule is, a classical step input to the process is needed in order to guess or compute the initial values for PID controller. This type of identification is simple to use and easily understood by the user. One of the outstanding professionals in process control industry honour by CONTROL's Process Automation Hall of Fame in February 2001 (CONTROL Magazine, 2004), Lipták (2001) stated that frequency domain is better left to mathematicians and the process control engineers should do their routine tuning in the time domain as that is the domain they live in and understand. Thus, this research solely concentrates on time-domain development and aims to be easily applicable to most users.

The two process models that are commonly used to approximate the dynamics of an industrial process are used. They are the first-order lag plus delay (FOLPD) model and second-order system plus delay (SOSPD) model, as shown below respectively:

$$G_{P1}(s) = \frac{K}{Ts+1}e^{-Ls} \tag{5.1}$$

$$G_{P2}(s) = \frac{K}{T^2s^2 + 2T\zeta s + 1}e^{-Ls} \tag{5.2}$$

where K is the process gain, T is the process time constant, ζ is the damping factor and L is the process dead-time or transport delay.

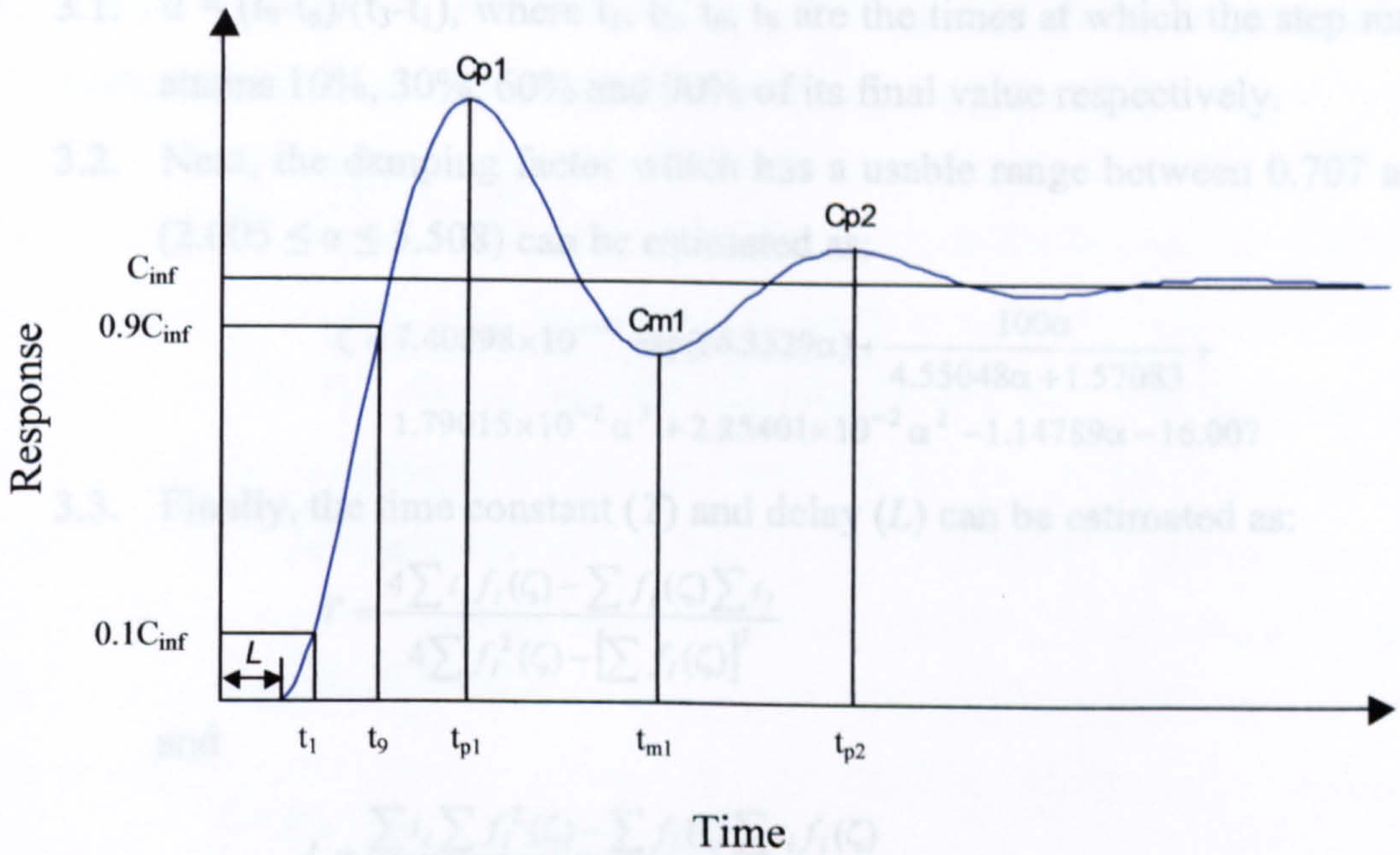

Figure 5.1 Typical Under-Damped Step-Response Curve

The plant identification method for FOLPD model is based on the classical reaction curve method (Åström and Hägglund, 1995). As for SOSPD model, the plant identification method used is a combination of the approximation methods proposed by Chen (1989), Huang and Huang (1993) and Huang and Chou (1994) so that it can cover a range of damping factor from 0 to 3. Simplicity is the main advantage of these approximation methods.

Based on Huang and Chou (1994), the damping factor (ζ) can be easily estimated using the maximum overshoot (M_p) of the step-response curve as:

$$\zeta = \sqrt{\frac{\ln^2(M_P)}{\pi^2 + \ln^2(M_P)}} \tag{5.3}$$

where $M_p = (C_{p1}-C_{inf})/C_{inf}$ (Figure 5.1). Accordingly, a systematic technique based on M_p as an index is used for estimating the SOSPD model parameters for $0 < \zeta \leq 3.0$ from step-response data:

1. Calculate the process gain (K) by dividing the steady-state output change by the input change;
2. Calculate M_p;
3. If C_{p1} is indefinite, say $M_p < 1.5\%$ (i.e., $\zeta > 0.8$) or so, the model parameters (ζ, T and L) are then estimated by the following steps (Huang and Huang, 1993):

3.1. $\alpha = (t_9-t_6)/(t_3-t_1)$, where t_1, t_3, t_6, t_9 are the times at which the step response attains 10%, 30%, 60% and 90% of its final value respectively.

3.2. Next, the damping factor which has a usable range between 0.707 and 3.0 ($2.005 \le \alpha \le 5.508$) can be estimated as:

$$\zeta = 7.40898\times10^{-40}\exp(16.3329\alpha) + \frac{100\alpha}{4.55048\alpha + 1.57083} + 1.79015\times10^{-2}\alpha^3 + 2.25401\times10^{-2}\alpha^2 - 1.14789\alpha - 16.007 \tag{5.4}$$

3.3. Finally, the time constant (T) and delay (L) can be estimated as:

$$T = \frac{4\sum t_i f_i(\zeta) - \sum f_i(\zeta)\sum t_i}{4\sum f_i^2(\zeta) - [\sum f_i(\zeta)]^2} \tag{5.5}$$

and

$$L = \frac{\sum t_i \sum f_i^2(\zeta) - \sum f_i(\zeta)\sum t_i f_i(\zeta)}{4\sum f_i^2(\zeta) - [\sum f_i(\zeta)]^2} \tag{5.6}$$

where

$$\sum t_i = t_1 + t_3 + t_6 + t_9 \tag{5.7}$$

$$\sum f_i(\zeta) = f_1(\zeta) + f_3(\zeta) + f_6(\zeta) + f_9(\zeta) \tag{5.8}$$

$$\sum f_i^2(\zeta) = f_1^2(\zeta) + f_3^2(\zeta) + f_6^2(\zeta) + f_9^2(\zeta) \tag{5.9}$$

$$\sum t_i f_i(\zeta) = t_1 f_1(\zeta) + t_3 f_3(\zeta) + t_6 f_6(\zeta) + t_9 f_9(\zeta) \tag{5.10}$$

$$f_1(\zeta) = 0.45465 + 0.06033\zeta + 0.01674\zeta^2 \tag{5.11}$$

$$f_3(\zeta) = 0.848967 + 0.071809\zeta + 0.19753\zeta^2 - 0.021823\zeta^3 \tag{5.12}$$

$$f_6(\zeta) = 1.08111 + 0.40977\zeta + 0.634313\zeta^2 - 0.093324\zeta^3 \tag{5.13}$$

$$f_9(\zeta) = 0.581618 + 0.875726\zeta + 3.64626\zeta^2 - 1.35143\zeta^3 + 0.173916\zeta^4 \tag{5.14}$$

4. If $1.5\% \le M_p \le 25\%$ (i.e., $0.4 \le \zeta \le 0.8$), then ζ can be computed using Equation 5.3, and T and L using Equations 5.5 and 5.6 but with different computation for $f_1(\zeta)$, $f_3(\zeta)$, $f_6(\zeta)$ and $f_9(\zeta)$ as shown below:

$$f_1(\zeta) = 0.451465 + 0.066696\zeta + 0.013639\zeta^2 \tag{5.15}$$

$$f_3(\zeta) = 0.800879 + 0.194550\zeta + 0.101784\zeta^2 \tag{5.16}$$

$$f_6(\zeta) = 1.202664 + 0.288331\zeta + 0.530572\zeta^2 \tag{5.17}$$

$$f_9(\zeta) = 1.941112 - 1.237235\zeta + 3.182373\zeta^2 \tag{5.18}$$

5. If $M_p > 25\%$ (i.e., $\zeta < 0.4$), then the following steps should be adopted (Chen, 1989) for the estimation of the model parameters (ζ, T and L):

$$C_\beta = \frac{C_{p1}C_{p2} - C_{m1}^2}{C_{p1} + C_{p2} - 2C_{m1}} \tag{5.19}$$

$$M'_p = \frac{1}{3}\left[\frac{C_{p1} - C_\beta}{C_\beta} + \frac{C_\beta - C_{m1}}{C_{p1} - C_\beta} + \frac{C_{p2} - C_\beta}{C_\beta - C_{m1}}\right] \tag{5.20}$$

$$\zeta = \frac{-\ln(M'_p)}{\sqrt{\pi^2 + \ln^2(M'_p)}} \tag{5.21}$$

$$T = \frac{(t_{m1} - t_{p1})\sqrt{1-\zeta^2}}{\pi} \tag{5.22}$$

$$L = 2t_{p1} - t_{m1} \tag{5.23}$$

5.2.2 PID Structure

The development of PIDeasy tuning rule is based on the ideal PID structure (2.1) and simple anti-windup scheme (2.8). The main reason is there are many tuning rules based on this structure and thus the performance can be compared easily. However, PIDeasy tuning rule is developed not only to support the ideal PID structure but also to support as many other available structures as possible. The ultimate aim is to reduce the burden of the user without having to make sure the controller structure suits the tuning rule or vice versa.

Modern design techniques have been helped tremendously by powerful simulation packages that are capable of taking saturations of the integrator into account during the design process. Hence, in the design we shall accommodate the treatment of saturation often found in PID practice.

5.2.3 Optimisation Objectives

The ultimate goal of a PID controller is to reduce the error between the process output and reference input. This needs to be satisfied under plant and environmental uncertainties, which is impossible in practical control system design due to control signal or actuator saturation (for e.g., voltage limit) and constraints on the rate of change of the control signal (for e.g., current limit). In fact, should there be no error regardless of time and frequency, the feedback system would become open-loop and this in turn would not guarantee a zero error with the presence of disturbance or model uncertainty.

Hence, a performance index (or fitness function in the context of evolutionary computation) must be devised to measure how close the actual performance is from the expected performance. For this, the performance indices and specifications need to reflect the qualitative specification requirements detailed in Section 2.2.7. Performance indices shall reflect all specifications that need to be considered in practice. They can be in the form of an overall composite objective or cost function, as commonly adopted by conventional optimisation technique. They can also, preferably, be in the form of multiple independent criteria, which can be handled easily and efficiently by evolutionary computation.

Thus, the optimisation objectives considered are the multiple costs in terms of the integral of time-weighted absolute error (ITAE) for both set-point tracking and load disturbance rejection performances and the rate of change in the control signal. It can be seen that only time-domain specifications are being considered here, as it can be argued that by minimising the error, it will indirectly lead to good stability margins (Li *et al.*, 2004). By just considering the two main objectives, it can also satisfy the other objectives that will be shown in the next chapter.

While most of the tuning rules are optimised based on load disturbance rejection performance, however this will lead to a rather oscillating response to set-point changes. The oscillating response can nevertheless be resolved by having a set-point filter. Moreover, simply based on load disturbance rejection performance, it will lead to poor stability margins and less robustness due to modelling error (which are illustrated in the Section 6.4). Thus, the aim of this design is to achieve optimal compromised performance based on the simplest structure.

5.2.4 Optimisation Process

In contrast with conventional optimisation algorithms, EAs can search for globally optimised solutions and can do so without the need for the existence of a derivative of the index or cost function. Furthermore, they can simultaneously deal with multiple objectives and hence require no weighting factors between competing performance indicators. All resulting 'non-dominant' solutions will form a so-called 'Parcto front', on which one solution is not dominantly 'better off' than another solution and hence on which all solutions are the 'best' in meeting multiple objectives. In this way, the user is being presented with a set of best-compromised solutions and the information on the trade-off, thus enable them to subjectively choose a suitable solution. Assuming the solutions have met the listed criteria, the final solution on each operating point chosen is based on the criteria of consistent gain and phase margin across a wide operating range. In order to ensure best optimal performance, the solutions found are compared against some selected tuning rules. This is done by displaying individual graphs of the solutions and selected tuning rules performance on each objective before making the final decision (Ang *et al.*, 2003).

Starting with FOLPD model (see (5.1)), the terms K_P, T_I, and T_D are optimised against the normalised delay, L/T. Based on each normalised delay ranging from 0.01 to 1000.0, the best compromised settings for the three terms are searched. This is a modification from PIDeasy™ (Li *et al.*, 1998), as the original tuning method is too aggressive and thus it is can be risky in the presence of modelling error. Corripio (2001) states that, when L/T is less than 0.1, most tuning rules tend to have impractical high gains and fast integral times. Thus, considering that, the only term that needs to be change from the original tuning rule is the proportional gain, K_P. The result is shown in Figure 5.2.

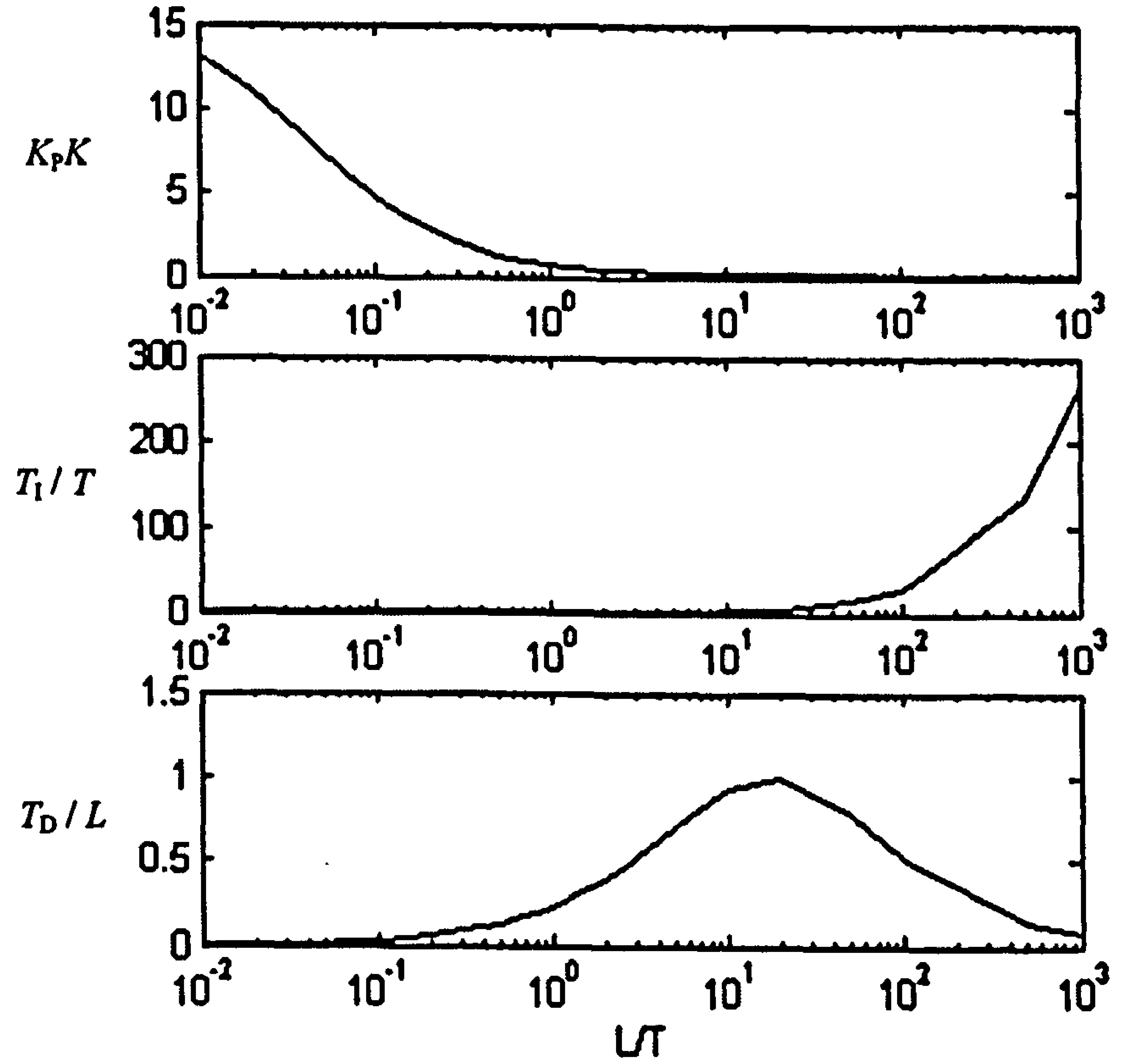

Figure 5.2 Optimised Values of K_P, T_I and T_D for FOLPD Model

As for the SOSPD model (see (5.2)), in addition to the normalised delay, the damping factor also needs to be considered in the search. Now based on each value of the damping factor, the normalised delay is scanned from 0.01 to 1000.0, and the best compromised settings for the three terms are searched for. This is a complex multi-level three-dimensional search, which can be very difficult for conventional optimiser. The result is shown in Figure 5.3.

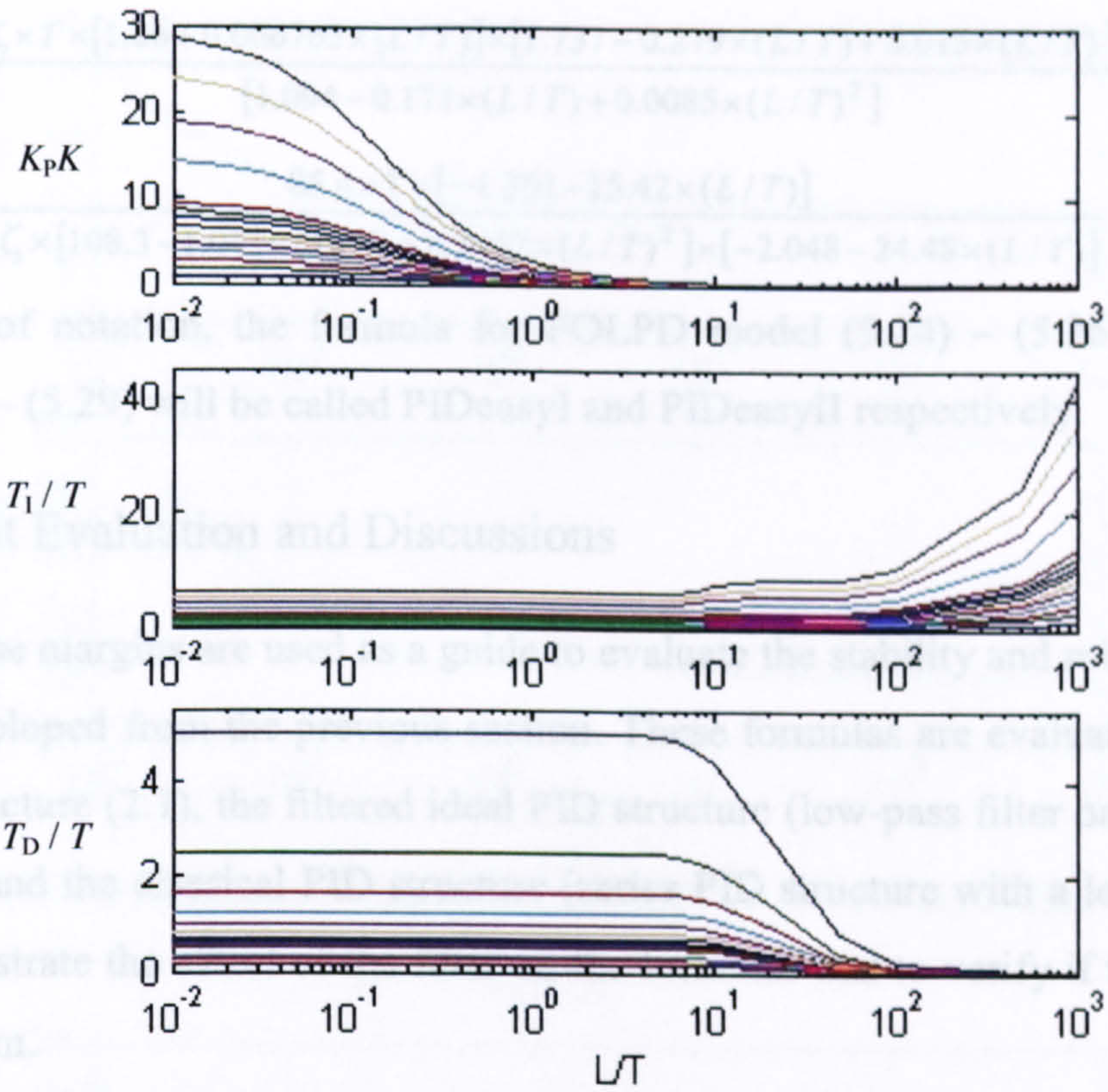

Figure 5.3 Optimised Values of K_P, T_I and T_D on Various Value of ζ for SOSPD Model

The target for the tuning rule is a formula-based similar to the ZN tuning rule. Thus, it can be easily implemented together with the modelling into a pocket calculator or a personal computer. The next step is to translate the results into a formula so that it can be easily computed. A curve fitting of the data onto a formula is being applied. The formula uses a polynomial structure and below is the result for FOLPD model:

$$K_P = \frac{0.652\times[1.0+1.1\times(L/T)+0.25\times(L/T)^2]}{K\times[0.04+1.02\times(L/T)+1.05\times(L/T)^2]} \quad (5.24)$$

$$T_I = \frac{T\times[1.0+1.18\times(L/T)]\times[1.0+1.4\times(L/T)+0.3\times(L/T)^2]}{[1.0+2.3\times(L/T)+1.33\times(L/T)^2]} \quad (5.25)$$

$$T_D = \frac{0.32\times L\times[1.0+1.03\times(L/T)]}{[1.0+1.6\times(L/T)+0.09\times(L/T)^2]\times[1.0+0.05\times(L/T)]} \quad (5.26)$$

The formula for SOSPD model is also using the same structure as FOLPD model but with an additional term, the damping factor (ζ). The result for SOSPD model is:

$$K_P = \frac{\zeta\times0.66\times[0.4022+1.251\times(L/T)+0.0095\times(L/T)^2]}{K\times[0.0267+0.1932\times(L/T)+0.99\times(L/T)^2]} \quad (5.27)$$

$$T_{\mathrm{I}} = \frac{\zeta \times T \times [1.26 + 0.006765 \times (L/T)] \times [1.737 - 0.279 \times (L/T) + 0.015 \times (L/T)^2]}{[1.094 - 0.171 \times (L/T) + 0.0085 \times (L/T)^2]} \quad (5.28)$$

$$T_{\mathrm{D}} = \frac{85.4 \times T \times [-1.301 - 15.42 \times (L/T)]}{\zeta \times [108.3 - 1.015 \times (L/T) + 0.2452 \times (L/T)^2] \times [-2.048 - 24.48 \times (L/T)]} \quad (5.29)$$

For ease of notation, the formula for FOLPD model (5.24) – (5.26) and SOSPD model (5.27) – (5.29) will be called PIDeasyI and PIDeasyII respectively.

5.2.5 Result Evaluation and Discussions

Gain and phase margins are used as a guide to evaluate the stability and robustness of the formulas developed from the previous section. These formulas are evaluated against the ideal PID structure (2.1), the filtered ideal PID structure (low-pass filter on differentiator term (2.16)) and the classical PID structure (series PID structure with a low-pass filter). This is to illustrate the effect of the filter on the formulas and to verify if there is a need to modify them.

Assuming that there is no modelling error when using the two models (5.1) and (5.2) on any process, the performance of PIDeasyI is evaluated based on FOLPD model by fixing the values of K and T and varies the L value from 0.01 sec. to 1000 sec. Figures 5.4 to 5.6 show the gain and phase margins of PIDeasyI on non-filtered ideal PID structure, filtered ideal PID structure with β = 3, 10, 20 and 30, and classical PID structure with β = 3, 10, 20 and 30, respectively. It is evident that the filter acting on differentiator, in this case, has only minimal effect on the overall performance. This is mainly because tuning rules based on FOLPD model always have a small derivative action. A typical first-order plant normally does not really require a differentiator as compared to second-order plant. All the different PID structures mainly differ in the derivative action, thus tuning rule based on FOLPD model is almost immune to the differences.

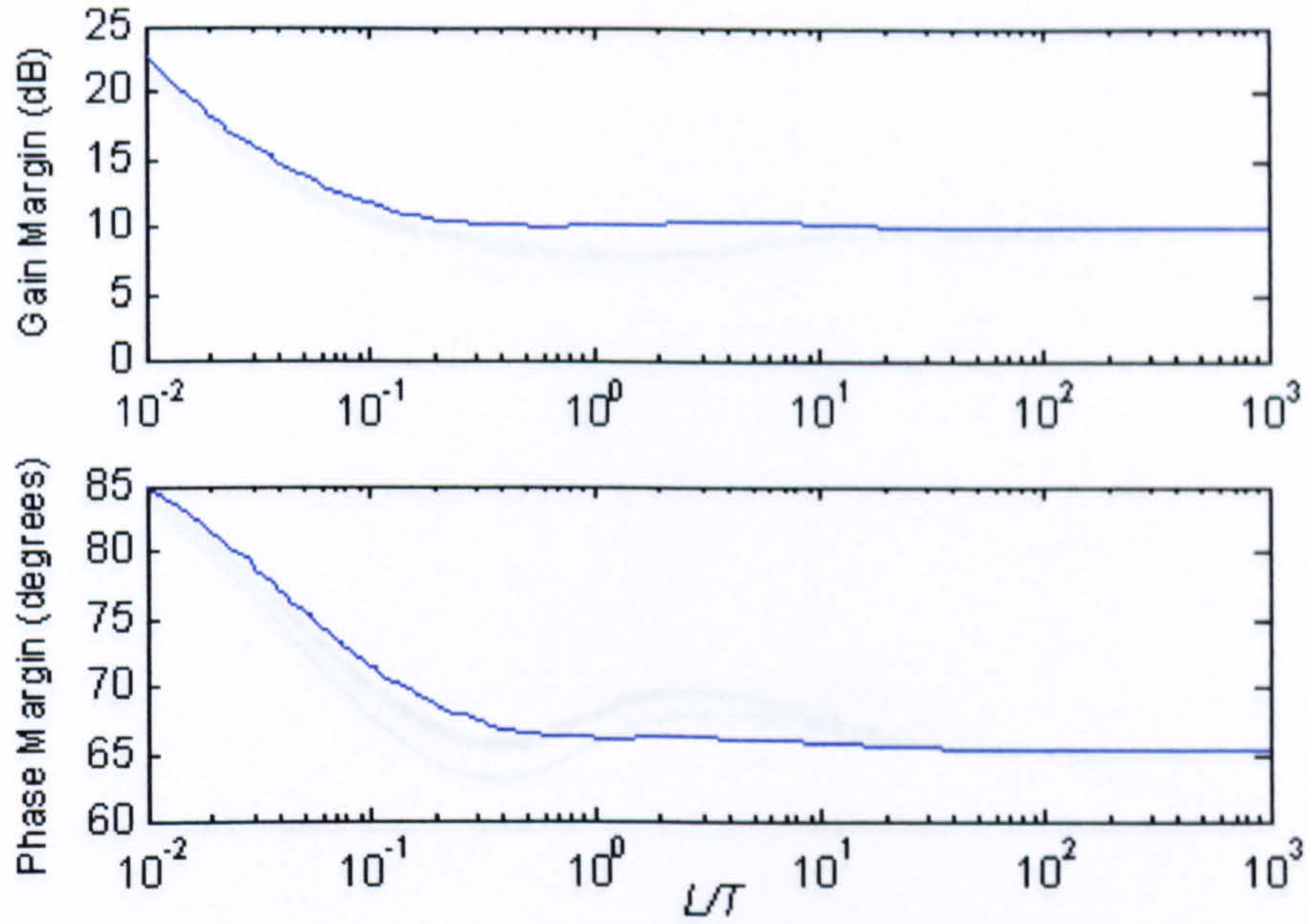

Figure 5.4 Gain and Phase Margins of PIDeasyI on 'Ideal Form' PID Controller

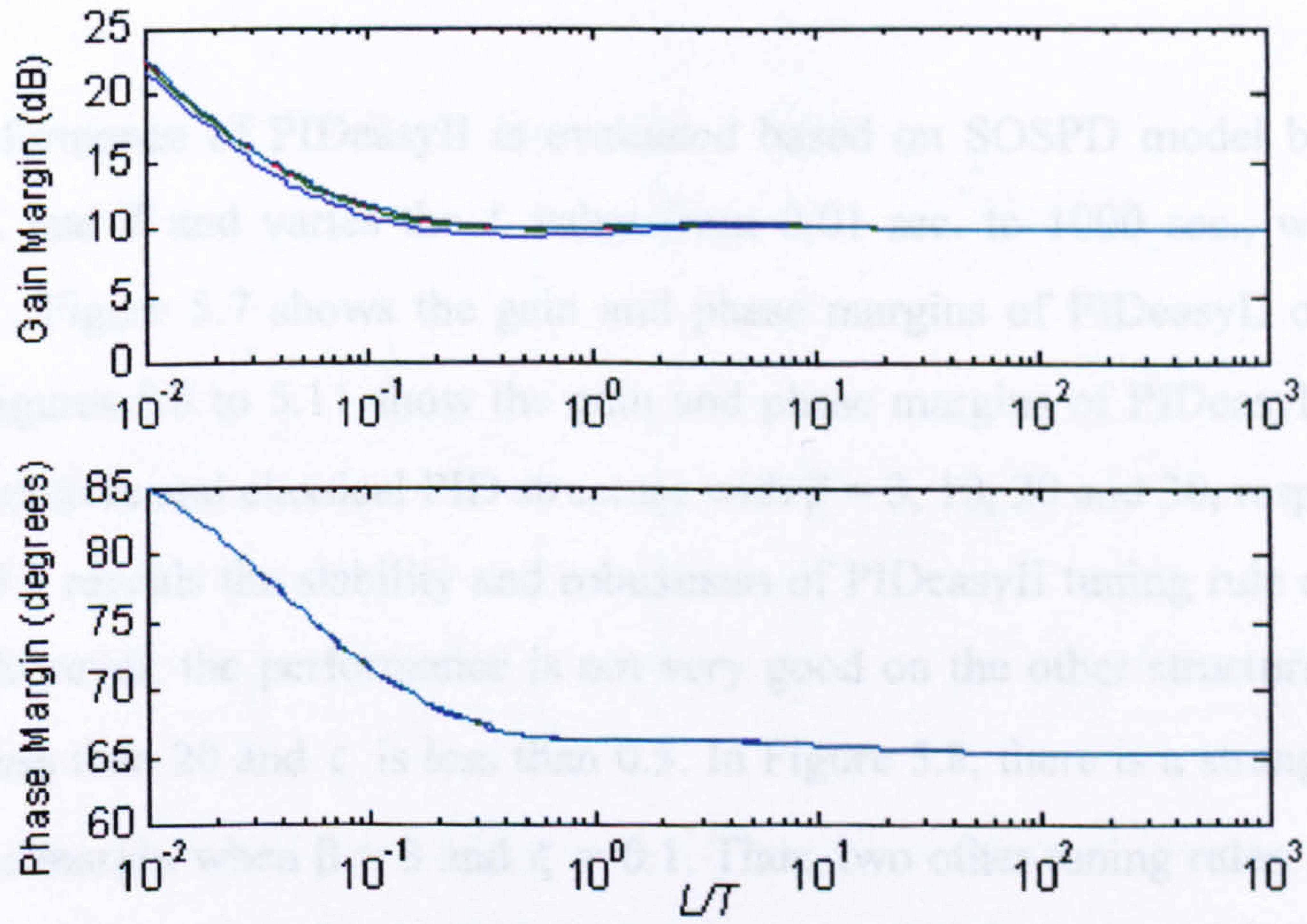

Figure 5.5 Gain and Phase Margins of PIDeasyI on Filtered 'Ideal Form' PID Controller with β = 3, 10, 20, 30

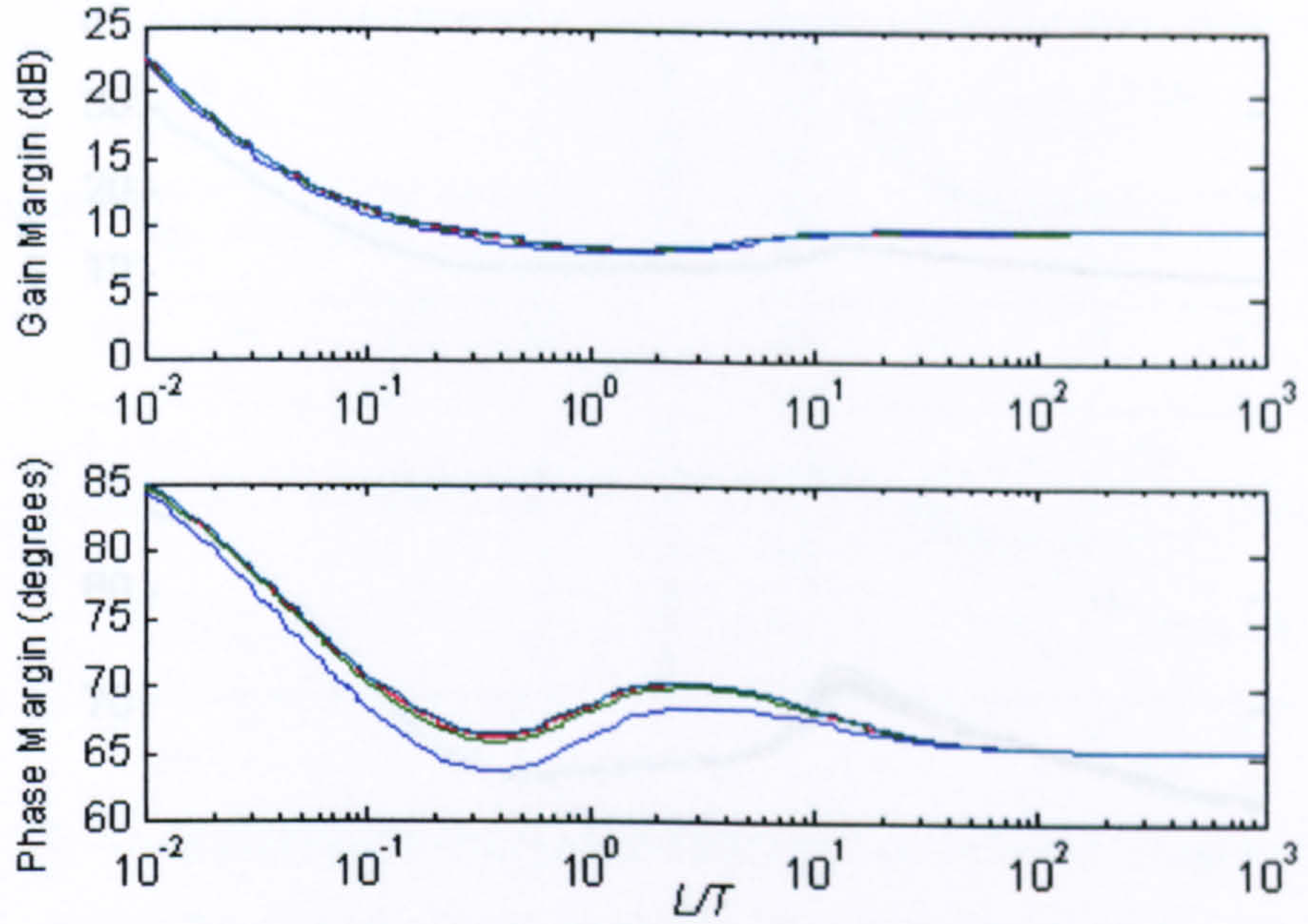

Figure 5.6 Gain and Phase Margins of PIDeasyI on Classical PID Controller with $\beta = 3$, 10, 20, 30

The performance of PIDeasyII is evaluated based on SOSPD model by fixing the values of K and T and varies the L value from 0.01 sec. to 1000 sec., with different values of ζ. Figure 5.7 shows the gain and phase margins of PIDeasyII on ideal PID structure. Figures 5.8 to 5.11 show the gain and phase margins of PIDeasyII on filtered ideal PID structure and classical PID structure with $\beta = 3$, 10, 20 and 30, respectively.

Figure 5.7 reveals the stability and robustness of PIDeasyII tuning rule on ideal PID structure. However, the performance is not very good on the other structures especially when β is less than 20 and ζ is less than 0.5. In Figure 5.8, there is a strange behaviour on the phase margin when $\beta = 3$ and $\zeta = 0.1$. Thus, two other tuning rules, proposed by G-K (Gorez and Klàn, 2000; O'Dwyer, 2003) and W-C (Wang and Clements, 1995; O'Dwyer, 2003), are used to verify this phenomenon. The results on the filtered ideal PID structure with $\beta = 3$ is shown in Figure 5.12. There is a tuning factor in W-C tuning rule. Stable performance is found by fixing it to 0.1 (robust but sluggish) for all values of ζ, especially when $\zeta = 0.1$. The objective here is not to show the optimise performance for each tuning rule, so no attempt is made to find the best tuning factor for W-C tuning rule for each ζ.

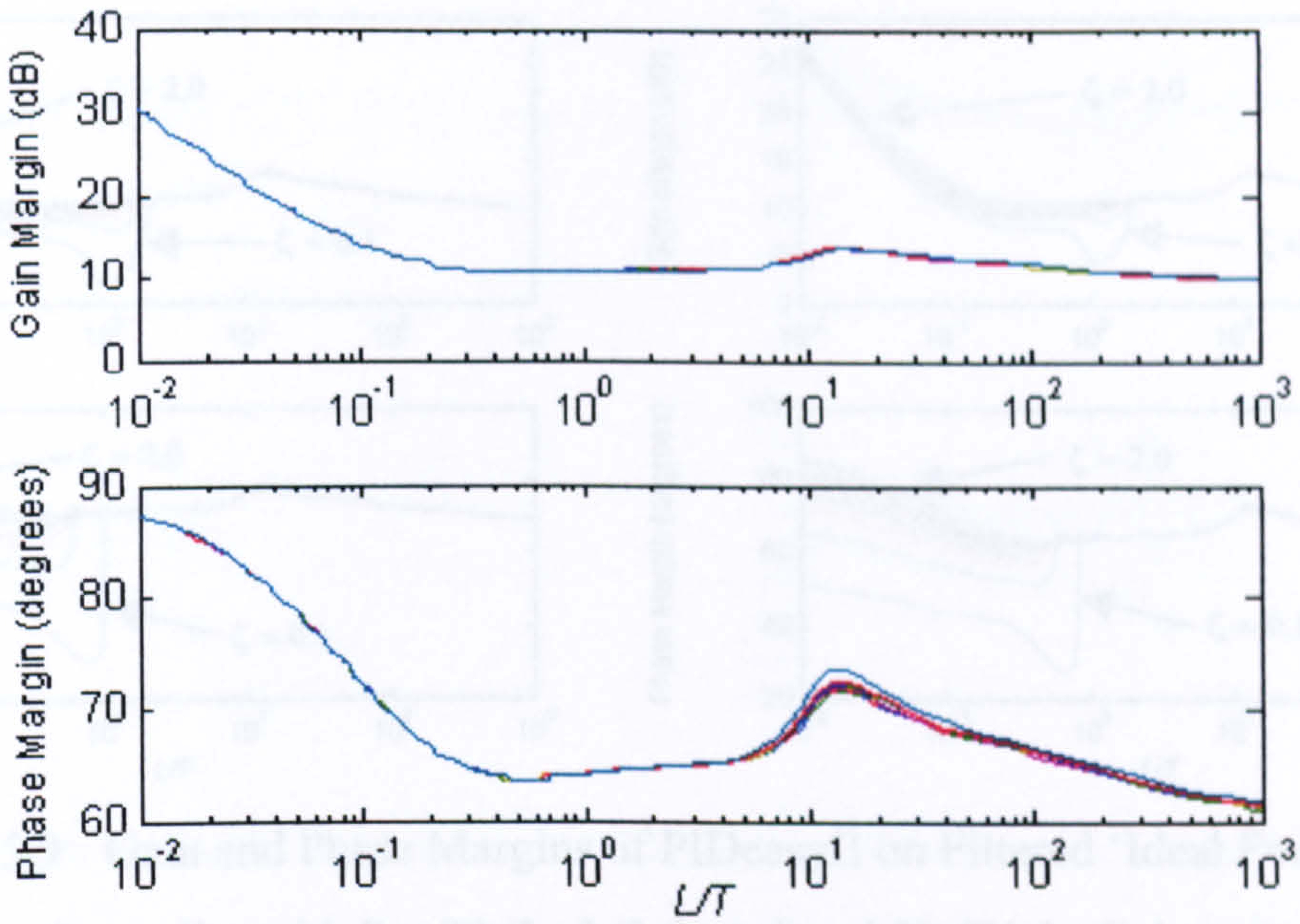

Figure 5.7 Gain and Phase Margins of PIDeasyII on 'Ideal Form' PID Controller

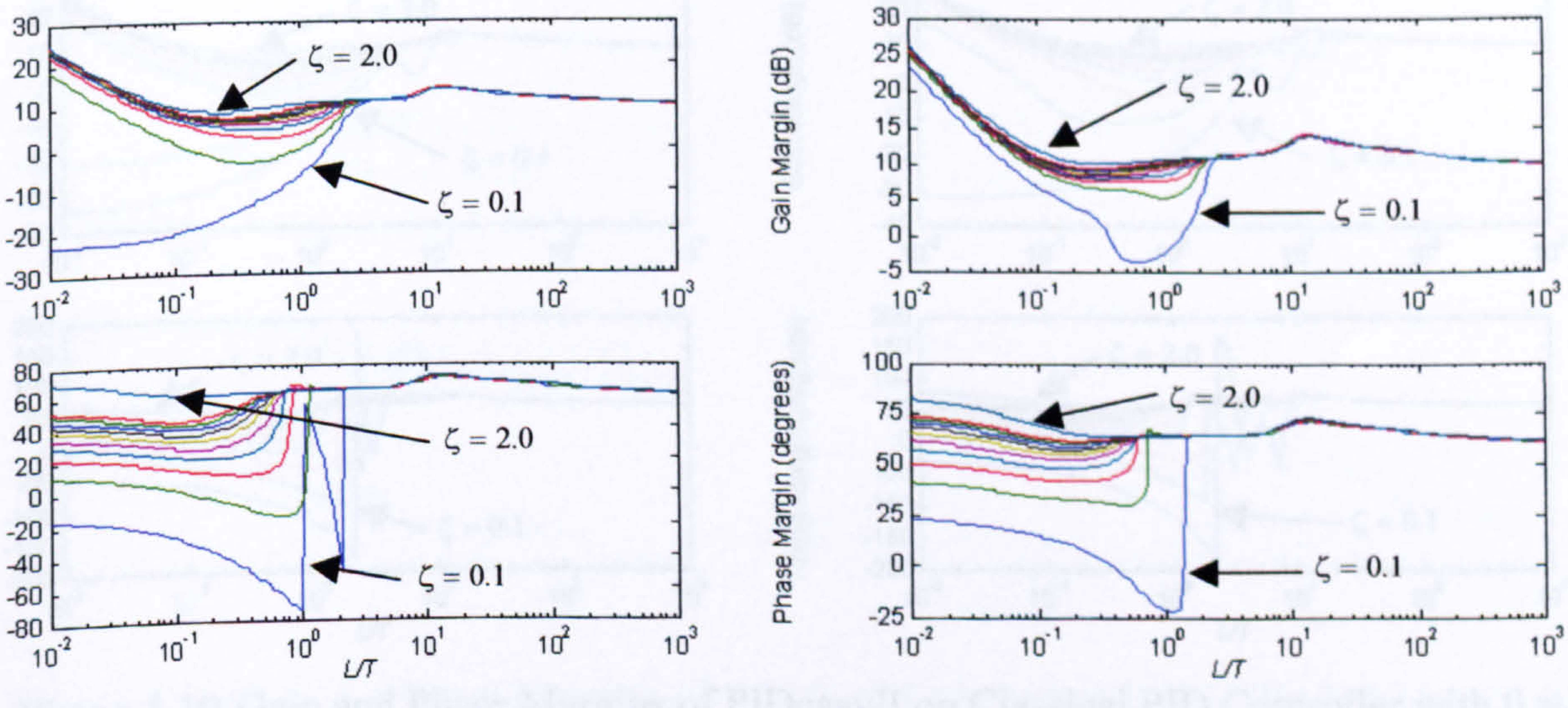

Figure 5.8 Gain and Phase Margins of PIDeasyII on Filtered 'Ideal Form' PID Controller with β = 3 (Left Column) and 10 (Right Column)

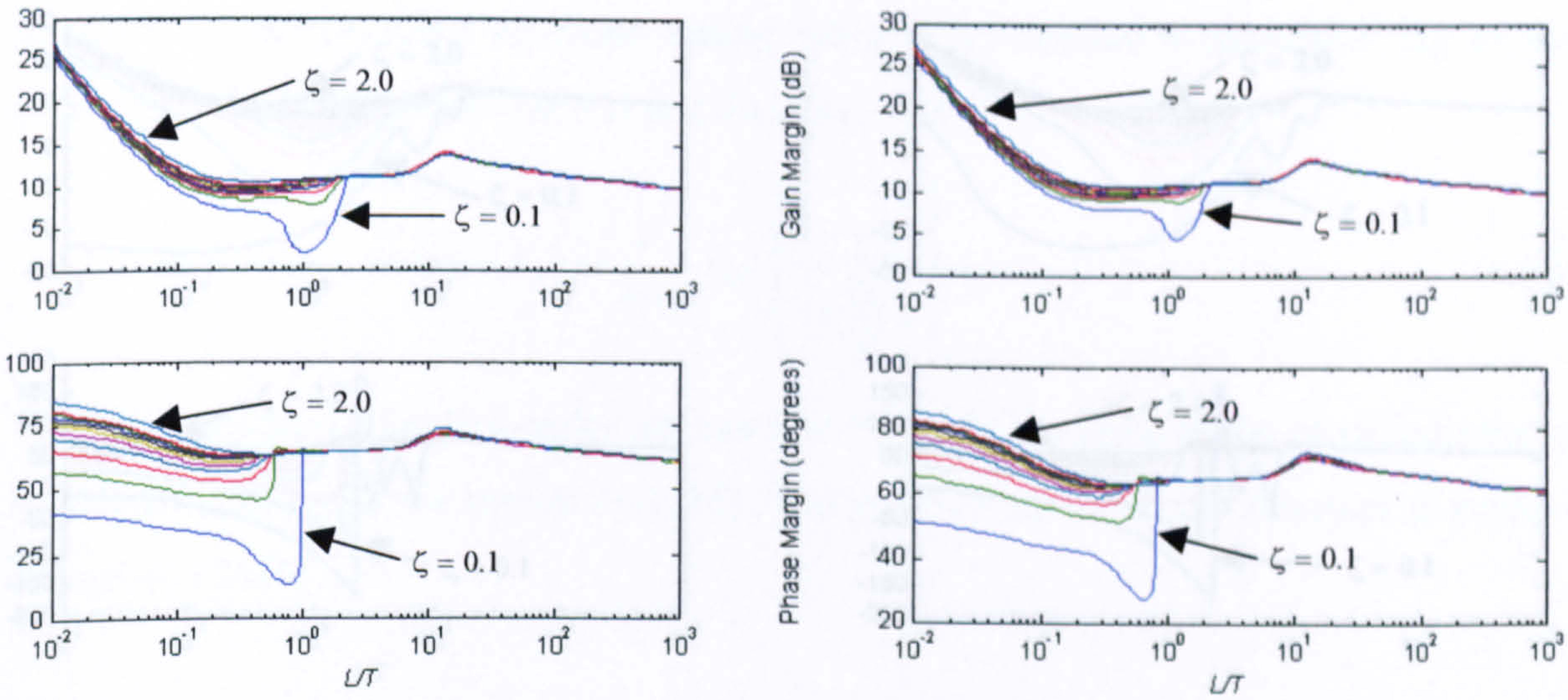

Figure 5.9 Gain and Phase Margins of PIDeasyII on Filtered 'Ideal Form' PID Controller with β = 20 (Left Column) and 30 (Right Column)

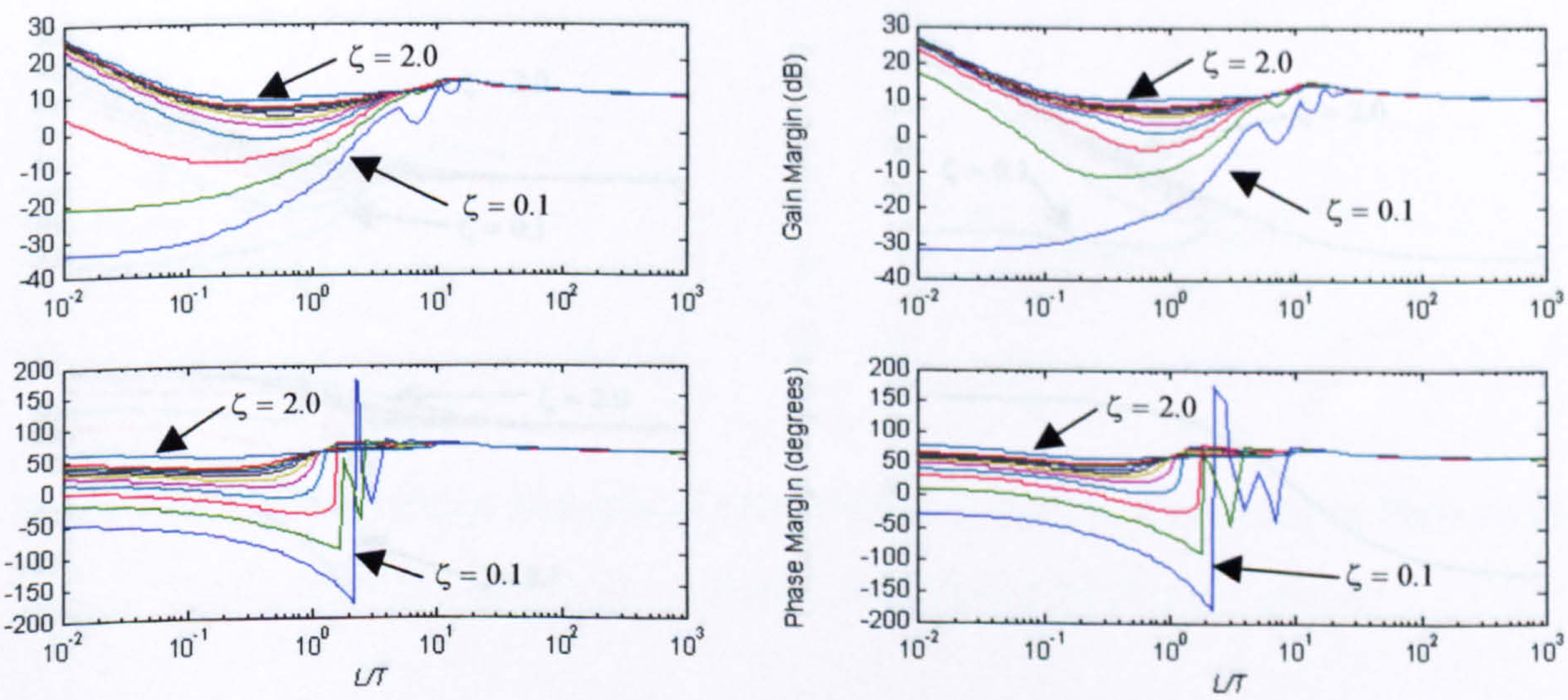

Figure 5.10 Gain and Phase Margins of PIDeasyII on Classical PID Controller with β = 3 (Left Column) and 10 (Right Column)

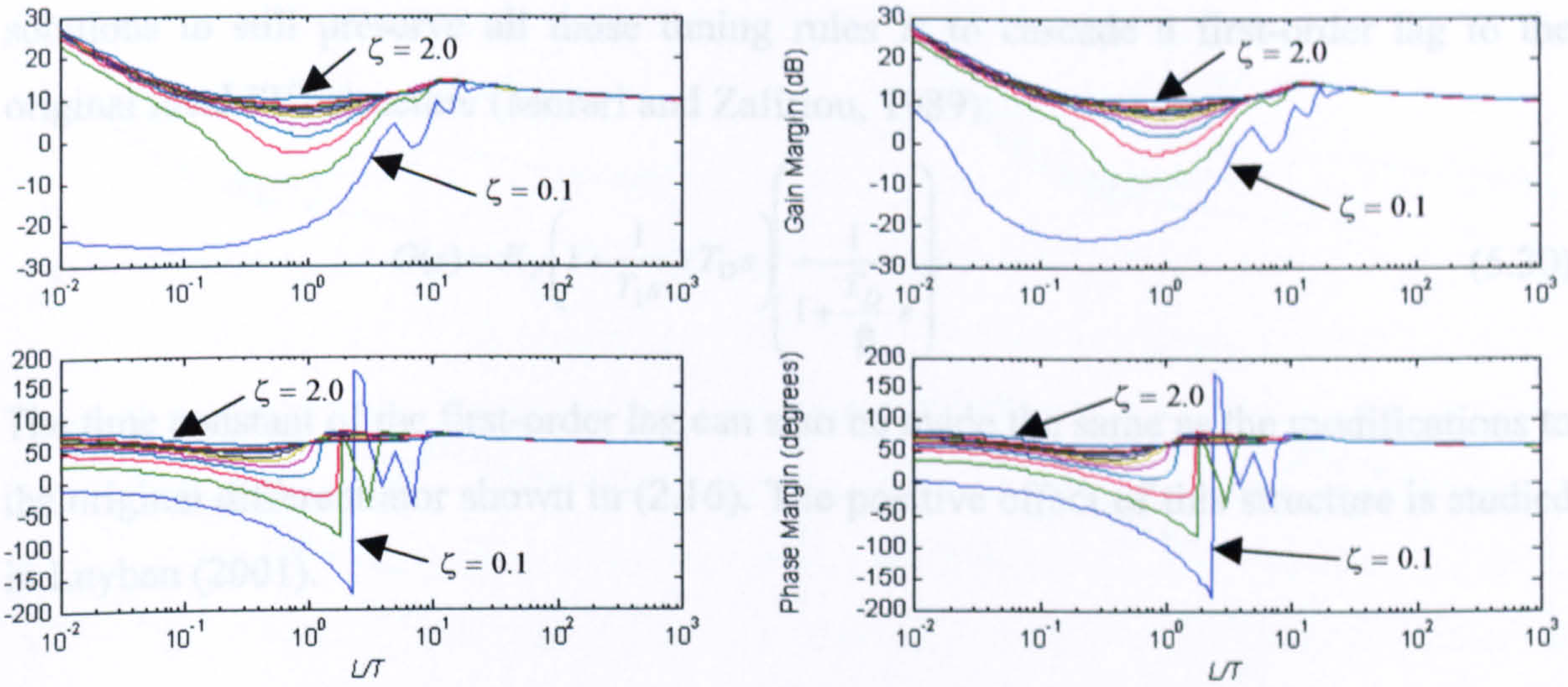

Figure 5.11 Gain and Phase Margins of PIDeasyII on Classical PID Controller with β = 20 (Left Column) and 30 (Right Column)

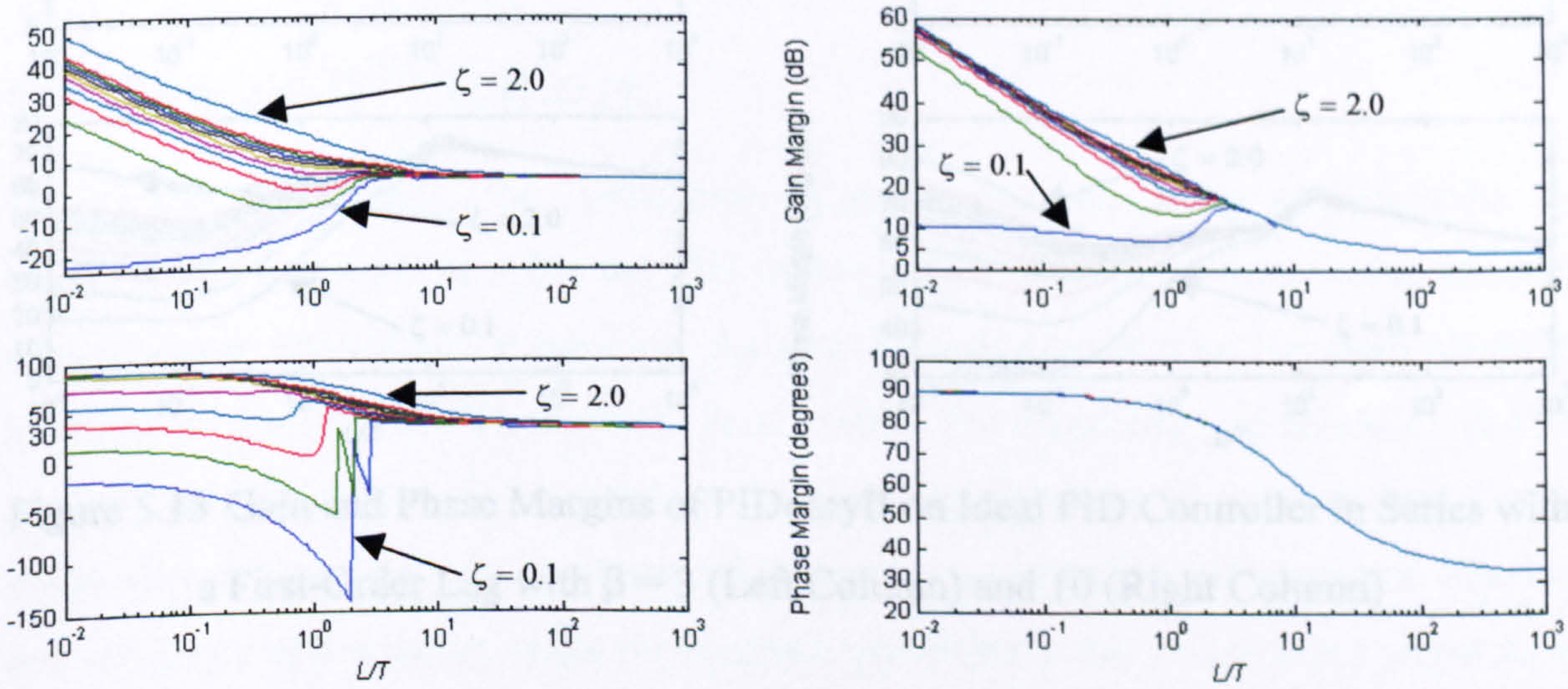

Figure 5.12 Gain and Phase Margins of G-K (Left Column) and W-C (Right Column) on Filtered 'Ideal Form' PID Controller with β = 3

Based on Figures 5.4 to 5.12, it is apparent why there are more tuning rules based on FOLPD model than SOSPD model (O'Dwyer, 2003). The main advantage is simply the developed tuning rules are usually immune to different PID structures. However, the usage is limited to only damped or slightly under-damped response.

It is pity that many tuning rules based on SOSPD model cannot be used because they are designed based on ideal PID structure, which is not employed practically. One of the solutions to still preserve all those tuning rules is to cascade a first-order lag to the original ideal PID structure (Morari and Zafiriou, 1989):

$$G(s) = K_{\rm P}\left(1+\frac{1}{T_{\rm I}s}+T_{\rm D}s\right)\left(\frac{1}{1+\frac{T_D}{\beta}s}\right) \tag{5.30}$$

The time constant of the first-order lag can also be made the same as the modifications to the original differentiator shown in (2.16). The positive effect of this structure is studied in Luyben (2001).

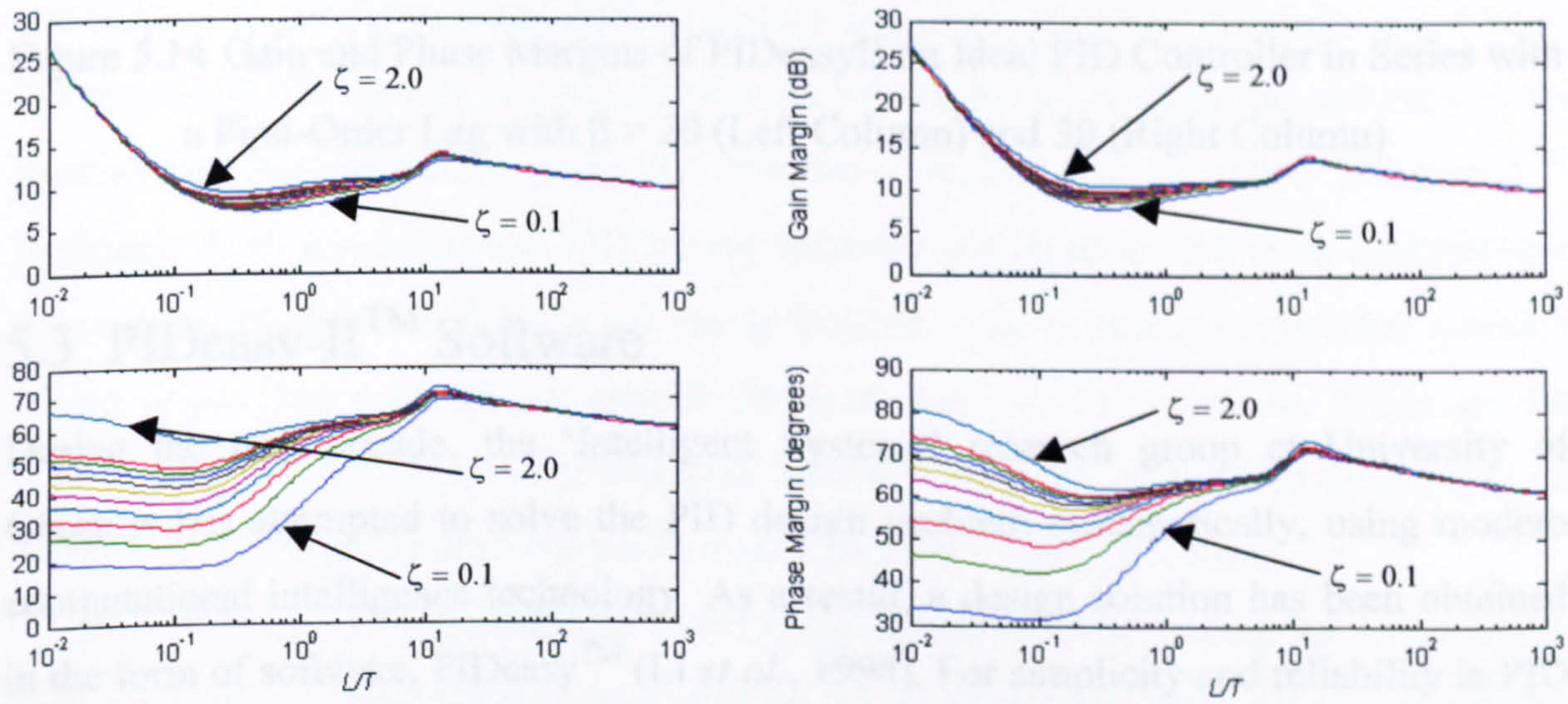

Figure 5.13 Gain and Phase Margins of PIDeasyII on Ideal PID Controller in Series with a First-Order Lag with $\beta = 3$ (Left Column) and 10 (Right Column)

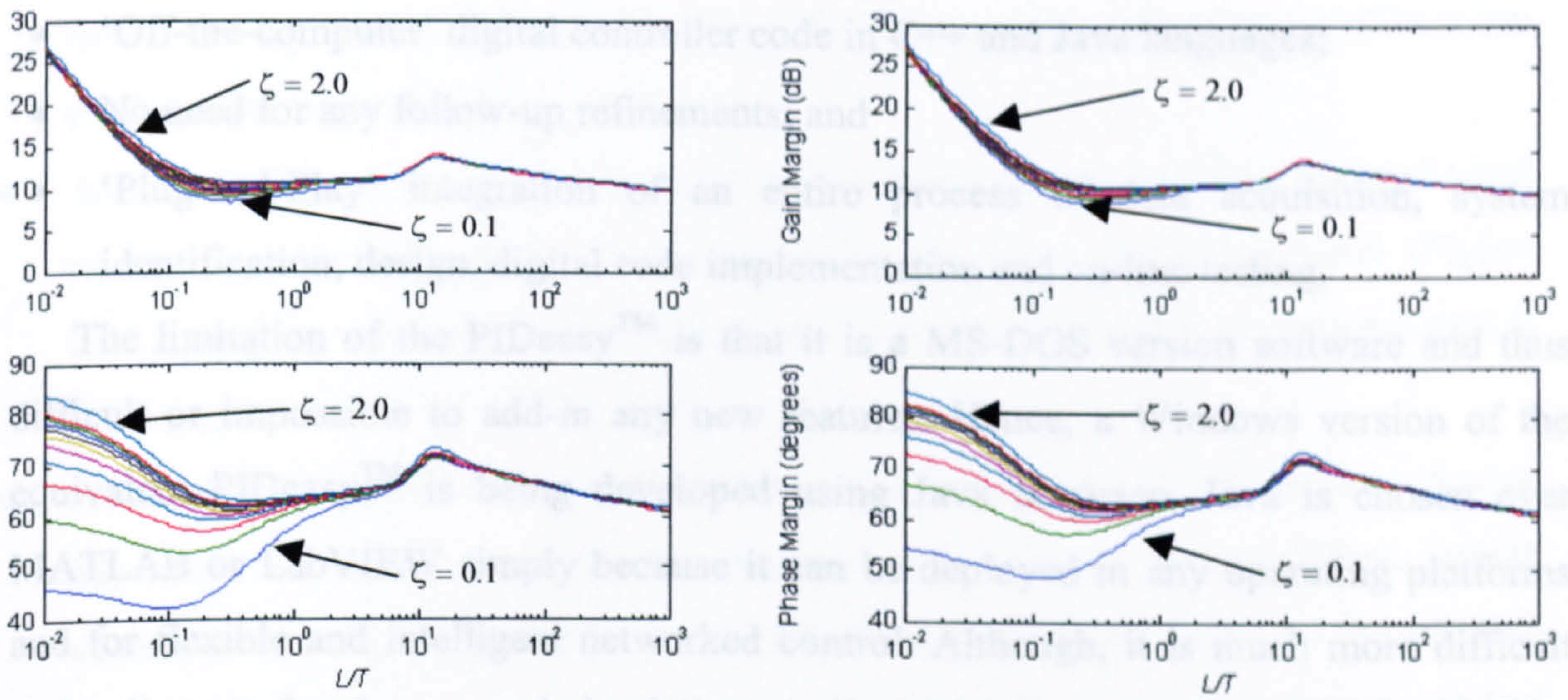

Figure 5.14 Gain and Phase Margins of PIDeasyII on Ideal PID Controller in Series with a First-Order Lag with β = 20 (Left Column) and 30 (Right Column)

5.3 PIDeasy-II™ Software

During the past decade, the 'Intelligent Systems' research group at University of Glasgow has attempted to solve the PID design problem systematically, using modern computational intelligence technology. As a result, a design solution has been obtained in the form of software, PIDeasy™ (Li *et al.*, 1998). For simplicity and reliability in PID applications, effort is made to maintain the controller structure in the 'standard form', while allowing optimal augmentation with simple and effective differentiator filtering and integrator anti-windup. High performance particularly that of transient response is offered through setting the controller parameters optimally in a fraction of a millisecond, as soon as changes in the process dynamics are detected. The optimality is multi-objective and is achieved by addressing existing problems at the roots using modern computational intelligence techniques.

The PIDeasy™ technology is targeted towards wider applications than the ZN based and other techniques that are currently available. This method coherently derives, in a fraction of a second, the optimal PID settings with highest possible control performance from the plant gain, time-constant and transport-delay. It offers:

- Optimal PID designs directly from off-line or on-line plant response;
- Generic and widest application to any first-order delayed plants;

- 'Off-the-computer' digital controller code in C++ and Java languages;
- No need for any follow-up refinements; and
- 'Plug-and-Play' integration of an entire process of data acquisition, system identification, design, digital code implementation and on-line testing.

The limitation of the PIDeasy™ is that it is a MS-DOS version software and thus difficult or impossible to add-in any new features. Hence, a Windows version of the equivalent PIDeasy™ is being developed using Java language. Java is chosen over MATLAB or LabVIEW simply because it can be deployed in any operating platforms and for flexible and intelligent networked control. Although, it is much more difficult and tedious to develop control simulations coding using Java, the accuracy is not being compromised. The results are compared with MATLAB to confirm the accuracy of the simulations. For continuity sake, this Windows version will be called PIDeasy-II™. Although there are numerous PID tuning software packages available as analysed and discussed in Section 3.3, they do not aim to facilitate user in testing some other available tuning rules. Thus they are not suitable for academic teaching purposes. There are two main goals that this software attempts to achieve. They are, to provide user a tool for comparing different tuning rules performance and for teaching purposes.

The PIDeasy-II™ is currently undergoing a performance testing to determine its stability level. Further enhancement will be done, should the need for higher stability level arises. Figures 5.15 to 5.18 show the four supported plant models and the available tuning rules for each model. In addition, there is also a user-defined PID parameters to perform fine tuning and the result is instantaneously reflect on the simulation as the user change each of the parameters. The following are some of the available features:

- Two types of time-domain simulations, namely, step response (Figure 5.19) and load disturbance rejection (Figure 5.20);
- Two types of frequency-domain simulations, namely, Nichols chart (Figure 5.21) and Nyquist plot (Figure 5.22);
- Various standard PID structures, namely, Ideal PID, Ideal PI-D, Ideal I-PD, Ideal PID with low-pass filter on Derivative, Ideal PI-D with low-pass filter on Derivative, Ideal I-PD with low-pass filter on Derivative, Ideal PID in series with a first-order lag, Ideal PI-D in series with a first-order lag, Ideal I-PD in series with a

first-order lag, Series PID and Classical PID. The naming conventions used here are explained in Section 2.2;

- Able to read in real process data, perform matching data to available selected plant models and perform off-line simulation. For the plant modelling panel (Figure 5.23), user can manually adjust each model parameters to match the process data if necessary;
- A simulator (Figure 5.24), which enable user to view the response as they change the reference signal, process gain, process time constant, process delay, injects disturbances or output noise. Note that the simulation is only based on the selected model and not the process data. However, this is a very useful tool as user can test how each tuning rule performs in the presence of modelling errors, load disturbances, or a combination of both, etc;
- Other features are actuator limits simulation, anti-windup setting and low-pass set-point filter.

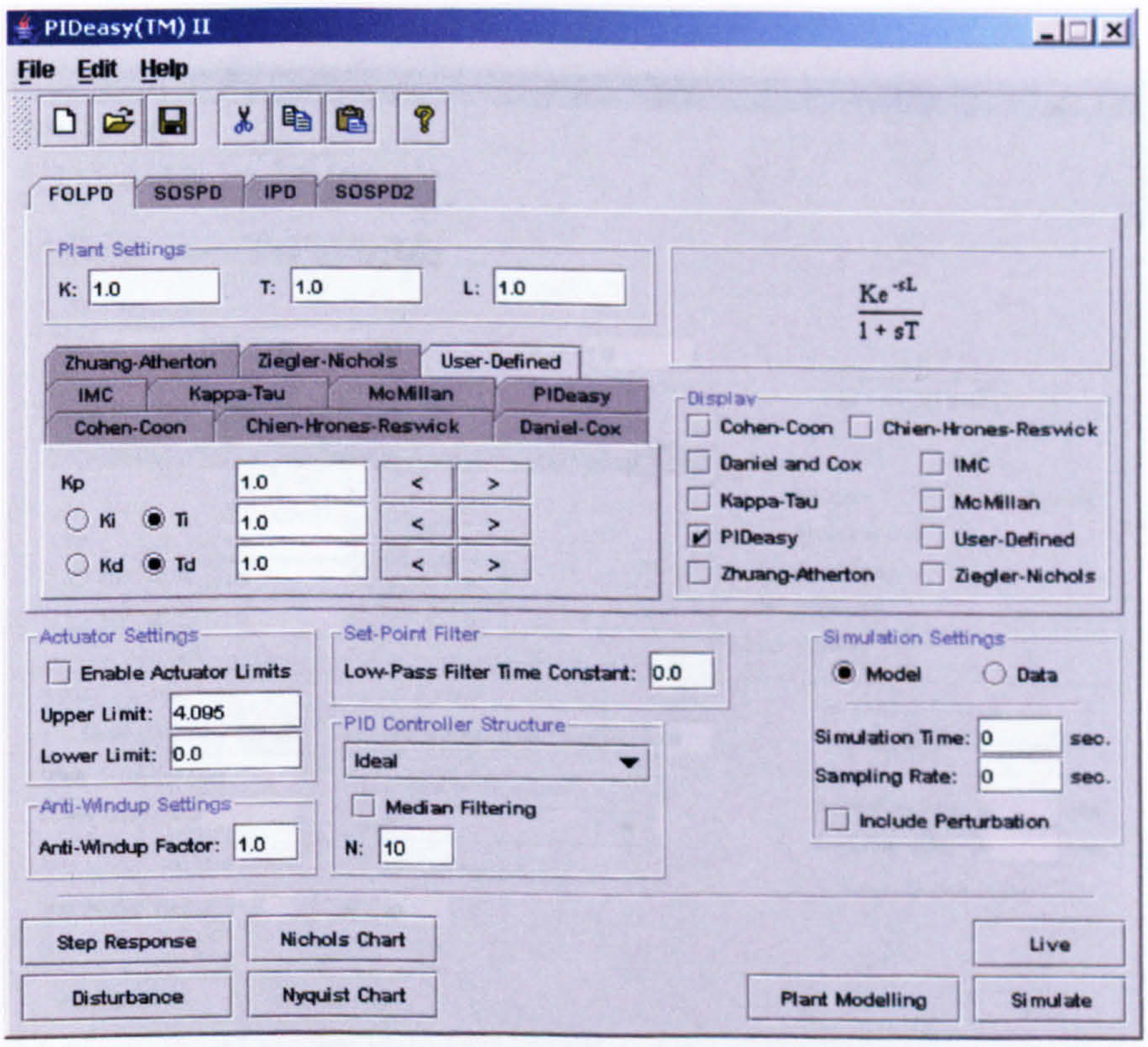

Figure 5.15 Main GUI showing a First-Order Model Panel

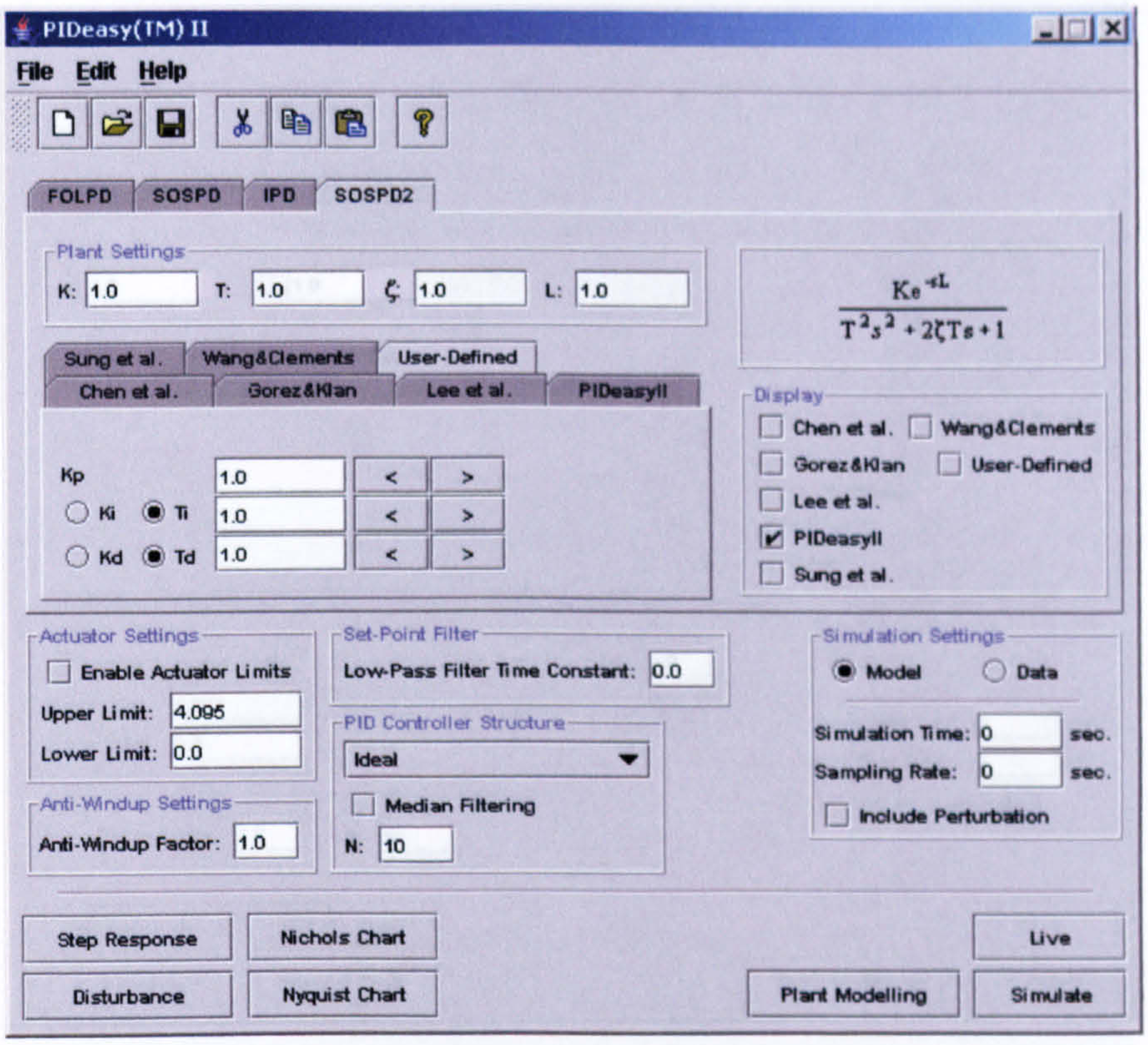

Figure 5.16 Main GUI showing a General Second-Order Model Panel

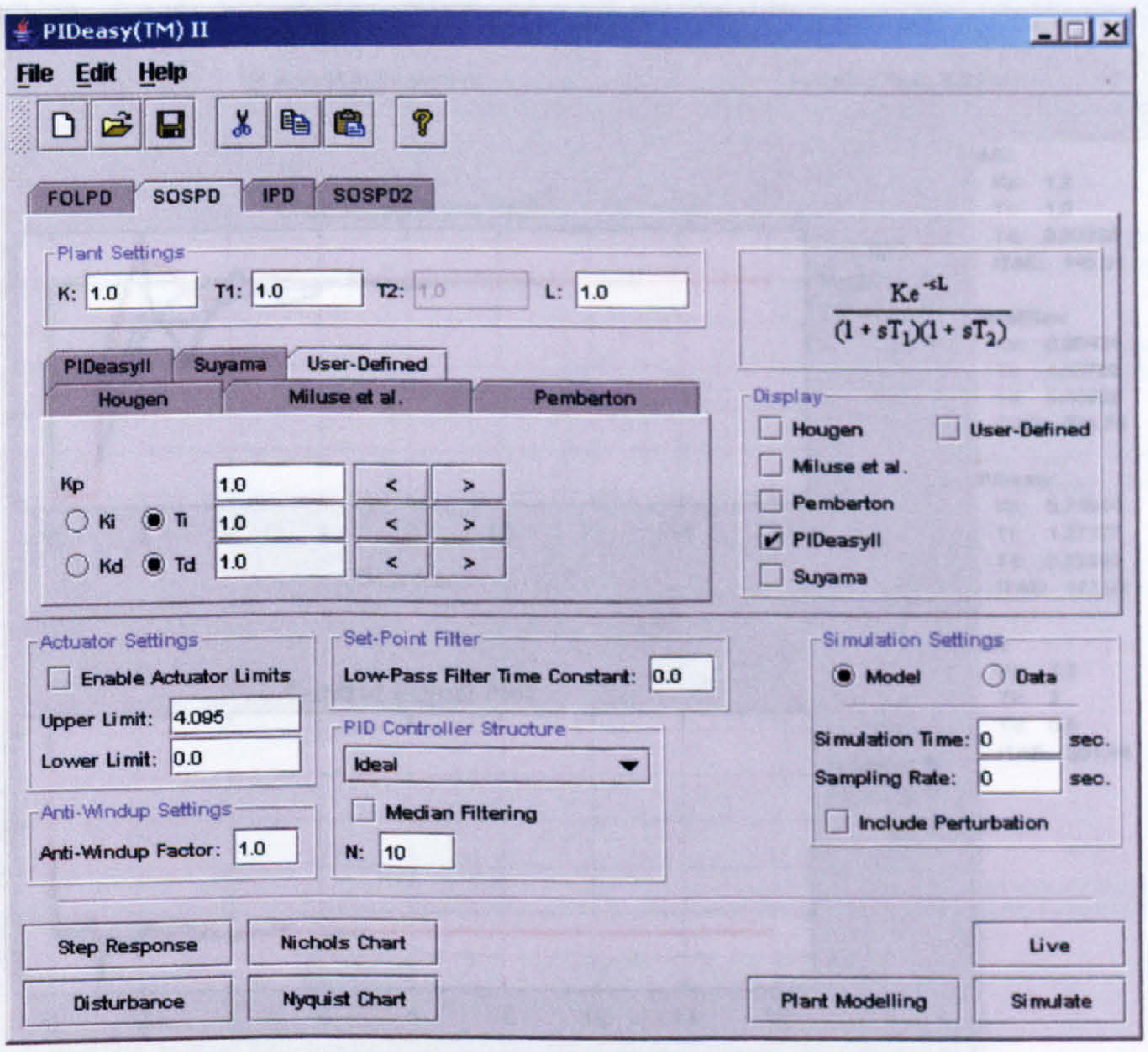

Figure 5.17 Main GUI showing a Second-Order Model with Repeated Pole Panel

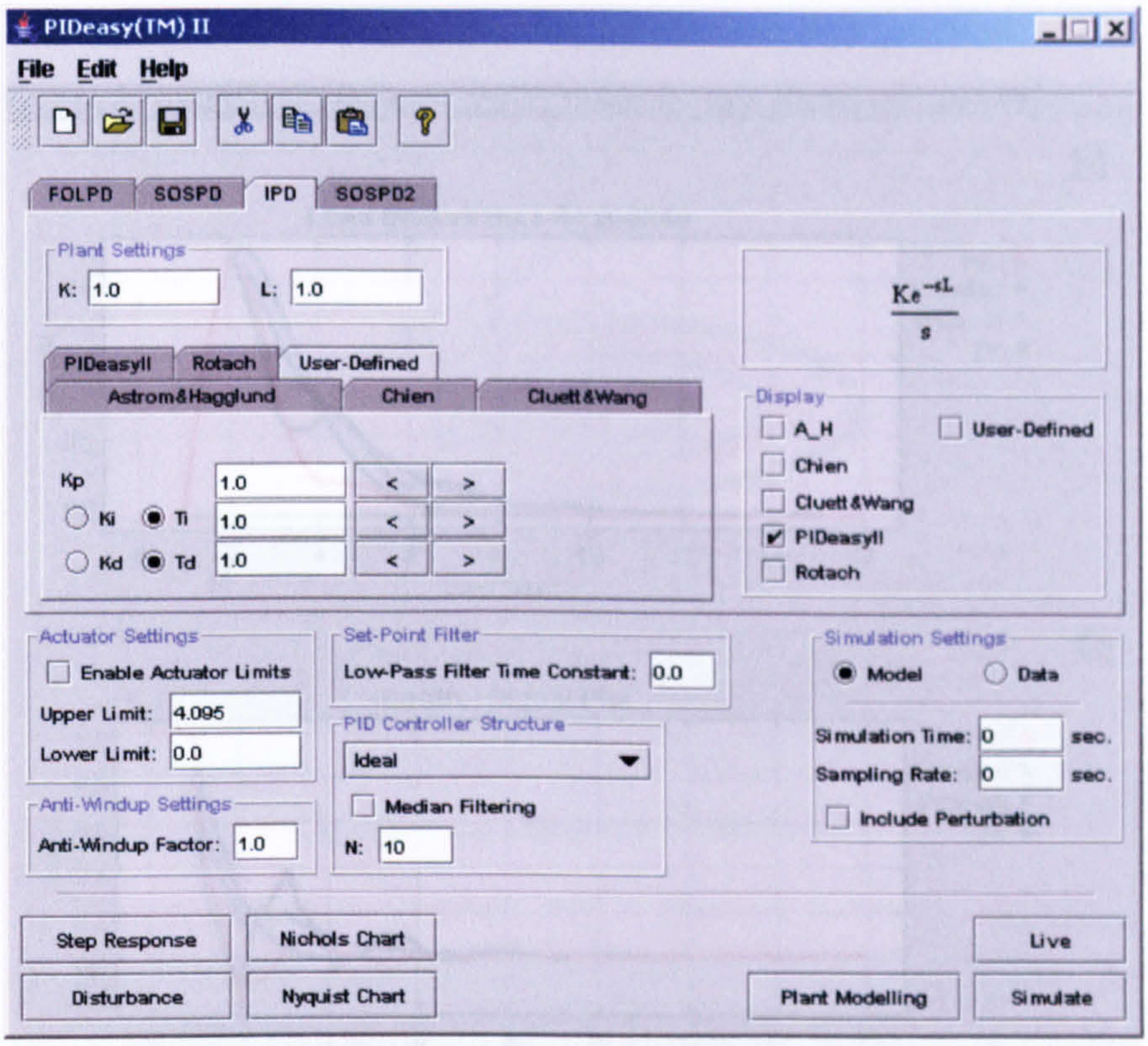

Figure 5.18 Main GUI showing an Integrating Process Model Panel

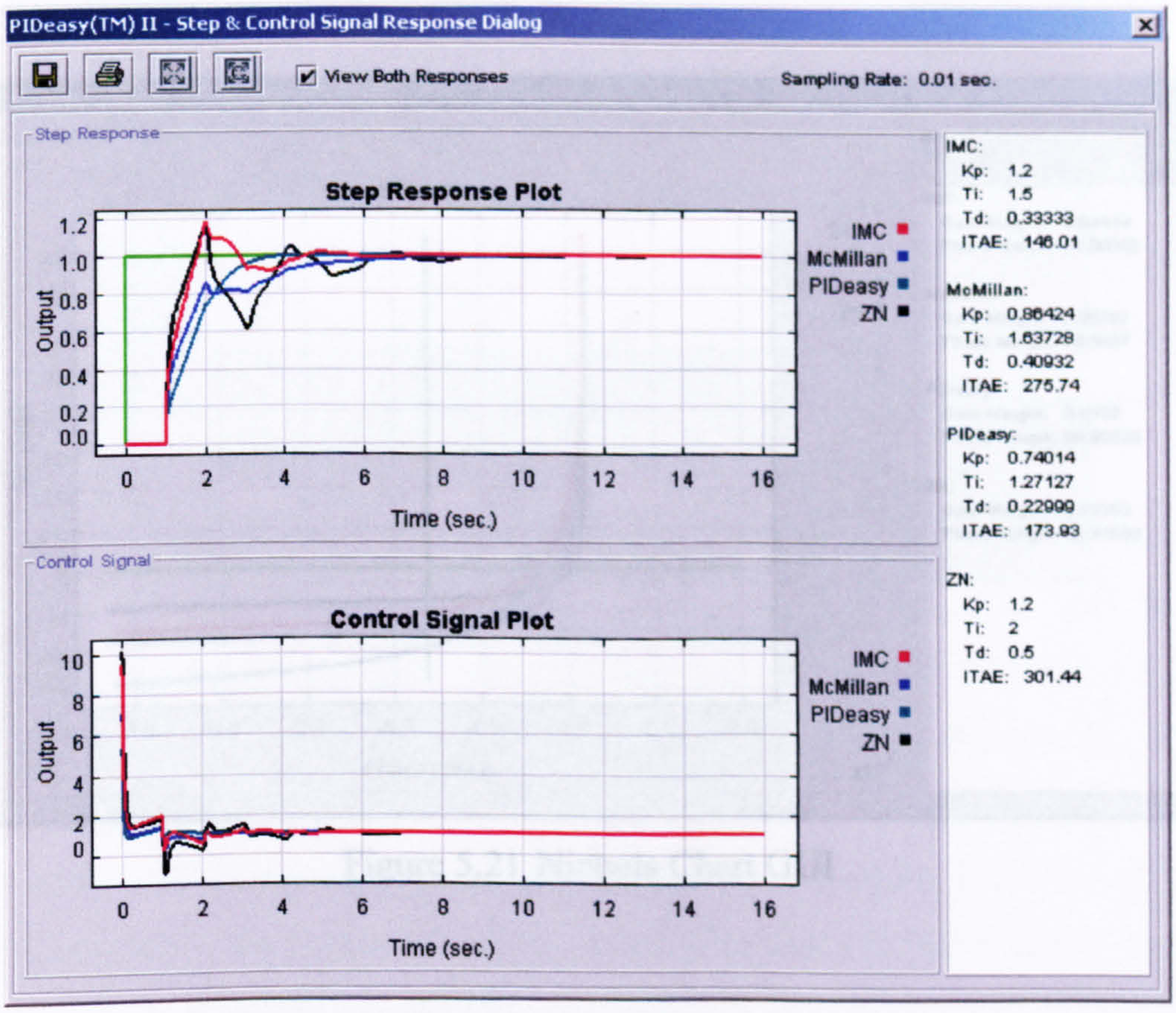

Figure 5.19 Step and Control Signal Response GUI

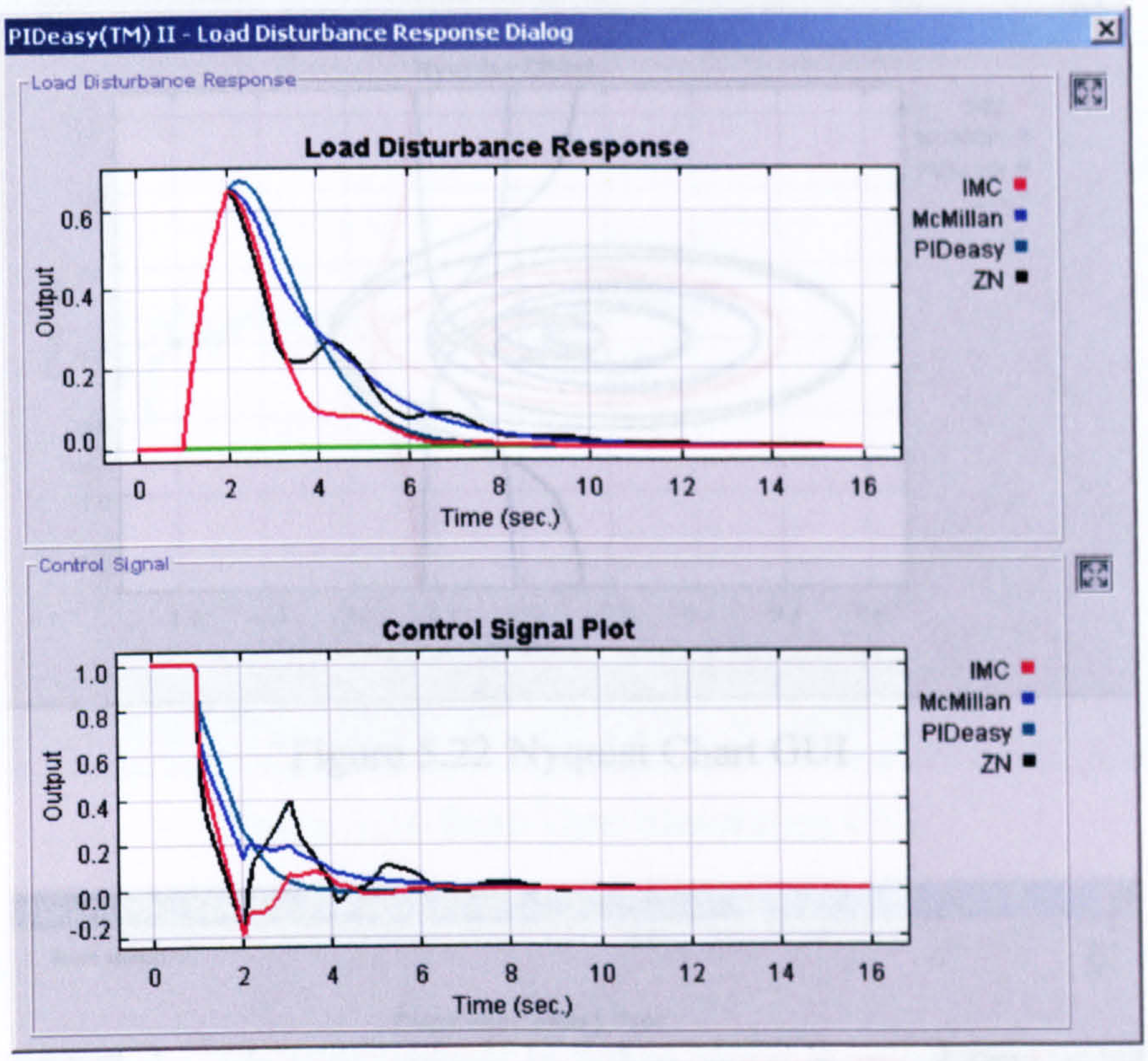

Figure 5.20 Load Disturbance Response GUI

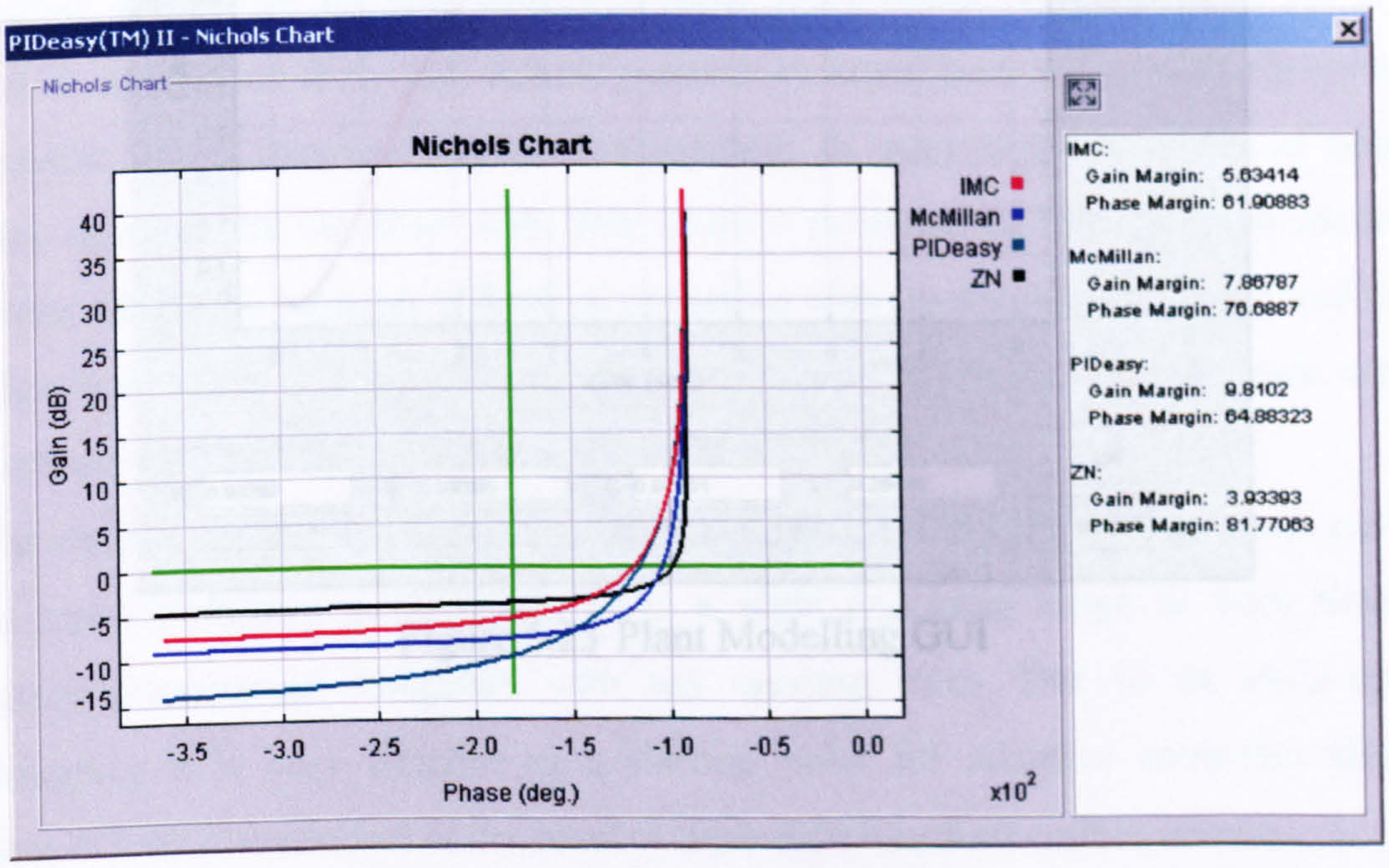

Figure 5.21 Nichols Chart GUI

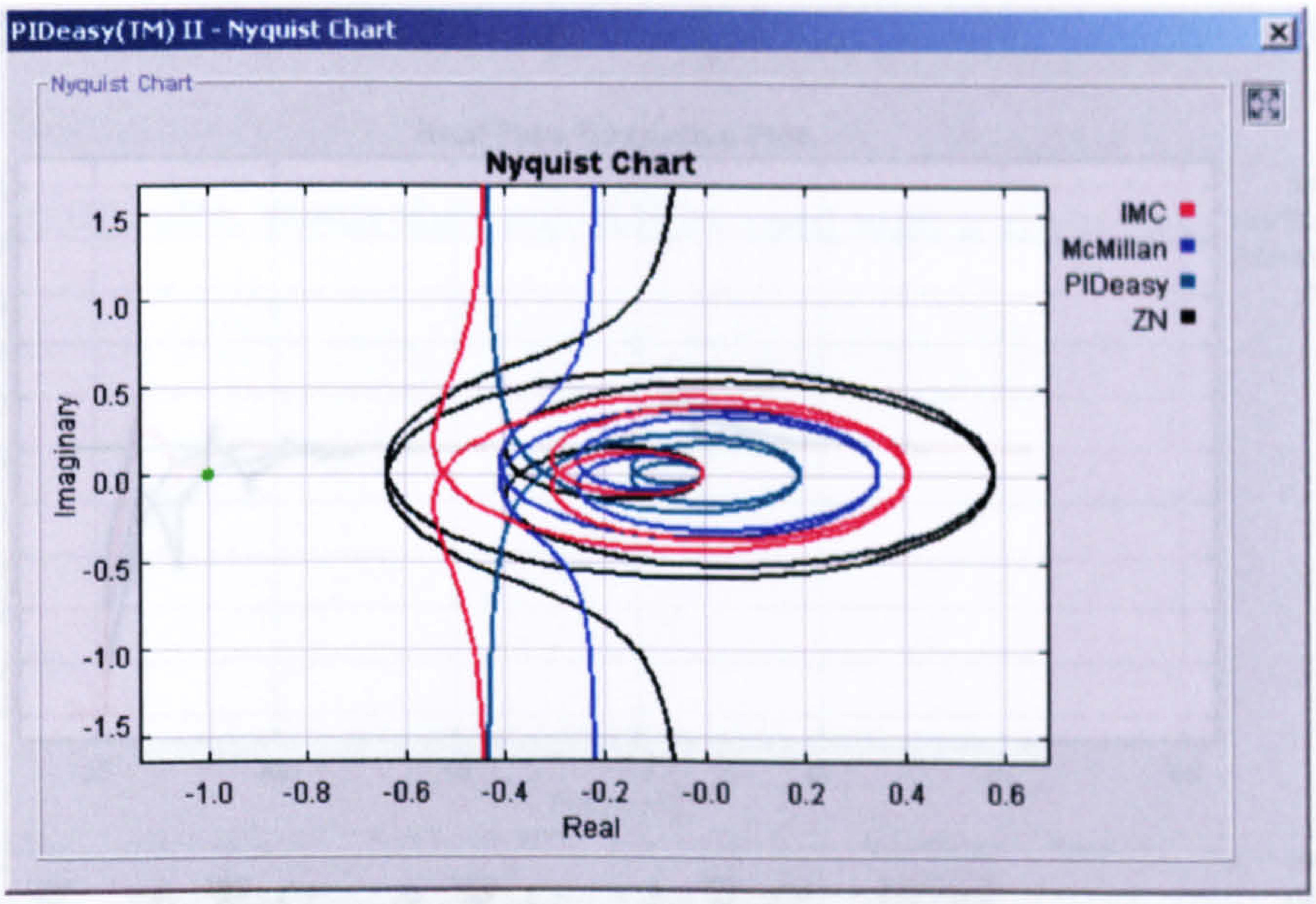

Figure 5.22 Nyquist Chart GUI

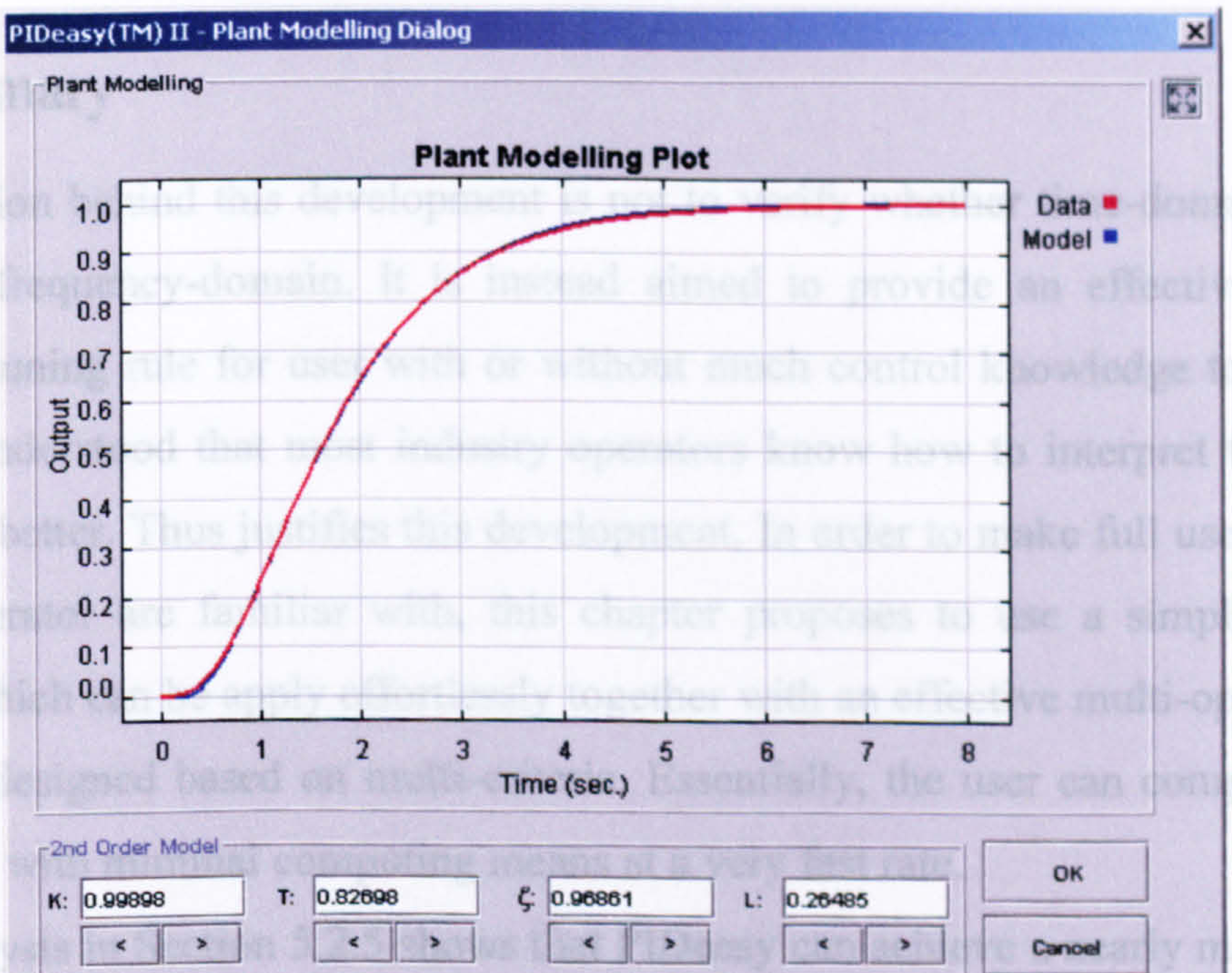

Figure 5.23 Plant Modelling GUI

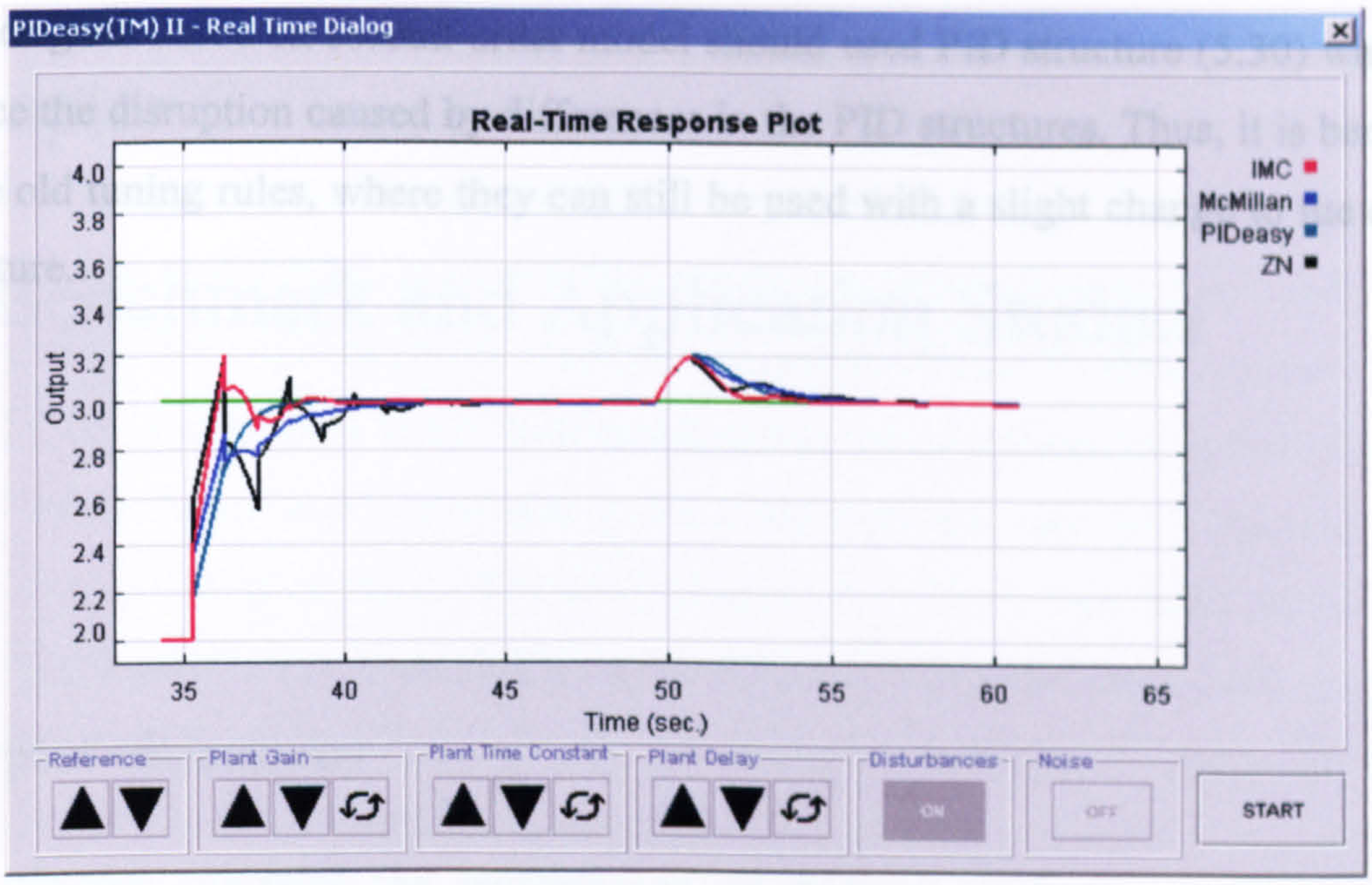

Figure 5.24 Real-Time Simulation GUI

5.4 Summary

The motivation behind this development is not to verify whether time-domain tuning is better than frequency-domain. It is instead aimed to provide an effective and well-understood tuning rule for user with or without much control knowledge to use it with ease. It is understood that most industry operators know how to interpret time-domain information better. Thus justifies this development. In order to make full use of what the industry operator are familiar with, this chapter proposes to use a simple modelling technique which can be apply effortlessly together with an effective multi-optimal tuning rule that is designed based on multi-criteria. Essentially, the user can compute a set of PID settings with minimal computing means at a very fast rate.

The analysis in Section 5.2.5 shows that PIDeasy can achieve a nearly multi-optimal, stable and satisfactory performance over a wide operating range in both time and frequency domains as compared with any existing rules. Due to its multi-optimal performance, it is very suitable as a starting point for adaptive controller (for e.g. Foxboro EXACT controller) or for local optimisation based on certain criteria.

The study of PIDeasy tuning rules on various PID structures indicated that tuning rule designed based on first-order model is more immune to the changes in the PID structures than those based on second-order model. It also shows that tuning rules that are designed based on second-order model should used PID structure (5.30) where it can reduce the disruption caused by differences in the PID structures. Thus, it is beneficial to those old tuning rules, where they can still be used with a slight change to the controller structure.

Chapter 6

Benchmark and Application Studies

Chapter objectives

This chapter evaluates the performance of PIDeasy performance on various PID benchmark test problems and actual real-time laboratory systems.

6.1 Introduction

In order to assess the performance of PIDeasy tuning rules, it is necessary to compare against other tuning rules on a set of recognised benchmark problems and some real-time applications. Two criterions are required for considerations in these studies. They are, namely, the performance of set-point tracking and load disturbance rejection. Most of the industrial non-oscillatory processes exhibit second-order type of response (S-shape) when a step signal is input. However, Section 3.4 shows that most of the practical modelling techniques still use FOLPD type of model. Thus, the two models (FOLPD and SOSPD) mentioned in Chapter 5 will be used together on each study and study the effects of the models on different processes.

6.2 Configuration Setup

It is advisable to show the configuration setup of the test so as to avoid any misunderstandings. For all the offline and online tests, the PID structure used is based on (5.30) and is of 'Type A' structure. The anti-windup scheme is based on (2.8) with $\gamma = 1$. No set-point filter or weighting is used for the input and the process output is unfiltered. This is to investigate the performance using the most basic configuration. The benchmark tests are conducted offline using computer simulation. The configuration setup for the offline benchmark tests is shown in Figure 6.1. The simulation is carried out using MATLAB version 6.5.0.180913a Release 13.

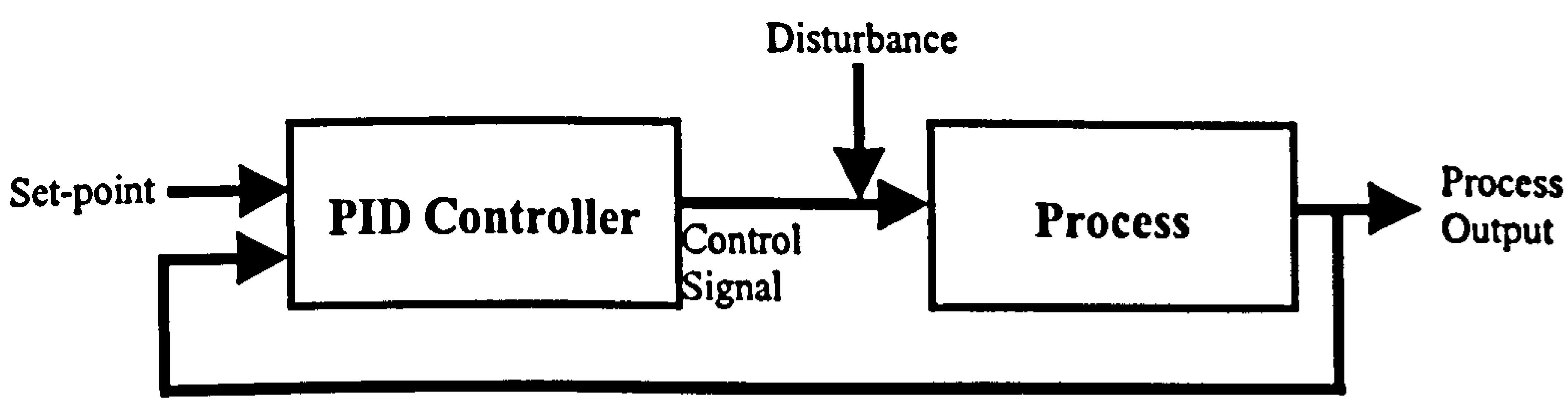

Figure 6.1 Configuration Setup for Offline Benchmark Tests

The online tests setup is shown in Figure 6.2. Nowadays, most controllers are digital in nature, operating on a cyclic basis rather than continuously as to analog controllers. Unlike analog instruments, digital devices must sample the controlled variable and compute and update the controller output at discrete time intervals. Consequently, the dynamic element of sampling plays a major role in most control loops and is often not given enough consideration. The sampling effect is normally being considered in the patented tuning rules but not common in academic literature. It affects performance, robustness, and tuning as well.

Moore *et al.* (1969) developed a simple correction for the controller tuning parameters to account for the effect of sampling. They pointed out that, when a continuous signal is sampled at regular intervals of time and then reconstructed by holding the sampled values constant for each sampling period, the reconstructed signal is effectively delayed by approximately one half the sampling intervals. Therefore, to correct for sampling, one half the sampling time is simply added to the dead time (or delay) obtained from the step response.

The online test will be conducted on the actual process using LabVIEW version 6.1 and National Instruments PCI-6024E multifunction data acquisition card.

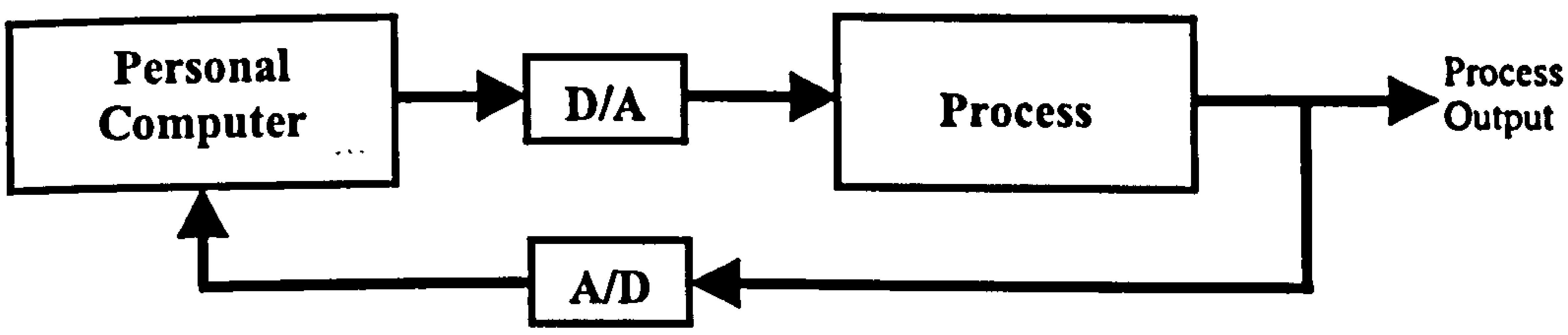

Figure 6.2 Configuration Setup for Online Tests

6.3 Tuning Rules

The selection of tuning rules has been made easier by the significant contribution of O'Dwyer (2003) where a huge collection of the available tuning rules is being compiled. The selected tuning rules are based on their capability on wide operating range instead of those optimised for a limited operating range. The selected tuning rules using FOLPD modelling are labelled as AMIGO (Åström and Hägglund, 2004), IMC (Morari and Zafiriou, 1989), McMillan (McMillan, 1984) and ZN (Ziegler and Nichols, 1942). Those using SOSPD modelling are labelled as G-K (Gorez and Klàn, 2000) and W-C (Wang and Clements, 1995). For clarity of the comparison, the selected tuning rules using FOLPD and SOSPD modelling methods are shown in Table 6.1 and 6.2. The original

PIDeasy tuning rule is also included in the comparison (Li *et al.*, 1998) together with PIDeasyI and PIDeasyII.

Table 6.1 Tuning Rules based on FOLPD Modelling

Rule	K_P	T_I (sec.)	T_D (sec.)	Comments
AMIGO	$\frac{1}{K}\left(0.2+0.45\frac{T}{L}\right)$	$\frac{0.4L+0.8T}{L+0.1T}L$	$\frac{0.5LT}{0.3L+T}$	
IMC	$\frac{2T+L}{2K(\lambda+L)}$	$T+0.5L$	$\frac{TL}{2T+L}$	$\lambda=\max(0.2T,0.25L)$
McMillan	$\frac{1.415T}{KL}\left\{\frac{1}{1+\left(\frac{T}{T+L}\right)^{0.65}}\right\}$	$L\left\{1+\left(\frac{T}{T+L}\right)^{0.65}\right\}$	$0.25L\left\{1+\left(\frac{T}{T+L}\right)^{0.65}\right\}$	
ZN	$\frac{1.2T}{KL}$	$2L$	$0.5L$	

Table 6.2 Tuning Rules based on SOSPD Modelling

Rule	K_P	T_I (sec.)	T_D (sec.)	Comments
G-K	$\frac{2\zeta T}{K(2\zeta T+L)}$	$2\zeta T$	$\frac{T}{2\zeta}$	
W-C	$\frac{2\lambda\zeta T}{K(1+\lambda L)}$	$2\zeta T$	$\frac{T}{2\zeta}$	$\lambda=1$, for all the tests

6.4 Benchmark Tests

Åström and Hägglund (2000) suggest a set of benchmark systems that they have collected from a wide range of sources following their years of research. The five sets of the proposed benchmark systems that are suitable for these parametric studies are:

$$G_1(s)=\frac{1}{(s+1)^n},\qquad n=1,2,3,4,8 \tag{6.1}$$

$$G_2(s)=\frac{1}{(s+1)(ns+1)(n^2s+1)(n^3s+1)},\qquad n=0.1,0.2,0.5 \tag{6.2}$$

$$G_3(s)=\frac{1-ns}{(s+1)^3}, \qquad n=0.1,0.2,0.5,1,2,5 \tag{6.3}$$

$$G_4(s)=\frac{1}{ns+1}e^{-s}, \qquad n=0.1,0.2,0.5,2,5,10 \tag{6.4}$$

$$G_5(s)=\frac{1}{(ns+1)^2}e^{-s}, \qquad n=0.1,0.2,0.5,2,5,10 \tag{6.5}$$

$$G_6(s)=\frac{n^2}{(s+1)(s^2+2\zeta ns+n^2)}, \qquad \zeta=0.1,\ n=1,2 \tag{6.6}$$

System $G_6(s)$ is only suitable for those tuning rules based on SOSPD modelling, as the response is oscillatory. Only n=1 and 2 are selected in this test because when n=5, the response cannot be modelled correctly using any of the methods used here. As for n=10, the response is damped and is quite similar to system $G_1(s)$ and $G_2(s)$. Hence, it is not included in system $G_6(s)$.

Based on the large number of test cases, it is impossible to view all the results in time-domain visually. Thus, the following indices are used as an indication of their performances. They are, namely, gain and phase margins (indicate as GM and PM respectively), ITAE on set-point tracking performance and with derivative (indicate as SP and SP' respectively) and ITAE on load disturbance rejection performance and with derivative (indicate as LD and LD' respectively). The system identification results based on FOLPD and SOSPD models are shown in Table 6.3 and 6.4 respectively. The results for each set of the benchmark systems are shown in Tables 6.5 to 6.10. The unit for T_I and T_D are in seconds, GM is decibel and PM is degrees.

$$\mathrm{SP},\mathrm{LD}=\sum_t t\,e(t) \tag{6.7}$$

$$\mathrm{SP'},\mathrm{LD'}=\sum_t t\,(e(t)+\dot{e}(t)) \tag{6.8}$$

In addition to the selected tuning rules, s-MOEA is also employed to search for the best possible settings for each test case. The search range for s-MOEA is set to [0,10] for K_P, T_I and T_D. Due to the nature of s-MOEA, selection criteria must be set before choosing the final solution from the Pareto set. Here, two solutions from the Pareto set will be used to compare with the tuning rules. The first solution is selected based on the best SP, LD, SP' and LD' performances that can defeat all the tuning rules if possible or else it will based on the best SP performance and will be denoted as s-MOEA1. The second solution is selected solely based on the best LD performance with no compromise on stability and will be denoted as s-MOEA2.

Note that s-MOEA employed here is only meant to verify the optimality of the results obtained by the tuning rules. It is not meant to compete with those tuning rules. Thus, the best result obtained by the tuning rules (except s-MOEA) for each system based on SP, LD, SP' and LD', is highlighted in red colour.

Table 6.3 Benchmark Systems Identification using FOLPD Model

System	K	T (sec.)	L (sec.)	Simulation Duration (sec.)	Sampling Rate (sec.)
$G_1(s)$, n=1	1.0	1.00027	0.0068	5	0.005
$G_1(s)$, n=2	1.0	1.47155	0.50163	15	0.01
$G_1(s)$, n=3	1.0	1.8226	1.1333	20	0.01
$G_1(s)$, n=4	1.0	2.11114	1.82891	20	0.01
$G_1(s)$, n=8	1.0	2.98157	4.91468	30	0.01
$G_2(s)$, n=0.1	1.0	1.00027	0.1168	10	0.01
$G_2(s)$, n=0.2	1.0	1.0195	0.2325	10	0.01
$G_2(s)$, n=0.5	1.0	1.17339	0.67817	10	0.01
$G_3(s)$, n=0.1	1.0	1.8178	1.23437	15	0.01
$G_3(s)$, n=0.2	1.0	1.8178	1.33437	15	0.01
$G_3(s)$, n=0.5	1.0	1.78413	1.65188	15	0.01
$G_3(s)$, n=1.0	1.0	1.72643	2.12476	30	0.01
$G_3(s)$, n=2.0	1.0	1.63025	2.83622	50	0.05
$G_3(s)$, n=5.0	1.0	1.5004	4.02519	100	0.05
$G_4(s)$, n=0.1	1.0	0.10099	1.00747	15	0.01
$G_4(s)$, n=0.2	1.0	0.20198	1.00493	15	0.01
$G_4(s)$, n=0.5	1.0	0.49533	1.00947	15	0.01
$G_4(s)$, n=2.0	1.0	1.99092	1.00574	20	0.01
$G_4(s)$, n=5.0	1.0	4.9773	1.00935	50	0.05
$G_4(s)$, n=10.0	1.0	9.88246	1.04479	50	0.05

$G_5(s)$, n=0.1	1.0	0.14427	1.05781	15	0.01
$G_5(s)$, n=0.2	1.0	0.29335	1.10454	15	0.01
$G_5(s)$, n=0.5	1.0	0.73577	1.25582	15	0.01
$G_5(s)$, n=2.0	1.0	2.94791	1.99219	20	0.01
$G_5(s)$, n=5.0	1.0	7.35774	3.50817	50	0.05
$G_5(s)$, n=10.0	1.0	14.71549	5.96633	100	0.05

Table 6.4 Benchmark Systems Identification using SOSPD Model

System	K	T (sec.)	ζ	L (sec.)	Simulation Duration (sec.)	Sampling Rate (sec.)
$G_1(s)$, n=1	1.0	0.17843	2.89542	0.005	5	0.005
$G_1(s)$, n=2	1.0	0.99861	0.99173	0.01991	15	0.01
$G_1(s)$, n=3	1.0	1.43065	0.90224	0.39445	20	0.01
$G_1(s)$, n=4	1.0	1.83856	0.84372	0.83793	20	0.01
$G_1(s)$, n=8	1.0	2.06312	0.99636	3.89728	30	0.01
$G_2(s)$, n=0.1	1.0	0.32614	1.69175	0.02403	10	0.01
$G_2(s)$, n=0.2	1.0	0.46196	1.31777	0.04725	10	0.01
$G_2(s)$, n=0.5	1.0	0.82698	0.96861	0.26485	10	0.01
$G_3(s)$, n=0.1	1.0	1.43216	0.90006	0.49436	15	0.01
$G_3(s)$, n=0.2	1.0	1.41095	0.90816	0.61263	15	0.01
$G_3(s)$, n=0.5	1.0	1.32677	0.93348	0.98738	15	0.01
$G_3(s)$, n=1.0	1.0	1.18463	0.98411	1.55475	30	0.01
$G_3(s)$, n=2.0	1.0	0.96646	1.08442	2.4135	50	0.05
$G_3(s)$, n=5.0	1.0	0.66361	1.33591	3.78853	100	0.05
$G_4(s)$, n=0.1	1.0	0.04274	1.40847	0.99566	15	0.01
$G_4(s)$, n=0.2	1.0	0.03569	2.89542	1.01178	15	0.01
$G_4(s)$, n=0.5	1.0	0.08854	2.89542	0.99932	15	0.01
$G_4(s)$, n=2.0	1.0	0.35431	2.44255	0.79144	20	0.01
$G_4(s)$, n=5.0	1.0	1.06532	2.44255	0.79144	50	0.05

$G_4(s)$, n=10.0	1.0	2.17511	2.37756	0.64634	50	0.05
$G_5(s)$, n=0.1	1.0	0.09246	1.04078	1.02079	15	0.01
$G_5(s)$, n=0.2	1.0	0.19807	0.99851	1.01486	15	0.01
$G_5(s)$, n=0.5	1.0	0.50096	0.99173	1.01435	15	0.01
$G_5(s)$, n=2.0	1.0	2.00849	0.98849	1.01448	20	0.01
$G_5(s)$, n=5.0	1.0	4.99306	0.99173	1.09956	50	0.05
$G_5(s)$, n=10.0	1.0	10.04706	0.98711	1.07443	100	0.05
$G_6(s)$, n=1	1.0	0.97796	0.13325	0.95	60	0.05
$G_6(s)$, n=2	1.0	0.466916	0.15283	0.77	30	0.01

Table 6.5 Results for $G_1(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=1	AMIGO	66.394	0.051	0.003	Inf.	85.7	0.19	0.01	7.06	0.22
	IMC	4.852	1.004	0.003	Inf.	90.9	8.69	47.95	50.57	93.38
	McMillan	104.301	0.014	0.003	Inf.	71.4	0.07	***0.0***	6.07	0.07
	PIDeasy	92.095	1.002	0.002	Inf.	100	***0.03***	2.12	***2.80***	4.26
	PIDeasyI	13.981	1.002	0.002	Inf.	91.6	1.06	14.77	15.91	29.71
	ZN	176.518	0.014	0.003	Inf.	98.3	***0.03***	***0.0***	3.36	***0.03***
	G-K	0.995	1.033	0.031	Inf.	91.6	210.08	369.55	407.24	572.27
	PIDeasyII	25.405	1.034	0.031	Inf.	134	0.58	8.36	9.31	16.81
	W-C	1.028	1.033	0.031	Inf.	91.7	198.30	354.96	390.39	553.51
	s-MOEA1	10.0	0.109	0.0	Inf.	58.1	4.52	0.57	51.25	5.44
	s-MOEA2	10.0	0.025	0.0	Inf.	30.5	4.27	0.22	91.54	4.49
n=2	AMIGO	1.520	1.065	0.228	Inf.	53.9	211.72	217.57	458.63	406.0
	IMC	2.164	1.722	0.214	Inf.	66.6	78.72	198.56	202.54	320.37
	McMillan	2.273	0.916	0.229	Inf.	45.1	209.82	135.92	504.25	290.44
	PIDeasy	1.948	1.610	0.137	Inf.	62	101.33	199.93	254.15	329.46
	PIDeasyI	1.796	1.610	0.137	Inf.	63.4	105.38	223.14	257.73	363.21
	ZN	3.520	1.003	0.251	Inf.	49.2	116.08	65.84	315.09	149.79
	G-K	0.990	1.981	0.503	Inf.	88.3	385.11	780.91	585.03	1060.8
	PIDeasyII	9.035	1.981	0.504	Inf.	77.6	***4.27***	***47.15***	***27.17***	***76.21***
	W-C	1.942	1.981	0.503	Inf.	87.1	97.37	300.89	200.03	448.06
	s-MOEA1	10.0	2.0	0.495	Inf.	76.4	3.19	43.02	24.04	69.27
	s-MOEA2	10.0	0.895	0.229	Inf.	50.9	37.78	10.33	146.57	29.57
n=3	AMIGO	0.924	1.647	0.478	32.9	56.3	618.35	978.41	1027.4	1466.6
	IMC	1.595	2.389	0.432	27.9	59.4	281.68	575.19	560.14	851.45
	McMillan	1.315	1.961	0.490	30.1	57.5	379.42	636.73	694.17	983.53
	PIDeasy	1.116	2.132	0.284	23	60.4	412.44	777.13	744.64	1154.7
	PIDeasyI	1.075	2.132	0.284	23.3	61.5	411.40	814.95	735.80	1198.1
	ZN	1.930	2.267	0.567	26.6	57.7	218.85	398.22	460.04	622.78
	G-K	0.867	2.582	0.793	31.3	81.3	737.70	1627.3	1035.7	2099.6

	PIDeasyII	2.869	2.583	0.791	20.9	59.9	***103.06***	***312.81***	***270.49***	***471.18***
	W-C	1.851	2.582	0.793	24.7	71.6	146.79	554.17	293.88	796.0
	s-MOEA1	3.376	2.788	0.884	18.4	56.9	97.21	297.88	273.40	436.73
	s-MOEA2	7.034	1.414	0.885	12.1	32.5	146.16	65.30	498.39	128.43
n=4	AMIGO	0.719	2.170	0.726	17.5	59.1	1141.8	2200.7	1662.1	2976.1
	IMC	1.323	3.026	0.638	12.4	57.3	660.71	1202.6	1137.5	1718.7
	McMillan	0.980	3.048	0.762	15.8	69.1	645.78	1879.6	963.93	2464
	PIDeasy	0.835	2.608	0.433	13.7	61.5	881.88	1846.9	1360.6	2521.3
	PIDeasyI	0.815	2.608	0.433	13.9	62.2	879.08	1890.1	1348.7	2566
	ZN	1.385	3.658	0.914	13.2	70.5	576.13	1607	897.33	2086.3
	G-K	0.787	3.102	1.09	17.9	76.9	1171.4	2732.6	1567.9	3417.3
	PIDeasyII	1.693	3.105	1.088	11.3	62.2	368.35	***952.74***	717.88	***1339.1***
	W-C	1.688	3.102	1.09	11.3	62.3	***367.49***	956.47	***715.15***	1343.8
	s-MOEA1	1.687	2.940	1.145	11.2	62	365.38	928.53	705.74	1318.8
	s-MOEA2	3.093	2.253	1.352	5.45	32.5	932.81	399.79	2127.6	702.88
n=8	AMIGO	0.473	4.102	1.644	12.1	64.4	5181.4	11724	6145	13705
	IMC	0.885	5.439	1.347	6.8	59.6	***3931.9***	***7419***	***5238.4***	***9263.7***
	McMillan	0.561	7.524	1.881	10.9	86.1	11066	16783	11969	18341
	PIDeasy	0.502	4.307	1.010	10.3	63.9	4500.9	11073	5454.6	13039
	PIDeasyI	0.498	4.307	1.010	10.4	64.2	4526.1	11182	5474.1	13151
	ZN	0.728	9.829	2.457	7.79	95.3	11128	15950	12131	17336
	G-K	0.513	4.111	1.035	9.94	61.1	4752.6	10744	5808.3	12742
	PIDeasyII	0.469	4.125	1.039	10.8	63.8	4900	11605	5880.2	13620
	W-C	0.839	4.111	1.035	5.67	42.2	7385.1	9833.5	9541.5	12522
	s-MOEA1	0.792	5.011	1.946	8.02	62.9	2771.2	7376.4	3617.4	8968.6
	s-MOEA2	1.109	4.766	2.611	4.49	57.2	4492.1	5577.3	6477.4	7345.9

Table 6.6 Results for $G_2(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=0.1	AMIGO	4.054	0.456	0.056	40.7	74	14.17	7.97	66.84	28.47
	IMC	3.341	1.059	0.055	40.6	80.2	8.21	43.39	40.39	84.62
	McMillan	6.276	0.226	0.056	34.9	40.5	14.89	3.35	99.44	19.73
	PIDeasy	5.525	1.033	0.035	40.9	70	3.16	22.72	26.48	46.94
	PIDeasyI	4.255	1.033	0.035	43.2	73.6	4.56	30.85	30.79	62.51
	ZN	10.277	0.234	0.058	30.3	46.8	6.35	***1.22***	53.14	***8.35***
	G-K	0.979	1.103	0.096	45.3	89.2	125.05	249.36	238.18	393.11
	PIDeasyII	11.921	1.104	0.096	23.6	78.8	***0.74***	10.99	***10.54***	22.22
	W-C	1.078	1.103	0.096	44.4	89.1	103.06	216.25	205.92	347.50
	s-MOEA1	10.0	1.094	0.086	26.3	78.3	0.94	13.06	12.50	26.51
	s-MOEA2	10.0	0.296	0.032	36.7	42.6	5.74	1.13	52.87	7.69
n=0.2	AMIGO	2.173	0.632	0.109	38.3	82.3	41.47	37.87	137.76	99.89
	IMC	2.603	1.136	0.104	31.9	73.7	14.19	68.10	59.97	130.6
	McMillan	3.309	0.436	0.109	29.5	43.7	42.10	19.71	172.15	70.10
	PIDeasy	2.867	1.084	0.066	33.3	66.5	13.63	54.96	64.38	110.38
	PIDeasyI	2.518	1.084	0.066	34.4	68.8	15.44	64.83	67.56	127.74
	ZN	5.262	0.465	0.116	25	47.6	20.09	***8.33***	99.53	***32.98***
	G-K	0.963	1.218	0.175	36	87.7	153.48	309.37	280.12	478.63

	PIDeasyII	8.117	1.218	0.175	17.5	67	***1.67***	20.25	***19.01***	40.03
	W-C	1.163	1.218	0.175	34.4	87.2	104.58	234.56	209.74	376.99
	s-MOEA1	8.750	1.211	0.174	16.9	65.1	1.47	18.42	18.73	36.57
	s-MOEA2	10.0	0.444	0.111	19.8	45.3	8.21	2.41	58.73	11.71
n=0.5	AMIGO	0.979	1.032	0.289	35.4	90.9	209.92	329.27	437.63	596.43
	IMC	1.657	1.512	0.263	17.8	61.9	80.11	218.0	217.48	381.68
	McMillan	1.404	1.182	0.296	19.3	57.2	129.97	212.21	307.74	405.21
	PIDeasy	1.192	1.359	0.172	18	61.9	124.62	281.86	296.49	488.04
	PIDeasyI	1.144	1.359	0.172	18.4	63	125.83	299.35	294.68	512.41
	ZN	2.076	1.356	0.339	16	57.5	66.28	129.70	192.53	254.02
	G-K	0.858	1.602	0.427	23.1	81	288.30	626.59	474.28	910.56
	PIDeasyII	2.703	1.603	0.426	13.1	60.7	***37.51***	***127.91***	***139.68***	***232.54***
	W-C	1.267	1.602	0.427	19.7	77.2	124.09	357.64	252.47	566.39
	s-MOEA1	2.703	1.595	0.430	13.1	60.9	37.50	126.84	139.54	231.10
	s-MOEA2	5.524	0.775	0.446	6.81	30.3	85.34	30.60	386.23	85.58

Table 6.7 Results for $G_3(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=0.1	AMIGO	0.863	1.698	0.513	23.8	57.1	647.60	1083.5	1051.2	1597.6
	IMC	1.524	2.435	0.461	18.8	58.7	306.76	633.35	602.37	937.93
	McMillan	1.216	2.116	0.529	21.2	60.5	361.46	735.47	652.04	1101.9
	PIDeasy	1.032	2.154	0.305	18.1	60.8	449.66	885.33	792.24	1299.1
	PIDeasyI	0.998	2.154	0.305	18.4	61.7	449.20	921.49	784.89	1339.9
	ZN	1.767	2.469	0.617	18.2	60.7	205.70	530.81	431.98	791.46
	G-K	0.839	2.578	0.796	23.9	79.7	756.67	1671.8	1062.2	2153.2
	PIDeasyII	2.347	2.580	0.794	15	60.5	***132.11***	***403.83***	322.51	***606.58***
	W-C	1.725	2.578	0.796	17.6	69.2	158.88	607.29	***321.87***	876.29
	s-MOEA1	2.435	2.612	0.866	14.2	61.8	127.59	403.37	314.74	600.27
	s-MOEA2	5.061	1.578	0.856	7.96	31.6	240.39	100.36	723.91	197.30
n=0.2	AMIGO	0.813	1.750	0.547	20.2	57.8	683.88	1223.3	1092	1773.2
	IMC	1.464	2.485	0.488	15	58.1	332.91	697.56	647.55	1032.7
	McMillan	1.134	2.267	0.567	17.7	63.3	347.36	858.87	618.19	1251.5
	PIDeasy	0.963	2.181	0.325	15.8	61	488.63	1002.4	843.49	1458.5
	PIDeasyI	0.935	2.181	0.325	16	61.9	488.71	1039	836.32	1499.5
	ZN	1.635	2.669	0.667	14.6	63.5	209.47	687.82	429.69	991.78
	G-K	0.807	2.563	0.777	20.4	77.8	778.83	1756	1094.9	2268.1
	PIDeasyII	1.910	2.565	0.776	12.9	61.4	***164.80***	***523.56***	372.54	***782.51***
	W-C	1.589	2.578	0.796	14.5	66.6	176.77	671.31	***361.16***	974.80
	s-MOEA1	1.992	2.614	0.850	12.1	63.2	156.08	522.82	358.26	773.65
	s-MOEA2	4.102	1.630	0.916	5.64	30.3	352.64	147.23	1015.0	289.38
n=0.5	AMIGO	0.686	1.885	0.646	15.4	59.8	814.33	1656.9	1246.2	2314.5
	IMC	1.264	2.610	0.565	10.1	57.5	407.46	947.61	769.49	1393.6
	McMillan	0.924	2.731	0.683	13	71.6	615.93	1610.4	910.15	2151.3
	PIDeasy	0.789	2.233	0.386	12.5	61.9	612.43	1414.8	993.93	2013.9
	PIDeasyI	0.773	2.233	0.386	12.7	62.5	614.38	1453.6	989.92	2055.6
	ZN	1.296	3.304	0.826	9.65	74.1	554.53	1381.2	829.24	1824.8
	G-K	0.715	2.477	0.711	15.2	72.3	821.41	2003	1169.1	2624.1

	PIDeasyII	1.147	2.480	0.711	11.1	62.1	319.89	1001.2	579.67	1458
	W-C	1.246	2.477	0.711	10.4	59.7	***304.87***	***884.91***	***579.42***	***1315***
	s-MOEA1	1.481	2.631	0.848	8.48	61.1	246.26	797.19	518.78	1183
	s-MOEA2	2.495	2.024	0.928	3.73	32.4	662.81	351.29	1675.6	707.93
n=1	AMIGO	0.566	2.063	0.776	11.8	60.9	1112.2	2590.6	1607	3532.6
	IMC	1.050	2.789	0.658	6.5	55.9	***571.17***	***1453.8***	***1057.9***	***2150.1***
	McMillan	0.721	3.386	0.847	9.4	79.4	1856.1	3728	2330	4608.5
	PIDeasy	0.626	2.301	0.468	9.99	61.6	832.11	2212.9	1268.7	3075.8
	PIDeasyI	0.617	2.301	0.468	10.1	62	838.66	2255.1	1271.1	3119.3
	ZN	0.975	4.250	1.062	5.76	86.7	1915.4	3627.6	2485.3	4417.3
	G-K	0.60	2.332	0.602	11.1	64.4	873.44	2429.2	1278.6	3290.9
	PIDeasyII	0.674	2.337	0.603	10.1	61.3	743.43	2073.4	1145.6	2899.4
	W-C	0.913	2.332	0.602	7.45	50.8	757.29	1558.4	1296.1	2398.5
	s-MOEA1	1.030	2.621	0.846	6.43	57.9	405.45	1386.8	768.74	2061.1
	s-MOEA2	1.391	2.374	0.923	3.63	42.9	755.40	902.80	1657.3	1613
n=2	AMIGO	0.459	2.306	0.932	7.96	59	326.92	880.48	452.18	1179
	IMC	0.860	3.048	0.758	2.66	47.5	377.40	***617.65***	718.11	1080.7
	McMillan	0.535	4.309	1.077	5.44	83.4	1151.6	1977.9	1330.8	2296.5
	PIDeasy	0.483	2.395	0.575	7.23	57.7	***251.61***	765.60	***365.33***	***1060***
	PIDeasyI	0.479	2.395	0.575	7.3	58	253.24	772.55	366.37	1066.4
	ZN	0.690	5.672	1.418	1.5	92.6	1423	2216.9	2396.2	2977
	G-K	0.465	2.096	0.446	6.44	50.8	376.19	891.07	548.12	1261.7
	PIDeasyII	0.384	2.107	0.447	8.14	58.4	350.75	930.44	482.86	1249
	W-C	0.614	2.096	0.446	4.02	36.6	663.57	1173.9	1025.6	1784.1
	s-MOEA1	0.681	2.666	0.843	4.62	53.2	151.23	570.61	262.52	849.29
	s-MOEA2	0.765	2.597	0.886	3.53	47.1	187.60	491.53	353.02	788.19
n=5	AMIGO	0.368	2.709	1.115	1.82	45.1	***654.49***	***2306.5***	1198	3781.6
	IMC	0.698	3.513	0.860	-3.2	-57.9	Unstable			
	McMillan	0.369	5.750	1.438	-0.698	22.2	Unstable			
	PIDeasy	0.364	2.581	0.720	2.62	39.6	674.28	2545.4	1021.2	3822.8
	PIDeasyI	0.363	2.581	0.720	2.65	39.9	662.25	2516.3	***1002.4***	***3771.6***
	ZN	0.447	8.050	2.013	-4.86	67.4	Unstable			
	G-K	0.319	1.773	0.248	0.411	6.72	45668	85327	63009	117630
	PIDeasyII	0.207	1.846	0.242	4.44	46.9	1203.4	3455.5	1552.6	4505.6
	W-C	0.370	1.773	0.248	-0.889	-15.3	Unstable			
	s-MOEA1	0.352	2.707	0.833	3.04	45.9	396.36	2013.2	604.76	2941.5
	s-MOEA2	0.363	2.708	0.825	2.78	44.2	415.64	1945	656.43	2931.4

Table 6.8 Results for $G_4(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=0.1	AMIGO	0.245	0.479	0.126	11.1	71.1	208.77	471.30	474.56	901.32
	IMC	0.480	0.605	0.084	5.75	63.6	***77.12***	***271.88***	***363.25***	***785.62***
	McMillan	0.117	1.220	0.305	11.2	90.5	4504.6	4995.3	5019.6	5592.6
	PIDeasy	0.210	0.370	0.094	10.3	65.7	132.12	374.32	385.53	785.63
	PIDeasyI	0.209	0.370	0.094	10.3	65.7	132.27	374.53	385.74	785.88
	ZN	0.120	2.015	0.504	7.89	93	6127	6516.3	6569	7037.3
	G-K	0.108	0.120	0.015	5.03	39.6	480.12	788.92	1255.9	1830.2

	PIDeasyII	0.060	0.167	0.008	13.3	70.9	476.12	833.44	851.63	1345.1
	W-C	0.060	0.120	0.015	10.1	61.8	228.05	511.72	567.02	996.77
	s-MOEA1	0.490	0.557	0.223	0.95	61.2	30.68	204.02	205.22	604.04
	s-MOEA2	0.501	0.556	0.230	0.55	60.6	30.97	202.80	210.82	612.67
n=0.2	AMIGO	0.290	0.552	0.202	11.3	69.8	185.47	462.67	445.40	891.98
	IMC	0.561	0.704	0.144	5.71	63.2	***79.42***	***280.07***	***371.38***	***785.62***
	McMillan	0.217	1.319	0.330	10.2	90.8	2554.8	3088.5	3048.8	3709.7
	PIDeasy	0.266	0.471	0.141	10.4	65.8	134.61	396.84	389.46	815.03
	PIDeasyI	0.265	0.471	0.141	10.4	65.8	135.14	397.62	390.18	815.95
	ZN	0.241	2.010	0.502	6.21	95.2	3639.4	4142.7	4149.4	4778.1
	G-K	0.170	0.207	0.006	5.69	43.3	406.21	713.58	1058.9	1605
	PIDeasyII	0.104	0.287	0.002	13	71	460.33	841.70	831.46	1358.8
	W-C	0.103	0.207	0.006	10	61.7	234.88	542.43	578.36	1040.9
	s-MOEA1	0.535	0.634	0.247	4.56	61.2	39.18	228.34	238.36	631.12
	s-MOEA2	0.558	0.633	0.260	3.8	60.2	41.44	225.84	253.78	649.37
n=0.5	AMIGO	0.421	0.763	0.313	11.2	66.8	171.57	493.97	439.57	914.99
	IMC	0.793	1.00	0.250	5.74	62.7	***81.67***	***285.59***	***378.75***	***687.83***
	McMillan	0.467	1.50	0.375	9.16	88.8	907.02	1401.6	1276.7	1915.1
	PIDeasy	0.433	0.767	0.196	10.3	65.5	135.72	449.94	393.40	857.23
	PIDeasyI	0.431	0.767	0.196	10.3	65.7	138.13	454.15	396.63	862.30
	ZN	0.589	2.019	0.505	5.33	97.4	1127.2	1617.6	1614.2	2175.8
	G-K	0.339	0.513	0.015	7.64	52.6	264.96	565.77	701.96	1156.2
	PIDeasyII	0.234	0.693	0.013	12.9	74	580.47	1070.5	969.33	1605.9
	W-C	0.256	0.513	0.015	10.1	61.7	232.16	585.22	573.97	1074.4
	s-MOEA1	0.719	0.891	0.293	6.56	60.8	53.60	275.90	282.92	634.11
	s-MOEA2	0.816	0.877	0.320	5.39	56.9	73.51	246.88	367.85	645.57
n=2	AMIGO	1.091	1.665	0.437	10.9	59.5	358.96	671.84	727.86	1033.6
	IMC	1.776	2.494	0.401	6.64	65.5	***74.68***	541.26	***324.60***	809.82
	McMillan	1.586	1.777	0.444	7.52	57	186.82	402.03	471.82	675.54
	PIDeasy	1.348	2.266	0.261	9.56	64.5	119.53	659.04	369.58	972.76
	PIDeasyI	1.283	2.266	0.261	9.99	65.8	139.05	707.66	397.42	1032.2
	ZN	2.375	2.011	0.503	3.39	54.6	256.03	***258.97***	986.20	***515.99***
	G-K	0.682	2.052	0.061	14	71.9	567.37	1517.2	963.28	2012.1
	PIDeasyII	0.947	2.065	0.061	11.1	65.2	238.45	922.86	559.26	1316
	W-C	1.049	2.052	0.061	10.2	62.3	212.12	782.65	537.54	1148.2
	s-MOEA1	1.802	2.381	0.342	6.89	61	70.83	484.72	322.13	747.22
	s-MOEA2	2.421	1.638	0.374	4.34	40.5	189.03	176.57	666.23	396.55
n=5	AMIGO	2.419	2.937	0.476	10.9	58	135.21	133.04	226.61	183.19
	IMC	2.734	5.482	0.458	9.71	73.7	48.35	298.20	102.94	355.25
	McMillan	3.698	1.905	0.476	7.44	41	118.43	58.42	223.56	97.94
	PIDeasy	3.214	5.256	0.291	9.24	64.3	***22.21***	224.35	***70.72***	273.75
	PIDeasyI	2.773	5.256	0.291	10.5	67.9	36.19	269.96	90.67	326.09
	ZN	5.917	2.019	0.505	3.09	38.5	70.24	***23.57***	233.61	***49.31***
	G-K	0.868	5.204	0.218	20.6	82.3	610.14	1336.2	749.97	1499.1
	PIDeasyII	3.007	5.210	0.218	9.81	63.5	26.98	239.72	80.92	292.36
	W-C	2.905	5.204	0.218	10.1	64.4	29.01	249.74	83.19	303.92
	s-MOEA1	4.215	5.446	0.383	6.49	60	15.77	173.0	69.55	212.82
	s-MOEA2	5.590	2.096	0.403	4.24	35.3	52.88	21.78	153.11	43.99

n=10	AMIGO	4.456	4.278	0.506	11.1	59.2	212.11	120.89	317.36	155.51
	IMC	3.444	10.405	0.496	13.2	79.7	142.80	774.93	220.54	849.41
	McMillan	6.911	2.024	0.506	7.65	36.7	190.13	43.71	335.72	71.08
	PIDeasy	6.095	10.173	0.315	9.39	65.4	28.05	379.24	77.58	421.08
	PIDeasyI	4.573	10.173	0.315	11.9	71.6	63.94	528.95	126.21	584.35
	ZN	11.351	2.090	0.522	3.19	36.1	77.18	***14.10***	230.77	***27.92***
	G-K	0.941	10.343	0.457	24.8	86.7	2159	4060.3	2389.5	4290.6
	PIDeasyII	7.088	10.349	0.456	7.23	66	***19.68***	330.25	***66.06***	366.35
	W-C	6.282	10.343	0.457	8.27	69.1	29.74	378.06	76.78	418.40
	s-MOEA1	7.034	10.0	0.361	7.95	62.8	25.25	312.75	71.95	348.80
	s-MOEA2	10.0	2.587	0.333	5.16	33.5	65.57	14.91	164.96	27.66

Table 6.9 Results for $G_5(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=0.1	AMIGO	0.261	0.531	0.165	11.2	70.5	223.93	518.31	504.03	972.12
	IMC	0.509	0.673	0.113	5.71	63.1	***93.71***	***312.69***	***413.07***	***861.95***
	McMillan	0.154	1.324	0.331	12	90.6	3773.6	4318.7	4295.4	4947.1
	PIDeasy	0.230	0.427	0.121	10.3	65.6	151.70	426.01	421.51	863.46
	PIDeasyI	0.229	0.427	0.121	10.3	65.6	151.98	426.41	421.87	863.92
	ZN	0.164	2.116	0.529	8.1	93.8	5210.3	5672.8	5694.3	6261.4
	G-K	0.159	0.192	0.044	5.49	42.2	446.45	772.58	1145.8	1733.2
	PIDeasyII	0.086	0.257	0.038	13.6	71.8	560.72	970.04	955.83	1514.9
	W-C	0.095	0.192	0.044	9.93	61.3	243.95	557.28	595.19	1070.4
	s-MOEA1	0.501	0.613	0.244	4.31	61	42.47	241.26	249.42	668.17
	s-MOEA2	0.514	0.611	0.258	3.68	60.4	43.40	239.94	256.34	679.95
n=0.2	AMIGO	0.320	0.659	0.259	11.2	68.8	217.05	557.28	506.43	1028.7
	IMC	0.612	0.846	0.192	5.66	62.5	***116.01***	***357.57***	472.77	***919.14***
	McMillan	0.276	1.505	0.376	11.4	90.6	2196.8	2794.1	2687	3428.7
	PIDeasy	0.302	0.589	0.175	10.3	65.4	171.62	496.49	***456.97***	960.19
	PIDeasyI	0.302	0.589	0.175	10.3	65.5	172.62	498.03	458.24	961.96
	ZN	0.319	2.209	0.552	7.66	96.2	3001.1	3573.5	3535.8	4237.6
	G-K	0.280	0.396	0.099	6.81	48.8	325.79	661.38	844.04	1410.9
	PIDeasyII	0.172	0.407	0.097	11.3	65.7	286.59	671.30	627.12	1185.1
	W-C	0.196	0.396	0.099	9.91	61.2	245.19	598.52	598.11	1133.7
	s-MOEA1	0.550	0.744	0.290	6.84	61	63.25	302.20	305.16	733.92
	s-MOEA2	0.595	0.748	0.312	6.15	59.4	68.38	294.97	340.12	761.90
n=0.5	AMIGO	0.464	1.031	0.415	11.7	65.3	299.64	798.26	648.78	1323.9
	IMC	0.869	1.364	0.339	6.27	61.1	***170.09***	***462.24***	568.06	***937.80***
	McMillan	0.544	1.913	0.478	9.77	86.9	1042.7	1723	1460.2	2291.7
	PIDeasy	0.490	1.074	0.256	10.3	64.7	229.28	712.51	***551.82***	1210.3
	PIDeasyI	0.486	1.074	0.256	10.3	64.9	232.90	721.52	555.45	1220.4
	ZN	0.703	2.511	0.628	6.17	96.1	1184.3	1827.8	1688.4	2372.6
	G-K	0.495	0.994	0.253	9.82	60.6	254.10	699.17	614.92	1232.2
	PIDeasyII	0.435	0.997	0.253	11	64.4	279.75	784.26	626.05	1310.7
	W-C	0.493	0.994	0.253	9.85	60.7	254.29	700.65	614.61	1233.1
	s-MOEA1	0.809	1.226	0.419	6.89	60	118.93	436.57	434.68	858.43
	s-MOEA2	0.948	1.220	0.476	5.36	56.6	162.23	389.66	597.14	857.29

n=2	AMIGO	0.866	2.748	0.828	17	58	1431.8	2601.4	2128.5	3343.1
	IMC	1.528	3.944	0.745	12.1	60	557.50	1627.8	1044.7	2109.9
	McMillan	1.221	3.416	0.854	13.9	61.6	758.95	1839.4	1268.4	2404.2
	PIDeasy	1.036	3.491	0.493	14.4	61.7	936.40	2234	1514.6	2866.7
	PIDeasyI	1.002	3.491	0.493	14.7	62.7	943.75	2349.7	1513.7	2995.5
	ZN	1.776	3.984	0.996	10.3	61.8	307.48	1369.5	683.86	1794.9
	G-K	0.797	3.971	1.016	17.2	77.2	1825.3	4101.4	2411.3	4840.8
	PIDeasyII	1.794	3.974	1.015	10.1	61.9	301.24	1345.2	***672.52***	1766.3
	W-C	1.971	3.971	1.016	9.28	59.2	***290.94***	***1182.9***	682.30	***1573***
	s-MOEA1	2.139	4.153	1.116	8.16	60.2	263.73	1160.4	669.18	1529.5
	s-MOEA2	3.688	2.679	1.096	3.73	30.8	952.35	369.94	2266.5	667.37
n=5	AMIGO	1.144	6.026	1.535	24.3	55.6	1359.2	1853.3	1645.1	2121.1
	IMC	1.830	9.112	1.416	20.5	63.2	504.40	1365.9	673.82	1530.5
	McMillan	1.671	6.231	1.558	21	51.6	1029.4	1179.1	1298.5	1380.8
	PIDeasy	1.422	8.319	0.917	22.8	61.2	758.07	1619.6	970.45	1816.8
	PIDeasyI	1.348	8.319	0.917	23.2	62.5	772.69	1744.6	982.21	1951
	ZN	2.517	7.016	1.754	16.8	53.1	561.27	647.44	756.32	770.35
	G-K	0.90	9.904	2.517	23.3	83.2	2050.1	4221.3	2284.6	4518.1
	PIDeasyII	3.786	9.909	2.512	10.9	63.1	89.19	641.04	***172.58***	723.01
	W-C	4.717	9.904	2.517	8.93	56.5	***86.35***	***492.37***	181.73	***557.44***
	s-MOEA1	4.790	10.0	2.489	8.88	55.9	84.63	492.64	181.21	557.12
	s-MOEA2	9.253	5.761	1.882	5.13	30.7	219.98	82.39	447.18	108.73
n=10	AMIGO	1.310	11.358	2.660	30.4	54.4	4773.2	5698	5308.4	6146.1
	IMC	1.987	17.699	2.480	27.2	64.5	1795.6	4625.5	2089.1	4907.2
	McMillan	1.937	10.749	2.687	26.9	47.7	4225.2	3586.6	4787.8	3943.1
	PIDeasy	1.654	16.353	1.595	31	61.1	2519.1	5089.7	2886.3	5410.8
	PIDeasyI	1.548	16.353	1.595	31.6	62.5	2584.6	5566.3	2948.3	5906.6
	ZN	2.960	11.933	2.983	22.5	50.3	2365.8	1865.1	2766.5	2071.1
	G-K	0.949	19.835	5.089	28.1	85.7	7792.9	15661	8219.4	16209
	PIDeasyII	5.952	19.843	5.079	12.1	64.8	***150.23***	1508.6	***246.97***	1602.9
	W-C	9.562	19.835	5.089	8.01	49.8	158.67	***886.27***	297.93	***941.46***
	s-MOEA1	10.0	10.0	4.463	8.76	50.4	475.53	316.19	631.51	358.36
	s-MOEA2	10.0	10.0	2.810	12.4	41.5	679.92	217.64	923.46	256.62

Table 6.10 Results for $G_6(s)$

	Rule	K_P	T_I	T_D	GM	PM	SP	LD	SP'	LD'
n=1	G-K	0.215	0.261	3.670	11.7	44.4	571.96	4251.9	1144.1	8419.8
	PIDeasyII	0.125	0.261	3.674	16.5	59	***180.26***	***2013.5***	***326.21***	***3943.8***
	W-C	0.134	0.261	3.670	15.9	57.2	193.12	2169.0	360.82	4256.9
	s-MOEA1	0.092	0.225	4.405	17.4	60.4	145.57	1510.9	236.17	2943.8
	s-MOEA2	0.216	0.585	3.828	10.4	33.4	457.45	670.30	907.39	1378.0
n=2	G-K	0.156	0.143	1.528	7.63	44.4	431.89	1675.4	1021.0	4747.2
	PIDeasyII	0.083	0.143	1.532	13.3	59.4	***404.57***	1214.2	***757.11***	3289.6
	W-C	0.081	0.143	1.528	13.4	59.9	409.85	***1213.7***	761.61	***3276.2***
	s-MOEA1	0.081	0.130	1.704	13.3	57	405.58	1178.8	759.26	3206.4
	s-MOEA2	0.063	0.097	2.686	18.8	52.2	534.91	1062.6	965.53	2822.9

System $G_1(s)$ has multiple equal poles and are very common. For large values of n, the system behaves like systems with long dead times. These types of systems have been used by controller manufacturers as test cases for a long time (Åström and Hägglund, 2000). Based on Table 6.5, it is evident that when n=1, the most ideal case is to have infinite high gains in order to have the best performance. However, in practice, that is not possible and advisable. As the lag gets longer, i.e., when n=8, IMC achieves the best overall performance among the tuning rules. It is being done with a compromise in the margins. However, s-MOEA1 shows that it is still possible to achieve a better performance than IMC without degrading the margins. s-MOEA2 indicates that, in order to have optimal load disturbance performance, then it must somehow sacrifice certain degree of stability robustness to achieve this. For system $G_1(s)$, PIDeasyII achieves a good compromise on set-point tracking and load disturbance performances, with a good degree of stability robustness. It also shows that for this type of system, using second-order modelling does help.

System $G_2(s)$ has four poles whose spacing is determined by parameter n. Since system $G_2(s)$ is similar to system $G_1(s)$, thus tuning rules based on second-order modelling should again outperform the rest. When n=0.1 and 0.2, it is impossible to achieve overall optimal performance as the s-MOEA search also fails to achieve that. When n=0.5, PIDeasyII can defeat all the other tuning rules on all performance indices. It can be seen based on s-MOEA1 that PIDeasyII is almost optimal with emphasis on set-point tracking performance. With the margins achieved on optimal load disturbance performance as indicated by s-MOEA2, it shows that PIDeasyII can at least provide a reasonable and safe initial setting for a PID controller.

System $G_3(s)$ has three equal poles and a right half plane zero. The difficulty of control increases with increasing n. Tuning rules based on second-order modelling dominates when n is less than 1.0. As n increases, even though PIDeasyII does not outperform the rest, it is still stable and robust based on the margins. This can be seen from Table 6.7 when n=1 and 2. This proves to be beneficial as when n=5, four out of nine of the tuning rules goes unstable. PIDeasyII has the best possible stability margins as compared with the rest.

System $G_4(s)$ is a classic system which has been used in many investigations of PID control. Many of the early tuning rules were derived based on this model. A drawback with the model is that it has slow roll-off at high frequencies. IMC totally outperforms the other tuning rules when n is less than 2. PIDeasy and PIDeasyI are just tailing behind IMC in these cases and with a better stability margins. The time-domain responses are also much smoother than IMC based on the four time-domain performance indices. When n=2, ZN outperform IMC on the load disturbance rejection. However, looking at the SP'/SP ratio, it is obvious that ZN time-domain response is quite oscillatory as compared with IMC or PIDeasyI.

System $G_5(s)$ is similar to system $G_4(s)$ except that it has more high frequency roll-off. IMC has again outperformed the other tuning rules when n is less than 2 and with PIDeasy and PIDeasyI tailing behind, just like the previous case. As n increase from 2 to 10, the beneficial of higher derivative gain takes effect as PIDeasyII and W-C dominates. In the case of n=10, s-MOEA1 and s-MOEA2 sub-optimal performances are due to the limit set on each parameters.

System $G_6(s)$ with a small damping factor, ζ, is not very suitable for PID control (Åström and Hägglund, 2000). It is easy to control if n is large. For this case, the n value is small; PIDeasyII still provides good performances and stability margins. Based on the results shown by s-MOEA1 and s-MOEA2, PIDeasyII performance is considered a good compromised solution based on the fact that it is a fixed structure formula.

The study in Section 5.2.5 shows that PIDeasyII should be capable of operating over a wide range of processes with good stability margins. This is further verified in all the 28 test cases conducted here. Next, in order to show that PIDeasyII is not just performing fine under computer simulations, it will be put to test on real processes.

The next three sections will cover three real processes of different characteristics. The first process is a DC motor where it has negligible transport delay and very fast responses. For this type of process, most tuning rules should not have any problem in controlling. Due to the expected small delay to time constant ratio, most of the tunings maybe too aggressive and thus might end up behaving like an on-off control. The second process is a heating system where it has a transport delay and the response is generally fast. In addition to the transport delay, this type of system is also vulnerable to atmospheric disturbances due to it nature. Thus, this will be a good test on the tuning rules robustness on disturbances and transport delay. The last process is a coupled tanks system where it has minimal transport delay and the response is very slow. This system can be tricky to control if the control scheme is still the same as the previous two. Normally, aggressive tuning is not advisable for this type of system, as it will cause the process output to oscillate or chatter. However, the control scheme will not be customised just to control this system, as the objective of the test is to confirm that if PIDeasy tuning methodology can provide good tuning based on the simplest control scheme. This is very important in the case where operator does not have a good understanding of a process.

6.5 LJ MS15 DC Motor Control Module

A DC motor can often be modelled simply as an LTI (Linear Time Invariant) plant where a small time-delay may appear. The motor is more difficult for velocity control, as it is a Type 0 system, where no integral element is evident in the system, and hence it will result in a steady-state error when following a step command. The LTI model of this system is given by the second-order differential equation:

$$\ddot{\omega}(t-0.06)+\left(\frac{JR+LB}{LJ}\right)\dot{\omega}(t-0.06)+\left(\frac{RB}{LJ}\right)\omega(t-0.06)=\left(\frac{K_T}{LJ}\right)v(t) \tag{6.9}$$

where

$v(t) \in [0V,5V]$	: the field control voltage with a saturation limit and allowing no braking voltage,
$\omega(t) \in R$	: the angular velocity calculated from a Gray-code shaft encoder,
$K_T = 13.5\ NmA^{-1}$	: the torque constant for a fixed armature constant,
$R = 9.2\ \Omega$	: the resistance of the winding,
$L = 0.25\ H$	: the winding inductance,
$J = 0.001\ kgm^2$	: the inertia of the motor shaft combined with a load, and
$B = 2.342 \times 10^3\ Nms$	: the friction coefficient of the shaft, changing to 1.34×10^3 Nms when an eddy current brake is released.

The basic principle of a velocity control system is that the controller attempts to keep the velocity constant by comparing the feedback signal with the command signal to compensate for changes which will occur when there are variations in load.

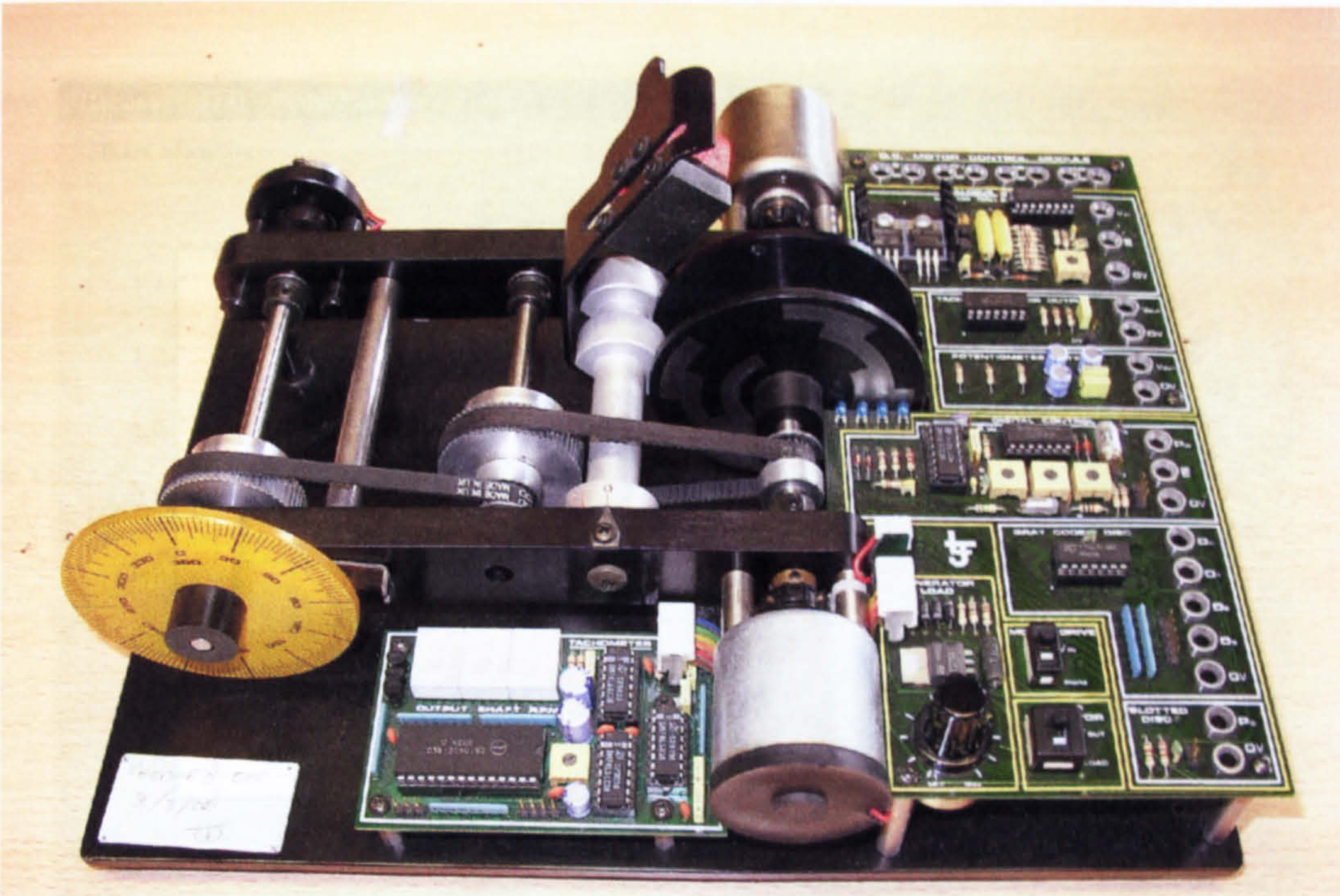

Figure 6.3 LJ MS15 DC Motor Control Module

6.5.1 Modelling and Tuning Process

An input and output relationship is first being established in order to verify its linearity. The MS15 DC motor linear behaviour is shown in Figure 6.4. Next, an open-loop step test is conducted on the system by injecting a 2 volts input with a sampling rate of 0.01 sec. The response captured is approximated by using the two models (5.1) and (5.2), as shown in Figure 6.5 and 6.6 respectively.

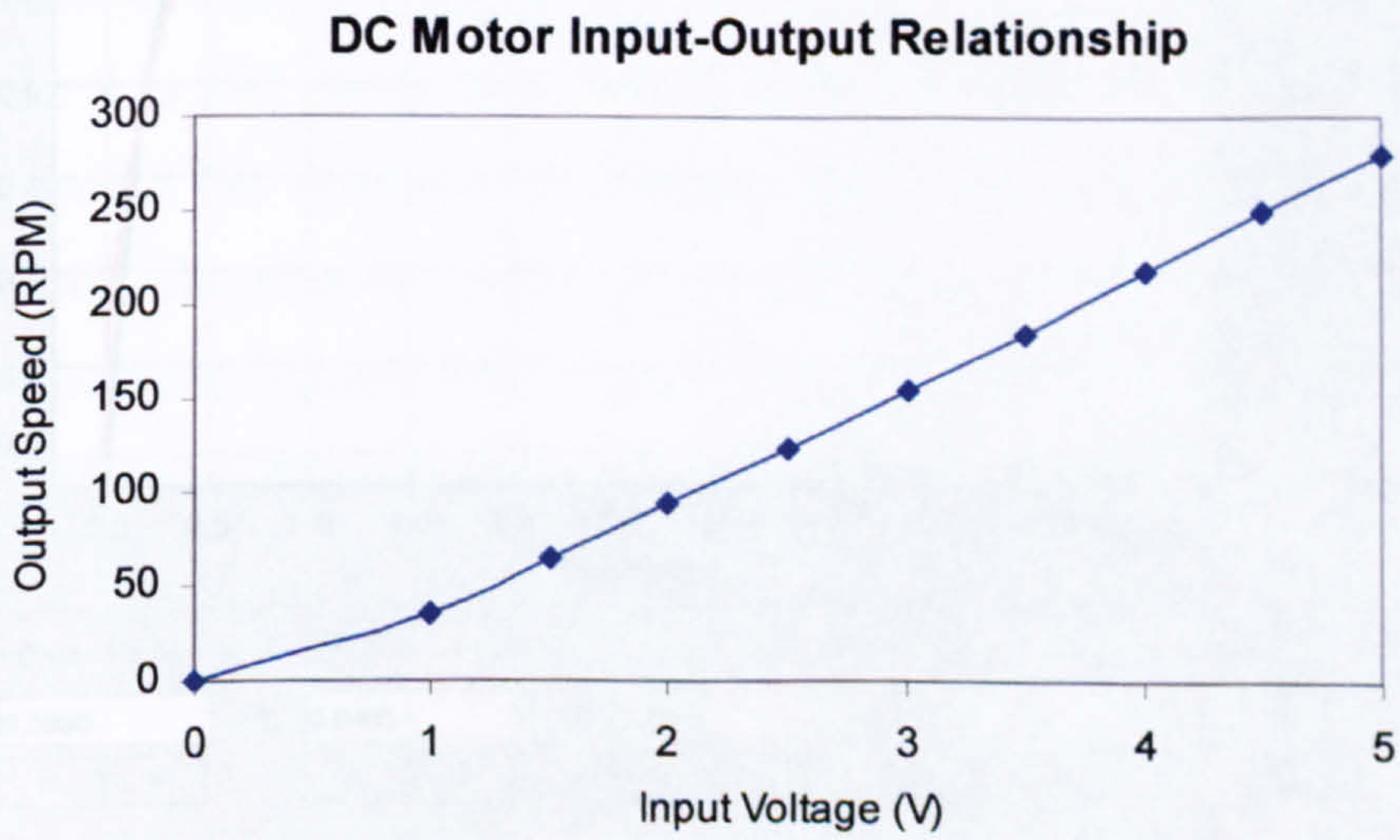

Figure 6.4 MS15 DC Motor Input-Output Relationship

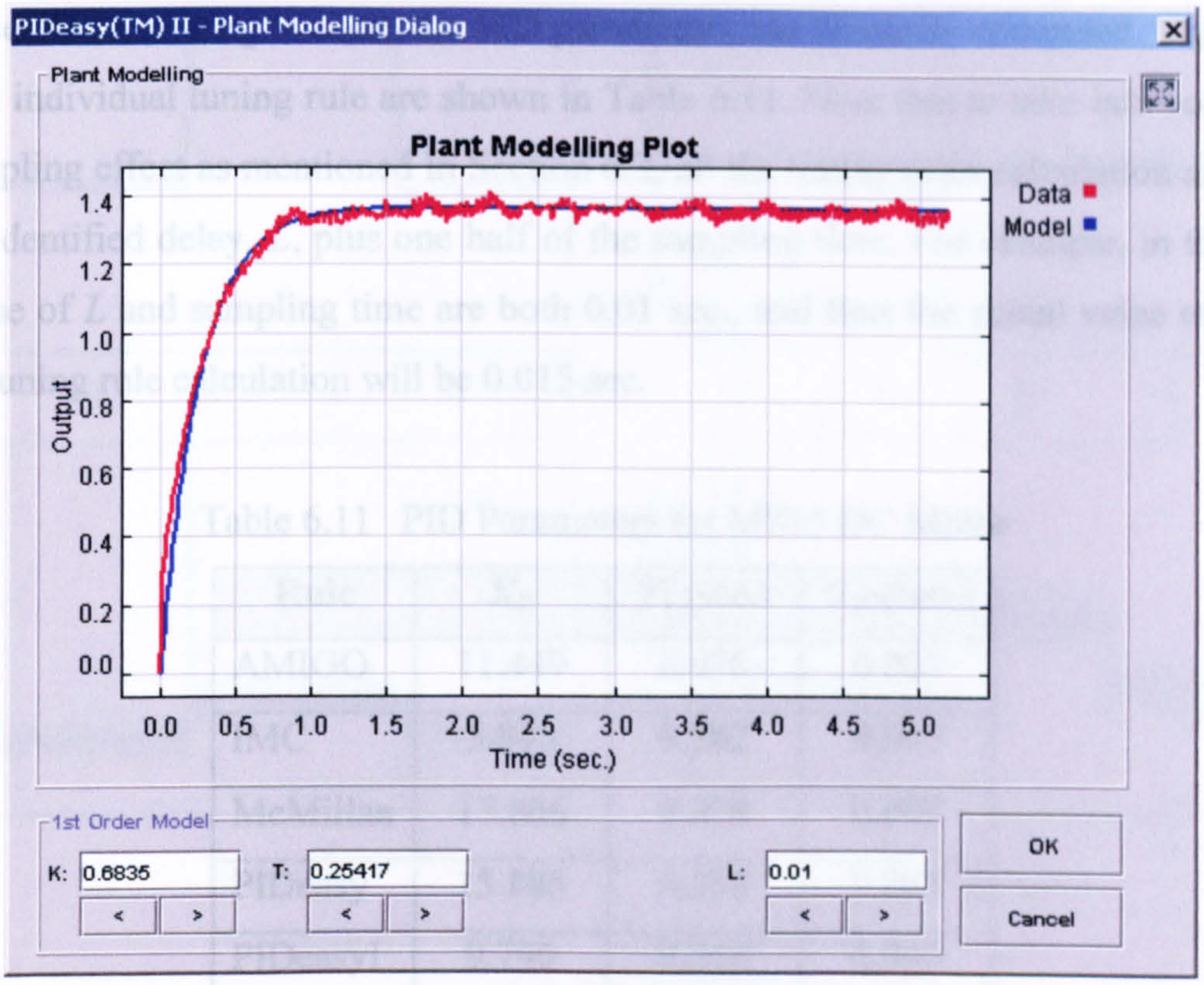

Figure 6.5 MS15 DC Motor Modelling using FOLPD Model

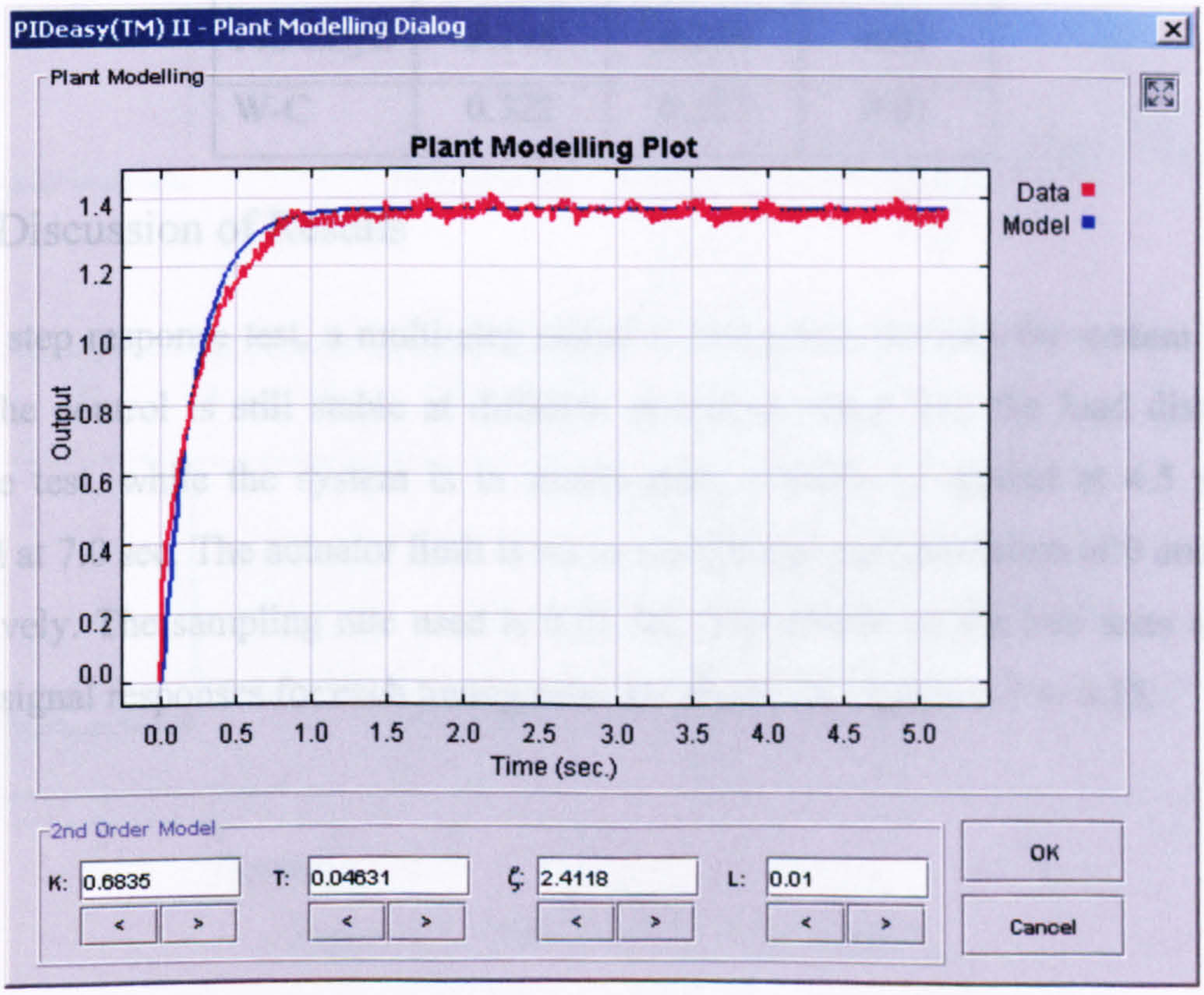

Figure 6.6 MS15 DC Motor Modelling using SOSPD Model

Once the modelling is done, the PID parameters can be easily computed. The results by each individual tuning rule are shown in Table 6.11. Note that to take into account of the sampling effect as mentioned in Section 6.2, all the tuning rules calculation are based on the identified delay, L, plus one half of the sampling time. For example, in this case, the value of L and sampling time are both 0.01 sec., and thus the actual value of L used by the tuning rule calculation will be 0.015 sec.

Table 6.11 PID Parameters for MS15 DC Motor

Rule	K_P	T_I **(sec.)**	T_D **(sec.)**
AMIGO	11.449	0.078	0.007
IMC	5.815	0.262	0.007
McMillan	17.866	0.029	0.007
PIDeasy	15.885	0.258	0.005
PIDeasyI	9.790	0.258	0.005
ZN	29.749	0.03	0.008
G-K	1.371	0.223	0.01
PIDeasyII	9.748	0.224	0.01
W-C	0.322	0.223	0.01

6.5.2 Discussion of Results

For the step response test, a multi-step signal is being injected into the system so as to verify the control is still stable at different operating range. For the load disturbance response test, while the system is in steady-state, a brake is applied at 4.5 sec. and released at 7.0 sec. The actuator limit is set to a minimum and maximum of 0 and 5 volts respectively. The sampling rate used is 0.01 sec. The results on the two tests and their control signal responses for each tuning rules are shown in Figures 6.7 to 6.15.

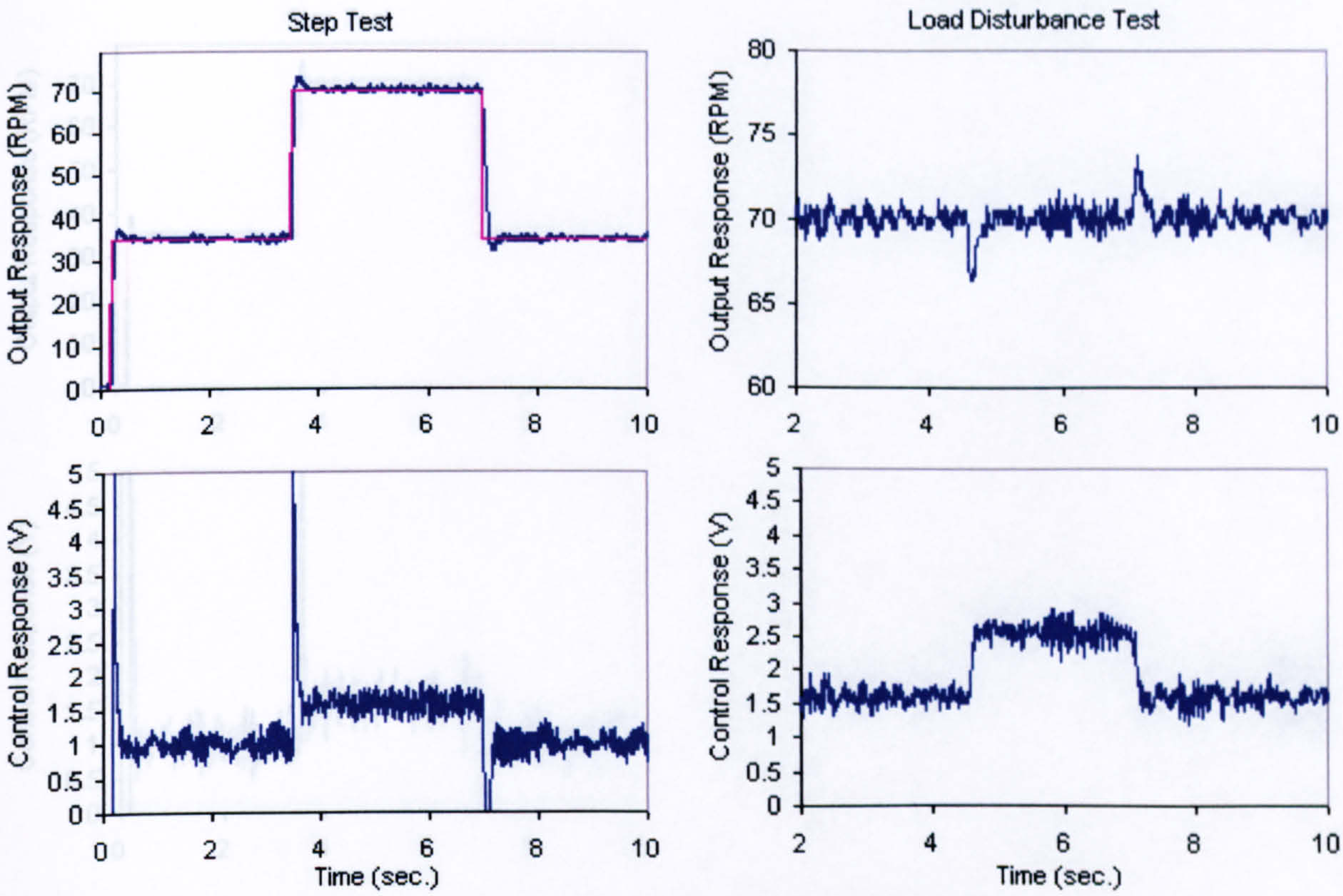

Figure 6.7 MS15 – AMIGO Test Results

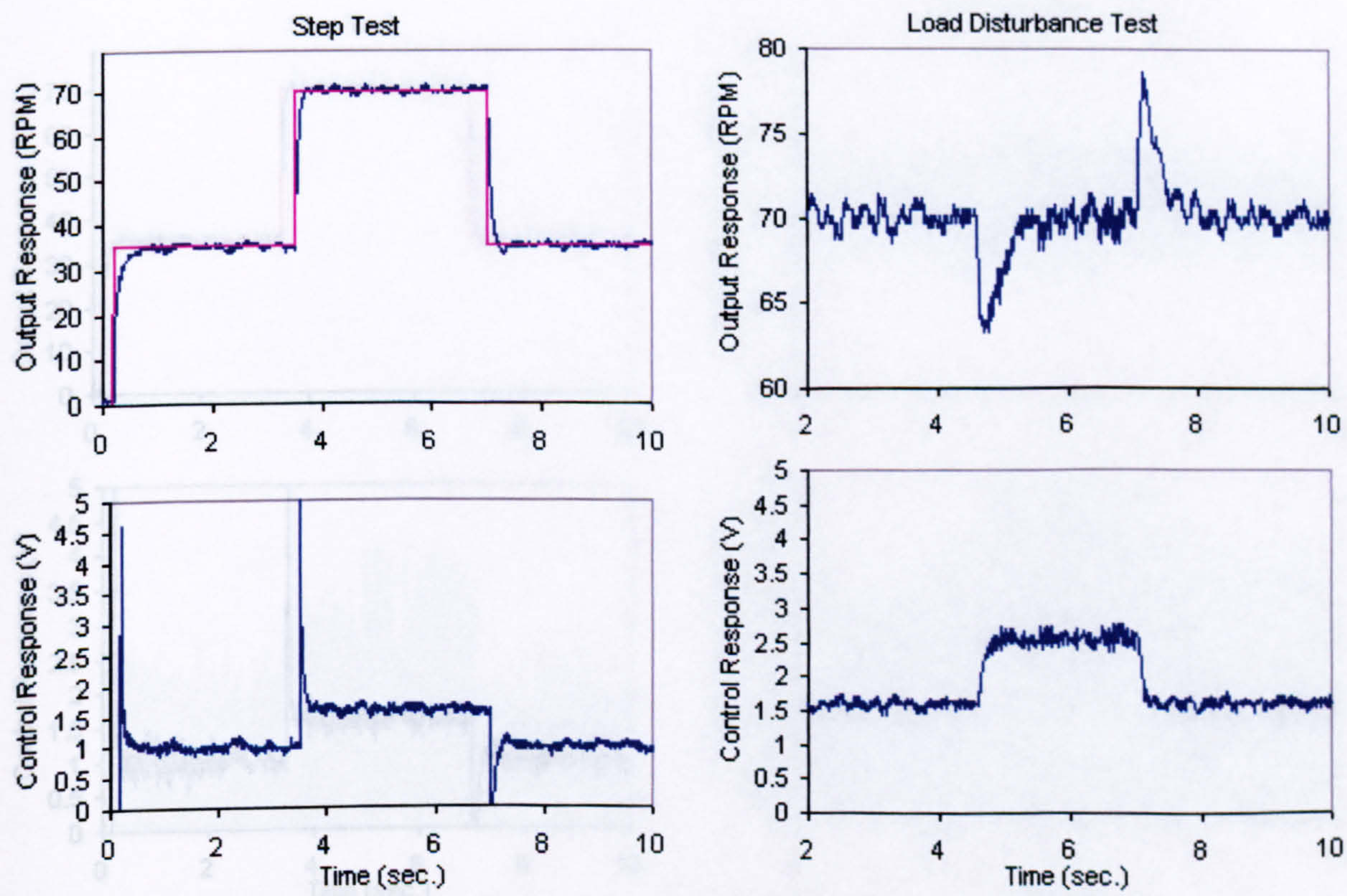

Figure 6.8 MS15 – IMC Test Results

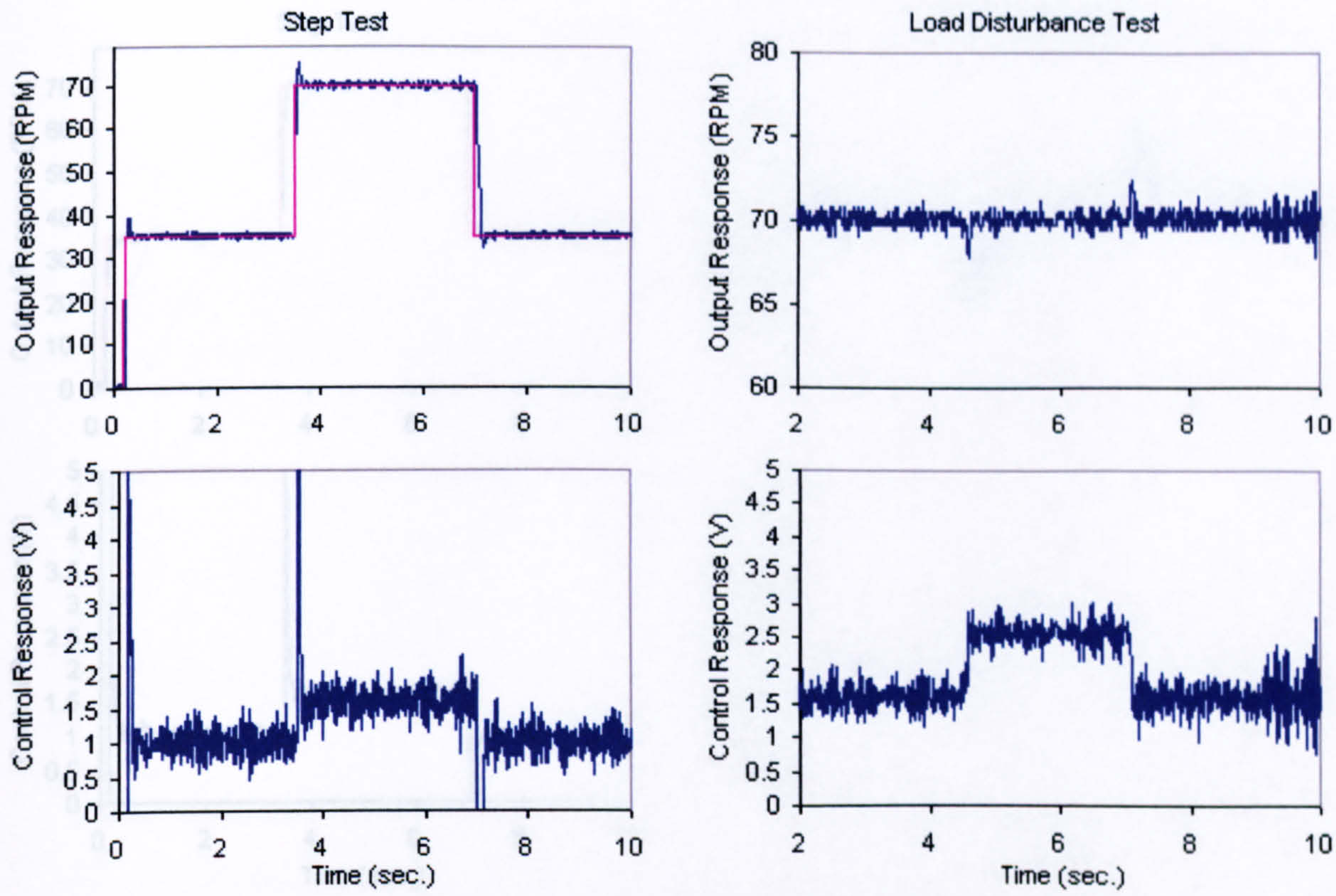

Figure 6.9 MS15 – McMillan Test Results

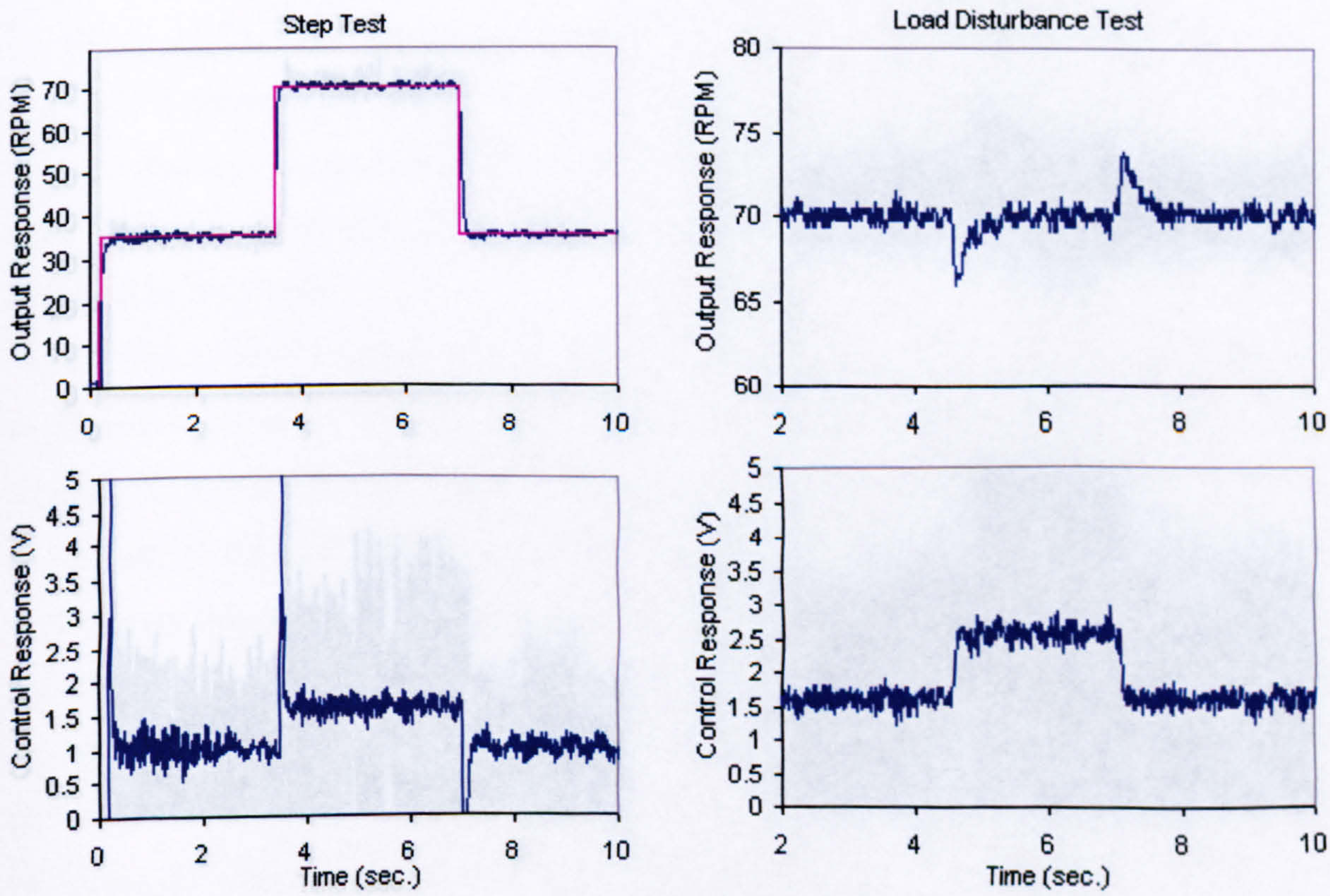

Figure 6.10 MS15 – PIDeasy Test Results

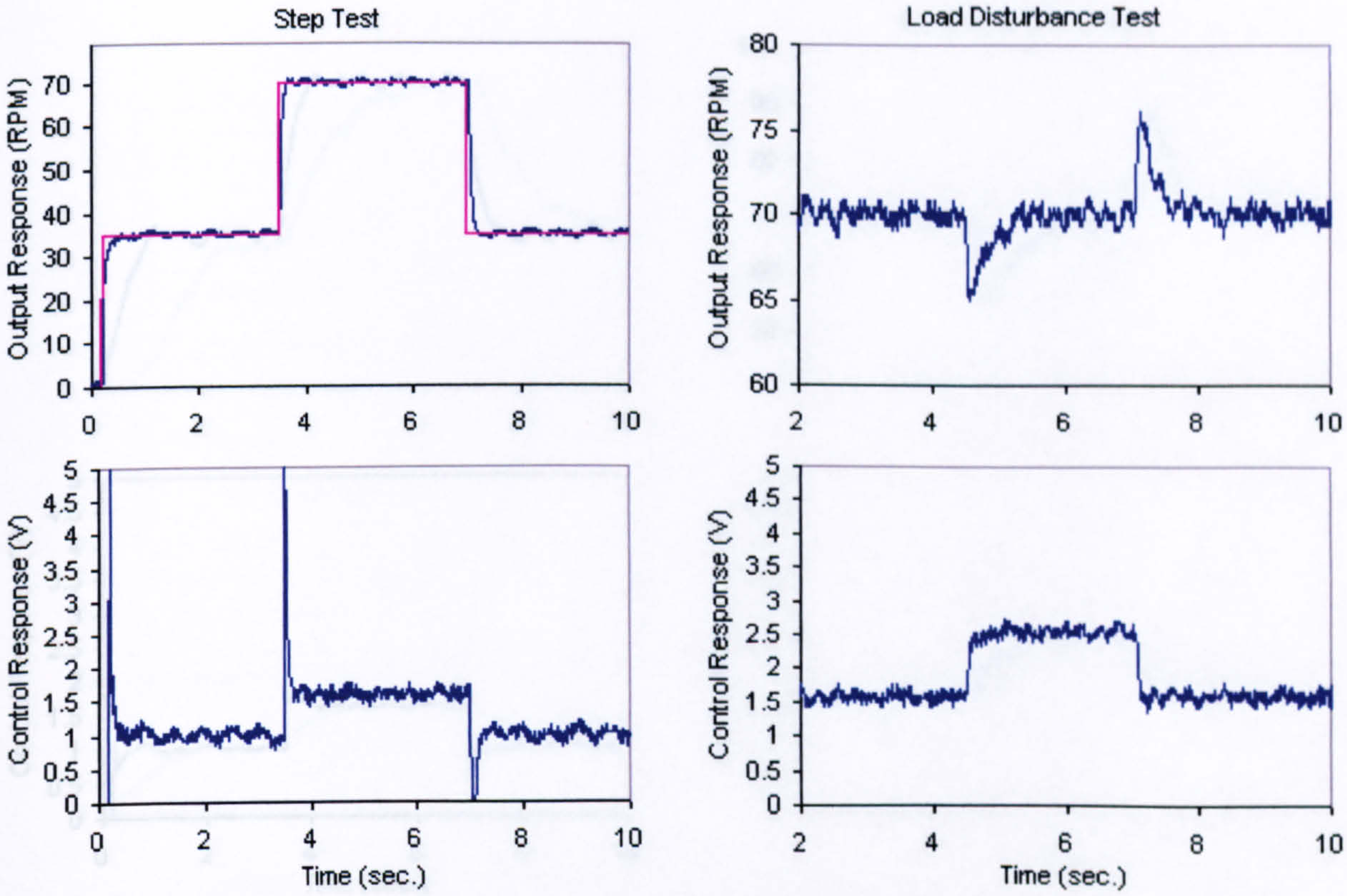

Figure 6.11 MS15 – PIDeasyI Test Results

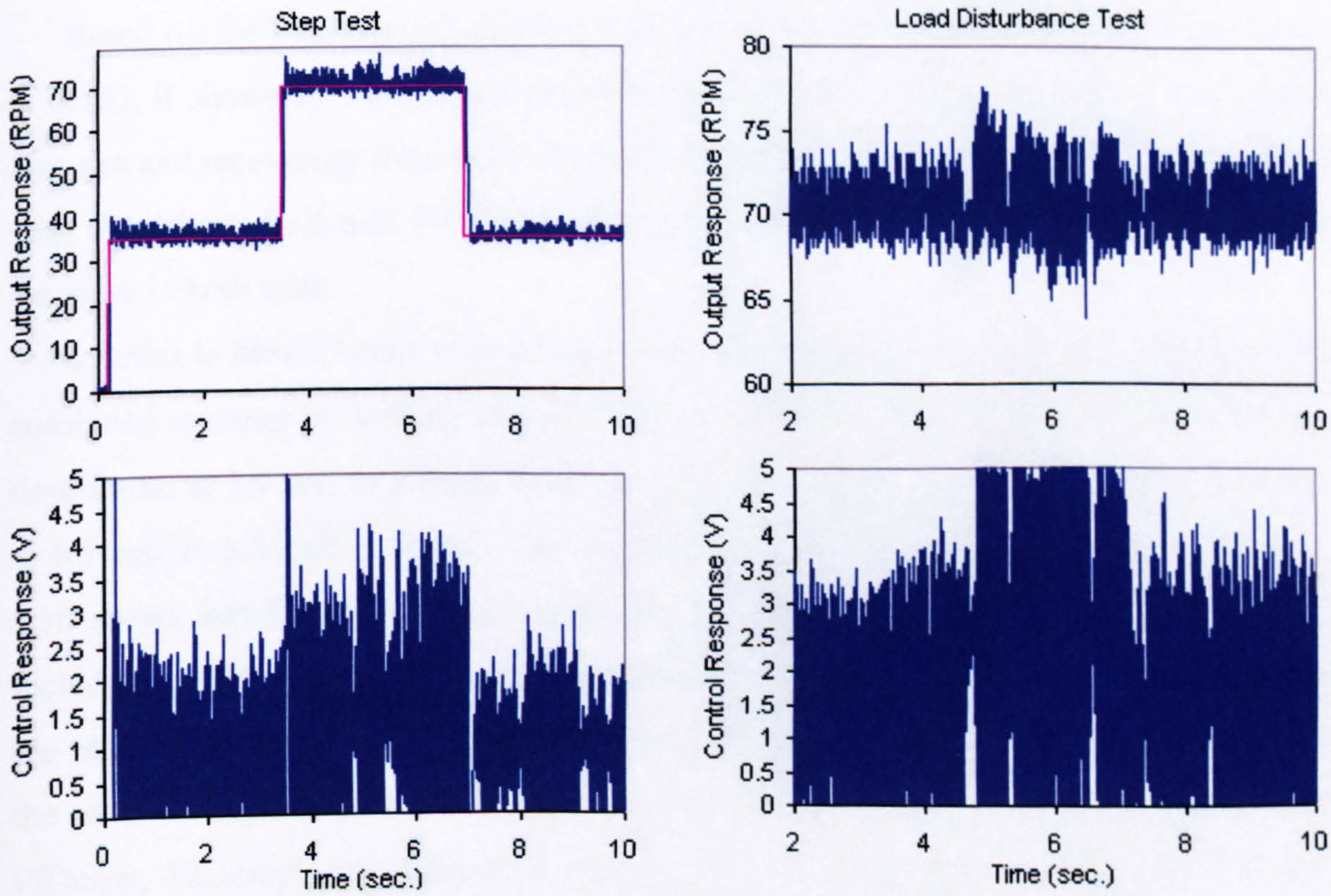

Figure 6.12 MS15 – ZN Test Results

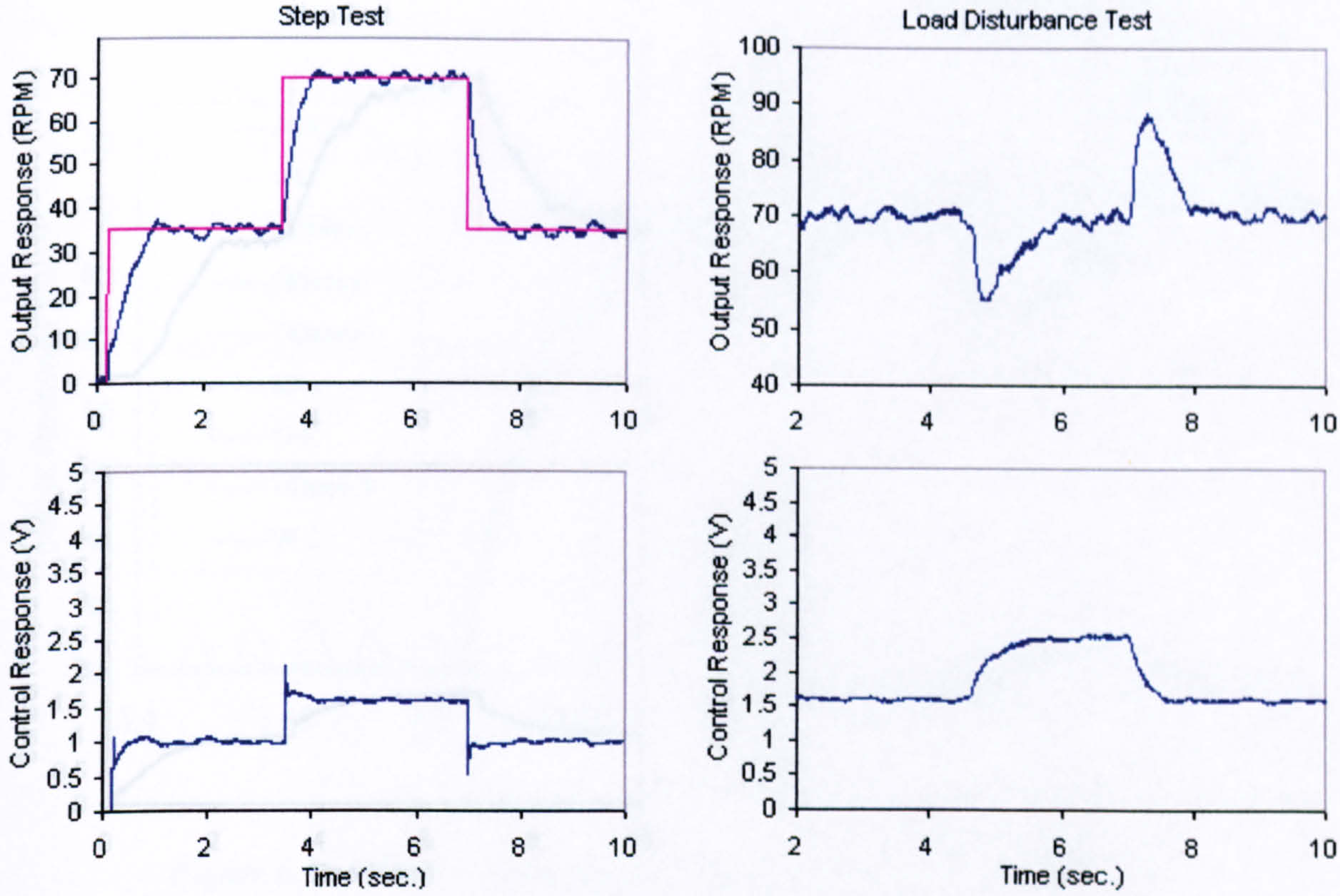

Figure 6.13 MS15 – G-K Test Results

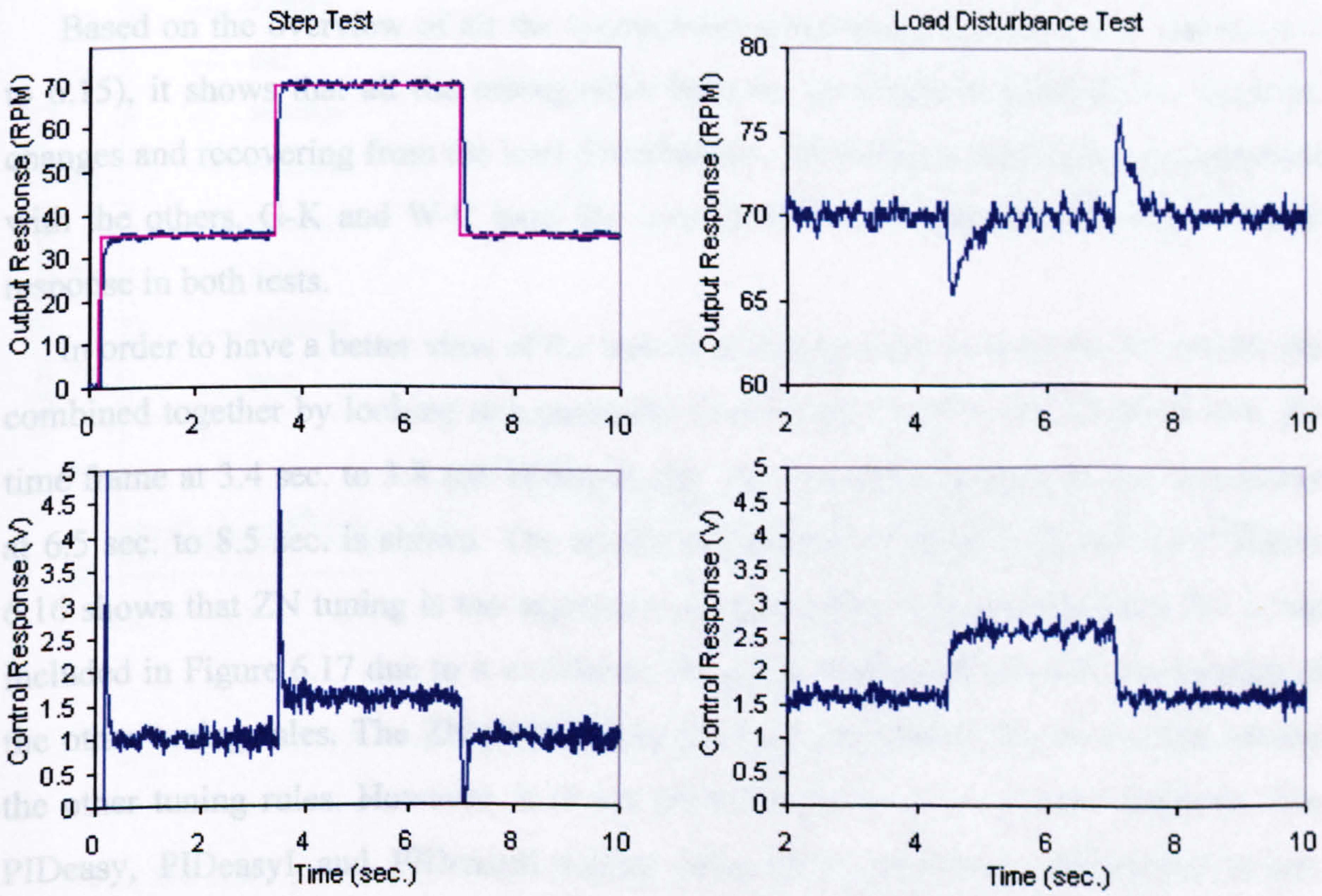

Figure 6.14 MS15 – PIDeasyII Test Results

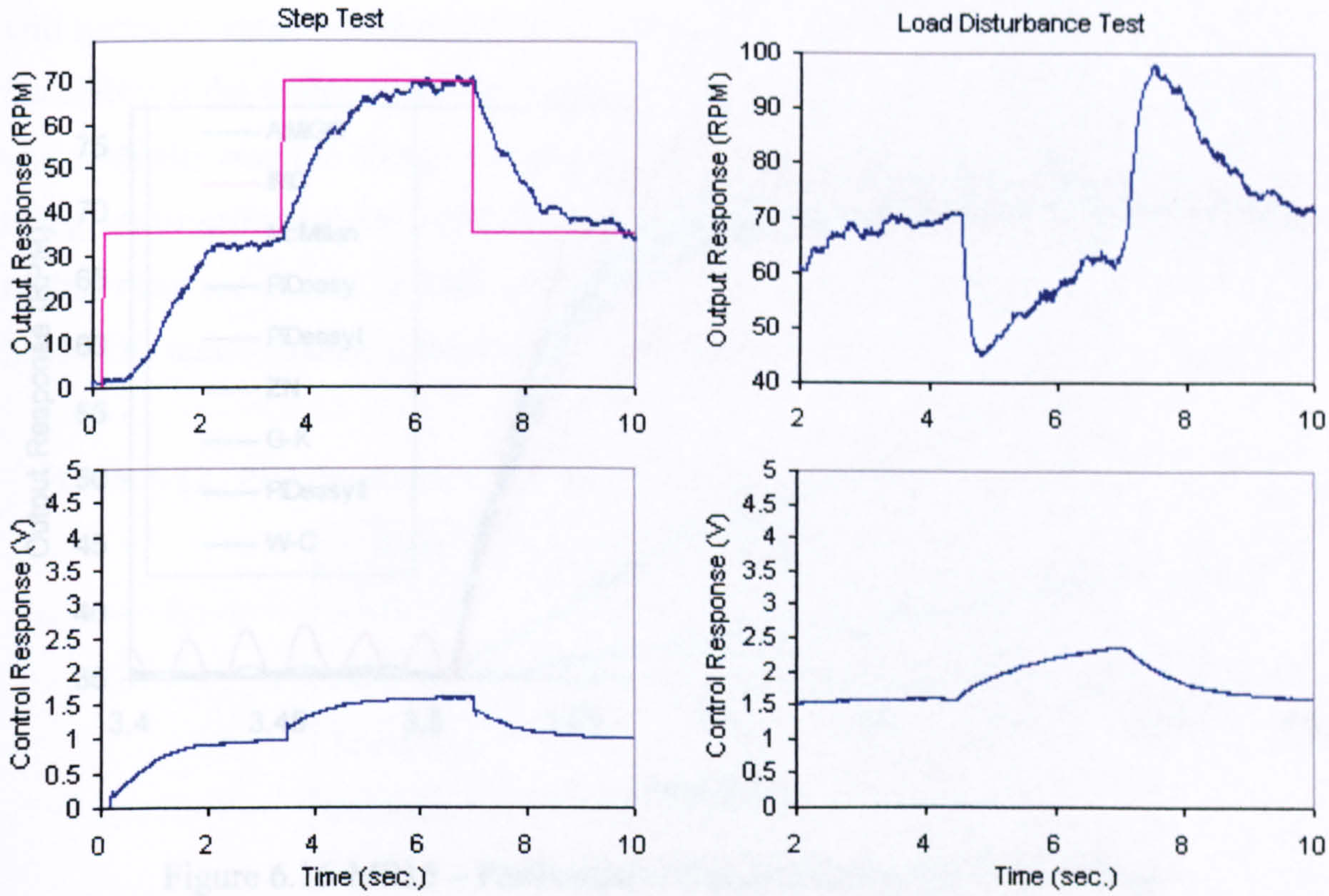

Figure 6.15 MS15 – W-C Test Results

Based on the overview of all the tuning rules performance (as shown in Figures 6.7 to 6.15), it shows that all the tuning rules have no problems in tracking the set-point changes and recovering from the load disturbances. ZN tuning is aggressive as compared with the others. G-K and W-C have the smoothest control response and hence slower response in both tests.

In order to have a better view of the individual tuning rule on each test, the results are combined together by looking at a particular time frame. For the step response test, the time frame at 3.4 sec. to 3.8 sec. is shown. For the load disturbance test, the time frame at 6.5 sec. to 8.5 sec. is shown. The results are shown in Figure 6.16 and 6.17. Figure 6.16 shows that ZN tuning is too aggressive as the output is chattering. Thus ZN is not included in Figure 6.17 due to it oscillatory response which will obscure the viewing of the other tuning rules. The ZN performance on load disturbance test is the best among the other tuning rules. However, it is not desirable due to it oscillatory response. The PIDeasy, PIDeasyI and PIDeasyII tuning rules have excellent performance in set-pointing tracking with no overshoot and reasonable performance in recovering from load disturbance.

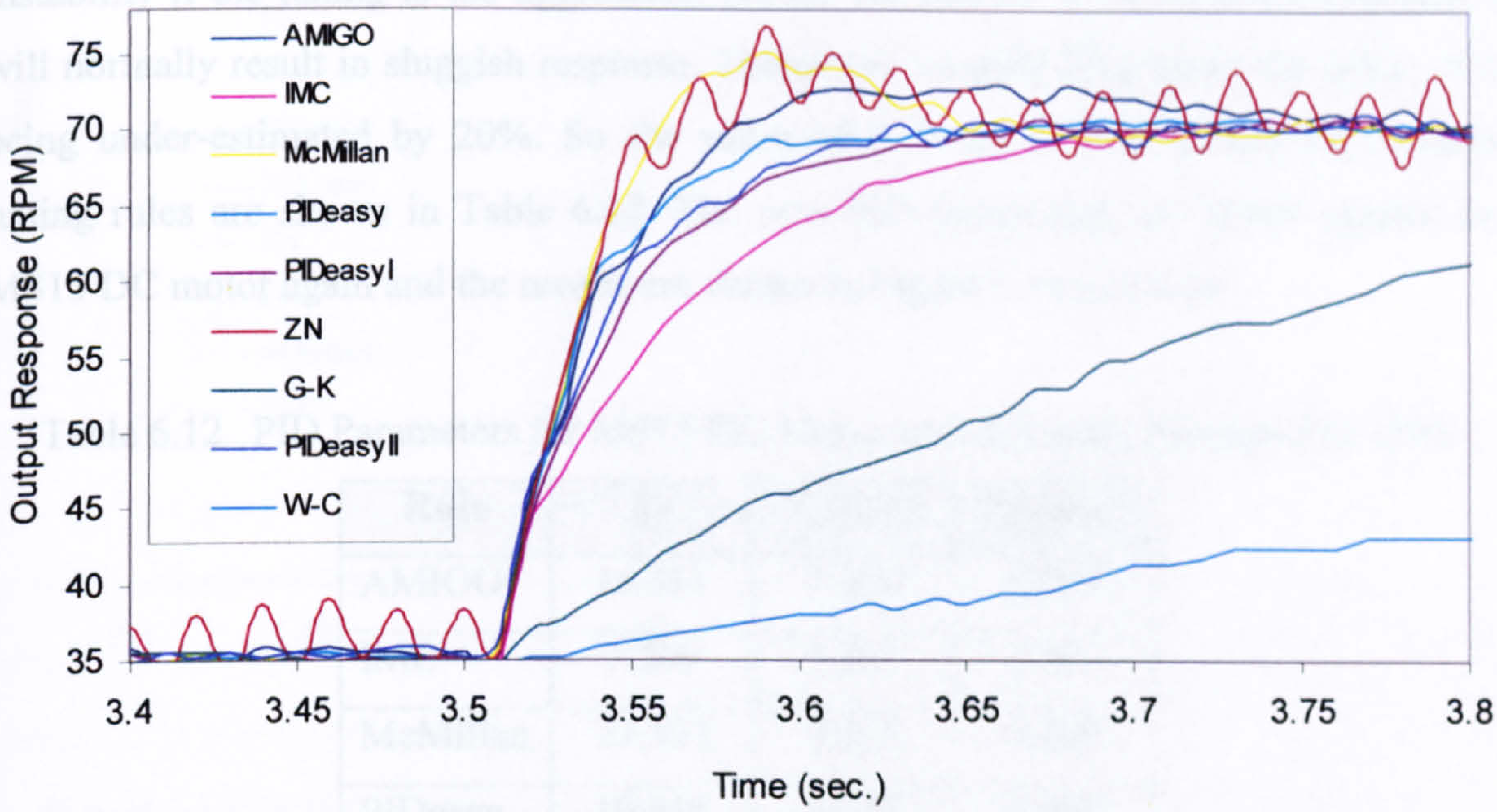

Figure 6.16 MS15 – Performance Comparison on Set-Point Change

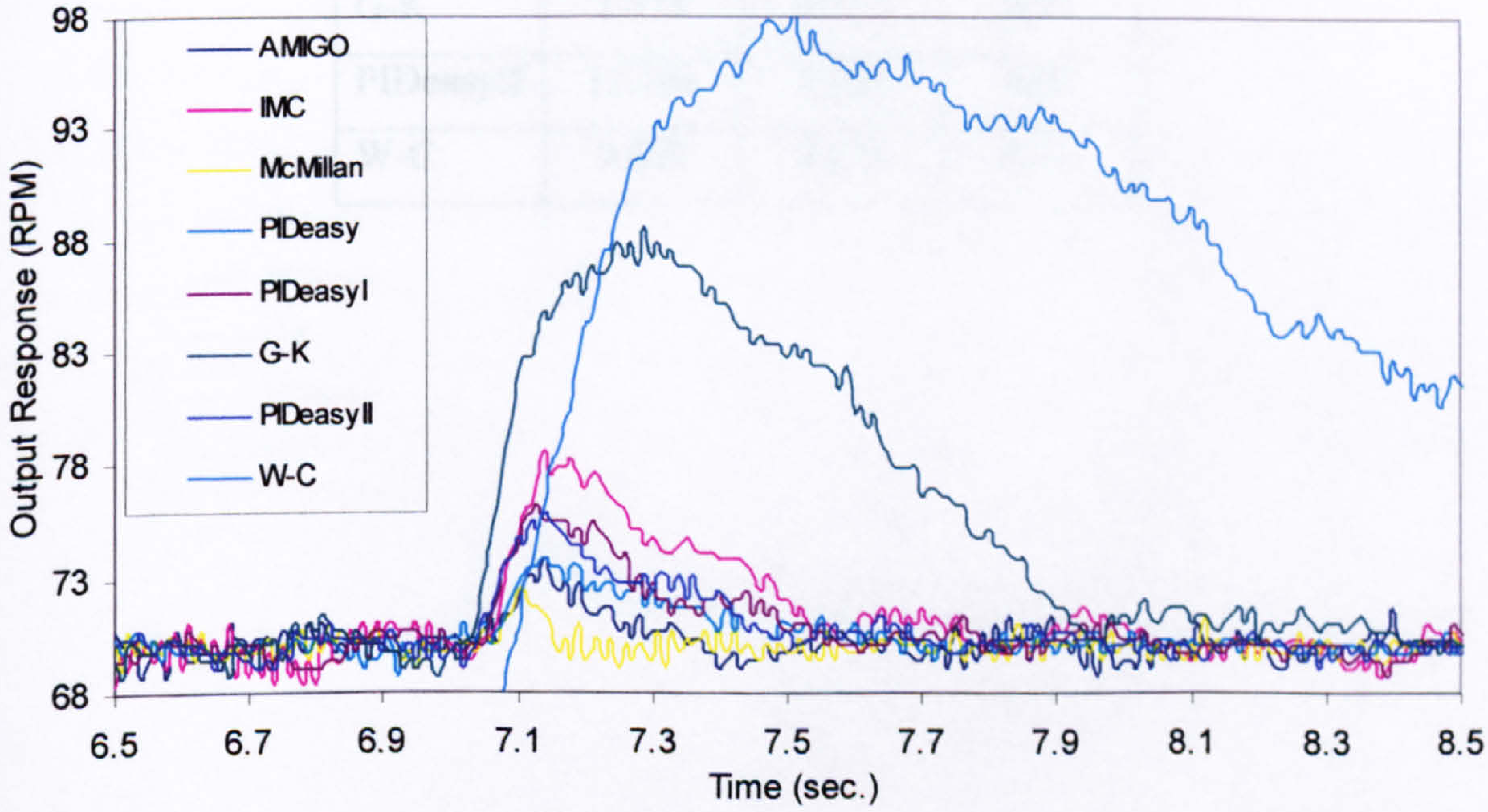

Figure 6.17 MS15 – Performance Comparison on Load Disturbance Rejection

Next, we shall proceed to investigate their robustness against modelling error. Since this is a fast response process, thus only the process gain, K, will be manipulated to investigate the robustness of the tuning rules. In the case of K being under-estimated, it will normally cause a tuning rule to have a higher value of K_P. This will in turn cause instability if the tuning is too aggressive. As for the case of K being over-estimated, it will normally result in sluggish response. Therefore, we shall investigate the effect of K being under-estimated by 20%. So the value of K becomes 0.5468 and the updated tuning rules are shown in Table 6.12. The new PID parameters are tested against the MS15 DC motor again and the results are shown in Figure 6.18 and 6.19.

Table 6.12 PID Parameters for MS15 DC Motor with K Under-Estimated by 20%

Rule	K_P	T_I (sec.)	T_D (sec.)
AMIGO	14.311	0.078	0.007
IMC	7.269	0.262	0.007
McMillan	22.333	0.029	0.007
PIDeasy	19.856	0.258	0.005
PIDeasyI	12.237	0.258	0.005
ZN	37.187	0.03	0.008
G-K	1.714	0.223	0.01
PIDeasyII	12.184	0.224	0.01
W-C	0.402	0.223	0.01

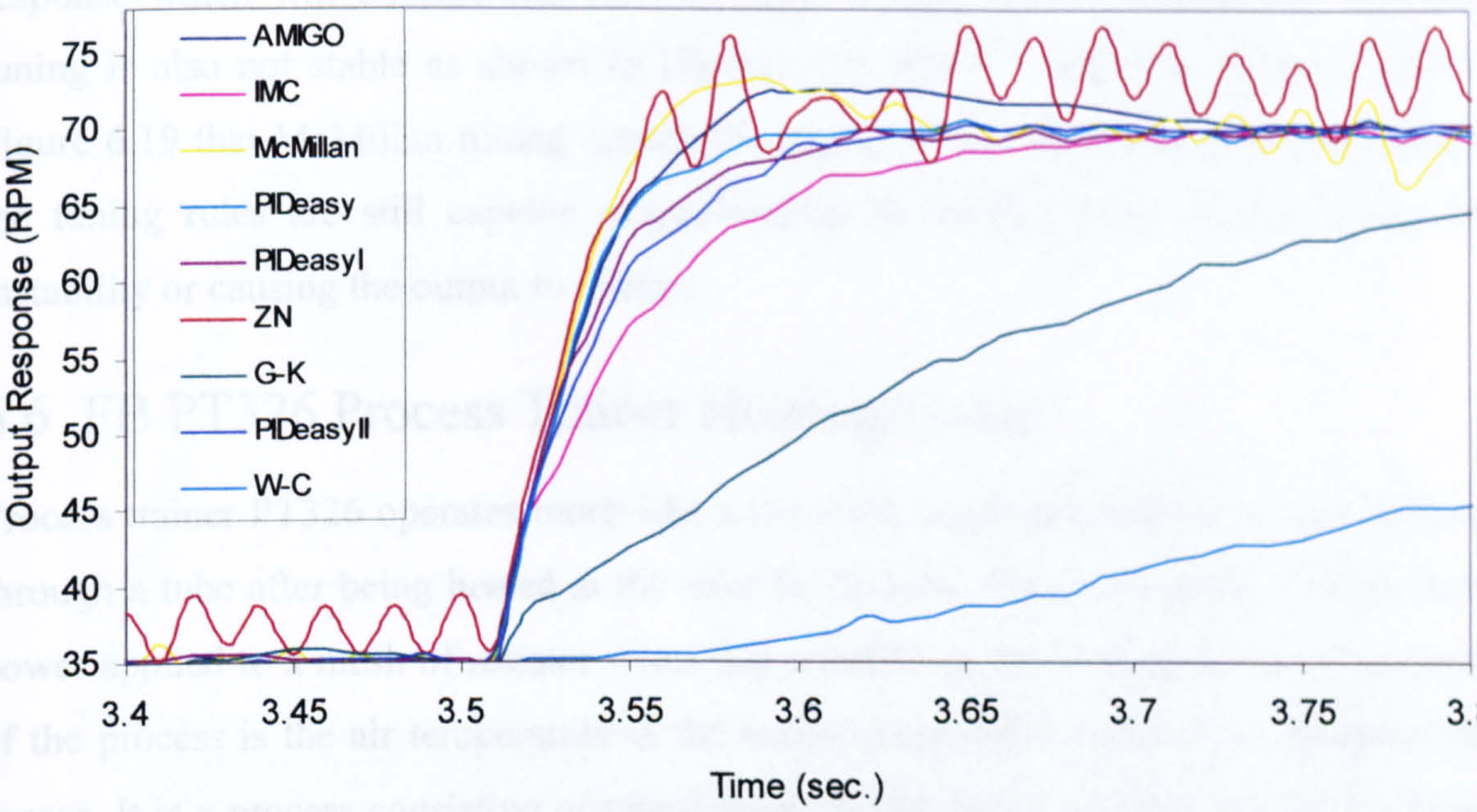

Figure 6.18 MS15 – Performance Comparison on Set-Point Change with *K* Under-Estimated by 20%

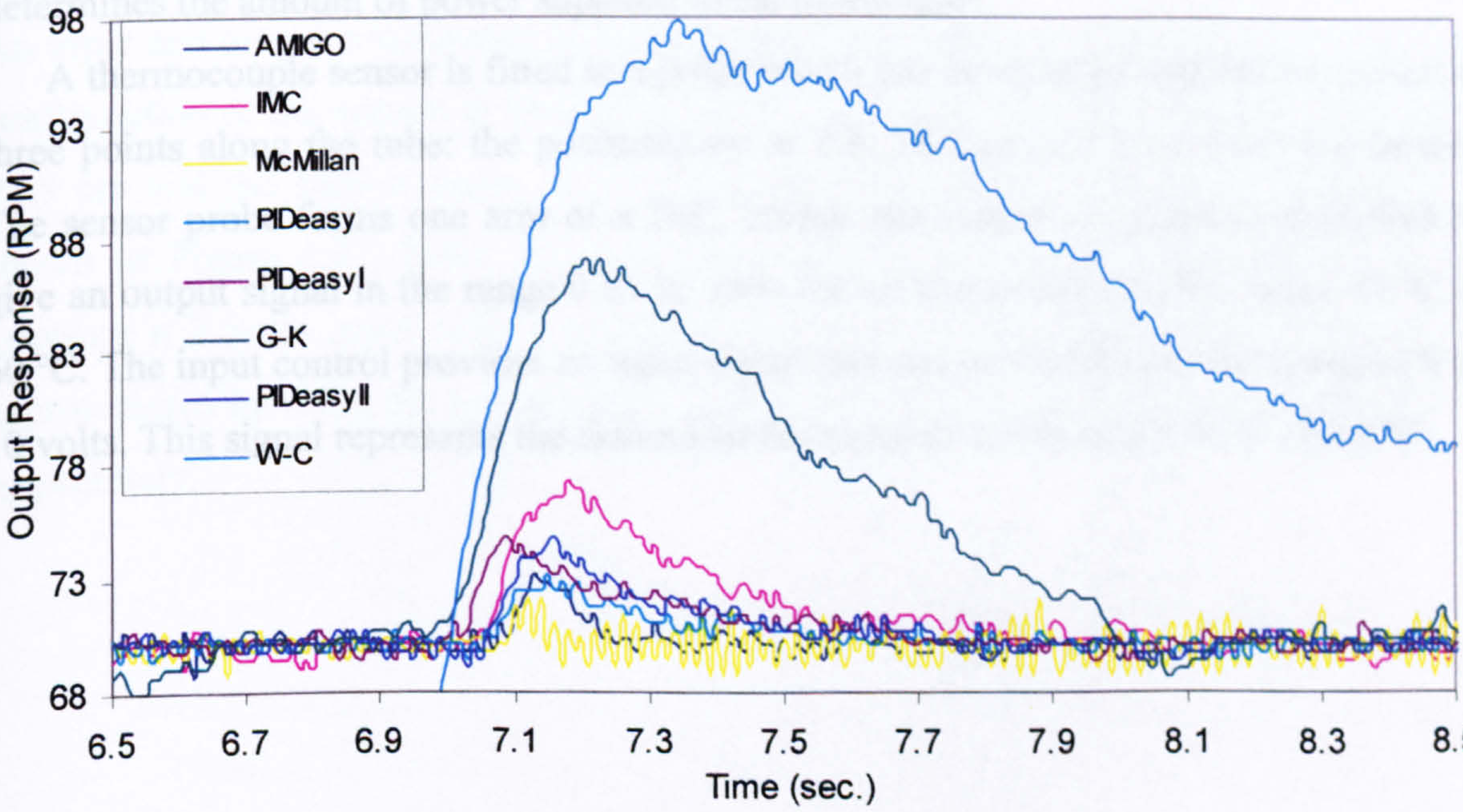

Figure 6.19 MS15 – Performance Comparison on Load Disturbance Rejection with *K* Under-Estimated by 20%

Here, it is expected ZN tuning to become even worst and this is demonstrated in Figure 6.18. Again, ZN result is not included in Figure 6.19 due to its oscillatory response which will obscure the view of other tuning rules. It seems that McMillan tuning is also not stable as shown in Figure 6.18 after 3.7 sec. It is further shown in Figure 6.19 that McMillan tuning causes the output to chatter. Other than that, most of the tuning rules are still capable of performing reasonably well without going into instability or causing the output to chatter.

6.6 FB PT326 Process Trainer Heating System

Process trainer PT326 operates much like a common hand-held hair dryer. Air is blown through a tube after being heated at the inlet to the tube. The input to the process is the power applied to a mesh of resistor wires that constitutes the heating device. The output of the process is the air temperature at the outlet, measured in volts by a thermocouple sensor. It is a process consisting of transferring energy to the air flowing past the heater so that the air in the tube is brought to a specified temperature. The purpose of the control equipment is to measure the air temperature at one of the three points in the tube, compare it with the value set by the operator and then generate a control signal which determines the amount of power supplied to the heater grid.

A thermocouple sensor is fitted in a probe which can be inserted into the air stream at three points along the tube: the positions are at 2.8, 14.0 and 27.9 cm from the heater. The sensor probe forms one arm of a D.C. bridge, the output of which is amplified to give an output signal in the range 0 to 10 volts for air temperature in the range 30 °C to 60 °C. The input control provides an input signal that can be varied over the range of 0 to 10 volts. This signal represents the desired air temperature in the range 30 °C to 60 °C.

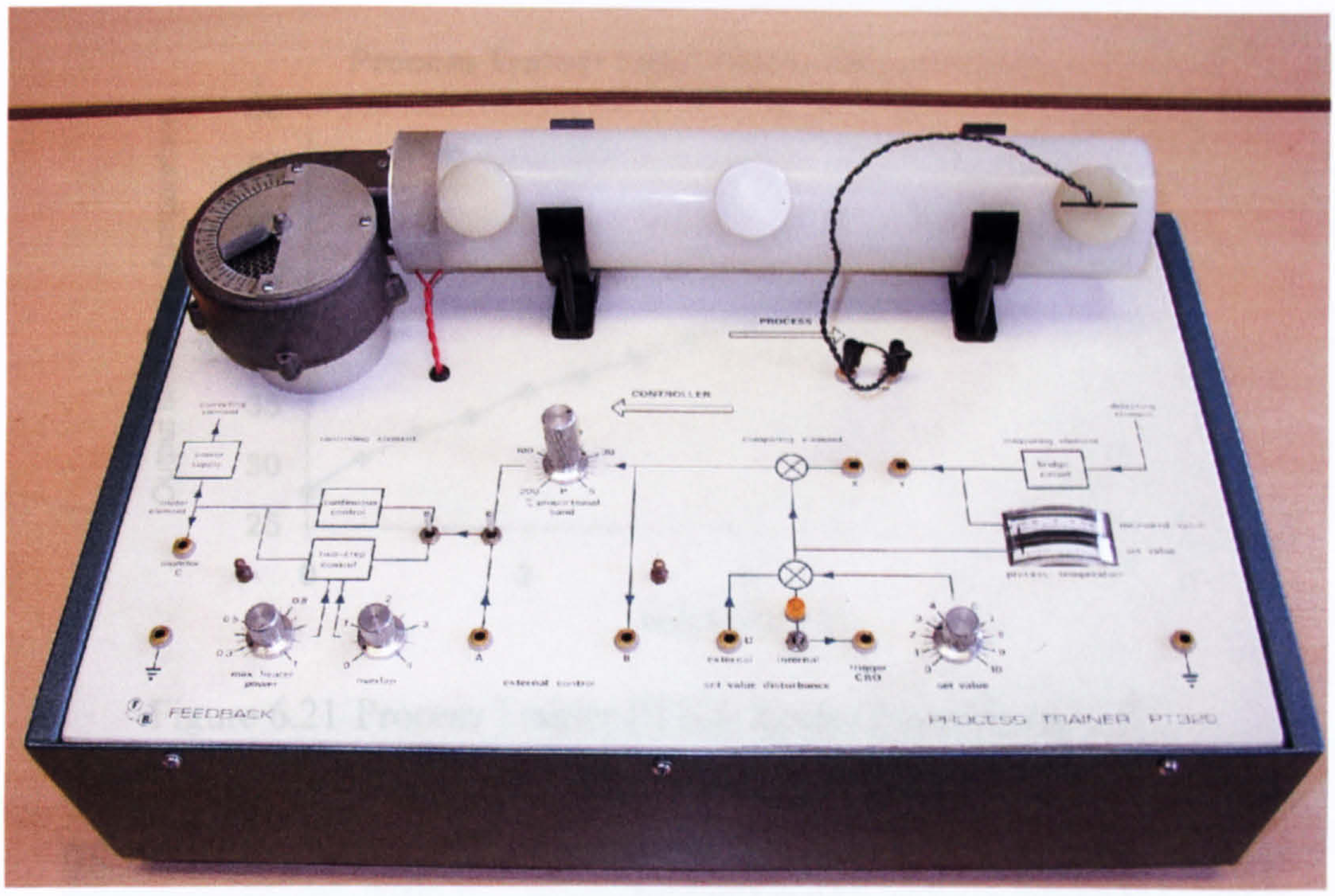

Figure 6.20 Process Trainer PT326 Heating System

6.6.1 Modelling and Tuning Process

An input and output relationship is being established in order to verify its linearity. The configuration of the system is fixed throughout the test with the sensor probe at position 27.9 cm and the blower angle at 50°. The process trainer PT326 linear behaviour is shown in Figure 6.21. Next, an open-loop step test is conducted on the system by injecting a 2 volts input with a sampling rate of 0.01 sec. The response captured is approximated by using the two models (5.1) and (5.2), as shown in Figure 6.22 and 6.23 respectively.

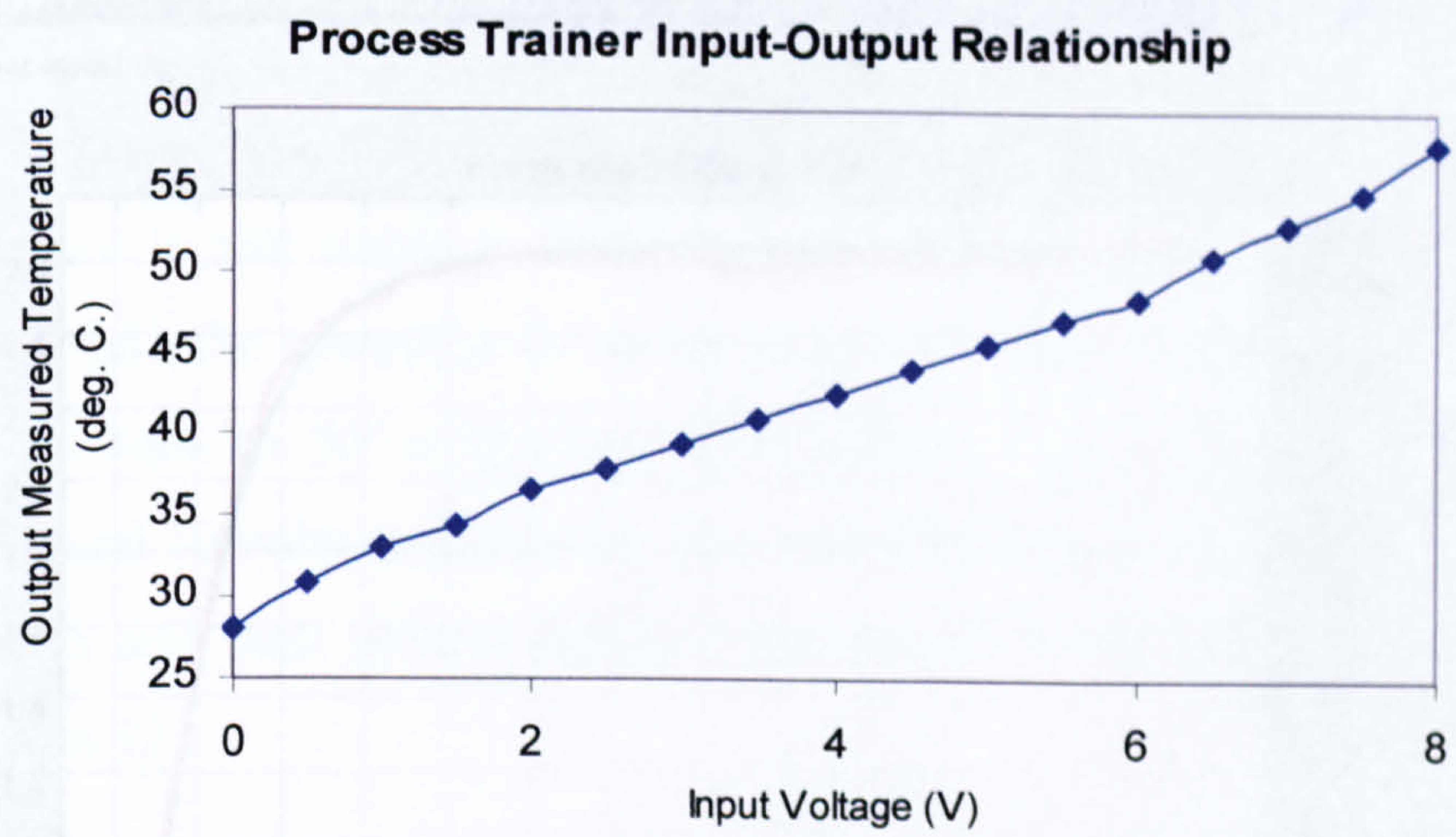

Figure 6.21 Process Trainer PT326 Input-Output Relationship

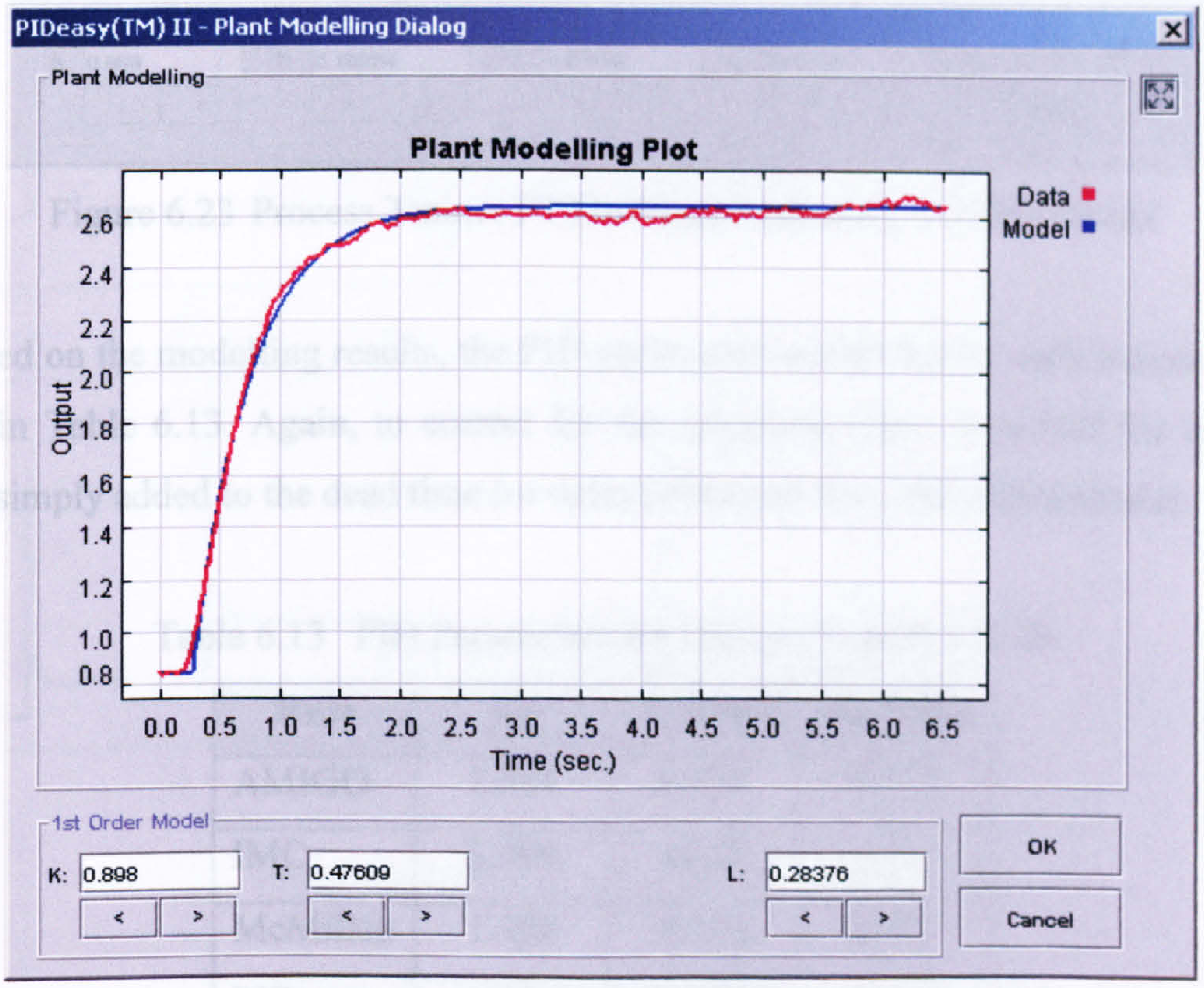

Figure 6.22 Process Trainer PT326 Modelling using FOLPD Model

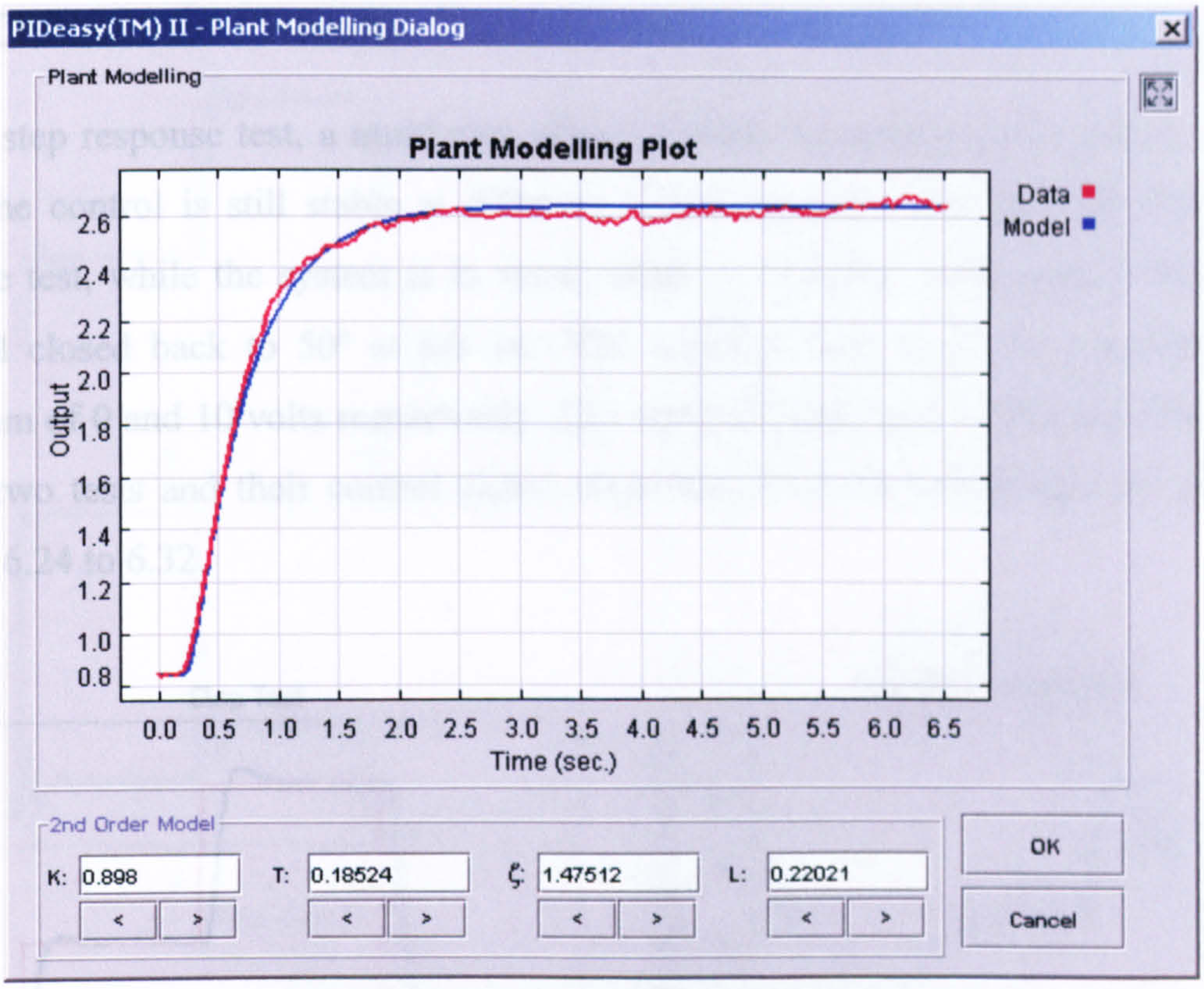

Figure 6.23 Process Trainer PT326 Modelling using SOSPD Model

Based on the modelling results, the PID parameters computed by each tuning rule are shown in Table 6.13. Again, to correct for the sampling effect, one half the sampling time is simply added to the dead time (or delay) obtained from the step response.

Table 6.13 PID Parameters for Process Trainer PT326

Rule	K_P	T_I **(sec.)**	T_D **(sec.)**
AMIGO	1.049	0.426	0.122
IMC	1.799	0.620	0.111
McMillan	1.498	0.501	0.125
PIDeasy	1.271	0.555	0.073
PIDeasyI	1.222	0.555	0.073
ZN	2.203	0.578	0.144
G-K	0.789	0.547	0.063
PIDeasyII	1.218	0.548	0.063
W-C	0.497	0.547	0.063

6.6.2 Discussion of Results

For the step response test, a multi-step signal is being injected into the system so as to verify the control is still stable at different operating range. For the load disturbance response test, while the system is in steady-state, the blower angle opened fully at 4.0 sec. and closed back to 50° at 8.0 sec. The actuator limit is set to a minimum and maximum of 0 and 10 volts respectively. The sampling rate used is 0.01 sec. The results on the two tests and their control signal responses for each tuning rule are shown in Figures 6.24 to 6.32.

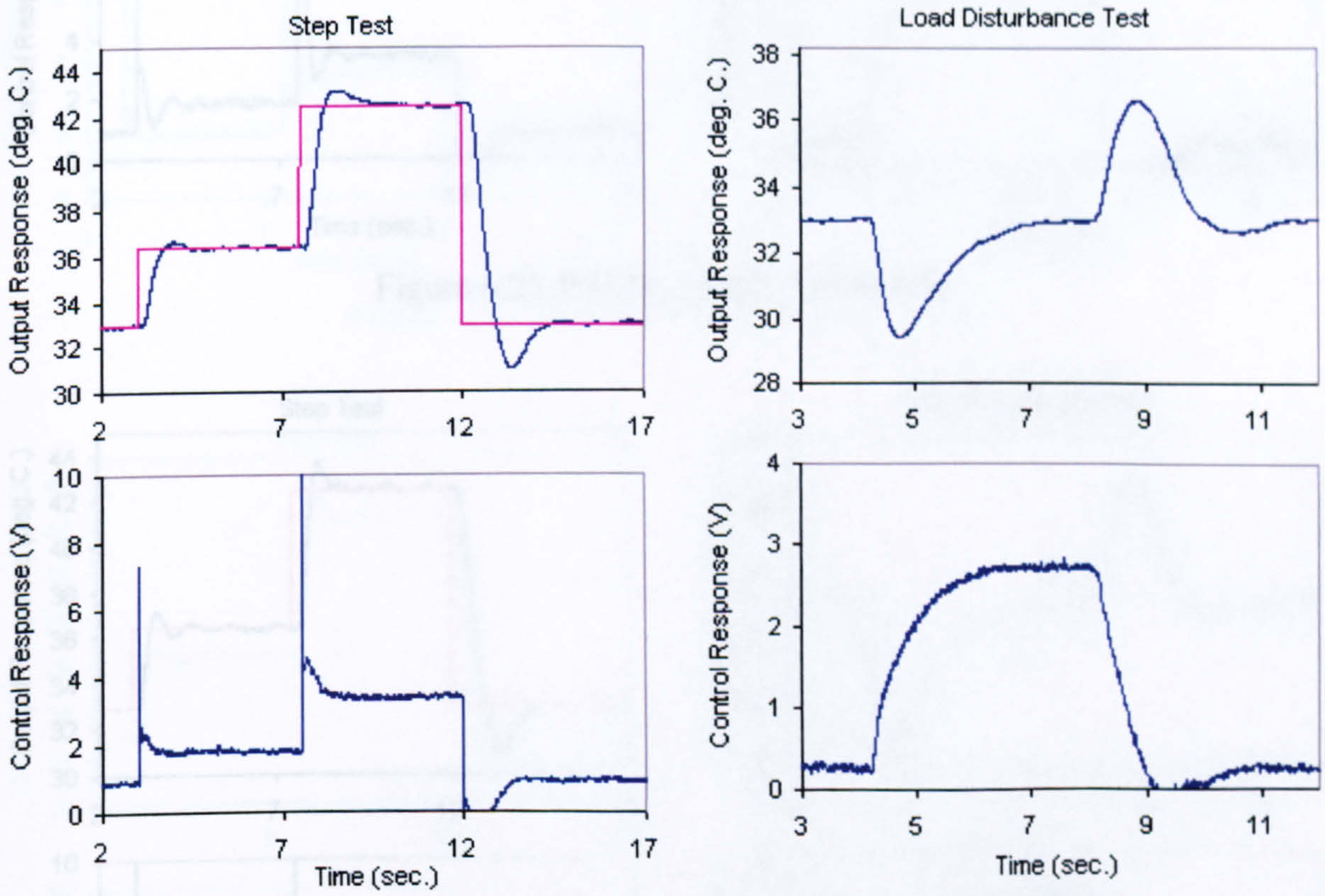

Figure 6.24 PT326 – AMIGO Test Results

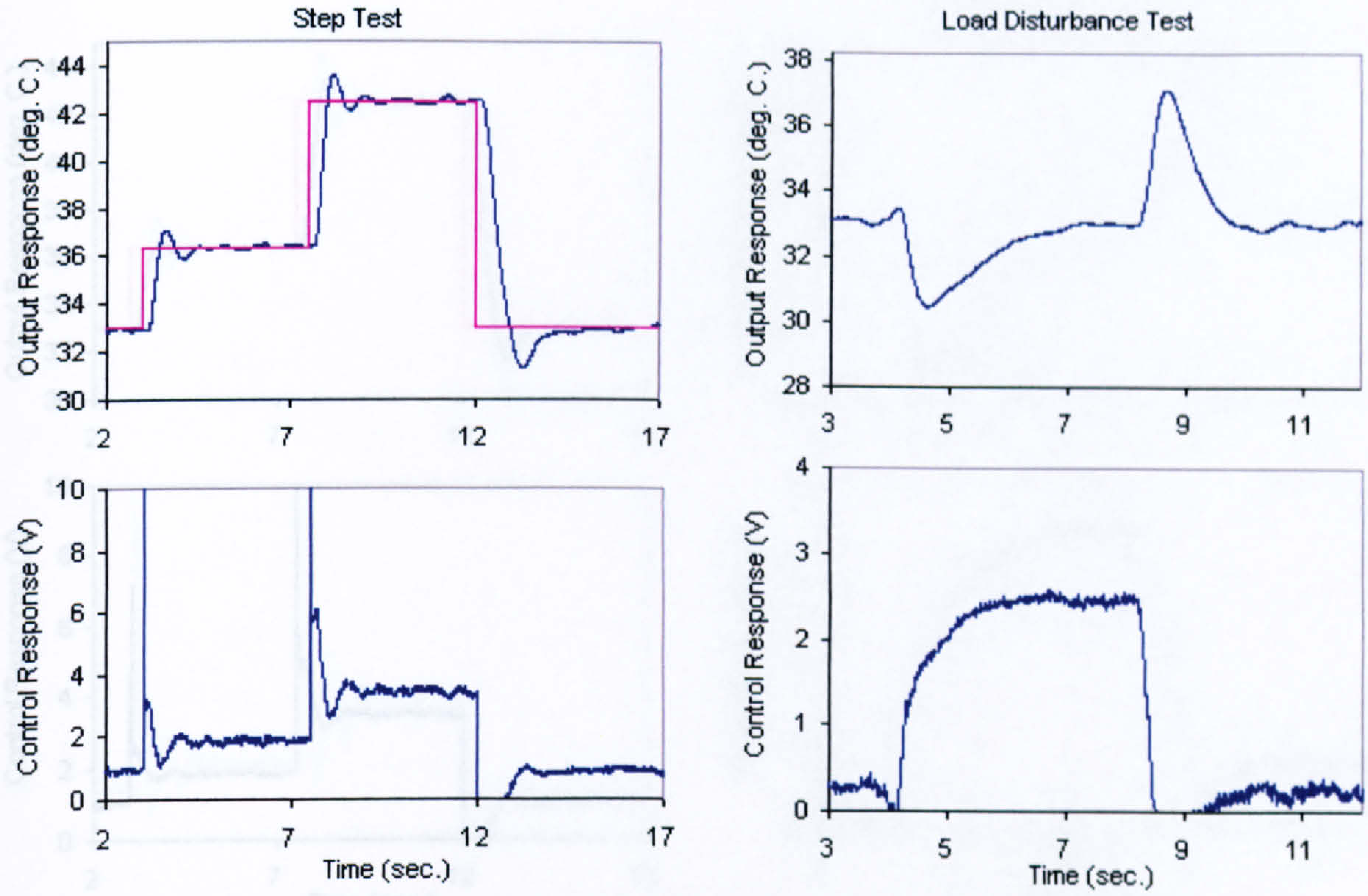

Figure 6.25 PT326 – IMC Test Results

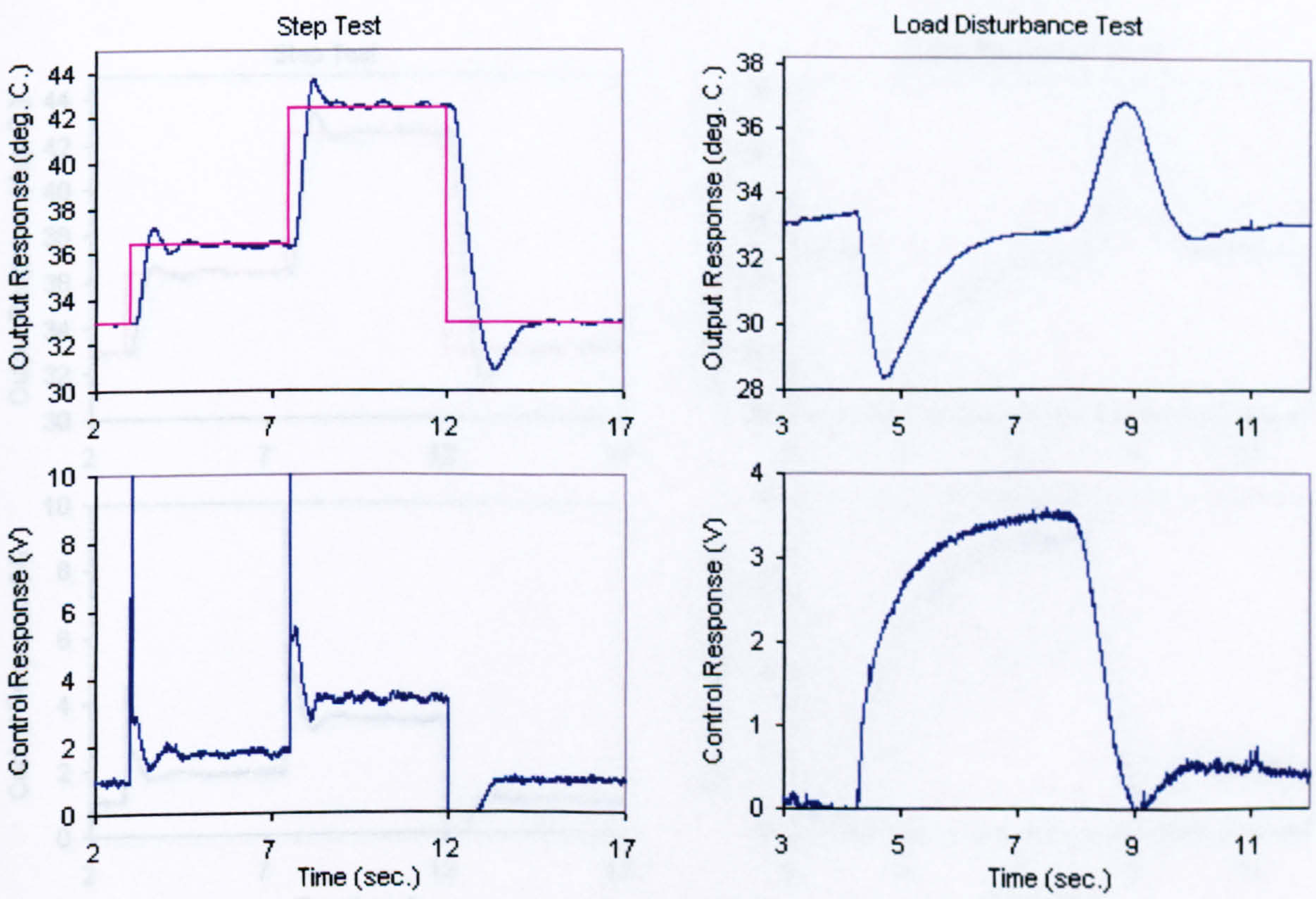

Figure 6.26 PT326 – McMillan Test Results

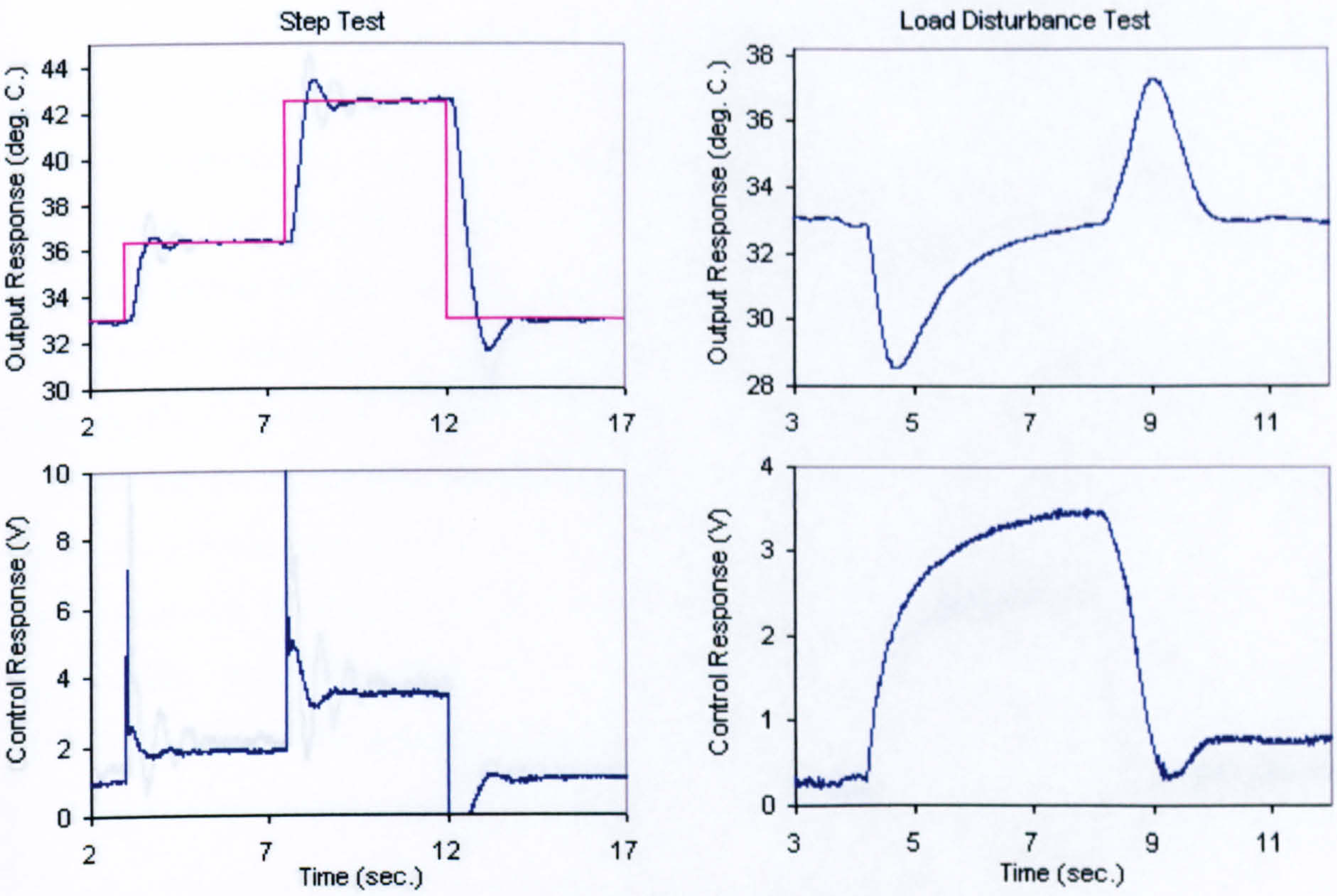

Figure 6.27 PT326 – PIDeasy Test Results

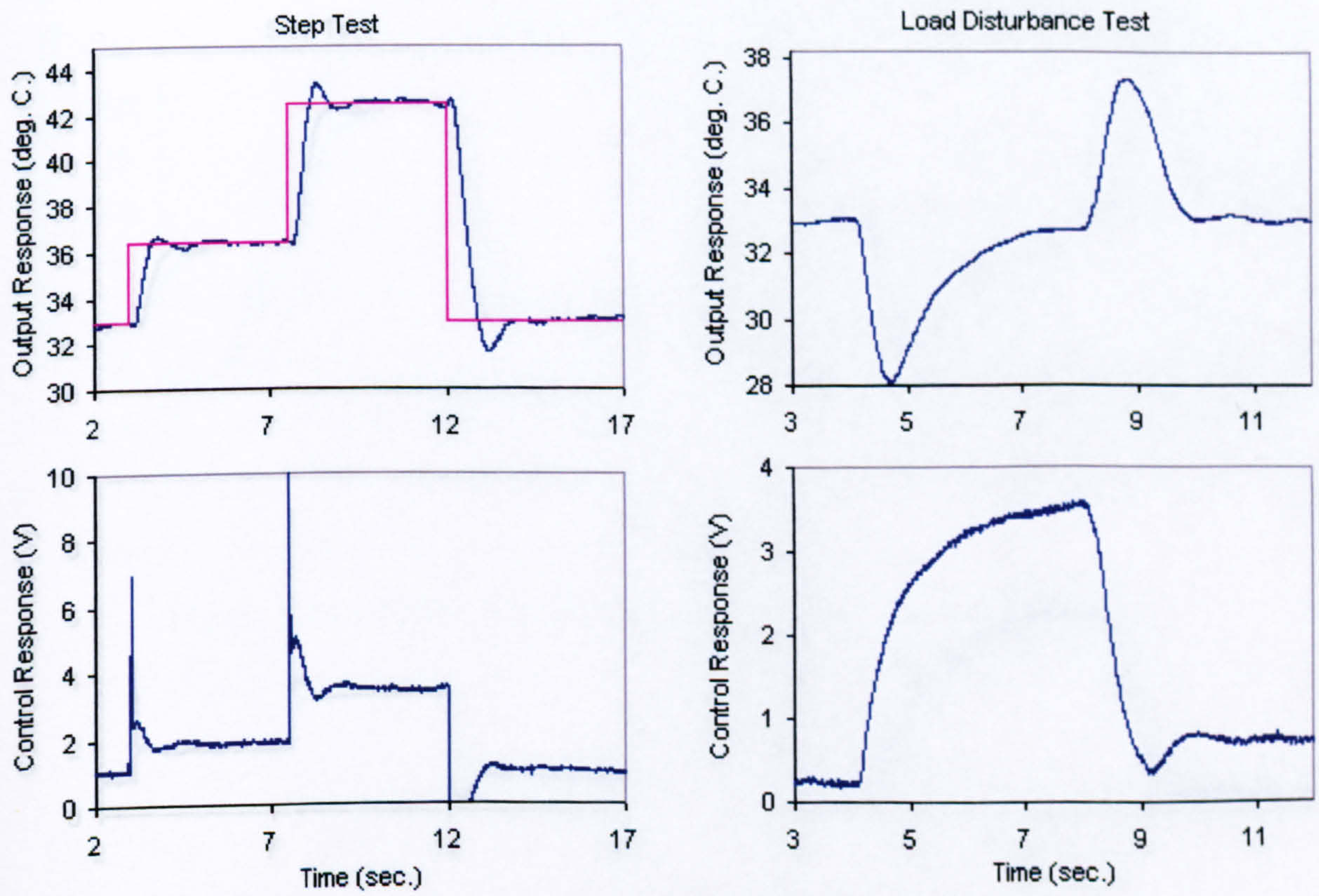

Figure 6.28 PT326 – PIDeasyI Test Results

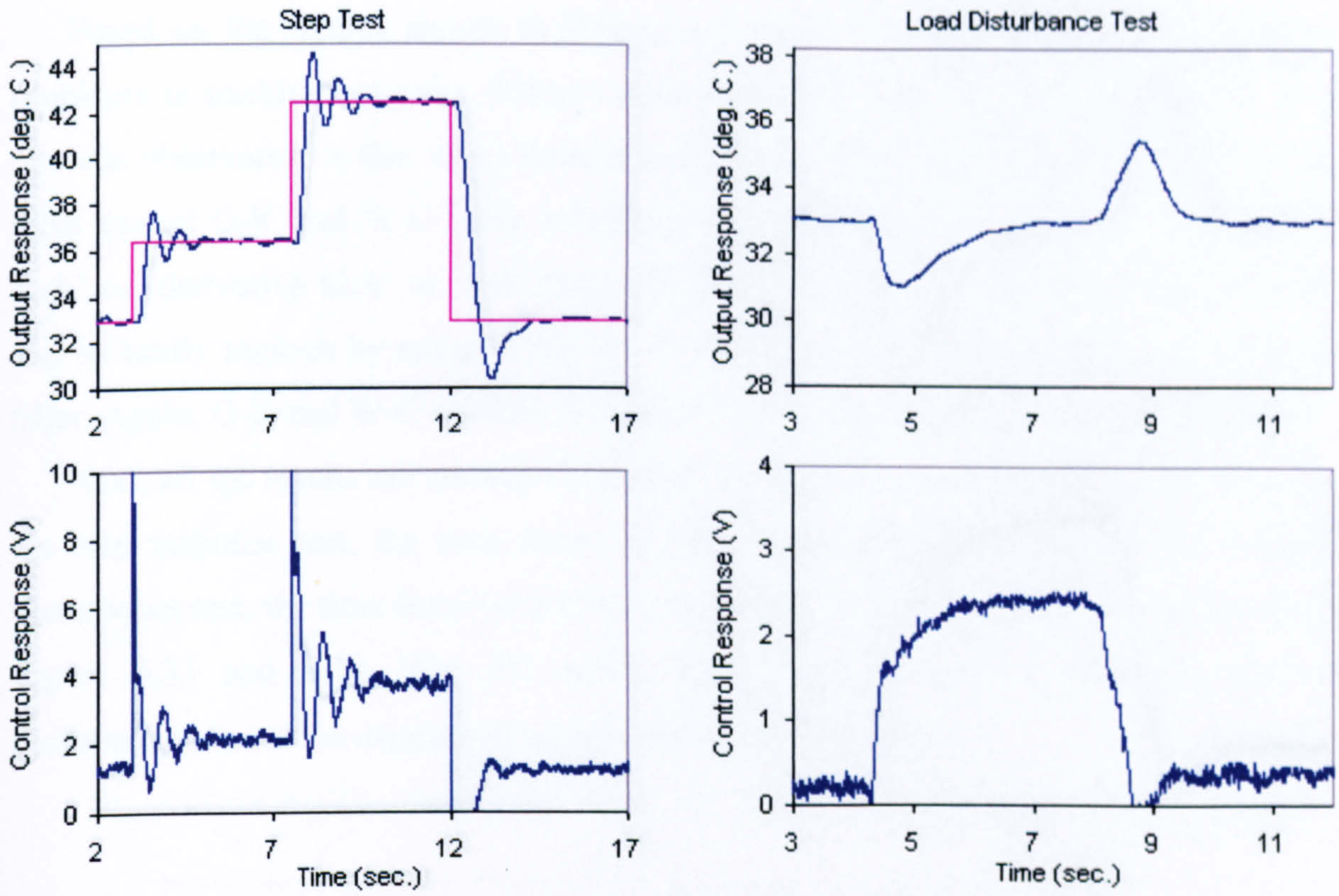

Figure 6.29 PT326 – ZN Test Results

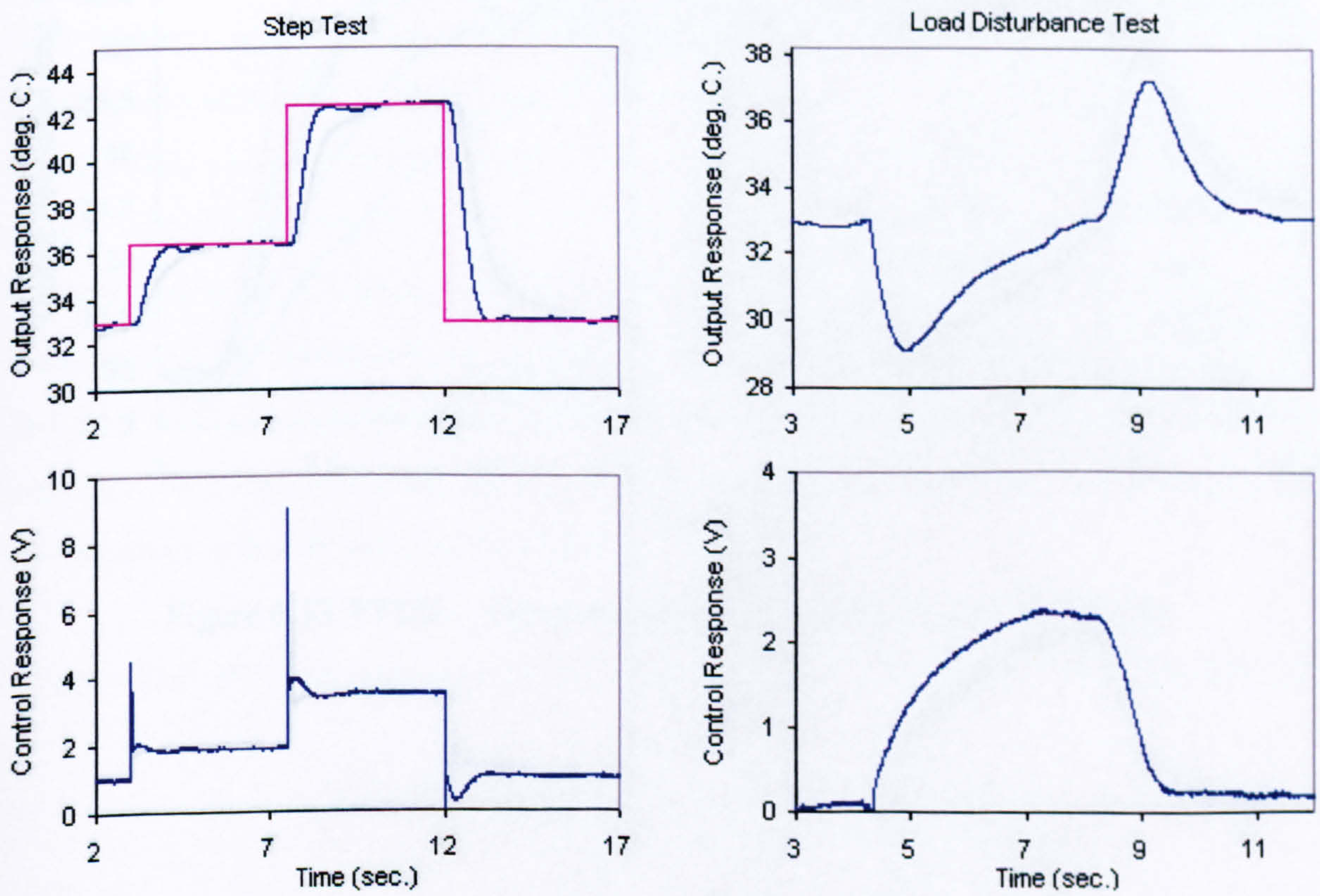

Figure 6.30 PT326 – G-K Test Results

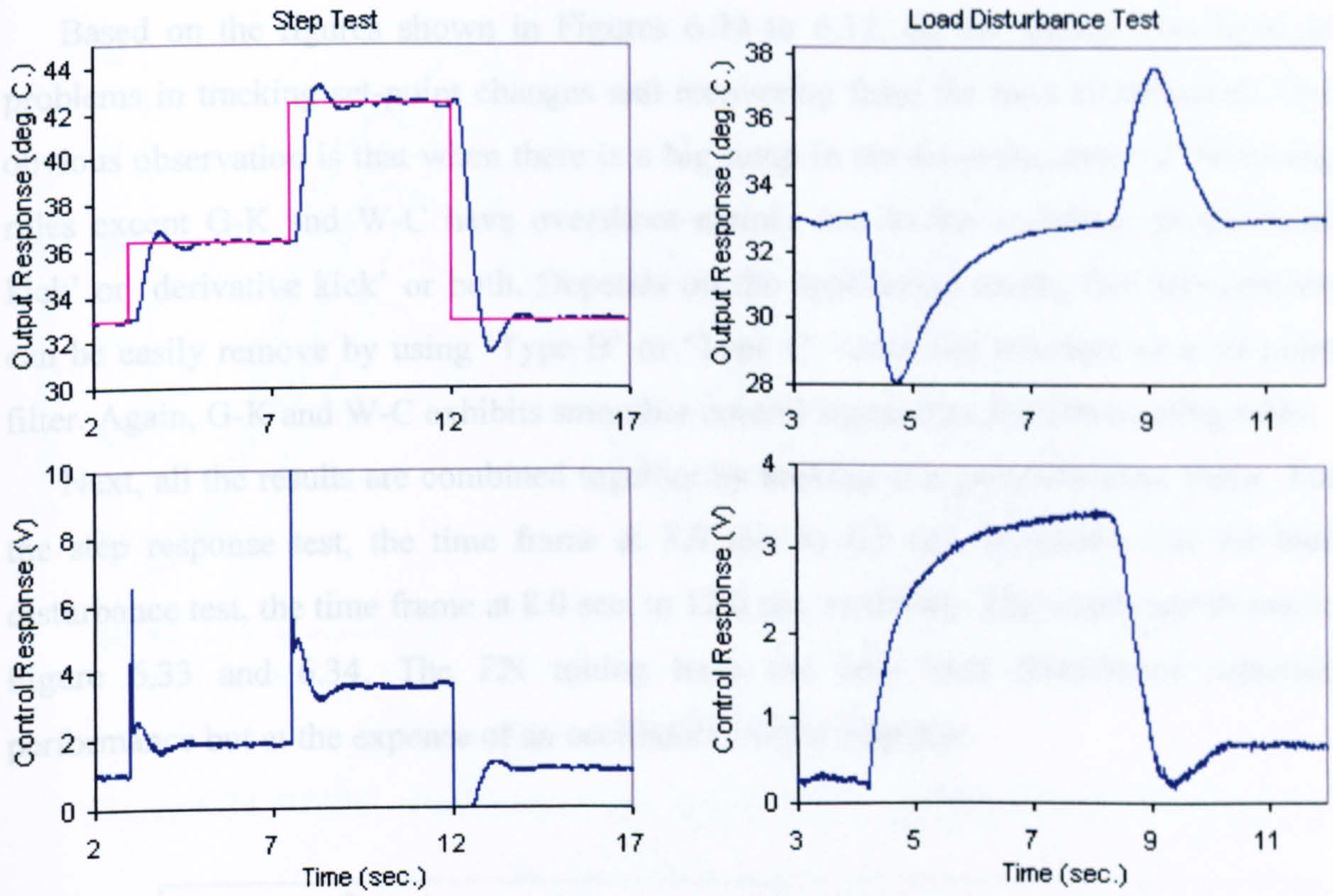

Figure 6.31 PT326 – PIDeasyII Test Results

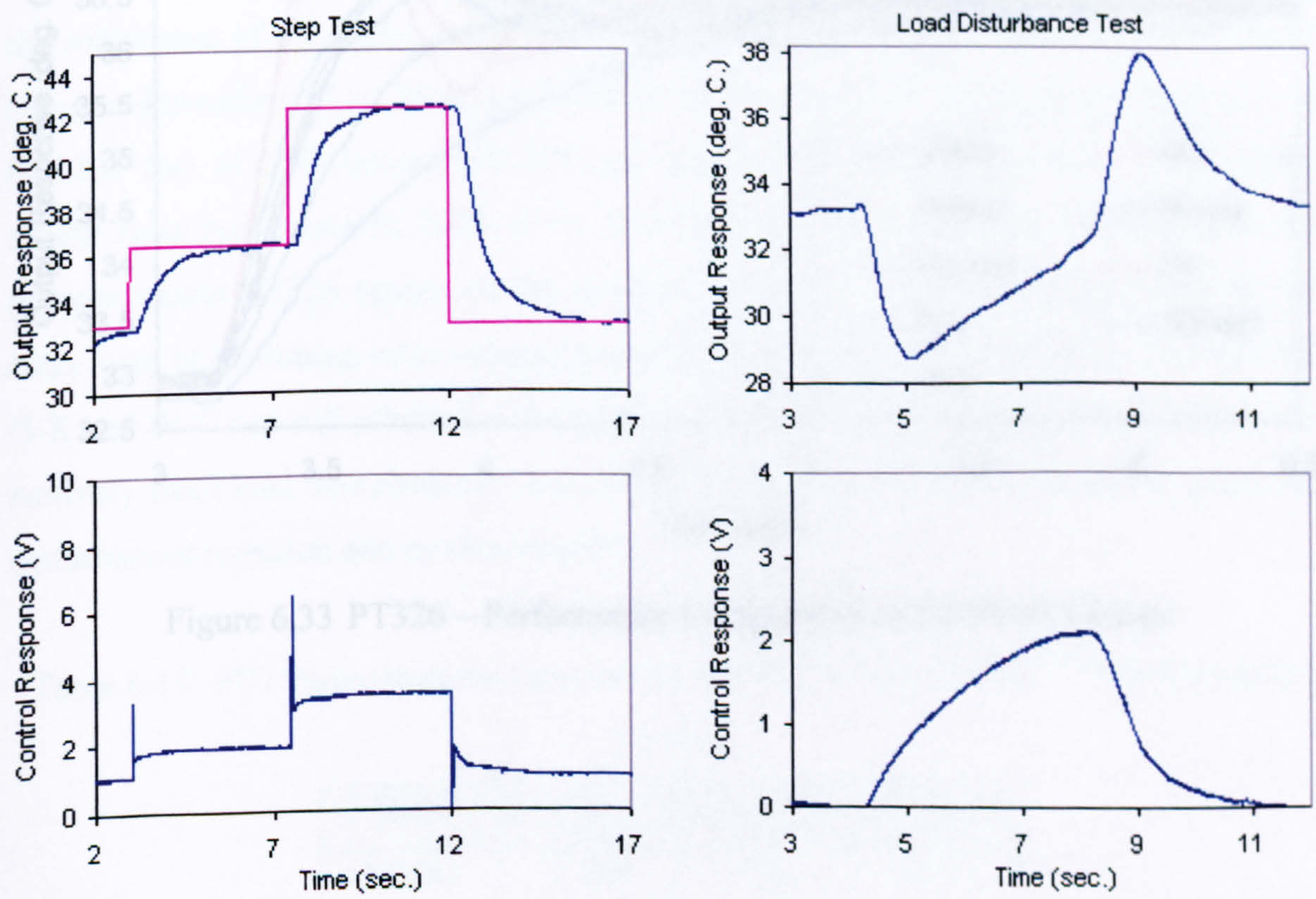

Figure 6.32 PT326 – W-C Test Results

Based on the figures shown in Figures 6.24 to 6.32, all the tuning rules have no problems in tracking set-point changes and recovering from the load disturbances. One obvious observation is that when there is a big jump in the set-point, most of the tuning rules except G-K and W-C have overshoot mainly due to the so-called 'proportional kick' or 'derivative kick' or both. Depends on the application needs, this phenomenon can be easily remove by using 'Type B' or 'Type C' controller structure or a set-point filter. Again, G-K and W-C exhibits smoother control signal than the other tuning rules.

Next, all the results are combined together by looking at a particular time frame. For the step response test, the time frame at 3.0 sec. to 6.5 sec. is shown. For the load disturbance test, the time frame at 8.0 sec. to 12.0 sec. is shown. The results are shown in Figure 6.33 and 6.34. The ZN tuning have the best load disturbance rejection performance but at the expense of an oscillatory output response.

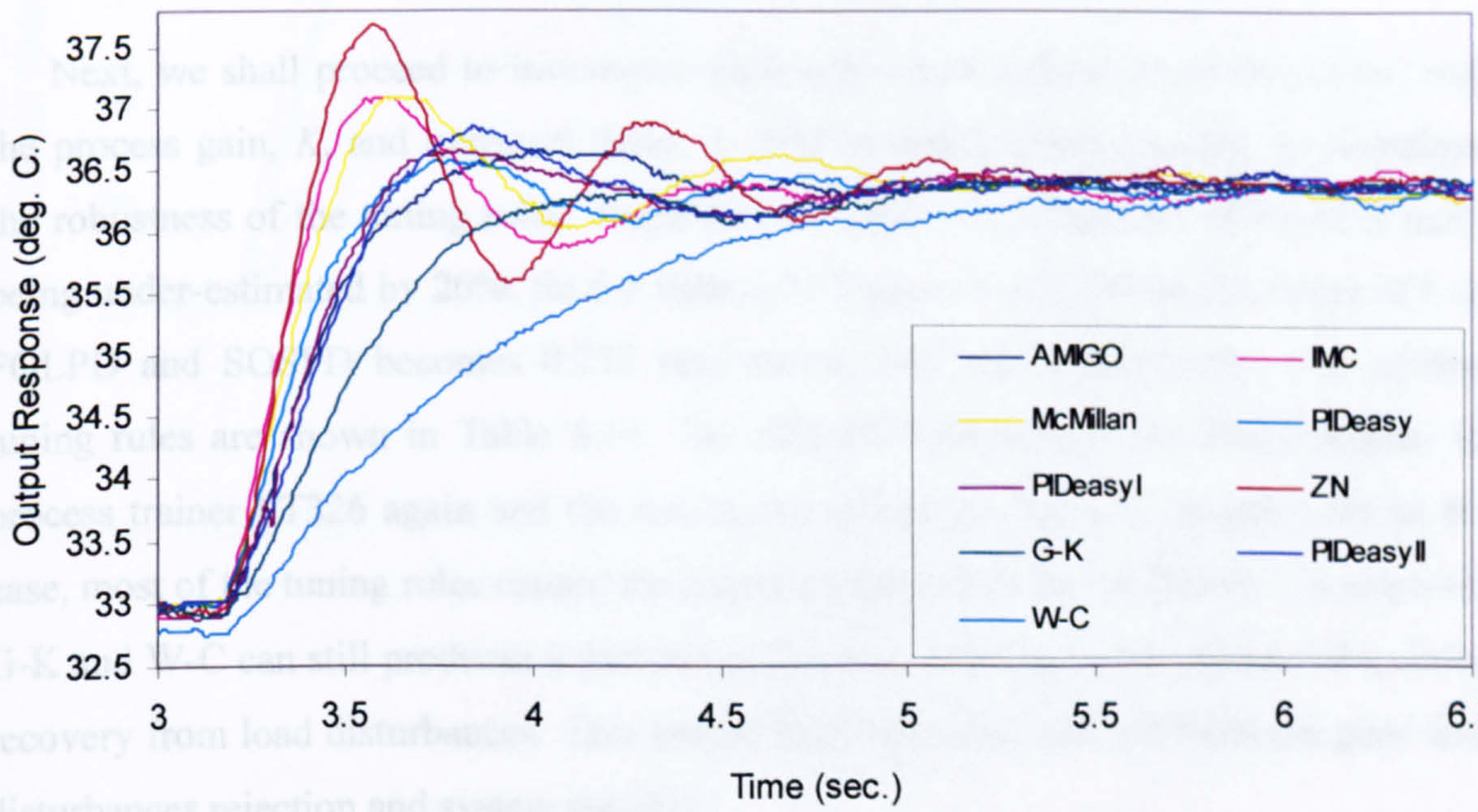

Figure 6.33 PT326 – Performance Comparison on Set-Point Change

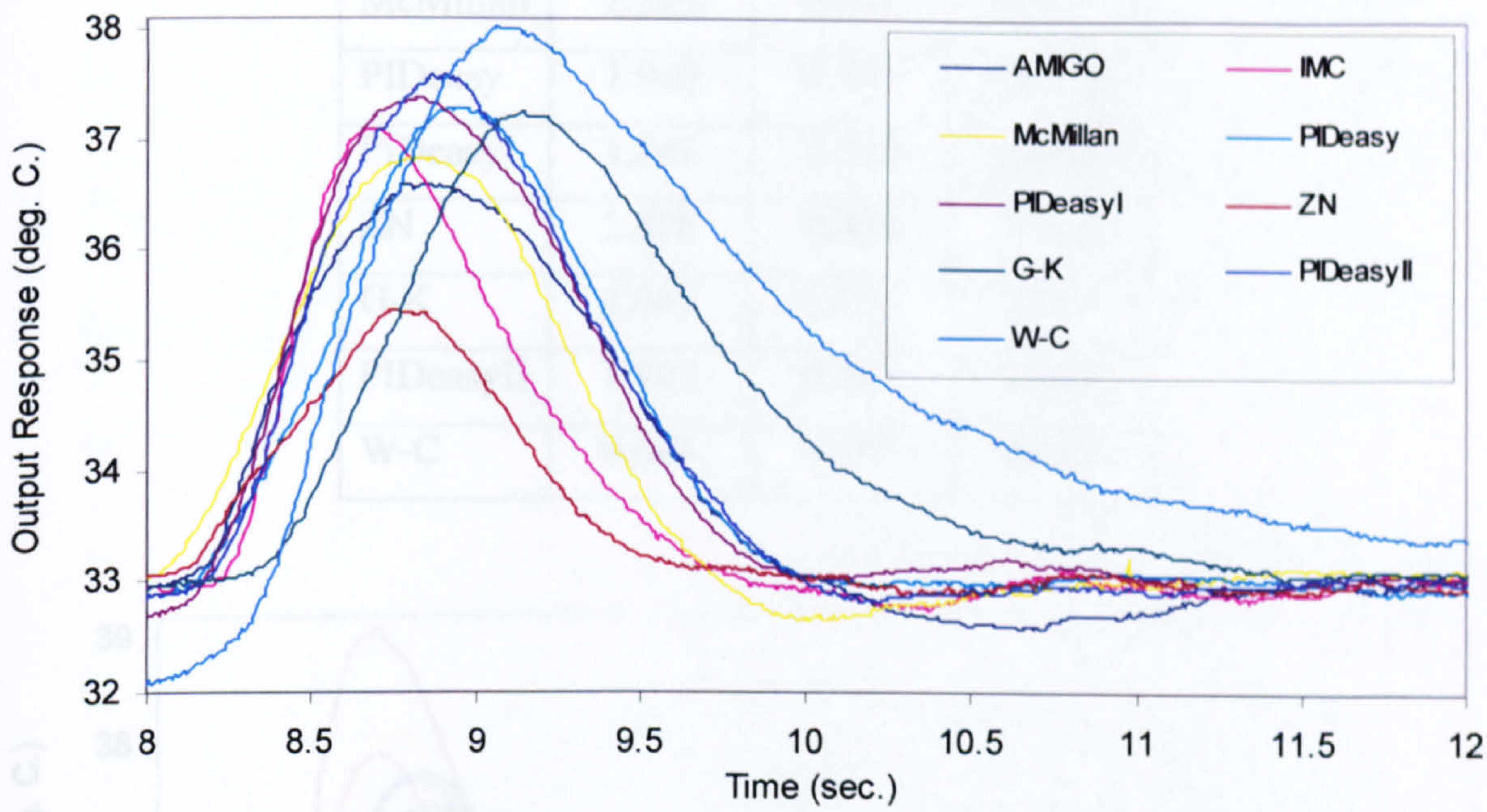

Figure 6.34 PT326 – Performance Comparison on Load Disturbance Rejection

Next, we shall proceed to investigate their robustness against modelling error. Both the process gain, K, and transport delay, L, will be manipulated together to investigate the robustness of the tuning rules. Therefore, we shall investigate the effect of K and L being under-estimated by 20%. So the value of K becomes 0.7184 and the value of L for FOLPD and SOSPD becomes 0.227 sec. and 0.1762 sec. respectively. The updated tuning rules are shown in Table 6.14. The new PID parameters are tested against the process trainer PT326 again and the results are shown in Figure 6.35 and 6.36. In this case, most of the tuning rules caused the output to become more oscillatory. As expected, G-K and W-C can still produces a damped output response but at the expense of a slower recovery from load disturbances. This simply highlights the trade-off between good load disturbances rejection and system stability.

Table 6.14 PID Parameters for Process Trainer PT326 with K and L Under-Estimated by 20%

Rule	K_P	T_I (sec.)	T_D (sec.)
AMIGO	1.564	0.393	0.101
IMC	2.519	0.592	0.093

McMillan	2.280	0.411	0.103
PIDeasy	1.940	0.540	0.060
PIDeasyI	1.841	0.540	0.060
ZN	3.428	0.464	0.116
G-K	1.045	0.547	0.063
PIDeasyII	1.905	0.547	0.063
W-C	0.644	0.547	0.063

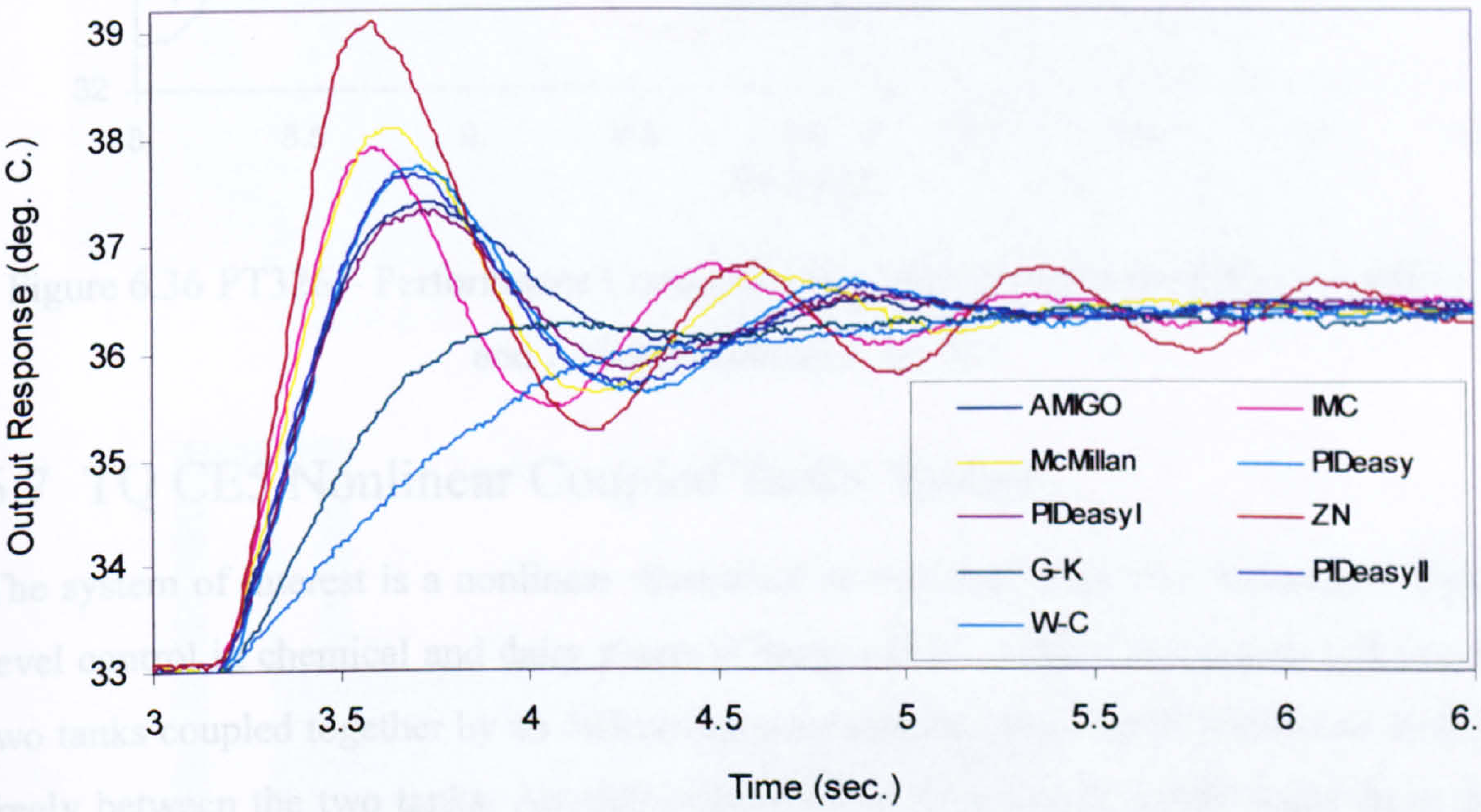

Figure 6.35 PT326 – Performance Comparison on Set-Point Change with *K* and *L* Under-Estimated by 20%

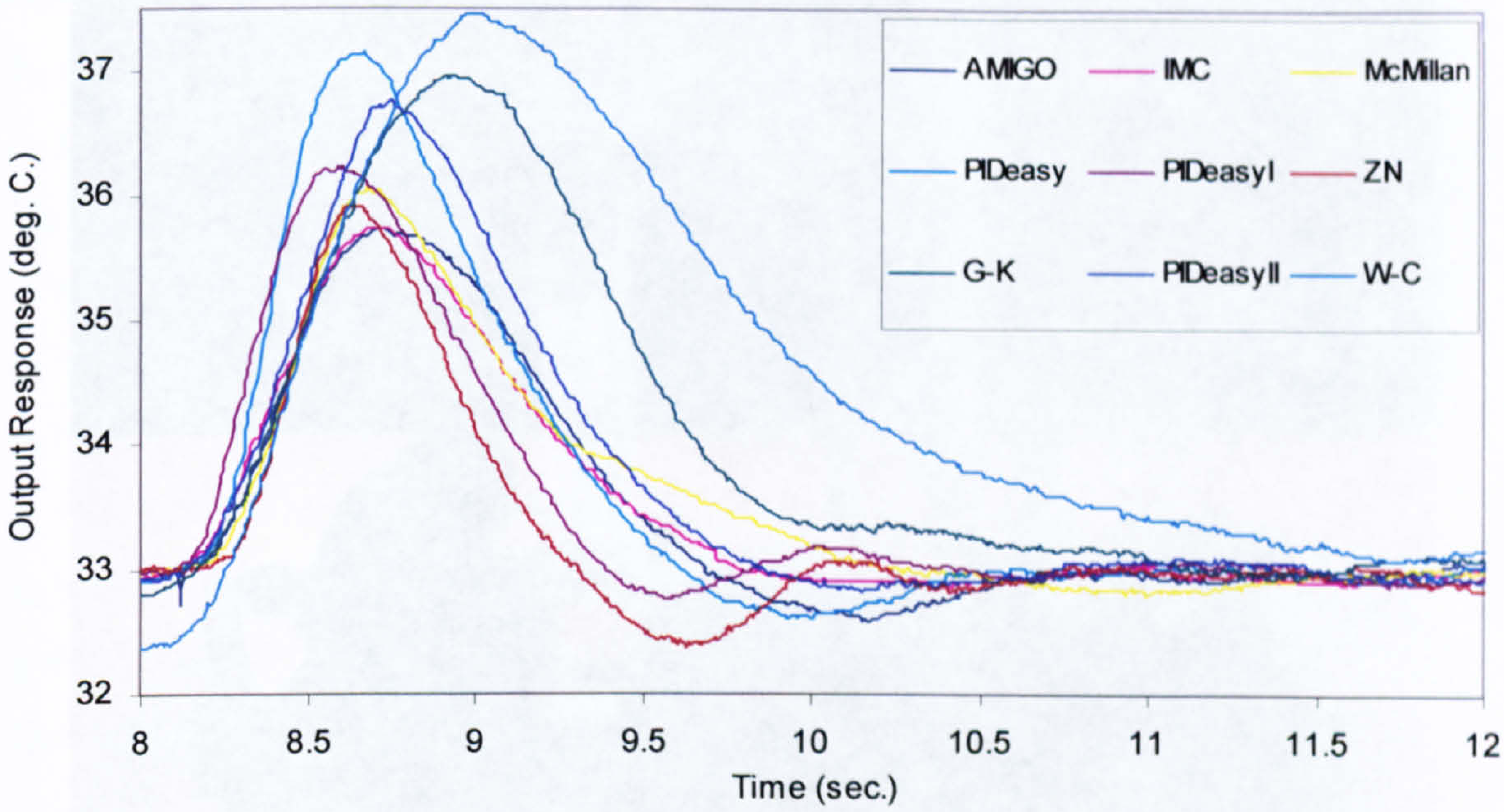

Figure 6.36 PT326 – Performance Comparison on Load Disturbance Rejection with *K* and *L* Under-Estimated by 20%

6.7 TQ CE5 Nonlinear Coupled Tanks System

The system of interest is a nonlinear dynamical system that is used to investigate liquid level control in chemical and dairy plants (Chong and Li, 2000). The system consists of two tanks coupled together by an orifice that connects the two. Liquid is allowed to flow freely between the two tanks. Another orifice in the second tank drains liquid from the tank freely. A pump controls the flow rate of liquid entering the system via the first tank. This setup is illustrated in the diagram in Figure 6.38.

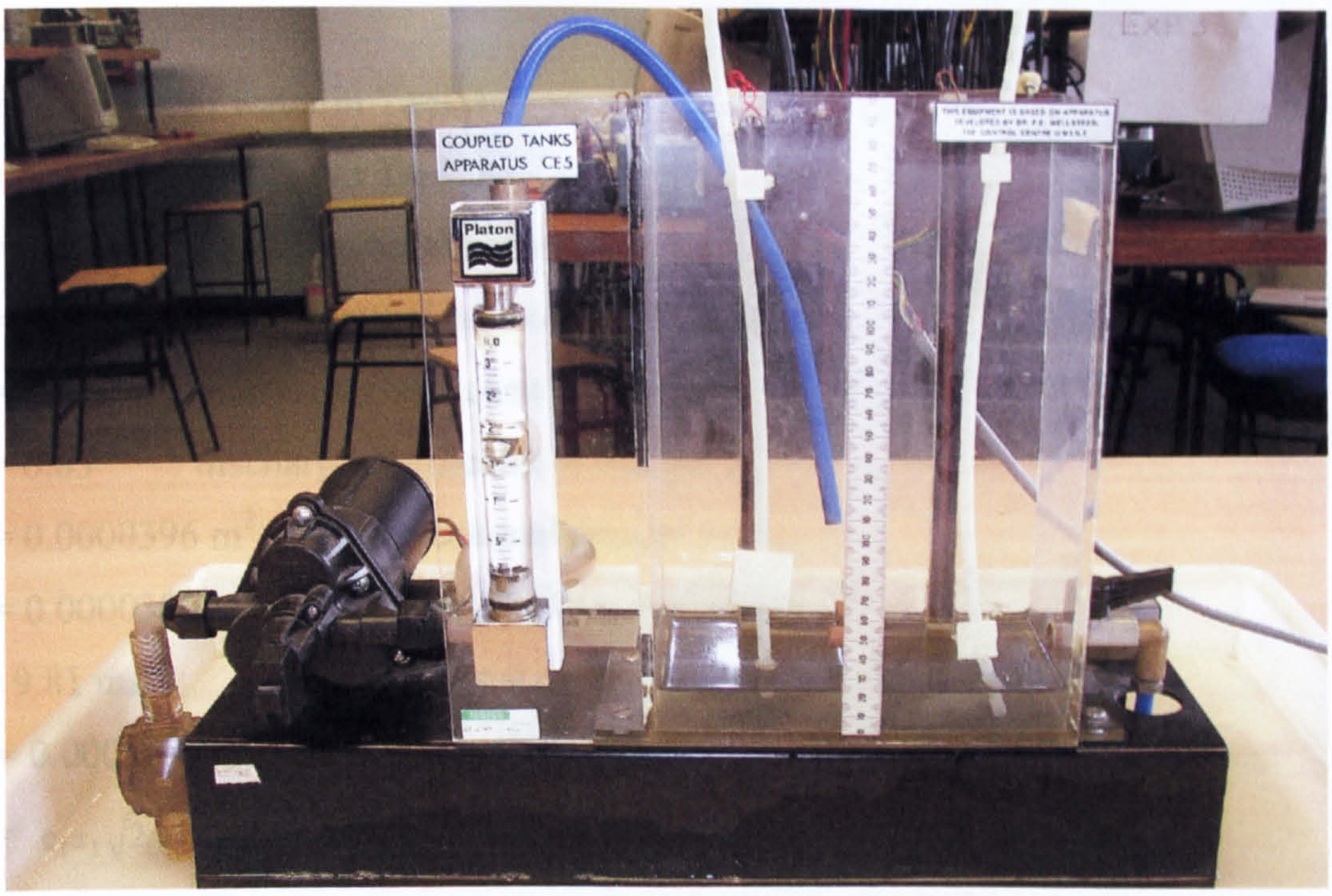

Figure 6.37 TQ CE5 Nonlinear Coupled Tanks System

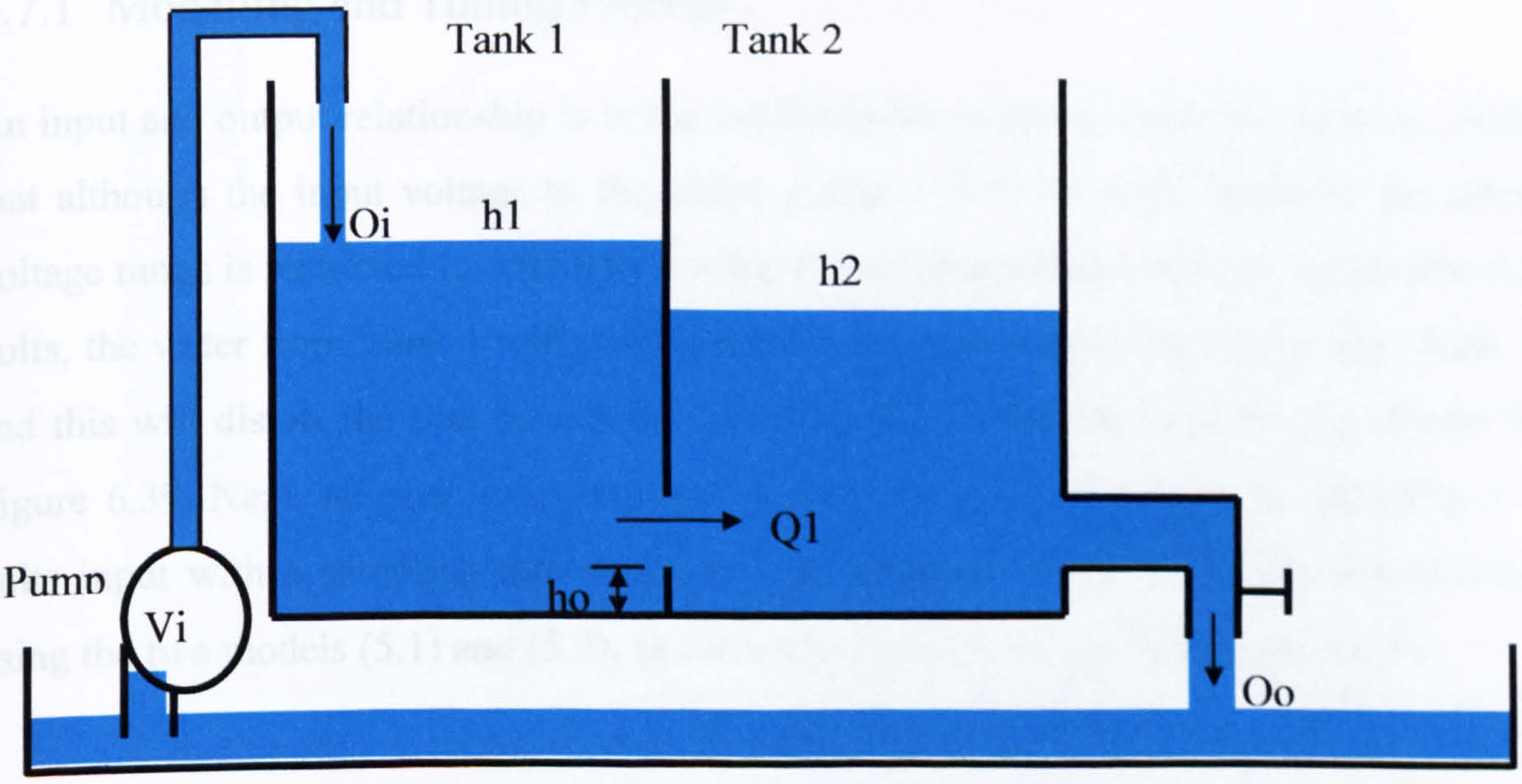

Figure 6.38 CE5 Coupled Tanks System Diagram

The dynamics of this system can be derived from first principles using Bernoulli's mass-balance and flow equations (Tan, 1997). Its behaviour is then approximated by the following state-space equation:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\operatorname{sgn}(h_1 - h_2)\dfrac{c_1 a_1}{A}\sqrt{2g\left|h_1 - h_2\right|} \\ \operatorname{sgn}(h_1 - h_2)\dfrac{c_1 a_1}{A}\sqrt{2g\left|h_1 - h_2\right|} - \dfrac{c_2 a_2}{A}\sqrt{2g(h_2 - H_0)} \end{bmatrix} + \begin{bmatrix} \dfrac{Q_1}{A} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ 0 \end{bmatrix} \tag{6.10}$$

where

h_1 m : height of water in Tank 1

h_2 m : height of water in Tank 2

$H_0 = 0.03$ m : minimum height of water in tank

$A = 0.01$ m^2 : cross sectional area of Tank 1 & 2

$c_1 = 0.53$: discharge coefficient of orifice 1

$c_2 = 0.63$: discharge coefficient of orifice 2

$a_1 = 0.0000396$ m^2 : cross sectional area of orifice 1

$a_2 = 0.0000386$ m^2 : cross sectional area of orifice 2

$g = 9.81$ ms^{-2} : gravitational constant

$Q_i = 0.000007$ $m^3s^{-1}V^{-1}$: pump flow rate

$Q_1 = c_1 a_1 \sqrt{2g(h_1 - h_2)}$ m^3s^{-1} : flow rate from Tank 1 to Tank 2

$Q_0 = c_2 a_2 \sqrt{2g(h_2 - H_0)}$ m^3s^{-1} : discharge rate

6.7.1 Modelling and Tuning Process

An input and output relationship is being established in order to verify its linearity. Note that although the input voltage to the motor pump is 0 to 10 volts, however the input voltage range is restricted to only 0 to 4 volts. If the input voltage were to increase to 4.5 volts, the water from Tank 1 will reach the limit and overflow from the top into Tank 2 and this will distort the true behaviour. The twin-tank nonlinear behaviour is shown in Figure 6.39. Next, an open-loop step test is conducted on the system by injecting a 3 volts input with a sampling rate of 1 sec. The response captured is approximated by using the two models (5.1) and (5.2), as shown in Figure 6.40 and 6.41 respectively.

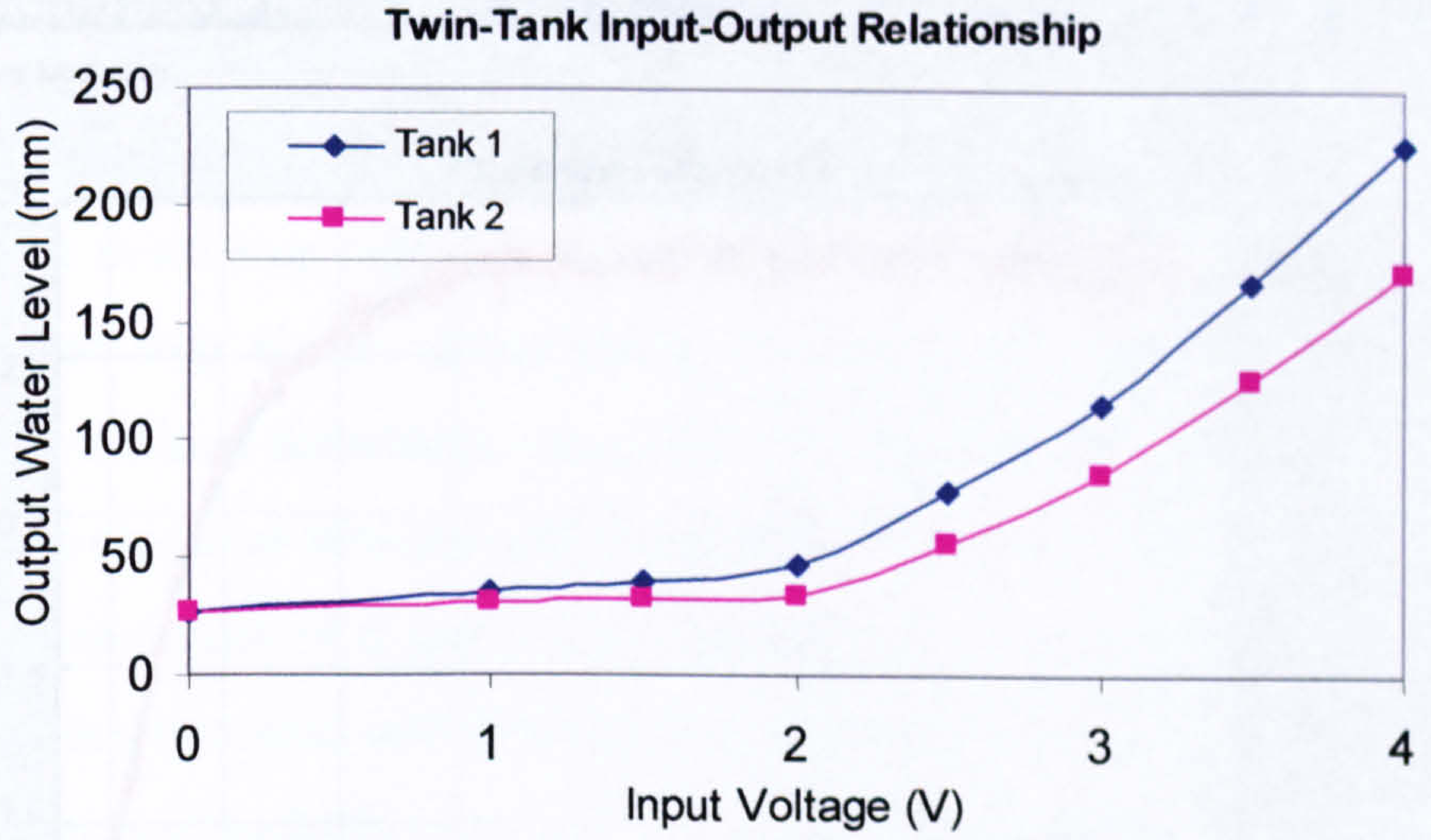

Figure 6.39 CE5 Coupled Tanks Input-Output Relationship

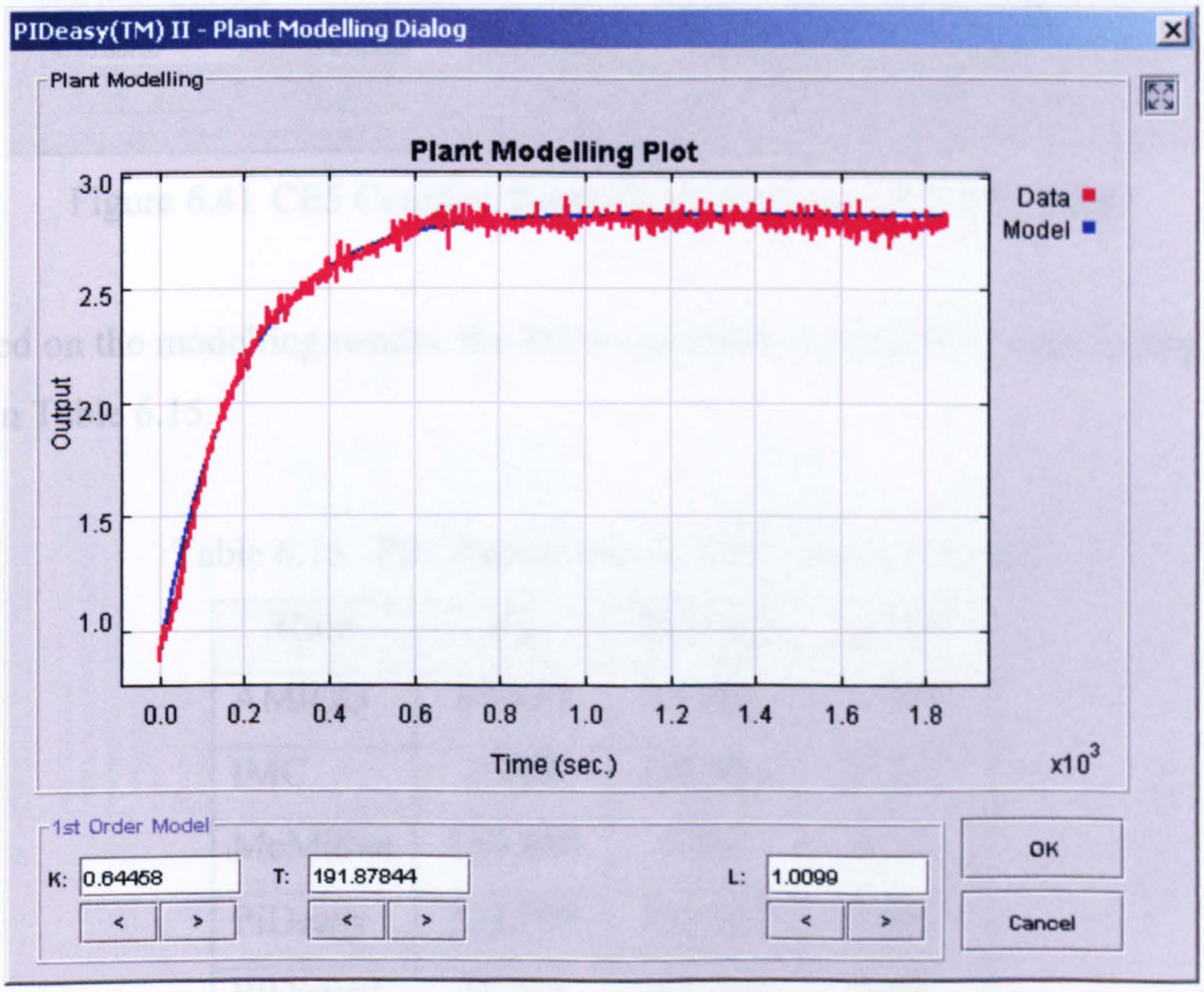

Figure 6.40 CE5 Coupled Tanks Modelling using FOLPD Model

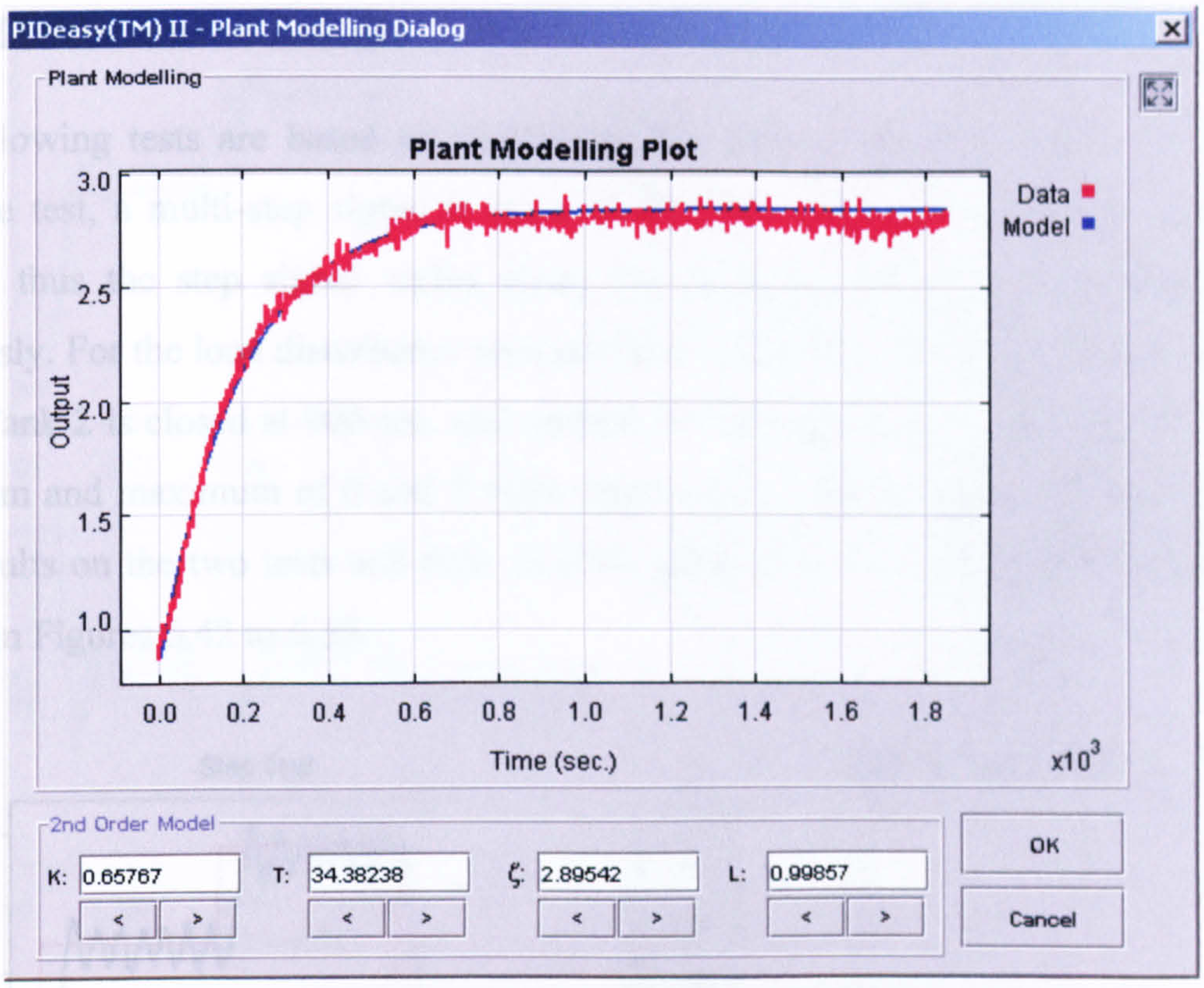

Figure 6.41 CE5 Coupled Tanks Modelling using SOSPD Model

Based on the modelling results, the PID parameters computed by each tuning rule are shown in Table 6.15.

Table 6.15 PID Parameters for CE5 Coupled Tanks

Rule	K_P	T_I **(sec.)**	T_D **(sec.)**
AMIGO	89.029	11.242	0.753
IMC	7.493	192.633	0.752
McMillan	139.840	3.012	0.753
PIDeasy	123.799	192.301	0.481
PIDeasyI	21.215	192.301	0.481
ZN	236.582	3.020	0.755
G-K	1.509	199.103	5.937
PIDeasyII	35.868	199.169	5.934
W-C	121.165	199.103	5.937

6.7.2 Discussion of Results

The following tests are based on controlling the water level of Tank 2. For the step response test, a multi-step signal is injected into the system. Since this is a nonlinear system, thus the step signal varies along the operating point of identification done previously. For the load disturbance response test, while the system is in steady-state, the tap at Tank 2 is closed at 900 sec. and opened at 1000 sec. The actuator limit is set to a minimum and maximum of 0 and 5 volts respectively. The sampling rate used is 1 sec. The results on the two tests and their control signal responses for each tuning rules are shown in Figures 6.42 to 6.50.

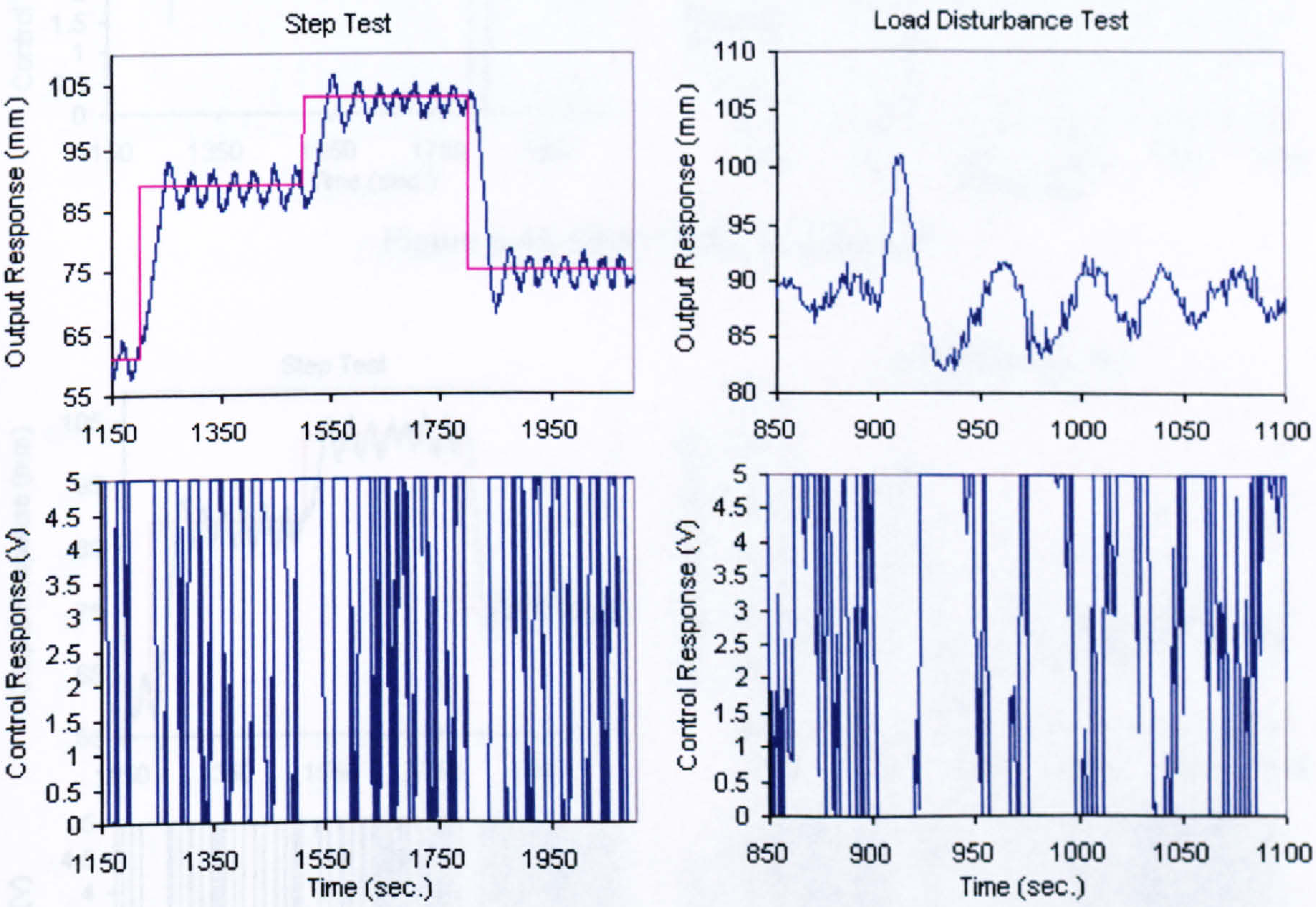

Figure 6.42 CE5 – AMIGO Test Results

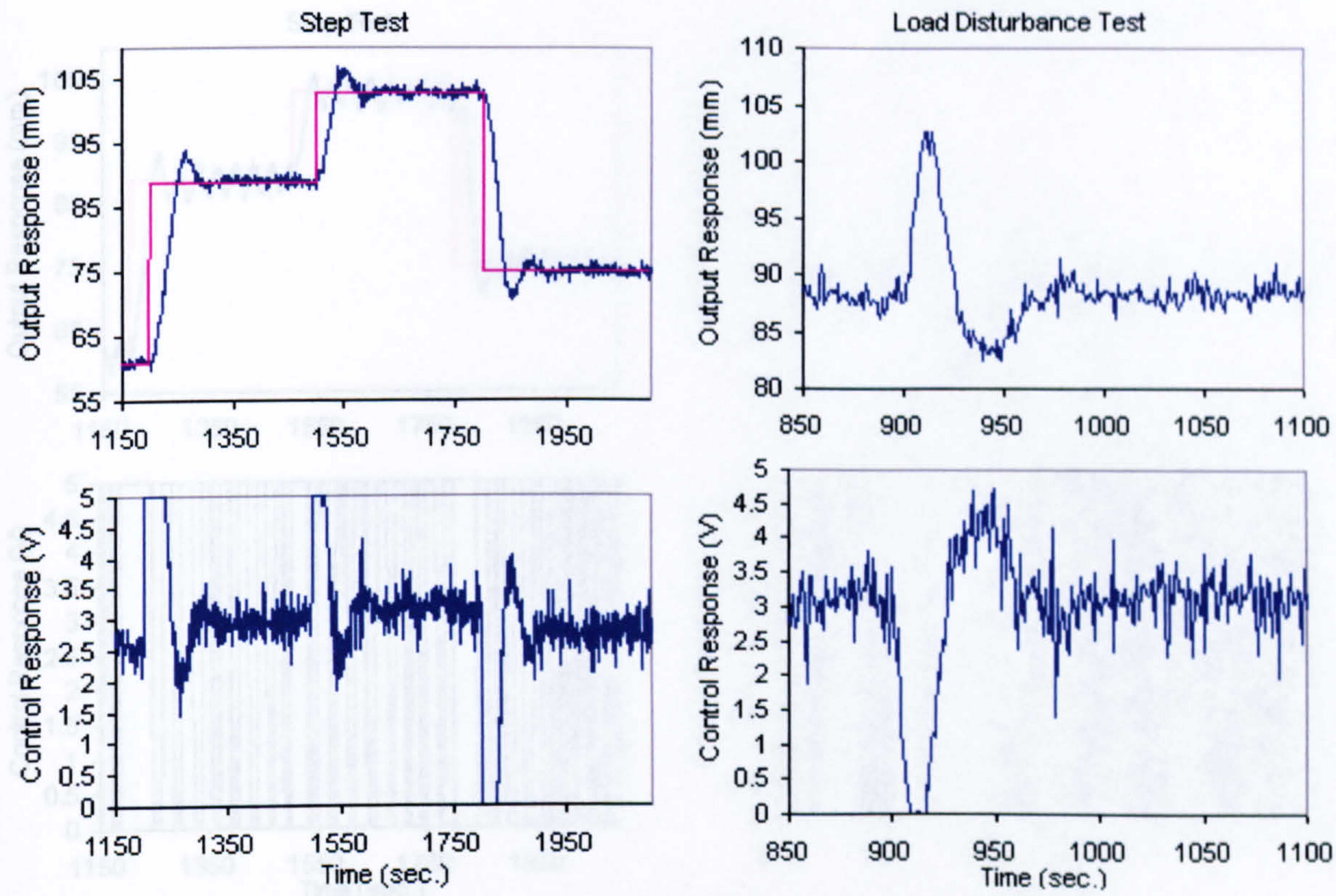

Figure 6.43 CE5 – IMC Test Results

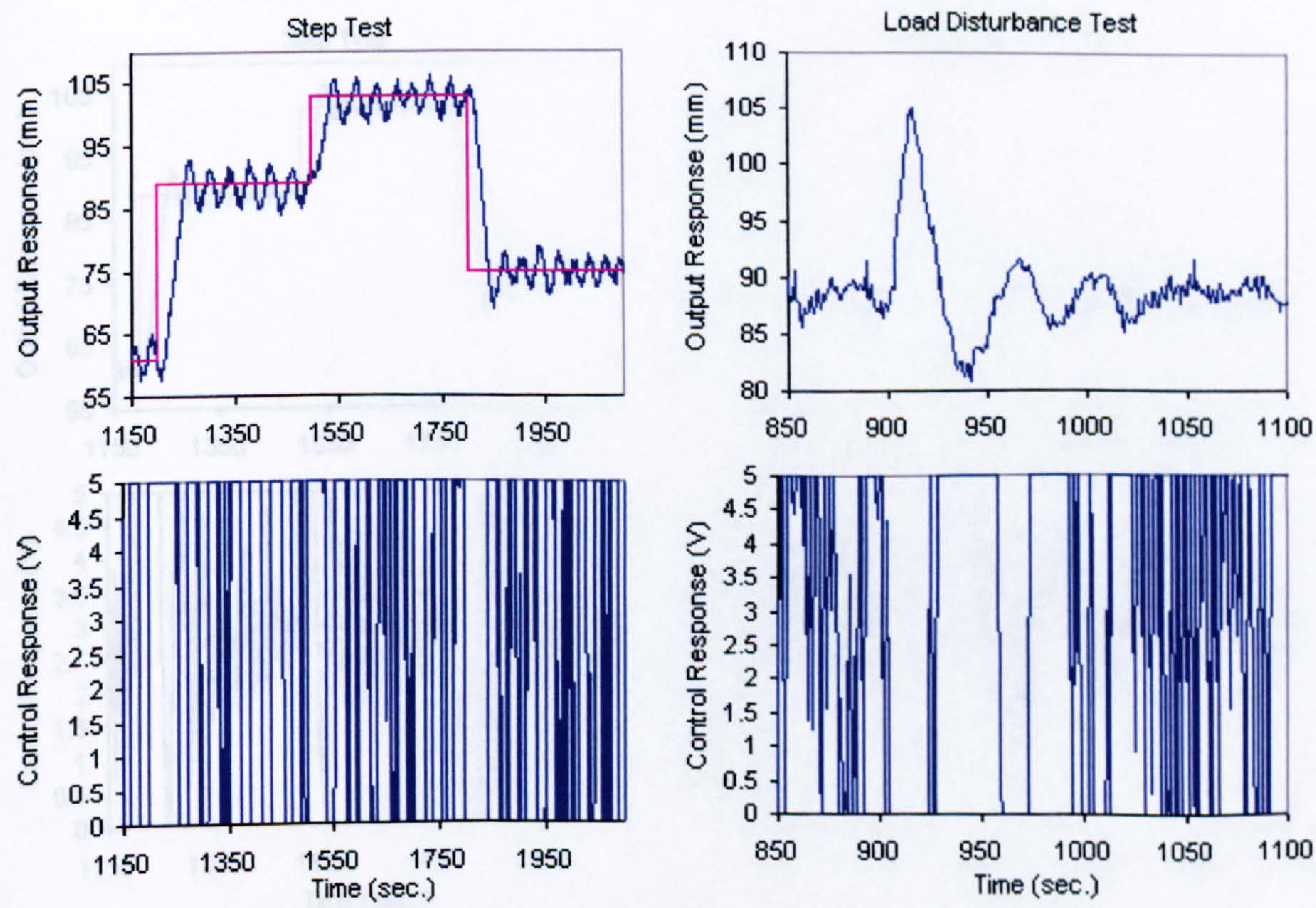

Figure 6.44 CE5 – McMillan Test Results

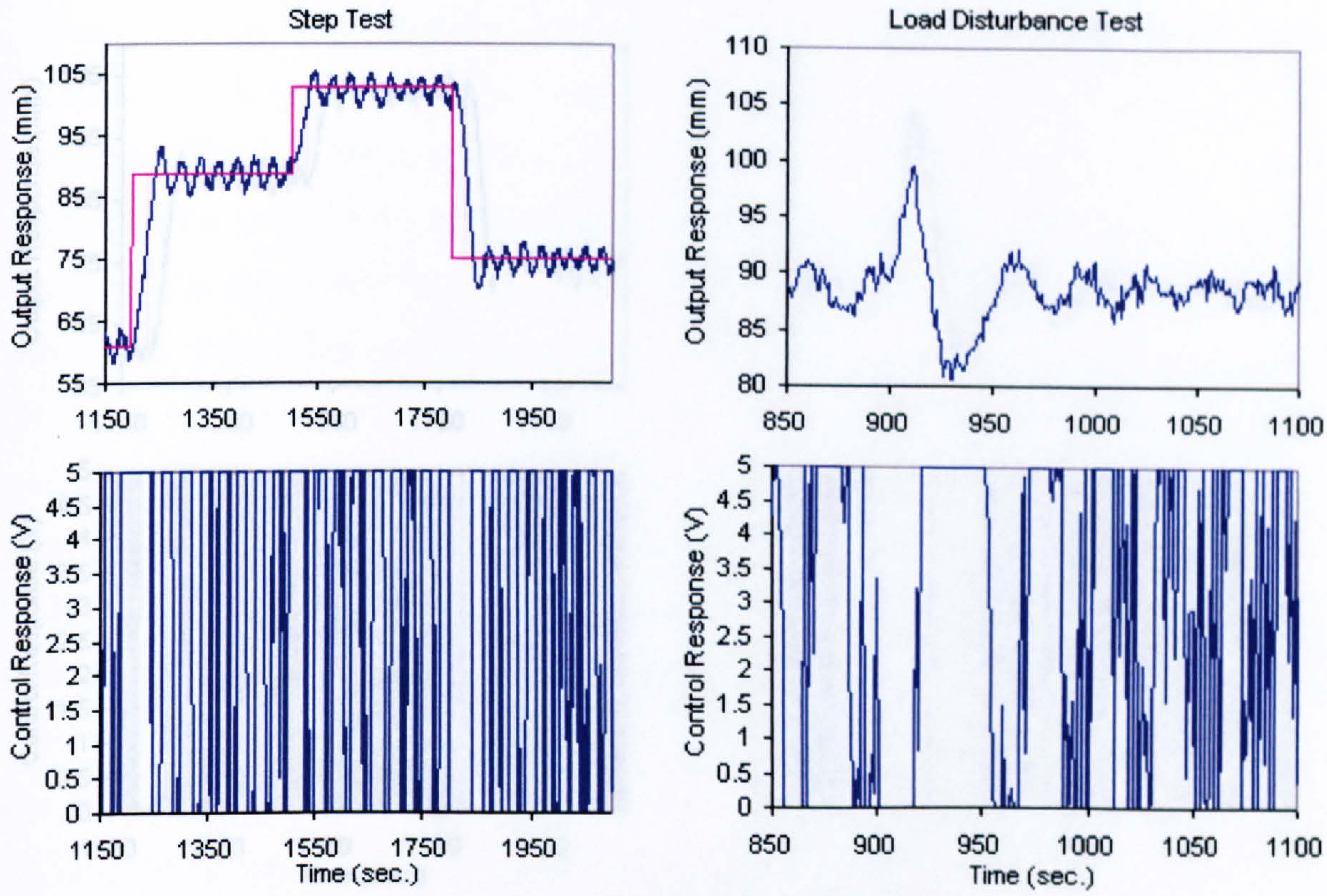

Figure 6.45 CE5 – PIDeasy Test Results

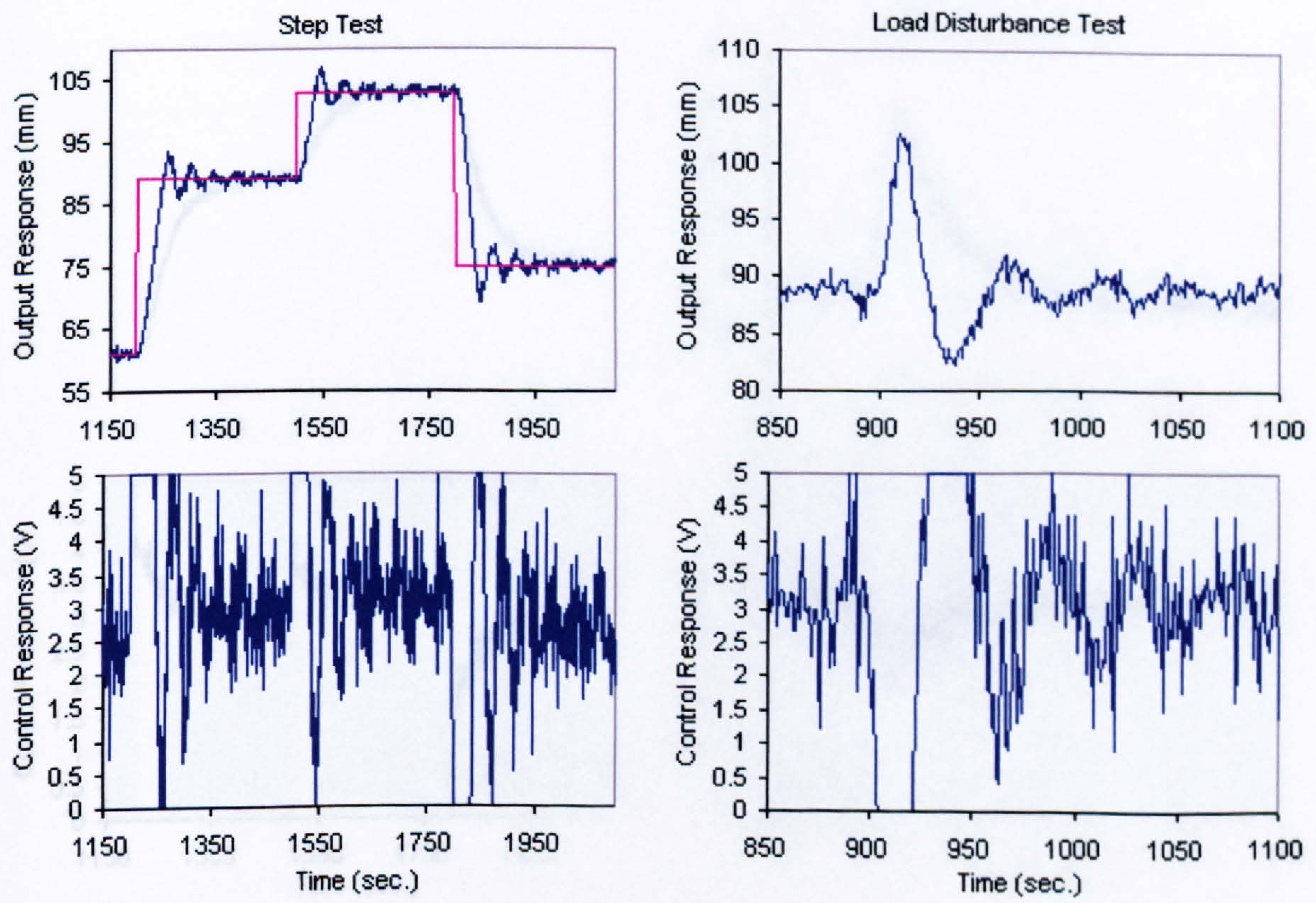

Figure 6.46 CE5 – PIDeasyI Test Results

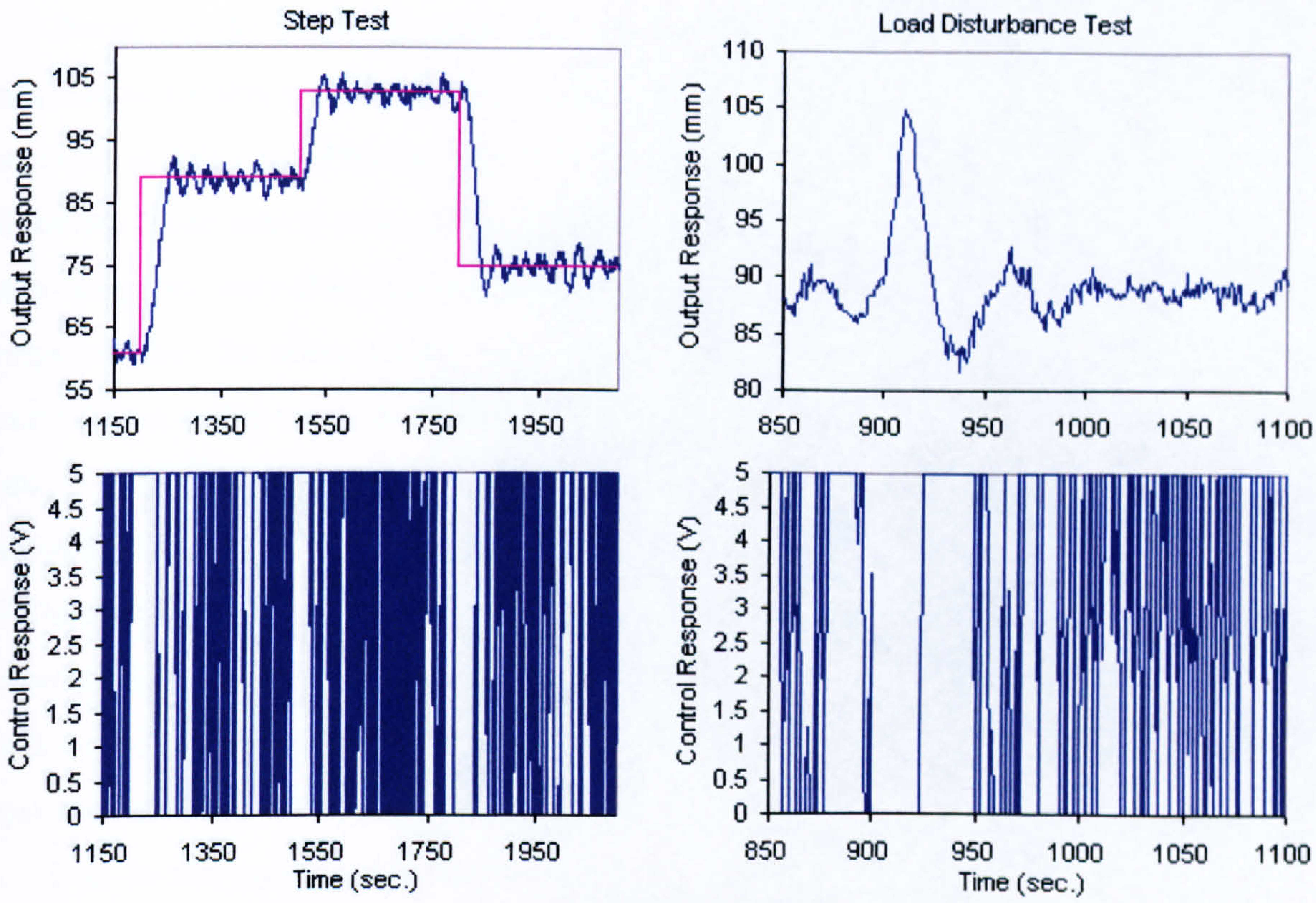

Figure 6.47 CE5 – ZN Test Results

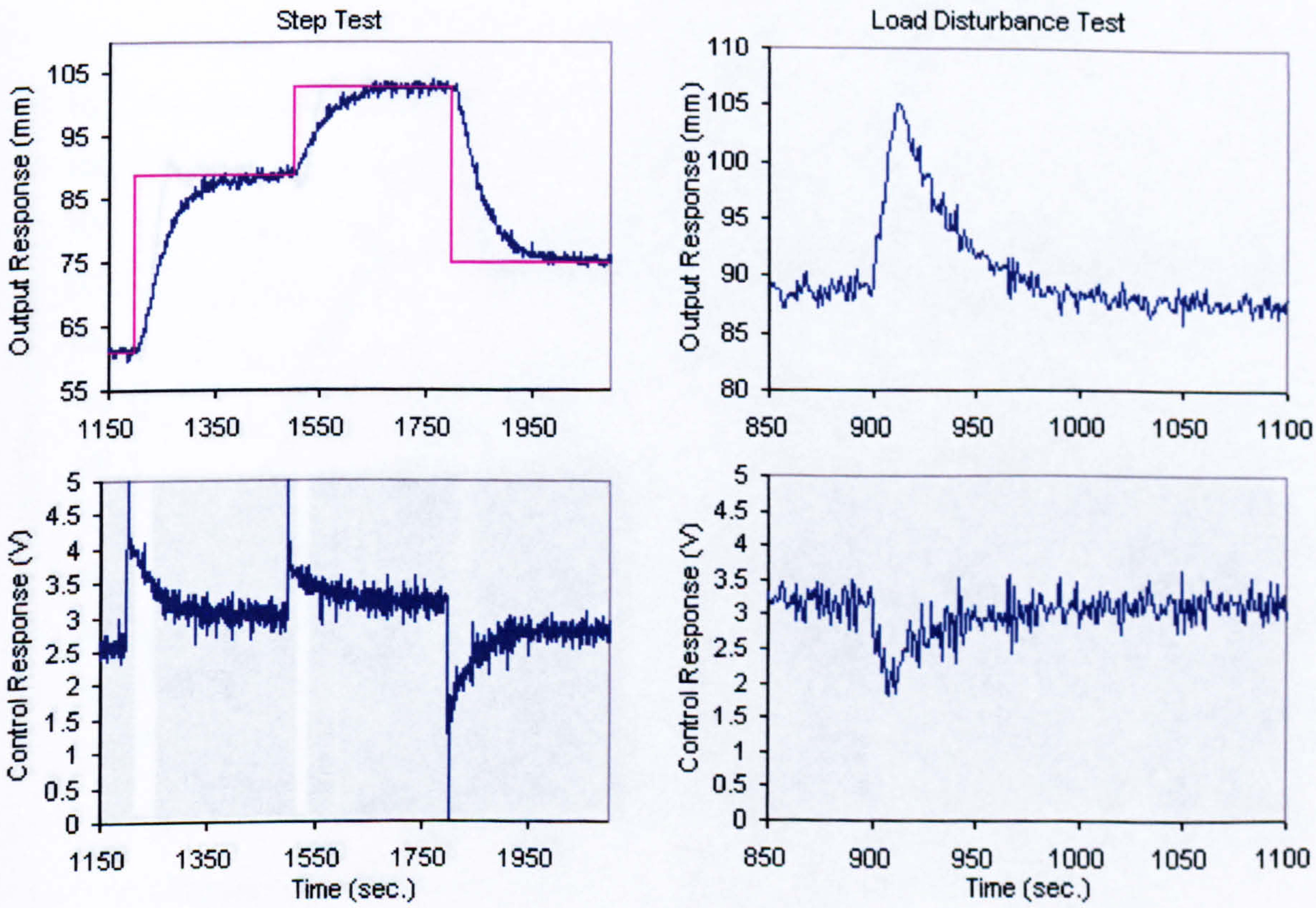

Figure 6.48 CE5 – G-K Test Results

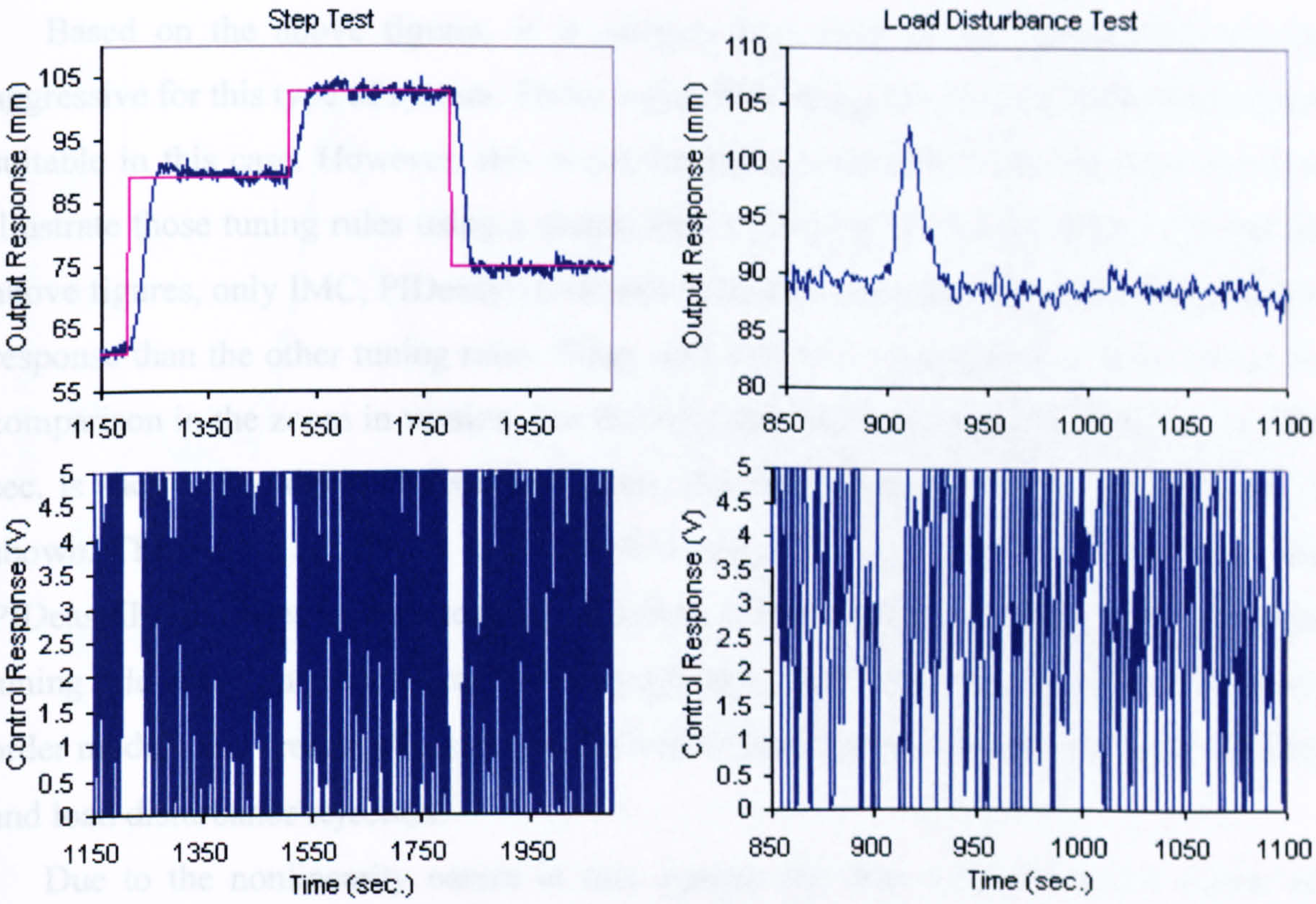

Figure 6.49 CE5 – PIDeasyII Test Results

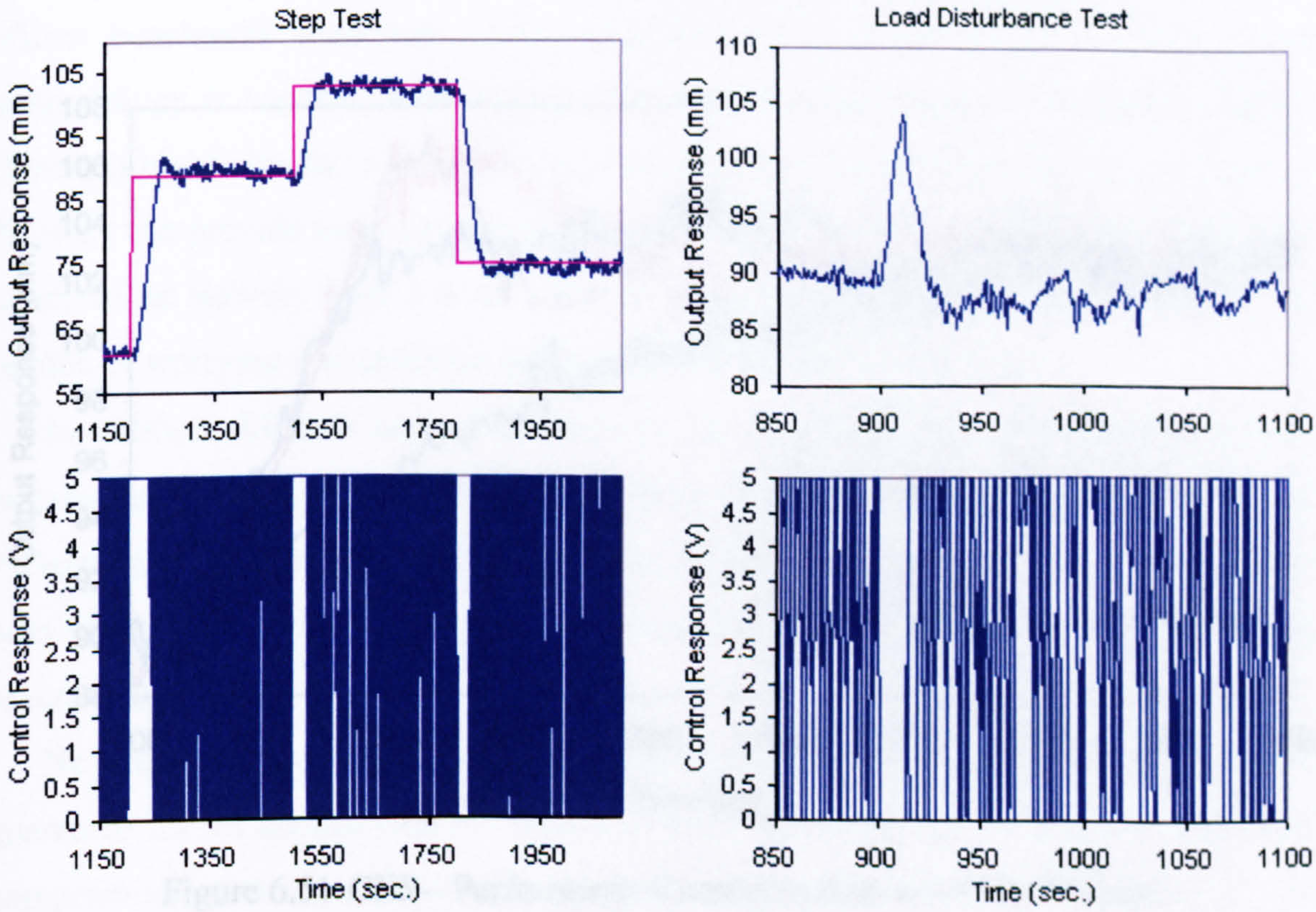

Figure 6.50 CE5 – W-C Test Results

Based on the above figures, it is obvious that most of the tuning rules are too aggressive for this type of system. Those controllers acting on error-squared will be more suitable in this case. However, this is not the main issue here since the objective is to illustrate those tuning rules using a simple PID controller. Based on Table 6.15 and the above figures, only IMC, PIDeasyI, G-K and PIDeasyII tunings produce slightly damped response than the other tuning rules. Thus, only this four tuning rules will be shown for comparison in the zoom in version. For the step test, the time frame at 1500 sec. to 1680 sec. is shown. For the load disturbance test, the time frame at 880 sec. to 1050 sec. is shown. The results are shown in Figure 6.51 and 6.52. It is instantly recognisable that PIDeasyII dominates in both tests. In addition, it illustrates that for this type of system tuning rule based on second-order model performs much better than those based on first-order model. The present of a stronger derivative proves useful in both set-point tracking and load disturbance rejection.

Due to the nonlinearity nature of this system, the step tests conducted earlier are enough to confirm that the robustness against modelling error. Thus, the tuning rules will not be tested again by changing any of the model parameters.

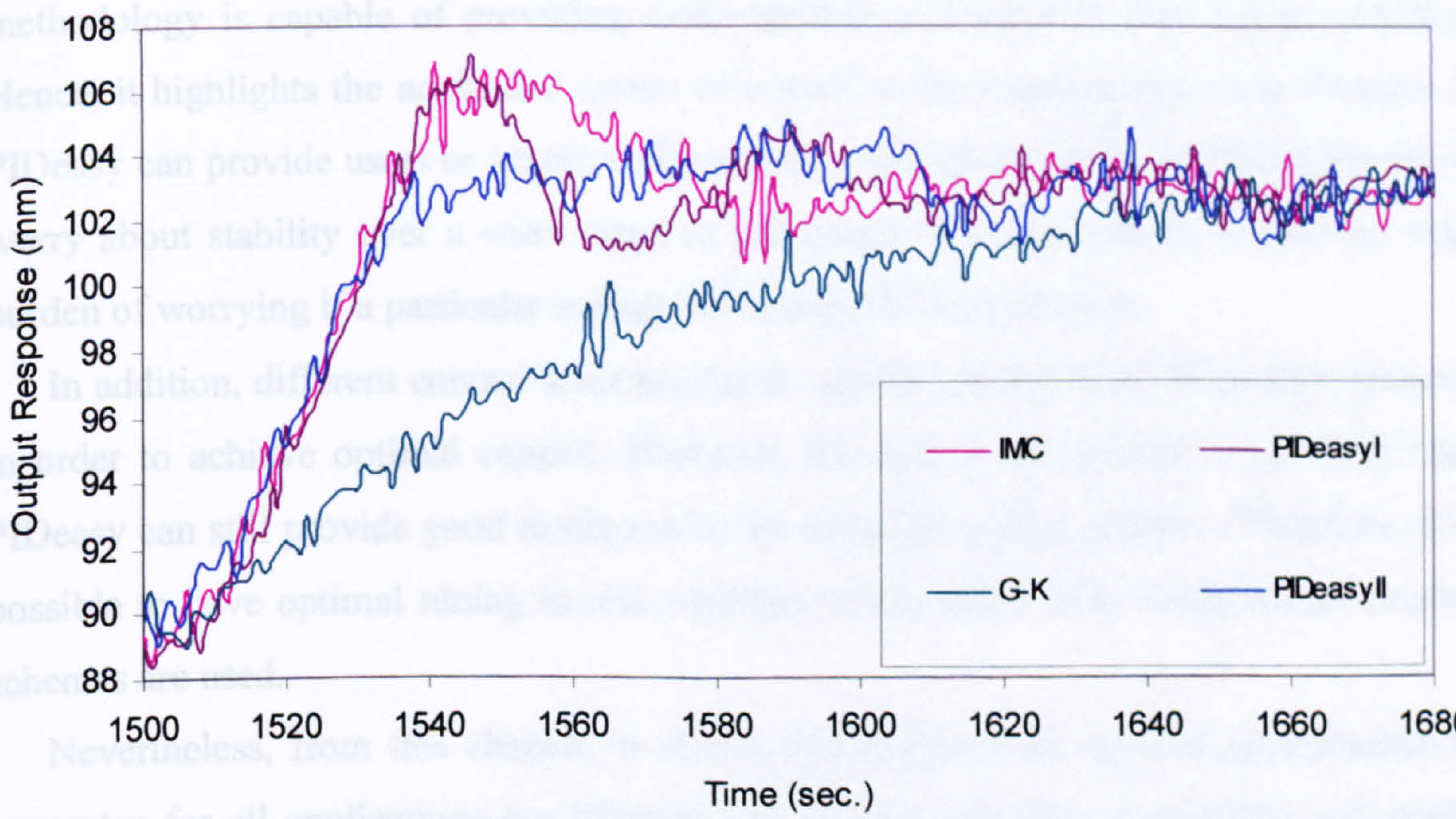

Figure 6.51 CE5 – Performance Comparison on Set-Point Change

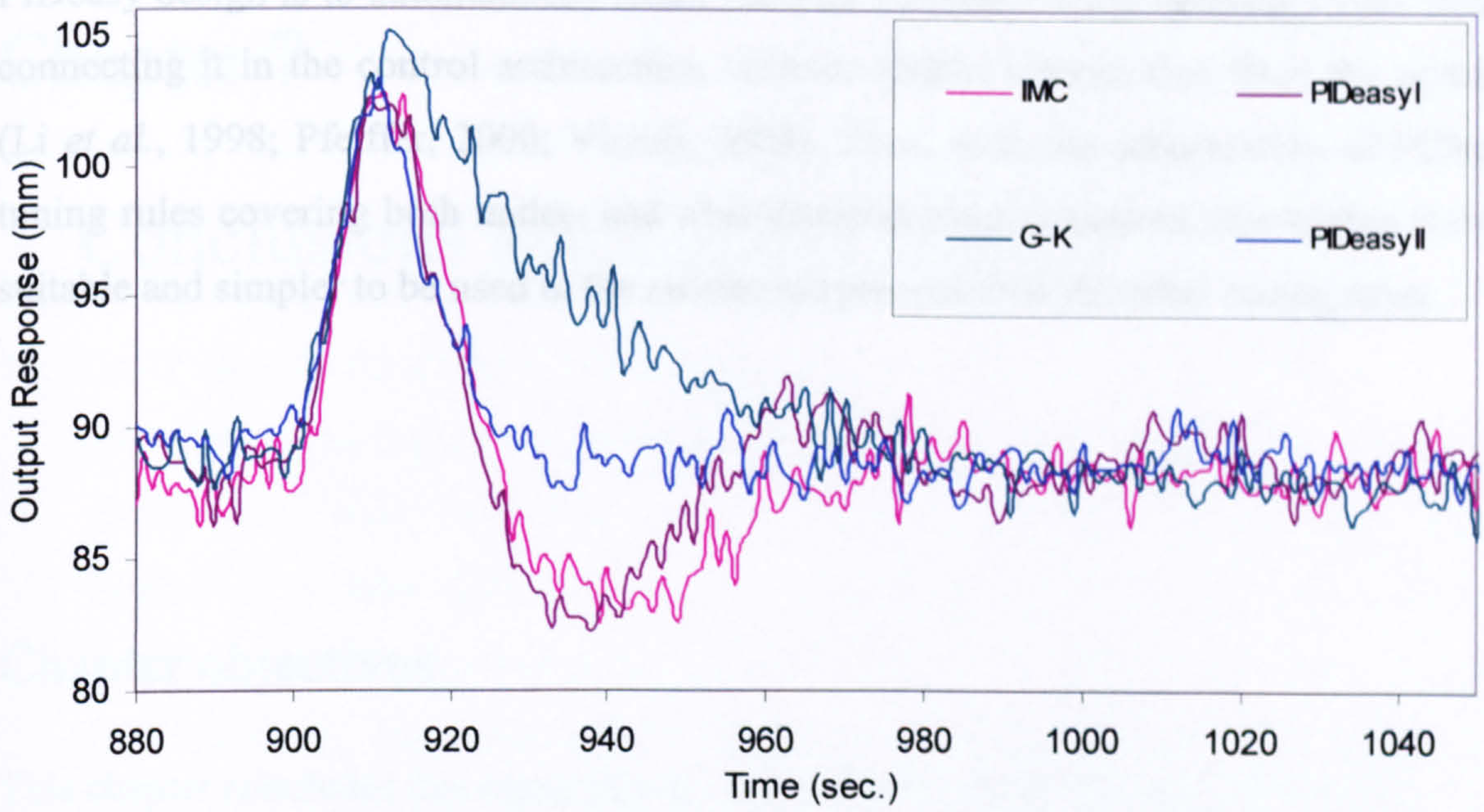

Figure 6.52 CE5 – Performance Comparison on Load Disturbance Rejection

6.8 Summary

Every tuning methodology is designed based on certain specifications. Based on all the offline benchmark tests and online application tests, it shows that PIDeasy tuning methodology is capable of providing multi-optimal performance with robust stability. Hence, it highlights the needs and merits of a multi-criteria design shown in Chapter 5. PIDeasy can provide users or adaptive algorithms, good initial settings without having to worry about stability over a wide range of processes. This can greatly reduce the user burden of worrying if a particular tuning rule is suitable for a process.

In addition, different control schemes can be applied on the three laboratory systems in order to achieve optimal control. However, the aim of the studies is to verify that PIDeasy can still provide good tunings with the simplest control scheme. Therefore, it is possible to have optimal tuning in real situation where much more complicated control schemes are used.

Nevertheless, from this chapter, it shows that although no optimal performance is guarantee for all applications but PIDeasy can at least provide a reasonable and stable compromised performance with minimal changes to the whole control structure. The tests conducted in this chapter thus verify the stability and robustness of PIDeasy tuning rules designed based on the first- and second-order models. The ultimate aim of the

PIDeasy design is to automatically make the PID controller work optimally after simply connecting it in the control architecture, without further intervention from the operator (Li *et al.*, 1998; Pfeiffer, 2000; Visioli, 2003). Thus, with the achievement of PIDeasy tuning rules covering both under- and over-damped plant responses, this makes it more suitable and simpler to be used in the automated process than the other tuning rules.

Chapter 7

Conclusions and Further Work

Chapter objectives

This chapter concludes this research and suggests some further enhancements to it.

7.1 Conclusions

This research is motivated by the wide application of PID control and problems tuning in it. The aim is to achieve the most optimal results by making use of the simplest method without over-complication. This research has successfully and satisfactorily achieved all the targets stated in the scope of this research. They are summarised as follows:

- By utilising the capability of multi-objective evolutionary algorithms, this research has devised a truly multi-objective PID tuning rule that significantly outperforms all other tuning rules based on multi-criteria optimality. For FOLPD plants, the evolved rules can cover a delay to time constant ratio from zero to infinity. For SOSPD plants, they also cover all possible dynamics found in practice. This is verified by the offline computer simulation tests and online laboratory system tests. These tests cover almost all aspects of real world situations and concerns such as optimal performance and robustness over a range of processes, noise, modelling error, load disturbances, atmospheric interferences etc.
- A software tool arising from the development of the PID tuning rules is being developed. It can be used for academic teaching where it allows user to test the performance of different tuning rules. The tool is expandable to accommodate any new tuning rules. It is platform independent due to the programming language used and is affordable to any user as compared to any of the existing tools where user needs to install expensive software applications.
- All the above PID developments are done based on the extensive study and analysis on the practical PID software packages, hardware modules and patents. It is through the study that reveals the need for the development of a software tool and a tuning rule designed based on multi-objective considerations.
- By devising a simple to understand and implement multi-objective evolutionary algorithm, s-MOEA. Its performance is comparable with any of the commonly cited evolutionary algorithms as verified by the tests on a wide range of multi-objective test problems. Thus this would be attractive to any user who wishes to try out MOEA without much development efforts.
- An achievement from the evolutionary computing side is the proposed visualisation technique, which greatly assists the comparison of results on multi-dimensional

data. This visualisation technique is a result of extensive study and analysis on the existing metrics for comparing multi-objective evolutionary algorithms.

In conclusion, this research has been successful and has satisfactorily addressed the problems raised in the Introduction chapter. Not only are the problems addressed, new methodology and software that are both efficient and easy-to-use, have been developed to assist future users in their work and research.

7.2 Further Work

With reference to the methodologies proposed and applied in this research, it would be beneficial to discuss some alternative ways that could further improve the work.

Starting with the evolutionary computing part, the development of MOEAs has been relatively mature. Further attempts to improve an MOEA will lead to minimal improvement only, which usually does not justify more research effort. However, a more formal and effective way of measuring MOEA performance is still lacking at present. Hence, this part of MOEA research should deserve more attention. Indirectly it can improve the performance and effectiveness of the algorithm by understanding the problems at hand better. Current frameworks proposed are still too complicated to be applied efficiently. Any future metrics should therefore be designed on an efficient framework with ease of implementation.

At present, the DD chart covers only the two main aspects of MOEA functional goals and is only targeted to unary type of metrics. Further attempts to extend the DD chart to cover other concerned area of MOEA performance and to other types of metrics would thus be useful.

A fully 'plug-and-play' automated PID controller would be valuable to users as it eases the process of tuning and commissioning a PID controller. Some simple additional steps can be added to the current design so as to achieve automated 'plug-and-play'. Since most PID controllers are now implemented digitally, following modifications can be carried out. First, an identification block can be added to identify the plant. This identification block can be implemented in any fashion as long as a plant model is supported. PIDeasy supports two types of plant models; PIDeasyII may be used in most cases unless strong derivative action is of concern. Then PIDeasyI can be used to compute PID parameters for critically- and over-damped plant response. PIDeasyII will be best for under-damped plants. Once PID parameters are optimally computed or updated, the PID controller can immediately resume control of the plant. For adaptive type of tuning, PIDeasy may also be applied for Model Predictive Control (MPC) where a continuously updated internal plant model exists in the control architecture. In this case, PIDeasy tuning rule can instantaneously compute a new set of PID parameters based on the updated internal plant model.

Last point of future work concerns PIDeasy-II™ software tool development. More tests can be done to refine the stability and accuracy of the simulation, although as currently stands it is usable for most cases. One valuable enhancement would be to make it capable of communicating with hardware through an Ethernet interface, so as to realise networked control. Then it will be possible to automate the whole process from identification, modelling, tuning and control, as discussed in the previous paragraph. Finally, another possible enhancement is to consider frequency domain identification and tuning. This would make the PIDeasy-II™ tool appeal to a wider audience.

References

ABB (2001a). *Instruction Manual of MICRO-DCI™ 53SL6000.* PN24991, U.S.A.

ABB (2001b). *Specification DataFile of Commander 355.* SS/C355, Issue 3.

ABB (2004). The ABB group [online]. Available from: <http://www.abb.com> [14 May 2004].

ACT (2004). ACT GmbH homepage [online]. Available from: <http://www.act-control.com> [14 May 2004].

ADAPTECH (2004). Adaptech – identification and advanced control [online]. Available from: <http://www.adaptech.com/Ang/menu_a.html> [14 May 2004].

Algosys Inc. (2004). Algosys – solutions & technology [online]. Available from: <http://www.algosys.com/solutions/othertech.asp> [14 May 2004].

Ang, K.H. (1996). *Discovering Tractable Ziegler-Nichols PID Tuning Rules by Genetic Programming.* Undergraduate 1995/96 Project Report No. 118, University of Glasgow, U.K.

Aoki, S. (1993). *Method and apparatus using fuzzy logic for controlling a process having dead time.* U.S. Patent No. 5272621.

Arcara, S.A. and Anderson, P. (1983). *Self tuning of P-I-D controller by conversion of discrete time model identification parameters.* U.S. Patent No. 4407013.

Artcon Inc. (2004). Artcon inc. [online]. Available from: <http://artcon.com/artcon.html> [14 May 2004].

Åström, K.J. and Hägglund, T. (1988). *Method and an apparatus for automatically tuning a process regulator.* U.S. Patent No. 4758943.

Åström, K.J., Hägglund, T., Hang, C.C., and Ho, W.K. (1993). Automatic tuning and adaptation for PID controllers – a survey. *Control Engineering Practice*, 1(4):699-714.

Åström, K.J. and Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. 2nd Edition, Instrument Society of America, Research Triangle Park, North Carolina.

Åström, K.J. and Hägglund, T. (1996). PID control. In: W.S. Levine (Ed), *The Control Handbook*, CRC Press/IEEE Press, pp. 198-209.

Åström, K.J. and Hägglund, T. (2000). Benchmark systems for PID control. In *Proceedings of the IFAC Workshop on Digital Control Past, Present and Future of PID Control*, Terrassa, Spain, pp. 165-166.

Åström, K.J. and Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9(11):1163-1175.

Åström, K.J. and Hägglund, T. (2004). Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635-650.

Bäck, T., Fogel, D.B., and Michalewicz, Z., editors (1997). *Handbook of Evolutionary Computation*. IOP Publishing Ltd and Oxford University Press.

Barlow, P. and Jones, K.R. (1980). *Three term (PID) controllers*. U.S. Patent No. 4214300.

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., and Stewart, W.R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9-32.

BESTune (2004). PID controller tuning [online]. Available from: <http://bestune.50megs.com> [14 May 2004].

Bienert, P., Rechenberg, I., and Schwefel, H.-P. (1966). *Messung kleiner Wandschubspannungen bei turbulenten Grenzschichten in Ablösenähe.* Technical Report of the Hermann Föttinger-Institute for Fluid Dynamics, Wi 8/45, Technical University of Berlin.

Bohn, C. and Atherton, D.P. (1995). An analysis package comparing PID anti-windup strategies. *IEEE Control Systems Magazine,* 15(2):34-40.

Bosman, P.A.N. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation,* 7(2):174-188.

Bristol II, E.H. (1974). *Pattern evaluation method and apparatus for adaptive control.* U.S. Patent No. 3798426.

Brosilow, C.B. (2002). Coleman brosilow [online]. Available from: <http://www.cwru.edu/cse/eche/people/faculty/brosilow/brosilow.htm> [30 June 2002].

Bunzemeier, A. (1998). *Self controllable regulator device.* U.S. Patent No. 5796608.

Cao, R. and McAvoy, T. (1990). Evaluation of a pattern recognition adaptive PID controller. *Automatica,* 26(4):797-801.

Chand, S. (1992). *Self-monitoring tuner for feedback controller.* U.S. Patent No. 5159547.

Chen, C.-L. (1989). A simple method for on-line identification and controller tuning. *American Institute of Chemical Engineers Journal,* 35(12):2037-2039.

Chong, G. and Li, Y. (2000). Trajectory controller network and its design automation through evolutionary computing. In *Proceedings of the EvoWorkshops 2000: Real-World Applications of Evolutionary Computing,* Volume 1803 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 139-146.

Chu, Y.T., Chen, J.K., Tseng, W.L., Tseng, C.G., Chang, S.F., and Chong, K.G. (1993). *Method and an apparatus for PID controller tuning.* U.S. Patent No. 5229699.

Coello Coello, C.A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems: An International Journal,* 1(3):269-308.

Coello Coello, C.A., Van Veldhuizen, D.A., and Lamont, G.B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic / Plenum Publishers, New York.

Cohon, J.L. (1978). *Multiobjective Programming and Planning.* Academic Press, New York.

Communications & Systems (2004). COTEMS [online]. Available from: <http://www.cotems.com/anglais/default.htm> [14 May 2004].

Control Arts Inc. (2004). Control arts inc: process control, alarm analysis, abnormal situation management, engineering analysis and operations management software [online]. Available from: <http://www.controlartsinc.com/index.html> [14 May 2004].

Control & Optimization Specialists (2004). COSpecialists – control & optimization specialists [online]. Available from: <http://www.cospecialists.com/index.html> [14 May 2004].

Control Engineering (1998). Single-loop controllers dominate marketplace [online]. Available from: <http://www.manufacturing.net/ctl/article/CA189397> [14 May 2004].

CONTROL Magazine (2004). CONTROL Magazine [online]. Avaliable from: <http://www.controlmag.com> [14 May 2004].

Control Soft Inc. (2004). ControlSoft inc. – the company with leading control technologies [online]. Available from: <http://www.controlsoftinc.com/index.shtml> [14 May 2004].

Cooper, D. (2004). Control station is software for process control training, tuning and analysis [online]. Available from: <http://www.engr.uconn.edu/control/index.html> [14 May 2004].

Corripio, A.B. (2001). *Tuning of Industrial Control Systems.* 2nd Edition, Instrumentation, Systems, and Automation Society, Research Triangle Park, North Carolina.

CP Engineering Systems Ltd (2004). NF CADET v12 page 1 [online]. Available from: <http://www.cpengineering.com/exthtmnf/NFcadetv12.html> [14 May 2004].

Cutler Johnston Corporation (2004). Products [online]. Available from: <http://www.cutlerjohnston.com/products.htm> [14 May 2004].

Dahlin, E.B. (1970). *Method for adjusting controller gain to control a process.* U.S. Patent No. 3532862.

Darwin, C. (1859). *On the origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life.* London, Murray.

Das, I. And Dennis, J. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63-69.

De Jong, K.A. (1975). *An Analysis of the Behavior of a class of Genetic Adaptive Systems.* PhD Thesis, University of Michigan, Dissertation Abstracts International 36(10), 5140B, University Microfilms No. 76-9381.

De Jong, K.A., Potter, M.A., and Spears, W.M. (1997). Using problem generators to explore the effects of epistasis. In *Proceedings of the 7th International Conference on Genetic Algorithms*, Michigan State University, East Lansing, U.S.A., pp. 338-345.

Deb, K. (1989). *Genetic Algorithms in Multimodal Function Optimization.* Masters Thesis, TCGA Report No. 89002, The Clearinghouse for Genetic Algorithms, University of Alabama, Tuscaloosa.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, U.K.

Deb, K. and Agrawal, R.B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115-148.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, pp. 849-858.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Hawaii, U.S.A., pp.825-830.

Delta Tau Data Systems Inc. (2004). PMAC executive pro suite [online]. Available from: <http://www.deltatau.com/PMACExecutiveSuite.htm> [14 May 2004].

DeltaV (2004). DeltaV | product data sheet [online]. Available from: <http://www.easydeltav.com/productdata/pds/index.asp> [14 May 2004].

Eder, H.H. (2003). New PID versions and options – useful features or just gimmicks? [online]. Available from: <http://www.manufacturing.net/ctl/article/CA328672> [13 August 2004].

EngineSoft (2003). PIDtune – integrated system identification and PID controller design [online]. Available from: <http://www.pidtune.com/index.html> [29 October 2003].

EnTech (2004). Emerson process management – entech – tuner module [online]. Available from: <http://www.emersonprocess.com/entechcontrol/products/tune/index.asp> [14 May 2004].

ExperTune Inc. (2004). Process performance monitoring & optimization, analysis, simulation, and PID tuning [online]. Available from: <http://www.expertune.com> [14 May 2004].

Farhang-Mehr, A. and Azarm, S. (2003). Minimal sets of quality metrics. In *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, Volume 2632 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 405-417.

Feng, W., Brune, T., Chan, L., Chowdhury, M., Kuek, C.K., and Li, Y. (1998). Benchmarks for testing evolutionary algorithms. In *Proceedings of the 3rd Asia-Pacific Conference on Control and Measurement*, China Aviation Industry Press, Dunhuang, P.R. China, pp. 134-138.

Feng, W. and Li, Y. (1999). Performance indices in evolutionary CACSD automation with application to batch PID generation. In *Proceedings of the 10th IEEE International Symposium on Computer Aided Control System*, Hawaii, U.S.A., pp. 486-491.

Fogel, L.J., Owens, A.J., and Walsh, M.J. (1966). *Artificial Intelligence Through Simulated Evolution*. John Wiley, New York.

Fonseca, C.M. (1995). *Multiobjective Genetic Algorithms with Applications to Control Engineering Problems*. PhD Thesis, Dept. of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

Fonseca, C.M. and Fleming, P.J. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, pp. 416-423.

Fonseca, C.M. and Fleming, P.J. (1995a). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1-16.

Fonseca, C.M. and Fleming, P.J. (1995b). Multiobjective genetic algorithms made easy: selection, sharing, and mating restriction. In *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield, U.K., pp. 45-52.

Foxboro (1995). *Technical Information of EXACT Tuning with 762, 760, and 740 Series Controllers.* TI039-200, August, U.S.A.

Foxboro (1996). *Product Specification of I/A Series® 716C 1/16 DIN Temperature Controller.* PSS 2C-1B5 A, U.S.A.

Friedberg, R.M. (1958). A learning machine: part I. *IBM Journal of Research and Development*, 2(1):2-13.

Fukumoto, S. (1989). *Self-tuning process controller.* U.S. Patent No. 4814968.

Gent, I.P., Grant, S.A., MacIntyre, E., Prosser, P., Shaw, P., Smith, B.M., and Walsh, T. (1997). *How Not To Do It.* Research Report 97.27, School of Computer Studies, University of Leeds, Leeds, U.K.

Gerry, J.P. (1987). A comparison of PID control algorithms. *Control Engineering*, 34(3):102-105.

Gerry, J.P. and Shinskey, F.G. (2004). PID controller specification (white paper) [online]. Available from: <http://www.expertune.com/PIDspec.htm> [14 May 2004].

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, Massachusetts.

Goldberg, D.E. (1993). Making genetic algorithms fly: a lesson from Wright brothers. *Advanced Technology for Developers*, 2:1-8.

Goodwin, G. and Crisafulli, S. (1998). *Method and apparatus for automated tuning of PID controllers.* Patent No. WO9812611.

Gorez, R. (1997). A survey of PID auto-tuning methods. *Journal A*, 38(1):3-10.

Gorez, R. and Klàn, P. (2000). Nonmodel-based explicit design relations for PID controllers. In *Proceedings of the IFAC Workshop on Digital Control Past, Present and Future of PID Control*, Terrassa, Spain, pp. 133-140.

Gough Jr., W.A.G. and Lyon, D.L. (1994). *Method and apparatus for adaptive control.* U.S. Patent No. 5335164.

Greenberg, H.J. (1990). Computational testing: why, how and how much. *INFORMS Journal on Computing*, 2(1):94-97.

Grefenstette, J.J. (1999). Evolvability in dynamic fitness landscapes: a genetic algorithm approach. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Washington D.C., U.S.A., pp. 2031-2038.

Hägglund, T. and Åström, K.J. (1985). *Method and an apparatus in tuning a PID-regulator.* U.S. Patent No. 4549123.

Hägglund, T. and Åström, K.J. (1991). Industrial adaptive controllers based on frequency response techniques. *Automatica*, 27(4):599-609.

Hagiwara, K. (1998). *Method and device for PID tuning.* Japan Patent No. 10333704.

Hang, C.C. and Sin, K.K. (1991). A comparative performance study of PID auto-tuners. *IEEE Control Systems Magazine*, August, 11(5):41-47.

Hansen, M.P. and Jaszkiewicz, A. (1998). *Evaluating the Quality of Approximations to the Non-dominated Set.* Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Hansen, P.D. (1995a). *Self-tuning controller that extracts process model characteristics.* U.S. Patent No. 5394322.

Hansen, P.D. (1995b). *Self-tuning controller.* U.S. Patent No. 5406474.

Hansen, P.D. and Bristol, E.H. (1996). *Self-tuning controller.* U.S. Patent No. 5587896.

Hayashibe, S. (1984). *Apparatus for tuning PID controllers in process control systems.* U.S. Patent No. 4441151.

Hiroi, K. (1996). *Method and apparatus for controlling a process having a control loop using feedback control.* U.S. Patent No. 5535117.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Honeywell International Inc. (2004). http://www.acs.honeywell.com/ichome/rooms/displaypages/layoutinitial [online]. Available from: <http://www.acs.honeywell.com> [14 May 2004].

Hooker, J.N. (1995). Testing heuristics: we have it all wrong. *Journal of Heuristics*, 1(1):33-42.

Huang, C.-T. and Huang, M.-F. (1993). Estimation of the second-order parameters from the process transient by simple calculation. *Industrial & Engineering Chemistry Research*, 32(1):228-230.

Huang, C.-T. and Chou, C.-J. (1994). Estimation of the underdamped second-order parameters from the system transient. *Industrial & Engineering Chemistry Research*, 33(1):174-176.

IEE Digest (1996). *Getting the Best Out of PID in Machine Control*. Digest of IEE PG16 Colloquium (96/287), London, U.K.

Iino, Y. and Shigemasa, T. (1989). *Adaptive process control system*. U.S. Patent No. 4882526.

Ingénierie Pour SIgnaux et Systèmes (2004). Easy PID tuning [online]. Available from: <http://www.ipsis.com/produits/pidtuning/pidtuning.htm> [14 May 2004].

Innovention Industries Inc. (2004). Tune plus – PID tuning product [online]. Available from: <http://www.innovin.com/tuneplus.htm> [14 May 2004].

IPCOS (2004). IPCOS creators in control [online]. Available from: <http://www.ipcos.be/index.html> [14 May 2004].

ISE Inc. (2004). Commander software data sheet [online]. Available from: <http://instserv.com/commande.htm> [14 May 2004].

Israel Electric Corporation (2004). Optimal PID tuning through best process identification [online]. Available from: <http://www.geocities.com/ugf4aggnn/CSTPmain.htm> [14 May 2004].

JC Systems Inc. (2004). Untitled document [online]. Available from: <http://www.jcsystemsinc.com/html/tool_oview.html> [14 May 2004].

Jones, D.F., Mirrazavi, S.K., and Tamiz, M. (2002). Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1-9.

Junk, K.W. (2000). *Optimal auto-tuner for use in a process control network.* U.S. Patent No. 6128541.

Katayama, R. (1992). *Automatic tuning PID control device.* Japan Patent No. 4076702.

Katayama, R. and Kajitani, Y. (1994). *Control parameter tuning unit and a method of tuning parameter for a control unit.* U.S. Patent No. 5295061.

Kato, T., Yoshida, O., Nakamura, M., and Arita, S. (1997). *Adaptive PI control method.* U.S. Patent No. 5691615.

Kaya, A. and Scheib, T.J. (1988). Tuning of PID controls of different structures. *Control Engineering*, 35(7):62-65.

Kennedy, J. and Spears, W.M. (1998). Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, U.S.A., pp. 78-83.

Kim, J.K. (1997). *Method of gain control using puzzy technique.* Korea Patent No. 9705554.

Kim, Y.S. (1999). *Unit and method for determining gains of a PID controller using genetic algorithm.* U.S. Patent No. 5971579.

Knowles, J.D. and Corne, D.W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149-172.

Knowles, J.D. and Corne, D.W. (2002). On metrics for comparing non-dominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation*, Hawaii, U.S.A., pp. 711-716.

Kobari, M. and 2 others (1994). *Auto-tuning PID controller*. Japan Patent No. 6095702.

Kraus, T.W. (1986). *Pattern-recognizing self-tuning controller*. U.S. Patent No. 4602326.

Kraus, T.W. (1990). *Pattern-recognizing self-tuning controller*. U.S. Patent No. RE33267.

Kursawe, F. (1990). A variant of evolution strategies for vector optimization. In *Proceedings of the 1st International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, pp. 193-197.

Lambda Controls (2004). Lambda controls: a process control optimization company [online]. Available from: <http://www.lambdacontrols.com> [14 May 2004].

Lane, J.D. and Scheib, T.J. (1988). *Apparatus and method using adaptive gain scheduling algorithm*. U.S. Patent No. 4768143.

L'Ecuyer, P. (1996). Simulation of algorithms for performance analysis. *INFORMS Journal on Computing*, 8(1):16-20.

Li, Y., Ng, K.C., Murray-Smith, D.J., Gray, G.J., and Sharman, K.C. (1996). Genetic algorithm automated approach to the design of sliding mode control systems. *International Journal of Control*, 63(4):721-739.

Li, Y., Feng, W., Tan, K.C., Zhu, X.K., Guan, X., and Ang, K.H. (1998). PIDeasy™ and automated generation of optimal PID controllers. In *Proceedings of the 3rd Asia-Pacific Conference on Control and Measurement*, China Aviation Industry Press, Dunhuang, P.R. China, pp. 29-33.

Lipták, B.G. (2001). Technical information: tuning of processes [online]. Available from: <http://www.dardemann-power-station-services.com/Tuning support page.htm> [13 April 2004].

Lloyd, S.G. (1994). *Tuning arrangement for turning the control parameters of a controller.* U.S. Patent No. 5283729.

Lu, Z.J. (2001). *Controllers that determine optimal tuning parameters for use in process control systems and methods of operating the same.* U.S. Patent No. 6253113.

Luo, R.F., Qin, S., and Chen, D. (2000). *System and method for closed loop autotuning of PID controllers.* U.S. Patent No. 6081751.

Luyben, W.L. (2001). Effect of derivative algorithm and tuning selection on the PID control of dead-time processes. *Industrial & Engineering Chemistry Research*, 40(16):3605-3611.

Marsh, P. (1998). Turn on, tune in. *New Electronics*, 31(4):31-32.

Mathur, A.K. and Samad, T. (1997). *Closed loop neural network automatic tuner.* U.S. Patent No. 5625552.

Matrikon Inc. (2004). Advanced process control – ProcessACT at Matrikon.com [online]. Available from: <http://www.matrikon.com/products/processact/ezytune.asp> [14 May 2004].

McGeoch, C.C. (1996). Toward an experimental method for algorithm simulation. *INFORMS Journal on Computing*, 8(1):1-15.

McMillan, G.K. (1984). Control loop performance. In *Proceedings of the ISA'84 International Conference and Exhibition: Advances in Instrumentation*, Houston, U.S.A., pp. 589-603.

Mendenhall, W. and Beaver, R.J. (1994). *Introduction to Probability and Statistics.* 9th Edition, Duxbury Press, International Thomson Publishing.

Metso Automation Inc. (2004). Metso automation – high level automation solutions, systems and services [online]. Available from: <http://www.metsoautomation.com> [14 May 2004].

Michalewicz, Z., Deb, K., Schmidt, M., and Stidsen, T. (2000). Test-case generator for nonlinear continuous parameter optimization techniques. *IEEE Transactions on Evolutionary Computation*, 4(3):197-215.

Minter, B.J. and Fisher, D.G. (1988). A comparison of adaptive controllers: academic vs industrial. In *Proceedings of the 1988 American Control Conference*, Atlanta, U.S.A., pp. 1653-1658.

Mitsuo, K. (1995). *Automatic tuning system for PID parameter.* Japan Patent No. 7168604.

Miyagaki, H., Sugano, A., Takita, A., Toyama, E., Shimizu, K., Tobita, H., Matsumoto, H., Nomura, M., and Kimura, T. (1993). *Process controller for controlling a process to a target state.* U.S. Patent No. 5268835.

Moore, C.F., Smith, C.L., and Murrill, P.W. (1969). Simplifying digital control dynamics for controller tuning and hardware lag effects. *Instrument Practice*, 23:45-49.

Morari, M. and Zafiriou, E. (1989). *Robust Process Control.* Prentice Hall, New York.

Mori, Y. and Shigemasa, T. (1986). *Multivariable proportional-integral-derivative process control apparatus.* U.S. Patent No. 4563734.

Morrison, R.W. and De Jong, K.A. (1999). A test problem generator for non-stationary environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Washington D.C., U.S.A., pp.2047-2053.

Murate, T. and Suzuki, S. (1989). *Method and a process control system using the method for minimizing hunting.* U.S. Patent No. 4855674.

National Instruments (2004). LabVIEW PID control toolset for windows – products and services – national instruments [online]. Available from: <http://sine.ni.com/apps/we/nioc.vp?cid=1394&lang=US> [14 May 2004].

Nomoto, K., Kirimoto, T., and Kondo, M. (1989). *Auto-tuning controller using fuzzy reasoning to obtain optimum control parameters*. U.S. Patent No. 4864490.

Nomura, M., Saito, T., Matsumoto, H., Shimoda, M., Kondoh, M., Miyagaki, H., Sugano, A., and Yokokawa, N. (1994). *Process control method and system for performing control of a controlled system by use of a neural network.* U.S. Patent No. 5311421.

O'Dwyer, A. (2003). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.

OPC Foundation (2004). The OPC foundation – dedicated to interoperability in automation [online]. Available from: <http://www.opcfoundation.org> [14 May 2004].

Orlin, J.B. (1996). On experimental methods for algorithm simulation. *INFORMS Journal on Computing*, 8(1):21-23.

Otani, T. (1991). *Process controller.* Japan Patent No. 3265902.

Pemberton, T.J. (1973). *Pulse test of digital control system.* U.S. Patent No. 3727035.

Pemberton, T.J. (1974). *Simplified procedure for tuning PID controllers*. U.S. Patent No. 3826887.

Peng, Y., Vrancic, D., and Hanus, R. (1996). Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. *IEEE Control Systems Magazine*, 16(4):48-57.

Pettit, J.W. and Carr, D.M. (1987). *Self-tuning controller.* U.S. Patent No. 4669040.

Pfeiffer, B.-M. (2000). Towards 'plug and control': self-tuning temperature controller for PLC. *International Journal of Adaptive Control and Signal Processing*, 14(5):519-532.

Plant Automation Services Inc. (2004). PAS home page [online]. Available from: <http://www.tunewizard.com> [14 May 2004].

Poloni, C. (1995). Hybrid GA for multi-objective aerodynamic shape optimisation. In: G. Winter *et al.* (Eds), *Genetic Algorithms in Engineering and Computer Science*, Chichester:Wiley, pp. 397-415.

Prechelt, L. (1996). A quantitative study of experimental evaluations of neural network algorithms: current research practice. *Neural Networks*, 9(3):457-462.

PROVOX (2004). PROVOX | product data | control [online]. Available from: <http://www.provox-secure.com/product/control.asp> [14 May 2004].

Quevedo, J. and Escobet, T., editors (2000). *Digital Control: Past, Present and Future of PID Control.* Proceedings of the IFAC Workshop, Terrassa, Spain.

Raczynski, S. (2004). PID controller settings [online]. Available from: <http://www.raczynski.com/pn/pids.htm> [14 May 2004].

Rockwell Automation (2004). RSTune – overview [online]. Available from: <http://www.software.rockwell.com/rstune> [14 May 2004].

Rose, R.E. (1999). *Method and apparatus for automatically tuning the parameters of a feedback control system.* U.S. Patent No. 5974434.

Rutledge, R.W. (1982). *Process control.* U.S. Patent No. 4346433.

Saito, T., Kawakami, J., Takahashi, S., Suehiro, T., Matsumoto, H., and Tachibana, K. (1990). *PID controller system.* U.S. Patent No.4903192.

Saito, T., Takahashi, S., Tachibana, K., Kawakami, J., and Yokokawa, N. (1992). *Self-tuning controller apparatus and process control system.* U.S. Patent No. 5153807.

Sakai, Y., Nakai, Y., Miyabe, A., and Kawano, T. (1989). *Self-tuning controller.* U.S. Patent No. 4881160.

Samad, T. (1998). *Nonlinear-approximator-based automatic tuner.* U.S. Patent No. 5847952.

Sarker, R., Mohammadian, M., and Yao, X., editors (2002). *Evolutionary Optimization.* Kluwer Academic Publishers, New York.

Sarker, R. and Coello Coello, C.A. (2002). Assessment methodologies for multiobjective evolutionary algorithms. In: R. Sarker *et al.* (Eds), *Evolutionary Optimization*, Kluwer Academic Publishers, New York, pp. 177-195.

Schaffer, J.D. (1987). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications*, New Jersey, U.S.A., pp. 93-100.

Schmidt, M. and Michalewicz, Z. (2000). Test-case generator TCG-2 for nonlinear parameter optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation*, San Diego, U.S.A., pp. 728-735.

Schott, J.R. (1995). *Fault Tolerant Design using Single and Multicriteria Genetic Algorithm Optimization.* Masters Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Schwefel, H.-P. (1988). Evolutionary learning optimum-seeking on parallel computer architectures. In *Proceedings of the International Symposium on Systems Analysis and Simulation 1988, I: Theory and Foundations*, Akademie-Verlag, Berlin, pp. 217-225.

Seem, J.E. and Haugstad, H.J. (1994). *Pattern recognition adaptive controller.* U.S. Patent No. 5355305.

Seem, J.E. and Decious, G.M. (1996). *Fast automatic tuning of a feedback controller.* U.S. Patent No. 5568377.

Shier, D.R. (1996). On algorithm analysis. *INFORMS Journal on Computing*, 8(1):24-26.

Shigemasa, T. (1984). *Process control apparatus*. U.S. Patent No. 4451878.

Shigemasa, T. and Takagi, Y. (1984). *Improved proportional integral-derivative control apparatus*. U.S. Patent No. 4466054.

Shigemasa, T. and Ichikawa, Y. (1985). *Digital PID process control apparatus*. U.S. Patent No. 4539633.

Shinskey, F.G. (1994). *Feedback Controllers for the Process Industries*. McGraw-Hill Inc., New York.

Siemens (2004). Siemens – SIMATIC technology – PID control [online]. Available from: <http://www.ad.siemens.de/simatic/regelsysteme/html_76/produkte/sb-pidcontr2.htm> [14 May 2004].

Sklaroff, M. (1992). *Adaptive controller in a process control system and a method therefor*. U.S. Patent No. 5170341.

SpecializedControl (2002). PID control tuning – analytical PID tuning methods for feedback control systems [online]. Available from: <http://www.specializedcontrol.com> [30 June 2002].

Straight-Line Control Co. Inc. (2004). Controller tuning 101 [online]. Available from: <http://members.aol.com/pidcontrol/software.html> [14 May 2004].

Suzuki, S. (1988). *Method of determining PID parameters and an automatic tuning controller using the method*. U.S. Patent No. 4754391.

Svarovsky, J.E., Baechtel, D.F., and Day, C.P. (1993). *Automatic tuning apparatus for PID controllers*. U.S. Patent No. 5223778.

Tadokoro, H. and 3 others (1992). *PID parameter automatic tuning method*. Japan Patent No. 4346102.

Tadokoro, H. and 2 others (1993). *Method for automatically tuning PID parameter.* Japan Patent No. 5073104.

Takahashi, S., Tachibana, K., and Saito, T. (1991). *Method and apparatus of automatically setting PID constants.* U.S. Patent No. 5043862.

Takatsu, H. and Kitano, K. (1992). *Process control device.* U.S. Patent No. 5166873.

Takeguchi, Y. (1999). *PID controller with automatic tuning function.* Japan Patent No. 11161301.

Takigawa, T. and 1 other (1984a). *Auto-tuning system for parameter of PID adjustor.* Japan Patent No. 59069807.

Takigawa, T. and 1 other (1984b). *Auto-tuning system of parameter of PID adjustor.* Japan Patent No. 59153202.

Tan, K.C. (1997). *Evolutionary Methods for Modelling and Control of Linear and Nonlinear Systems.* PhD Thesis, Department of Electronics and Electrical Engineering, University of Glasgow, U.K.

Tan, K.C. and Li, Y. (1997). Evolutionary system identification in the time domain. *Journal of Systems and Control Engineering – Proceedings of the Institution of Mechanical Engineers (Part I)*, 211(5):319-323.

Tan, K.C., Lee, T.H., and Khor, E.F. (2001a). Incrementing multi-objective evolutionary algorithms: performance studies and comparisons. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, Volume 1993 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 111-125.

Tan, K.C., Lee, T.H., and Khor, E.F. (2001b). Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5(6):565-588.

Techmation Inc. (2004). Techmation [online]. Available from: <http://protuner.com> [14 May 2004].

Teng, D. and Wang, S.P. (1997). *Auto-tuning controller and method of use therefor.* U.S. Patent No. 5649062.

Teng, D. and Wang, S.P. (1998). *Cascade tuning controller and method of use therefor.* U.S. Patent No. 5805447.

TiPS Inc. (2002). TiPSWEB tune-a-fish page [online]. Available from <http://www.tipsweb.com/tuneafish.htm> [1 October 2002].

Ueda, T., Nagami, Y., Saito, T., and Zhang, Z.M. (1994). *PID control unit.* U.S. Patent No. 5331541.

Van Veldhuizen, D.A. and Lamont, G.B. (1998). *Multiobjective Evolutionary Algorithm Research: A History and Analysis.* Technical Report TR-98-03, Dept. of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson, AFB, Ohio.

Van Veldhuizen, D.A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* PhD Thesis, Dept. of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Versteeg, H.J., Jansma, H.J., and Turner, K. (1986). Evaluation of commercially available adaptive controllers. *Journal A*, 27(3):120-126.

Viennet, R., Fontiex, C., and Marc, I. (1996). Multicriteria optimization using a genetic algorithm for determining a Pareto set. *International Journal of Systems Science*, 27(2):255-260.

Visioli, A. (2003). Time-optimal plug&control for integrating and FOPDT processes. *Journal of Process Control*, 13(3):195-202.

Visual Solutions Inc. (2004). Visual solutions, inc. – VisSim, VisSim/Comm & VisSim/DSP home page [online]. Available from: <http://www.vissim.com/products/addons/optimize.html> [14 May 2004].

Wang, L., Barnes, T.J.D., and Cluett, W.R. (1995). New frequency-domain design method for PID controllers. *IEE Proceedings D – Control Theory and Applications*, 142(4):265-271.

Wang, Q.G. and Hang, C.C. (2000). *Frequency-domain adaptive controller.* U.S. Patent No. 6076951.

Wang, Q.G., Hang, C.C., and Bi, Q. (2002). *Apparatus for relay based multiple point process frequency response estimation and control tuning.* U.S. Patent No. 6438431.

Wang, T.S. and Clements, W.C. (1995). Adaptive multivariable PID control of a distillation column with unknown and varying dead time. *Chemical Engineering Communications*, 132:1-13.

Weinzierl, K. (2002). *Method for generating control parameters from a response signal of a controlled system and system for adaptive setting of a PID controller.* U.S. Patent No. 6353766.

Westinghouse Process Control (2004). Westinghouse process control: products: ovation: descriptions: tuning tools [online]. Available from: <http://www.westinghousepc.com/ovation/details-tuningtools.cfm> [14 May 2004].

White III, C.A. (1992). *Self-learning memory unit for process controller and self-updating function generator.* U.S. Patent No. 5126933.

Wilkie, J., Johnson, M., and Katebi, R. (2002). *Control Engineering: An Introductory Course.* Palgrave, New York.

Wilson, C. and Callen C. (2004). Close process control translates to quality heat treated parts [online]. Available from: <http://www.industrialheating.com/CDA/ArticleInformation/coverstory/BNPCoverStoryItem/0,2830,116383,00.html> [14 May 2004].

Wolpert, D.H. and Macready, W.G. (1995). *No Free Lunch Theorems for Search.* Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wolpert, D.H. and Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67-82.

Wojsznis, W.K. and Blevins, T.L. (1995). *System and method for automatically tuning a process controller*. U.S. Patent No. 5453925.

Wojsznis, W.K. and Blevins, T.L. (2001). *Adaptive feedback/feedforward PID controller*. Patent No. WO0198845.

Xiera Technologies Inc. (2004). GeneX [online]. Available from: <http://www.xiera.com/genex.htm> [14 May 2004].

Xue, D. (2004). MATLAB central file exchange – CtrlLAB [online]. Available from: <http://www.mathworks.com/matlabcentral/fileexchange/index.jsp> [14 May 2004].

Yamamoto, S. (1991). *Adaptive controller*. Japan Patent No. 3118606.

Yu, C.C. (1998). *Use of saturation relay feedback in PID controller tuning*. U.S. Patent No. 5742503.

Yu, J.J., Jo, H.J., Lee, C.K., and Park, J.M. (1994). *Tuning method of PID controller*. Korea Patent No. 9407530.

Ziegler, J.G and Nichols, N.B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, vol. 64, pp. 759-768.

Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms – a comparative study. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, pp. 292-301.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257-271.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2):173-195.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., and Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117-132.

Zou, H.H., Hedstrom, K.P., Warrior, J., and Hays, C.L. (1997). *Field based process control system with auto-tuning*. U.S. Patent No. 5691896.

Zou, H.H. and Brigham, S.E. (1998). *Process control system with asymptotic auto-tuning*. U.S. Patent No. 5818714.