TRAJECTORY-SCHEDULING CONTROL SYSTEMS AND THEIR MULTI-OBJECTIVE DESIGN AUTOMATION

A DISSERTATION SUBMITTED TO

THE DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

OF UNIVERSITY OF GLASGOW

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Gregory Chow Ye Chong

January 2006

PAGE

NUMBERING

AS ORIGINAL

Abstract

This research develops the concept of a Trajectory-Scheduling Network (TSN). A TSN system comprises nodes of linear subsystems placed along the operating trajectory of a nonlinear plant to be controlled. The TSN is to enable linear time-invariant system analysis and design methods to be extended to nonlinear systems for their entire operating envelope or setpoint trajectory without needing linearisation. The thesis encompasses the analysis of TSN systems and their multi-objective design methods.

TSN nodes are networked through interpolation and activation, similar to a gain-scheduling or local model/controller network. However, to achieve accuracy and ease of commissioning without requiring a large number of nodes, an algorithm has been developed first to identify optimum transition nodes within the entire operating envelope. Then the TSN approaches a nonlinear plant globally, not just locally, without requiring linearisation. If desired or necessary, global optimisation provides an enhancement in the design process for TSNs. Since optimising only one aspect (a single objective) of performance while compromising others is undesirable, multi-objective designs have been developed concurrently to deliver or improve multiple aspects of performance.

Following the development of a TSN, it is applied to nonlinear system modelling, and this TSN is termed a Trajectory-Scheduling Model (TSM). A TSM possesses the same properties and design features as the TSN generic framework. A nonlinear system, a coupled liquid-tank, is used to examine this modelling technique. Results verify the feasibility and effectiveness of the methods developed and validates the TSM.

Further, the TSN technique is applied to nonlinear controller design, by way of a Trajectory-Scheduling Controller (TSC) network. It is illustrated through the design of a networked, easy-to-understand and easy-to-use PID control system for the coupled liquid-tank. Results show that the methods developed offer a high-performance linear control system with nonlinear capabilities to handle practical systems operating in a broad range and to cope with conflict between setpoint following at transient and disturbance rejection at steady state. This method is then applied to the PID network design problems for two nonlinear chemical processes.

As a TSN can be interpreted as a linear system, with respect to individual nodes, the controllers can thus be analysed using existing classical or conventional methods. For example, frequency-domain based stability margin and robustness analysis methods can be applied to determine the stability of each node and to assess the robustness of the overall network. Such examples are shown in this thesis. It is believed that the methodologies developed here provide a novel tool for nonlinear plant modelling and control system design, which is effective for a wide setpoint range and the entire operating envelope, hence ease nonlinear control system analysis and design problems by linear means.

List of Publications

Journals

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*, 1(1), 83-95.

Ang, K.H., Chong, G., and Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4), 559-576.

Li, Y., Ang, K.H., Chong, G. (2006). Patents, Software, and Hardware for PID Control - An overview and analysis of the current art. *IEEE Control Systems Magazine*, 26(1), 2006. (Invited paper, currently in press).

Li, Y., Ang, K.H., Chong, G. (2006). PID Control System Analysis and Design - Problems, remedies, and a way forward. *IEEE Control Systems Magazine*, 26(1), 2006. (Invited paper, currently in press).

Book Chapter

Ang, K.H., Chong, G., and Li, Y. (2004). Visualization technique for analyzing non-dominant Pareto optimality. In: Tan, K.C. *et al.* (Eds), *Recent Advances in Simulated Evolution and Learning*, World Scientific Series on Advances in Natural Computation, 2(18), 327-345.

Conference Proceedings

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*, Lecture Notes in Computing Science, Springer-Verlag, Berlin, 1803, 139-146.

Chong, G. and Li, Y. (2000). Evolving Trajectory Controller Networks from Linear Approximation Model network. In *Proceeding of the 2000 Congress on Evolutionary Computation*, San Diego, USA, 1, 251-255.

Chong, G. and Li, Y. (2001). Soft computing based networking technique and its application to control system design and identification. In *Proceedings of the Joint 9th IFSA Congress and 20th NAFIPS Conference*, Vancouver, Canada, 2734-2739.

Feng, W., Li, Y., and Chong, G. (2001). Tractable neurocontroller design and application to ship control with actuator limits . In *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, 1282-1287.

Chong, G. and Li, Y. (2002). Multi-objective design of trajectory networks controller for nonlinear plants and stability analysis. In *Proceedings of the 5th Asia-Pacific Conference on Control and Measurement*, Dali, P.R. China, 185-190.

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 Simulation Evolution and Learning*, Singapore, 840-844.

Chong, G. and Li, Y. (2002). Nonlinear system modelling and control along the operating locus. In *Proceedings of the 8th annual Conference of Chinese Automation and Computing Society in the UK*, U.K., 59.

Ang, K.H., Chong, G., and Li, Y. (2002). 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, 2, 1139-1144.

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

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

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., 41-46.

Li, Y., Chong, G. and Ang, K.H. (2005). Trajectory-Scheduled Controller Networks and Application in Nonlinear Process Control. In *Proceedings of the annual Conference of Chinese Automation and Computing Society in the UK*, Sheffield U.K., 10 September.

Table of Contents

Abstract i

Acknowledgements iii

List of Publications iv

Table of Contents vii

List of figures x

List of Tables xiv

Nomenclature xv

Chapter 1 Introduction 1

1.1 Motivation 2

1.2 Approach 3

1.3 Contributions 4

1.4 Outline of the Thesis 6

Chapter 2 Networking Methods for Nonlinear Control System Modelling and Design 8

2.1 Multiple Linear Networks for Nonlinear Control 9

2.2 Gain-Scheduling Network 9

2.3 Local Model Network 10

2.4 Local Controller Network 11

2.5 Linear Approximation Model 13

2.6 Framework of Networking 14

2.6.1 Structure Networking 14

2.6.2 Parameter Networking 17

2.7 Activation Functions 19

2.8 Activation Variables 24

2.8.1 Input and Setpoint Elements in the Activation Variables 25

2.8.2 Output and Feedback Elements in the Activation Variables 26
2.8.3 Additional Elements in Activation Variables 27
2.9 Operating Nodes 28
2.10 Conclusion 29

Chapter 3 Trajectory-Scheduling Networks Suitable for a Range of Setpoints and Operating Envelopes 31

3.1 Multiple Setpoints and TSN Nodes 32
3.1.1 Formulation 33
3.1.2 Illustration 36
3.2 TSN Construction 42
3.2.1 Interpolation 44
3.2.2 Hybrid Activation Variables 46
3.3 Summary 47

Chapter 4 Automating TSN Construction with Multiple Objectives through Evolutionary Computation 49

4.1 Control System Design Objectives 50
4.2 Multi-Objective Optimisation 52
4.2.1 Pareto Optimality 52
4.2.2 Goal and Priority 53
4.3 Evolutionary Search Algorithms 53
4.4 MATLAB Based Design Automation Framework for TSN 57

Chapter 5 TSN for Nonlinear System Modelling 59

5.1 System Modelling 60
5.2 Trajectory-Scheduling Models 61
5.3 Nonlinear Coupled System Modelling (Case 1) 64
5.3.1 System Identification with TSM 66
5.3.2 Validation and Analysis 68
5.4 Nonparametric System Modelling 70
5.5 Discussions 75

Chapter 6 TSN for Nonlinear Control System Design 76

6.1 PID Controllers Feasible as TSC Nodes 77

6.2 TSC Construction 79

6.3 TSC Design Methods 83

6.3.1 Generating Controller Nodes Using a CACSD Package 83

6.3.2 Deriving Trajectory Controllers from TSM 84

6.3.3 Evolving TSC from Multi-Setpoint Response Data 86

6.3.4 TSC through Evolution 89

6.4 TSC Network Design for a Nonlinear Chemical Process (Case 2) 92

6.4.1 Controller Design 94

6.4.2 Results 96

6.5 TSC for a Polymerisation Reaction Process (Case 3) 97

6.6 Multi-Objective Design 101

6.7 Guided Search 105

6.7.1 Multi-Objective Decision Making 109

6.7.2 Evolving High-Performance TSC Systems Using MOEA 111

6.8 LabVIEW Based TSC 114

6.9 Summary 115

Chapter 7 Analysis of Trajectory-Scheduling Networks 117

7.1 Linear representation of the TSN Systems 118

7.1.1 Analysis TSM-TSC with Classical Methods 119

7.1.2 TSC Activation and System Stability 123

Chapter 8 Conclusions and Further Work 125

8.1 Conclusion 126

8.2 Future Perspective 127

References 128

List of figures

Figure 2.1 Local Controller Network....12
Figure 2.2 Interpolated LAM network....14
Figure 2.3 The multiple sub-task approach....15
Figure 2.4 Parameter-varying Local Model Network in a state-space representation....19
Figure 2.5 Normalised Cubic spline interpolation....21
Figure 2.6 Normalised Cubic interpolation....21
Figure 2.7 Normalised Piecewise cubic Hermite interpolation....22
Figure 2.8 Normalised Linear interpolation....22
Figure 2.9 Normalised Linear extrapolation....23
Figure 2.10 Normalised cubic Hermite extrapolation....23
Figure 2.11 Input or setpoint activation....26
Figure 2.12 Output/feedback activation....27
Figure 3.1 Algorithm for finding operating nodes (from Case 2)....33
Figure 3.2 Smooth parameters....35
Figure 3.3 Diversify parameters....35
Figure 3.4 Trajectory with 3 nodes....37
Figure 3.5 Trajectory with 5 nodes....37
Figure 3.6 Trajectory with 4 nodes....38
Figure 3.7 Trajectory with 5 nodes....39
Figure 3.8 Trajectory with 6 nodes....39
Figure 3.9 Trajectory with 8 nodes....40
Figure 3.10 normalize tolerance of 0.1. Five nodes found....41
Figure 3.11 normalize tolerance of 0.01. Twelve nodes found....41
Figure 3.12 Parameterisation of TSN....42
Figure 3.13 Trajectory-Scheduling Network....43
Figure 3.14 A simple linear interpolation scheduling networking for multiple linear systems....44
Figure 3.15 Hybrid activation (intermediate)....46
Figure 3.16 Hybrid activation (boost)....47
Figure 4.1 A feedback control system with a model-following command....50

Figure 4.2 Computer-automated design by artificial evolution 54
Figure 4.3 Exhaustive search of PID controller 55
Figure 4.4 Close loop response of plant from EA search 56
Figure 4.5 MATLAB TSC-TSM design and analysis package. 58
Figure 5.1 Block diagram of a single input single output system 60
Figure 5.2 General ABCD block diagram 62
Figure 5.3 Trajectory-Scheduling Model 63
Figure 5.4 Evolving a TSM network from plant step response data 64
Figure 5.5 Nonlinear twin-tank couple system 65
Figure 5.6 The nonlinearity of the twin tank system presented as 3D open loop step response 66
Figure 5.7 The static model and operating nodes of the twin-tank system 67
Figure 5.8 Linear interpolation of the TSM Twin-tank system 68
Figure 5.9 Validation of TSM network (step response) 69
Figure 5.10 Validation of TSM network (continuous response) 69
Figure 5.11 Regeneration of plant response with and without noise using convolution methods 72
Figure 5.12 Step response of TSM, convolution, linear and spline 74
Figure 6.1 PID Structure – Ideal Form 78
Figure 6.2 PID Structure – Parallel Form 78
Figure 6.3 PID Structure – Series Form 79
Figure 6.4 Activation region of Linear Model Network and Local Model Network 80
Figure 6.5 Trajectory-Scheduling Controller 82
Figure 6.6 Trajectory-Scheduling PID Controller. 83
Figure 6.7 Direct design from plant response using PIDeasy™ 84
Figure 6.8 Closed loop responses of the TSC at operating points including the unseen ones at 0.075m and 0.125m. 85
Figure 6.9 Closed loop responses of the TSC at unseen operating points. 85
Figure 6.10 Closed loop time responses of TSC via convolution method. 86
Figure 6.11 Closed loop control signal of TSC via convolution method. 87
Figure 6.12 Closed loop responses of speed control (model free designed) 88
Figure 6.13 Closed loop responses of speed control (IMC designed) 88

Figure 6.14 Objective function for optimising the controller at one reference point. 89
Figure 6.15 Evaluation points in the operating envelope. 90
Figure 6.16 Closed loop responses of the TSC at operating points including the unseen ones at 0.075m and 0.125m. 91
Figure 6.17 Closed loop responses of the TSC at unseen operating points. 91
Figure 6.18 Performance of TSC, where disturbance occur at t=200 and 700 s. 92
Figure 6.19 The equilibrium manifold of the nonlinear process within its operating envelope 93
Figure 6.20 Open-loop tests of the nonlinear process within its operating envelope 94
Figure 6.21 Weighting function of TSC 95
Figure 6.22 Performance of the TSC regulator subject to a 20% load disturbance 96
Figure 6.23 Performance reliability of the PID network in the entire operating envelope 97
Figure 6.24 Open-loop step response from (0.01-0.1) at equal-interval of 0.01 (kg h-1) 98
Figure 6.25 Operating range x2=[2-10] (kg cm-2); u1=[0.0235-0.1269] (kg h-1)..........99
Figure 6.26 Close-loop response. $r(t)=5kg\ cm^{-2}$ with $[x_1,x_2]=[0,0]$ initial condition. 100
Figure 6.27 Close-loop response. $r(t)=5kg\ cm^{-2}$ with $[x_1,x_2]=[0,0]$ initial condition, and 20% load disturbance 100
Figure 6.28 Good transient but chattering control signal. 101
Figure 6.29 Suppressing the control signal 102
Figure 6.30 Pareto optimal solutions divided by a conventional design as a reference point 103
Figure 6.31 Performance of all TSC on Pareto front 104
Figure 6.32 Performance of TSC fall in region with both objectives optimised 105
Figure 6.33 Pareto optimal solutions divided by a conventional design as a reference point 106
Figure 6.34 Step response of individual solution fall in the region with both improvements compared to IMC tuning 107
Figure 6.35 Step response in region where only ISE minimise 108
Figure 6.36 Step response in region where only Control Energy was reduce. 109
Figure 6.37 Additional Z-N tuned PID as reference point. 110

Figure 6.38 Step response of good region MOEA solution 111
Figure 6.39 2D and 3D view of non-dominated solutions for 3 objectives MOEA and selection region reference to IMC. 112
Figure 6.40 Normalised and sorted view of *non-dominated* sets from MOEA and selection region reference to IMC 113
Figure 6.41 Step response from selected solution (with load disturbance). 114
Figure 6.42 LabVIEW online TSC controller interface 115
Figure 7.1 Static system during transition 118
Figure 7.2 Step response of the TSM network 119
Figure 7.3 Bode plot of the TSM network 120
Figure 7.4 Root locus of the TSC-TSM network 120
Figure 7.5 Classical analysis methods applied to TSC-TSM system (linear interpolation) 121
Figure 7.6 Classical analysis methods applied to TSC-TSM system (Piecewise Cubic Hermite interpolation) 122
Figure 7.7 Classical analysis methods applied to TSC-TSM system (Spline interpolation) 122
Figure 7.8 Overall performance of the TSC-TSM system 123
Figure 7.9 Individual controller signals and their activation according to step change. 124

List of Tables

Table 5.1: The coefficients of the twin-tank 65
Table 6.1: PID controller parameters by local tuning....84
Table 6.2: PID controller parameters by global tuning....85
Table 6.3: PID controller parameters search through evolution simultaneously. 90

Nomenclature

GA	Genetic Algorithm
GP	Genetic Programming
EA	Evolutionary Algorithm
IMC	Internal Model Control
LAM	Linear Approximation Model
LCN	Local Controller Network
LMN	Local Model Network
LPV	Linear Parameter-Variant
LTI	Linear Time Invariant
MOEA	Multi-Objective Evolutionary Algorithm
TSC	Trajectory-Scheduling Controller
TSM	Trajectory-Scheduling Model
TSN	Trajectory-Scheduling Network

Chapter 1

Introduction

1.1 Motivation

Most real-world systems encountered in control engineering are nonlinear (Shorten and Narendra 1998), but control methods practised on these systems are mostly linear (Levine 1996). This fact and hence the need for linear control techniques that are applicable to nonlinear control problems have motivated this research.

There are many ways to tackle a nonlinear process. The use of multiple models to control nonlinear systems has a log history beginning with gain scheduling (Shamma and Athans 1990). The more recent ideas are Neural Networks (Jacobs and Jordan 1993; Li and Haeussler 1996; Murray-Smith *et al.* 1992), Local Model Networks (Johansen and Foss 1992a), adaptive gain scheduling (Narendra and George 2002) and Fuzzy Logic (Takagi and Sugeno 1985). Other approaches like linear and nonlinear Auto Regressive Moving Average with eXogenous inputs (ARMAX) type models are also widely used to model dynamic systems (Åström and Bohlin 1965; Johansen and Foss 1992b).

The capability of the methods mention can be readily exploited in this context. However, the representation achieved by these methods is obscure (black-box). There are no simple ways to analyse the properties of the model achieved by them. In other words, there is no way to analyse the performance except by trial and error. While the knowledge-driven (clear-box) method are derived from physicochemical laws, but its case-by-case characteristics can make its application difficult and time consuming for industrially relevant control systems.

Recently, grey-box approaches have been developed, which combine the clear-box and black-box (Gawthrop *et al.* 1993; Tan *et al.* 1997). Using the knowledge-driven method (Wang *et al.* 1997), the model can have a relatively simple structure; using the data-driven method, its parameters can be estimated and its learning ability can be improved. Although this kind of approach appears suitable for the control of complex industrial processes, it still requires a systematic effort to select, organize, and coordinate many different methods into one efficient, practical framework for process control.

Therefore, there is an urgent need to develop a widely applicable modelling technique that can provide advantages of both knowledge-driven and data-driven methods. There is also an urgent need to develop a corresponding control system design method for such a nonlinear process, so that a control system can be designed for the entire operating envelope of the process. Finally, it is essential that the system developed can be analysed.

1.2 Approach

Assisting modelling and design of control systems, intelligent methods such as rule-based expert systems (Skeirik 1990), multivariate statistical partial least squares (Chen *et al.* 1998), and genetic algorithms (Goldberg 1989) have been used and found very powerful in control engineering applications. The aim of the work reported in this thesis is hence to explore the most recent development in these techniques for solving problems discussed in Section 1.1.

In particular, neural network and fuzzy system techniques (Woosoon Yim and Singh 1995) have shown their arbitrary functional approximation capability in a wide range of nonlinear dynamic system modelling and control system design applications. However, the networks are in the form of a "black-box" representation, in which a model developed can be difficult to interpret in terms of physical meaning to practising engineers. Such a model can also be complex due to the excessive number of network weights and other parameters. This erects barriers to the application of conventional analysis and control theories. Therefore, to overcome the complexity of the existing networks, a much simpler network based on the 'Linear Approximation Model' (LAM) (Li and Tan 2000) is developed.

Extended in this thesis as a primary step in developing nonlinear control systems using existing linear techniques, hence, the LAM system identification and modelling technique is extended to a formal 'Trajectory-Scheduling Model' (TSM).

Based on LAM, a much more comprehensive network is developed (Chong and Li 2000a). This is termed 'Trajectory-Scheduling Network' (TSN) (Chong and Li 2002a; Chong and Li 2000b). TSN is a network that could be used in modelling and control, capable of tackling the nonlinear control problems. TSN is relatively simple and is formed with a set of conventional linear elements. It is a strategy of utilising linear synthesis technique for solving nonlinear problems. Each elements of TSN can be of a simple form, such as state-space models, proportional plus integral plus derivative (PID), transfer function or transfer function matrix based. TSN may then satisfy design specifications in the full operating range of a nonlinear system, unlike a linear one, which can only satisfy part of it. Such a nonlinear controller network applied the TSN technique is term 'Trajectory-Scheduling Controller' (TSC).

Finally, the combined TSM-TSC network can be interpreted in a classical way. Therefore, it can be analysed easily using conventional methods locally and globally. It is clear that such a network can overcome many control problems and satisfies modern-day requirements. This thesis will detail the development of this technique.

1.3 Contributions

The main contributions of the thesis are:

- Critical review and thus extension of performance metrics for both single and multiple objective optimisation and search algorithms. The current visualisation technique for assessing non-dominant solution sets is limited to two objectives. This has been extended to higher dimensional data, with a novel visualisation technique developed for identifying and presenting the best Evolutionary Algorithms (EA) solutions.

- Critical review and thus solution to complexity and analysis problems in existing multiple model networking techniques.
- Extension of the seminal LAM technique to a complete TSM technique, and application of evolutionary computation to derive TSMs automatically. A TSM eliminates the need of linearising and prior derivation of a first-principle's model.
- Development of a novel operating-point-scheduling technique for the entire operating envelope of a given process. This is coupled with a node acquiring technique based on the process nonlinearity, reducing the number of networking parameters and increasing application speed for TSM and TSC.
- Development of a flexible and straightforwardly applicable TSC, and also methods of obtaining TSC from existing control designs or CAD software, as well as through multi-objective evolutionary search techniques. These methods allow *a priori* knowledge to be incorporated in the design if desired. The results are presented with multi-objective visualisation, which enables the user to supervise (if desired) the final design for performance that meets his/her requirements.
- Development of a linear parameter-variant technique and tool to enable analysis of a combined TSM and TSC network using classical stability analysis methods.
- Development of a MATLAB based GUI software tool automatically to build a TSM for a nonlinear process, to evolve the corresponding TSC, and to analyse the close-loop system. This also makes it easy to implement in real-time with LabVIEW and to test against real physical systems.

1.4 Outline of the Thesis

This thesis consists of eight chapters.

Chapter 1 presents the motivation and approach to carry out this research and the thesis contributions. This chapter also presenting the background aspects of the modelling and control problems.

Chapter 2 begins with background on networking techniques used in control systems and their distinctions. An overview of networking techniques, which are based on operating region decomposition, is given. Aspects of the structure selection are discussed and problems in control systems design are presented.

Chapter 3 presents the TSN structure and the algorithm to search the optimum transition nodes. The significant of finding appropriate transition nodes are discussed. A step-by step construction of the TSN network is illustrated, and finally a novel hybrid activation technique is proposed.

Chapter 4 analyses the proposed evolutionary technique and the tools developed to construct the proposed design. The main advantages on multi-objective over single objective evolutionary algorithm are discussed, and the tools developed for carry out the search are shown.

Chapter 5 presents detailed modelling techniques used for nonlinear systems. This chapter contain the full illustration on using TSM in modelling a plant from various aspects, including the modelling without using any mathematical model but solely on plant data only.

Chapter 6 presents network controller design methods for nonlinear systems using the proposed networking technique and its validation. The benefits of using multi-objective design for TSC are discussed. A unique visualisation technique is proposed in multi-objective designed of the TSC. Finally, the implementation of TSC on LABVIEW real-time hardware for online experiments.

Chapter 7 discusses stability analysis of controller networks using classical methods. In this chapter, a method of deriving TSM-TSC network into linear model and the verification for the entire operating trajectory are demonstrated.

Chapter 8 draws conclusions and highlights future work.

Chapter 2

Networking Methods for Nonlinear Control System Modelling and Design

2.1 Multiple Linear Networks for Nonlinear Control

Control theory is well developed and understood for linear systems. Since one knows how to design a controller from a linear model, the multiple linear modelling and control can be easily applied to nonlinear systems. More importantly, the analysis of the properties of the representation achieved by these networks can be easily carried out since the linear control theory can be used to analyse locally the properties of each local model or controller. The main advantage of multiple linear model network approach is that they are capable of generalisation, the computation can be performed in parallel and *a priori* knowledge can be incorporated in their architecture. These significant advantages reveal the powerful potential of these multiple linear model approaches for the control of nonlinear systems. The complexity of a plant arises from at least three different sources. First, the plant dynamics involving disturbances, transport delays and non-minimum phase zeroes in the process. Secondly, the number of inputs and outputs – this is generally described in Multi-Input Multi-Output (MIMO) systems. Finally, the nonlinearity of the plant and actuator saturation.

In this thesis, the emphasis is on the development of a TSC capable of handling problems arising from plant nonlinearity using intelligent multi-objective search technique. Therefore, only Single-Input Single-Output (SISO) nonlinear plants having simple dynamics or complex plants to a certain extent will be considered.

2.2 Gain-Scheduling Network

The basic idea of the gain scheduling (Levine 1996) approach is to design different (linear) controllers for different operating conditions of the plant. When in operation, the parameters of these controllers are interpolated by a scheduler, resulting in a time-varying control scheme. The main advantage of this technique is that a relatively simple standard design technique (such as PID control) can be used to compose a nonlinear controller.

Although gain scheduling has been proven to work successfully in many applications (Leith *et al.* 2000; Wada and Osuka 1997), it is difficult to obtain analytical results for stability and robustness of this control approach. The main problem is that the influence of the scheduler has to be taken into account. It can be neglected if the scheduling variable varies slowly in which case stability can easily be proven (Shamma and Athans 1990). Although gain scheduling have been proven useful but it also have some limitations and potential dangers (Shamma and Athans 1992; Shorten 1996).

Traditional gain scheduling is limited to a design based on local models, which are linearisation of the process's dynamics around the equilibria. While this yields good results when the process is operating close to its equilibrium manifold, performance can be unsatisfactory for transients, which go through operating regions far away from the equilibrium manifold. Hunt and Johansen (Hunt and Johansen 1997) showed how a controller design based on local models which are not limited to linearisation around process equilibria, but which cover off-equilibrium transient operating conditions, can significantly improve the performance of the gain scheduling controller.

2.3 Local Model Network

The LMN was introduced in Poggio and Girosi (Poggio and Girosi 1990) and further extended for modelling and control purposes by Johansen and Foss (Johansen and Foss 1992b; Johansen and Foss 1993). The idea underlying the other network was introduced in Middleton *et al.* (Middleton *et al.* 1988) and further extended in Morse and Weller (Morse 1990; Morse *et al.* 1992; Weller and Goodwin 1994) and coined as the "hysteresis switching algorithm". This algorithm aims to achieve stability whereas, the "Multiple

Switched Model" (MSM) extensively studied by Narendra *et al.* (Narendra *et al.* 1995; Narendra and Balakrishnan 1997), but closely related to the former algorithm, is used for improving the control performance whilst dealing with systems having their parameters changing quickly through time (e.g. nonlinear systems).

The LMN consists of a single layer network of local plant models and a gating system often composed of radial basis functions. The various local models are activated by an input vector X varying with time that usually corresponds to the vector of parameters of a Nonlinear Auto Regressive Moving Average with eXogenous inputs (NARMAX). Therefore, each local model can be interpreted as a local NARMAX model of the plant.

An important advantage of this method is the facility to transform the LMN into a local controller network. Since each local model is linear, there are several straightforward methods arising from control theory that can be used to transform the linear models into local linear controllers. The early work of Johansen and Foss (Johansen and Foss 1992b; Johansen and Foss 1993) used a proportional and integral (PI) design method to transform each model into a controller. In Gawthrop and Ronco (Gawthrop and Ronco 1996) the models are used to develop predictive controllers. A pole placement technique is used in Hunt and Johansen (Hunt and Johansen 1997) and Gollee and Murray-Smith (Gollee and Murray-Smith 1997). A model reference adaptive control design method has been used to transform the local models into controllers in Sharman *et al.* (Sharman *et al.* 2002).

2.4 Local Controller Network

LCN is closely related to a gain scheduling approach which is a standard technique to handle systems having different operating regimes (Shamma and Athans 1992; Shamma and Athans 1990). In the gain scheduling approach, the gains of the controllers are changed according to the current operating condition. In the LCN approaches, this idea is extended in such a way that different control structures can be used to meet different requirements involved by different operating regimes.

The design of a controller is usually based on a model of the process. A large number of powerful design and analysis techniques for linear controllers whose design is based on a linear model of the process, see for example (Åström and Hagglund 1995; Franklin *et al.* 1991; Kailath 1980; Middleton and Graham 1990). These include PID controller, optimal control, pole placement etc.

If a model of the process is available in the form of an LMN with linear local models, a straightforward approach is to use a linear design technique to obtain a local linear controller for each local model. The local controllers can then be interpolated using the same scheduler as the LMN, which is referred in Figure 2.1 as a LCN. The main assumption here is that, once the system has been decomposed into locally valid models, a similar decomposition can be used to design a corresponding controller.

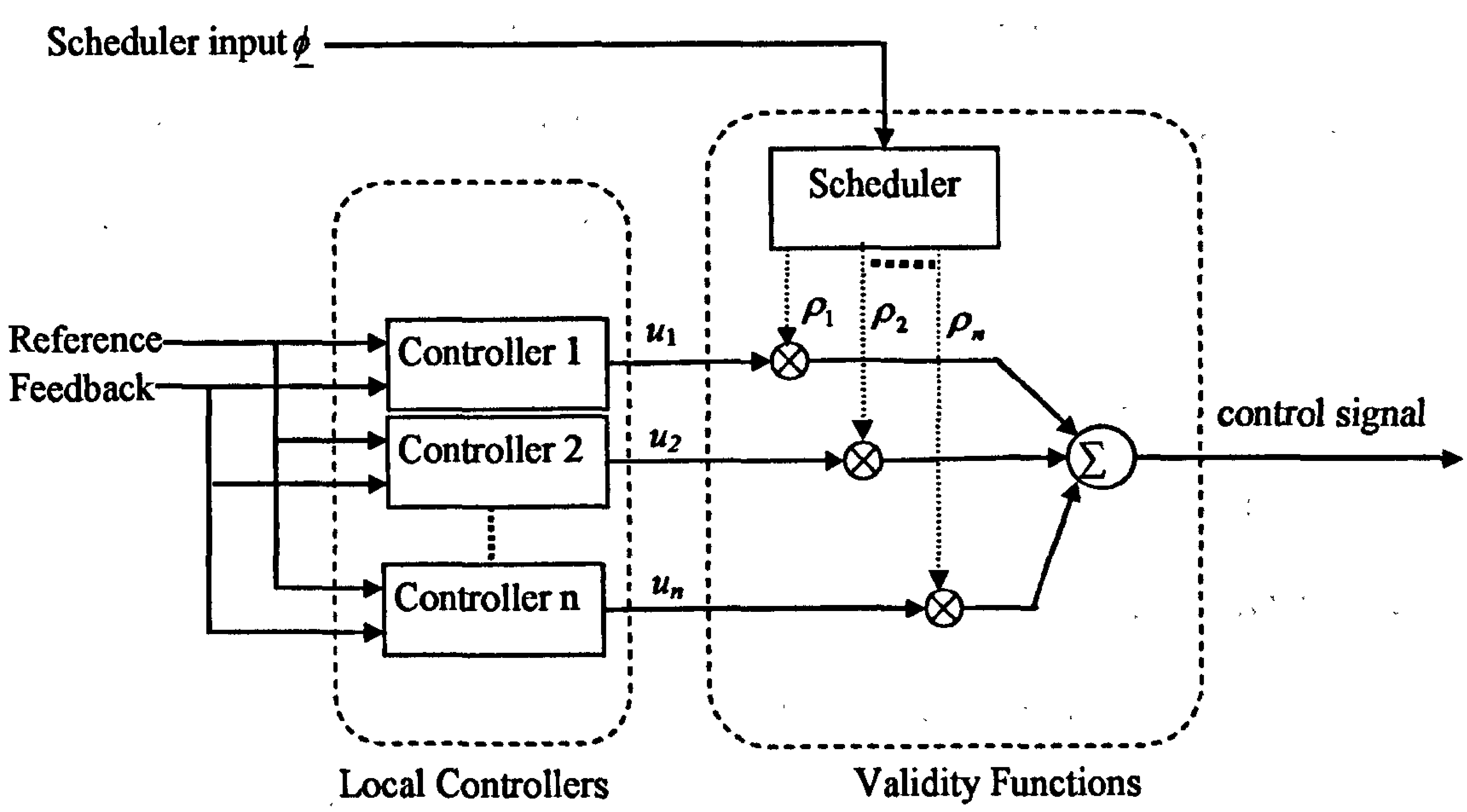

Figure 2.1 Local Controller Network

LCN have been used successfully in a number of applications (Gollee and Hunt 1997; Johansen *et al.* 1998). The approach is closely related to fuzzy control. It can be interpreted as a constructive way of designing gain-scheduling controllers. Instead of designing a network of controller off-line, controllers can be designed on-line. The interpolated LMN can be used to design a corresponding time-varying controller at each sampling instant. While this approach is closely related to adaptive control (Åström and Wittenmark 1989), it is computationally expensive as a new controller needs to be designed at each sampling instant. An alternative approach is suggested in Ronco (Ronco 1997), where a new controller is only designed if the performance requirements cannot be met by the current controller. Such a design approach is, however, potentially sensitive to noise and disturbances.

2.5 Linear Approximation Model

A LAM (Li and Tan 2000) approximates step response data by a transfer function or a transfer function matrix. This is similar to representing a nonlinear system response with high fidelity by convolution or harmonic analysis. This simple technique eases the difficulties encountered in conventional linearisation without the need for an initial nonlinear model. Unlike a LMN, LAM gives a straightforward approximation in the entire trajectory ranging from the initial condition to the setpoint, whilst a local model network is applicable only around the initial condition.

In modelling a nonlinear system, a set of LAMs (Figure 2.2) obtained at different operating points can be networked conveniently by simple linear local interpolation to produce an unseen operating level if required. Compared with a local model network, a LAM network offers the advantage of obtaining each linear model directly from step response and hence the advantage of a wide range of validity of individual models that can act stand-alone in a certain degree.

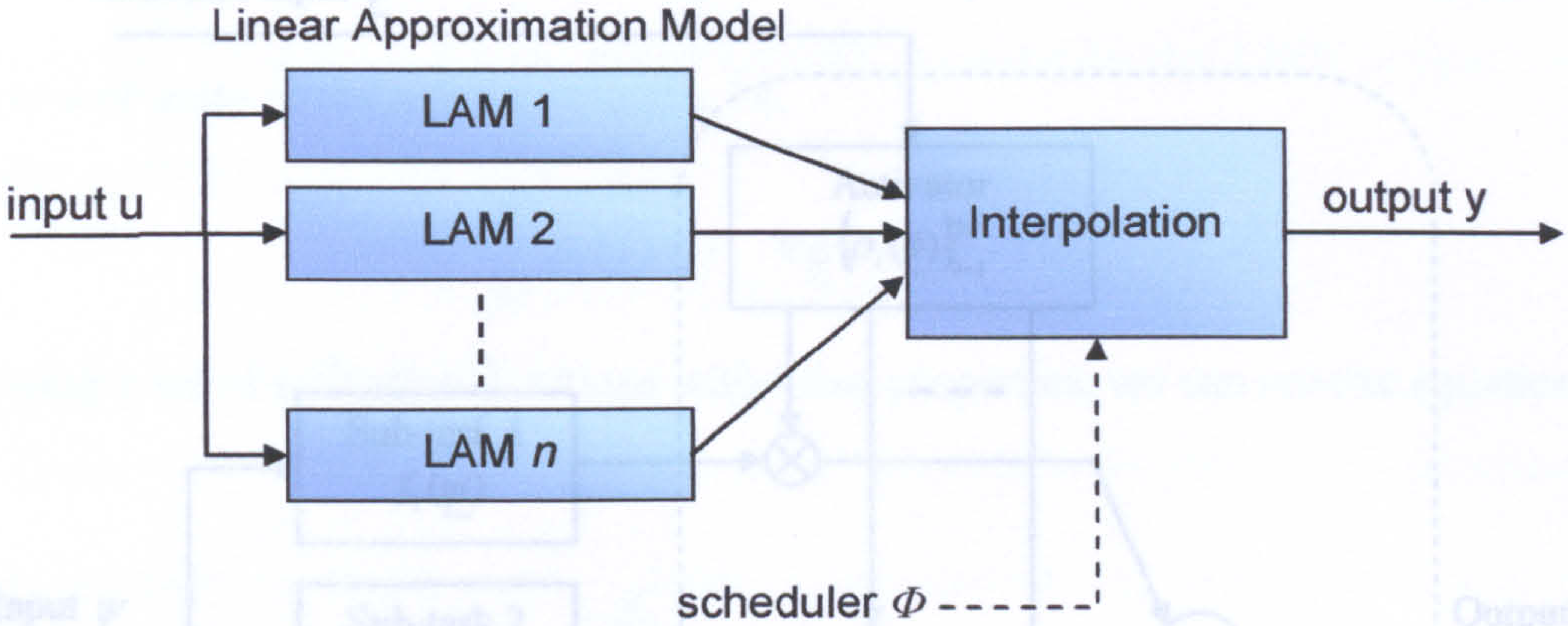

Figure 2.2 Interpolated LAM network

2.6 Framework of Networking

Based on the network techniques used in modelling and control, the overall networks can be divided into 2 main categories: structure networking and parameter networking. In structure networking, each element or sub-task contain its own entities, e.g. the state or the integration action within it. While in parameter networking, only one structure is used, but the parameters within the structure change accordingly.

2.6.1 Structure Networking

The basic idea of the structure networking approach employed is to divide a complex, nonlinear system into smaller, simpler sub-tasks. Each sub-task can then be handled independently by a simpler model/controller. An activator (scheduler/supervisor) decides how relevant the sub-tasks are for the current operating condition, and weights (interpolate) them accordingly. The overall system is composed as the sum of all weighted sub-tasks. Such a structure is shown in Figure 2.3.

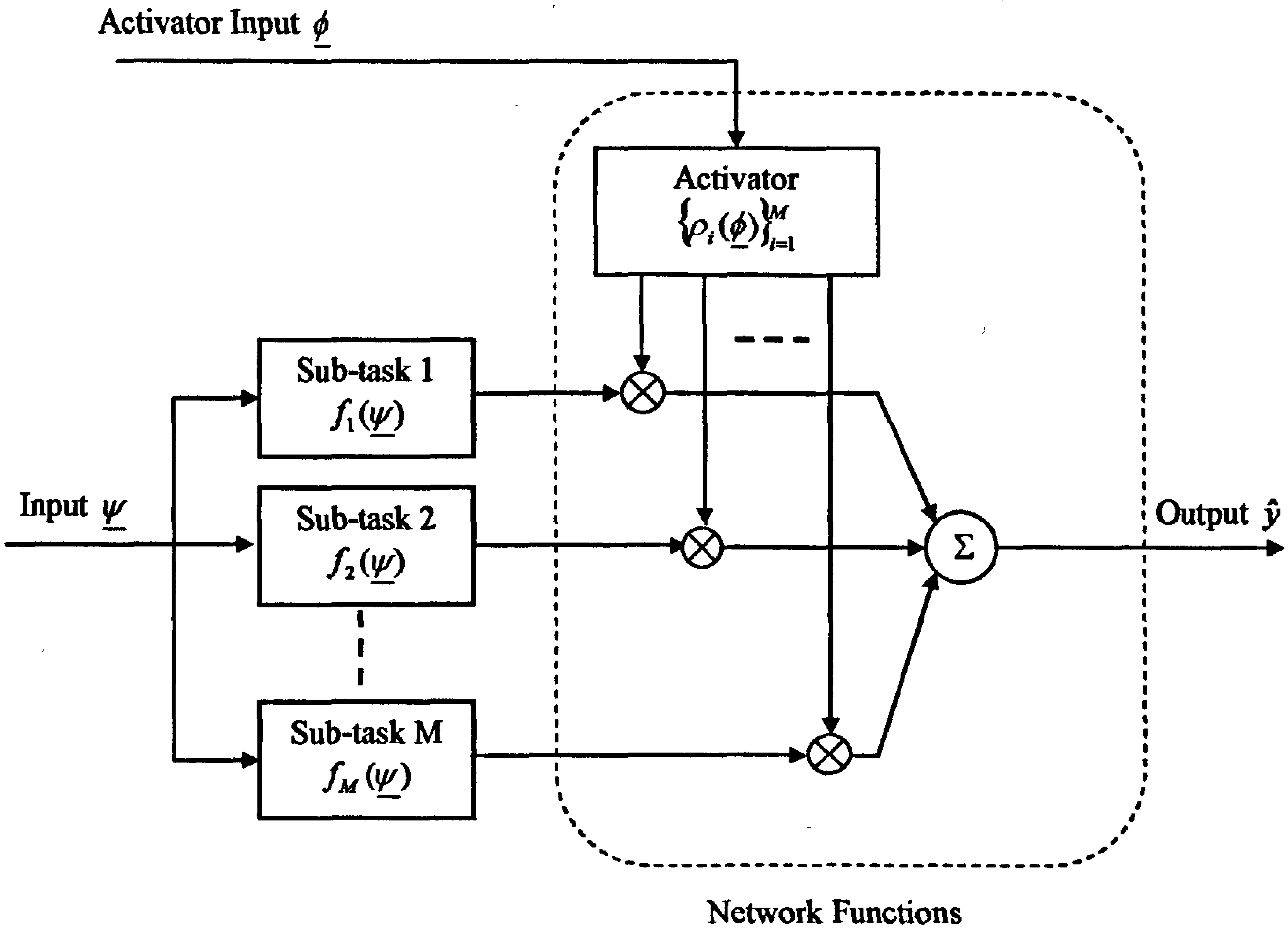

Figure 2.3 The multiple sub-task approach

The sub-tasks can generally be of any form, e.g. linear or nonlinear, input-output or state-space form, empirical or based on physical analysis. It is often straightforward to incorporate a prior knowledge at this stage. When a sub-task is a local model it is also known as Local Model Networks (LMN) (Johansen and Foss 1992a), alternatively if sub-task is a local controller it is know as Local Controller Network (LCN) (Murray-Smith and Johansen 1997).

In this structure networking, we consider

$$y = f(\underline{\psi}), \tag{2.1}$$

with $\underline{\psi} \in \Psi \subset \mathbb{R}^{n_\psi}$ the input vector, $y \in \Upsilon \in \mathbb{R}$ the scalar output, and $f : \Psi \to \Upsilon$ a smooth nonlinear continuous function. By introducing an activator that consists of M Scalar functions $\rho_i(\underline{\phi}) : \Phi \to [0,1]$, with $\underline{\phi} \in \Phi \subset \mathbb{R}^{n_\phi}$ being the activation variables. These functions are smooth with a localized region of activity. The set of functions $\{\rho_i\}_{i=1}^{M}$ forms a partition of unity of the activation space, i.e.

$$\sum_{i=1}^{M} \rho_i(\underline{\phi}) = 1 \quad \forall \underline{\phi} \in \Phi . \tag{2.2}$$

Employing a set of activation functions with these properties, we can rewrite equation (2.1) as

$$y = \sum_{i=1}^{M} \rho_i(\underline{\phi}) f(\underline{\psi}) . \tag{2.3}$$

If, for a given $\rho_i(\underline{\phi}) \approx 1$, a local approximation of f exists,

$$f_i(\underline{\psi}) \approx f(\underline{\psi}) \text{ if } \rho_i(\underline{\phi}) \approx 1 , \tag{2.4}$$

then an approximation to equation (2.3) can be formulated as

$$y \approx \hat{y} = \sum_{i=1}^{M} \rho_i(\underline{\phi}) f_i(\underline{\psi}) . \tag{2.5}$$

Here the functions f_i, which are locally valid approximations of the global functions f, are defined as local sub-tasks. The functions set $\{\rho_i\}_{i=1}^{M}$ define the validity of the corresponding local sub-tasks; they are thus called *activation functions*. The function ρ_i has the largest value for those operation conditions where the function f_i is the best approximation to f, and is closed to zero elsewhere. The overlap between neighbouring activation functions results in interpolation between sub-tasks. The advantage of this approach being the network could be easily expanded by adding an identical element into the structure. However, from another point of view, the entities contained in the elements are different from each other, issues similar to gain scheduling where scheduling variable have to vary slowly is a setback.

2.6.2 Parameter Networking

Assuming that sub-tasks is a local model network, considering that LMN in time-invariant nonlinear system in the state space form,

$$\begin{aligned}\dot{\underline{x}}(t) &= f(\underline{x}(t), u(t-T_d)); \qquad \underline{x}(0)=\underline{x}_0 \\ y(t) &= g(\underline{x}(t))\end{aligned} \quad . \tag{2.6}$$

Here $f(x)$ and $g(x)$ are nonlinear continuous differentiable functions. For simplicity, we restrict ourselves to single input – single output system, the linearised system can be written as

$$\begin{aligned}\dot{\underline{x}}(t) &= A\underline{x}(t) + \underline{b}u(t-T_d) + \underline{d}^x \\ y(t) &= \underline{c}^T\,\underline{x}(t) + d^y\end{aligned} \quad , \tag{2.7}$$

with the bias terms $\underline{d}^x$ and d^y defined as

$$\begin{aligned}\underline{d}x &= f(\underline{x}^o, u^o) - A\underline{x}^o - \underline{b}u^o \\ dy &= y^o - c^T\,\underline{x}^o\end{aligned} \quad . \tag{2.8}$$

In the state space description (2.6), we have two nonlinear functions, f and g, which can be approximated by means of local functions decomposition, applying functions (2.5), the system can be rewritten as a weighted sum of local models,

$$\begin{aligned}\dot{\underline{x}}(t) &= \sum_{i=1}^{M} \rho_i(\underline{\phi}(t)) f_i(\underline{x}(t), u(t-Td)); \; \underline{x}(0)=\underline{x}_0 \\ y(t) &= \sum_{i=1}^{M} \rho_i(\underline{\phi}(t)) g_i(\underline{x}(t))\end{aligned} \quad , \tag{2.9}$$

which is a LMN representation of the system (2.6). Employing the concept of local linearisation for different operating conditions, we choose to work with linear local state-space representations as described by equation (2.7).This results in

$$\begin{aligned}f_i(\underline{x}(t), u(t)) &= A_i\underline{x}(t) + \underline{b}_i u(t-Td) + \underline{d}_i^x \\ g_i(\underline{x}(t)) &= \underline{c}_i^T\,\underline{x}(t) + d_i^y\end{aligned} \quad , \tag{2.10}$$

with i=1,...,M. The overall system can then be approximated as

$$\underline{\dot{x}}(t)=\sum_{i=1}^{M}\rho_i(\underline{\phi}(t))\left[A_i\underline{x}(t)+\underline{b}_i u(t-Td)+\underline{d}_i^x\right],\ \underline{x}(0)=\underline{x}_0$$
$$y(t)=\sum_{i=1}^{M}\rho_i(\underline{\phi}(t))\left[\underline{c}_i^T\underline{x}(t)+d_i^y\right] \quad (2.11)$$

Such network can also be described as a linear parameter-varying (LPV) (Kajiwara *et al.* 1999) system,

$$\underline{\dot{x}}(t)=A(\underline{\phi}(t))\underline{x}(t)+\underline{b}(\underline{\phi}(t))u(t-T_d)+\underline{d}^x(\underline{\phi}(t));\quad \underline{x}(0)=\underline{x}_0$$
$$y(t)=\underline{c}^T(\underline{\phi}(t))\underline{x}(t)+d^y(\underline{\phi}(t)) \quad (2.12)$$

where $A,\underline{b},\underline{d}^x,\underline{c}$ and d^y are the interpolated parameters of the local models,

$$A(\underline{\phi})=\sum_{i=1}^{M}\rho_i(\underline{\phi})A_i,\quad \underline{b}(\underline{\phi})=\sum_{i=1}^{M}\rho_i(\underline{\phi})\underline{b}_i,\quad \underline{d}^x(\underline{\phi})=\sum_{i=1}^{M}\rho_i(\underline{\phi})\underline{d}_i^x,$$
$$\underline{c}(\underline{\phi})=\sum_{i=1}^{M}\rho_i(\underline{\phi})\underline{c}_i,\quad \underline{d}^y(\underline{\phi})=\sum_{i=1}^{M}\rho_i(\underline{\phi})\underline{d}_i^y \quad (2.13)$$

The parameters of (2.12) depend only on the activator $\underline{\phi}$. This type of parameter varying structure is depicted in Figure 2.4. The goodness of such an approach is that it is capable of reacting faster to scheduling variables, therefore overcoming the deficiency in structure networking. Despite having to breakdown the entities in the structure into each schedulable parameter, this network is much more robust and easy to manipulate.

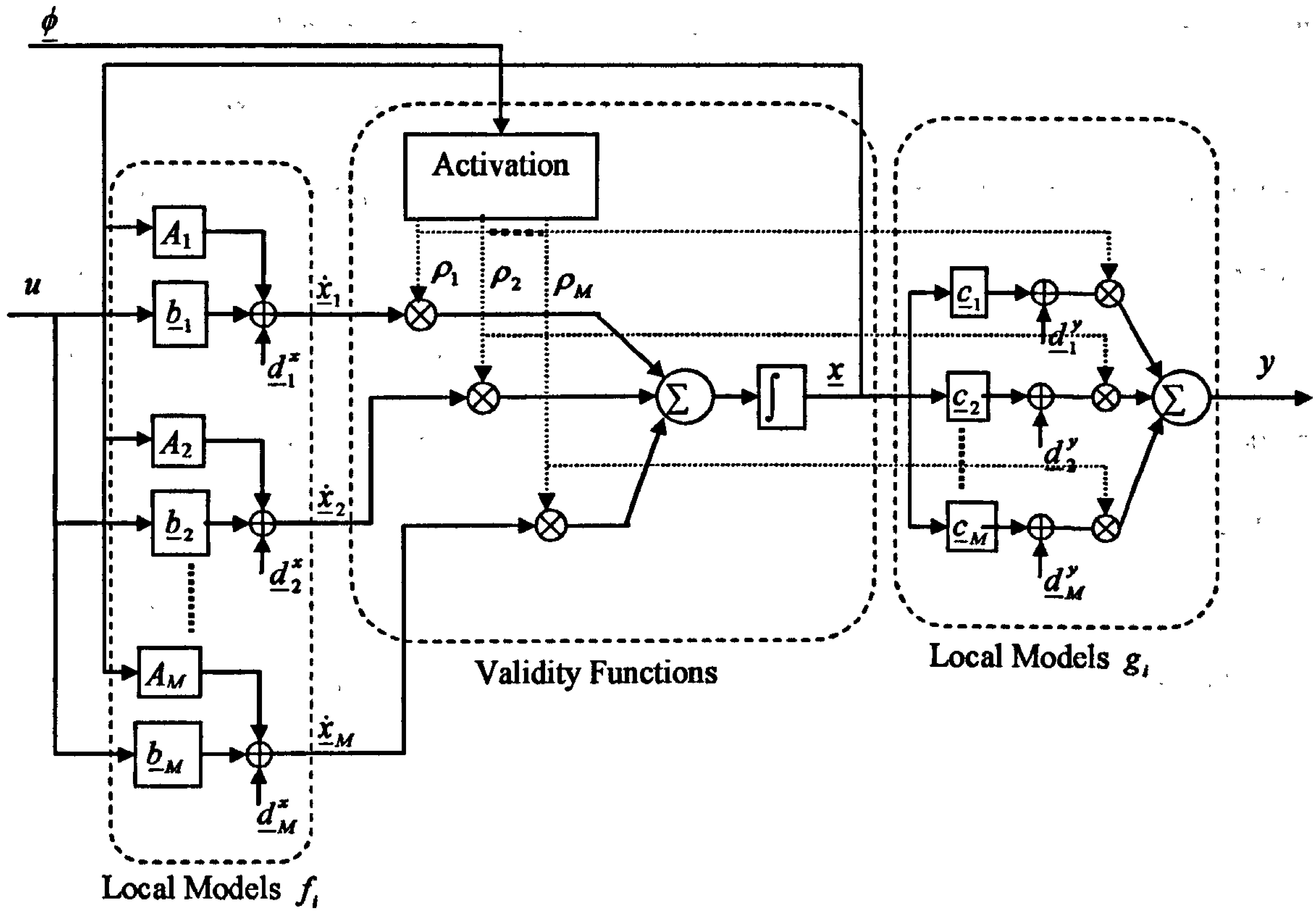

Figure 2.4 Parameter-varying Local Model Network in a state-space representation

In this thesis, we adopt parameter-varying structure for model and controller in a much simpler form, which will be discussed in chapter 3.

2.7 Activation Functions

Almost all multiple model networks are built from a simple basic principle, interpolation and weighting (Stilwell and Rugh 1998). Neural network, Fuzzy Logic, ARMAX or NARMAX adopt multiple layer of interpolation and rules to incorporate each of the layers and weigh each of the nodes based on some activation or scheduling according to the reference or operating points.

There is no systematic way to determine the required structure to model a given system. The most serious problem perhaps is that the network representation is a black-box, the properties of the model cannot be analysed. In another hand, LMN, LCN and LAM are much simpler networks with a clear structure that can be handled easily. In this thesis, only such clear structure is adopted. It is far more important to apply a clear design in practical or physical plant than having a blank box that works in the dark.

Networking/interpolation can be of linear or nonlinear type. In most cases nonlinear interpolation are more complex and exposed to a high order of variation in parameters during the transition of operating points. Although nonlinear interpolation can more accurately represent a given model, they require more computation power and cannot safely use on any given plant without understanding the dynamics of that plant, while linear interpolation is not subjected to such uncertainties.

To satisfy the approximation described in equation (2.5) we require the following properties from a set of activation functions:

a) A activation function transforms its input to a value between 0 and 1:

$$\{\rho_i : \Phi \rightarrow [0,1]\}_{i=1}^{M} \tag{2.14}$$

b) The value of activation function decreases with increasing distance of the input from its maximum (its 'centre'). The activation converges to zero for inputs that are far away from the 'centre'.

c) The set of activation functions forms a partition of unity of its input space, i.e.

$$\sum_{i=1}^{M} \rho_i(\underline{\phi}) = 1 \quad \forall \underline{\phi} \in \Phi \subset \mathbb{R}^{n_\phi} \tag{2.15}$$

To ensure that every point in the input space is covered to the same degree, any interpolation functions with the properties tested could be applied as a activation function, i.e., Spline , Cubic Spline, Cubic, Nearest neighbour, Piecewise cubic Hermite, linear and extrapolation (see Figure 2.5 to Figure 2.10).

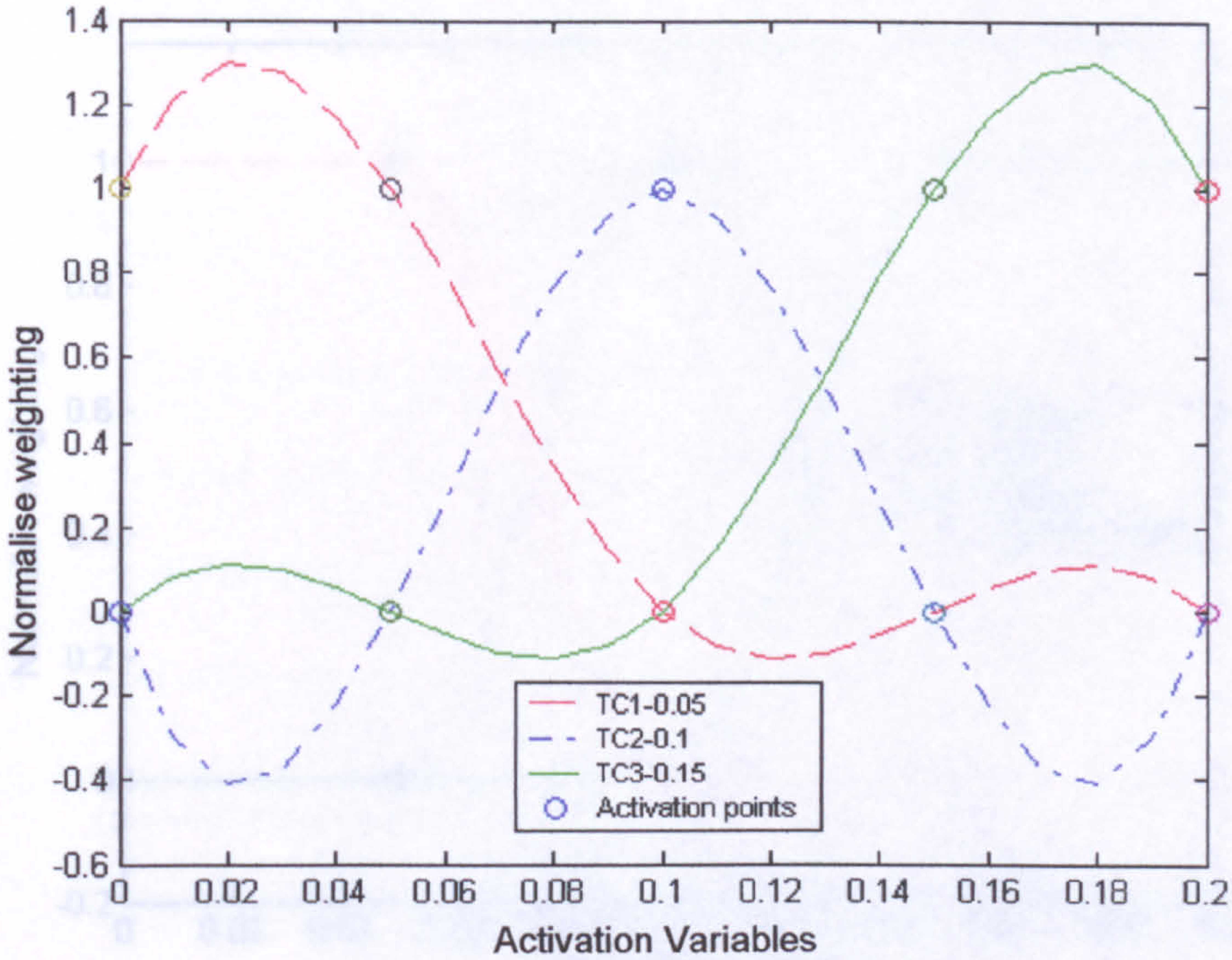

Figure 2.5 Normalised Cubic spline interpolation

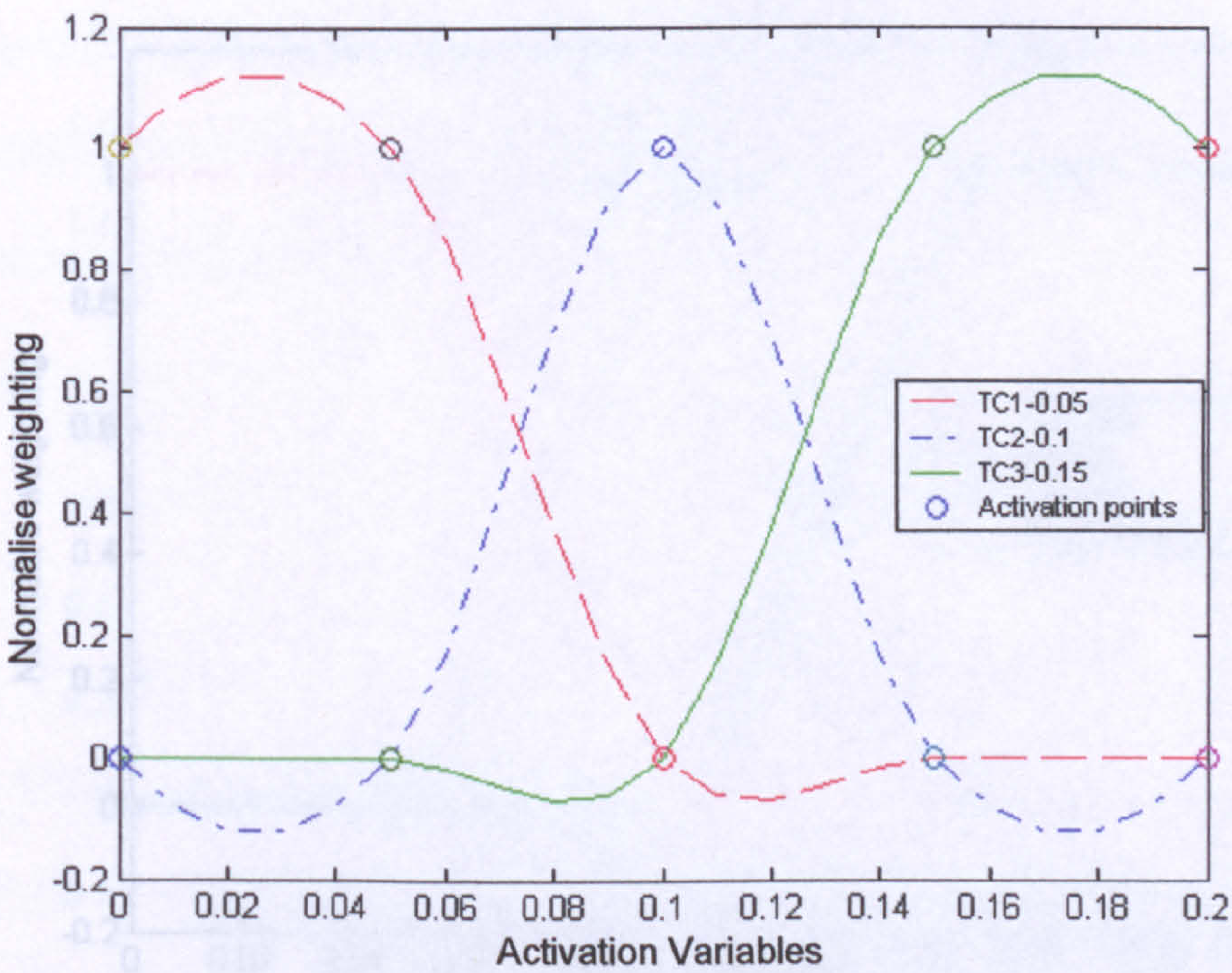

Figure 2.6 Normalised Cubic interpolation

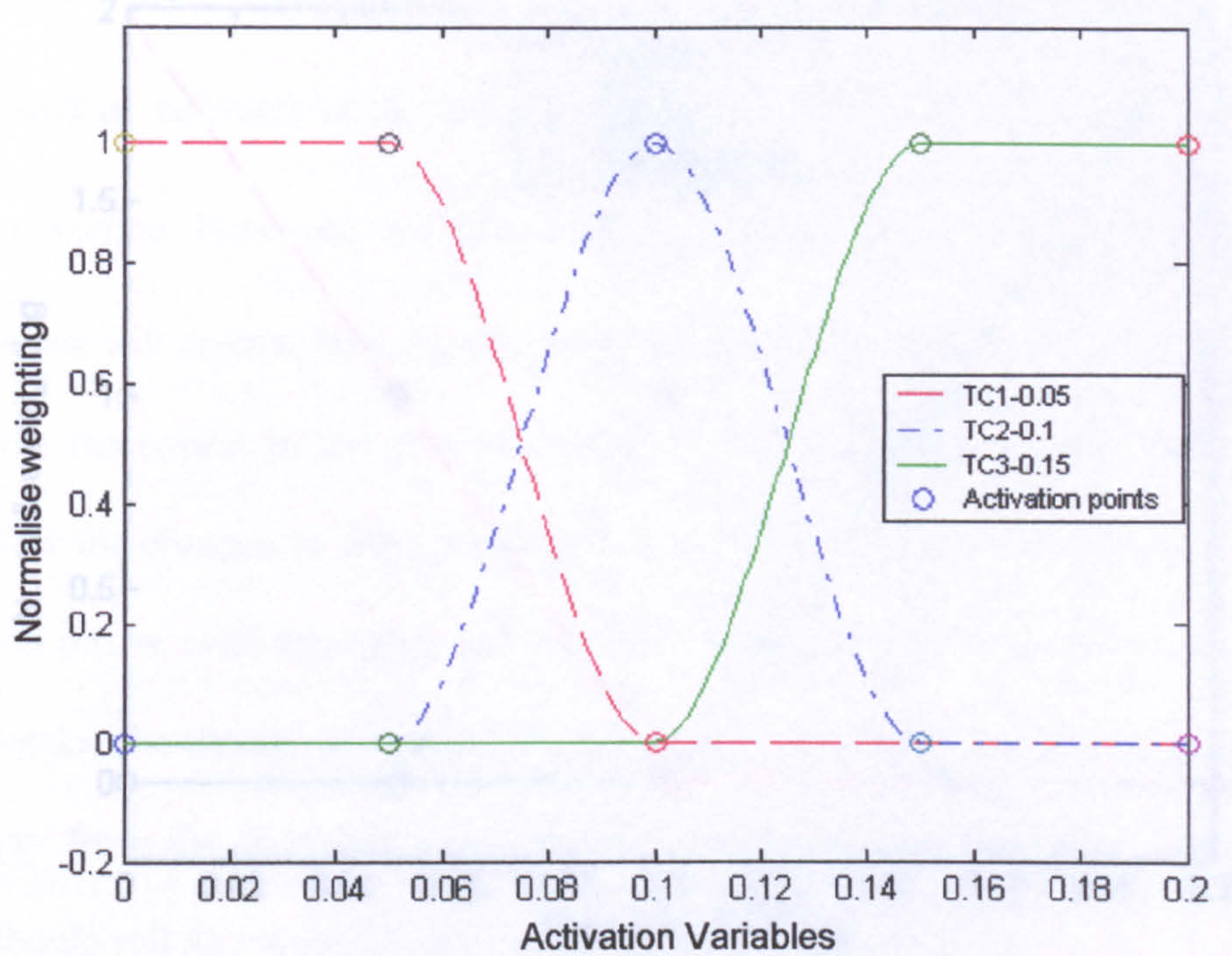

Figure 2.7 Normalised Piecewise cubic Hermite interpolation

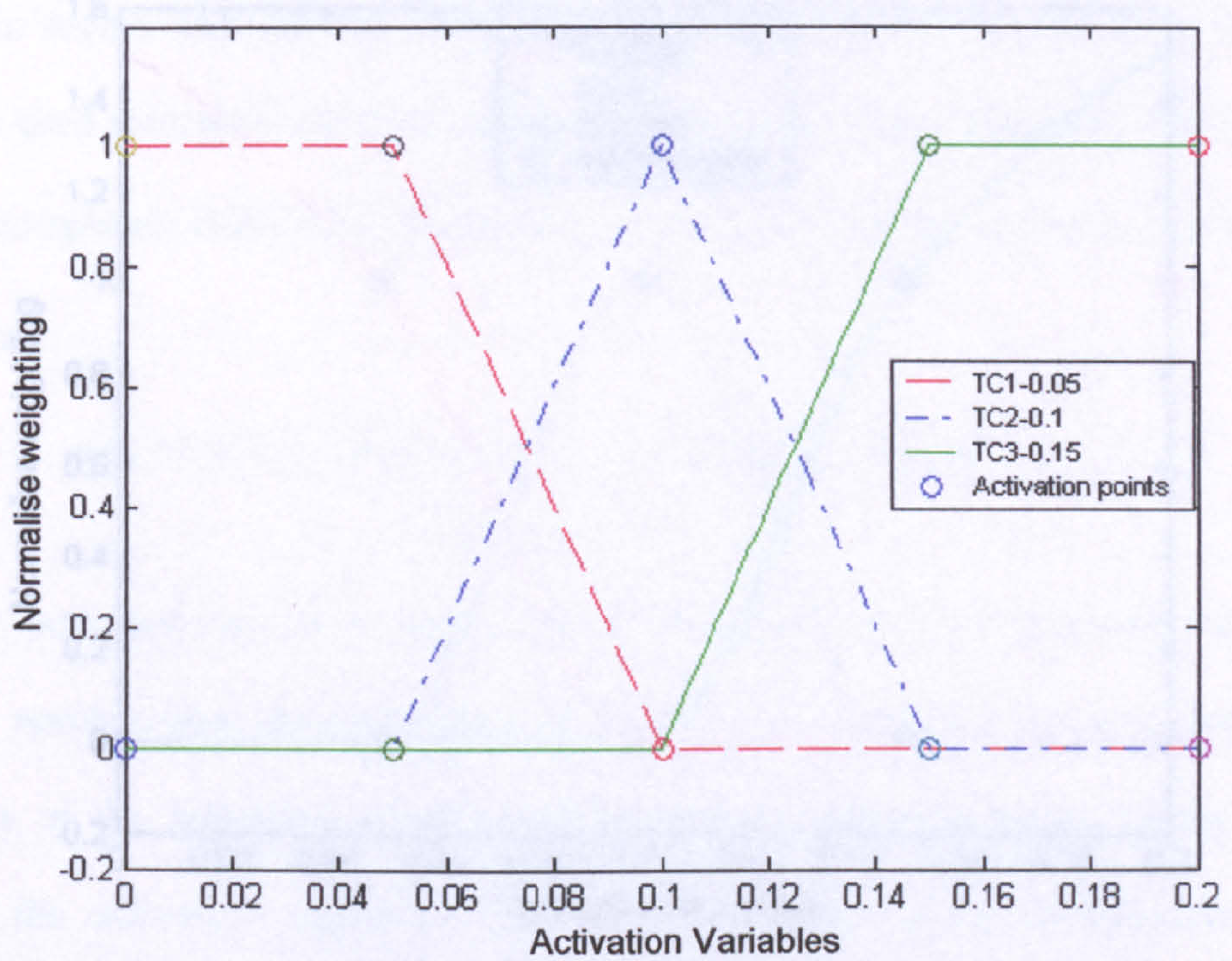

Figure 2.8 Normalised Linear interpolation

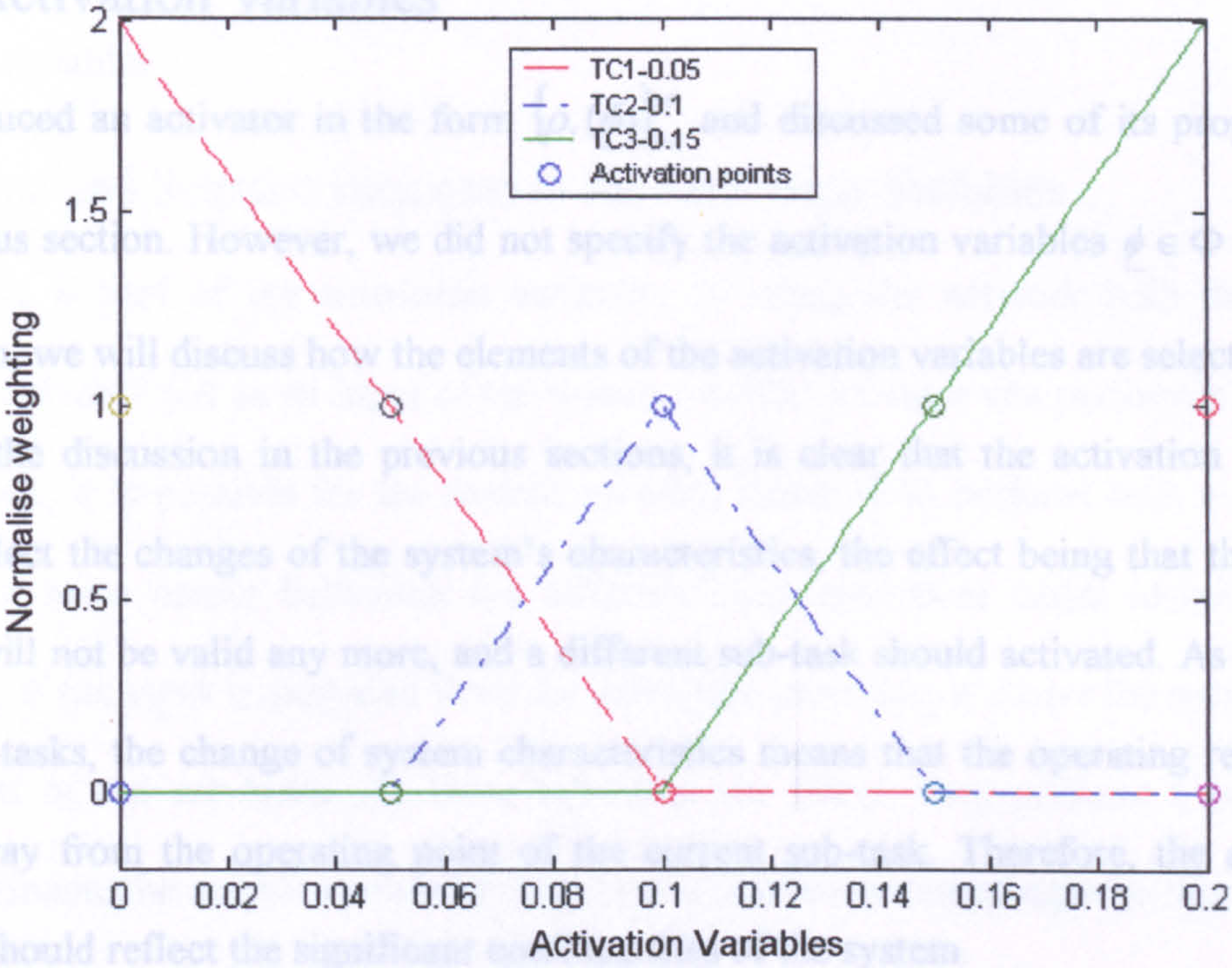

Figure 2.9 Normalised Linear extrapolation

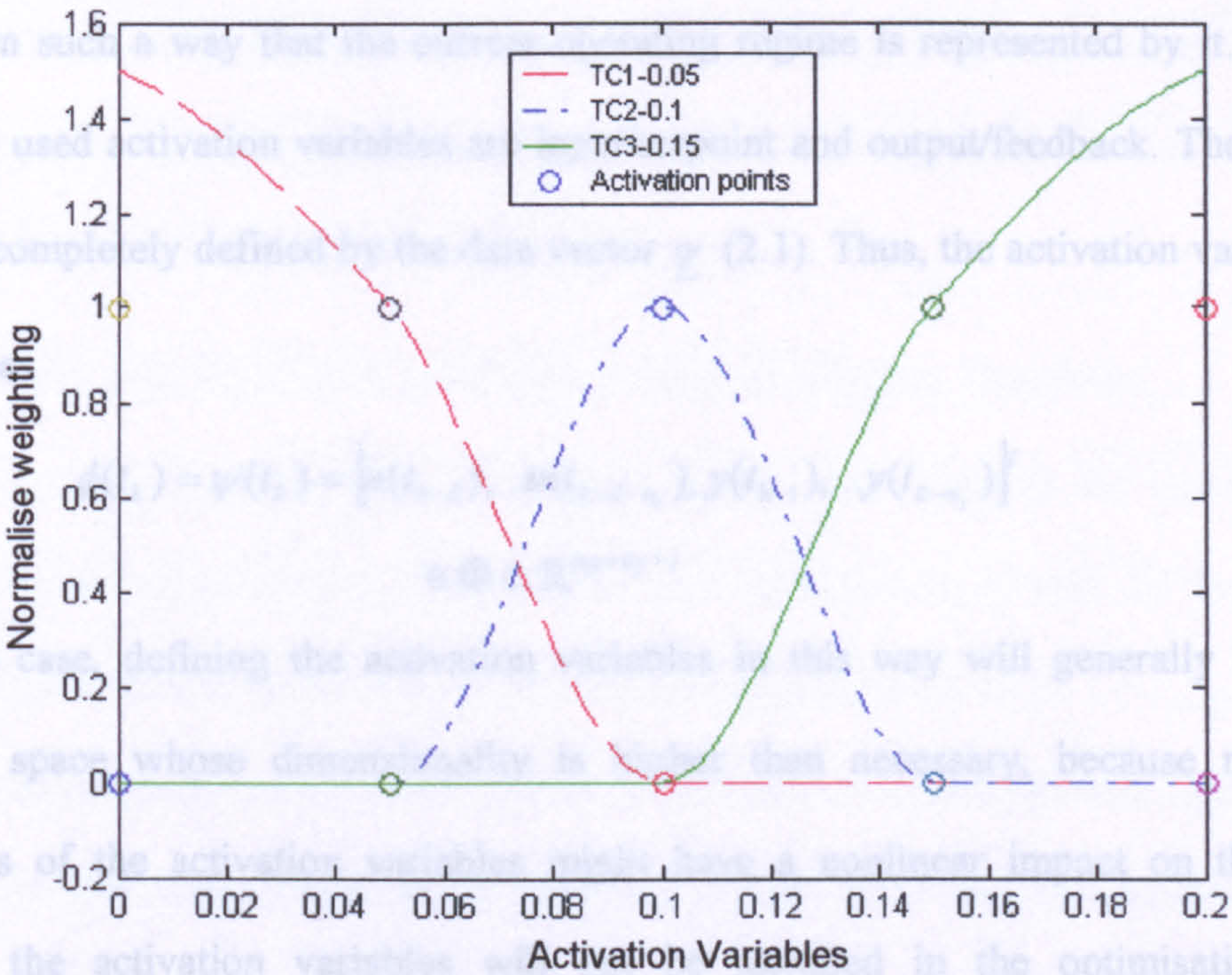

Figure 2.10 Normalised cubic Hermite extrapolation

2.8 Activation Variables

We introduced an activator in the form $\{\rho_i(\underline{\phi})\}_{i=1}^{M}$ and discussed some of its properties in the previous section. However, we did not specify the activation variables $\underline{\phi} \in \Phi \subset \mathbb{R}^{n_\phi}$. In this section, we will discuss how the elements of the activation variables are selected.

From the discussion in the previous sections, it is clear that the activation variables should reflect the changes of the system's characteristics, the effect being that the current sub-task will not be valid any more, and a different sub-task should activated. As we adopt linear sub-tasks, the change of system characteristics means that the operating regime has moved away from the operating point of the current sub-task. Therefore, the activation variables should reflect the significant nonlinearities of the system.

A change of the system's characteristics is linked to a change of the operating regime of the task. A straightforward approach is therefore selecting the elements of the activation variables in such a way that the current operating regime is represented by it. The most commonly used activation variables are input/setpoint and output/feedback. The operating regime is completely defined by the data vector $\underline{\psi}$ (2.1). Thus, the activation variables can be chosen as

$$\phi(t_k) = \psi(t_k) = \left[u(t_{k-d}),..u(t_{k-d-n_u}), y(t_{k-1}),..y(t_{k-n_y})\right]^T \in \Phi \subset \mathbb{R}^{nu+ny+1} \tag{2.16}$$

In this case, defining the activation variables in this way will generally lead to an activation space whose dimensionality is higher than necessary, because not all the dimensions of the activation variables might have a nonlinear impact on the system, therefore, the activation variables will not be included in the optimisation of the networking system in this research. The use of *a priori* knowledge about the networking system is of great importance here. We will therefore discuss some aspects of the activation variables.

2.8.1 Input and Setpoint Elements in the Activation Variables

If the input is part of the activation variables, it enters the network both though the nonlinear activator and as an input of the system. As the activator can perform a many-to-one mapping, it is possible for the overall network structure to perform such a mapping, i.e. to have same output behaviour for different input sequences under identical initial conditions. If the input is excluded from the activation variables, it enters the network only as an input of the sub-tasks. As these sub-tasks are linear, they perform a one-to-one mapping. Hence, the overall system will perform a one-to-one mapping with respect to the input: different input condition sequences will necessarily lead to different output patterns, assuming the same initial conditions (see Figure 2.11), thus, leading to a more general network structure.

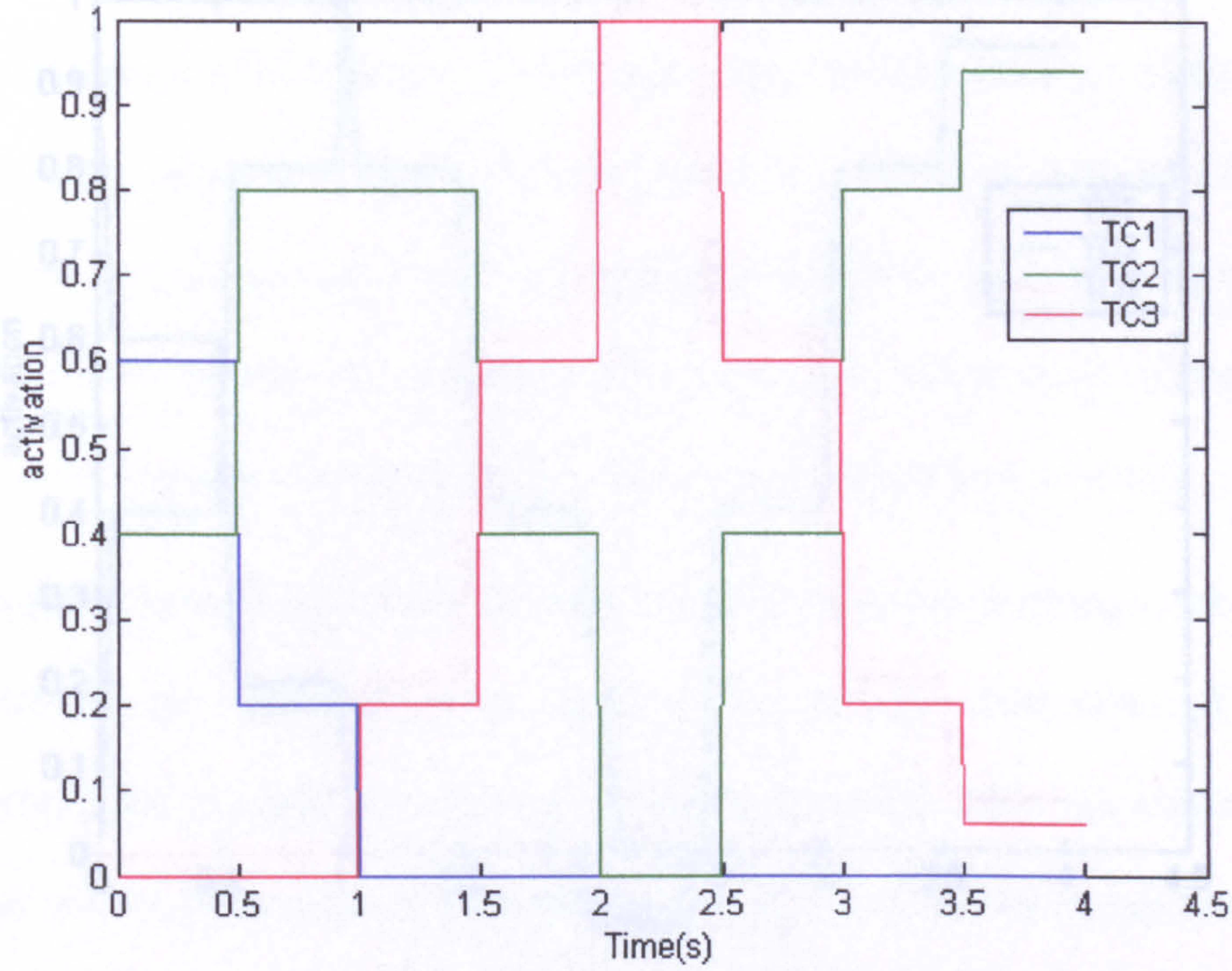

Figure 2.11 Input or setpoint activation

The setpoint activation method is one of the most commonly use method in networking systems. This method switches the controllers instantaneously from one to another. From the figure above, we can see the instant step change of controllers. Such a method gives a fast response to setpoint change.

2.8.2 Output and Feedback Elements in the Activation Variables

If output elements (i.e. the state, or delayed values of the output) are included in the activation variables, changes to the process require readjustment of both the parameters of the local models and the parameters of the activator (see Figure 2.12).

If output elements are excluded from the activation variables, limited changes of the plant require only readjustment of the parameters of the sub-tasks. The parameters of the activator can remain unchanged.

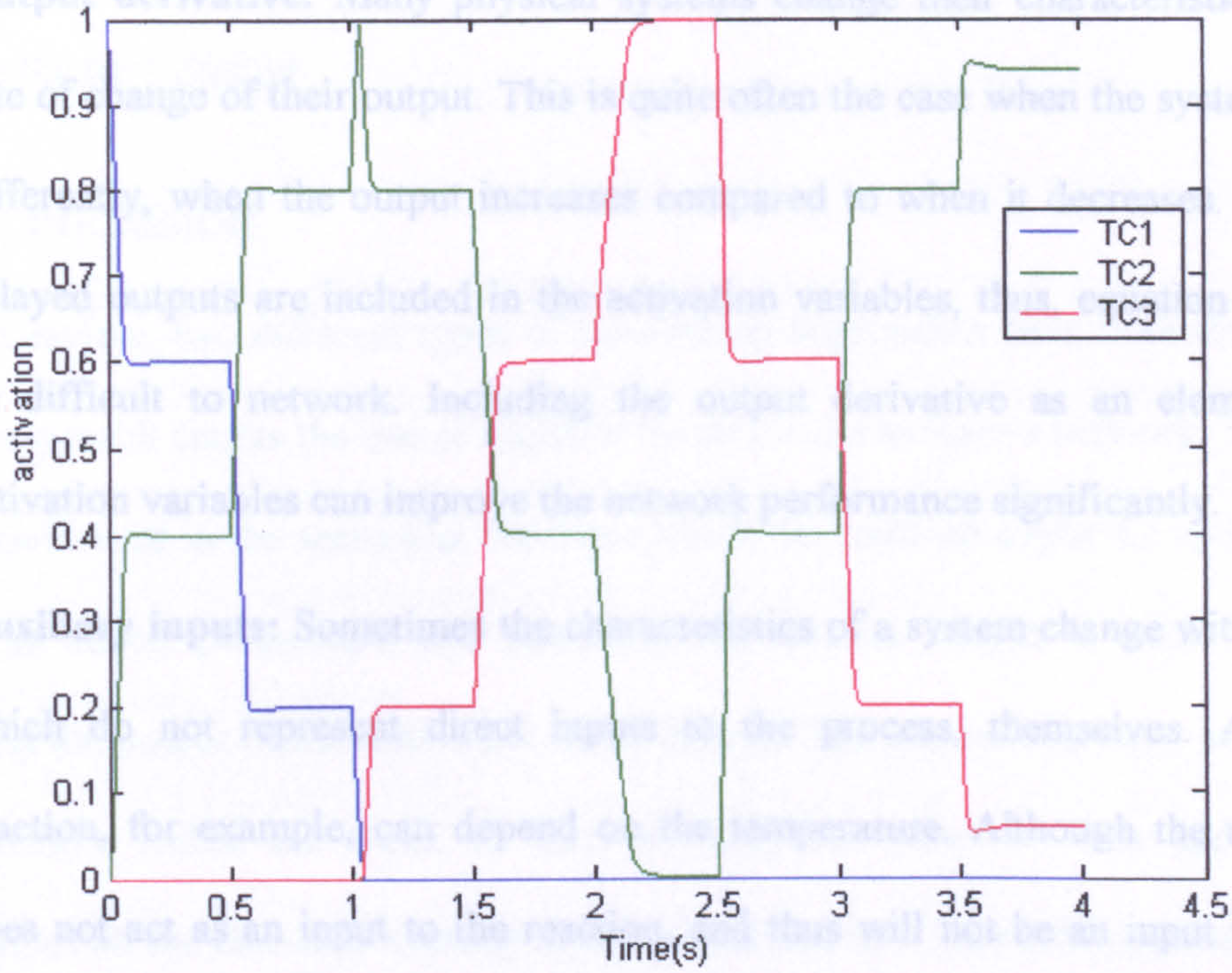

Figure 2.12 Output/feedback activation

Output activation method switches the controller gradually according to the output of a plant. This is a stable way to schedule controllers but have a slower response time to setpoint change and sensitive to output noise. One numerical error found is the zero response due to initial zero state of all values.

2.8.3 Additional Elements in Activation Variables

So far, we have only discussed the use of input and output as activation variables. Depending on the system to be networked, it might be useful to take additional variables as potential elements for the activation variables into account:

- **Filtered inputs:** The input of the process can often change very rapidly. When inputs are directly used for activation, this leads to an abrupt switch to a different sub-task. It is usually desirable to change the sub-task characteristics more smoothly. Employing a low-pass filtered version of the input for activation can improve the switching characteristics of the sub-task significantly.

- **Output derivative:** Many physical systems change their characteristics with the rate of change of their output. This is quite often the case when the system behaves differently, when the output increases compared to when it decreases. If only the delayed outputs are included in the activation variables, thus, equation (2.16) will be difficult to network. Including the output derivative as an element of the activation variables can improve the network performance significantly.

- **Auxiliary inputs:** Sometimes the characteristics of a system change with variables, which do not represent direct inputs to the process, themselves. A chemical reaction, for example, can depend on the temperature. Although the temperature does not act as an input to the reaction, and thus will not be an input to the local models, it is important as an element of the activation variables. Thus, it might be necessary to include auxiliary variables in the activation variables, which are not inputs to the local models.

2.9 Operating Nodes

To have an effective network, it is essential to find the optimal number and assign the operating points or nodes/position of each sub-task to an appropriate location. Although these elements can be included in the search or optimisation process, using *a priori* knowledge with human intervention will reduce the search time tremendously and produce a safer and more effective networking system. There are various techniques to find networking nodes, the simplest method being uniform distribution between the activation spaces.

In this research, we have a novel technique to find an effective non-uniform, nonlinear asymmetrical nodes. Using the process data collected, a static model is formed within the operating range and then a trajectory is formed along the static model. We will explain this method in the next chapter.

2.10 Conclusion

In this review, two different types of networking approaches have been distinguished. The first approach entails the use of multiple linear blocks to form a network. This method is more convenient in the sense that the linear block designed on a specific operating point retained its tuned parameters and structure, the overall output is achieved through networking the output of the individual linear blocks. The second approach intends to use the parameters within the block to form a parameter network. This approach is to breakdown of all the elements existing in the blocks and later networking those parameters of the system.

In a LMN that adopts parameter networking, requires linearisation and hence firstly a nonlinear model needs to be built for carrying out the linearisation. Otherwise, controlled perturbation, which can induce unnecessary plant upset, would be needed in order to obtain a locally valid linear model at each operating point under consideration. Moreover, the state in the LMN also needs to be networked, therefore, increase the complexity of the networking system.

LCN and LAM adopted structure networking. In LCN, each of the local controller is design from a locally valid models of the LMN. Therefore, each of the local controller in LCN is bond by the validity region of the LMN. This implied higher number of linear controller block used.

Unlike LMN, LAM does not require linearisation, each linear model can obtain directly from step response data, eliminate the need of intermediate nonlinear model use in LMN. Although LAM eliminate linearisation step, but it only valid for step response where the setpoint is fixed. Varying the setpoint requires regeneration of the model from initial state, therefore post a problem in continuous time modelling. Other than the problems stated, there are also no systematic way to obtain optimum operating nodes for the network.

The underlying aim of this research is to develop an easy-to-use 'quasi-linear' modelling and control technique for nonlinear systems, particularly when these systems need to operate in an operating envelope as opposed to at one fixed point. Critical review carried out on existing multiple linear model based control system techniques has revealed their complexity and inadequacy.

Chapter 3

Trajectory-Scheduling Networks Suitable for a Range of Setpoints and Operating Envelopes

The estimation of the trajectory nodes and model dynamics are highly correlated (Platt 1991). The nonlinearity of the local region of the operating space is directly related to the position of trajectory nodes assigned to the system. Different methods have been applied to determine the activity region surrounding the trajectory nodes which depend on their distance and location (Tan *et al.* 2004). The simplest methods consist of uniformly distributing the trajectory nodes on the operating space. This is obviously not suitable in many cases since the nonlinearity of the operating space cannot be expected to be homogeneous.

To have an effective network, it is essential to find the optimal number and assign the operating points or nodes/position of each sub-task at an appropriate location. Although these elements can be included in the search or optimisation process, using *a priori* knowledge with human intervention will reduce the search time tremendously and produce a safer and more effective networking system (Chong and Li 2002b).

3.1 Multiple Setpoints and TSN Nodes

To find a set of effective operating nodes for a process, the first step is to capture the nonlinearity of the process. Using a chemical process as an example, the response between the input and the steady-state output of the process are plotted in Figure 3.1. Such a plot is also known as static model. The operating range is between 0 and 1.

Starting by inserting 2 nodes to the operating trajectory, the first node is at lowest operating trajectory (node 1); another node is at the highest operating trajectory (node 3). As we can see, 2 nodes cannot approximate nonlinear trajectory effectively. Therefore, One new node will be inserted in between nodes (1) and (3) using the algorithm,

$$\mathrm{N} \Leftarrow \Delta d_{n,n+1}\, max$$

where N is a new node satisfy the condition $\Delta d_{n,n+1}\,max > \tau$, where τ is the tolerance which can be specified by user.

With the new node (2) added ($\Delta d_{1,3}\,max > \tau$), we now have 3 operating nodes along the operating trajectory. To satisfy the condition $\Delta d_{n,n+1}\,max > \tau$, $\Delta d_{1,2}\,max$ and $\Delta d_{2,3}\,max$ will be calculated and checked against τ. In this case both ($\Delta d_{1,2}\,max$; $\Delta d_{2,3}\,max$)$< \tau$, so there will be no new nodes inserted into the operating trajectory. The resulting operating nodes for this example were node (1) at 0, node (2) at 0.553 and node (3) at 1.

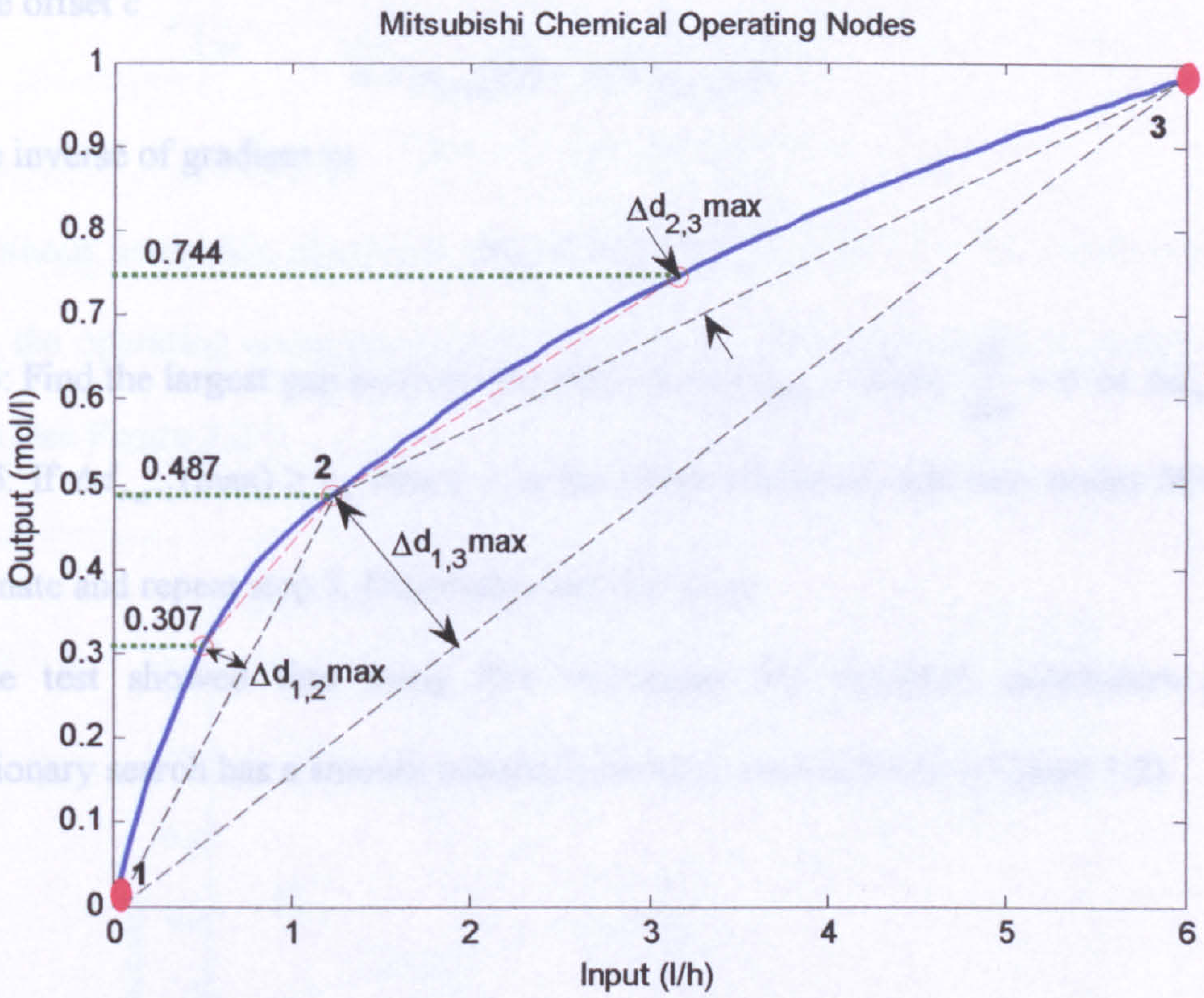

Figure 3.1 Algorithm for finding operating nodes (from Case 2)

3.1.1 Formulation

To generalise the formula, we first normalise the static model. Assigned x=input, y=output and n is the operating point, the normalise the static model is

$$y_{norm}(n)=\frac{y(n)-y(n_{\min})}{y(n_{\max})-y(n_{\min})},\; x_{norm}(n)=\frac{x(n)-x(n_{\min})}{x(n_{\max})-x(n_{\min})} \tag{3.1}$$

Step 1: Assigned the two most significant nodes N as: $N(1)=[x_{norm}(n_{\min}), y_{norm}(n_{\min})]$ and

$N(2)=[x_{norm}(n_{\max}), y_{norm}(n_{\max})]$

Step 2: Sort the nodes incrementally.

Step 3: Find the gradient (m) between the nodes $N(n)$ to $N(n+1)$.

$$m=\frac{y_{norm}(n)-y_{norm}(n+1)}{x_{norm}(n)-x_{norm}(n+1)} \tag{3.2}$$

and the offset c

$$c=y_{norm}(n)-m\times x_{norm}(n) \tag{3.3}$$

get the inverse of gradient m

$$m_{Inv}=-\frac{1}{m} \tag{3.4}$$

Step 4: Find the largest gap between the slope m and y_{norm}, where $\frac{dy}{dm}=0$ or $\Delta d_{n,n+1}(\max)$

Step 5: If $\Delta d_{n,n+1}(\max)\geq\tau$, where τ is the given tolerance, add new nodes $N(+1)$ to the coordinate and repeat step 3. Otherwise, end the route

The test showed that using this technique, the resultant parameters from the evolutionary search has a smooth transition between each node (see Figure 3.2).

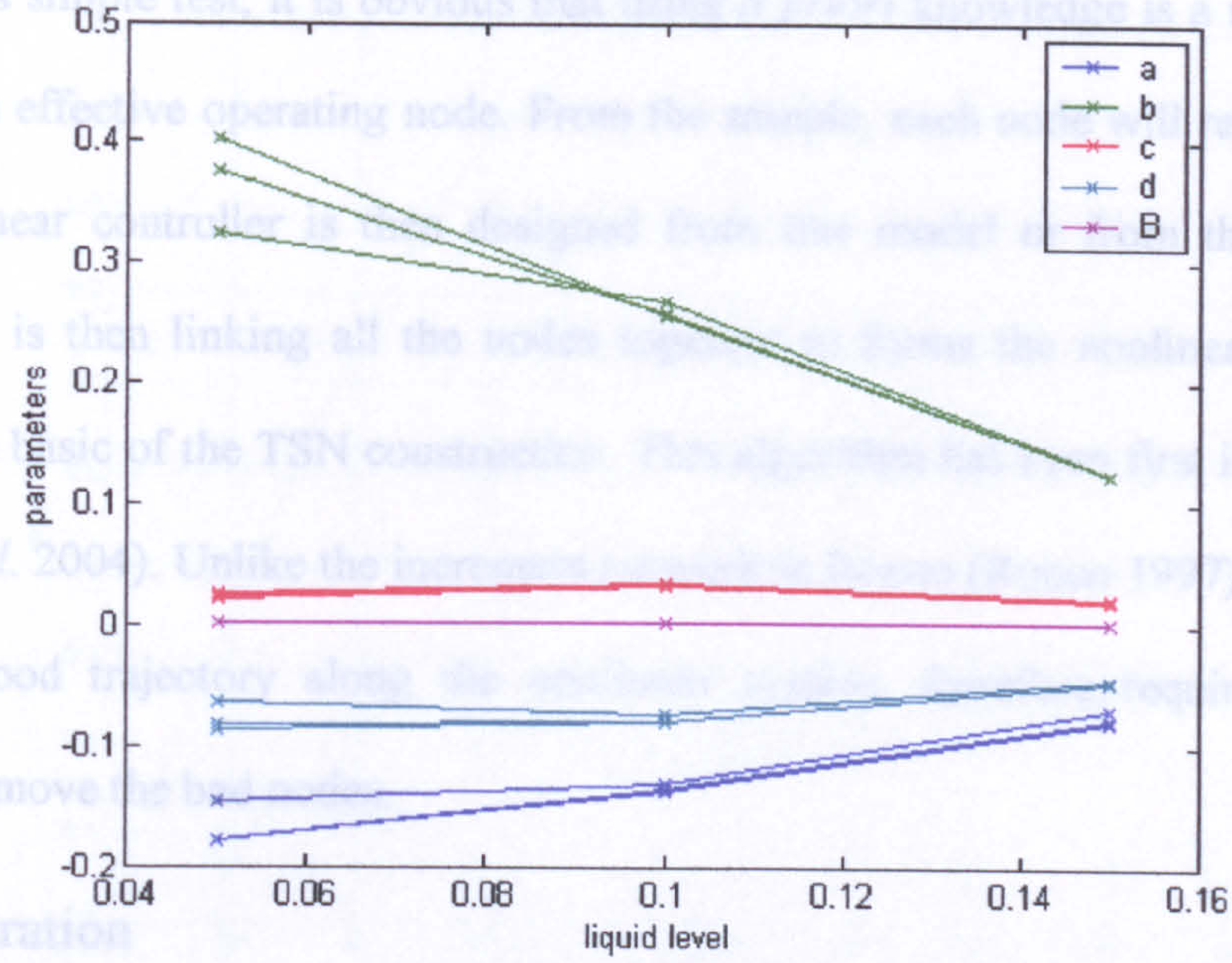

Figure 3.2 Smooth parameters

Without using this algorithm, the resultant parameters from the evolutionary search where the operating nodes are search elements, we have an extreme diversify parameter values (see Figure 3.3).

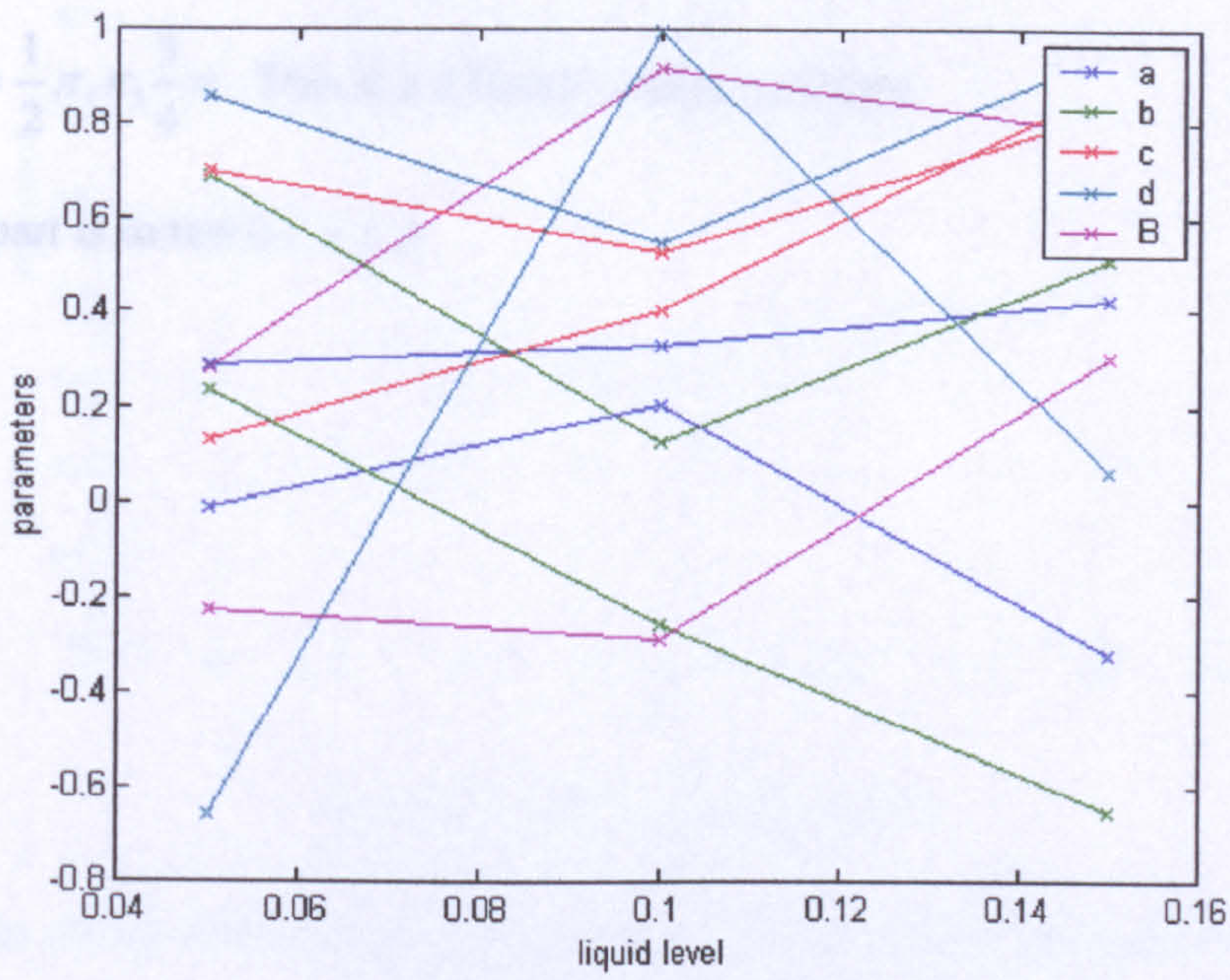

Figure 3.3 Diversify parameters

From this simple test, it is obvious that using *a priori* knowledge is a much safer way in finding an effective operating node. From the sample, each node will represent a linear model. A linear controller is then designed from this model or from the node. Linear interpolation is then linking all the nodes together to forms the nonlinear system. This feature is the basic of the TSN construction. This algorithm has been first introduced in Li *et al.* (Li *et al.* 2004). Unlike the increment network in Ronco (Ronco 1997), this algorithm provide a good trajectory along the nonlinear system, therefore requiring no further pruning to remove the bad nodes.

3.1.2 Illustration

In order to illustrate the capability of the TSN we will consider the control of a nonlinear system. This system is described by the following equation

$$y = \sin(x) \tag{3.5}$$

where sin(x) is the system's nonlinearity and $0 \le x \le 2\pi$.

This function is non monotonic since a change of sign occurs around the operating condition $x = \frac{1}{2}\pi, \pi, \frac{3}{4}\pi$. This is a difficult control problem.

The first part is to test $0 \le x \le \pi$.

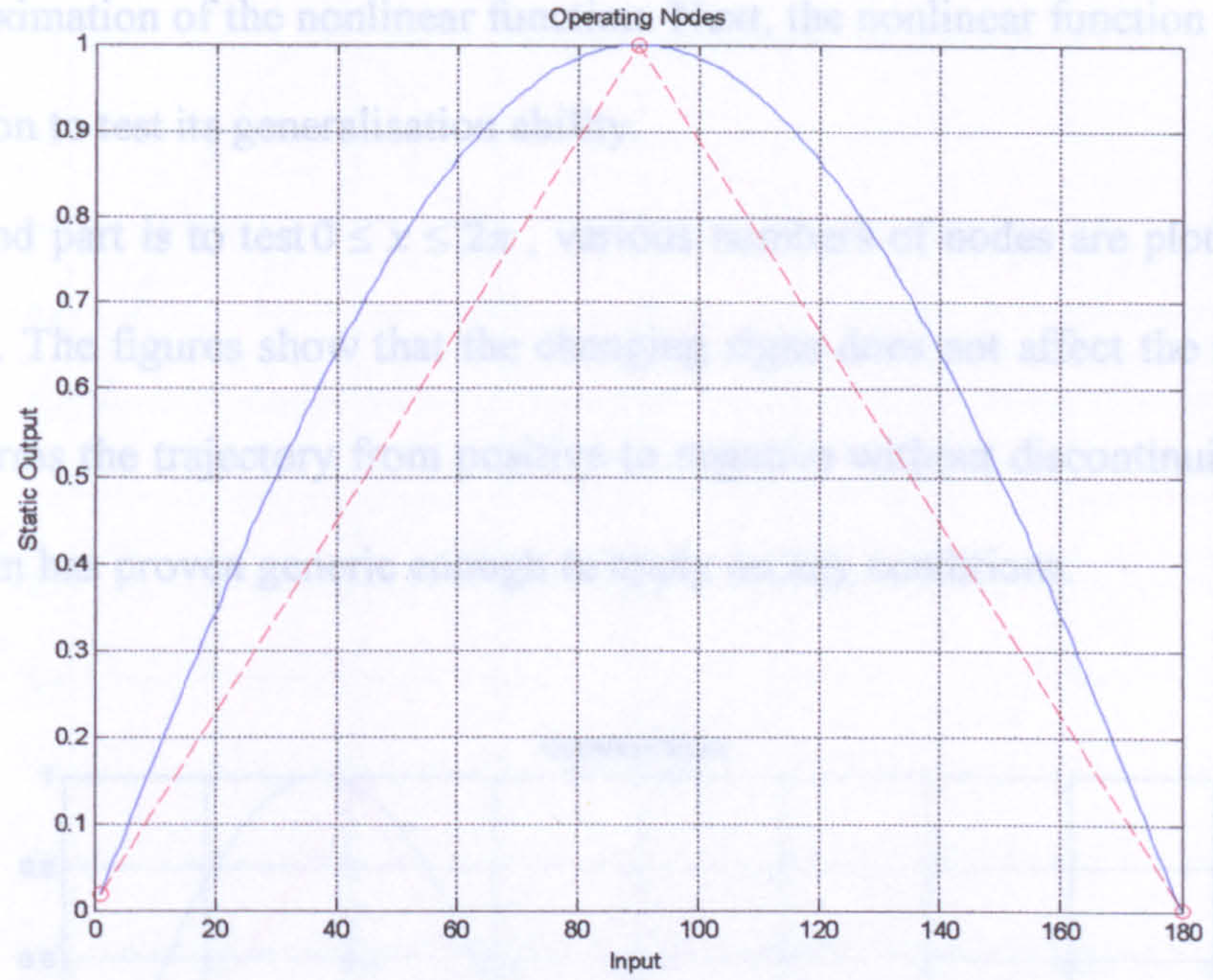

Figure 3.4 Trajectory with 3 nodes

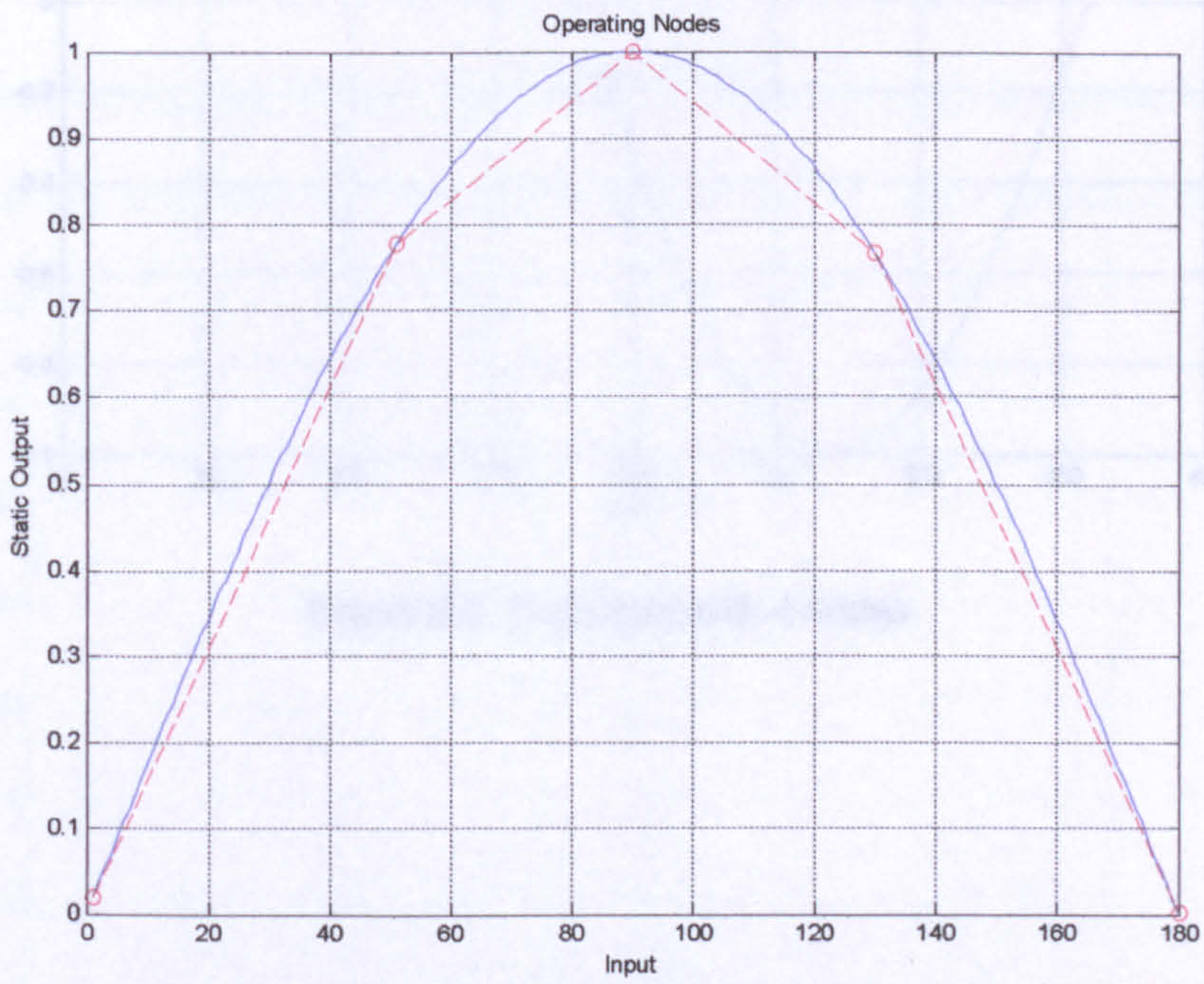

Figure 3.5 Trajectory with 5 nodes

The figures above show the results of linear trajectory along the positive region of the *sin* function. It is clear that the linear trajectory can shapes closely to the curves, providing a good approximation of the nonlinear function. Next, the nonlinear function is extended to negative region to test its generalisation ability.

The second part is to test $0 \leq x \leq 2\pi$, various numbers of nodes are plot in Figure 3.6 to Figure 3.9. The figures show that the changing signs does not affect the algorithm and capable to forms the trajectory from positive to negative without discontinuity. Therefore, such algorithm has proven generic enough to apply on any conditions.

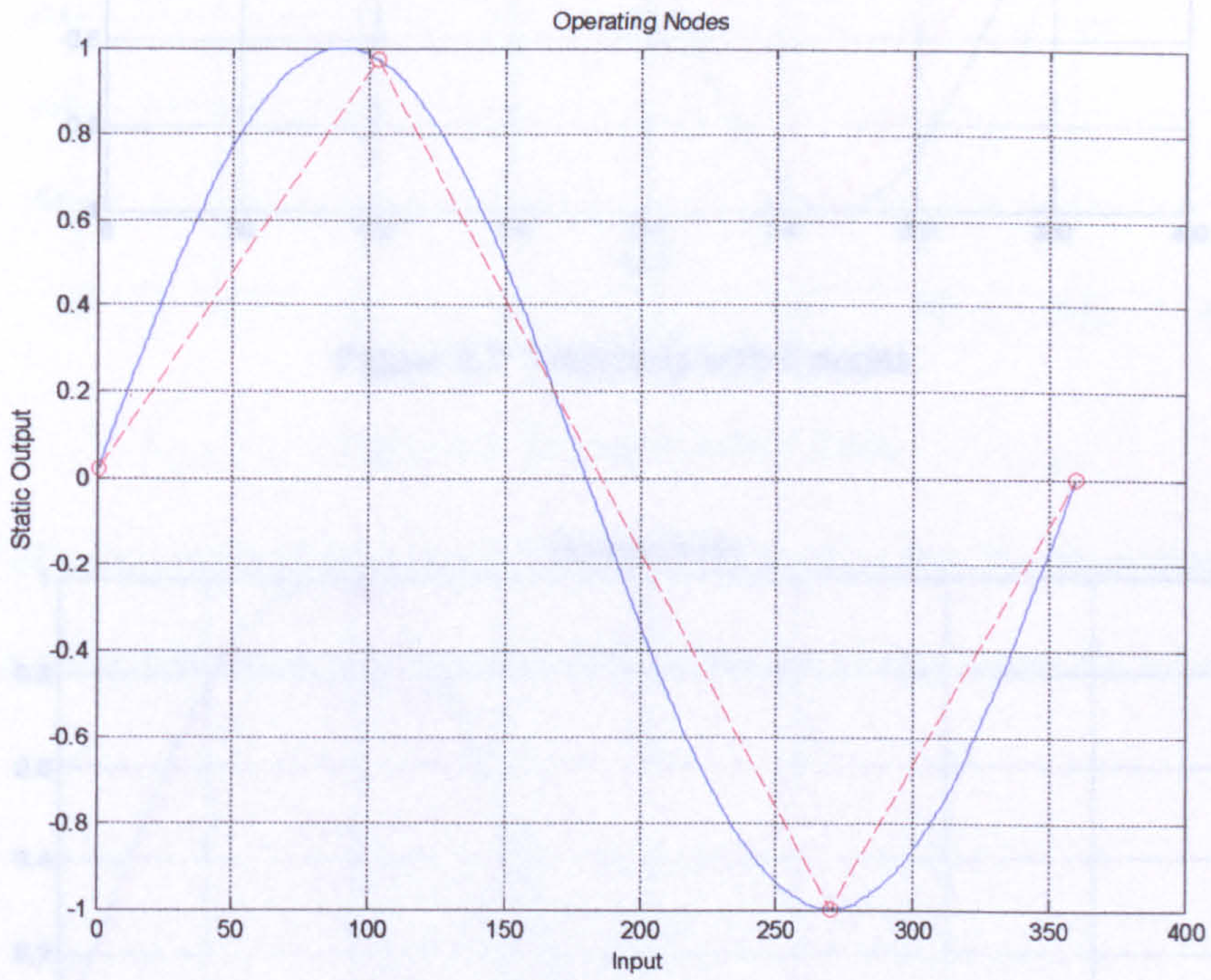

Figure 3.6 Trajectory with 4 nodes

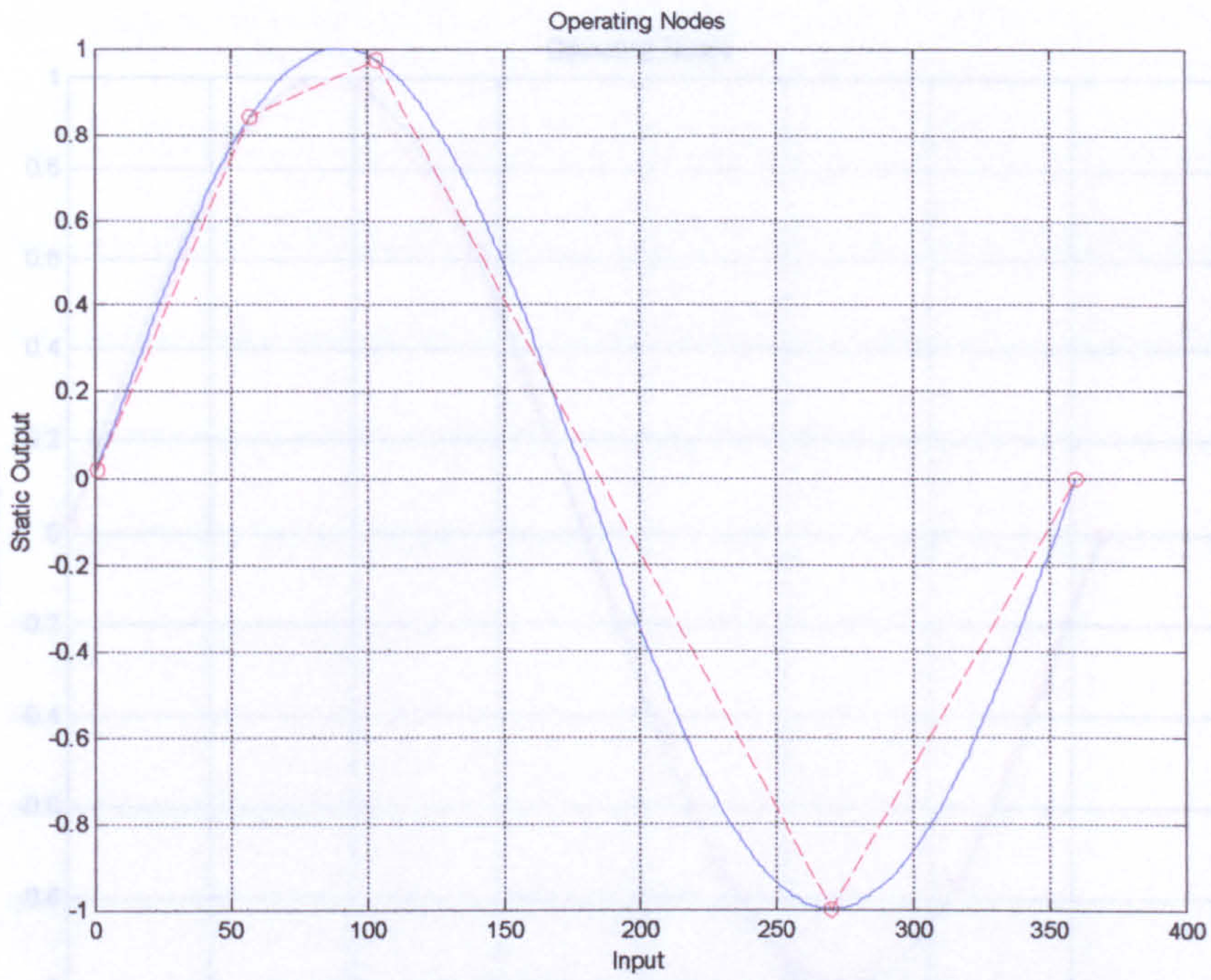

Figure 3.7 Trajectory with 5 nodes

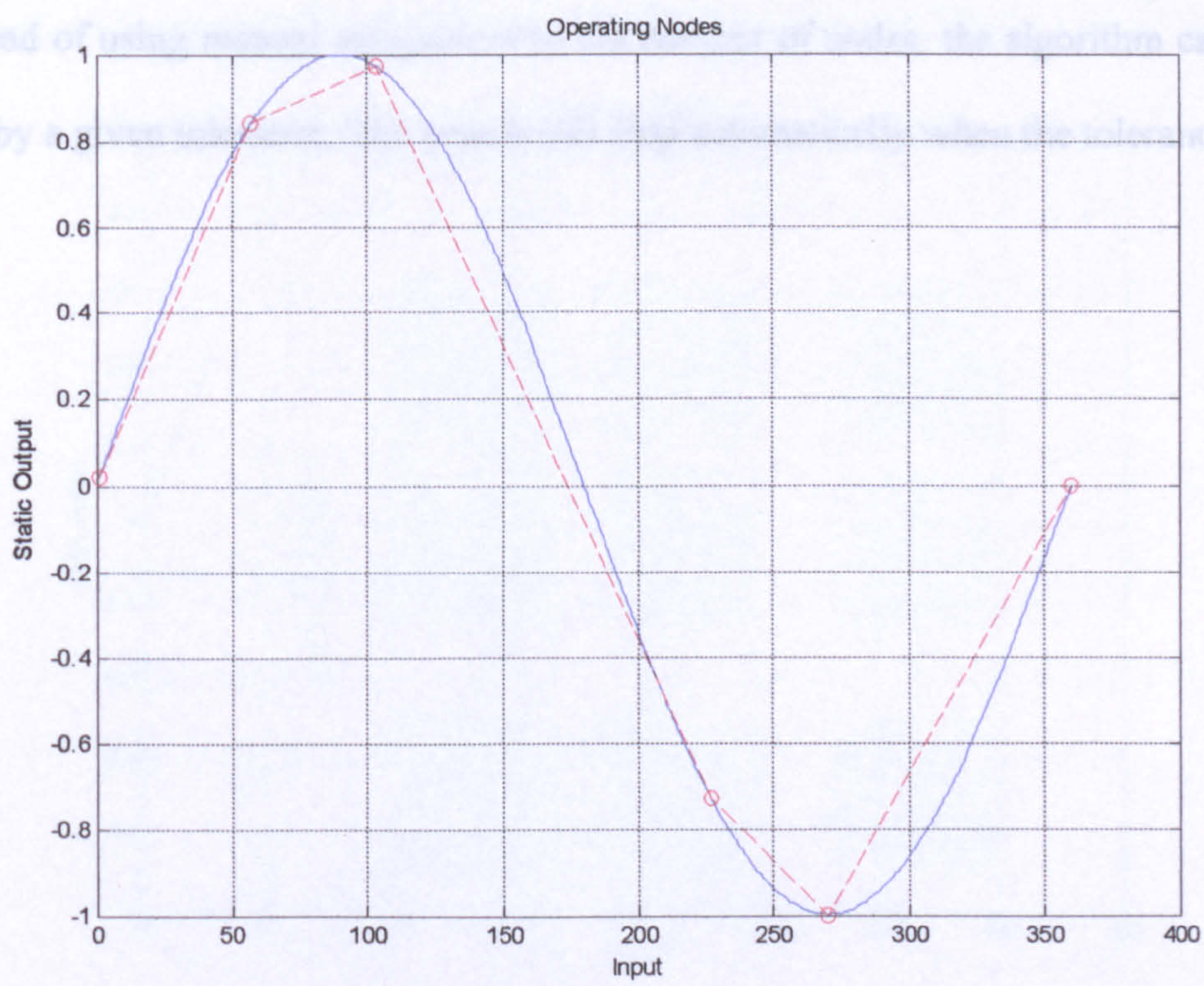

Figure 3.8 Trajectory with 6 nodes

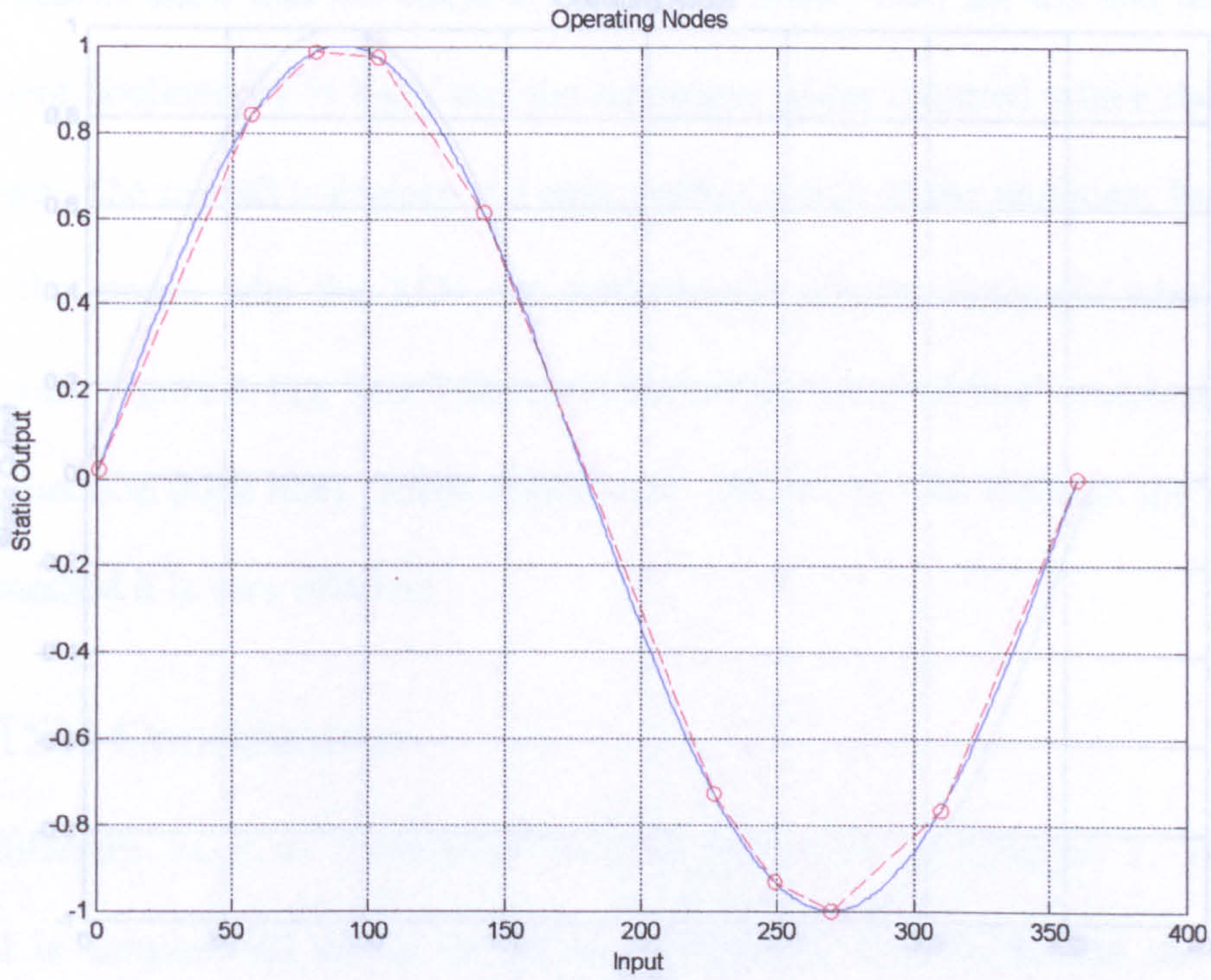

Figure 3.9 Trajectory with 8 nodes

Instead of using manual assignment to the number of nodes, the algorithm can also be defined by a given tolerance. The search will stop automatically when the tolerance meets.

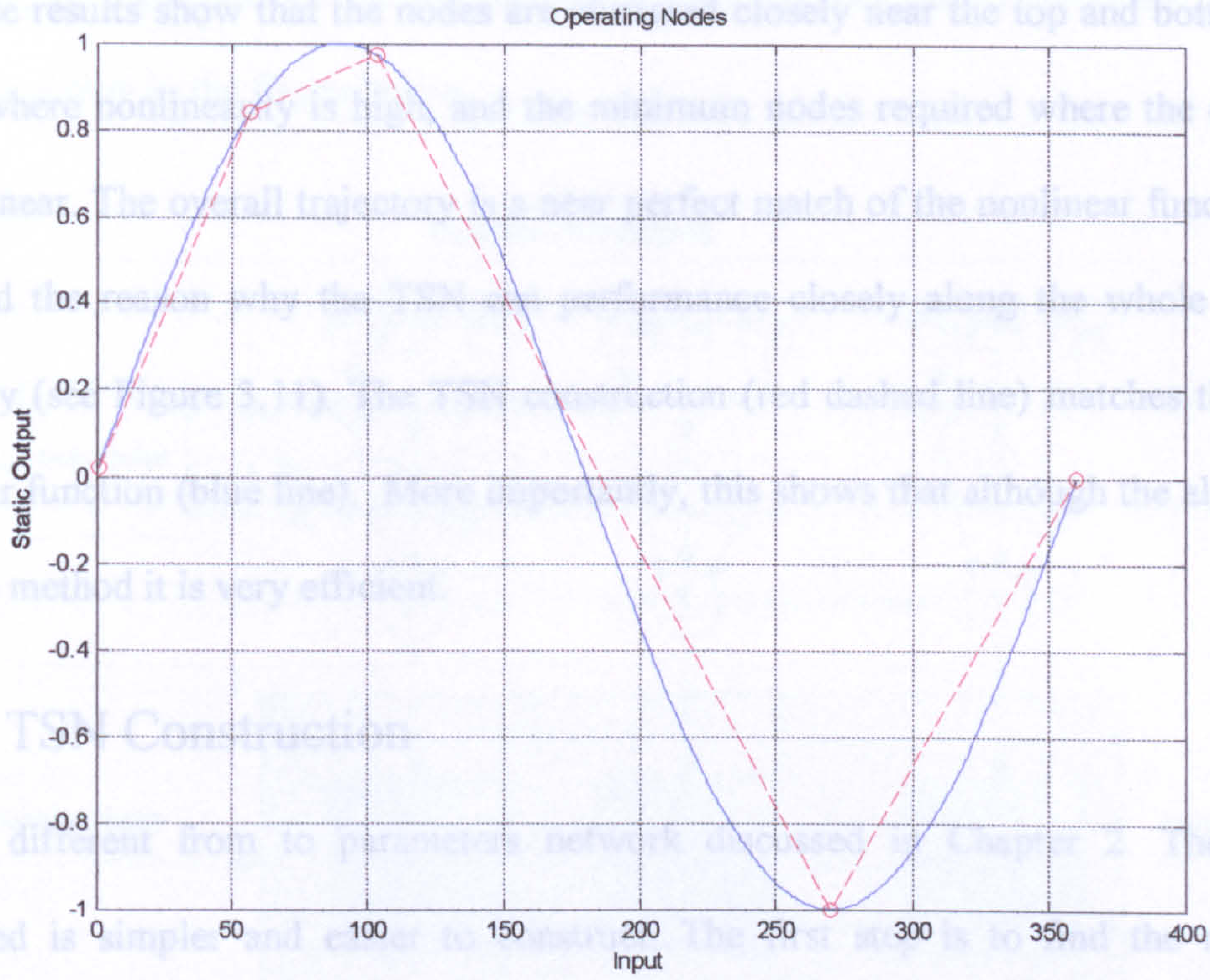

Figure 3.10 normalize tolerance of 0.1. Five nodes found.

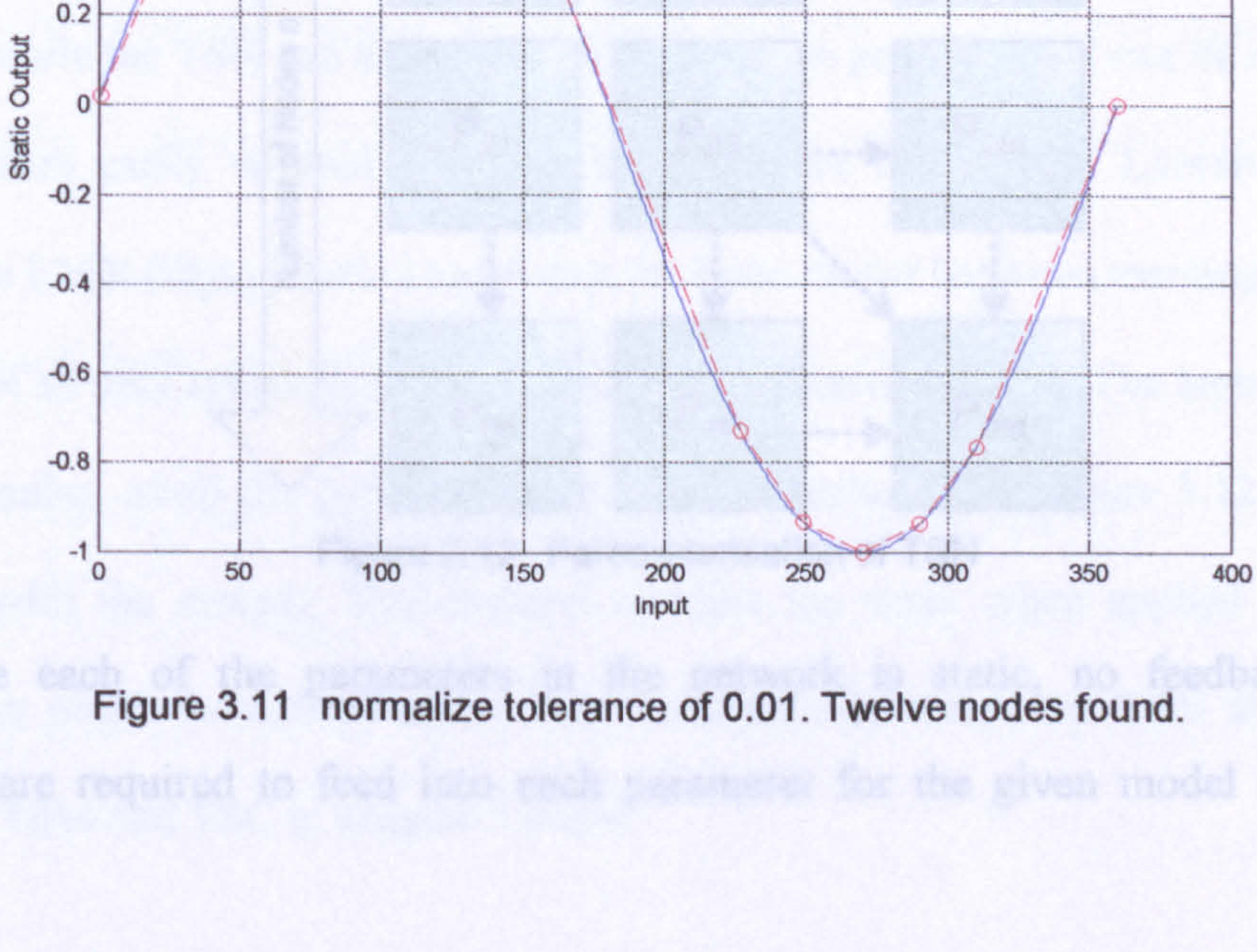

Figure 3.11 normalize tolerance of 0.01. Twelve nodes found.

These results show that the nodes are clustered closely near the top and bottom of the curves where nonlinearity is high, and the minimum nodes required where the curves are nearly linear. The overall trajectory is a near perfect match of the nonlinear function. This is indeed the reason why the TSN can performance closely along the whole nonlinear trajectory (see Figure 3.11). The TSN construction (red dashed line) matches the desired nonlinear function (blue line). More importantly, this shows that although the algorithm is a simple method it is very efficient.

3.2 TSN Construction

This is different from to parameters network discussed in Chapter 2. The network developed is simpler and easier to construct. The first step is to find the number of elements embedded in the given model or controller, then find the number of nodes used in the network. The parameterization of such a network will be distributed as follows:

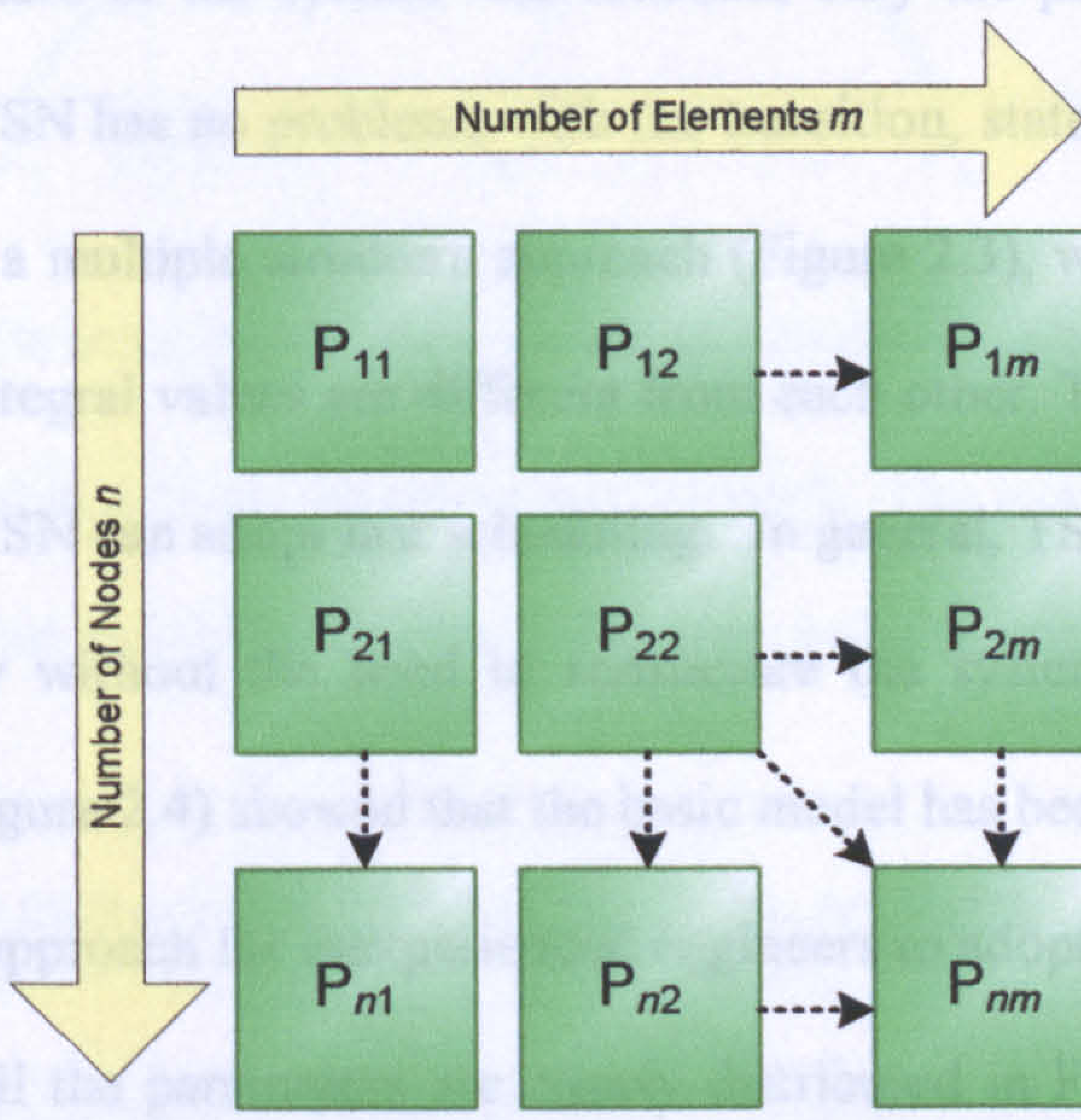

Figure 3.12 Parameterisation of TSN

Because each of the parameters in the network is static, no feedbacks or state transitions are required to feed into each parameter for the given model or controller.

Therefore, by interpolating the parameters in each element, the constructed network in general is shown in the figure below;

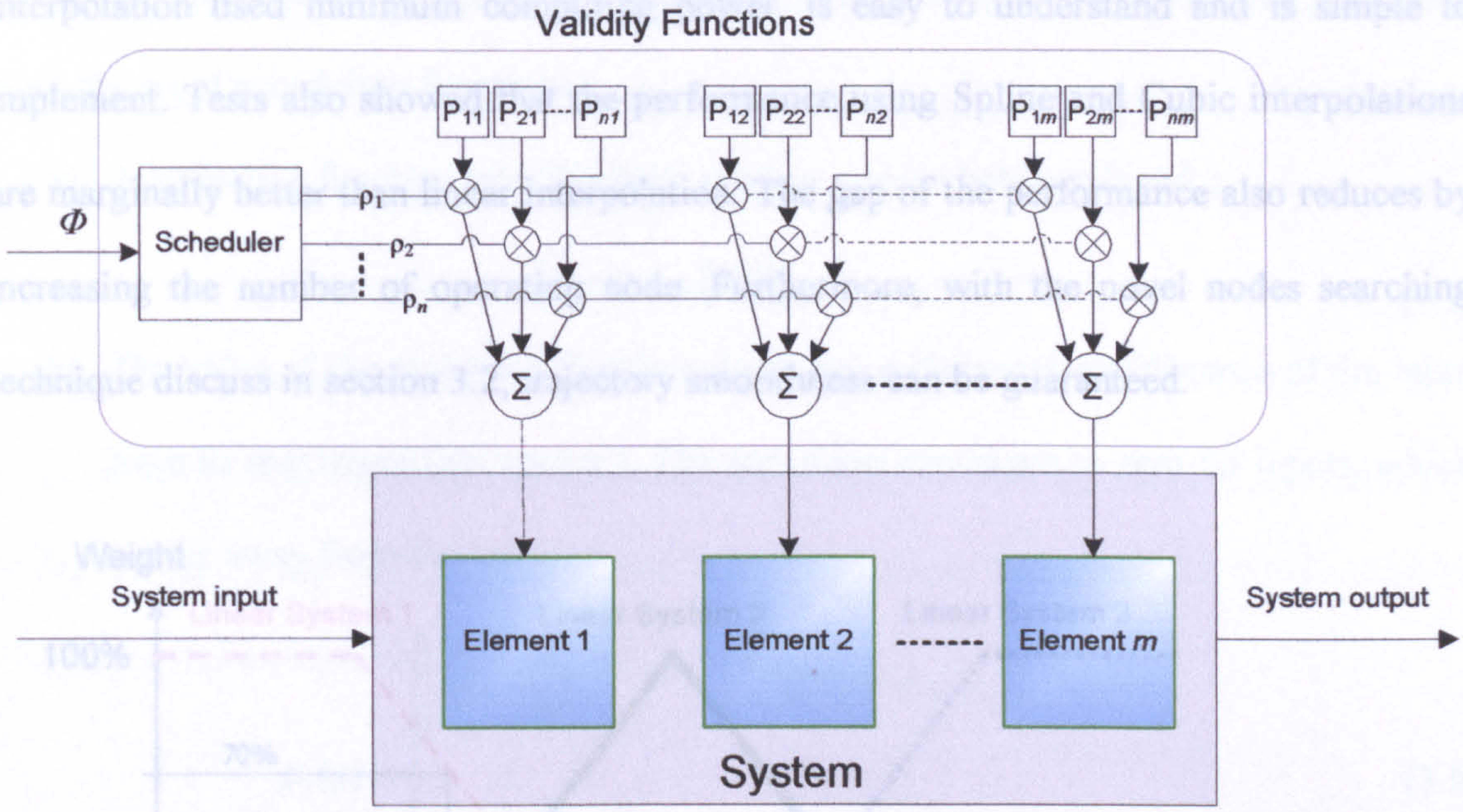

Figure 3.13 Trajectory-Scheduling Network

In TSN, the structure of the system was retained, only the parameters in the system vary. Therefore, the TSN has no problems with the transition, state and integral embedded in the system. Unlike a multiple structure approach (Figure 2.3), within each sub-task, the transition, state and integral values are different from each other. Thus, the scheduling has to be slow while the TSN can adopt fast scheduling. In general, TSN can be applied to any type of system easily without the need to restructure the system. Literature review in chapter 2 on LMN (Figure 2.4) showed that the basic model has been rearranged internally. This was not an easy approach for inexperienced engineers to adopt. The layout of the TSN is much simpler, as all the parameters are clearly distributed in Figure 3.12 and later act externally with the system. This method remains the same when applied on nonlinear modelling or nonlinear controller. To demonstrate its potential, the TSN will be used to construct a TSM and TSC in chapter 5 and 6.

3.2.1 Interpolation

In this work, we will restrict ourselves to the use of linear interpolation. Linear interpolation used minimum computing power, is easy to understand and is simple to implement. Tests also showed that the performance using Spline and Cubic interpolations are marginally better than linear interpolation. The gap of the performance also reduces by increasing the number of operating node. Furthermore, with the novel nodes searching technique discuss in section 3.2, trajectory smoothness can be guaranteed.

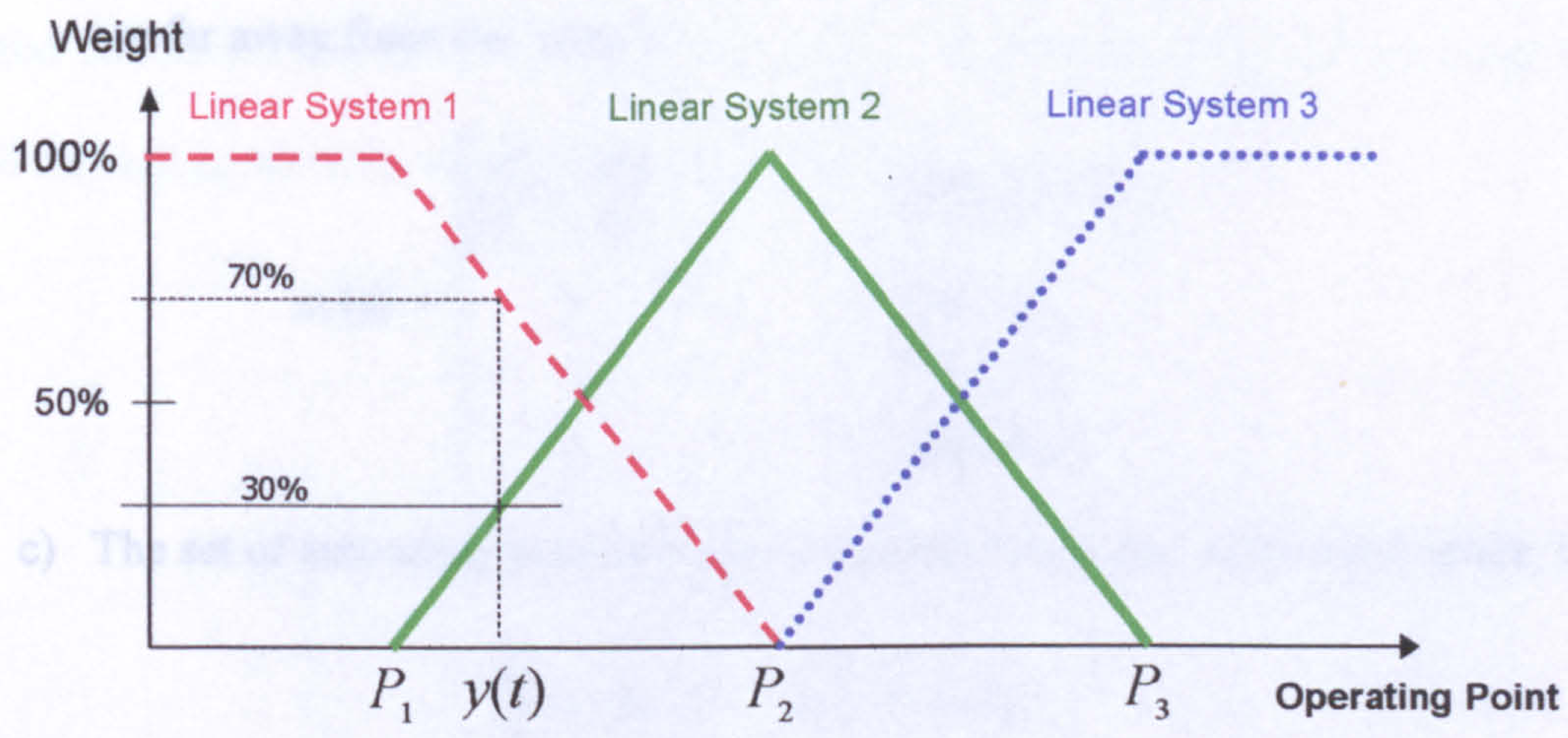

Figure 3.14 A simple linear interpolation scheduling networking for multiple linear systems

The figure illustrates a linear interpolation network. The example shows the sum of 70% of system 1 output and 30% of system 2 output was the total contribution for output $y(t)$.

The general linear interpolation formulation can be defined as:

$$y(t) = \begin{cases} \dfrac{P_{i+1} - y(t)}{P_{i+1} - P_i} \times \hat{y}_i(t) + \dfrac{y(t) - P_i}{P_{i+1} - P_i} \times \hat{y}_{i+1}(t) & if\ P_i \le y(t) \le P_{i+1} \\ \hat{y}_i & if\ y(t) < P_1 \\ \hat{y}_n & if\ y(t) > P_n \end{cases} \quad (3.6)$$

where the outputs $\hat{y}_i(t)$ are interpolated giving a final output $y(t)$ using equation (3.6), P_i is the operating nodes, i=1,...,n-1 and n is the number of operating nodes implemented. To satisfy the approximation described in equation (3.6) we require the following properties from the set of scheduling functions:

a) A activation function transforms its input to a value between 0 and 1:

$$\{\rho_i : \Phi \rightarrow [0,1]\}_{i=1}^{n} \tag{3.7}$$

b) The value of the activation function decreases with increasing distance of the input from its maximum (its 'centre'). The activation converges to zero for inputs, which are far away from the 'centre'.

$$\rho_i(\phi) = \begin{cases} \left|\dfrac{P_{i+1} - \phi}{P_{i+1} - P_i}\right| & \textit{if } P_i \le \phi \le P_{i+1} \\ 1 & \textit{if } \phi < P_1 \\ 1 & \textit{if } \phi > P_n \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

c) The set of activation functions forms a partition of unity of its input space, i.e.

$$\sum_{i=1}^{n} \rho_i(\underline{\phi}) = 1 \quad \forall \underline{\phi} \in \Phi \subset \mathbb{R}^{n_\phi} \tag{3.9}$$

TSN uses one general function for all systems, either modelling or control. Such network can also be described as an LPV system.

$$E_j(\underline{\phi}) = \sum_{i=1}^{n} \rho_{ij}(\underline{\phi}) E_{ij}, \tag{3.10}$$

The parameters of (3.10) depend only on the activator $\underline{\phi}$. This type of parameters varying structure is depicted in Figure 3.13. The goodness of such an approach is that it is capable of reacting faster to scheduling variables, therefore overcoming the deficiency in structure networking. Despite having to break down the entities in the structure into each schedulable parameter, this network is much more robust and easy to manipulate.

3.2.2 Hybrid Activation Variables

In chapter 2, we learned that most of the activations are singular; they are either the input or the output of a given process. The weakness of using only one activation variable is that only one variable is monitored at any one time. Therefore, the response can be slow and unable to change according to the variables that are not in used.

By combining the input and output variables, this novel activation method provides the intermediate performance of the two. This method eliminates the delayed and zero response using only the output as the scheduler during the start-up when most values and initial conditions are zeros. On the other hand, a small percentage of the input variables will be used to increase the speed of the activation.

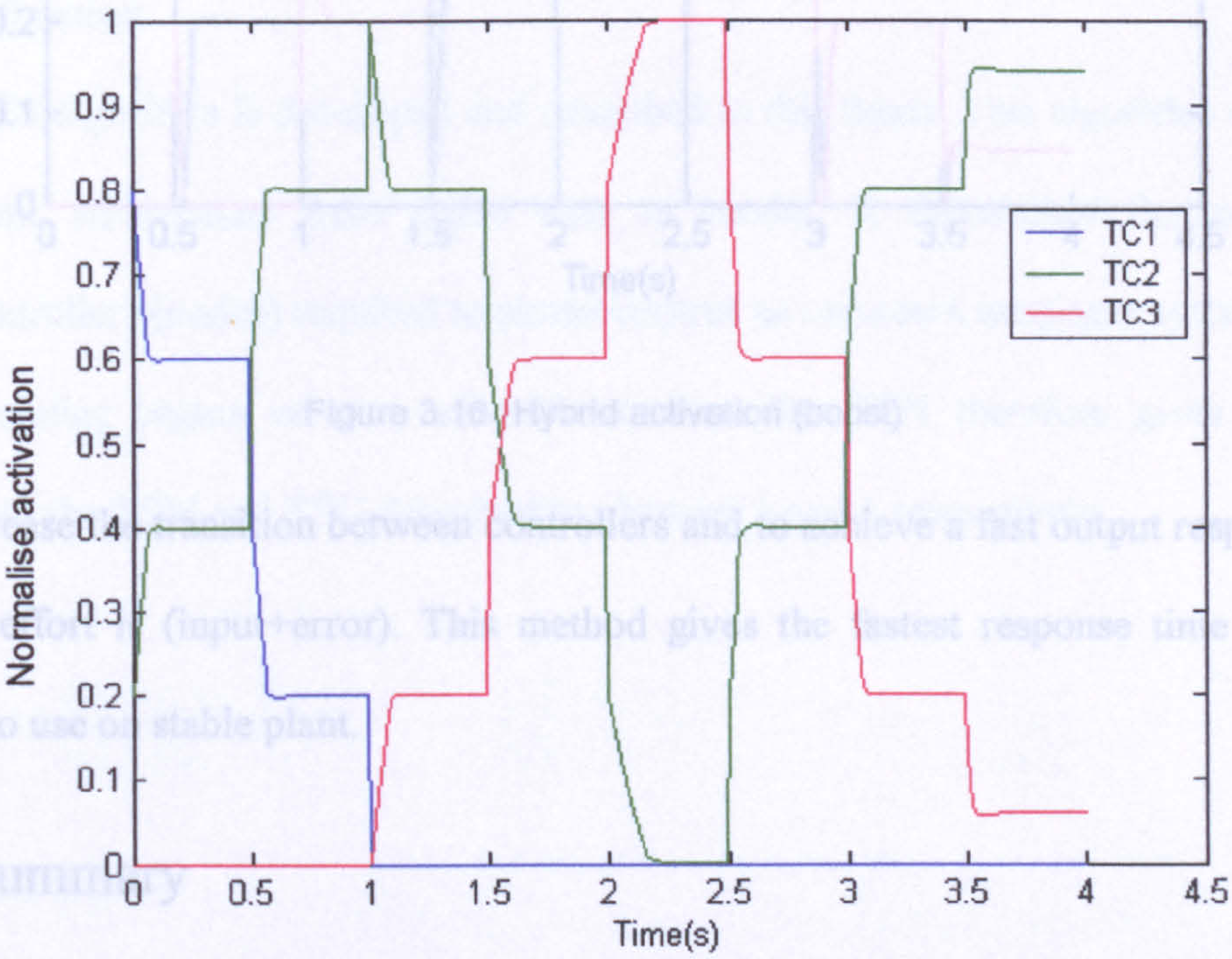

Figure 3.15 Hybrid activation (intermediate)

This method is the compensation between input and output activation. The activation is (intput+output)/2 or the so called intermediate activation. The performance is between the two formal methods.

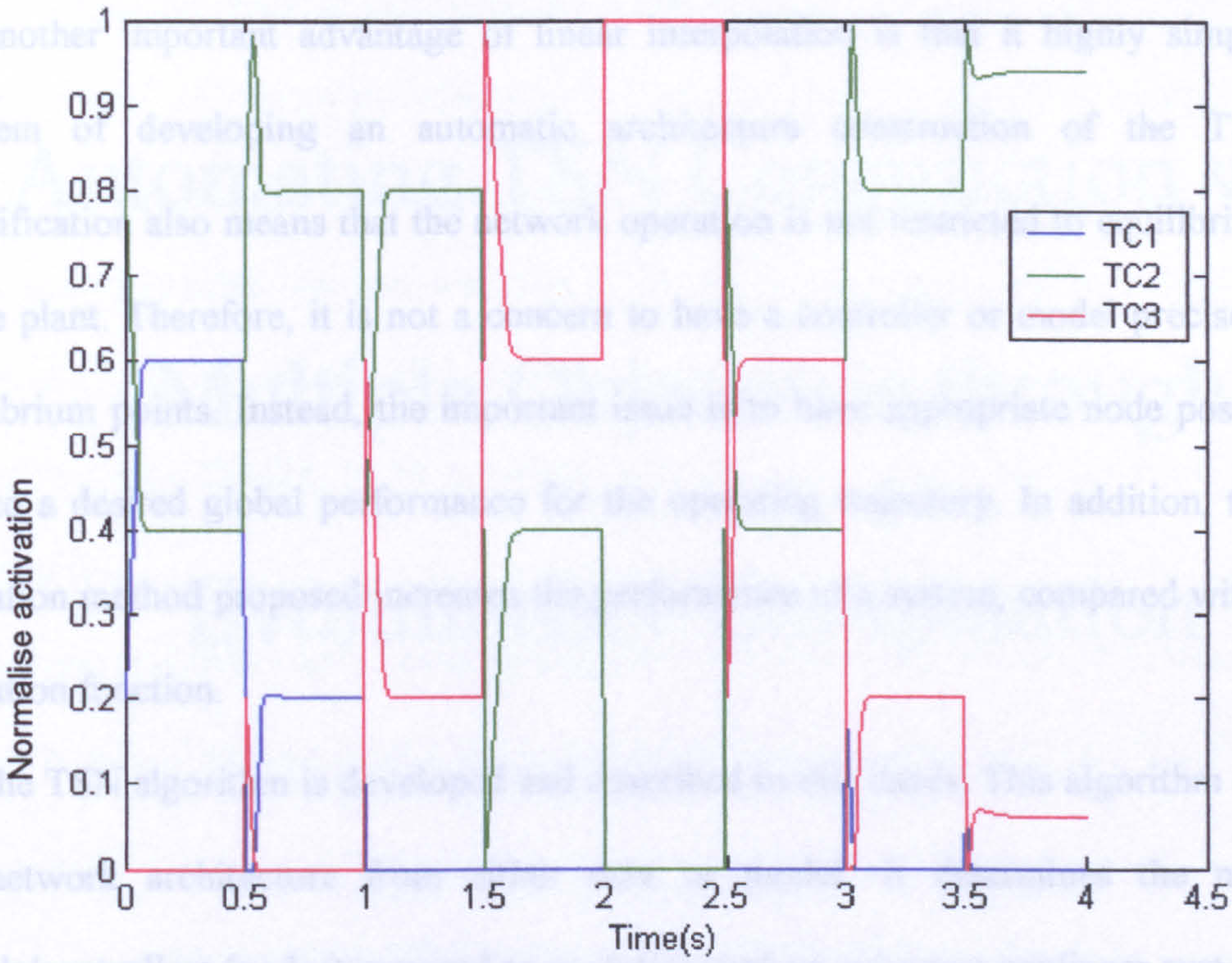

Figure 3.16 Hybrid activation (boost)

To increase the transition between controllers and to achieve a fast output response, the activation effort is (input+error). This method gives the fastest response time but only advisable to use on stable plant.

3.3 Summary

The TSN is a modified version of the LMN. The main difference is that the TSN is achieved through a static model of the process whereas the LMN requires further linearisation of the nodes found. There are several important advantages arising from this simple networking. Among them is the facility to determine the neighbourhood of the operating condition using the static model. This leads to better interpolation capability of the network as well as a straightforward understanding of the network activity. The activation of no more than two models/controllers each time also has the advantage of implying very few computations that made it possible for real time software controller.

Another important advantage of linear interpolation is that it highly simplifies the problem of developing an automatic architecture construction of the TSN. This simplification also means that the network operation is not restricted to equilibrium points of the plant. Therefore, it is not a concern to have a controller or model precisely on the equilibrium points. Instead, the important issue is to have appropriate node positions that lead to a desired global performance for the operating trajectory. In addition, the hybrid activation method proposed increases the performance of a system, compared with a single activation function.

The TSN algorithm is developed and described in this thesis. This algorithm constructs the network architecture from either data or model. It determines the number of models/controllers (nodes) required to model/control an unknown nonlinear system as well as the operating region of each model/controller. The TSN therefore gives complete autonomy to the TSM and TSC described in chapters 5 and 6 respectively.

Chapter 4

Automating TSN Construction with Multiple Objectives through Evolutionary Computation

4.1 Control System Design Objectives

An evolution algorithm provides globally optimal solutions to engineering design problems by emulating natural evolution. Therefore, the performance index (cost function) is used for evaluating the systems performance. A performance index is a number that indicates the “goodness” of system performance. A control system is considered optimal when the values of the parameters are chosen so that the selected performance index is minimum or maximum depending on the situation. The optimum values depend directly on the performance index selected. In addition, a performance index must yield a single positive number or zero. Finally, to be practical, a performance index must be easily computed analytically.

Consider a generic unity negative feedback control system of a given plant $G(s)$ with a controller $H(s)$. Refer to Figure 4.1 for notations. Without loss of generality, for the case $F(s) = 1$,

$$E(s) = R(s) - Y(s) = \frac{1}{1 + H(s)G(s)}\left[R(s) - G(s)D(s)\right] \quad (4.1)$$

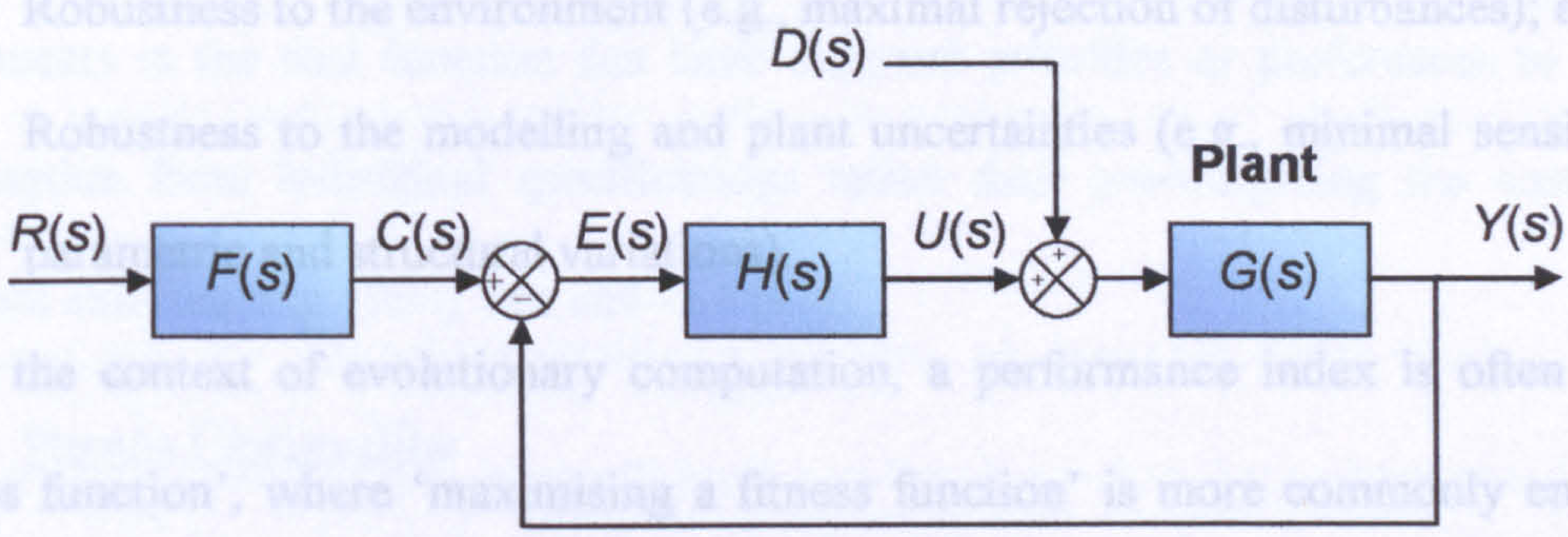

Figure 4.1 A feedback control system with a model-following command

where $D(s)$ is usually a disturbance signal, which may be coloured and also be modelled to include the plant uncertainty. The ultimate objective of a control system design is hence to find an $H(s)$ such that

$$E(s) = 0, \quad \forall s, D(s) \tag{4.2}$$

or

$$e(t) = L^{-1}\{E(s)\} = 0, \quad \forall t, d(t) \tag{4.3}$$

This ultimate objective means that Condition (4.2) or (4.3) need to be satisfied regardless of plant uncertainties, which is impossible in a practical control system design.

Hence, a performance index, $J : \mathbf{R}^n \to \mathbf{R}^+$, is often used to measure *how close* the above ultimate objective is met, where n is the number of parameters that needs to be determined in the design. For this, performance indices and specifications need to reflect the following qualitative requirements (Kashiwagi 1983; Levine 1996; Li *et al.* 1995b).

- Good relative stability (e.g., good gain and phase margins);
- Excellent steady-state accuracy (e.g., minimal or no steady-state errors);
- Excellent transient response (e.g., minimal rise-time, settling-time, overshoots and undershoots);
- Robustness to the environment (e.g., maximal rejection of disturbances); and
- Robustness to the modelling and plant uncertainties (e.g., minimal sensitivities to parametric and structural variations).

In the context of evolutionary computation, a performance index is often termed a 'fitness function', where 'maximising a fitness function' is more commonly encountered than 'minimising a cost function', although an evolutionary algorithm (EA) can do both maximisation and minimisation in one process. For convenience, a cost function can be converted easily into a fitness function by, for example, f: $\mathbf{R}^+ \to \mathbf{R}^+$,

$$f_i(H) = \frac{1}{1 + J_i(H)} \in (0,1], \tag{4.4}$$

where i is the number of functions used. Special cases of the performance index J are two commonly used indices as listed below.

The *Integral of Absolute Error (IAE)* (Levine 1996):

$$J_{\text{IAE}} = \sum_t |e(t)| = \|e(t)\|_1 \tag{4.5}$$

The *Integral of Square Error (ISE)* (Levine 1996):

$$J_{\text{ISE}} = \sum_t e^2(t) = \|e(t)\|_2^2 \tag{4.6}$$

4.2 Multi-Objective Optimisation

GA have been recognised to be well-suited to multi-objective optimisation (Fonseca and Fleming 1993; Goldberg 1989). Unlike conventional methods that linearly combine multiple attributes to form a composite scalar objective function, a multi-objective evolutionary algorithm (MOEA) (Li *et al.* 1996) incorporates the concept of Pareto's domination to evolve a family of non-dominated solutions at multiple points along the Pareto optimal frontier simultaneously and efficiently. By combining the Pareto dominance with partial preference information in the form of a priority vector, each of the individual components in the cost function can have different priorities or preferences to guide the optimisation from individual specifications rather than pre-weighting the cost function (Fonseca and Fleming 1993; Tan and Li 1997).

4.2.1 Pareto Optimality

Pareto-based fitness assignment was first proposed by Goldberg (Goldberg 1989), as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. The method assigns rank 1 to the non-dominated individuals and removing them from contention, then finding a new set of non-dominated individuals, ranked 2, and so forth. Tournament selection based on Pareto dominance was later proposed by Horn *et al.* (Horn *et al.* 1994). In addition to the individuals competing in each tournament, a number of other individuals in the population were used to help determine whether the competitors were dominant or not. Sharing is used to determine the selection if both competitors were either dominant or non-dominant.

4.2.2 Goal and Priority

Goal and priority information is often naturally available from the problem formulation, although they are not necessarily utilised in a strict sense. The cost assignment method described earlier can be modified to accommodate goal information (Fonseca and Fleming 1993; Tan and Li 1997) in a similar way to that used by conventional goal attainment method (Grace 1992). The method of goal attainment entails the construction of a set of *goal* values for the objective functions.

4.3 Evolutionary Search Algorithms

Evolutionary computation based design techniques make use of simulation results just like a human designer, and 'intelligently' transform the simulation problem into its reverse problem of design. A multiple coefficient design space characterised by a performance index is usually multi-modal, which is hard to accommodate by traditional optimisation methods.

Emulating the Darwinian-Wallace principle of 'survival-of-the-fittest' in natural selection and genetics have led to today's success in Evolutionary computation (Li *et al.* 1996). Evolutionary algorithms (EA) (Fogel 1995) such as Genetic Algorithms (GA) (Goldberg 1989), have been found to be very effective and efficient in searching a poorly understood, irregular and complex space for optimisation and machine learning. Such an algorithm evaluates performances of candidate solutions at multiple points simultaneously and thus efficiently approaches the global optimum. As summarised in Figure 4.2, an EA encodes a candidate design in an artificial 'chromosome' and then vary these chromosomes for improvements, generation by generation, in a similar way to natural evolution. This is in effect a parallel search and machine learning process, in which the EA makes use of past trial information in a similarly intelligent manner to human designers. The EA can start designs from the application engineer's existing library or from an initial population of random candidates. A number of automatically 'evolved' top-performing candidates will finally merge as optimal designs.

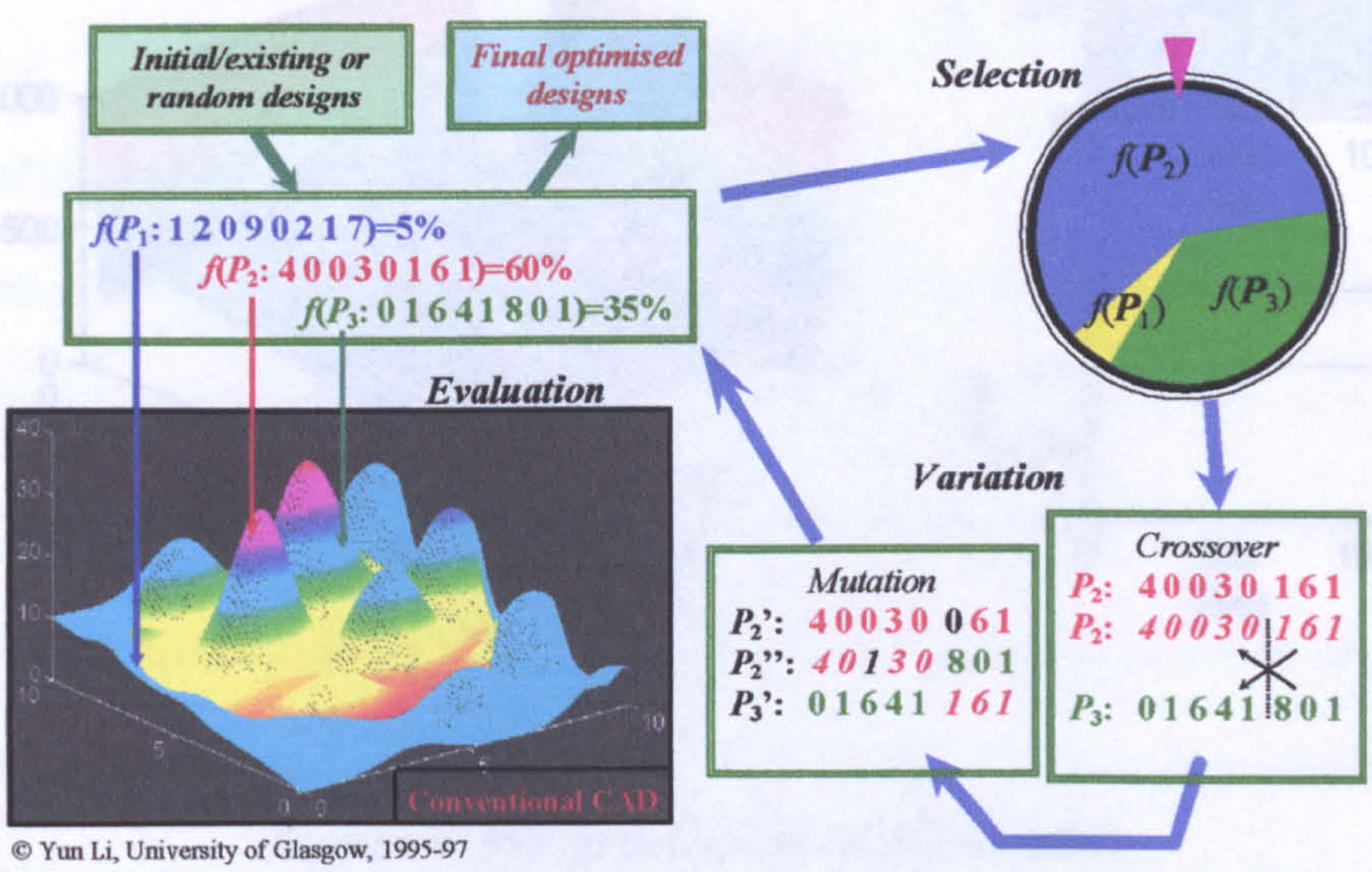

Figure 4.2 Computer-automated design by artificial evolution

Evolutionary computation can search multi-objective, globally optimised solutions to many practical engineering problems that cannot be solved by conventional means. Its unique searching and adaptive learning powers have facilitated design automation, meeting multiple design objectives, offering design quality improved beyond the present performance bounds, and reducing design cycle and time-to-market dramatically. A conventional CACSD package that provides simulation results, taking into account actuator saturation, is used to evaluate the performance of candidate controllers in terms of plant outputs, close-loop errors and control signal provision. Artificial evolution then enables CACSD to become CAutoCSD. By trading off precision slightly using nondeterministic adjustments, the EA exponentially reduces the search time compared with exhaustive search and thus provides much improved tractability and efficiently in design automation (Li *et al*. 1995a; Li and Haeussler 1996; Ng 1995; Tan 1997).

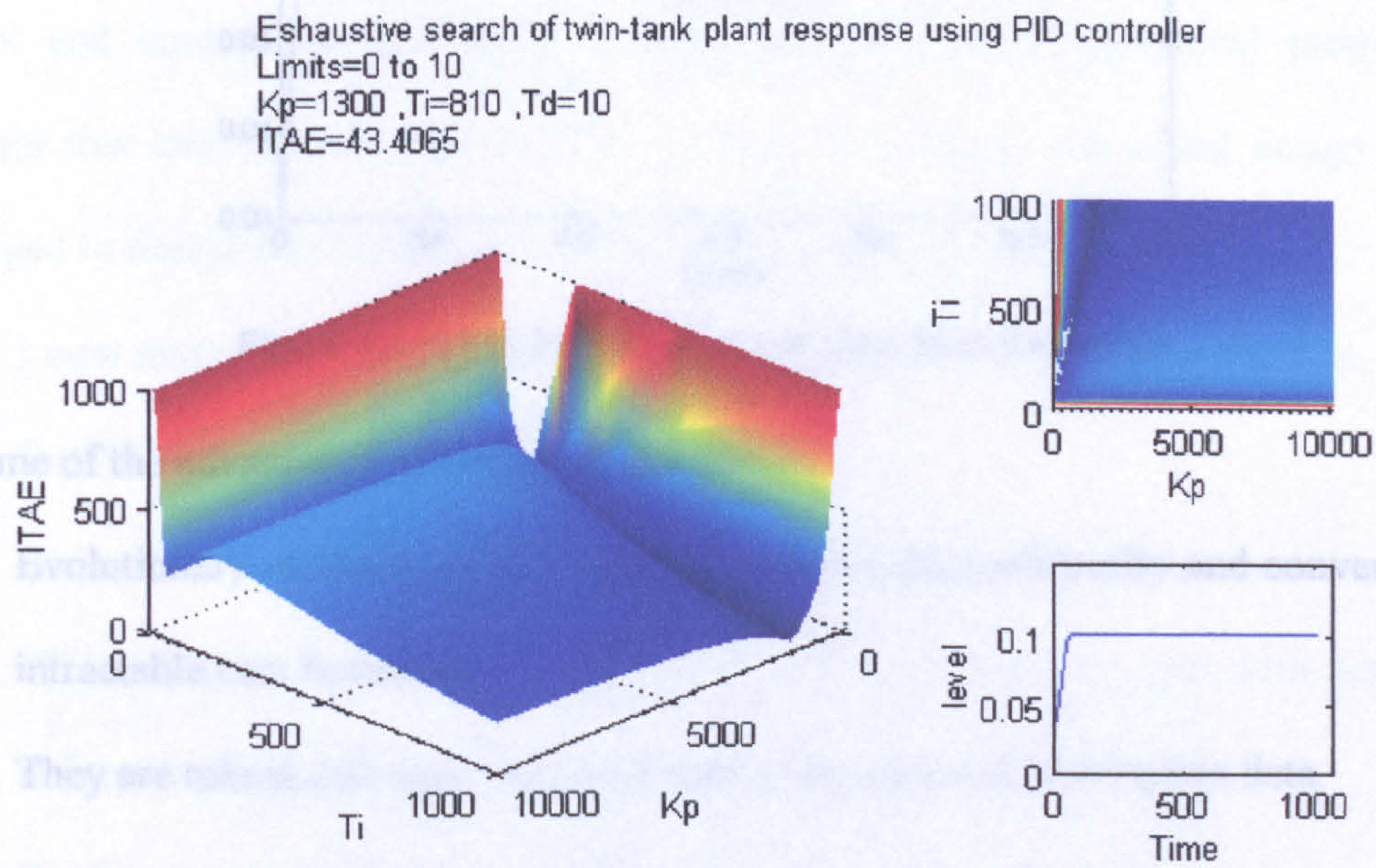

Figure 4.3 Exhaustive search of PID controller

Illustration in Figure 4.3 show that exhaustive search method took 1.5 hrs to find an optimum PID controller while the evolutionary method (Figure 4.4) took approximately 1 minute to find the best performance matching the exhaustive search method.

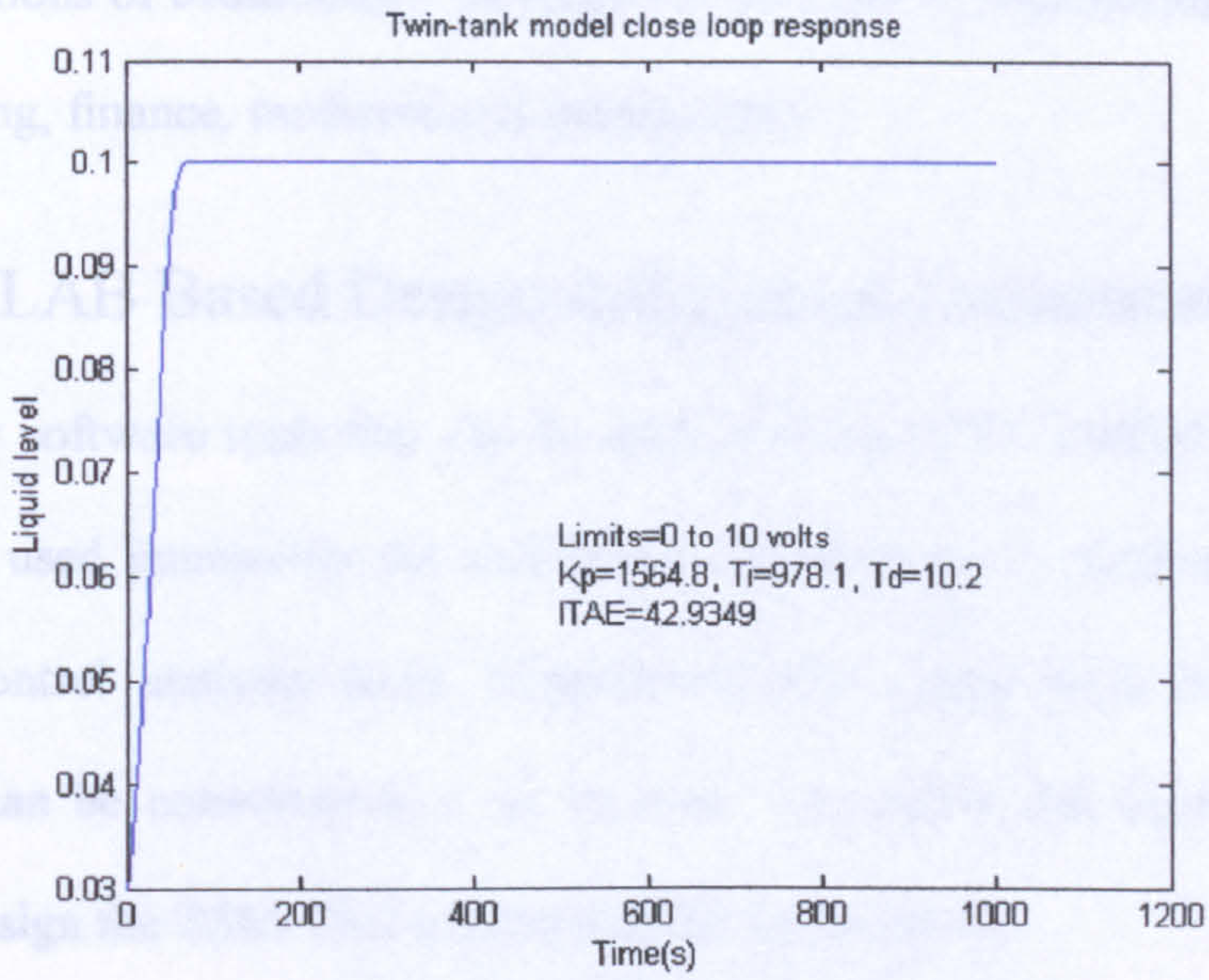

Figure 4.4 Close loop response of plant from EA search

Some of the advantages of EA are as follow:

- Evolutionary methods offer the ability to deal with nonlinearity and conventionally intractable cost functions.
- They are robust and cope well with noisy, inaccurate or incomplete data.
- They can provide solutions to problems that were previously out of range problems that cannot be solved by analytical mathematical techniques, which involve so many variables that other methods would take too long to solve them, or that have frequently and unpredictably shifting goals.
- They are modular and therefore portable; because the evolutionary mechanism is separated from the problem representation they can be transferred from problem to problem.
- They provide an extremely open and flexible approach to design, allowing arbitrary constraints, simultaneous multiple objectives and the mixing of continuous and discrete parameters.

The applications of evolutionary methods are not only in engineering; it is also widely used in marketing, finance, medicine and management.

4.4 MATLAB Based Design Automation Framework for TSN

There are many software tools that can be used to design TSN control system. Of these, MATLAB was used intensively for simulation and analysis, it came with a rich control toolbox and control analysis tools. Easy-to-use and simple high level programming language that can be converted to C in no time. The MATLAB based design package developed to design the TSM-TSC has many build-in functions:

- Local search, EA and MOEA for model identification, TSM identification, tuning for PID and TSC.
- Come with 5 sample processes, twin tank, Mitsubishi chemical/process, VenDeVu, PH process or custom defines (transfer functions/state-space), also with custom define controller (transfer functions).
- Finding static model and operating nodes.
- Setting for noise, load disturbance, and limits.
- Setting for MOEA and EA
- Classical analyses enable; Bode, Nyquist and Root locus.
- Save/load enables

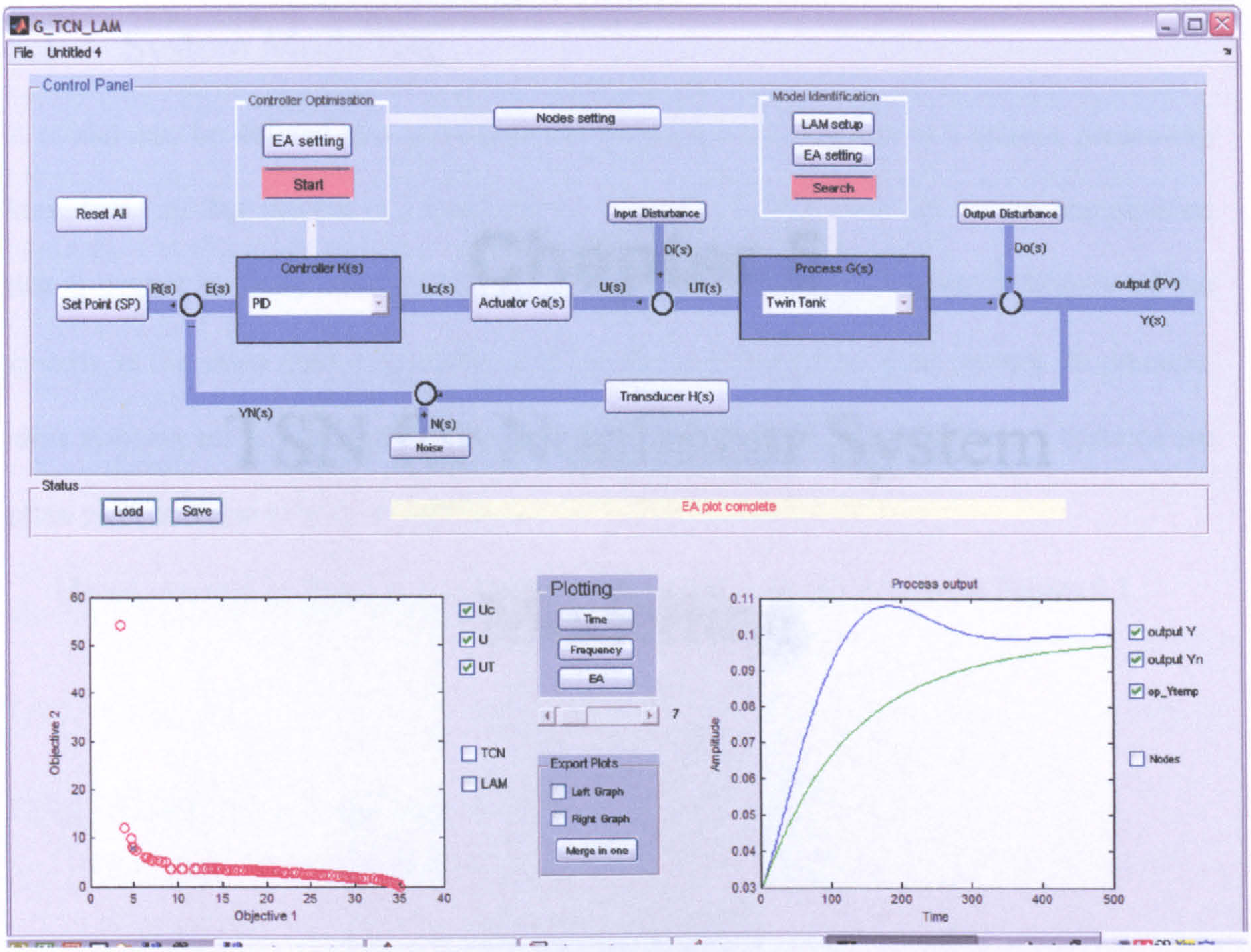

Figure 4.5 MATLAB TSC-TSM design and analysis package.

Chapter 5

TSN for Nonlinear System Modelling

5.1 System Modelling

A model may be defined as a representation of the essential aspects of a system, presenting knowledge of that system in a usable form. A useful model must not be too complicated that it cannot be understood and therefore unsuitable for predicting the behaviour of the system, at the same time it has to be accurate enough to represent the system. In practice, most systems are nonlinear with distributed parameters; linear models for such systems are often used because of their simplicity.

Hence, we will be dealing in this study, with the system represented in Figure 5.1.

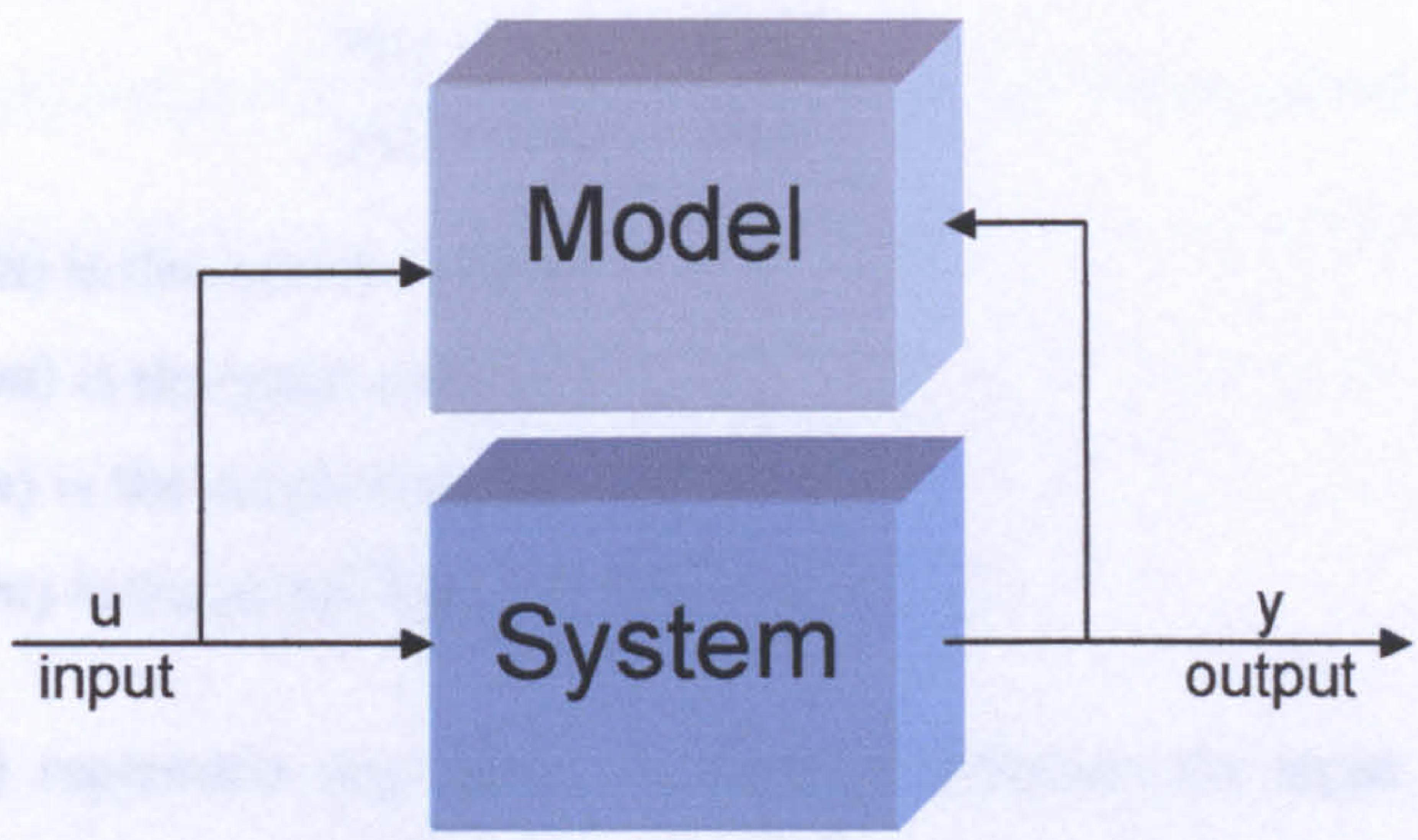

Figure 5.1 Block diagram of a single input single output system

Since system modelling appears to be primordial for the design of controllers, the first concern of this chapter is the determination of a general model for nonlinear systems. We will then describe the method used during this research to determine the values of the parameters of a model. Having described the model, and how to estimate its parameters the design method so called the 'Trajectory-Scheduling Model' (TSM) will be presented. This modelling method will be illustrated according to a couple of examples.

5.2 Trajectory-Scheduling Models

Based upon the LAM, this simple method has extended to highly nonlinear modelling network using the general structure of the TSN introduced in chapter 3. Using the TSN structure with the LAM as linear model blocks, the TSM is then formed.

Given a system of,

m inputs,

n states,

r outputs ,

the full state space system is given by,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}, \tag{5.1}$$

where A (size $n\times n$) is the system matrix,

B (size $n\times m$) is the input matrix,

C (size $r\times n$) is the output matrix,

D (size $r\times m$) is the direct feed through matrix.

The matrix D represents any direct connections between the input and the output. However, in many simple cases, the D matrix is zero.

The rewriting of state variable equations in vector-matrix form will occur often in state space work, the step of identifying the number of states (n), inputs (m) and outputs (r) automatically sets up the size of the ABCD matrices to be filled:

$$\begin{aligned}\dot{x}(t) &= A_{n\times n}x(t) + B_{n\times m}u(t) \\ y(t) &= C_{r\times n}x(t) + D_{r\times m}u(t)\end{aligned} \tag{5.2}$$

As we are only interested in single input single output 2 states model,

thus, n=2, m=1 and r=1. Therefore

$$\text{A: } (2\times 2);\ \text{B: } (2\times 1);\ \text{C: } (1\times 2);\ \text{D: } (1\times 1). \tag{5.3}$$

In the time domain, the general state variable diagram then looks like Figure 5.2.

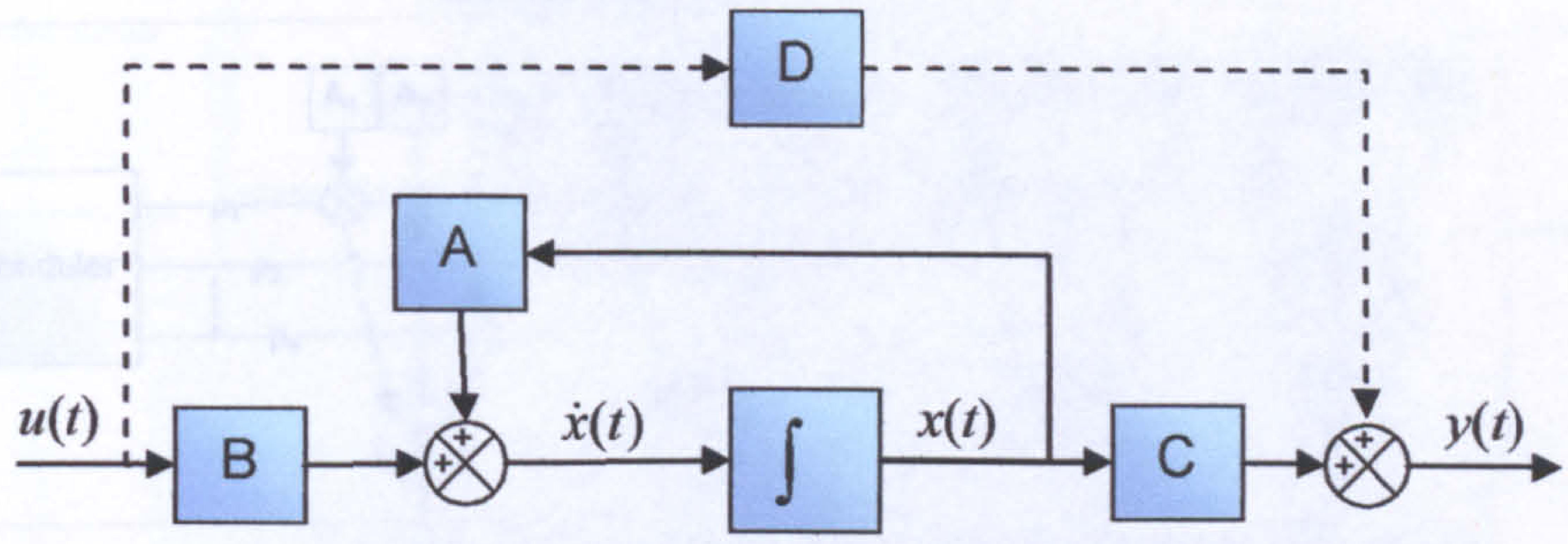

Figure 5.2 General ABCD block diagram

We have used dotted lines connect the D matrix, since for many cases the D matrix will be zero, and these connections will not be present.

In TSM, each model used is a linear model identical to Figure 5.2. Therefore, unlike LMN, TSM does not require linearisation. For a network consisting of i=1,…,M number of models. The overall system can then be approximated as

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{M} \rho_i(\underline{\phi}(t))\left[A_i x(t) + B_i u(t)\right] \\ y(t) &= \sum_{i=1}^{M} \rho_i(\underline{\phi}(t))\left[C_i x(t) + D_i u(t)\right], \end{aligned} \tag{5.4}$$

where A, B, C, D are the interpolated parameters of the linear models,

$$\begin{aligned} A(\underline{\phi}) &= \sum_{i=1}^{M} \rho_i(\underline{\phi})A_i, & B(\underline{\phi}) &= \sum_{i=1}^{M} \rho_i(\underline{\phi})B_i, \\ C(\underline{\phi}) &= \sum_{i=1}^{M} \rho_i(\underline{\phi})C_i, & D(\underline{\phi}) &= \sum_{i=1}^{M} \rho_i(\underline{\phi})D_i \end{aligned} \tag{5.5}$$

The parameters of (5.5) depend only on the scheduler $\underline{\phi}$. This type of parameter varying structure is much simpler then (2.12). The overall TSM structure is shown in the figure below.

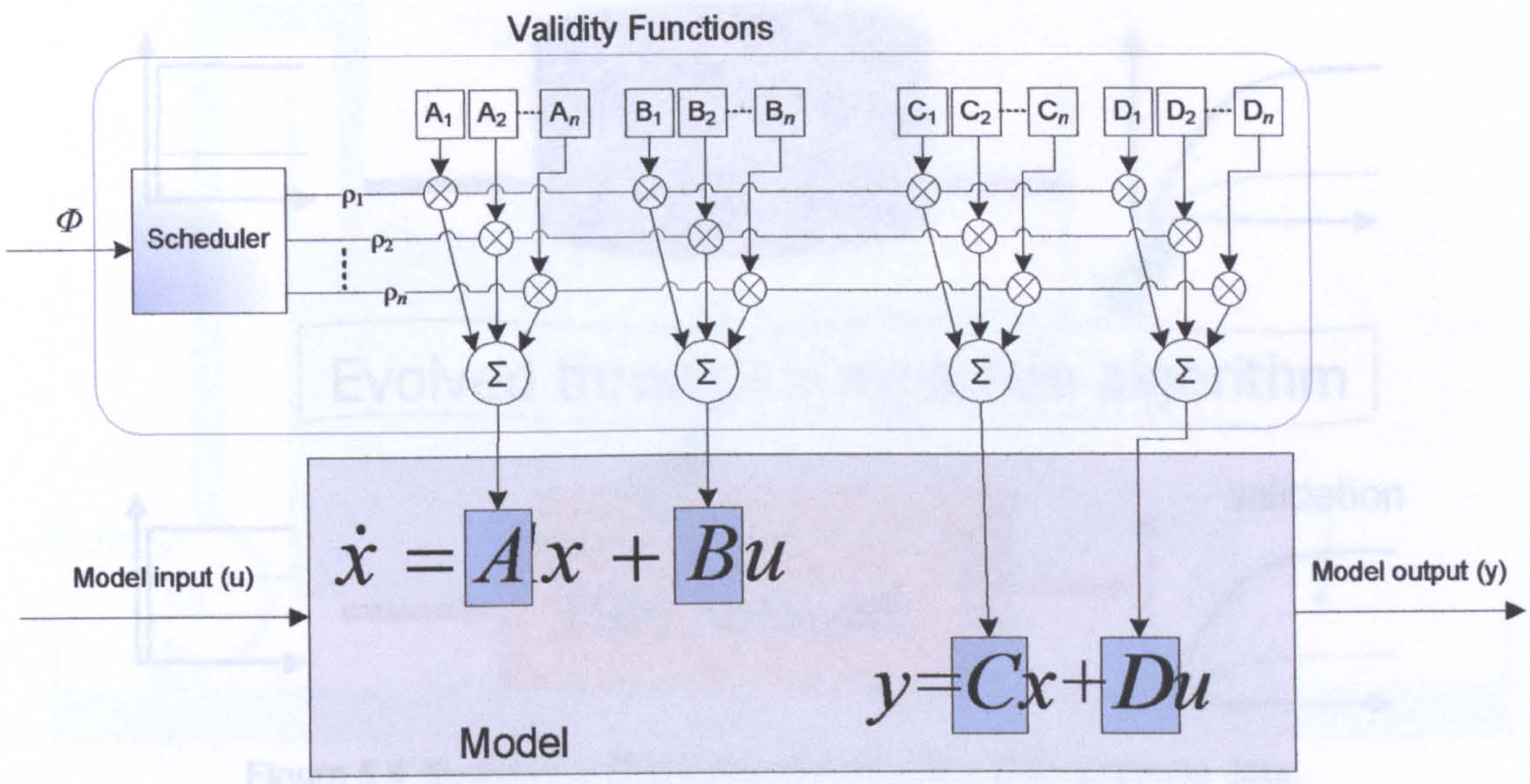

Figure 5.3 Trajectory-Scheduling Model

For practical engineering systems, an evolution-based TSM modelling technique has been developed. This simple and easily understood method is applicable to continuous-time modelling and is capable of obtaining an optimal nominal model from a wide range of step response data.

To achieve high modelling quality and fast generation of TSM, an evolutionary algorithm may be used to search for globally optimal solutions. For more adaptiveness, further fine-tuning by local optimisation method may be applied. This method is illustrated in Figure 5.4.

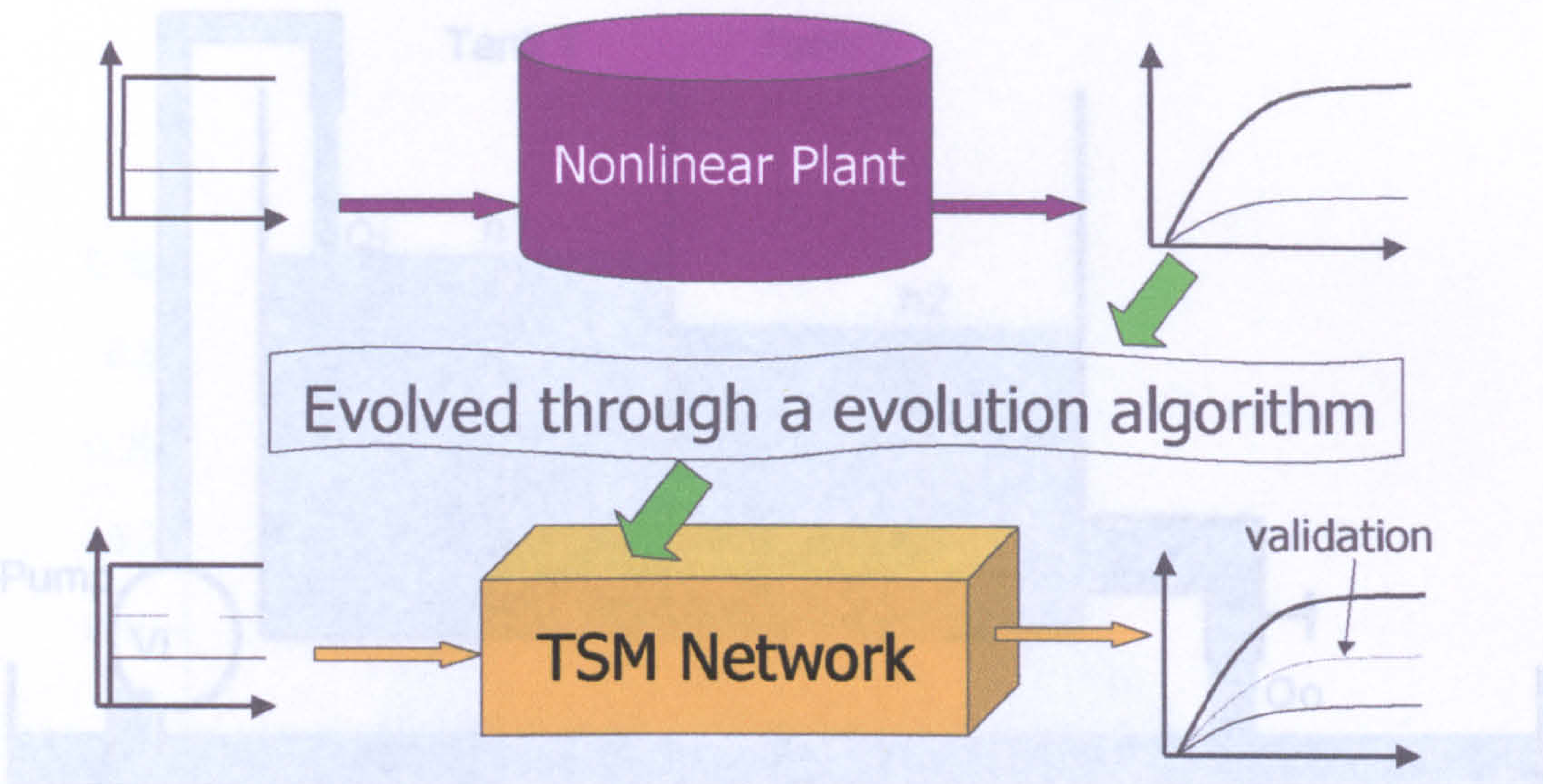

Figure 5.4 Evolving a TSM network from plant step response data.

5.3 Nonlinear Coupled System Modelling (Case 1)

The diagram shown below is a twin-tank coupled non-linear hydraulic system that model liquid-level found in chemical and diary plant. The scale down model can also be found in the laboratory. Based on the Bernoulli's mass-balance and flow equations, the system structure is described in (5.6) and the process is shown in Figure 5.5.

$$\begin{bmatrix}\dot{h}_1\\ \dot{h}_2\end{bmatrix}=\begin{bmatrix}-\operatorname{sgn}(h_1-h_2)\dfrac{c_1a_1}{A}\sqrt{2g|h_1-h_2|}\\ \operatorname{sgn}(h_1-h_2)\dfrac{c_1a_1}{A}\sqrt{2g|h_1-h_2|}-\dfrac{c_2a_2}{A}\sqrt{2g\left(h_2-H_0\right)}\end{bmatrix}+\begin{bmatrix}\dfrac{Q_1}{A} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}v_1\\ 0\end{bmatrix} \tag{5.6}$$

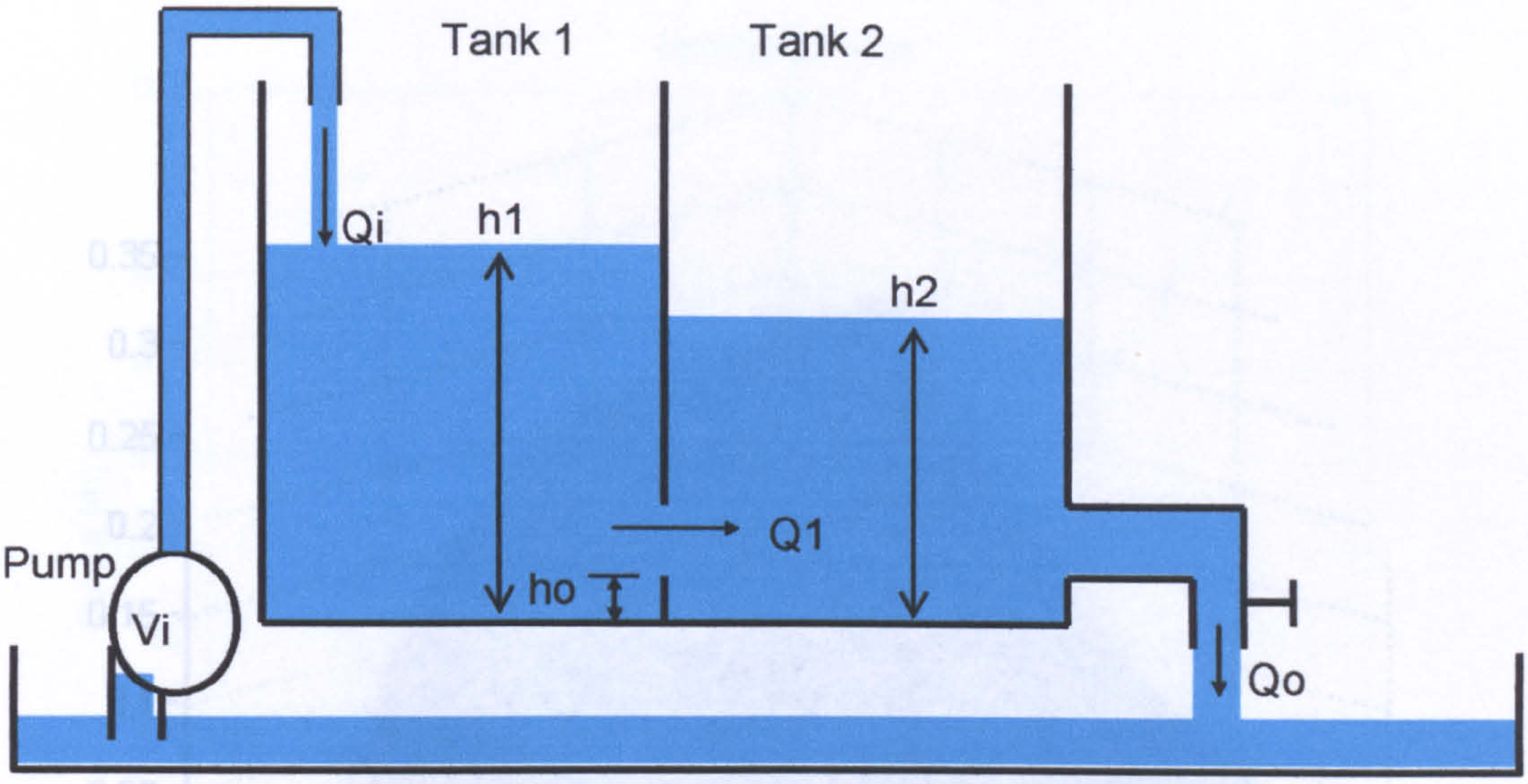

Figure 5.5 Nonlinear twin-tank couple system

Table 5.1: The coefficients of the twin-tank

Height Of water in tank 1	h_1 m
Height Of water in tank 2	h_2 m
minimum height of water in tank	$H_0 = 0.03$ m
Cross sectional area of tank 1&2	$A = 0.01\ m^2$
Discharge coefficient of orifice 1	$c_1 = 0.53$
Discharge coefficient of orifice 2	$c_2 = 0.63$
Cross sectional area of orifice 1	$a_1 = 0.0000396\ m^2$
Cross sectional area of orifice 2	$a_2 = 0.0000386\ m^2$
Gravitational constant	$g = 9.81\ m\ s^{-2}$
Pump Flow rate	$Qi = 0.000007\ (m^3\ s^{-1}\ V^{-1})$
Flow rate from tank 1 to tank 2	$Q_1 = c_1 a_1 \sqrt{2g(h_1 - h_2)}\quad (m^3\ s^{-1})$
Discharge rate	$Qo = c_2 a_2 \sqrt{2g(h_2 - H_0)}\ (m^3\ s^{-1})$

The step response of the twin-tank was plotted in 3D forms (Figure 5.6). The nonlinearity of the plant can be clearly seen.

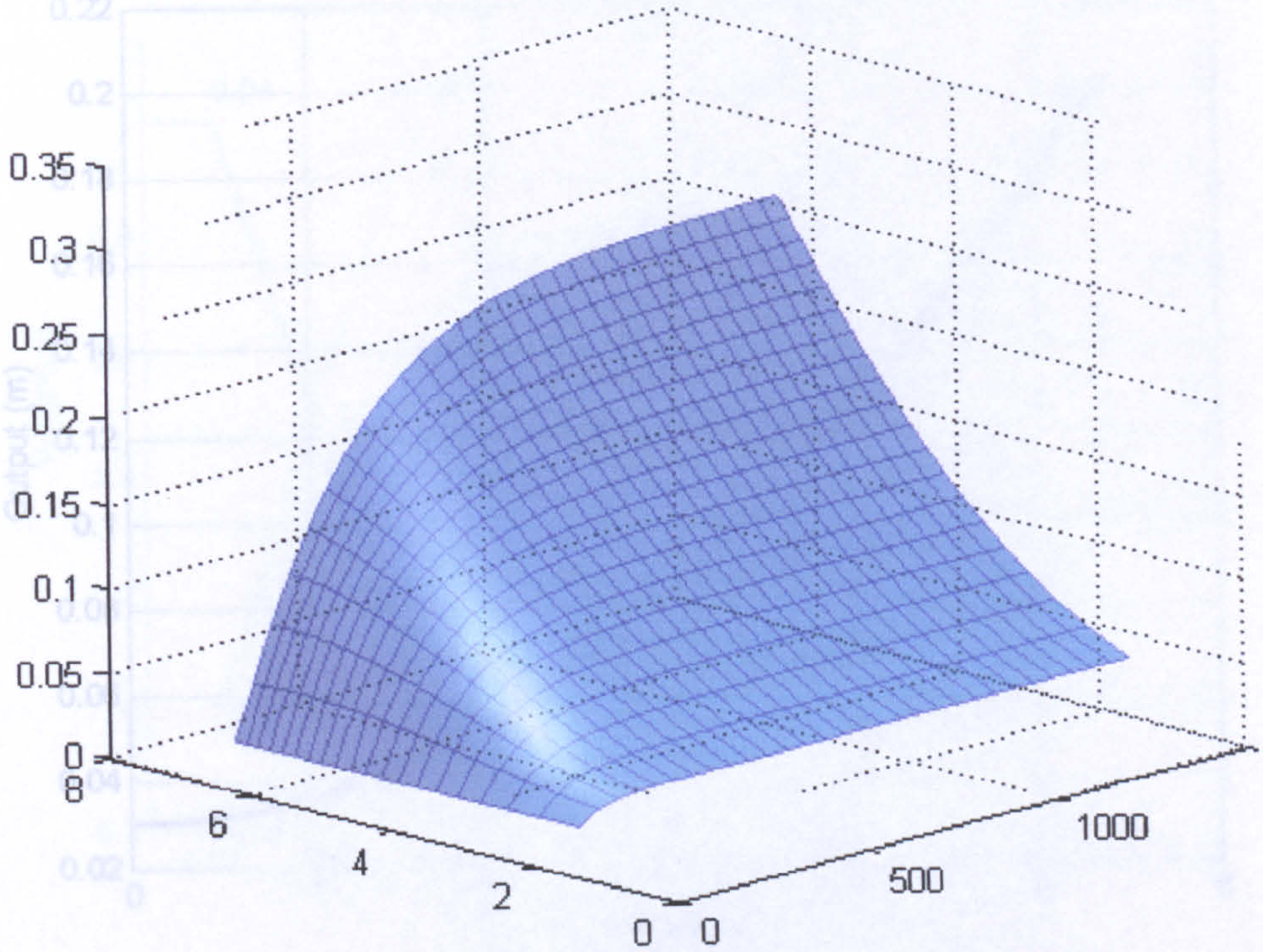

Figure 5.6 The nonlinearity of the twin tank system presented as 3D open loop step response

5.3.1 System Identification with TSM

The first step is to obtain a static model of the twin tank (Figure 5.7). This can easily be done by generating an input versus output data from equation (5.6). The next step involves finding the best transition/operating point for the plant using the algorithm described in chapter 3.2. From the node search algorithm, the best transition nodes found to be at 0.04m, 0.07m and 0.13m.

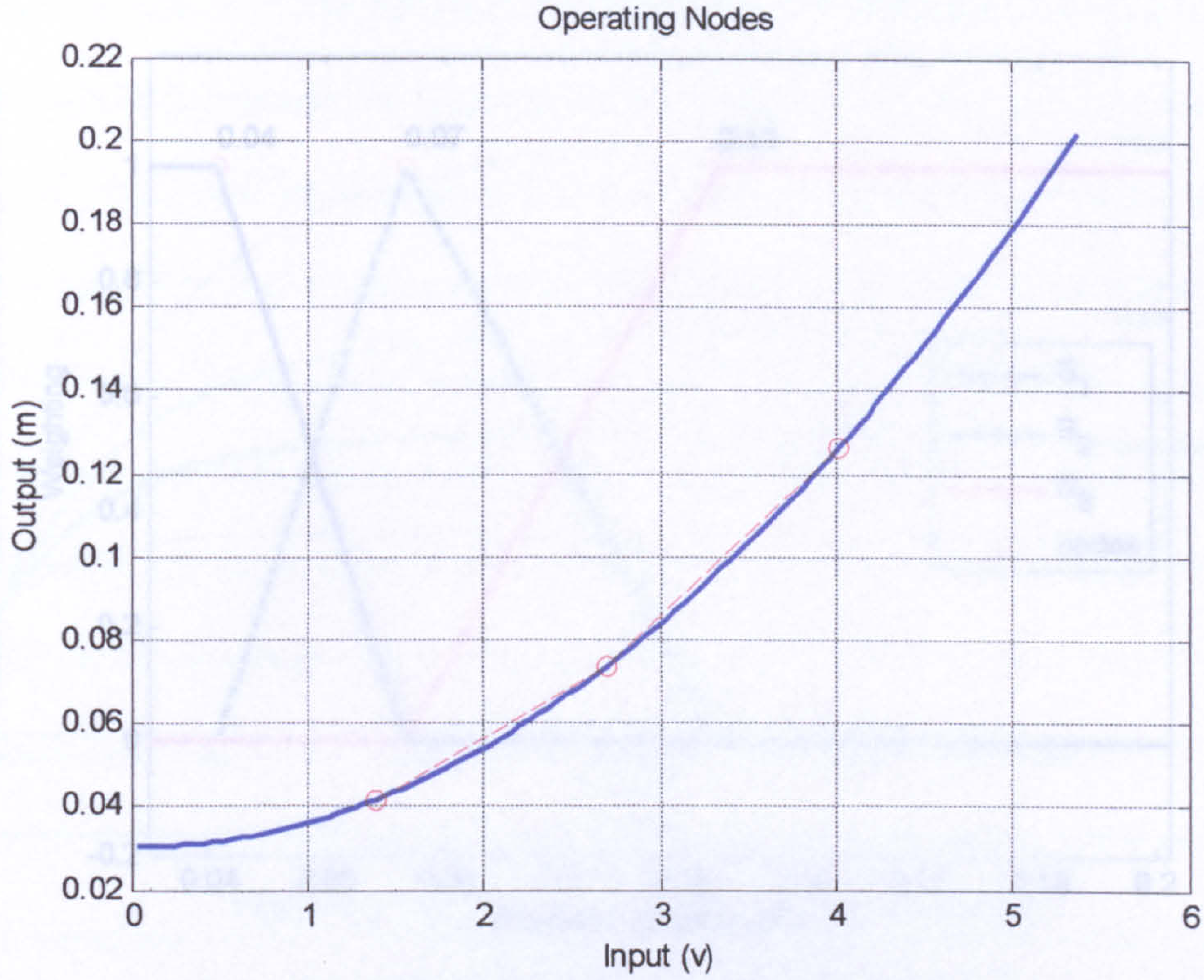

Figure 5.7 The static model and operating nodes of the twin-tank system

In step 3, we construct a TSM at each of these transition points using the linear model given by the equation (5.7).

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U \tag{5.7}$$

Finally, linear interpolation will be use to link all the nodes to together as shown in Figure 5.8.

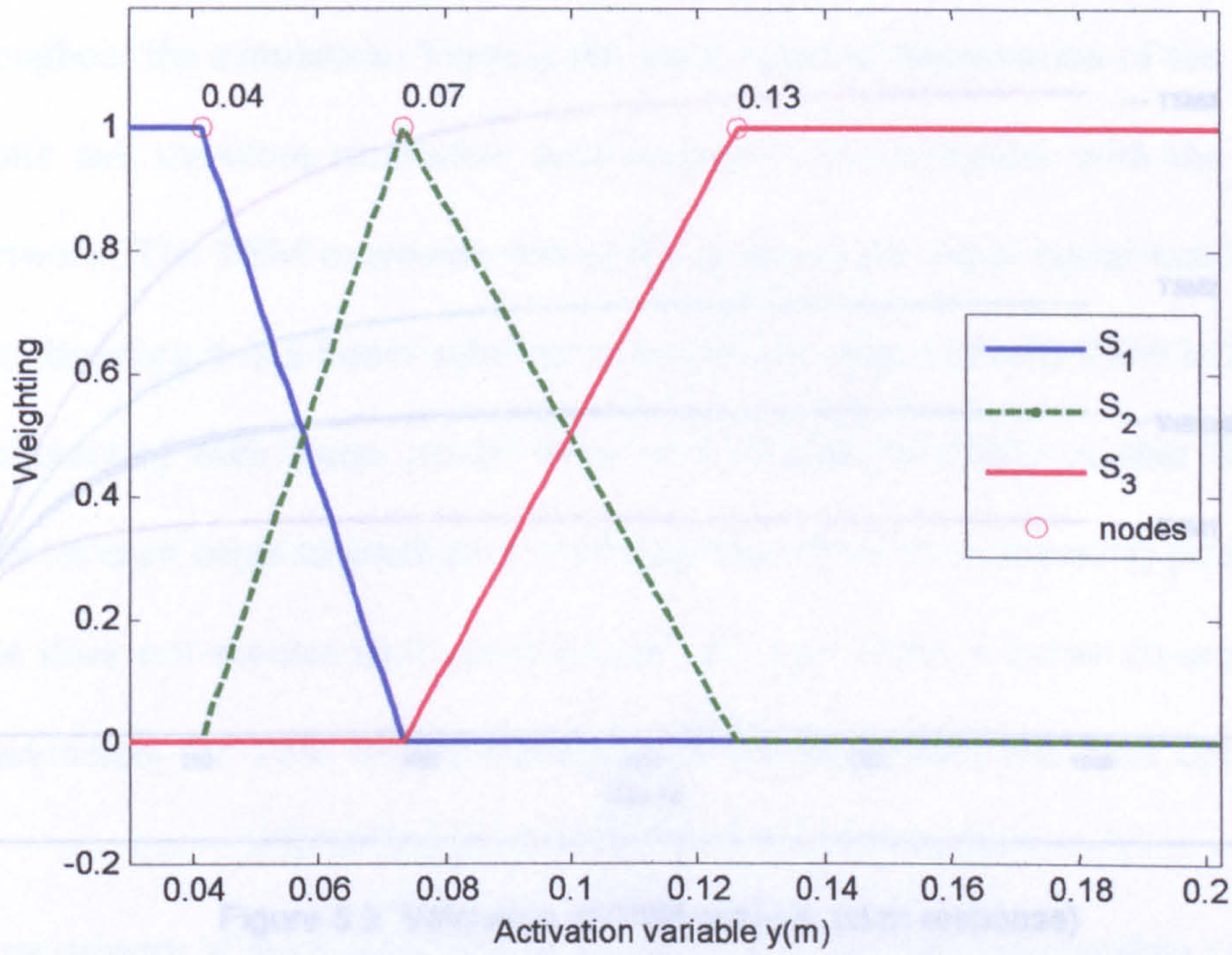

Figure 5.8 Linear interpolation of the TSM Twin-tank system

5.3.2 Validation and Analysis

In modelling of the twin-tank plant for the whole operating region, the 3 nodes TSM was evolved through EA. All the parameters for TSM were obtained simultaneously using evolutionary search methods. From the TSM produced, a response of an unseen operating point was generated. In Figure 5.9 shows that the identified model at various unseen operating points matched the actual system.

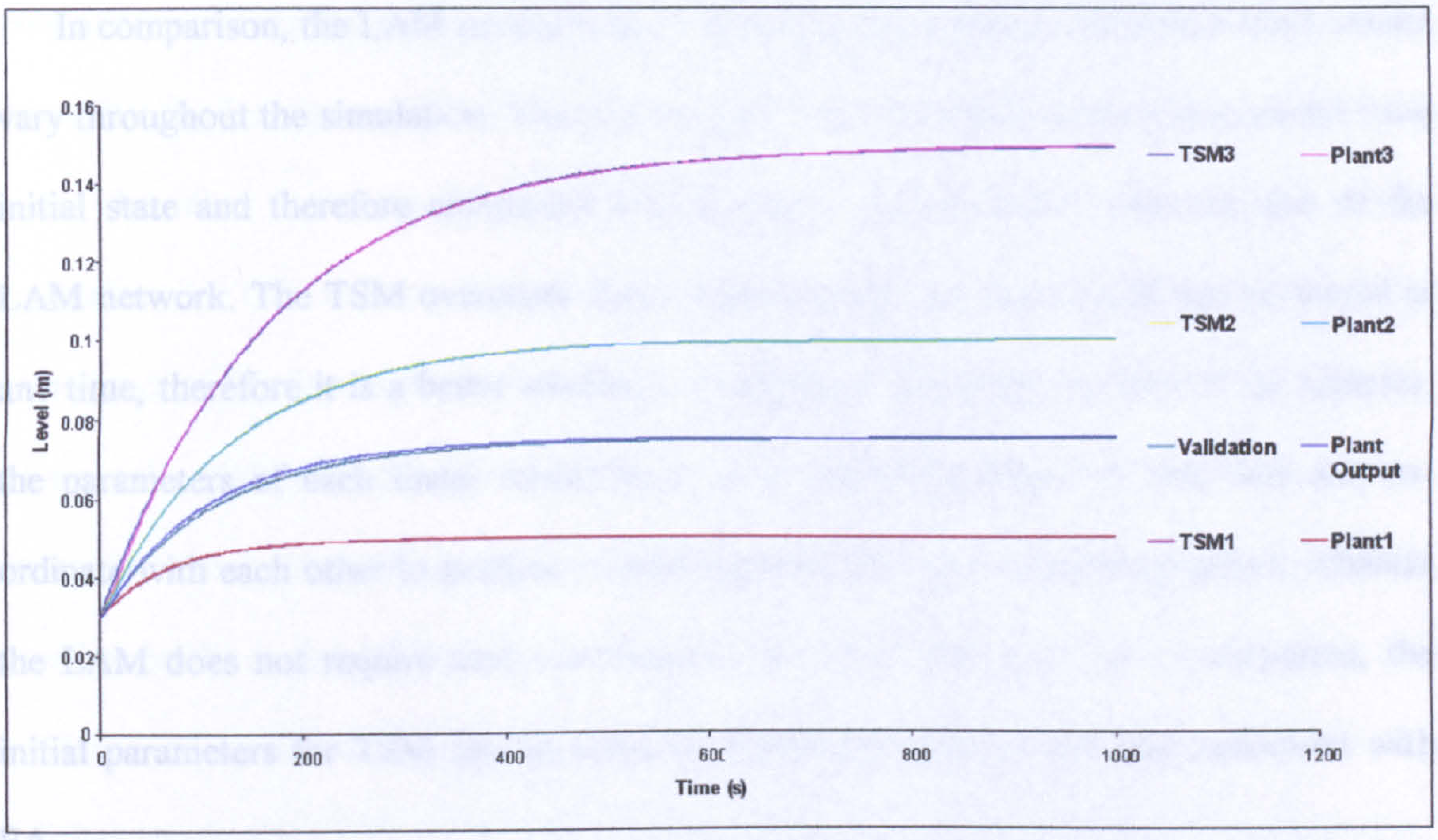

Figure 5.9 Validation of TSM network (step response)

The TSM is not limited to producing a step response, indeed it can mimic any response from any given initial conditions. The result shows that the TSM fit the actual nonlinear system accurately (see Figure 5.10). From the results, it was clear that this method is suitable to use for continuous-time modelling.

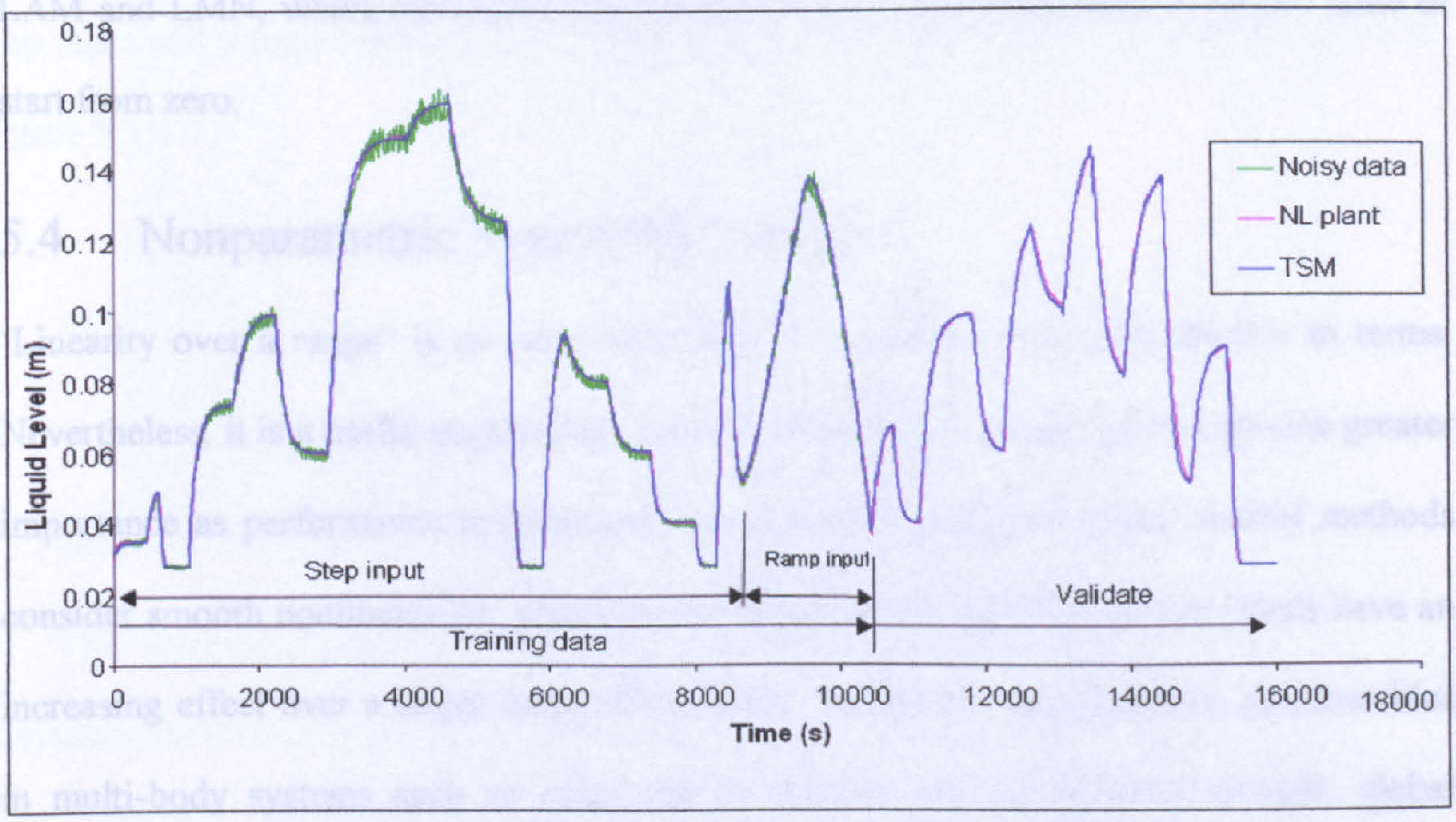

Figure 5.10 Validation of TSM network (continuous response)

In comparison, the LAM network only valid for a step response where the input cannot vary throughout the simulation. Varying the input requires regeneration of the model from initial state and therefore simulation time increases proportionally with the size of the LAM network. The TSM overcome this problem where the input signal can be varied at any time, therefore it is a better solution in modelling. Although the TSM is far superior, the parameters of each linear model have to be chosen carefully so that they can co-ordinate with each other to produce a valid response for unseen operating points, whereas the LAM does not require such coordination. To give TSM a better co-ordination, the initial parameters for TSM can be aided by identification tools and later optimised with EA.

The superiority of using evolution methods in modelling are that the data used need not be specified and their ability to learn through a simple cost function. This flexible method is preferable over classical methods because it has a wider range of selection and handles many types of industrial problems. As for this parameter-varying TSM network, there is only one state at any given time, therefore it can start from any operating point unlike the LAM and LMN, where the initial condition for each local model have to be the same or start from zero.

5.4 Nonparametric System Modelling

'Linearity over a range" is an oxymoron; that is, a mathematical contradiction in terms. Nevertheless, it is a useful engineering concept. However, nonlinear effects assume greater importance as performance requirements become more stringent. Many control methods consider smooth nonlinearities, which are linearisable near equilibrium and which have an increasing effect over a larger range of operation. Geometric and kinematic nonlinearities in multi-body systems such as spacecraft or robotics are examples of smooth, global nonlinearities. Control theorists often assume that these nonlinearities are sufficiently well known that the functions and their derivatives can be used in transformation techniques.

On the other hand, many control applications require accurate motion over small amplitudes. In this regime, the nonlinearities tend to be non-smooth and possibly discontinuous. Friction is a common example of a non-smooth nonlinearity. In addition, non-smooth nonlinearities such as friction and backlash may possess hidden, un-measurable states. The ability to perform identification depends on the nature of the plant, as well as on the environment.

Motivated by the above problems, original data collected should not be filtered or modified, such data include friction, noise and some unknowns. Using convolution methods, the original response can be reproduced. To proceed, the plant unit-impulse response data can be obtained from the step response as given by:

$$g(t) = \dot{y}_s(t) / A \tag{5.8}$$

where $g(t)$ is the data collected, $\dot{y}_s(t)$ is the output at different operating point and A is the current input state.

This may be regarded as such an infinite-order "model", since convoluting this with the step input will yield a high fidelity reconstruction of the step response, as indeed shown in Figure 5.11.

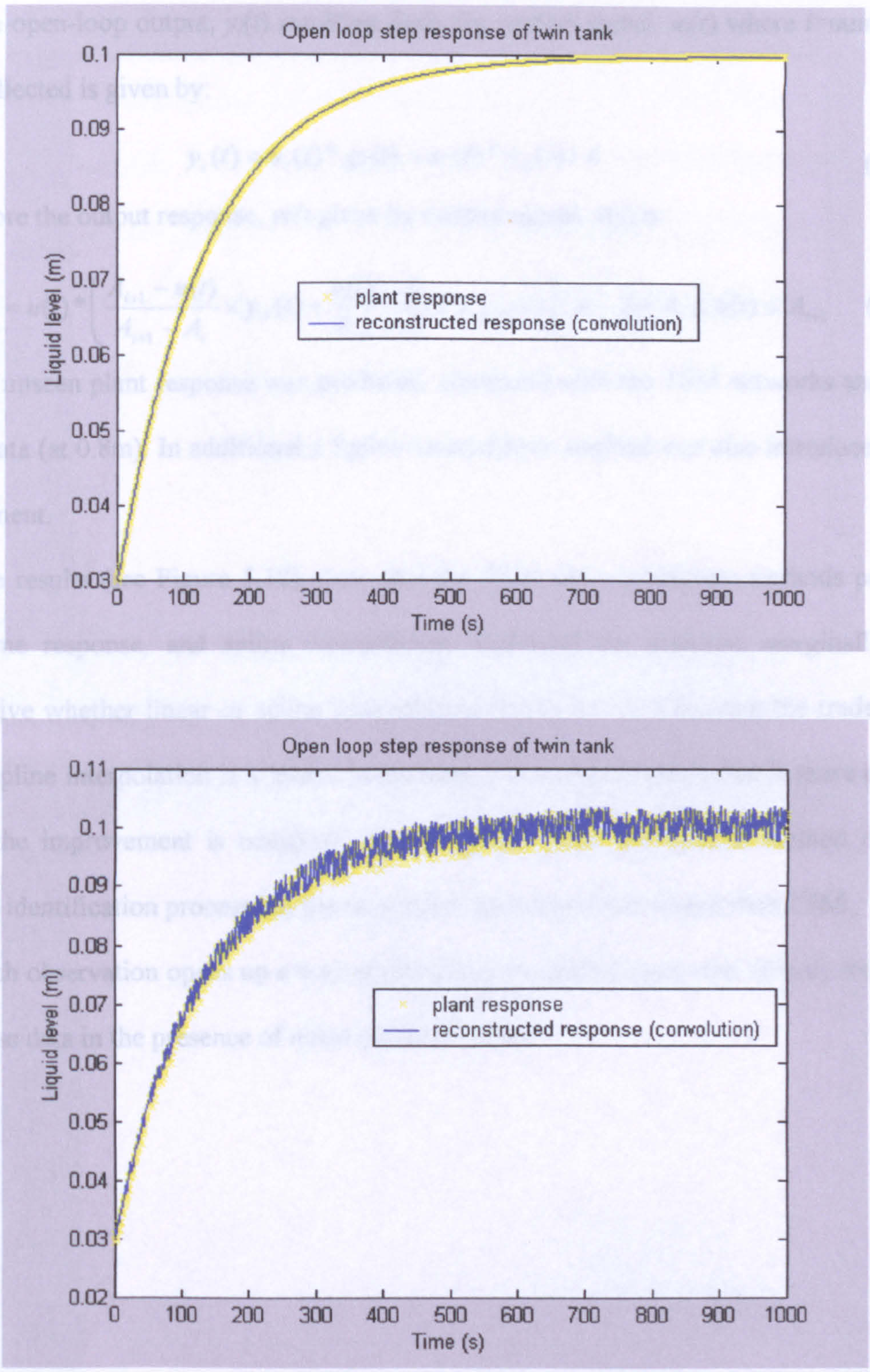

Figure 5.11 Regeneration of plant response with and without noise using convolution methods

The same multiple model technique can be used to network the response data to approximate a nonlinear plant. Using the same linear networking technique, multiple data were collected at different operating points.

The open-loop output, $y_i(t)$ resulting from the control signal, $u_i(t)$ where i=numbers of data collected is given by:

$$y_i(t) = u_i(t) * g_i(t) = u_i(t) * \dot{y}_{si}(t) / A \quad (5.9)$$

Therefore the output response, $y(t)$ given by control signal, $u(t)$ is:

$$y(t) = u(t) * \left(\frac{A_{i+1} - u(t)}{A_{i+1} - A_i} \times y_{si}(t) + \frac{u(t) - A_i}{A_{i+1} - A_i} \times y_{s(i+1)}(t) \right) / A \quad for\, A_i \le u(t) \le A_{i+1} \quad (5.10)$$

An unseen plant response was produced, compared with the TSM networks and actual plant data (at 0.8m). In additional a Spline interpolation method was also introduced in the experiment.

The results (see Figure 5.12) show that the TSM and convolution methods produced the same response, and spline interpolation improved the response marginally. It is subjective whether linear or spline interpolation should be used because the trade-off for using spline interpolation is a longer processing time and a structure that is more complex when the improvement is marginal. Another issue is the convolution method does not require identification process but the simulation time was much longer than TSM.

Such observation opens up a way of designing an optimal controller directly from plant response data in the presence of noise and uncertainty.

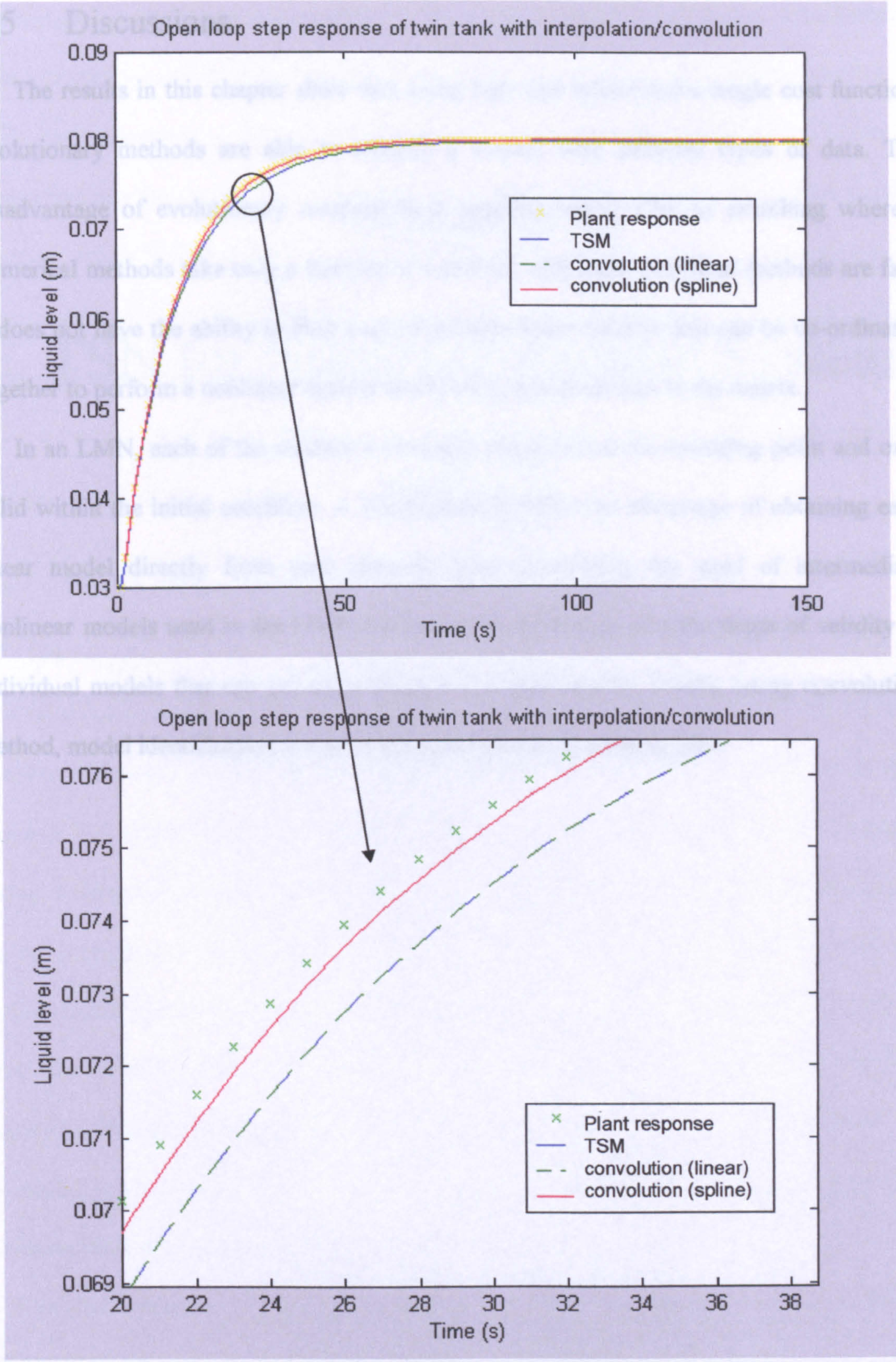

Figure 5.12 Step response of TSM, convolution, linear and spline

5.5 Discussions

The results in this chapter show that using only one model and a single cost function, evolutionary methods are able to identify a system with different types of data. The disadvantage of evolutionary methods is it requires longer time in searching whereas numerical methods take only a fraction of a second. Although numerical methods are fast, it does not have the ability to find a set of suitable linear models that can be co-ordinated together to perform a nonlinear system due to many identical data in the matrix.

In an LMN, each of the models is normally linearised at the operating point and only valid within the initial condition. A TSM network offers the advantage of obtaining each linear model directly from step response data, eliminating the need of intermediate nonlinear models used in the LMN and hence the advantage of wide range of validity of individual models that can act stand-alone in a certain degree. Finally, using convolution method, model identification can be eliminated using only process data.

Chapter 6

TSN for Nonlinear Control System Design

The aim of this chapter is to describe the 'Trajectory-Scheduling Controller' (TSC) design method based upon a TSN in chapter 3, which overcomes the deficiency associated with LCN. Each element of a TSC can be of a simple linear controller, such as PID. The tuning parameters can be obtained directly from a set of step response data at several typical operating levels for fast prototyping. Since step response data are often readily available in control engineering practice, such TSC can be evolved automatically and optimally from these data using appropriate model. The overall controller is co-ordinated and evolved along the entire operating trajectory in the operating envelope, tackling the control problem of practical or nonlinear plants.

6.1 PID Controllers Feasible as TSC Nodes

PID control can be used in TSC effectively and efficiently, although other types of controllers can also be used. PID control offers the simplest and yet most efficient solution to many real-world control problems. Its three-term functionality covers treatment to both transient and steady-state responses. 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 and covers more than 90% of industrial usage. 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.* 2005).

The PID controller come in many forms, it can also be implemented as extreme forms of phase-lag and phase-lead compensators. The PID controller is also known as the 'Three-

Term' controller where K_P is the proportional gain, T_I the integral time constant and T_D the derivative time constant, whose transfer function in 'ideal form' is shown in Figure 6.1.

$$G(s) = K_P\left(1 + \frac{1}{T_I s} + T_D s\right) \tag{6.1}$$

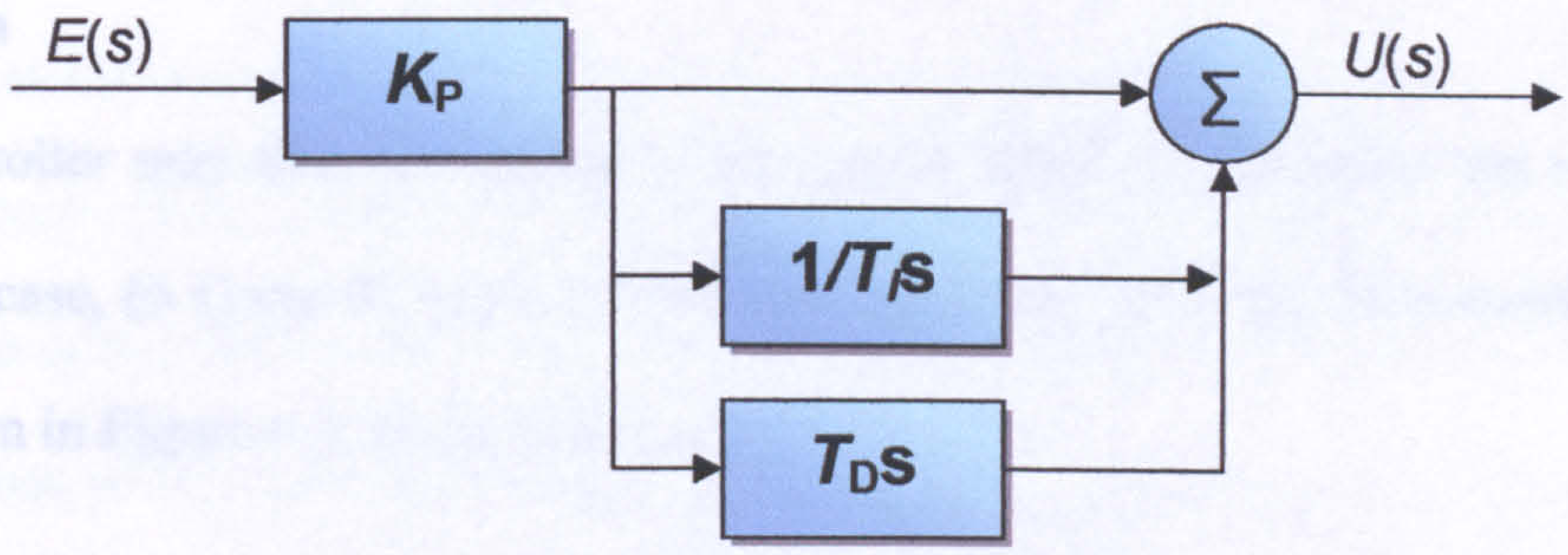

Figure 6.1 PID Structure – Ideal Form

Two other common type are 'parallel form' (Figure 6.2) and 'series form' (Figure 6.3).

Parallel Form

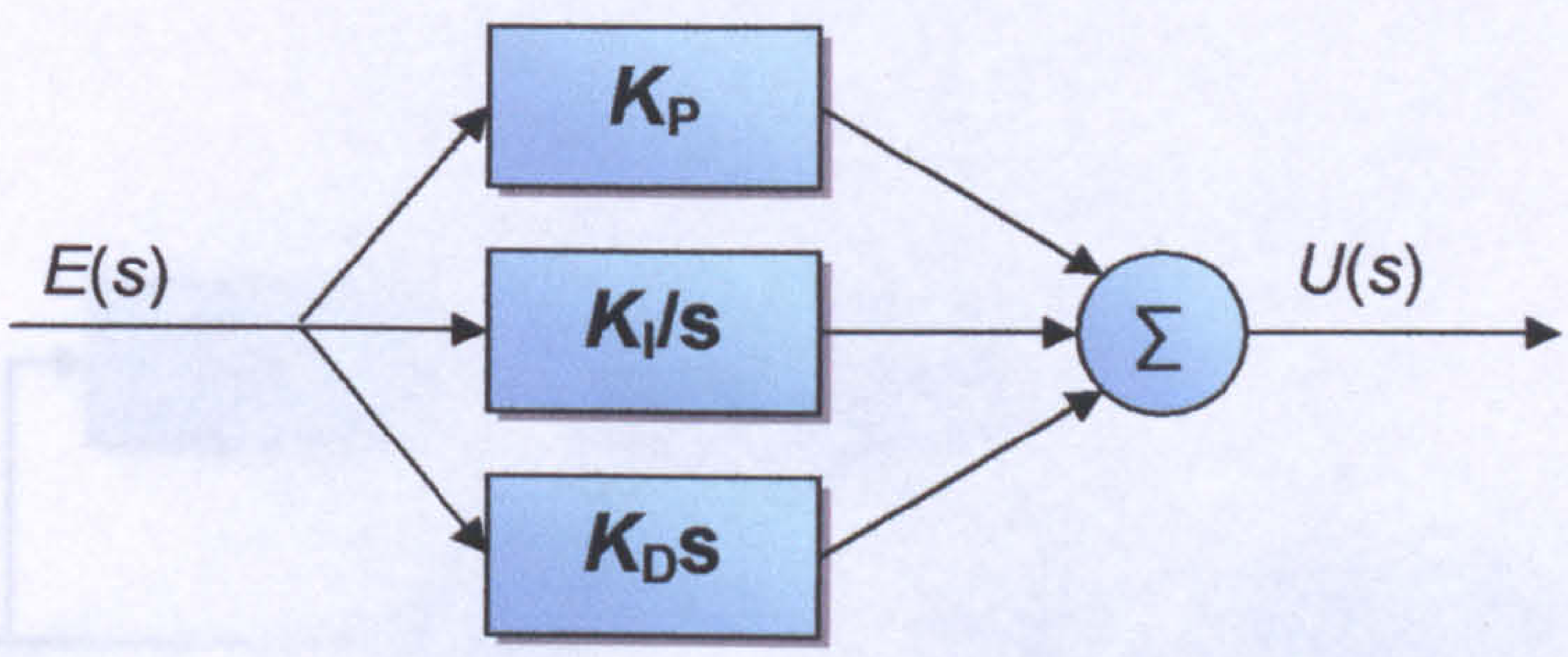

Figure 6.2 PID Structure – Parallel Form

The parallel form PID controller can be written as:

$$G(s) = K_P + \frac{K_I}{s} + K_D s\,, \tag{6.2}$$

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.

Series Form

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, (6.1) can be implemented as a cascade of a PD and PI controller (Li *et al.* 1998), shown in Figure 6.3, in the form:

$$G(s) = K_P(\alpha + T_D s)\left(1 + \frac{1}{\alpha T_I s}\right), \tag{6.3}$$

where,

$$\alpha = \frac{1 \pm \sqrt{1 - 4T_D / T_I}}{2} > 0 . \tag{6.4}$$

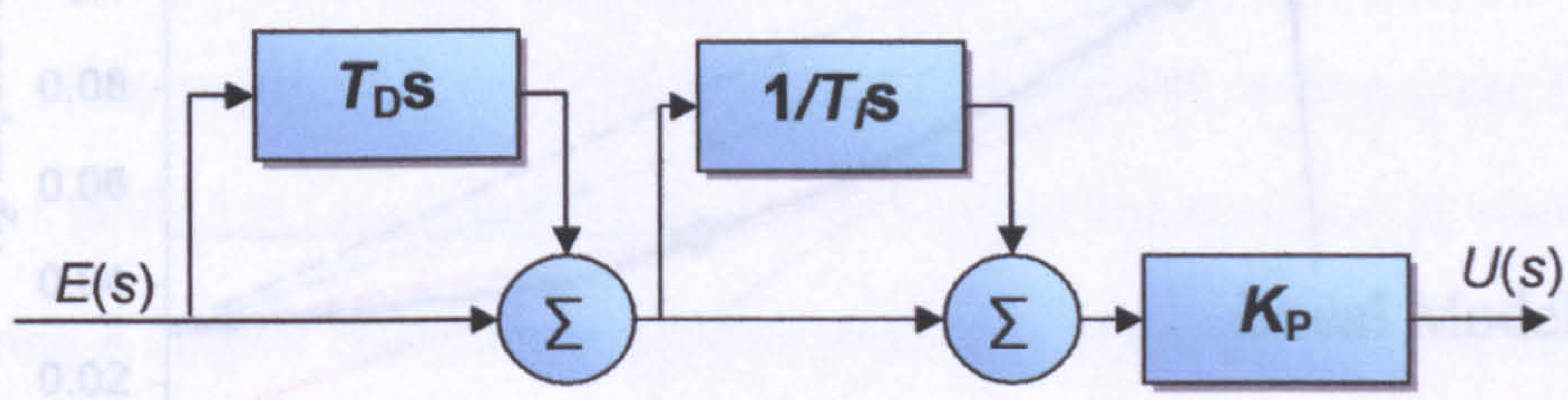

Figure 6.3 PID Structure – Series Form

6.2 TSC Construction

In this methods, each of the linear systems was represented by a transfer function or transfer function matrix. This is similar to represent a nonlinear system response with high fidelity by convolution or harmonic analysis. This simple technique eases the difficulties encountered in conventional linearisation without the need for an initial nonlinear model. Unlike a LMN, TSM gives a straightforward approximation in the entire trajectory ranging from the initial condition to the setpoint, whilst a LMN applicable only around the setpoint. (see Figure 6.4 twin-tank coupled system)

Once the trajectory of a static model has been predicted, control measures are required to ensure that the trajectory data is applied in a consistent manner when calculating the many trajectory-related parameters required by the controllers of the system. This thesis describes the queue control techniques used in generating the predicted trajectory and the subsequent use of the trajectory data. The queue control logic that is discussed requires a minimum amount of main storage while a task is waiting to be performed.

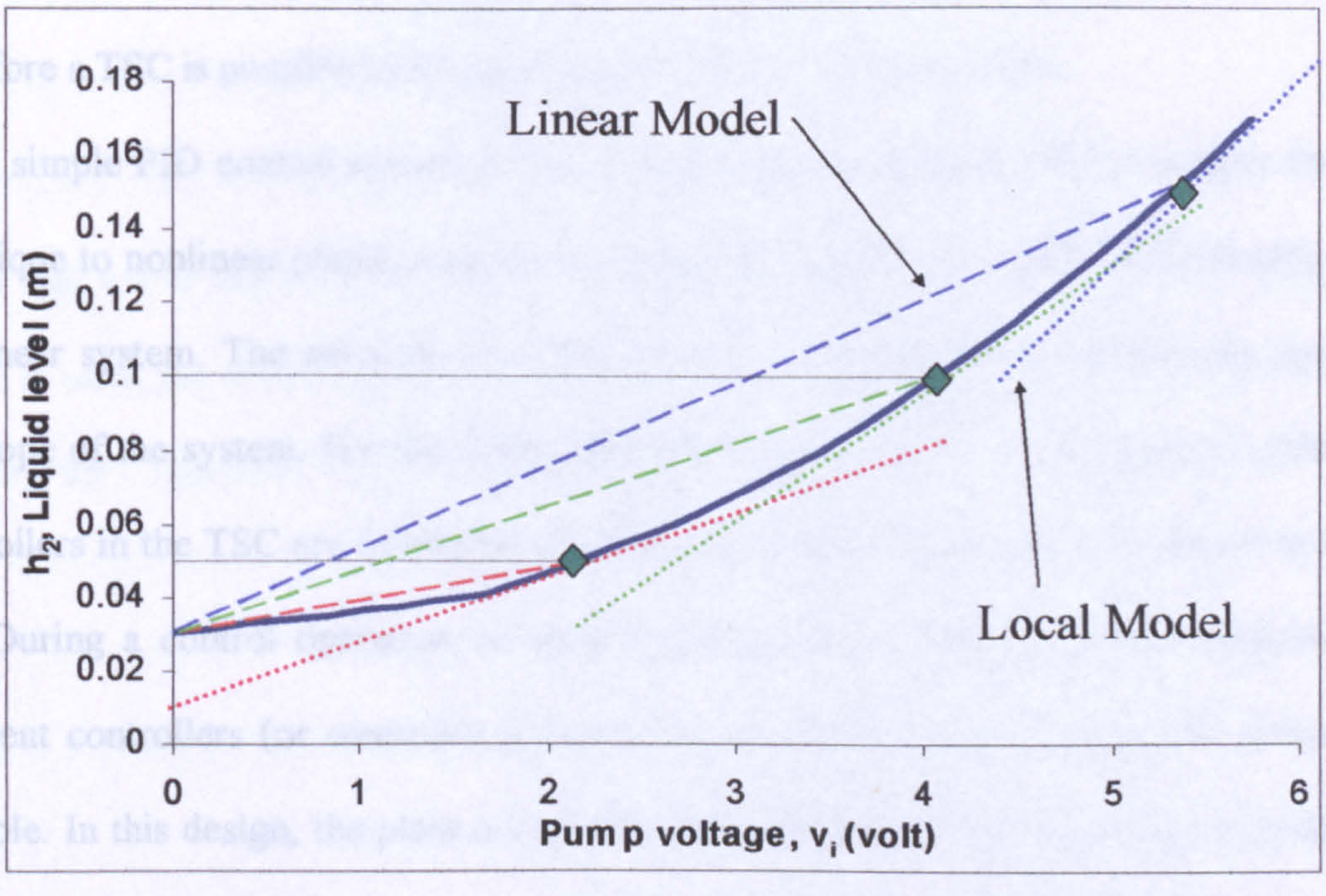

Figure 6.4 Activation region of Linear Model Network and Local Model Network

In modelling of a nonlinear system, a set of linear system obtained at different operating points was networked conveniently by simple linear local interpolation (chapter 3.2.1) to produce an unseen operating level.

The TSC developed exhibits simple and effective control system design in an operating envelope of a nonlinear plant. In the controller design, the example shows that a linear TSC offers high performance control for a nonlinear system in the entire operating envelope. TSC designed by evolutionary computation shown that the technique can accommodate many practical requirements imposed by engineering specifications, since it does not require differentiation of performance index.

A direct convolution method for controller design from step response data, bypassing the system identification stage has been developed. Validation against linear and nonlinear plants has shown that the performance of a controller evolved from the response data was better. This approach offered a step towards autonomy in building control system. Where, convolution operation has been extended to include a variable type of input excitations, therefore a TSC is possible to design directly from off or on-line data.

A simple PID control system can be designed easily out of a TSM. To apply the TSC technique to nonlinear plants, controllers were designed for the entire TSM network of the nonlinear system. The network is to provide adequate performance across the operating envelope of the system. For the TSM obtained in the previous section (case 1), three PID controllers in the TSC are scheduled or simply switched between them as shown in Figure 6.6. During a control operation, a variable indicating operating point is monitored and different controllers (or controller parameters) are activated according to this scheduling variable. In this design, the plant output $y(t)$ was used as scheduling variable to weight the output of the controllers.

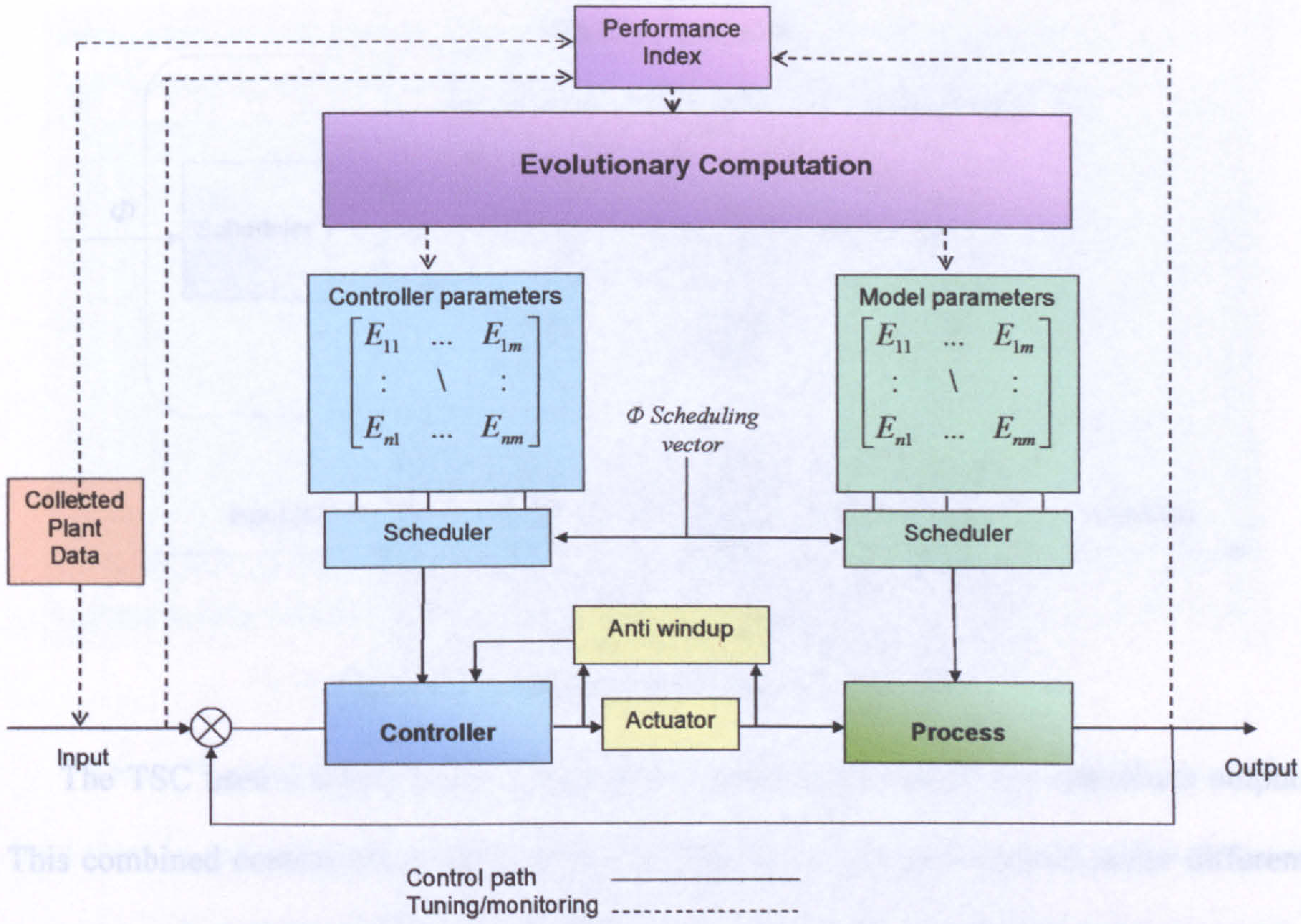

Figure 6.5 Trajectory-Scheduling Controller

In real world most of the actuators have its limit. If a controller with an integrating action is used, the error may continue to be integrated when the actuator saturates, leading to "wind up". Therefore, anti-windup was implemented in the TSC developed here.

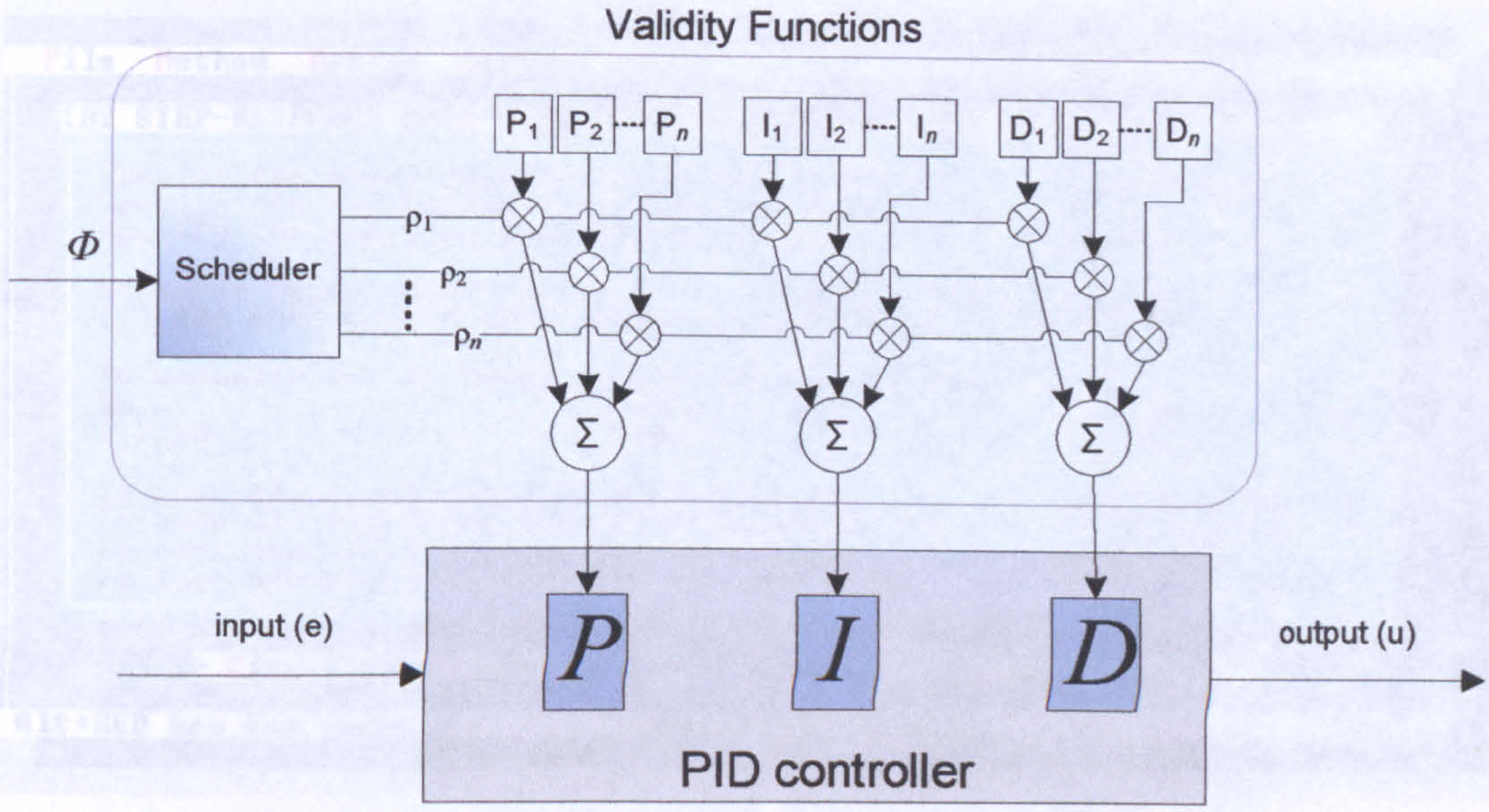

Figure 6.6 Trajectory-Scheduling PID Controller.

The TSC used a simple linear interpolation schedule to weight the controllers output. This combined control effort offers a good global close-loop performance under different operating condition of the nonlinear system in a simple way. Using the same principles, interpolation may also apply to the controller parameters.

6.3 TSC Design Methods

6.3.1 Generating Controller Nodes Using a CACSD Package

Individual PID controllers from a step-response trajectory of the twin-tank coupled system (case 1) to each of the three operating points were generated from the PIDeasy™ (see Figure 6.7) design package.

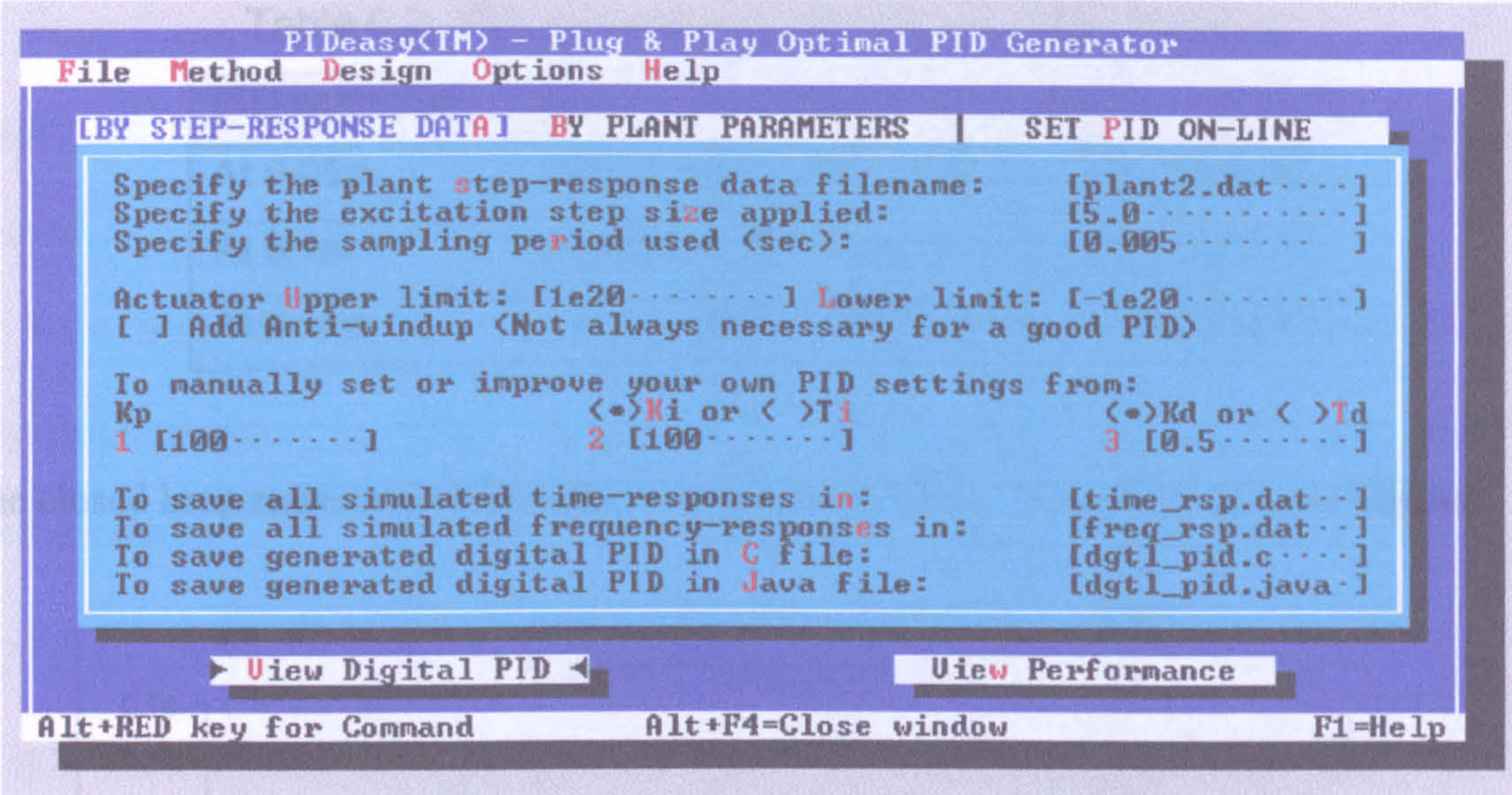

Figure 6.7 Direct design from plant response using PIDeasy™

PIDeasy™ analyses step response data and generates an appropriate PID controller from them in split seconds. At each operating point, the step response of the TSM produces a corresponding PID controller. This tuning method is known as local tuning. The locally tuned controller parameters are shown in Table 6.1.

Table 6.1: PID controller parameters by local tuning

PID controller	Kp	Ki	Kd
At 0.05m	3510.21	96.19	0.5353
At 0.1m	1350.86	172.707	1.34237
At 0.15m	1012.74	225.447	1.77881

6.3.2 Deriving Trajectory Controllers from TSM

To evolve a TSC out of TSM, all the parameters of the three linear controllers and the scheduling weights are evolved simultaneously in an operating envelope. This tuning method is known as global tuning. At the end of the search, the controller parameters obtained are tabulated in Table 6.2.

Table 6.2: PID controller parameters by global tuning

PID controller	Kp	Ki	Kd
At 0.05m	2000.175034	1351.85888	1010.193409
At 0.1m	86.58769437	164.8790683	237.7647766
At 0.15m	1.652132227	1.979571326	1.754395542

The closed loop responses of the locally tuned and globally tuned TSC were compared.

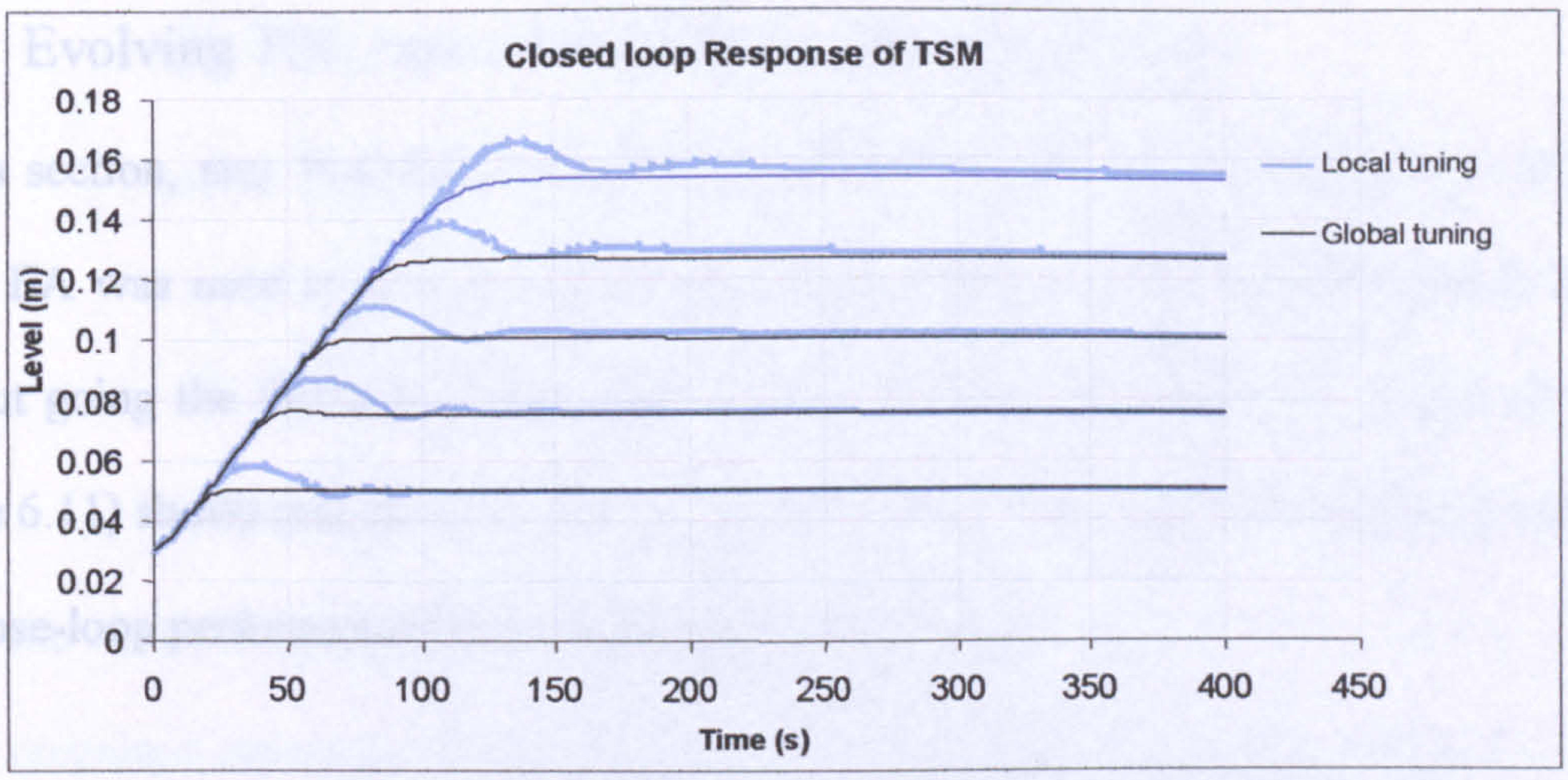

Figure 6.8 Closed loop responses of the TSC at operating points including the unseen ones at 0.075m and 0.125m.

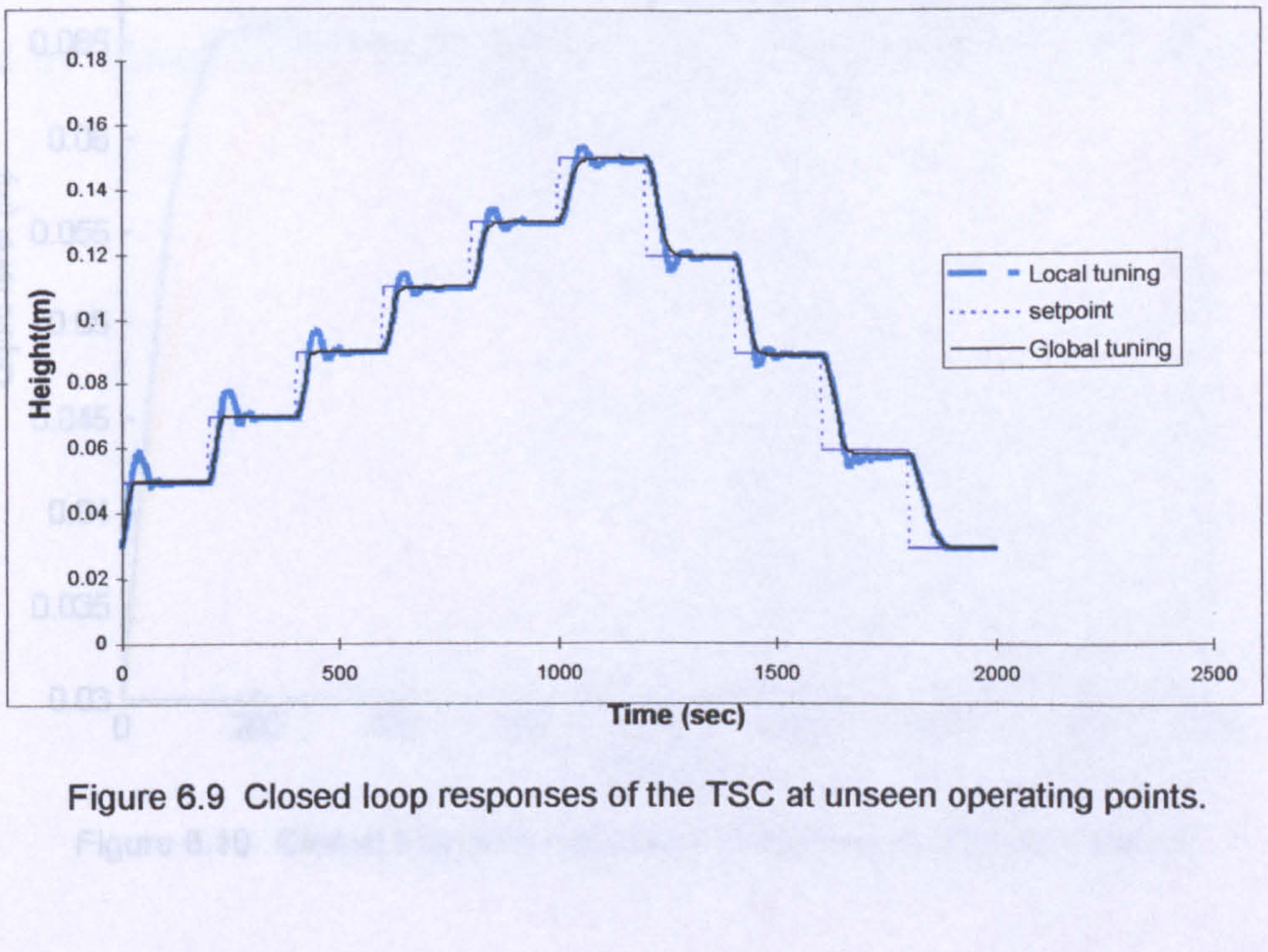

Figure 6.9 Closed loop responses of the TSC at unseen operating points.

From the results, it can be seen that the performance of the globally tuned method (see Figure 6.8 and Figure 6.9) exceeds that of the locally tuned method providing a fast optimal solution to the nonlinear control problem.

Note that in the locally tuned method, each PID controller was generated using 'linear' step-response data, but is now tested against the nonlinear plant. This reveals the need for network tuning.

6.3.3 Evolving TSC from Multi-Setpoint Response Data

In this section, step response data of twin-tank plant (case 1) is collected at 0.05m and 0.1m. EA was used to evolve a PID controller towards optimal performance at 0.065m without going the through system identification process. The result (see Figure 6.10 and Figure 6.11) shows that the close-loop performance of this design method is very similar to the close-loop performance of nonlinear twin-tank plant.

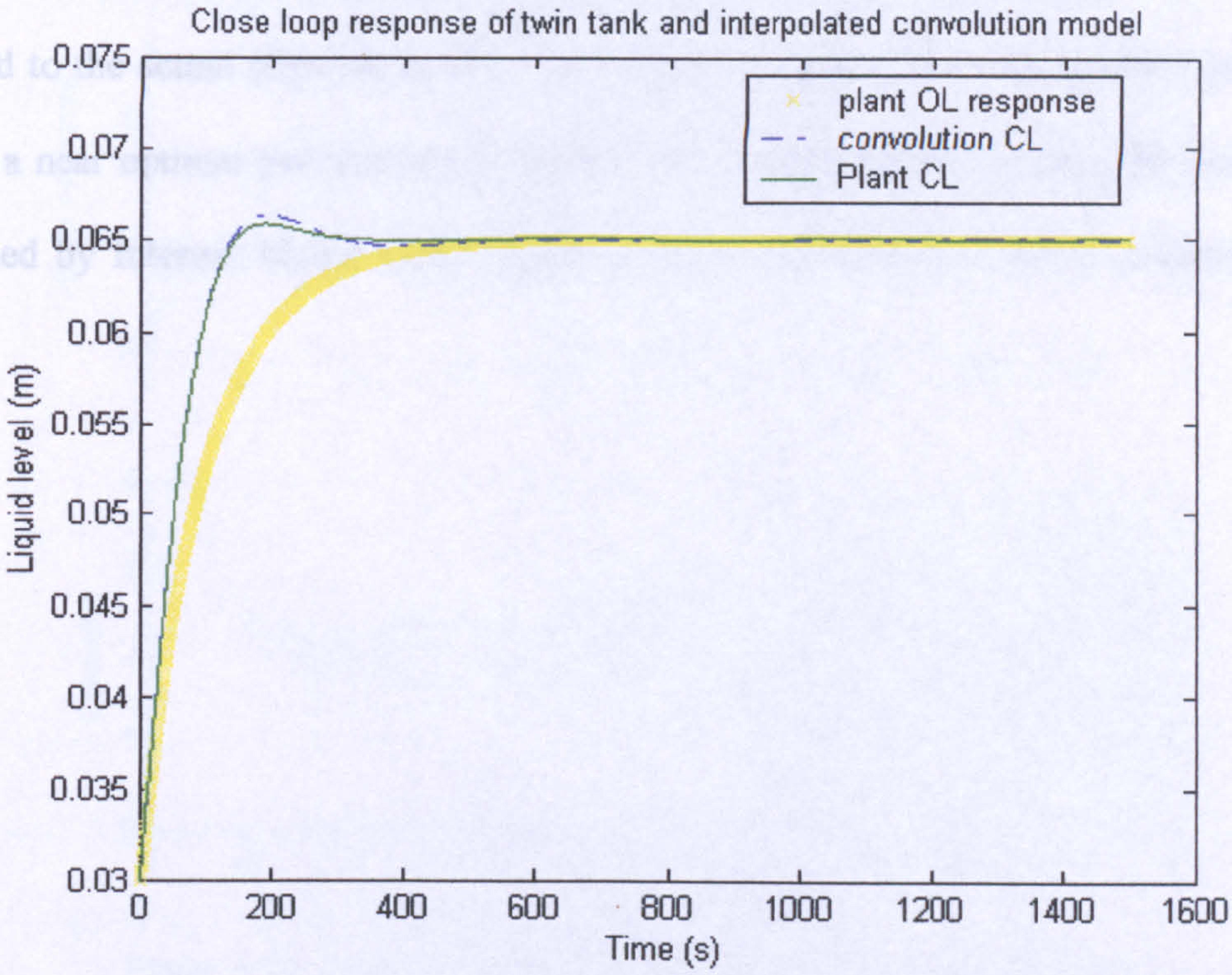

Figure 6.10 Closed loop time responses of TSC via convolution method.

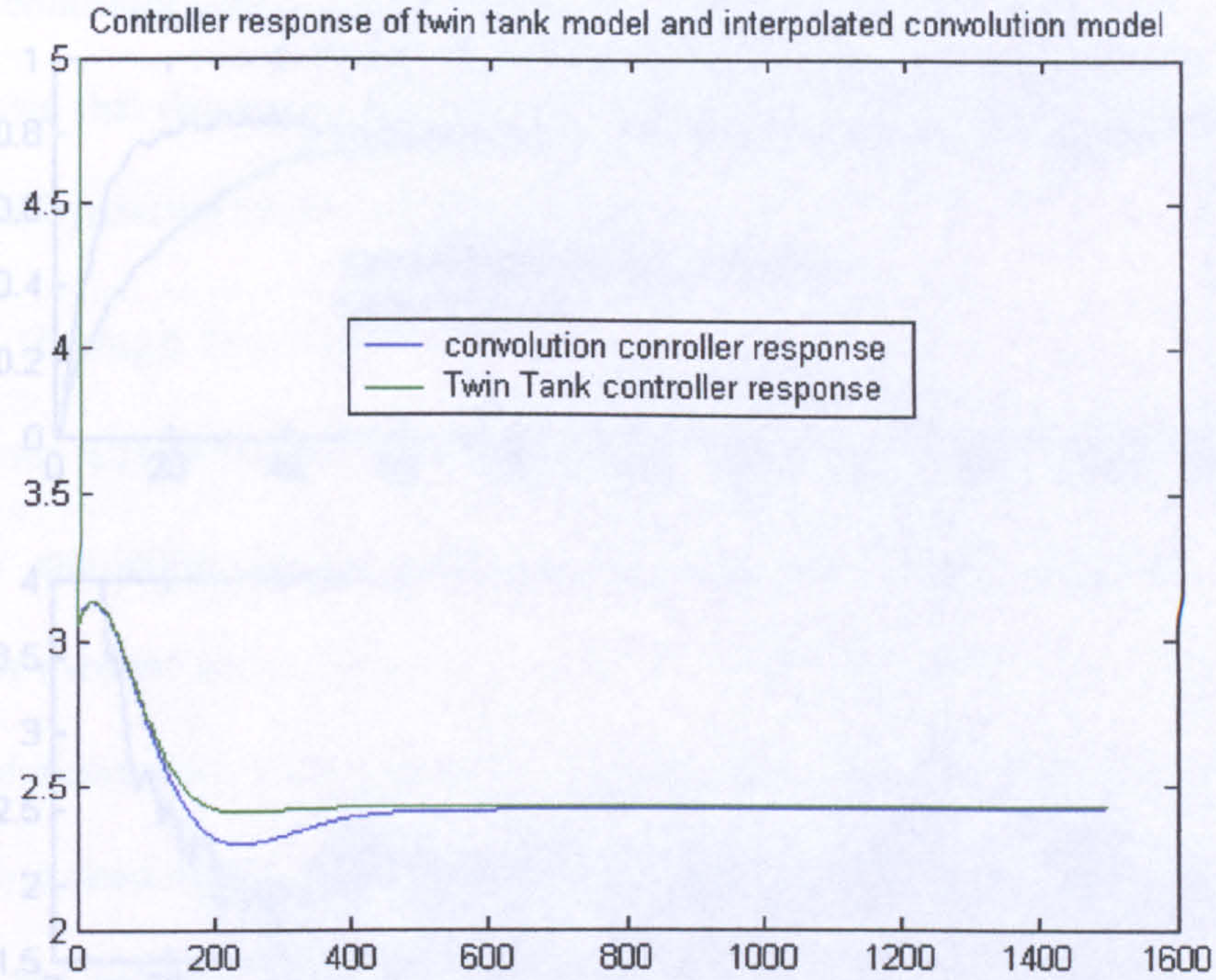

Figure 6.11 Closed loop control signal of TSC via convolution method.

In the next test, a physical step response data is collected from an electrical motor, which contained unknown noise. A PID controller was evolved using the original data and applied to the actual physical model. The close-loop response of this model free designed offers a near optimal performance (Figure 6.12), matching or better than the performance designed by Internal Model Control (IMC) (Dong and Brosilow 1997) methods (Figure 6.13).

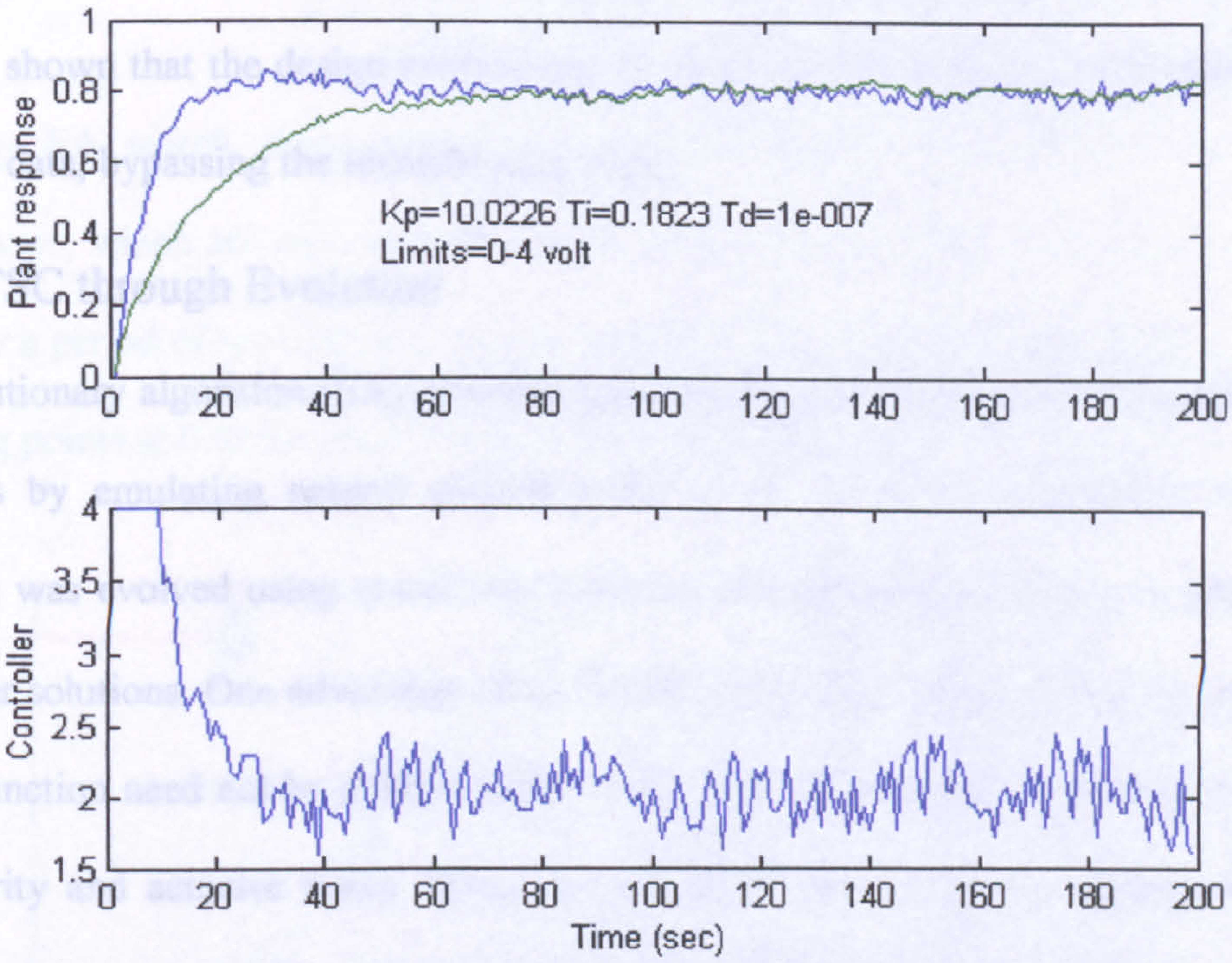

Figure 6.12 Closed loop responses of speed control (model free designed)

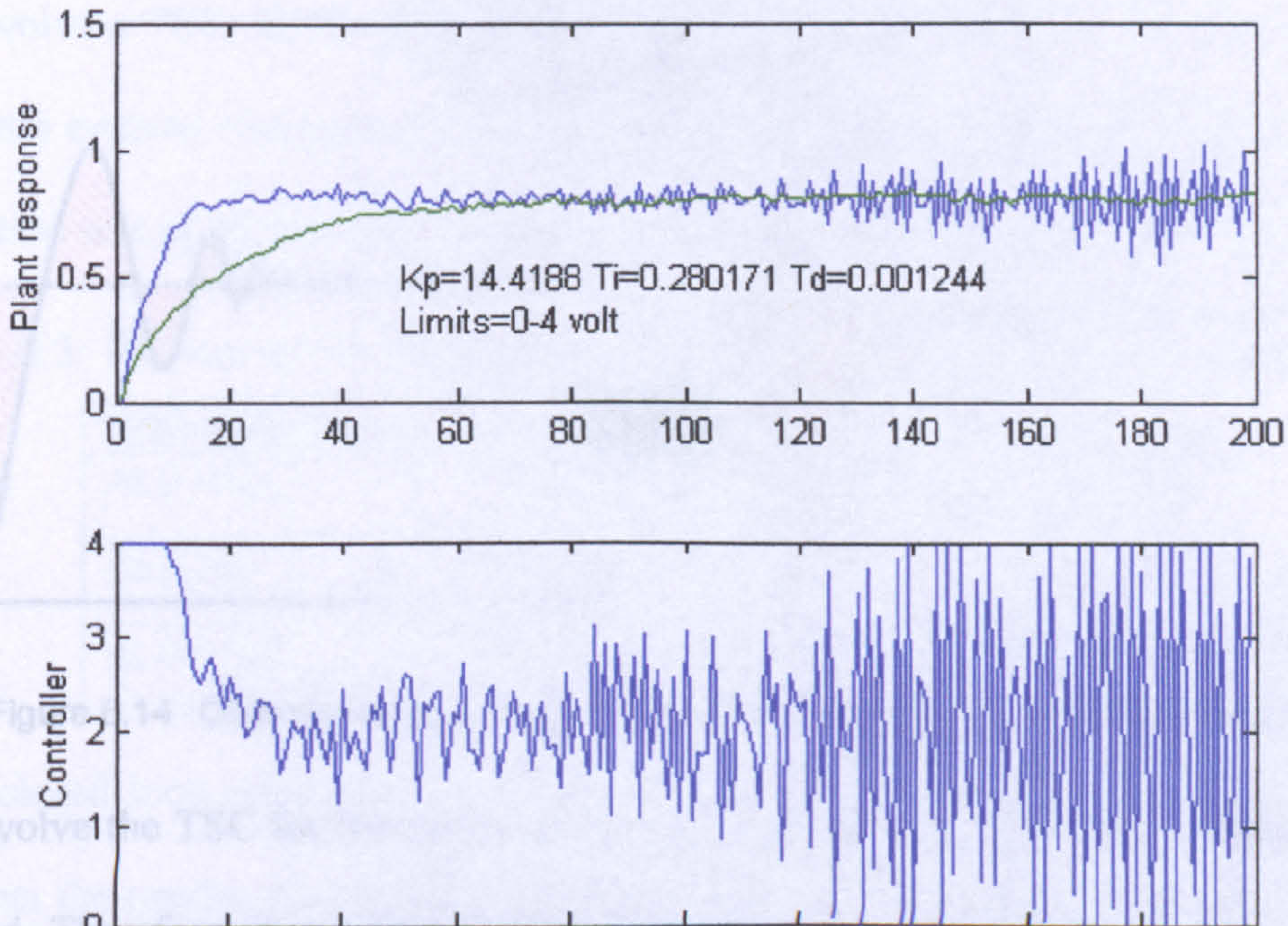

Figure 6.13 Closed loop responses of speed control (IMC designed)

A direct controller design method from step response data has been developed. Results obtained shown that the design method can be automated by efficient evolution from step response data, bypassing the identification stage.

6.3.4 TSC through Evolution

An evolutionary algorithm (EA) provides globally optimal solutions to engineering design problems by emulating natural evolution (Li *et al.* 1996). A population of potential solutions was evolved using crossover, mutation and selection operators to arise at better and better solutions. One advantage of an EA for search and tuning is that the objective or fitness function need not be differentiable. This is useful for global optimisation involving nonlinearity and actuator limits. Here, the objective function to be minimised is the L_1 norm of all errors across the closed loop response within a given time period m as shown in Figure 6.14.

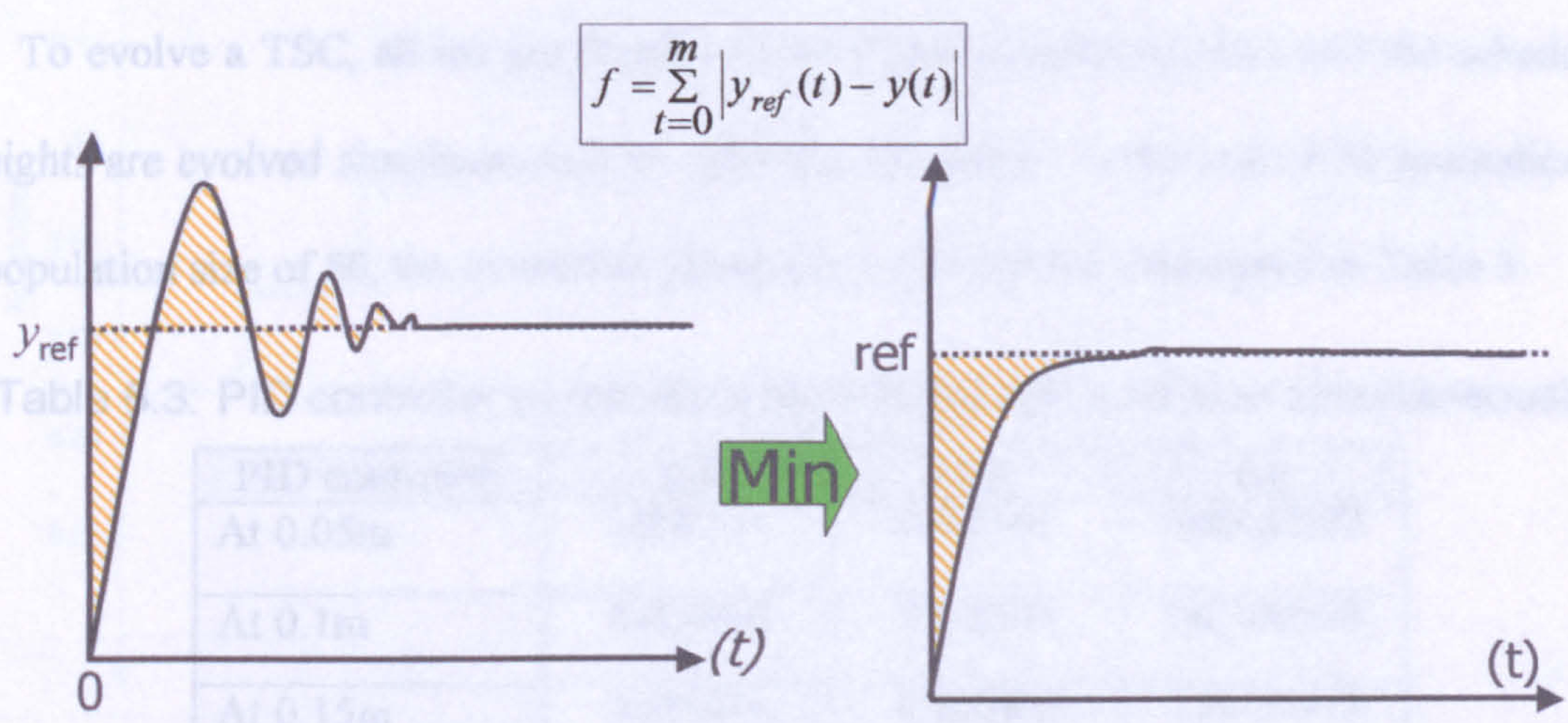

Figure 6.14 Objective function for optimising the controller at one reference point.

To evolve the TSC for the entire operating envelope, a few pre-select reference points were used. Therefore, at n reference operating points the overall cost function is given in equation (4). Where e(t) is the tracking error.

$$J = \sum_{ref=1}^{n} \sum_{t=0}^{m} |e(t)| \tag{6.5}$$

In the EA search, three reference levels were used to evaluate the error tracking performance, which are representative of the whole operating trajectory. Each reference is tested for a period of m=1000 sec. The evolved TSC will be used to test against two unseen operating points at 0.075m and 0.125m as shown in Figure 6.15.

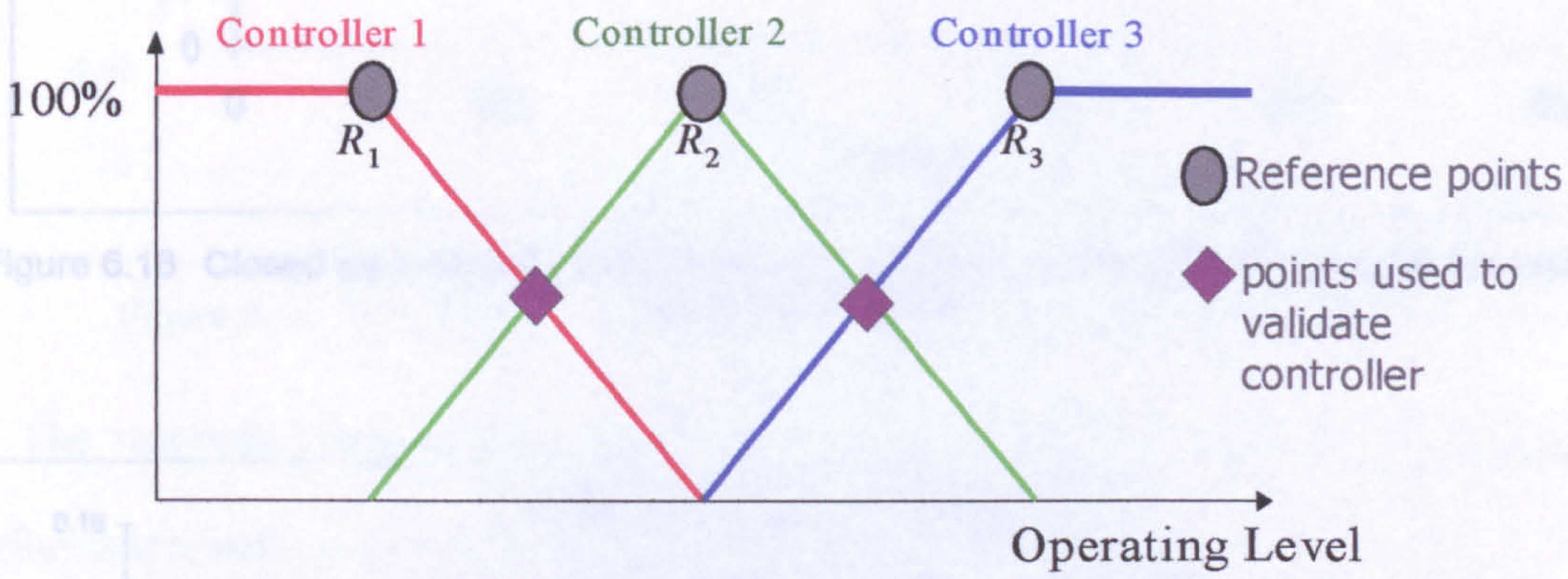

Figure 6.15 Evaluation points in the operating envelope.

To evolve a TSC, all the parameters of the three linear controllers and the scheduling weights are evolved simultaneously in operating envelope. At the end of 50 generations of a population size of 50, the controller parameters obtained were tabulated in Table 3.

Table 6.3: PID controller parameters search through evolution simultaneously.

PID controller	Kp	Ki	Kd
At 0.05m	3500.011	2499.963	1999.93287
At 0.1m	349.9258	500.2581	500.640639
At 0.15m	3.323444	4.834564	7.84161825

The closed loop responses of the finally evolved TSC are shown in Figure 6.16 and 6.17. From the results, it can be seen that the performances are excellent providing a fast optimal solution to the nonlinear control problem.

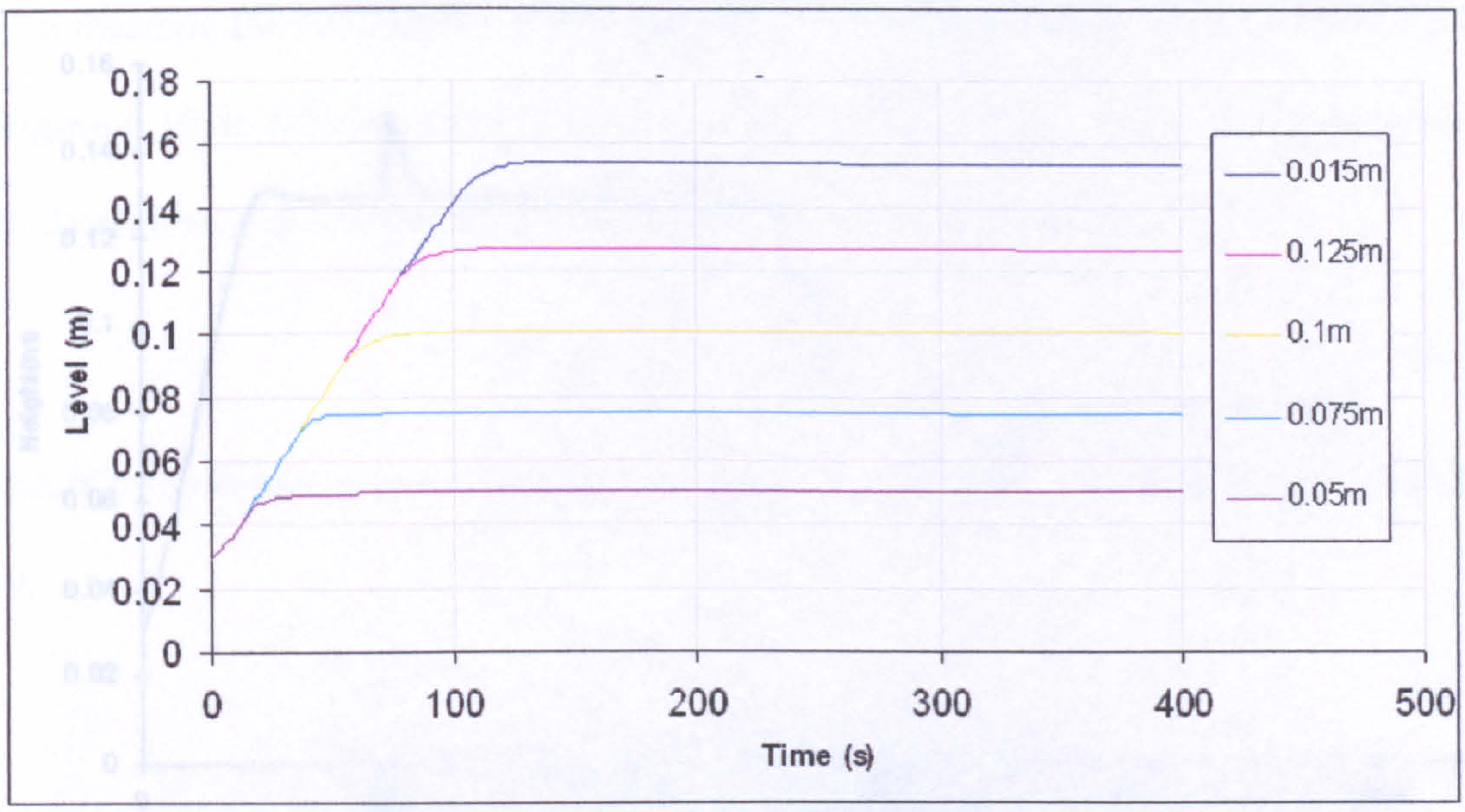

Figure 6.16 Closed loop responses of the TSC at operating points including the unseen ones at 0.075m and 0.125m.

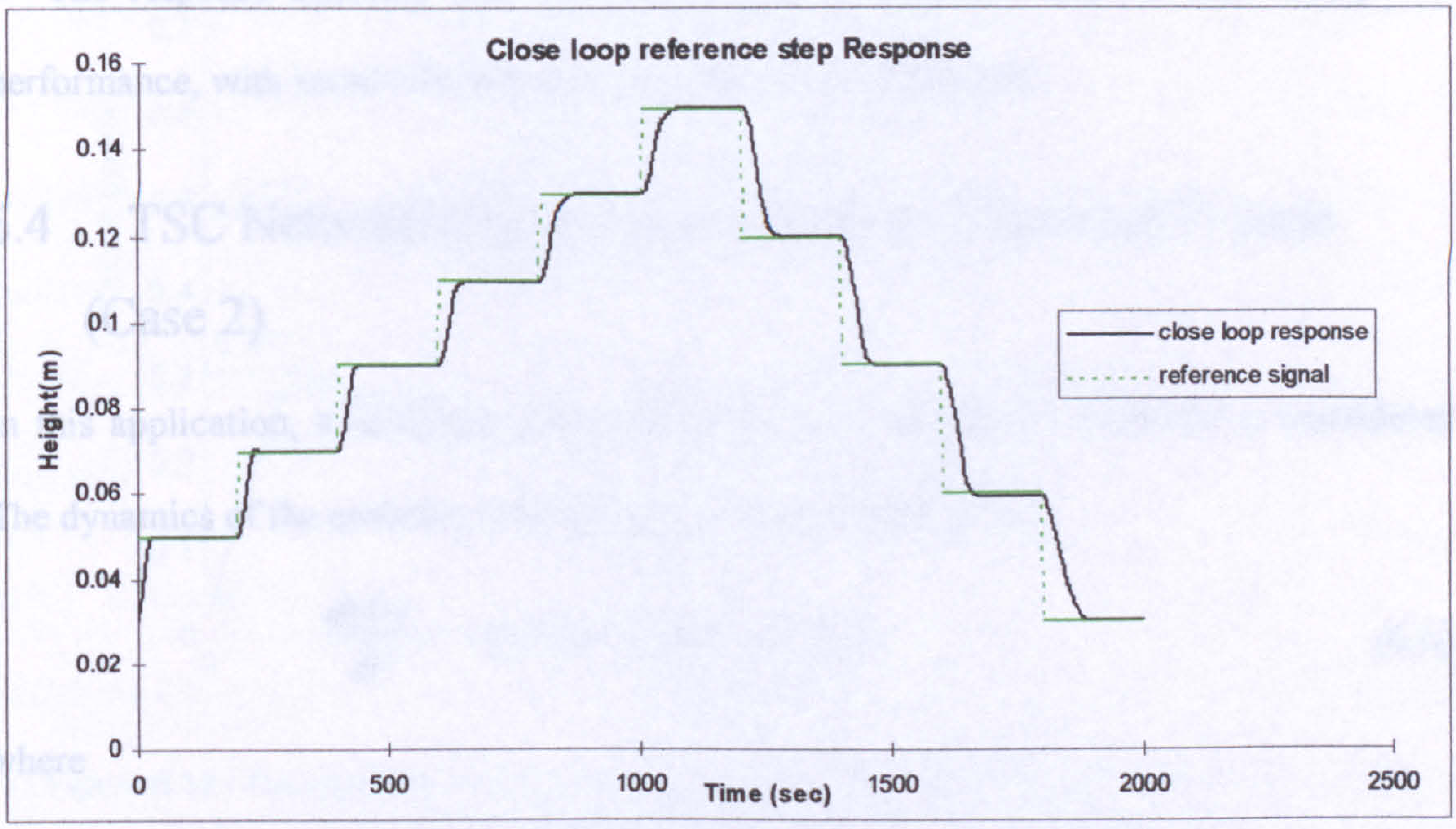

Figure 6.17 Closed loop responses of the TSC at unseen operating points.

Using the same result, the performance of the TSC was tested against some disturbance. A plant disturbance of 0.02m was injected into the plant at t=200 and 700 second as shown in Figure 6.18.

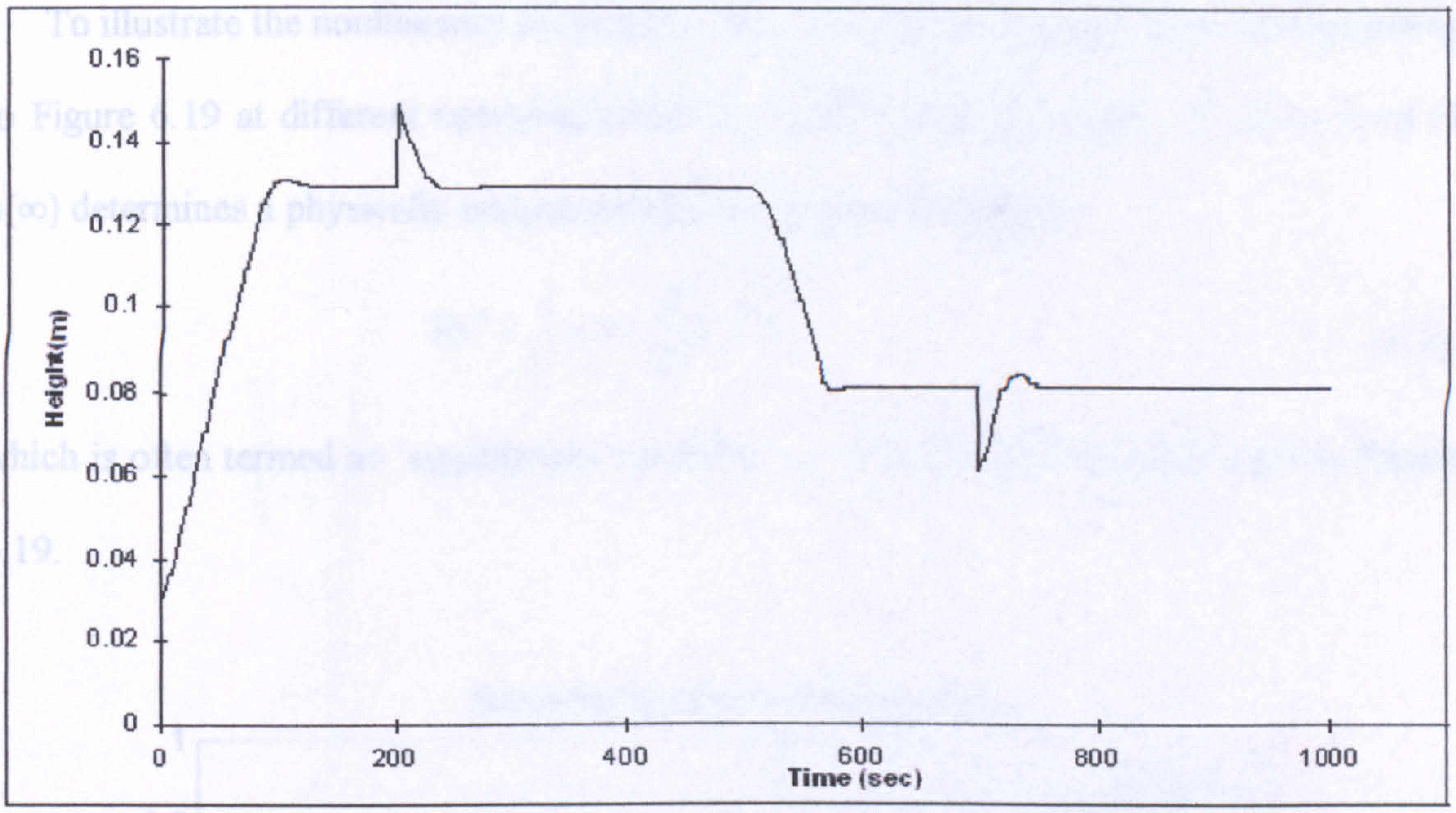

Figure 6.18 Performance of TSC, where disturbance occur at t=200 and 700 s.

The response confirms that the TSC does yield a good transient and steady state performance, with some robustness against the plant uncertainties.

6.4 TSC Network Design for a Nonlinear Chemical Process (Case 2)

In this application, a nonlinear chemical process at Mitsubishi Chemicals is considered. The dynamics of the constant-temperature reaction is modelled by:

$$\frac{dy(t)}{dt} = -Ky^2(t) + \frac{1}{V}[d - y(t)]u(t) \tag{6.6}$$

where

$y(t)$ = concentration in the outlet stream (mol/l)

$u(t)$ = flow rate of the feed stream (l/h)

K = rate of reaction (l/mol/l·h)

V = reactor volume (l)

d = concentration in the inlet stream (mol/l)

To illustrate the nonlinearity of the plant, the static model at steady-state corresponding to Figure 6.19 at different operating points or equilibria may be used. A given level of $u(\infty)$ determines a physically unique $y(\infty)$ by the parabolic equation:

$$Ky^2 + \frac{1}{V}u\,y - \frac{d}{V}u = 0 \tag{6.7}$$

which is often termed an 'equilibrium manifold', as illustrated by the solid curve in Figure 6.19.

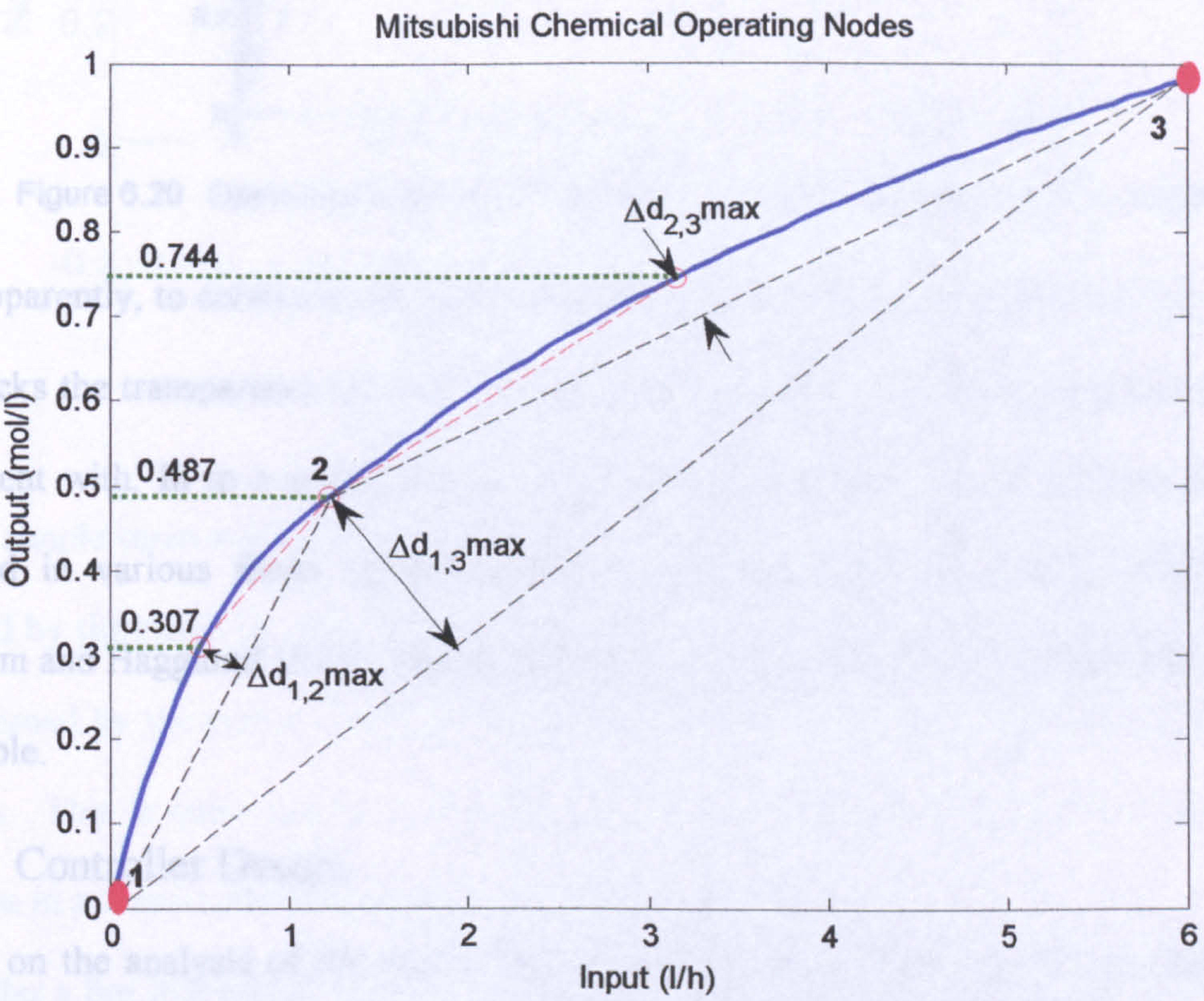

Figure 6.19 The equilibrium manifold of the nonlinear process within its operating envelope

A static model is often the first step in investigating a nonlinear process. The model can be used to determine the range of control signals, the sizes of the actuators and the resolution of selected sensors. In practice, such a static model can be obtained either from close-loop or open-loop tests as shown in Figure 6.20. Note that the tests have a physical interpretation only for a stable process (Åström and Hagglund 1995).

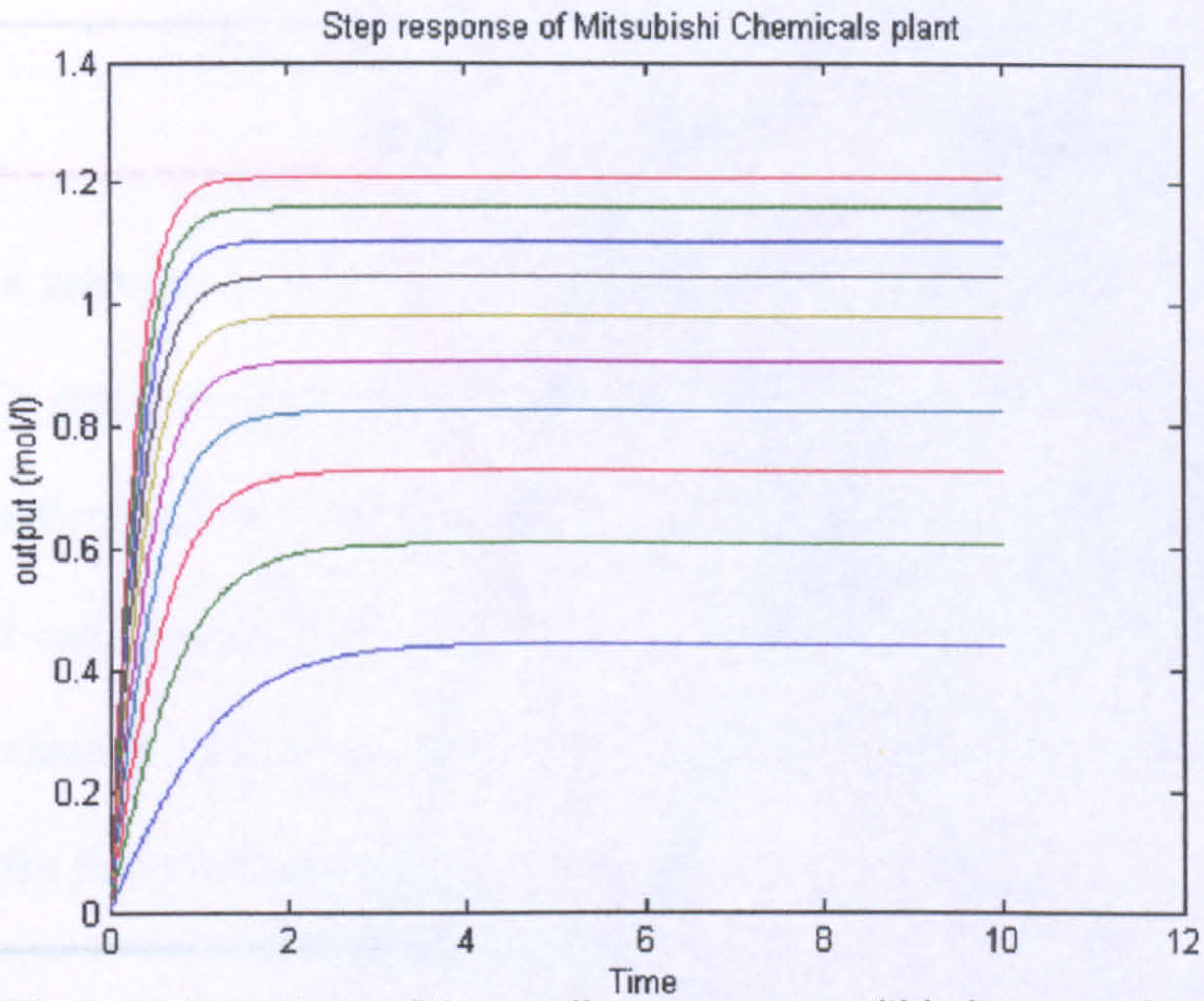

Figure 6.20 Open-loop tests of the nonlinear process within its operating envelope

Apparently, to control such a nonlinear process, a nonlinear controller may be used, but this lacks the transparency on stability and familiarity that a practising engineer would be confident with. In to a recent survey, over 90% of industrial control systems in use are realised in various forms of proportional plus integral plus derivative (PID) control (Åström and Hagglund 1995). Hence, the use and design of a simple PID control system is desirable.

6.4.1 Controller Design

Based on the analysis of the process and its model, clearly, the use of a straightforward PID control would be inadequate. For such a nonlinear plant, therefore, the use of a TSC is proposed (Chong and Li 2002b). In this application, each node of the TSC is a straightforward three-term PID controller, placed along the operating trajectory as shown in Figure 6.19. The nodes found at [0.307, 0.487, 0.744] will be interpolated as shown in Figure 6.21.

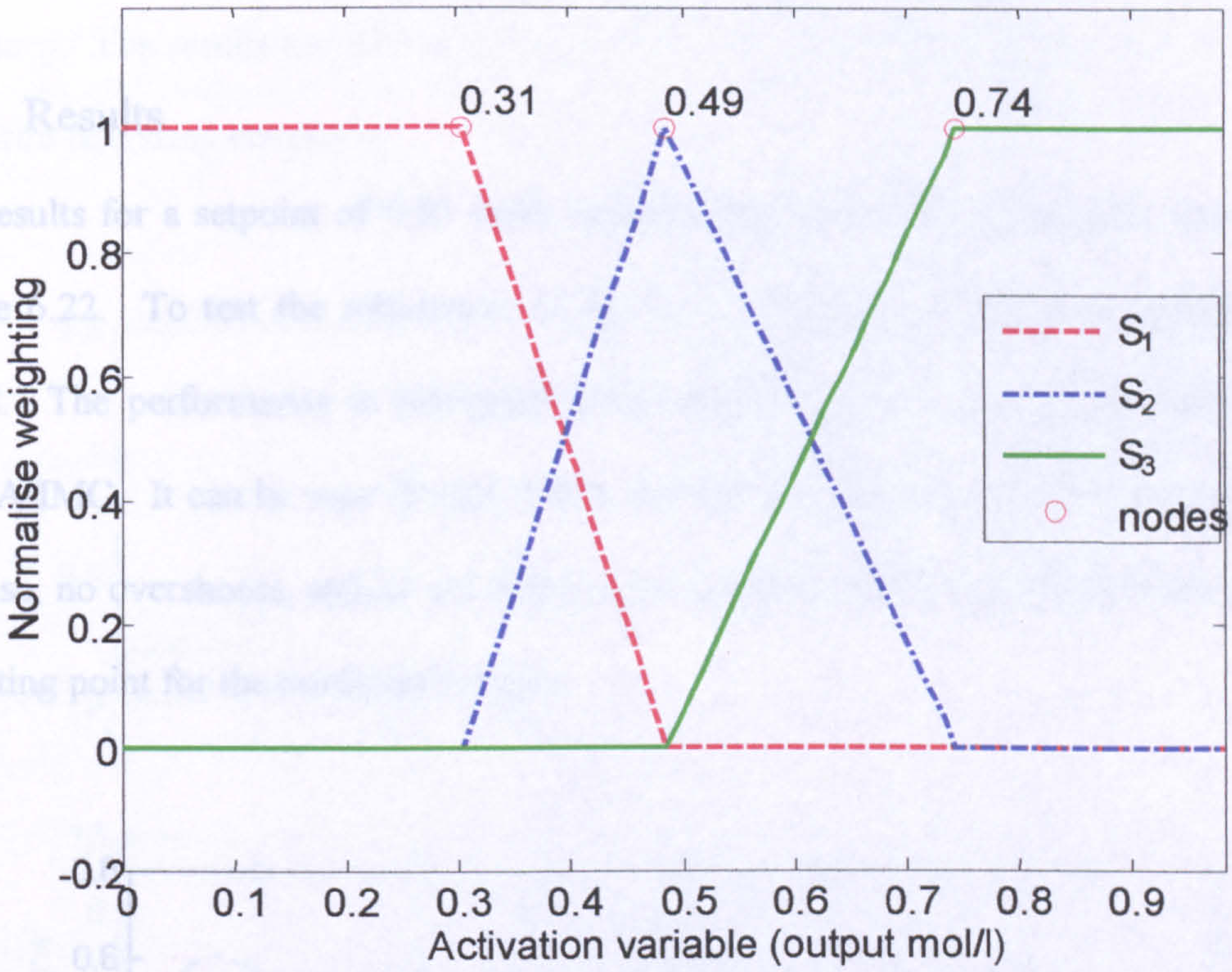

Figure 6.21 Weighting function of TSC

A simple three-node TSC is designed, under a hard constraint on the input flow rate as limited by the range [0 10] l/h. The scheduling between the three PID controller nodes is determined by the output levels, using simple triangular-shaped activation functions S_1, S_2 and S_3. This is quite similar to assigning the degree of memberships in the fuzzification process in a fuzzy control system (Chowdhury and Li 1998).

After a fast individual local PID tuning from IMC tuning rules,

$$u(t) = [S_1 \quad S_2 \quad S_3]\begin{bmatrix} 9.82 & 1.22 & 0.0376 \\ 15.6 & 0.784 & 0.0241 \\ 28.6 & 0.481 & 0.0137 \end{bmatrix}\begin{bmatrix} 1 \\ p^{-1} \\ p \end{bmatrix} e(t) \tag{6.8}$$

the TSC is then optimised through MOEA. The overall controller hence takes the form:

$$u(t) = [S_1 \quad S_2 \quad S_3]\begin{bmatrix} 35.76 & 0.751 & 382.8 \\ 24.98 & 0.268 & 74.58 \\ 21.68 & 0.189 & 451.68 \end{bmatrix}\begin{bmatrix} 1 \\ p^{-1} \\ p \end{bmatrix} e(t) \tag{6.9}$$

where p is the differentiation operator.

6.4.2 Results

The results for a setpoint of 0.53 mol/l, specified by Mitsubishi Chemicals, are shown in Figure 6.22. To test the robustness of the TSC regulator, a 20% load disturbance was added. The performance is compared with PID controllers obtained through IMC and MOEA-IMC. It can be seen (Figure 6.22) that the TSC offers a good performance with a fast rise, no overshoots, and an extremely good rejection to the load disturbance at various operating point for the nonlinear process.

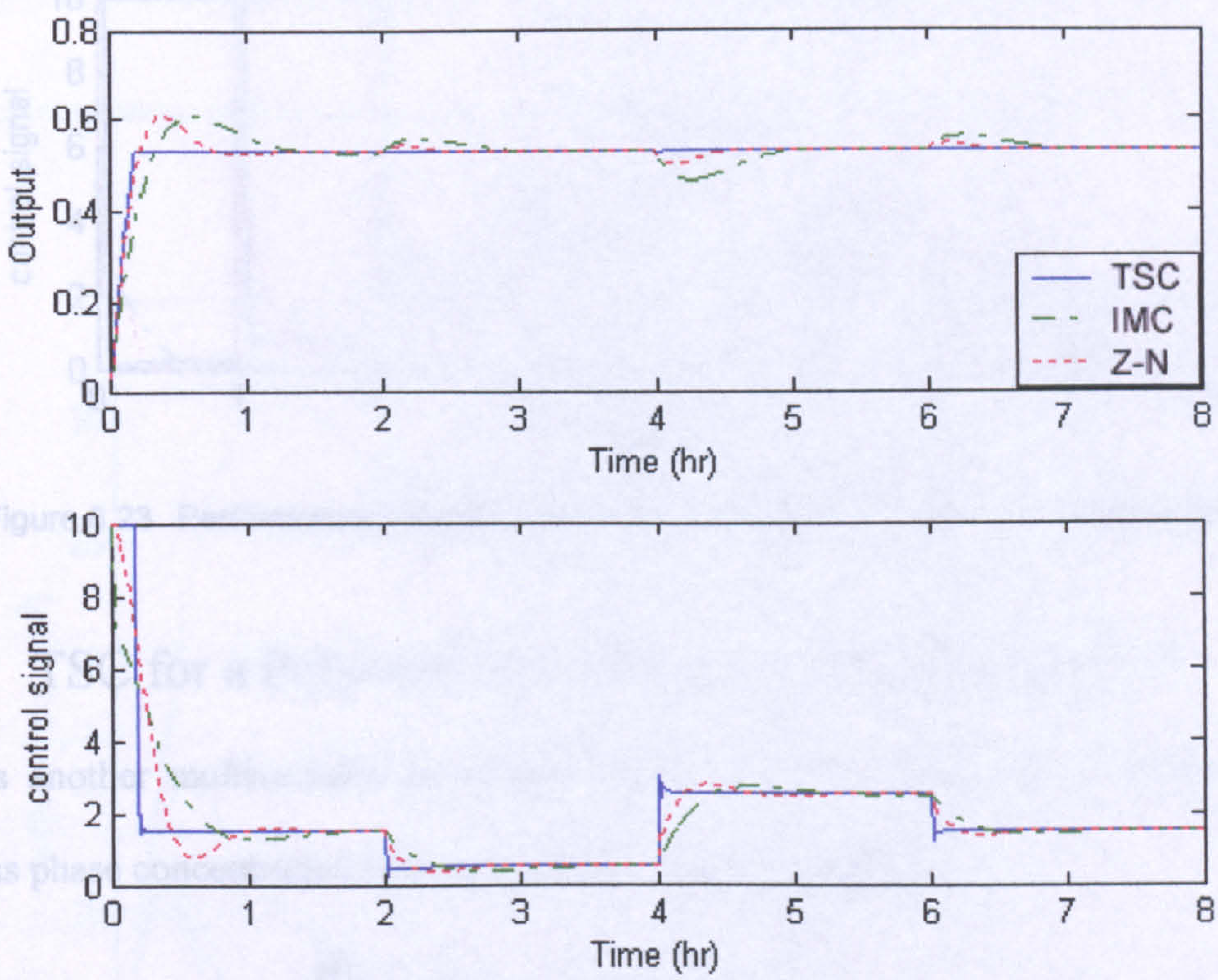

Figure 6.22 Performance of the TSC regulator subject to a 20% load disturbance

In essence, a Linear Time Invariant (LTI) building block based TSC is a nonlinear controller overall. The switching between the nodes is through soft activation and hence imposed no threat of actuator damage. To thoroughly test the performance of the PID network designed for nonlinear, the system was driven throughout the allowed operating trajectory. The results are shown in Figure 6.23. It can be seen that it is indeed reliable in the entire operating envelope.

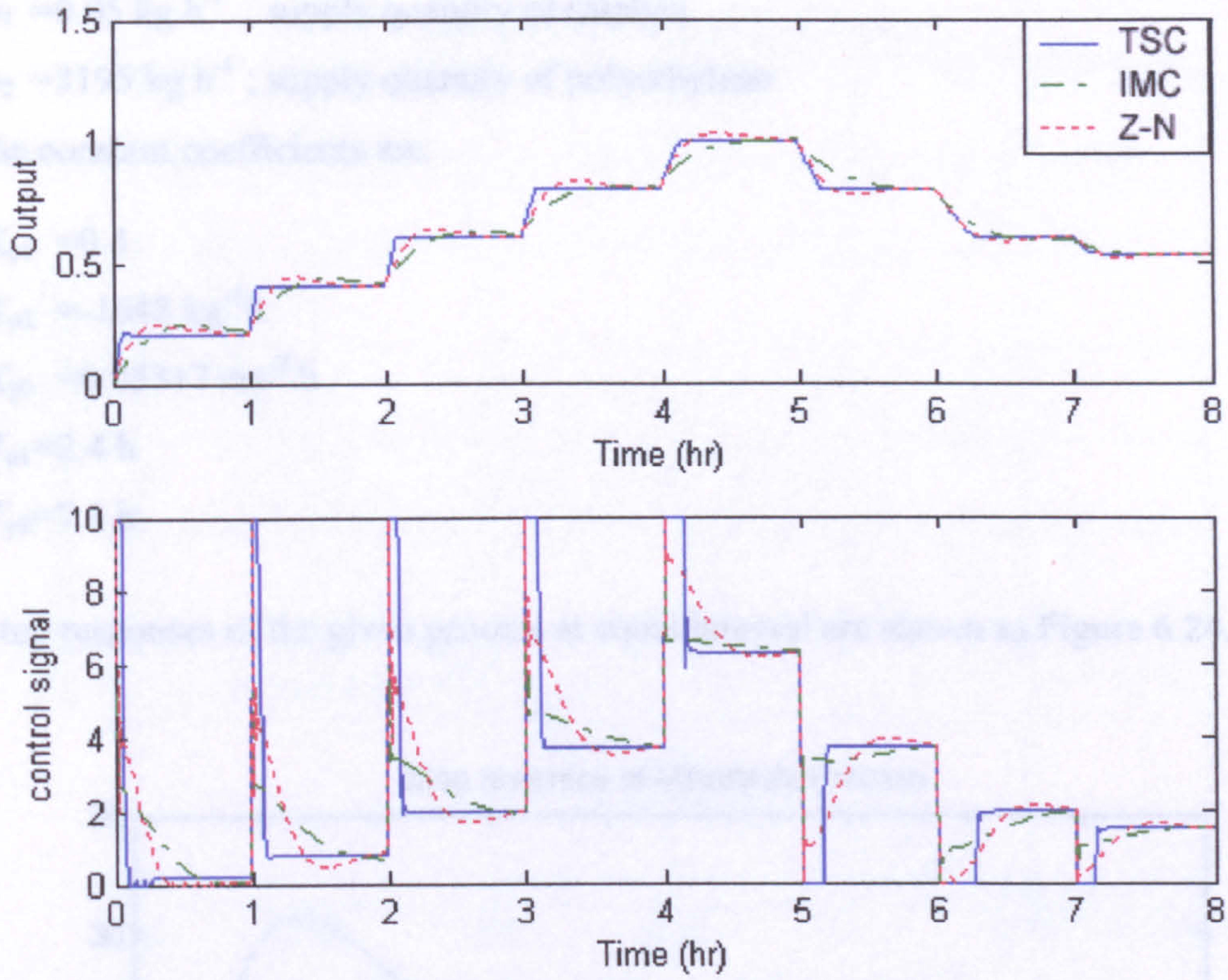

Figure 6.23 Performance reliability of the PID network in the entire operating envelope

6.5 TSC for a Polymerisation Reaction Process (Case 3)

This is another multivariable nonlinear system, but the control task is SISO, where a gaseous phase concentration of polymerisation reaction is given by

$$\frac{dx_1}{dt} = \frac{1}{T_{p1}}(-x_1 + K_{p1}u_1) \tag{6.10}$$

$$\frac{dx_2}{dt} = \frac{1}{T_{p2}}(K_{p2}x_1x_2 - x_2 + K_{p3}u_2) \tag{6.11}$$

$$y = x_2 \tag{6.12}$$

where the operating point is at: The control task is to use u1 to control x2.

x_1 =0.02 kg h^{-1} ; consumption velocity of catalyst

x_2 =5.0 kg cm^{-2} ; gas density

u_1 =0.05 kg h^{-1} ; supply quantity of catalyst

u_2 =3195 kg h^{-1} ; supply quantity of polyethylene

and the constant coefficients are:

K_{p1} =0.4

K_{p2} =-1648 $kg^{-1}h$

K_{p3} =0.05317 cm^{-2} h

T_{p1}=2.4 h

T_{p2}=7.1 h

The step responses of the given process at equal interval are shown as Figure 6.24.

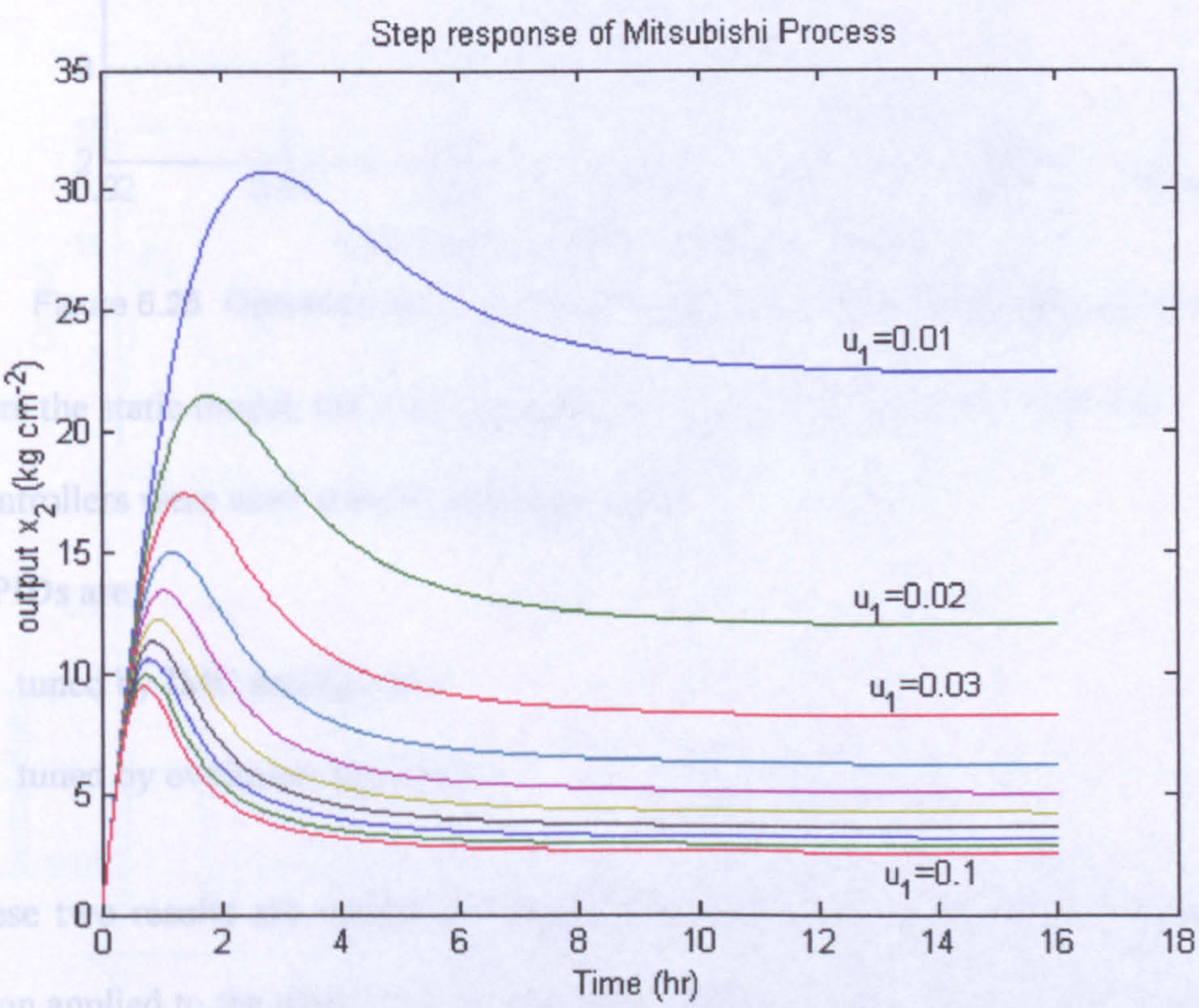

Figure 6.24 Open-loop step response from (0.01-0.1) at equal-interval of 0.01 (kg h-1)

From Figure 6.24, the effective range of the output should be around 5 kg cm^{-2}. Assume that output will be regulated from 2-10 kg cm^{-2}. A static model was constructed for this operating range as shown in Figure 6.25.

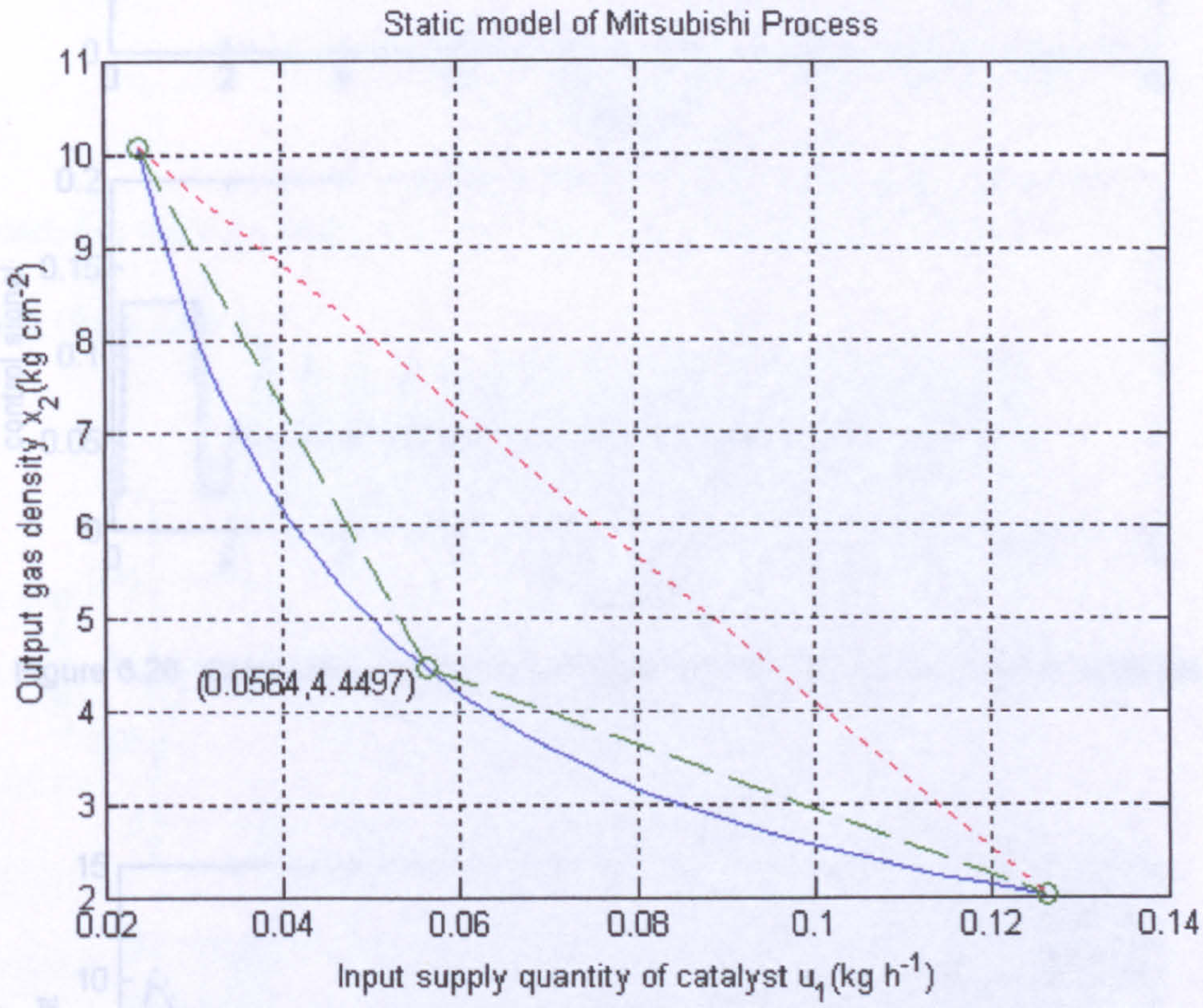

Figure 6.25 Operating range x2=[2-10] (kg cm-2); u1=[0.0235-0.1269] (kg h-1)

From the static model, the TSC operating node found to be at [2 4.4497 10].

PID controllers were used at these operating nodes.

These PIDs are:

a) tuned by IMC tuning rules

b) tuned by evolution algorithm

These two results are compared. Figure 6.26 and Figure 6.27 shows the various test condition applied to the plant. It is obvious that EA tuning is superior to IMC tuning.

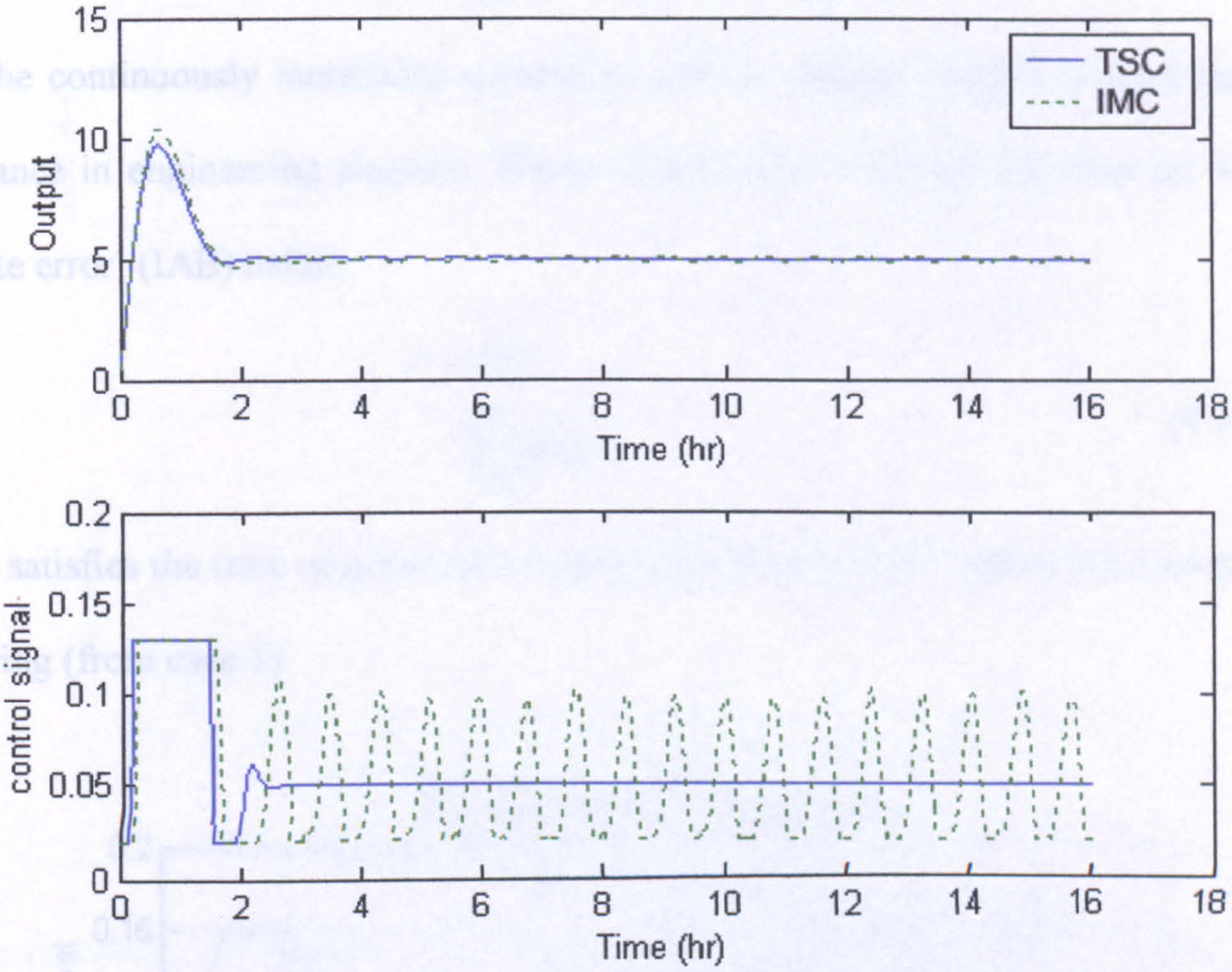

Figure 6.26 Close–loop response. r(t)=5kg cm^{-2} with [x_1,x_2]=[0,0] initial condition.

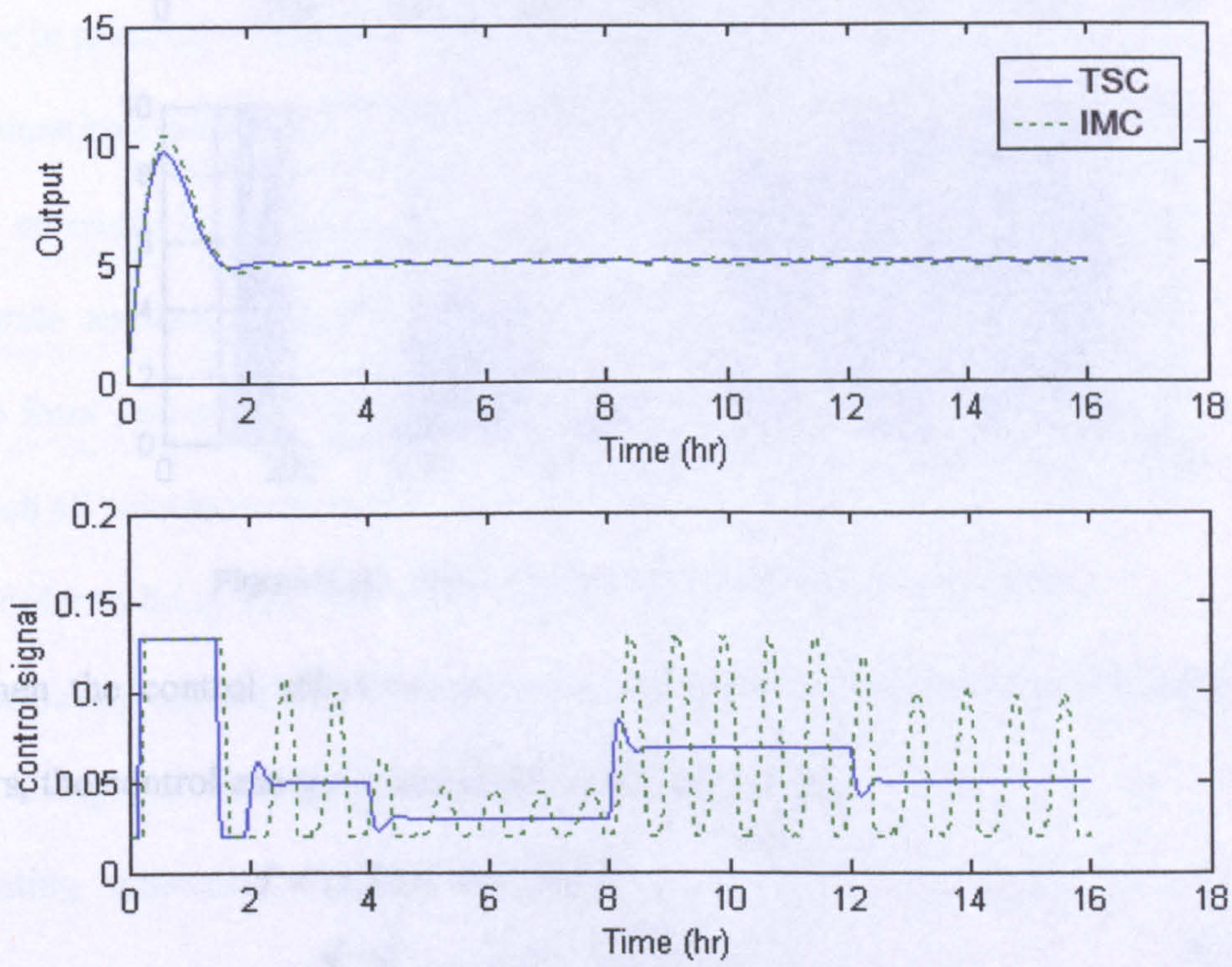

Figure 6.27 Close–loop response. r(t)=5kg cm^{-2} with [x_1,x_2]=[0,0] initial condition, and 20% load disturbance.

6.6 Multi-Objective Design

With the continuously increasing computing power, optimal control is receiving a wider acceptance in engineering practice. When optimisation is carried out over an 'integral of absolute error' (IAE) index,

$$J = IAE = \sum_{t=0}^{n} |e(t)| \tag{6.13}$$

it only satisfies the time response of a system (see Figure 6.28), where the control signal is chattering (from case 1)

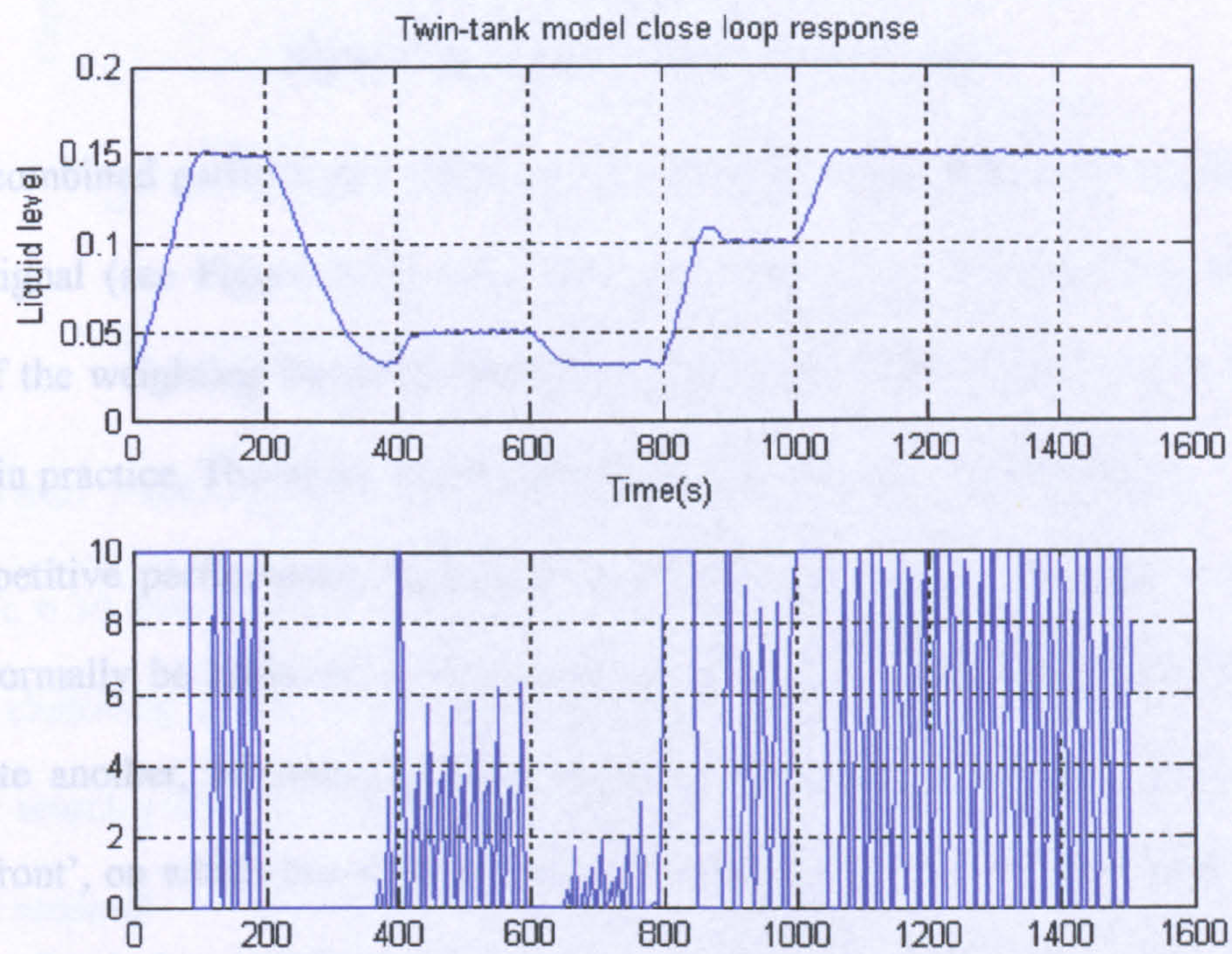

Figure 6.28 Good transient but chattering control signal.

When the control effort needs to be considered alongside the IAE performance it delivers, the control energy is added to augment the index.

$$J = \alpha_1 IAE + \alpha_2 IAED = \sum^{s} \sum_{t=0}^{n} \alpha_1 |e(t)| + \alpha_2 \left| \frac{du}{dt}(t) \right| \tag{6.14}$$

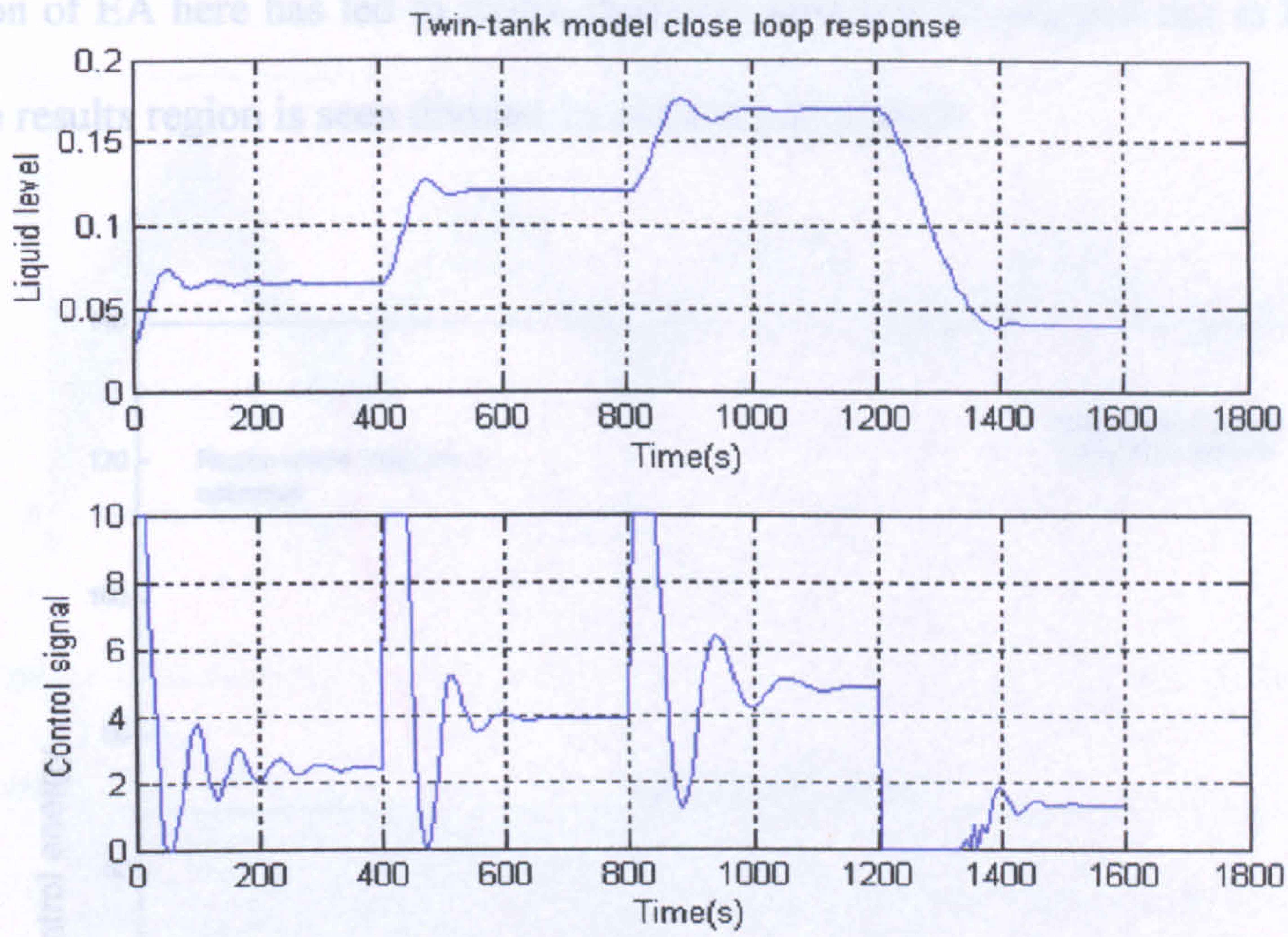

Figure 6.29 Suppressing the control signal

The combined performance index into one fitness proves effective in suppressing the control signal (see Figure 6.29), note that it is not a true multi-objective solution, the choice of the weighting factors between the errors and controls proves to be a nontrivial exercise in practice. Therefore, a true multi-objective solution is preferable.

Competitive performance indicators are usually not reconcilable, e.g., a small error cannot normally be achieved with a small control effort. As solving one problem may deteriorate another, all resulting 'equally optimal' solutions will thus form a so-called 'Pareto front', on which one solution is not dominantly 'better off' than another and hence on which all solutions are the 'best' in terms of multiple objectives (Herreros *et al.* 2000).

In order to bring out, and verify the optimality of, Pareto solutions, the IMC method is conventionally applied here to the design of a PID controller as a benchmark or reference point. Using the IMC reference, the search space for optimisation can be reduced, eliminating unnecessary processing time beyond acceptable stability margins. The application of EA here has led to multi-objective solutions all mapped out in Figure 6.30, where the results region is seen divided by the reference point.

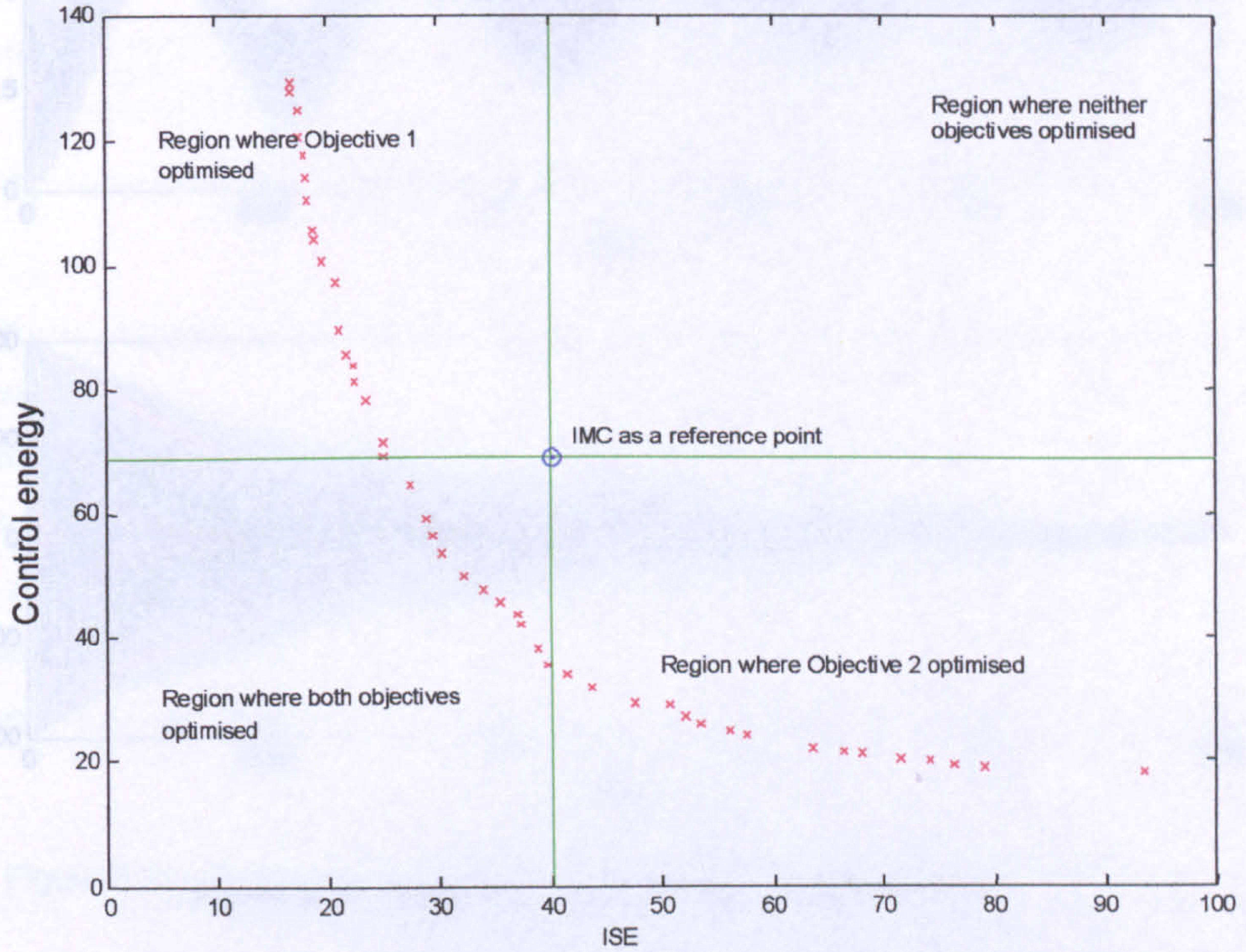

Figure 6.30 Pareto optimal solutions divided by a conventional design as a reference point.

Figure 6.30 also reveals how much improvement in reducing both control error and potential chattering by the weightless Pareto optimisation technique. This EA based multi-objective search yields much balanced optimal solutions compared with conventional and objective methods.

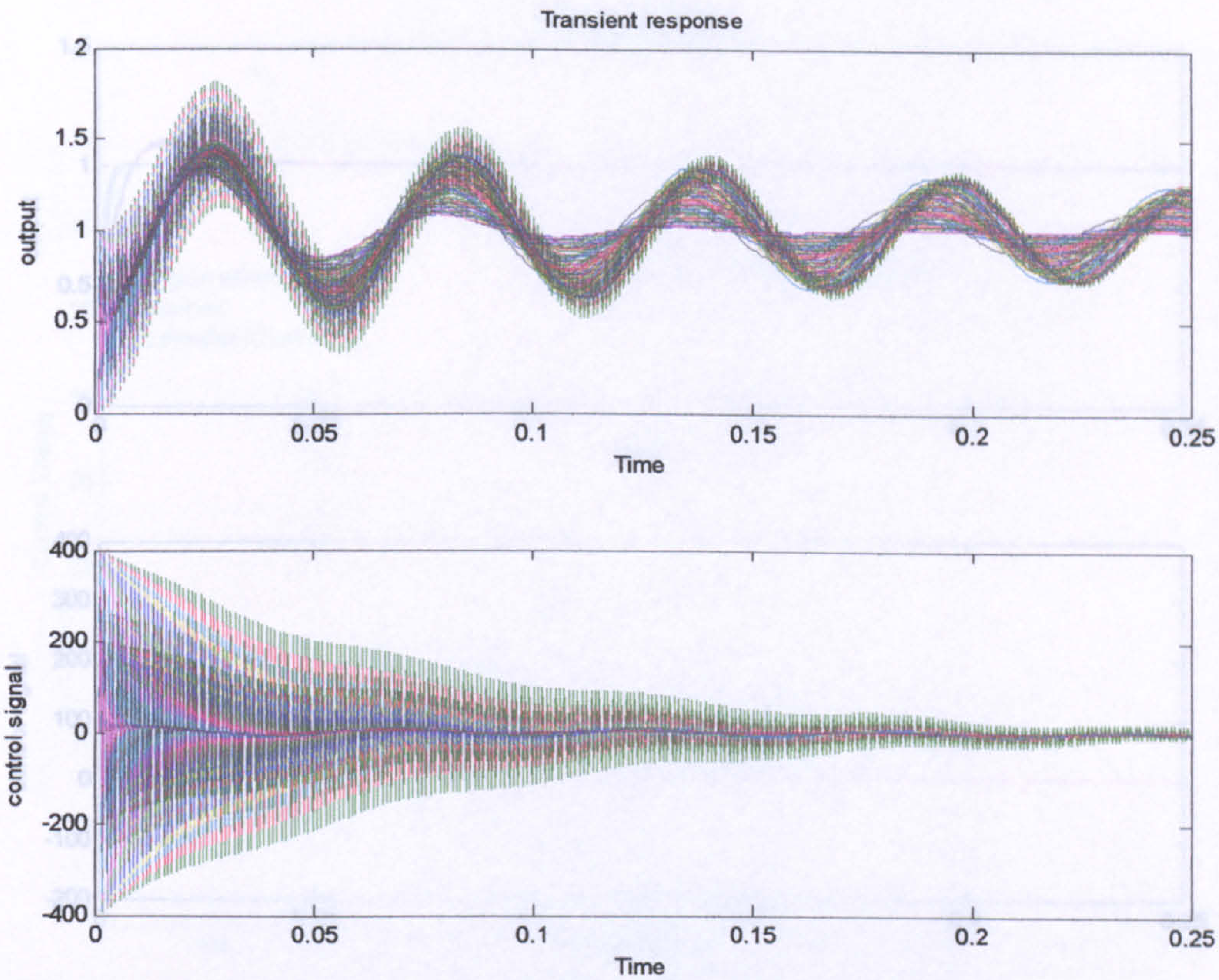

Figure 6.31 Performance of all TSC on Pareto front

Figure 6.31 shows that all the TSC controllers fall on the Pareto front was stable although some oscillation occurs.

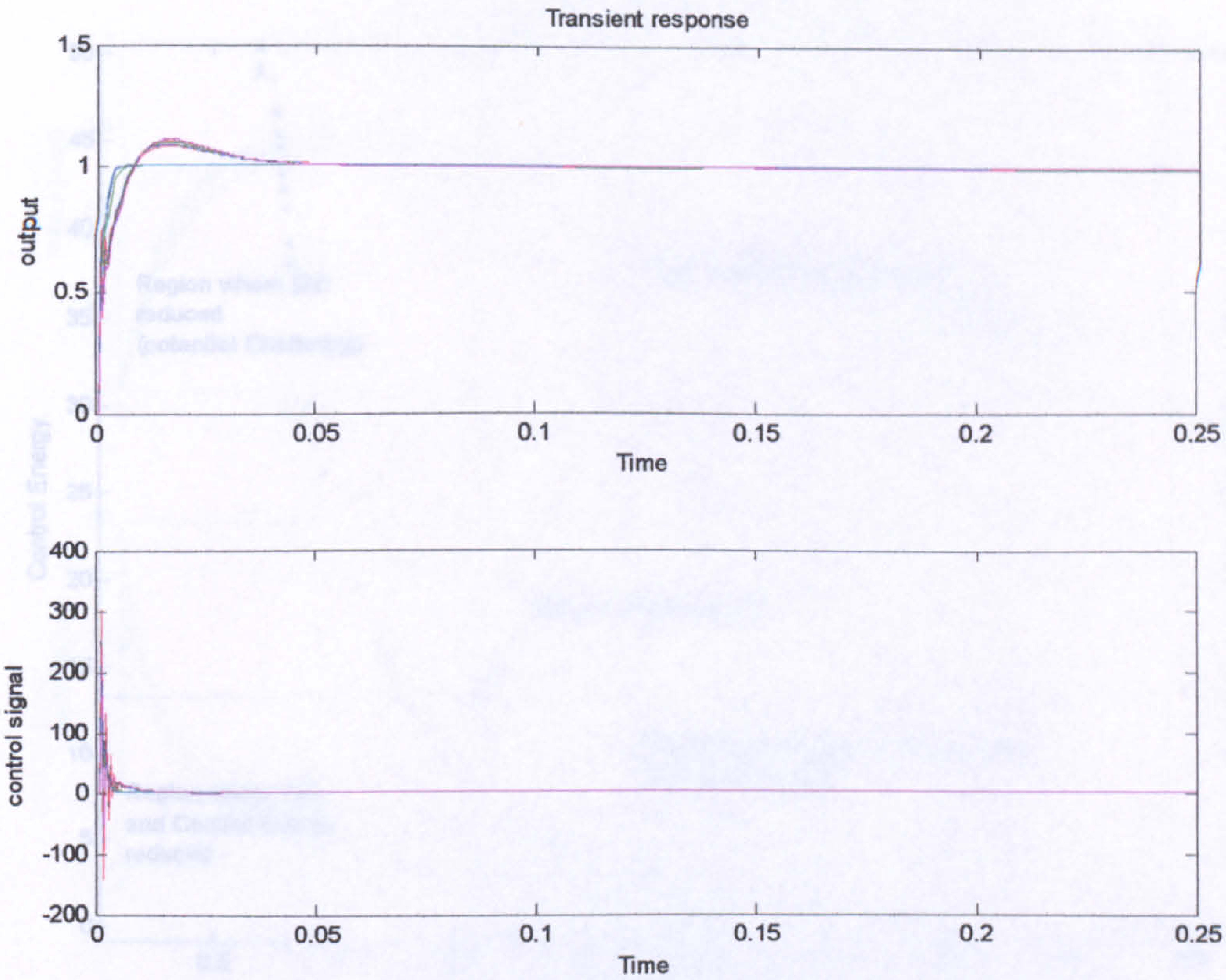

Figure 6.32 Performance of TSC fall in region with both objectives optimised

Figure 6.32 shows that the TSC controllers fall on the region where both objectives are optimised have an excellent step response and less control signal chattering.

6.7 Guided Search

In this example, a nonlinear chemical process (case 2) similar to section 6.4 is considered. The application of Multi-Objective Evolutionary Algorithm (MOEA) here has led to multi-objective solutions all mapped out in Figure 6.33, where the result was divided into four regions by the IMC reference point.

It also reveals how much improvement in reducing both control error and potential chattering by the weightless Pareto optimisation technique. This MOEA based multi-objective search yields much more balanced optimal solutions compared to conventional single objective methods.

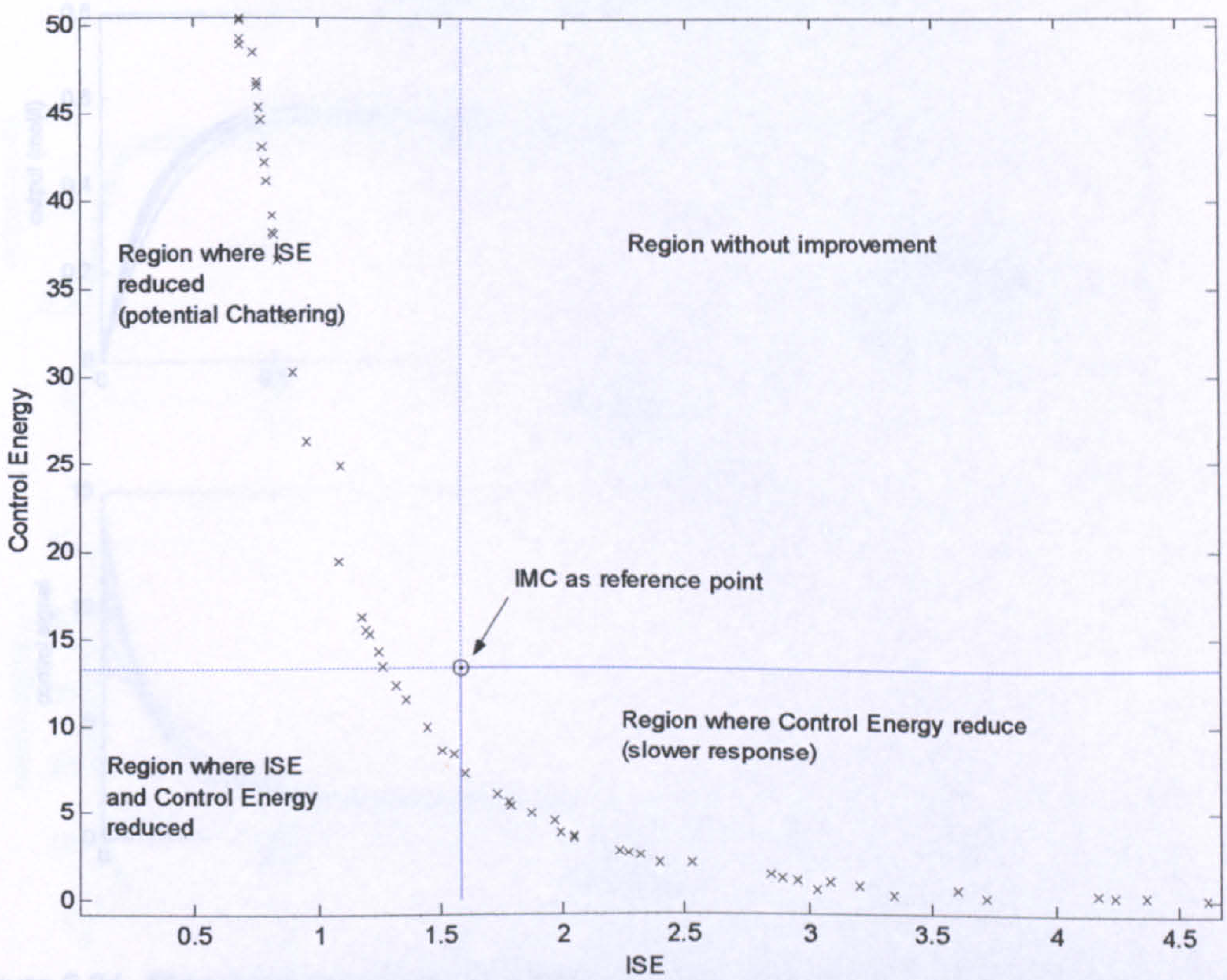

Figure 6.33 Pareto optimal solutions divided by a conventional design as a reference point.

Figure 6.34 shows the solution found in the region where there are improvements made on both objectives compared to the IMC design.

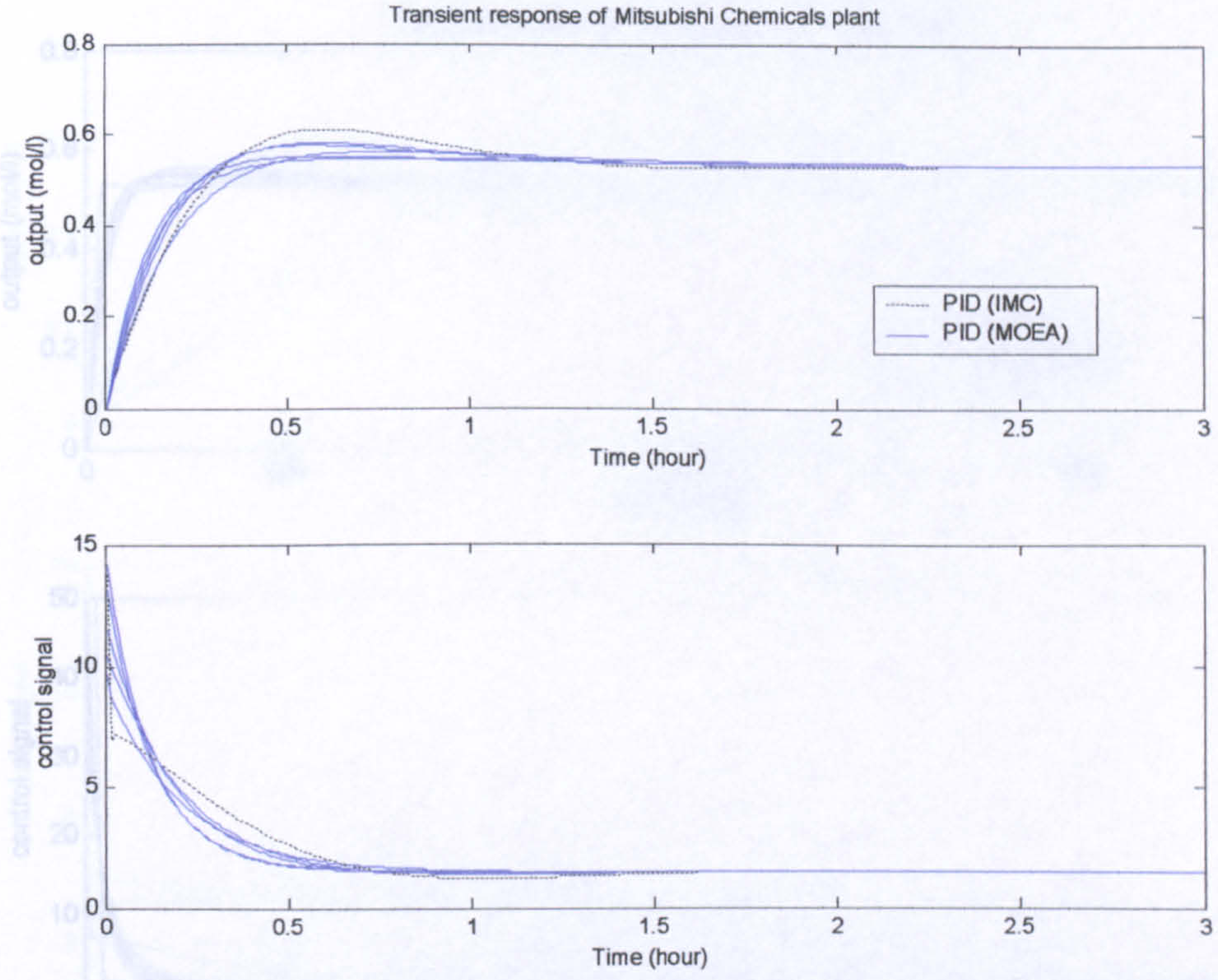

Figure 6.34 Step response of individual solution fall in the region with both improvements compared to IMC tuning.

By comparing these solutions with IMC, it is obvious that any one of these solutions exceeds the performance of the IMC design without the need of adding an additional weighting when choosing a controller from the *non-dominated* solutions set.

The step responses of the solutions fall in region where only ISE was minimised are plotted in Figure 6.35. Those responses look very ideal, thus, are not practical to implement in real physical systems. Therefore, these solutions will be eliminated when choosing a controller.

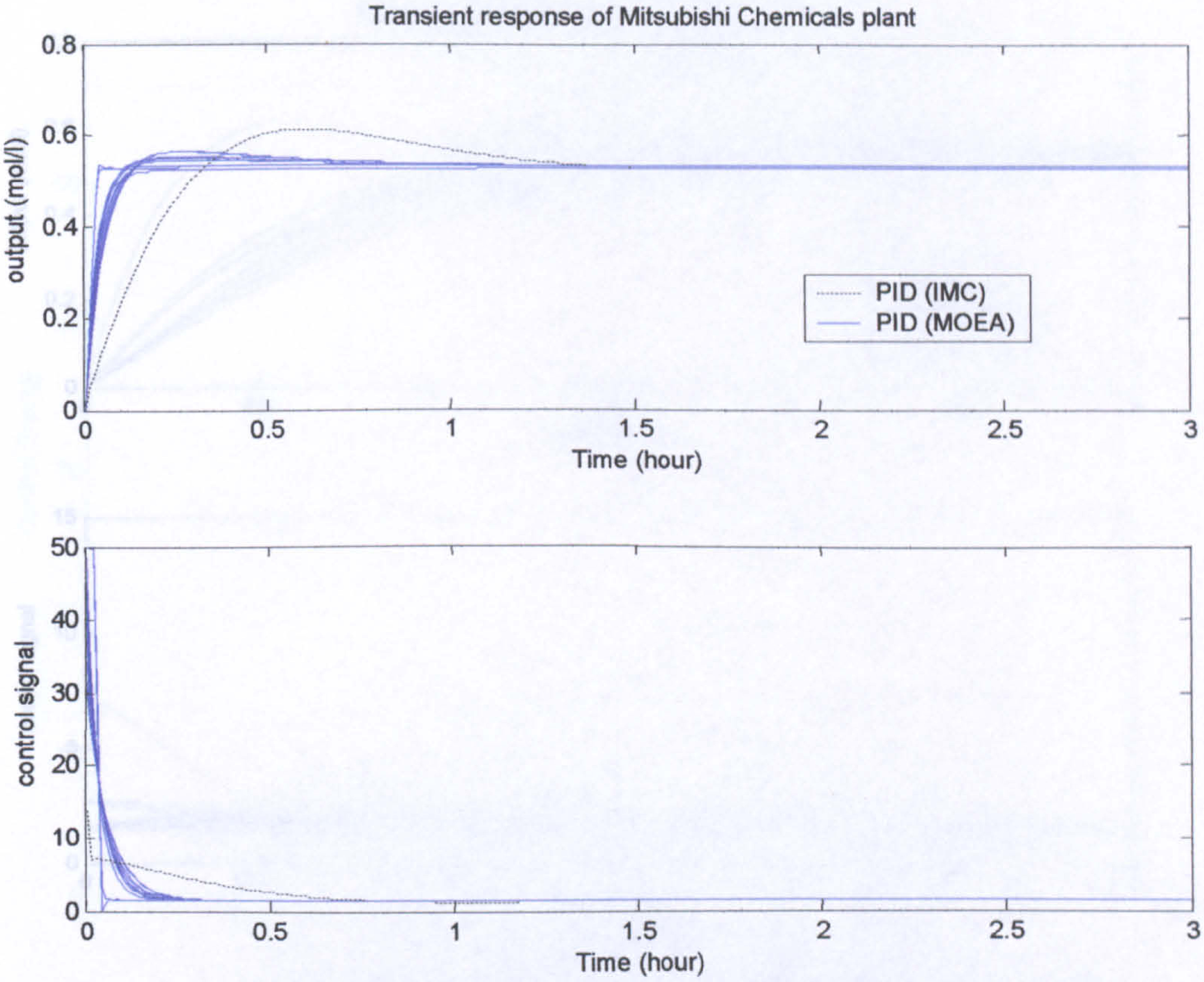

Figure 6.35 Step response in region where only ISE minimise.

Finally, in the region where control energy was minimised, the step responses (Figure 6.34) are too sluggish, some of the output did not reach the setpoint. Therefore, these solutions will not be considered for controller selection.

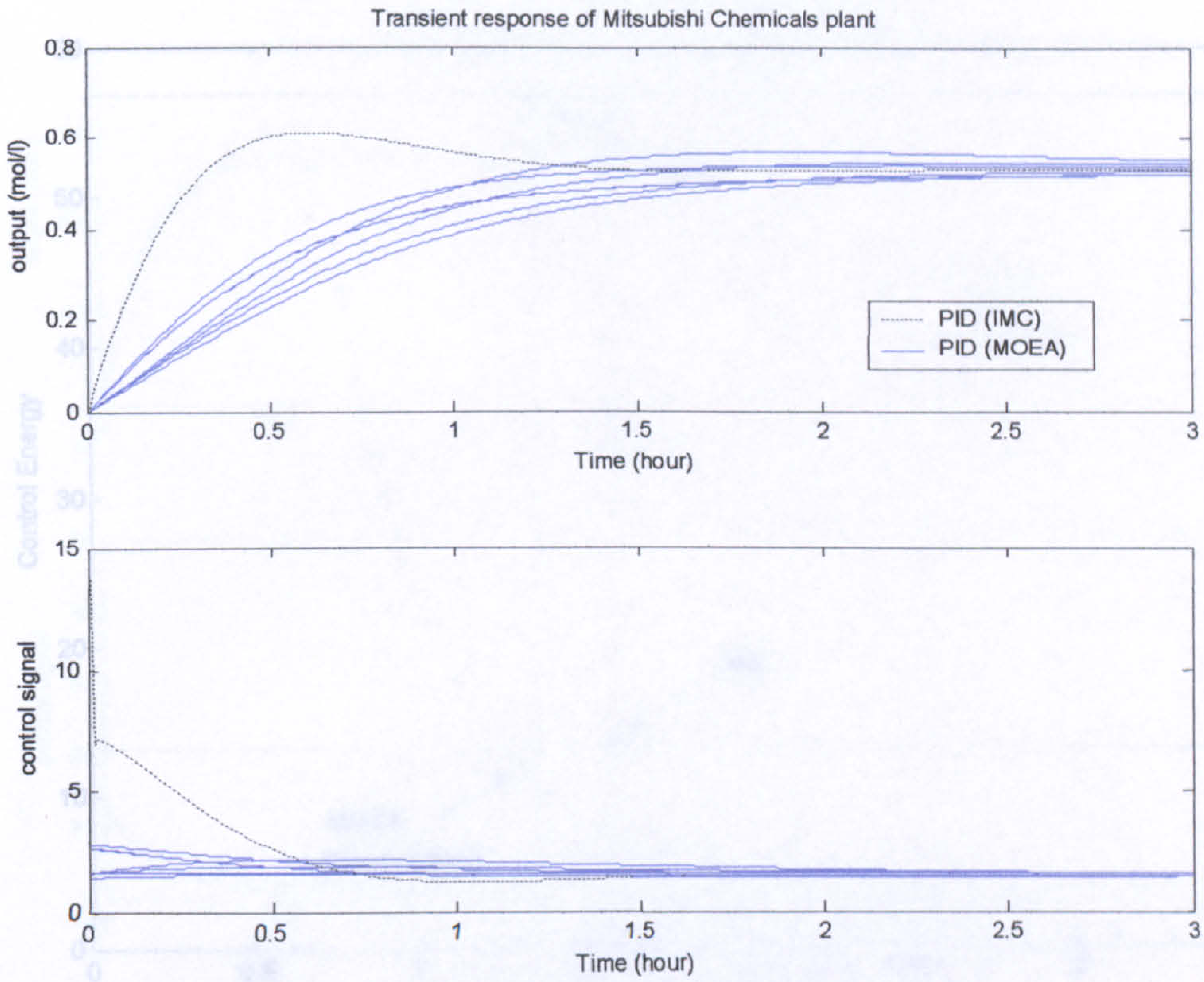

Figure 6.36 Step response in region where only Control Energy was reduce.

6.7.1 Multi-Objective Decision Making

From Figure 6.33, the selection of high-performance low-chattering weightless controller has been minimised to a few solutions. In some cases, when the good solutions are more than needed, the selection can be narrowed down to one or two solutions. This can be achieved by adding one or more reference points from different tuning methods. In Figure 6.37, Z-N designed PID was inserted as one of the additional reference point, minimising the good solutions from *non-dominated* sets to two.

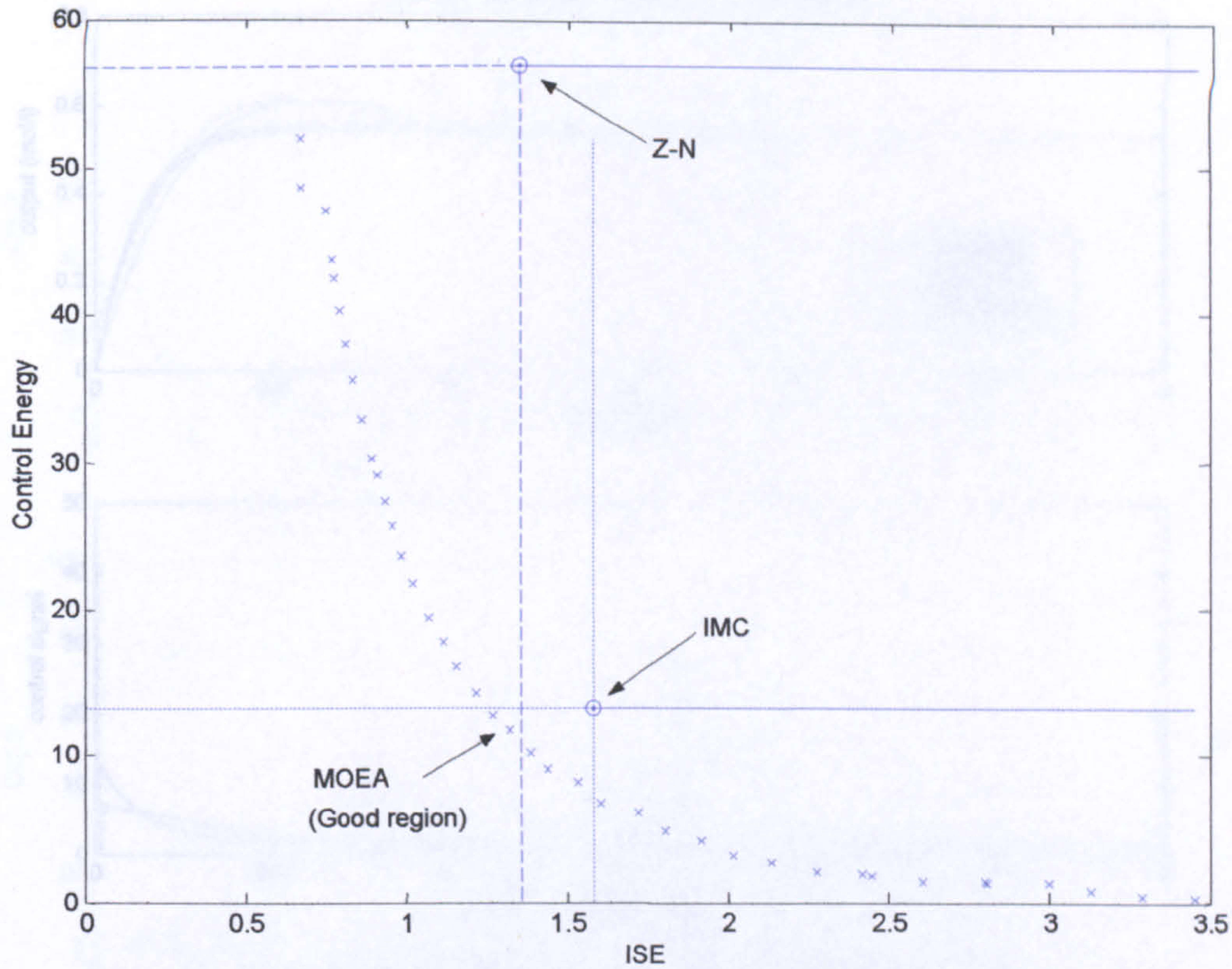

Figure 6.37 Additional Z-N tuned PID as reference point.

Using two reference points, Z-N and IMC, the optimum controllers were narrowed down to two. The step responses were plotted and shown in Figure 6.38 below.

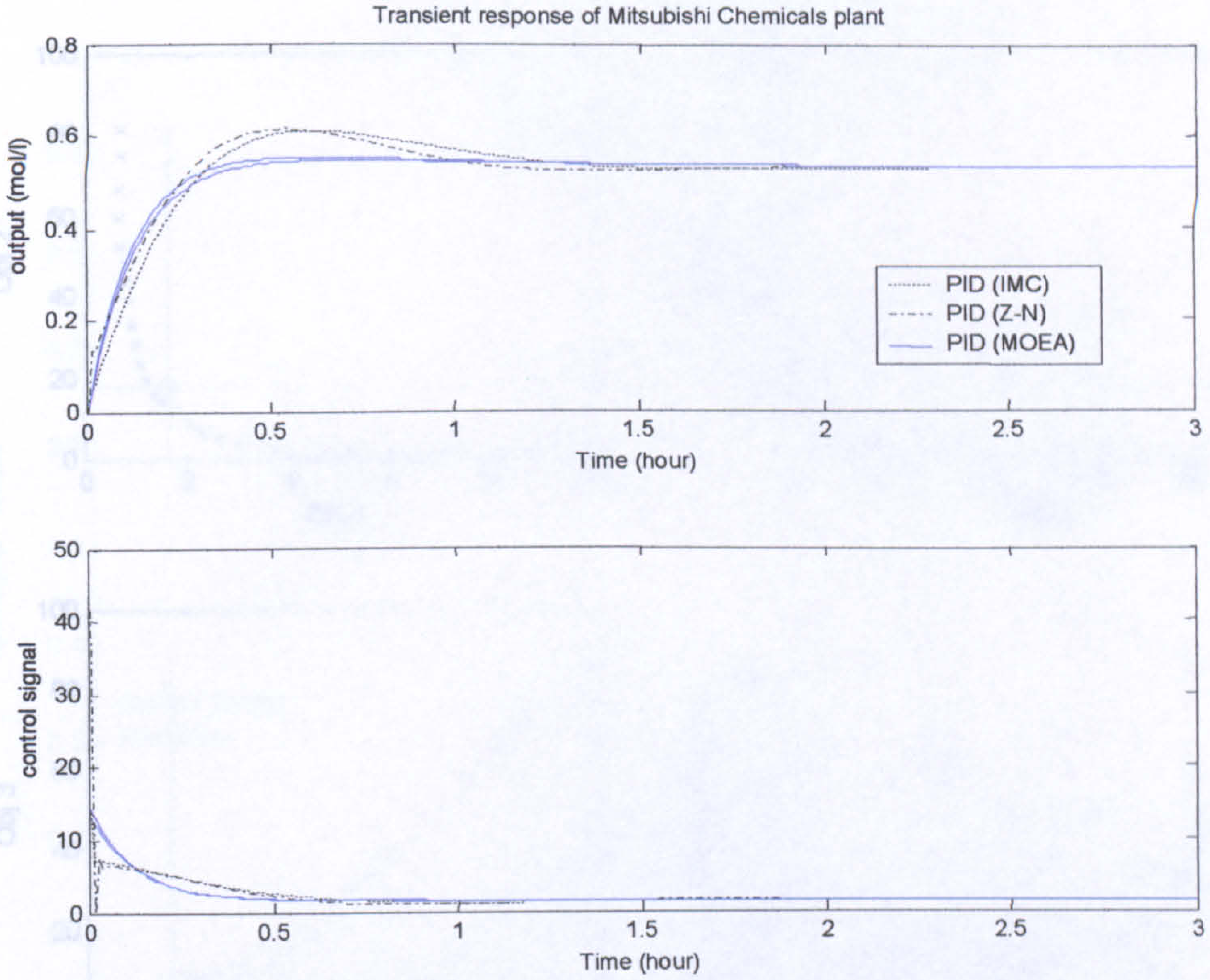

Figure 6.38 Step response of good region MOEA solution

6.7.2 Evolving High-Performance TSC Systems Using MOEA

The MOEA design methods are not restricted to PID controllers and the objective function can be more than two. Furthermore, the plant can be linear or nonlinear and constraints can be included in the design. In the following example, a TSC was applied to the nonlinear chemical plant used in the previous test. Three objectives used were ISE, control energy and load disturbance. Although TSC cannot compare with IMC in many ways, IMC can still be used as the guideline when choosing TSC from the *non-dominated* sets. The test results are shown below.

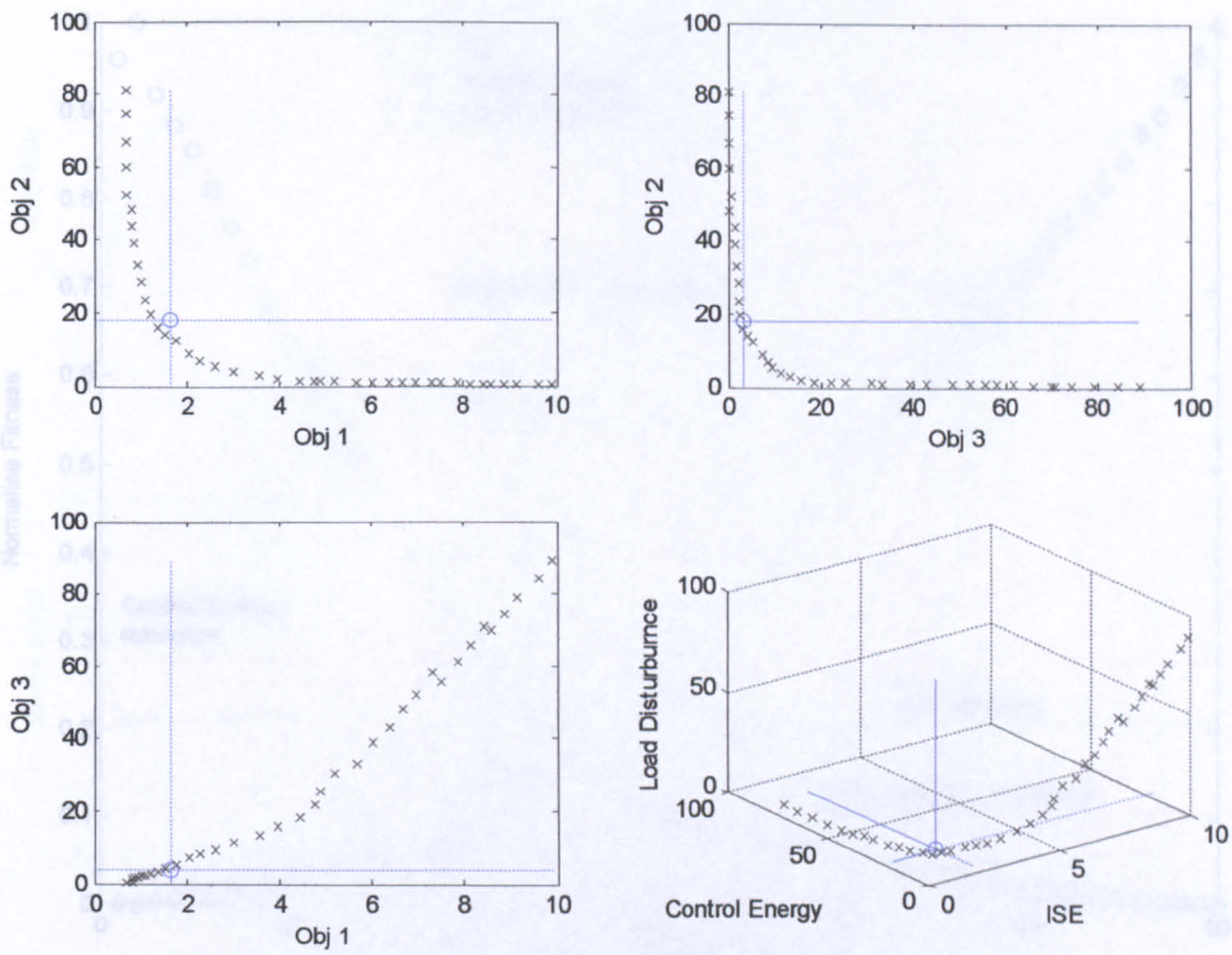

Figure 6.39 2D and 3D view of non-dominated solutions for 3 objectives MOEA and selection region reference to IMC.

In Figure 6.39, the selection region *can be classified* clearly from 2D and 3D plots. This method cannot be used on MOEA with 4 or more objectives. Therefore, another visualisation and selection method has to be adapted.

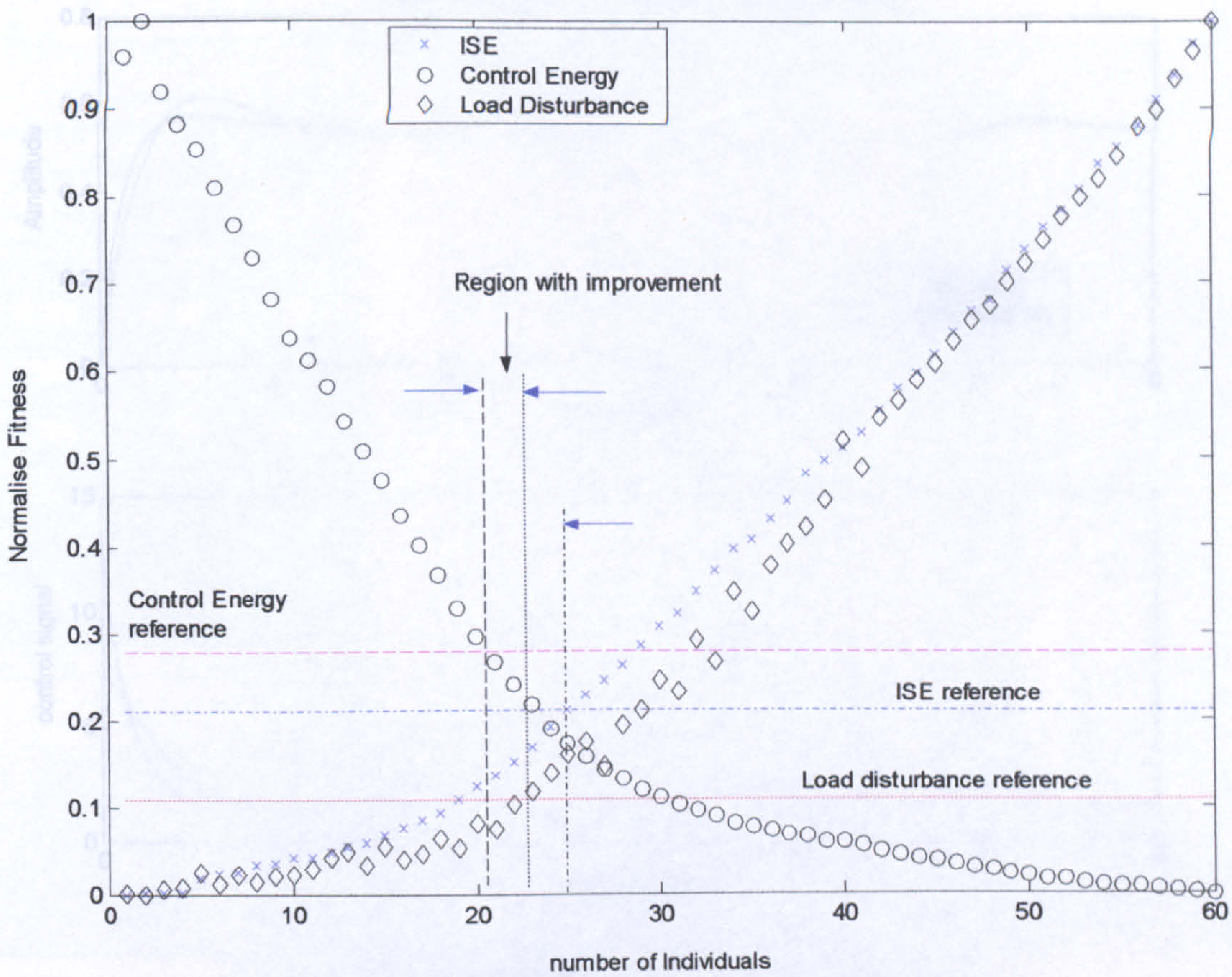

Figure 6.40 Normalised and sorted view of *non-dominated* sets from MOEA and selection region reference to IMC

The selection technique in Figure 6.40 can be used on four or more objectives. From the figure, the solutions that fall under 'region with improvement' are the chosen controllers. Their performances were plotted and shown in Figure 6.41. The plot shows that the selected PID from MOEA has a faster step response, better load disturbance and requires less control energy.

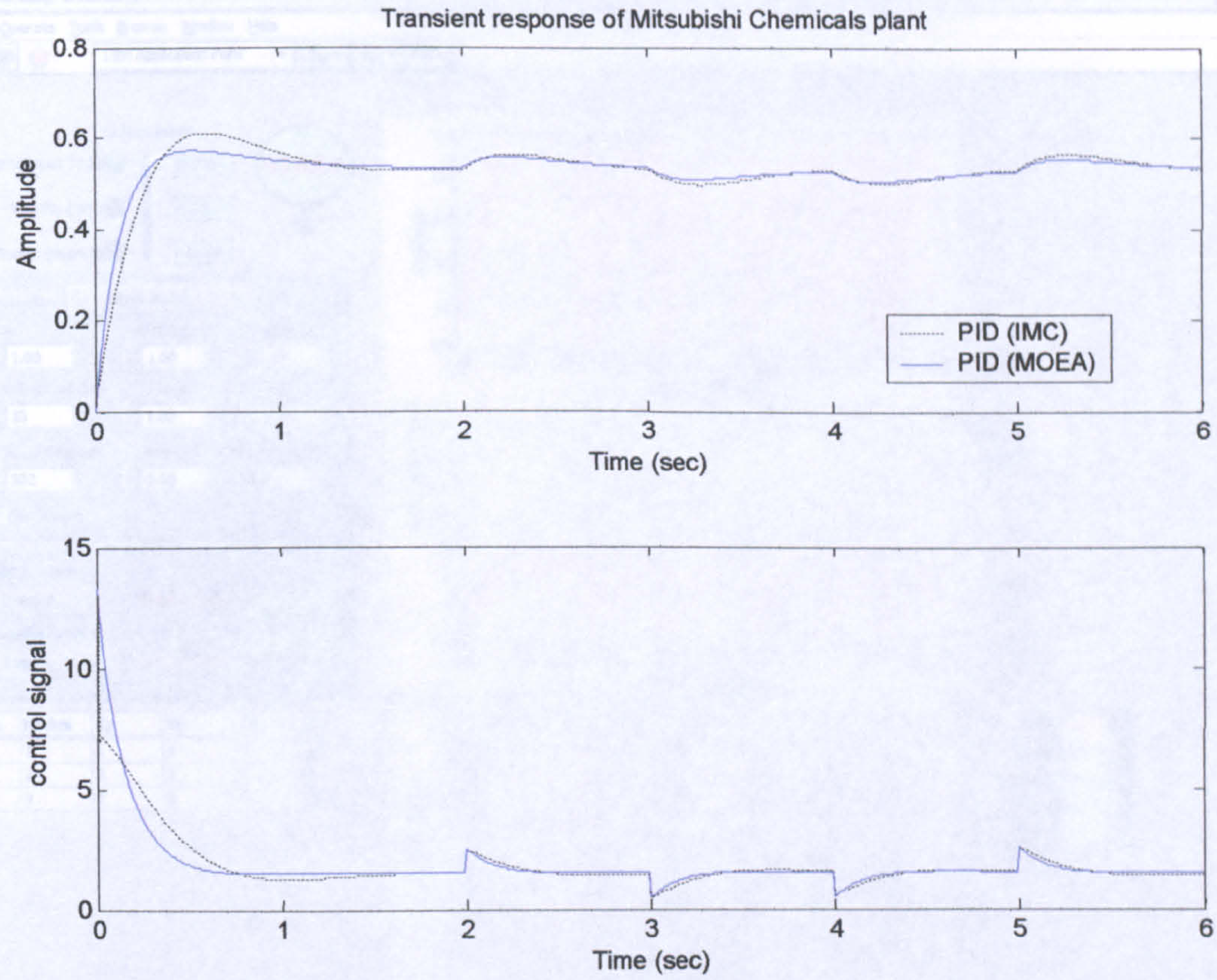

Figure 6.41 Step response from selected solution (with load disturbance).

6.8 LabVIEW Based TSC

A LabVIEW TSC/PID real time online controller has been developed to test the method proposed. The controller comes with these functions:

- TSC with infinite controllers
- Series or parallel, P, I, PI, PID, ideal PID, PI-D, I-P, I-PD, PID-LPF controllers
- Limits/anti wind-up
- Digital input enables
- Spline, linear, polynomial and rational interpolation/extrapolation scheduling
- Save/load enables

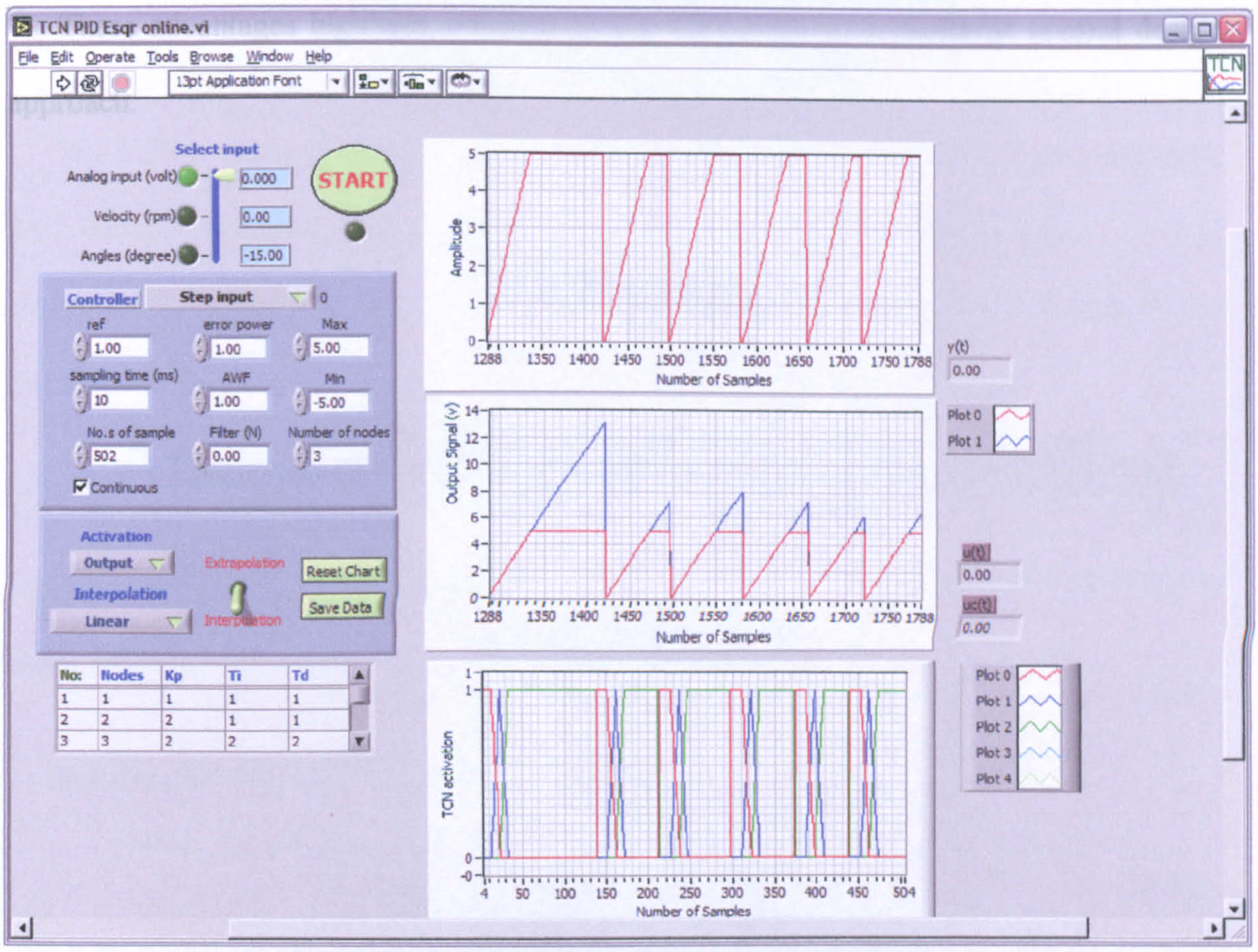

Figure 6.42 LabVIEW online TSC controller interface

6.9 Summary

The TSC offers several important advantages.

- The TSC develops a clear representation of the controllers since each TSC is linear. This enables analysis of the overall properties and makes the conversion of the TSC-TSM network into a linear system at any operating point more straightforward.
- A priori knowledge can be use for TSC design
- The learning of TSC is fast and straightforward, and various tuning rules and search tools can be applied.
- Capabilities to network controllers with various transfer functions.

These advantages highlight potential of the TSC-TSM as a nonlinear control design approach.

Chapter 7

Analysis of Trajectory-Scheduling Networks

7.1 Linear representation of the TSN Systems

The TSN is made of a linear network. The interpolated linear systems can be extracted from the TSN instantly at any given operating point. Adapting the equations (2.2 to 2.5) in section 2.6.2, at any given operating point (ψ), the parameters P_ψ that extracted from the TSN can be derived as:

$$P_\psi = \sum_{i=1}^{M} P_i(\psi)\rho_i(\phi)\,, \tag{7.1}$$

thus, the linear model G_ψ extracted from the TSM is

$$G_\psi = \sum_{i=1}^{M} G_i(\psi)\rho_i(\phi)\,, \tag{7.2}$$

and the controller C_ψ extracted from the TSC is

$$C_\psi = \sum_{i=1}^{M} C_i(\psi)\rho_i(\phi)\,. \tag{7.3}$$

Therefore, the extracted close loop system of the network at ψ is illustrated as Figure 7.1.

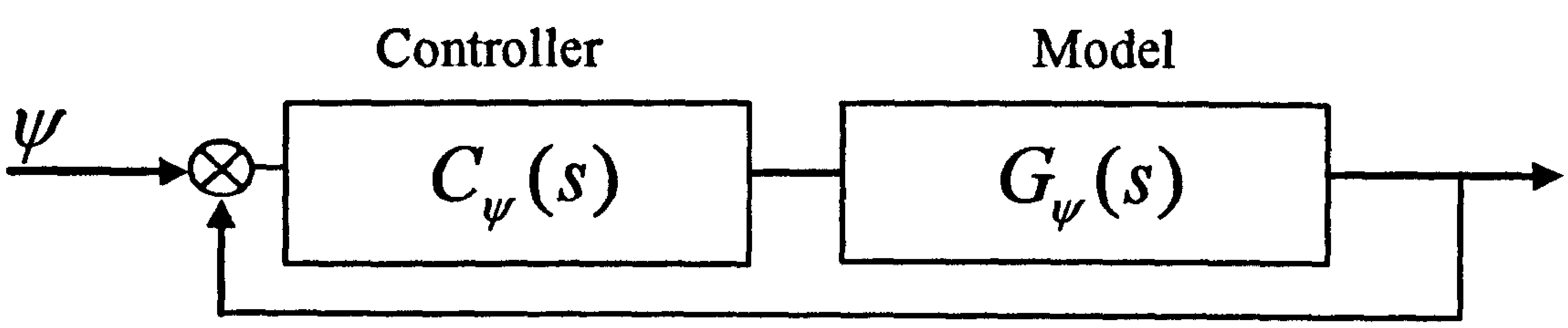

Figure 7.1 Static system during transition

Hence, the close loop transfer function can be derive as

$$CG_\psi(s) = \frac{C_\psi G_\psi}{1 + C_\psi G_\psi}\,. \tag{7.4}$$

Since the extracted system is linear, any classical LTI analysis methods can be applied, such as the Bode plots in an open loop and a root locus in a closed loop. By analysing (7.4) for the full operating trajectory of the TSC-TSM networks, any nonlinear systems represented by it can be studied. Hence, the stability of the nonlinear systems for the whole operating trajectory can be ensured.

7.1.1 Analysis TSM-TSC with Classical Methods

Using the twin-tank couple process as an example, the system was set to operate at the level from 0.04m to 0.14m. Using the data collected from these operating ranges, the TSM and TSC controllers are evolved. The obtained TSC-TSM systems are analysed using the method suggested above. The system was broken down into 0.01m steps and plotted in the figure below:

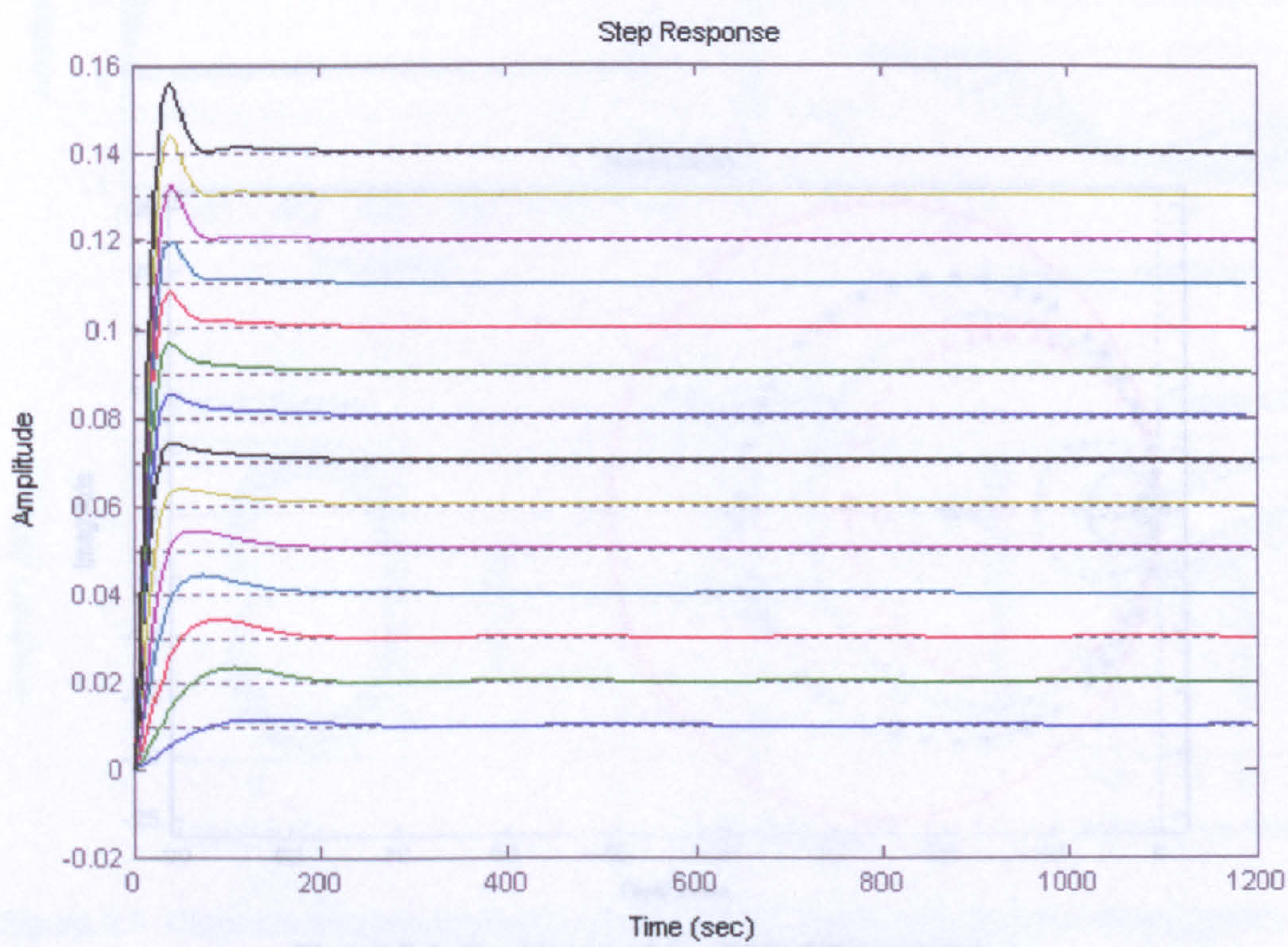

Figure 7.2 Step response of the TSM network

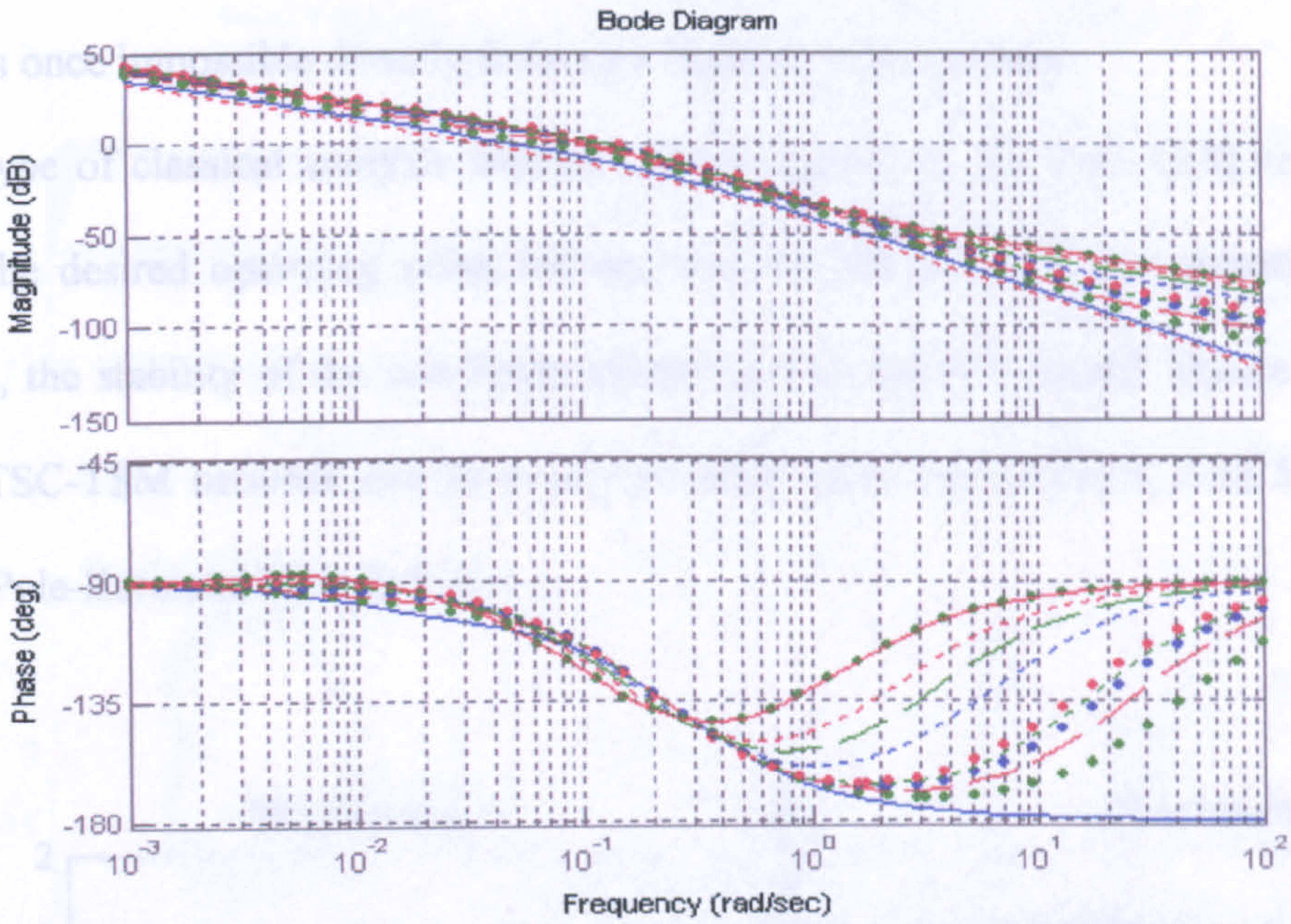

Figure 7.3 Bode plot of the TSM network

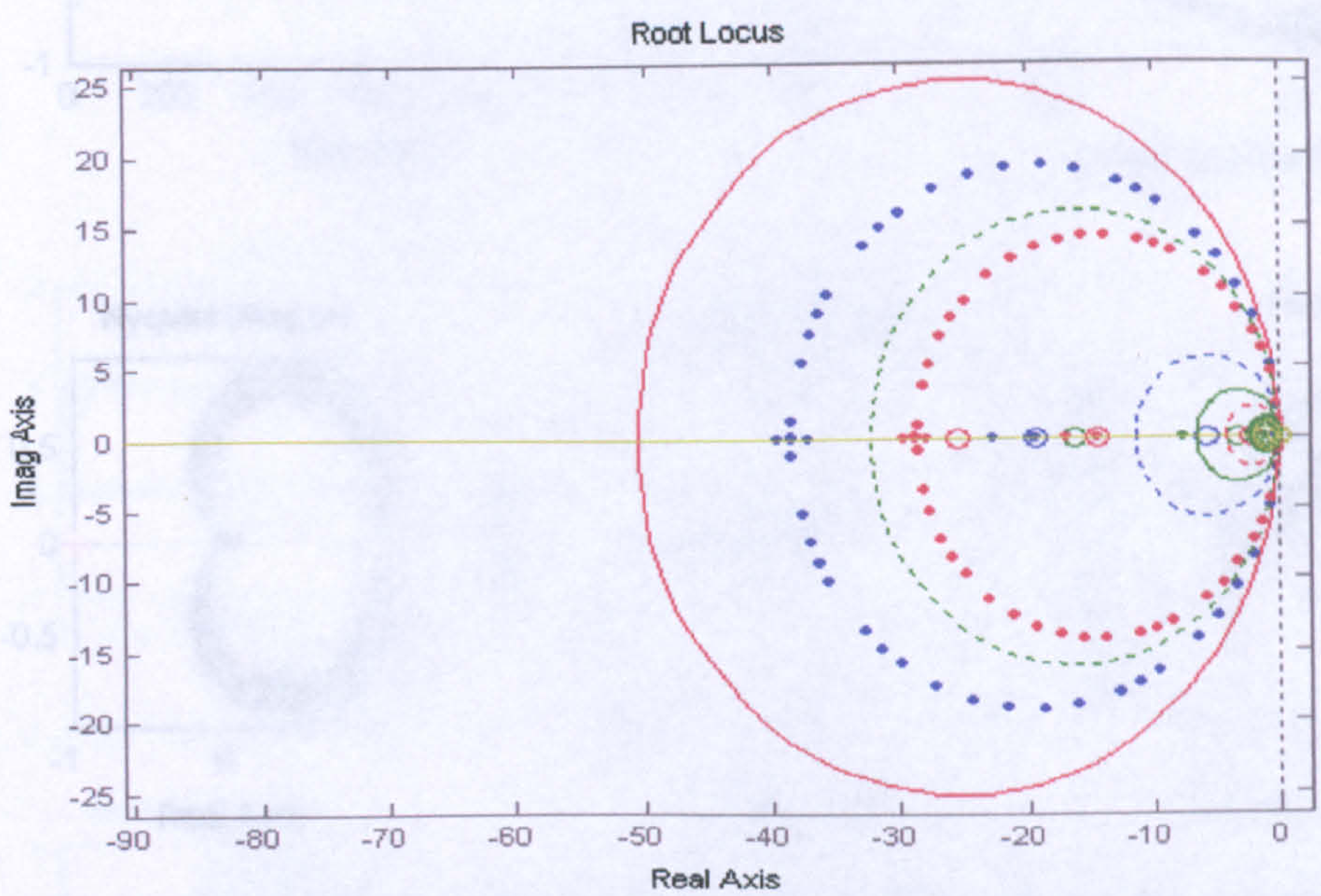

Figure 7.4 Root locus of the TSC-TSM network

Refer to the step, Bode and root locus plot. The results show that each of these linear paths, covering the complete operating region, was well within the stability margins, therefore giving a simple and direct way to design and analyse nonlinear control systems, which was once impossible directly from sets of plant response data.

Any type of classical analysis method can be applied to the TSC-TSM network, not only on the desired operating point but also on the full range of the operating region. Therefore, the stability of the non-linear system can be easily assessed. Figure 7.5 shows that the TSC-TSM network can be explored using linear interpolation with Step, Bode, Nyquist, Pole-Zero and Nichols Chart.

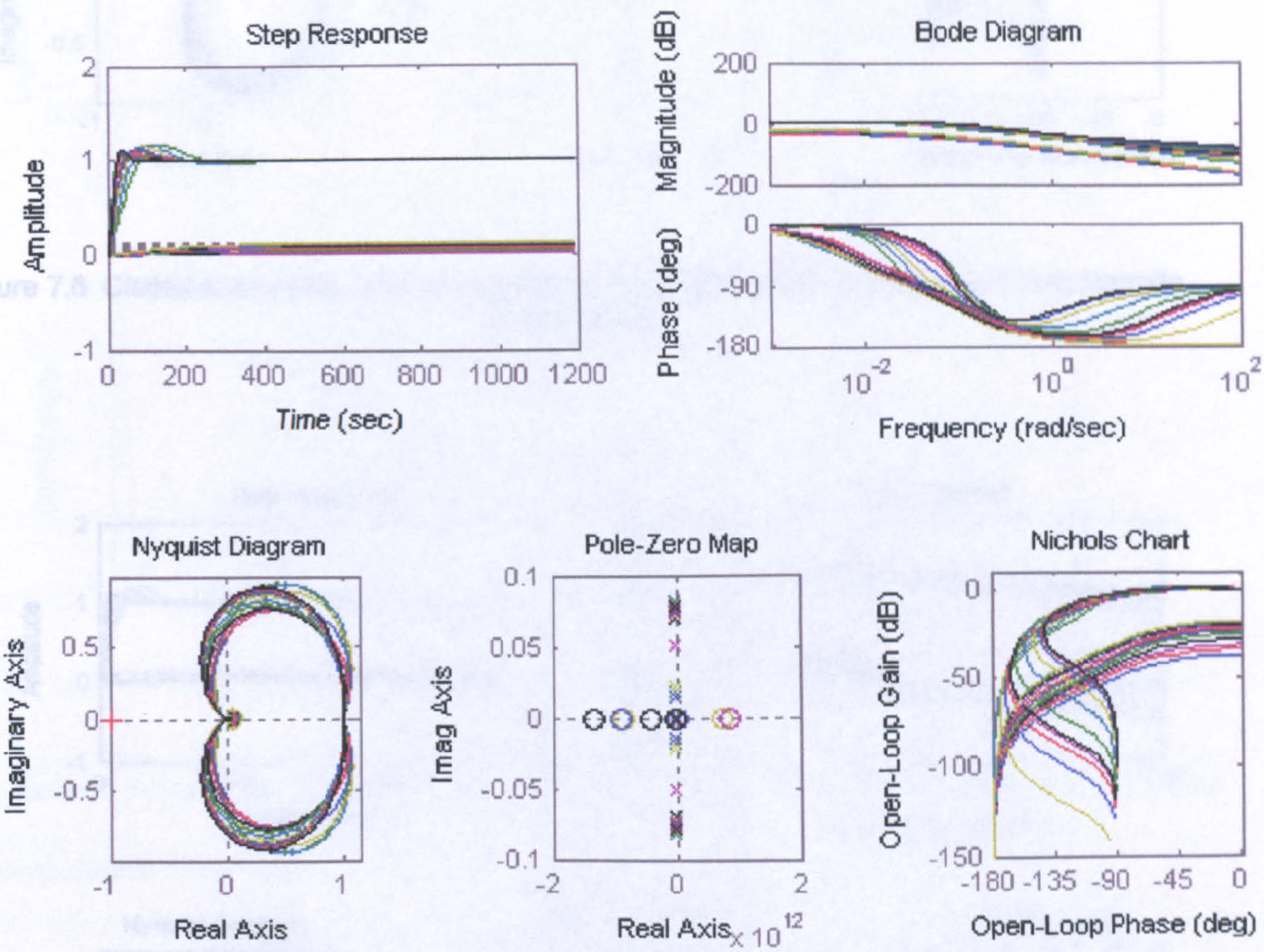

Figure 7.5 Classical analysis methods applied to TSC-TSM system (linear interpolation)

This analysis is not restricted to linear interpolation of the TSC-TSM network only, investigations were also made on two other types of interpolation, namely the Piecewise Cubic Hermite Interpolation (Figure 7.6) and the Spline Interpolation (Figure 7.7).

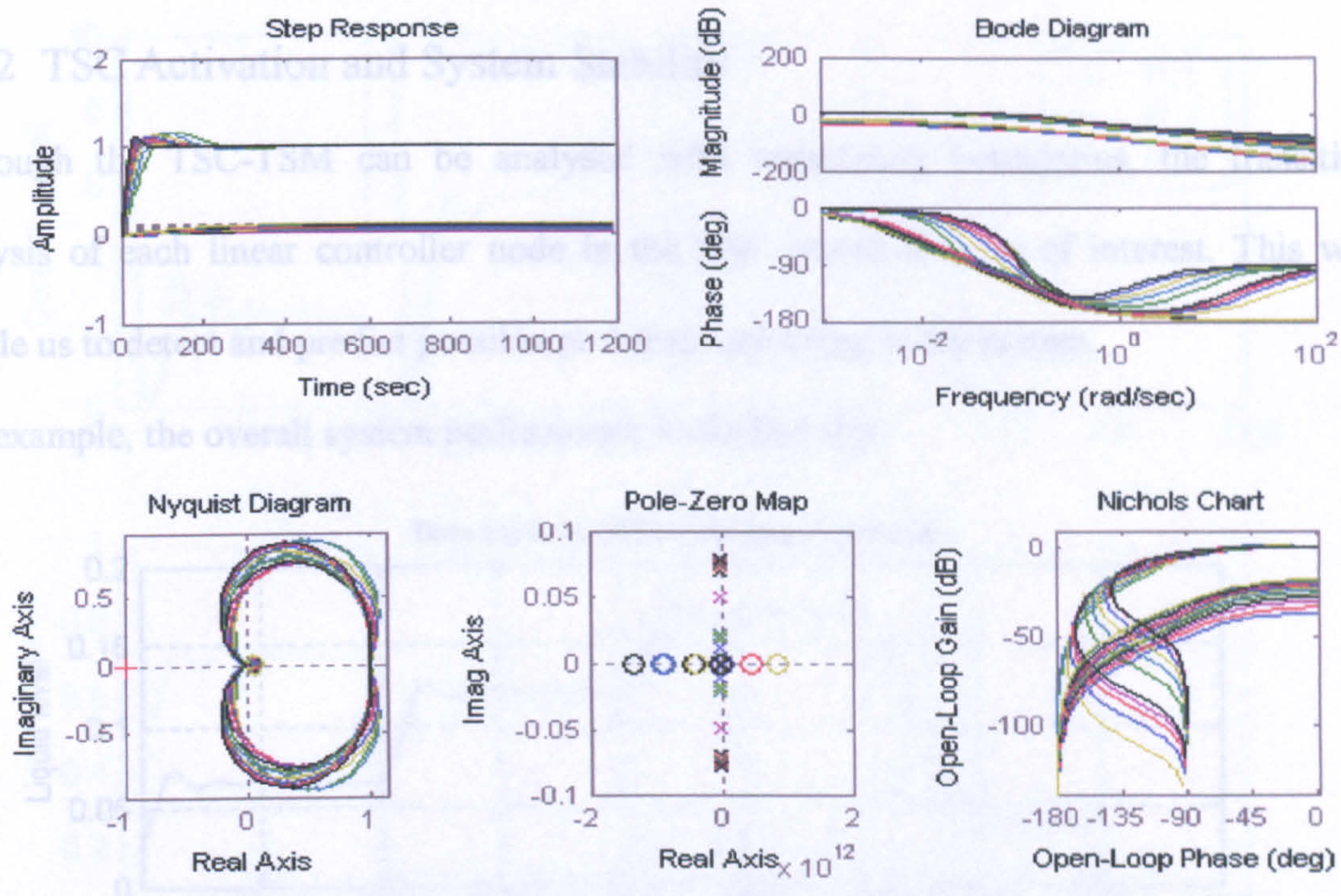

Figure 7.6 Classical analysis methods applied to TSC-TSM system (Piecewise Cubic Hermite interpolation)

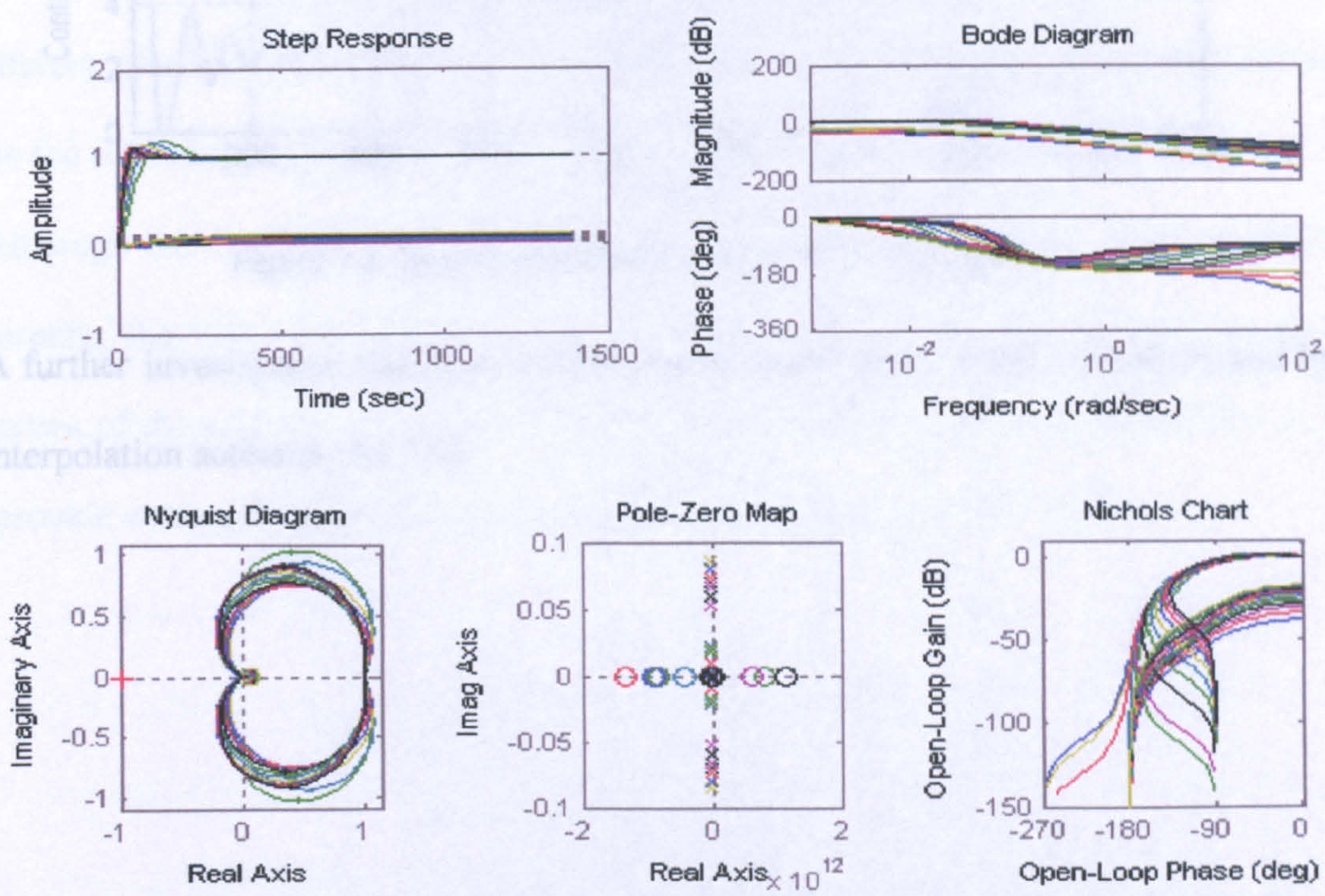

Figure 7.7 Classical analysis methods applied to TSC-TSM system (Spline interpolation)

7.1.2 TSC Activation and System Stability

Although the TSC-TSM can be analysed with uncertainty boundaries, the transition analysis of each linear controller node in the TSC would also be of interest. This will enable us to detect and predict possible problems occurring in the system.

For example, the overall system performance looks like this:

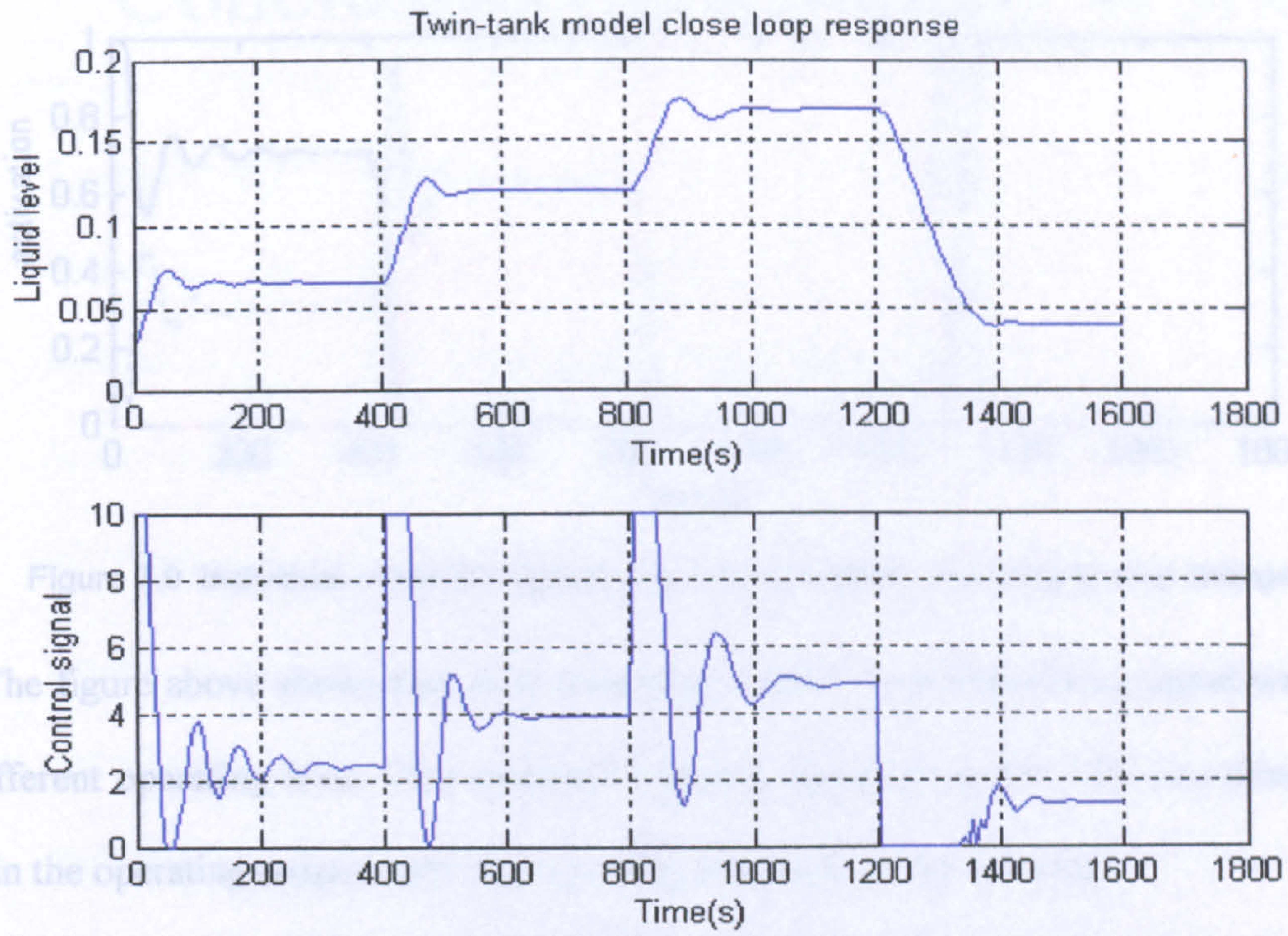

Figure 7.8 Overall performance of the TSC-TSM system

A further investigation has been carried out to assess each of the controllers and how the interpolation activates the TSC.

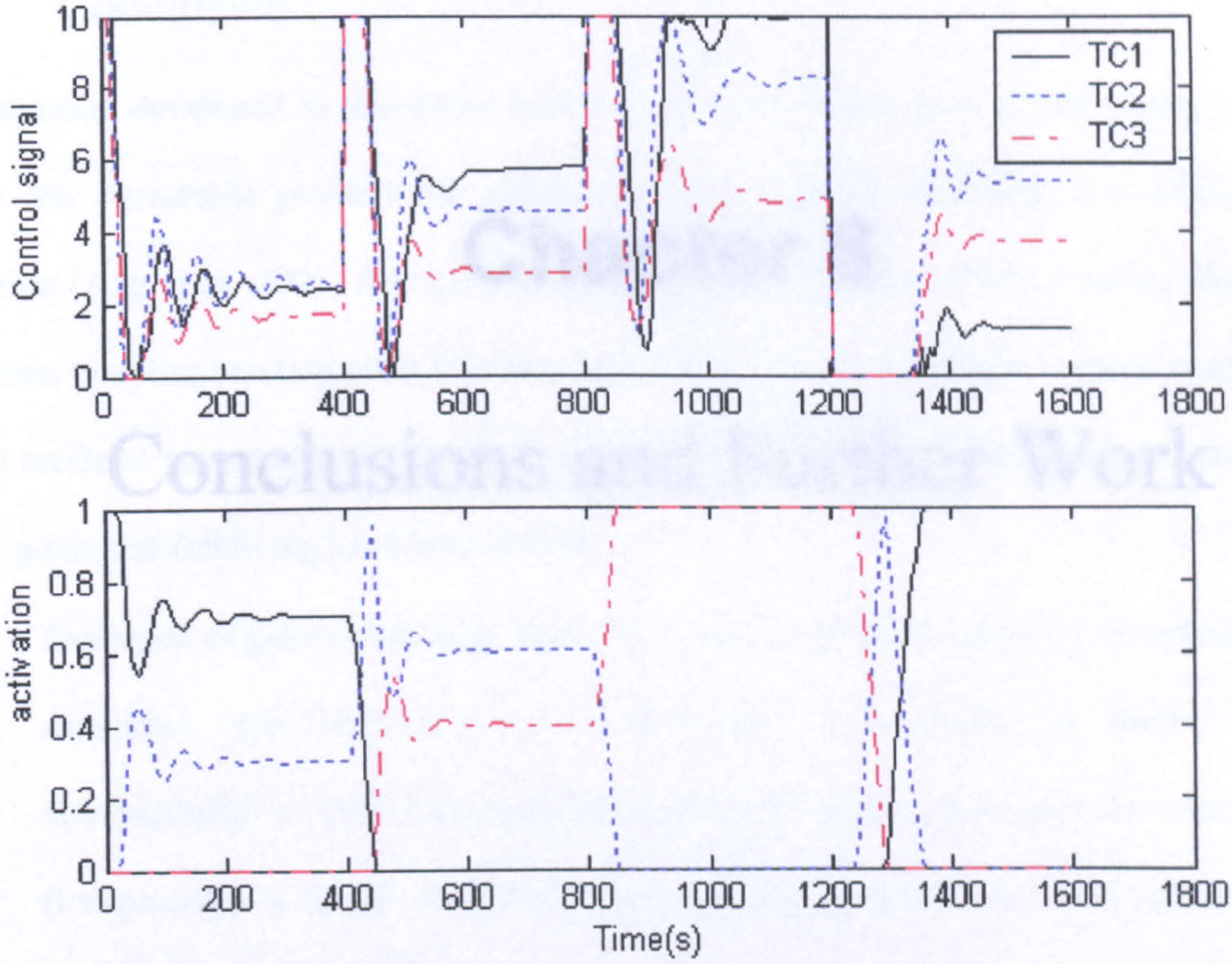

Figure 7.9 Individual controller signals and their activation according to step change.

The figure above shows that each controller exhibits a synchronising signal waveform at different operating level. This evidence supports the fact that the TSC co-ordinate well within the operating range under the crossover activation of the network.

Although one can argue that linear analysis techniques are generally inappropriate for interpreting the TSN system and the results of such an analysis do not reflect all the properties of the underlying parameter-variant system, it must be remembered that it can still provide extremely valuable insight in terms of local and global stability.

Chapter 8

Conclusions and Further Work

8.1 Conclusion

The solution developed in this thesis builds on from the existing work and endeavours to solve the intractable problem of addressing both setpoint following and disturbance rejection (Ang *et al.* 2005; Åström and Hagglund 1995; Levine 1996). Another objective has been to utilise existing plant I/O data at operating levels of interest, without needing to inject artificial perturbation to the plant at operating point under consideration. Achieving these goals, the following have been developed.

- Extension of gain-scheduling, LMN/LCN and LAM techniques to a complete TSM technique, and application of evolutionary computation to derive TSMs automatically. A TSM eliminates the need of linearising and prior derivation of a first-principle's model or the need for artificial perturbation to the plant under operation.

- Extension of performance metrics for both single and multiple objective optimisation and search algorithms. The current visualisation technique for assessing non-dominant solution sets is limited to two objectives. This has been extended to higher dimensional data, with a novel visualisation technique developed for identifying and presenting the best EAs solutions.

- Development of a flexible and straightforwardly applicable TSC, and also methods of obtaining TSC from existing control designs or CAD software, as well as through multi-objective evolutionary search techniques. These methods allow *a priori* knowledge to be incorporated in the design if desired. The results are presented with multi-objective visualisation, which enables the user to supervise (if desired) the final design for performance that meets his/her requirements.

- Development of a linear parameter-variant technique and tool to enable analysis of a combined TSM and TSC network using classical stability analysis methods.
- Development of a novel operating-point-scheduling technique for the entire operating envelope of a given process. This is coupled with a node acquiring technique based on the process nonlinearity, reducing the number of networking parameters and increasing application speed for TSM and TSC.
- Development of a MATLAB based GUI software tool automatically to build a TSM for a nonlinear process, to evolve the corresponding TSC, and to analyse the close-loop system. This also makes it easy to implement in real-time with LabVIEW and to test against real physical systems.

8.2 Future Perspective

With reference to the methodology and techniques developed, the development of multiple model networks has been relatively mature. However, a nonlinear network analysis is still lacking at present. Hence, this part of research should deserve more attention. The future work includes a theoretical development of a full suite of nonlinear network analysis.

A fully 'plug-and-play' automated nonlinear system modelling and design tool in the form of a professional software package would be valuable to industrial users. Some simple additional steps can be added to the current design to achieve automated 'plug-and-play'. Finally, one valuable enhancement would be to make the software-based system Ethernet enabled, where users can 'plug-and-play' remotely through the Internet.

References

Ang, K.H., Chong, G., and Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4), 559-576.

Åström, K.J. and Bohlin, B. (1965). Numerical identification of linear dynamic systems from normal operating records. *IFAC Symposium on Self-Adaptive Systems*, Teddington, England.

Åström, K.J. and Hagglund, T. (1995). PID Controllers: Theory, Design and Tuning. *Instrument Society of America.*

Åström, K.J. and Wittenmark, B. (1989). *Adaptive Control.*, Addison-Wesley, Reading, MA.

Chen, G., Ti, M., and Piovoso, M.J. (1998). A multivariate statistical controller for on-line quality improvement. In *International Proceeding on Control*, 8(2), 139-149.

Chong, G. and Li, Y. (2000a). Evolving Trajectory Controller Networks from Linear Approximation Model network. In *Proceeding of the 2000 Congress on Evolutionary Computation*, San Diego, USA, 1, 251-255.

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

Chong, G. and Li, Y. (2002a). Multi objective design of trajectory networks controller for nonlinear plants and stability analysis. In *Proceedings of the 5th Asia-Pacific Conference on Control and Measurement*, Dali, P.R. China, 185-190.

Chong, G. and Li, Y. (2002b). Nonlinear system modelling and control along the operating locus. In *Proceedings of the 8th annual Conference of Chinese Automation and Computing Society in the UK*, U.K., 59.

Chowdhury, M. and Li, Y. (1998). Learning fuzzy control by evolutionary and advantage reinforcements. *International Journal of Intelligent Systems*, 13(10-11), 949-974.

Dong, J. and Brosilow, C.B. (1997). Design of robust multivariable PID controllers via IMC. In *Proceedings of the American Control Conference*, 5, 3380-3384.

Fogel, D.B. (1995). *Evolutionary Computation.*, IEEE Press, Piscataway, NJ.

Fonseca, C.M. and Fleming, P.J. (1993). Genetic algorithms for multiobjective optimisation: Formulation, Discussion and Generalisation. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan-Kauffman, 416-423.

Franklin, G.F., Powell, J.D., and Emami-naeini, A. (1991). *Feedback control of dynamic systems.*, Addison-Wesley.

Gawthrop, P.J., Jezek, J., Jones, R.W., and Sroka, I. (1993). Grey-box model identification. *Control-Theory and Advanced Technology*, 9(1), 139-157.

Gawthrop, P.J. and Ronco, E. (1996). Local model networks and self-tuning predictive control. In *Proceedings of 4th IEEE Mediterranean Symposium on New Directions in Control and Automation*, 261-265.

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

Gollee, H. and Hunt, K.J. (1997). Nonlinear modelling and control of electrically stimulated muscle: a local model network approach. *International Journal of Control*, 68(6), 1259-1288.

Gollee, H. and Murray-Smith, D.J. (1997). Validation of an empirical model structure by analysing model properties. In *International Conference on Engineering Applications of Neural Networks*, Stockholm, Sweden.

Grace, A. (1992). *Optimisation Toolbox User's Guide.*, The MathWorks, Inc.

Herreros, A., Baeyéns, E., and Perán, J.R. (2000). Design of PID Controllers using Multiobjective Genetic Algorithms. *IFAC Digital Control: Past, Present and Furture of PID Control.*

Horn, J., Nafpliotis, N., and Goldberg, D.E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1, 82-87.

Hunt, K.J. and Johansen, T.A. (1997). Design and analysis of gain-scheduled control using local controller networks. *International Journal of Control*, 66(5), 619-651.

Jacobs, R.A. and Jordan, M.I. (1993). Learning piecewise control strategies in a modular neural network architecture. *IEEE Transaction on Systems, Man, and Cybernetics*, 23(2), 337-345.

Johansen, T.A. and Foss, B.A. (1992b). A NARMAX model representation for adaptive control based on local models. *Modelling, Identification and Control*, 13(1), 25-39.

Johansen, T.A. and Foss, B.A. (1993). Constructing NARMAX models using ARMAX models. *International Journal of Control*, 58(5), 1125-1153.

Johansen, T.A. and Foss, B.A. (1992a). A narmax model representation for adaptive control based on local model. *Modelling, Identification and Control*, 13(1), 25-39.

Johansen, T.A., Hunt, K.J., Gawthrop, P.J., and Fritz, H. (1998). Off-equilibrium linearisation and design of gain scheduled control with application to vehicle speed control. *Control Engineering Practice*, 6(2), 167-180.

Kailath, T. (1980). *Linear Systems.*, Prentice-Hall, New York.

Kajiwara, H., Apkarian, P., and Gahinet, P. (1999). LPV techniques for control of an inverted pendulum. *IEEE Control Systems Magazine*, 19(1), 44-54.

Kashiwagi, H. (1983). *Automatic Control.*, Asakura, Tokyo.

Leith, D.J., Murray-Smith, R., and Leithead, W.E. (2000). Nonlinear Structure Identification: A Gaussian Process Prior/Velocity-based approach. *Control*, Cambridge.

Levine, W.S. (1996). *Control Handbook.*, CRC Press/IEEE Press.

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*, 1(1), 83-95.

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 *The Third Asia-Pacific Conference on Measurement and Control*, Dunhuang, China, Plenary paper, 29-33.

Li, Y. and Haeussler, A. (1996). Artificial evolution of neural networks and its application to feedback control. *International Journal of Artificial Intelligence in Engineering*, 10(2), 143-152.

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

Li, Y., Ng, K.C., Tan, K.C., Gray, G.J., McGookin, E.W., Murray-Smith, D.J., and Sharman, K.C. (1995a). Automation of linear and nonlinear control systems design by evolutionary computation. In *Proceeding of IFAC Youth Automation Conference*, Beijing, China, 53-58.

Li, Y. and Tan, K.C. (2000). Linear approximation model network and its formation via evolutionary computation. *Academy Proceedings in Engineering Sciences (SADHANA)*, Indian Academy of Sciences, 25(112), 97-110.

Li, Y., Tan, K.C., Ng, K.C., and Murray-Smith, D.J. (1995b). Performance based linear control system design by genetic evolution with simulated annealing. In *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA, 731-736.

Middleton, R.H., Goodwin, G.C., and Mayne, D.Q. (1988). Design issues in adaptive control. *IEEE Transaction on Automatic Control*, 33(1), 50-58.

Middleton, R.H. and Graham, C.G. (1990). *Digital control and estimation: a unified approach.*, Prentice-Hall, Inc.

Morse, A.S. (1990). Toward a unified theory of parameter adaptive control-tunability 1002-1012. *IEEE Transaction on Automatic Control*, 35(9).

Morse, A.S., Mayne, D.Q., and Goodwin, G.C. (1992). Applications of hysteresis switching in parameter adaptive control. *IEEE Transaction on Automatic Control*, 37(9), 1343-1354.

Murray-Smith, R. and Johansen, T.A. (1997). *Multiple Model Approaches to Modelling and Control.*, Taylor and Francis.

Murray-Smith, R., Neumerkel, D., and Sbarbaro-Hofer, D. (1992). Neural Networks for Modelling and Control of a Non-linear Dynamic System. *IEEE Symposium on Intelligent Control*, Glasgow, 404-409.

Narendra, K.S. and Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2), 171-187.

Narendra, K.S., Balakrishnan, J., and Ciliz, M.K. (1995). Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems Magazine*, 15(3), 37-51.

Narendra, K.S. and George, K. (2002). Adaptive control of simple nonlinear systems using multiple models. In *Proceedings of the American Control Conference*, 3, 1779-1784.

Ng, K.C. (1995). *Switching Control Systems and Their Design Automation via Genetic Algorithms*. PhD Thesis, Department of Electronics and Electrical Engineering, University of Glasgow.

Platt, J. (1991). A resource-allocating network for function interpolation. *Neural Computation*, 3, 213-225.

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. In *Proceedings of the IEEE*, 78(9), 1481-1497.

Ronco, E. (1997). *Incremental polynomial controller networks: Two Self-organising Non-linear Controller*. PhD Thesis, Department of Mechanical Engineering, University of Glasgow, UK, 132-177.

Shamma, J.S. and Athans, M. (1992). Gain scheduling: Potential hazards and possible remedies. *IEEE Control Systems Magazine*, 12(3), 101-107.

Shamma, J.S. and Athans, M. (1990). Analysis of gain scheduled control for non-linear plants. *IEEE Transactions on Automatic Control*, 35, 898-907.

Sharman, S.K., McLoone, S., and Irwin, G.W. (2002). Genetic algorithms for local model and local controller network design. In *Proceedings of the American Control Conference*, 2, 1693-1698.

Shorten, R. (1996). *A Study of Hybrid Dynamical Systems with Application to Automobile Control*. PhD Thesis, Department of Electrical and Electronic Engineering. University College Dublin, Ireland.

Shorten, R.N. and Narendra, K.S. (1998). Investigating the stability of a class of hybrid system. *Computing & Control Engineering Journal*, 9(2), 81-88.

Skeirik, R.D. (1990). Expert systems for closed-loop process control. In *Proceeding of Instrurmentation in the Chemical and Petroleum Industries*, 21, 27-37.

Stilwell, D.J. and Rugh, W.J. (1998). Interpolation methods for gain scheduling. In *Proceedings of the 37th IEEE Conference on Decision and Control*, 3, 3003-3008.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transaction on Systems, Man, and Cybernetics*, 15(1), 116-132.

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

Tan, K.C. and Li, Y. (1997). Multi-objective genetic algorithm based time and frequency domain design unification of control systems. *IFAC International Symposium on Artificial Intelligence in Real-Time Control*, Kuala Lumpur, Malaysia.

Tan, K.C., Li, Y., Gawthrop, P.J., and Glidle, A. (1997). Evolutionary Grey-box Modelling for Practical Systems. *Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, UK.

Tan, W., Marquez, H.J., and Chen, T. (2004). Operating point selection in multimodel controller design. In *Proceedings of the American Control Conference 2004*, 4, 3652-3657.

Wada, T. and Osuka, K. (1997). Gain scheduled control of nonlinear systems based on the linear-model-sets identification method. In *Proceedings of the 36th IEEE Conference on Decision and Control*, 1, 412-417.

Wang, J., Wang, F.L., Li, M., and Yang, Y. (1997). A Knowledge-Based Controller Used in Process Control Systems. In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5(1), 47-57.

Weller, S.R. and Goodwin, G.C. (1994). Hyteresis switching adaptive control of linear multivariable system. *IEEE Transaction on Automatic Control*, 39(7), 1360-1375.

Woosoon Yim and Singh, S.N. (1995). Trajectory control of flexible manipulator using macro-micro manipulator system. In *Proceedings of the 34th IEEE Conference on Decision and Control*, 3, 2841-2846.

Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. *Transactions of ASME*, 64, 759-768.