



AN EVALUATION OF MODEL-BASED PREDICTIVE CONTROL

Student: Daniel Czarkowski, B.Eng. M.Sc.

Supervisor: Tom O'Mahony, Ph.D.

Department of Electronic Engineering,
Cork Institute of Technology

Submitted to the Higher Education and Training Awards Council (HEATEC)
in partial fulfilment of the requirements for
Degree of Master Engineering at CIT

Declaration

I hereby declare that this submission is my own work and thesis contains no material that has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

Signature of author

Certified by

Dr. Tom O'Mahony

Date

Abstract

This thesis attempts to determine whether advanced control algorithms offer any real benefits to the process industry in terms of optimising single-input single output process units. To establish this, an ideal PID controller, Two Degree of Freedom (2 DOF) PID Controller and the *Generalised Predictive Controller* (GPC) are compared. A common design philosophy is applied based on optimising performance subject to constraints on robustness. Specifically, three different designs are examined: minimum IAE subject to constraints on the gain and phase margin, minimum IAE subject to constraints on the modulus margin and minimum IAE subject to constraints on the input sensitivity function. Each of these optimisation problems are solved using a genetic algorithm.

The control algorithms and design methodologies are evaluated in simulation using a range of thirteen benchmark systems common in the process industry. Subsequently, the controllers and design strategies are evaluated on a real-time laboratory scale process — a flexible link. The results indicate that the GPC algorithm performs notably better than the other two controllers on the benchmark simulation study. However, when the GPC was applied to the flexible link the benefits are not so obvious. The 2-DOF PID controller achieves a really good trade-off between performance, robustness and ease-of-tuning. These issues are undoubtedly one of the reasons for the success of the structure in practice and the name *Industrial PID controller* is well deserved.

While each of the three proposed controller design techniques worked well, the design based on the input sensitivity function performed best. This is especially noticeable on the real-time system as the nature of the technique ensures that the high-frequency gain of the controller is adequately shaped and resulting in good immunity to high-frequency noise.

The novelty in this work primarily arises from the problem domain studied and the proposed controller design methodologies. This is particularly true for the designs that minimised the IAE subject to constraints on the modulus margin and input sensitivity function. Additional novelty arises from the nature of the penalty functions to avoid constraint violation and the modifications that were made to the canonical genetic algorithm to reduce the computational time.

Acknowledgements

First of all I would like to thank my supervisor, Dr Tom O’Mahony, for his insights into *the way of thinking*, his long hours spent on chatting and listening to my sometimes scattered thoughts and helping me channel them in the right direction. During the study he took an active interest in my work and encouraged me to keep at it when I could not see the light at the end of the tunnel. For him, I wish to dedicate a poem written by Anthony de Mello, *One Minute Wisdom*.

“*What does your Master teach?*”

asked a visitor.

“*Nothing,*” *said the disciple.*

“*Then why does he give discourses?*”

“*He only points the way — he teaches
nothing.*”

Veronica, like Tom, you were helping me in the structuring the research. I appreciate your support and waiting for me for the last two years. Your love is the rock on which we are going to build our lives.

I wish to thank professor Donald E. Knuth for the \TeX typesetting language and Leslie Lamport for developing the \LaTeX . In this field it is also worth to mentioning Arno Linnemann, who wrote the MATLAB `laprint` function which makes easier to export figures. I would also like to take this opportunity to thank researches from the University of Sheffield for developing the *Genetic Algorithm Toolbox* under the GNU General Public License.

I would like to thank to all the members of the CIT Mountaineering Club, Photographic Society and many Irish buddies for making my stay in Ireland unforgettable. Finally, I wish to gratefully acknowledge the financial support I have received from the Irish Government under the HEAPRTLI cycle II scheme and also a part was financed by the European Regional Development Fund, ERDF.

Daniel

Contents

1	Introduction	11
1.1	Review of existing controller comparisons	12
1.2	Problem statement	14
1.3	Organisation of the thesis	15
2	Controllers	17
2.1	PID controller	17
2.2	Two degree of freedom PID controller	18
2.3	Model-Based Predictive Controller	19
2.4	Control law of the GPC	20
2.5	GPC tuning parameters	27
2.5.1	The sampling period, h	28
2.5.2	The minimum output horizon, N_1	30
2.5.3	The maximum output horizon, N_2	31
2.5.4	The control horizon, N_u	31
2.5.5	The control weighting coefficient, λ	32
2.5.6	Effect of the T polynomial	32
2.6	Gradient techniques versus stochastic methods	34
3	Objective functions	36
3.1	Genetic Algorithms	36
3.1.1	Generate initial population	37

3.1.2	Ranking	38
3.1.3	Selection	38
3.1.4	Crossover	38
3.1.5	Mutation	39
3.1.6	Evaluation of the fitness function	39
3.1.7	Terminating the GA	40
3.1.8	Cancellation of the output horizon	40
3.2	Time domain performance criteria	41
3.3	Metrics to measure robustness	43
3.3.1	Gain and phase margin specification	43
3.3.2	Modulus margin, MM	44
3.3.3	Resonance peak of the closed-loop system M_p	44
3.3.4	Delay margin, DM	45
3.4	An overview of robust tuning techniques	45
3.4.1	Gain and phase margin specifications	46
3.4.2	Techniques based on sensitivity functions	50
3.4.3	Resume of tuning techniques	55
3.5	Objective functions	56
3.5.1	Gain and phase margin specifications	56
3.5.2	Modulus margin	58
3.5.3	Input sensitivity function	59
4	Simulations	61
4.1	Models	61
4.2	Choosing the sampling period	64
4.3	Optimisation environment	74
4.4	Results	79
4.4.1	The objective function based on GPM method	79
4.4.2	The objective function with constraints on MM	83
4.4.3	Summary of the results	86

5	Real-time application	87
5.1	Flexible link	87
5.1.1	Identification of the flexible link	89
5.1.2	Tuning strategy	90
5.1.3	Results	91
5.1.4	Discussion	98
6	Conclusions and future research	100
6.1	Conclusions	100
6.2	Summary of contributions	103
6.3	Future research directions	104
	Bibliography	119
	Appendix A	120
A.1	Objective function incorporating GPM	120
A.1.1	The ideal PID controller	120
A.1.2	The 2-DOF PID controller	123
A.1.3	The GPC controller	125
A.2	Objective function incorporating the sensitivity functions	127
A.2.1	The ideal PID controller	127
A.2.2	The 2-DOF PID controller	128
A.2.3	The GPC controller	129
	Appendix B, tuning of the flexible link	130
B.1	Tuning based on GPM specification	130
B.2	Tuning based on GPM specification	131
B.3	Tuning based on Modulus Margin specification	131
B.4	Tuning based on input sensitivity function	132
	Appendix C, figures discussed in section 2.6	134

List of Figures

2.1	Feedback loop	18
2.2	Block diagram of 2-DOF PID controller	19
2.3	Predictive control law	21
2.4	Block diagram for equivalent linear form of GPC	27
2.5	Poles (\times) and zeros (\circ) when the system in equation 2.41 is sampled with changing h	28
2.6	Presentation of bandwidth	30
2.7	Effect of the T polynomial on the IAE_{reg} and the robustness criteria	33
3.1	Flow chart of the Genetic Algorithm	37
3.2	Example of single point crossover	39
3.3	Example of mutation	39
3.4	Error of the best individuals	40
3.5	Typical response, $y(t)$, to the step input $r(t)$	41
3.6	Typical response of control error to step or impulse disturbance	43
3.7	Gauss curve	56
3.8	Penalty function for the GPM method	57
3.9	Penalty function for the modulus margin specification	59
3.10	Penalty function for the input sensitivity function	60
4.1	Sampling period versus performance	73
4.2	Flow diagram of optimisation environment	75

4.3	Effect of the look up table	78
4.4	Nyquist diagram of open-loop transfer function with controllers tuned for $G_{13}(s)$	80
4.5	Set-point following for $G_1(s)$, $A_m \geq 6dB$	83
4.6	Set-point following for $G_3(s)$, $A_m \geq 6dB$	84
4.7	Nyquist plot of $G_{12}(z)$	86
5.1	Diagram of the flexible link: (i) side view (ii) top view	88
5.2	Block diagram of the PC connected with the flexible link	88
5.3	Set point signal applied to the flexible link	91
5.4	IAE of the control error	92
5.5	IAE of the control signal	92
5.6	Bode magnitude plots of (a) input sensitivity function \mathcal{U} and (b) controller transfer function resulting from the GA optimisation based on $A_m = 6dB$ and $\phi_m = 45^\circ$	94
5.7	Complementary sensitivity function resulting from the GA optimisation based on $A_m = 6dB$ and $\phi_m = 45^\circ$	95
5.8	Set-point following of the flexible link	96
5.9	IAE of the average plant uncertainty	96
5.10	Set-point following of the full-loaded flexible link	97
C.1	Study of the search area for the system G_1	135
C.2	Study of the search area for the system G_5	136
C.3	Study of the search area for the system G_5	137
C.4	Figure C.5(a) revisited	138
C.5	Study of the search area for the system G_{10}	139
C.6	Study of the search area for the system G_{10}	140

Nomenclature

\mathbf{G}	Internal matrix in the GPC law
\mathbf{H}	Internal matrix in the GPC law
Δ	Differencing operator $1 - z^{-1}$
λ	Control weighting sequence
ω_B	Bandwidth in the frequency domain
ω_c	Frequency where the Nyquist plot intersects the unit circle
ω_n	Natural frequency
ω_s	Sampling frequency
ϕ_m	Phase margin
$\underline{\phi_m}$	Minimum phase margin
$\underline{A_m}$	Minimum gain margin
ζ	Damping factor
A	Plant denominator polynomial in backward shift operator z^{-1}
A_m	Gain margin
B	Plant numerator polynomial in backward shift operator z^{-1}

$C(s)$ Controller transfer function

d Time delay of a system (*samples*)

DM Delay margin

$e(t)$ Control error

$G(s)$ Process transfer function

G_{gap} Generation gap in Genetic Algorithms

h Sampling period

i Imaginary unit

IAE_{both} IAE of servo response and regulator response

IAE_{reg} IAE of regulator response

IAE_{servo} IAE of servo response

IAE_{um} IAE of control signal for the nominal model of the flexible link

IAE_{up} IAE of control signal for the flexible link

IAE_{ym} IAE of control error for the nominal model of the flexible link

IAE_{yp} IAE of control error for the flexible link

K_d Derivative gain in PID controller

K_i Integrating gain in PID controller

K_p Proportional gain in PID controller

$L(s)$ Open-loop transfer function

L_{ind} Length of the chromosome structure

M_p Resonance peak of the complementary sensitivity transfer function $\mathcal{T}(s)$

M_s	Resonance peak of the sensitivity transfer function $\mathcal{S}(s)$
M_u	Resonance peak of the input sensitivity transfer function $\mathcal{U}(s)$
MM	Modulus margin
N_1	Minimum cost horizon for GPC control law
N_2	Maximum cost horizon for GPC control law
n_a	Order of the process pole polynomial
n_b	Order of the process zero polynomial
N_u	Control horizon
N_{best}	Number of the best individuals
N_{pop}	Number of populations
P_m	Mutation probability
R	Polynomial of the GPC linear control law corresponding to the first row of $\mathbf{H} \times \mathbf{G}$ product
S	Polynomial of the GPC control law corresponding to the first row of the product $\mathbf{H} \times \mathbf{F}$
T	GPC observer polynomial
T_d	Time delay of a system (<i>sec</i>)
T_i	Integration time
T_r	Rise time
T_s	Settling time of servo response
T_{rd}	Settling time of regulatory response
$u(t)$	Control signal

$u_d(t)$	Disturbed control signal
$y(t)$	Process output
y_d	Overshoot at disturbance
y_p	Overshoot of step response
y_{max}	Peak amplitude of step response
y_{ss}	Steady state value
$\mathcal{S}(s)$	Sensitivity transfer function
$\mathcal{T}(s)$	Complementary sensitivity transfer function
$\mathcal{U}(s)$	Input sensitivity transfer function
CARIMA	Controlled Autoregressive and Integrated Moving Average
CPP	Cancellation Pole Placement
DAU	Data Acquisition Unit
DMC	Dynamic Matrix Control
GA	Genetic Algorithm
GPC	Generalised Predictive Control
IAE	Integrated Absolute Error
ISE	Integrated Squared Error
ITSE	Integrated of Time Squared Error
LQG	Linear Quadratic Gaussian
LRPC	Long Range Predictive Control
MBPC	Model Based Predictive Control

NOMENCLATURE

OSAC One Step Ahead Control

PRBS Pseudo Random Binary Sequence

SISO Single Input Single Output

SUS Stochastic Universal Sampling

Chapter 1

Introduction

PID feedback control remains a very popular control strategy in industrial processes. According to Åström and Hägglund (2001) more than 90% of all control loops involve the PID controller. The method is well understood and accepted by operating personnel and control engineers due to the intuitive simplicity of the algorithm. Detailed knowledge of a process is not required to successfully implement PID control, which is well suited for many processes with different dynamics.

Besides the PID controller, many advanced control algorithms exist, for example predictive control. The predictive control philosophy was first advocated by Smith (1957) though many variations now exist. To demonstrate the advantage of the predictive control methodology consider two control strategies. The first one uses only the current value of the system output as feedback while the alternative strategy uses knowledge of the future behaviour of the output to adjust the manipulated variable. Which control strategy is expected to perform better? Clearly, the answer is the second control strategy, as the necessary control action can be taken beforehand if the future behaviour of the system can be predicted. This is the reason why much research has focused on predictive control.

For the majority of control problems faced by industry, the omnipresent PID controller provides sufficient capability. Nevertheless, the PID controller is still restricted in its performance when controlling processes with long time delay, nonminimum phase behaviour or unusual dynamics. For more complex control problems, advanced techniques

such as *Model Based Predictive Control* (MBPC), also referred to as *Model Predictive Control* (MPC), may achieve better control performance. Even though advanced techniques may provide superior control, there are situations where they are not utilised for different reasons. A major barrier to the successful implementation of advanced controllers is complexity. In contrast to the PID controller, a model of a process has to be obtained when the MBPC algorithm is implemented. Certainly, this limits its popularity.

1.1 Review of existing controller comparisons

Over the past few decades, control engineers, (particularly academics) have devised a bewildering range of control algorithms to overcome the limitations of the ideal PID controller. Frequently, these proposals are justified based on superior simulation performance with a PID controller. In many cases, these claims are unjustified because the PID controller was poorly designed and tuned, e.g. using the Ziegler–Nichols technique (Wang and Kwok, 1994; Leonard, 1997; Neto and Embirucu, 2000; Visioli, 2000; Burden et al., 2001; Sree et al., 2004). Thus, the contribution of this thesis is to compare an appropriately designed and well-tuned PID controller with an advanced controller algorithm. The motivation for this study is to determine whether the advanced control algorithms can offer any real benefits to the process industry in terms of optimising *Single-Input Single-Output* (SISO) process units by, for example, maximising throughput, minimising waste, minimising energy, etc. As a representative of the MBPC paradigm, the *Generalised Predictive Control* (GPC) philosophy was evaluated.

A number of researchers have previously compared the PID and the GPC algorithms. For example, Camacho and Bordons (1995, sec 8.2) applied GPC and a PI controller to a solar power plant and found that the GPC performed faster and with less overshoot than the PI controller. However, their results suffer from the bias mentioned above; the authors note that “the results obtained by the PI controller, tuned by the Ziegler–Nichols open-loop tuning rules, were very oscillatory, better results were obtained by E.R. Velasco, after a long commissioning period where «optimal» PID parameters were obtained”. Additional comparisons were performed by Owens and Warwick (1988),

Jolly and Bentsman (1993), Josi et al. (1997) and Burden et al. (2001). The first two of these papers compare GPC with a PI(D) controller and the GPC algorithm is reported to perform better: the former in simulation while an experimental continuous casting process was used in the Jolly & Bentsman evaluation. Both reports are deficient in that no details of the PI(D) tuning procedure were supplied. Though the comparison performed by Josi et al. (1997) is more thorough, few details of the PID algorithm are presented. The authors state that a cascade configuration with feedforward control and gain scheduling was used, but the particulars of the design process were not provided. This advanced classical controller is compared with a number of model-based schemes on an experimental shell and tube heat exchanger. MBPC is rated as more favourable than the classical technique but a process model-based control scheme that could cater for the process non-linearities was found to be the most favourable.

A pilot-scale desalination process was used by Burden et al. (2001) to compare a Ziegler-Nichols tuned PID controller with a constrained model-based predictive approach. Not surprisingly, these researchers reported that the PID approach “failed to control the plant properly”. An additional GPC/PI comparison was performed by Fikar and Draeger (1995). In this contribution both controllers were applied to a neutralisation reactor and it was shown that the adaptive GPC is able to control the strongly nonlinear plant and that it behaves much better than the “carefully tuned PI controller”. The authors also pointed out that “the GPC parameters have been chosen with some care”. Owens and Warwick (1988) also compared an adaptive GPC with a fixed “well-tuned” PID controller. Such comparisons are disingenuous towards the PI(D) controller as the non-linear adaptive structure is obviously going to outperform a fixed linear structure on a strongly non-linear process or simulation.

In contrast to the foregoing comparisons, the work reported by Liu and Daley (1999a) is noteworthy in that an advanced tuning strategy was applied to design the PID controller. The authors compared a PID control scheme, a fixed parameter predictive controller and an adaptive controller on a non-linear atmospheric test rig with a commercial combustor. The PID controller was tuned using the Simplex optimisation method to achieve a specified second-order response for the identified NO_x model. Experi-

mental tests indicated that the adaptive algorithm outperformed both fixed algorithms while the fixed predictive scheme achieved better tracking performance than the optimal PID scheme. In the work conducted by O'Mahony and Downing (1999), the PID controller was tuned using a pole-placement design technique advocated by Warwick and Kang (1996). The authors demonstrated that, for a simple process (level control of a laboratory-scale single tank system), the GPC offers few enhancements over the PID controller.

Mohtadi and Clarke (1986) and O'Mahony and Downing (2000b) compared GPC, the *Linear Quadratic Gaussian* (LQG) controller and the *Cancellation Pole Placement* (CPP) controller. Each of these controllers has a 2-DOF structure. The authors noticed that the GPC is relatively robust to mismodelling but it is quite sensitive to the effect of high-frequency measurement noise. The CPP controller resulted in a smoother control signal and was subsequently the controller of choice for the investigated application. The GPC controller has also been compared with \mathcal{H}_∞ control (Ordys et al., 2000). Both controllers were successfully applied to a laser scanner system. The authors found that the \mathcal{H}_∞ controller was easier to tune and provided better performance while the GPC was capable of removing overshoots.

1.2 Problem statement

The aim of this thesis is to investigate whether advanced control algorithms offer any significant performance improvement for SISO loops relative to the ideal PID controller. In contrast to much of the research outlined in the preceding paragraphs, this thesis will endeavour to perform as fair a comparison as possible. This will be achieved by applying a systematic controller design philosophy to both the PID and the advanced control algorithm. This design philosophy is based on a review of existing approaches which can be broadly classified as being based on (i) gain and phase margins and (ii) optimising a sensitivity function criterion. Both approaches are investigated and the optimisation is performed using a *Genetic Algorithm* (GA). In addition to the ideal PID structure, the performance of a Two Degree of Freedom (2-DOF) PID controller will also be evaluated. The inclusion of this algorithm is important as the ideal PID

controller is rarely applied in industry and therefore the results will be more relevant to the process industry. Furthermore, incorporating the 2-DOF PID controller removes a source of bias — namely controller structure. As this controller and the GPC have two degrees of freedom, the structures are similarly complex, the degrees of freedom are the same and the number of tuning parameters are similar. Each of the three controllers will be evaluated, in simulation, using thirteen benchmark processes (Åström and Hägglund, 2000; Åström et al., 1998) that are representative of typical process transfer functions. Finally, a real-time evaluation will also be performed on a laboratory-scale flexible link system.

1.3 Organisation of the thesis

The remainder of the thesis is organised as follows:

Chapter 1 introduces the reader to the research problem and provides an introduction to the remainder of the thesis.

Chapter 2 describes the three controllers that are to be evaluated; the ideal PID controller, the two degree of freedom PID controller and the Generalised Predictive Controller. These controllers are widely documented in the literature and hence only a brief description is presented. In addition, the GPC tuning parameters are analysed to demonstrate the non-convex relationship between these parameters and classical performance indices such as the integral of error criterion. The analysis is presented to demonstrate the need for alternative optimisation routines, such as Genetic Algorithms.

Chapter 3 presents the Genetic Algorithm optimisation technique. The performance criteria used in the controller evaluation are also described. First, the servo performance is considered and definitions of typical performance metrics are given. In section 3.3 robustness criteria are presented. A review of existing tuning methods is provided in section 3.4 and then the critical component of the optimisation technique, the objective function, is detailed.

Chapter 4 analyses the thirteen models and a brief description of the chosen sampling period is presented. An overview of the optimisation environment and the control algorithm implementation details are given. The selected controller structures are tuned for each of the thirteen models using (i) a design that minimises the *Integral of Absolute Error* (IAE) criterion subject to constraints on the gain and phase margin, and (ii) a design that minimises the IAE criterion subject to constraints on a sensitivity function. The results for each of the controller structures are summarised and discussed.

Chapter 5 evaluates the controllers and design philosophies on a real-time system. The three previously mentioned controllers are designed and implemented on the real-time system. The results of the real-time evaluation are presented and discussed.

Chapter 6 draws conclusions from the results and arguments presented in the thesis and suggests possible directions for future research.

Appendix A lists results — controller coefficients and performance criteria — obtained from the simulation evaluation for designs based on the gain and phase margin criteria and the modulus margin.

Appendix B details the results obtained from the flexible link.

Chapter 2

Controllers

In this chapter three controllers are described. The ideal PID controller, the two degree of freedom PID controller (2-DOF PID) and the *Generalised Predictive Controller* (GPC). These controllers are widely documented in the literature and hence only a brief description of the control laws are presented here. In addition, the GPC tuning parameters are analysed to demonstrate the non-convex relationship between these parameters and classical performance indices such as the integral of error criterion. The analysis is presented to demonstrate the need for alternative optimisation routines, such as Genetic Algorithms.

2.1 PID controller

The terms P, I, D, may be interpreted as: P-proportional action which effects the present, I-integral action (past), and D-derivative action (future control error). In the case of pure proportional control, the control action is simply proportional to the control error. The main function of integral action is to make sure that the process output agrees with the set-point in steady state. The aim of the derivative action is to improve closed-loop stability.

Figure 2.1 illustrates a block diagram representation of a typical feedback loop, where $U_d(s)$ is the disturbing control signal. The loop consists of a controller, $C(s)$, and a process $G(s)$. The aim of the regulator is to minimise the error which is the difference

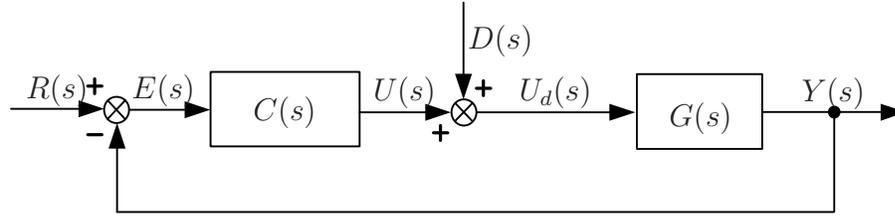


Figure 2.1 Feedback loop

between the set-point, $R(s)$, and actual output, $Y(s)$. In other words, the controller calculates the signal $U(s)$ such that $E(s)$ is zero, despite the occurrence of disturbances and/or changes to the set-point command.

The structure of the ideal PID controller is given by equation 2.1

$$U(s) = E(s) \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (2.1)$$

where $E(s)$ is

$$E(s) = R(s) - Y(s) \quad (2.2)$$

In equation 2.1, K_p , K_i and K_d are controller parameters.

2.2 Two degree of freedom PID controller

Equation 2.1 is known as a single degree of freedom control law because the controller parameters may be optimised for servo performance or regulatory performance but not both. Most industrial controllers are considerably more complex than the “textbook” control law of equation 2.1 and a *Two Degree of Freedom PID Controller*, (2-DOF PID), is given by equation 2.3

$$U(s) = K_p (b \cdot R(s) - Y(s)) + \frac{K_i}{s} (R(s) - Y(s)) + \frac{sK_d (c \cdot R(s) - Y(s))}{1 + \frac{sK_d}{K_p N}} \quad (2.3)$$

where N is the derivative filter and $\{b, c\}$ are weightings that influence the set-point response. Equation 2.3 may be restructured as

$$U(s) = R(s)F(s) - Y(s)H(s) \quad (2.4)$$

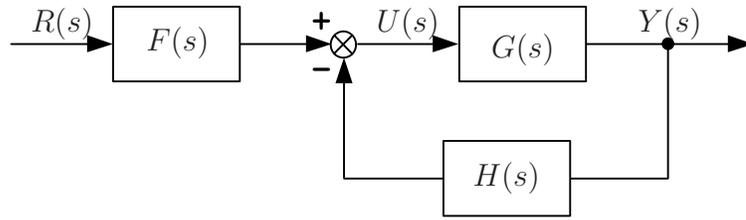


Figure 2.2 Block diagram of 2–DOF PID controller

where $F(s)$ and $H(s)$ are defined by equations 2.5 and 2.6 respectively. This representation is illustrated in figure 2.2.

$$F(s) = K_p b + \frac{K_i}{s} + \frac{sK_d}{1 + \frac{sK_d}{K_p N}} c \quad (2.5)$$

$$H(s) = K_p + \frac{K_i}{s} + \frac{sK_d}{1 + \frac{sK_d}{K_p N}} \quad (2.6)$$

Clearly, the industrial PID structure corresponds to a 2–DOF control law. The transfer function $H(s)$ may be designed to yield optimal regulator performance while the transfer function $F(s)$ can be chosen to yield good servo performance. Note that K_p , K_i , K_d and N appear in both of these transfer functions and hence it is not possible to specify the regulator performance independently of the servo performance. Thus, the 2–DOF PID controller is not a true 2–DOF controller. However, the parameters b and c may be utilised to enhance the servo performance while not affecting the regulator performance. For the ideal PID controller $F(s)$ and $H(s)$ are equal, given by equation 2.1.

2.3 Model–Based Predictive Controller

Model–Based Predictive Control (MBPC) is a field which has attracted much research interest and attention. The idea of MBPC can be traced to the work of Kalman (1960a), (1960b) when the *Linear Quadratic Gaussian* (LQG) concept was introduced. MBPC has found wide spread applications in areas where it is economically feasible to develop an accurate process model, e.g. in the petrochemical industry (Clarke, 1988; Grimble and Ordys, 2001). These applications are typically described by non–linear, multi–variable, constrained process models for which the MBPC philosophy is ideally suited.

While different control schemes with predictive capability are now available (Qin and Badgwell, 2003), one popular class of predictive control is the *Generalised Predictive Control* (GPC) introduced by Clarke et al. (1987a), (1987b). The initial derivation of the GPC controller was for SISO systems; however *Multi-Input Multi-Output* (MIMO) implementations (Camacho and Bordons, 1995, ch. 5), constrained versions, (Camacho, 1993; de Madrid et al., 1994) and also continuous-time versions (Demircioğlu, 1994; Ronco et al., 1999) also exist.

As distinct from PID controllers, predictive controllers know the reference *a priori*, therefore the system can react before changes occur, thus avoiding the effects of delay in the process response. The methodology of predictive control is presented in figure 2.3. The future outputs for an appointed prediction horizon, N_2 , also called the maximum cost horizon, are predicted at each sample, t , using a model of the process. These predicted outputs, $y(t+k|t)$ ¹ for $k = N_1 \dots N_2$, depend on the known values up to instant t and the future control signal, $u(t+k|t)$, $k = N_1 - 1 \dots N_2 - 1$, which is calculated by optimising a cost function in order to keep the output close to the set-point. The control effort is included in the objective function and weighted by a parameter λ , see equation 2.9. An explicit solution can be obtained if the criterion is quadratic, the model is linear and there are no constraints; otherwise an iterative optimisation method has to be used. The control signal, $u(t|t)$, is sent to the process whilst the remaining calculated control signals are rejected because at the next sampling instant $y(t+1)$ is already known and the procedure is repeated with this new value and all the sequences are brought up to date. Thus, $u(t+1|t+1)$ is calculated (which in principle will be different to $u(t+1|t)$ because of the new information available) using the most recent information, (Camacho and Bordons, 1995).

2.4 Control law of the GPC

There are three major components in the design of a GPC:

- A model of the system to be controlled. This model is used to predict the system

¹This notation refers the predicted value y at the instant $t+k$ calculated at instant t .

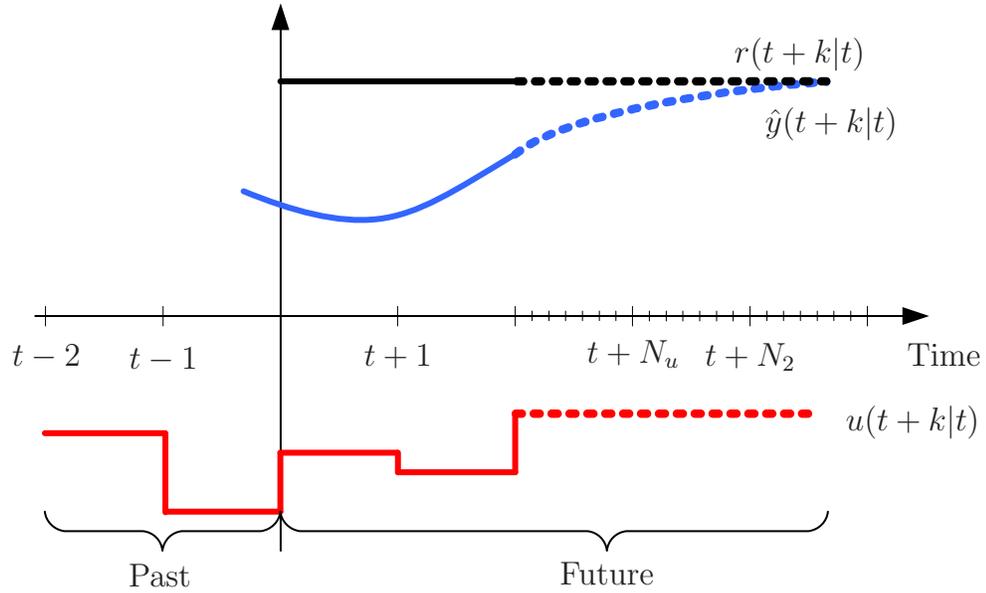


Figure 2.3 Predictive control law

output over the prediction horizon. Consider the SISO *Controlled AutoRegressive and Integrated Moving Average* (CARIMA) model of the form:

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + C_d(z^{-1})\frac{\xi(t)}{\Delta} \quad (2.7)$$

where B is the numerator of a plant given by a polynomial in the backward shift operator z^{-1} . The order of the B polynomial is denoted n_b .

A is the denominator of a plant given by a polynomial in the backward shift operator z^{-1} . The order of the A polynomial is denoted n_a .

$\xi(t)$ is an uncorrelated random sequence

Δ is the difference operator defined as $1 - z^{-1}$.

To derive a j -step ahead predictor, the Diophantine equation 2.8 is solved.

$$1 = E_j(z^{-1})A\Delta + z^{-j}F_j(z^{-1}) \quad (2.8)$$

where E_j and F_j are the corresponding controller polynomials with degrees $j-1$ and n_a , respectively (Clarke et al., 1987a), (1987b).

- The various predictive control algorithms utilise different cost functions. The general aim is that the future output should follow a pre-determined reference

signal, taking into account the control effort required to do so. The GPC cost function is given by equation 2.9

$$J = \sum_{j=N_1}^{N_2} [\hat{y}(t+j) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \quad (2.9)$$

where

N_1 is the minimum cost horizon, subject to $N_1 \leq N_2$

N_2 is the maximum cost horizon

N_u is the control horizon, subject to $N_u \leq N_2$

λ is the control weighting sequence, subject to $\lambda \geq 0$

ω is the future reference trajectory.

- Minimising the cost function yields the optimal control output. The minimisation of equation 2.9, assuming no constraints on future controls, results in the projected control increment vector given by equation 2.10

$$\tilde{\mathbf{u}} = (\mathbf{G}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \quad (2.10)$$

as will now be demonstrated.

Referring to equation 2.7, note that if the disturbance is modelled correctly i.e. if an accurate estimate for the polynomial $C(z^{-1})$, i.e. $\hat{C}(z^{-1}) = C(z^{-1})$, is obtained, a minimum variance predictor results, leading to a control law optimised for maximum quality. Alternatively, an assumed model, $T(z^{-1}) = C(z^{-1})$, can lead to enhanced robustness. This latter interpretation is the one generally adopted in the GPC literature, (Clarke et al., 1987a), (1987b) and will also be adopted in this thesis.

Given the model 2.7 the objective is then to predict the future output $\hat{y}(t+j)$ so that corrective control action, $u(t)$, can be invoked at time instant t . The problem then relies on the optimal j -step ahead prediction of $y(t)$:

$$\hat{y}(t+j) = \frac{B}{A}u(t+j-1) + \frac{T}{A\Delta}\xi(t+j) \quad (2.11)$$

Given j , $A(z^{-1})$, $B(z^{-1})$, $\{u(t)\}_{t-n_b}^t$, the deterministic component

$$\hat{y}(t+j) = \frac{B}{A}u(t+j-1) \quad (2.12)$$

can be predicted exactly. In contrast, the additive stochastic disturbance consists of both a discernible and unpredictable entity. For optimal predictions it is necessary to include the observable component of the stochastic disturbance in the j -step ahead predictor. To achieve this the rational function $T/A\Delta$ is split into a quotient and remainder by application of the division algorithm as follows:

$$\frac{T}{A\Delta} = E_j(z^{-1}) + z^{-j} \frac{F_j}{A\Delta} \quad (2.13)$$

$$T = E_j A\Delta(z^{-1}) + z^{-j} F_j \quad (2.14)$$

where E_j is a polynomial of degree $j-1$ and F_j is a polynomial of degree $\max\{n_a, n_t-1\}$ where n_t is the order of the T polynomial. Multiplying 2.11 by $E_j A\Delta$ and substituting for $E_j A\Delta$ from equation 2.14 yields

$$T(z^{-1})\hat{y}(t+j) = G'_j \Delta u(t+j-1) + F_j \hat{y}(t) + E_j T \xi(t+j) \quad (2.15)$$

$$\hat{y}(t+j) = G'_j \Delta u^f(t+j-1) + F_j \hat{y}^f(t) + E_j \xi(t+j) \quad (2.16)$$

In equation 2.16 the superscript f denotes quantities filtered by T . This j -step-ahead predictor runs independently of the process and, for this precise reason, is not suited for practical applications. To account for possible model mismatch or unmodelled disturbances it is necessary to incorporate information about the current state of the process output. This may simply be achieved by replacing $\hat{y}^f(t)$ in equation 2.16 with the actual measured output $y(t)/T = y^f(t)$. Note also that since E_j is a polynomial of degree $j-1$ the noise component $E_j \xi(t+j)$ is comprised of terms that are all in the future and therefore the optimal j -step ahead predictor is defined by

$$\hat{y}(t+j) = G'_j \Delta u^f(t+j-1) + F_j y^f(t) \quad (2.17)$$

and the corresponding prediction error is –

$$\varepsilon(t) = E_j \xi(t+j) \quad (2.18)$$

The predictor defined by equation 2.17 may be combined with a quadratic cost which is subsequently minimised to obtain the optimal future control sequence $\tilde{\mathbf{u}} = [\Delta u(t), \Delta u(t+1), \dots]$. This optimisation is hindered by the fact that the first component on the right

hand side of equation 2.17 consists of both past and future terms of the manipulated variable. To simplify the optimisation problem, it is thus necessary to split this component into its constituent past, $\Delta u(t - 1)$, and future, $\Delta u(t + j - 1)$, components. This may be achieved by noting that

$$G'_j \Delta u^f(t + j - 1) = \frac{G'_j}{T} \Delta u(t + j - 1) \quad (2.19)$$

and, by invoking the division algorithm, the term G'_j/T may be split into a quotient and remainder as follows:

$$\frac{G'_j}{T} = G_j + z^{-j} \frac{\bar{G}_j}{T} \quad (2.20)$$

$$G'_j = G_j T + z^{-j} \bar{G}_j \quad (2.21)$$

where the order of G_j is $j - 1$ and \bar{G}_j is of degree $\max(n_b, n_t) - 1$. Substituting for G'_j in equation 2.17 yields the following, preferable, representation for the j -step ahead predictor

$$\hat{y}(t + j) = G_j \Delta u(t + j - 1) + \bar{G}_j \Delta u^f(t - 1) + F_j y^f(t) \quad (2.22)$$

This is now in the correct form in that the predicted outputs are directly related to future but unknown control increments. In addition, the predictor can now be represented by two terms; one depending entirely on future values of the manipulated variable; the other on past values of the manipulated variable and past measured outputs.

Equation 2.22 represents a prediction of the output at a single time instant j . To improve the reliability of the predictions the process output may be predicted over a wider band. GPC considers a bank of predictions for which j varies from a small value to a large value: these are known as the initial, N_1 , and final, N_2 , prediction horizons. Consequently 2.22 may be restated as

$$\hat{y}(t + j) = G_j \Delta u(t + j - 1) + \bar{G}_j \Delta u^f(t - 1) + F_j y^f(t) \quad N_1 \leq j \leq N_2 \quad (2.23)$$

In key-vector form this representation is:

$$\hat{\mathbf{y}} = \mathbf{G} \bar{\mathbf{u}} + \mathbf{f}; \quad \mathbf{f} = \bar{\mathbf{G}} \Delta u^f(t - 1) + \mathbf{F} y^f(t) \quad (2.24)$$

where the vectors are defined as:

$$\hat{\mathbf{y}} = [\hat{y}(t + N_1), \dots, \hat{y}(t + N_2)]^T$$

$$\begin{aligned}\tilde{\mathbf{u}} &= [\Delta\hat{u}(t + N_1 - 1), \dots, \Delta\hat{u}(t + N_u - 1)]^T \\ \mathbf{f} &= [f(t + N_1), \dots, f(t + N_2)]^T\end{aligned}$$

and the matrix \mathbf{G} is of dimension $[N_2 - N_1 + 1 \times N_u]$ and defined by

$$\mathbf{G} = \begin{bmatrix} G_{N_1} \\ \vdots \\ G_{N_2} \end{bmatrix} = \begin{bmatrix} g_{N_1-1} & \cdots & g_0 & 0 \\ g_{N_1} & & & \\ & \ddots & & g_0 \\ & \vdots & \vdots & \vdots \\ g_{N_2-1} & \cdots & & g_{N_2-N_u} \end{bmatrix} \quad (2.25)$$

The apparent inconsistencies in the indexing of the elements of the \mathbf{G} matrix may be resolved by noting that for each row, G_j , the first $j - 1$ elements, $g_0 \dots g_{j-1}$, remain unchanged, consequently the polynomials G_j are defined as

$$\begin{aligned}G_{N_1} &= g_0 + g_1 z^{-1} + \dots + g_{N_1-1} z^{N_1-1} \\ G_{N_1+1} &= g_0 + g_1 z^{-1} + \dots + g_{N_1-1} z^{N_1-1} + g_{N_1} z^{N_1} \\ &\vdots \\ G_{N_2} &= g_0 + g_1 z^{-1} + \dots + g_{N_1-1} z^{N_1-1} + \dots + g_{N_2-1} z^{N_2-1}\end{aligned}$$

The GPC control law, equation 2.9, is arrived at by minimising a quadratic cost function comprising the future values of the predicted errors plus some weighting on the future control signals. Note that for $N_1 = 1$, equation 2.9 is equivalent to minimising

$$J = \sum_{j=1}^{N_2} \{[\hat{y}(t + j) - w(t + j)]^2 + \lambda(j) [\Delta u(t + j - 1)]^2\} \quad (2.26)$$

subject to the equality constraint

$$\Delta u(t + j - 1) = 0 \quad \text{for } N_u < j \leq N_2 \quad (2.27)$$

Hence the incorporation of a control horizon $N_u \leq N_2$ is equivalent to placing infinite weights i.e. $\lambda(j) = \infty$, on the control signal increments for $j > N_u$. The cost 2.9 may also be re-written in matrix notation as:

$$J = (\hat{\mathbf{y}} - \mathbf{w})^T (\hat{\mathbf{y}} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} \quad (2.28)$$

Using (2.24) yields the expanded cost

$$J = (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w})^T(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w}) + \lambda\tilde{\mathbf{u}}^T\tilde{\mathbf{u}} \quad (2.29)$$

Multiplying through yields

$$J = \tilde{\mathbf{u}}^T(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})\tilde{\mathbf{u}} + 2\tilde{\mathbf{u}}^T\mathbf{G}^T(\mathbf{f} - \mathbf{w}) + (\mathbf{f} - \mathbf{w})^T(\mathbf{f} - \mathbf{w}) \quad (2.30)$$

The minimum value of the cost 2.30 is obtained by differentiating w.r.t. $\tilde{\mathbf{u}}$ and setting the result to zero i.e.

$$\frac{\delta J}{\delta \tilde{\mathbf{u}}} = 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})\tilde{\mathbf{u}} + 2\mathbf{G}^T(\mathbf{f} - \mathbf{w}) = 0 \quad (2.31)$$

$$\Rightarrow \tilde{\mathbf{u}} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \quad (2.32)$$

Equation 2.32 gives the whole trajectory of future control increments and as such it is an open-loop strategy. To close the loop, only the first element of $\tilde{\mathbf{u}}$ is extracted and applied to the system and the optimisation recomputed at time instant $t + 1$. This strategy is called the *receding horizon principle* and is one of the key features in the MPC concept. Recall the first element of $\tilde{\mathbf{u}}$ is $\Delta u(t)$ so that the applied control signal $u(t)$ is defined by

$$u(t) = u(t - 1) + \Delta u(t) \quad (2.33)$$

For the purpose of closed-loop analysis it is possible to rearrange the GPC control law, equation 2.32, into an equivalent linear control law. To achieve this it is convenient to assume that the equivalent linear control law is causal i.e. no future set-point information is incorporated. If the first row of the $N_u \times N_2 - N_1 + 1$ matrix $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$ is denoted by

$$h = [h_{N_1} \quad \dots \quad h_{N_2}] \quad (2.34)$$

then the current control increment, $\Delta u(t)$, may be calculated from

$$\Delta u(t) = [h_{N_1} \quad \dots \quad h_{N_2}] \cdot \begin{bmatrix} w(t) - \bar{G}_{N_1}\Delta u^f(t-1) - F_{N_1}y^f(t) \\ \vdots \\ w(t) - \bar{G}_{N_2}\Delta u^f(t-1) - F_{N_2}y^f(t) \end{bmatrix} \quad (2.35)$$

$$= \sum_{j=N_1}^{N_2} h_j \cdot w(t) - \sum_{j=N_1}^{N_2} h_j \bar{G}_j \cdot \Delta u^f(t-1) - \sum_{j=N_1}^{N_2} h_j F_j \cdot y^f(t) \quad (2.36)$$

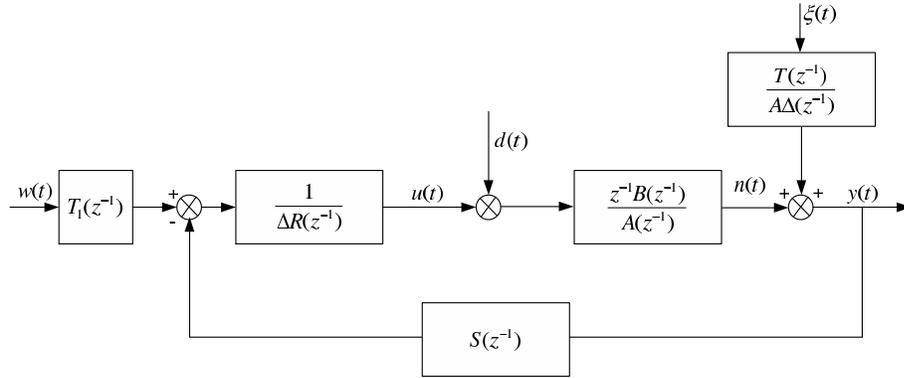


Figure 2.4 Block diagram for equivalent linear form of GPC

Multiplying across by $T(z^{-1})$ and re-arranging yields the following linear causal 2DOF control law

$$R(z^{-1})\Delta u(t) = T_1(z^{-1})w(t) + S(z^{-1})y(t) \quad (2.37)$$

where

$$R(z^{-1}) = T + z^{-1} \sum_{j=N_1}^{N_2} h_j \bar{G}_j ; \quad \deg(R) = \max(n_t, n_b) \quad (2.38)$$

$$T_1(z^{-1}) = T \sum_{j=N_1}^{N_2} h_j ; \quad \deg(T_1) = n_t \quad (2.39)$$

$$S(z^{-1}) = \sum_{j=N_1}^{N_2} h_j F_j ; \quad \deg(S) = \max(n_a, n_t - N_1) \quad (2.40)$$

The corresponding block diagram is illustrated in figure 2.4.

2.5 GPC tuning parameters

In the ideal PID controller structure there are only three tuning parameters, K_p , K_i and K_d . Tuning the GPC controller is more complex, as there are more tuning parameters available. Some of them, like the T polynomial, (assuming perfect modelling) only influence the disturbance rejection properties while other parameters, like the prediction horizons, influence both the disturbance rejection and the robustness properties. In this section the tunable parameters are briefly described. For a more complete study of predictive controller tuning rules, see Rani and Unbehauen (1997).

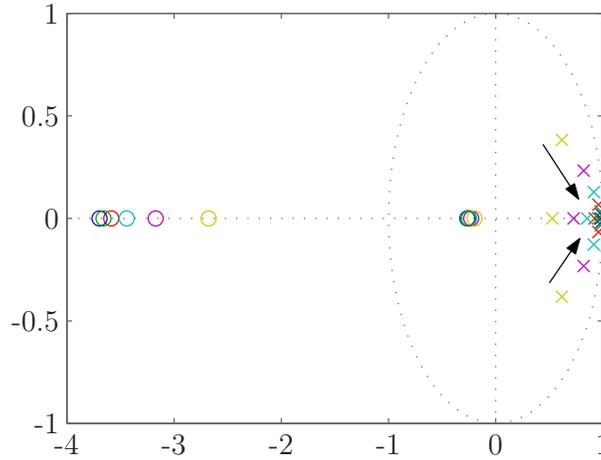


Figure 2.5 Poles (\times) and zeros (\circ) when the system in equation 2.41 is sampled with changing h

2.5.1 The sampling period, h

In the design of a digital controller it is necessary to choose a sampling period, denoted in this thesis as h . If the sampling period too long control of the process is more difficult as large deviations following a load disturbance may arise. However, short sampling periods require a larger N_2 and hence increase the load on the processor. Also, poles will move towards the point $z = -1$ which may cause numerical problems. To demonstrate this, consider the continuous-time transfer function

$$G(s) = \frac{1}{(s+1)(s^2+s+1)} \quad (2.41)$$

Decreasing the sampling period over the range $h = 1.28, 0.64, 0.32 \dots 0.02$ moves the poles towards -1 , as illustrated in figure 2.5. When the transfer function, $G(s)$, is sampled using the zero-order hold method with $h = 0.02$ the coefficients of the discrete-time transfer function $G(z)$ decrease, which can cause numerical problems. The guidelines for choosing the sampling period are presented in the following sections.

Settling time

The settling time, T_s , is defined as the time taken for the process step response to reach and remain within 2% of its steady state value. The sampling period should be less than

$1/10^{th}$ of either the open or closed-loop settling time depending on which is shorter.

$$h = 0.1 \cdot T_s \tag{2.42}$$

For an oscillatory system this technique is inappropriate because, in general, the sampling period will be too long. This technique is very simple and recommended by Isermann (1981).

Rise time

Rise time, T_r , is either defined as the inverse of the largest slope of the step response or the time taken for the response to pass from 10% to 90% of its steady state value, y_{ss} . The sampling period can be obtained from equation 2.43

$$h = \frac{T_r}{N_r} \tag{2.43}$$

where N_r is the number of sampling periods per rise time and T_r is either the open or closed-loop rise time depending on which is shorter. For first order systems, Åström (1990, sec. 3.7) recommends that N_r should be chosen between 4 – 10.

Analysis in the frequency domain — the damped frequency method

The sampling period can be related to the damped frequency of the closed-loop system. Given the natural frequency, ω_n , and the damping factor, ζ , the sampling period is obtained from

$$h = \frac{2\pi}{N\omega_n\sqrt{1-\zeta^2}} \tag{2.44}$$

where the parameter N is the ratio of damped period to sampling period. It is reasonable to choose N between 25 – 75, see Åström (1990, sec. 9.2).

Analysis in the frequency domain — the bandwidth method

Another technique presented by Åström (1990, sec. 2.6) is based on a frequency domain analysis and uses the bandwidth of the closed-loop system. Illustration of the bandwidth, ω_b , is shown in figure 2.6 and is defined as the first frequency where the gain drops below 70.79 percent ($-3dB$) of its d.c. value. Reasonable sampling frequencies are ten

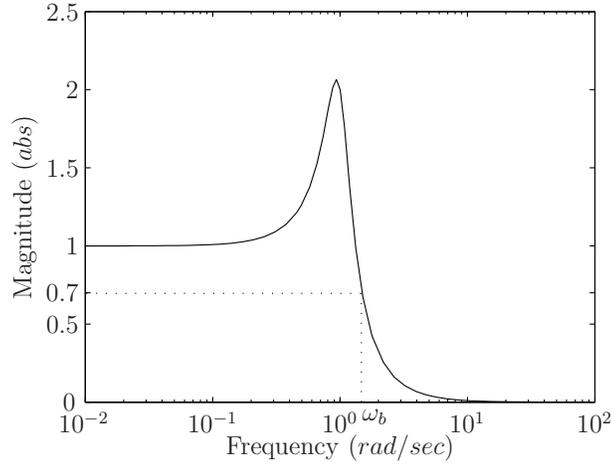


Figure 2.6 Presentation of bandwidth

to thirty times the bandwidth of the closed-loop system, as defined by equation 2.45

$$10\omega_b \leq \omega_s \leq 30\omega_b \quad (2.45)$$

where ω_s is the sampling frequency, defined as

$$\omega_s = \frac{2\pi}{h} \quad (2.46)$$

Analysis in the frequency domain — the natural frequency method

Another technique presented by Åström (1990, sec. 9.2) states that the sampling period should be chosen as shown in equation 2.47

$$\omega_n h = 0.2 - 0.6 \quad (2.47)$$

where ω_n is the desired natural frequency of the closed-loop system. These various techniques are applied in chapter 4 to obtain discrete-time models for the GPC.

2.5.2 The minimum output horizon, N_1

In general, it is recommended to set this parameter to 1 (Clarke et al., 1987a; Lambert, 1987). However for systems with dead time where the delay d is exactly known, there is no benefit in setting N_1 to be less than d since this results in superfluous calculations because $y(t+1)$ cannot be affected by the first action $u(t)$. If d is unknown or variable,

then N_1 can generally be set to 1 and the degree of $B(z^{-1})$ increased to encompass all possible values of d , (Clarke et al., 1987a).

2.5.3 The maximum output horizon, N_2

If the plant has an initially negative-going nonminimum phase response, N_2 should be chosen so that the later positive-going output samples are included in the cost. In discrete-time, this implies that N_2 exceeds the degree of $B(z^{-1})$. In practice, however, a rather larger value of N_2 is suggested,

$$d \leq N_2 \leq t_s \tag{2.48}$$

where t_s is the 95% settling time, (McIntosh et al., 1991; García and Morari, 1982).

2.5.4 The control horizon, N_u

For a simple plant (e.g. open-loop stable though with possible dead time and non-minimum phase behaviour), a value $N_u = 1$ generally gives acceptable control, (Clarke et al., 1987a). Small values of N_u have the added advantage that the computational burden is reduced, which is particularly important in the adaptive case, when fast calculations are necessary. This effect is evident from equation 2.25 where the dimension of the matrix \mathbf{G} is a function of N_u . The choice $N_u = 1$ also avoids numerical problems associated with inverting the $N_u \times N_u$ matrix $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})$ which reduces to a scalar and an inverse is guaranteed for $\lambda > 0$. In general, increasing N_u results in the corresponding process output response becoming more active until a stage is reached where any further increase in makes little difference, (Lambert, 1987). However, many researchers (McIntosh et al., 1991) advocate using N_2 rather than N_u to obtain a faster closed-loop response as numerical conditioning and computational complexity are not adversely affected. For more complex systems somewhat larger values of N_u have been useful. For example, Clarke et al. (1987a) found that good control was achieved when N_u is at least equal to the number of unstable or badly damped poles of the process transfer function.

2.5.5 The control weighting coefficient, λ

The control weighting coefficient can be used to penalise the control signal. It is particularly useful in negotiating the trade-off between minimising the error and minimising energy requirements. A non-zero value of λ also ensures that the matrix $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})$ maintains full rank and thus improves the numerical properties of the GPC algorithm, equation 2.10.

2.5.6 Effect of the T polynomial

The T polynomial, also known as the observer polynomial, is principally used to negotiate the trade-off between disturbance rejection and robustness. It should not be selected on the basis of good robustness properties alone because of its influence on the speed of disturbance rejection; slow observer poles are to be preferred for robustness but fast poles for good disturbance rejection. Thus a trade-off between the two must exist when designing an appropriate T polynomial, (Robinson and Clarke, 1991).

Optimality properties, such as minimum variance, are possible if the T polynomial is equal to $C_d(z^{-1})$ of equation 2.7. However, accurate identification of $C_d(z^{-1})$ is rare in practice (Robinson and Clarke, 1991) in which case T can be thought of as a fixed design polynomial and can be used to represent prior knowledge about the process noise. Predictors based on a fixed observer will not be optimal but a good design generally provides other important properties such as robustness to unmodelled dynamics. This latter interpretation is the one most commonly used in the GPC and has been demonstrated by many researchers e.g. Mohtadi (1988), McIntosh et al. (1989) and O'Mahony (2002). The latter researcher, for example, derives the following alternate expression for the GPC prediction error

$$\varepsilon(t+j) = \frac{A\Delta}{T} \left[\left(M - \frac{B}{A} \right) u(t+j-1) \right] \quad (2.49)$$

where $\varepsilon(t+j) = y(t+j) - \hat{y}(t+j)$ is the prediction error and M represents the true process dynamics. Robustness to model uncertainty may be achieved by ensuring that the filter $A\Delta/T$ has a low-pass characteristic over the frequency range where model uncertainty is significant; typically at high frequencies. This design philosophy effectively reduces

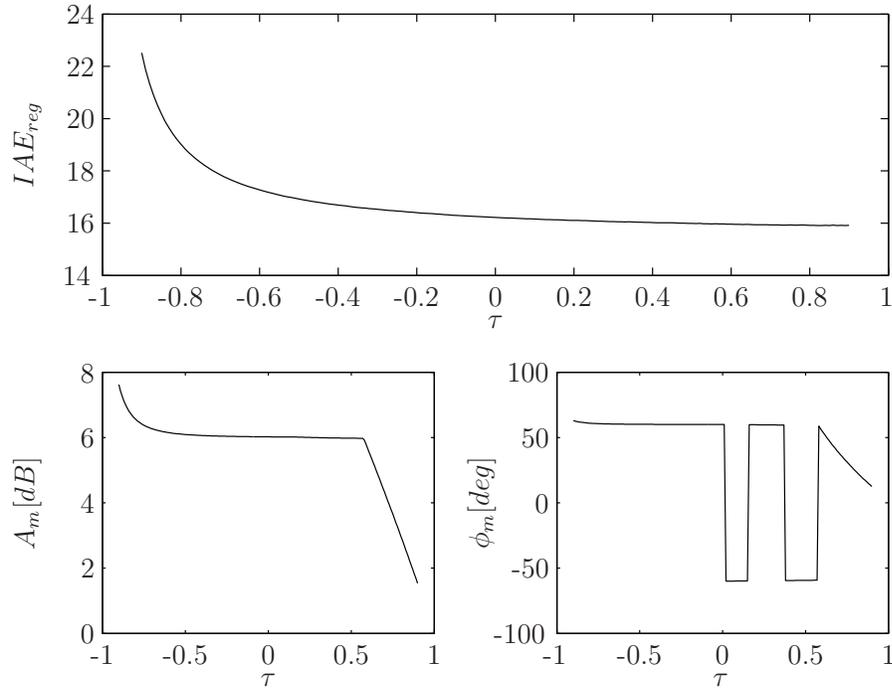


Figure 2.7 Effect of the T polynomial on the IAE_{reg} and the robustness criteria

the high-frequency gain of the GPC control law without affecting its low-frequency properties (Mohtadi, 1988; Robinson and Clarke, 1991).

To demonstrate the influence of the T polynomial consider the process

$$G_3(s) = \frac{1}{(s+1)^3} e^{-15s} \quad (2.50)$$

which is sampled every 0.7 seconds. The discrete-time system has a time delay of 21.4 samples hence, in accordance with section 2.5.2, the parameter N_1 is set to 22. The remaining GPC parameters were set by trial-and-error to achieve a satisfactory closed-loop response. These parameter settings are

$$\begin{aligned} N_1 &= 22 & N_2 &= 70 & N_u &= 1 & \lambda &= 1 \cdot 10^{-6} \\ T(z^{-1}) &= 1 - \tau(z^{-1}) \end{aligned} \quad (2.51)$$

In figure 2.7 robustness criteria such as the gain margin, A_m (equation 3.8), and the phase margin, ϕ_m (equation 3.10) are plotted as a function of the observer pole, τ . Both of these criteria are classical measures of robustness. The disturbance rejection performance is also illustrated using the *Integrated Absolute Error of the regulatory*

response, IAE_{reg} , criterion. This is plotted as a function of τ assuming a step size of 0.01 and is clearly discontinuous, though it is not clear why this is the case.

All three criteria, IAE_{reg} , A_m and ϕ_m , are maximised for $\tau = -1$ implying a robust closed-loop response but with poor disturbance rejection properties. As $\tau \rightarrow 0$ the disturbance rejection properties improve but the gain and phase margins decrease. Beyond $\tau = 0$, the IAE_{reg} remains approximately constant but both A_m and ϕ_m become distinctly non linear, with ϕ_m displaying distinct local minima.

2.6 Gradient techniques versus stochastic methods

It was necessary to select an appropriate optimisation routine to design the PID and GPC controllers. Techniques which do not employ optimisation, e.g. tuning rules, are usually easy to implement. However, they do not always work well and this motivated the use of optimisation techniques. To demonstrate the complexity of the optimisation problem, consider the following transfer functions.

$$G_1(s) = \frac{1}{(s+1)^3} \quad (2.52)$$

$$G_5(s) = \frac{1-2s}{(s+1)^3} \quad (2.53)$$

$$G_{10}(s) = \frac{100}{(s+10)^2} \left(\frac{1}{s+1} + \frac{0.5}{s+0.05} \right) \quad (2.54)$$

Model $G_1(s)$: The solution space for the ideal PID and GPC controllers is presented in figure C.1. Figure C.1(a) illustrates the Integrated Absolute Error of the servo response, IAE_{servo} , defined by equation 3.6 as a function of the parameters K_p , K_i and $K_d = 6$ for the PID controller. Likewise, figure C.1(b) illustrates the Modulus Margin, MM (section 3.3.2), as a function of the same parameters. The solution space in figure C.1(a) is convex and it is easy to find the minimum using gradient methods. However, an objective function based on the MM is not convex. Furthermore, the graphical solution becomes really non-convex if the derivative gain, K_d , is incorporated and gradient based techniques could fall foul to local maxima. This was the motivation for considering alternatives, such as genetic algorithms.

Model $G_5(s)$: The methodology outlined in the previous paragraph was also applied to the model $G_5(s)$ except that the gain K_d was set to 0.5. Figures C.2(a) and C.2(b) present the surface for the PID controller. The IAE criterion yields a convex surface while the MM criterion clearly results in a non-convex solution space. Figures C.3(a) and C.3(b) illustrate the IAE criterion and the MM criterion as a function of the GPC tuning parameters N_1 and N_2 . Though not obvious from the graph figure C.3(a) is non-convex. The global minimum occurs for $N_1 = 18$, $N_2 = 19$ and equals 3.24. A local minimum occurs at $N_1 = 17$, $N_2 = 19$ and is 6% larger than the global minimum. These coefficients do not correspond to any of the conventional GPC parameter settings, such as recommended by McIntosh et al. (1989), and it would be difficult to determine this solution based on trail and error. The MM criterion as a function of N_1 and N_2 yields a more complex surface with at least four local maxima and further supports the rationale for using non-traditional optimisation techniques.

Model $G_{10}(s)$: Consider figure C.5, the minimum $IAE_{servo} = 0.25$ for the PID controller occurs for $K_p = 11$, $K_i = 1$ assuming a constant $K_d = 1$. However, the solution space is quite flat, it is non-convex and the next local minimum occurs for $K_p = 11$, $K_i = 25$, which is demonstrated in figure C.4. The solution for the maximum MM in figure C.5(b) is non-convex and a local maximum occurs at $K_p = 1$ and $K_i = 40$. The GPC IAE_{servo} and the MM do not always decrease simultaneously. Examining C.6(a) and C.6(b) it is evident that the minimum IAE_{servo} and maximum value of MM coincide for $N_1 = 1$, $N_2 = 2$. To demonstrate the non-convex surface in figure C.6(b) the scale on the vertical axis is 0 to 0.2. However, a maximum of 0.88 occurs for $N_1 = 1$, $N_2 = 2$.

These three simulation studies have illustrated that, in general, common optimisation techniques such as linear and quadratic programming have limited validity as the majority of the solution surfaces are multimodal. These classical methods are susceptible to local minima and a global solution may not be found. This statement is even more applicable if the dimension of the search space is increased, e.g. if λ or the T polynomial is included or combined criteria e.g. $IAE_{servo} + MM$ are used. For this reason, a non-convex optimisation routine was applied to optimise the controller parameters.

Chapter 3

Objective functions

The optimisation technique, Genetic Algorithms, is presented in this chapter. The performance criteria used in the controller evaluation are described. First, the servo performance is considered and definitions of typical performance metrics are given. Then, in section 3.3, robustness criteria are presented. An overview of tuning methods is provided in section 3.4 and then the critical component of the optimisation technique, the objective function, is detailed.

3.1 Genetic Algorithms

The *Genetic Algorithms* (GA) approach is an intuitive and mature search and optimisation technique based on the principles of natural evolution and population genetics. Typically, the GA starts with little or no knowledge of the correct solution and depends entirely on responses from an interacting environment and its evolution operators to arrive at good solutions. By dealing with several independent points, the GA samples the search space in parallel and hence is less susceptible to converging to a suboptimal solution. In this way, the GA has been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality or false optima, as may occur with gradient descent techniques. Thus, the GA has been recognised as a powerful tool in many control applications. A current study of GA theory was done by Schmitt (2001).

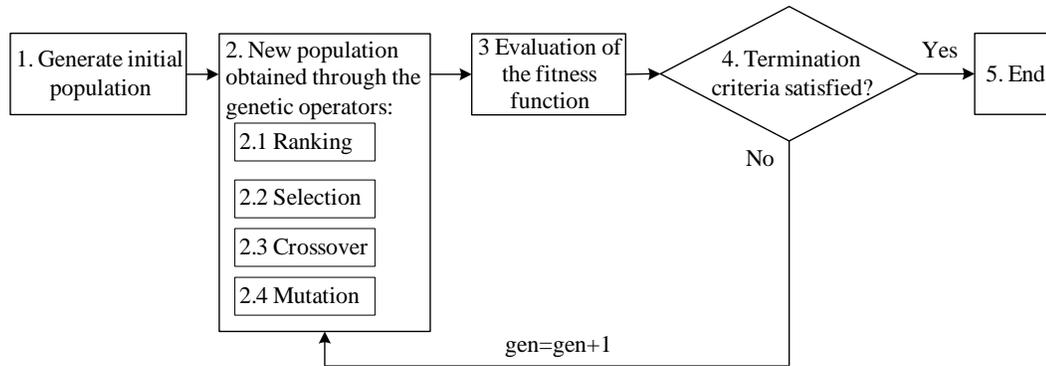


Figure 3.1 Flow chart of the Genetic Algorithm

In this study a simple or canonical genetic algorithm, (Goldberg, 1989; Cao and Wu, 1999), was designed using functions from the *Genetic Algorithm Toolbox*¹, (Chipperfield et al., 1994). A flow chart of the GA is shown in figure 3.1 and the following sections summarise the principal components.

3.1.1 Generate initial population

In keeping with the flow chart shown in figure 3.1, the GA starts with an initial population which describes the possible set of solutions via binary chromosomes. A fixed number of GA parameters has to be assigned at this stage in the code. For the results presented in this thesis the following constants were assumed:

- *Population size* is set to 50. If the population size is small and the problem is complex (non-convex) the GA has less genetic material to work with. Typically, a population is composed of between 30 and 100 individuals, (Zalzala and Fleming, 1997, sec. 1.2.1).
- *Number of variables* depends on the controller to be designed. For the ideal PID controller three variables are tuned, for the 2-DOF PID controller six variables and for the GPC controller eight tuning variables are optimised.
- *Resolution of tuning variables*. In the GA, the resolution for each parameter is described by the number of bits and the upper and lower bounds on that parameter.

¹Published under the GNU General Public License, <http://www.shef.ac.uk/~gaipp/>

The number of bits is defined as:

$$N_{bits} = \log_2 \left(\frac{N_{up} - N_{down}}{R} + 1 \right) \quad (3.1)$$

where N_{up} defines the upper bound on the search area, N_{down} the lower bound, and R is the resolution. For the $T(z^{-1})$ polynomial and the tuning variables of the PID controllers the resolution was set to 0.001, for λ it was set to 0.1 while for N_1 , N_2 and N_u the resolution is unity.

- *Coding.* A binary representation of the chromosomes was used.

3.1.2 Ranking

Individuals are ranked according to their objective fitness values. The best individual is ranked as 2, the worst as 0 and the remaining individuals were linearly ranked between these two limits.

3.1.3 Selection

The individuals chosen for breeding are probabilistically selected based on their fitness or ranking. Fitter individuals have a higher probability and are more likely to be selected. However, less fit individuals may also be chosen. This ensures genetic variety. A common selection strategy known as *Stochastic Universal Sampling* (SUS) was used. To make sure that the best individuals proceed to the next generation, a generation gap, G_{gap} , is set and is defined in equation 3.2.

$$G_{gap} = \frac{N_{pop} - N_{best}}{N_{pop}} \quad (3.2)$$

N_{pop} is the number of individuals in the population and the number of best individuals is denoted as N_{best} . Typically 1 to 5 individuals are passed from the current generation to the next unchanged. In this application, N_{best} was set equal to five.

3.1.4 Crossover

The crossover operator recombines pairs of individuals with a given probability to produce the offspring. The crossover rate was set to 0.7, the default value according to Chip-

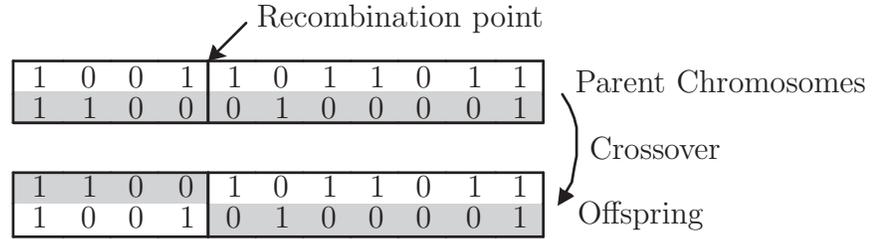


Figure 3.2 Example of single point crossover

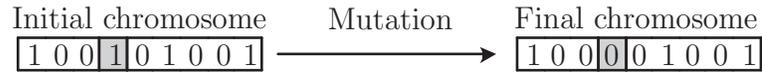


Figure 3.3 Example of mutation

perfield et al. (1994). The simplest recombination strategy is to employ single point crossover as illustrated in figure 3.2.

3.1.5 Mutation

The mutation operator introduces new genetic material by random changes of a single bit. It works on each individual by alternating the value of a randomly selected bit position, see figure 3.3. The mutation probability, P_m , is obtained from equation 3.3

$$P_m = \frac{0.7}{L_{ind}} \tag{3.3}$$

where L_{ind} is the length of the chromosome structure. Typically, the probability for bit mutation is in the range 0.001 to 0.01, see O’Mahony (2002).

3.1.6 Evaluation of the fitness function

The *fitness function* indicates how good a candidate solution is. Descriptions of the chosen functions are detailed in section 3.5.

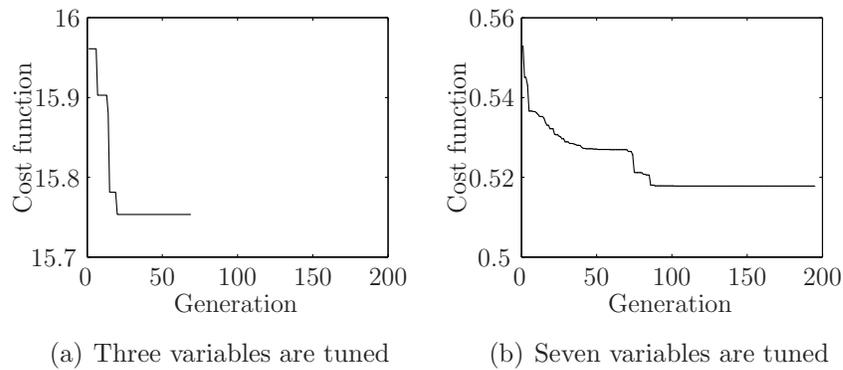


Figure 3.4 Error of the best individuals

3.1.7 Terminating the GA

The GA runs in the loop as long as predefined user criteria are not achieved, see the flow chart 3.1. The algorithm may be terminated if:

- a predefined maximum number of the generations has been reached
- the algorithm has converged, e.g. the cost function over a specified number of generations has remain unchanged
- the GA has reached a prespecified minimum value of the cost function.

A combination of the first two criteria was used and the maximum number of iterations was set to 200. As shown in figure 3.4(a), if the algorithm converged quickly then the GA terminated prior to the predesigned value. In this case, the GA terminated at the 69th generation. In most cases, the maximum number of generations was not achieved. When the T polynomial was included for tuning, the problem became more complex and a larger number of generations was required. An example of this is shown in figure 3.4(b).

3.1.8 Cancellation of the output horizon

The genetic algorithm does not have information about the problem which it is going to solve. The results are based on a continuous evaluation of the objective function. It is necessary to assume that the output predictive horizon is greater than the initial

predictive horizon, $N_1 \leq N_2$. The GA works on a probabilistic method and hence there is a possibility that $N_1 > N_2$, which is forbidden by the GPC control law. A simple “if” statement was implemented in the GA to prevent this from occurring.

```

if N2<N1
    N2=N1+N2-1
end

```

3.2 Time domain performance criteria

This section defines typical performance criteria that characterise the servo plus regulatory responses. The servo response refers to the closed-loop response, $y(t)$, to an applied unity set-point input, $r(t)$, see figure 3.5. Based on this figure the following criteria are considered.

Rise time, T_r , is either defined as the inverse of the largest slope of the step response or the time taken for the response to go from 10% to 90% of its steady state value, y_{ss} .

Settling time, T_s , is the the time taken for a unit step response to reach and remain within an absolute band of 0.02 units.

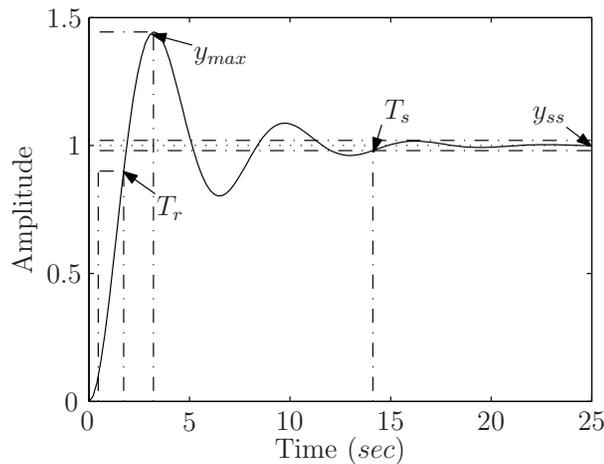


Figure 3.5 Typical response, $y(t)$, to the step input $r(t)$

Overshoot, y_p , is the ratio of the difference between the first peak and the steady state value of the step response.

$$y_p = \frac{y_{max} - y_{ss}}{y_{ss}} \quad (3.4)$$

In industrial control applications it is common to specify an overshoot of less than 10%. In many situations it is desirable, however, to have an overdamped response with no overshoot, (Åström and Hägglund, 1995).

Integrated Error, IE, is defined by equation 3.5

$$IE = \sum_{k=0}^{t_1} e(k) \quad (3.5)$$

where $e(k)$ is the control error and t_1 is the simulation end time (steady state error equals zero). The IE criterion is a natural choice for control of quality variables for a process which is characterised by an overdamped response. This criterion can be misleading however since, for example, it will be zero for an oscillatory system with no damping.

Integrated Absolute Error, IAE, is another very common performance criterion. The IAE of the servo response is given by equation 3.6

$$IAE_{servo} = \sum_{k=0}^{t_1} |e(k)| \quad (3.6)$$

Regulator response is defined as the closed-loop response, $Y(s)$, to a disturbance input, $D(s)$, as illustrated by the block diagram in figure 2.1. A typical response is presented in figure 3.6. In this case the following criteria, presented by Åström (1987), are considered:

Overshoot after disturbance, y_d , is defined as $y_d = -\frac{e_2}{e_1}$, where the peaks e_1 and e_2 are presented in figure 3.6.

Disturbance settling time, T_{rd} , is the time taken for a unit step disturbance response to reach and remain within an absolute band of 0.02 units.

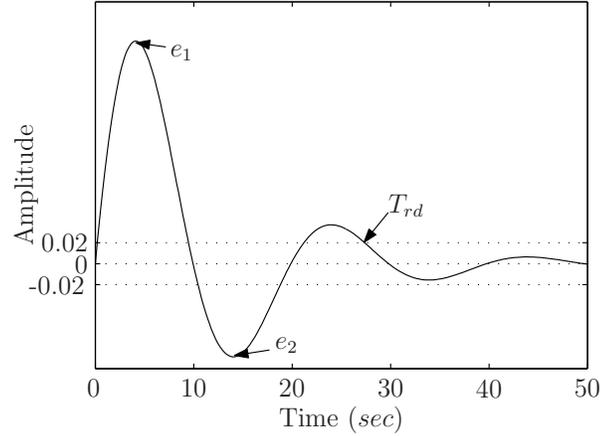


Figure 3.6 Typical response of control error to step or impulse disturbance

Integrated Absolute Error, IAE, of the regulator response is given by equation 3.7

$$IAE_{reg} = \sum_{k=0}^{t_1} |-y(k)| \quad (3.7)$$

3.3 Metrics to measure robustness

In practice, models are not perfect and a discrepancy will always exist between the identified model and the actual plant. For the controller, which is designed using the identified model, to be successful on the actual plant, the controller must be robust to process–model mismatch. Standard measures of robustness are presented in the following sections.

3.3.1 Gain and phase margin specification

Gain margin, A_m , is defined as the factor by which the gain must change to force the closed–loop system to marginal stability, (Phillips and Nagle, 1990; Åström and Hägglund, 1995).

$$A_m = \frac{1}{|L(i\omega_u)|} \quad (3.8)$$

where the ultimate frequency, ω_u , is the frequency at which $L(i\omega_u) = -\pi$. L is defined as an open–loop transfer function described by equation 3.9.

$$L(s) = C(s)G(s) \quad (3.9)$$

Phase margin, ϕ_m , is defined as the angle through which the Nyquist diagram must be rotated so that the diagram intersects the $(-1, i0)$ point, (Phillips and Nagle, 1990; Åström and Hägglund, 1995).

$$\phi_m = \pi + \arg(L(i\omega_c)) \quad (3.10)$$

The gain cross-over frequency, ω_c , is the frequency where the Nyquist plot intersects the unit circle $|L(i\omega)| = 1$. Typical values of ϕ_m range from 30° to 60° . The gain and phase margin definitions for the discrete-time system and for the continuous-time system are exactly the same.

3.3.2 Modulus margin, MM

The Modulus Margin, MM , is defined by Landau et al. (1998) as the radius of the circle centred on the critical point $(-1, i0)$ and tangent to the Nyquist plot of the open-loop transfer function, $L(s)$. Recommended practical values for the modulus margin are $MM \geq 0.5(-6dB)$. The inverse of the MM corresponds to the maximum value of the sensitivity function $\mathcal{S}(s)$ on the Bode plot and is denoted by M_s .

$$M_s = \max_{0 \leq \omega \leq \infty} (\mathcal{S}(i\omega)) \quad (3.11)$$

where

$$\mathcal{S}(s) = \frac{1}{1 + C(s)G(s)} \quad (3.12)$$

According to Åström and Hägglund (1995), reasonable values of M_s are in the range 1.3 (robust tuning) to 2.0 (more aggressive tuning).

3.3.3 Resonance peak of the closed-loop system M_p

The value of M_p is the size of the resonance peak of the closed-loop system obtained from a frequency domain analysis of the complementary sensitivity function, \mathcal{T} , detailed in equation 3.14.

$$M_p = \max_{0 \leq \omega \leq \infty} (\mathcal{T}(i\omega)) \quad (3.13)$$

$$\mathcal{T}(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} \quad (3.14)$$

Typical values of M_p are in the range of 1.0–1.5, (Åström et al., 1998).

3.3.4 Delay margin, DM

Delay margin, DM , represents the additional time delay that the closed-loop system can tolerate prior to instability. The phase lag introduced by a pure time delay, T_d , is

$$\angle\phi(\omega) = \omega T_d \quad (3.15)$$

Therefore, to convert the phase margin to a delay margin, i.e. to compute the additional delay which will lead to instability, equation 3.16 is used.

$$DM = \left| \frac{\phi_m}{\omega_c} \right| \quad (3.16)$$

If the Nyquist plot intercepts the unit circle at several frequencies, ω_c^i , characterised by corresponding phase margins of ϕ_{mi} , the delay margin is defined by

$$DM = \min_i \left| \frac{\phi_{mi}}{\omega_c^i} \right| \quad (3.17)$$

For adequate robustness, the delay margin should be greater than one sampling period for discrete-time systems, (Landau et al., 1998, sec. 8.3.1).

3.4 An overview of robust tuning techniques

Regardless of the design technique used, controllers are always designed based on information about the dynamic behaviour of a process. The model accuracy varies but is never perfect. Furthermore, the behaviour of the plant itself changes with time and these changes are rarely captured in the model. It is most desirable that the controller be insensitive to this kind of *model uncertainty*, i.e. the controller should be *robust*. In addition, the design must satisfy some minimum performance specifications, e.g.

- good disturbance rejection,
- adequate set-point following.

Traditionally, PID controllers have largely been designed from a performance point of view, (Åström et al., 1993; O'Dwyer, 2000a,b,c). More recently, the importance of robustness has been realised and robust PID controllers may now be achieved using a myriad of approaches. In keeping with Yaniv and Nagurka (2004), this thesis will collate robust PID controllers into two categories:

- techniques utilising gain and phase margin specifications,
- techniques employing sensitivity functions.

3.4.1 Gain and phase margin specifications

The design of PID controllers to satisfy some gain and phase margin specifications may be achieved via the use of tuning rules, graphical techniques and, recently, techniques based on *Artificial Intelligence* (AI) — numerical methods. Some of the techniques presented in this section overlap. For example: crossover frequencies may be obtained using numerical optimisation and subsequently a tuning rule is applied. In this thesis, these cases are categorised as tuning rule techniques, i.e. the classification is based on the manner in which the controller parameters are obtained. The following discussion is not restricted to the ideal and 2-DOF PID structures that are considered in this thesis but attempts to provide the reader with a general overview of design philosophies that incorporate robustness. Where appropriate, the specific PID algorithm relevant to the design is mentioned.

Techniques based on tuning rules

There are a multitude of PID controller design techniques based on the use of tuning rules and the interested reader is referred to O'Dwyer (2003) for an excellent overview. This section will concentrate only on tuning rules that incorporate gain and phase margin criteria in the design and will provide a brief overview of these techniques. Numerous PID controller designs based on these criteria exist, for example, the proposals by Åström and Hägglund (1984), Franklin et al. (1986), Ogata (1990), Ho et al. (1995) and Cluett and Wang (1997). The disadvantage of some of these methods is that the transfer function of the controlled process is restricted to a *First Order Lag Plus time Delay* (FOLPD) model. As reported in Shafiei and Shenton (1997), some of these methods may be unable to render solutions for all gain and phase margin requirements and sometimes the controllers do not perform satisfactorily. Fung et al. (1998) proposed a design method for the ideal PI controllers which can achieve user-specified gain and phase margins. These authors derive equations that define the boundaries for the ideal

parallel PID algorithm given a specified gain and phase margin. These equations are then plotted to define the $\{K_p K_i K_d\}$ parameter plane corresponding to the required level of stability margins. Points on the boundaries of the regions can be associated with the phase and gain crossover frequencies and hence allow direct consideration of bandwidth requirements. Some guidance on transient performance optimisation is also provided incorporate gain and phase margin (GPM).

Ho et al. (1998b) proposed another GPM method which has the advantage that it can be applied to high order systems with time delay. This paper also presented other tuning formulae for the ideal PID controller that can satisfy both robustness and optimum performance requirements in terms of the *Integral Square Error* (ISE) criterion. Ho et al. (1999) studied the relationship between the performance index ISE, gain margin, phase margin and gave recommendations for improving the phase margin and performance of the ideal PI and PID structures. The authors discussed the classic trade-off problem: phase margin versus performance and demonstrated that for a given gain margin there is a phase margin that minimises the ISE. PID tuning rules to achieve a user specified gain margin for open-loop unstable processes were presented by Ho and Xu (1998). Another technique which achieves a desired gain and phase margin was presented by Wang et al. (1999). This method can yield an exact solution for a general linear process and is based on the utilisation of the frequency response of the process. The proportional gain of the controller is calculated using the critical point information and the desired gain margin. The phase margin is then used to calculate the integral and derivative gains. An advantage of this method is that the closed-loop bandwidth can be specified as a desired factor of the open-loop bandwidth. A quantitative robust stability criterion that is based on the open-loop frequency response was developed by Wang et al. (2002) for the parallel PID controller structure. The authors proposed a tuning algorithm which guarantees the required gain and phase margins of the closed-loop system for a second order plant plus time delay and applied the technique to an uncertain system.

An autotuning procedure for PID controllers in terms of gain and phase margin procedure was presented by Zhuang and Atherton (1993). The least squares fit technique

was used to obtain formulae and based on these formula, the controller gains were calculated for a FOLPD model. The tuning technique is not suitable for systems with a long dead time.

Graphical methods

Relative to tuning rules, robust PID controller design based on graphical techniques has received scant attention. To the author's knowledge, only two significant contributions exist. Shafiei and Shenton (1997) proposed a technique, based on a parallel PI/PD/PID controller formulation, that allows a direct design for simultaneous minimum phase and gain margins as well as transient performance and maximum bandwidth specifications and thus may be regarded as a more powerful alternative to the approach of Åström and Hägglund (1984), Ho et al. (1995). By solving given equations, the user is able to plot a two dimensional parameter plane and, based on that, choose the desired controller parameters. The technique is based on previous work, Shafiei and Shenton (1994).

The method introduced by Huang and Wang (2001) is based on searching the parameter area for the controller gains to achieve a compromise between good tracking performance and system robustness with respect to external disturbances. This method can be applied to stable and unstable systems of arbitrary order and the controller design has considerable flexibility. One of the advantages is that exact gain and phase margin specifications may be achieved. Again, the authors assume that the PID controller can be described by the ideal parallel or non-interacting algorithm.

Numerical methods

An improvement of the GPM proposed by Ho et al. (1995) was achieved via a fuzzy neural network approach presented by Chu and Teng (1999). According to the authors, the proposed *Fuzzy Neural controller based on Gain and Phase margin specifications* (FNGP) gives better performance than the GPM even when the plant is first-order with time delay. A fuzzy neural approach, proposed by Lee and Teng (2002), is based on *Adaptive Network using Fuzzy Interference System* (ANFIS). Advantages of this neural network design approach are that the trained neural network automatically tunes

the parameters of an ideal parallel PID controller for different gain and phase margin specifications so that neither numerical methods nor graphical methods are required. An alternative method to achieve specified gain and phase margins using AI methods, based on the *Direct Nyquist Array* (DNA) was presented by Ho et al. (1998a).

The tuning formulation of Ho et al. (1995) was applied to a parallel structure of fuzzy PID control systems by Xu et al. (2000). A comparison with the ideal PID controller was presented via a second order nonlinear plant with time delay — two tanks coupled by an orifice. The authors demonstrated, via theoretical and experimental results, that the fuzzy PID controller has the nonlinear properties of higher controller gains when the system is away from its steady state and a lower control profile when set–point changes occur. As a result, these nonlinear properties provide the fuzzy PID controller with the ability to outperform the conventional PID control system.

Liu and Daley (1999b) proposed an alternative technique for satisfying gain and phase margin specifications for the ideal PID controller structure. These authors considered the controller tuning as an optimisation problem designed to satisfy a set of frequency–domain performance requirements: gain margin, phase margin, crossover frequency and steady–state error and solved the problem using min–max optimisation (Gill et al., 1981) and the method of inequalities, (Zakian and Al-Naib, 1973). The proposed design was demonstrated via an application to a rotary hydraulic system. Its performance was compared with six alternative PID tuning rules — none of which included gain and phase margin criteria — and significantly improved dynamic performance was achieved.

Since the early 1990’s, Genetic Algorithms have been used to determine the PID controller parameters. Initially, simple cost functions, e.g. integral of squared error, were specified (Porter and Jones, 1992; Hwang and Thompson, 1993; Porter and Hicks, 1993; Wang and Kwok, 1994; Porter and Hicks, 1995; Ota and Omatu, 1996; Cheng and Hwang, 1998; Rensburg et al., 1998; Mitsukura et al., 1999; O’Mahony et al., 2000). More recently the versatility of the GA approach has been exploited and more complex cost functions incorporating robustness constraints have been proposed. However, the number of GA–based designs for PID controllers using gain and phase margins is

still quite limited. One such gain and phase margin optimisation problem was proposed by Jones et al. (1996). A frequency-domain model of the process was generated by identifying several points below, at, and above the crossover frequency. This model provided estimates of the actual gain and phase margins, denoted GA_m and $G\phi_m$, respectively. The following cost function was specified $(\mathcal{G}A_m - \mathcal{D}A_m)^2 + (\mathcal{G}\phi_m - \mathcal{D}\phi_m)^2$ where $\mathcal{D}A_m = 2$ and $\mathcal{D}\phi_m = 45^\circ$ represent the desired gain and phase margins respectively. This cost function was optimised using a simple binary coded genetic algorithm with a population size of 100 individuals and 1000 generations. The proposal was demonstrated on a laboratory-scale airflow temperature control system.

Mitsukura et al. (1997) proposed a somewhat convoluted approach for tuning the ideal velocity-based discrete PID controller using genetic algorithms. In their approach the authors' equate the discrete PID algorithm with the generalised minimum variance control law (Clarke and Gawthrop, 1979) assuming a second-order discrete-time process model. They then derive abstract tuning parameters; λ which is related to stability and, by extension, the gain and phase margin, σ which is related to the rise-time and δ a parameter related to the damping factor. These parameters are specified by the user and the PID coefficients are optimised to achieve the specification. The design procedure is closely related to the traditional pole-placement philosophy but suffers from a lack of intuitiveness.

3.4.2 Techniques based on sensitivity functions

Though the use of sensitivity functions for controller design is topical, the idea is not new. The concept was popularised in the PID literature by Shinskey (1990) who outlined a series of controller evaluation tests based on optimal load disturbance rejection and measured by sensitivity constraints. This idea was developed by Persson (1992), Persson and Åström (1992), who applied a constraint to M_s . The use of both M_s and M_p as design parameters was suggested by Schei (1994).

Modern techniques are extensions of these ideas where robustness is specified as a sensitivity constraint and performance optimised through the integral of error criterion. The combined optimisation problem is non trivial and numerical approaches are gener-

ally applied to realise the PI/PID controller gains. A few graphical solutions have also been suggested.

Numerical methods

A novel and effective PI controller tuning technique based on non-convex optimisation was reported by Åström et al. (1998). The authors described a numerical method based on the optimisation of load disturbance rejection with constraints on the sensitivity function and weighting of the set-point response. The paper also demonstrated that the design problem of minimising the IE of the load disturbance is equivalent to maximising the integral gain, K_i . The tuning was restricted to the PI algorithm of equation 3.18

$$U(s) = K_p (b \cdot R(s) - Y(s)) + \frac{K_i}{s} (R(s) - Y(s)) \quad (3.18)$$

where b is a controller parameter. An advantage of this numerical method is that it is applicable to a wide range of systems and robustness is easily specified through the criteria M_s and M_p .

Another design approach to solve the non-convex optimisation problem of Åström et al. (1998) was presented by Hwang and Hsiao (2002). The approach is based on regarding the equality constraint set on the controller gain parameters as a two-dimensional value set in the complex plane and uses the notion of principal points to characterise its boundary. The method can handle sensitivity and complementary sensitivity constraints simultaneously without using an iterative procedure. As distinct from the method presented by Åström et al. (1998) the authors considered the ideal PI and the ideal PID controller. The method leads to a less conservative design approach than that proposed by Åström et al. (1998) in which both constraints are combined into a single and more conservative constraint.

In Wang and Shao (2000), robustness was achieved by the parameter λ

$$\frac{1}{\lambda} = \max_{0 \leq \omega < \infty} |\operatorname{Re}[L(j\omega)]| \quad (3.19)$$

which specified the shape of the Nyquist curve. The loop transfer function is defined by equation 3.9 and $C(s)$ is the ideal parallel PI controller, equation 3.20.

$$U(s) = E(s) \left\{ K_p + \frac{K_i}{s} \right\} \quad (3.20)$$

In addition to the robustness parameter λ , the design minimised the integrated control error to obtain good performance. These authors gave a number of simulation examples for the PI controller which achieved high performance over a wide range of linear self-regulating processes.

The combined use of genetic/evolutionary algorithms and sensitivity functions began to appear from the late 1990's onwards. For example, Poulin and Pomerleau (1997) presented a tuning method for PID controllers based on the contours of the Nichols chart and M_p . The approach offers the possibility of simultaneously handling (1) the maximum peak overshoot, (2) a lower bound on the amplitude and phase margins, such that minimum stability margins are guaranteed, and (3) approximate specifications on the closed-loop bandwidth. The method can be applied to almost all types of processes. Subsequently, Poulin and Pomerleau (1999) developed a PI design methodology for integrating processes that bounds the maximum peak resonance of the closed-loop transfer function. The peak resonance constraint is equivalent to bounding the complementary sensitivity which can be converted to bounding the sensitivity function.

Similarly Lieslehto (2001) chose to incorporate multiple criteria, closed-loop bandwidth, resonance peak M_p , maximum sensitivity M_s as well as the traditional gain and phase margin criteria into his GA formulation. While multiple criteria are considered the cost function reduces to a single-objective function by considering only the worst-case offender for each candidate solution. This GA optimisation was performed on a two-degree of freedom PID algorithm with set-point weighting and first-order filter and demonstrated via a simulation example.

In contrast, Shen (2001) chose to use the dominant pole assignment method proposed by Åström and Hägglund (1995) as the basis for his GA-based tuning approach. In the dominant pole assignment method the dominant poles of the system are assigned such that the integration of the error caused by a step load disturbance is minimised subject to a constraint on the maximum sensitivity. Set-point weighing is then used to optimise the set-point response. Shen (2001) chose to combine these steps into a single optimisation function $J = J_s + J_d$, where J_s is the IAE due to a set-point change and J_d is the IAE due to a disturbance change. This function is minimised subject to the constraints of

(i) stability and (ii) $1.4 \leq M_s \leq 2.0$. A two degree of freedom PID structure (an ideal PID controller with set-point weighting) was utilised and a simulation example is used to demonstrate the improved performance over the tuning rules proposed by Åström and Hägglund.

Robust designs are frequently based on the H_∞ norm of some suitably chosen sensitivity function. Chen et al. (1995) chose to tune the PID coefficients by solving a mixed H_2/H_∞ optimisation problem via genetic algorithms. The authors proposed to solve the problem in three stages, effectively reducing the multi-objective problem to a single-objective problem. In the first stage, the Routh–Hurwitz criterion is applied to determine the stability domain for the three PID parameters. Secondly, a subset of the stability domain in the PID parameter space from step one is specified so that the H_∞ constraint (robustness constraint or disturbance attenuation constraint) is satisfied. In the third step, the design problem becomes, in the subset domain of the H_∞ constraint of step 2, optimise the H_2 tracking performance. The authors focused on the ideal non-interacting PID algorithm and a more efficient algorithm was subsequently proposed by Chen and Cheng (1998). The performance of this mixed H_2/H_∞ formulation was evaluated by Lagunas et al. (2003) on a pilot-scale DC servomotor. A similar H_2/H_∞ design philosophy was adopted by de Moura Oliveira and Jones (2000) where the H_2 tracking performance was optimised subject to a H_∞ norm on the sensitivity function i.e. the ISE tracking was optimised subject to a constraint on M_s . The primary contribution of this research was to present a modified GA which preserves genetic diversity. The authors demonstrate the effectiveness of the proposal via two simulation examples; one SISO and one MIMO.

True multi-objective GA approaches to the mixed H_2/H_∞ PID optimisation problem have been proposed by Takahashi et al. (1997), Kawabe and Tagami (1997), Krohling et al. (1997), Krohling and Rey (2001) Herreros et al. (2002), amongst others. For example, Takahashi et al. (1997) consider a cost function that consists of four objectives: optimisation of (1) the H_2 and (2) the H_∞ norm from reference to output transfer function; (3) the H_2 and (4) the H_∞ norm from the disturbance to output transfer function. In addition these authors consider two constraints (i) a maximum constraint on the absolute

values of the closed-loop poles motivated by the recognition that optimal controllers may realise very slow responses; (ii) maximum values on the controller gains. In addition it was assumed that the process structure was assumed known but with uncertain parameters e.g. the uncertainty was defined as a convex polytope. To solve this problem the min-max formulation was adopted e.g. the cost function was minimised for the worst-case model uncertainty. The paper focused on deriving a new gradient-based optimisation algorithm to solve the problem. A multi-objective GA using pareto-optimal solutions is then used to validate the results of the approximate gradient technique. The authors note that the gradient-based algorithm was considerable more efficient — execution time was between 30 to 450 times faster. Herreros et al. (2000, 2002) proposed a generic multi-objective mixed H_2/H_∞ cost function which could incorporate set-point following, rejection of load disturbances, robustness with respect to model uncertainties, attenuation of measurement noise and actuator limitations. The authors also consider a stability constraint and assume a generic PID controller structure. Like Takahashi et al. (1997) the possibly competitive nature of the multiple objectives is negotiated via the use of pareto-optimal solutions. A relatively advanced genetic algorithm is proposed and the authors focus on comparing their multiobjective robust control design algorithm with alternative multiobjective GA proposals and with formal *Linear Matrix Inequality* (LMI) based approaches.

Similarly, Kawabe and Tagami (1997) assume that the process contains parametric uncertainty and can be modelled as a polytopic system. The proposal focuses on designing an optimal 2-DOF PID controller to minimise the H_2 norm of (i) the regulator and (ii) the servo performance. Rather than considering pareto-optimal solutions as in Takahashi et al. (1997) this proposal reformulates the problem in terms of Linear Matrix Inequalities and subsequently applies a GA to solve this set of inequalities. The paper concludes with some simulation examples.

Krohling et al. (1997) as well as Krohling and Rey (2001) consider the same problem formulation as Chen et al. (1995), i.e. they optimise the H_2 tracking performance subject to constraints on stability and a H_∞ norm that encapsulates the disturbance rejection capability. Both norms are optimised by separate genetic algorithms with periodic

information exchange to ensure the “true” optimum is achieved. However, since the cost essentially consists of a single objective with a constraint it is not clear why two GA are required and the advantages of the proposal, if any, are not detailed. A design example, utilising the ideal parallel PID structure, completes the paper.

3.4.3 Resume of tuning techniques

It is clear from the previous sections that robust controller design laws may be realised through the use of gain and phase margin criteria and/or by constraining the open or closed-loop sensitivity functions in some fashion. It is evident from this review that the majority of the techniques outlined in section 3.4.1 are unsuitable for the purposes of this research because the design cannot be readily applied to the Generalised Predictive Controller, i.e. a tuning rule for PI/PID controller will not realise the GPC parameters. The possible exceptions are the designs proposed by Liu and Daley (1999b) and Jones et al. (1996). In the former, the authors, Liu & Daley, apply numerical optimisation (min–max optimisation) to determine optimum controller parameters and a similar procedure will be applied in this study but the numerical optimisation will be performed by a GA and the cost function will minimise and IAE performance criterion subject to constraints on the gain and phase margin. In contrast, Jones et al. (1996) only considered optimising the gain and phase margins. It is the author’s experience that incorporating a transient performance specification is useful as it helps to direct the GA and results in quicker convergence. In addition, formulating the problem to achieve a single desired value of gain and phase margin is too restrictive and may potentially be mutually exclusive. A more realistic objective is to specify a region within which the gain and phase is acceptable as is advocated in this thesis and detailed presently.

The majority of the techniques discussed in section 3.4.2 can be applied to both the PID controller and the GPC since, in general, a numerical optimisation is performed. The problem is then to determine the most appropriate cost function, i.e. what combination of constraints, sensitivity functions, and integral of error criteria will yield the most suitable/practical controller design?

The following section describes the proposed tuning strategies based on the gain and

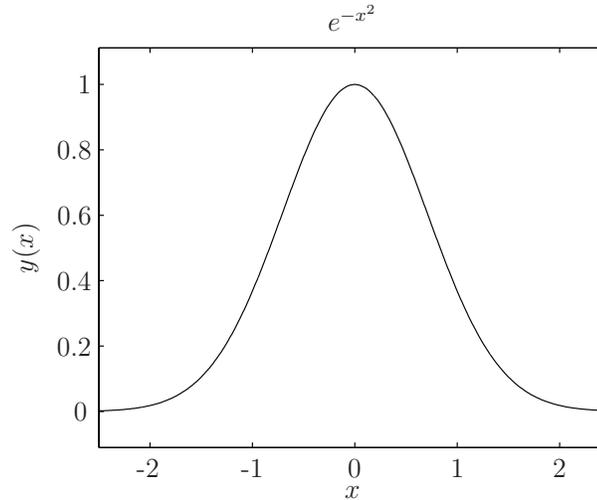


Figure 3.7 Gauss curve

phase margin, 3.5.1, the modulus margin, 3.5.2, and the input sensitivity function, 3.5.3.

3.5 Objective functions

As mentioned in section 3.4.1, the first of the proposed designs involves the use of gain and phase margin specifications. This section describes how this design was achieved by first considering typical requirements for both the gain and the phase margin. For example, Seborg et al. (1989) specified that typical minimum requirements are a gain margin of $A_m > 4.6dB$ and a phase margin of $\phi_m > 30^\circ$. The authors Ho et al. (1995), Fung et al. (1998), Wang et al. (1999) assumed a $A_m = 9.5dB$ and a $\phi_m = 60^\circ$ in their controller design while Åström and Hägglund (1995, p. 126) recommend that the gain margin should be in the range $6dB - 14dB$ and that the phase margin should be in the range $30^\circ - 60^\circ$. In this thesis, the minimum gain margin was chosen as $\underline{A_m} = 6dB$ and the phase margin $\underline{\phi_m} = 45^\circ$.

3.5.1 Gain and phase margin specifications

The objective function in the controller design was then to minimise the IAE subject to the constraints that the gain margin was greater than $6dB$ and the phase margin was greater than 45° . These constraints were implemented as follows. First consider the

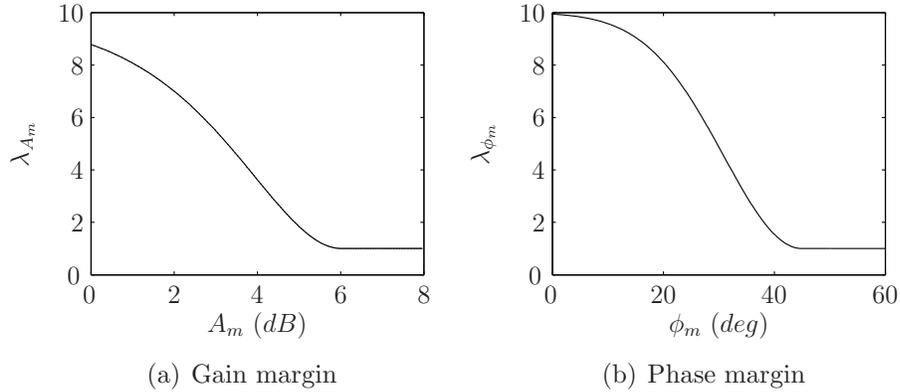


Figure 3.8 Penalty function for the GPM method

standard Gaussian function

$$y = e^{-x^2} \quad (3.21)$$

which is illustrated in figure 3.7. If this function is modified by appropriate truncation and scaling, it is possible to generate curves such as those illustrated in figure 3.8. Consider figure 3.8(a): if the gain margin is less than $6dB$ then the contribution by this function to the overall cost is large. Hence, the cost function will be heavily weighted, regardless of the IAE, while if the gain margin is greater than $6dB$ the contribution is constant at unity. Mathematically, these curves are represented by equations 3.22 and 3.23.

$$\lambda_{A_m} = \begin{cases} 10 - 9e^{-2(\underline{A}_m - A_m)^2} & \forall A_m < \underline{A}_m \\ 1 & \forall A_m \geq \underline{A}_m \end{cases} \quad (3.22)$$

$$\lambda_{\phi_m} = \begin{cases} 10 - 9e^{-0.0025(\underline{\phi}_m - \phi_m)^2} & \forall \phi_m < \underline{\phi}_m \\ 1 & \forall \phi_m \geq \underline{\phi}_m \end{cases} \quad (3.23)$$

where A_m is the measured gain margin, \underline{A}_m is the specified minimum value, both expressed as absolute values, ϕ_m is the measured phase margin (in degrees) and $\underline{\phi}_m$ is the specified minimum value.

Clearly, provided the robustness constraints are satisfied, these objective functions reduce to minimising the IAE. The individual objective functions are defined by

$$J_{PID} = \min_{K_p, K_i, K_d} \{IAE_{both} \cdot \lambda_{A_m} \cdot \lambda_{\phi_m}\} \quad (3.24)$$

$$J_{2-DOF} = \min_{K_p, K_i, K_d, b, c, N} \{IAE_{both} \cdot \lambda_{A_m} \cdot \lambda_{\phi_m}\} \quad (3.25)$$

$$J_{GPC} = \min_{N_1, N_2, N_u, \lambda, T(z^{-1})} \{IAE_{both} \cdot \lambda_{Am} \cdot \lambda_{\phi m}\} \quad (3.26)$$

where the IAE of the servo and regulatory response is given by equation 3.27

$$IAE_{both} = \sum_{k=0}^{t_0-1} |e(k)| + \sum_{k=t_0}^{t_1} |e(k)| \quad (3.27)$$

t_0 is the time at which a unit step disturbance, $D(s)$, is applied (see figure 2.1) and t_1 is the simulation end time (steady state error equals zero). Therefore, the IAE servo is calculated over the period $0 < t < t_0$, and the IAE regulator is calculated over the period $t_0 \leq t \leq t_1$.

3.5.2 Modulus margin

A variety of sensitivity functions have been proposed to aid the design of robust control systems. Landau et al. (1998) reported that a good robustness measure is the modulus margin, MM , described in section 3.3.2. These authors recommended practical values for the modulus margin namely $MM \geq 0.5(-6dB)$. Likewise, Åström and Hägglund (1995) suggested that reasonable values of MM are in the range from 0.77 ($M_s = 1.3$, robust tuning) to 0.5 ($M_s = 2.0$, aggressive tuning). Based on these recommendations, the value $\underline{MM} = 0.6$ ($\overline{M_s} = 1.667$) was selected as a desired minimum value.

As with the gain and phase margin, the mathematical description of the penalty function for the MM is defined by equation 3.28.

$$\lambda_{MM} = \begin{cases} 10 - 9e^{-20(\underline{MM} - MM)^2} & \forall \quad MM < \underline{MM} \\ 1 & \forall \quad MM \geq \underline{MM} \end{cases} \quad (3.28)$$

and hence suitable objective functions are equations 3.29, 3.30 and 3.31.

$$J_{PID} = \min_{K_p, K_i, K_d} \{IAE_{both} \cdot \lambda_{MM}\} \quad (3.29)$$

$$J_{2-DOF} = \min_{K_p, K_i, K_d, b, c, N} \{IAE_{both} \cdot \lambda_{MM}\} \quad (3.30)$$

$$J_{GPC} = \min_{N_1, N_2, N_u, \lambda, T(z^{-1})} \{IAE_{both} \cdot \lambda_{MM}\} \quad (3.31)$$

where IAE_{both} is defined by equation 3.27.

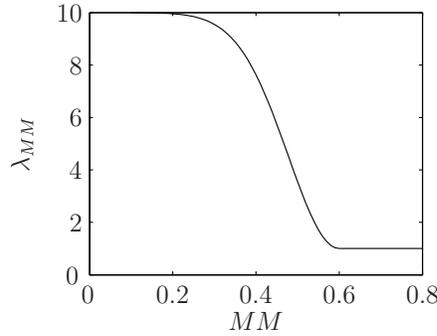


Figure 3.9 Penalty function for the modulus margin specification

3.5.3 Input sensitivity function

In figure 2.1, the closed-loop transfer function between the output, $Y(s)$, and the output disturbance, $U_d(s)$, can be used as a measurement of the disturbance rejection qualities of the system. This function is called the *input sensitivity function*, $\mathcal{U}(s)$, and is defined by equation 3.32

$$\mathcal{U}(s) = \frac{C(s)}{1 + C(s)G(s)} \quad (3.32)$$

The maximum value of this sensitivity function, M_u , is defined by equation 3.33

$$M_u = \max_{0 \leq \omega \leq \infty} \left\{ \frac{C(i\omega)}{1 + C(i\omega)G(i\omega)} \right\} \quad (3.33)$$

Ideally, for good disturbance rejection performance, M_u , must be close to one in the range $(0 \leq \omega < \infty)$, (Normey-Rico and Camacho, 1999; O'Mahony and Downing, 2000c). Using the M_u , another robust design method is presented. The objective functions are:

$$J_{2-DOF} = \min_{K_p, K_i, K_d, b, c, N} \{IAE_{both} \cdot \lambda_{M_u}\} \quad (3.34)$$

$$J_{GPC} = \min_{N_1, N_2, N_u, \lambda, T(z^{-1})} \{IAE_{both} \cdot \lambda_{M_u}\} \quad (3.35)$$

where λ_{M_u} is defined by equation 3.36 and graphically represented in figure 3.10.

$$\lambda_{M_u} = \begin{cases} 10 - 9e^{-0.0333(\overline{M}_u - M_u)^2} & \forall M_u > \overline{M}_u \\ 1 & \forall M_u \leq \overline{M}_u \end{cases} \quad (3.36)$$

The GA design procedure using objective functions based on

- the minimum gain and phase margin

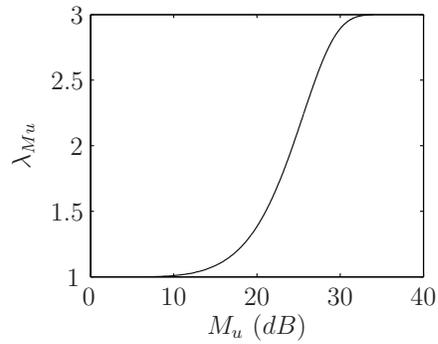


Figure 3.10 Penalty function for the input sensitivity function

- the modulus margin

are evaluated in simulation in chapter 4. However, the design based on the input sensitivity function is not. The primary reason for this is that the designs using A_m/ϕ_m and MM did not perform as well as anticipated in real-time (chapter 5) and this motivated the use of the input sensitivity function. However, time did not permit the author evaluate this technique using the simulation analysis of chapter 4.

Chapter 4

Simulations

4.1 Models

The controllers were optimised using Genetic Algorithms based on the GPM and *MM* objective functions described in chapter 3. This chapter presents some details of the optimisation environment and proposes a novel solution to reduce the computation time associated with the GA optimisation. Finally, both of the tuning strategies are applied to the 13 benchmark models, the results are presented and discussed.

The systems which are discussed in this section were proposed in a paper by Åström et al. (1998) and in a workshop by Åström and Hägglund (2000). Models from both publications will be considered in this thesis and are summarised in table 4.1. The first six models are representative of standard process control systems. All of the models describe linear processes and are specified by a transfer function, $G(s)$, which is analytical with finite poles and, possibly, an essential singularity at infinity. This description covers finite dimensional systems with time delay (equations 4.3 and 4.8).

Table 4.1 Benchmark process models used to evaluate proposed controller design procedure

$$G_1(s) = \frac{1}{(s+1)^n} \quad n = 1, 2, 3, 4, 8 \quad (4.1)$$

$$G_2(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)} \quad \alpha = 0.1, 0.2, 0.5, 1.0 \quad (4.2)$$

$$G_3(s) = \frac{1}{(s+1)^3} e^{-15s} \quad (4.3)$$

$$G_4(s) = \frac{1}{s(s+1)^2} \quad (4.4)$$

$$G_5(s) = \frac{1-\alpha s}{(s+1)^3} \quad \alpha = 0.1, 0.2, 0.5, 1.0, 2.0, 5.0 \quad (4.5)$$

$$G_6(s) = \frac{9}{(s+1)(s^2+\alpha s+9)} \quad \alpha = 0.2, 0.5, 1.0, 2.0 \quad (4.6)$$

$$G_7(s) = e^{-s} \quad (4.7)$$

$$G_8(s) = \frac{e^{-s}}{s} \quad (4.8)$$

$$G_9(s) = \frac{1}{10s+1} \quad (4.9)$$

$$G_{10}(s) = \frac{100}{(s+10)^2} \left(\frac{1}{s+1} + \frac{0.5}{s+0.05} \right) \quad (4.10)$$

$$G_{11}(s) = \frac{150}{(s+10)^2(s+1)} \quad (4.11)$$

$$G_{12}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)} \quad (4.12)$$

$$G_{13}(s) = \frac{1}{s^2-1} \quad (4.13)$$

Systems such as $G_1(s)$ are common in the process industry and for large values of n the system behaves like a system with long time delay. Åström et al. (1998) used the value $n = 3$, hence the function described by equation 4.1 will be also considered in this thesis.

$$G_1(s) = \frac{1}{(s+1)^3} \quad (4.14)$$

The system $G_2(s)$ represents a process that has a dominant pole(s) and high-order dynamics whose separation is determined by the parameter α . For the case where $\alpha = 1$ the system $G_2(s)$ is identical to the system $G_1(s)$, for $n = 4$. These two systems, $G_1(s)$, $G_2(s)$, represent processes which are relatively easy to control. In this thesis the specific case where $\alpha = 0.2$, as presented in equation 4.15, will be considered.

$$G_2(s) = \frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)} \quad (4.15)$$

System $G_3(s)$ is also similar to the model $G_1(s)$, however a time delay of 15(sec) is included. This makes the control problem slightly more difficult. $G_4(s)$ is a relatively simple lag plus integrating system. In contrast, the transfer function $G_5(s)$ contains inverse unstable dynamics (zero in the right half of the s -plane) which can complicate the control problem. With reference to equation 4.5, as α is increased the zero moves towards the origin of the pole-zero map and the difficulty of control increases, (Åström and Hägglund, 2000). To demonstrate this, consider the system $G_5(s)$ in a unity gain negative feedback loop. For $\alpha = 0.1$ the system has a gain margin of 15.8dB, when $\alpha = 2$ the gain margin is equal to 1.16dB and when $\alpha = 5$ the system is unstable, $A_m = -6dB$. The peak response of the transfer function, $G_5(j\omega)$, also increases with increasing α . In this study, the choice $\alpha = 2$ was made, which is presented in equation 4.16.

$$G_5(s) = \frac{1 - 2s}{(s + 1)^3} \quad (4.16)$$

The underdamped system, equation 4.6, was included in Åström et al. (1998) but not presented at the workshop, (Åström and Hägglund, 2000). The system has two complex poles with the relative damping $\zeta = \alpha/6$. When the parameter α is decreased the system becomes more difficult to control. In this thesis, the choice $\alpha = 2$ was made and the transfer function is defined by equation 4.17.

$$G_6(s) = \frac{9}{(s + 1)(s^2 + 2s + 9)} \quad (4.17)$$

A time-delay of one second was chosen for systems $G_7(s)$ and $G_8(s)$ yielding the models:

$$G_7(s) = e^{-s} \quad (4.18)$$

$$G_8(s) = \frac{e^{-s}}{s} \quad (4.19)$$

The model $G_9(s)$ was not included by Åström et al. (1998) or Åström and Hägglund (2000) but included as it is representative of some simple well-defined problems such as a liquid level control problem. Closed-loop control of such systems is easy.

The process described by equation 4.10 contains both slow and fast modes. This system has two fast modes with time constants 0.1(sec), one mode with a time constant of 1(sec) and a slow mode with a time constant of 20(sec). The system has a static

gain of 10 and the step response is dominated by the slow time constants, but it is the faster modes that are critical for the closed-loop system, (Åström et al., 1998). The transfer function equation 4.11 is similar, however the slowest term was removed which simplifies the control problem somewhat. System $G_{12}(s)$ is also characterised by diverse dynamics, incorporating an integrator, a pair of slow poles at $s = -1$, a fast pole at $s = -36$ and a pair of zeros at $s = -6$. The final system, $G_{13}(s)$, is open-loop unstable with poles at $s = 1$. The system is representative of classical control problems, e.g. inverted pendulum and would be characterised as a system that is “difficult” to control.

4.2 Choosing the sampling period

The models given by equations of table 4.1 can be used to evaluate the PID and 2-DOF PID controllers as both of these structures are defined in continuous time. While a continuous-time GPC formulation is also possible, and reputedly offers many advantages (Demirioglu and Gawthrop, 1991; Demircioglu and Karasu, 2000), this thesis will consider the original, and more common, discrete-time algorithm. Clearly then, to design the controller a sampled version of the process transfer functions is required. Each discrete-time model was obtained using the standard Z-transform, assuming the presence of a zero-order-hold circuit at the input. In section 2.5.1 some general comments were made regarding the choice of sampling period and some rules were presented to help guide this choice. Many of these rules are formulated in terms of the “desired” or “specified” closed-loop dynamics. Given the lack of specifications, and the fact that the desired response is not quantifiable, an indicative closed-loop response was obtained by applying negative feedback with a unity gain controller. The exceptions are systems $G_7(s)$ and $G_{13}(s)$. Since a proportional controller in a negative feedback loop does not stabilise the open-loop unstable transfer function, $G_{13}(s)$, a stabilising PD controller was designed using root-locus techniques. The controller transfer function is $C(s) = 3.3(s + 1.14)$. A unity gain negative feedback loop also fails to stabilise system $G_7(s)$ and therefore the proportional gain was reduced (arbitrarily) to a value of 0.1. In hindsight, a better choice might have been to use the closed-loop responses obtained from the application of the PID and/or 2-DOF PID controllers, but this was

not considered at the time.

System	Rule	Sampling period (sec)	
		Min	Max
$G_1(s)$	OL Settling time (eqn. 2.42)	0.752	
	CL Settling time (eqn. 2.42)	0.84	
	OL Rise time (eqn. 2.43)	0.422	1.055
	CL Rise time (eqn. 2.43)	0.187	0.467
	Damped frequency (eqn. 2.44)	0.0967	0.29
	Bandwidth (eqn. 2.45)	0.18	0.539
	Natural frequency (eqn. 2.47)	0.22	0.68
$G_2(s)$	OL Settling time (eqn. 2.42)	0.418	
	CL Settling time (eqn. 2.42)	0.228	
	OL Rise time (eqn. 2.43)	0.22	0.6
	CL Rise time (eqn. 2.43)	0.0933	0.233
	Damped frequency (eqn. 2.44)	0.0629	0.189
	Bandwidth (eqn. 2.45)	0.0909	0.273
	Natural frequency (eqn. 2.47)	0.08	0.24
$G_3(s)$	OL Settling time (eqn. 2.42)	2.24	
	CL Settling time (eqn. 2.42)	N/A	
	OL Rise time (eqn. 2.43)	0.422	1.055
	CL Rise time (eqn. 2.43)	0.25	0.625
	Damped frequency (eqn. 2.44)	0.479	1.44
	Bandwidth (eqn. 2.45)	0.2	0.6
	Natural frequency (eqn. 2.47)	1.14	3.43
$G_4(s)$	OL Settling time (eqn. 2.42)	1	
	CL Settling time (eqn. 2.42)	3.09	
	OL Rise time (eqn. 2.43)	0.345	0.863
	CL Rise time (eqn. 2.43)	0.174	0.435
	Damped frequency (eqn. 2.44)	0.112	0.337
	Bandwidth (eqn. 2.45)	0.191	0.574

4.2 Choosing the sampling period

	Natural frequency (eqn. 2.47)	0.29	0.88
$G_5(s)$	OL Settling time (eqn. 2.42)		0.849
	CL Settling time (eqn. 2.42)		7.39
	OL Rise time (eqn. 2.43)	0.614	1.5
	CL Rise time (eqn. 2.43)	0.3	0.77
	Damped frequency (eqn. 2.44)	0.101	0.303
	Bandwidth (eqn. 2.45)	0.105	0.314
	Natural frequency (eqn. 2.47)	0.241	0.722
$G_6(s)$	OL Settling time (eqn. 2.42)		0.375
	CL Settling time (eqn. 2.42)		0.683
	OL Rise time (eqn. 2.43)	0.218	0.545
	CL Rise time (eqn. 2.43)	0.0619	0.1548
	Damped frequency (eqn. 2.44)	0.031	0.0916
	Bandwidth (eqn. 2.45)	0.0565	0.169
	Natural frequency (eqn. 2.47)	0.07	0.2
$G_7(s)$	OL Settling time (eqn. 2.42)		0.99
	CL Settling time (eqn. 2.42)		0.22
	OL Rise time (eqn. 2.43)	0	0
	CL Rise time (eqn. 2.43)	0.039	0.154
	Damped frequency (eqn. 2.44)	0.027	0.08
	Bandwidth (eqn. 2.45)	0	N/A
	Natural frequency (eqn. 2.47)	0.051	0.154
$G_8(s)$	OL Settling time (eqn. 2.42)		0.1
	CL Settling time (eqn. 2.42)		0.129
	OL Rise time (eqn. 2.43)		0
	CL Rise time (eqn. 2.43)	0.079	0.199
	Damped frequency (eqn. 2.44)	0.0626	0.188
	Bandwidth (eqn. 2.45)	0.0977	0.293
	Natural frequency (eqn. 2.47)	0.0437	0.146
	OL Settling time (eqn. 2.42)		3.91

4.2 Choosing the sampling period

$G_9(s)$	CL Settling time (eqn. 2.42)	1.96	
	OL Rise time (eqn. 2.43)	2.22	5.55
	CL Rise time (eqn. 2.43)	1.11	2.75
	Damped frequency (eqn. 2.44)	N/A	
	Bandwidth (eqn. 2.45)	1.05	3.15
	Natural frequency (eqn. 2.47)	1	3
$G_{10}(s)$	OL Settling time (eqn. 2.42)	7.65	
	CL Settling time (eqn. 2.42)	1.03	
	OL Rise time (eqn. 2.43)	4.32	10.8
	CL Rise time (eqn. 2.43)	0.41	1.025
	Damped frequency (eqn. 2.44)	N/A	
	Bandwidth (eqn. 2.45)	0.188	0.563
	Natural frequency (eqn. 2.47)	0.7	2.25
$G_{11}(s)$	OL Settling time (eqn. 2.42)	0.412	
	CL Settling time (eqn. 2.42)	0.112	
	OL Rise time (eqn. 2.43)	0.223	0.557
	CL Rise time (eqn. 2.43)	0.067	0.167
	Damped frequency (eqn. 2.44)	0.04	0.12
	Bandwidth (eqn. 2.45)	0.0654	0.196
	Natural frequency (eqn. 2.47)	0.046	0.139
$G_{12}(s)$	OL Settling time (eqn. 2.42)	0.1	
	CL Settling time (eqn. 2.42)	1.78	
	OL Rise time (eqn. 2.43)	0.331	0.827
	CL Rise time (eqn. 2.43)	0.182	0.455
	Damped frequency (eqn. 2.44)	0.11	0.329
	Bandwidth (eqn. 2.45)	0.19	0.57
	Natural frequency (eqn. 2.47)	0.252	0.757
	OL Settling time (eqn. 2.42)	N/A	
	CL Settling time (eqn. 2.42)	0.214	
	OL Rise time (eqn. 2.43)	N/A	

$G_{13}(s)$	CL Rise time (eqn. 2.43)	0.07	0.21
	Damped frequency (eqn. 2.44)	0.422	1.26
	Bandwidth (eqn. 2.45)	0.07	0.22
	Natural frequency (eqn. 2.47)	0.12	0.36

Table 4.2 Application of sampling rules to model transfer functions

Table 4.2 summarises the application of these rules to the list of process models under consideration. Some rules, such as the Bandwidth method (eqn. 2.45) provide an upper and lower bound and these are listed as max and min respectively in table 4.2 while others, such as the settling time method yield a single nominal value. In table 4.2 the settling time and rise-time were calculated from the MATLAB step-response plots while ω_b , ω_n and ζ were determined using the functions `damp` and `bandwidth`.

Prior to discussing the results presented in table 4.2 a few brief explanatory remarks are appropriate. System $G_3(s)$, when controlled using a unity feedback controller, results in an inordinately long settling time ($\sim 1600sec$) and that accounts for the n/a entry for the closed-loop settling time method. System $G_7(s)$ has, in theory, an infinite bandwidth and this accounts for the n/a entry for equation 2.45. System $G_{13}(s)$ is open-loop unstable hence a value for the open-loop rise-time and settling-time could not be computed. In closed-loop, systems $G_9(s)$ and $G_{10}(s)$ both yield a critically damped response for which $\zeta = 1$ and equation 2.44 becomes unbounded hence the n/a entries in those rows. The rise- and settling-times for the integrating systems $G_4(s)$, $G_8(s)$ and $G_{12}(s)$ were computed from the MATLAB impulse response plot. The time-delay was included in the calculation of the settling time for systems $G_3(s)$, $G_7(s)$ and $G_8(s)$.

An analysis of table 4.2 would immediately suggest that the various rules yield quite a diverse range of recommendations — differing by up to a factor of seventy for $G_5(s)$. That being said, reasonably consistent results are obtained from the last four methods CL Rise time; damped frequency; bandwidth; natural frequency. This is especially true for systems $G_1(s)$, $G_2(s)$, $G_4(s)$, $G_6(s)$, $G_7(s)$, $G_8(s)$, $G_9(s)$, $G_{11}(s)$ and $G_{12}(s)$.

System $G_3(s)$ is very oscillatory in closed-loop and the relatively small values of ζ and ω_n ($\zeta = 0.01$; $\omega_n = 0.17$) would account for the rather larger values obtained using the damped and natural frequency methods. With system $G_5(s)$ the calculation based on the closed-loop rise-time is slightly distorted as the effect of the inverse response was incorporated within the rise-time metric itself. If this was excluded, and the metric measured from +10% (as opposed to $\pm 10\%$) to +90% of the final value of the steady-state response, this metric would be in line with the others. Finally, the PD controller designed for system $G_{13}(s)$ may have adversely effected the results for this system; the closed-loop response is almost critically damped and this may account for the somewhat more disparate results obtained in this case.

With the exception of $G_5(s)$ (and $G_7(s)$ for which the bandwidth method fails) the closed-loop rise time and closed-loop bandwidth methods are almost perfectly correlated. Since $G_5(s)$ is inverse-unstable this result is not totally unexpected. The discrepancy is not large — the numbers are only out by a factor of 3. In contrast, the settling time method, in general, suggests the use of a relatively long sampling period. This may be attributed to the fact that the settling time is not the best measure of a system’s “speed of response”. The metric is distorted by time-delay, inverse unstable characteristics and low damping factor. In contrast, the rise-time metric is relatively immune to these attributes. Based on this observation it is rational to suggest that equation 2.42 be modified to yield smaller values for h . This hypothesis is supported by two recent texts — that were not available to the author at the commencement of this study; Zhu’s *Multivariable System Identification for Process Control* (Zhu, 2001) and Corriou’s *Process Control* (Corriou, 2004). In the former text the author recommends that the sampling period be chosen according to $0.01t_s \leq h \leq 0.1t_s$ while Corriou suggests the choice $0.067t_s \leq h \leq 0.167t_s$. Applying either of these lower bounds would clearly result in recommendations that are more consistent with the results obtained from the other rules.

In general, the philosophy advocated in this thesis has been to tend towards the minimum recommended value for the sampling period. There are a number of reasons for this:

1. A continuous-time system has, in effect, a sampling period of 0(sec). A fair comparison between digital and continuous would then imply that the minimum possible value of sampling period be chosen.
2. In a digital control system, the sampling period limits the maximum achievable bandwidth of the system. To get fast time-domain responses (i.e. large bandwidth) a “small” sampling period is required. This is true for both set-point tracking and disturbance rejection.
3. As mentioned previously, a unity gain proportional controller was used to generate an indicative closed-loop response. This choice was motivated by simplicity. It is highly probable that the GPC — in fact any controller — will result in much better performance, namely faster rise-times, settling times and wider bandwidth, and hence even shorter sampling periods than those indicated in table 4.2 might be appropriate.

To the author’s knowledge the main disadvantages associated with fast sampling are (i) numerical sensitivity and (ii) computational load. The numerical sensitivity primarily arises from the application of the standard Z-transform which, for high sampling frequencies, results in (a) the system poles tending to the point $(-1, i0)$ in the z-plane (b) the numerator coefficients, when expressed as a polynomial tending towards zero and (c) for many systems, inverse unstable dynamics. Since the GPC does not involve a pole-zero cancellation the latter is not a restriction. The numerical sensitivity is a real problem when performing a discrete-time *parameter estimation* experiment at high sampling frequencies. Numerical errors, round-off errors, sensor noise, etc. will then all combine to shift the coefficients of the estimated discrete-time numerator and denominator. Because the system poles (in discrete time) are all bunched about the point $(-1, i0)$ and the numerator coefficients are quite small, deviations of three or four decimal places are significant — the interested reader is referred to Isermann (1980), for a more comprehensive discussion. An inverse Z-transform may then transform these relatively small deviations in discrete-time to significant movements in the Laplace plane and results in continuous-time poles/zeros that are quite removed from the actual system dynamics.

The idealised scenario studied here, i.e. the actual continuous-time dynamics are available, an estimation experiment is not required, sensor noise is not present, means that this problem is considerably reduced. In these simulations, the system is transformed to discrete-time using MATLAB's double precision floating point arithmetic and all subsequent processing, including Simulink simulations, utilises the same precision. Therefore, the use of a common platform for computation and simulation maintains the integrity of the continuous-to-discrete and discrete-to-continuous transformations and enables considerably higher sampling frequencies than might be advocated in practice.

A high sampling frequency also results in additional computational overhead. This overhead naturally arises due to the reduced time slot within which the calculations must be performed. With time-delayed systems an additional problem arises — the dimension of the discrete-time system increases with reductions in the sampling period. Most direct digital controller design procedures result in a controller that is, at least, the same order as the process model. Therefore, high-order controller transfer functions may result. These present additional computational load — and if the controller is to be implemented on a separate system an additional source of numerical sensitivity (Keel and Bhattacharyya, 1997). If, as is the case here, the design and implementation are integrated then this problem is circumvented. In the GPC case, the controller parameter N_2 should generally be chosen to be longer than the system settling time — measured in samples. A small sampling period will result in a long settling time (measured in sampling periods). This can present its own challenges if the controller is to be designed and implemented on-line as in an adaptive setting, as the dimension of the matrix \mathbf{G} (see equation 2.25) is directly related to N_2 . In a non-adaptive setting the GPC controller can be implemented as two difference equations (figure 2.4) and therefore this is not a significant issue.

Based on these considerations it is the author's considered opinion that, in this case, the advantages associated with employing a high sampling frequency outweigh the disadvantages and therefore the author tended towards the minimum recommended value of sampling period for each of the systems $G_1(s)$ — $G_{13}(s)$. Table 4.3 summarises this choice. However, during the course of this research it was necessary to revisit this

Model	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}
h	0.1	0.1	0.2	0.2	0.1	0.1	0.1	0.1	1	0.1	0.1	0.1	0.1

Table 4.3 Choice of sampling period for systems G_1 to G_{13}

	PID			2DOF PID			GPC			
Model	IAE_{both}	A_m	ϕ_m	IAE_{both}	A_m	ϕ_m	$h(sec)$	IAE_{both}	A_m	ϕ_m
$G_2(s)$	0.409	22.70	45	0.27	15.59	45	0.1	0.46	6.13	44

Table 4.4 PID, 2–DOF PID and GPC controllers optimised for model G_2

choice. The motivation for this is as follows. Consider the process model $G_2(s)$. When the PID and 2–DOF PID controllers were tuned for this model, using the objective function equation 3.24 — minimise the combined IAE criterion subject to minimum constraints on the gain margin ($A_m \geq 6dB$) and phase margin ($\phi_m \geq 45^\circ$), the results of table 4.4 were obtained. Sampling $G_2(s)$ at 0.1 second intervals yields the discrete–time model

$$G_2(z) = \frac{0.0086944(z + 2.432)(z + 0.1505)(z + 0.001186)}{(z - 3.727e - 006)(z - 0.08208)(z - 0.6065)(z - 0.9048)} \quad (4.20)$$

The GPC controller was now tuned using the same criterion and the third column of table 4.4 was obtained. It is evident, and quite unexpected, that the GPC controller does not outperform the other controllers. Similar results were obtained for systems $G_4(s)$, $G_8(s)$, $G_9(s)$, $G_{10}(s)$, $G_{11}(s)$, $G_{12}(s)$ and $G_{13}(s)$.

Considering system $G_2(z)$ it is clear, due to the inherent one sample delay caused by the ZOH, that the minimum error that can be obtained is 0.1sec (it takes a minimum of one sample to reach the set–point). To outperform the 2–DOF PID controller the remainder of the (combined servo and regulator) dynamic error response must be less than 0.17(sec) which is only feasible if a one–step–ahead design is realised. The logical solution is to reduce the sampling period. This logic can be analytically supported by noting that the PID controller yields a closed–loop rise–time of 0.19(sec) while the 2–DOF PID controller yields a closed–loop rise–time of 0.23(sec) and, according to equation 2.43, a minimum sampling period of 0.02(sec) could be applied. A simulation study was conducted to quantify the exact effect of the sampling period on the combined

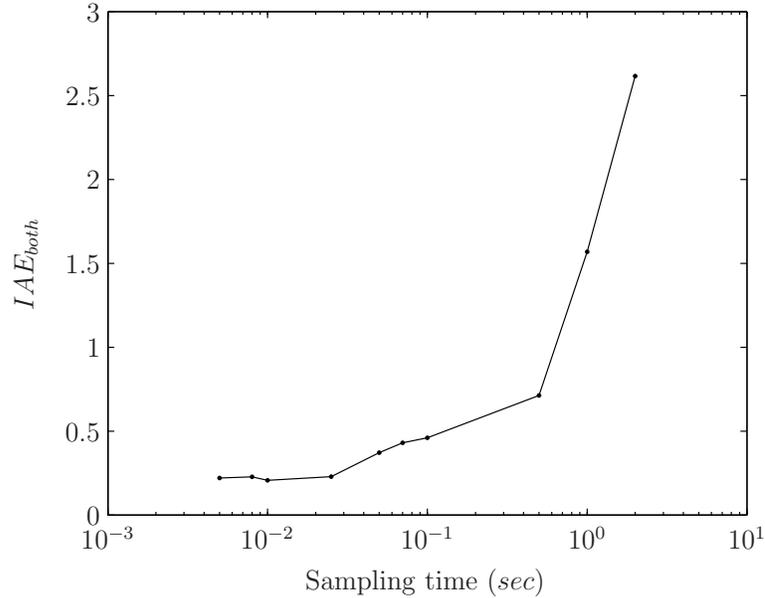


Figure 4.1 Sampling period versus performance

servo plus regulator integral of error response. In this study, the sampling period was varied as follows: $h := 2, 1, 0.5, 0.1, 0.07, 0.05, 0.025, 0.01, 0.008, 0.005$. For each value of h the corresponding discrete-time transfer function $G_2(z)$, was determined and an optimal GPC controller was designed based on equation 3.26.

Figure 4.1 presents the variation of the combined integral of absolute error (IAE_{both}) with sampling period (presented using a logarithmic scale for clarity). The IAE_{both} decreases appreciably with decreasing h until the sampling period $h = 0.005$ is reached. Beyond this point the IAE_{both} metric becomes relatively independent of the sampling frequency. Based on these considerations a revised sampling period of $h = 0.01$ would appear to be a sensible choice. The corresponding discrete-time transfer function is —

$$G_2(z) = \frac{4.8514 \cdot 10^{-6}(z + 7.49)(z + 0.7356)(z + 0.07135)}{(z - 0.2865)(z - 0.7788)(z - 0.9512)(z - 0.99)} \quad (4.21)$$

Compared with equation 4.21, the numerator coefficients have decreased three-fold and would clearly be susceptible to numerical errors arising from a parameter estimation experiment and the location of all three zeros has shifted to the left. The four poles have migrated closer to the $(-1, i0)$ point in the z -plane but are still distinguishable which 4.21 would suggest that this choice of h is not too small. If the GPC is designed based on equation table 4.5 results.

	PID			2DOF PID			GPC			
Model	IAE_{both}	A_m	ϕ_m	IAE_{both}	A_m	ϕ_m	$h(sec)$	IAE_{both}	A_m	ϕ_m
$G_2(s)$	0.409	22.70	45	0.27	15.59	45	0.01	0.208	11.30	45

Table 4.5 PID, 2–DOF PID and revised GPC controllers optimised for model G_2

Model	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}
h	0.1	0.01	0.2	0.01	0.1	0.1	0.1	0.02	0.2	0.01	0.01	0.01	0.01

Table 4.6 Choice of revised sampling period for systems G_1 to G_{13}

A similar analysis was performed on systems $G_4(s)$, $G_8(s)$, $G_9(s)$, $G_{10}(s)$, $G_{11}(s)$, $G_{12}(s)$, $G_{13}(s)$; table 4.6 lists the revised sampling periods that resulted. The sampling period for system $G_3(s)$ also had to be modified. The sampling period of $h = 0.2sec$ (as indicated in table 4.5) results in a discrete–time delay of 75 sampling periods and a 78th–order GPC controller. The high–order, of both the process and the controller, resulted in severe numerical problems and, in general, the optimisation failed. It is important to point out that it is the author’s belief that this issue is not due to a limitation of the GPC algorithm, but is a function of its implementation. The underlying GPC code is implemented as a Simulink S–function written in C source code where the delay is modelled as leading zeros in the discrete–time numerator polynomial. This inherently results in a high–order process model and a numerically weak implementation algorithm. This algorithm needs to be improved to deal with high–order/delay–dominant processes — possibly by extracting the delay and dealing with this separately — however this is an issue for future work. A trial and error procedure revealed that a sampling period of $h = 0.7$ solved this particular problem (the delay is now reduced to 21 sampling periods) and therefore this choice was adopted during the following experiments. Table 4.6 summarises the sampling periods that were chosen for each of the process models.

4.3 Optimisation environment

Prior to discussing the results it is appropriate to make a few brief comments regarding the implementation details. The cost function of section 3.5.1 and 3.5.2 was optimised

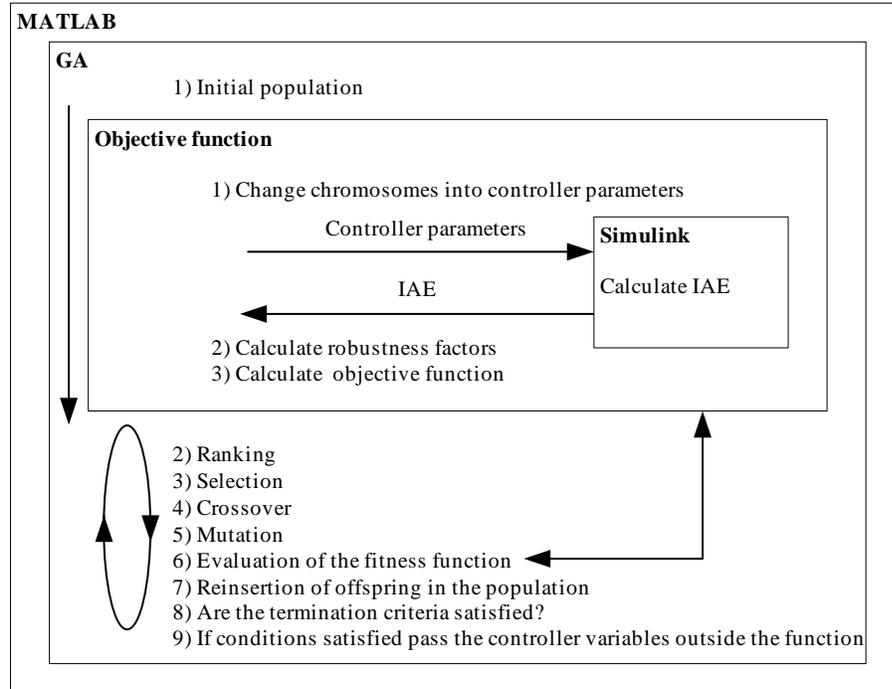


Figure 4.2 Flow diagram of optimisation environment

in the MATLAB/Simulink environment using a GA. Time did not permit an evaluation of the input sensitivity criteria, section 3.5.3. This function will be discussed in the real-time experiments. The flow chart in figure 4.2 illustrates the optimisation procedure. In this figure the outer block presents the MATLAB environment within which the GA code executes. The GA layer, in turn, calls the objective function routine which interacts with the Simulink environment to determine the IAE criteria. First, the GA parameters such as: range, accuracy, number of tuning parameters are defined and an initial population is generated. The GA iterates as long as the specified termination condition(s) are not fulfilled. The robustness factors (gain and phase margins, modulus margin) are calculated in MATLAB and the IAE is calculated in Simulink. For the systems with time delay an eight order Padé approximation is used in the calculation of the *MM*. A high-order Padé approximation produces transfer functions with clustered poles. Because such pole configurations tend to be very sensitive to perturbations, Padé approximations greater than 10^{th} order should be avoided, (Mathworks, 2004a). Trial and error experimentation suggested that an eight order approximation was a sensible choice.

Both the PID and 2-DOF PID controllers were implemented in continuous-time while the GPC controller was implemented as a discrete-time system. Hence, it was necessary to choose a sampling period for the GPC controlled systems but not for the other two. The choice of sampling period was discussed in the previous section and values used are listed in table 4.6. The control signal was constrained to the range $-10 \leq u(k) \leq 10$ in all of the simulations.

Extensive trial and error indicated that a common set of Simulink simulation parameters was possible, namely:

- Solver: variable step ODE45
- Output refine factor 3
- Relative error tolerance $1 \cdot 10^{-6}$

The exception is model $G_7(s)$ where the relative error tolerance was set to the default value, $1 \cdot 10^{-3}$, due to numerical problems when the PID controllers were evaluated. The same relative error tolerance and the output refine factor of 3, was used for model $G_9(s)$, due to a high computational burden.

The time taken to optimise the controller parameters varied. For a simple model controlled by the ideal PID controller the optimisation took about 15 minutes on a *Pentium IV, 1.8 GHz* PC. However, tuning the GPC controller was more time consuming and took up to two hours for models G_5 and G_{12} . A trade-off between computation time and attaining the global optimum exists and can be negotiated by:

- Choosing the range over which the coefficients were tuned.
- Changing the population size.
- Changing the relative error tolerance parameter in Simulink.

Evaluating the objective function is the most time consuming component and requires 99% of the computation time. If the PID controller is considered, the calculation of the IAE in the Simulink model absorbed 67% of the processing time while the function margin, which returned the gain and the phase margins, required 32% of the time. In

contrast, the GA operators (e.g. crossover, mutation, etc.) were very efficient and do not require further optimisation. The calculation of the IAE for the PID controller could be improved by the MATLAB `lsim` function, however, the GPC algorithm was implemented in Simulink and for consistency this environment was maintained. The fixed step solver is in general more time consuming, however, the objective function evaluation is more accurate and therefore this choice was maintained. These solvers are well documented in the Mathworks manual (Learning Simulink, 2004) and were also analysed by O'Mahony and Downing (2000a).

Evaluating the objective function is the most time consuming component and requires 99% of the computation time. If the PID controller is considered, the calculation of the IAE in the Simulink model absorbed 67% of the processing time while the function `margin`, which returned the gain and the phase margins, required 32% of the time. In contrast, the GA operators (e.g. crossover, mutation, etc.) were very efficient and do not require further optimisation. The calculation of the IAE for the PID controller could be improved by the MATLAB `lsim` function, however, the GPC algorithm was implemented in Simulink and for consistency this environment was maintained. The fixed step solver is in general more time consuming, however, the objective function evaluation is more accurate and therefore this choice was maintained. These solvers are well documented in the Mathworks manual and were also analysed by O'Mahony and Downing (2000a).

A solution to speed the GA up is to implement a look up table. As was mentioned before, the evaluation of the fitness function is the most time consuming part, so a faster GA implies that the objective function must be executed quicker. GAs are stochastic search techniques and a fitness function with the same controller parameters may be evaluated a number of times in each population. The number of unnecessary computations increases with each generation, as the algorithm converges. To avoid this, a matrix (look up table) is generated, where previously computed GA chromosomes are stored with the equivalent value of the objective function. The algorithm checks whether the fitness function was previously evaluated and, if so, the objective function is not executed; instead the solution is passed through. A disadvantage of this method is that

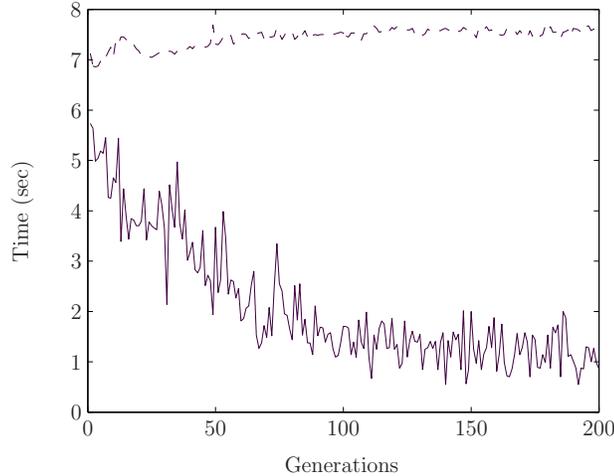


Figure 4.3 Effect of the look up table

the matrix grows with each member of the population that is evaluated. However, this is not a serious restriction as the MATLAB `find` function is relatively fast and this solution provides a faster off-line optimisation.

Consider the GPC parameter tuning problem on the previously mentioned PC. Without the look up table the optimisation takes almost 25 minutes and the population of 50 individuals requires, on average, 7.45(sec) to be evaluated. The time may vary depending on the value of N_2 . If the look up table is implemented, individuals require approximately 2.1(sec) to be evaluated and the optimisation is performed in 7 minutes. This performance improvement is illustrated in figure 4.3, where the x-axis represents the number of generations and the y-axis shows the time which is required to evaluate a single generation. The dashed plot presents the GA solution without the look up table, while the solid line corresponds to the case where the look up table is used. Note that the initial population has been evaluated and this data has already been stored in the matrix. As was mentioned before, 5% of the population is passed to the next generation unchanged. This is the reason why both algorithms do not start at the average value of 7(sec). The evaluation of this design is three times faster when the look up table is included even though the dimension of the final matrix was 2860 rows and 8 columns (7 tuning parameters and the result of objective function). The combination of the GA with a look up table significantly reduces the computation time required to tune the GPC controller, while the PID controller tuning time is reduced as well. This method

model	PID			2-DOF PID			GPC			
	IAE_{both}	A_m	ϕ_m	IAE_{both}	A_m	ϕ_m	$h(sec)$	IAE_{both}	A_m	ϕ_m
G_1	2.08	∞	44	1.62	14.64	45	0.1	1.242	9.14	48
G_2	0.409	22.7	45	0.27	15.59	45	0.01	0.208	11.3	45
G_3	42.9	6.27	64	41	6.02	64	0.7	33.9	6	60
G_4	5.46	∞	44	1.4	9.24	9	0.01	0.838	11.87	71
G_5	10.7	5.94	58	9.94	5.88	55	0.1	8.79	6	59
G_6	1.44	∞	45	1.15	13.34	45	0.1	0.859	6.97	45
G_7	2.59	∞	63	2.55	7.05	63	0.1	2.11	6.02	60
G_8	7.39	8.02	47	4.74	5.82	44	0.02	2.83	9.33	71
G_9	1.03	∞	78	0.746	∞	69	0.2	0.533	6.08	45
G_{10}	0.38	∞	45	0.3	13.04	45	0.01	0.195	12.65	51
G_{11}	0.351	∞	45	0.259	25.57	44	0.01	0.166	9.26	45
G_{12}	1.09	∞	45	0.729	∞	45	0.01	0.531	7.69	52
G_{13}	0.746	-11.68	50	0.354	-16.42	46	0.02	0.382	6.09	62

Table 4.7 Optimised IAE_{both} with constraints $A_m \geq 6dB$ and $\phi_m \geq 45^\circ$

is most applicable when the objective function evaluation is time consuming. To the author's best knowledge, this solution has not been presented in the literature to date.

4.4 Results

4.4.1 The objective function based on GPM method

The simulation results for the first design ($A_m \geq 6dB, \phi_m \geq 45^\circ$) are presented in table 4.7 while appendix A presents the details of the controller parameters and performance criteria. Clearly, the gain and phase margin criteria were satisfied for all systems with the exception of the inverted pendulum model, $G_{13}(s)$, where a negative gain margin was returned for the ideal PID and 2-DOF controllers. This does not imply that the system is unstable. In both cases the GA was able to determine PID parameters that stabilised the open-loop unstable system, $G_{13}(s)$, as is indicated by the small, finite IAE value returned. The result is merely implying that the use of the gain (and phase) margin criterion is misleading for this system. Since the system is open-loop unstable, stability may only be evaluated by applying Nyquist's General Stability Criterion,

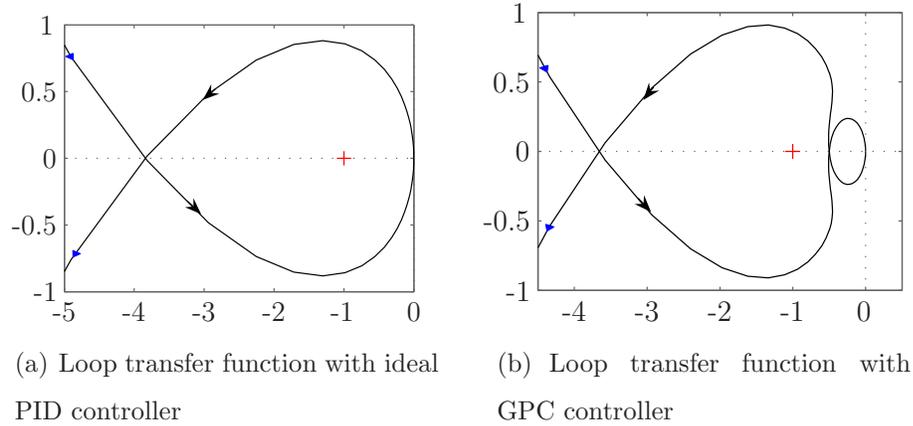


Figure 4.4 Nyquist diagram of open-loop transfer function with controllers tuned for $G_{13}(s)$

namely:

$$Z = N + P \quad (4.22)$$

where P is the number of unstable open-loop poles, N is the number of encirclements of the critical $\{-1, i0\}$ point (N is positive for clockwise encirclements and negative for counter-clockwise encirclements) and Z is the number of closed-loop poles in the right-half of the s -plane. The process transfer function contains one unstable open-loop pole and since the ideal PID controller is restricted to having a pole at the origin, $P = 1$. Therefore, to obtain a stable closed-loop system $N = -1$, i.e. the loop transfer function must encircle the critical point at least once in a counter-clockwise direction. An examination of the Nyquist plot for the resulting ideal PID controller (figure 4.4(a)) reveals that this is indeed the case. A similar result exists for the 2-DOF PID controller.

In figure 4.4, the loop transfer function intersects the negative real axis at -3.8 and therefore a negative gain margin is returned. The Nyquist plot of the loop transfer function for the GPC controlled system is illustrated in figure 4.4(b). The same counter-clockwise encirclement is evident. The loop transfer function is sixth-order (as opposed to the third-order loop transfer function that results from the ideal PID design) and it may be the case that the larger number of design parameters enabled a slightly more intricate curve that intersects the negative real axis in two places. The first intersection, at approximately -0.5, results in a positive gain margin of 6dB's. However, the A_m

criterion (for all three controllers) is misleading for this system and must be treated with caution. It is also possible that the phase margin criterion will return misleading results for open-loop unstable systems. In general, the stability of open-loop unstable systems may only be assessed by examining the number of encirclements of the critical point. Clearly then, the gain and phase margin criteria may not (reliably) be used to tune open-loop unstable systems and this represents a significant limitation of the method.

From table 4.7 it is clear that the GPC and 2-DOF PID controllers yield a lower IAE value (while satisfying the minimum robustness constraints) than the ideal PID controller. It is not surprising that the 2-DOF controllers should outperform the 1-DOF controller, however the GPC controller performs significantly better than the 2-DOF PID controller — on average the IAE was reduced by 24%. The most significant reductions occurred for the two integrating plants, $G_4(s)$ and $G_8(s)$ where the IAE was reduced by 40%. Model $G_{13}(s)$ represents the only exception to this rule — the GPC controller yielded a higher (by 8%) IAE value than the 2-DOF PID controller. However, the fact that the latter did not satisfy the minimum A_m specification may have biased these results. A cursory examination of the loop gain margins reveals that, on average, the A_m associated with the 2-DOF PID controller is larger than the corresponding value of the GPC controller. This would suggest that a classic compromise — robustness versus performance — exists. However, this hypothesis is questionable as the phase margin associated with the GPC design, is on average, the same or greater than that for the 2-DOF PID controller.

To investigate the robustness versus performance idea further a second design was performed where the minimum gain margin was increased to $14dB$. The results are summarised in table 4.8. Surprisingly, the results in this case are even clearer. The results from the 1-DOF PID controller design will be ignored for the present. Likewise, system G_{13} has already been discussed in some depth, and will be omitted from the following discussion. Examining the table, it is evident that the GPC clearly outperforms the 2-DOF PID controller for systems G_1 , G_4 , G_5 , G_6 , G_7 , G_8 , G_{10} and G_{11} in that the combined IAE is lower, the gain margins are approximately equivalent and equal

model	PID			2-DOF PID			GPC			
	IAE_{both}	A_m	ϕ_m	IAE_{both}	A_m	ϕ_m	$h(sec)$	IAE_{both}	A_m	ϕ_m
G_1	2.05	∞	45	1.63	14.59	45	0.1	1.329	13.82	43
G_2	0.407	22	45	0.256	17.02	45	0.01	0.233	14	53
G_3	53.9	14	63	53.7	14.01	63	0.7	56	13.95	75
G_4	4.47	∞	45	3.51	14.76	45	0.01	0.838	14	70
G_5	15.6	14.01	60	15.6	14.02	57	0.1	12.8	14.17	67
G_6	1.47	∞	45	1.15	14.22	45	0.1	0.921	14.08	45
G_7	2.56	∞	62	3.09	14.19	61	0.1	2.51	14.44	62
G_8	9.7	14.22	44	8.93	13.99	44	0.02	2.5	14.81	∞
G_9	1.03	∞	77	0.584	∞	80	0.2	0.548	14.15	47
G_{10}	0.377	∞	45	0.331	13.99	45	0.01	0.195	14.01	50
G_{11}	0.349	∞	45	0.297	14.33	45	0.01	0.189	14.01	51
G_{12}	1.09	∞	45	0.588	∞	48	0.01	0.531	14.06	53
G_{13}	0.746	-11.68	50	0.352	-20.37	44	0.02	0.3877	14.02	61

Table 4.8 Optimised IAE_{both} with constraints $A_m \geq 14dB$ and $\phi_m \geq 45^\circ$

to 14dB's while the phase margins are either similar $\{G_6, G_7\}$ or the GPC results in a larger phase margin metric. The GPC results in a smaller IAE value for system G_2 while the 2-DOF PID controller results in a larger A_m . A similar result is obtained for system G_{12} . For model G_9 the 2-DOF PID controller yields better gain and phase margin indices but the GPC minimises the IAE. The only system where the 2-DOF yields a better IAE metric is model G_3 and it is probable that if a smaller sampling period could have been applied that this result would also have been overturned.

Overall, the GPC reduced the IAE value, on average, by approximately 24% while satisfying the minimum robustness criteria that were specified. The GPC and 2-DOF PID algorithms both have similar structures (they both have two degrees of freedom) and a comparable number of tuning parameters and in this context the superior performance of the GPC was a little surprising and would suggest that this controller may offer some real benefit to the process industry.

A selection of closed-loop responses are illustrated in figures 4.5 and 4.6 where the solid line represents the response of the GPC controller, the dashed line that of the 1-DOF PID controller and the dotted line the 2-DOF PID controller. In each of these

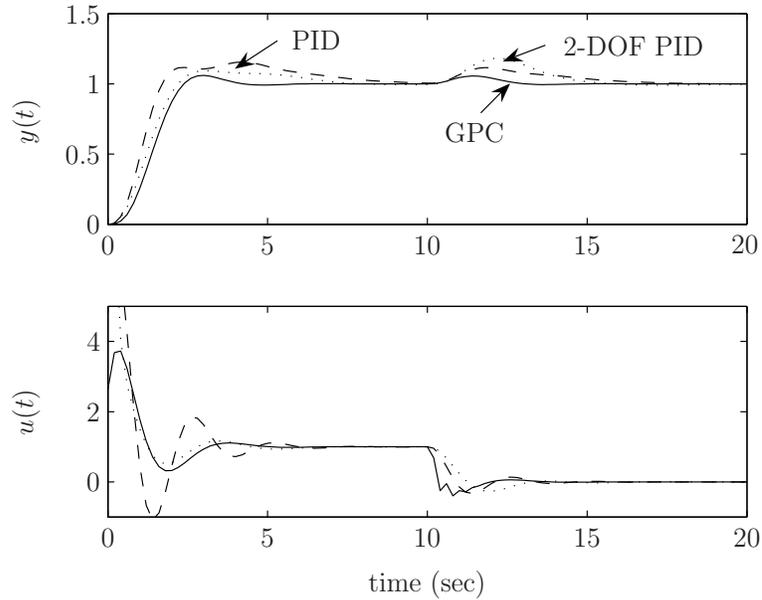


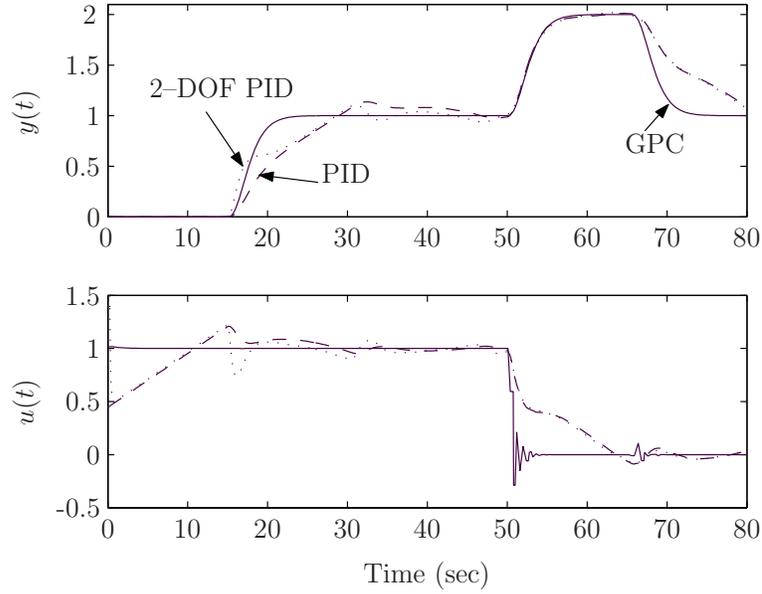
Figure 4.5 Set-point following for $G_1(s)$, $A_m \geq 6dB$

figures the upper plot illustrates the closed-loop response to a unit step input occurring at $t = 0$ and to a unit step disturbance. In figure 4.5 the disturbance was applied at $t_0 = 10(sec)$, while in figure 4.6 the disturbance occurs at time $t_0 = 35(sec)$. Also shown are the corresponding control signals. The superior performance of the GPC controller is clearly evident in each case.

4.4.2 The objective function with constraints on MM

Table 4.9 summaries the tuning results when the objective function was given by equations: 3.29, 3.30 and 3.31. The constraint on the modulus margin ($MM \geq 0.6$) was satisfied for all controllers except for the GPC controller when model $G_8(s)$ was used. Initially, the controllers were tuned with a constraint on the maximum value on the Bode plot of the complementary sensitivity function, $M_p \leq 1.5$. However, this did not realise any real benefit, as the tables in appendix A.2 show that this constraint was satisfied anyhow. Consequently, the cost function was simplified and only the MM was included.

From table 4.9 can be seen that when the MM constraint is fulfilled, the gain and/or phase margins are, in some cases, negative. This is especially true for the GPC design.

Figure 4.6 Set-point following for $G_3(s)$, $A_m \geq 6dB$

Consider system $G_2(s)$ which is open-loop stable. The corresponding discrete-time system, obtained through the standard Z-Transform with $h = 0.01$ is also stable. The GPC which results from the GA optimisation, may, with reference to figure 2.4, be represented by the following two transfer functions:

$$G_{PF} = \frac{T_1(z)}{S(z)} = \frac{0.00517(z + 0.824)(z + 0.727)(z - 0.708)}{(z - 0.7718)(z - 0.2865)(z^2 - 1.739z + 0.7687)} \quad (4.23)$$

$$G_c = \frac{S(z)}{\Delta R(z)} = \frac{27181(z - 0.7718)(z - 0.2865)(z^2 - 1.739z + 0.7687)}{(z - 1)(z + 1.232)(z + 0.7344)(z + 0.04304)} \quad (4.24)$$

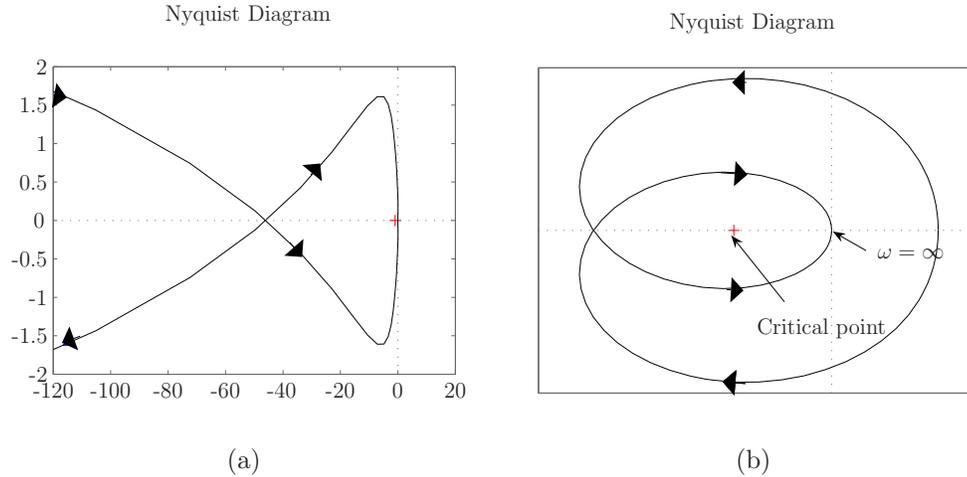
Clearly the design results in a controller, $G_c(z)$, which has a pole outside the unit circle. Therefore the open-loop transfer function is unstable and, as with system G_{13} , the standard gain and phase margin criteria are no longer applicable. A Nyquist plot of the open-loop transfer function reveals that the critical point is encircled once in the counter-clockwise direction and therefore the closed-loop system is stable. An analysis of, G_4 , G_6 , G_{10} and G_{11} reveals that each of these closed-loop systems are stable (as indicated by the finite IAE values) and that the negative gain and/or phase margin index is a direct result of an unstable open-loop transfer function. For such systems the classical gain and phase margin criteria are unreliable and stability must be ascertained via encirclements of the critical point on the Nyquist diagram.

model	PID				2-DOF PID				GPC				
	IAE_{both}	A_m	ϕ_m	MM	IAE_{both}	A_m	ϕ_m	MM	$h(sec)$	IAE_{both}	A_m	ϕ_m	MM
$G_1(s)$	2.01	∞	43	0.60	1.7	20.61	45	0.60	0.1	1.192	0.61	38	0.61
$G_2(s)$	0.433	28.24	47	0.66	0.337	12.90	50	0.60	0.01	0.207	-4.09	-36	0.60
$G_3(s)$	43.5	7.93	64	0.60	42.1	7.94	64	0.60	0.7	37.1	7.95	64	0.60
$G_4(s)$	5.02	∞	42	0.59	3.81	22.21	41	0.59	0.01	0.816	-6.48	-38	0.60
$G_5(s)$	12	8.27	60	0.60	11.7	8.15	58	0.60	0.1	10.4	7.90	65	0.60
$G_6(s)$	1.62	∞	90	0.60	1.38	17.06	66	0.60	0.1	0.775	-6.52	37	0.60
$G_7(s)$	2.61	∞	61	0.67	2.63	8.18	63	0.61	0.1	2.45	7.97	64	0.60
$G_8(s)$	8.525	10.37	38	0.6	7.962	8.12	41	0.6	0.02	10.36	6.36	47	0.52
$G_9(s)$	1.03	∞	78	1.02	0.589	∞	86	1.00	0.2	0.536	8.32	45	0.60
$G_{10}(s)$	0.432	∞	42	0.60	0.33	18.42	45	0.60	0.01	0.195	-5.33	-37	0.60
$G_{11}(s)$	0.375	∞	42	0.60	0.296	19.53	45	0.60	0.01	0.165	-4.29	-37	0.60
$G_{12}(s)$	1.44	-33.19	51	0.86	0.819	-20.12	35	0.60	0.01	0.531	10.08	50	0.63
$G_{13}(s)$	1.13	-12.09	62	1.00	0.476	-10.13	38	0.61	0.1	0.387	7.90	36	0.60

Table 4.9 Optimised IAE_{both} with constraint $MM \geq 0.6$

A similar analysis can be applied to system $G_{12}(z)$ for the 1-DOF and 2-DOF PID controllers. In both cases the gain margin criterion indicates that the system is unstable. For the 1-DOF controlled system, MATLAB generates the (incomplete) Nyquist plot of figure 4.7 (A). However, the open-loop transfer function consists of a double pole at the origin (one due to the process transfer function while the other results from the integral term of the PID controller). A double pole at the origin creates a $+360^\circ$ arc in the clockwise direction that is infinitely large. A sketch of the real Nyquist diagram (not to scale because of the infinite arc) is presented in figure B. The critical point is encircled once in the clockwise direction and once in the anti-clockwise direction and the net encirclement is zero. Since the open-loop transfer function has no unstable poles, $P = 0$, and equation 4.22 yields $Z = 0$. Hence, the conventional gain margin metric yields an incorrect result and stability can be (properly) assessed via the Nyquist diagram. The negative gain margin for system $G_{13}(s)$ is a direct result of the open-loop unstable pole and has already been discussed previously.

With the singular exception of system G_8 , the GPC yields better performance in that the IAE criterion is minimised while, at the same time, the minimum robustness

Figure 4.7 Nyquist plot of $G_{12}(z)$

criterion that was specified are achieved. On average, the GPC reduces the combined (servo plus regulator) IAE by almost 19% compared with the 1-DOF PID controller and by 12% compared with the 2-DOF controller. This performance gain is somewhat less than was achieved using the gain and phase margin criteria but still represents a considerable gain in performance.

4.4.3 Summary of the results

Thirteen models, three objective functions and three controllers have been evaluated. In summary, a few brief comments about the results are appropriate. First, the models will be compared. A general opinion is that models with time delay were more difficult to tune than the others. Many difficulties occurred due to the continuous-time description of the models. While the continuous-time transfer function is more intuitive, the discrete-time or a state-space description is easier to implement and numerical problems (arising from numerical integration routines) do not occur. However, the choice of sampling period is not trivial. For some models, like the inverted pendulum, G_{13} , PID controllers with satisfactory A_m and ϕ_m criteria were not found.

These simulation results, on their own, may be very misleading. The designs which work well on a linear model may fail when applied to a nonlinear plant with high frequency noise and uncertain model. To further investigate this issue, real-time experiments are presented in the next chapter.

Chapter 5

Real-time application

In this chapter the controllers and the objective functions are evaluated based on a real-time system. Design strategies based on the gain and phase margin, modulus margin and the maximum value of the input sensitivity function on the Bode plot are analysed. The three previously mentioned controllers are implemented and the results are discussed.

5.1 Flexible link

A flexible link was used to evaluate the controllers in a real-time application. It was designed by *Quanser*¹ and a diagram is presented in figure 5.1. It consists of a thin piece of steel, the link, which is controlled by a DC motor. The system output is measured via two sensors. The first sensor (potentiometer $10k\Omega$) measures the angle of the link (denoted by b in figure 5.1) while the second sensor, a strain gauge, is used to measure the deflection of the tip (denoted by a in figure 5.1). The overall plant output — the tip position, is determined by summing the outputs from each sensor. However, since the strain gauge is more sensitive than the potentiometer, the following scaling was suggested by Cunningham (2003) and applied here

$$y(t) = a(t) + \frac{b(t)}{9.13} \quad (5.1)$$

The flexible link is effectively a position servo control problem where the objective is to rapidly and accurately control the tip position and minimise the naturally occurring

¹www.quanser.com

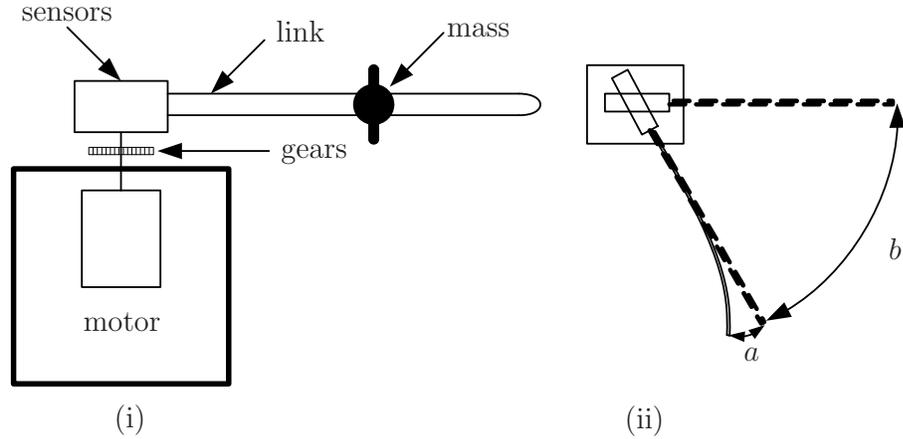


Figure 5.1 Diagram of the flexible link: (i) side view (ii) top view

modes of vibration due to its flexible nature. These modes of vibration can be enhanced, and model uncertainty introduced, by attaching a mass to the link. The mass may be moved from the tip position towards the fixed end of the link. This application considers five locations for the 84 gram mass. The nominal model corresponds to the mass positioned at 150 units² from the motor shaft (half-loaded). The other possibilities correspond to the unloaded link, the case where the mass is positioned 100 units from the shaft (quarter-loaded), the mass at 200 units (three quarter-loaded) and the fully-loaded link (i.e mass at 250 units).

The flexible link is controlled from a PC via a *Data Acquisition Unit*, denoted as DAU, in the block diagram, figure 5.2. The controller is designed in *Simulink, version 3.0*, while the environment and the executable code is created using the *Real-Time*

²150 units equal 27 centimeters

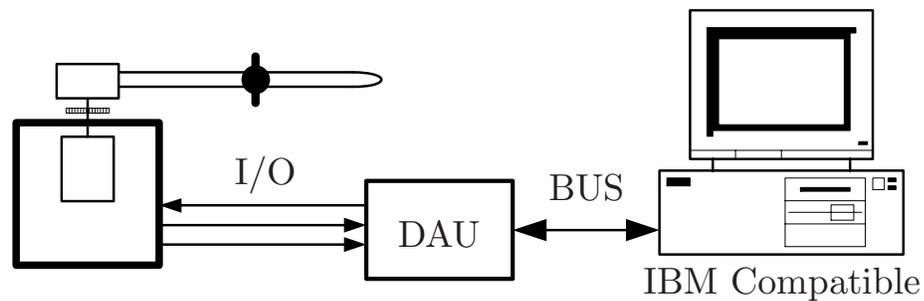


Figure 5.2 Block diagram of the PC connected with the flexible link

Workshop Toolbox, version 3.0.0. This executable is downloaded and executes on the PC's processor, in this case a 450MHz *Intel Pentium III*. Communication with the flexible link occurs via the *MultiQ-3* data acquisition unit while the companion software, *Wincon*³, enables data acquisition, monitoring and control of the real-time experiment.

5.1.1 Identification of the flexible link

The models used to describe the flexible link were obtained from Cunningham (2003). In his research the following methods were applied: step identification; impulse identification; *Pseudo Random Binary Sequence* (PRBS) correlation; closed-loop indirect identification; closed-loop direct identification; iterative closed-loop identification (controller redesign). The author found that the impulse identification gave the most accurate model and chose the sampling time as $h = 0.01$. Based on this work the nominal model, (mass 150), is given by equation 5.2

$$G(z) = \frac{-0.01147(z - 1.16)(z - 0.1569)(z^2 - 1.376z + 0.7694)}{(z - 1)(z^2 - 1.786z + 0.822)(z^2 - 1.267z + 0.9421)} \quad (5.2)$$

the unloaded model by equation 5.3

$$G(z) = \frac{-0.011557(z - 1.253)(z + 0.2826)(z^2 - 1.211z + 0.6305)}{(z - 1)(z^2 - 1.835z + 0.8812)(z^2 - 0.4738z + 0.8164)} \quad (5.3)$$

the mass at 100 by equation 5.4

$$G(z) = \frac{-0.013618(z - 1.189)(z + 0.1436)(z^2 - 1.621z + 0.8828)}{(z - 1)(z^2 - 1.797z + 0.8495)(z^2 - 1.342z + 0.9201)} \quad (5.4)$$

the mass at 200 by equation 5.5

$$G(z) = \frac{-0.012424(z - 1.143)(z + 0.1543)(z^2 - 1.222z + 0.5924)}{(z - 1)(z^2 - 1.791z + 0.8129)(z^2 - 0.5981z + 0.9817)} \quad (5.5)$$

the mass at 250 by equation 5.6

$$G(z) = \frac{-0.0066031(z - 1.169)(z - 0.8365)(z + 0.9972)(z + 0.4297)}{(z - 1)(z^2 - 1.9z + 0.9063)(z^2 + 1.52z + 0.9846)} \quad (5.6)$$

These models were validated by subjecting both the models and the flexible link to a step input. The time-domain responses of both the models and the real system were compared and a close correlation between both was observed. Hence, the models were judged to be sufficiently accurate for controller design purposes. Additional detail on the model validation can be found in Cunningham (2003).

³delivered from the Quanser[®]

5.1.2 Tuning strategy

The controllers were applied to the flexible link to evaluate them on a real-time system. Firstly, the controller parameters were tuned in the MATLAB/Simulink environment using the nominal model, equation 5.2. In Simulink the IAE_{both} was calculated using the general purpose fixed step solver (ODE5, Dormand–Prince formula) with a sampling period of 0.01(sec). In keeping with equation 3.27 the controllers were tuned with the simulation end time set to $t_1 = 5(sec)$ and the disturbance was applied at $t_0 = 3(sec)$.

The gain and phase margins were calculated using discrete transfer functions for the GPC controller. Since the PID controllers were implemented as continuous-time systems (and the process models discrete) a conversion was necessary for the gain and phase margins to be computed. In all instances involving the PID and 2–DOF PID controllers, the process transfer function was converted to a continuous-time transfer function using the inverse Z–transform, assuming the presence of a zero–order–hold on the input. The gain and phase margins were then determined for the continuous-time system.

The first design was based on the gain and phase margin tuning technique, presented in section 3.5.1. The IAE of the servo and regulatory response was optimised subject to $\underline{A}_m = 6dB$ and $\underline{\phi}_m = 45^\circ$, see equation 5.7. The second design, equation 5.8, assumed a larger minimum gain margin and the third design employed the modulus margin. However, the real-time results (detailed in section 5.1.3) for these three designs were not encouraging — specifically the GA designed GPC was unable to stabilise the real-time system, though adequate performance was achieved with both PI structures. Hence a fourth design based on the input sensitivity function was proposed. This decision was made at an advanced stage in the project and time did not permit for a comprehensive simulation analysis as detailed in chapter 4. The value $M_u = 6dB$ was chosen based on trial and error and an examination of the input sensitivity function returned from previous designs. The cost functions are listed below.

$$J_1 = \min \{IAE_{both} \cdot \lambda_{Am} \cdot \lambda_{\phi m}\} \quad s.t. \underline{A}_m = 6dB, \underline{\phi}_m = 45^\circ \quad (5.7)$$

$$J_2 = \min \{IAE_{both} \cdot \lambda_{Am} \cdot \lambda_{\phi m}\} \quad s.t. \underline{A}_m = 14dB, \underline{\phi}_m = 45^\circ \quad (5.8)$$

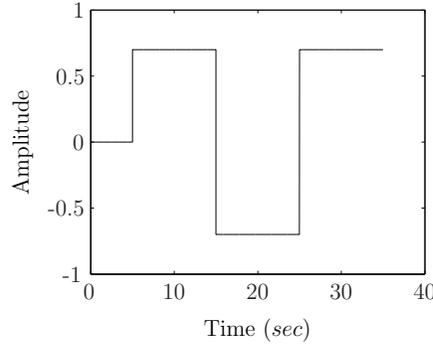


Figure 5.3 Set point signal applied to the flexible link

$$J_3 = \min \{IAE_{both} \cdot \lambda_{MM}\} \quad s.t. \underline{MM} = 0.6 \quad (5.9)$$

$$J_4 = \min \{IAE_{both} \cdot \lambda_{Mu}\} \quad s.t. \overline{Mu} = 6dB \quad (5.10)$$

The fourth design cannot be used to optimise the ideal PID controller, given by equation 2.1, due to the pure derivative action. The M_u for this controller tends to infinity. For this reason a filter on the derivative action, N , was introduced and the new control law is given by equation 5.11. This new structure was optimised with respect to equations 5.7–5.10. The controller coefficients are listed in Appendix B.

$$U(s) = E(s) \left(K_p + \frac{K_i}{s} + \frac{K_d s}{1 + \frac{K_d s}{N}} \right) \quad (5.11)$$

Clearly, this structure is also a one degree of freedom controller with a first order filter on the derivative action. There is an additional tuning parameter, N , which assumes integer values.

5.1.3 Results

The experiments that follow used the reference signal presented in figure 5.3. Initially, the control performance was evaluated in simulation. The IAE of the control error, $y(t) - r(t)$ is presented in figure 5.4 for the three controllers designed using equations 5.7–5.10. In these, and subsequent figures, the x-axis represents the controller design philosophy that was applied, with design 1 corresponding to equation 5.7 and design 4 to equation 5.10. The y-axis represents the IAE criterion. The IAE values

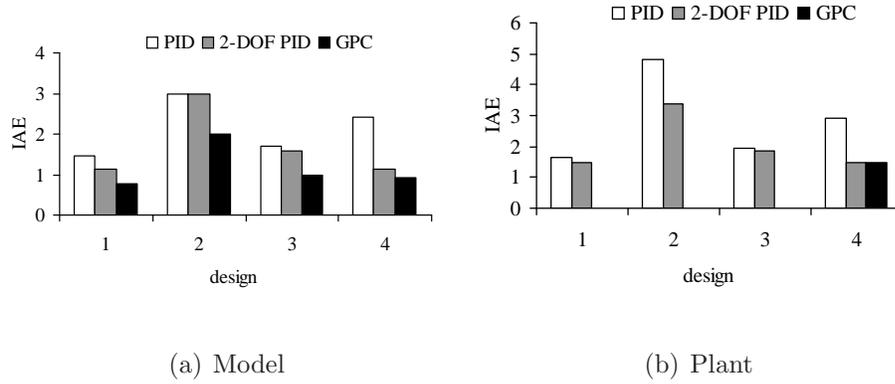


Figure 5.4 IAE of the control error

are listed in appendix B (page 130). In figure 5.4(a) the simulation results are presented, while figure 5.4(b) shows results obtained from the real-time system. Similarly, figure 5.5 illustrates the IAE of the control signal in simulation and real-time. The real-time experiments were repeated three times and the average values presented.

The following sections evaluate these results from two perspectives (i) the choice of objective function and (ii) the controller structure that was used.

Controller design strategy

It can be concluded from the simulation results presented in figure 5.4(a) that the objective function J_2 is not suited to this application. The IAE of the robust design, J_2 , is more than twice that for J_1 , while approximately the same amount of energy is required to move the tip, figure 5.5. The other three designs give similar results and

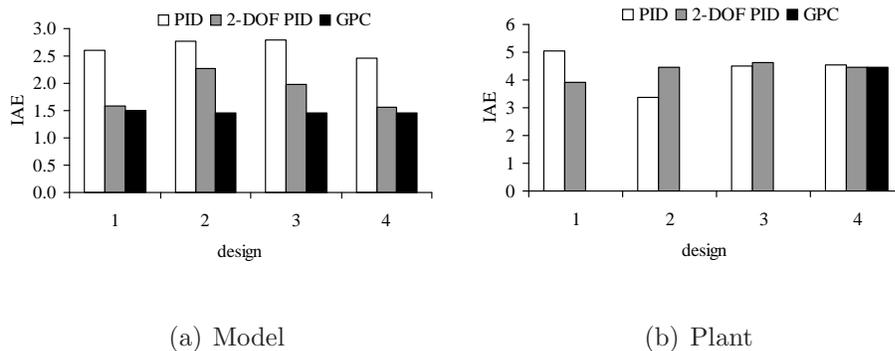


Figure 5.5 IAE of the control signal

could work properly, when applied to the real-time system. When the controllers were applied to the flexible link, figure 5.4(b), a significantly sluggish servo response was noted for the second design. From the IAE of the control signal it is clear that each of the designs utilised a similar amount of energy and, in this context, neither design is superior.

Controller evaluation

Figure 5.4(a) indicates that the GPC controller outperforms the other two controllers. However, when the GPC controller was applied to the flexible link, figure 5.4(b), it did not work for the first three designs. This is due to the high frequency gain of the controller which amplified existing measurement noise and resulted in unstable performance. This extreme sensitivity to measurement noise was evident in the real-time application where the motor, and consequently the link, was vigorously perturbed by control action. Over a very short time period the amplitude of this response reached the limits of the physical system and the experiment was aborted. The origins of the problem are evident from figure 5.6. Consider the controller design based on the nominal model, equation 5.2, using the GA optimisation based on a minimum gain margin of $6dB$ and a minimum phase margin of 45° .

Figure 5.6 illustrates the Bode magnitude response of both the input sensitivity function (above) and the controller transfer function (below). It is well appreciated that the high-frequency gain of the ideal PID controller is a significant factor that limits applications of this controller yet, from figure 5.6(b) note that the GPC design has a high-frequency gain ($\sim 60dB$ at $\omega = 300rad/s$) that is almost 100 times that of the ideal PID controller ($\sim 20dB$ at $\omega = 300rad/s$). Recall also that the input sensitivity function is one measure of how disturbances affect the control signal i.e. with reference to figure 2.4 it defines the relationship $U(z)/D(z)$ and thus how a disturbance, such as unmeasurable noise, will affect the control signal. From figure 5.6(a) it is evident that the GPC design results in a much larger input sensitivity function (by a factor of eight at $\omega = 300rad/s$) than the PID controller and hence accounts for the poor performance of the GPC controller in practice. A similar problem occurs for the other two designs

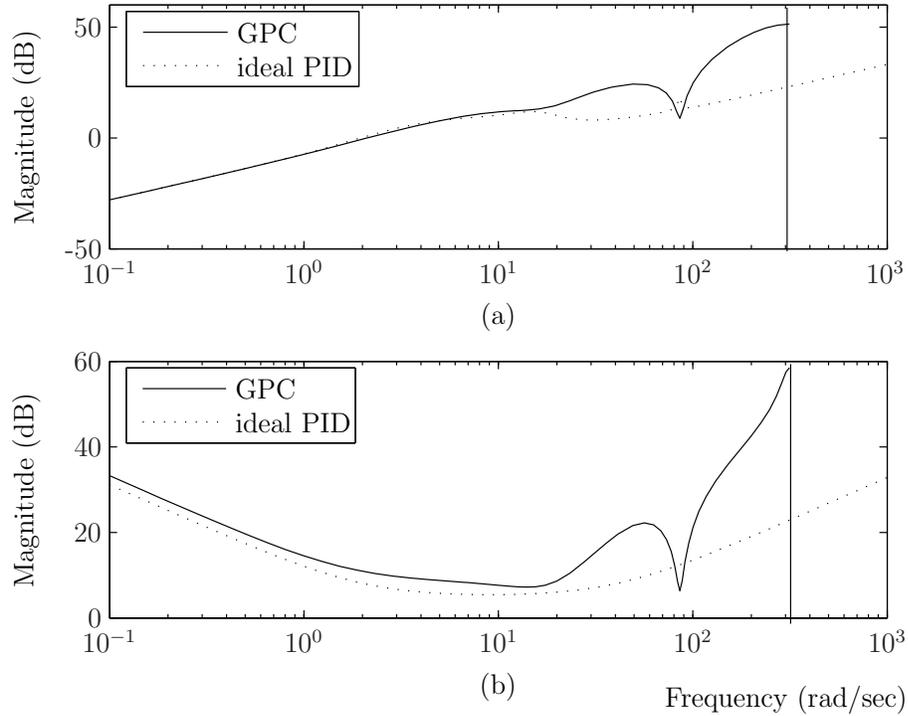


Figure 5.6 Bode magnitude plots of (a) input sensitivity function \mathcal{U} and (b) controller transfer function resulting from the GA optimisation based on $A_m = 6dB$ and $\phi_m = 45^\circ$ and is documented in terms of the parameter, M_u , in equations B.6 and B.9 of appendix B (compare, for example, with equations B.5 and B.8).

Examining the magnitude response of the complementary sensitivity function, figure 5.7, reveals that the GPC design has a larger bandwidth than the ideal PID controller but, more significantly, that the magnitude response of the GPC design increases to a maximum of 3dB's at $\omega = 300rad/s$. The shape of the complementary sensitivity function is known to influence a system's robustness to model uncertainty. For example, Doyle's stability criterion (Doyle, 1984) gives the result —

$$|\Delta_L(j\omega)| \cdot |T(j\omega)| < 1 \quad \forall \omega \in \mathfrak{R} \quad (5.12)$$

where $\Delta_L(j\omega)$ represents the plant–model mismatch as multiplicative uncertainty, defined as

$$\Delta_L(j\omega) = \frac{|G(j\omega) - G_o(j\omega)|}{|G_o(j\omega)|}; \quad (5.13)$$

G being the actual process and G_o the nominal model. The implications of equation 5.12

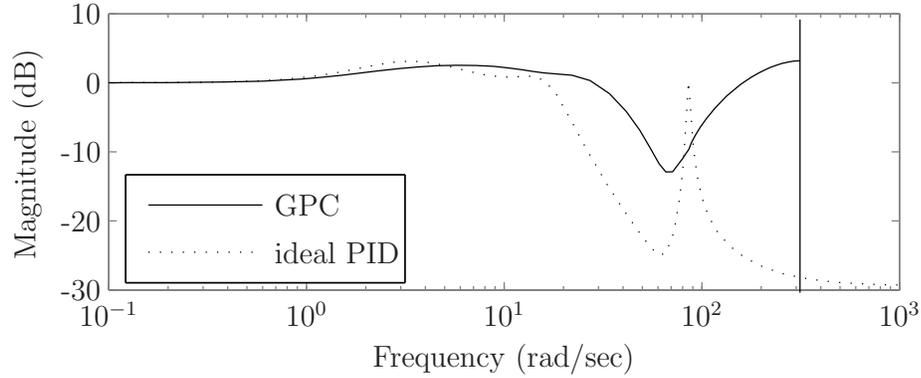


Figure 5.7 Complementary sensitivity function resulting from the GA optimisation based on $A_m = 6dB$ and $\phi_m = 45^\circ$

are that if, as is commonly the case, $\Delta_L(j\omega)$ is characterised by high-gain at high frequency then T must conversely have low gain at high frequencies. This is clearly not the case for the GPC design, see figure 5.7, and therefore the GPC design will be very sensitive to modelling inaccuracies and this would also account for the poor performance of the controller in practice. In contrast, note that the ideal PID controller results in an adequately-shaped complementary sensitivity function, though the spike at $\omega \approx 100rad/s$ is undesirable (the nominal model has a resonance at this frequency).

In summary then, the GA optimisation yielded a $T(z^{-1})$ polynomial that gave satisfactory deterministic disturbance rejection properties (as measured by the IAE). However, only the fourth design, which directly penalised the peak amplitude response of the input sensitivity function, yielded adequate performance in the presence of measurement noise and unmodelled dynamics. The other two candidate controllers provided stable closed-loop performance for each of the designs. Examining the performance of these controllers it is noticeable that the 2-DOF PID structure performs better than the ideal PID algorithm in terms of the IAE of the control error.

A selection of the best closed-loop responses are illustrated in figure 5.8, where the dashed line represents the response of the ideal PID controller, the dotted line denotes the 2-DOF PID controller and the solid line the GPC controller. The ideal PID controller and the 2-DOF PID controller were tuned using the first design, J_1 . The IAE of the PID servo response is $IAE_{yp} = 1.65$. The 2-DOF PID controller performs with

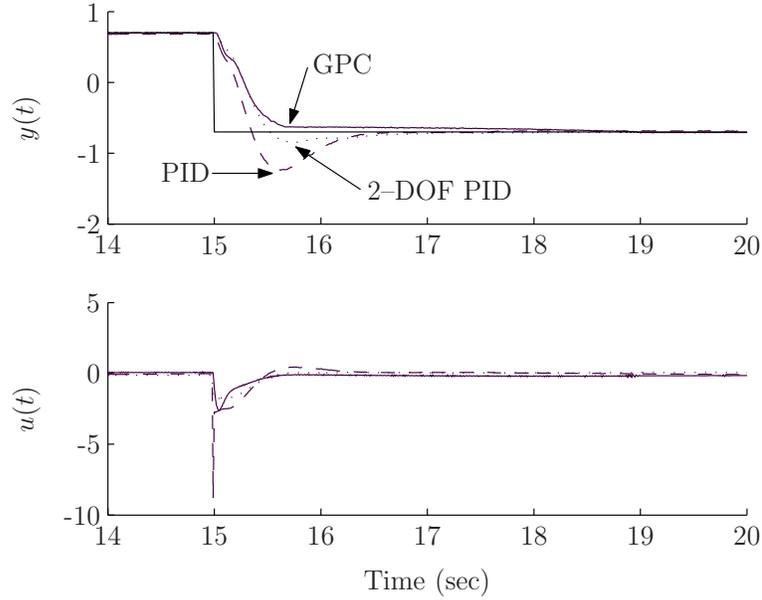


Figure 5.8 Set-point following of the flexible link

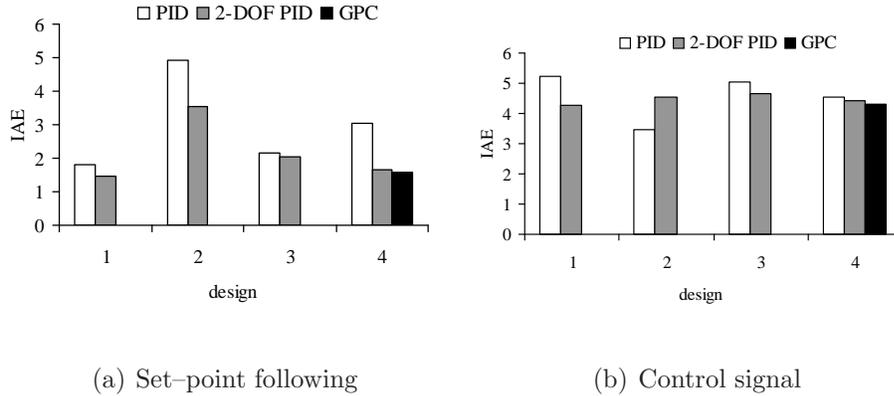


Figure 5.9 IAE of the average plant uncertainty

$IAE_{yp} = 1.47$, while the GPC controller was tuned using the fourth objective function and performs with $IAE_{yp} = 1.46$. The 2-DOF controllers perform slightly better than the one degree of freedom controller.

Plant uncertainty

As was mentioned in the introduction, uncertainty may be incorporated by changing the position of the mass on the flexible link. The previous controller designs were based on the nominal model (half-loaded link), however four other configurations are also

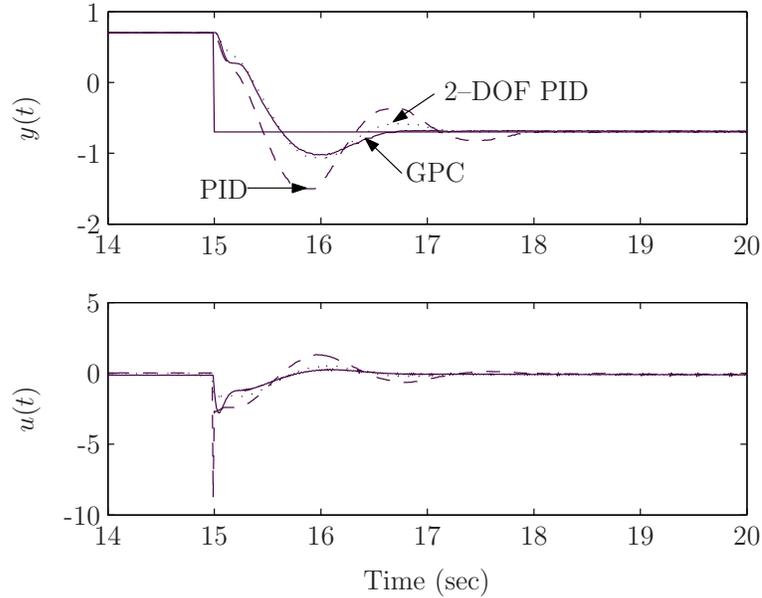


Figure 5.10 Set-point following of the full-loaded flexible link

possible. This section examines the performance of (i) the chosen objective function and (ii) the controller structures when uncertainty is incorporated.

In figure 5.9 the results for model uncertainty are presented. The IAE value represents the mean IAE of the five experiments performed on the flexible link. The five experiments corresponded to a) no load b) quarter-loaded c) half-loaded (nominal) d) three quarter-loaded and e) fully-loaded link. Likewise, 5.9(b) represents the average integrated absolute value of the control signal. Clearly, the controllers that satisfied the nominal model also worked well over the range of load variations that were considered. The average IAE in figure 5.9(a) is larger than that of figure 5.4(b) by 6%.

In general, the same trends are evident in that design two was overly robust and design four yielded a stable closed-loop response for all controllers. Likewise, the PID and 2-DOF PID controllers provided stable performance in all cases while the GPC only worked for design four. The ideal PID and 2-DOF PID controllers provide comparable performance for design one and three while the 2-DOF PID and GPC controllers provide almost identical performance for design four.

In all cases it was found that the worst case (stable) response occurred when the link was fully loaded. Figure 5.10 presents a typical response, where the PID controllers were tuned using the first design and the GPC controller the fourth. The dashed line

presents the response of ideal PID controller, the dotted line that of the 2-DOF PID controller and the GPC controller is denoted using the solid line. As shown, the GPC and 2-DOF PID display similar levels of performance, while the ideal PID controller has significant overshoot and the initial control signal is very aggressive.

5.1.4 Discussion

The application to the flexible link demonstrates that the 2-DOF PID controller provides a better level of performance than the ideal PID controller structure. This statement is also supported by the extensive simulation study conducted in chapter 4. These results only serve to support what is commonly known in the control community. In some cases, such as second design of figure 5.9, a trade-off between performance and control activity occurs but, in the majority, the 2-DOF controller performs better on both counts.

The gain and phase margin criteria worked very well for both PID controllers. This is especially true for the case where the minimum robustness specification were $A_m = 6dB$ and $\phi_m = 45^\circ$. In contrast, the generalised predictive controller did not yield adequate performance, in practice, when tuned using gain and phase margin criteria. It is the author's contention that this difference may be attributed to the large number of tuning parameters associated with the GPC. For example, it is frequently possible to satisfy minimum A_m and ϕ_m specifications using the (primarily) servo parameters, N_1 , N_2 , N_u and λ (or some subset of these) and with the T-polynomial set to unity. This may result in a controller transfer function that displays very high gain at high frequencies and is therefore sensitive to measurement noise and mismodelling effects. In essence, satisfaction of classical robustness indices does not guarantee that the GPC will perform well in practice. Specifically, even with adequate gain and phase margins the controller may display very high gain at high frequencies.

Alternatively, it is possible to think of the GPC optimisation as an over-determined problem — with an excess of tuning parameters and inconsistent results. For example, it is possible to satisfy minimum A_m and ϕ_m specifications using the (primarily) servo parameters, N_1 , N_2 , N_u and λ (or some subset of these) while maintaining the T-polynomial fixed or by fixing the servo parameters and optimising the T-polynomial. In

general, a unique solution does not exist and re-running the same optimisation problem will return a different set of coefficients that yield a similar optimum. Thus, the GPC is, in a sense, disadvantaged by the large number of tuning parameters associated with this controller.

In the GPC case, a thorough understanding of the controller, and the effect of its tuning parameters is required if the controller is to be optimised in practice. Furthermore, relatively sophisticated designs appear to be required, e.g. input sensitivity function. Provide both of these issues are addressed, the controller performance can be optimised but this application would suggest that the effort might be in vain. Figures 5.4(a) and 5.4(b) demonstrate that the 2-DOF PID controller yields practically identical performance on the flexible link. Figure 5.7, which evaluates the performance over a range of operating points, indicates that the 2-DOF PID is as robust to uncertainty as the GPC. In both cases the 2-DOF has an important advantage — the simple A_m, ϕ_m - based designs perform admirably. In the GPC case the same design yielded an unstable closed-loop.

Chapter 6

Conclusions and future research

6.1 Conclusions

Just as academicians do not agree on which control algorithms and tuning techniques are the best, industrial practitioners also have very different views on the state of the art in process control technology. Model-Based Predictive Control has been developing since the late seventies and during this time many different implementations have evolved. Indeed, the ideal PID controller also has many different realisations and hundreds of tuning techniques have been developed. In this scenario, any engineer who is going to employ a controller to a SISO system could be disorientated. Hence, the main effort of this thesis has been to evaluate the Generalised Predictive Controller, when applied to the SISO system, and to compare its performance with that resulting from the other two PID controllers. Performance was compared over a range of benchmark process transfer functions where the controller parameters were tuned using the same objective function. This allows the controllers to be compared like-with-like. The design philosophy was based on minimising performance (IAE) subject to specified minimum constraints on robustness (gain/phase margin, modulus margin, etc.). Finally the controllers were evaluated on a real-time laboratory-scale process.

A general conclusion is that, for the specified design criteria, the GPC clearly outperformed the two PID controllers in simulation. As with every comparison a trade-off exists. In this case, the GPC performed better than the ideal PID controller but, on

the other hand, it could be argued that the ideal PID controller displayed better robustness properties. In general, the gain margins associated with PID controller were considerably larger than the specified minimum values. In contrast, in the GPC designs the minimum gain margin was satisfied with little additional margin. This point is debatable however, as an examination of the phase margin criterion will elucidate. This clear superiority of the GPC controller did not translate into real-time performance. Of the real-time experiments performed, the GPC algorithm failed to stabilise the system for three out of the four designs. This is in clear contrast to the other two designs which were very readily translated into real-time performance. Reasons for this failure were identified. This study would suggest that a significant amount of prior knowledge is required for a successful GPC implementation. This, coupled with the complexity of the algorithm, the significant number of tuning parameters and the lack of established tuning techniques would suggest that the GPC algorithm has little merit for SISO process loops. In contrast, the industrial 2-DOF PID controller is a relatively simple algorithm with well-understood tuning parameters and the design philosophy translates directly into real-time results. The 2-DOF PID controller resulted in a significant reduction in measured IAE — over 12% — relative to the ideal PID controller. Considering that the PI controller is the current norm in the process industry it is clear that “advanced” control algorithms do offer real benefits — however this study would suggest that complexity beyond the 2-DOF PID controller may be of little advantage.

Of the four design philosophies, the design based on the input sensitivity function worked best, in that it realised a successful design for all three controller structures. The approach has the advantage in that it directly addresses closed-loop robustness and the sensitivity to high-frequency noise — which was not addressed in the other three designs. However, the design suffers from the disadvantage that there are few guidelines to aid the choice of a suitable upper bound for this function. A value of 6dB was chosen for the real-time application, but different processes would have different requirements and alternative values may need to be found. The input sensitivity methodology is less suited to the 1-DOF PID controller, primarily due to the nature of the controller which exhibits high-gain at high-frequencies. Consequently, the real-time results were less impressive

and the design based on a minimum gain margin of 6dB reduced the measured IAE by approximately 75% relative to the design based on the input sensitivity function.

The optimisation based on minimum gain and phase margin criteria has the advantage that recommendations for practical minimum gain and phase margins abound. The design based on a 6dB gain margin worked particularly well for the ideal PID controller while the experimental IAE value returned for the 2-DOF PID controller was practically identical to that achieved using the input sensitivity function. However, the design based on the 14dB specification was less appropriate and resulted in a sluggish response on the real-time system and in the simulation study also. Thus, for simple controllers a design based on the classical gain and phase margin criteria is suitable, while a consideration of the input sensitivity function does offer an advantage for more complex structures; the modulus margin criteria would not appear to have any significant benefit as a controller design paradigm.

Though the use of a genetic algorithm for controller design is not novel, the approach adopted in this thesis, i.e. minimising performance subject to constraints on robustness criteria, is particularly true for the designs that combine the IAE with the modulus margin and the IAE with constraints on the input sensitivity function. The use of penalty functions to direct the search for optimal controller coefficients would appear to be a novel and effective method of realising the trade-off between performance and robustness while the use of a look up table to reduce the computational complexity adds further novelty to this work. It is especially useful when the calculation of the objective function is time consuming. However, if the objective function consisted of a simple mathematical equation, the use of a look up table may increase the overall computation time. For the designs presented in this thesis a reduction in computation time resulted in all cases — especially with the GPC optimisation.

Given the ready availability of personal computers, and the flexibility that genetic algorithms offer, this approach offers a very real and practical alternative to traditional controller tuning approaches. A notable advantage of the technique is that the tuning strategy is completely open and can be easily tailored to accommodate additional requirements if, and when, they arise. For example, the tutorial given by Fleming (2004)

demonstrated a GA where the objective function was extended to incorporate settling time, rise time and percent overshoot. However, as with all “black-box” approaches caution must be advised. The GA design critically depends on the specified objective function and the “best” objective function remains elusive. Furthermore, in many cases engineering knowledge will be required to place carefully chosen constraints on the parameters to be tuned. This is particularly evident with the GPC approach where the variables N_1 , N_2 , λ and the T -polynomial all need careful selection if a feasible, practical and stable design is to be achieved.

Finally, it was found that the performance of the GPC controller significantly depended on the chosen sampling period. From the guidelines presented in section 2.5.1 it was found that, in general, methods based on the frequency analysis were more suitable. The method based on the settling time is very misleading for oscillatory systems while the rise time method is very intuitive and suitable for stable systems.

This thesis has only been able to touch on the most general features of the controller comparison. It is obvious that if one of the controllers was “the best”, then the others would not be applied in industry. Most industrial processes are SISO which can be well controlled by PID controllers, (Åström and Hägglund, 2001) hence, extra controller complexity is not recommended. However, a trade-off always exists and a particular controller must be chosen based on the circumstances of a particular application. Based on this assumption, the controller tuning should be done with suitable constraints on the control signal and the robustness parameters.

6.2 Summary of contributions

The main contributions of this thesis are that:

- Three controller structures were evaluated on a wide variety of simulation transfer functions and on a real-time system. The results of the evaluation indicated that the generalised predictive controller did not realise its potential in real-time and that, as an advanced control algorithm, the 2-DOF PID was a more practical alternative.

- A number of novel controller design philosophies were proposed and evaluated. Each of the design paradigms were based on optimising performance subject to constraints on robustness criteria. In general, it was found that a robustness constraint based on the input sensitivity function was most practical, however, for the 1-DOF PID structure classical gain and phase margin criteria were superior.
- The controller parameters were optimised using a genetic algorithm. A novel penalty function was incorporated to direct the GA while a look up table was implemented to accelerate execution time.

6.3 Future research directions

This study is limited by the fact that the controllers were evaluated on a single system. While the dynamics of this system are not trivial (integrating, badly damped, inverse unstable) the system does not display any appreciable time delay. In order to validate the work that has been carried out, an evaluation on additional real-time systems is needed. In particular, a performance evaluation on a system with long time delay would be particularly illuminating as predictive controllers would be expected to outperform PID controllers in such applications. A further limitation of the study is that only the servo response of the flexible link was investigated (as the system is fundamentally a servo tracking problem). Many methods presented in the current literature focus on the load disturbance properties rather than the set-point following, see the review by O'Dwyer (2003) or the recent paper by Leva (2005). A real-time evaluation of the load disturbance properties of these controllers would complete the analysis conducted in this thesis.

Further research could focus on the design based on the input sensitivity, which worked particularly well in practice. It would be very interesting to apply this tuning technique to the benchmark test and compare with the results obtained for the other methods, unfortunately time constraints did not permit this to be achieved. In particular, it would be interesting to determine if there is any upper bound which suits all of the benchmark models and if practical recommendations can be specified. The

author's experience would suggest that such a value does not exist, however a relation may be found. An alternative is to improve the first three designs and directly include the aggressiveness of the control signal. This could be achieved by adding measurement noise to the output of the nominal model in the simulation study and augmenting the objective function to, perhaps, include weighting on the integrated value of the control signal. This design with penalised control signal could be particularly suitable for the GPC controller. However, it would not be trivial to choose a suitable weighting penalty on the control signal as this would vary dramatically depending on the particular application and this is the major problem with this.

Recent work in the genetic algorithm domain (Śmierzchalski and Michalewicz, 2000; Grosman and Lewin, 2002) has focused on evolution programming, also called genetic programming, rather than the simple genetic algorithm. Many researchers are also advocating the use of a multi-objective GA (da Fonseca, 1995; Herreros et al., 2002). As future work, it may be worth employing a multi-objective GA or a hybrid multi-objective genetic programming, instead of the penalty factors developed in this thesis. This would complicate the optimisation tool but the benefit is that faster optimisation would be obtained. It is the author's opinion that as long as the objective function remains the same identical results should be obtained. Hence, this is not a critical point for future research.

Bibliography

- Åström, K. J. (1987). Adaptive feedback control. *Proceedings of the IEEE*, 75(2):185–217.
- Åström, K. J. (1990). *Computer Controlled Systems — theory and design*. Prentice–Hall.
- Åström, K. J. and Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5):645–651.
- Åström, K. J. and Hägglund, T. (1995). *PID controllers: theory, design, and tuning*. Instrument Society of America, 2nd edition.
- Åström, K. J. and Hägglund, T. (2000). Benchmark systems for PID control. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 165–166, Terrassa, Spain.
- Åström, K. J. and Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9(11):1163–1175.
- Åström, K. J., Hägglund, T., Hang, C., and Ho., W. (1993). Automatic tuning and adaptation for PID controllers — a survey. *Control Engineering Practice*, 1(4):699–714.
- Åström, K. J., Panagopoulos, H., and Hägglund, T. (1998). Design of PI controllers based on non-convex optimization. *Automatica*, 34(5):585–601.

- Burden, A., Deshpande, P., and Watters, J. (2001). Advanced process control of a B-9 Permasep permeator desalination pilot plant. *Desalination*, 133:271–283.
- Camacho, E. F. (1993). Constrained generalized predictive control. *IEEE Transactions on Automatic Control*, 38(2):327–332.
- Camacho, E. F. and Bordons, C. (1995). *Model Predictive Control in the process industry*. ISBN 3-540-19924-1. Springer – Verlag London Limited.
- Cao, Y. J. and Wu, Q. H. (1999). Teaching genetic algorithm using MATLAB. *International Journal of Electrical Engineering Education*, 36:139–153.
- Chen, B. and Cheng, Y. (1998). A structure specified \mathcal{H}_∞ optimal control design for practical applications: a genetic approach. *IEEE Transactions on Control Systems Technology*, 6(6):707 – 718.
- Chen, B.-S., Cheng, Y.-M., and Lee, C.-H. (1995). A genetic approach to mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal PID control. *IEEE Control Systems Magazine*, 15(5):51–60.
- Cheng, S.-L. and Hwang, C. (1998). Designing PID controllers with a minimum IAE criterion by a differential evolution algorithm. *Chemical Engineering Communications*, 170:83–115.
- Chipperfield, A., Fleming, P., Pohlheim, H., and Fonseca, C. (1994). *Genetic Algorithm Toolbox user's guide*. Department of Automatic Control and Systems Engineering, University of Sheffield, 1.2 edition.
- Chu, S.-Y. and Teng, C.-C. (1999). Tuning of PID controllers based on gain and phase margin specifications using fuzzy neural network. *Fuzzy Sets and Systems*, 101:21–30.
- Clarke, D. and Gawthrop, P. (1979). Self-tuning control. *Control Theory and Applications*, 126(6):633–640.
- Clarke, D. W. (1988). Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine*, 8(8):49–55.

- Clarke, D. W., Mohtadi, C., and Tuffs, P. (1987a). Generalized predictive control — Part I the basic algorithm. *Automatica*, 23(2):137–148.
- Clarke, D. W., Mohtadi, C., and Tuffs, P. (1987b). Generalized predictive control — Part II extensions and interpretations. *Automatica*, 23(2):149–160.
- Cluett, W. and Wang, L. (1997). New tuning rules for PID control. *Pulp and paper Canada*, 3(6):52–55.
- Corriou, J.-P. (2004). *Process Control Theory and Applications*. Number 1-85233-776-1. Springer Verlag.
- Cunningham, P. (2003). Multiple model control with intelligent switching to ensure robustness to process variations. Master’s thesis, Cork Institute of Technology, Ireland, Cork, Ireland.
- Cutler, C. and Ramaker, B. (1980). Dynamic matrix control - a computer control algorithm. In *Proceedings Joint Automatic Control Conference*, Paper WP5-B, San Francisco.
- da Fonseca, C. M. M. (1995). *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*. PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield.
- de Madrid, A., Santos, M., Dormido, S., and Morilla, F. (1994). *Constrained generalized predictive control with dynamic programming*. Advances in Model-based predictive control. Oxford University Press.
- de Moura Oliveira, P. and Jones, A. (2000). Co-evolutionary design of PID control structures. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 179–187, Terrassa, Spain.
- Demircioglu, H. and Karasu, E. (2000). Generalised predictive control. A practical application and comparison of discrete- and continuous-time versions. *IEEE Control Systems Magazine*, 20(5):36–47.

- Demirciođlu, H. (1994). *Continuous-time generalised predictive control (CGPC): Implementation Issues*. Advances in Model-based predictive control. Oxford University Press.
- Demirioglu, H. and Gawthrop, P. (1991). Continuous-time generalised predictive control (CGPC). *Automatica*, 27(1):55–75.
- Doyle, J. (1984). Onr/honeywell workshop on advances in multivariable control, minneapolis, minn. Lecture notes.
- Fikar, M. and Draeger, A. (1995). Adaptive predictive control of a neutralization reactor. In *Preprints of the 10th Conference Process Control*, volume 1, pages 153–156, Matliare, Slovakia.
- Fleming, P. (2004). Evolutionary computing extends the application domain for practical optimisation. In *The Irish Signals and Systems Conference*, Queen’s University Belfast, Northern Ireland.
- Franklin, G. F., Powell, J. D., and Baeini, A. E. (1986). *Feedback Control of dynamics systems*. Addison–Wesley, Reading, MA.
- Fung, H.-W., Wang, Q.-G., and Lee, T.-H. (1998). PI tuning in terms of gain and phase margins. *Automatica*, 34(9):1145–1149.
- García, C. E. and Morari, M. (1982). Internal model control I. A unifying view and some new results. *Ind. Eng. Chem. Process. Des. Dev.*, 21:308–323.
- García, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice — a survey. *Automatica*, 25(3):335–348.
- Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical optimisation*. New York: Academic Press.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation and machine learning*. Addison Wesley Publishing Company.

- Grimble, M. and Ordys, A. (2001). Predictive control for industrial applications. *Annual Reviews in Control*, 25:13–24.
- Grosman, B. and Lewin, D. R. (2002). Automated nonlinear model predictive control using genetic programming. *Computers and Chemical Engineering*, 26:631–640.
- Herreros, A., Baeyens, E., and Perán, J. (2000). Design of PID controllers using multiobjective genetic algorithms. In *IFAC Workshop on Digital Control: Past, Present and Future of PID Control*, pages 277–282, Terrassa, Spain.
- Herreros, A., Baeyens, E., and Perán, J. R. (2002). MRCD: a genetic algorithm for multiobjective robust control design. *Engineering Applications of Artificial Intelligence*, 15(3-4):285–301.
- Ho, W., Lee, T., and Tay, E. (1998a). Knowledge-based multivariable PID control. *Control Engineering Practice*, 6(7):855–864.
- Ho, W., Lim, K., Hang, C., and Ni, L. (1999). Getting more phase margin and performance out of PID controllers. *Automatica*, 35:1579–1585.
- Ho, W., Lim, K., and Xu, W. (1998b). Optimal gain and phase margin tuning for PID controllers. *Automatica*, 34(8):1009–1014.
- Ho, W. K., Hang, C. C., and Cao, L. S. (1995). Tuning of PID controllers based on gain and phase margin specifications. *Automatica*, 31(3):497–502.
- Ho, W. K. and Xu, W. (1998). PID tuning for unstable processes based on gain and phase-margin specifications. *IEE Proceedings, Control Theory & Applications*, 145(5):392–396.
- Huang, Y. J. and Wang, Y.-J. (2001). Robust PID controller for non-minimum phase time delay systems. *ISA Transactions*, 40:31–39.
- Hwang, C. and Hsiao, C.-Y. (2002). Solution of a non-convex optimization arising in PI/PID control design. *Automatica*, 38(11):1895–1904.

- Hwang, W. and Thompson, W. (1993). An intelligent controller design based on genetic algorithms. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 1266 –1267.
- Isermann, R. (1980). Practical aspects of process identification. *Automatica*, 16:575–587.
- Isermann, R. (1981). *Digital control systems*. Springer–Verlag, Heidelberg.
- Jolly, T. and Bentsman, J. (1993). Generalised predictive control with dynamic filtering for process control applications. In *Proc. American Control Conf.*, pages 1741– 1745, San Francisco, California, USA,.
- Jones, A., Ajlouni, N., and Uzam, M. (1996). Online frequency domain identification and genetic tuning of PID controllers. In *IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 261 –266.
- Josi, N., Murugan, P., and Rinehart, R. (1997). Experimental comparisons of control strategies. *Control Engineering Practice*, 5(7):885–896.
- Kalman, R. E. (1960a). Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana*, 5:102–119.
- Kalman, R. E. (1960b). A new approach to linear filtering and prediction problems. *Transactions of ASME, Journal of Basic Engineering*, 87:35–45.
- Kawabe, T. and Tagami, T. (1997). A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom. In *Proceedings of the 1997 IEEE International Symposium on Intelligent Control*, pages 119 – 124, Istanbul Turkey.
- Keel, L. and Bhattacharyya, S. (1997). Robust, fragile, or optimal? *IEEE Trans. Automatic Control*, 42(8):1098 – 1105.
- Kim, D. H. (2003). Comparison of PID controller tuning of power plant using immune and genetic algorithms. In *IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, pages 169 – 174, Dept. of Instrum. & Control Eng., Hanbat Nat. Univ., Daejeon, South Korea.

- Krohling, R. and Rey, J. (2001). Design of optimal disturbance rejection PID controllers using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 5(1):78 – 82.
- Krohling, R. A. (1997). Design of PID controller for disturbance rejection: a genetic optimization approach. In *Proc. 2nd IEE/IEEE International Conference Genetic Algorithms In Engineering Systems: Innovations And Applications*, page 498503, Glasgow, Scotland.
- Krohling, R. A., Jaschek, H., and Rey, J. P. (1997). Designing PI/PID controllers for a motion control system based on genetic algorithms. In *International Symposium on Intelligent Control*, pages 125–130, Istanbul, Turkey. Proceedings of the 12th IEEE.
- Lagunas, R., Martínez-García, J. C., Soria-López, A., and Fernández-Anaya, G. (2003). Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ based PID control via genetic algorithms: An experimental evaluation. In *European Control Conference*. University of Cambridge, IEEE Control Systems Society.
- Lambert, M. (1987). *Adaptive Control of Flexible Systems*. PhD thesis, Oxford University.
- Landau, I., Lozano, R., and M'Saad, M. (1998). *Adaptive control*. Springer-Verlag London Limited.
- Lee, C.-H. and Teng, C.-C. (2002). Tuning of PID controllers for unstable process based on gain and phase margin specifications: a fuzzy neural approach. *Fuzzy Sets and Systems*, 128:95–106.
- Leonard, F. (1997). An IAE optimum control tuned by one parameter. In *European Control Conference*, Brussels, Belgium.
- Leva, A. (2005). Autotuning process controller with improved load disturbance rejection. *Journal of Process Control*, 15(2).
- Lieslehto, J. (2001). PID controller tuning using evolutionary programming. In *American Control Conference*, volume 4, pages 2828–2833.

- Lima, J. and Ruano, A. (January 2000). Neuro-genetic PID autotuning: time invariant case. *Mathematics and Computers in Simulation*, 51:287–300.
- Liu, G. and Daley, S. (1999a). Design and implementation of an adaptive predictive controller for combustor no_x emissions. *Journal of Process Control*, 9:485–491.
- Liu, G. P. and Daley, S. (1999b). Optimal-tuning PID controller design in the frequency domain with application to a rotary hydraulic system. *Control Engineering Practice*, 7(7):817–926.
- Mathworks (2004a). Control system toolbox manual.
- Mathworks, M. (2004b). Learning simulink. available on Mathworks webpage www.mathworks.com/academia/student_version/learnsimulink.pdf.
- McIntosh, A., Shah, S., and Fisher, D. (1989). Experimental evaluation of adaptive control in the presence of disturbances and model-plant mismatch. In Shah, S. and Dumont, G., editors, *Adaptive Control Strategies for Industrial Use*, pages 145–172. Springer-Verlag.
- McIntosh, A., Shah, S., and Fisher, D. (February 1991). Analysis of adaptive generalized predictive control. *The Canadian Journal of Chemical Engineering*, 69:97–110.
- Mitsukura, Y., Yamamoto, T., and Kaneda, M. (1997). A genetic tuning algorithm of PID parameters. In *IEEE International Conference on Computational Cybernetics and Simulation*, volume 1, pages 923 –928.
- Mitsukura, Y., Yamamoto, T., and Kaneda, M. (1999). A design of self-tuning PID controllers using a genetic algorithm. In *American Control Conference*, volume 2, pages 1361 – 1365.
- Mohtadi, C. (1988). On the role of pre-filtering in parameter estimation and control. In *Workshop on Adaptive Control Strategies for Industrial Use*, Banff, Canada.
- Mohtadi, C. and Clarke, D. W. (1986). Generalised predictive control, LQ, or pole-placement. In *Proceedings on 25th conference on decision and Control*, pages 1536–1541, Athens, Greece. IEEE.

- Mwembeshi, M. M., Kent, C. A., and Salhi, S. (2004). A genetic algorithm based approach to intelligent modelling and control of pH in reactors. *Computers & Chemical Engineering*, 28(9):1743–1757.
- Neto, C. A. and Embirucu, M. (2000). Tuning of PID controllers: an optimisation-based method. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 367–372, Terrassa, Spain.
- Normey-Rico, J. and Camacho, E. (1999). Robustness effects of a prefilter in a Smith predictor — based generalised predictive controller. *IEE Proceedings, Control Theory & Applications*, 146(2):179–185.
- O’Dwyer, A. (2000a). PI and PID controller tuning rules for time delay processes: a summary. Available on-line 12.08.04, <http://citeseer.nj.nec.com/dwyer00pi.html>.
- O’Dwyer, A. (2000b). A summary of PI and PID controller tuning rules for process with time delay. Part I: PI controller tuning rules. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 159–164, Terrassa, Spain.
- O’Dwyer, A. (2000c). A summary of PI and PID controller tuning rules for process with time delay. Part I: PID controller tuning rules. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 211–216, Terrassa, Spain.
- O’Dwyer, A. (2003). *Handbook of PI and PID Controller Tuning Rules*. ISBN 1-86094-342-X. Imperial College Press.
- Ogata, K. (1990). *Modern Control Engineering*. Prentice-Hall, 2nd edition.
- O’Mahony, T. (2002). *Robust generalised predictive control — an optimal design for uncertain systems*. PhD thesis, Cork Institute of Technology.
- O’Mahony, T. and Downing, C. (1999). PID & GPC: comparative study. In *Irish Signals and Systems Conference*, pages 61–68, Galway.
- O’Mahony, T. and Downing, C. (2000a). Computer aided control system design - goldmine or minefield. In Fagan, A. and O’Feely, editors, *Proceedings of the Irish Signals and Systems Conference*, pages 450–457, Dublin.

- O'Mahony, T. and Downing, C. (2000b). CPP, LQG or GPC? In *Proceedings 5th Symposium on Quantitative Feedback Theory and Robust Frequency Domain Methods*, pages 261–268, Public University of Navarre, Pamplona, Spain.
- O'Mahony, T. and Downing, C. (2000c). Structurally stable synthesis of the generalised predictive controller. In Fagan, A. and O'Feely, editors, *Proceedings of the Irish Signals and Systems Conference*, pages 281–288, Dublin.
- O'Mahony, T., Downing, C., and Fatla, K. (2000). Genetic algorithms for PID parameter optimisation: minimising error criteria. In *Proceedings of the Process Control and Instrumentation*, pages 148–153, Glasgow, Scotland.
- Onnen, C., Babuška, R., Kaymak, U., Sousa, J., Verbruggen, H., and Isermann, R. (1997). Genetic algorithm for optimization in predictive control. *Control Engineering Practice*, 5(10):1363–1372.
- Ordys, A. W., Stoustrup, J., and Smillie, I. (2000). Comparisons of \mathcal{H}_∞ control and generalised predictive control for a laser scanner system. *Proceedings of the IMechE*, 214(1):283–297.
- Ota, T. and Omatu, S. (1996). Tuning of the PID control gains by GA. In *IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 272–274, Kauai, HI, USA.
- Owens, D. and Warwick, K. (1988). Extended predictive control. In *Proc. IEE Int. Conf. CONTROL*, Oxford, England.
- Persson, P. (1992). *Towards Autonomous PID Control*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden.
- Persson, P. and Åström, K. J. (1992). Dominant pole design — a unified view of PID controller tuning. In Dugard, L., M'Saad, M., and Landau, I., editors, *IFAC Adaptive Systems in Control and Signal Processing*, pages 377–382, Grenoble, France.
- Phillips, C. and Nagle, H. (1990). *Digital control system analysis and design*. Prentice-Hall International Editions, second edition.

- Porter, B. and Hicks, D. (1995). Genetic design of unconstrained digital PID controllers. In *Proceedings of the National Aerospace and Electronics Conference, NAECON*, volume 1, pages 478–485, Dayton, OH, USA.
- Porter, B. and Hicks, D. L. (1993). Genetic design of adaptive digital model-following flight-mode control systems. In *Navigation and control conference*, Monterey, USA.
- Porter, B. and Jones, A. (1992). Genetic tuning of digital PID controllers. *Electronics letters*, 28(9):843–844.
- Poulin, E. and Pomerleau, A. (1997). Unified PID design method based on a maximum peak resonance specification. *IEE Proceedings, Control Theory & Applications*, 144(6):566–574.
- Poulin, E. and Pomerleau, A. (1999). PI settings for integrating processes based on ultimate cycle information. *IEEE Transactions on Control Systems Technology*, 7(4):509–511.
- Qin, S. and Badgwell, T. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764.
- Rani, K. and Unbehauen, H. (1997). Study of predictive controller tuning methods. *Automatica*, 33(12):2243–2248.
- Rensburg, P. V., Shaw, I., and Wyk, J. V. (1998). Adaptive PID control using a genetic algorithm. In *Second International Conference on Knowledge-Based Intelligent Electronic Systems*, volume 2, pages 133–138, Adelaide, SA Australia.
- Richalet, J., Rault, A., Testud, J., and Papon, J. (1976). Algorithmic control of industrial processes. In *Proceedings 4th IFAC Symposium on Identification and System Parametric Estimation*, pages 1119–1167.
- Robinson, B. and Clarke, D. W. (1991). Robustness effects of a prefilter in generalized predictive control. *IEE Proceedings D*, 138(1):2–8.

- Ronco, E., Gawthrop, P. J., and Hill, D. J. (1999). A practical continuous-time non-linear generalised predictive controller. Technical report ee-99001, Electrical and Information Engineering School, The University of Sydney, NSW 2006, Australia.
- Schei, T. S. (1994). Automatic tuning of PID controllers based on transfer function estimation. *Automatica*, 30(12):1983–1989.
- Schmitt, L. M. (2001). Fundamental study theory of genetic algorithms. *Theoretical Computer Science*, 259:1–61.
- Seborg, D., Edgar, T., and Mellichamp, D. (1989). *Process Dynamics and Control*. John Wiley & Sons, New York.
- Shafiei, Z. and Shenton, T. (1997). Frequency-domain design of PID controllers for stable and unstable systems with time delay. *Automatica*, 33(12):2223–2232.
- Shatiei, Z. and Shenton, T. (1994). Tuning of PID-type controllers for stable and unstable systems with time delay. *Automatica*, 30:1609–1615.
- Shen, J.-C. (2001). New tuning method for PID controller. In *Conference on Control Applications*, pages 459–464, Mexico City.
- Shinskey, F. G. (1990). How good are our controllers in absolute performance and robustness? *Measurement and Control*, 23:114–121.
- Śmierzchalski, R. and Michalewicz, Z. (2000). Modeling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Transactions on Evolutionary computation*, 4(3):227–241.
- Smith, O. (1957). Close control of loops with dead time. *Chemical Engineering Progress*, 53(5):217–219.
- Sree, R. P., Srinivas, M. N., and Chidambaram, M. (2004). A simple method of tuning PID controllers for stable and unstable FOPTD systems. *Computers and Chemical Engineering*. Article in press, Available online 12 August 2004, <http://dx.doi.org/10.1016/j.compchemeng.2004.04.004>.

- Takahashi, R., Peres, P., and Ferreira, P. (1997). Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ guaranteed cost PID design. *Control Systems Magazine*, 17(5):37–47.
- Visioli, A. (2000). Fuzzy logic based tuning of PID controllers for plants with under-damped response. In *IFAC Workshop on Digital Control: Past Present and Future of PID Control*, pages 577–582, Terrassa, Spain.
- Vlachos, C., Williams, D., and Gomm, J. (1999). Genetic approach to decentralised PI controller tuning for multivariable processes. *IEE Proceedings — Control Theory and Applications*, 146(1):58–64.
- Wang, P. and Kwok, D. (1994). Optimal design of PID process controllers based on genetic algorithms. *Control Engineering Practice*, 2(4):641–648.
- Wang, Q.-G., Fung, H.-W., and Zhang, Y. (1999). PID tuning with exact gain and phase margin. *ISA Transactions*, 38:243–249.
- Wang, Q.-G., Hang, C., Yang, Y., and He, J. (2002). Quantitative robust stability analysis and PID controller design. *IEE Proceedings, Control Theory & Applications*, 149(1):3–7.
- Wang, Y.-G. and Shao, H.-H. (2000). Optimal tuning for PI controller. *Automatica*, 36:147–152.
- Warwick, K. and Kang, Y. H. (1996). Relationship between deadbeat and pole-placement discrete-time PID controllers. *Transactions of the Institute of Measurement and Control*, 18(5):262–266.
- Xu, J.-X., Hang, C.-C., and Liu, C. (2000). Parallel structure and tuning of a fuzzy PID controller. *Automatica*, 36:673–684.
- Yaniv, O. and Nagurka, M. (2004). Design of PID controllers satisfying gain margin and sensitivity constraints on a set of plants. *Automatica*, 40(1):111–116.
- Zakian, V. and Al-Naib, U. (1973). Design of dynamical and control systems by the method of inequalities. *IEE Proceedings*, 120(11):1421–1427.

- Zalzala, A. and Fleming, P. (1997). *Genetic Algorithms in engineering systems*. The Institution of Electrical Engineers, London.
- Zhu, Y. (2001). *Multivariable system identification for process control*. Number ISBN 0-08-043985-3. Pergamon Press.
- Zhuang, M. and Atherton, D. (1993). Automatic tuning of optimum PID controllers. *IEE Proceedings D*, 140(3):216–224.
- Ziegler, J. G. and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions ASME*, 64:759–768.

Appendix A

A.1 Objective function incorporating GPM

A.1.1 The ideal PID controller

No of system	K_p	K_i	K_d	IAE_{both}	M_p (abs)	M_s (abs)	A_m (dB)	ϕ_m (deg)	DM (sec)	MM (abs)	Disturbance			Step response			
											T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
Optimised IAE_{both} with constraints $A_m \geq 6dB$ and $\phi_m \geq 45^\circ$																	
$G_1(s)$	6.78	2.63	5.98	2.08	1.32	1.64	∞	44	0.36	0.61	5.9	0.69	0.39	1.7	1	15	8.3
$G_2(s)$	18.3	20	1.66	0.409	1.33	1.53	22.70	45	0.08	0.65	1.1	0.00	0.05	0.36	0.19	25	2.1
$G_3(s)$	0.446	0.0507	1.23	42.9	1.00	1.95	6.27	64	20.66	0.51	75	3.24	20	23	10	14	49
$G_4(s)$	1.87	0.659	2.95	5.46	1.33	1.63	∞	44	0.55	0.61	14	11.41	2	3.5	1.5	36	16
$G_5(s)$	0.605	0.226	0.463	10.7	1.13	2.08	5.94	58	3.98	0.48	15	41.26	5.7	4.9	2.1	7	10

No of system	K_p	K_i	K_d	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
					(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
$G_6(s)$	1.9	2.14	0.729	1.44	1.47	2.00	∞	45	0.24	0.50	3.3	4.52	0.52	0.93	0.64	14	4.9
$G_7(s)$	0.418	0.901	0.0146	2.59	1.00	1.52	∞	63	1.05	0.66	5.8	31.89	1.5	1.4	0.52	32	6.1
$G_8(s)$	0.864	0.243	0.397	7.39	1.41	1.66	8.02	47	0.96	0.60	9	1.55	4.3	3.1	0.82	39	9.4
$G_9(s)$	40.2	40.1	0.0915	1.03	1.12	0.99	∞	78	0.33	1.01	0	0.00	0.025	1	0.83	35	4
$G_{10}(s)$	10.9	20	1.2	0.38	1.32	1.57	∞	45	0.07	0.64	0.95	0.05	0.05	0.33	0.16	28	1.5
$G_{11}(s)$	12.4	20	1.38	0.351	1.32	1.59	∞	45	0.07	0.63	1	0.00	0.05	0.3	0.16	21	1.6
$G_{12}(s)$	20	11.8	7.99	1.09	1.71	1.31	∞	45	0.24	0.76	2	0.00	0.085	1	0.56	40	4.6
$G_{13}(s)$	32.5	48.7	6.53	0.746	1.73	1.18	-11.68	50	0.12	0.85	0.79	0.00	0.021	0.72	0.28	65	2.5

Optimised IAE_{both} with constraints $A_m \geq 14dB$ and $\phi_m \geq 45^\circ$

$G_1(s)$	6.59	2.89	5.64	2.05	1.32	1.64	∞	45	0.37	0.61	5.4	1.49	0.36	1.7	1	18	7.7
$G_2(s)$	19.4	29.5	1.9	0.407	1.32	1.54	22.00	45	0.08	0.65	0.77	0.00	0.034	0.37	0.18	31	1.8
$G_3(s)$	0.224	0.039	0.449	53.9	1.00	1.45	14.00	63	28.22	0.69	86	2.34	27	27	21	2.9	70
$G_4(s)$	1.89	0.946	3.84	4.47	1.32	1.63	∞	45	0.47	0.61	17	23.83	1.8	4	1.6	44	17
$G_5(s)$	0.234	0.156	0.224	15.6	1.01	1.53	14.01	60	6.64	0.65	27	37.88	8.2	7.3	5.9	6.1	22
$G_6(s)$	1.88	2.17	0.8	1.47	1.44	1.95	∞	45	0.23	0.51	5.2	6.03	0.52	0.96	0.64	17	4.8
$G_7(s)$	0.419	0.937	0.0293	2.56	1.00	1.43	∞	62	1.08	0.70	5.7	35.61	1.4	1.4	0.54	33	3.7
$G_8(s)$	0.739	0.156	0.195	9.7	1.39	1.62	14.22	44	1.04	0.62	15	0.00	6.6	3.1	0.93	43	12
$G_9(s)$	29.4	24.8	0.0305	1.03	1.14	1.00	∞	77	0.44	1.00	1.1	0.00	0.04	0.99	0.85	28	4.1
$G_{10}(s)$	11.6	20	1.33	0.377	1.32	1.59	∞	45	0.07	0.63	0.97	0.05	0.05	0.33	0.16	23	1.6
$G_{11}(s)$	12.2	20	1.34	0.349	1.32	1.59	∞	45	0.07	0.63	0.99	0.00	0.05	0.3	0.16	21	1.6
$G_{12}(s)$	20	11.8	7.99	1.09	1.71	1.31	∞	45	0.24	0.76	2	0.00	0.085	1	0.56	40	4.6

No of system	K_p	K_i	K_d	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
					(<i>abs</i>)	(<i>abs</i>)	(<i>dB</i>)	(<i>deg</i>)	(<i>sec</i>)	(<i>abs</i>)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
$G_{13}(s)$	32.5	48.7	6.53	0.746	1.73	1.18	-11.68	50	0.12	0.85	0.79	0.00	0.021	0.72	0.28	65	2.5

A.1.2 The 2-DOF PID controller

No of system	K_p	K_i	K_d	b	c	N	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
								(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
Optimised IAE_{both} with constraints $A_m \geq 6dB$ and $\phi_m \geq 45^\circ$																				
$G_1(s)$	4.01	2.32	3.2	0.715	0.999	7	1.62	1.13	1.78	14.64	45	0.53	0.56	4.8	5.74	0.49	1.1	1.2	4.3	6.5
$G_2(s)$	19.1	20	2.01	0.829	0.106	14	0.27	1.00	1.64	15.59	45	0.07	0.61	1.1	0.00	0.05	0.22	0.23	2.4	0.49
$G_3(s)$	0.455	0.0513	1.24	0.999	0.999	29	41	1.01	2.01	6.02	64	21.00	0.50	75	4.70	20	21	11	11	58
$G_4(s)$	11.9	9.84	10.4	0.449	0.999	31	1.4	5.98	6.54	9.24	9	0.05	0.15	2.2	20.32	0.15	1.3	0.92	20	8.8
$G_5(s)$	0.594	0.234	0.425	0.999	0.999	20	9.94	1.18	2.10	5.88	55	3.66	0.48	13	39.96	5.6	4.3	2.6	1	6
$G_6(s)$	1.65	3.31	0.792	0.789	0.997	15	1.15	1.43	2.15	13.34	45	0.25	0.47	4.2	16.15	0.41	0.74	0.61	16	5.5
$G_7(s)$	0.222	0.911	9.62	0.734	0.991	1	2.55	1.00	1.80	7.05	63	1.08	0.56	6.1	35.04	1.5	1.5	0.59	29	5.1
$G_8(s)$	1.06	0.325	0.477	0.676	0.999	13	4.74	1.04	2.08	5.82	44	0.70	0.48	8.4	0.66	3.1	1.6	0.54	34	5.7
$G_9(s)$	20	19.2	0.0525	0.594	0.716	27	0.746	1.02	1.00	∞	69	0.55	1.00	1.6	2.66	0.055	0.69	1.1	6.4	3.7
$G_{10}(s)$	6.36	15.4	0.776	0.648	0.999	4	0.3	1.05	1.78	13.04	45	0.10	0.56	0.95	3.79	0.071	0.23	0.22	4.4	1.3
$G_{11}(s)$	11.1	19.6	1.22	0.731	0.998	31	0.259	1.16	1.64	25.57	44	0.07	0.61	0.98	0.00	0.051	0.21	0.22	1.8	0.37
$G_{12}(s)$	20	11.8	9.07	0.673	0.997	8	0.729	1.41	1.31	∞	45	0.21	0.77	2.1	0.01	0.085	0.64	0.81	2.8	3.5
$G_{13}(s)$	45.2	50	8.29	0.630	0.987	6	0.354	1.27	1.35	-16.42	46	0.08	0.74	0.52	0.00	0.02	0.33	0.37	3.2	0.93

No of system	K_p	K_i	K_d	b	c	N	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
								(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
Optimised IAE_{both} with constraints $A_m \geq 14dB$ and $\phi_m \geq 45^\circ$																				
$G_1(s)$	4.29	2.28	3.62	0.760	0.992	8	1.63	1.19	1.79	14.59	45	0.49	0.56	5.2	4.48	0.49	1.2	1.2	5.4	6.6
$G_2(s)$	18.6	32	1.95	0.674	0.155	19	0.256	1.00	1.60	17.02	45	0.08	0.63	0.75	0.00	0.031	0.22	0.24	1.5	0.4
$G_3(s)$	0.224	0.039	0.468	0.992	0.992	30	53.7	1.00	1.44	14.01	63	28.26	0.69	87	2.38	27	27	24	2.4	69
$G_4(s)$	1.22	0.906	3.06	0.566	0.992	23	3.51	1.26	1.81	14.76	45	0.57	0.55	19	43.77	2.8	2.3	1.4	20	16
$G_5(s)$	0.224	0.166	0.244	0.992	0.992	18	15.6	1.05	1.60	14.02	57	5.98	0.63	27	37.92	8.3	7.3	6.3	8	22
$G_6(s)$	1.66	3.27	0.81	0.798	0.992	17	1.15	1.41	2.11	14.22	45	0.24	0.47	4.2	16.16	0.41	0.74	0.64	16	5.4
$G_7(s)$	0.0976	0.732	0.849	0.992	0.992	1	3.09	1.01	1.49	14.19	61	1.44	0.67	5.3	6.63	1.6	1.6	0.98	6.6	4
$G_8(s)$	0.663	0.146	0.176	0.605	0.992	31	8.93	1.01	1.57	13.99	44	1.15	0.64	12	0.12	6.8	2.1	2.2	3.5	8.6
$G_9(s)$	47.1	49.4	0.299	0.581	0.985	27	0.498	1.00	1.00	∞	81	0.29	1.00	0	4.95	0.024	0.48	0.8	0.14	1.3
$G_{10}(s)$	6.77	9.99	0.702	0.775	0.992	4	0.331	1.12	1.75	13.99	45	0.10	0.57	1.4	0.10	0.1	0.23	0.21	3.4	0.8
$G_{11}(s)$	7.94	9.99	0.839	0.845	0.992	5	0.297	1.19	1.76	14.33	45	0.09	0.57	1.5	0.00	0.1	0.2	0.21	2.6	0.75
$G_{12}(s)$	50	22.3	14	0.767	0.992	4	0.588	1.44	1.23	∞	48	0.15	0.81	0	0.00	0.045	0.54	0.6	2.3	2.1
$G_{13}(s)$	50	35.8	9.18	0.752	0.969	5	0.352	1.31	1.43	-20.37	44	0.07	0.70	0	0.00	0.028	0.32	0.37	1.1	0.59

A.1.3 The GPC controller

No of system	h	N_1	N_2	N_u	λ	$T(z^{-1})$	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response		
	(sec)						(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	
Optimised IAE_{both} with constraints $A_m \geq 6dB$ and $\phi_m \geq 45^\circ$																			
$G_1(s)$	0.1	1	128	16	0.078	$(1 + 0.651z^{-1})(1 + 0.439z^{-1})(1 + 0.245z^{-1})$	1.24	1.00	1.63	9.14	48	0.22	0.62	1.63	12.82	0.04	1.20	1.07	7.9
$G_2(s)$	0.01	8	11	1	$1 \cdot 10^{-6}$	$(1 + 0.801z^{-1})(1 + 0.737z^{-1})(1 - 0.900z^{-1})$	0.208	1.00	1.41	11.30	45	0.02	0.71	0.0	0	0.001	0.2	0.1	5.1
$G_3(s)$	0.70	22	70	1	$1 \cdot 10^{-6}$	$(1 + 0.329z^{-1})(1 - 0.166z^{-1})(1 + 0.445z^{-1})$	33.9	1.00	2.00	6.00	60	2.27	0.50	38.3	0	16	17.9	4.1	0.0
$G_4(s)$	0.01	8	72	1	0.078	$(1 + 0.308z^{-1})(1 + 0.501z^{-1})(1 + 0.590z^{-1})$	0.838	1.02	1.40	11.87	71	0.05	0.71	0.0	15	$\simeq 0$	0.8	0.6	10.6
$G_5(s)$	0.15	8	31	2	$1 \cdot 10^{-6}$	$(1 - 0.898z^{-1})(1 - 0.898z^{-1})(1 - 0.897z^{-1})$	8.79	2.06	2.01	6.00	59	3.88	0.50	14.3	49	5.6	3.2	1.3	0.1
$G_6(s)$	0.10	4	30	6	0.23	$(1 + 0.423z^{-1})(1 + 0.357z^{-1})(1 + 0.395z^{-1})$	0.859	1.00	1.86	6.97	45	0.13	0.54	1.4	14	0.088	0.8	0.7	9.3
$G_7(s)$	0.10	21	69	1	0.47	$(1 - 0.002z^{-1})(1 + 0.028z^{-1})(1 - 0.016z^{-1})$	2.11	1.00	2.00	6.02	60	1.00	0.50	2.1	0	1.1	1.1	0.0	99.0
$G_8(s)$	0.02	50	86	12	6.5	$(1 - 0.896z^{-1})(1 - 0.453z^{-1})(1 - 0.896z^{-1})$	2.83	1.00	4.40	9.33	71	0.12	0.23	3.5	0.6	1.5	1.4	0.5	2.1
$G_9(s)$	0.2	1	163	8	$1 \cdot 10^{-6}$	$(1 - 0.451z^{-1})(1 - 0.435z^{-1})(1 + 0.051z^{-1})$	0.533	1.00	1.99	6.08	45	0.15	0.50	0	0	0.01	0.52	0.18	0.2
$G_{10}(s)$	0.01	1	59	9	0.078	$(1 - 0.996z^{-1})(1 + 0.641z^{-1})(1 + 0.698z^{-1})$	0.195	1.00	1.41	12.65	51	0.03	0.71	0.0	4.5	0.003	0.2	0.2	3.2
$G_{11}(s)$	0.01	9	9	1	$1 \cdot 10^{-6}$	$(1 + 0.648z^{-1})(1 - 0.826z^{-1})(1 + 0.548z^{-1})$	0.166	1.00	1.59	9.26	45	0.02	0.63	0.0	1.8	$\simeq 0$	0.2	0.1	6.0
$G_{12}(s)$	0.01	16	32	1	$1 \cdot 10^{-6}$	$(1 + 0.245z^{-1})(1 - 0.906z^{-1})(1 - 0.962z^{-1})$	0.531	1.00	1.81	7.69	52	0.01	0.55	0.0	0.16	$\simeq 0$	0.5	0.4	5.6
$G_{13}(s)$	0.02	1	55	11	0.079	$(1 + 0.929z^{-1})(1 + 0.608z^{-1})(1 + 0.380z^{-1})$	0.382	1.00	1.99	6.09	62	0.08	0.50	0	3.40	$\simeq 0$	0.38	0.43	1.5

No of system	h	N_1	N_2	N_u	λ	$T(z^{-1})$	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response		
														T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p
Optimised IAE_{both} with constraints $A_m \geq 14dB$ and $\phi_m \geq 45^\circ$																			
$G_1(s)$	0.1	6	95	10	0.078	$(1 - 0.433z^{-1})(1 + 0.751z^{-1})(1 + 0.344z^{-1})$	1.33	1.00	1.49	13.82	43	0.21	0.67	1.97	4.97	0.06	1.27	1.27	3.3
$G_2(s)$	0.01	4	25	2	0.078	$(1 + 0.228z^{-1})(1 + 0.517z^{-1})(1 + 0.899z^{-1})$	0.233	1.00	1.25	14.00	53	0.04	0.80	0.0	9.8	0.0027	0.2	0.2	7.0
$G_3(s)$	0.70	23	23	1	$1 \cdot 10^{-6}$	$(1 + 0.081z^{-1})(1 + 0.161z^{-1})(1 - 0.977z^{-1})$	56	1.00	1.29	13.95	75	55.86	0.77	147.0	0	43	15.9	0.7	11.2
$G_4(s)$	0.01	8	72	1	0.078	$(1 + 0.809z^{-1})(1 + 0.389z^{-1})(1 + 0.074z^{-1})$	0.838	1.02	1.27	14.00	70	0.05	0.78	0.0	15	0.0015	0.8	0.6	10.6
$G_5(s)$	0.15	2	54	4	0.16	$(1 - 0.934z^{-1})(1 - 0.947z^{-1})(1 + 0.017z^{-1})$	12.8	1.02	1.43	14.17	67	8.18	0.70	22.0	40	8.3	4.5	4.2	0.0
$G_6(s)$	0.10	2	46	4	0.23	$(1 + 0.749z^{-1})(1 - 0.762z^{-1})(1 + 0.682z^{-1})$	0.921	1.03	1.57	14.08	45	0.17	0.64	1.7	6.1	0.17	0.8	0.6	11.7
$G_7(s)$	0.10	10	67	2	5.6	$(1 - 0.711z^{-1})(1 - 0.382z^{-1})(1 + 0.608z^{-1})$	2.51	1.00	1.77	14.44	62	1.43	0.56	3.4	0	1.5	1.1	0.3	8.5
$G_8(s)$	0.02	63	122	10	0.16	$(1 - 0.922z^{-1})(1 - 0.739z^{-1})(1 - 0.922z^{-1})$	2.5	1.02	4.16	14.81	∞		0.24	3.7	0.01	1.4	1.1	0.2	2.6
$G_9(s)$	0.2	1	75	8	$1 \cdot 10^{-6}$	$(1 - 0.529z^{-1})(1 - 0.443z^{-1})(1 - 0.443z^{-1})$	0.548	1.00	1.52	14.15	47	0.25	0.66	0.80	0	0.03	0.52	0.18	0
$G_{10}(s)$	0.01	1	59	9	0.078	$(1 - 0.996z^{-1})(1 + 0.854z^{-1})(1 + 0.306z^{-1})$	0.195	1.00	1.25	14.01	50	0.03	0.80	0.0	4.6	$\simeq 0$	0.2	0.2	3.2
$G_{11}(s)$	0.01	3	21	1	0.078	$(1 + 0.882z^{-1})(1 - 0.122z^{-1})(1 + 0.212z^{-1})$	0.189	1.00	1.30	14.01	51	0.03	0.77	0.0	10	$\simeq 0$	0.2	0.2	7.1
$G_{12}(s)$	0.01	16	32	1	$1 \cdot 10^{-6}$	$(1 - 0.713z^{-1})(1 - 0.864z^{-1})(1 - 0.956z^{-1})$	0.531	1.00	1.38	14.06	53	0.02	0.72	0.0	7.8	$\simeq 0$	0.5	0.4	5.6
$G_{13}(s)$	0.02	1	21	1	0.079	$(1 + 0.506z^{-1})(1 + 0.239z^{-1})(1 - 0.020z^{-1})$	0.388	1.00	1.26	14.02	61	0.06	0.79	0	6.77	0	0.38	0.41	6.2

A.2 Objective function incorporating the sensitivity functions

A.2.1 The ideal PID controller

No of system	K_p	K_i	K_d	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
					(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
Optimised IAE_{both} with constraint $MM \geq 0.6$																	
$G_1(s)$	6.98	2.87	5.63	2.01	1.37	1.68	∞	43	0.35	0.60	5.5	0.57	0.35	1.7	0.98	19	7.7
$G_2(s)$	10	10	0.701	0.433	1.28	1.51	28.24	47	0.13	0.66	1.7	0.00	0.1	0.33	0.22	26	1.7
$G_3(s)$	0.419	0.0488	0.968	43.5	1.00	1.67	7.93	64	22.09	0.60	75	2.66	21	23	11	9.1	60
$G_4(s)$	2.12	0.634	3.2	5.02	1.41	1.68	∞	42	0.49	0.59	14	6.47	1.8	3.2	1.5	30	15
$G_5(s)$	0.463	0.197	0.368	12	1.01	1.67	8.27	60	5.03	0.60	15	40.79	6.4	5.5	3.5	3.8	15
$G_6(s)$	1.43	1.65	0.578	1.62	1.08	1.67	∞	90	1.09	0.60	3.6	3.83	0.65	0.97	0.77	12	5.1
$G_7(s)$	0.419	0.937	0.0195	2.61	1.00	1.50	∞	61	1.06	0.67	4.6	35.60	1.5	1.5	0.5	36	7.2
$G_8(s)$	0.685	0.641	3.4	8.85	1.75	1.68	8.10	36	0.89	0.60	0.95	17	5.36	3.91	0.95	53	13.38
$G_9(s)$	34.2	32.2	0.177	1.03	1.14	0.98	∞	78	0.38	1.02	0.62	0.00	0.031	1	0.83	32	4.1
$G_{10}(s)$	10	10	0.859	0.432	1.39	1.67	∞	42	0.08	0.60	1.7	0.04	0.1	0.33	0.16	25	2.3
$G_{11}(s)$	10	10	0.798	0.375	1.39	1.67	∞	42	0.08	0.60	1.7	0.00	0.1	0.28	0.17	24	1.7
$G_{12}(s)$	10	6.31	6.47	1.44	1.51	1.17	-33.19	51	0.33	0.86	3	1.86	0.17	1.3	0.76	38	4.6
$G_{13}(s)$	10	6.72	4.52	1.13	1.51	1.00	-12.09	62	0.24	1.00	2.5	0.13	0.15	0.98	0.51	40	3.3

A.2.2 The 2-DOF PID controller

No of system	K_p	K_i	K_d	b	c	N	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response			
								(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	T_s
Optimised IAE_{both} with constraint $MM \geq 0.6$																				
$G_1(s)$	4.64	2.19	3.3	0.698	0.992	15	1.7	1.08	1.68	20.61	45	0.51	0.60	5.3	1.05	0.47	1.2	1.5	2.4	6.2
$G_2(s)$	8.12	9.99	0.966	0.845	0.992	2	0.337	1.05	1.66	12.90	50	0.14	0.60	1.6	0.00	0.1	0.24	0.24	3.6	0.94
$G_3(s)$	0.42	0.0488	0.927	0.992	0.992	25	42.1	1.00	1.67	7.94	64	22.01	0.60	77	3.19	21	21	12	4.5	59
$G_4(s)$	1.52	0.511	1.98	0.599	0.999	26	3.81	1.19	1.69	22.21	41	0.67	0.59	14	9.87	2.4	1.6	1.8	5.4	9.7
$G_5(s)$	0.458	0.204	0.382	0.999	0.999	30	11.7	1.03	1.67	8.15	58	4.72	0.60	19	40.96	6.4	5.3	4.7	3.1	15
$G_6(s)$	1.15	2.3	0.521	0.915	0.999	16	1.38	1.08	1.68	17.06	66	0.81	0.60	4.8	13.67	0.58	0.81	0.77	14	3.8
$G_7(s)$	0.0195	0.868	9.99	0.008	0.961	19	2.63	1.00	1.64	8.18	63	1.19	0.61	5.2	24.48	1.4	1.4	0.65	21	4.1
$G_8(s)$	0.672	0.205	0.273	0.612	0.985	20	7.962	1.11	1.67	8.12	41	1.05	0.60	0	20	5.73	3.62	1.83	11	9.55
$G_9(s)$	49.2	20.3	0.0122	0.798	0.519	1	0.589	0.99	1.00	∞	86	0.31	1.00	0	0.03	0.049	0.54	0.88	0.004	1.3
$G_{10}(s)$	8.38	10	0.848	0.804	0.999	9	0.33	1.16	1.67	18.42	45	0.09	0.60	1.6	0.06	0.1	0.23	0.22	2	0.79
$G_{11}(s)$	9.66	10	1	0.873	0.998	12	0.296	1.21	1.67	19.53	45	0.08	0.60	1.7	0.00	0.1	0.2	0.21	2.8	0.7
$G_{12}(s)$	10	8.16	6.61	0.581	0.973	5	0.819	1.50	1.67	-20.12	35	0.21	0.60	2.5	5.09	0.14	0.68	0.81	4.5	4.5
$G_{13}(s)$	10	10	4.2	0.560	0.999	4	0.476	1.41	1.63	-10.13	38	0.14	0.61	1.9	4.77	0.11	0.37	0.45	2.8	1

A.2.3 The GPC controller

No of system	h	N_1	N_2	N_u	λ	$T(z^{-1})$	IAE_{both}	M_p	M_s	A_m	ϕ_m	DM	MM	Disturbance			Step response		
	(sec)						(abs)	(abs)	(dB)	(deg)	(sec)	(abs)	T_{rd}	y_d	IAE_{reg}	IAE_{servo}	T_r	y_p	
Optimised IAE_{both} with constraint $MM \geq 0.6$																			
$G_1(s)$	0.1	1	39	5	0.078	$(1 - 0.322z^{-1})(1 + 0.399z^{-1})(1 + 0.187z^{-1})$	1.19	1.00	1.65	11.24	38	0.17	0.61	1.67	3.35	0.04	1.15	1.20	2.1
$G_2(s)$	0.01	8	10	1	$1 \cdot 10^{-6}$	$(1 - 0.708z^{-1})(1 + 0.727z^{-1})(1 + 0.824z^{-1})$	0.207	1.00	1.67	-4.09	-36	0.00	0.60	0.0	5.2	$\simeq 0$	0.2	0.1	5.3
$G_3(s)$	0.70	22	25	2	$1 \cdot 10^{-6}$	$(1 - 0.911z^{-1})(1 + 0.040z^{-1})(1 + 0.054z^{-1})$	37.1	1.00	1.67	7.95	64	23.09	0.60	61.0	0	21	15.9	0.7	6.1
$G_4(s)$	0.01	1	158	16	0.078	$(1 + 0.666z^{-1})(1 + 0.631z^{-1})(1 + 0.630z^{-1})$	0.816	1.00	1.67	-6.48	-38	0.00	0.60	0.0	5.2	$\simeq 0$	0.8	0.7	3.6
$G_5(s)$	0.15	4	40	3	0.078	$(1 - 0.928z^{-1})(1 - 0.557z^{-1})(1 - 0.935z^{-1})$	10.4	1.40	1.67	7.90	65	5.68	0.60	17.9	47	6.6	3.8	2.5	0.0
$G_6(s)$	0.10	2	18	4	0.16	$(1 + 0.399z^{-1})(1 - 0.095z^{-1})(1 + 0.741z^{-1})$	0.775	1.00	1.67	-6.52	37	0.11	0.60	1.2	14	0.074	0.7	0.6	9.0
$G_7(s)$	0.10	10	13	10	$1 \cdot 10^{-6}$	$(1 + 0.410z^{-1})(1 - 0.802z^{-1})(1 + 0.397z^{-1})$	2.45	1.00	1.68	7.97	64	1.46	0.60	3.8	0	1.4	1.1	0.0	39.0
$G_8(s)$	0.02	60	143	1	0.2	$(1 - 0.993z^{-1})(1 + 0.852z^{-1})(1 + 0.677z^{-1})$	10.4	1.00	1.93	6.36	47	1.21	0.52	17.06	0	8.11	2.25	2.74	0.1
$G_9(s)$	0.2	1	106	12	$1 \cdot 10^{-6}$	$(1 - 0.420z^{-1})(1 - 0.396z^{-1})(1 - 0.231z^{-1})$	0.536	1.00	1.67	8.32	45	0.17	0.60	0.40	0	0.01	0.52	0.18	0
$G_{10}(s)$	0.01	1	59	9	0.078	$(1 - 0.996z^{-1})(1 + 0.706z^{-1})(1 + 0.764z^{-1})$	0.195	1.00	1.66	-5.33	-37	0.00	0.60	0.0	4.5	0.003	0.2	0.2	3.2
$G_{11}(s)$	0.01	8	9	1	$1 \cdot 10^{-6}$	$(1 - 0.657z^{-1})(1 + 0.191z^{-1})(1 + 0.841z^{-1})$	0.165	1.00	1.67	-4.29	-37	0.00	0.60	0.0	7.6	$\simeq 0$	0.2	0.1	6.4
$G_{12}(s)$	0.01	16	32	1	$1 \cdot 10^{-6}$	$(1 - 0.318z^{-1})(1 - 0.959z^{-1})(1 - 0.887z^{-1})$	0.531	1.00	1.60	10.08	50	0.01	0.63	0.0	5.4	$\simeq 0$	0.5	0.4	5.6
$G_{13}(s)$	0.02	1	55	11	0.079	$(1 + 0.631z^{-1})(1 + 0.592z^{-1})(1 + 0.475z^{-1})$	0.382	1.00	1.63	8.01	61	0.08	0.62	0.0	3.40	$\simeq 0$	0.38	0.43	1.5

Appendix B, tuning of the flexible link

The following nomenclature is assumed here:

- IAE_{ym} is IAE of control error for the nominal model
- IAE_{yp} is IAE of control error for the plant
- IAE_{um} is IAE of control signal for the nominal model
- IAE_{up} is IAE of control signal for the plant

B.1 Tuning based on GPM specification

Results of tuning controllers based on $\underline{A}_m = 6dB$ and $\underline{\phi}_m = 45^\circ$, for details see equation 5.7.

The ideal PID controller

$$\begin{aligned} K_p &= 1.872 & K_i &= 3.611 & K_d &= 0.044 \\ A_m &= 6.02dB & \phi_m &= 45^\circ & MM &= 0.48 & DM &= 0.15(sec) \\ IAE_{ym} &= 1.45 & IAE_{yp} &= 1.65 & IAE_{um} &= 2.61 & IAE_{up} &= 5.05 \end{aligned} \quad (B.1)$$

The 2-DOF PID controller

$$\begin{aligned} K_p &= 0.064 & K_i &= 2.480 & K_d &= 127 & b &= 0.200 & c &= 0.667 & N &= 26 \\ A_m &= 6.02dB & \phi_m &= 45^\circ & MM &= 0.49 & DM &= 0.16(sec) & M_u &= 15.81dB \\ IAE_{ym} &= 1.13 & IAE_{yp} &= 1.47 & IAE_{um} &= 1.59 & IAE_{up} &= 3.91 \end{aligned} \quad (B.2)$$

The GPC controller

$$\begin{aligned}
N_1 &= 1 & N_2 &= 123 & N_u &= 4 & \lambda &= 0.77 & h &= 0.01 \\
T(z^{-1}) &= (1 - 0.9756z^{-1})(1 - 0.1254z^{-1})(1 - 0.1088z^{-1}) \\
A_m &= 6.02dB & \phi_m &= 44^\circ & MM &= 0.48 & DM &= 0.10(sec) & M_u &= 51.38dB \\
IAE_{ym} &= 0.77 & IAE_{yp} &= \infty & IAE_{um} &= 1.50 & IAE_{up} &= \infty
\end{aligned} \tag{B.3}$$

B.2 Tuning based on GPM specification

Results of tuning controllers based on $\underline{A_m} = 14dB$ and $\underline{\phi_m} = 45^\circ$, for details see equation 5.8.

The ideal PID controller

$$\begin{aligned}
K_p &= 0.749 & K_i &= 1.155 & K_d &= 0.018 \\
A_m &= 13.99dB & \phi_m &= 44^\circ & MM &= 0.75 & DM &= 0.34(sec) \\
IAE_{ym} &= 2.98 & IAE_{yp} &= 4.81 & IAE_{um} &= 2.77 & IAE_{up} &= 3.38
\end{aligned} \tag{B.4}$$

The 2-DOF PID controller

$$\begin{aligned}
K_p &= 0.171 & K_i &= 0.932 & K_d &= 127 & b &= 0.067 & c &= 1.000 & N &= 3 \\
A_m &= 14.02dB & \phi_m &= 45^\circ & MM &= 0.74 & DM &= 0.37(sec) & M_u &= 18.23dB \\
IAE_{ym} &= 2.98 & IAE_{yp} &= 3.37 & IAE_{um} &= 2.28 & IAE_{up} &= 4.44
\end{aligned} \tag{B.5}$$

The GPC controller

$$\begin{aligned}
N_1 &= 4 & N_2 &= 93 & N_u &= 3 & \lambda &= 9.8 & h &= 0.01 \\
T(z^{-1}) &= (1 - 0.9854z^{-1})(1 + 0.3499z^{-1})(1 - 0.1625z^{-1}) \\
A_m &= 13.94dB & \phi_m &= 59^\circ & MM &= 0.74 & DM &= 0.33(sec) & M_u &= 47.91dB \\
IAE_{ym} &= 1.99 & IAE_{yp} &= \infty & IAE_{um} &= 1.46 & IAE_{up} &= \infty
\end{aligned} \tag{B.6}$$

B.3 Tuning based on Modulus Margin specification

Results of tuning controllers based on $\underline{MM} = 0.6$, for details see equation 5.9.

The ideal PID controller

$$\begin{aligned}
K_p &= 1.457 & K_i &= 2.930 & K_d &= 0.035 \\
A_m &= 8.01dB & \phi_m &= 44^\circ & MM &= 0.60 & DM &= 0.19(sec) \\
IAE_{ym} &= 1.71 & IAE_{yp} &= 1.93 & IAE_{um} &= 2.79 & IAE_{up} &= 4.48
\end{aligned} \tag{B.7}$$

The 2-DOF PID controller

$$\begin{aligned}
K_p &= 0.441 & K_i &= 2.516 & K_d &= 127 & b &= 0.067 & c &= 0.867 & N &= 2 \\
A_m &= 8.14dB & \phi_m &= 41^\circ & MM &= 0.59 & DM &= 0.18(sec) & M_u &= 32.86dB \\
IAE_{ym} &= 1.59 & IAE_{yp} &= 1.88 & IAE_{um} &= 1.98 & IAE_{up} &= 4.61
\end{aligned} \tag{B.8}$$

The GPC controller

$$\begin{aligned}
N_1 &= 1 & N_2 &= 31 & N_u &= 1 & \lambda &= 2 & h &= 0.01 \\
T(z^{-1}) &= (1 - 0.9678z^{-1})(1 - 0.2933z^{-1})(1 - 0.1352z^{-1}) \\
A_m &= 8.03dB & \phi_m &= 46^\circ & MM &= 0.58 & DM &= 0.13(sec) & M_u &= 49.24dB \\
IAE_{ym} &= 0.98 & IAE_{yp} &= \infty & IAE_{um} &= 1.46 & IAE_{up} &= \infty
\end{aligned} \tag{B.9}$$

B.4 Tuning based on input sensitivity function

Results of tuning controllers based on $\overline{M}_u = 6dB$, for details see equation 5.10.

The ideal PID controller

$$\begin{aligned}
K_p &= 0.176 & K_i &= 2.128 & K_d &= 1.114 & N &= 1 \\
A_m &= 9.41dB & \phi_m &= 51^\circ & MM &= 0.65 & DM &= 0.29(sec) & M_u &= 6.23dB \\
IAE_{ym} &= 2.43 & IAE_{yp} &= 2.90 & IAE_{um} &= 2.45 & IAE_{up} &= 4.55
\end{aligned} \tag{B.10}$$

The 2-DOF PID controller

$$\begin{aligned}
K_p &= 0.300 & K_i &= 4.957 & K_d &= 255 & b &= 0.056 & c &= 0.020 & N &= 6 \\
A_m &= 3.94dB & \phi_m &= 31^\circ & MM &= 0.35 & DM &= 0.08(sec) & M_u &= 7.03dB \\
IAE_{ym} &= 1.14 & IAE_{yp} &= 1.49 & IAE_{um} &= 1.56 & IAE_{up} &= 4.44
\end{aligned} \tag{B.11}$$

The GPC controller

$$N_1 = 4 \quad N_2 = 101 \quad N_u = 8 \quad \lambda = 0.091 \quad h = 0.01$$

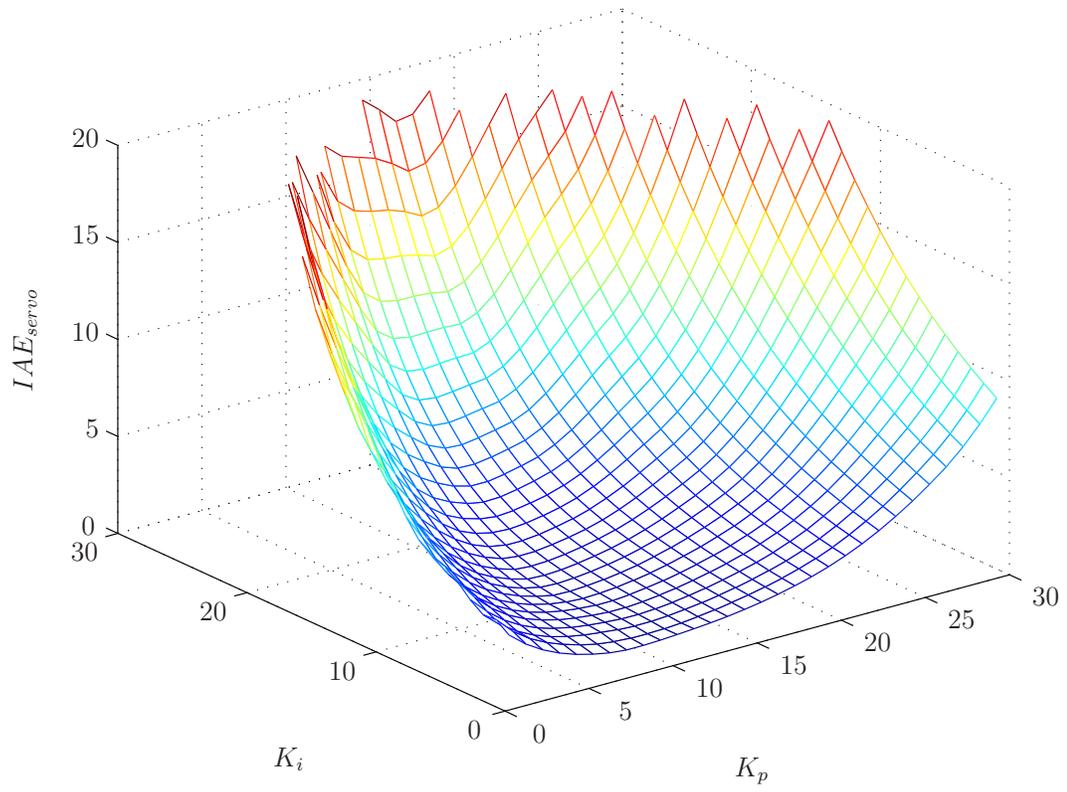
$$T(z^{-1}) = (1 - 0.9688z^{-1})(1 - 0.9551z^{-1})(1 - 0.8410z^{-1})$$

$$A_m = 8.26dB \quad \phi_m = 42^\circ \quad MM = 0.55 \quad DM = 0.17(sec) \quad M_u = 8.39dB$$

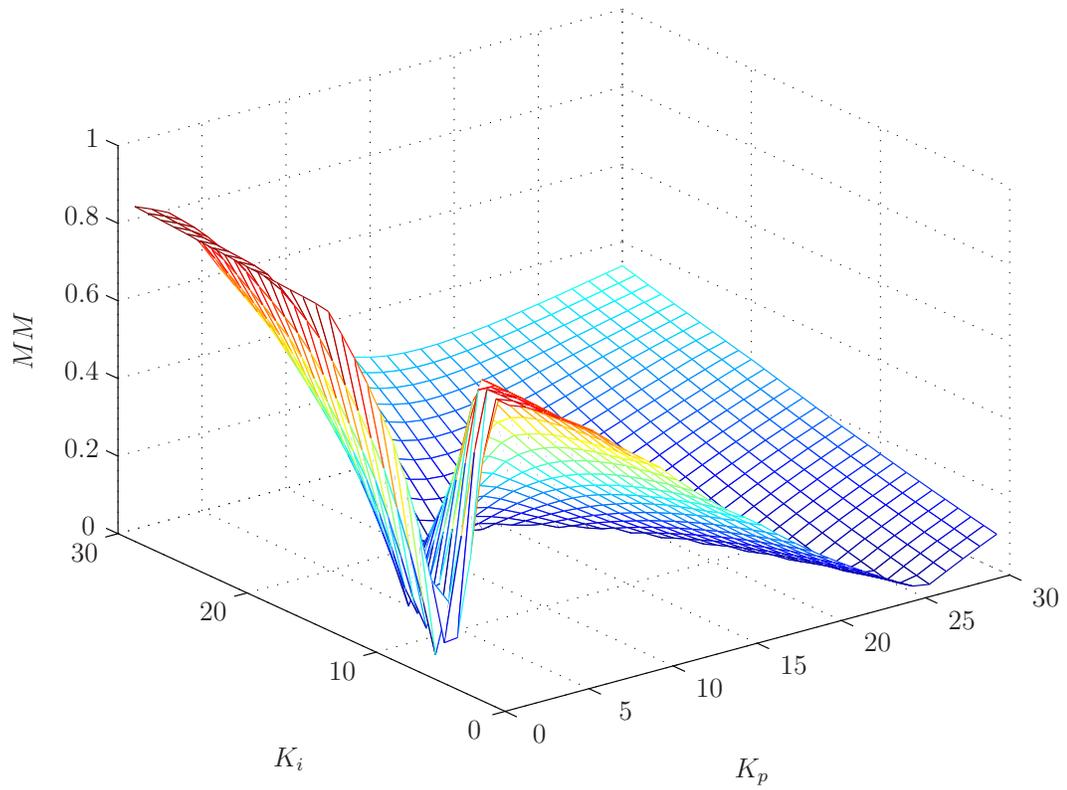
$$IAE_{ym} = 0.93 \quad IAE_{yp} = 1.46 \quad IAE_{um} = 1.46 \quad IAE_{up} = 4.44$$

(B.12)

Appendix C, figures discussed in section 2.6

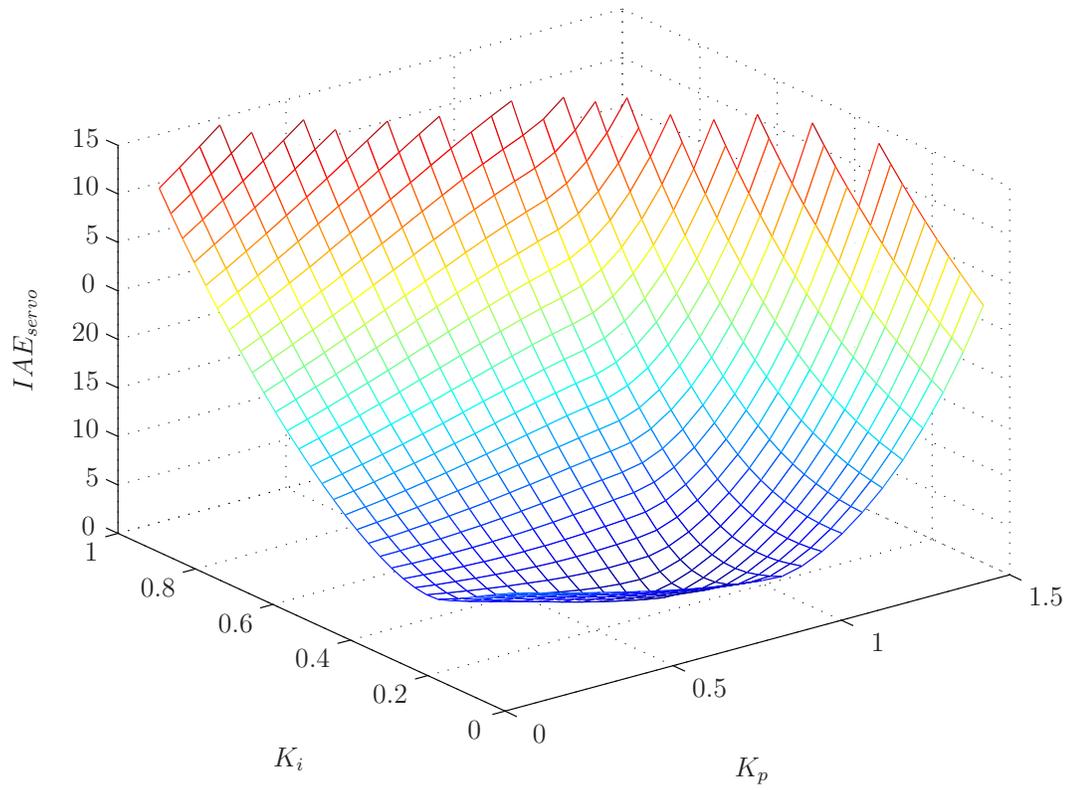


(a) *PID*

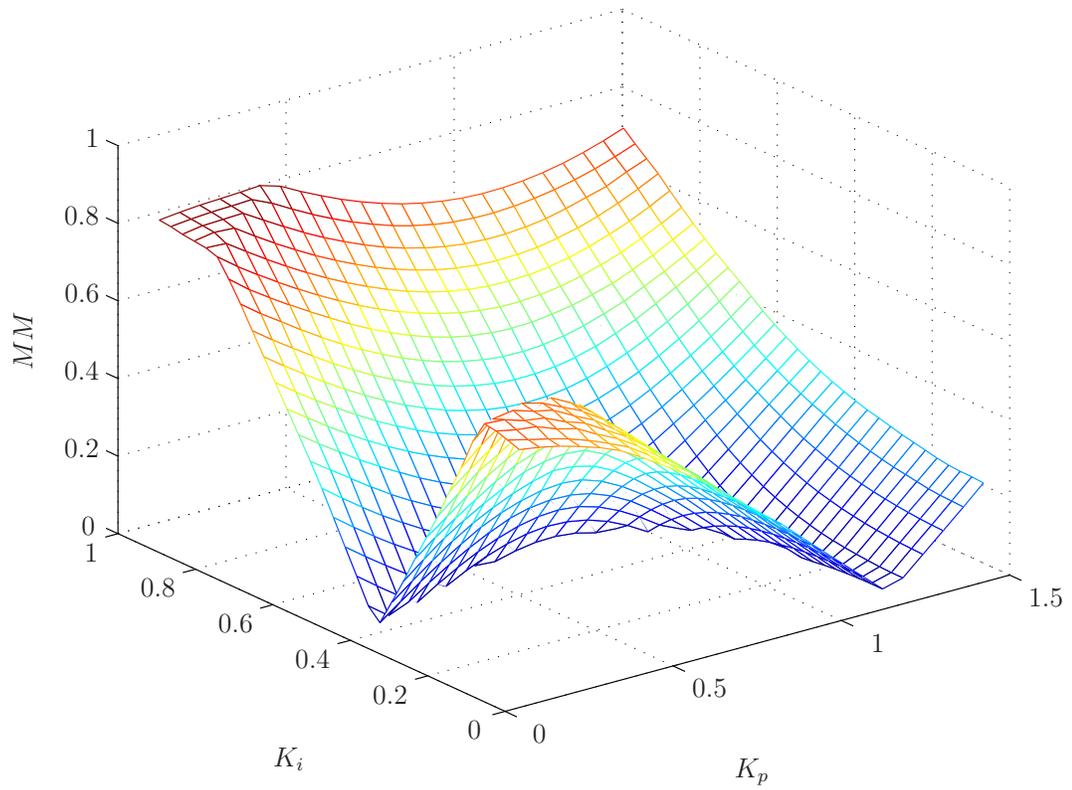


(b) *PID*

Figure C.1 Study of the search area for the system G_1

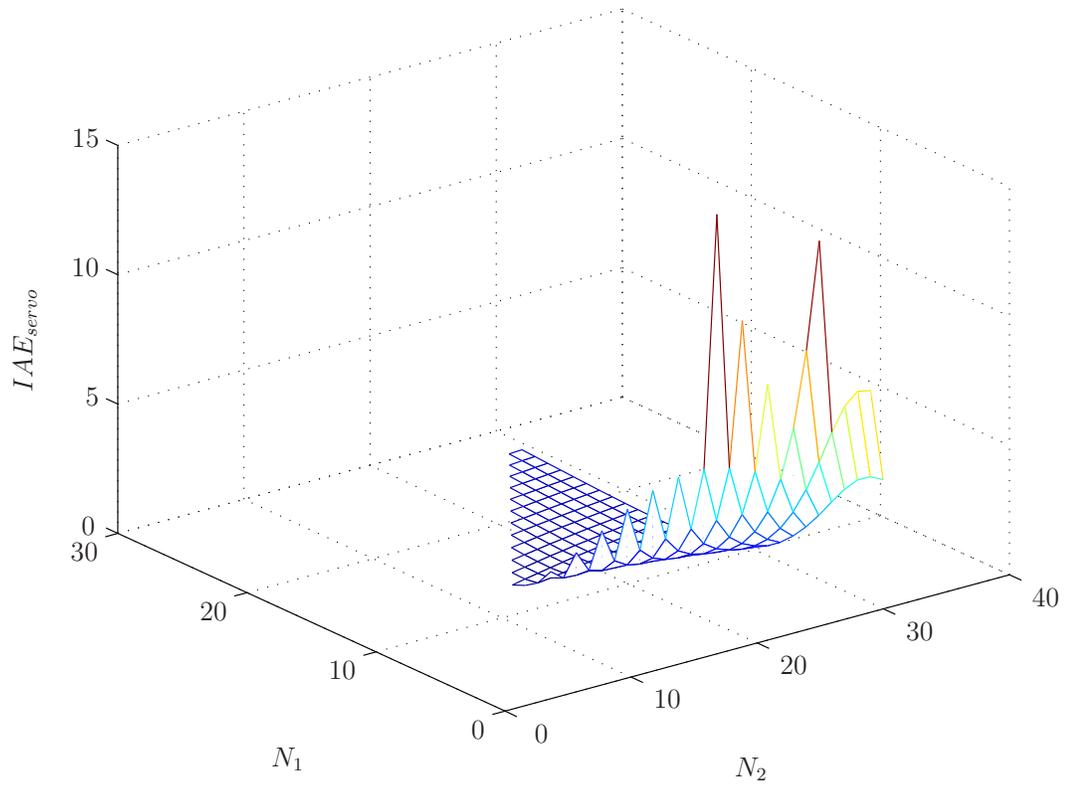


(a) *PID*

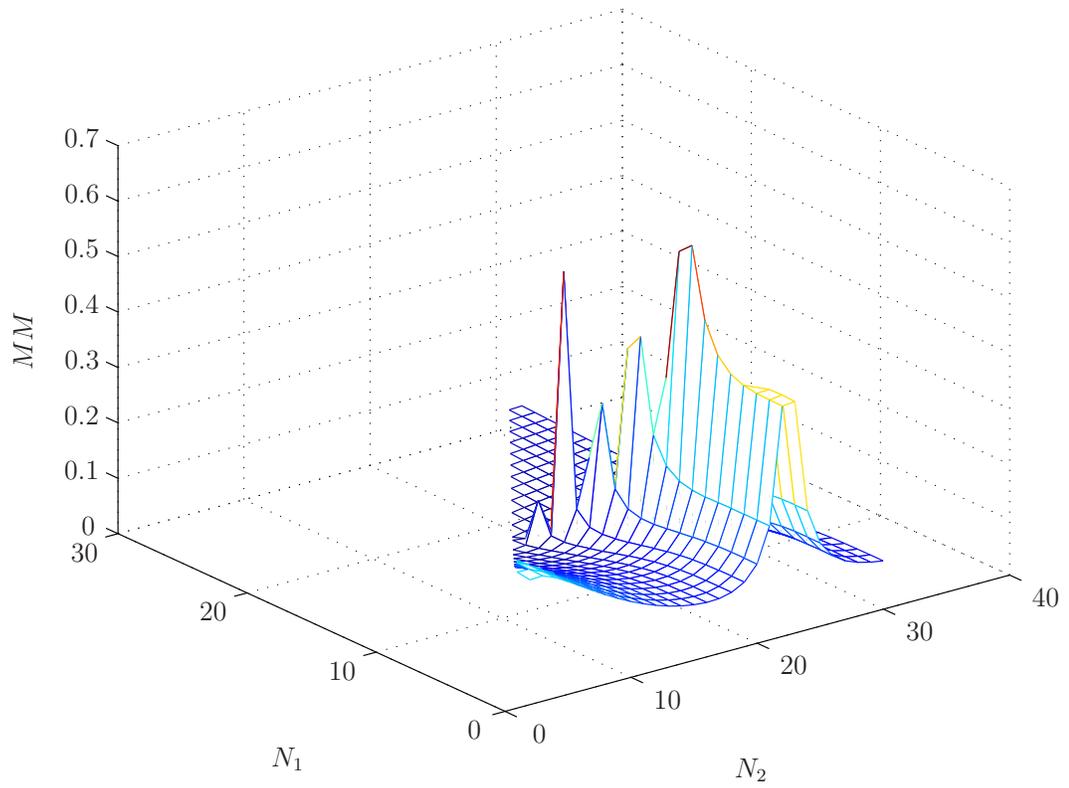


(b) *PID*

Figure C.2 Study of the search area for the system G_5



(a) *GPC*



(b) *GPC*

Figure C.3 Study of the search area for the system G_5

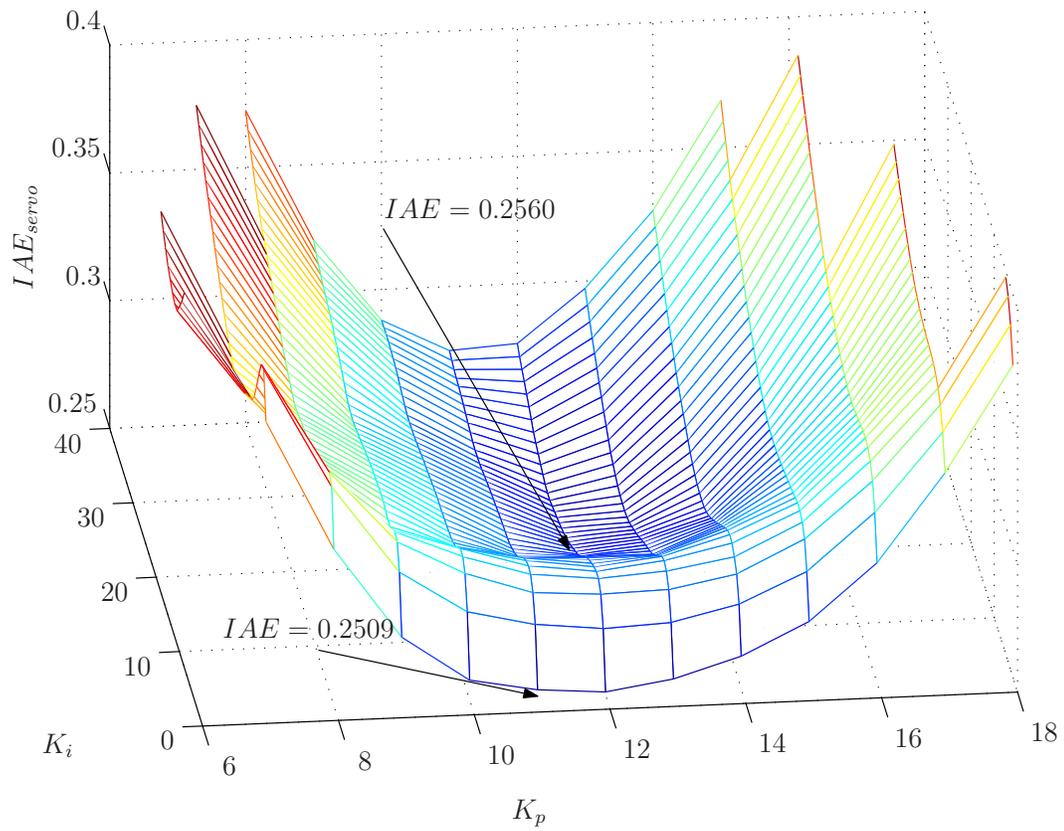
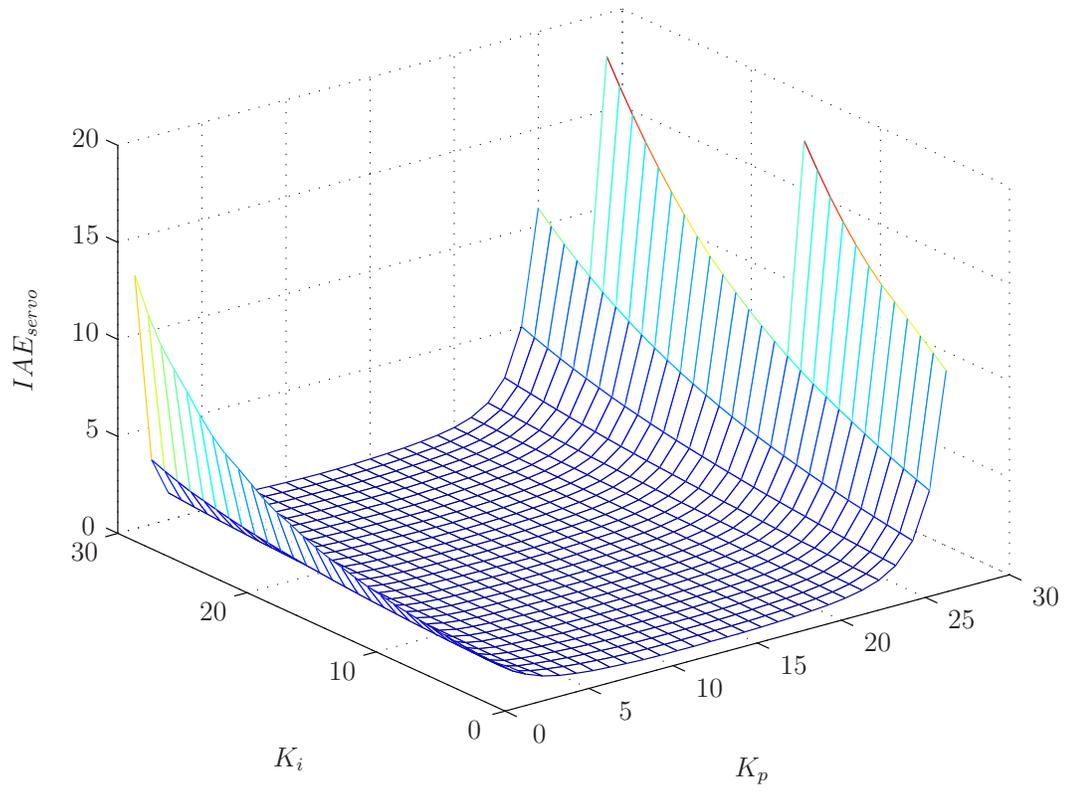
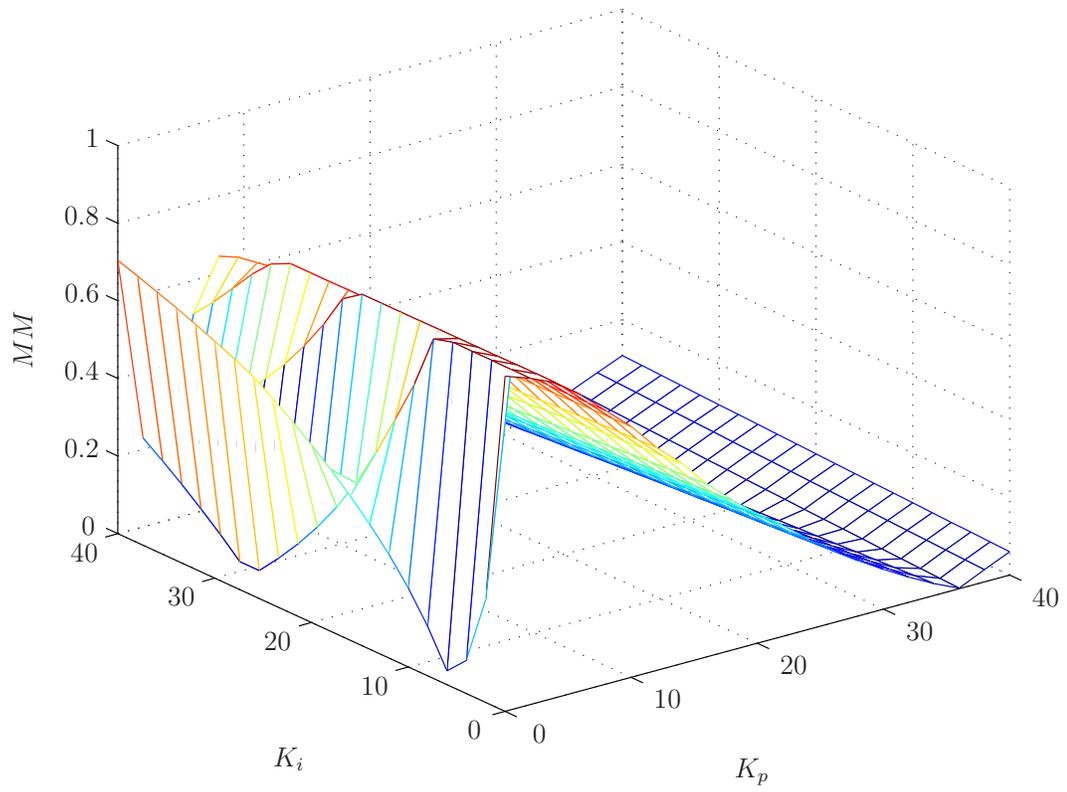


Figure C.4 Figure C.5(a) revisited

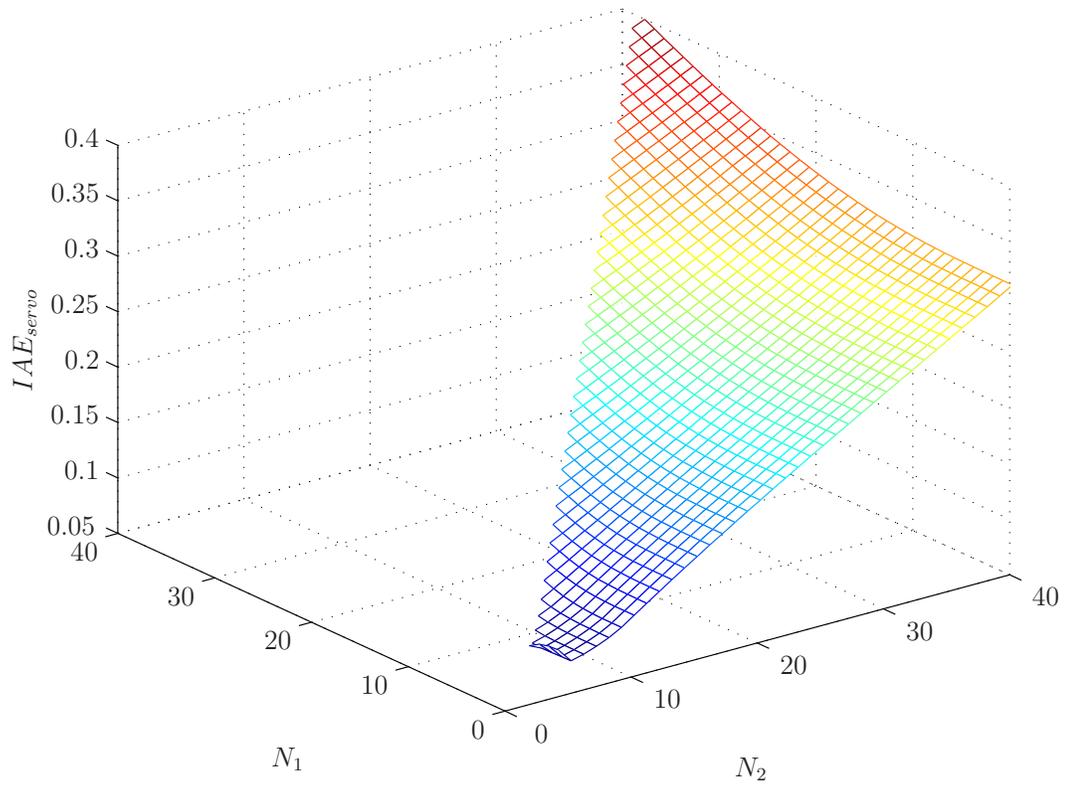


(a) *PID*

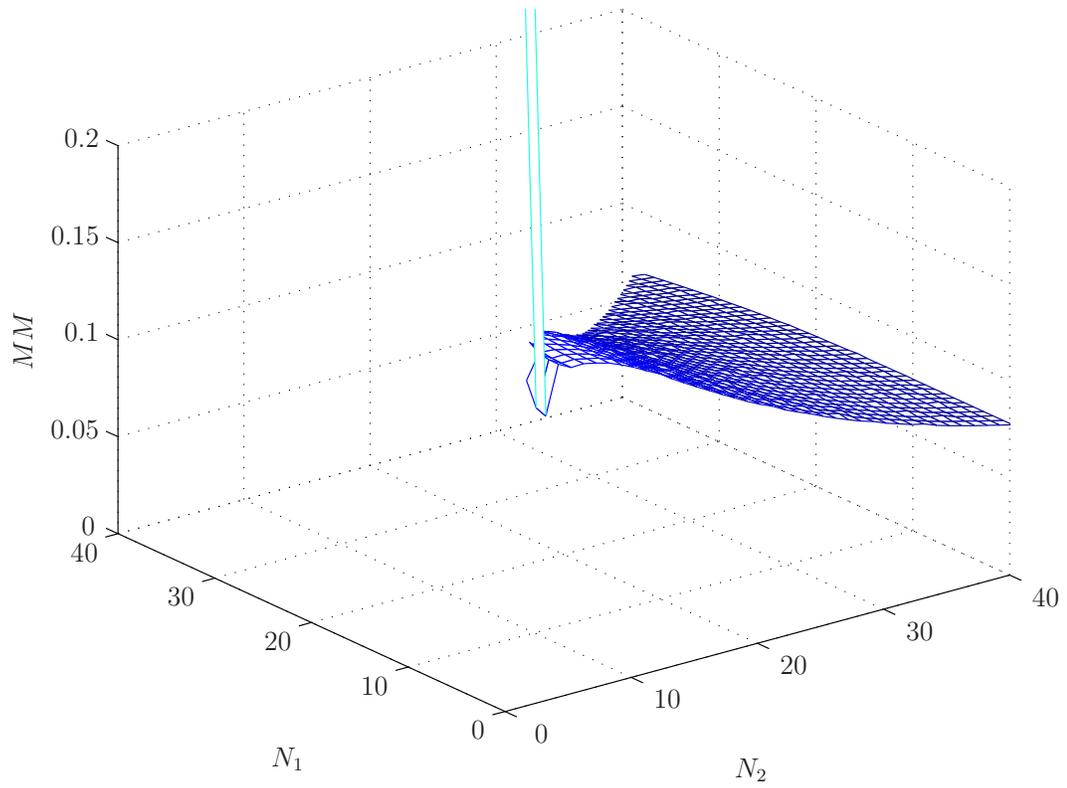


(b) *PID*

Figure C.5 Study of the search area for the system G_{10}



(a) *GPC*



(b) *GPC*

Figure C.6 Study of the search area for the system G_{10}