The Development of Advanced Multivariable, Linear and Nonlinear Control Design Methods with Applications to Marine Vehicles

Peter Martin

Industrial Control Centre
Department of Electronic and Electrical Engineering
University of Strathclyde
50 George Street
Glasgow G1 1QE
Scotland
The United Kingdom

Submitted for the degree of Doctor of Philosophy
May 2005

Declaration of Author's Rights

Abstract

This thesis primarily concerns control and identification of FPSO and Shuttle Tanker vessels, where nonlinear hydrodynamics raise the associated issue of nonlinear control. A 3-DOF model is presented for investigating Dynamic Positioning control, a problem where directional thrusters maintain ship position and heading against environmental disturbances. The coupled, multivariable dynamics are controlled using rapid tuning techniques to decouple the plant, yielding successful multivariable PI feedback designs. Identification of a coupled FPSO and Shuttle Tanker is achieved using an MLP neural network. Initially, the network is trained with simulation data for proof of concept, before employing real data from a Mitsubishi Heavy Industries scale model. Identification is successful, but performance degrades with increasing wave height. Two adaptive controllers are developed, based on polynomial LQG and LQGPC optimal control theory. The first uses a standard stochastic cost, approximated to produce a restricted structure controller that permits optimisation across several plant models at once, yielding a multiple model controller. Augmenting linearised ship models with online identification produces adaptive control giving interesting trade-offs between robustness and performance. The second adaptive controller is very similar, but based on a multi-step predictive cost function. Both controllers are applied to FPSO surge axis velocity control, where the LQGPC version produces better performance for a wave-induced reference. A multivariable nonlinear controller is examined for "sandwich" systems consisting of a linear transfer function "sandwiched" between input and output nonlinearities of a particular form. This system description is substituted into the solution of a time-varying polynomial optimal control problem, where the assumption of a frozen plant at each sampling instant requires slowly-varying plant signals in practice. The controller is successfully applied to a 2 × 2 plant with deadzone input and backlash output, with a demonstration that the performance is superior to a well-tuned linear controller.

Acknowledgements

I would like to wholeheartedly thank my supervisor, Professor Mike Grimble, for his support, encouragement and faith in my ability throughout my continuing association with the Industrial Control Centre. I am also hugely grateful to my 2nd supervisor, Doctor Reza Katebi, for his guidance and calmly supportive influence during the creation of this thesis.

Within the ICC, I would like to thank the secretaries, Drew, Sheena, Anne and Sheila for being friendly and helpful, even when I was being a nuisance. I met many fellow PhD students and good friends in the ICC, so I would like to thank them for their humour and willingness to discuss Life, the Universe, Control and Everything - Jonas "Skål" Balderud, Baris "Baz B" Bulut, Arkadiusz "Arek" Dutka, Leonardo "Señor" Giovanini, David Greenwood, Gerald Hearns, Pawel "Dzien Dobry" Majecki, Helen Markou, Ahmed "Salam Alaikum" Medhat, Stuart Robertson, John "How's England?" Russell, Alberto "Señor" Sanchez, Alaa Shawky, Drew "Beg Marn" Smith, Damien Uduehi and Matthew Wade. In the ISC company, I always found Andy Clegg, Dave Anderson, Gerrit van der Molen and Andy Buchanan to be very helpful.

Away from study, my old flatmates, Michael Grant, Tassos Trontis, Arturo Vega-Gonzalez and Vasilis Protopappas, made Glasgow a great place to be, often out for a few beers or a Wednesday International Night. I must thank the following girls, Joanne Heydon, Ezia Casulli, Tamaki Saito and Helen Cook for managing to put up with me even when I was a pain during my studies. My mates from home, The Lads, were always an e-mail away from cracking me up with laughter when I felt fed up. At BAE Systems, Matt MacDonald, Nick Brignall, Paul Ingleby, Brian Mann and David Robinson were a good bunch, even if I was late after spending way too long on this thesis the previous night. To any other friends I met during my years in Scotland, thanks for the happy memories.

Dedication

To My Family

Preface

Original Contributions of the Thesis

The original contributions of the thesis can be summarised as follows:

1. Successful demonstration that existing multivariable techniques due to Davison, Penttinen and Koivo, and Maciejowski can be applied to the ship dynamic positioning (DP) problem for rapid preliminary tuning.

2. It is established that a neural network has the capability to successfully learn and generalise the dynamics of a coupled-ship system from real data provided by a Mitsubishi Heavy Industries 1/50th scale model.

3. A new kind of adaptive controller is proposed based on polynomial optimal LQG theory, which combines the robustness of multiple models and the performance of standard adaptive control with the simplicity of a restricted structure controller. A successful ship DP example is given.

4. A novel adaptive predictive controller is proposed based on polynomial optimal LQGPC theory, which combines the robustness of multiple models and the performance of standard adaptive control with the simplicity of a restricted structure controller. A successful ship DP simulation example is given, where the performance is better than in the earlier LQG case.

5. A nonlinear control design idea based on time-varying control is extended to the multivariable case, where a MIMO example shows that superior performance over a fixed linear controler is possible.

Publications arising from this work

[2004] P. Martin and M. R. Katebi, *Multivariable PID Tuning of Dynamic Ship Positioning Control Systems*, Submitted to The IMarEST Journal of Marine Engineering, Science and Technology.

[2004] P. Martin and M. J. Grimble, *Multivariable Nonlinear Control*, Submitted to International Journal of Systems Science.

[2002] M. J. Grimble and P. Martin, *Restricted Structure Adaptive Predictive Control of Nonlinear Systems*, Proceedings of the IEEE Conference on Control Applications jointly with CACSD, Glasgow, Scotland, September 18th-20th 2002.

[2001] M. J. Grimble and P. Martin, *Adaptive Control Using Controllers of Restricted Structure*, Proceedings of the 7th IFAC Workshop on Adaptation and Learning in Control and Signal Processing, Cernobbio-Como, Italy, August 29th-31st 2001.

[2001] P. Martin, M. R. Katebi, I. Yamamoto, K. Daigo, E. Kobayashi, M. Matsuura, M. Hashimoto, H. Hirayama, N. Okamoto, *Neural Networks for System Identification of Coupled-Ship Dynamics*, Proceedings of the IFAC Conference on Control Applications in Marine Systems, Glasgow, Scotland, July 18th-20th 2001, Paper WA2.5.

[2000] P. Martin, I. Yamamoto, M. R. Katebi, M. J. Grimble, *Multivariable PI Tuning in Dynamic Position Control of Ships*, Proceedings of the 5th IFAC Conference on Manoeuvring and Control of Marine Craft, Aalborg, Denmark, August 23rd-26th 2000, pp.75-80.

Outline of the Thesis

Chapter One presents a short introduction to control theory applied to marine systems, followed by a discussion of the background and motivation for the forthcoming ideas in the thesis, in the areas of Dynamic Positioning (DP), ship identification and nonlinear control.

Chapter Two is an investigation of simple multivariable tuning techniques applied to the ship Dynamic Positioning (DP) problem. A non-linear ship model is detailed and linearised to form the basis of much of the work in the thesis. Four candidate tuning methods are introduced, each one decoupling the plant in a different range of frequencies. The control system structure is then given before applying each technique in turn to a linearised model. The results show that it is possible to rapidly tune a true multivariable controller for the DP problem, which will satisfy the constraints on thruster input demand and produce low positioning error.

Chapter Three is an application of neural networks to the problem of identifying coupled ship dynamics. The initial identification is performed on the model used in Chapter Two, in order to confirm that the technique is feasible for use with a real ship. The results are positive in this case, hence data supplied by Mitsubishi Heavy Industries from a 1/50th scale model is used in place of simulation data. It is found that the network is able to learn the dynamics, although the results are operating point dependent and are best when wave disturbances are small.

Chapter Four develops an adaptive controller of restricted structure. The technique is based on a multiple model optimal control solution for a system stated in polynomial form. A frequency domain cost is minimised across a set of linear models, to produce a controller in PI form. The set consists of several linearised models at common operating points, and the adaptive aspect is introduced by updating the set of linear models with another one that is identified via recursive least squares. At each sample step, the previous identified model is discarded and replaced by the latest data. In this way, the controller provides robustness via optimisation across a known set, and performance gains by incorporating the latest information.

Chapter Five builds on the previous Chapter by including a multi-step predictive cost criteron in the algorithm. This linear quadratic Gaussian predictive control (LQGPC) problem is initially stated in state-space form, with stochastic disturbance and reference generating models. Optimising over the future control signals allows the cost to be posed in a standard LQG form, thus leading once more to an adaptive multiple-model restricted structure controller. The example in both Chapters Four and Five is the ship DP control problem, allowing comparisons to be drawn between predictive and non-predictive controllers.

Chapter Six introduces a multivariable nonlinear controller that is based on the solution to a time-varying optimal control problem. The nonlinear plant is stated as a linear transfer function with input and output non-linearities, a so-called "sandwich" system. These non-linearities are posed in a particular form such that they act as a time-varying multiplier of their input signals. By freezing the time-varying description at each sampling instant, the infinite-time optimisation problem becomes tractable, hence a solution is obtained by calculating the controller online using nonlinearity-dependent polynomials. This approximation is not too great provided that the plant input and output signals vary slowly. The controller is simple to implement and a 2 × 2 system example is simulated, which yields results demonstrating that the achievable performance is superior to a fixed linear controller.

Chapter Seven provides a summary of the previous Chapters and presents new directions for research leading on from the current work.

See **Appendix A** for the notation used throughout the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Control and marine applications

Feedback control ideas have been applied to marine vessels for almost a century now, with the first known example being an automatic ship steering mechanism constructed by Elmer Sperry in 1911. This contraption, known affectionately as "Metal Mike" (www.sperry-marine.com), used the gyrocompass patented by H. Anschutz, in combination with automatic gain adjustments, to compensate for changing sea states. Over a decade later, Nicholas Minorsky had been studying the behaviour of expert ship pilots, culminating in the creation of a position feedback controller with three terms operating on the heading error. The detailed analysis in Minorsky (1922) was responsible for the now widely-used Proportional-Integral-Derivative (PID) controller.

This development preceded the so-called Classical Control period, where luminaries such as Nyquist (1932), Black (1934), Bode (1940) and Evans (1950) developed a framework for analysis and design of feedback control systems. Within this framework, it became possible to tune the response of a Minorsky course-keeping controller with quantifiable stability margins and an understanding of the behaviour of each PID term. Subsequent applications of control to marine vessels include forward speed regulation, where a hierarchy of control loops is implemented such that propellor pitch, engine speed and thrust are controlled to achieve a desired speed with optimal propulsion efficiency. Track-keeping systems combine course-keeping and speed regulation with lateral sway control to guide a ship along a route between way points.

Turning controllers allow the response of a heading change to be determined, often using a model reference scheme, rather than relying on the regulatory dynamics of a course-keeping controller. Roll stabilisation is an important application, as it provides comfort for crew and passengers, prevents cargo damage and assists operations such as aircraft take-off and landing. The actuation is achieved by the rudder, adjustable fins or a combination of the two, where controller design has been carried out with a range of classical and modern control techniques. All of the above applications are described in more detail in Fossen (1994).

1.2 Dynamic ship positioning

The marine application in this thesis is known as Dynamic Positioning (DP), where the problem is to maintain a fixed vessel position and heading against environmental disturbances by using directional thrusters. The ship model in question is for a Floating Production, Storage and Offloading (FPSO) vessel belonging to Mitubishi Heavy Industries (MHI) of Japan. The model is multivariable, highly coupled and nonlinear, but is linearised at several operating points for use during the thesis. The vessel is subject to current forces and a standard disturbance spectrum due to wind and waves, all described in more detail in Chapter 2.

The DP controller is used for drilling and offloading activities and is essential for safe operation, often in harsh conditions. The majority of previous control designs are linear, ranging from the early independent PID loop control of

Schneider (1969), through the LQG designs of Balchen et al. (1976) and Grimble et al. (1980a), with a H_∞ approach recently by Katebi et al. (1997). Fossen and Grøvlen (1998) have introduced nonlinear observers and observer backstepping to DP, while Yamamoto et al. (1998) have investigated fuzzy logic control with some success. In Chapters 2, 4 and 5 of this thesis, some new control ideas are applied to the DP problem, beginning with linear multivariable decoupling techniques to facilitate rapid multivariable PID tuning for early ship trials or academic comparisons. PID control for DP has largely been ignored after the introduction of LQG designs, thus the simple ideas in Chapter 2 are an original contribution and form the basis for the remainder of the ship applications in the thesis.

After this consideration of the multivariable aspect of DP, Chapter 4 addresses the nonlinearities to some extent. A restricted structure polynomial LQG idea is extended to the multiple-model adaptive case, where linear representations of the ship at different operating points form the set of multiple models. An on-line optimisation is performed across this set, augmented by an identified linear model, so that the controller combines knowledge of potential ship descriptions with a representation at a given sampling instant. Although not rigorously shown, it is believed that such a control scheme provides a combination of robustness, via the fixed set of models, and performance enhancement, via the inclusion of an identified model. The structure of the controller is PI, giving the benefits of simplicity of implementation and transparency of operation.

The controller in Chapter 5 is similar to the above, but the optimisation is instead performed on a multi-step predictive cost function. Both controllers suffer with numerical problems and hence are applied to a SISO element of the overall ship. The controllers give a novel contribution, both in terms of extending restricted structure control techniques to a multi-model adaptive case and applying them to the DP problem.

1.3 Ship identification

As an adjoint to the main investigation of DP control, Chapter 3 of this thesis looks at identification of ship dynamics with a neural network. This study was motivated by a problem encountered by MHI of Japan, where two ships are coupled together with a metal hawser when oil is offloading from an FPSO to a Shuttle Tanker. This problem has been investigated analytically in Morishita et al. (2001) using hydrodynamical equations. The authors concluded that the dynamics are complicated and there are numerous modes, stable and unstable operating points present. The dynamics of such a coupled system are complex and liable to modelling errors using first principle derivations, hence it was decided to explore the potential of a "black-box" technique in this thesis.

Neural networks have a limited history of application to marine vessel motion identification, with Haddara and Xu (1999) being one of the few examples, although with a focus on heave and pitch dynamics. The only published research known to this author on identifying motion relevant to DP with neural networks is by Hardier (1995), who uses a multilayer perceptron (MLP) with tapped-delay lines to learn the surge, sway, yaw and roll dynamics of a $1/12th$ scale model of the Charles de Gaulle aircraft carrier.

In Chapter 3 an MLP is also used, but without the recurrence of the tapped-delay lines. This is a simpler approach than Hardier, as the mapping is from the forces on the ship to body accelerations, hence integrators are introduced to give velocity and position signals. This method does produce cumulative errors in velocity and position, but in practice these errors would be cancelled by a controller with integral action. The contribution of Chapter 3 comes from demonstrating the potential of a non-recurrent MLP for ship identification and from an application to a coupled-ship system.

1.4 Nonlinear control

The thesis has a theme of nonlinearities running through it as a consequence of the ship application. Chapter 3 details true nonlinear identification, whilst Chapters 4 and 5 address the problem through an optimisation across multiple linear models. Practical techniques are in short supply for control system design applied to a nonlinear plant, thus the final Chapter attempts to make a modest contribution in the area of nonlinear multivariable system control. In general, nonlinear analysis and design approaches require approximations, complex mathematics or are rather impractical. Methods such as Describing Functions, Geometric Control and Lyapunov theory fall respectively into these categories. In Chapter 6, the controller has a rigorous foundation, yet is practical and simple to apply, although some approximations are inevitably made.

The nonlinear control idea is based on the theory of time-varying linear systems, where the behaviour of the nonlinearity is captured by a linear "snapshot" of the system at a given sampling instant. A time-varying polynomial optimal control problem is solved, which is then adapted for use with a so-called nonlinear "sandwich" system. This system description comprises a linear transfer function block sandwiched between input and output nonlinear functions of a particular form, thus encompassing a considerable set of possible nonlinear plants. The polynomials representing the nonlinear plant are substituted into the time-varying solution and, assuming that the plant changes slowly, an approximate LQG type of solution is arrived at.

The plant in the example of the final Chapter is not a ship, unlike the previous Chapters, as the algorithm is not numerically robust for multivariable systems of order greater than one. Therefore, a fictitious 2×2 plant with deadzone input and backlash output nonlinearites is used to demonstrate the value of the controller.

Chapter 2

Ship Modelling and Multivariable Controller Tuning

This Chapter introduces the ship Dynamic Positioning (DP) control problem, details a nonlinear multivariable simulation model for a Floating Production, Storage and Offloading (FPSO) vessel and presents four model-free linear methods for rapid tuning of a multivariable PID controller. Controller structures are discussed and the methods are applied to the simulation model at an operating point, thereby justifying the use of linear tuning techniques. It is shown that some of the methods are appropriate and that it is straightforward to produce a controller of acceptable performance within the thruster input constraints of the ship. Simulation results are presented and analysed at the end of the Chapter.

2.1 Introduction

This Chapter investigates the application of simple multivariable controller tuning techniques to the Dynamic Positioning (DP) of a Floating Production, Storage and Offloading (FPSO) vessel. The problem is described in van Calcar and Morgan (1975) as follows: "Dynamic Positioning (DP) of a vessel refers to the process of automatically controlling the vessel's thrusters and/or main screws to maintain the vessel at a fixed position and heading and/or at a precise speed along a selected track". Fossen (1994) offers the definition: "A dynamically positioned vessel is a vessel which maintains its position (fixed location or predetermined track) exclusively by means of active thrusters". In this Chapter, DP will be interpreted in the more modern sense, where speed control is neglected.

Published research in this area dates back over thirty years to early papers such as Schneider (1969) and Harbonn (1971). Schneider (1969) discusses DP in a very general sense, detailing the sonar position measurement system and general aims of the feedback control law, and gives results taken from the Glomar Challenger drilling vessel. There is no mathematical analysis of the vessel or controller in the paper, but the Glomar Challenger control inputs are described as a bow and stern side thruster in addition to a main screw, and the controller is some form of Proportional, Integral and Derivative (PID) control in three independent loops. It is stated in the paper that position variations of less than 6 metres are "normal" in $1.1km$ water depth with sea state 1 and a variable wind of $3.5 - 4.5m/s$. The water depth is significant here because the accuracy of sonar position measurement decreases with increasing depth, and the maximum vessel deviation limit is stated as 3% of depth. In this case, deviations are less than 0.5%. Also, notably, filtering of the position measurements is not mentioned.

The work of Harbonn (1971) details a five year study carried out with Terebel, a French floating platform. An inclinometer, a taut wire to the ocean bed with angle sensor, is used as position sensor for this $900Tonne$ vessel. The accuracy of measurement with this system is also dependent on water depth. Fore and aft pivot-type propellors with a thrust maximum of $2.7Tonnes$ are used as control inputs. Again, three independent PID loops are used for control, and the tuning is carried out by trial and error - A controller frequency response is given in the paper, and wave-filtering of the derivative term is included. Station-keeping accuracy is quoted as better than 15 metres when drilling at a depth of $2.6km$, 0.5% as with the Glomar Challenger, although the weather conditions are not mentioned.

van Calcar and Morgan (1975) and Morgan (1978) review the state-of-the-art in DP at that time. PID control is still in use, with the addition of a feedforward term acting on the sensed wind speed and direction. A digital adaptive wave filtering technique, used by the authors' company, is also mentioned in van Calcar and Morgan (1975).

Shortly after this state-of-the-art, a modern control approach was taken for the first time by Balchen et al. (1976). Until then, control had always consisted of separate PID loops tuned in a heuristic fashion. The new approach was to model the ship in state-space, and to split the state vector into separate low and high frequency components. The low frequency component is based on ship dynamics, wind, current and low frequency wave motion and the high frequency component is based on a model of expected high frequency wave conditions. An extended Kalman filter is then utilised to predict overall ship motion as the sum of the two components, but only the low frequency state is used for control. By doing so, wave filtering is effectively carried out within the Kalman filter, and the need to introduce notch wave filters, with associated phase lag, is removed. The controller is a state-feedback controller with proportional and integral action, designed using standard Linear Quadratic Gaussian (LQG) theory.

A whole series of papers based on the Kalman filtering technique followed, notably Grimble et al. (1980a) use a steady-state Kalman filter, Grimble (1980b) investigates combined state and state-estimate feedback, Grimble et al. (1980c) discuss extended and steady-state Kalman filter designs and compare notch filtering with Kalman filtering of ship position. Fotakis et al. (1982) compare controller designs using the characteristic locus and multivariable root locus with optimal methods of design, where a Kalman filter is used to provide state and output estimates in both cases. Sælid et al. (1983) introduce a new low frequency model and a wave frequency adaptation algorithm, and Fung and Grimble (1983) describe a self-tuning Kalman filter which adapts to the high frequency wave disturbance.

Recently, other modern control techniques have been applied to the DP problem. Sørenson et al. (1996) describe an LQG design incorporating a reference model for calculating feasible trajectories of vessel motion, a wind feedforward controller and a model reference feedforward controller. Katebi et al. (1997) use H_∞ robust control theory to produce a design for a linearised ship model which can cope with perturbations in the operating point whilst retaining stability. Fossen and Grøvlen (1998) introduce a nonlinear observer to avoid the necessity of a set of linearised models, and nonlinear feedback control using observer backstepping. Global exponential stability is then proven using Lyapunov stability theory. Strand et al. (2001) bring together several of these recent ideas and review modelling, observer and control design. Yamamoto et al. (1998) describe a fuzzy logic controller based on a nonlinear programming algorithm, and compare the results favourably with conventional PID control.

The use of PID for DP has fallen out of favour due to the development of the more advanced control strategies above. However, for preliminary ship sea trials and for academic studies giving comparisons of modern controllers with PID, it is always useful to be able to quickly tune a simple controller of reasonable performance. Model-free design methods in the spirit of Ziegler and Nichols (1942) or the Stability Limit, Cohen-Coon and Tyreus-Luyben methods in Kiong et al. (1999) are suitable for such tuning. Limited information about the plant from a step response, or frequency response at one frequency, is used to produce acceptable closed-loop behaviour with little effort. These simple methods are intended for single-input single-output (SISO) systems, but in this Chapter a nonlinear multi-input multi-output (MIMO) ship model will be presented and 18 different operating points found. Hence, a quick method of tuning a MIMO PID controller at each operating point is desirable, so that gain scheduling could be used for overall control.

It would of course be possible to use Ziegler-Nichols or one of the other methods on each input and output of the ship separately, in order to produce independent PID controllers as with Schneider (1969) and Harbonn (1971). However, a ship exhibits significant interactions between the various loops, thus a true multivariable strategy is preferable.

Maciejowski (1989) details several model-based off-line MIMO design techniques, including Sequential Loop Closing, the Characteristic Locus method, Nyquist-arrays, and Quantitative Feedback Theory. The Sequential Loop Closing method is simple and involves designing SISO controllers and closing the loops one at a time, where the effects of previously closed loops are taken into account. An extension to this is the Sequential Return-Difference method proposed by Mayne (1979), whereby a decoupling matrix is added before designing the individual controllers. Maciejowski (1989) suggests a choice of this matrix which will be discussed in Section 2.3 of this Chapter. The idea of the Characteristic Locus method is to select the compensator structure so that each characteristic function of the cascaded plant and compensator, $G(s)K(s)$, is the product of an individual plant, $G(s)$, and compensator, $K(s)$, characteristic function. It is then possible to design a compensator for each plant characteristic function using SISO Nyquist loci.

Nyquist-array methods involve the use of an array of graphs, where each graph is the Nyquist plot of an element of a transfer function matrix. With the Inverse Nyquist-array (INA) method, neighbourhoods of the inverse Nyquist loci known as Ostrowski Bands are treated as if they are SISO inverse Nyquist loci for the purposes of control design. The Direct Nyquist-array (DNA) method uses neighbourhoods of the normal Nyquist loci known as Gershgorin Bands to predict closed-loop stability, but does not take into account the effect of loop coupling as with INA. In both cases, the return difference, $I + G(s)K(s)$, must possess the property of Diagonal Dominance, where the union of Gershgorin Bands must exclude the origin. This requirement coupled with extensive inspection of the Nyquist arrays during design produces a rather involved design technique.

Quantitative Feedback Theory (QFT) is based on the premise that the amount of uncertainty of a plant, and the precise specification of tolerable closed-loop behaviour given the uncertainty, may be quantified by bounds on the frequency response at each frequency. The design technique involves selecting a diagonal controller and a reference pre-filter so that the frequency responses satisfy these bounds. Maciejowski (1989) also describes the multivariable case of the well-known LQG and H_∞ optimal design techniques.

In addition, there is the Biggest Log Tuning (BLT) method of Luyben (1986). Tuning begins by calculating the Ziegler-Nichols (Z-N) settings for each loop and introducing a tuning factor, F, which scales the Z-N gains. The value of F is then varied until the maximum of a log function of the plant characteristic equation equals some empirically decided value. This value is $2NdB$ for an $N \times N$ system. Reasonable performance is shown in the results for several process control examples.

Internal Model Control (IMC) was introduced by Garcia and Morari (1982) and the method of obtaining PID parameters for this structure is described in Rivera et al. (1986). The basic idea is that the closed-loop system contains a model of the plant in parallel with the actual plant, and the signal which is fed back to the comparator is the difference between the outputs of this internal model and the real plant. The controller is then the inverse of the model ignoring any time delays and right half-plane zeros. The PID controller structure is found to follow naturally from the IMC design procedure for many simple transfer function models in process control. An extension of IMC to robust multivariable PID control is given in Dong and Brosilow (1997), where a PID structure is obtained by expanding the controller transfer function with a Maclaurin series to the first three terms. The controller parameters are then a function of the internal model and filters.

None of the aforementioned methods really satisfy the desire for a quick and easy way to tune a multivariable controller in the manner of the SISO techniques mentioned above, however. Katebi et al. (2000) survey the various MIMO PID tuning methods within the control literature and suggest some techniques which are more appropriate for simple designs. The model-based methods above are reviewed and some model-free on-line "autotuning" techniques are given, so-called because it is possible to automate them to the extent of simply pushing a button, indeed their SISO counterparts are widely used in industry. Zhuang and Atherton (1994) have used a MIMO extension of the Ziegler-Nichols method for autotuning of two-input two-output (TITO) systems. Loh et al. (1993) extend the relay feedback idea of Åström and Hägglund (1995) by tuning one loop at a time in the manner of Sequential Loop Closing. Palmor et al. (1995) also use relay feedback, but tune both loops of a TITO process together. Wang et al. (1997) describe a method for autotuning fully cross-coupled multivariable PID controllers from decentralised relay feedback. A thorough review of the state of PID autotuning is given by Yu (1999).

The model-free off-line methods given by Davison (1976), Penttinen and Koivo

(1980) and Maciejowski (1989) are the focus of this Chapter, however, as they require little design effort and are based on step tests or frequency responses at a single point. The robust servomechanism problem was introduced by Davison (1976) to provide a general controller design method with guarantees of asymptotic stability and asymptotic tracking given disturbances of a particular form, and plant model perturbations that did not result in closed-loop instability. More pertinently, a detailed mathematical model of the plant is not necessary and the controller may be constructed based on simple open-loop tests. Penttinen and Koivo (1980) suggest a way to diagonalise the plant at very low and very high frequencies, which is in fact an extension of Davison's work. The decoupling matrix at a particular bandwidth frequency, suggested in Maciejowski (1989) with regard to the Sequential Return-Difference method, is also investigated, as an intermediate approach to that of Penttinen and Koivo (1980). These three techniques, and a simple new combination of all three, will be described in more detail in Section 2.3.

In the following Sections of this Chapter, a novel contribution to DP control design will be made via the application of the multivariable PID design methods above. Before discussing these ideas further, the ship simulation model to be utilised during the thesis will be presented.

2.2 Ship modelling

Mathematical modelling of the motion of ships, underwater vehicles, and high speed craft is an extended exercise in the classical mechanics of Newton, Lagrange, Euler and Kirchhoff. This exercise is superbly summarised in the work of Fossen (1994), in which a complete analysis of a 6 degree-of-freedom model is given. The six different components of motion are surge, sway, heave, roll, pitch, and yaw. Two different coordinate frames are used, one is the body-fixed frame which moves with the ship, the other is the earth-fixed frame, which is an inertial reference. The motion components and coordinate frames are shown in Figure 2.1.

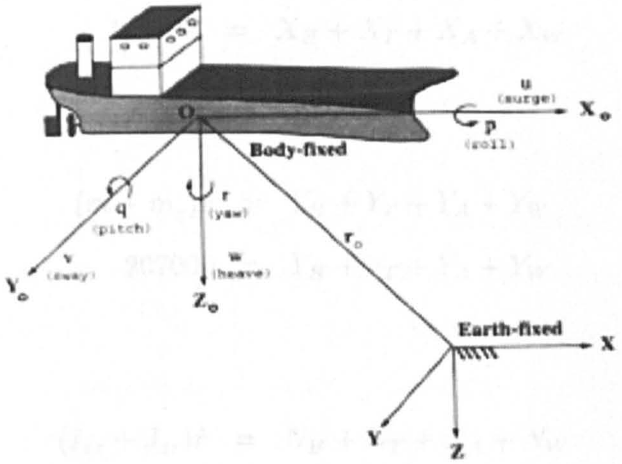

Figure 2.1: Ship and coordinate frames, from Fossen(1994)

For the purposes of dynamic position control, only the surge, sway, and yaw motions are of interest. Heave, pitch, and roll control are important for a comfortable ride on high speed vessels but are ignored here, where the objective is to keep the ship at a particular longitude, latitude and heading regardless of the crew's stomach. The nonlinear model and parameters used throughout the thesis are the result of a case study carried out for Mitsubishi Heavy Industries Ltd. A list of parameters and their values is to be found in Appendix A. The general model is applicable to a range of vessels, given the correct parameters, but in this thesis will apply to FPSOs and Shuttle Tankers, such as pictured in Figure 2.2.

2.2.1 Equations of motion

If the origin of the body-fixed coordinate system coincides with the ship centre of gravity, the Coriolis effect is ignored, and taking into account the sea kinetic energy, the equations of motion are as follows:

$$\begin{aligned}(m+m_x)\dot{u} &= X_H + X_T + X_A + X_W \\ 13200\dot{u} &= X_H + X_T + X_A + X_W \qquad (2.1)\end{aligned}$$

$$\begin{aligned}(m+m_y)\dot{v} &= Y_H + Y_T + Y_A + Y_W \\ 20700\dot{v} &= Y_H + Y_T + Y_A + Y_W \qquad (2.2)\end{aligned}$$

$$\begin{aligned}(I_{zz}+J_{zz})\dot{r} &= N_H + N_T + N_A + N_W \\ 94000000\dot{r} &= N_H + N_T + N_A + N_W \qquad (2.3)\end{aligned}$$

where each m denotes mass, and I and J denote ship inertia. u, v, and r are the surge, sway and yaw body-fixed velocities respectively. X, Y, and N stand for surge, sway and yaw forces and moments, and the subscripts $_H$, $_T$, $_A$, and $_W$ stand for hydrodynamic forces, thruster forces, and wind and wave forces respectively.

Figure 2.2: MODEC Venture 1 FPSO - Picture taken from www.mes.co.jp

The vessel under investigation possesses three azimuth thrusters - stern left and right with a maximum of $170T$ force each, as well as a bow thruster with 130T maximum. Thruster location is illustrated in Figure 2.3. The mathematical model of the ship, however, does not take into account the independence of these thrusters. Forces are simply lumped into surge and sway direction forces, and an overall moment, N_T. It is assumed that a thruster allocation algorithm provides the inputs to each thruster. Therefore, it is not possible to check the three thrusters separately, so the vector magnitude of surge and sway forces is instead calculated in the simulation later to ensure that it is less than $470T$ - the maximum possible force from all three engines. In addition, the maximum rate of change of the thrust vector is $50T/s$ and the moment may not exceed 10^5Tm.

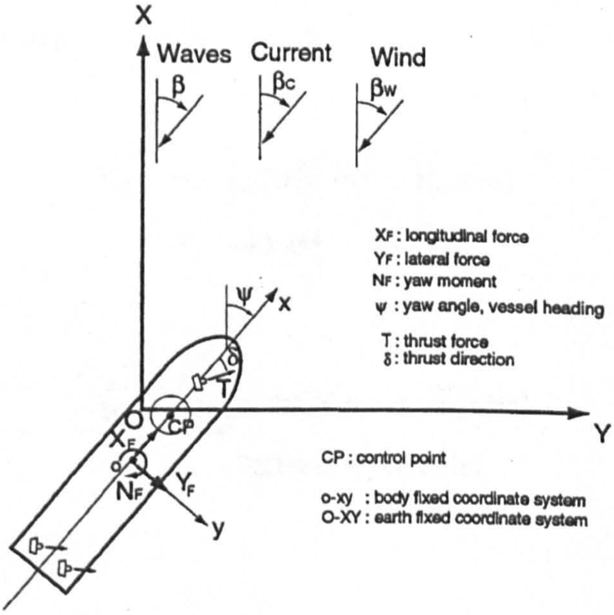

Figure 2.3: Plan view of the ship with thrusters illustrated

2.2.2 Hydrodynamic forces

The hydrodynamic forces due to the damping effect of the water are:

$$\begin{aligned} X_H &= \frac{1}{2}\rho Ld(X'_{vv}v^2) + \frac{1}{2}\rho L^2 d(LX'_{rr}r^2 + X'_{vr}vr - \frac{X'_{uu}u|u|}{L}) \\ &= -1.11v^2 - 247000r^2 - 3310vr - 24.5u|u| \end{aligned} \tag{2.4}$$

$$\begin{aligned} Y_H &= Y_{H_v} + \frac{1}{2}\rho L^2 d(Y'_r Ur + LY'_{rr}r|r| + Y'_{vr}v|r|) \\ &= Y_{H_v} + 3280Ur + 1300000r|r| - 19500v|r| \end{aligned} \tag{2.5}$$

$$\begin{aligned} N_H &= N_{H_v} + \frac{1}{2}\rho L^3 d(N'_r Ur + LN'_{rr}r|r| + N'_{vr}v|r|) \\ &= N_{H_v} - 432000Ur - 121000000r|r| + 1880000v|r| \end{aligned} \tag{2.6}$$

When $|\beta_C - \Psi| < \pi/4$

$$\begin{aligned} Y_{H_v} &= \frac{1}{2}\rho Ld(Y'_v vU + Y'_{vv}v|v|) \\ &= -42.4vU - 177v|v| \end{aligned} \tag{2.7}$$

$$\begin{aligned} N_{H_v} &= \frac{1}{2}\rho L^2 d(N'_v vU + N'_{vv}v|v|) \\ &= -7390vU + 3620v|v| \end{aligned} \tag{2.8}$$

When $|\beta_C - \Psi| \geq \pi/4$

$$\begin{aligned} Y_{H_v} &= -0.7sin(\mu_C)\frac{1}{2}\rho LdU^2 \\ &= -136sin(\mu_C)U^2 \end{aligned} \tag{2.9}$$

$$\begin{aligned} N_{H_v} &= -0.1sin(2\mu_C)\frac{1}{2}\rho L^2 dU^2 \\ &= -4980sin(2\mu_C)U^2 \end{aligned} \tag{2.10}$$

$$U = (u^2 + v^2)^{1/2} \tag{2.11}$$

$$\mu_C = tan^{-1}(\frac{v}{u}) \tag{2.12}$$

where ρ is water density, L is ship length, and d is ship draught. U is the magnitude and μ_C is the angle of the total ship velocity vector in the body-fixed coordinate system. All X', Y', and N' are constant hydrodynamic derivatives, Ψ is the ship heading and β_C is the angle of current in the earth-fixed coordinates.

The environmental disturbances acting on the ship are waves, wind and current, described in the next three subsections.

2.2.3 Wave forces

The effect of wave disturbances is split into two components, slowly varying wave drifting forces and rapidly varying wave exciting forces, also known as $1st$ and $2nd$ order waves respectively. The parameters in the following mathematical description are found by fitting forces and moments to data based on the spectrum of Pierson and Moskowitz (1963).

Wave drifting forces

The wave drifting forces span the frequency range 0.02 to $0.49 rad/s$ with peak values in the surge, sway and yaw direction of $50T$ ($\mu_W \approx 0$), 200T ($\mu_W \approx \pm\pi/2$), and 4000Tm ($\mu_W \approx \pm\pi/4$) respectively for heavy seas. They are zero mean and tend to slowly push the ship off position in one direction then another. A successful dynamic position control system is able to reject these disturbances. The mathematical description follows:

$$\begin{aligned} X_{WD} &= \sum_{m=1}^{N}\sum_{n=1}^{N} -0.0475 a_m a_n cos(\mu_W)\rho g L cos\{(\omega_m - \omega_n)t \\ &\quad +(\kappa_m - \kappa_n)(X cos\beta_W + Y sin\beta_W) + (\varepsilon_m - \varepsilon_n)\} \end{aligned} \tag{2.13}$$

$$\begin{aligned} Y_{WD} &= \sum_{m=1}^{N}\sum_{n=1}^{N} -0.225 a_m a_n sin^5(\mu_W)\rho g L cos\{(\omega_m - \omega_n)t \\ &\quad +(\kappa_m - \kappa_n)(X cos\beta_W + Y sin\beta_W) + (\varepsilon_m - \varepsilon_n)\} \end{aligned} \tag{2.14}$$

$$\begin{aligned} N_{WD} &= \sum_{m=1}^{N}\sum_{n=1}^{N} -0.015 a_m a_n sin(2\mu_W)\rho g L^2 cos\{(\omega_m - \omega_n)t \\ &\quad +(\kappa_m - \kappa_n)(X cos\beta_W + Y sin\beta_W) + (\varepsilon_m - \varepsilon_n)\} \end{aligned} \tag{2.15}$$

$$\mu_W = \beta_W - \Psi \tag{2.16}$$

where every a denotes a wave amplitude component, μ_W is the incident angle of waves in the body-fixed coordinates, g is the gravitational constant and t is time. All ω are frequencies of wave components, all $\kappa = \omega^2/g$ are wave numbers, and all ϵ are wave phase angles. In the earth-fixed coordinate system, β_W is the angle of waves, and X and Y denote ship position.

Wave exciting forces

The expressions for these forces are given below:

$$X_{WE} = \sum_{n=1}^{N} a_n f_{XWE}(\mu_W) cos(\omega_n t + \kappa_n (X cos\beta_W + Y sin\beta_W) + \varepsilon_n + \varepsilon_{XWE}(\mu_W)) \tag{2.17}$$

$$Y_{WE} = \sum_{n=1}^{N} a_n f_{YWE}(\mu_W) cos(\omega_n t + \kappa_n (X cos\beta_W + Y sin\beta_W) + \varepsilon_n + \varepsilon_{YWE}(\mu_W)) \tag{2.18}$$

$$N_{WE} = \sum_{n=1}^{N} a_n f_{NWE}(\mu_W) cos(\omega_n t + \kappa_n(X cos\beta_W + Y sin\beta_W) + \varepsilon_n + \varepsilon_{NWE}(\mu_W)) \tag{2.19}$$

$$f_{XWE}(\mu_W) = 500sin^2(2\mu_W) + 110(cos(6\mu_W) + 1) \tag{2.20}$$

$$f_{YWE}(\mu_W) = 3800sin^3\mu_W \tag{2.21}$$

$$f_{NWE}(\mu_W) = 140000sign(sin(\mu_W))sin^2(2\mu_W) \tag{2.22}$$

$$\varepsilon_{XWE}(\mu_W) = 1.8sin(1.5\mu_W) + 0.6(1 + cos(\mu_W)) \tag{2.23}$$

$$\varepsilon_{YWE}(\mu_W) = 2.5sin^2(1.5\mu_W) - 0.45cos^2(0.5\mu_W)(4 + cos(3\mu_W)) \tag{2.24}$$

$$\varepsilon_{NWE}(\mu_W) = 3.2sin(1.5\mu_W) - 0.7(1 + cos(\mu_W)) \tag{2.25}$$

where all f are wave force functions and all ϵ are wave phase angle functions.

The wave exciting forces span the frequency range 0.45 to 1.08 rad/s with peak values in the surge, sway and yaw directions of $1600T$ ($\mu_W \approx \pm\pi/4$), $10000T$ ($\mu_W \approx \pm\pi/2$), and $400000Tm$ ($\mu_W \approx \pm\pi/4$) respectively for heavy seas. Due to their large magnitude and high frequency, it is a waste of control energy to attempt to counteract these disturbances as they are zero mean and do not affect the average ship position. Therefore, feedback controllers require position measurement filters for the wave exciting forces, as detailed in Section 2.4. Furthermore, even with filtering it is possible to saturate the ship thrusters if the angle, μ_W, between the waves and ship heading is too large. The dynamic position control system is usually activated when the ship heading is close to being directly into the waves, to minimise thruster activity and energy consumption.

Total wave force

The total force exerted on the ship due to waves is the sum of the drifting and exciting forces.

$$X_W = X_{WD} + X_{WE} \tag{2.26}$$

$$Y_W = Y_{WD} + Y_{WE} \tag{2.27}$$

$$N_W = N_{WD} + N_{WE} \tag{2.28}$$

2.2.4 Wind forces

Wind consists of an average plus a fluctuating velocity component. The wind forces are functions of the velocity as follows:

$$\begin{aligned} X_A &= -1.15sin(1.4\mu_A + \pi/3)\frac{1}{2}\rho_A A_{AT} V_A^2 \\ &= -0.0633sin(1.4\mu_A + \pi/3)V_A^2 \end{aligned} \tag{2.29}$$

$$\begin{aligned} Y_A &= -0.9sin(\mu_A)\frac{1}{2}\rho_A A_{AL} V_A^2 \\ &= -0.283sin(\mu_A)V_A^2 \end{aligned} \tag{2.30}$$

$$\begin{aligned} N_A &= -0.087sin(2\mu_A)\frac{1}{2}\rho_A L A_{AL} V_A^2 \\ &= -7sin(2\mu_A)V_A^2 \end{aligned} \tag{2.31}$$

$$\mu_A = tan^{-1}(\frac{v_A}{u_A}) \tag{2.32}$$

$$V_A^2 = u_A^2 + v_A^2 \tag{2.33}$$

$$u_A = u_{abs} + U_A(t)cos(\beta_A - \Psi) \tag{2.34}$$

$$v_A = v_{abs} + U_A(t)sin(\beta_A - \Psi) \tag{2.35}$$

$$U_A(t) = U_{A0} + u_{A0}(t) \tag{2.36}$$

$$u_{A0}(t) = \sum_{i=1}^{N} a_{Ai}cos(\omega_{Ai}t + \varepsilon_{Ai}) \tag{2.37}$$

where μ_A is the angle and V_A is the magnitude of the total wind velocity vector in body-fixed coordinates. β_A is the angle of wind in earth-fixed coordinates, ρ_A is the density of air, A_{AT} is the transverse wind area, and A_{AL} is the lateral wind area. u_A and v_A are the surge and sway components of wind velocity and u_{abs} and v_{abs} are absolute ship velocities in the body-fixed directions. The absolute velocities are simply the surge and sway body-fixed velocities minus that of the current in the opposite direction. This is an assumption which is valid only in the steady state when the current has had enough time to act on the ship, but the assumption is made nevertheless. $U_A(t)$ is the wind velocity relative to the earth and consists of a constant average component, U_{A0}, and a fluctuating component $u_{A0}(t)$. a_{Ai}, ω_{Ai}, and ϵ_{Ai} are the amplitude, frequency and phase respectively of the fluctuating component.

In high winds of $20m/s$ (or $45mph$), the peak values in surge, sway and yaw directions are $25T$, $140T$, and $2500Tm$ respectively. A successful dynamic position control system is able to reject these disturbances.

2.2.5 Coordinate systems and transformations

The ship model detailed so far has only taken account of acceleration and velocity with respect to body-fixed coordinates. The ship position which we wish to control is with respect to a set of earth-fixed coordinates, (X, Y, Ψ). The transformation between the two coordinate systems is defined by:

$$X = \int (u_{abs} cos\Psi - v_{abs} sin\Psi) dt \tag{2.38}$$

$$Y = \int (u_{abs} sin\Psi + v_{abs} cos\Psi) dt \tag{2.39}$$

$$\Psi = \int r dt \tag{2.40}$$

$$\underline{y} = \begin{bmatrix} u_{abs} \\ v_{abs} \\ r_{abs} \end{bmatrix} = \underline{x} - \underline{c} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} - \begin{bmatrix} u_c \\ v_c \\ 0 \end{bmatrix} \tag{2.41}$$

with transfer function matrix:

$$\underline{Y}(s) = \begin{bmatrix} X \\ Y \\ \Psi \end{bmatrix} = \frac{1}{s} T_{BE} \underline{y}(s) = \frac{1}{s} \begin{bmatrix} cos\Psi & -sin\Psi & 0 \\ sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{abs} \\ v_{abs} \\ r_{abs} \end{bmatrix} \tag{2.42}$$

for constant ship heading, Ψ, at an operating point. The effect of the current is included in this formulation by defining the absolute ship velocity to be the body-fixed velocity minus the velocity of the current in the opposite direction, as already stated.

The reference signal of the controller will be in the earth-fixed coordinate scheme, therefore another transformation is required for the thrusters as follows:

$$\underline{e}(s) = \begin{bmatrix} e_x \\ e_y \\ e_\Psi \end{bmatrix} = T_{EB} \underline{E}(s) = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_X \\ E_Y \\ E_\Psi \end{bmatrix} \tag{2.43}$$

where $\underline{e}$ and $\underline{E}$ are the vectors of position error in the body-fixed and earth-fixed coordinates respectively. In Section 2.4, Figures 2.7, 2.8 and 2.11 show the location of the transformations for feedback control. Figure 2.3 in Section 2.2.1 illustrates the various forces acting on the type of ship in question, as well as the body and earth-fixed coordinate systems.

2.2.6 Model linearisation

Although the controller tuning techniques to be investigated in this Chapter are not model-based, the linearised ship model is now described and later used for control design. The tuning techniques require only step tests and the determination of frequency responses at a single frequency. If the nonlinear ship simulation settles at a stable operating point and these simple tests are applied, the correspondence between the simulation and linearised model results is almost exact. Hence, throughout this Chapter, when linear model results are given, it is assumed that exactly the same results could have been obtained from the nonlinear simulation.

The nonlinear model is linearised using a 1st order Taylor expansion about an operating point, $\underline{z}_0$. This consists of body-fixed ship velocities, heading, thruster forces and current angle, $\underline{z} = (u, v, r, \Psi, \beta_C, X_T, Y_T, N_T)$, with wave and wind forces equal to zero. Manipulating equations (2.1) to (2.12) to give expressions for acceleration:

$$\dot{u} = f(\underline{z}) = \dot{u}_0 + \Delta\dot{u} \simeq f(\underline{z}_0) + \frac{\partial f}{\partial u}\Delta u + \frac{\partial f}{\partial v}\Delta v + \frac{\partial f}{\partial r}\Delta r + \frac{\partial f}{\partial X_T}\Delta X_T \qquad (2.44)$$

$$\dot{v} = g(\underline{z}) = \dot{v}_0 + \Delta\dot{v} \simeq g(\underline{z}_0) + \frac{\partial g}{\partial u}\Delta u + \frac{\partial g}{\partial v}\Delta v + \frac{\partial g}{\partial r}\Delta r + \frac{\partial g}{\partial Y_T}\Delta Y_T \qquad (2.45)$$

$$\dot{r} = h(\underline{z}) = \dot{r}_0 + \Delta\dot{r} \simeq h(\underline{z}_0) + \frac{\partial h}{\partial u}\Delta u + \frac{\partial h}{\partial v}\Delta v + \frac{\partial h}{\partial r}\Delta r + \frac{\partial h}{\partial N_T}\Delta N_T \qquad (2.46)$$

At the operating point, $\dot{u}_0$, $\dot{v}_0$, $\dot{r}_0$, $f(\underline{z}_0)$, $g(\underline{z}_0)$ and $h(\underline{z}_0)$ are all equal to zero and so we have the state space model:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + E\underline{d} \qquad (2.47)$$

$$\underline{y} = C\underline{x} \tag{2.48}$$

where

$$A = \begin{bmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} & \frac{\partial f}{\partial r} \\ \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} & \frac{\partial g}{\partial r} \\ \frac{\partial h}{\partial u} & \frac{\partial h}{\partial v} & \frac{\partial h}{\partial r} \end{bmatrix}_{\underline{z}_0}, \; B = E = \begin{bmatrix} \frac{1}{m+m_x} & 0 & 0 \\ 0 & \frac{1}{m+m_y} & 0 \\ 0 & 0 & \frac{1}{I_{zz}+J_{zz}} \end{bmatrix} \tag{2.49}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; \underline{x} = \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta r \end{bmatrix}, \; \underline{u} = \begin{bmatrix} \Delta X_T \\ \Delta Y_T \\ \Delta N_T \end{bmatrix}, \; \underline{d} = \begin{bmatrix} X_A + X_W \\ Y_A + Y_W \\ N_A + N_W \end{bmatrix} \tag{2.50}$$

and the expressions for the partial derivatives are given in Appendix B. Manipulating into transfer function form:

$$\begin{aligned} \underline{x}(s) &= (sI - A)^{-1}B\underline{u}(s) + (sI - A)^{-1}E\underline{d}(s) \\ &= G_V(s)\underline{u}(s) + G_d(s)\underline{d}(s) \end{aligned} \tag{2.51}$$

where G_V and G_d are actually identical as $d(s)$ is just another force input. When under closed-loop control, the ship reference position is given in the earth-fixed coordinate system of equation (2.42). At an operating point, the ship velocity and current velocity cancels so that $u_{abs} = u_0 + \Delta u - u_c = \Delta u$, $v_{abs} = v_0 + \Delta v - v_c = \Delta v$ and $r_{abs} = r_0 + \Delta r - r_c = \Delta r$. The final transfer function, using equation (2.42), from thruster force to earth-fixed positions is:

$$\underline{Y}(s) = \begin{bmatrix} X \\ Y \\ \Psi \end{bmatrix} = \frac{1}{s}T_{BE}\{G_V(s)\underline{u}(s) + G_d(s)\underline{d}(s)\} \tag{2.52}$$

When there are no thruster force changes from the operating point, $\underline{Y}(s) = \frac{1}{s}T_{BE}G_d(s)\underline{d}(s)$, and the earth-fixed position simply depends upon the wind and waves. This is a disturbance, and is independent of feedback control.

2.2.7 Operating point selection

Operating points exist where the ship has settled to a steady state, meaning that no acceleration is occurring. Hence, given (u_0, v_0, r_0), the objective is to find (X_{T0}, Y_{T0}, N_{T0}) such that equations (2.44) to (2.46) are equal to zero. There are potentially infinite permutations of $\underline{z}_0$, so a finite set of representative points must be chosen. The maximum magnitude of the ship thrust vector in the (X_T, Y_T) plane is 470T and N_T may takes values in the interval $[-10^5, 10^5]$Tm. Therefore, values of velocity, u_0, v_0 and r_0, can be selected to produce an evenly distributed subset of thrust vectors from the possible (X_T, Y_T) disc, such as those illustrated in Figure 2.4, and moments from the N_T interval.

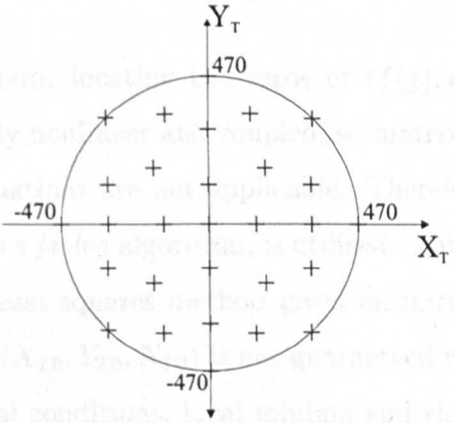

Figure 2.4: Possible ship thrust vectors

A trivial solution occurs when $(u_0, v_0, r_0) = (0, 0, 0)$ and $(X_{T0}, Y_{T0}, N_{T0}) = (0, 0, 0)$. In this case the A matrix in equation (2.49) equals zero, and therefore:

$$\begin{aligned} \underline{x}(s) &= (sI)^{-1}B\underline{u}(s) + (sI)^{-1}E\underline{d}(s) \\ &= \begin{bmatrix} \frac{1}{(m+m_x)s} & 0 & 0 \\ 0 & \frac{1}{(m+m_y)s} & 0 \\ 0 & 0 & \frac{1}{(I_{zz}+J_{zz})s} \end{bmatrix} (\underline{u}(s) + \underline{d}(s)) \end{aligned} \tag{2.53}$$

and

$$\underline{Y}(s) = T_{BE} \begin{bmatrix} \frac{1}{(m+m_x)s^2} & 0 & 0 \\ 0 & \frac{1}{(m+m_y)s^2} & 0 \\ 0 & 0 & \frac{1}{(I_{zz}+J_{zz})s^2} \end{bmatrix} (\underline{u}(s) + \underline{d}(s)) \qquad (2.54)$$

It will be shown in Section 2.4 that T_{BE} can be ignored for controller design, thus the system in equation (2.54) is simply one of diagonal double integrators. Clearly, this is not a difficult multivariable problem as there is no interaction between the three loops. A design can easily be carried out by examining each loop separately as a Single-Input Single-Output problem, and this case will be ignored henceforth.

Otherwise for this system, locating the zeros of $(f(\underline{z}), g(\underline{z}), h(\underline{z}))$ is not easy. The ship model is highly nonlinear and coupled, so matrix techniques for solving linear simultaneous equations are not applicable. Therefore, a nonlinear search routine, namely Matlab's *fsolve* algorithm, is utilised. This algorithm solves nonlinear equations by a least squares method given an initial guess, (X'_T, Y'_T, N'_T). A unique zero for each (X_{T0}, Y_{T0}, N_{T0}) is not guaranteed regardless of (u_0, v_0, r_0), and sensitivity to initial conditions, local minima and slow convergence may be encountered. In practice, however, the *fsolve* algorithm seems to converge to the same minimum regardless of the initial value of (X_T, Y_T, N_T).

Table 2.1 lists eighteen different operating points, intended to cover some of the thrust vectors shown in Figure 2.4, including maximums around the circumference of the circle. The operating points are a function of the incident current, in that a useful steady state is reached when the body-fixed velocities exactly cancel the opposing current velocities. Hence, there is no resultant acceleration or velocity in the earth-fixed coordinates.

Table 2.1: Stable operating points

Current Velocity	Angle	Ship Velocity			Thruster Forces		
U_C	$\beta_C - \Psi$	u_0	v_0	r_0	X_{T0}	Y_{T0}	N_{T0}
(m/s)	$(^o)$	(m/s)	(m/s)	$(^o/s)$	(T)	(T)	(Tm)
1	0	1	0	0	24.5	0	0
2	0	2	0	0	98.0	0	0
3	0	3	0	0	220	0	0
4	0	4	0	0	392	0	0
1	20	0.940	0.342	0	21.8	35.2	2110
1	40	0.766	0.643	0	14.8	100	3260
1	60	0.500	0.866	0	6.96	118	4310
1	80	0.174	0.985	0	1.82	134	1700
1	100	-0.174	0.985	0	0.338	134	-1700
1	120	-0.500	0.866	0	-5.29	118	-4310
1	140	-0.766	0.643	0	-13.9	87.6	-4900
1	160	-0.940	0.342	0	-21.5	46.6	-3200
1	180	-1	0	0	-24.5	0	0
0	*any*	0	0	0.5	18.8	-99.1	9220
0	*any*	0	0	1	75.2	-396	36892
4.38	0	4.38	0	0	470	0	0
1.87	80	0.325	1.84	0	6.35	470	5960
0	*any*	0	0	1.08	87.8	-462	43000

Operating points 1 to 4 are for the ship pointing directly into current of different velocities, and produce thruster excitation in the surge direction only. Operating points 5 to 13 are for fixed current velocity but varying angles of incidence. As the angle passes through 90^o, the surge and yaw thrust signals change in sign, and when the angle reaches 180^o the signals are of equal magnitude but opposite sign to the 0^o case. Operating points 14 and 15 are an attempt to find a non-accelerating state when the ship is rotating. The current velocity has to be set to zero in this case, because u and v will not settle to constant non-zero values for non-zero r_0. The $16th$ and $17th$ operating points give maximum thrust in the surge and sway directions respectively, and the final operating point drives N_T as high as it will go before the (X_T, Y_T) thrust vector hits the maximum.

The controller tuning methods to be used are only applicable to stable linear systems, hence this Chapter will be focussed on the stable operating points given in Table 2.1. A more extensive list could easily be compiled, but the eighteen shown serve to demonstrate general ship behaviour.

The most straightforward linear ship representations are found when $u_0 \in [-4.38, 4.38]$ and $(v_0, r_0) = (0, 0)$. A specific example of this simple case is $(u_0, v_0, r_0) = (2, 0, 0)$, with resulting thruster forces $X_{T0} = 98.0$, $Y_{T0} = 0$, and $N_{T0} = 0$. The open-loop transfer function matrix at this point is:

$$\frac{1}{s}T_{BE}G_V(s) = \begin{bmatrix} \frac{7.6\times10^{-5}}{s(s+0.0075)} & 0 & 0 \\ 0 & \frac{4.8\times10^{-5}(s+0.0092)}{s(s+0.0067\pm j0.0066)} & \frac{3.4\times10^{-9}}{s(s+0.0067\pm j0.0066)} \\ 0 & \frac{-7.6\times10^{-9}}{s(s+0.0067\pm j0.0066)} & \frac{1.1\times10^{-8}(s+0.0041)}{s(s+0.0067\pm j0.0066)} \end{bmatrix} \tag{2.55}$$

There is no interaction of the surge direction with the sway and yaw forces, but a most definite coupling between sway and yaw as depicted in the Bode magnitude plot of $G_V(s)$ in Figure 2.5. Clearly, the sway thruster has more effect on the yaw output than the yaw thruster itself for frequencies below $1rad/s$. For other operating points where v_0 and r_0 are non-zero, the open-loop transfer function matrix contains no zero elements, hence coupling and interaction is more of an issue, and is dealt with in the next Section on tuning methods.

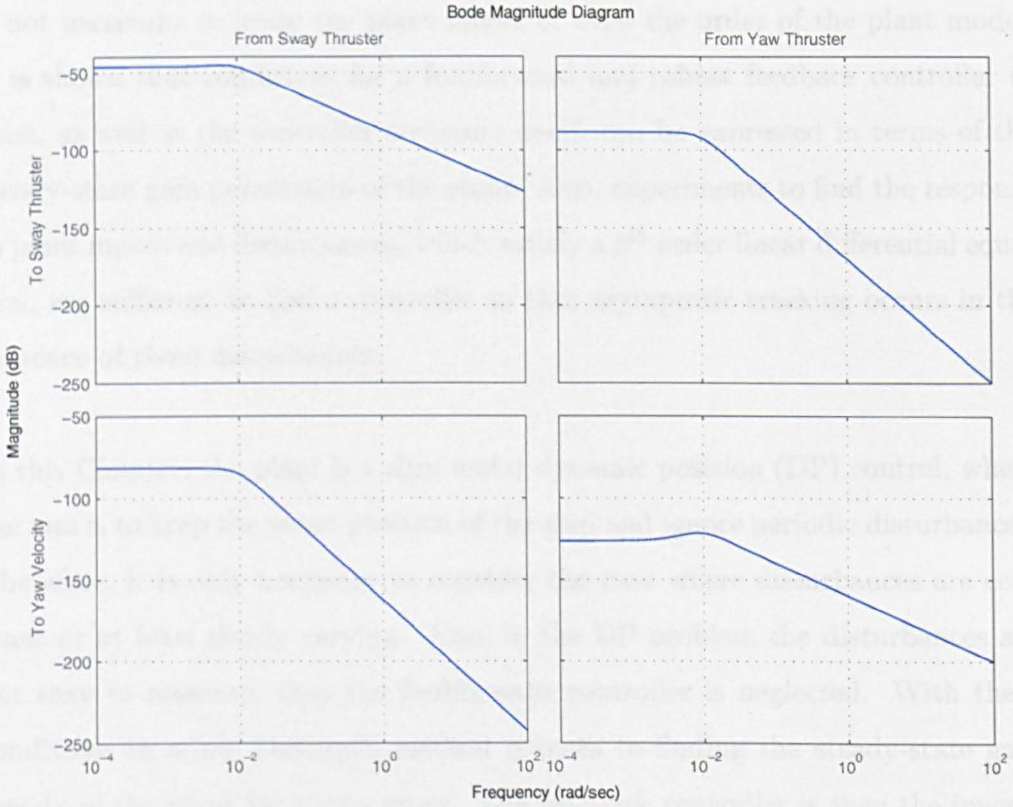

Figure 2.5: Bode sway and yaw magnitude response

2.3 Tuning methods

The techniques of tuning to be covered in this Chapter include the Davison method, the Penttinen and Koivo method, the Maciejowski method, and a new combination of all three.

2.3.1 Davison method

The approach outlined in the Davison (1976) paper is taken with the only assumptions on the plant being that it is linear, time-invariant and open-loop stable. It is not necessary to know the plant model or even the order of the plant model. It is shown that conditions for a feedforward and robust feedback controller to exist, as well as the controller structure itself, can be expressed in terms of the steady-state gain parameters of the plant. Also, experiments to find the response to plant inputs and disturbances, which satisfy a p^{th} order linear differential equation, are sufficient to find a controller so that asymptotic tracking occurs in the presence of these disturbances.

In this Chapter, the plant is a ship under dynamic position (DP) control, where the aim is to keep the mean position of the ship and ignore periodic disturbances. Therefore, it is only necessary to consider the case where disturbances are constant or at least slowly varying. Also, in the DP problem the disturbances are not easy to measure, thus the feedforward controller is neglected. With these conditions in mind, Davison's method reduces to finding the steady-state gain matrix of the plant for a step input. The feedback controller is then the inverse of this matrix, provided that it is of full rank, multiplied by the error signal integral. Note that there is no proportional term in this case and a multiplier, ϵ, is included for tuning the resulting closed-loop system.

The expression for the controller is:

$$\underline{u}(s) = K_i \frac{1}{s} \underline{e}(s), K_i = \epsilon G^{-1}(0) \tag{2.56}$$

where K_i is essentially an integral feedback gain, G(s) is the square open-loop transfer function matrix, and the scalar, ϵ, is the tuning parameter. The procedure for determining ϵ is known as "tuning the regulator on-line" and simply consists of making adjustments, starting with a very small positive value, and increasing so the output response of the closed-loop plant for a step function input has a maximum speed of response. Note that each of the multivariable loops is adjusted simultaneously.

The product of $G(s)$ with K_i approaches ϵI as frequency approaches zero, resulting in a diagonalised cascaded plant and controller at steady-state/low frequencies. The closed-loop transfer function is then

$$\lim_{s\to 0}(I+GK)^{-1}GK = \begin{bmatrix} \frac{\epsilon}{\epsilon} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\epsilon}{\epsilon} \end{bmatrix} = I \tag{2.57}$$

where $K = \frac{K_i}{s}$. From this, it is obvious that non-zero ϵ will produce zero steady-state error.

This approach has been applied successfully to chemical processes, where step tests can be used to find $G(0)$. In the DP case, the ship is simply moved to an operating point before applying step inputs. The deviation from the operating point thruster input and velocity output values is used to find the $G(0)$ matrix.

2.3.2 Penttinen-Koivo method

The Penttinen and Koivo (1980) technique alters the Davison method slightly to achieve a diagonalised plant at very low and very high frequencies. The expression for the controller is:

$$\underline{u}(s) = (K_p + K_i\frac{1}{s})\underline{e}(s), K_p = (CB)^{-1}, K_i = \epsilon G^{-1}(0) \tag{2.58}$$

The CB matrix comes from the state-space plant model, or in the absence of a model it is possible to perform tests to quickly determine the value of CB. Observe that $\dot{\underline{y}} = C\dot{\underline{x}} = CA\underline{x} + CB\underline{u}$. If $\underline{x} = 0$ or the plant is at an operating point, then $\dot{\underline{y}} = CB\underline{u}$ or $\Delta\dot{\underline{y}} = CB\Delta\underline{u}$ at the instant an input is applied. Thus, by applying a unit step to each input in turn and measuring the gradient of each output immediately after:

$$CB = \left[\begin{array}{cccc} \dot{\underline{y}}_1 & \dot{\underline{y}}_2 & \ldots & \dot{\underline{y}}_m \end{array}\right] \tag{2.59}$$

where m is the number of plant inputs and $\dot{\underline{y}}_k$ is the output gradient in response to the k^{th} input step.

The reasoning behind this choice of matrix can most easily be seen using an argument given in Mayne (1979) as follows. Given a plant in state-space form, the Laurent series expansion of the transfer function:

$$G(s) = C(sI - A)^{-1}B \tag{2.60}$$

is:

$$G(s) = \frac{CB}{s} + \frac{CAB}{s^2} + \frac{CA^2B}{s^3} + \ldots \tag{2.61}$$

Therefore, at high frequencies, $G(s) \to CB/s$, and $G(s)K_p \to I/s$.

The proportional gain matrix can be selected as $K_p = (CB)^{-1}$p, where p is a constant scalar tuning parameter. To "tune the regulator on line", p is increased from a small positive value until the closed-loop response for a step-input reference signal is acceptable. p is then reduced slightly and ϵ is increased from a small positive value until the maximum speed of closed-loop response is achieved.

The product of $G(s)$ with K_p and K_i/s approaches pI/s and $CB\epsilon G(0)^{-1}/s^2$ respectively at high frequencies. The K_i/s term will generally be negligible compared to K_p at high frequencies, resulting in a closed-loop transfer function:

$$(I + GK)^{-1}GK = \begin{bmatrix} (s+p)^{-1}p & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (s+p)^{-1}p \end{bmatrix}_{(s\ large)} \tag{2.62}$$

where $K = (K_p + K_i/s)$. From this, it is evident that properly selected p and ϵ will produce good high and low frequency tracking. A brief proof of robustness for stable open-loop plants is given in the Penttinen and Koivo (1980) paper.

2.3.3 Maciejowski method

The approach adopted by Maciejowski (1989) can be interpreted as an interim version of the previous two methods. The technique proposed is to diagonalise the system in the vicinity of bandwidth, ω_b, then to introduce separate controllers in each of the loops of the multivariable system. In keeping with the earlier P+I controller tuning ideas, this yields proportional, integral and derivative matrix terms:

$$K_p = pG^{-1}(j\omega_b), K_i = \epsilon G^{-1}(j\omega_b), K_d = \delta G^{-1}(j\omega_b) \tag{2.63}$$

where p, ϵ and δ are scalar tuning parameters.

If a plant model is available then this method requires the frequency response at a single point. Otherwise, experimental application of sinusoidal inputs to the actual plant at the desired frequency will give values for gain and phase. In the case of a nonlinear system, this experimental approach is not strictly valid, but low amplitude sinusoidal excitation about the operating point yields a very close approximation to the linearised result.

Clearly, $G^{-1}(j\omega_b)$ will produce complex gains, but to realise such a controller, the gains must be real. Hence, the "Align" algorithm of MacFarlane and Kouvaritakis

(1977) is used to produce a real approximation of the inverse of $G(j\omega)$. This algorithm finds a constant real gain matrix, M, such that

$$J(M, \Theta) = (G(j\omega_b)M - e^{j\Theta})^T(G(j\omega_b)M - e^{j\Theta}), \Theta = diag(\theta_1, ..., \theta_n) \qquad (2.64)$$

is minimised. The product of $G(j\omega_b)$ and M is then as close as possible to a diagonal matrix with elements of unity magnitude. If we let $K_p = M$, this produces desirable properties in a multivariable system as each loop will be almost decoupled. To illustrate the effect of using this algorithm, we look at the ideal case where $J(M, \Theta) = 0$. The closed-loop transfer function of this system at $\omega = \omega_b$ will be:

$$(I + G(j\omega_b)K_p)^{-1}G(j\omega_b)K_p = diag(\frac{e^{j\theta_1}}{1 + e^{j\theta_1}}, ..., \frac{e^{j\theta_n}}{1 + e^{j\theta_n}}) \qquad (2.65)$$

From this, it is possible to comment on the gain and phase of the ideal open-loop system, and therefore on the closed-loop system stability.

The gain of the open-loop system at ω_b is unity. The closed-loop gain depends solely on the open-loop phase and is infinite if $\theta = \pi \pm 2n\pi, n \in \mathbb{Z}$. The idea of Maciejowski's method is that the closed-loop bandwidth is ω_b. This occurs when $\theta = -\pi/2 \pm n\pi, n \in \mathbb{Z}$. Here, $\frac{e^{j(\frac{\pi}{2} \pm n\pi)}}{1+e^{j(\frac{\pi}{2} \pm n\pi)}} = 0.5 \pm 0.5j$ and the closed-loop gain is $\frac{1}{\sqrt{2}}$. Of course, as the open-loop phase is $-\pi/2 \pm n\pi$, the closed-loop system will be stable for $\theta = \pm\pi/2$ with negative feedback and unstable for all other n, by Bode plot phase margin considerations.

In summary, this method aims to create nearly-decoupled unity gain open-loop transfer functions from a coupled transfer function matrix. If the open-loop phase is close to $-\pi/2$, then the bandwidth will be close to ω_b and the closed-loop system will be stable. Of course, this is the ideal case and if the open-loop phase is greater than $-\pi/2$, the bandwidth will be less than ω_b. Conversely, open-loop phase lower than $-\pi/2$ will result in a higher bandwidth. If the open-loop phase is $-\pi$ or less, the system will be unstable. This analysis applies to proportional control only, as M is a constant gain matrix. Fine tuning may be achieved with the addition of the integral or derivative terms in equations (2.63). Again, following on from the preceding methods, the p, ϵ and δ parameters should be increased from small positive values until the desired closed-loop performance is achieved.

2.3.4 Combined method

A simple logical extension to the three methods above is to use the Maciejowski diagonalisation in K_p, the steady state gain inverse in K_i, and Penttinen and Koivo's idea for proportional gain in K_d. Therefore:

$$K_p = pG^{-1}(j\omega_b), K_i = \epsilon G^{-1}(0), K_d = \delta(CB)^{-1} \tag{2.66}$$

The motivation is that each gain is suited to each PID term as a consequence of frequency domain characteristics. The integrator is dominant over the other terms at zero frequency, where K_i produces complete decoupling. The derivative is dominant at very high frequencies and K_d removes coupling there. K_p is midway and attempts to remove interaction around the bandwidth.

2.3.5 Tuning methods - Discussion of limitations

The four methods detailed are limited as design tools in that Davison and Penttinen-Koivo depend upon a stable open-loop plant for their robustness and stability proofs. There is no such proof for Maciejowski's method - The fact that $|GK|$ can be set close to the identity matrix, I, at or near the selected bandwidth does not give any guarantees. However, for individual design cases the methods may still be applied and stability and robustness assured by inspection. In the case of multivariable plants, the Generalised Nyquist Stability Criterion (GNSC), see

Maciejowski (1989), is the favoured design aid. The idea of the GNSC is that knowing the number of open-loop unstable poles of $det[G(s)K(s)]$ allows us to decide whether the closed-loop will have unstable poles based on the Characteristic Loci. These Loci are the graphs of the eigenvalues of $G(s)K(s)$ as s goes around the Nyquist contour. If the number of clockwise encirclements of the -1 point equals the number of unstable open-loop poles of $det[G(s)K(s)]$, then the closed-loop system is stable.

Another limitation of the design methods is in the process of "tuning the regulator on line". Increasing the various scalar tuning parameters from zero until the desired response is achieved does not take into account input constraints or cross-coupling of the multivariable loops. Hence, in the design investigation, it is necessary to trade off the speed of response, disturbance rejection, decoupling, and actuator constraints against each other.

2.4 Ship control

Before applying any of the tuning methods it is necessary to examine the particular characteristics of the ship control problem that are likely to influence and constrain any final design. The forces incident on a vessel due to the environment - wind, waves, current - can be considerably greater than the available force from the ship's thrusters. Thus, it is necessary to both avoid actuator saturation and to save fuel by ignoring disturbances which cannot be effectively cancelled. To this end, filters can be employed to attenuate non-essential components of the spectrum of measured variables. To be specific, notch filtering of position measurement at the frequency of dominant wave exciting forces, $\omega_n = 0.6rad/s$, produces good steady-state tracking, whilst removing high-frequency zero-mean forces that do not affect the average position of the ship.

The transfer function matrix of the filters to be used in the design is:

$$N(s) = \frac{s^2 + 0.2\omega_n s + \omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \times I = n(s)I \tag{2.67}$$

with the Bode plot for $n(s)$ given in Figure 2.6. The 'depth' of the notch depends on the relation between the coefficients of $\omega_n s$ in the numerator and denominator. $\frac{1}{10} \times 2 = 0.2$, so the notch is $-20dB$. A numerator coefficient equal to 0.02, or $\frac{1}{100}$ of 2 gives a notch of $-40dB$, and so on.

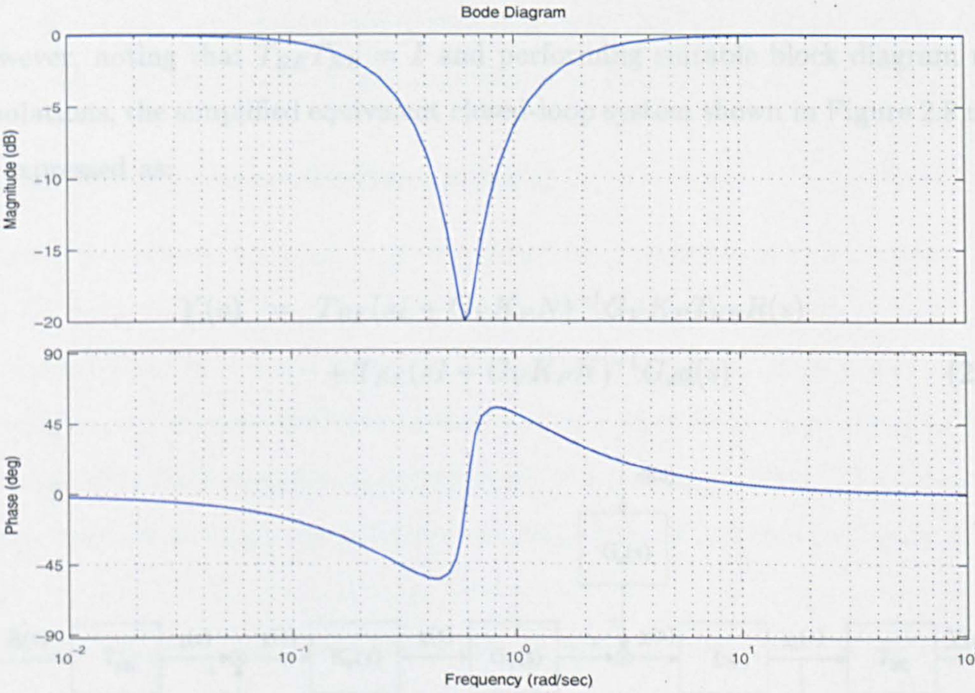

Figure 2.6: Notch filter Bode plot

The resulting closed-loop ship control with transfer function:

$$\begin{aligned}\underline{Y}(s) &= (sI + T_{BE}G_V K_P T_{EB} N)^{-1} T_{BE} G_V K_P T_{EB} \underline{R}(s) \\ &\quad + (sI + T_{BE}G_V K_P T_{EB} N)^{-1} T_{BE} G_d \underline{d}(s)\end{aligned} \tag{2.68}$$

is shown in Figure 2.7.

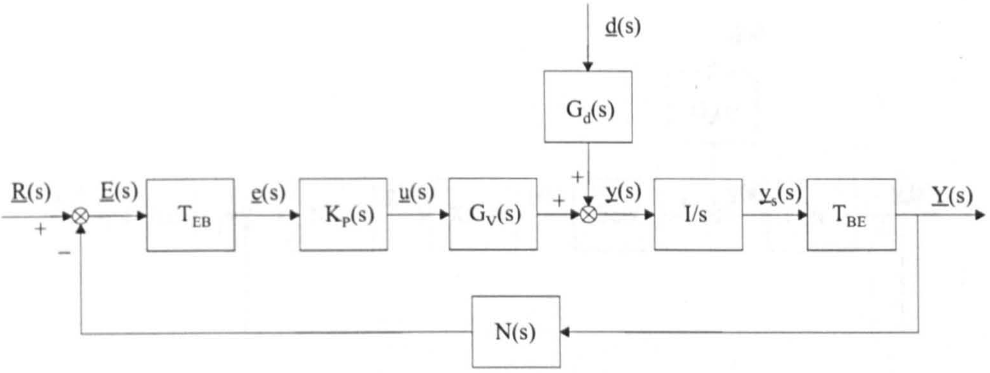

Figure 2.7: Control scheme with output notch filter

However, noting that $T_{BE}T_{EB} = I$ and performing suitable block diagram manipulations, the simplified equivalent closed-loop system shown in Figure 2.8 may be expressed as:

$$\begin{aligned} \underline{Y}(s) &= T_{BE}(sI + G_V K_P N)^{-1} G_V K_P T_{EB} \underline{R}(s) \\ &\quad + T_{BE}(sI + G_V K_P N)^{-1} G_d \underline{d}(s) \end{aligned} \tag{2.69}$$

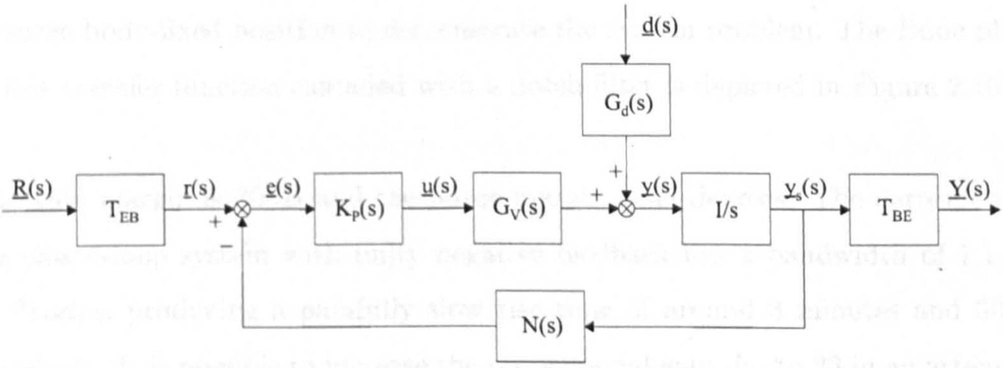

Figure 2.8: Rearranged control scheme with output notch filter

The coordinate transformations do not appear within this equivalent system closed loop and it is clear that their effect can be ignored for control design. A final block manipulation of $N(s)$ produces the equivalent unity feedback system in Figure 2.9, neglecting the coordinate transformations.

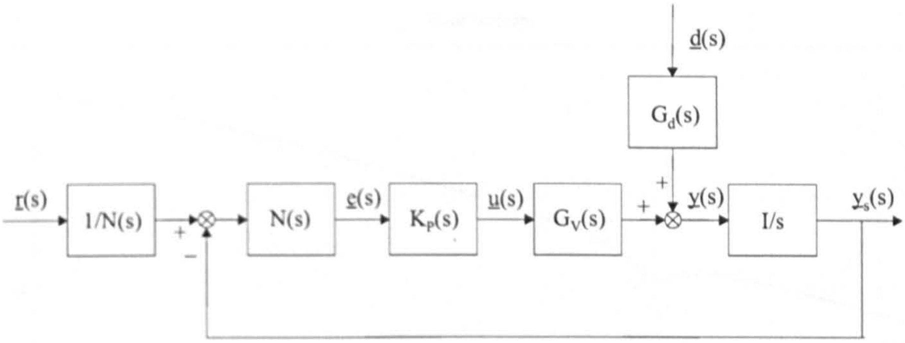

Figure 2.9: Unity feedback control scheme

$N(s)$ is diagonal and hence interchangeable with $K_P(s)$ or $G_V(s)$ in the loop, hence the open-loop plant to be compensated is $N(s)G_V(s)/s$.

Limitation of position feedback scheme

There is a problem, however, with the closed-loop response of this system. This can be demonstrated by examination of transfer function matrix (2.55). This expression is for $\Psi = 0$ and therefore is equal to $G_V(s)/s$. It is necessary only to investigate the transfer function in the top left corner from surge thruster force to surge body-fixed position to demonstrate the system problem. The Bode plot of this transfer function cascaded with a notch filter is depicted in Figure 2.10.

The gain margin is $30dB$ and the phase margin is 42 degrees. The corresponding closed-loop system with unity negative feedback has a bandwidth of $1.1 \times 10^{-2} rad/s$, producing a painfully slow rise time of around 3 minutes and 30% overshoot. It is possible to increase the proportional gain, k_p, to 33 in an attempt to increase the bandwidth, but the closed-loop response becomes marginally stable. At this point derivative action, k_d, may be introduced in an attempt to stabilise the system and decrease the overshoot. $k_d = 100$ increases the bandwidth to $7.2 \times 10^{-2} rad/s$ but results in a highly oscillatory response. Reducing k_p to 1 gives an overshoot of 3% but slow response once again. Removing the propor-

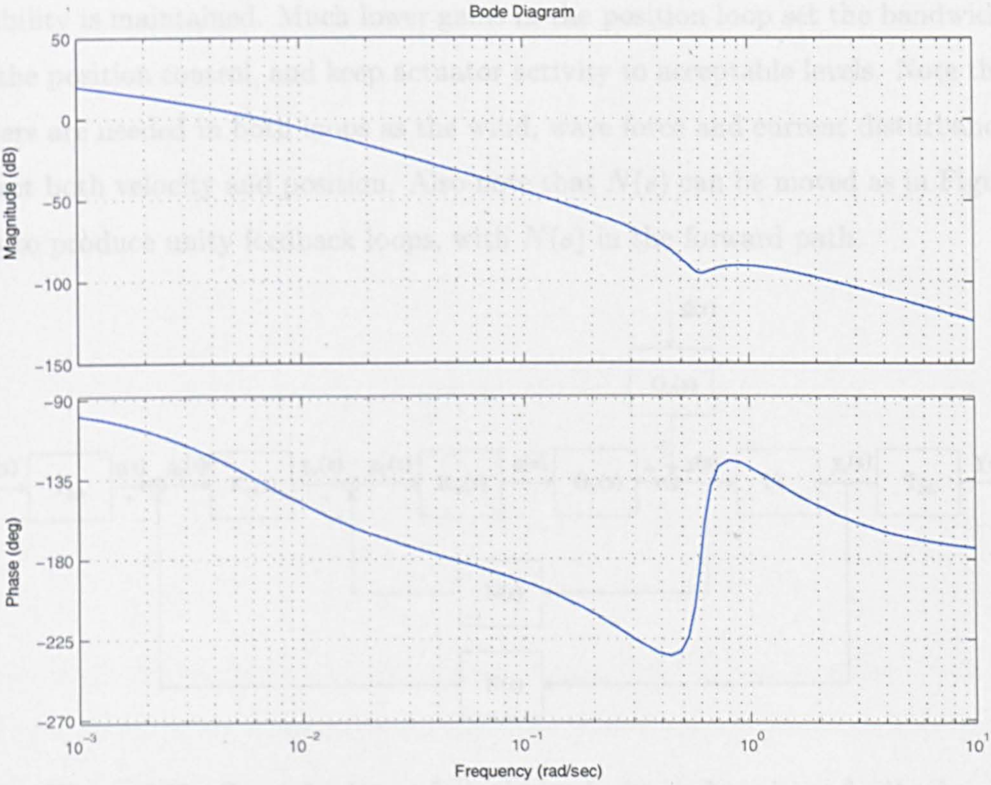

Figure 2.10: Bode plot of top left element of $\frac{G_V}{s}N$

tional gain altogether and increasing k_d to 1000 produces a rise time of around 1 minute, but with 17% steady-state error. Introducing integral gain is simply not possible, as the extra phase shift removes any possibility of satisfactory response. From this quick investigation it becomes clear that the phase shift introduced by notch filtering invariably leads to undesirable response or destabilisation of the closed-loop system, as has been noted before in Grimble et al. (1980a).

Position and velocity feedback

One solution to this problem without abandoning PID control completely is to introduce a velocity feedback loop as illustrated in Figure 2.11. If the gain $K_V(s)$ is sufficiently large, then the inner loop can provide enough phase margin such that notch filters can be introduced while closed-position-loop performance and stability is maintained. Much lower gains in the position loop set the bandwidth of the position control, and keep actuator activity to acceptable levels. Note that filters are needed in both loops as the wind, wave force and current disturbances affect both velocity and position. Also note that $N(s)$ can be moved as in Figure 2.9 to produce unity feedback loops, with $N(s)$ in the forward path.

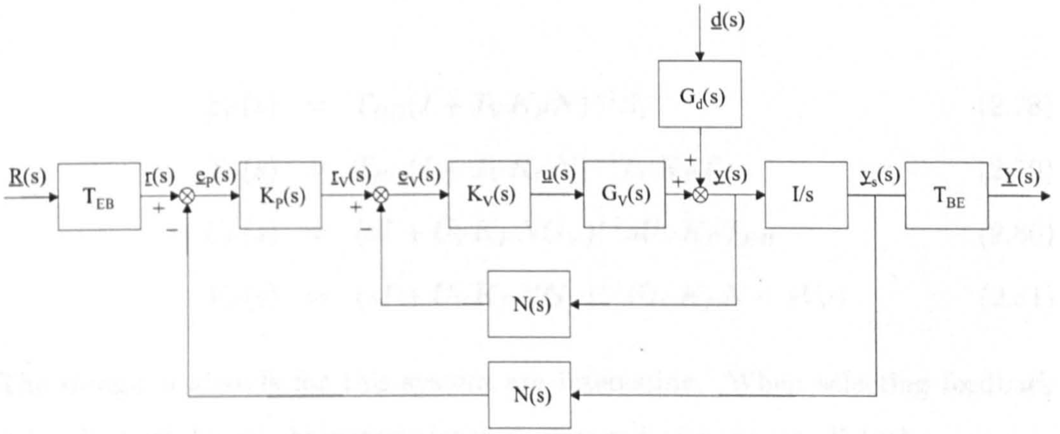

Figure 2.11: Control scheme featuring velocity and position feedback

The closed-loop transfer functions of note in the velocity loop are:

$$\underline{y}_s(s) = T_V(s)\underline{r}_V(s) + S_V(s)G_d(s)\underline{d}(s) \tag{2.70}$$

$$\underline{u}(s) = U_V(s)\underline{r}_V(s) - V_V(s)G_d(s)\underline{d}(s)) \tag{2.71}$$

where

$$S_V(s) = \frac{I}{s}(I + G_V K_V N)^{-1} \tag{2.72}$$

$$T_V(s) = \frac{I}{s}(I + G_V K_V N)^{-1} G_V K_V \tag{2.73}$$

$$U_V(s) = (I + K_V N G_V)^{-1} K_V \tag{2.74}$$

$$V_V(s) = (I + K_V N G_V)^{-1} K_V N \tag{2.75}$$

In the position loop, the closed-loop transfer functions of note are:

$$\underline{Y}(s) = T_P(s)\underline{R}(s) + S_P(s)G_d(s)\underline{d}(s) \quad (2.76)$$

$$\underline{u}(s) = U_P(s)\underline{R}(s) - V_P(s)G_d(s)\underline{d}(s) \quad (2.77)$$

where

$$S_P(s) = T_{BE}(I + T_V K_P N)^{-1} S_V \quad (2.78)$$

$$T_P(s) = T_{BE}(I + T_V K_P N)^{-1} T_V K_P T_{EB} \quad (2.79)$$

$$U_P(s) = (sI + U_V K_P N G_V)^{-1} s U_V K_P T_{EB} \quad (2.80)$$

$$V_P(s) = (sI + U_V K_P N G_V)^{-1}(U_V K_P N + s V_V) \quad (2.81)$$

The design trade-offs for this system are interesting. When selecting feedback gains for both loops, characteristics such as speed of response, disturbance rejection, decoupling, and actuator constraints must all be taken into account. High gains may produce fast responses and good disturbance rejection but with unnacceptable actuator activity. Reducing the gain may bring thruster forces to within the constraints, but the system will respond slowly and be pushed away from the setpoint more easily. The problem of coupling may also appear, whereby control energy is expended on one input to cancel the effect of another control input. This interaction is clearly undesirable and gives an inefficient overall control system, but may have to be tolerated to some extent given other possibly opposing design issues.

Now that the requirements of the control are clear, the tuning methods can be applied to test their utility. Once again, the linearised system of equation (2.55) at $(u_0, v_0, r_0) = (2, 0, 0)$ will be under investigation.

2.4.1 Velocity loop design

Davison method

Beginning with velocity loop design and the Davison method, values for $(N(0)G_V(0))^{-1}$ are required. Noting that $N(0) = I$, therefore:

$$K_V(s) = \frac{K_i}{s} = \epsilon_V \frac{G_V(0)^{-1}}{s} = \frac{1}{s}\begin{bmatrix} 99.0 & 0 & 0 \\ 0 & 85.6 & -6630 \\ 0 & 14900 & 872000 \end{bmatrix}_{\epsilon_V=1} \tag{2.82}$$

This method does produce a stable closed-loop system with zero steady-state error as stated in Section 2.3.1 - The Characteristic Loci of $G_V(s)K_V(s)N(s)$ shows that the closed-loop system is stable for $0 < \epsilon_V \leq 0.012$. However, integral action only is included here, so the problem of excessive phase lag remains. The Bode plot of the top left element of $G_V(s)K_V(s)N(s)$ is identical to Figure 2.10 but for the scaling factor, $99\epsilon_V$. As noted in Section 2.4, the performance of such a system in closed-loop is unacceptable. No advantage would be gained, in terms of greater phase margin for a notch filter, by using velocity feedback with this controller, hence the Davison method is of no use in this case.

Penttinen-Koivo method

With the Penttinen-Koivo method, $(CB)^{-1}$ corresponding to NG_V is required. $N(s)_{s\to\infty} = I$, so this yields:

$$\begin{aligned} K_V(s) &= K_p + \frac{K_i}{s} = p_V(CB)^{-1} + \epsilon_V\frac{G_V(0)^{-1}}{s} \\ &= \begin{bmatrix} 13100 & 0 & 0 \\ 0 & 20700 & 0 \\ 0 & 0 & 94500000 \end{bmatrix}_{p_V=1} + \frac{1}{s}\begin{bmatrix} 99.0 & 0 & 0 \\ 0 & 85.6 & -6630 \\ 0 & 14900 & 872000 \end{bmatrix}_{\epsilon_V=1} \end{aligned} \tag{2.83}$$

The problems with the Davison method may be overcome using the above controller. In fact, bearing in mind that the velocity loop was introduced in order to create extra phase margin for the position notch filter, it may seem wise to use proportional-only control. However, removing the $G_V(0)^{-1}/s$ term from the controller by setting $\epsilon_V = 0$ raises the issue of interaction between the sway and yaw directions, as ilustrated by the Bode plot of $G_V K_V N$ in Figure 2.12. High-frequency interaction is not a problem when $p_V = 1, \epsilon_V = 0$, as the off-diagonal elements roll-off at $40dB/decade$ compared to $20dB/decade$ for the diagonals, but the system is coupled at low frequencies. Introducing a small amount of integral action produces steady state decoupling without much penalty in phase margin, therefore a non-zero value of ϵ_V will be used.

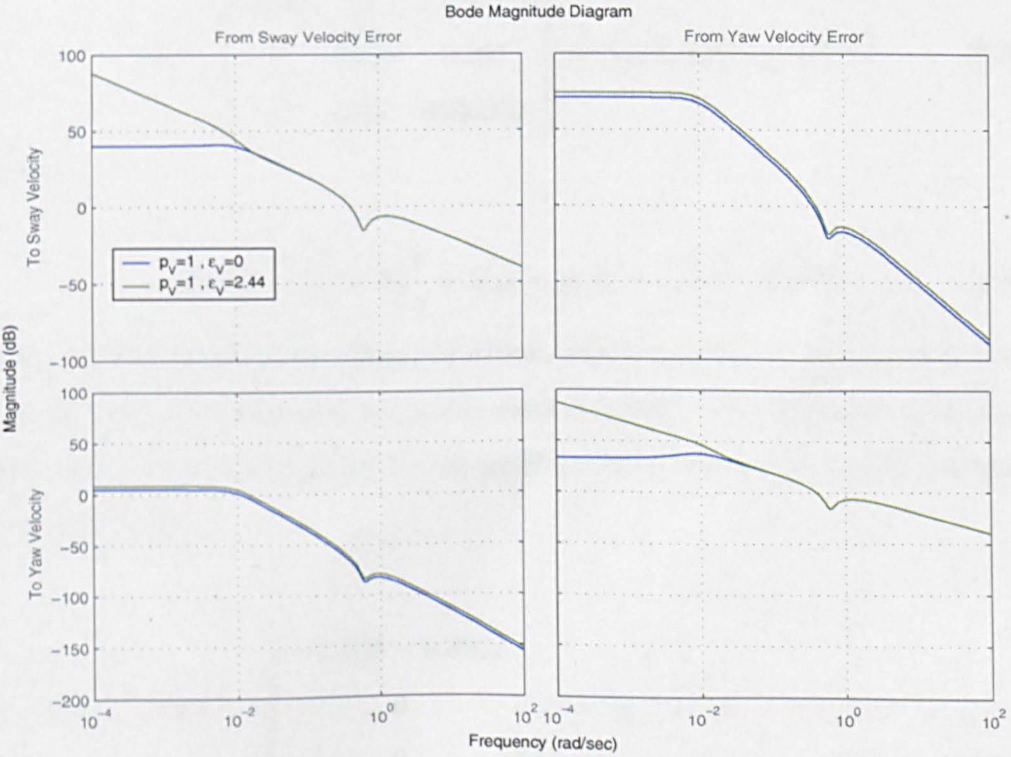

Figure 2.12: Bode sway and yaw magnitude response with notch filter

Setting $p_V = 1$ and adding the integral term, the closed-loop system is stable for values of ϵ_V from 0 to 24.4. Selecting $\epsilon_V = 2.44$, or 10% of the unstable value, is a good starting point for tuning and provides robustness to uncertainty of the ship frequency domain characteristics.

Maciejowski method

The Maciejowski method involves selecting a desired bandwidth for the system and attempting to decouple the sway and yaw directions using the Align algorithm, which finds an approximate plant inverse. The bandwidth in this case is selected to be $0.2rad/s$, so that the system responds faster than the position-feedback-only case in Figure 2.10, but does not attempt to reject high frequency wave exciting forces. This gives:

$$K = \begin{bmatrix} 3280 & 0 & 0 \\ 0 & 5160 & -107 \\ 0 & 385 & 6630000 \end{bmatrix} \approx (G_V(j0.2)N(j0.2))^{-1} \tag{2.84}$$

where

$$K_V(s) = K_p + K_i\frac{1}{s} + K_d s = p_V K + \epsilon_V K\frac{1}{s} + \delta_V K s \tag{2.85}$$

However, the nature of the Align algorithm, which is to find a real approximation to a complex matrix inverse, precludes exact inversion. The consequence of this is that the product of $G_V(j0.2)N(j0.2)$ and the rough inverse, K, is fairly coupled as shown below:

$$G_V(j0.2)KN(j0.2) = \begin{bmatrix} -0.506 - 0.862j & 0 & 0 \\ 0 & -0.521 - 0.854j & -0.384 + 0.234j \\ 0 & 0.000684 - 0.000394j & -0.140 - 0.244j \end{bmatrix} \tag{2.86}$$

The gain from sway input to sway output is twice that from the yaw input, and the gain from yaw input to yaw output is much greater than from the sway input. However, the gain magnitude from yaw input to sway output is 60% greater than the gain to the yaw output. Therefore, the sway thruster will have to work to compensate sway errors due to the yaw thruster. It is possible for the system to be more decoupled if a higher bandwidth is chosen. However, the reason for choosing $0.2rad/s$ bandwidth is that this will produce good disturbance rejection below $0.2rad/s$ and avoid excessive thruster forces.

Like the Penttinen-Koivo method, it is possible to include integral action without destabilising the closed-loop system. Setting $p_V = 1$ and adding the integral term, the closed-loop system is stable for values of ϵ_V from 0 to 0.31, hence the 10% value, $\epsilon_V = 0.031$, is used as an initial value for controller tuning. Non-zero ϵ_V is unsuccessful in low frequency decoupling, unlike the Penttinen-Koivo approach, by inspection of Figure 2.13. This is likely to result in an inefficient controller.

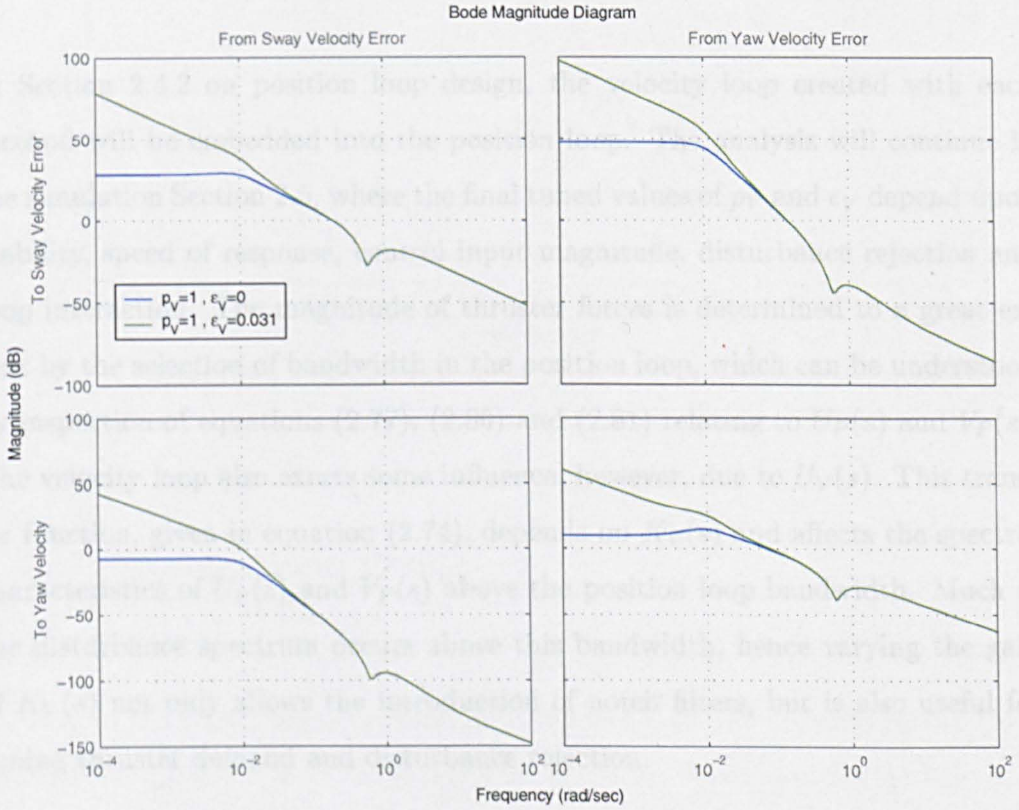

Figure 2.13: Bode sway and yaw magnitude response with notch filter

δ_V is set to zero for the ship example, in order to keep actuator excursion low. This also applies to the next method.

Combined method

In the combined case, equation (2.84) is used in the K_p term and equation (2.82) in the K_i term. Thus:

$$K_V(s) = K_p + K_i\frac{1}{s} = p_V K + \epsilon_V \frac{G_V(0)^{-1}}{s} \tag{2.87}$$

Setting $p_V = 1$ and considering the integral term, the closed-loop system is stable for values of ϵ_V from 0 to 1.53, suggesting $\epsilon_V = 0.153$ as a first value for tuning the controller. Again, non-zero ϵ_V will be used for decoupling at low frequencies, producing a Bode plot of $G_V K_V N_V$ similar to Figure 2.12 rather than Figure 2.13.

In Section 2.4.2 on position loop design, the velocity loop created with each method will be embedded into the position loop. The analysis will continue in the simulation Section 2.5, where the final tuned values of p_V and ϵ_V depend upon stability, speed of response, control input magnitude, disturbance rejection and loop interaction. The magnitude of thruster forces is determined to a great extent by the selection of bandwidth in the position loop, which can be understood by inspection of equations (2.77), (2.80) and (2.81) relating to $U_P(s)$ and $V_P(s)$. The velocity loop also exerts some influence, however, due to $U_V(s)$. This transfer function, given in equation (2.74), depends on $K_V(s)$ and affects the spectral characteristics of $U_P(s)$ and $V_P(s)$ above the position loop bandwidth. Much of the disturbance spectrum occurs above this bandwidth, hence varying the gain of $K_V(s)$ not only allows the introduction of notch filters, but is also useful for tuning thruster demand and disturbance rejection.

The improvement over a system without velocity feedback can be demonstrated by examining the left upper transfer function of $T_V(s)$. A frequency response plot when cascaded with a notch filter is shown in Figure 2.14 with the indicated gains. Clearly, contrasting this with the Bode plot for the ship with no velocity feedback in Figure 2.10, it is immediately obvious that the phase response roll-off occurs at a higher frequency. Similar results are seen for the sway and yaw loops.

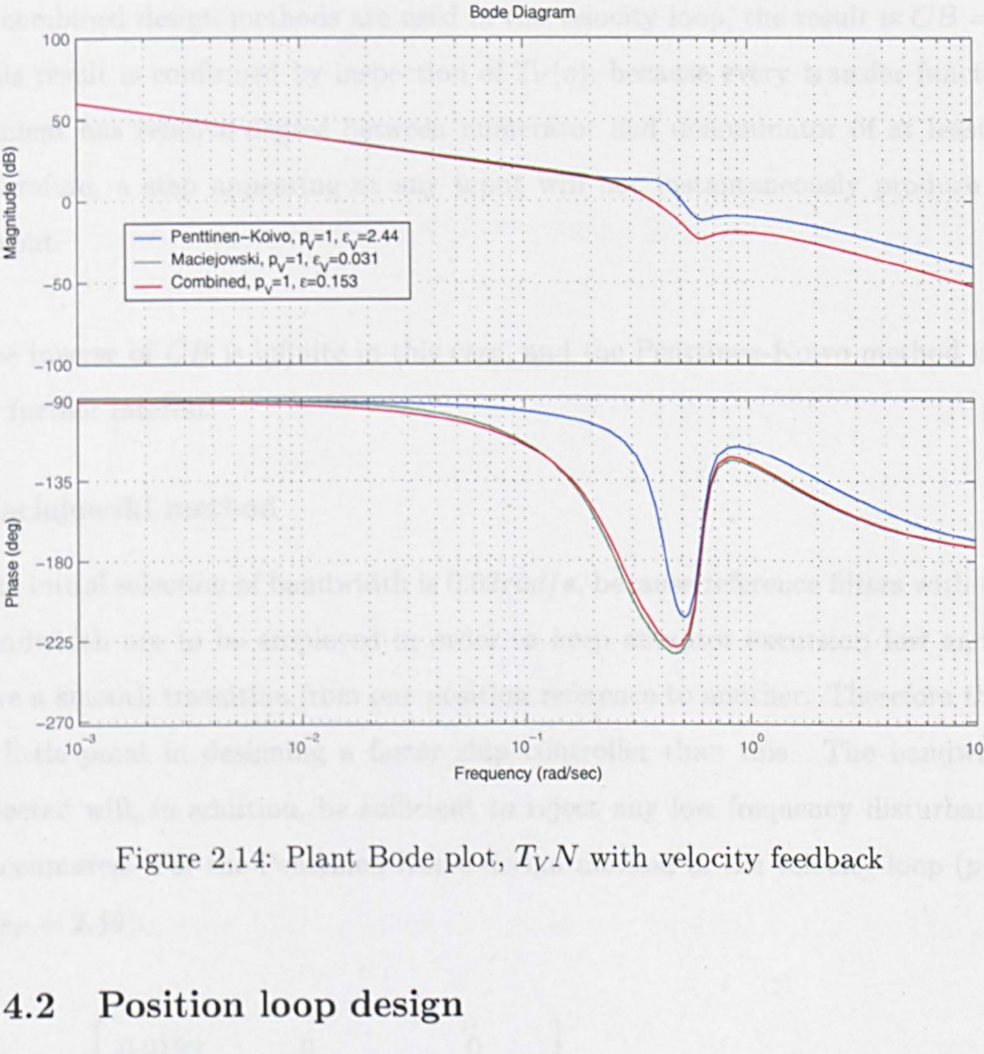

Figure 2.14: Plant Bode plot, $T_V N$ with velocity feedback

2.4.2 Position loop design

Davison method

Moving to the position loop control, it is evident that the Davison method will produce zero gains, due to the integrators in $T_V(s)$. Therefore, the method is of no value in this case either.

Penttinen-Koivo method

The Penttinen-Koivo method presents another difficulty because there is no readily available state-space representation for $T_V(s)$. Hence the method of identifying CB from step tests is utilised. However, when the Penttinen-Koivo, Maciejowski or combined design methods are used in the velocity loop, the result is $CB = 0$. This result is confirmed by inspection of $T_V(s)$, because every transfer function element has relative degree between numerator and denominator of at least 2. Therefore, a step appearing at any input will not instantaneously produce an output.

The inverse of CB is infinite in this case, and the Penttinen-Koivo method is of no further interest.

Maciejowski method

The initial selection of bandwidth is $0.02 rad/s$, because reference filters with this bandwidth are to be employed in order to keep actuator excursion low and to give a smooth transition from one position reference to another. Therefore there is little point in designing a faster ship controller than this. The bandwidth selected will, in addition, be sufficient to reject any low frequency disturbances encountered. For the Penttinen-Koivo design method in the velocity loop ($p_V = 1, \epsilon_V = 2.44$):

$$K = \begin{bmatrix} 0.0199 & 0 & 0 \\ 0 & 0.0198 & -0.00125 \\ 0 & 0.000000652 & 0.0185 \end{bmatrix} \approx (T_V(j0.02)N(j0.02))^{-1} \qquad (2.88)$$

where

$$K_P(s) = (K_p + K_i\frac{1}{s} + K_d s) = (p_P K + \epsilon_P K\frac{1}{s} + \delta_P K s) \qquad (2.89)$$

$$T_V(j0.02)N(j0.02)K =$$

$$\begin{bmatrix} -0.0144-0.999j & 0 & 0 \\ 0 & -0.0143-0.999j & -0.254+0.00362j \\ 0 & 0.000135-0.00000133j & -0.00927-0.931j \end{bmatrix} \qquad (2.90)$$

The system is clearly well decoupled at this frequency, and once again the Characteristic Loci is plotted to provide stable values for p_P and ϵ_P. When $p_P = 1$, the integral action scaling may take values $0 < \epsilon_P < 0.91$.

For the Maciejowski design method in the velocity loop ($p_V = 1, \epsilon_V = 0.031$):

$$K = \begin{bmatrix} 0.0194 & 0 & 0 \\ 0 & 0.0193 & -0.00656 \\ 0 & 0.0000127 & 0.0138 \end{bmatrix} \approx (T_V(j0.02)N(j0.02))^{-1} \qquad (2.91)$$

$$T_V(j0.02)N(j0.02)K =$$

$$\begin{bmatrix} -0.0367-0.999j & 0 & 0 \\ 0 & -0.0308-0.999j & 0.428-0.0132j \\ 0 & -0.00113+0.000175j & -0.115-0.749j \end{bmatrix} \qquad (2.92)$$

The system is also fairly well decoupled at this frequency, and the Characteristic Loci indicates stability for $p_P = 1$ and $0 < \epsilon_P < 0.08$.

Using the combined design method in the velocity loop ($p_V = 1, \epsilon_V = 0.153$):

$$K = \begin{bmatrix} 0.0202 & 0 & 0 \\ 0 & 0.0202 & 0.00857 \\ 0 & 0.00000713 & 0.0106 \end{bmatrix} \approx (T_V(j0.02)N(j0.02))^{-1} \qquad (2.93)$$

$$T_V(j0.02)N(j0.02)K =$$

$$\begin{bmatrix} -0.0826-0.997j & 0 & 0 \\ 0 & -0.0780-0.997j & -0.493+0.0385j \\ 0 & -0.000580+0.000128j & -0.124-0.562j \end{bmatrix} \qquad (2.94)$$

The system is decoupled, although the magnitude of the bottom right element is only 24% greater than the element above. The Characteristic Loci indicates stability for $p_P = 1$ and $0 < \epsilon_P < 0.073$.

The top left transfer function of $V_P(s)G_d(s)$, from equation (2.77), and corresponding Bode plot, in Figure 2.15, demonstrates the presence of a notch in the frequency response from wave disturbances to the thruster input at $0.6 rad/s$. The three plots are for different tuning methods in the velocity loop and with the gains indicated. Similar plots may be obtained for the sway and yaw loops.

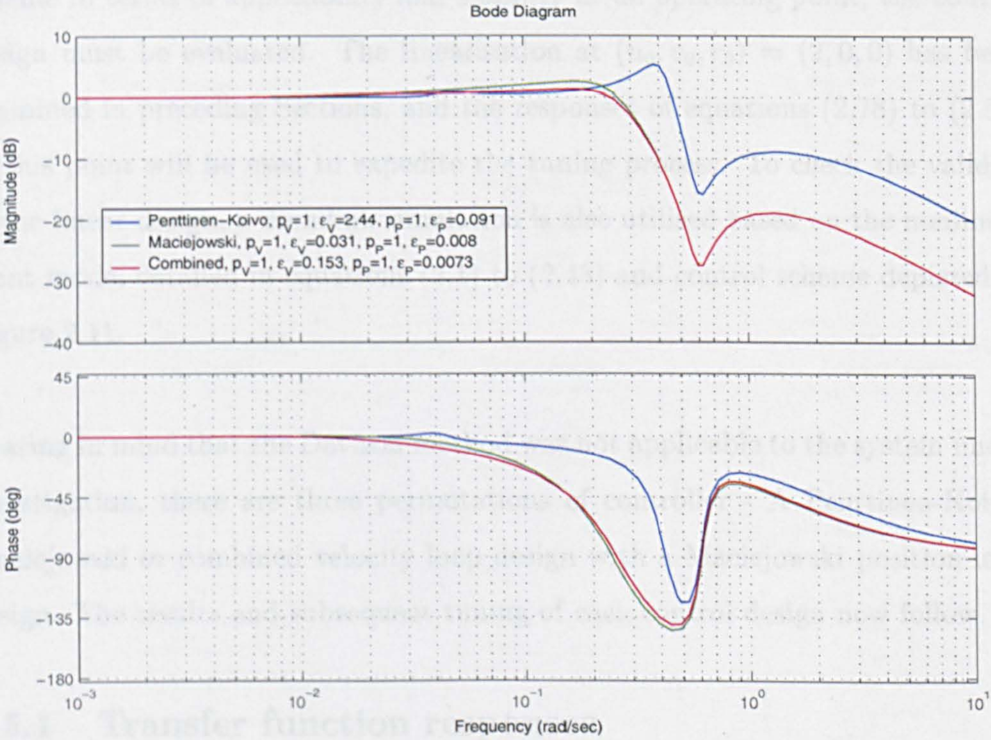

Figure 2.15: Disturbance-to-control-input Bode plot

Combined method

As with the Davison method, the K_i gains will be zero, hence this is just a special case of the Maciejowski method where $\epsilon_P = 0$.

For each position loop controller, $\delta_P = 0$ to avoid large thruster forces, and ϵ_P is chosen to be 10% of the upper limit of stable values as a first guess for controller tuning. The final values of p_P and ϵ_P will depend on stability, control input magnitude, overshoot, settling time and decoupling as given in the next Section on simulation.

2.5 Controller tuning and simulation

Having assessed the tuning methods for suitability in the dynamic position control scheme in terms of applicability and stability at an operating point, the control design must be evaluated. The linearisation at $(u_0, v_0, r_0) = (2, 0, 0)$ has been examined in preceding Sections, and the responses of equations (2.78) to (2.81) at this point will be used to expedite the tuning process. To check the validity of the linear design, a Simulink simulation is also utilised based on the nonlinear plant model detailed in equations (2.1) to (2.43) and control scheme depicted in Figure 2.11.

Bearing in mind that the Davison method was not applicable to the system under investigation, there are three permutations of controller - A Penttinen-Koivo, Maciejowski or combined velocity loop design with a Maciejowski position loop design. The results and subsequent tuning of each control design now follow.

2.5.1 Transfer function responses

To make a quick initial assessment of each controller, responses from equations (2.78) to (2.81) are required. These transfer functions are linear representations of the system at an operating point, so provided that the ship remains in the neighbourhood of this point, the responses should closely agree with the nonlinear simulation. In Section 2.4.2 it was noted that reference filters of bandwidth $0.02 rad/s$ are to be employed to keep actuator excursion low and to give a smooth transition from one position reference to another. In the simulation, first order lags with transfer function, $F_R(s)$, are used to achieve this:

$$\underline{R}(s) = F_R(s)\underline{R}_{demand}(s) = diag\{\frac{1}{50s+1}, \frac{1}{50s+1}, \frac{1}{50s+1}\}\underline{R}_{demand}(s) \quad (2.95)$$

Figure 2.16 shows the multivariable responses of $T_P(s)F_R(s)$ to unit step reference demands on each input, with zero disturbance. For the Pentinnen-Koivo velocity loop case, the parameters are ($p_V = 1, \epsilon_V = 2.44, p_P = 1, \epsilon_P = 0.091$). In the Maciejowski velocity loop case, the values are ($p_V = 1, \epsilon_V = 0.031, p_P = 1, \epsilon_P = 0.008$), and in the combined case, ($p_V = 1, \epsilon_V = 0.153, p_P = 1, \epsilon_P = 0.0073$).

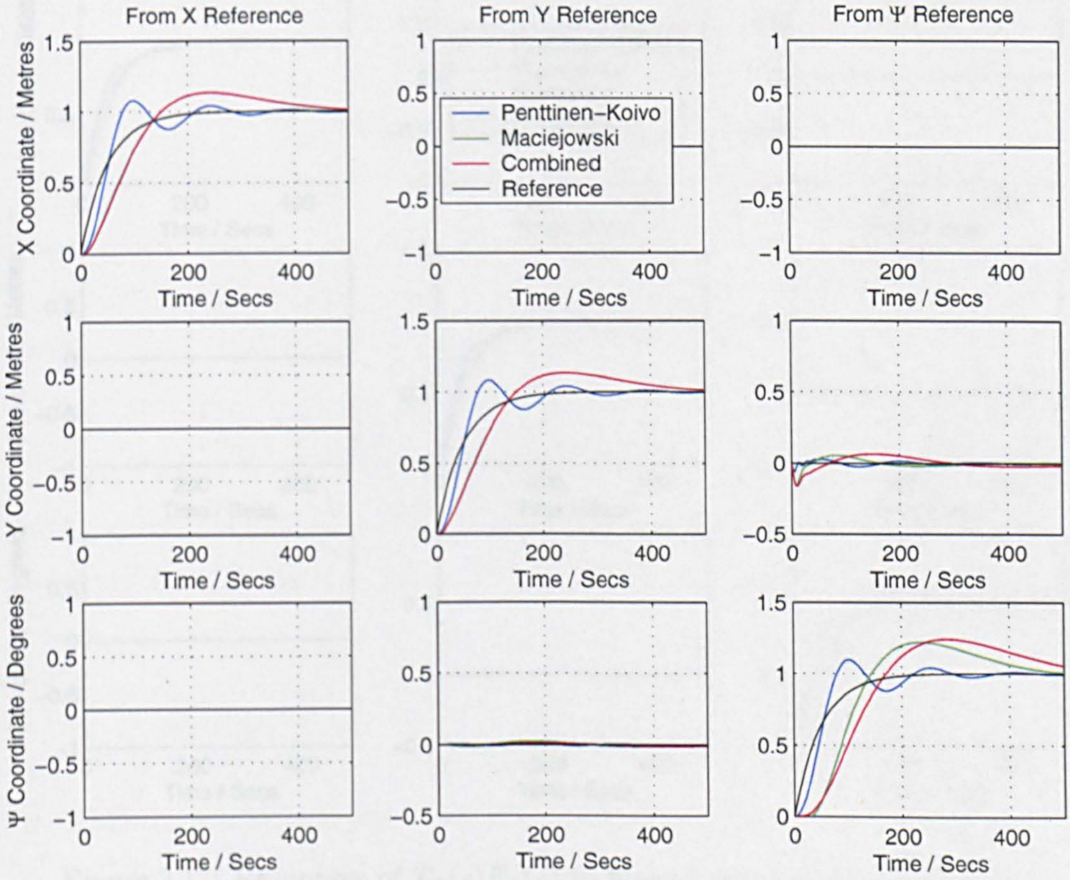

Figure 2.16: Responses of $T_P(s)F_R(s)$ to filtered step reference demand

Cross-coupling is not a problem, judging from the Y response to a step in Ψ and vice versa. However, the diagonal elements exhibit considerable overshoot and long settling times for all three cases, and oscillation in the Penttinen-Koivo case. Increasing p_P in fact improves the responses in all three cases. Letting $p_P = 3$ and altering ϵ_P to the new 10% value of the upper limit of stability provides the plot in Figure 2.17. For the Pentinnen-Koivo velocity loop case, the parameters are now $(p_V = 1, \epsilon_V = 2.44, p_P = 3, \epsilon_P = 0.225)$. In the Maciejowski velocity loop case, the values are $(p_V = 1, \epsilon_V = 0.031, p_P = 3, \epsilon_P = 0.016)$, and in the combined case, $(p_V = 1, \epsilon_V = 0.153, p_P = 3, \epsilon_P = 0.0175)$.

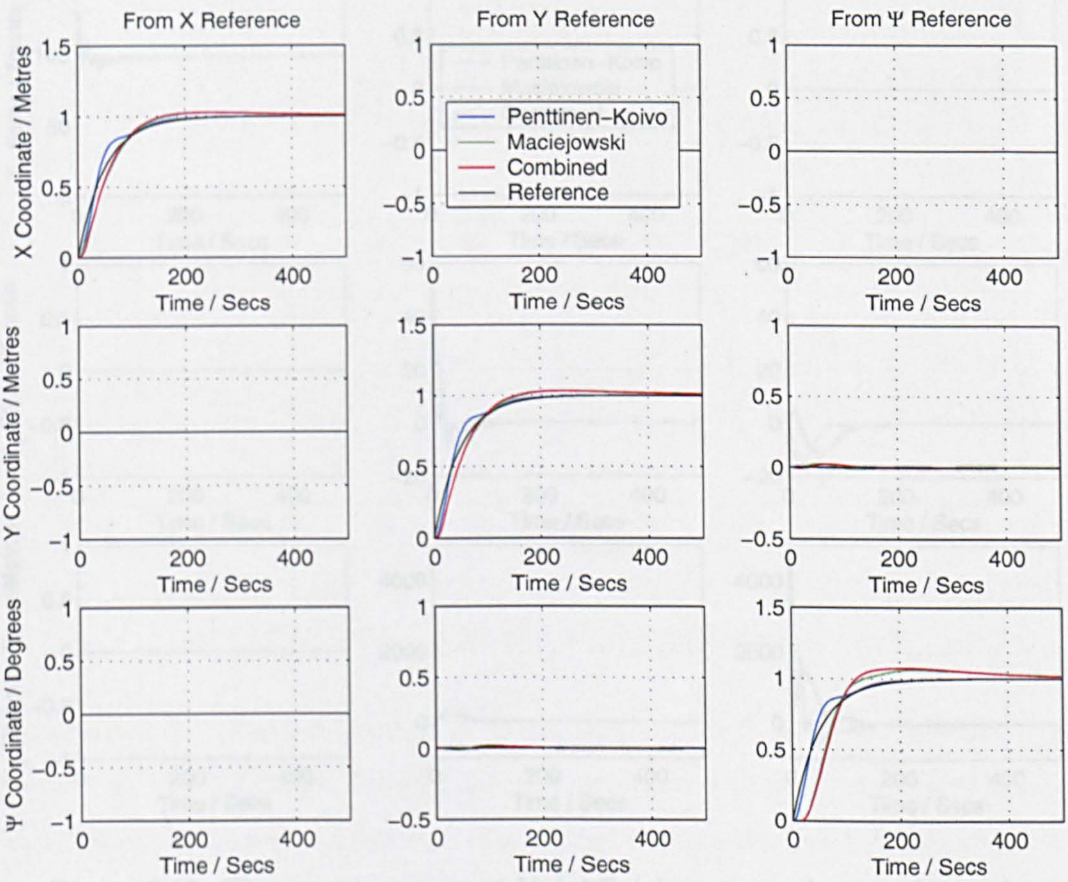

Figure 2.17: Responses of $T_P(s)F_R(s)$ to filtered step reference demand

Overshoot and settling times are now reduced and the performance is acceptable for this application.

Figure 2.18 shows the corresponding surge and sway thrust responses of $U_P(s)F_R(s)$ with the parameters given on the previous page, and a bias of $98T$ added to the surge plot to account for the operating point. In the nonlinear simulation, there would also be a slight offset on the right hand plots due to the current acting at 1^o to the ship heading when the reference point is reached. All peaks are well within the thruster maximum force of $470T$ and moment of 10^5Tm, and the rate limit of $50T/s$ is not violated.

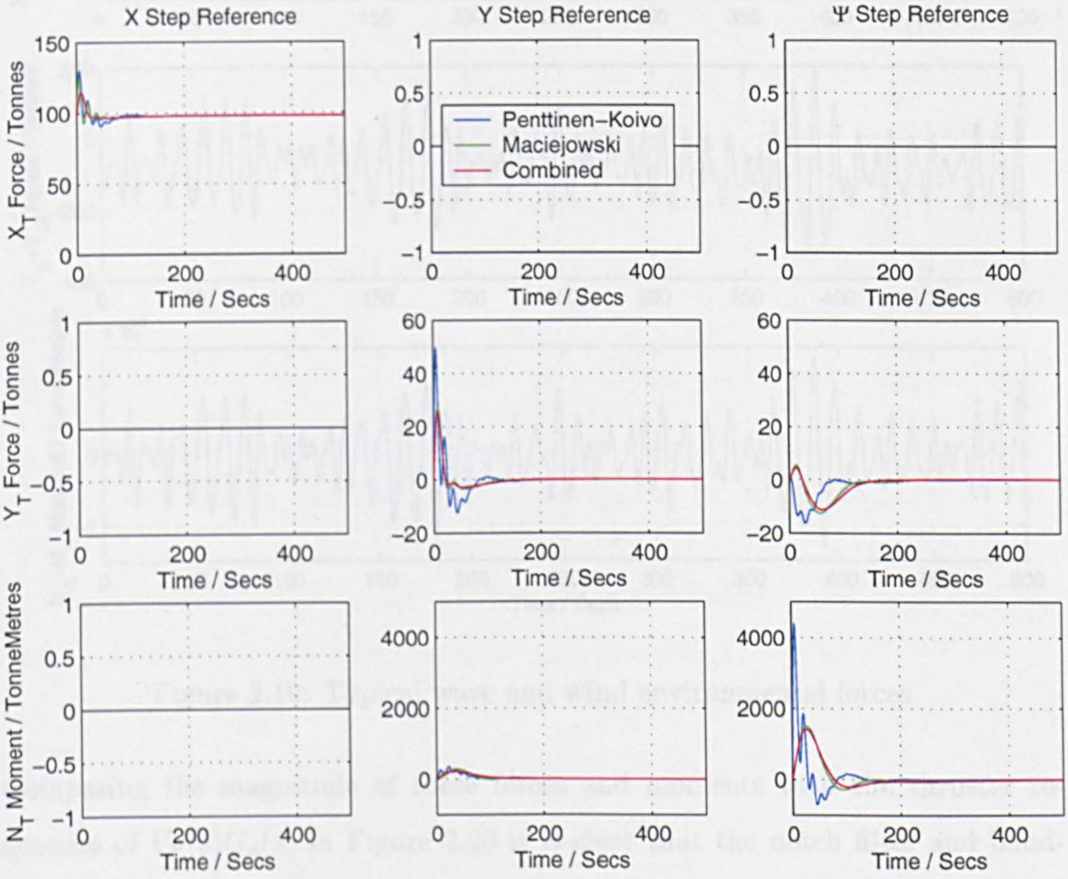

Figure 2.18: Thruster responses of $U_P(s)F_R(s)$ to step reference demand

In the simulation later, the average wind velocity, U_{A0}, is set to 20m/s at an angle $\beta_A = \Psi_R + \pi/4$, and the waves are for heavy seas at an angle $\beta_W = \Psi_R + \pi/12$, where Ψ_R is the heading reference. Figure 2.19 depicts typical wind and wave environmental forces in each direction during the simulation.

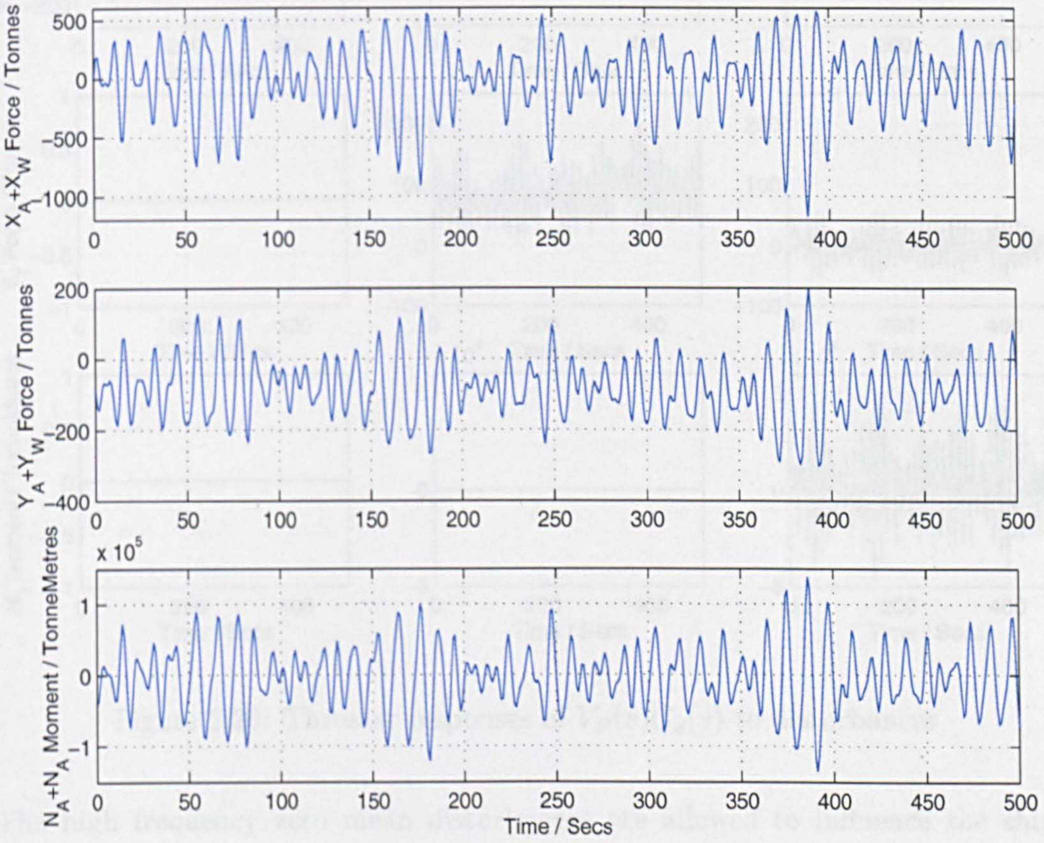

Figure 2.19: Typical wave and wind environmental forces

Comparing the magnitude of these forces and moments with the thruster responses of $V_P(s)G_d(s)$ in Figure 2.20 it is clear that the notch filter and bandwidth selection prevents total rejection of the disturbance, instead focussing on the low frequency components.

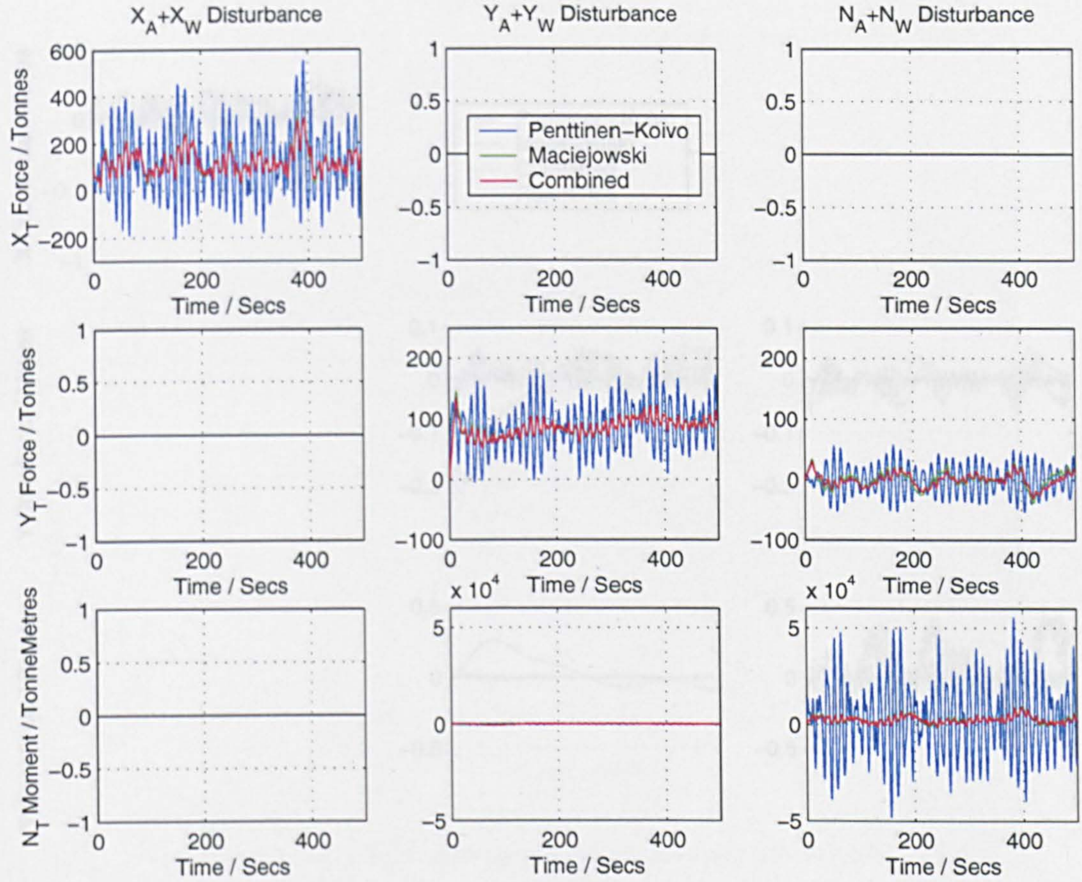

Figure 2.20: Thruster responses of $V_P(s)G_d(s)$ to disturbances

The high frequency zero mean disturbances are allowed to influence the ship motion, as depicted in the responses of $S_P(s)G_d(s)$ in Figure 2.21.

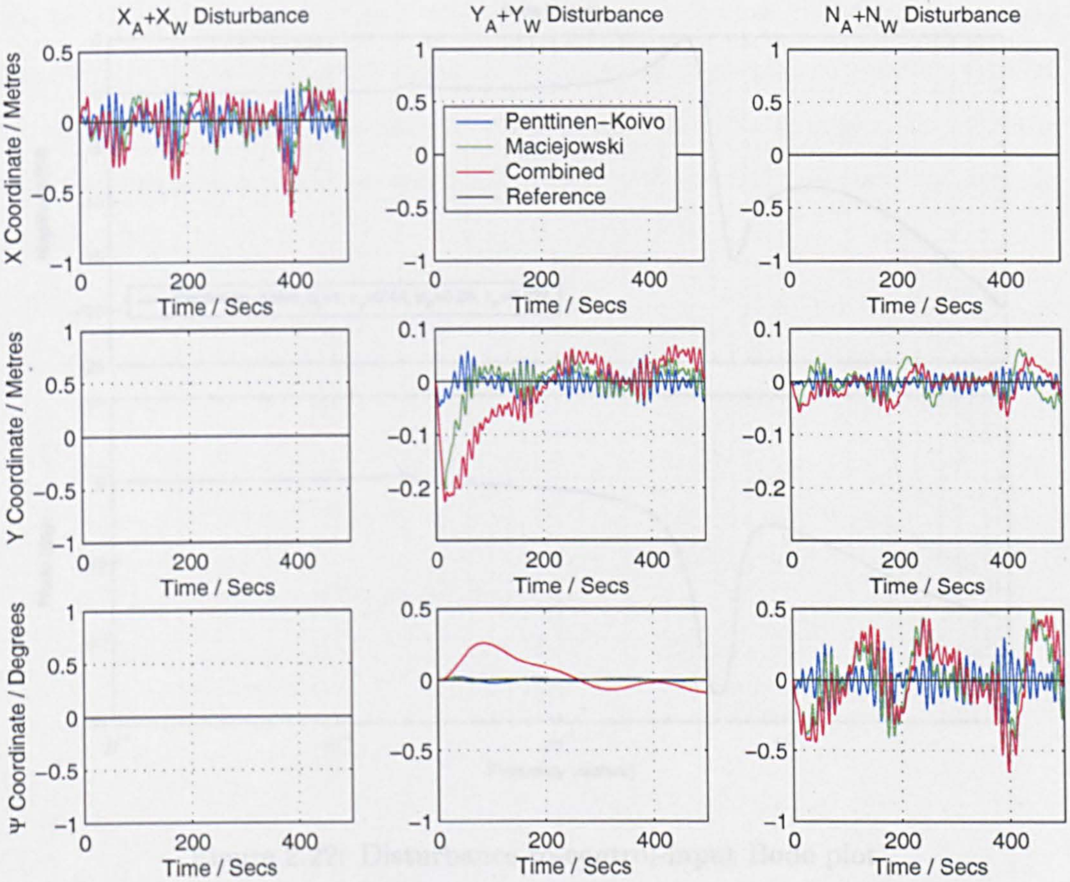

Figure 2.21: Position responses of $S_P(s)G_d(s)$ to disturbances

The surge thrust reaches peaks of greater than $470T$ in the Penttinen-Koivo linear responses and the rate of change peaks at $200T/s$, so clearly some tuning is required. Unfortunately, simply reducing p_P does not reduce the thruster magnitude as desired. This can be seen by setting $p_P = 0.25, \epsilon_P = 0.024$ and plotting the frequency response from wave disturbances to the thruster input in Figure 2.22. Comparing this with Figure 2.15 where $p_P = 1, \epsilon_P = 0.091$, the only difference is a very small resonant peak at $0.02rad/s$. The characteristics above $0.1rad/s$ are the same, hence the response to disturbances above this point is the same. In addition, p_P has already been increased from 1 to 3 in order to improve the step reference response, so it would be counterproductive to decrease p_P again.

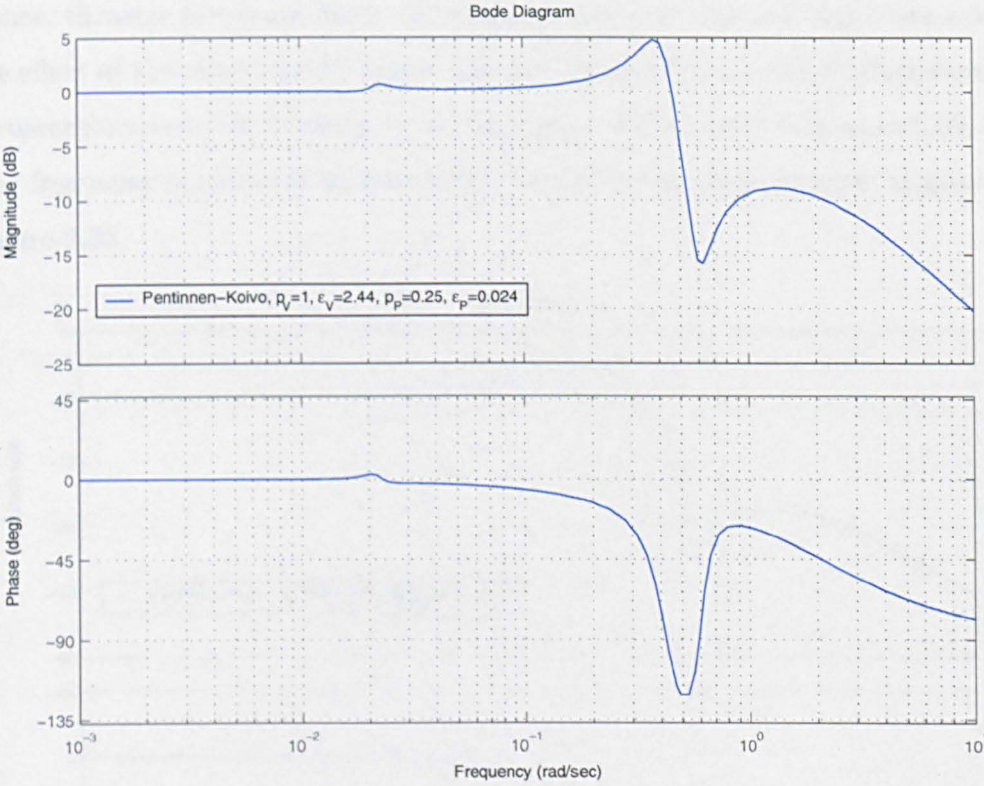

Figure 2.22: Disturbance-to-control-input Bode plot

As mentioned in Section 2.4.1, adjusting $K_V(s)$ affects the spectral characteristics of the position loop above the bandwidth frequency. To proceed, p_V must be reduced in the velocity loop and a new gain matrix, K, found in the position loop. The thruster peaks are in excess of the $470T$ maximum by around 15%, but the rate of change is four times larger than the acceptable $50T/s$. Decreasing p_V from 1 to 0.3, ϵ_V from 2.44 to 0.155 and recalculating K at $0.02rad/s$ for the Penttinen-Koivo velocity loop produces:

$$K = \begin{bmatrix} 0.0202 & 0 & 0 \\ 0 & 0.0201 & -0.094 \\ 0 & 0.00000479 & 0.0197 \end{bmatrix} \tag{2.96}$$

By comparison with equation (2.88), the off-diagonal elements show an increase due to greater velocity loop interaction between the sway and yaw directions.

Hence, thruster forces are likely to increase slightly so that one input can cancel the effect of the other control input. The reduction of p_V should produce smaller thruster forces overall, however. Maintaining $p_P = 3$ and reducing ϵ_P to 0.03, the new frequency response from wave disturbances to the thruster input is given in Figure 2.23.

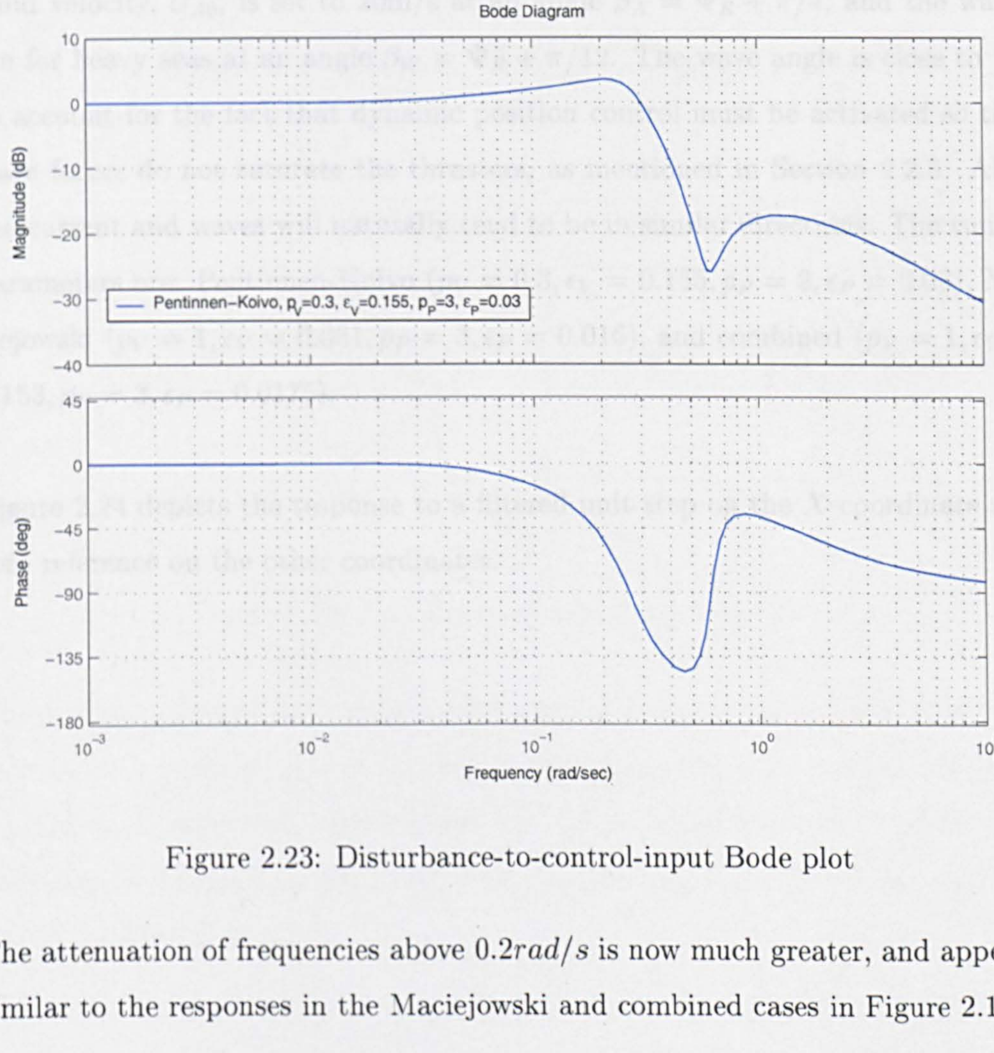

Figure 2.23: Disturbance-to-control-input Bode plot

The attenuation of frequencies above $0.2rad/s$ is now much greater, and appears similar to the responses in the Maciejowski and combined cases in Figure 2.15.

2.5.2 Simulation results

Having tuned the controller based on linearised models, the Simulink simulation results are now presented. The simulation is initialised so that the ship remains near to the operating point. From Table 2.1, when the body-fixed ship velocity is $2.0m/s$ in the surge direction and zero in the other directions, the surge thruster is at $98.0T$. Therefore, if the ship is regulated near to a slowly-changing or constant setpoint in the simulation, $\underline{R} = [X_R, Y_R, \Psi_R]^T$, and the current in the surge direction, u_c, is set to $2.0m/s$, $\beta_C = \Psi_R$, the value of X_T and u_0 should be $98.0T$ and $2.0m/s$ on average in order to counteract the fixed current. Average wind velocity, U_{A0}, is set to 20m/s at an angle $\beta_A = \Psi_R + \pi/4$, and the waves are for heavy seas at an angle $\beta_W = \Psi_R + \pi/12$. The wave angle is close to Ψ_R to account for the fact that dynamic position control must be activated so that wave forces do not saturate the thrusters, as mentioned in Section 2.2.3. Also, the current and waves will naturally tend to be in similar directions. The tuning parameters are: Pentinnen-Koivo ($p_V = 0.3, \epsilon_V = 0.155, p_P = 3, \epsilon_P = 0.03$), Maciejowski ($p_V = 1, \epsilon_V = 0.031, p_P = 3, \epsilon_P = 0.016$), and combined ($p_V = 1, \epsilon_V = 0.153, p_P = 3, \epsilon_P = 0.0175$).

Figure 2.24 depicts the response to a filtered unit step on the X coordinate and zero reference on the other coordinates.

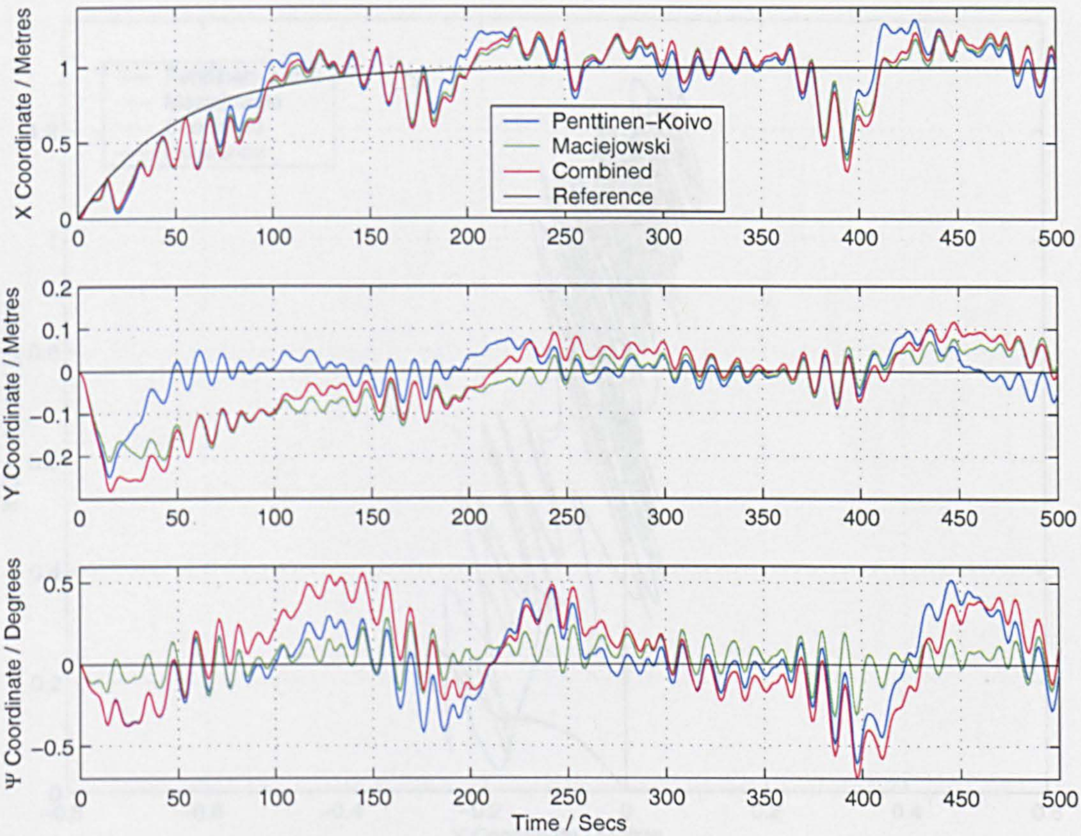

Figure 2.24: Position response of simulated ship

The corresponding plot of surge versus sway position is shown in Figure 2.25. Mean radial position error is $0.147m$ for Penttinen-Koivo in the velocity loop, $0.151m$ for Maciejowski, and $0.171m$ for the combined method. The heading error standard deviations are 0.222^o (Penttinen-Koivo), 0.104^o (Maciejowski), and 0.264^o (combined).

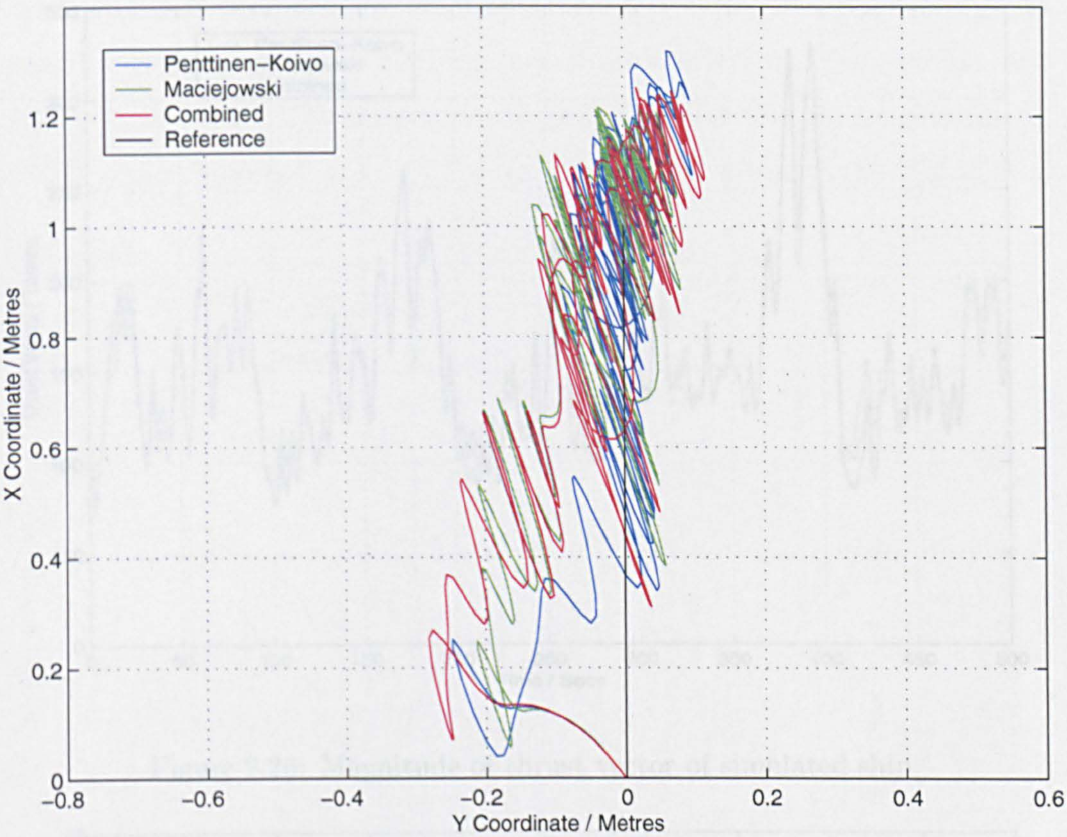

Figure 2.25: Overhead view of simulated ship position

Figures 2.26 and 2.27 depict the (X_T, Y_T) thrust vector magnitude and its rate of change during the simulation. The vector has a mean of $153T$ (Pentinnen-Koivo), $153T$ (Maciejowski) and $152T$ (combined) and reaches peaks of $328T$ (Pentinnen-Koivo), $332T$ (Maciejowski), and $327T$ (combined), well within the limit of $470T$. For the rate of change, the standard deviation is $11.1T/s$ (Pentinnen-Koivo), $12.8T/s$ (Maciejowski) and $11.0T/s$ (combined) and the maximum magnitude is $37.3T/s$ (Penttinen-Koivo), $45.8T/s$ (Maciejowski), $35.9T/s$ (combined), well within the limit of $50T/s$.

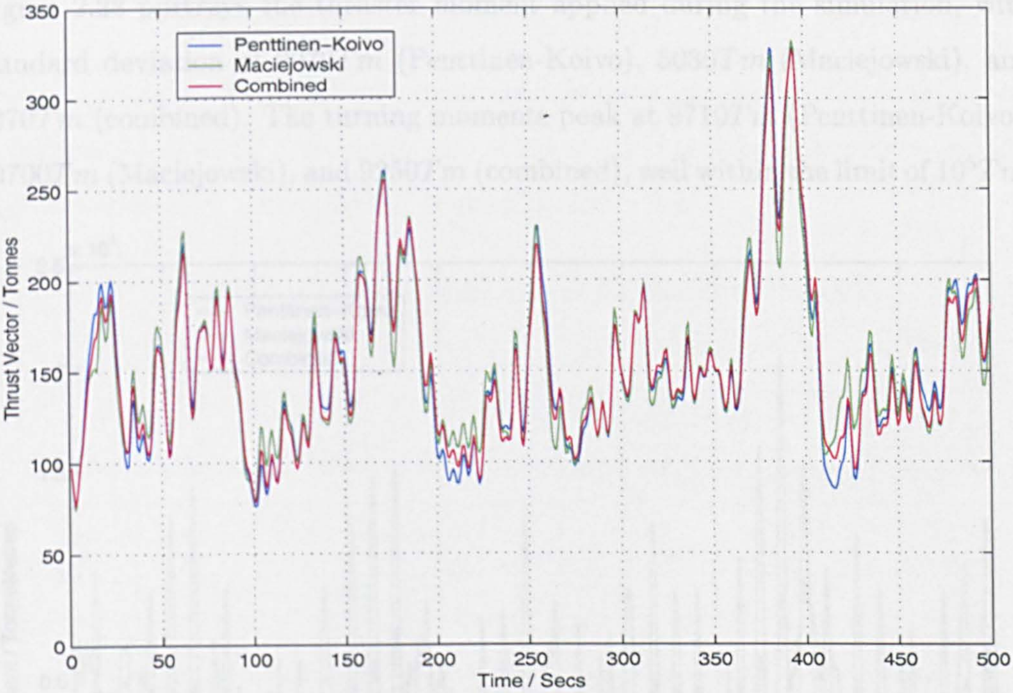

Figure 2.26: Magnitude of thrust vector of simulated ship

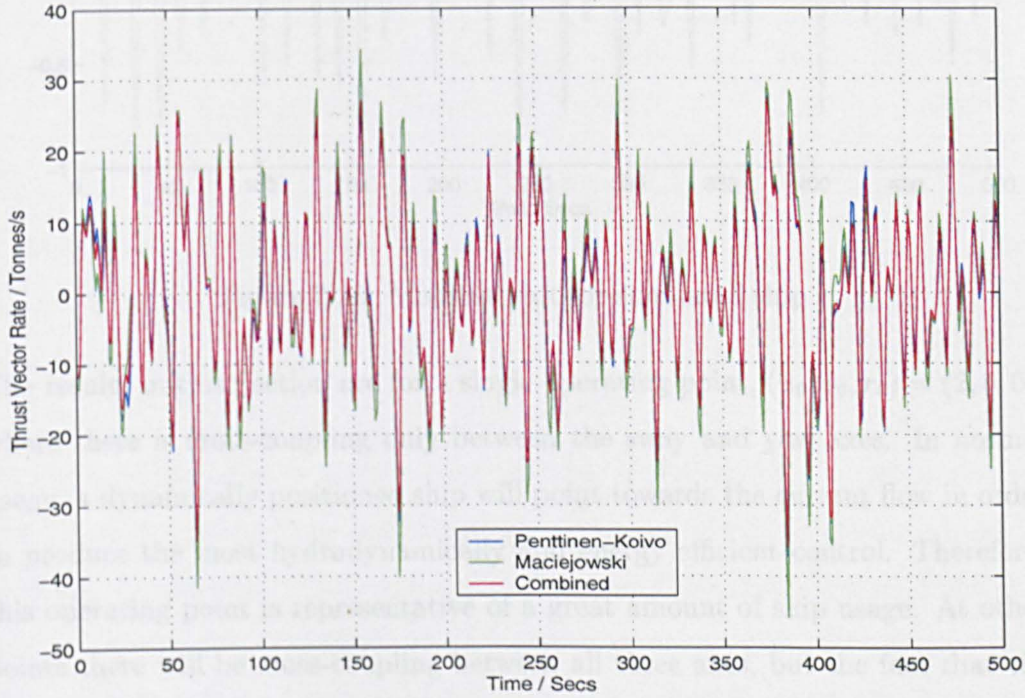

Figure 2.27: Rate of change of thrust vector magnitude of simulated ship

Figure 2.28 portrays the thruster moment applied during the simulation, with standard deviation of $2070Tm$ (Penttinen-Koivo), $5030Tm$ (Maciejowski), and $1870Tm$ (combined). The turning moments peak at $9710Tm$ (Penttinen-Koivo), $20700Tm$ (Maciejowski), and $9250Tm$ (combined), well within the limit of $10^5 Tm$.

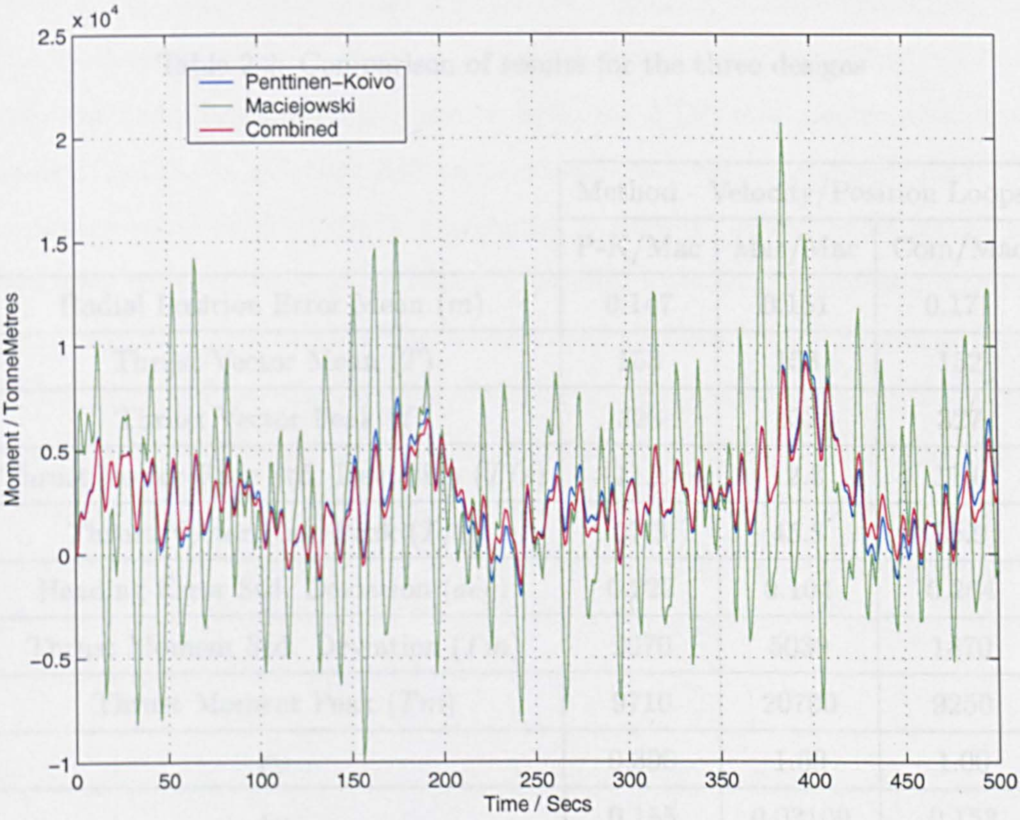

Figure 2.28: Moment plot for simulated ship

The results in this Section are for a single operating point, $(u_0, v_0, r_0) = (2, 0, 0)$, where there is cross-coupling only between the sway and yaw axes. In normal usage, a dynamically positioned ship will point towards the current flow in order to produce the most hydrodynamically and energy efficient control. Therefore, this operating point is representative of a great amount of ship usage. At other points there will be cross-coupling between all three axes, but the fact that the model-free multivariable tuning methods are successful for interaction between sway and yaw suggests that little difficulty would be encountered in other cases.

2.6 Discussion

Having produced three different multivariable dynamic ship positioning designs satisfying the thruster constraints and achieving acceptable performance and disturbance rejection, a comparison is given in Table 2.2.

Table 2.2: Comparison of results for the three designs

	Method - Velocity/Position Loops		
	P-K/Mac	Mac/Mac	Com/Mac
Radial Position Error Mean (m)	0.147	0.151	0.171
Thrust Vector Mean (T)	153	153	152
Thrust Vector Peak (T)	328	332	327
Thrust Vector Rate Std. Deviation (T/s)	11.1	12.8	11.0
Thrust Vector Rate Peak (T/s)	37.3	45.8	35.9
Heading Error Std. Deviation (deg)	0.222	0.104	0.264
Thrust Moment Std. Deviation (Tm)	2070	5030	1870
Thrust Moment Peak (Tm)	9710	20700	9250
p_V	0.300	1.00	1.00
ϵ_V	0.155	0.03100	0.153
p_P	3.00	3.00	3.00
ϵ_P	0.0300	0.0160	0.0175

Outright performance is not the only criterion of success for a "quick and easy" multivariable tuning technique, although performance figures are an important medium of assessment. The mean and peak of the thrust vector is almost identical in each case, but the Mac/Mac radial position error mean is 2.72% greater than for P-K/Mac and the Com/Mac error is 16.3% greater. However, the thrust vector rate peak is least in the Com/Mac case, although only by 3.8% from P-K/Mac.

The thrust vector rate standard deviations are almost identical for P-K/Mac and Com/Mac. From this evidence it could be argued that the P-K/Mac technique is superior, but first examine the heading error and thruster moment results.

The Mac/Mac case exhibits the smallest heading error standard deviation, but also the largest thrust moment standard deviation and peak. The P-K/Mac controller has a heading error standard deviation 2.13 times greater, but thrust moment standard deviation 2.43 times smaller and thrust moment peak 2.13 times smaller. Therefore, if it is acceptable to allow a larger heading error by a particular multiple, the control energy expended is smaller by a greater proportion. The Com/Mac controller has a heading error standard deviation 2.54 times greater than the Mac/Mac case, but thrust moment standard deviation 2.69 times smaller and thrust moment peak 2.24 times smaller. Again, the control energy is decreased by a greater amount than the error is increased, although the peak thrust shows the opposite trend. The Mac/Mac method is perhaps suffering from the cross-coupling indicated in Figure 2.13, producing a less efficient controller for sway and yaw than is possible.

The heading error standard deviation of 0.222^o in the P-K/Mac case is entirely acceptable, added to the low expenditure of control energy. Thus, again it appears that the P-K/Mac method is preferable and, in fact, delivers the best overall performance of the three controllers under inspection. This is perhaps slightly surprising given the fact that the Penttinen-Koivo method is intended for decoupling at high frequencies and the ship operates at low frequencies. It was necessary to reduce the scalar tuning gain for the P-K tuned velocity loop in order to achieve this performance, so it could be argued that more tuning was involved for the P-K/Mac method during the comparison of so-called "rapid tuning" techniques. Without any adjustment of p_V, Penttinen-Koivo is clearly the worst method, as the thrust magnitude and rate limits are violated.

It is not certain that the performance of the Mac/Mac and Com/Mac controllers cannot also be improved with further tuning. The integral action scaling gains were chosen fairly arbitrarily to be 10% of the stable value and the decoupling bandwidth of $0.2rad/s$ could be increased until the P-K gain matrix resulted at the upper limit. Decreasing the bandwidth is also a possibility, although stability will become a problem as the gain matrix approaches the Davison gain at the lower limit.

In summary, all three methods produce acceptable multivariable performance with little effort, although the P-K/Mac controller requires slightly more tuning from the initial unity proportional scalar gain in the velocity loop. Achieving ultimate performance was not the aim of this Chapter, but an interesting piece of further work would be to vary the decoupling bandwidth and scalar tuning gains for each method. It is possible that one method would yield clearly superior performance, although the results of this Chapter suggest that any advantage would be small and would involve an interplay between tracking error, control energy, bandwidth selection and decoupling.

2.7 Conclusions

This Chapter began by introducing the ship Dynamic Positioning (DP) control problem, where the objective is to maintain the position and heading of a vessel in the sea by using active thrusters only. This problem has been the subject of research for over thirty years and the design tools have ranged from heuristic PID tuning to modern LQG and H_∞. The problem is multivariable in nature due to interactions between the surge, sway and yaw directions of movement, and the mathematical model may not be known. However, the problem of rapid preliminary tuning has never previously been addressed whilst taking these factors into account. The main contribution of this Chapter was to investigate four potential techniques for addressing this particular problem.

A nonlinear multivariable model for a Floating Production, Storage and Offloading (FPSO) vessel was presented, which encompasses the hydrodynamic forces in addition to disturbances due to wind, waves and current. This model and the parameters used throughout the thesis were the result of a case study carried out for Mitsubishi Heavy Industries Ltd.

Four methods for rapid tuning of a multivariable PID controller were given. These methods are model-free and intended for easily producing an initial controller design based on step tests or application of sinusoidal inputs at a single frequency. Although a model was used for simulation, it is straightforward to apply these tests in reality where no model is known. Controller structures were discussed and the four methods were applied to the linearised ship model at an operating point. The Davison method proved to be of little use, as the ship model in question possesses frequency domain characteristics such that the technique cannot be applied. Penttinen and Koivo, Maciejowski and the combined approaches, however, produced control schemes with a few short, simple steps.

The Penttinen-Koivo technique involves applying a unit step to each input in turn and measuring the initial gradient of the output response and the steady state gain. The values of gradient are then incorporated in a square matrix to form a proportional control term, and the steady state values are arranged in another square matrix to produce the integrator term, both with multiplicative scalar tuning parameters. The Maciejowski technique involves applying a sinusoidal input to each input in turn and measuring the magnitude and phase response at the outputs. The frequency of the sinusoid is chosen to be the desired bandwidth of the closed-loop system. The magnitude of the sinusoid is small to provide an almost linear response at the operating point. The response is expressed as a matrix of complex numbers, for which a real approximate inverse is found. The resulting matrix is used as proportional and integral controller terms, also with scalar tuning parameters. For the combined method, the real approximate inverse matrix is used for the proportional controller term and the steady state gain inverse is used for the integral term. The Pentinnen-Koivo technique provides decoupling at very low and very high frequencies, the Maciejowski method provides approximate decoupling at the chosen bandwidth, whereas the combined method decouples at the selected bandwidth and very low frequencies.

The control scheme consists of velocity and position feedback loops, tuned in that order. The generalised Nyquist stability criterion was used to find a range of suitable values for the tuning parameters. This would be not be an option where a model is not available, but served to expedite the investigation rather than having to "tune the regulator on-line". The controllers were easily tuned to meet design criteria of stability, speed of response, disturbance rejection, decoupling and avoidance of actuator saturation. Simulation results were presented and it was concluded that using the Penttinen-Koivo method in the velocity loop and Maciejowski method in the position loop provides the best performance, by a small margin only, although with slightly more tuning.

The main flaw of the design methods detailed in this Chapter is the dependence on a point where decoupling can take place in the frequency range of interest. The Davison method is of no use where integrators are present in the plant, and the Pentinnen-Koivo method requires that high frequency motions are desirable in a system. Maciejowski's approach can produce decoupling through a large range of frequencies, but selection of the wrong bandwidth can result in strong interaction between one or more of the system modes, and therefore an ineffective control system.

The main advantage of the design approaches given is the speed and ease with which an effective multivariable PID controller can be produced. The advantage to industry is that money can be saved on design effort, and also on commissioning time. A benefit for academics is that a MIMO PID controller can easily be obtained when making comparisons with more advanced techniques. Of course, further fine-tuning of individual loops can take place after using these methods, but the initial design can serve as a useful starting point.

One extension of this work could be to try alternative bandwidths for the velocity loop and to experiment more with the scalar tuning gains. It is conceivable that the control system performance with the ship example could be improved with experimentation. Another extension would be to try a large number of operating points to confirm the wide applicability of the techniques, then to apply gain scheduling as the overall control scheme.

Chapter 3

Neural Network Ship Identification

This Chapter describes an examination of techniques for identifying the dynamical behaviour of a ship from input and output data. The situation when two ships influence each other's motion via a hawser is studied in particular, and a neural network is applied to the task. The concept of a neural network is described and the elements of the mathematical model additional to the previous Chapter are detailed. An experiment is carried out using data from a simulation, which produces results to encourage further investigation using real data. This data is obtained from a 1/50th scale model in a tank with controllable environment. The subsequent results of neural network training for different wave heights and excitation are satisfactory, although deterioration is seen as disturbances increase. Network performance is analysed at the end of the chapter, and use of neural nets for control applications is discussed.

3.1 Introduction

Modelling of vessels for DP control generally involves deduction of the equations of motion from first principles, see Fossen (1994) for example, and quantification of parameters from physical knowledge. This will produce a model as given in the previous Chapter. Another approach is to perform system identification, whereby a mathematical model is created using data from observations of system behaviour. Ljung (1999) deals with a broad range of such techniques in some depth, beginning with simple linear single-input single-output (SISO) time-invariant systems with clear structure based on physical insight. A range of system complexity is then covered, up to nonlinear multivariable time-varying systems with "black-box" internals.

The ship identification problem is towards the complex end of that range, as it is certainly nonlinear and multivariable. Time-variation may also be a factor as the ship encounters changes in its environment which affect the dynamics. The ship is more "grey box" than "black box" because physical laws can be used to analyse likely behaviour influenced by known parameters. However, the model given in equations (2.1) to (2.37), for example, is subject to various modelling approximations and uncertainty over quantities such as the hydrodynamic derivatives.

Fossen (1994) and Fossen et al. (1996) deal with parameter estimation of a DP ship model using a state-augmented extended Kalman filter, see Gelb et al. (1988). Lack of persistent excitation is noted as a significant impediment to identification in the DP problem, hence the procedure is performed off-line and is broken into a number of manoeuvres. Firstly, the ship is excited in the surge direction alone using the main propellors to identify three important parameters in the model. These parameters are frozen and used in the next stage where sway and yaw excitation are applied together via bow and aft tunnel thrusters to identify another six quantities. The final stage involves freezing the nine known parameters so far and exciting the bow azimuth thruster only, to determine one more parameter. The Kalman filter is presented with several data sets from different manoeuvre sequences in order to improve identifiability and reduce parameter drift. The estimated model behaviour is compared with experimental results from the real ship and produces good agreement.

The identification of DP ship dynamics is clearly not a simple problem, further complicated by lack of persistent excitation and coupling between the three degrees of freedom. Ljung (1999) suggests that it is usually preferable to work with state-space models in the multivariable case, but still gives the generalisation of various SISO polynomial methods to multi-input multi-output (MIMO) models. Subspace identification for state-space is covered, this is in more detail in Van Overschee and DeMoor (1996), but the method is supposed to be used with linearly behaving systems. The ship is also not suited to the nonlinear Wiener (output nonlinearity) or Hammerstein (input nonlinearity) models, so attention must unavoidably turn to a more general system representation. Ljung (1999) refers to basis function expansions or networks for this purpose, of which neural networks are a good example.

Neural networks are well suited to identification of arbitrary highly nonlinear systems, when used as a mathematical mechanism for "learning" a quite general mapping from an r-dimensional real space to an s-dimensional real space. Their other function is pattern recognition, not of interest here, which involves a different structure and mode of learning. A neural network consists of a large number of individual processing units called neurons, which may be arranged in parallel within layers and interconnected in such a way that every neuron in one layer can influence the input to every neuron in the next layer. This is known as a feedforward network, and if the outputs from neurons may also feed back to influence neurons in the same or previous layers, this is called a recurrent network. These basic issues are well covered in Pham and Liu (1995) and Narendra and Parthasarathy (1990).

In order for a neural network to approximate a desired mapping, it must go through a supervised learning process, see Haykin (1994). An input is presented to the network and the output from the final layer is compared with a desired target output. The error between the two is used to update the network connections to give more accurate results. The process repeats until satisfactory performance is achieved.

In Page et al. (1993), a recurrent modified Elman network is able to identify SISO systems of up to third order. In Pham and Liu (1995), feedforward networks are used to identify SISO systems of arbitrary order using tapped-delay-lines i.e. input and output signals from the past and present. The drawback of feedforward nets for representing dynamical systems is that they do not have dynamic memory, so tapped-delay-lines become necessary although the system order and number of delays required may not be known. However, in the ship model of Chapter 2, a linear state-space model is derived where accelerations are a non-dynamic nonlinear multivariable function of force and velocity. Hence, dynamic memory is not necessary to characterise the system.

In this Chapter, with the above consideration in mind, the most commonly used feedforward network, the multilayer perceptron (MLP), is applied to a ship identification problem. The Chapter details an investigation into the use of a neural network for modelling the motions of a Shuttle Tanker when coupled to a Floating Production, Storage, and Offloading (FPSO) vessel via a hawser, as represented by Figure 3.1. The circled numbers are (1) Hawser Angle with FPSO, (2) Relative Position of FPSO and Shuttle Tanker, and (3) Hawser Angle with Shuttle Tanker. A turret towards the front of the vessel anchors the FPSO, and control comes from a stern azimuth thruster. The shuttle tanker is pulled away from the FPSO by a tugboat, hence keeping the hawser in tension, and control comes from a bow azimuth thruster. The shuttle tanker also has a rear propeller and rudder, but these are not used when the hawser couples the two ships. This configuration is used by Mitsubishi Heavy Industries (MHI) Ltd. when an FPSO is offloading oil. The parameters of the simulation model in Section 3.4 and subsequent real tank test data in Section 3.5 are supplied by MHI.

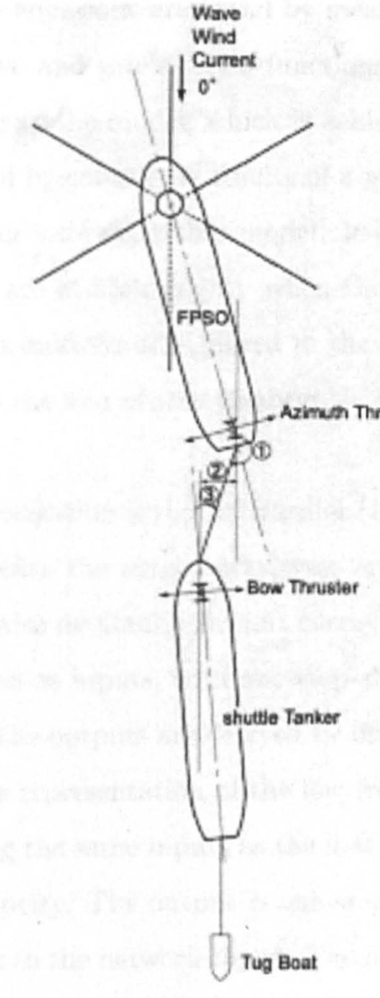

Figure 3.1: Plan view of coupled ships

Neural networks have previously been used in marine systems identification and control, but a survey of the literature indicates that control has been the predominant area of research. This will be discussed in Section 3.6. The potential of neural networks for identification has been explored in aspects of the overall ship system, such as diesel engine modelling in Roskilly and Mesbahi (1996b) and Xiros and Kyrtatos (2001), but there is only a small body of research into the identification of ship motion. Haddara and Xu (1999) look at ship coupled heave-pitch motions, modelled by two so-called "random decrement" equations. Two of the parameters in these equations are found by measuring the period of damped free oscillations in heave and pitch. Two functions must then be identified to represent the remainder of the model, which is achieved using a neural network. The method is validated by comparing results of a numerical simulation with experimental data from an icebreaker ship model. It is found that the predictions by the neural network are of high fidelity when the ship is lightly damped. Of course, heave and pitch motions are ignored in the DP problem, so this work is not entirely relevant to the aim of this Chapter.

A directly relevant investigation is that of Hardier (1995), who uses an MLP with tapped-delay lines to learn the surge, sway, yaw and roll dynamics of a 1/12th scale model of the Charles de Gaulle aircraft carrier. A network is trained using fin and rudder deflection as inputs, with one-step-ahead roll angle, roll, yaw and sway rate as outputs. The outputs are delayed by one time step and fed back into the network to improve representation of the low frequency dynamics. A second network is trained using the same inputs as the first, plus propellor rotation rate, yaw rate and surge velocity. The output is one-step-ahead surge velocity, which is delayed and fed back to the network input. The first network is applicable at a fixed surge velocity only, hence three versions are trained at 15, 20, and 25 knots. The second network is used to create a prediction of surge velocity, which inputs to an interpolation law acting on the outputs of the three velocity-dependent networks. Training occurs for $2.5m$ wave height and the results are described as "satisfactory".

The contribution of this Chapter is to identify ship dynamics for use in DP of coupled ships, using a neural network. This problem has been investigated analytically in Morishita et al. (2001) using hydrodynamical equations. The authors concluded that the dynamics are complicated and there are numerous modes, stable and unstable operating points present. The aim of the following is to dispense with analysis and simply look at the viability of learning an input-output model. This is not an entirely original idea, as the work of Hardier (1995) described above is clearly similar. However, the fact that no recurrence is involved in the network and that the dynamics are for a coupled-ship system provides a problem of a slightly different nature.

The Chapter is divided into several Sections. Section 3.2 explains the mathematics behind neural networks and the manner in which they are used. Section 3.3 describes the mathematical model used to both analyse ship motion and to produce a computer simulation, and explains how this model suggests the inputs and outputs to be used with the neural net. Section 3.4 details an initial attempt at training neural networks based on data from the computer simulation. Results and Figures are given to illustrate the effectiveness of producing acceleration, velocity, and position signals for three degrees of freedom of the ship.

Section 3.5 then gives results using data supplied by Mitsubishi from their 1/50th scale model tank tests. These results are the most important as they give an idea of the realistic potential of neural networks for ship identification, which appears to be fairly high. Section 3.6 discusses the results given in the main body of the Chapter, describes the use of neural networks for control, and suggests ideas for further work. Conclusions are drawn in Section 3.7.

3.2 Neural networks

The individual processing units of a neural network are called neurons, and the output of each neuron, see Figure 3.2, is a function of the weighted sum of the neuron inputs, as detailed in the following equation:

$$\begin{aligned} a &= f(\sum_{i=1}^{n} w(i)p(i) + b) \\ &= f(\underline{w}^T \underline{p} + b) \end{aligned} \tag{3.1}$$

where a is the output, $p(i)$ is the i^{th} input, $w(i)$ is the weight on the i^{th} input, b is a bias term, and f is known as the activation function.

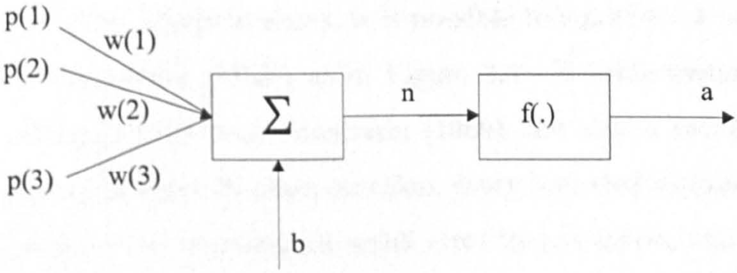

Figure 3.2: Neuron

A layer of neurons is arranged as in Figure 3.3 so that it obeys:

$$\underline{a} = f(W\underline{p} + \underline{b}) \tag{3.2}$$

where $\underline{a}$ is the vector of outputs from all neurons in a layer, W is the weight matrix, $\underline{p}$ is the vector of inputs to all neurons in a layer, $\underline{b}$ is the bias vector, and f is the activation function which is the same for every neuron in a given layer. The activation function can take the form of a step, a straight line, a log-sigmoid, hyperbolic tangent or radial basis function depending on the application.

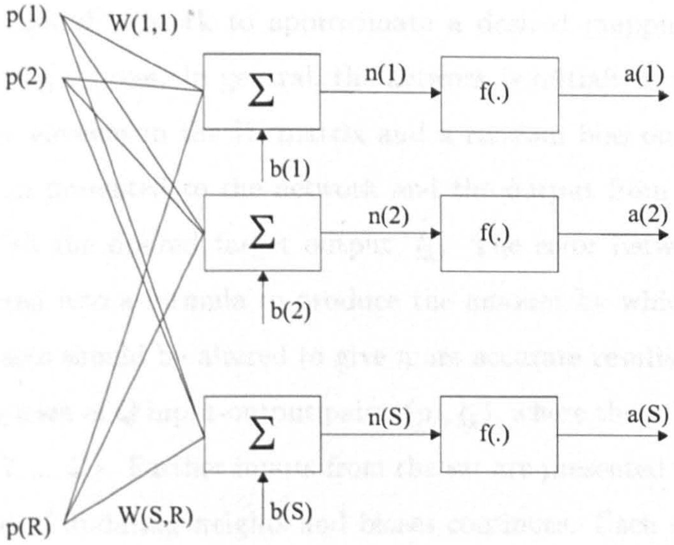

Figure 3.3: Layer of neurons

Using several of these layers in series, it is possible to construct a network called a Multi-Layer Perceptron (MLP) as in Figure 3.4. Notable features of MLPs, proved by Cybenko (1989) and Funahashi (1989), are that a two-layer network can exactly represent every Boolean function, every bounded continuous function can be approximated with arbitrarily small error by two layers, and any function can be approximated to arbitrary accuracy by three layers.

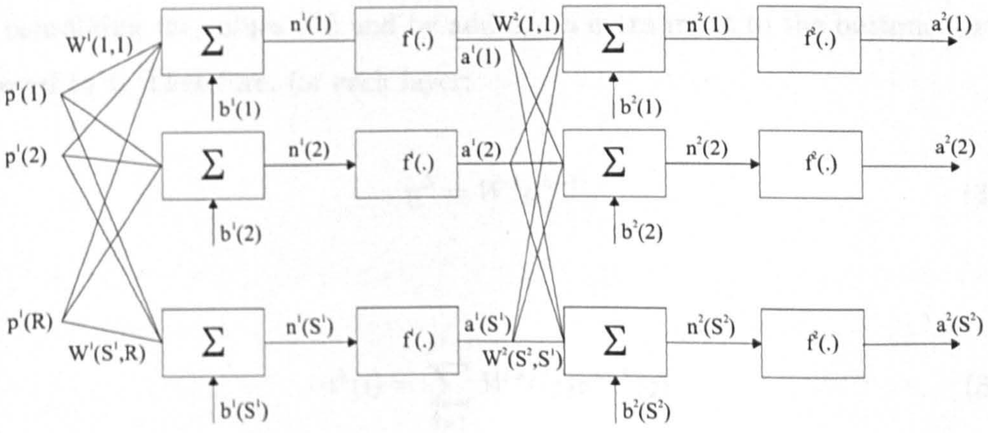

Figure 3.4: Multi-layer perceptron

In order for a neural network to approximate a desired mapping, it must go through a learning process. In general, the network is initialised with a random weight on every element in the W matrix and a random bias on every neuron. An input is then presented to the network and the output from the final layer is compared with the desired target output, $\underline{t}_k$. The error between the two is then incorporated into a formula to produce the amount by which the network weights and biases should be altered to give more accurate results. The network is trained using a set of Q input-output pairs, $(\underline{p}_k, \underline{t}_k)$, where the subscript k is the index, $k = \{1, 2, ..., Q\}$. Further inputs from the set are presented to the network and the process of updating weights and biases continues. Each presentation of an input/output pair and alteration of weights and biases is known as an epoch, and the algorithm continues until the maximum number of epochs is reached, or the error between target and actual output satisfies a predefined criterion.

3.2.1 Notation

The superscript λ will be used as the index for the network layers, $\lambda = \{1, 2, ..., \Lambda\}$, and $\underline{n}^\lambda$ will denote the vector of values after the summing junction in the λ^{th} layer, as in Figure 3.4. It will be assumed that the biases have been incorporated into the weight matrices. This is achieved by adding an extra column to the end of W containing the values of $\underline{b}$ and by adding an extra input to the bottom row of $\underline{p}$ equal to 1. Therefore, for each layer:

$$\underline{n}^\lambda = W^\lambda \underline{a}^{\lambda-1} \tag{3.3}$$

or

$$n^\lambda(i) = \sum_{j=1}^{S^{\lambda-1}} W^\lambda(ij) a^{\lambda-1}(j) \tag{3.4}$$

where $W^\lambda(ij)$ is the i^{th} row and j^{th} column of the weight matrix for the λ^{th} layer. $S^{\lambda-1}$ is the number of neurons in the $(\lambda - 1)^{th}$ layer. Also note that $\underline{a}^{\lambda-1} = \underline{p}^\lambda$.

The output of each layer is:

$$\underline{a}^{\lambda} = f^{\lambda}(\underline{n}^{\lambda}) \tag{3.5}$$

or

$$a^{\lambda}(i) = f^{\lambda}(n^{\lambda}(i)) \tag{3.6}$$

where $f^{\lambda}(.)$ is the transfer function used in the λ^{th} layer.

3.2.2 The backpropagation learning rule

The process of learning for MLP neural networks adjusts the weights and biases one layer at a time working backwards from the output layer, hence the name "backpropagation". The cost function for backpropagation is:

$$J = \frac{1}{2}\sum_{k=1}^{Q}[\underline{t}_k - \underline{a}_k^{\Lambda}]^T[\underline{t}_k - \underline{a}_k^{\Lambda}] \tag{3.7}$$

where $\underline{t}_k$ is the k^{th} target output and $\underline{a}_k^{\Lambda}$ is the output from the final network layer in response to the k^{th} training input vector.

The method of steepest descent is used to calculate the change in the $(i,j)^{th}$ element of W after each presentation of the training data to the network. The expression for this change is:

$$\Delta W^{\lambda}(ij) = -\eta\frac{\partial J}{\partial W^{\lambda}(ij)} \tag{3.8}$$

where η is called the learning rate. The idea behind this method is to find values of W that minimise J by moving along the line where $\partial J/\partial W$ is a maximum. The solution is iterative and the weights will be updated at each training step until a global minimum for J is reached.

An implementable expression for $\partial J/\partial W^\lambda(ij)$ is therefore required. This will be given in the following derivation. Defining J_k as the contribution to the cost from the k^{th} training error:

$$J_k = \frac{1}{2}[\underline{t}_k - \underline{a}_k^\Lambda]^T[\underline{t}_k - \underline{a}_k^\Lambda] \tag{3.9}$$

so that

$$J = \sum_{k=1}^{Q} J_k \tag{3.10}$$

allows us to work with $\partial J_k/\partial W^\lambda(ij)$ rather than $\partial J/\partial W^\lambda(ij)$. Returning to (3.4) see that:

$$n_k^\lambda(i) = \sum_{j=1}^{S^{\lambda-1}} W^\lambda(ij) a_k^{\lambda-1}(j) \tag{3.11}$$

and

$$a_k^\lambda(i) = f(n_k^\lambda(i)) \tag{3.12}$$

Using the chain rule, split $\partial J_k/\partial W^\lambda(ij)$ into a product:

$$\frac{\partial J_k}{\partial W^\lambda(ij)} = \frac{\partial J_k}{\partial n_k^\lambda(i)} \frac{\partial n_k^\lambda(i)}{\partial W^\lambda(ij)} \tag{3.13}$$

From (3.11)

$$\frac{\partial n_k^\lambda(i)}{\partial W^\lambda(ij)} = a_k^{\lambda-1}(j) \tag{3.14}$$

so an expression for the first term in the product is required, defined as:

$$\delta_k^\lambda(i) = -\frac{\partial J_k}{\partial n_k^\lambda(i)} \tag{3.15}$$

so that

$$\Delta W^{\lambda}(ij) = \eta \sum_{k=1}^{Q} \delta_k^{\lambda}(i) a_k^{\lambda-1}(j) \tag{3.16}$$

Again using the chain rule, express $-\delta_k^{\lambda}(i)$ as:

$$\frac{\partial J_k}{\partial n_k^{\lambda}(i)} = \frac{\partial J_k}{\partial a_k^{\lambda}(i)} \frac{\partial a_k^{\lambda}(i)}{\partial n_k^{\lambda}(i)} \tag{3.17}$$

From (3.12), see that:

$$\frac{\partial a_k^{\lambda}(i)}{\partial n_k^{\lambda}(i)} = \frac{\partial f^{\lambda}}{\partial n_k^{\lambda}(i)} \tag{3.18}$$

and so an expression for $\partial J_k / \partial a_k^{\lambda}(i)$ is needed. There are two cases to be covered here, one is for the output layer ($\lambda = \Lambda$) and the other is for the remaining layers ($\lambda = 1, ..., \Lambda - 1$). For the output layer, recall that:

$$\begin{aligned} J_k &= \frac{1}{2}[\underline{t}_k - \underline{a}_k^{\Lambda}]^T [\underline{t}_k - \underline{a}_k^{\Lambda}] \\ &= \frac{1}{2}\sum_{i=1}^{S^{\Lambda}} [t_k(i) - a_k^{\Lambda}(i)]^2 \end{aligned} \tag{3.19}$$

Therefore:

$$\frac{\partial J_k}{\partial a_k^{\Lambda}(i)} = -[t_k(i) - a_k^{\Lambda}(i)] \tag{3.20}$$

and the delta vector for the output layer is constructed from:

$$\delta_k^{\Lambda}(i) = [t_k(i) - a_k^{\Lambda}(i)] \frac{\partial f^{\Lambda}}{\partial n_k^{\Lambda}(i)} \tag{3.21}$$

For the remaining layers:

$$\frac{\partial J_k}{\partial a_k^{\lambda}(i)} = \sum_{l=1}^{S^{\lambda+1}} \frac{\partial J_k}{\partial n_k^{\lambda+1}(l)} \frac{\partial n_k^{\lambda+1}(l)}{\partial a_k^{\lambda}(i)} \tag{3.22}$$

From (3.11):

$$n_k^{\lambda+1}(i) = \sum_{j=1}^{S^\lambda} W^{\lambda+1}(ij)\,a_k^\lambda(j) \tag{3.23}$$

leading to

$$\frac{\partial n_k^{\lambda+1}(l)}{\partial a_k^\lambda(i)} = W^{\lambda+1}(li) \tag{3.24}$$

From (3.15):

$$\frac{\partial J_k}{\partial n_k^{\lambda+1}(l)} = -\delta_k^{\lambda+1}(l) \tag{3.25}$$

Substituting (3.24) and (3.25) into (3.22) gives:

$$\frac{\partial J_k}{\partial a_k^\lambda(i)} = -\sum_{l=1}^{S^{\lambda+1}} \delta_k^{\lambda+1}(l)\,W^{\lambda+1}(li) \tag{3.26}$$

and the delta vector for the remaining layers is constructed from:

$$\delta_k^\lambda(i) = \sum_{l=1}^{S^{\lambda+1}} \delta_k^{\lambda+1}(l)\,W^{\lambda+1}(li)\,\frac{\partial f^\lambda}{\partial n_k^\lambda(i)} \tag{3.27}$$

3.2.3 Training algorithm

The theory necessary to construct an algorithm for training an MLP has been derived above. The steps are as follows:

1. Initialise the network with random weights and biases
2. Propagate each training input $\underline{p}_k$ forward through the network, using the current weights and recording the $\underline{p}_k^\lambda$ vectors for each layer
3. From the output of the last layer, calculate the training error $\underline{t}_k - \underline{a}_k^\Lambda$
4. Use the training error to calculate the delta vector for the output layer
5. Calculate the delta vectors for all of the hidden layers, moving backwards from the output layer

6. Repeat steps 2-5 for each input-output pair in the training set. Use the accumulated delta vectors and input vectors to update the weight matrices

7. Repeat steps 2-6, known as an epoch, until either a pre-determined number of epochs have been completed or the cost J decreases to a pre-determined value

When the algorithm is finished, it is desirable to check the standard of learning of the network. This is achieved using a set of validation data which is different from the training data but was produced by the same mapping to be learned. This also consists of inputs and target outputs. The validation steps are as follows:

1. Propagate each validation input $\underline{p}_v$ forward through the trained network

2. From the output of the last layer, calculate the validation error $\underline{t}_v - \underline{a}_v^\Lambda$

3. Repeat steps 1-2 for each input-output pair in the validation set

4. Calculate the cost J

This cost will indicate that training has been successful if is not much greater than the training cost. Of course, visual inspection of the network output compared with the desired output should also be carried out, to ensure that there are not anomalous spikes which are having little effect on the cost.

The validation data set allows the user to observe how successfully the network generalises the learnt function to previously unencountered data points. If the generalisation is good, then the user can be confident that the function has been well approximated by the network. If not, it may be necessary to use another training data set, so that the unknown features of the function can be learnt.

3.3 Mathematical ship model

Although Mitsubishi Heavy Industries have a working 1/50th scale replica of the coupled FPSO and shuttle tanker, there has also been development of a simulation based on a nonlinear mathematical model, given in equations (2.1) to (2.37). The two ships use the same model with different mass and hydrodynamic parameters, plus the X_T, Y_T and N_T signals are augmented by forces due to the turret, hawser and tugboat.

3.3.1 Turret forces

The turret anchors the FPSO to the ocean floor via four steel cables, nominally acting at heading angles of 0, $\pi/2$, π and $-\pi/2$ from the turret. The distance from turret to anchoring for each cable is given by:

$$\begin{aligned}
d_{ta1} &= \sqrt{(L_c - X_{FPSO} + d_{to}(1 - cos\Psi_{FPSO}))^2 + (-Y_{FPSO} - d_{to}sin\Psi_{FPSO})^2} \\
d_{ta2} &= \sqrt{(d_{to}(1 - cos\Psi_{FPSO}) - X_{FPSO})^2 + (L_c - Y_{FPSO} - d_{to}sin\Psi_{FPSO})^2} \\
d_{ta3} &= \sqrt{(-L_c - X_{FPSO} + d_{to}(1 - cos\Psi_{FPSO}))^2 + (-Y_{FPSO} - d_{to}sin\Psi_{FPSO})^2} \\
d_{ta4} &= \sqrt{d_{to}(1 - cos\Psi_{FPSO}) - X_{FPSO})^2 + (-L_c - Y_{FPSO} - d_{to}sin\Psi_{FPSO})^2}
\end{aligned} \tag{3.28}$$

and the angle at which each cable acts is given by:

$$\begin{aligned}
\theta_{ta1} &= tan^{-1}\frac{-Y_{FPSO} - d_{to}sin\Psi_{FPSO}}{L_c - X_{FPSO} + d_{to}(1 - cos\Psi_{FPSO})} \\
\theta_{ta2} &= tan^{-1}\frac{L_c - Y_{FPSO} - d_{to}sin\Psi_{FPSO}}{d_{to}(1 - cos\Psi_{FPSO}) - X_{FPSO}} \\
\theta_{ta3} &= tan^{-1}\frac{-Y_{FPSO} - d_{to}sin\Psi_{FPSO}}{-L_c - X_{FPSO} + d_{to}(1 - cos\Psi_{FPSO})} \\
\theta_{ta4} &= tan^{-1}\frac{-L_c - Y_{FPSO} - d_{to}sin\Psi_{FPSO}}{d_{to}(1 - cos\Psi_{FPSO}) - X_{FPSO}}
\end{aligned} \tag{3.29}$$

where d_{taj} is the distance and θ_{taj} is the angle for the jth cable. X_{FPSO} and Y_{FPSO} are the earth-fixed X and Y coordinates of the FPSO and Ψ_{FPSO} is the heading. $L_c = 1714m$ is cable length and $d_{to} = 29.75m$ is the distance from the turret to the origin of the body-fixed coordinates. Each cable produces a tension force of:

$$T_j = \begin{cases} 43(d_{taj} - 1701) & , d > 1701 \\ 0 & , d \leq 1701 \end{cases} \tag{3.30}$$

where T_j is tension in the jth cable. The forces experienced by the ship via the turret are finally described by:

$$X_{trt} = \sum_{j=1}^{4} T_j cos\theta_{taj} cos\Psi_{FPSO} + \sum_{j=1}^{4} T_j sin\theta_{taj} sin\Psi_{FPSO} \tag{3.31}$$

$$Y_{trt} = \sum_{j=1}^{4} T_j sin\theta_{taj} cos\Psi_{FPSO} - \sum_{j=1}^{4} T_j cos\theta_{taj} sin\Psi_{FPSO} \tag{3.32}$$

$$N_{trt} = d_{to}(\sum_{j=1}^{4} T_j sin\theta_{taj} cos\Psi_{FPSO} - \sum_{j=1}^{4} T_j cos\theta_{taj} sin\Psi_{FPSO}) \tag{3.33}$$

where X_{trt}, Y_{trt} and N_{trt} are the surge, sway and yaw components.

3.3.2 Hawser forces

The hawser attaches the FPSO to the shuttle tanker. The distance from the stern of the FPSO to the bow of the shuttle tanker is given by:

$$\begin{aligned} d &= ((X_{FPSO} - X_{ST} - 0.5(L_{FPSO} cos\Psi_{FPSO} + L_{ST} cos\Psi_{ST}))^2 \\ &\quad +(Y_{FPSO} - Y_{ST} - 0.5(L_{FPSO} sin\Psi_{FPSO} + L_{ST} sin\Psi_{ST}))^2)^{\frac{1}{2}} \end{aligned} \tag{3.34}$$

and the angle of the hawser is given by:

$$\theta = tan^{-1} \frac{Y_{FPSO} - Y_{ST} - 0.5(L_{FPSO} sin\Psi_{FPSO} + L_{ST} sin\Psi_{ST})}{X_{FPSO} - X_{ST} - 0.5(L_{FPSO} cos\Psi_{FPSO} + L_{ST} cos\Psi_{ST})} \tag{3.35}$$

where X_{ST} and Y_{ST} are the earth-fixed X and Y coordinates of the shuttle tanker, Ψ_{ST} is the heading, and $L_{FPSO} = 255.7m$ and $L_{ST} = 230.0m$ are the lengths of each ship. The tension in the hawser is governed by the following:

$$T = \begin{cases} 0 & ,d < 48.34 \\ 3.6425d - 176.09 & ,d < 58.545 \\ 12.813d - 712.98 & ,d < 59.545 \\ 21.39d - 1223.7 & ,d \geq 59.545 \end{cases} \tag{3.36}$$

Resolving the tension into X and Y components:

$$T_X = T cos\theta, T_Y = T sin\theta \tag{3.37}$$

the forces experienced by each ship via the hawser are then described by:

$$\begin{aligned} X_{hwsFPSO} &= -T_X cos\Psi_{FPSO} - T_Y sin\Psi_{FPSO} \\ Y_{hwsFPSO} &= T_X sin\Psi_{FPSO} - T_Y cos\Psi_{FPSO} \\ N_{hwsFPSO} &= 0.5L_{FPSO}(T_Y cos\Psi_{FPSO} - T_X sin\Psi_{FPSO}) \end{aligned} \tag{3.38}$$

$$\begin{aligned} X_{hwsST} &= T_X cos\Psi_{ST} + T_Y sin\Psi_{ST} \\ Y_{hwsST} &= T_Y cos\Psi_{ST} - T_X sin\Psi_{ST} \\ N_{hwsST} &= 0.5L_{ST}(T_Y cos\Psi_{ST} - T_X sin\Psi_{ST}) \end{aligned} \tag{3.39}$$

where X_*, Y_* and N_* are the surge, sway and yaw components.

3.3.3 Tug boat forces

The tug boat is attached to the stern of the shuttle tanker and creates a force of $30 Tonnes$ at a heading of π radians. This produces surge, sway and yaw forces as follows:

$$X_{tug} = -30cos\Psi_{ST}, Y_{tug} = 30sin\Psi_{ST}, N_{tug} = -15L_{ST}sin\Psi_{ST} \tag{3.40}$$

3.3.4 Neural network teaching function

The model is useful for control design, but it also gives an idea as to the true relationship between the system variables, provided that the model is a good representation of the real ships. The significance of this is that the model can be used to decide which inputs and outputs are the best choice for the neural network. As noted earlier, a feedforward neural network does not have dynamic memory so it is necessary to rearrange the equations of motion of a ship into a form such that integration and differentiation are avoided. This rearrangement was performed earlier in equations (2.44) to (2.46), but now the full order model is required rather than a first order expansion. This is stated below:

$$\dot{u} = \frac{X_H + X_F + X_A + X_W}{m + m_x} = f(v, r, u, \Psi, \beta_C, X_F, X_A, X_W) \tag{3.41}$$

$$\dot{v} = \frac{Y_H + Y_F + Y_A + Y_W}{m + m_y} = g(u, r, v, \Psi, \beta_C, Y_F, Y_A, Y_W) \tag{3.42}$$

$$\dot{r} = \frac{N_H + N_F + N_A + N_W}{I_{zz} + J_{zz}} = h(r, u, v, \Psi, \beta_C, N_F, N_A, N_W) \tag{3.43}$$

where $X_F = X_T + X_{trt} + X_{hws} + X_{tug}$, $Y_T = Y_T + Y_{trt} + Y_{hws} + Y_{tug}$ and $N_T = N_T + N_{trt} + N_{hws} + N_{tug}$. Hence, f, g, and h are nonlinear coupled functions with eight inputs and three outputs, to be learnt by the neural network.

Both ships use the same model with different parameters, as stated earlier, so to demonstrate the effectiveness of neural networks for identification it is only really necessary to identify one ship at a time. Later, the networks could be joined to form a complete system.

3.4 System identification using Simulink data

Having decided upon the inputs and outputs of the neural network, the next step is to attempt to identify the ship dynamics using data from the Simulink simulation. This is to establish confidence in the network and eliminate any minor problems in the process, before moving on to real data.

For simplicity, the current is set to $U_C = 1m/s$ at $\beta_C = 0$, producing an operating point of $(u_0, v_0, r_0) = (1, 0, 0)$ when $\Psi = 0$. No wind or wave forces are applied. The two-ship system is then excited using a simple control law to apply square waves to the yaw and sway inputs of both shuttle tanker and FPSO, dependent on the FPSO heading only. The setpoint begins at 10^o, with $Y_{T(FPSO)} = -8.45$, $N_{T(FPSO)} = 1080$, $Y_{T(ST)} = -10.0$, $N_{T(ST)} = -1070$. When Ψ_{FPSO} reaches 10^o, the setpoint flips to -10^o with $Y_{T(FPSO)} = 8.45$, $N_{T(FPSO)} = -1080$, $Y_{T(ST)} = 10.0$, $N_{T(ST)} = 1070$. This sequence repeats until sufficient data has been gathered. The hawser is kept in tension throughout by 18 Tonnes of forward thrust from the FPSO, $X_{T(FPSO)} = 18$, and 30 Tonnes of reverse thrust from the tugboat. The combination of all of these controlled inputs also produces reactive force inputs from the FPSO turret and the hawser itself.

The shuttle tanker is chosen as the ship to identify, where the neural network input is a 7×1 vector consisting of u, v, r, $\Psi - \beta_C$, the sum of thruster, turret, hawser and tug forces in the surge direction, the sum in the sway direction and also the yaw torques. The output is a vector consisting of $\dot{u}$, $\dot{v}$ and $\dot{r}$. The simulation is run for 6000 data points, and the middle 2000 points are used for neural network training. The validation data is the entire set of 6000 points and, as a means of comparison, the network is implemented in Simulink such that the validation output may be compared with the network output (as shown in later Figures).

The neural network is a three-layer MLP with seven inputs and three outputs, but the internal structure is not completely defined by this description. A decision must be taken on the number of neurons and the activation function in each layer, although this is not an exact science and is open to trial and error. The number of neurons in the output layer must be three, as there are three outputs, but the first and second layers can be adjusted. Increasing the number of neurons will improve the accuracy of network function approximation to a point, beyond which the network begins to learn noise and idiosyncrasies of the data. This will lead to poor generalisation to other data sets, so a balance must be found when selecting the number of neurons. After experimentation, it is found that 14 input layer neurons and 7 hidden layer neurons, twice and the same as the number of inputs respectively, produce acceptable results. As for the activation function, this does not seem to noticeably influence the results in this application, so a standard log-sigmoid is used.

For the first attempt at network training, the number of epochs is 500 and the mean square error (MSE) between target and actual outputs is 1.4×10^{-5} at the end of training. The results are not good, however, as the network output contains large numbers of spikes and generally fails to match the validation data. The mean square output (MSO) is only 2.5×10^{-6}, so by comparison with the MSE it is evident that the training is unsuccessful. After some experimentation with number of epochs, it appears that ill-conditioned data is the reason for poor performance of the neural network - The acceleration in the surge direction is almost three orders of magnitude greater than the yaw acceleration. Normalising all of the data such that the maximum of every signal is equal to unity eases numerical problems within the neural net, as noted by Roskilly and Mesbahi (1996a).

The training algorithm is run once again with 500 epochs and normalised data, yielding a mean square error of 3.1×10^{-4} for MSO of 0.69. Clearly, this is a great improvement, although the MSE gradient is still high enough after 500 epochs to justify a longer training period. Hence, the algorithm is run for 1000 epochs, giving a MSE of 2.5×10^{-4} and results as depicted in Figures 3.5, 3.6, and 3.7. These Figures show the comparison between 6000 seconds of output data recorded from Simulink, and the output of the neural net that has been trained on only the middle third. The only significant errors occur in the first third where the ship oscillations are beginning to build. The performance of the neural net is satisfactory in these examples.

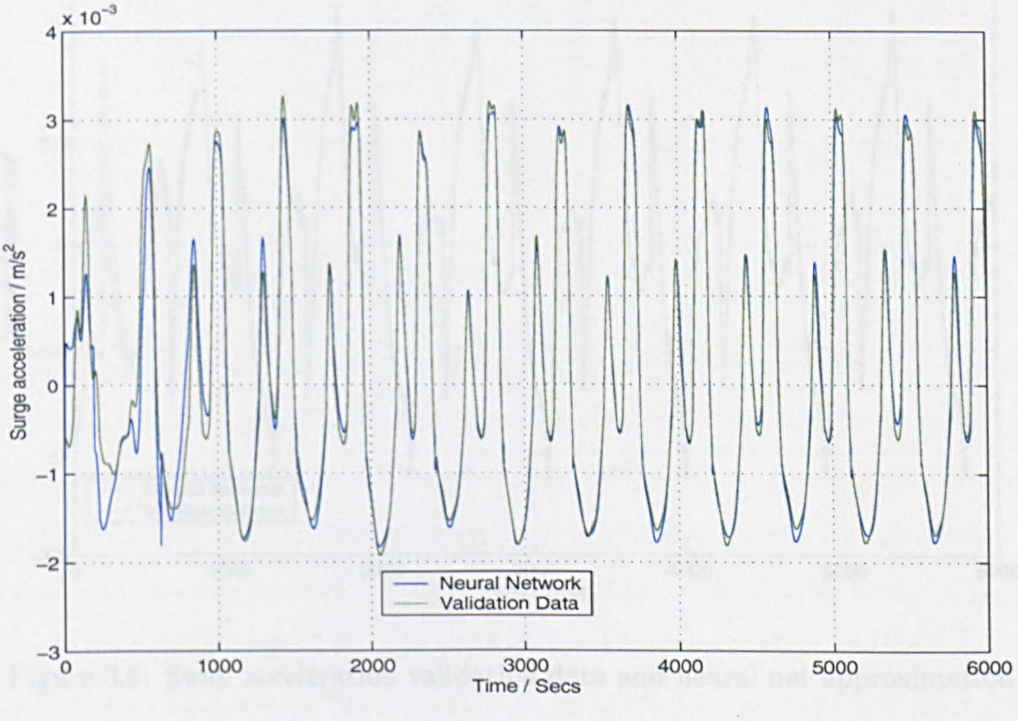

Figure 3.5: Surge acceleration validation data and neural net approximation

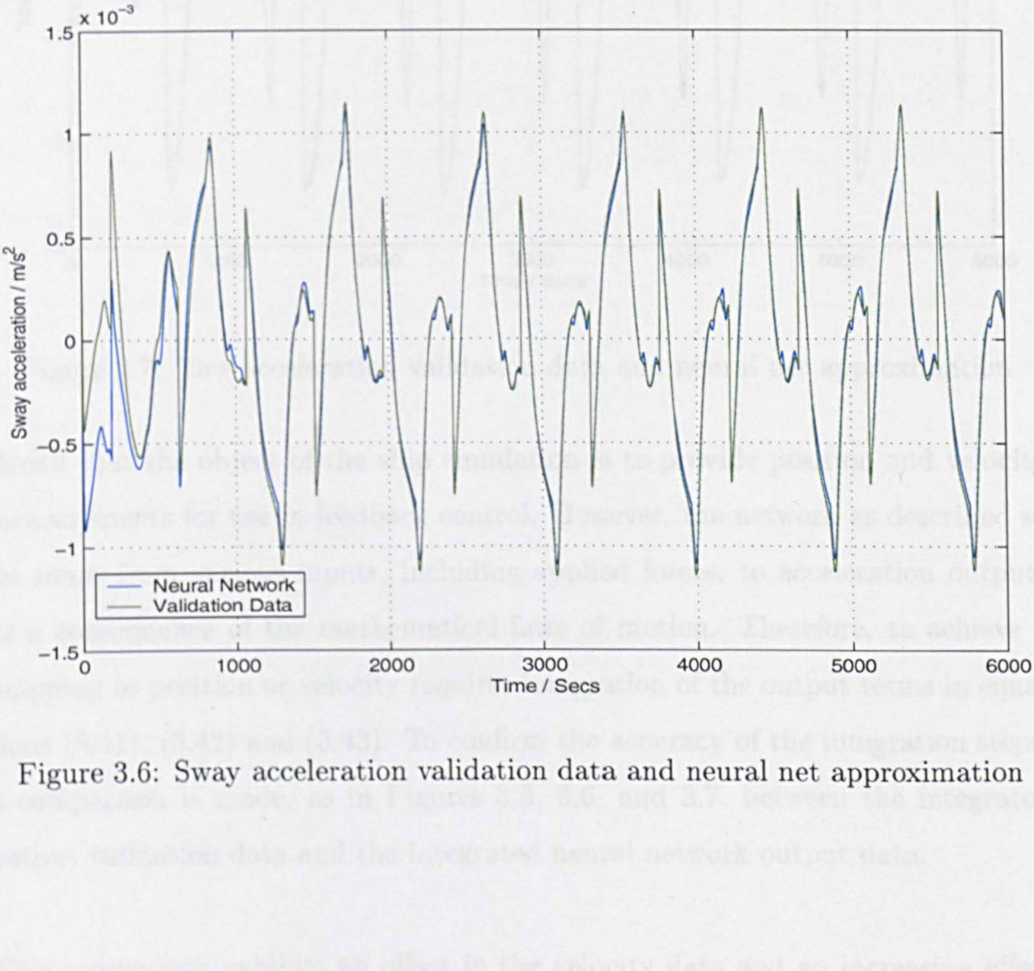

Figure 3.6: Sway acceleration validation data and neural net approximation

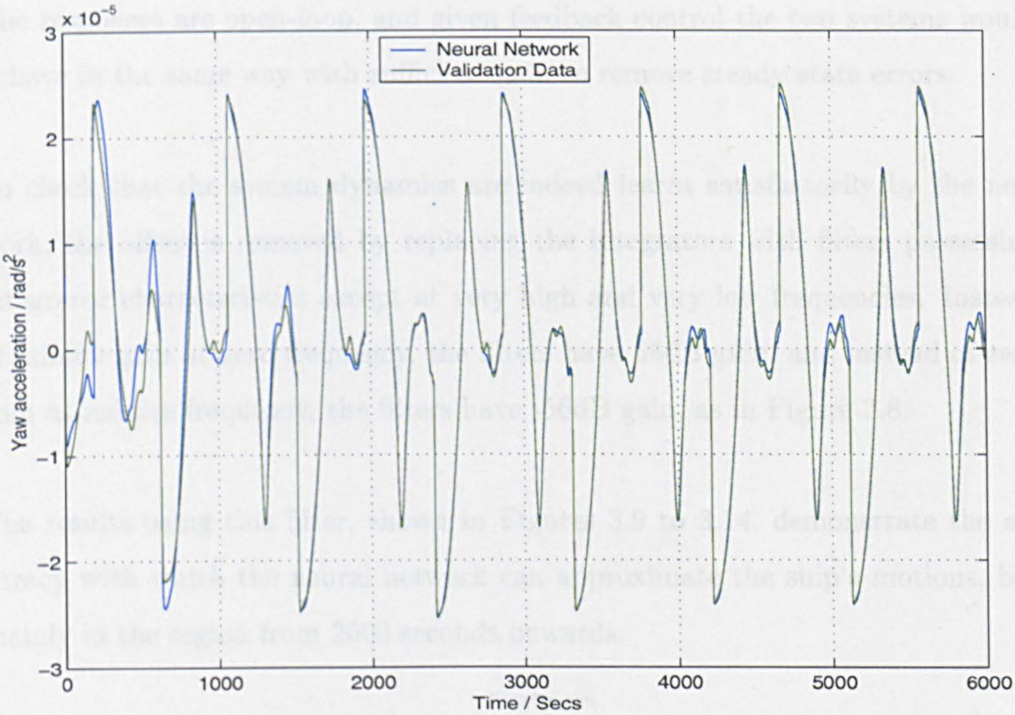

Figure 3.7: Yaw acceleration validation data and neural net approximation

Recall that the object of the ship simulation is to provide position and velocity measurements for use in feedback control. However, the network as described so far maps from various inputs, including applied forces, to acceleration outputs as a consequence of the mathematical laws of motion. Therefore, to achieve a mapping to position or velocity requires integration of the output terms in equations (3.41), (3.42) and (3.43). To confirm the accuracy of the integration steps, a comparison is made, as in Figures 3.5, 3.6, and 3.7, between the integrated output validation data and the integrated neural network output data.

This comparison exhibits an offset in the velocity data and an increasing offset in the position data due to the accumulation of small errors in acceleration of Figures 3.5/3.6/3.7. Nonetheless, the "shape" of these neural net responses appear to be almost identical to the desired shape, so it is felt that the non-DC system dynamics are well approximated and the offset is not a great concern.

The responses are open-loop, and given feedback control the two systems would behave in the same way with sufficient gain to remove steady-state errors.

To check that the system dynamics are indeed learnt satisfactorily by the network, the offset is removed by replacing the integrators with filters possessing integrator characteristics except at very high and very low frequencies. Instead of infinite gain at zero frequency, the filters have 56dB gain, and instead of zero gain at infinite frequency, the filters have -56dB gain, as in Figure 3.8.

The results using this filter, shown in Figures 3.9 to 3.14, demonstrate the accuracy with which the neural network can approximate the ship's motions, but mainly in the region from 2000 seconds onwards.

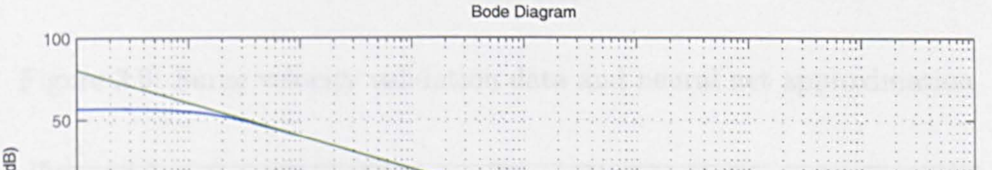

Figure 3.8: Filter frequency response used to approximate an integrator

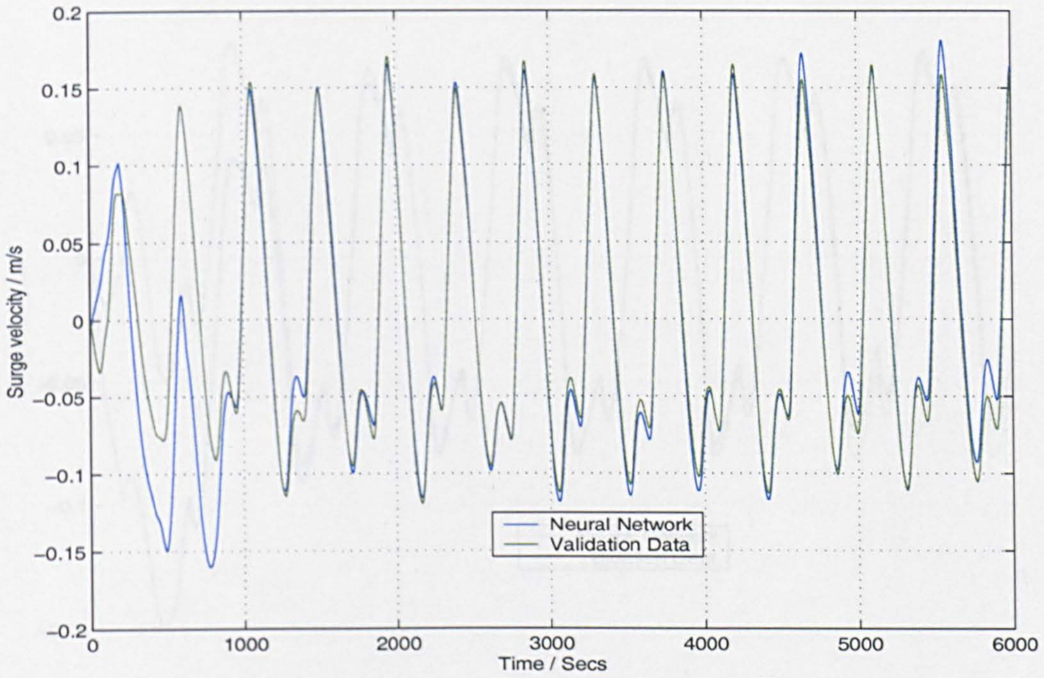

Figure 3.9: Surge velocity validation data and neural net approximation

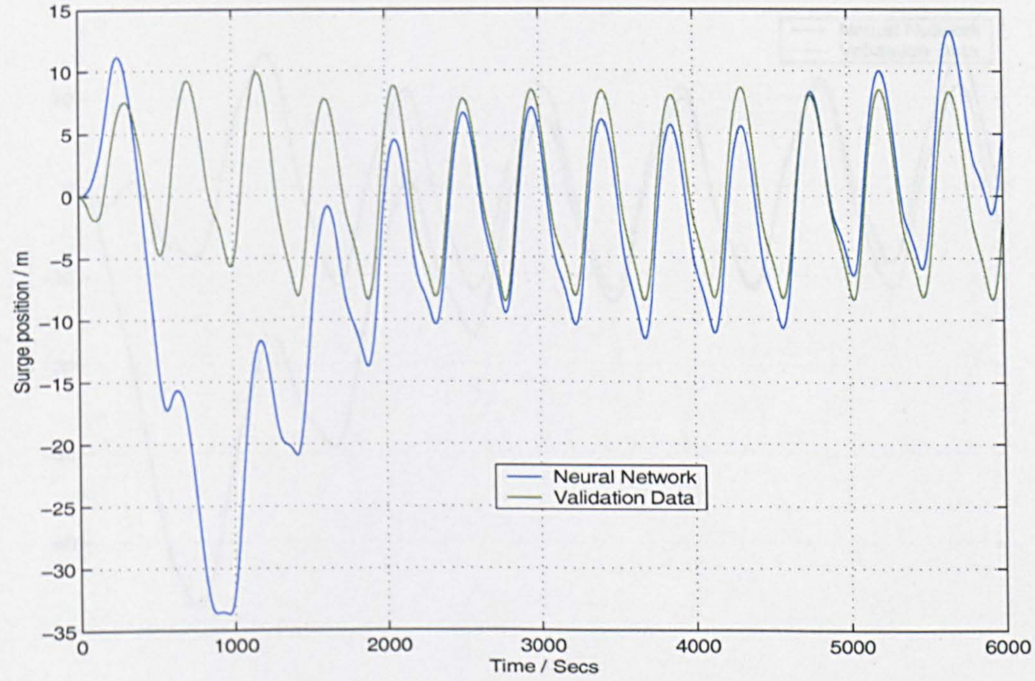

Figure 3.10: Surge position validation data and neural net approximation

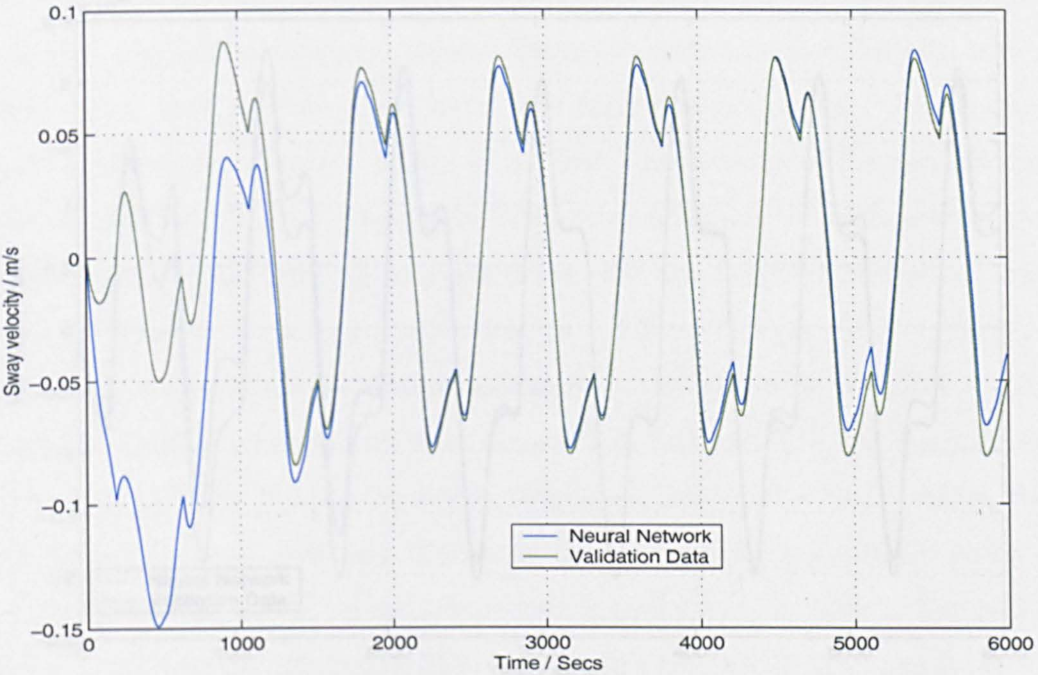

Figure 3.11: Sway velocity validation data and neural net approximation

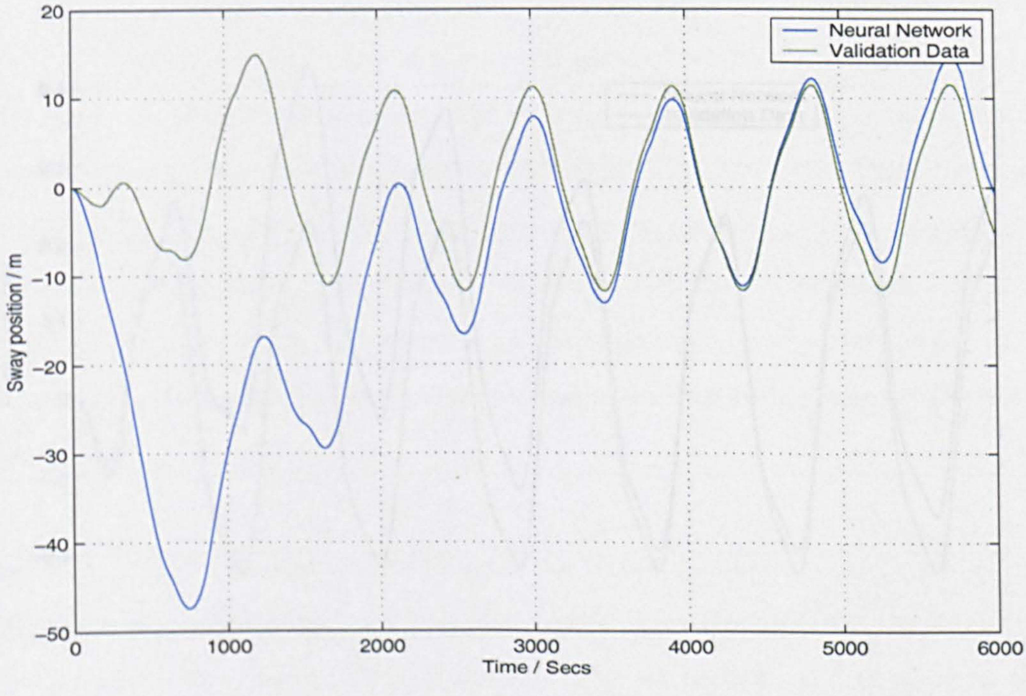

Figure 3.12: Sway position validation data and neural net approximation

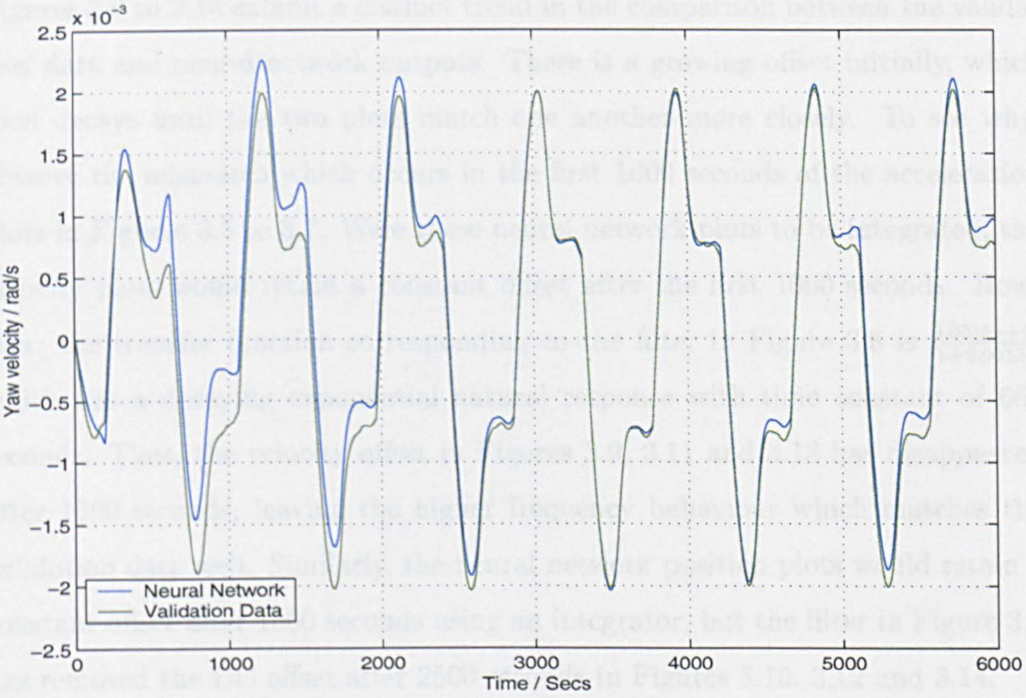

Figure 3.13: Yaw velocity validation data and neural net approximation

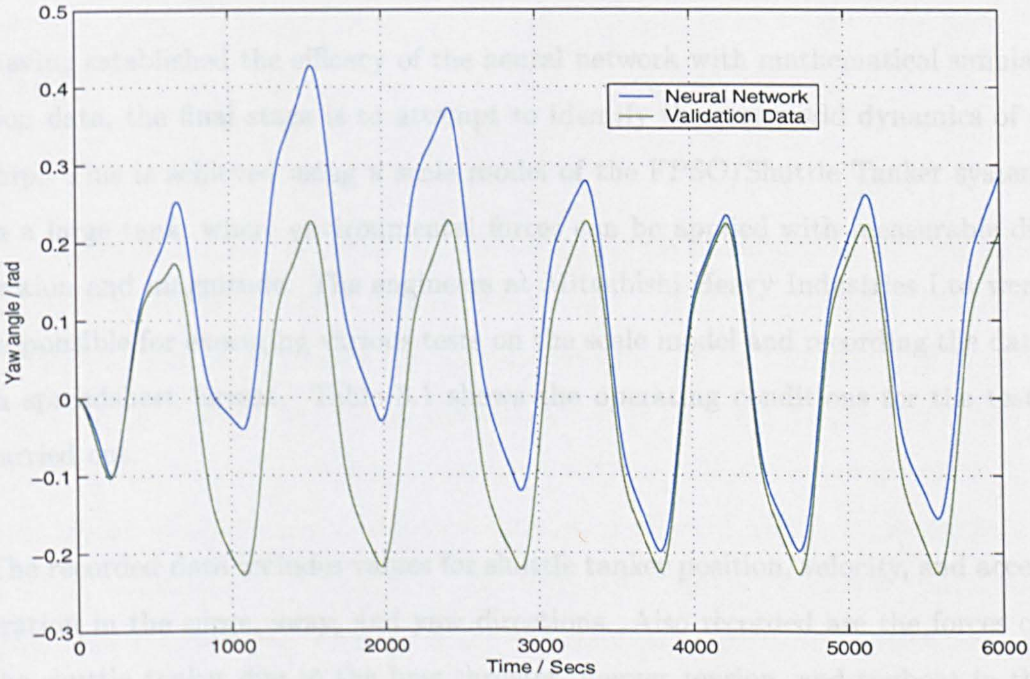

Figure 3.14: Yaw angle validation data and neural net approximation

Figures 3.9 to 3.14 exhibit a distinct trend in the comparison between the validation data and neural network outputs. There is a growing offset initially, which then decays until the two plots match one another more closely. To see why, observe the mismatch which occurs in the first 1000 seconds of the acceleration plots in Figures 3.5 to 3.7. Were these neural network plots to be integrated, the velocity plots would retain a constant offset after the first 1000 seconds. However, the transfer function corresponding to the filter in Figure 3.8 is $\frac{0.0015s+1}{s+0.0015}$, which has a decaying exponential natural response with time constant of 667 seconds. Thus, the velocity offset in Figures 3.9, 3.11 and 3.13 has disappeared after 1500 seconds, leaving the higher frequency behaviour which matches the validation data well. Similarly, the neural network position plots would retain a constant offset after 1500 seconds using an integrator, but the filter in Figure 3.8 has removed the DC offset after 2500 seconds in Figures 3.10, 3.12 and 3.14.

3.5 System identification using tank test data

Having established the efficacy of the neural network with mathematical simulation data, the final stage is to attempt to identify the real-world dynamics of a ship. This is achieved using a scale model of the FPSO/Shuttle Tanker system in a large tank, where environmental forces can be applied with measurable direction and magnitude. The engineers at Mitsubishi Heavy Industries Ltd were responsible for executing various tests on the scale model and recording the data in spreadsheet format. Table 3.1 shows the operating conditions for the tests carried out.

The recorded data includes values for shuttle tanker position, velocity, and acceleration in the surge, sway, and yaw directions. Also recorded are the forces on the shuttle tanker due to the bow thruster, hawser tension, and tugboat in the surge, sway and yaw directions. The wave height and FPSO angle are the other variables to be recorded. It is noted the shuttle tanker's stern propeller provides no force and the rudder is kept at zero degrees.

Table 3.1: Operating conditions for tank tests

			FPSO			Shuttle Tanker	
Exp. No.	Current Vel. (knots)	Wave Ht. (m)	Heading Change (deg)	Thruster Angle (deg)	Thruster Force (tons)	Bow Thruster Force (tons)	Tug Force (tons)
61871	2	0	15	90	30	0	30
61877	2	2.0	15	90	30	0	30
61875	2	3.0	15	90	30	0	30
61879	2	4.0	15	90	30	0	30
61827	2	0	0	0	0	0→20	30
61828	2	0	0	0	0	0→20	30
61837	2	0	0	90	0→30	0	30
61855	2	3.0	0	90	0→30	0	30
61862	2	4.0	0	90	0→30	0	30

In terms of neural network training, it is assumed that a real ship follows the time-invariant mathematical laws stated in equations 3.41, 3.42, and 3.43. Therefore, the inputs and outputs used are almost the same as for the Simulink data in Section 3.4. Namely, the neural network input is an 8×1 vector consisting of u, v, r, y, the sum of thruster, hawser and tug forces in the surge direction, the sum in the sway direction, the yaw torques, and wave height. The output is a vector consisting of $\dot{u}$, $\dot{v}$ and $\dot{r}$. The significant difference here is that waves have been incorporated, whereas in the Simulink case the ships are operating in a completely calm environment. Again, the shuttle tanker is chosen as the ship to be identified. The number of neurons is 16 in the first layer, 8 in the second layer and 3 in the output layer. These values are twice and equal to the number of inputs and equal to the number of outputs respectively. The activation functions are log-sigmoids in each layer.

The first attempt at network training is made using the first half of the data set from experiment 61871 in Table 3.1. Once again, the data is normalised to the maximum values for each variable, and a 500-epoch training run is attempted. The mean square error (MSE) is 3.1×10^{-4} given a mean square signal of 0.53. Using the whole data set for verification demonstrates good generalisation by the network in the surge and sway directions, but rather more significant errors for yaw. Hence, it is decided to re-train using the first three quarters of the data set. This time, the MSE is 6.3×10^{-4} for a mean square signal of 0.64. This result appears to be slightly worse numerically, but the yaw errors are significantly reduced. Figures, 3.15, 3.16, and 3.17 demonstrate the effective generalisation by the network after 3470 seconds, where the data is outside of the training set.

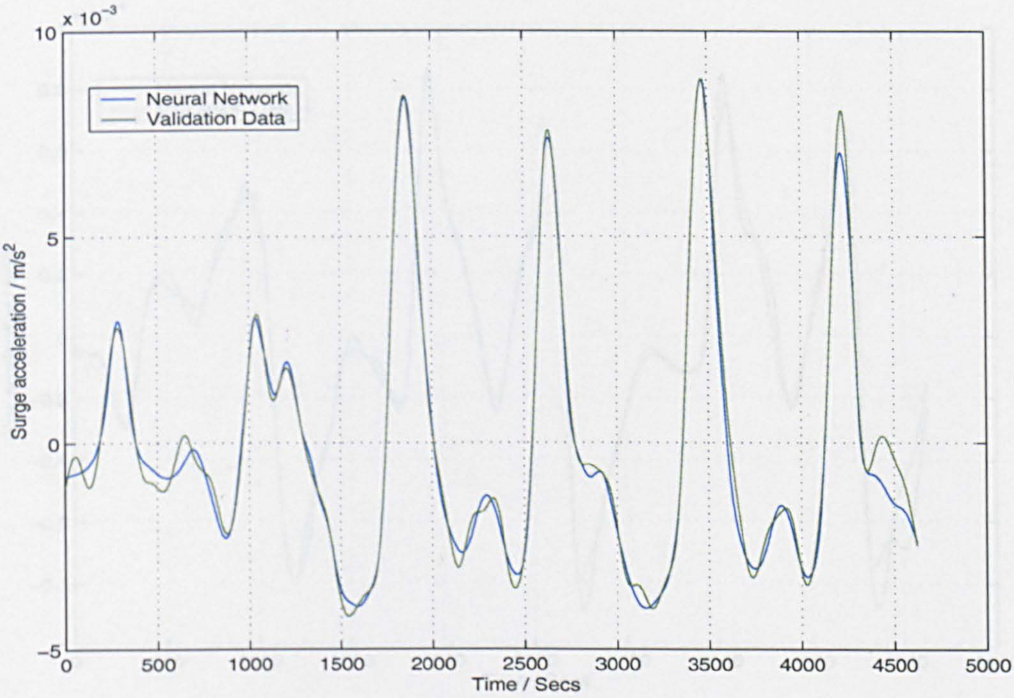

Figure 3.15: Surge acc. validation data (61871) and neural net approximation

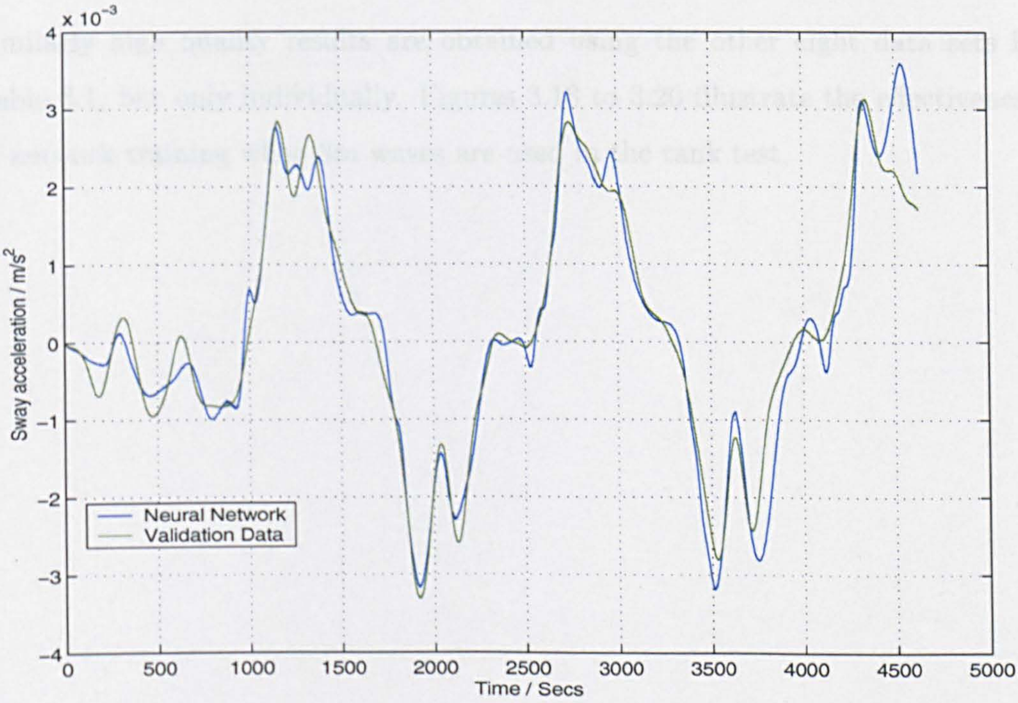

Figure 3.16: Sway acc. validation data (61871) and neural net approximation

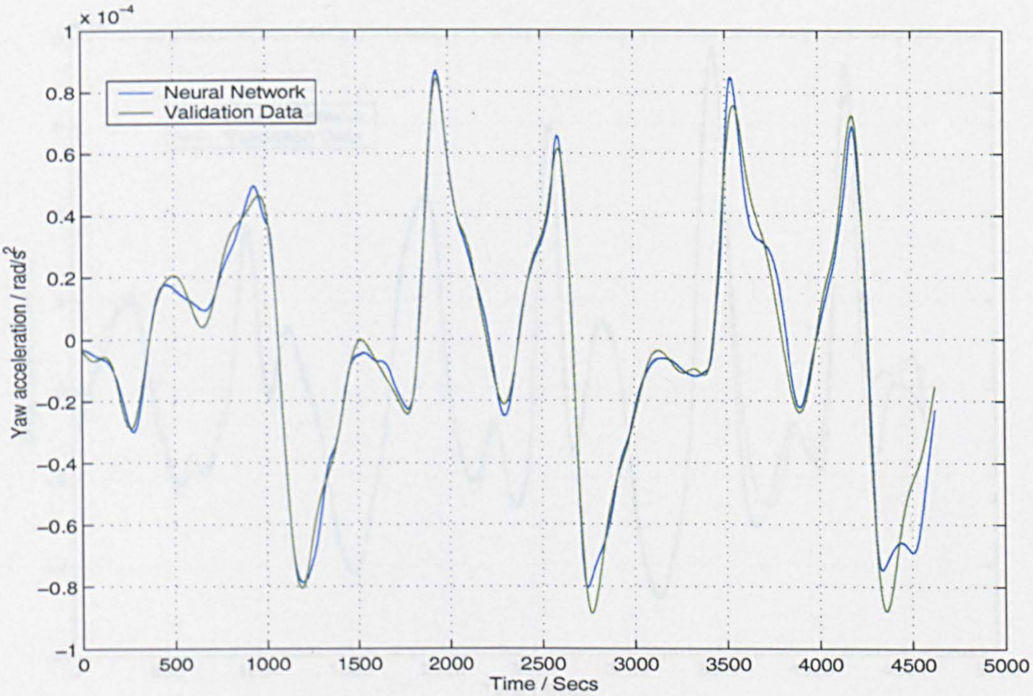

Figure 3.17: Yaw acc. validation data (61871) and neural net approximation

Similarly high quality results are obtained using the other eight data sets in Table 3.1, but only individually. Figures 3.18 to 3.20 illustrate the effectiveness of network training when $3m$ waves are used in the tank test.

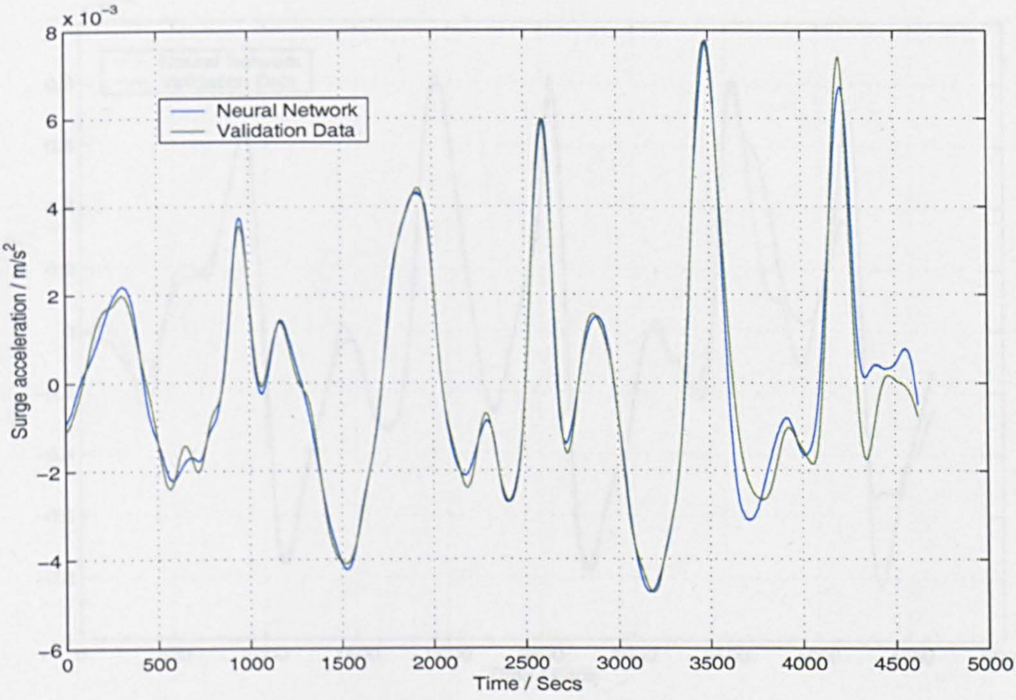

Figure 3.18: Surge acc. validation data (61877) and neural net approximation

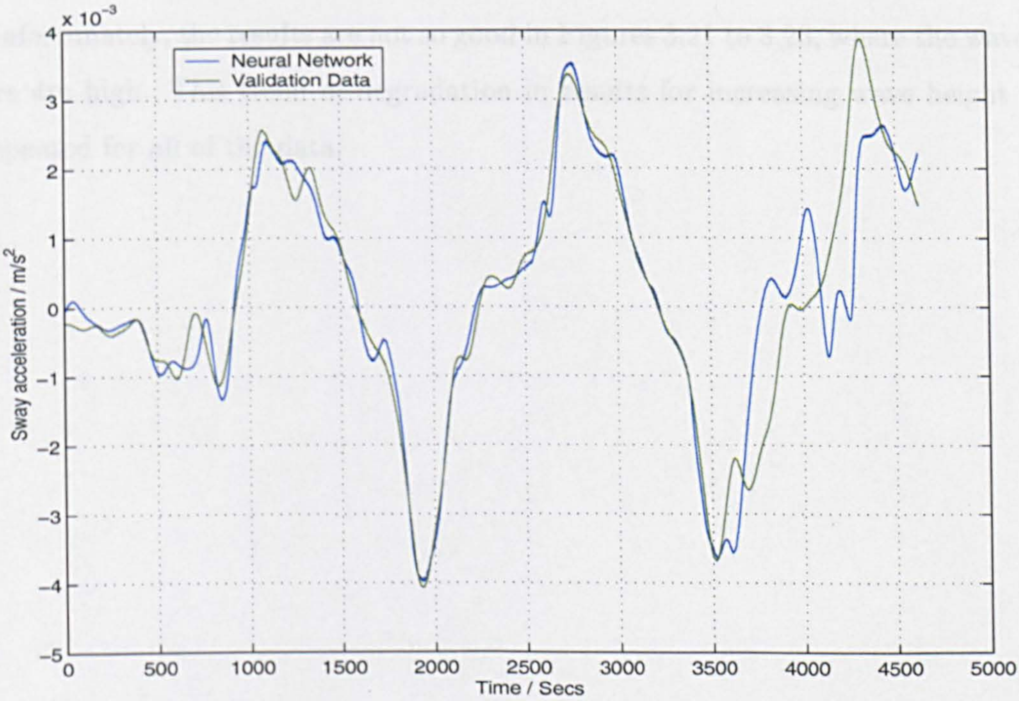

Figure 3.19: Sway acc. validation data (61877) and neural net approximation

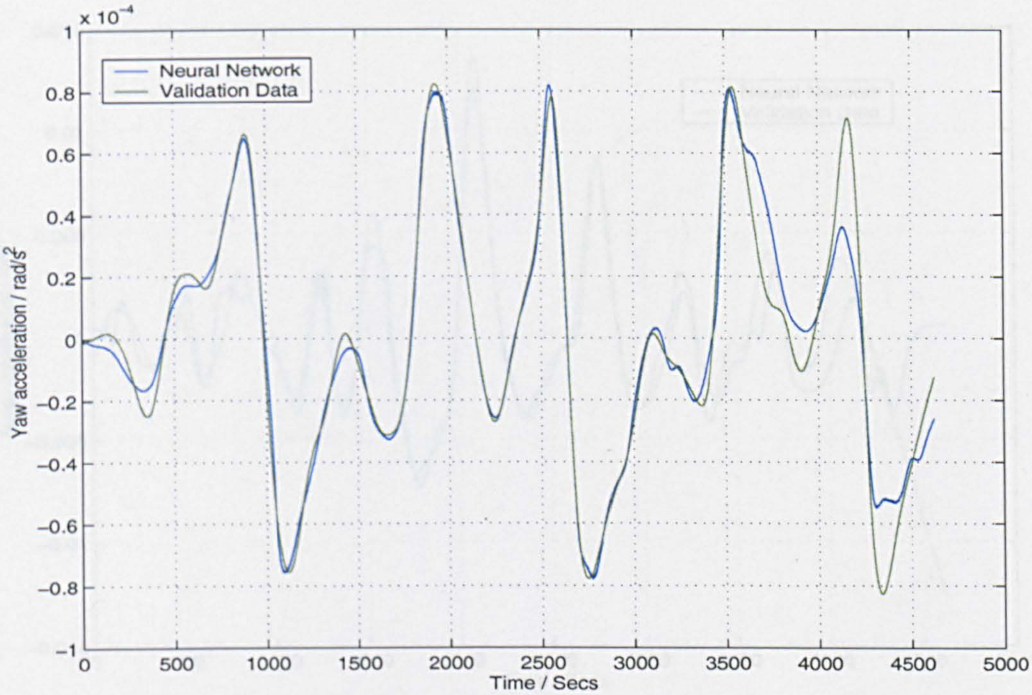

Figure 3.20: Yaw acc. validation data (61877) and neural net approximation

Unfortunately, the results are not so good in Figures 3.21 to 3.23, where the waves are $4m$ high. This trend of degradation in results for increasing wave height is repeated for all of the data.

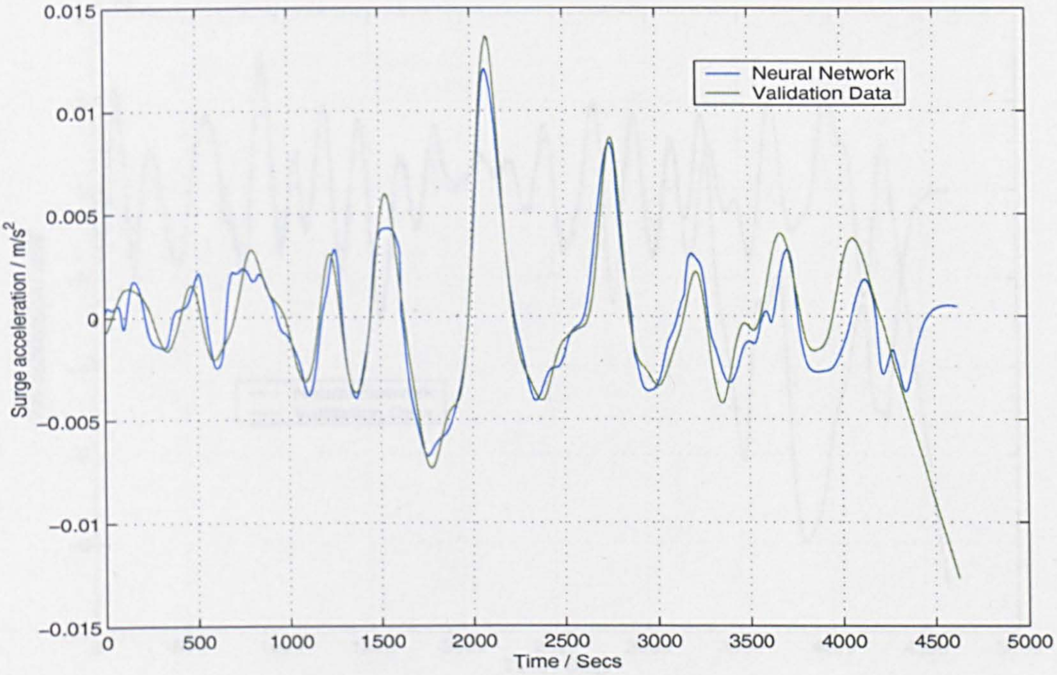

Figure 3.21: Surge acc. validation data (61879) and neural net approximation

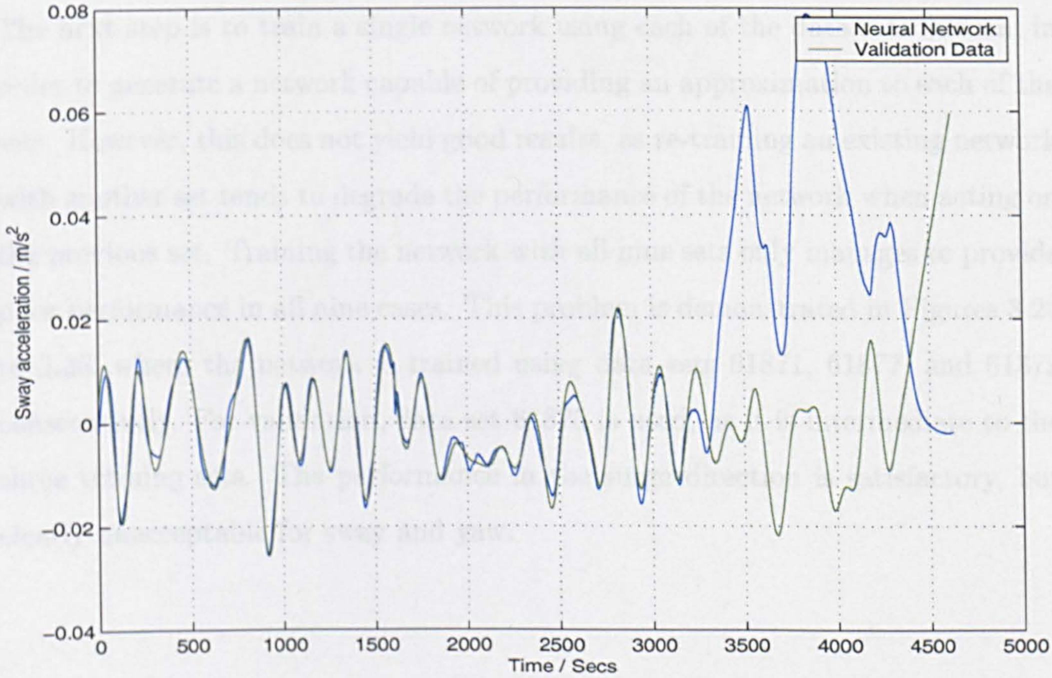

Figure 3.22: Sway acc. validation data (61879) and neural net approximation

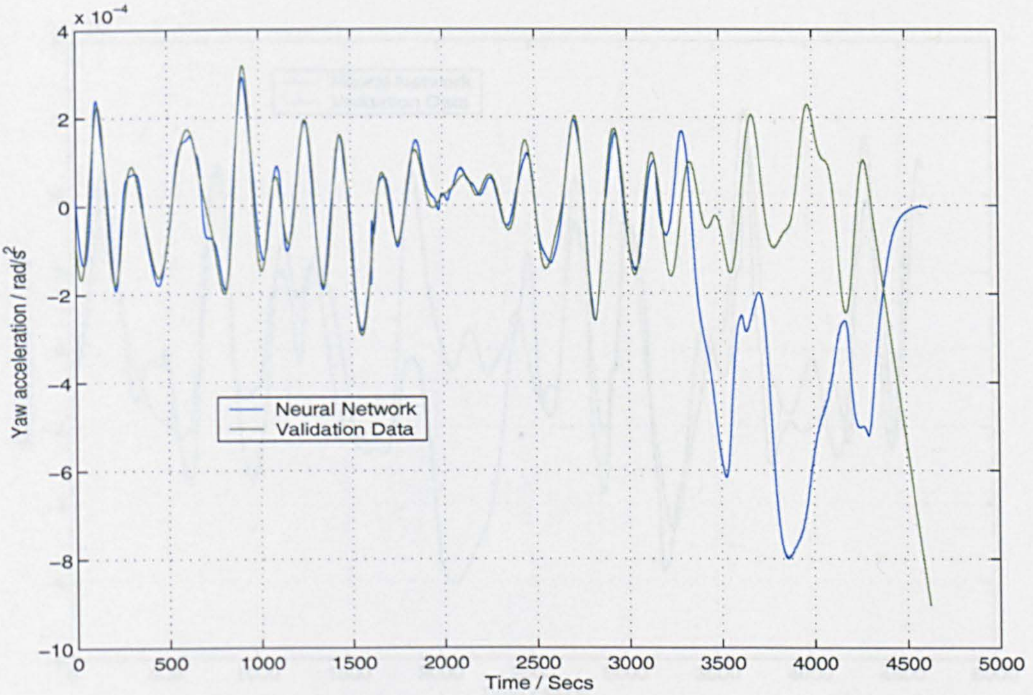

Figure 3.23: Yaw acc. validation data (61879) and neural net approximation

The next step is to train a single network using each of the data sets in turn, in order to generate a network capable of providing an approximation to each of the sets. However, this does not yield good results, as re-training an existing network with another set tends to degrade the performance of the network when acting on the previous set. Training the network with all nine sets only manages to provide poor performance in all nine cases. This problem is demonstrated in Figures 3.24 to 3.26, where the network is trained using data sets 61871, 61877, and 61879 consecutively. For validation, data set 61875 is used, as it is intermediate to the three training sets. The performance in the surge direction is satisfactory, but clearly unacceptable for sway and yaw.

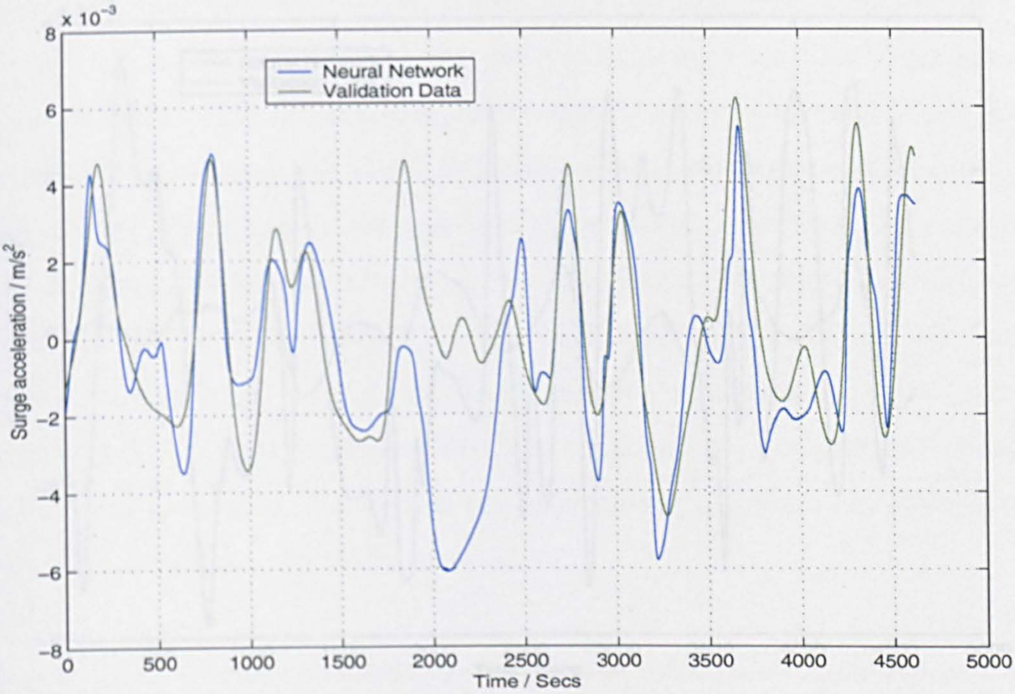

Figure 3.24: Surge acc. validation data (61875) and neural net approximation

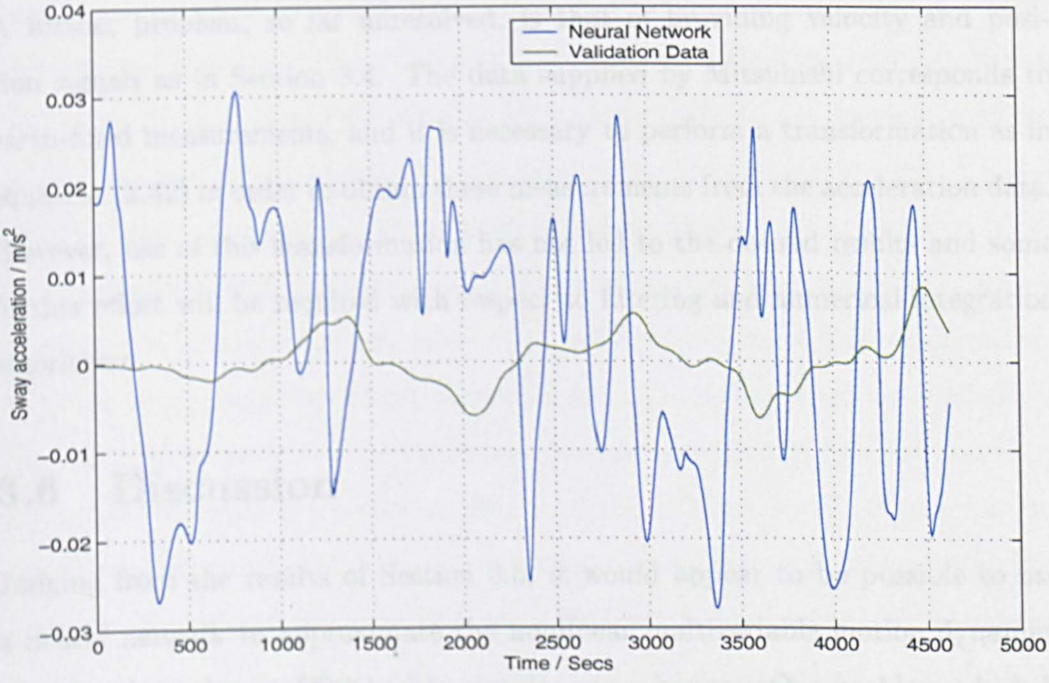

Figure 3.25: Sway acc. validation data (61875) and neural net approximation

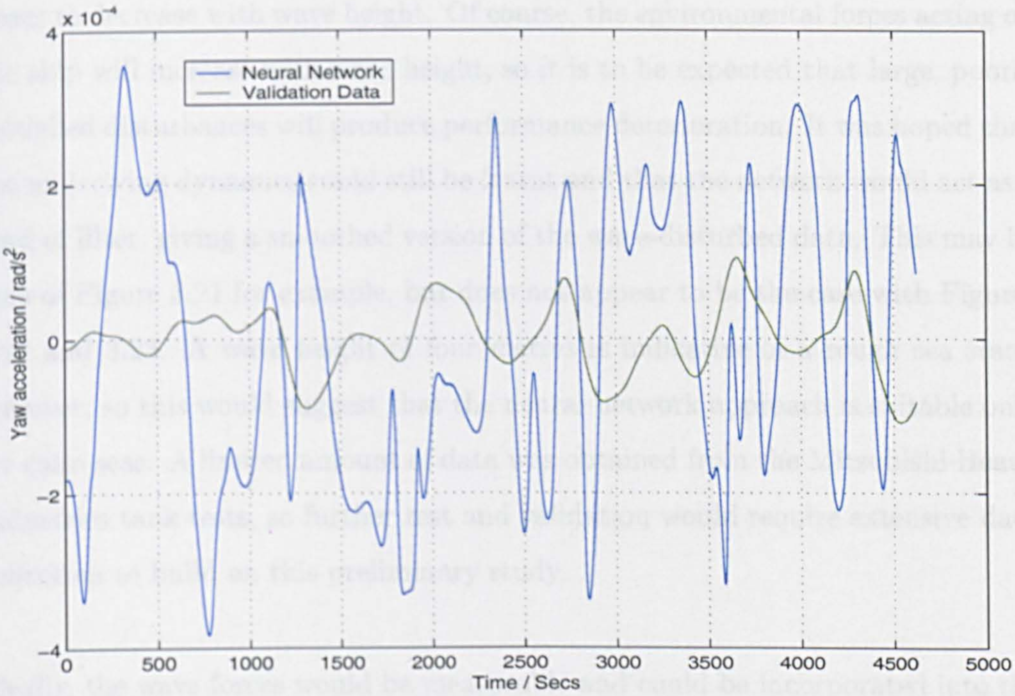

Figure 3.26: Yaw acc. validation data (61875) and neural net approximation

A further problem, so far unresolved, is that of providing velocity and position signals as in Section 3.4. The data supplied by Mitsubishi corresponds to earth-fixed measurements, and it is necessary to perform a transformation as in equation (2.42) in order to obtain these measurements from the acceleration data. However, use of this transformation has not led to the desired results and some further effort will be required with respect to filtering and numerical integration algorithms.

3.6 Discussion

Judging from the results of Section 3.5, it would appear to be possible to use a neural network to approximate the nonlinear multivariable motion dynamics of a shuttle tanker or FPSO when coupled via a hawser. One problem which is evident by examination of Figures 3.15 to 3.23 is that neural net performance appears to decrease with wave height. Of course, the environmental forces acting on the ship will increase with wave height, so it is to be expected that large, poorly modelled disturbances will produce performance deterioration. It was hoped that the underlying dynamics could still be learnt and that the network would act as a kind of filter, giving a smoothed version of the wave-disturbed data. This may be true of Figure 3.21 for example, but does not appear to be the case with Figures 3.22 and 3.23. A wave height of four metres is indicative of a rough sea state, however, so this would suggest that the neural network approach is suitable only for calm seas. A limited amount of data was obtained from the Mitsubishi Heavy Industries tank tests, so further test and validation would require extensive data collection to build on this preliminary study.

Ideally, the wave forces would be measurable and could be incorporated into the input data vector. In reality, it is not possible to isolate and directly measure the ship forces due to waves only, so there are two possibilities to augment the input vector, using knowledge of the wave spectrum and the plant respectively. Waves are often modelled according to Pierson and Moskowitz (1963) and produce forces as in Section 2.2.3. Given the amplitude, frequency, phase and incident angle of the fundamental wave components, it is reasonable that the neural network could infer the force components due to waves. The problem would then be to select sensors to measure the relevant signals. The second possibility would be to utilise values of ship translational and rotational inertia, and a simple Newtonian model, to give a rough estimate of wave forces in each of the three degrees of freedom. The known thruster forces, ship inertia and accelerations could be used to predict ship behaviour, with any errors attributed to the incident waves. In this way, an estimator of wave forces could be constructed and used in neural network training. Of course, this transfers some of the modelling effort from the neural network to the engineer, but ship inertias are usually known quite accurately and the wave estimator would be simple, see Martin (1999) for example.

Another problem is that the network experiences difficulty in learning to approximate more than one data set simultaneously. This may be due to the nine data sets representing rather disparate operating conditions, giving the network conflicting dynamics to learn. The investigation was carried out with the assumption that the ship dynamics do not change greatly over time and with environmental differences, so an idea to make progress with this problem is to use several networks, one for each likely operating point, then to train each network extensively using data from each point. The ship's motions could then be obtained from one network corresponding to the present operating conditions, or from a weighted sum of the various network outputs. Such an approach was suggested in Eikers and Karim (1999).

The final problem encountered in Section 3.5 is the transformation from acceleration outputs to earth-fixed velocity and position signals. Further examination of the processes used in producing the tank test data is required, and examination of the use of integrators leading to offsets, similarly to Section 3.4. The use of feedback control should mean that offsets are unimportant provided that the remaining ship dynamics are an accurate representation of the real ship.

As mentioned in Section 3.1, neural networks have predominantly been explored for their application to control of marine systems rather than identification, hence a discussion of possible further work on the DP control problem with neural networks now follows. There are various approaches to control which exploit the ability of neural networks to learn arbitrary non-linear mappings. The following examples are discussed in Warwick et al. (1992).

Supervised control involves learning the actions of a human when it has proven difficult to design an effective automatic controller using accepted classical or modern control techniques. If the control task is particularly arduous or repetitive, then replacing the human operator with a neural network is clearly advantageous. Direct Inverse control involves learning the plant inverse, which is then used in series with the plant to create an overall identity mapping. Of course, this technique suffers from sensitivity to model uncertainty and may not be used with an unstable plant. In Model Reference control, the closed-loop system is expected to follow the output from a reference model generating the desired response. The error between the actual system and the reference is used to train the network, which acts as the controller.

Internal Model control is another possibility, where the closed-loop system contains an internal neural network model of the plant in parallel with the actual plant. The signal which is fed back to the comparator is the difference between the outputs of this internal model and the real system. The controller is then the inverse of the neural network model, with the property that the closed loop is stable given that both plant and controller are stable. Further, under the assumption that the inverse is exact, then the system output will perfectly follow the reference with no steady-state offset. If this assumption is invalid, then robustness to model uncertainty may be attained by introducing a filter in cascade with the controller.

One more application of a neural network is to produce a form of predictive control. In this case, the network learns to predict the next plant output as a function of previous inputs and current and previous outputs. This prediction is used when minimising a performance cost function in order to compute the optimal control signal. The prediction is only one step ahead but, for a nonlinear system, will be more accurate than that obtained from a linear ARMA (AutoRegressive Moving Average) model, for example.

Burns (1995) has used a neural network as ship autopilot at various forward speeds. An optimal state-feedback controller is tuned for a set of velocities, then the network must learn to approximate the correct gain with varying speed. The results show that the network performs in a slightly sub-optimal manner, but has the advantage that it does not have to re-compute controller parameters as velocity varies. Vukic et al. (2000) combine two neural networks with conventional PID velocity and position control loops on a six degree-of-freedom underwater vessel. The scheme is adaptive in that the network attempts to minimise the error signal from each loop through continual training. The networks produce control input signals in addition to the PID controllers and results show that tracking performance is improved, plus the controller is robust to large changes in plant dynamics.

Unar and Murray-Smith (1999) use radial basis function (RBF) networks and local model networks (LMNs) to investigate the autopilot problem. RBF network function approximation is superior to that of MLPs, although the number of basis functions must be equal to the number of data points, which may be large. An LMN is a set of RBFs weighted by some activation function, so that the overall output is a weighted sum of RBF outputs. The advantages are that the structure is transparent and a priori knowledge may be incorporated. Each type of network is trained via supervision by PID controllers, which have themselves been tuned for different operating speeds. Simulation results show that the neural networks provide performance improvements compared with a single PID controller operating over a range of conditions.

Some other examples include Yuh (1990), Fortuna and Muscato (1996), and Zhang et al. (1996), but Hardier (1997) is most relevant to the ships dealt with in this Chapter, as it is an extension of Hardier (1995) discussed in Section 3.1. Hardier (1997) trains a controller network to minimise a cost function of weighted output errors and inputs, using the identified ship neural network as a predictor of the real ship behaviour. The controller is recurrent and consists of a number of stages intended to implement filtering, gain matrices and control constraint handling. The gain stage is acted on by a subnetwork as a function of the operating condition. The controller is trained for several hours in high sea conditions and the performance is again described as "satisfactory". No comparison is made with a classical controller or a combined Kalman filter and state estimate feedback scheme, so it is not known whether the network presents any significant performance advantage.

No research publications have been found which investigate DP control of a coupled ship system specifically. Hence, future work could concentrate on classical or neural controllers to regulate the position of each ship individually and to maintain separation jointly.

3.7 Conclusions

This Chapter began by discussing the problem of ship modelling and identification of parameters for use in DP control. A suitable ship model must be nonlinear and multivariable, so the number of appropriate techniques is actually fairly limited. In the past, this problem has been approached by deriving a simplified model and finding the relevant parameters approximately, or with more sophistication by using an extended Kalman filter. Neural networks are a valuable technique for approximating nonlinear, multivariable functions, and clearly also have potential.

The particular problem addressed in this Chapter was to investigate the use of a multilayer perceptron feedforward network for identifying ship dynamics of a coupled FPSO and shuttle tanker. The main contribution of the Chapter was to demonstrate that the network could learn and generalise the dynamics from real data provided by a Mitsubishi Heavy Industries 1/50th scale model.

The theoretical principles of neural networks and their mode of learning were presented, in addition to a mathematical model to describe the forces acting on the ships due to a turret, hawser and tugboat. The coupled ships were modelled in Simulink using the equations from Chapter 2, plus the additional forces mentioned. Data from this simulation was then used to check the viability of using a neural network for the problem. The differential equations describing each ship were rearranged to state acceleration in terms of velocities, angles and forces, thereby removing the need for memory in the network. A ship could therefore be expressed as a function with eight inputs and three outputs.

The simulation data was gathered by implementing a simple control law to yaw the ships back and forth. This was sufficient to excite the various dynamics and forces due to thrusters, turret, hawser and tugboat. The data from the shuttle tanker was assembled into a training set for the neural network. With normalised data, it was discovered that the dynamics were learnt well and generalised from the small training set to the complete validation set. Integrating to generate velocity and position signals also demonstrated that the network was successful in learning, although a filter was required to remove steady-state offset. The non-DC performance was otherwise encouraging, so it was possible to progress to using real data from Mitsubishi's scale model.

Here, input and output signals were as in the simulation case, but with the addition of wave height as an input. Mitsubishi conducted several tests to excite the coupled ships in a tank with controllable environment. The first network training procedure was conducted with data from a still environment, that is with no waves applied. Again using normalised data, the network outputs were found to match the data with little error. Further training with a wave height of two metres yielded acceptable results, but with four metre wave height, the errors were not insignificant. The network was also trained using all of the data together from the previous three cases, before being validated with data from three metre wave height. In this case, the results were of poor quality. Indeed, any attempt to train the network with more than one data set at a time resulted in significant errors, even for validation with a data set used in training.

The main flaw in the use of neural networks is that approximating more than one data set simultaneously does not appear to be easily achievable. In further work, several networks could be used, one for each likely operating point, with each network trained extensively using data from each point. Ship motion could be produced by one network corresponding to the present operating condition, or from a weighted sum of the various network outputs. A further, less fundamental flaw is the transformation from acceleration outputs to earth-fixed velocity and position signals. It may be necessary to use a different process in production of the tank test data, and to take care in the use of integrators for velocity and position signals.

The main advantage of the neural network approach is that it is not necessary to know the precise structure of the equations of motion or to identify particular parameters. The multivariable character of the problem does not present a difficulty either, as the interconnected nature of the network is well-suited to cope with this. The ship is simply an input-output mapping which can behave in a highly nonlinear manner with no loss of accuracy in signal reproduction.

Chapter 4

Restricted-Structure Multiple-Model Adaptive Control

This Chapter describes the application of a novel adaptive controller to the ship DP control problem, where multiple linear models are used to describe the nonlinear ship, and the controller is of restricted structure. The adaptive controller is introduced by detailing the underlying polynomial-based optimal control theory for a single model, before approximating the cost integral so that a restricted-structure solution across multiple models may be found. Augmenting a set of representative linear models with an identified ship model in the online optimisation completes the adaptive control algorithm. The properties of the restricted-structure solution are presented with a single model example, before applying the adaptive controller to the surge velocity loop of the ship problem in Chapter 2. *Successful simulation results are presented and analysed at the end of the Chapter, and related problems of algorithm convergence and simultaneous stabilisation are discussed.*

4.1 Introduction

The multivariable PID controller tuning idea for DP developed in Chapter 2 is limited by the fact that each controller design applies at only one operating point.

This fact presents difficulties from the control theoretic perspective. A controller which is well tuned or optimal at one linearised operating point may have poor performance or even be unstable at another.

The problem of poor performance may be addressed by analysing the system nonlinearities and producing a fixed nonlinear control law, or by varying linear controller parameters, known as adaptation. The theory of nonlinear control is less well developed and understood than that for linear systems, so by assuming that the plant is simply shifting from one linear operating point to another, the more developed linear theory may be used. The problem of instability is understood in terms of robustness to unmodelled dynamics, and has been studied using H_∞ control theory, introduced by Zames (1981). H_∞ control design tends to produce poorer performance, as the controller is detuned in order to avoid instability when plant gain varies unpredictably. The two problems are related by the common thread of uncertain model variation. In the past, attempts have been made to address the difficulties by the use of multiple models and adaptive control.

Multiple-model solutions date back to the 1970s and significant early works include Lainiotis (1976) and Athans et al. (1977). In the work of Lainiotis (1976), multiple Kalman filters are employed to improve the accuracy of the state estimate in control problems. Athans et al. (1977) control an F-8C aircraft using the MMAC (Multiple-Model Adaptive Control) method. This involves finding models of 16 different equilibrium flight conditions and carrying out a complete LQG (Linear Quadratic Gaussian) design, see Burl (1999) for example, for each one. During operation, the actual control signal for the aircraft is applied to each of the 16 Kalman filters created during the LQG design. Each LQG controller then produces an optimal control signal for the flight condition to which it corresponds. Additionally, each Kalman filter generates a residual vector that is used to produce conditional probabilities that the model in question is the true one. The control signal which actually drives the aircraft is then computed as the weighted sum of the 16 LQG ‘optimal‘ control signals, where the weighting is the probability for the corresponding model. Simulation results show that the MMAC approach is potentially applicable to adaptive aircraft control, although the authors note that careful tuning of the Kalman filters is important in order that identification of the "true" model and subsequent control are not adversely affected. The authors also point out that a persistent excitation signal is necessary for identification and that, overall, a theoretical basis for the approach is lacking, due to the highly nonlinear nature of the plant and control scheme. In this regard, it is stated that extensive simulation results must be relied on to judge the performance of the algorithm, in the absence of a suitable analytical framework.

During the 1990s, K. S. Narendra has been particularly active in the field of multiple models and, in Narendra et al. (1995), gives an accessible account of a general approach, including switching and tuning. The use of multiple models is justified by the fact that the input-output characteristics of a system will generally change when operating in different environments. A single model may change too slowly during identification, leading to transient errors, or be unable to adequately represent the plant across the range of operating conditions. Thus, a control scheme similar to Athans et al. (1977) is proposed. In the linear case, the plant is represented by a transfer function with time-varying parameters. A finite set of possible parameter vectors is selected, then a controller is designed for each vector such that tracking error is smaller than a certain constant. These parameter vectors are also used to generate output estimates at each time step and some measure of identification error is computed to determine the model which best represents the plant. The controller pre-designed for the best model is then applied to the real plant, but an alternative is to use some sort of learning control if the identification error is particularly large and the plant is operating in a previously unencountered region. Adaptation then occurs until the steady-state error is smaller than some constant. This new model and controller is stored in memory by the control algorithm for later use. Issues such as controller structure, the model switching scheme and the use of fixed or adaptive models are then dealt with, and it is noted that fixed models produce a more computationally efficient algorithm, but an adaptive model gives desired steady-state accuracy. Narendra suggests using one adaptive model along with several fixed models to yield an efficient control architecture.

In contrast to Athans et al. (1977), there is more of a theoretical underpinning to the work, and stability is proved in Narendra and Balakrishnan (1994a) and Narendra and Balakrishnan (1994b) in the linear continuous time case and Narendra and Xiang (2000) in the linear discrete time case. Novel switching and tuning schemes are explored in Narendra and Balakrishnan (1997) and the stochastic case is described in Narendra and Driollet (2001). Ippoliti and Longhi (2004) take these ideas and apply them to several examples of real systems, using minimum variance as the control design algorithm, and employing $n + 1$ linear models, with n fixed and 1 identified model. Proofs of stability are given and the examples demonstrate reduced tracking error for the multiple-model case compared to an adaptive-only case of minimum variance control. The concept of multiple models has also been developed for applications in system identification, controller design using fuzzy techniques and even control of a chaotic system. These ideas are all presented in Johansen and Foss (Eds.) (2001).

Adaptive controllers date back to the 1950s, with Kalman (1958) and Gregory (Ed.) (1959), for example. The field is quite mature now, with Åström and Wittenmark (1995) providing a fairly comprehensive introduction dealing with four main types of adaptive systems: Gain scheduling, model-reference adaptive control, self-tuning regulators and dual control. Gain scheduling is a scheme by which the controller parameters are varied according to changes in a particular operating condition. In practice, this takes the form of a look-up table that specifies gains corresponding to operating point. In model-reference adaptive control, desired closed-loop behaviour is determined by a reference model. The error between plant and model output forms the input to an adjustment mechanism that alters the controller parameters in order to minimise the square of the error. These first two strategies alter the control gains directly without recourse to the plant parameters. A self-tuning regulator, however, uses parameter estimates within a design algorithm to update the control. Strictly speaking, self-tuning refers only to plants with unknown constant parameters where an initial "tuning-in" mechanism is required, but in the present discussion the wider sense of continual tuning to track parameter variation is used. The certainty equivalence principle is invoked here, where it is assumed that the estimates are perfectly equal to the true plant. This may often be a reasonable assumption, but the final adaptive scheme, dual control, incorporates uncertainty into the design. Unfortunately, the approach is complicated and does not yield practicable results without approximations.

The multiple-model adaptive schemes discussed earlier essentially fall into the self-tuning regulator set of adaptive controllers. The approach to multiple-model adaptive control in this Chapter is also within the self-tuning regulator class, but differs from previous approaches in that a controller is not designed for each fixed model in advance but is incorporated into an online optimisation. The strategy followed is to minimize an LQG criterion across n+1 separate linear models, where n are fixed and 1 is identified. This approach is novel in itself, but additionally the controller is restricted to a desired low-order structure. A simple analytic solution cannot be obtained, as in the case where the controller structure is unconstrained, but a relatively straightforward optimisation problem can be established which provides the desired solution. The algorithm depends upon recent advances in restricted-structure optimal control, Grimble (1999a), and in control of multiple-model systems, Grimble (1999b), first combined in Grimble (2000). The aim is to provide a degree of robustness to parameter uncertainty by optimising across a set of possible models, whilst attempting to produce good performance by incorporating an estimate of the current system parameters. In this way, the problems highlighted at the beginning of the Chapter, of poor performance and possible instability due to changing system parameters, are addressed concurrently.

The reasons for creating restricted-structure controllers were highlighted in a Bode prize lecture, transcripted in Anderson (1993). Practising engineers want controllers that are low complexity, discrete-time and free of numerical problems. Unfortunately, modern optimal design methods such as LQG and H_∞ give controllers with order equal to or greater than the plant. Hence, complex plant descriptions inevitably yield optimal but undesirably complex control designs. There are three routes to a low order controller in this situation. Firstly, a high order design may be produced from a high order model, followed by controller reduction. Secondly, the model may be reduced, see Gugercin and Antoulas (2000), before carrying out a design on the outcome of this reduction. Finally, there are direct design methods, taking the high order plant and generating a low order controller without an intermediate step, see Hyland and Bernstein (1984). The second route tends to produce a poor controller, as the approximation step occurs first in the process and any errors are carried through to the design. The third route is problematic because it is hard to understand the procedure intuitively and gain convergent results. The first route seems to hold most promise, as the approximation step occurs last, after an optimal design is known, and the restricted-structure algorithm in this Chapter follows this route.

Early work in this area focussed on reducing the controller order without paying attention to the structure, PID or lead-lag for example. Anderson and Liu (1989) emphasise that controller reduction must take into account closed-loop behaviour, as an open-loop procedure does not necessarily respect stability, bandwidth and robustness. Hence, a frequency-weighted model problem is presented and solved using truncation of an internally balanced realisation, Hankel norm approximation and q-cover approximation. Additionally, the controller solution of an LQG problem is given in a transfer function fractional representation, before the solution schemes above are applied and compared. The authors conclude that different methods are superior dependent upon control objectives, and whether the controller is open-loop unstable or not. The structure of the controller is not discussed.

Hjalmarsson et al. (1994) present a numerical optimisation approach to an LQG type problem, where the plant and controller are stated in discrete-time transfer function form, and the desired response is given by a reference model. The cost is given as a function of the controller parameters, then a gradient descent procedure is used to find the minimum, with stochastic approximation to the cost gradient. This gradient is computed from measured closed-loop data and so the approach is not model-based. A proof of convergence is given and an example with a specified extended-PID controller structure is examined. The restricted-complexity controller converges to a good approximation of the desired response after 14 iterations. A drawback of this method is that it involves three separate experiments in order to gather the data, thus the optimisation cannot be carried out online.

Landau, Karimi and Hjalmarsson (Eds.) (2003) deals with solutions to an active suspension system benchmark problem using controllers of restricted complexity, of which Chable et al. (2003) is the only paper to deal with reduction of controller order from a full-order design. The approach taken is known as robust modal control and involves the solution of a QP (Quadratic Programming) problem, where the quadratic to be minimised is the frequency response error between the full and reduced order controllers. The equality constraint is used to specify the structure of the reduced controller and the inequality constraint places restrictions on the gain at particular frequencies. The full-order design for the benchmark problem is carried out using the mixed H_∞ method, producing a $25th$ order controller. The robust modal algorithm produces a $4th$ order controller, providing good results in comparison with the H_∞ design. This approach is successful, but does not take closed-loop behaviour into account as part of the algorithm. Thus, this method may not be wholly reliable, for the reasons given by Anderson and Liu (1989).

In this Chapter, a novel contribution is made to the field of adaptive control via the application of a multiple-model restricted-structure algorithm to the ship DP problem. The theory behind this approach is now developed, beginning with a statement of the LQG problem.

4.2 LQG problem formulation

State-space form

This problem was first posed in state-space, following developments by Kalman (1960a) and Kalman (1960b). The discrete-time problem is as follows:

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{4.1}$$

$$y(t) = Cx(t) + v(t) \tag{4.2}$$

where state $x(t) \in \mathbb{R}^{n\times 1}$, input $u(t) \in \mathbb{R}$, process noise $w(t) \in \mathbb{R}^{n\times 1}$, output $y(t) \in \mathbb{R}$, and measurement noise $v(t) \in \mathbb{R}$. The constant matrices are $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$ and $C \in \mathbb{R}^{1\times n}$. The noise signals are assumed to be white, zero mean, Gaussian and uncorrelated, with covariance:

$$E[ww^T] = Q_w, E[vv^T] = R_v \tag{4.3}$$

The standard discrete-time steady-state LQG stochastic control problem involves finding the minimum of the time-domain performance index:

$$J = \lim_{t\to\infty} E\left[\hat{x}^T(t)Q\hat{x}(t) + u^T(t)Ru(t)\right] \tag{4.4}$$

where $\lim_{t\to\infty} E\left[.\right]$ denotes the steady-state expected value operation and $\hat{x}(t)$ is the optimal state estimate from a Kalman filter. Q is a symmetric, positive semi-definite real matrix and R is a symmetric, positive definite real matrix. To minimise the index, the algebraic Riccati equation must be solved to find the optimal state feedback gain and the filter algebraic Riccati equation must be solved to find the optimal estimator gain.

Polynomial form

The related polynomial problem was first solved by Shaked (1976) and is detailed along with related developments in Grimble and Johnson (1994). The equations for the system shown in Figure 4.1 take the form:

$$\begin{aligned} y(t) &= Wu(t) + d(t) = A^{-1}Bu(t) + d(t) && (4.5)\\ d(t) &= W_d\xi(t) = A^{-1}C_d\xi(t) && (4.6)\\ r(t) &= W_r\zeta(t) = A^{-1}E\zeta(t) && (4.7) \end{aligned}$$

where W is the plant, W_d is the disturbance model, W_r is the reference model and all signals are real scalars. The external white noise sources, $\xi(t)$ and $\zeta(t)$, are zero mean and mutually statistically independent. The covariances for these signals are without loss of generality taken to be unity. The plant is assumed to be free of unstable hidden modes and the reference, W_r , and disturbance, W_d, subsystems are assumed to be asymptotically stable.

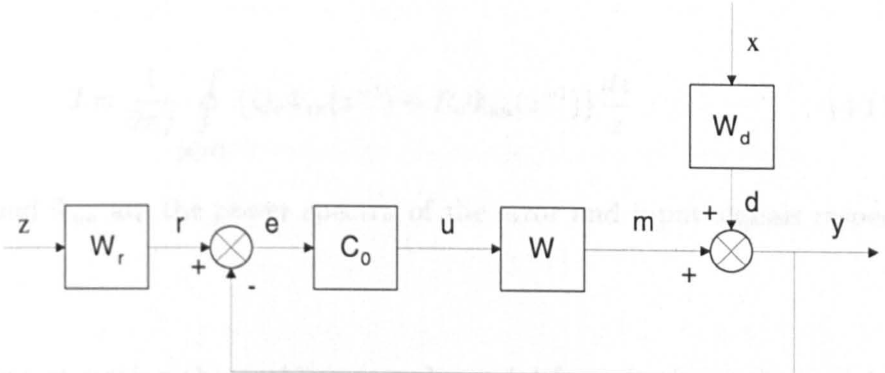

Figure 4.1: Closed-loop system

Tracking error and control input are given by:

$$
\begin{aligned}
e(t) &= r(t) - y(t) && (4.8)\\
u(t) &= C_0 e(t) = C_{0d}^{-1} C_{0n} e(t) && (4.9)
\end{aligned}
$$

A, B, C_d, E, C_{0d} and C_{0n} are polynomials in z^{-1}, which is the unit delay operator or complex number, e^{-sT}, dependent on context. The various polynomials are not necessarily coprime but the system elements are assumed to be free of unstable hidden modes. For notational simplicity, the z^{-1} arguments are often omitted.

The steady-state LQG stochastic control time-domain performance index for this system is:

$$
\begin{aligned}
J &= \lim_{t\to\infty} E\left[(q_c e(t))^2 + (r_c u(t))^2\right]\\
&= \lim_{t\to\infty} E\left[Q_c e^2(t) + R_c u^2(t)\right] && (4.10)
\end{aligned}
$$

where $Q_c, R_c > 0$. Using Parseval's theorem, equation (4.10) may be equivalently stated in the z-domain as:

$$J = \frac{1}{2\pi j} \oint\limits_{|z|=1} \{Q_c\Phi_{ee}(z^{-1}) + R_c\Phi_{uu}(z^{-1})\}\frac{dz}{z} \tag{4.11}$$

where Φ_{ee} and Φ_{uu} are the power spectra of the error and input signals respectively.

An advantage of stating the problem in polynomial form is that it is straightforward to use dynamic instead of constant weights. Let q_c and r_c in equation (4.10) be time-dependent functions convolved with $e(t)$ and $u(t)$ respectively, see Shaked (1976) and Grimble and Johnson (1988):

$$J \;=\; \lim_{t\to\infty} E\left[(q_c(t) * e(t))^2 + (r_c(t) * u(t))^2\right] \tag{4.12}$$

In the z-domain:

$$\begin{aligned} J &= \frac{1}{2\pi j} \oint\limits_{|z|=1} \{q_c(z^{-1})\Phi_{ee}(z^{-1})q_c^*(z^{-1}) + r_c(z^{-1})\Phi_{uu}(z^{-1})r_c^*(z^{-1})\}\frac{dz}{z} \\ &= \frac{1}{2\pi j} \oint\limits_{|z|=1} \{Q_c(z^{-1})\Phi_{ee}(z^{-1}) + R_c(z^{-1})\Phi_{uu}(z^{-1})\}\frac{dz}{z} \end{aligned} \tag{4.13}$$

where $Q_c(z^{-1}) = \dfrac{Q_n}{A_q^*(z^{-1})A_q(z^{-1})}$, $R_c(z^{-1}) = \dfrac{R_n}{A_r^*(z^{-1})A_r(z^{-1})}$, and $Q_n = Q_n^*$, $R_n = R_n^*$. Here, we define $x^*(z^{-1}) = x^T(z)$.

In this way it is possible to exert a finer influence over the costed signals. The frequency domain problem can be viewed as weighting the power spectra $\Phi_{ee}(z^{-1})$ and $\Phi_{uu}(z^{-1})$ with the filters $Q_c(z^{-1})$ and $R_c(z^{-1})$ respectively.

4.3 LQG problem solution

In order to minimise equation (4.13), expressions for the spectra are required. From Figure 4.1, it is straightforward to obtain the following relationship:

$$y(z^{-1}) = WMr(z^{-1}) + Sd(z^{-1}) \tag{4.14}$$

where the control sensitivity function is $M = C_0(1 + WC_0)^{-1}$, and $S = (1 + WC_0)^{-1}$ is the sensitivity function.

From Grimble and Johnson (1988), $\Phi_{yy}(z^{-1}) = W(z^{-1})\Phi_{uu}(z^{-1})W^*(z^{-1})$ when $y(z^{-1}) = W(z^{-1})u(z^{-1})$. Hence:

$$\begin{aligned}\Phi_{ee}(z^{-1}) &= (1 - WM)\Phi_{rr}(z^{-1})(1 - WM)^* + S\Phi_{dd}(z^{-1})S^* \\ &= (1 - WM)\Phi_{rr}(z^{-1})(1 - WM)^* + (1 - WM)\Phi_{dd}(z^{-1})(1 - WM)^* \\ &= (1 - WM)\Phi_{ff}(z^{-1})(1 - WM)^* \end{aligned} \tag{4.15}$$

and

$$\begin{aligned}\Phi_{uu}(z^{-1}) &= M\Phi_{rr}(z^{-1})M^* + M\Phi_{dd}(z^{-1})M^* \\ &= M\Phi_{ff}(z^{-1})M^* \end{aligned} \tag{4.16}$$

where $\Phi_{ff}(z^{-1}) = \Phi_{rr}(z^{-1}) + \Phi_{dd}(z^{-1})$ and r and d are uncorrelated.

Dropping the dependence of signals on z^{-1} where obvious, the minimisation of the cost function proceeds as follows:

$$\begin{aligned}
J &= \frac{1}{2\pi j}\oint\limits_{|z|=1}\{Q_c(z^{-1})\Phi_{ee}(z^{-1}) + R_c(z^{-1})\Phi_{uu}(z^{-1})\}\frac{dz}{z} \\
&= \frac{1}{2\pi j}\oint\limits_{|z|=1}\{Q_c(1-WM)\Phi_{ff}(z^{-1})(1-WM)^* + R_c(z^{-1})M\Phi_{ff}(z^{-1})M^*\}\frac{dz}{z} \\
&= \frac{1}{2\pi j}\oint\limits_{|z|=1}\{(W^*Q_cW + R_c)M\Phi_{ff}M^* - Q_c\Phi_{ff}(WM - W^*M^* + 1)\}\frac{dz}{z} \\
&= \frac{1}{2\pi j}\oint\limits_{|z|=1}\{(Y_cMY_f)(Y_cMY_f)^* - Q_c\Phi_{ff}(WM\frac{Y_cY_f}{Y_cY_f} - W^*M^*\frac{Y_c^*Y_f^*}{Y_c^*Y_f^*} + 1)\}\frac{dz}{z}
\end{aligned} \tag{4.17}$$

where $W^*Q_cW + R_c = Y_c^*Y_c = \dfrac{D_c^*D_c}{A_c^*A_c}$, $A_c = AA_qA_r$, $\Phi_{ff} = Y_f^*Y_f = \dfrac{D_f^*D_f}{A^*A}$.

Later on, it is necessary to solve:

$$D_c^*D_c = B^*A_r^*Q_nA_rB + A^*A_q^*R_nA_qA \tag{4.18}$$

$$D_f^*D_f = EE^* + C_d^*C_d \tag{4.19}$$

for D_c and D_f. This is known as spectral factorisation.

Using a completing the squares argument on equation (4.17):

$$J = \frac{1}{2\pi j}\oint\limits_{|z|=1}\{(Y_cMY_f - \frac{Q_cW^*\Phi_{ff}}{Y_c^*Y_f^*})(Y_cMY_f - \frac{Q_c^*W^*\Phi_{ff}^*}{Y_c^*Y_f^*})^* + \Phi_0\}\frac{dz}{z} \tag{4.20}$$

where $\Phi_0 = -\dfrac{W^*WQ_c^*Q_c\Phi_{ff}^*\Phi_{ff}}{Y_c^*Y_cY_f^*Y_f} + Q_c\Phi_{ff}$.

The cost function is now in a form such that the terms in Φ_0 are independent of the feedback controller, C_0. The remaining difficulty is that the product in J must be split into guaranteed stable and unstable components. This splitting can be achieved using two Diophantine equations, but first the terms in the product must be expanded into desired form:

$$Y_c M Y_f = \frac{D_c}{A_c}\frac{C_0}{1+WC_0}\frac{D_f}{A} = \frac{D_c D_f C_{0n}}{A_c(AC_{0d}+BC_{0n})} \tag{4.21}$$

$$\frac{Q_c W^* \Phi_{ff}}{Y_c^* Y_f^*} = \frac{B^*}{A^*}\frac{Q_n}{A_q^* A_q}\frac{Y_f}{Y_c^*} = \frac{B^* Q_n D_f A_c^*}{A^* A_q^* A_q A D_c^*} = \frac{B^* Q_n D_f A^* A_q^* A_r^*}{A^* A_q^* A_q A D_c^*} = \frac{B^* Q_n D_f A_r^*}{D_c^* A A_q} \tag{4.22}$$

Thus, $Y_c M Y_f - \frac{Q_c W^* \Phi_{ff}}{Y_c^* Y_f^*}$ is equivalent to:

$$\begin{aligned} & \frac{D_c^* D_c D_f C_{0n}}{D_c^* A_c(AC_{0d}+BC_{0n})} - \frac{B^* Q_n D_f A_r^*}{D_c^* A A_q} \\ = & \frac{(B^* A_r^* Q_n A_r B + A^* A_q^* R_n A_q A) D_f C_{0n}}{D_c^* A_c(AC_{0d}+BC_{0n})} - \frac{B^* A_r^* Q_n D_f}{D_c^* A A_q} \\ = & \frac{(B^* A_r^* Q_n D_f) B A_r C_{0n} + (A^* A_q^* R_n D_f) A A_q C_{0n}}{D_c^* A_c(AC_{0d}+BC_{0n})} - \frac{B^* A_r^* Q_n D_f}{D_c^* A A_q} \end{aligned} \tag{4.23}$$

Substituting the Diophantine equations:

$$D_c^* G_0 z^{-g} + F_0 A A_q = B^* A_r^* Q_n D_f z^{-g} \tag{4.24}$$

$$D_c^* H_0 z^{-g} - F_0 B A_r = A^* A_q^* R_n D_f z^{-g} \tag{4.25}$$

it is then trivial to show that:

$$Y_c M Y_f - \frac{Q_c W^* \Phi_{ff}}{Y_c^* Y_f^*} = \frac{C_{0n} H_0 A_q - C_{0d} G_0 A_r}{A_w(AC_{0d}+BC_{0n})} - \frac{F_0 z^g}{D_c^*} = T_1^+ + T_1^- \tag{4.26}$$

where the first term is strictly stable and the second term is strictly unstable.

Now:

$$J = \frac{1}{2\pi j}\oint_{|z|=1} \{(T_1^+ + T_1^-)(T_1^+ + T_1^-)^* + \Phi_0\}\frac{dz}{z} \tag{4.27}$$

The T_1^- and Φ_0 terms are independent of the controller, C_0, hence the optimal control problem reduces to finding C_0 such that

$$J_{csl} = \frac{1}{2\pi j} \oint_{|z|=1} \{T_1^+ T_1^{+*}\} \frac{dz}{z} \tag{4.28}$$

is minimised. In the optimal case, simply set T_1^+ to zero. That is, let:

$$C_0 = \frac{C_{0n}}{C_{0d}} = \frac{G_0 A_r}{H_0 A_q} \tag{4.29}$$

To summarise, solving the optimal control problem involves finding the solution to two spectral factors and two Diophantine equations:

$$\begin{aligned}
D_c^* D_c &= B^* A_r^* Q_n A_r B + A^* A_q^* R_n A_q A \\
D_f^* D_f &= EE^* + C_d^* C_d \\
D_c^* G_0 z^{-g} + F_0 A A_q &= B^* A_r^* Q_n D_f z^{-g} \\
D_c^* H_0 z^{-g} - F_0 B A_r &= A^* A_q^* R_n D_f z^{-g}
\end{aligned}$$

where the unknowns are D_c, D_f, F_0, G_0 and H_0. There are, in fact, an infinite number of spectral factors, but the solutions D_c and D_f in the form $d_0 + d_1 z^{-1} + d_2 z^{-2} + ...$ are used. The solution to the Diophantine equations is unique provided that F_0 is of smallest degree, n_{f0}, and the shift $g > n_{f0}$. The procedure is detailed in Kucera (1979) and is implemented in numerous mathematical software packages. The package adopted throughout this thesis is Matlab.

4.4 Numerical algorithm for restricted-structure solution

By setting $T_1^+ = 0$, the order of the optimal controller is determined by the order of the various system polynomials and weightings, and the structure is simply a rational function. Constraining C_0 to be of a certain order and structure, PID for example, means that T_1^+ cannot simply be set to zero, and the solution will be sub-optimal. To minimise J in this case requires a method for minimising T_1^+ with respect to C_{0n} and C_{0d}, as follows.

It is clear that T_1^+ can be written in the form:

$$T_1^+ = \frac{C_{0n}L_1 - C_{0d}L_2}{C_{0n}L_3 + C_{0d}L_4} \tag{4.30}$$

where $L_1 = H_0A_q$, $L_2 = G_0A_r$, $L_3 = A_wB$, and $L_4 = A_wA$. Now assume that C_0 has a PI structure, although this is by no means the only viable structure, so that:

$$C_{0n} = K_p(1 - z^{-1}) + K_i \tag{4.31}$$

and

$$C_{0d} = (1 - z^{-1}) \tag{4.32}$$

C_{0n} includes the unknown gains to be optimised and C_{0d} is known. However, T_1^+ will be nonlinear in K_p and K_i, rendering equation (4.28) particularly difficult to minimise directly. Nevertheless, if the values of K_p and K_i are assumed known in the denominator of T_1^+, an iterative solution is possible, as will be shown later.

To proceed, T_1^+ must be split into real and imaginary components, where z^{-1} is now interpreted as e^{-sT}. Let the superscripts r and i denote the real and imaginary parts of a complex function, so that:

$$C_{0n} = C_{0n}^r + jC_{0n}^i \; and \; C_{0d} = C_{0d}^r + jC_{0d}^i \tag{4.33}$$

The numerator term may be split into components, through comparison with (4.31):

$$C_{0n} = K_p \alpha_0(z^{-1}) + K_i \alpha_1(z^{-1}) = C_{0n}^r + jC_{0n}^i \tag{4.34}$$

where:

$$\alpha_0 = (1 - z^{-1}) \ , \ \alpha_1 = 1 \tag{4.35}$$

and hence

$$C_{0n}^r = K_p \alpha_0^r + K_i \alpha_1^r \ , \ C_{0n}^i = K_p \alpha_0^i + K_i \alpha_1^i \tag{4.36}$$

The denominator in T_1^+ is assumed known, thus let:

$$T_1^+ = C_{0n} L_{n1} - L_{n2} \tag{4.37}$$

where L_{n1} and L_{n2} are therefore assumed known and defined as:

$$L_{n1} = \frac{L_1}{C_{0n} L_3 + C_{0d} L_4} \ , \ L_{n2} = \frac{C_{0d} L_2}{C_{0n} L_3 + C_{0d} L_4} \tag{4.38}$$

Substituting from (4.33):

$$\begin{aligned} T_1^+ &= (C_{0n}^r + jC_{0n}^i)(L_{n1}^r + jL_{n1}^i) - (L_{n2}^r + jL_{n2}^i) \\ &= C_{0n}^r L_{n1}^r - C_{0n}^i L_{n1}^i - L_{n2}^r + j(C_{0n}^i L_{n1}^r + C_{0n}^r L_{n1}^i - L_{n2}^i) \end{aligned} \tag{4.39}$$

and after substitution from (4.36), obtain:

$$\begin{aligned} T_1^+ &= K_p \alpha_0^r L_{n1}^r + K_i \alpha_1^r L_{n1}^r - K_p \alpha_0^i L_{n1}^i - K_i \alpha_1^i L_{n1}^i - L_{n2}^r \\ &\quad + j(K_p \alpha_0^i L_{n1}^r + K_i \alpha_1^i L_{n1}^r + K_p \alpha_0^r L_{n1}^i + K_i \alpha_1^r L_{n1}^i - L_{n2}^i) \end{aligned} \tag{4.40}$$

A vector form of the above equation that will enable the optimisation to be performed for the single-model case, with respect to the unknown gains, $x = (K_p, K_i)$, is:

$$\begin{bmatrix} T_1^{+r} \\ T_1^{+i} \end{bmatrix} = F \begin{bmatrix} K_p \\ K_i \end{bmatrix} - L = Fx - L \tag{4.41}$$

where

$$F = \begin{bmatrix} (\alpha_0^r L_{n1}^r - \alpha_0^i L_{n1}^i) & (\alpha_1^r L_{n1}^r - \alpha_1^i L_{n1}^i) \\ (\alpha_0^i L_{n1}^r + \alpha_0^r L_{n1}^i) & (\alpha_1^i L_{n1}^r + \alpha_1^r L_{n1}^i) \end{bmatrix}, \; L = \begin{bmatrix} L_{n2}^r \\ L_{n2}^i \end{bmatrix} \tag{4.42}$$

Setting $z = e^{j\omega T}$, where ω is angular frequency and T is sample time, note the result $(T_1^{+r})^2 + (T_1^{+i})^2 = T_1^+(e^{-j\omega T})T_1^+(e^{j\omega T})$. Now recall that we wish to minimise equation (4.28), which can be restated as follows:

$$\begin{aligned} J_{csl} &= \frac{1}{2\pi j} \oint\limits_{|z|=1} T_1^+ T_1^{+*} \frac{dz}{z} \\ &= \frac{T}{2\pi} \int\limits_0^{2\pi/T} T_1^+(e^{-j\omega T})T_1^+(e^{j\omega T}) d\omega \\ &= \frac{T}{2\pi} \int\limits_0^{2\pi/T} (T_1^{+r})^2 + (T_1^{+i})^2 d\omega \\ &= \frac{T}{2\pi} \int\limits_0^{2\pi/T} (Fx - L)^T (Fx - L) d\omega \end{aligned} \tag{4.43}$$

This cost function can be optimised directly, but a simple iterative solution can be obtained if the integral is approximated by a summation with a sufficient number of frequency points, $\{\omega_1, \ldots, \omega_k, \ldots, \omega_N\}$. ω_N is the Nyquist frequency. The optimisation is then carried out by minimising the summation, in the manner of Yukitomo et al. (1998), where the new cost is:

$$\begin{aligned} J_0 &= \sum_{k=1}^{N} (Fx - L)^T (Fx - L) \\ &= (b - Ax)^T (b - Ax) \end{aligned} \tag{4.44}$$

and

$$A = \begin{bmatrix} F(\omega_1) \\ \vdots \\ F(\omega_N) \end{bmatrix}, b = \begin{bmatrix} L(\omega_1) \\ \vdots \\ L(\omega_N) \end{bmatrix}, x = \begin{bmatrix} K_p \\ K_i \end{bmatrix} \tag{4.45}$$

The multiple-model solution can be found by minimising the weighted sum of the costs of each individual system model, J_{0j} , as below:

$$\begin{aligned} J_0 &= \sum_{j=1}^{n} p_j J_{0j} \\ &= \sum_{j=1}^{n} p_j (b_j - A_j x)^T (b_j - A_j x) \\ &= (\underline{b} - \underline{A}x)^T P (\underline{b} - \underline{A}x) \end{aligned} \tag{4.46}$$

where

$$\underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \underline{A} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}, P = diag\{p_1, \ldots, p_n\} \tag{4.47}$$

Assuming the matrix $\underline{A}^T P \underline{A}$ is not singular, the least squares optimal solution, see Ljung (1999) for example, follows as:

$$x = (\underline{A}^T P \underline{A})^{-1} \underline{A}^T P \underline{b} \tag{4.48}$$

Of course, as the assumption was made that the solution x was already known in the denominator of T_1^+, this is a case where the method of successive approximation, as in Luenberger (1969), can be used. This involves a transformation T such that $x_{n+1} = T(x_n)$. Under appropriate conditions, the sequence $\{x_n\}$ converges to a solution of the original equation. Since this optimisation problem is non-linear there may be not be a unique minimum. The following algorithm, with a system identification algorithm incorporated, does always appear to converge to an optimal solution in many industrial examples and can be used to compute the restricted-structure LQG adaptive controller.

Algorithm 4.4.1 (Adaptive restricted-structure control algorithm) .

1. *Define* N *(number of frequency points),* $\omega_1, \ldots, \omega_N$*,* N_f *(number of fixed models), and* P *(model probabilities)*
2. *Initialise* $K_p = K_i = 1$ *(arbitrary choice)*
3. *Define* $\alpha_0(z^{-1})$*,* $\alpha_1(z^{-1})$ *(using (4.35))*
4. *Compute* $C_{0n}(z^{-1}) = K_p\alpha_0(z^{-1}) + K_i\alpha_1(z^{-1})$
5. *Compute* $C_{0d}(z^{-1}) = \alpha_0(z^{-1})$
6. *For* $j = 1$ *to* N_f
   - (a) *Using* A_j*,* B_j*,* C_{dj}*,* E_j*,* Q_c *and* R_c*, solve for the spectral factors* D_{cj} *and* D_{fj}*, and the Diophantine equations for* G_{0j}*,* H_{0j}*, and* F_{0j}*.*
   - (b) *Create* L_{1j}*,* L_{2j}*,* L_{3j}*,* L_{4j}*,* L_{n1j}*, and* L_{n2j}*.*
   - (c) *For all chosen frequencies, calculate* $L^r_{n1j}(\omega T)$*,* $L^i_{n1j}(\omega T)$*,* $L^r_{n2j}(\omega T)$*,* $L^i_{n2j}(\omega T)$*,* $\alpha^r_0(\omega T)$*,* $\alpha^i_0(\omega T)$*,* $\alpha^r_1(\omega T)$*,* $\alpha^i_1(\omega T)$*,* $C^r_{0d}(\omega T)$*,* $C^i_{0d}(\omega T)$*.*
   - (d) *Assemble* $A_j = \begin{bmatrix} F_j(\omega_1) \\ \vdots \\ F_j(\omega_N) \end{bmatrix}$ *and* $b_j = \begin{bmatrix} L_j(\omega_1) \\ \vdots \\ L_j(\omega_N) \end{bmatrix}$
7. *Estimate current* A*,* B*, and* C_d *polynomials using a recursive least squares algorithm.*
8. *Repeat steps 6(a) to (d) for the identified polynomials.*

9. *Stack the $N_f + 1$ A and b matrices to form $\underline{A}$ and $\underline{b}$*

10. *Calculate the restricted-structure controller gains, $x = (\underline{A}^T P \underline{A})^{-1} \underline{A}^T P \underline{b}$*

11. *If the cost is lower than the previous cost, repeat steps 8 to 10 using the new C_{0n}. Otherwise, use previous controller gains to compute the feedback controller $C_{0n}(z^{-1}) = K_p \alpha_0(z^{-1}) + K_i \alpha_1(z^{-1})$ and $C_0(z^{-1}) = C_{0n}(z^{-1})/C_{0d}(z^{-1})$.*

12. *Implement controller in feedback loop and go back to step 7.*

4.5 A remark on numerical difficulties

The controller calculations are performed to double precision using Matlab, or in practice most industrial processors operate to double precision also. This mean that numbers are stored in a quantised fashion within the computer using 64 bits. 52 of the bits are used for the mantissa, 11 bits for the exponent and 1 bit for the sign. The mantissa stores binary numbers of decimal magnitude less than 1, which are then multiplied by two to the power of the exponent to produce a number. The number in the mantissa may range from 2^{-52} up to $1 - 2^{-52}$, but any number less than 2^{-52} is ignored and considered to be zero. The consequence of this is that calculations carried out in Matlab are subject to rounding errors such that:

$$k(1 + \frac{\epsilon}{2}) = \begin{cases} k & , \epsilon \leq 2^{-52} \\ k + k\frac{\epsilon}{2} & , \epsilon > 2^{-52} \end{cases} \tag{4.49}$$

This is not such a problem in general use, but when successive multiplications, divisions or cancellations are required in a calculation, the rounding errors can become significant. This is found to be the case when performing spectral factorisation and solving Diophantine equations for polynomials of "high order". This effect is difficult to predict and can also occur for "low order" polynomials where the coefficients are of very different orders of magnitude. Hence, in the following example, efforts are made to reduce the order of polynomials and make simplifications where this can be done with limited effect on the outcome. Numerical issues are often overlooked during academic exercises, so it is encouraging to see attention drawn to them in a recent special section of the IEEE Control Systems Magazine, Varga et al. (2004).

4.6 Application to dynamic ship positioning

Algorithm 4.4.1 will now be applied to the ship DP problem described in Chapter 2, where the mathematical model is given in equations (2.1) to (2.37) and Figure 2.3 depicts the forces acting on the ship. The theory above may be extended to the multivariable case in a fairly straightforward manner, but due to the numerical inadequacies of double precision arithmetic and the fact that a SISO example is enough to demonstrate the algorithm, only the surge axis of the ship will be considered. Hence, the plant is described by the top left transfer functions of equations (2.48) and (2.51), and the overall control system for the surge axis is depicted in Figure 4.2.

The notation in this Figure does not indicate whether the system is continuous or discrete. The ship problem has been dealt with in continuous time until now, but the nature of Algorithm 4.4.1 requires that the problem be transformed to discrete time. Introducing a zero order hold at the input of g_V and assuming that $X_A + X_W$ is constant between samples, the discrete description for the surge axis becomes:

$$\Delta u(t) = g_V(z^{-1})\Delta X_T(t) + g_d(z^{-1})(X_A(t) + X_W(t)) \tag{4.50}$$

The argument, z^{-1}, denotes the unit-delay operator or the z-transform complex number depending on interpretation. The argument, t, denotes a sampled signal. The relevant closed-loop transfer functions are:

$$\begin{aligned}
\Delta u(t) &= t_V(z^{-1})r_{uabs}(t) + s_V(z^{-1})g_d(z^{-1})(X_A(t) + X_W(t)) && (4.51)\\
\Delta X_T(t) &= u_V(z^{-1})r_{uabs}(t) - v_V(z^{-1})g_d(z^{-1})(X_A(t) + X_W(t)) && (4.52)
\end{aligned}$$

where

$$\begin{aligned}
s_V(z^{-1}) &= (1 + g_V k_V n)^{-1} && (4.53)\\
t_V(z^{-1}) &= (1 + g_V k_V n)^{-1} g_V k_V && (4.54)\\
u_V(z^{-1}) &= (1 + k_V n g_V)^{-1} k_V && (4.55)\\
v_V(z^{-1}) &= (1 + k_V n g_V)^{-1} k_V n && (4.56)
\end{aligned}$$

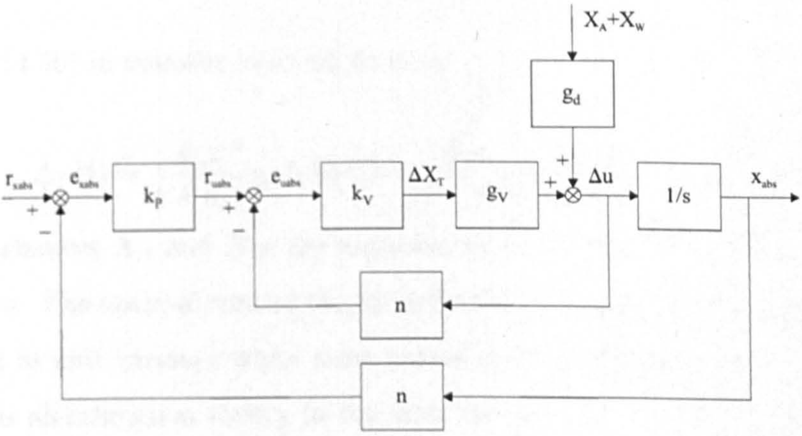

Figure 4.2: Scalar control scheme for surge axis

The adaptive controller will be applied to the velocity loop with the position loop fixed, as the effect of k_V is to desensitise s_V, t_V, u_V and v_V to changes in g_V. The consequence of this effect is that controlling t_V is not an interesting problem for adaptation due to change in system parameters. k_P may be fixed with negligible change in performance.

For motion in the surge direction only, g_V is a first order transfer function, and when sway and yaw motion is included, g_V acquires an extra two poles and zeros.

However, these poles and zeros are not dominant or unstable and are close to cancelling each other, so the transfer function may still be assumed first order. This assumption also helps to avoid numerical problems. Further, the notch filter, n, is ignored for control design as it only affects the frequency response close to $0.6rad/s$. The bandwidth of the velocity loop will be tuned to be in the region of $0.2rad/s$ as in Chapter 2, so n will act at above the crossover frequency where the impact on stability is reduced. Designing the controller for g_V rather than $g_V n$ keeps the plant order low, but additionally prevents the optimal controller from cancelling the effect of n. A $-20dB$ notch in $g_V n$ at $0.6rad/s$ would tend to produce a frequency response peak in k_V at $0.6rad/s$. This is clearly undesirable as the intention is to reduce control action at wave frequencies.

Equation (4.50) in transfer function form is:

$$\Delta u(t) = \frac{b_1 z^{-1}}{1+a_1 z^{-1}} \Delta X_T(t) + \frac{b_1 z^{-1}}{1+a_1 z^{-1}} (X_A(t) + X_W(t)) \qquad (4.57)$$

The disturbances X_A and X_W are unmeasured and cause low frequency oscillations of Δu. The optimal control theory in Section 4.2 demands that disturbances be defined as unit variance white noise passed through a filter, W_d. Furthermore, the system identification theory in the next Section describes the disturbance in terms of current and past estimation errors. For these reasons, equation (4.57) is now restated as the approximation:

$$\Delta u(t) = \frac{b_1 z^{-1}}{1+a_1 z^{-1}} \Delta X_T(t) + \frac{d_1(1+c_1 z^{-1})}{1+a_1 z^{-1}} \xi(t) \qquad (4.58)$$

where $\xi(t)$ is normalised estimation error and d_1 is the amplitude of the estimation error. In the form of equations (4.5) and (4.6) for the optimal control problem:

$$\begin{aligned} \Delta u(t) &= W \Delta X_T(t) + d(t) = A^{-1} B \Delta X_T(t) + d(t) \\ d(t) &= W_d \xi(t) = A^{-1} C_d \xi(t) \end{aligned}$$

where $A = (1+a_1z^{-1})(1-z^{-1})$, $B = b_1z^{-1}(1-z^{-1})$ and $C_d = d_1(1+c_1z^{-1})(1-z^{-1})$. Furthermore, if $E = 1 + a_1z^{-1}$ in equation (4.7), then:

$$r(t) = W_r\zeta(t) = \frac{1}{1-z^{-1}}\zeta(t)$$

Defining W_r as an integrator produces a simple and justifiable representation of the reference. Theoretically, $\xi(t)$ and $\zeta(t)$ are assumed to be mutually statistically independent white noise sources of zero mean and unity covariance. In practice, the reference and disturbance signals are mutually statistically independent and both $d(t)$ and $r(t)$ may contain DC and low frequency components. This is consistent with the use of an integrator as the white noise filter.

4.6.1 Single model example

To demonstrate the mechanism of the optimisation and illustrate the properties of full order and restricted-structure controllers, an example for a single realisation of W, W_d and W_r is presented. At an operating point where $u_c = u_0 = 2m/s$:

$$g_V(s) = \frac{7.60 \times 10^{-5}}{s + 0.00750} \tag{4.59}$$

Cascading with a zero order hold and z-transforming with a sample time of 1 second:

$$g_V(z^{-1}) = W(z^{-1}) = \frac{7.57 \times 10^{-5}z^{-1}}{1 - 0.993z^{-1}} \tag{4.60}$$

The sampling period of 1 second gives a sampling frequency of $2\pi = 6.28rad/s$ and therefore a Nyquist frequency of $3.14rad/s$. This is sufficient to prevent aliasing, see Franklin et al. (1998), because the Nyquist frequency is over three times greater than the highest component in the system signal spectrum, which ranges from DC up to $1rad/s$ for high frequency disturbances.

Letting:

$$W_d(z^{-1}) = \frac{0.01(1+z^{-1})}{1-0.993z^{-1}}\ ,\ W_r(z^{-1}) = \frac{1}{1-z^{-1}} \tag{4.61}$$

therefore $A = (1-0.993z^{-1})(1-z^{-1})$, $B = 7.57\times10^{-5}z^{-1}(1-z^{-1})$, $C_d = 0.01(1+z^{-1})(1-z^{-1})$ and $E = (1-0.993z^{-1})$.

Let the error and control weightings be:

$$Q_c = \frac{5000}{(1-z^{-1})(1-z)}\ ,\ R_c = 10^{-3} \tag{4.62}$$

Solving equations (4.18) and (4.19) yields:

$$\begin{aligned} D_c &= 0.0423-0.103z^{-1}+0.0839z^{-2}-0.0235z^{-3} \\ D_f &= 1.00-0.993z^{-1}-0.000100z^{-2} \end{aligned}$$

The Diophantine equations give:

$$\begin{aligned} F_0 &= 0.831-0.853z^{-1}-1.49z^{-2} \\ G_0 &= 35.4+0.269z^{-1}-35.1z^{-2} \\ H_0 &= 0.0423-0.0208z^{-1}-0.00000217z^{-2} \end{aligned}$$

Therefore the controller is:

$$C_{0f} = \frac{C_{0n}}{C_{0d}} = \frac{G_0A_r}{H_0A_q} = \frac{35.4+0.269z^{-1}-35.1z^{-2}}{0.0423-0.0631z^{-1}+0.0208z^{-2}+0.000002172z^{-3}} \tag{4.63}$$

The restricted-structure algorithm is executed for a PI controller where $N = 15$, $\omega_1 = 10^{-4}$, $\omega_N = \pi$ and the frequency points are logarithmically spaced. The iteration begins with $x = (K_p, K_i) = (1,1)$ as the initial guess, and Table 5.1 shows the convergence of the algorithm to a minimum after five iterations, where the cost, J_0, is calculated using the previous value of x in the denominator of L_{n1} and L_{n2}.

Table 4.1: Restricted-structure algorithm iterations

Iteration Number	$x = (K_p\ ,\ K_i)$	Cost, J_0
0	(1 , 1)	-
1	(3236.064 , 24.876)	2.837×10^8
2	(3235.885 , 24.876)	8.523×10^5
3	(3235.887 , 24.874)	8.713×10^5

The cost increases on the third iteration, hence the algorithm ceases. Clearly, the first iteration almost gives the final answer, because only the units and decimal places change thenceforth. In practice, the algorithm can produce greater fluctuations in x from one iteration to the next and take more or less steps to converge dependent on the system and the restricted controller structure. However, given a range of frequency points that cover pertinent features of the full-order controller and a restricted structure which is sufficiently 'rich' in degrees of freedom, the iterative algorithm always seems to converge to a solution.

The restricted-structure controller in this case is:

$$C_{0r} = 3236 + \frac{24.9}{1 - z^{-1}} = \frac{3261 - 3236z^{-1}}{1 - z^{-1}} \qquad (4.64)$$

Figure 4.3 shows a comparison between the Bode plots of C_{0f} and C_{0r}. The magnitude plots are identical up to $10^{-1}rad/s$ and the phase plots are identical up to $10^{-2}rad/s$. Above these frequencies, the full-order plots "roll off" where the restricted-structure plots do not, but the restricted-structure closed-loop performance is not greatly affected.

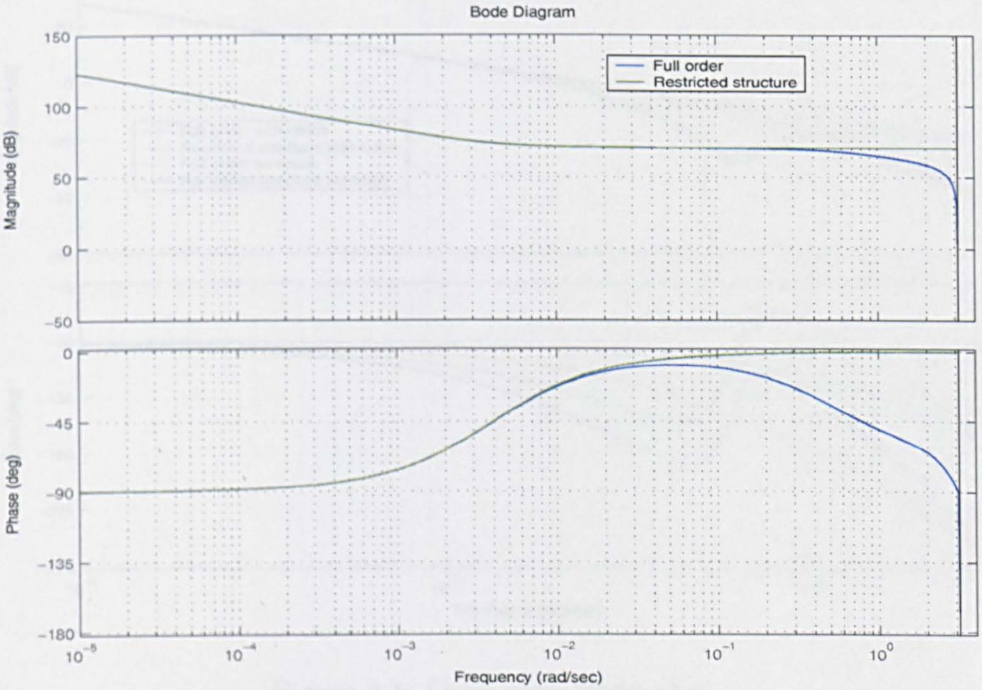

Figure 4.3: Bode plots of full order and PI structure controllers

Recall that the notch filter was neglected in the plant description, as the controller was to be designed so that the cross-over frequency occurred at below the notch frequency. Figure 4.4 shows a comparison between the Bode plots of $C_{0f}(z^{-1})(Wn)(z^{-1})$, $C_{0r}(z^{-1})(Wn)(z^{-1})$, $C_{0f}(z^{-1})W(z^{-1})$ and $C_{0r}(z^{-1})W(z^{-1})$.

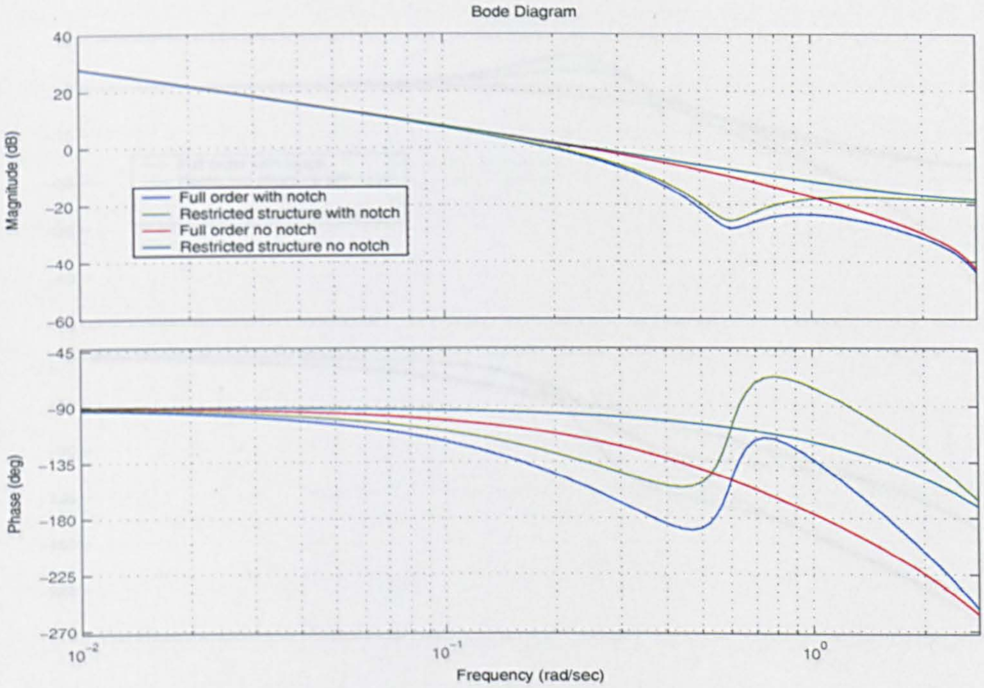

Figure 4.4: Open loop Bode plots

Here, the frequency response is similar for all four plots up to $2 \times 10^{-1} rad/s$, at which point the gain is roughly $0dB$. Figures 4.5, 4.6 and 4.7 show the corresponding comparisons between the Bode plots of $t_V(z^{-1})$ and the step responses of $t_V(z^{-1})$ and $u_V(z^{-1})$.

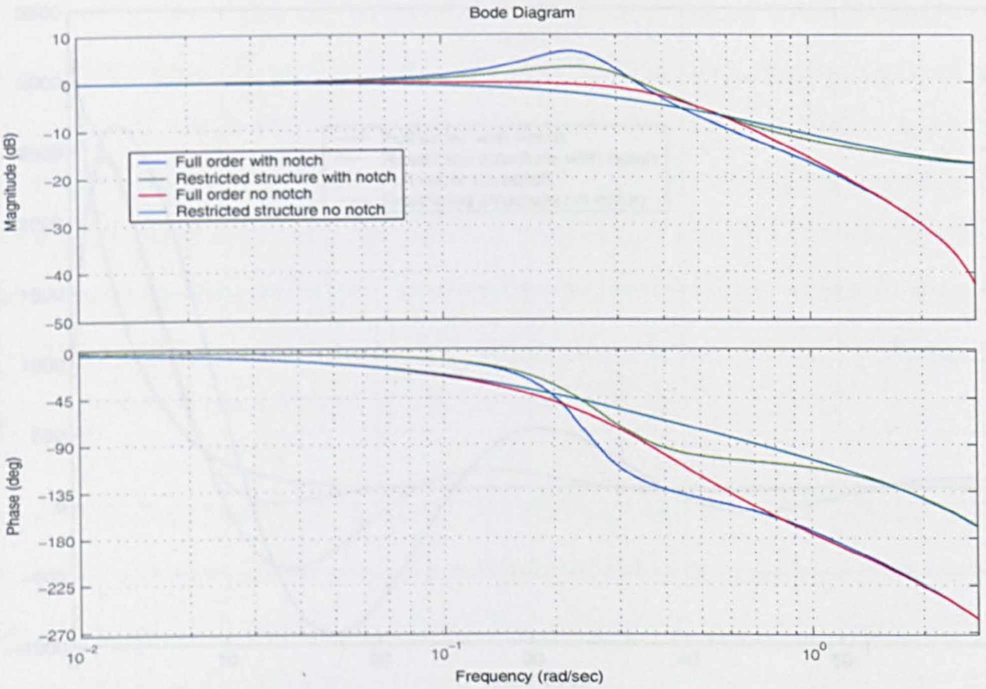

Figure 4.5: Closed-loop Bode plots

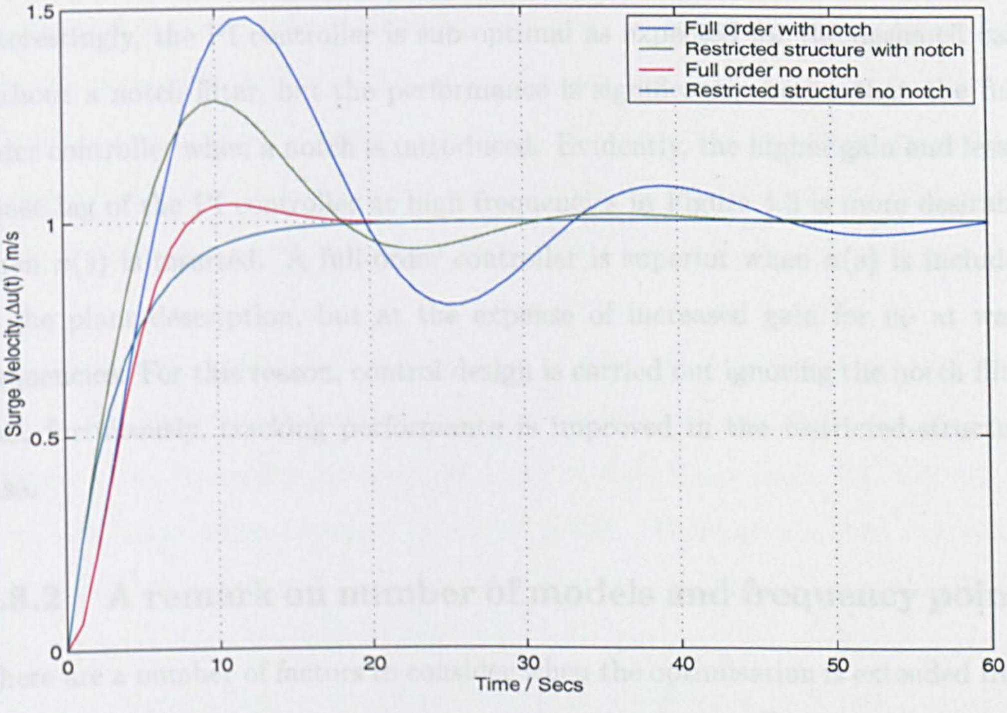

Figure 4.6: Closed-loop t_V step responses

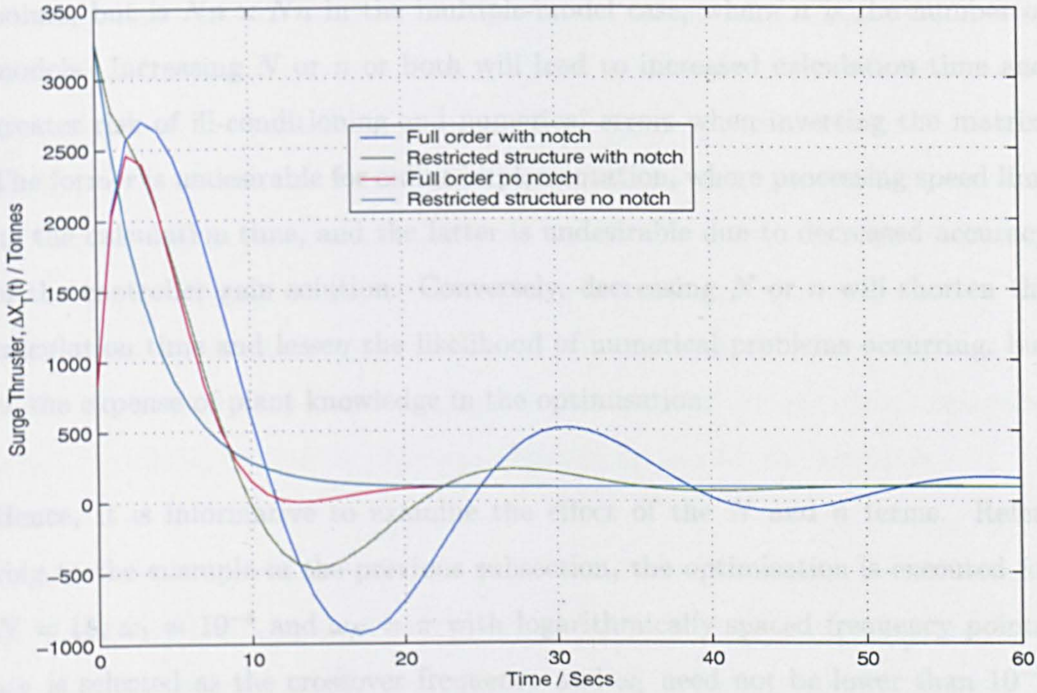

Figure 4.7: Closed-loop u_V step responses

Interestingly, the PI controller is sub-optimal as expected for the designed case without a notch filter, but the performance is significantly better than the full-order controller when a notch is introduced. Evidently, the higher gain and lesser phase lag of the PI controller at high frequencies in Figure 4.3 is more desirable when $n(s)$ is inserted. A full-order controller is superior when $n(s)$ is included in the plant description, but at the expense of increased gain for v_V at wave frequencies. For this reason, control design is carried out ignoring the notch filter and, fortuitously, tracking performance is improved in the restricted-structure case.

4.6.2 A remark on number of models and frequency points

There are a number of factors to consider when the optimisation is extended from the single model to the multiple-model case. Firstly, the $\underline{A}^T P \underline{A}$ matrix in equation (4.48) is $N \times N$ in the single model case, where N is the number of frequency points, but is $Nn \times Nn$ in the multiple-model case, where n is the number of models. Increasing N or n or both will lead to increased calculation time and greater risk of ill-conditioning and numerical errors when inverting the matrix. The former is undesirable for online implementation, where processing speed limits the calculation time, and the latter is undesirable due to decreased accuracy of the controller gain solution. Conversely, decreasing N or n will shorten the calculation time and lessen the likelihood of numerical problems occurring, but at the expense of plant knowledge in the optimisation.

Hence, it is informative to examine the effect of the N and n terms. Referring to the example in the previous subsection, the optimisation is executed for $N = 15$, $\omega_1 = 10^{-4}$ and $\omega_N = \pi$ with logarithmically spaced frequency points. ω_N is selected as the crossover frequency and ω_1 need not be lower than 10^{-4}, as the full-order controller simply approaches infinite gain when zero frequency is approached. A feature of the restricted structure algorithm is selecting this frequency window to suit a particular problem. In the example, the Bode plot content of interest occurs at higher frequencies around the crossover point at $0.2rad/s$. Other examples tend to support the view that a window around the crossover point is generally a good choice, see Grimble (1999b) for corroboration. After choosing a window, comes selection of N.

In the preceding example, it is acceptable to decrease N without loss of fidelity because the frequency domain behaviour of the full-order controller is simple. For a higher order controller, however, N must be sufficiently large that any roll-on or roll-off behaviour is captured by the N points. The situation may not be entirely straightforward as N increases, although the preceding example is affected little, because there is a risk of "overfitting". In that case, the restricted structure controller may attempt to compensate for plant dynamics which have little effect on performance, resulting in idiosyncratic behaviour. Conversely, a restricted structure controller that cannot account for important plant characteristics may in fact be of too low an order and incapable of good performance. To summarise, N can be small for full-order controllers with simple dynamics, i.e. few Bode plot minima and maxima, and a low order restricted controller, but would be expected to increase with controller complexity and order. However, too large an N may see the reduced order controller attempting to make unnecessary frequency domain compensation and will slow down the optimisation for online applications. A deeper investigation of the relationship between full and restricted controller order and the value of N would be an interesting area for further work.

Analagously to choosing N to suit plant and controller order, the choice of n is dictated by the severity of plant nonlinearities. In subsection 4.6.4 to follow, the ship simulation example uses three fixed models and one identified model to capture the ship nonlinearities. The ship surge axis with different velocities is an example of a relatively gentle nonlinearity, where the pole moves slightly from one operating point to the next, see Table 4.2. Thus, a small set of models is sufficient to describe the general behaviour of the ship in the example. Performing the multiple-model optimisation for models $j = 1$ to 3 in the Table, with equal probability weighting on each model, produces the restricted structure controller Bode plot in Figure 4.8, where the 3 full-order optimal controller Bode plots are shown for comparison. The restricted structure controller is, in a sense, intermediate to the 3 full-order controllers.

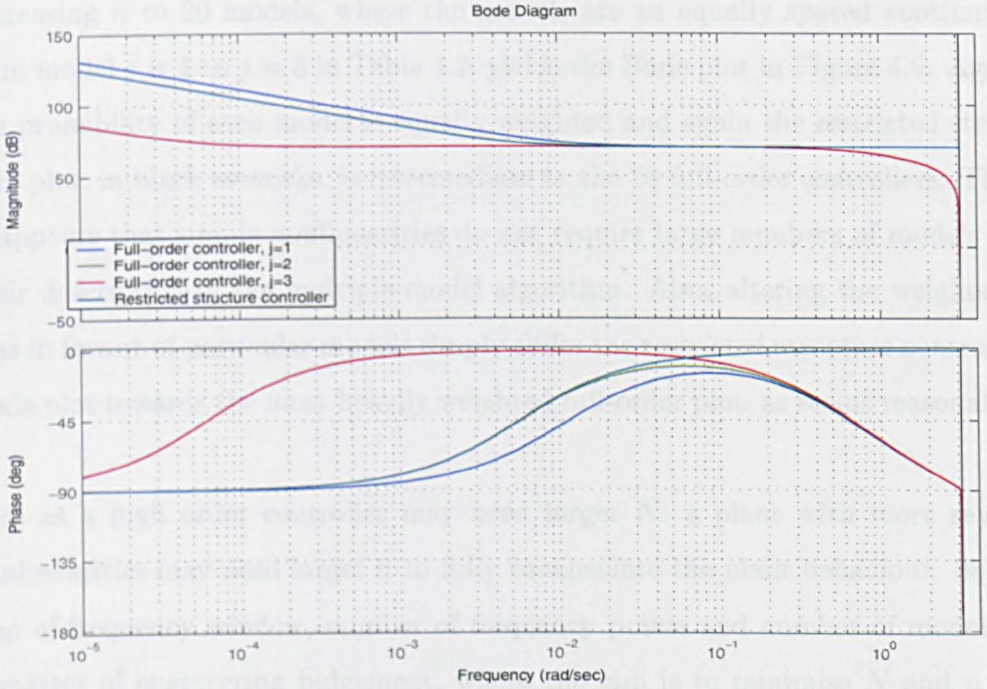

Figure 4.8: Individual full-order and restricted structure controller Bode plots

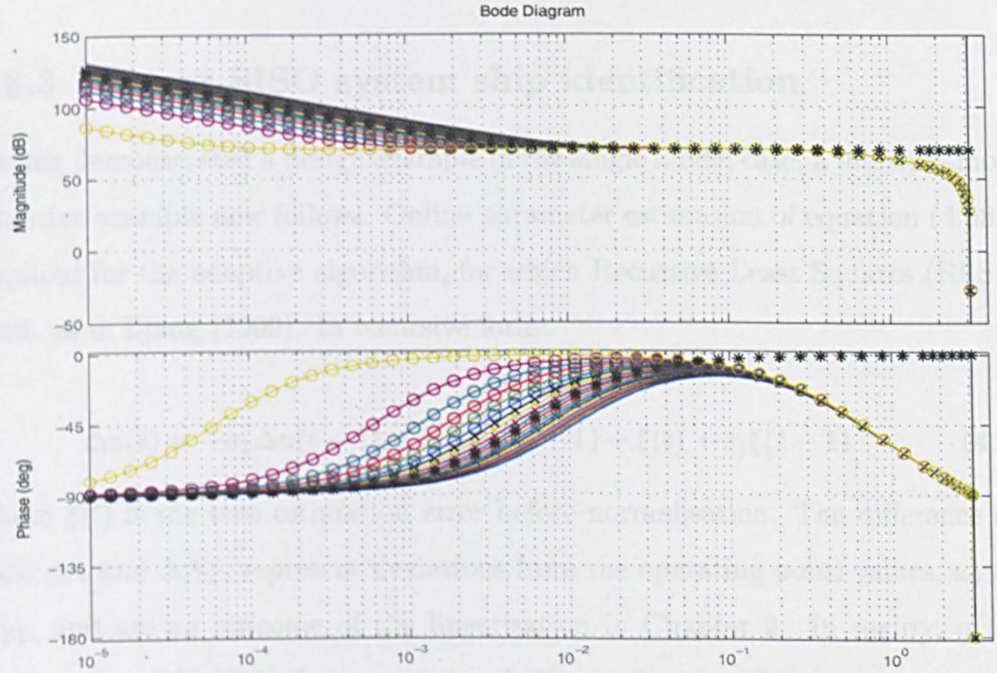

Figure 4.9: Individual full-order and restricted structure controller Bode plots

Increasing n to 20 models, where the models are an equally spaced continuum from model $j = 1$ to $j = 3$ in Table 4.2, yields the Bode plot in Figure 4.9. Again the probability of each model is equally weighted and again the restricted structure plot, in black asterisks, is intermediate to the 20 full-order controllers. Thus it appears that simple nonlinearities do not require large numbers of models for their description in the multiple-model algorithm. Also, altering the weights to bias in favour of particular models simply shifts the restricted structure controller Bode plot towards the most heavily weighted full-order plot, as seems reasonable.

Just as a high order controller may need larger N, a plant with more severe nonlinearities may need larger n to fully encapsulate the plant behaviour. Selection of frequency window, number of frequency points and number of models is a matter of engineering judgement, where the aim is to minimise N and n for algorithmic speed, but using large enough values to cover for the order of control and extremity of the nonlinearities.

4.6.3 Linear SISO system ship identification

Having demonstrated a design example in the single model case, a multiple-model adaptive example now follows. Online parameter estimation of equation (4.58) is required for the adaptive algorithm, for which Recursive Least Squares (RLS) is used, as in Ljung (1999). In recursive form:

$$\Delta u(t) = -a_1 \Delta u(t-1) + b_1 \Delta X_T(t-1) + \xi(t) + c_1 \xi(t-1) \tag{4.65}$$

where $\xi(t)$ is the true estimation error before normalisation. The difference signals, Δu and ΔX_T, represent deviations from the operating point values, u_0 and X_{T0}, and are an outcome of the linearisation in Chapter 2. In reality, $u(t)$ is measured and $X_T(t)$ is the control signal. Hence, for identification, the structure of equation (4.65) is maintained but the real signals are used instead:

$$u(t) = -a_1 u(t-1) + b_1 X_T(t-1) + \xi(t) + c_1\xi(t-1) \tag{4.66}$$

Of course, any model identified with this form will only ever be an approximation to the true nonlinear plant, but the approximation using equation (4.66) becomes progressively better than finding a_1 and b_1 analytically, as X_{T0} increases. To see this, recall that $u = u_0 + \Delta u$ and $X_T = X_{T0} + \Delta X_T$, so the identified parameters in equation (4.66) would be the same as the analytically derived versions if:

$$\begin{aligned} u_0 + \Delta u(t) &= -a_1(u_0 + \Delta u(t-1)) + b_1(X_{T0} + \Delta X_T(t-1)) + \xi(t) + c_1\xi(t-1) \\ \Rightarrow u(t) &= -a_1 u(t-1) + b_1 X_T(t-1) + \xi(t) + c_1\xi(t-1) \end{aligned} \tag{4.67}$$

However, this requires that the steady-state gain of the non-linear plant obeys:

$$\begin{aligned} u_0 &= -a_1 u_0 + b_1 X_{T0} \\ \Rightarrow \frac{u_0}{X_{T0}} &= \frac{b_1}{1+a_1} \end{aligned} \tag{4.68}$$

Compare the steady-state gains of the non-linear and linearised plants. From equations (2.1) and (2.4) at an operating point, $(u_0, v_0, r_0) = (u_0, 0, 0)$, note that:

$$\dot{u}_0 = 0 = \frac{-\rho L d X_{uu}|u_0|}{2(m+m_x)} u_0 + \frac{1}{m+m_x} X_{T0} \tag{4.69}$$

Hence:

$$\frac{u_0}{X_{T0}} = \frac{2}{\rho L d X_{uu}|u_0|} \tag{4.70}$$

From equations (2.44) and (B.1), note that:

$$\Delta\dot{u}(t) = \frac{-\rho L d X_{uu}|u_0|}{(m+m_x)} \Delta u(t) + \frac{1}{m+m_x} \Delta X_T(t) \tag{4.71}$$

Laplace-transforming:

$$\Delta u(s) = \frac{1}{(m+m_x)s + \rho L d X_{uu}|u_0|}\Delta X_T(s) \tag{4.72}$$

Hence, the steady-state gain is:

$$\frac{\Delta u}{\Delta X_T} = \frac{1}{\rho L d X_{uu}|u_0|} = \frac{b_1}{1+a_1} \tag{4.73}$$

If a zero order hold is introduced to equation (4.72) then z-transformed, the steady-state gain is unchanged, so the steady-state gain of the non-linear plant is twice that of the linearised plant, by comparison of equations (4.70) and (4.73). In conclusion, it has been shown that the non-linear plant obeys:

$$\begin{aligned} \frac{u_0}{X_{T0}} &= \frac{2b_1}{1+a_1} \\ \Rightarrow u_0 &= -a_1 u_0 + 2b_1 X_{T0} \end{aligned} \tag{4.74}$$

rather than equation (4.68). Thus, when $u_0 = 0$ and $X_{T0} = 0$, the parameters identified using equation (4.66) are the same as the analytically derived parameters. However, increasing X_{T0} produces an increasing disparity between the identified and derived parameters. Hence, a controller based on identified parameters has potentially better performance and robustness than that based on a linearisation, due to lower mismatch between the actual plant and the model. This result is true in simulation, and is strengthened in reality where the mismatch between analytical and identified models is likely to be even greater.

In matrix-vector notation, equation (4.66) is:

$$u(t) = \underline{x}^T(t)\underline{\Theta} + \xi(t) \tag{4.75}$$

where

$$\begin{aligned} \underline{x}^T(t) &= \begin{bmatrix} u(t-1) & X_T(t-1) & \xi(t-1) \end{bmatrix} \\ \underline{\Theta}^T &= \begin{bmatrix} -a_1 & b_1 & c_1 \end{bmatrix} \end{aligned}$$

The aim of least squares is to find:

$$\min_{\hat{\underline{\Theta}}}(\underline{u}(t) - X(t)\hat{\underline{\Theta}})^T(\underline{u}(t) - X(t)\hat{\underline{\Theta}}) \tag{4.76}$$

where

$$\underline{u}(t) = \begin{bmatrix} u(1) \\ \vdots \\ u(t) \end{bmatrix}, \; X(t) = \begin{bmatrix} \underline{x}^T(1) \\ \vdots \\ \underline{x}^T(t) \end{bmatrix} \tag{4.77}$$

The solution is:

$$\hat{\underline{\Theta}} = (X^T(t)X(t))^{-1}X^T(t)\underline{u}(t) \tag{4.78}$$

Clearly, this is unsuitable for online identification, because the $X^T(t)$ matrix and $\underline{u}(t)$ vector would grow over time without limit. For this reason, recursive least squares is used, and the data vector is updated at each time step. The algorithm in this case is:

Algorithm 4.6.1 (Recursive least squares algorithm) .

1. *Form $\underline{x}(t)$ using new data*
2. *Form $\epsilon(t) = u(t) - \underline{x}^T(t)\underline{\Theta}(t-1)$*
3. *Form $P(t) = \frac{1}{\lambda}P(t-1)\left[I - \frac{\underline{x}(t)\underline{x}^T(t)P(t-1)}{\lambda + \underline{x}^T(t)P(t)\underline{x}(t+1)}\right]$*
4. *Update $\hat{\underline{\Theta}}(t) = \hat{\underline{\Theta}}(t-1) + P(t)\underline{x}(t)\epsilon(t)$*
5. *Repeat for next time step*

λ is a so-called "forgetting factor", meaning that the effect of past data on the current estimate decreases with time. As a rule of thumb, the algorithm "remembers" $\frac{1}{1-\lambda}$ data points.

4.6.4 Simulation results

The example in this Section is based on the simulation in Chapter 2, but the surge axis velocity loop is to be controlled by the multiple-model adaptive controller. The surge position loop, and sway and yaw loops, have fixed controllers designed using the techniques in Chapter 2. The simulation runs for 1600 seconds, with initial reference signals of $(X, Y, \Psi) = (0, 0, 60^o)$ and current condition $(U_C, \beta_C) = (1, 0^o)$. The ship is held at this reference for 400 seconds before turning into the current, so that $(X, Y, \Psi) = (0, 0, 0^o)$ after 800 seconds. See Figure 4.20 later. This manoeuvre produces a change in operating point for the surge axis from $u_0 = U_C cos(\pi/3) = 0.5$ to $u_0 = U_C cos(0) = 1$. The reference then changes to $(X, Y, \Psi) = (1, 0, 0^o)$ to test the reference following capability at this operating point. So far, the manoeuvre is representative of realistic DP ship operation. After 1000 seconds, the current velocity increases linearly up to $(U_C, \beta_C) = (3, 0^o)$ on 1400 seconds, then remains constant until the end of the simulation. See Figure 4.17 later. Such a change in current is not particularly realistic, but does serve to demonstrate the adaptive controller performance for further changes in operating point.

The multiple-model adaptive controller utilises four linear models to represent the ship at various operating points, where $j \in \{1, 2, 3, 4\}$ is the model index for the transfer functions and polynomials given in equations (4.5) - (4.7), restated below:

$$W_j = \frac{b_{1j}z^{-1}(1-z^{-1})}{(1+a_{1j}z^{-1})(1-z^{-1})} = A_j^{-1}B_j \tag{4.79}$$

The disturbance transfer function is:

$$W_{dj} = \frac{d_{1j}(1+c_{1j}z^{-1})(1-z^{-1})}{(1+a_{1j}z^{-1})(1-z^{-1})} = A_j^{-1}C_{dj} \tag{4.80}$$

The reference transfer function is:

$$W_{rj} = \frac{(1+a_{1j}z^{-1})}{(1+a_{1j}z^{-1})(1-z^{-1})} = A_j^{-1}E_j \tag{4.81}$$

The A_j and B_j polynomials are time-invariant for $j \in \{1,2,3\}$, and the parameters are obtained as in the single model case of Section 4.6.1, for linearised $g_V(z^{-1})$ at $u_0 = 0m/s$, $u_0 = 1.5m/s$, and $u_0 = 3m/s$. These operating points span the range of velocities to be used in the simulation. Again, the sampling period is 1 second. The identified parameters, A_4 and B_4, are used to complete the set of models in the optimisation.

The parameter c_1 in C_d is set to 1 and d_1 is set to 0.0175 for the time-invariant models, in order to match the disturbances encountered in the simulation. To understand this, note that the white noise filtered by W_d is assumed to have unity covariance and standard deviation, and that the wind and wave disturbances have a standard deviation of 460. Further note that the disturbances add onto the thruster input in the Chapter 2 model, justifying the common denominator for plant and disturbance. Thus, if b_1 in equation (4.57) is 7.60×10^{-5}, then the effect of the waves can be approximated by $(7.60 \times 10^{-5}) \times 460 = 0.035$ multiplied by a unity standard deviation signal. As there are two terms in the numerator of equation (4.80), it is assumed that this value can be divided by two to give d_1. In the simulation Figure 4.12, the identified c_1 in C_{d4} tends to oscillate around 1 and the estimation error, d_1, standard deviation is around 0.01, so these are clearly reasonable assumptions for the fixed models. Table 4.2 shows the values of the parameters for each j.

The reference transfer function is set to $W_r = \dfrac{1}{1-z^{-1}} = A^{-1}E$ for all j. Integrated white noise represents the generally low frequency drifting nature of the

Table 4.2: Polynomial parameters

j	u_0	a_{1j}	b_{1j}	c_{1j}	d_{1j}
1	0	-1.00	7.601×10^{-5}	1	0.0175
2	1.5	-0.9944	7.580×10^{-5}	1	0.0175
3	3	-0.9889	7.559×10^{-5}	1	0.0175
4	0.5-3	RLS ID	RLS ID	RLS ID	RLS ID

velocity reference signal, r_{uabs}.

The error weighting in the LQG cost function is:

$$Q_c = \frac{5000}{(1-z^{-1})(1-z)} \tag{4.82}$$

This weights the error power spectrum, Φ_{ee}, heavily at low frequencies and lightly at high frequencies. This produces desirable closed-loop characteristics of zero steady-state error and good low frequency tracking, without the ship trying to reject high frequency disturbances. The control weighting in the LQG cost function is:

$$R_c = 10^{-3} \tag{4.83}$$

Here, the input power spectrum, Φ_{uu}, is weighted equally at all frequencies.

The controllers used for the sway and yaw velocity and position loops are taken from Chapter 2 (P-K/Mac case), where the linearisation is performed for $(u_0, v_0, r_0) = (2, 0, 0)$. Clearly, the ship only operates at this point after 1200 seconds of the simulation, but the focus of this example is on the surge velocity loop, so tuning of other loops is not an important consideration. During the simulation, the sway and yaw controllers produce stable closed-loop behaviour and acceptable performance," which is sufficient to meet the requirements here. Hence, the controller design in Chapter 2 is clearly robust to changes in operating point.

It is assumed that the surge loops are decoupled from the sway and yaw loops, then the controller for the surge position loop is also taken from Chapter 2. As noted in Section 4.6, the gain k_V tends to produce minimal variation in transfer functions s_V, t_V, u_V and v_V. Therefore, a fixed controller in the surge position loop will be acceptable in this application.

Figure 4.10 portrays the operation of the adaptive controller.

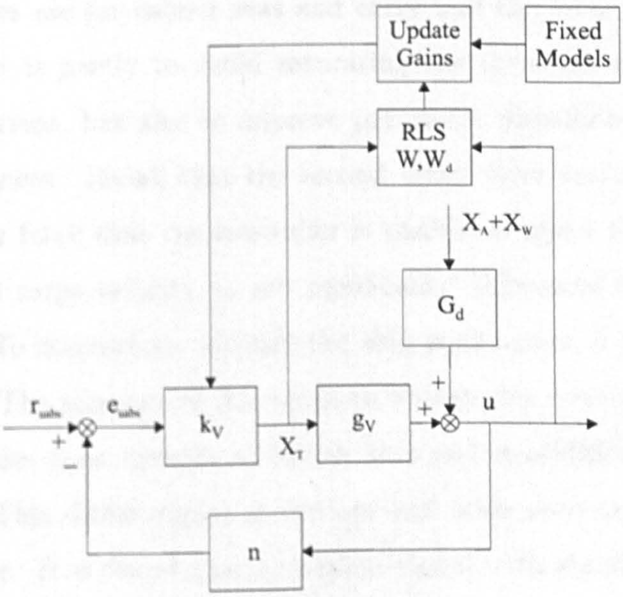

Figure 4.10: Adaptive control scheme

The recursive least squares algorithm 4.6.3 uses a forgetting factor of $\lambda = 0.95$. This produces a memory of approximately 20 data points. A shorter memory than this can make the identified parameters rather 'twitchy', making large changes from one sample time to the next and possibly producing instability. A longer memory tends to smooth out the change in identified parameters, but also leads to slower responsiveness to variations and hence to poorer control based on "out of date" parameters.

The changes in k_V are made every 4 samples. It is acceptable to make the adaptation every 4 seconds because the ship parameters do not change appreciably from one sample to the next. Simulation trials have been carried out for shorter periods of adaptation, but there is no performance advantage and the computational load is much greater. Longer periods do exhibit a degradation in performance, so 4 seconds is the best choice.

The wind disturbances in the simulation are the same as in Chapter 2, but the wave disturbances are for calmer seas and carry half the force of those used in Chapter 2. This is partly to avoid saturating the thrusters when the ship is at 60^o to the current, but also to improve parameter identifiability in the presence of disturbances. Recall that the second order wave exciting disturbances are of such large force that the controller is unable to reject them. Therefore, the variations in surge velocity, u, are significantly influenced by X_W, which is unmeasurable. To successfully identify the ship parameters, u and X_T must be well correlated. The presence of X_W tends to worsen this correlation, and so the adaptive controller must operate in calmer seas and in addition a dither signal is introduced. This dither signal is random and adds onto the X_T demanded by the controller. It is found that a random signal with standard deviation of $25Tonnes$ passed through a filter with bandwidth $0.1rad/s$ improves the parameter identification. The bandwidth is selected as $0.1rad/s$ because the dominant activity of both X_T and u is above this frequency, due to the wave exciting forces.

To establish the properties of the control scheme under investigation, it is necessary to run the simulation with an assortment of weighting values for probability p_j in equation (4.46). Starting with zero weight on the identified model and increasing by 0.1 for each simulation run, with the remaining weight shared equally among the fixed models, it is possible to see the effect of the adaptation. By evaluating equation (4.12) for each probability weighting, where $\lim_{t\to\infty}$ is ignored, the graph in Figure 4.11 is produced, depicting total cost versus identified-model weighting. Figure 4.11 shows that the cost is highest for the pure multiple-model case, $p_4 = 0$, decreasing monotonically to $p_4 = 0.9$, but increasing again for the pure adaptive solution, $p_4 = 1$. $p_4 = 0.7$ and $p_4 = 0.8$ are also superior to the pure adaptive case.

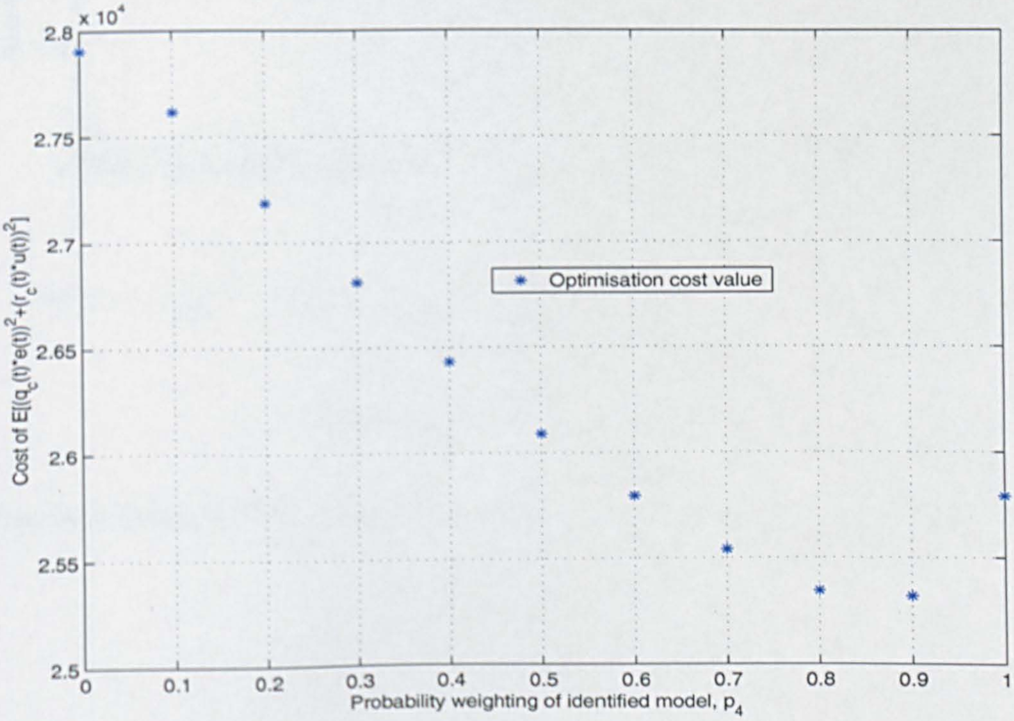

Figure 4.11: Cost versus identified model weight

The following Figures are for the $p_4 = 0.9$ case. The first 200 seconds of the simulation are with a fixed controller from the $p_4 = 0$ case. This is because the recursive least squares algorithm begins with estimated $(a_1, b_1, c_1) = (0, 0, 0)$ and, whilst the estimated a_1 and b_1 converge fairly quickly to sensible values, the c_1 parameter is much slower, taking over 150 seconds. Figure 4.12 shows how the parameter estimates evolve during the simulation.

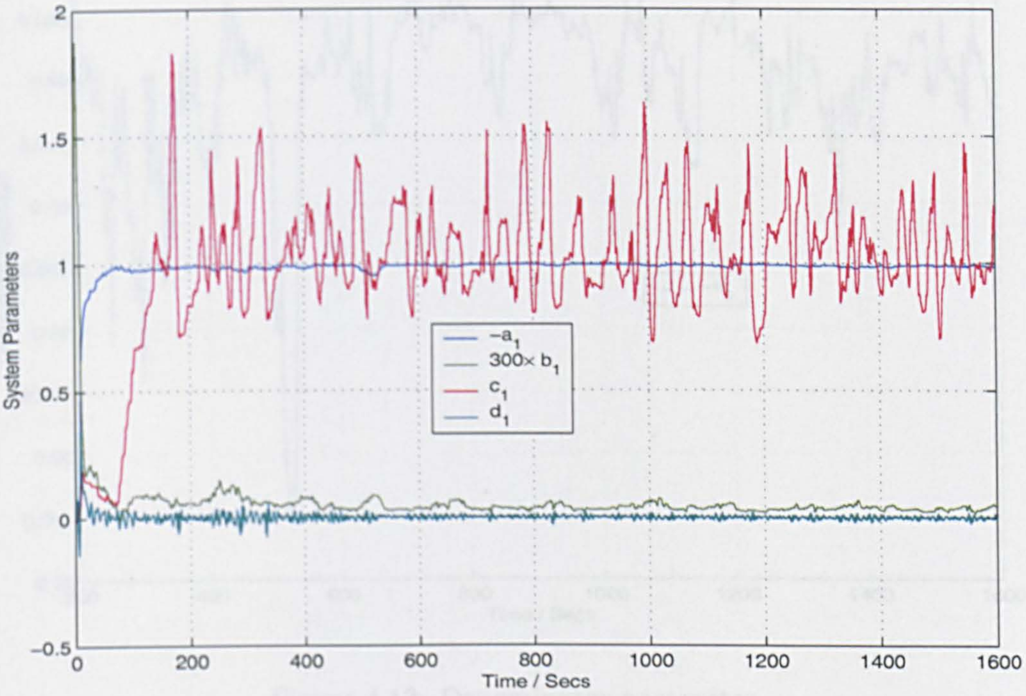

Figure 4.12: System parameters

Figures 4.13 and 4.14 show the detail of the a_1 and b_1 estimates after 200 seconds.

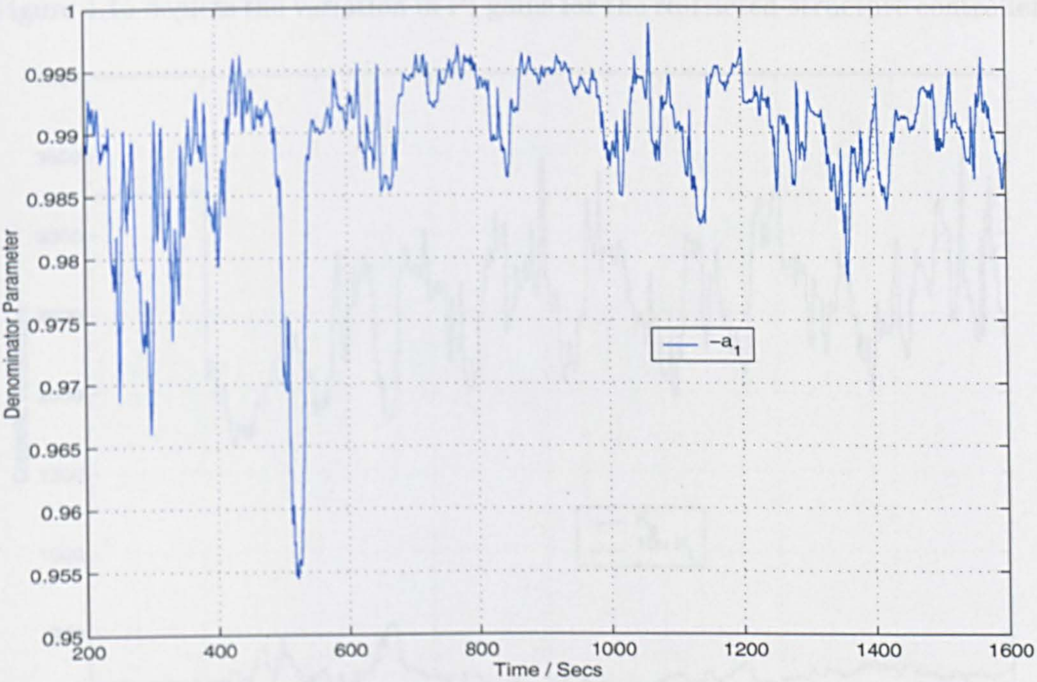

Figure 4.13: Denominator parameter

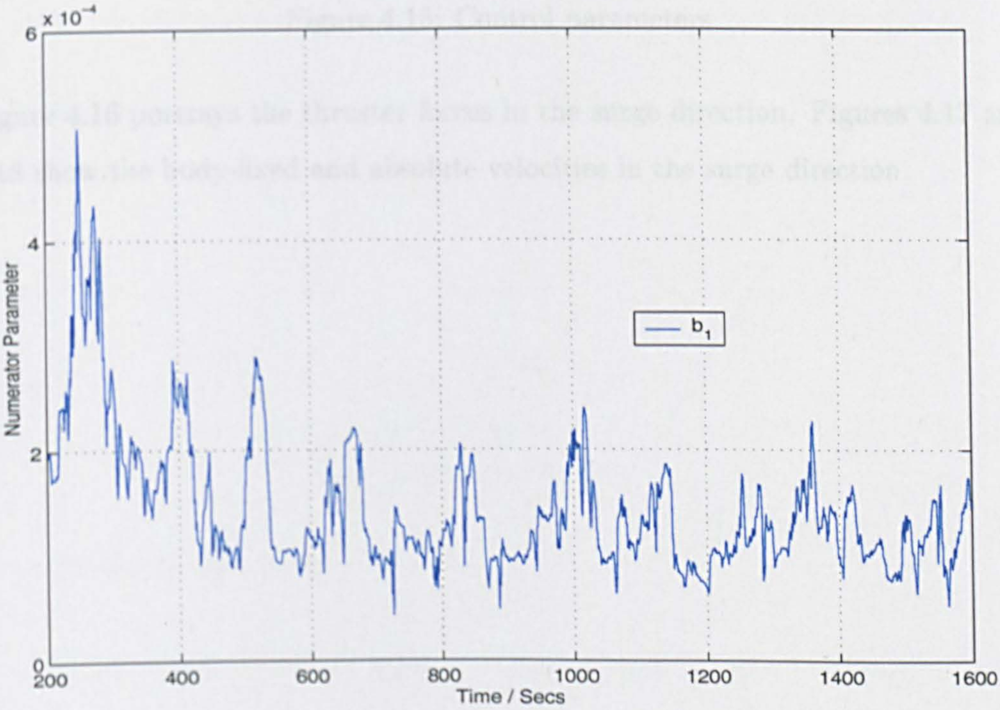

Figure 4.14: Numerator parameter

Figure 4.15 depicts the variation in PI gains for the restricted-structure controller.

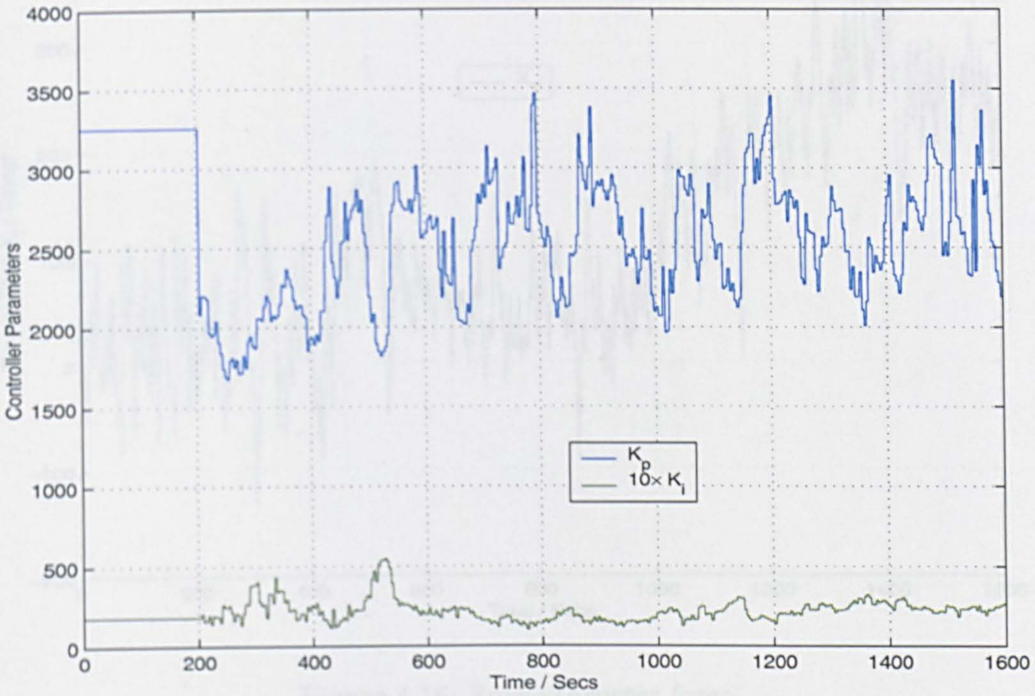

Figure 4.15: Control parameters

Figure 4.16 portrays the thruster forces in the surge direction. Figures 4.17 and 4.18 show the body-fixed and absolute velocities in the surge direction.

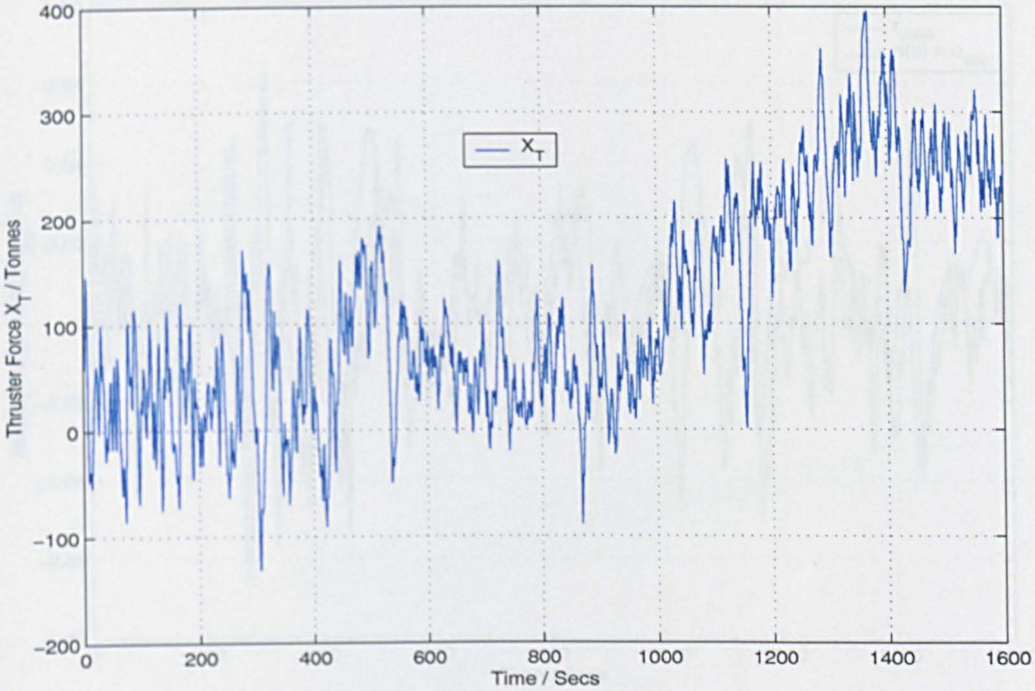

Figure 4.16: Surge thruster force

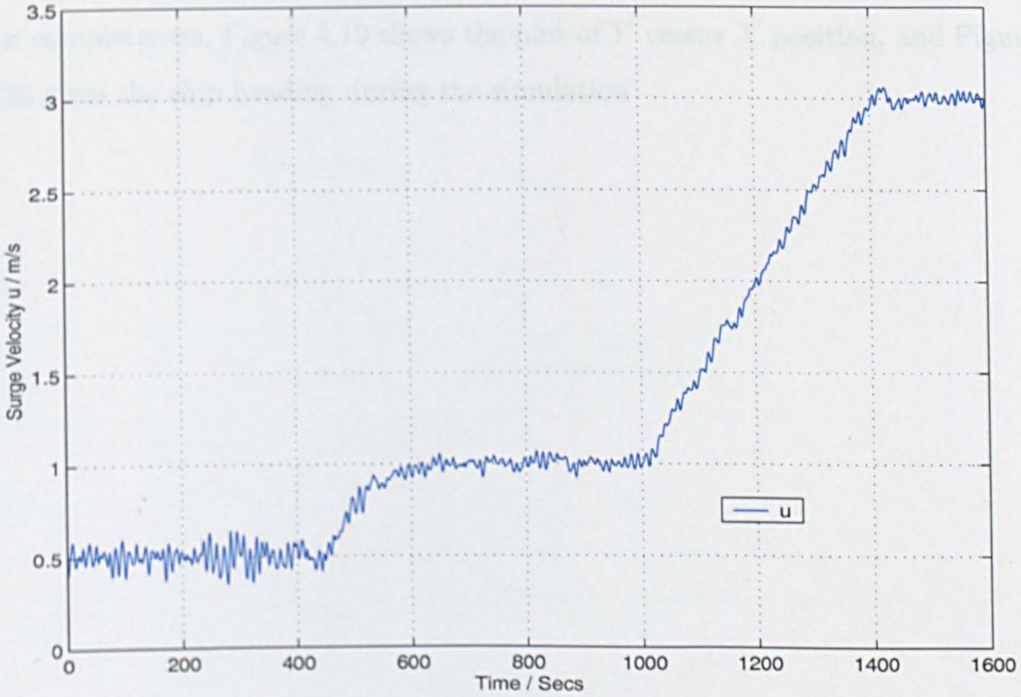

Figure 4.17: Surge velocity

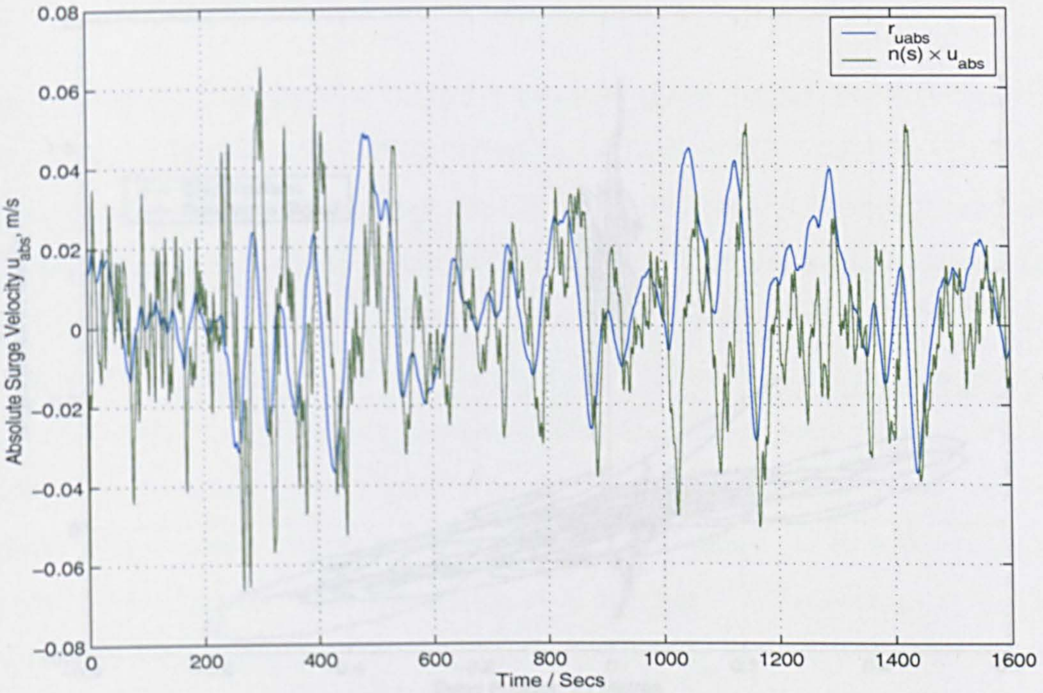

Figure 4.18: Absolute surge velocity

For completeness, Figure 4.19 shows the plot of Y versus X position, and Figure 4.20 gives the ship heading during the simulation

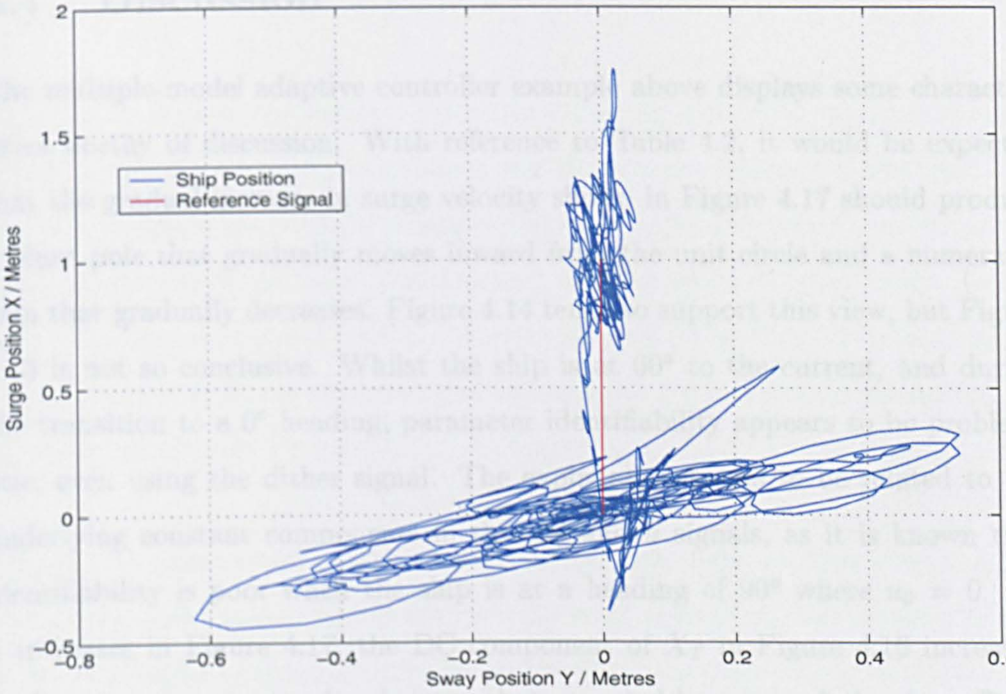

Figure 4.19: Overhead view of simulated ship position

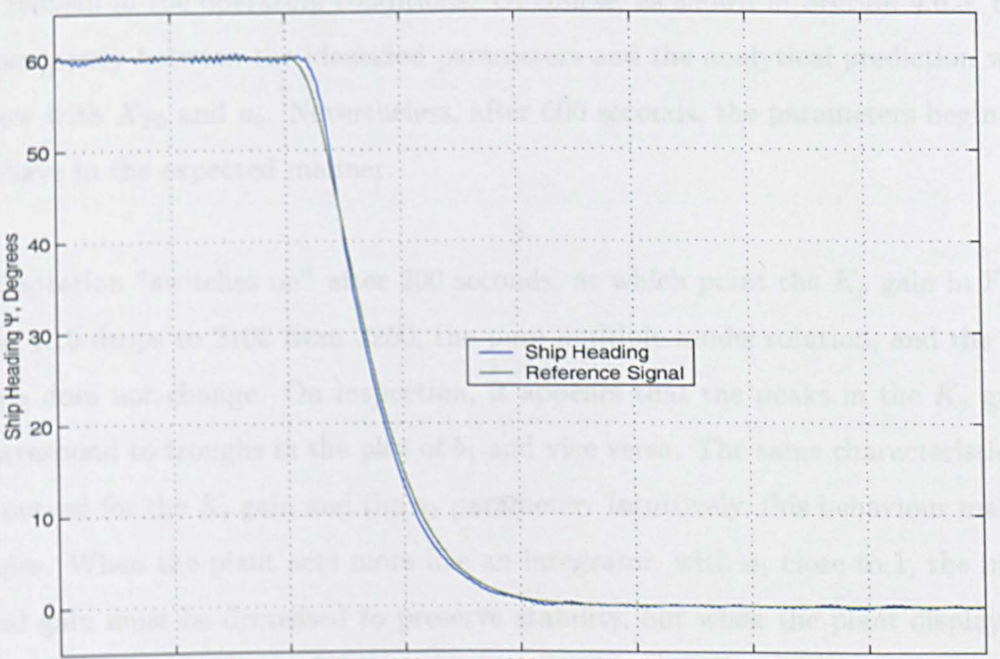

Figure 4.20: Heading of simulated ship

4.7 Discussion

The multiple-model adaptive controller example above displays some characteristics worthy of discussion. With reference to Table 4.2, it would be expected that the gradual increase in surge velocity shown in Figure 4.17 should produce a plant pole that gradually moves inward from the unit circle and a numerator gain that gradually decreases. Figure 4.14 tends to support this view, but Figure 4.13 is not so conclusive. Whilst the ship is at 60^o to the current, and during the transition to a 0^o heading, parameter identifiability appears to be problematic, even using the dither signal. The explanation seems to be related to the underlying constant components in the X_T and u signals, as it is known that identifiability is poor when the ship is at a heading of 90^o where $u_0 = 0$. As u increases in Figure 4.17, the DC component of X_T in Figure 4.16 increases, leading to parameter results closer to that expected by extrapolation from Table 4.2. Reducing the wave disturbances improves identifiability, but at the expense of realism in the operating conditions. Of course, as shown in Section 4.6.3, the discrepancy between the identified parameters and the analytical prediction will grow with X_{T0} and u_0. Nevertheless, after 600 seconds, the parameters begin to behave in the expected manner.

Adaptation "switches on" after 200 seconds, at which point the K_p gain in Figure 4.15 drops to 2100 from 3250, the pure multiple-model solution, and the K_i gain does not change. On inspection, it appears that the peaks in the K_p gain correspond to troughs in the plot of b_1 and vice versa. The same characteristic is observed for the K_i gain and the a_1 parameter. Intuitively, this behaviour makes sense. When the plant acts more like an integrator, with a_1 close to 1, the integral gain must be decreased to preserve stability, but when the plant displays a shorter time constant, after 500 seconds for example, the integral gain may be increased. Correspondingly, when the numerator gain, b_1, is large, a large K_p gain is not required, but when b_1 is small, K_p must be increased to preserve performance.

Figure 4.18 depicts the velocity reference following capability of the controller, which demonstrates good performance and disturbance rejection properties. Figure 4.19 portrays the position reference following capability when combined with fixed controllers in the surge position loop and the sway and yaw loops. Unfortunately, this result is not comparable with the results in Chapter 2, because the wave disturbance forces are lower and the operating point is different during the step response in this Chapter.

Figure 4.11 raises interesting issues regarding robustness, performance and stability. The performance figures are clear for each value of p_4, but robustness and stability margins of the controller are unknown. Such issues could potentially be addressed via the theory of simultaneous stabilisation (SS), whereby a single controller is able to simultaneously stabilise a number of plants. Vidyasagar (1985) has shown that the problem of stabilising a nonlinear plant, which is linearised about $n+1$ operating points, is equivalent to simultaneously stabilising n plants with a stable compensator. This problem is unsolved except when $n = 1$ and little has appeared towards a general solution. Ghosh (1986) has found sufficient conditions for SS of 3 different plants and Blondel et al. (1991) have presented necessary conditions for simultaneous stabilisability of more than 2 plants. There are other results of this type, but most lead to further unsolved problems. Blondel (1994) in fact concludes that the problem is undecidable by rational operations for more than two plants. Abdallah et al. (1994) restrict the class of plants to those with the same relative degree and right half-plane zeros, before providing sufficient conditions for SS and a method for computing a rational compensator.

Djaferis (1995) gives results on SS in terms of a family of polynomials with uncertainties. It is shown that stability of a family of polynomials can be proven from stability of a finite number of polynomials. However, this result requires the use of overbounding whereby a larger set of polynomials, containing the family of interest, must be considered. Several examples of controllers produced from the solution of an H_∞ problem are shown to provide SS, but this does not suggest a general technique for establishing the existence of a controller with the SS property. Nevertheless, the method is of interest and provides scope for further work on stability and robustness of the multiple-model approach in this Chapter. Fernandez-Anaya et al. (2001) and Muramatsu et al. (1999) approach SS using interpolated controllers. That is, the system is described by linear interpolation of proper stable coprime factorisations of the plant models, and the controller is a linear interpolation of coprime factorisations of stabilising controllers for these representative models. Muramatsu et al. (1999) apply this idea to two models and give necessary and sufficient condition for the plant to be stabilisable by the interpolated controller. This stability is shown to depend upon SS of the two linear models. Fernandez-Anaya et al. (2001) further this work by introducing nonlinear perturbations into the plant parameters and obtain sufficient conditions for SS using interpolated controllers. Again, this work may be of interest if it can be shown that the multiple-model controller in this Chapter may be expressed as an interpolation of controllers for individual plant models. Clearly, this is a fertile area for research.

In Section 4.4, the problem of convergence of the iterative restricted-structure algorithm was mentioned. This may not be easy to establish, but is obviously desirable to bring rigour to the algorithm. Knowing from Section 4.4 that $(K_p, K_i)_{n+1} = x_{n+1} = T(x_n) = (\underline{A}^T(x_n)P\underline{A}(x_n))^{-1}\underline{A}^T(x_n)P\underline{b}(x_n)$, a sequence $\{x_n\}$ is produced. Convergence could be proved by showing that $\{x_n\}$ is a Cauchy sequence i.e. for any $\varepsilon > 0$ there exists N such that $m, n > N \Rightarrow \|x_m - x_n\| < \varepsilon$. However, the nonlinear nature of T renders this a non-trivial problem and a possibility for further work.

For the simulation example in Section 4.6.4, the probability weightings are distributed equally among the fixed models and the remainder goes to the identified model. This is rather a broad assumption, hence another avenue of development could be to allocate realistic probabilities based on some algorithm as in Athans et al. (1977), mentioned in the Introduction. In that paper, the Kalman filter residuals for each linear model were used to generate weightings so that a weighted sum of the different optimal control signals could be applied to the plant. A neural network could perhaps be trained to produce probability weightings for each fixed linear model in the multiple-model optimisation. In this way, the optimisation would be performed across a more representative model set than simply assuming all models are equally likely to describe the true plant.

In the Introduction to this Chapter, whilst discussing adaptive control schemes, the technique of gain scheduling was described. Gain scheduling is a scheme where several linear control designs are carried out at different operating points of a nonlinear system. In practice, as the plant nears a particular operating point, a look-up table is consulted to retrieve the appropriate pre-designed gain. This is "switched in" and the gain from the previous operating point is "switched out". Shamma and Athans (1992) warn that this technique is susceptible to loss of stability and introduction of right half-plane zeros, but nevertheless gain scheduling is widely used in industry and seems suited to the example in this Chapter. Rugh (1991) and Shamma and Athans (1990) have created a theoretical framework for gain scheduling, previously a method used heuristically due to the simplicity of implementation. Åström and Wittenmark (1995) give an example of a gain-scheduled ship autopilot, but it would be informative to apply the method to the ship DP in this Chapter, for comparison with the multiple-model adaptive approach. This could produce further interesting work, particularly if the recent theoretical framework is used for performance, stability and robustness analysis.

4.8 Conclusions

This Chapter began by highlighting the fact that the ship controller design in Chapter 2 is applicable at only one operating point. The potential for instability and poor performance from this controller at other operating points is due to the effect of changing plant dynamics combined with an invariant controller. Solutions to this problem have previously involved nonlinear, robust, multiple-model or adaptive control theories. The nonlinear approach suffers from the fact that system analysis is more complicated and the theory is not so well developed, whereas robust solutions tend to produce poorer performance than might otherwise be possible.

The idea developed in this Chapter was to combine the benefits of a multiple-model and adaptive controller into one scheme, where the structure of the controller is restricted to a particular form. Unlike previous multiple-model schemes, the technique in this Chapter does not involve designing separate controllers for each model in the set, but instead performs an optimisation across the whole set. The multiple-model theory in question is based on standard optimal control theory, which is introduced in both state-space and polynomial form. The optimisation in polynomial form is then approximated by performing the cost minimisation at a finite set of frequencies, where the cost is restated in matrix-vector form and minimised using a least squares solution. This matrix-vector form introduces a further approximation whereby the controller is restricted to be of a particular structure. It is necessary to include a guess at the correct controller gains in order to produce the least squares solution. Thus, an iterative algorithm results that uses functions of the previous solution to produce the next answer.

The algorithm is extended to the multiple-model case by "stacking" matrices for each model, before computing the least squares solution across the whole set. At this point, the adaptive aspect of the algorithm is introduced by using plant parameters identified online from input-output data to add to the "stack" of matrices. Every time the optimisation is performed, the previous identified model is discarded and replaced with the latest version. This adaptive multiple-model controller is applied to the ship DP problem from Chapter 2, but first a single model example is detailed to illustrate features of the restricted-structure controller. It is seen that only four iterations are required to produce gains for PI control of the ship's surge velocity loop. The full-order controller is $3rd$ order, whereas the PI controller is only $1st$ order, yet the frequency responses between DC and $0.1rad/s$ are identical. The notch filter is ignored in order to keep the plant model order low. This is desirable to avoid numerical errors due to the use of 64 bit arithmetic, and is justifiable because the notch occurs at above the crossover frequency. Fortuitously, the step response of the restricted-structure controller is actually superior to the optimal controller when the notch filter is included in the plant but not in the model. Without a notch in the plant, the restricted-structure response is only marginally inferior to the full-order control.

Use of the recursive least squares algorithm is detailed, before a full DP example is described. The 3 degree of freedom simulation from Chapter 2 is used, but the adaptive multiple-model controller is only applied to the surge velocity loop. The other velocity loops take gains from the Penttinen-Koivo method in Chapter 2, and the position loop gains are produced with the Maciejowski technique. Linearised models are taken from three representative operating points, and the fourth model uses parameters identified with recursive least squares.

The ship executes a manoeuvre over 1600 seconds with calmer wave disturbances than in Chapter 2, to avoid thruster saturation when the ship heading points well away from the direction of the incident waves. The performance of the restricted-

PI-structure controller is evaluated for various probability weightings on each linear model. It is observed that performance is worst when the identified model is not included in the optimisation and only linearisations from three representative operating points are taken into account. As the probability weighting on the identified model is increased, the performance gradually improves to a peak when the probability is 0.9. The "pure" adaptive case, where the identified model probability equals 1, exhibits poorer performance than when the probability is 0.7, 0.8 or 0.9.

Hence, it is believed that the technique presented in this Chapter provides a combination of the benefits of adaptive and multiple-model optimal controller designs in one scheme. An adaptive controller is able to adapt to changing system parameters at the expense of possible instability, as the present controller depends upon an estimate of the current plant model only. A multiple-model optimal controller gives greater assurance of stability over a wide range of operating points with the expense of conservative performance. A multiple-model adaptive controller is intermediate to these two schemes. It provides a certain amount of confidence in stability, due to the weighted effect of fixed known models in the optimisation, plus a performance enhancement due to the incorporation of system identification knowledge from one sample point to the next. The restricted structure of the control law provides simplicity of implementation, and transparency of the solution to those acquainted with much-used classical control laws. Thus, the main contribution of this Chapter was to propose a new kind of adaptive controller which combines the benefits of existing control schemes, and also to apply this to the ship DP problem.

A flaw in this technique is lack of stability or robustness proofs for the multiple models and also the convergence of the restricted-structure algorithm needs to be established. Simultaneous stabilisation theory is suggested as an approach to the first problem, and empirical evidence suggests that convergence is likely to be assured if a suitable procedure can be found. Establishing these two results would also reinforce the perceived advantages of the adaptive controller, which could be further improved by allocating realistic probability weightings for the fixed models in some manner. It is also suggested that a gain scheduling example could be used for comparison with the controller in this Chapter, as there are similarities between the two approaches in that a set of linearised models is required in both cases.

Chapter 5

Restricted-Structure Multiple-Model Adaptive Predictive Control

In this Chapter, linear quadratic Gaussian predictive control (LQGPC) theory is developed in a similar manner to Chapter 4, *yielding a restricted-structure adaptive* predictive *controller. The problem statement involves a multi-step cost function, where the system is described in state-space with stochastic disturbance and reference generating models. An optimisation is performed over the future control signals, leaving the cost in a standard LQG form. An approximation to the cost integral is made and a restricted-structure solution across multiple models is described. The online optimisation is performed across a set of several linearised models with one identified model, to produce an adaptive algorithm. The example is again the ship DP control problem, allowing comparisons to be drawn between predictive and non-predictive controllers. Successful simulation results are presented and analysed at the end of the Chapter, and it is found that the predictive controller outperforms the standard LQG controller of Chapter* 4 *when the reference signal approximates filtered white noise.*

5.1 Introduction

In the previous Chapter, the solution to an LQG optimal control problem was approximated so that the optimisation could be performed across a set of multiple models for a controller of restricted structure. More recently, the LQG solution methodology, outlined in Section 4.2, has been applied to polynomial-form predictive control problems, known as linear quadratic Gaussian predictive control (LQGPC), in Grimble (1990) and Grimble (1995). The state-space solution of LQGPC followed in Ordys and Grimble (1996) and was extended to a restricted-structure form in Grimble (2001a). In this Chapter, the restricted-structure LQGPC method will be furthered to the multiple-model adaptive case, as in Chapter 4, then applied to the ship DP problem for comparison.

To put the LQGPC approach and the aims of this Chapter into context, it is informative to look at the background of predictive control. The use of prediction in control engineering problems dates back to Wiener (1949), who showed how to optimally predict the future values of a signal given a current observation of that signal corrupted with noise. The prediction is optimal in the sense that a mean square error criterion is minimised. The solution of the problem was considered impractical by engineers, however, and it was Kalman (1960b) who offered the first truly practical prediction and filtering of noisy signals using the now familiar state-space approach. Another early use of prediction in control theory was the Smith predictor in Smith (1959), for time-delay systems. This controller contains an internal model of the plant minus the delay, d, hence the plant output is effectively predicted d seconds into the future. The overall effect is to remove the delay from the closed loop for feedback control design.

Prediction was further utilised in the minimum variance (MV) regulator algorithm of Åström (1970) for stochastic systems, where the mean square output error is minimised at k steps in the future. The introduction of input and reference terms in the cost produced the generalised minimum variance (GMV) controller, see Clarke and Gawthrop (1975). Although these algorithms use a model of the plant when computing the control signal, the first example of modern model-based predictive control (MPC), known as model predictive heuristic control (MPHC), was given in Richalet et al. (1978). The common features of MPC algorithms are that a multi-step, generally quadratic, cost function is minimised given a finite future reference trajectory, a plant model and predictions based on that model, to produce an optimal control signal. The plant inputs and outputs, plus states with a state-space approach, are predicted over a finite horizon that always begins at the current time step - This is known as "receding-horizon control". The optimisation generally involves solution of a least squares problem, or a quadratic programming (QP) problem for constrained MPC. Constraints on plant variables are straightforward to handle for most MPC methods and this is one of the main advantages over other control techniques. These advantages include ease of understanding for the operator, straighforward extension to the multivariable case, operation close to or at the constraints, and ability to easily deal with dead-time and nonminimum phase characteristics.

The early MPHC algorithm was improved upon in the DMC method of Cutler and Ramaker (1980). Further advances include de Keyser and van Cauwenberghe (1981), Peterka (1984) and Ydstie (1984), but it was not until Clarke et al. (1987) that a general MPC controller could cope with a nonmininum-phase, open-loop unstable plant of unknown order with time-delay. This generalised predictive control law (GPC) was extended to cope with input constraints in Tsang and Clarke (1988) and both input and output constraints in Camacho (1993). GPC is popular in the process industries and has since been extended from the polynomial to the state-space setting in Ordys and Clarke (1993). This was not the first example of state-space MPC, early examples being Li et al. (1989) and

Ricker (1990), although state-space is relatively recent in this regard. In fact, predictive control was first addressed using step or impulse response data, followed by a transfer function or polynomial approach, but it is the state-space setting which is now considered most general, in Maciejowski (2002) for example, as it includes the other approaches and naturally describes multivariable systems.

LQGPC was introduced in Grimble (1990) in an attempt to combine the stability properties and intuitive tuning of polynomial LQG with the future reference knowledge of MPC. Two forms of controller are investigated, the first using the standard LQG cost but with knowledge of p future setpoint values, and the second including prediction error terms in the cost. For an example plant with large input delay and a nonminimum phase characteristic, improved output response is demonstrated for LQGPC over standard LQG. In Grimble (1995), the cost function and system description is similar to that used in GPC, but the solution again follows the polynomial LQG route. For each step in the cost function, a different controller transfer function generates the predicted control action, unlike in Grimble (1990) which is sub-optimal by comparison. The stability and robustness properties are the same as for an LQG optimal controller. This time, the output response of a submarine heading control example is superior to a non-predictive controller. Additionally, it is shown how the problem can be stated in matrix-vector form under the assumption of an ideal predictor, so that the constrained solution may be found using a QP solver. The disadvantage of both LQGPC schemes compared to standard MPC lies in the potential numerical problems of solving spectral factors and Diophantine equations associated with polynomials, and in the fact that the reference signal is filtered white noise rather than a known deterministic trajectory.

The aforementioned MPC literature deals exclusively with linear plant descriptions, but inevitably nonlinear systems are now being explored, in Gattu and

Zafiriou (1992), Doyle et al. (1995), Bemporad et al. (1997), Gawthrop et al. (1998) and Kouvaritakis and Cannon (Eds.) (2001), for example. The most elementary solution to nonlinear MPC is simply to replace the linear prediction model with a nonlinear one. The consequence of this is that optimisation convexity is lost, linear least squares and QP are redundant, and it is necessary to exploit nonlinear programming (NLP) algorithms to find the optimal control signal. For example, Sutton and Bitmead (1998) use Lagrange multipliers and a nonlinear state-space model, whereas Piché et al. (2000) use a feasible path reduced gradient solver and a neural network model. However, with NLP techniques, it is difficult to estimate the duration of an optimisation or even if termination will occur, so the major disadvantage is the unreliability for on-line predictive control. Of course, the accuracy of predicted signals depends upon a model that matches well with the true plant, so a linear model will not produce good MPC performance with a nonlinear or time-varying linear plant.

At present, it is therfore most practical to implement MPC for a nonlinear plant using some linear approximation. Yu et al. (1992) describe the response of a heart patient's mean arterial pressure and cardiac output to inputs of dopamine and a vasodilator with 36 linearised models. The controller is based on GPC and the multiple-model adaptation is based on the MMAC method of Athans et al. (1977) described in Section 4.1. Gopinathan et al. (1998) describe a multiple-model predictive scheme used to control an F/A-18A aircraft carrier landing manoeuvre subject to actuator failures. The switching and tuning ideas from Narendra and Balakrishnan (1997) are used to choose the linear model closest to the present plant dynamics, where the main aim is to produce fault-tolerant MPC in the presence of hard constraints. Huzmezan and Maciejowski (1998) use a quasi linear-parameter-varying (LPV) missile model, where the parameters depend on the system state rather than some exogenous variable. Again the emphasis is on fault detection, where failures adjust the constraints in the MPC algorithm.

The quasi-LPV model may be used in one of two ways, where predictions are obtained either by freezing the parameters at one point for the whole horizon, or updating the parameters at each step according to the system state. The results demonstrate that the updating strategy potentially yields higher performance.

In this Chapter, a novel contribution is made to the field of adaptive control in the form of a multiple-model restricted-structure adaptive predictive controller, although with many similarities to Chapter 4. Again, the multiple models and adaptation are used to deal with plant uncertainty, with a restricted structure controller implemented for simplicity and numerical reliability. A further parallel with Chapter 4 is that the optimisation occurs across the whole set of linear models, rather than for an individual model in the manner of Yu et al. (1992) and Gopinathan et al. (1998) above. Sunan et al. (2002) present GPC controllers with a PI structure where the plant must be of first-order-plus-delay form, and PID structure where the plant must be in second-order-plus-delay form. This is not as general as the restricted-structure method in this Chapter. Sunan et al. (2002) also describe an adaptive predictive controller, whereby recursive least squares plant identification is combined with GPC. A stability proof is given in the presence of model uncertainty for a particular class of SISO nonlinear systems. No such proof exists for the adaptive predictive controller in this Chapter, but it applicable to quite general MIMO nonlinear plants over a range of operating points.

The controller is, in a sense, a hybrid of state-space and polynomial LQGPC, as the initial formulation is in state-space, but the restricted-structure solution requires a polynomial LQG cost function. A state-space optimisation is carried out for the control signal from 1 to $H_u - 1$ steps ahead, which then reduces the problem to a standard LQG problem without prediction, solved using polynomial methods. The theory behind this approach is now developed, beginning with a statement of the stochastic MPC problem.

5.2 Predictive control problem formulation

In a conventional non-predictive optimal control problem, it is standard practice to define a cost function such that the present tracking error and control input are penalised to some degree. However in the predictive problem definition, the penalty includes both present and estimated future values of tracking error and control input. For a multivariable system with white noise input signals, the performance index to be minimised can be defined in the time domain as in Grimble (2001b):

$$\begin{aligned} J &= E\left\{ \lim_{T\to\infty} \frac{1}{2T} \sum_{t=-T}^{T} J_t \right\} \\ J_t &= \sum_{j=1}^{H_p} Q_j(r_h(t+j) - y_h(t+j))^2 + \sum_{j=0}^{H_u-1} R_j u(t+j)^2 \end{aligned} \tag{5.1}$$

where $E\{.\}$ is the unconditional expectation operator and y_h and r_h are generalised output and reference signals respectively. The error and control weightings, Q_j and R_j, need not remain fixed over the sequence of j's.

5.2.1 Plant model

The $y_h(.)$ and $u(.)$ signals to be costed in (5.1) are contained in the linear, time-invariant, discrete-time state-space system representation given below:

$$x_m(t+1) = A_m x_m(t) + B_m u(t) + D_m \xi_m(t) \tag{5.2}$$

$$z_m(t) = C_m x_m(t) + v_m(t) \tag{5.3}$$

$$y_h(t) = H_m x_m(t) \tag{5.4}$$

The state $x_m(t) \in \mathbb{R}^n$, control input $u(t) \in \mathbb{R}^l$, disturbance $\xi_m(t) \in \mathbb{R}$, observation $z_m(t) \in \mathbb{R}^m$, output noise $v_m(t) \in \mathbb{R}^m$, and generalised output $y_h \in \mathbb{R}^m$.

Also, $cov[v_m(t), v_m(\tau)] = R_{f1}\delta_{t\tau} > 0$ and $cov[\xi_m(t), \xi_m(\tau)] = \delta_{t\tau}$. The reason for introducing a generalised output is that H_m is able to cost a linear combination of states in addition to the actual plant output.

Having established the plant equations, an estimator is required to predict the inferred output for j steps ahead. This can be derived by repeatedly shifting (5.4) forward in time and substituting (5.2) into the equation obtained. The estimator is stated below:

$$Y_h = Hx_m(t) + GU + NW \tag{5.5}$$

$$Y_h = \begin{bmatrix} y_h(t+1) \\ y_h(t+2) \\ \vdots \\ y_h(t+H_p) \end{bmatrix}, H = \begin{bmatrix} H_m A_m \\ H_m A_m^2 \\ \vdots \\ H_m A_m^{H_p} \end{bmatrix}, U = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+H_u-1) \end{bmatrix}$$

$$G = \begin{bmatrix} H_m B_m & 0 & \cdots & 0 \\ H_m A_m B_m & H_m B_m & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ H_m A_m^{H_p-1} B_m & H_m A_m^{H_p-2} B_m & \cdots & H_m A_m^{H_p-H_u} B_m \end{bmatrix}$$

$$N = \begin{bmatrix} H_m D_m & 0 & \cdots & 0 \\ H_m A_m D_m & H_m D_m & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ H_m A_m^{H_p-1} D_m & H_m A_m^{H_p-2} D_m & \cdots & H_m D_m \end{bmatrix}, W = \begin{bmatrix} \xi_m(t) \\ \xi_m(t+1) \\ \vdots \\ \xi_m(t+H_p-1) \end{bmatrix}$$

where $u(t+j)$ is assumed zero for $j \geq H_u$, because these inputs do not appear in the cost function. The vectors and matrices have dimensions as follows: $Y_h \in \mathbb{R}^{mH_p}$, $H \in \mathbb{R}^{mH_p \times n}$, $G \in \mathbb{R}^{mH_p \times lH_u}$, $U \in \mathbb{R}^{lH_u}$, $N \in \mathbb{R}^{mH_p \times H_p}$, $W \in \mathbb{R}^{H_p}$.

5.2.2 Reference model

The $r_h(.)$ signals in (5.1) are obtained from a separate linear, time-invariant, discrete-time state-space system representation, described analogously to the plant as:

$$x_{r(H_p)}(t+1) = A_r x_{r(H_p)}(t) + D_r \xi_r(t) \tag{5.6}$$

$$z_r(t) = C_r x_{r(H_p)}(t) + v_r(t) \tag{5.7}$$

$$r_h(t+H_p) = H_r x_{r(H_p)}(t) \tag{5.8}$$

The state $x_{r(H_p)}(t) \in \mathbb{R}^m$, driving white noise input $\xi_r(t) \in \mathbb{R}$, output $z_r(t) \in \mathbb{R}^m$ and generalised reference $r_h(t) \in \mathbb{R}^m$. Also, $cov[v_r(t), v_r(\tau)] = R_{fo}\delta_{t\tau} > 0$ and $cov[\xi_r(t), \xi_r(\tau)] = \delta_{t\tau}$.

To create the reference signals at $\{t+1, t+2, \ldots, t+H_p-1\}$, further state variables are introduced by delaying $x_{r(H_p)}(t)$. The state equation hence created is:

$$\begin{bmatrix} x_{r(H_p)}(t+1) \\ x_{r(H_p-1)}(t+1) \\ \vdots \\ x_{r(2)}(t+1) \\ x_{r(1)}(t+1) \end{bmatrix} = \begin{bmatrix} A_r & 0 & \cdots & & 0 \\ H_r & 0 & \cdots & & \vdots \\ 0 & I & & & \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x_{r(H_p)}(t) \\ x_{r(H_p-1)}(t) \\ \vdots \\ x_{r(2)}(t) \\ x_{r(1)}(t) \end{bmatrix} + \begin{bmatrix} D_r \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \xi_r(t)$$

$$x_R(t+1) = A_R x_R(t) + D_R \xi_r(t) \tag{5.9}$$

$$z_r(t) = \begin{bmatrix} C_r & \cdots & 0 \end{bmatrix} x_R(t) + v_r(t) = C_R x_R(t) + v_r(t) \tag{5.10}$$

where the terms in equations (5.9) and (5.10) are interpreted in the obvious manner.

The vector x_R is the "wrong way up" to use as a reference. Hence, to give the same orientation as Y_h, the following matrix-vector product is formed:

$$\begin{bmatrix} r_h(t+1) \\ r_h(t+2) \\ \vdots \\ r_h(t+H_p-1) \\ r_h(t+H_p) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & I \\ 0 & 0 & \cdots & I & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & I & \cdots & 0 & 0 \\ H_r & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{r(H_p)}(t) \\ x_{r(H_p-1)}(t) \\ \vdots \\ x_{r(2)}(t) \\ x_{r(1)}(t) \end{bmatrix} \tag{5.11}$$

$$R_h = H_R x_R(t) \tag{5.12}$$

again with an obvious definition of terms in (5.12). $R \in \mathbb{R}^{mH_p}$, $H_R \in \mathbb{R}^{mH_p \times mH_p}$, $x_R \in \mathbb{R}^{mH_p}$, $A_R \in \mathbb{R}^{mH_p \times mH_p}$, $D_R \in \mathbb{R}^{mH_p}$.

5.2.3 Total system model

The total system, shown in Figure 5.1, can be represented in state-space form by using an enlarged state vector $X(t) = \begin{bmatrix} x_R(t) & x_m(t) \end{bmatrix}^T \in \mathbb{R}^{(mH_p+n)\times 1}$:

$$X(t+1) = \begin{bmatrix} A_R & 0 \\ 0 & A_m \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ B_m \end{bmatrix} u(t) + \begin{bmatrix} D_R & 0 \\ 0 & D_m \end{bmatrix} \begin{bmatrix} \xi_r(t) \\ \xi_m(t) \end{bmatrix} \tag{5.13}$$

$$X(t+1) = AX(t) + Bu(t) + D\xi(t) \tag{5.14}$$

$$\begin{bmatrix} z_r(t) \\ z_m(t) \end{bmatrix} = \begin{bmatrix} C_R & 0 \\ 0 & C_m \end{bmatrix} X(t) + \begin{bmatrix} v_r(t) \\ v_m(t) \end{bmatrix} \tag{5.15}$$

$$Z(t) = CX(t) + v_m(t) \tag{5.16}$$

The matrices C and D are partitioned to give $C_{11} = \begin{bmatrix} C_R & 0 \end{bmatrix} \in \mathbb{R}^{m\times(mH_p+n)}$, $C_{21} = \begin{bmatrix} 0 & C_m \end{bmatrix} \in \mathbb{R}^{m\times(mH_p+n)}$, $D_{11} = \begin{bmatrix} D_R \\ 0 \end{bmatrix} \in \mathbb{R}^{mH_p+n}$, and $D_{12} = \begin{bmatrix} 0 \\ D_m \end{bmatrix} \in \mathbb{R}^{mH_p+n}$. These partitions are used later, in the definition of the system transfer function matrices.

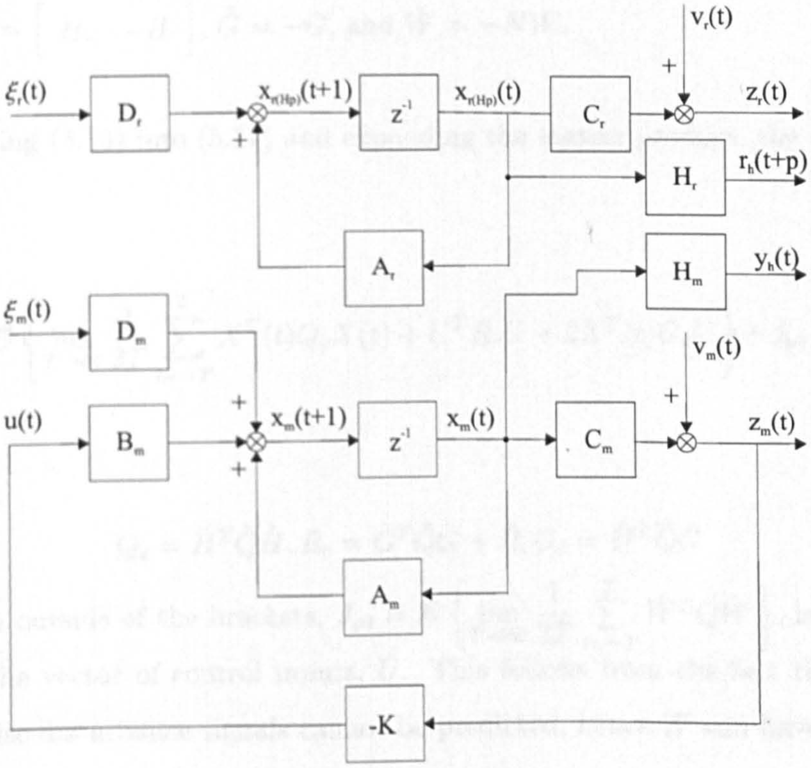

Figure 5.1: Plant Model and Reference Generator

5.2.4 Cost function manipulation

At this point, the system description is in a form where the optimal control criterion (5.1) can be used. Defining $\tilde{Q} = diag\{Q_1, \ldots Q_{H_p}\}$, $\tilde{R} = diag\{R_0, \ldots R_{H_u-1}\}$, J_t can be stated more simply in terms of matrices:

$$J_t = (R_h - Y_h)^T \tilde{Q}(R_h - Y_h) + U^T \tilde{R} U \tag{5.17}$$

Using the definitions in (5.5) and (5.12), the reference error can be stated as:

$$\begin{aligned} E = R_h - Y_h &= \begin{bmatrix} H_R & -H \end{bmatrix} X(t) - GU - NW \\ &= \tilde{H}X(t) + \tilde{G}U + \tilde{W} \end{aligned} \tag{5.18}$$

where $\tilde{H} = \begin{bmatrix} H_r & -H \end{bmatrix}$, $\tilde{G} = -G$, and $\tilde{W} = -NW$.

Substituting (5.18) into (5.17) and expanding the matrix product, the cost (5.1) becomes:

$$J = E\left\{ \lim_{T\to\infty} \frac{1}{2T} \sum_{t=-T}^{T} X^T(t)Q_cX(t) + U^TR_cU + 2X^T(t)G_cU \right\} + J_{p0} \tag{5.19}$$

where:

$$Q_c = \tilde{H}^T\tilde{Q}\tilde{H}, R_c = \tilde{G}^T\tilde{Q}\tilde{G} + \tilde{R}, G_c = \tilde{H}^T\tilde{Q}\tilde{G} \tag{5.20}$$

The term outside of the brackets, $J_{p0} = E\left\{ \lim_{T\to\infty} \frac{1}{2T} \sum_{t=-T}^{T} \tilde{W}^T\tilde{Q}\tilde{W} \right\}$, is independent of the vector of control inputs, U. This follows from the fact that future white noise disturbance signals cannot be predicted, hence W can have no influence on the U vector calculated at time t.

The mimimisation of the predictive control criterion can be simplified by noting that only $u(t)$ in U is required for the feedback control law. Therefore, partitioning the controls vector into $u(t)$ and $U^f = \begin{bmatrix} u(t+1) & \cdots & u(t+H_u-1) \end{bmatrix}^T$ components allows separate calculations for current and future inputs. To achieve this result, begin by partitioning $\tilde{G}$ and $\tilde{R}$ in accordance with U:

$$\tilde{G} = \begin{bmatrix} -G_1 & -G_2 \end{bmatrix}, \tilde{R} = \begin{bmatrix} R_0 & 0 \\ 0 & \tilde{R}_{22} \end{bmatrix}$$

where $\tilde{R}_{22} = diag\{R_1, \ldots, R_{H_u-1}\}$. The definitions of R_c and G_c can now be expressed in terms of these partititions:

$$R_c = \begin{bmatrix} R_{c1} & R_{c3} \\ R_{c3}^T & R_{c2} \end{bmatrix} = \begin{bmatrix} G_1^T\tilde{Q}G_1 + R_0 & G_1^T\tilde{Q}G_2 \\ G_2^T\tilde{Q}G_1 & G_2^T\tilde{Q}G_2 + \tilde{R}_{22} \end{bmatrix} \tag{5.21}$$

$$G_c = \begin{bmatrix} G_{c1} & G_{c2} \end{bmatrix} = \begin{bmatrix} -\tilde{H}^T\tilde{Q}G_1 & -\tilde{H}^T\tilde{Q}G_2 \end{bmatrix} \tag{5.22}$$

Denoting the summand in (5.19) with $I(t)$ and using the new terms in (5.21) and (5.22):

$$\begin{aligned} I(t) &= X^TQ_cX + u^TR_{c1}u + 2X^TG_{c1}u \\ &\quad + (U^{fT}R_{c2}U^f + 2u^TR_{c3}U^f + 2X^TG_{c2}U^f) \end{aligned} \tag{5.23}$$

To find the minimum of $I(t)$ with respect to U^f, the gradient of $I(t)$ is required:

$$\frac{\partial I(t)}{\partial U^f} = 2(U^{fT}R_{c2} + u^T(t)R_{c3} + X^T(t)G_{c2}) \tag{5.24}$$

The optimal solution occurs when this gradient is zero. Equating (5.24) with zero, the optimal controls vector is:

$$U^f = -R_{c2}(R_{c3}^Tu(t) + G_{c2}^TX(t)) \tag{5.25}$$

Substituting (5.25) into (5.23) eliminates U^f, leaving I(t) in the desired form:

$$I(t) = X^T(t)\bar{Q}_cX(t) + u^T(t)\bar{R}_cu(t) + 2X^T(t)\bar{G}_cu(t) \tag{5.26}$$

where $\bar{Q}_c = Q_c - G_{c2}R_{c2}G_{c2}^T$, $\bar{R}_c = R_{c1} - R_{c3}R_{c2}^{-1}R_{c3}^T$, $\bar{G}_c = G_{c1} - G_{c2}R_{c2}^{-1}R_{c3}^T$ and $\bar{Q}_c \in \mathbb{R}^{(mH_p+n)\times(mH_p+n)}$, $\bar{R}_c \in \mathbb{R}^{l\times l}$ and $\bar{G}_c \in \mathbb{R}^{(mH_p+n)\times l}$

5.3 Predictive control problem solution

In the previous Section, the cost criterion (5.1) was manipulated to provide an expression in terms of the defined system matrices. Also, the solution to the optimal control problem was derived for the vector of future controls. Thus, a predictive optimal cost criterion has been created such that the current state vector and current control input are weighted in the time domain. In order to find the minimum of this cost for a restricted-structure controller, a frequency domain version of the cost produces a solution. Thus, using Parseval's relation:

$$\begin{aligned} J_p &= E\left\{ \lim_{T\to\infty} \frac{1}{2T} \sum_{t=-T}^{T} X^T(t)\bar{Q}_c X(t) + u^T(t)\bar{R}_c u(t) + 2X^T(t)\bar{G}_c u(t) \right\} \\ &= \frac{1}{2\pi j} \oint_{|z|=1} trace\{\bar{Q}_c \Phi_{XX}(z^{-1}) + 2\bar{G}_c \Phi_{uX}(z^{-1}) + \bar{R}_c \Phi_{uu}(z^{-1})\} \frac{dz}{z} \quad (5.27) \end{aligned}$$

where the total cost is $J = J_p + J_{p0}$. Hence, to begin the solving the frequency domain problem, expressions for Φ_{XX}, Φ_{uu}, and Φ_{uX} in terms of the various external inputs are derived below.

Using the z-transform of (5.14), it is straightforward to obtain the following relationship:

$$\begin{aligned} X(z^{-1}) &= \Phi(z^{-1})Bu(z^{-1}) + \Phi(z^{-1})D\xi(z^{-1}) \\ &= \bar{W}u(z^{-1}) + \bar{W}_r \xi_r(z^{-1}) + \bar{W}_d \xi_m(z^{-1}) \quad (5.28) \end{aligned}$$

where

$$\begin{aligned} \Phi(z^{-1}) &= (zI - A)^{-1} = (I - Az^{-1})^{-1}z^{-1} = \bar{A}_p^{-1}z^{-1} \\ \bar{W}(z^{-1}) &= \Phi B = \bar{A}_p^{-1}z^{-1}B = \bar{A}_p^{-1}\bar{B}_p = \bar{B}_{1p}\bar{A}_{1p}^{-1} \\ \bar{W}_r(z^{-1}) &= \Phi D_{11} = \bar{A}_p^{-1}z^{-1}D_{11} \quad (5.29) \\ \bar{W}_d(z^{-1}) &= \Phi D_{12} = \bar{A}_p^{-1}z^{-1}D_{12} \end{aligned}$$

Substituting (5.28) into (5.16), obtain:

$$\begin{aligned} z_m(z^{-1}) &= Wu(z^{-1}) + W_d\xi_m(z^{-1}) + v_m(z^{-1}) && (5.30) \\ z_r(z^{-1}) &= W_r\xi_r(z^{-1}) + v_r(z^{-1}) && (5.31) \end{aligned}$$

where

$$\begin{aligned} W &= C_{21}\Phi B = A_p^{-1}B_p = B_{1p}A_{1p}^{-1} \\ W_r &= C_{11}\Phi D_{11} = A_{rp}^{-1}C_{rp} && (5.32) \\ W_d &= C_{21}\Phi D_{12} = A_{dp}^{-1}C_{dp} \end{aligned}$$

The optimal control with a single degree of freedom is:

$$u(z^{-1}) = -K(z^{-1})z_m(z^{-1}) = -K_nK_d^{-1}z_m(z^{-1}) \qquad (5.33)$$

Following an argument similar to that in Chapter 4, J_p may be minimised with respect to K. The derivation is rather lengthy and may be found in Appendix C.

To summarise, the solution involves solving two spectral factors:

$$D_{cp}^*D_{cp} = \bar{B}_{1p}^*\bar{Q}_c\bar{B}_{1p} + \bar{A}_{1p}^*\bar{R}_c\bar{A}_{1p} + \bar{B}_{1p}^*\bar{G}_c\bar{A}_{1p} + \bar{A}_{1p}^*\bar{G}_c^*\bar{B}_{1p} \qquad (5.34)$$

$$D_{dp}D_{dp}^* = C_{dp}C_{dp}^* + A_{dp}R_{f1}A_{dp}^* \qquad (5.35)$$

two regulating Diophantine equations:

$$z^{-g_1}D_{cp}^*G_{1p}^c + F_{1p}^c\bar{A}_p = (\bar{B}_{1p}^*\bar{Q}_c + \bar{A}_{1p}^*\bar{G}_c^*)z^{-g_1} \qquad (5.36)$$

$$z^{-g_1}D_{cp}^*H_{1p}^c - F_{1p}^c\bar{B}_p = (\bar{A}_{1p}^*\bar{R}_c + \bar{B}_{1p}^*\bar{G}_c)z^{-g_1} \qquad (5.37)$$

and two filtering Diophantine equations:

$$z^{-g_2} G^f_{1p} D^*_{dp} + \bar{A}_p F^f_{1p} = D_{12} C^*_{dp} z^{-g_2} \tag{5.38}$$

$$z^{-g_2} H^f_{1p} D^*_{dp} - C_{21} z^{-1} F^f_{1p} = R_{f1} A^*_p z^{-g_2} \tag{5.39}$$

The controller is then:

$$K = K_c(z\bar{A}_p + K_{f1}C_{21} + BK_c)^{-1} K_{f1} \tag{5.40}$$

where $K_c = H^{c-1}_{1p} G^c_{1p}$ and $K_{f1} = G^f_{1p} H^{f-1}_{1p}$. $G^c_{1p} \in \mathbb{R}^{l \times mH_p+n}$, $H^c_{1p} \in \mathbb{R}^{l \times l}$, $F^c_{1p} \in \mathbb{P}^{l \times mH_p+n}$, $G^f_{1p} \in \mathbb{R}^{mH_p+n \times m}$, $H^f_{1p} \in \mathbb{R}^{m \times m}$ and $F^f_{1p} \in \mathbb{P}^{mH_p+n \times m}$.

An interesting property of the solution is that the controller order does not necessarily increase with lengthened prediction and input horizons. The $\bar{Q}_c$ and $\bar{G}_c$ matrices increase in dimension with lengthening prediction horizon, but $\bar{R}_c$ is unnaffected. None of these three matrices are altered dimensionally by lengthening input horizon. Hence, the dimension and polynomial order of the spectral factors, D_{dp} and D_{cp}, are unchanged. To see this, note that $\bar{A}_{1p}$, A_{dp} and C_{dp} are unaltered by horizons and $\bar{B}_{1p}$ only acquires extra zero elements.

The dimension of the Diophantine equation solutions is changed, but the polynomial order is not. Again, this is due to unaltered $\bar{A}_{1p}$, C_{dp} and unchanged order of D_{dp} and D_{cp}. $\bar{B}_{1p}$ and $\bar{B}_p$ have extra zero elements, and $\bar{A}_p$ gains unity elements which are multiplied by extra elements in F^c_{1p} and F^f_{1p}. Thus the Diophantine equation solutions change with horizons, but not the polynomial order.

In the controller, the structure of $\bar{A}_p$, $K_{f1}C_{21}$ and BK_c is such that the order of the determinant of $(z\bar{A}_p + K_{f1}C_{21} + BK_c)$, which forms the denominator of K, is unchanged by prediction and control horizons. Of course, K is affected in some manner by these horizons, but the change is seen in the frequency domain. This will be illustrated later in Section 5.6.1.

5.4 Numerical algorithm for restricted-structure solution

The optimal solution to the predictive optimal control problem simply requires T_d^+ in equation (C.31) to be set to zero. The order of the controller is then determined by the system and cost weightings, and the structure is a matrix of rational functions. In the case of a restricted-structure control law, it is necessary that equation (C.34) be minimised with respect to the parameters of the given controller structure. The following is similar to that detailed in Section 4.4, and it will again be assumed that K is a PI controller.

Making the appropriate substitutions in equation (C.31), as in equation (C.38), obtain:

$$\begin{aligned} T_d^+ &= H_{1p}^c([I + K_c S_f B]K - K_c S_f K_{f1})(A_p + B_p K)^{-1} D_{dp} \\ &= H_{1p}^c([I + K_c S_f B]K_n - K_c S_f K_{f1} K_d)(A_p K_d + B_p K_n)^{-1} D_{dp} \\ &= (L_1 K_n - L_2 K_d)(L_3 K_n + L_4 K_d)^{-1} D_{dp} \qquad (5.41) \end{aligned}$$

where $S_f = \bar{A}_p z^{-1}(I + K_{f1} A_p^{-1} \bar{C}_{2p})^{-1} = (z\bar{A}_p + K_{f1} C_{21})^{-1}$, $L_1 = H_{1p}^c[I + K_c S_f B]$, $L_2 = H_{1p}^c K_c S_f K_{f1}$, $L_3 = B_p$, $L_4 = A_p$ and K is the matrix fraction:

$$K = K_n K_d^{-1} = (K_p \alpha_0 + K_i \alpha_1)\alpha_0^{-1} \qquad (5.42)$$

$K_p, K_i \in \mathbb{R}^{l \times m}$, $\alpha_0 = diag\{1 - z^{-1}\}$ and $\alpha_1 = I$.

T_d^+ is non-linear in K_p and K_i, meaning that (C.34) is difficult to minimise directly. However, an iterative solution is possible, as in Chapter 4, if the values of K_p and K_i in the denominator of T_d^+ are assumed known. It is possible to solve for the controller gains in the multivariable case, but the examples dealt with later are SISO. In that case, defining $L_{n1} = L_1 D_{dp}/(L_3 K_n + L_4 K_d)$ and $L_{n2} = L_2 K_d D_{dp}/(L_3 K_n + L_4 K_d)$, T_d^+ becomes linear in K_n:

$$T_d^+ = L_{n1} K_n - L_{n2} \tag{5.43}$$

This is analogous to equation (4.37) and the restricted-structure solution is identical to that in Section 4.4 with L_{n1} and L_{n2} as above and T_d^+ instead of T_1^+. Hence, using the multiple-model solution, the following adaptive restricted-structure predictive control algorithm is created.

Algorithm 5.4.1 (Adaptive restricted-structure control algorithm) .

1. *Define N (number of frequency points),* $\omega_1, \ldots, \omega_N$*,* N_f *(number of fixed models), and P (model probabilities)*

2. *Initialise* $K_p = K_i = 1$ *(arbitrary choice)*

3. *Define* $\alpha_0(z^{-1})$*,* $\alpha_1(z^{-1})$

4. *Compute* $K_n(z^{-1}) = K_p\alpha_0(z^{-1}) + K_i\alpha_1(z^{-1})$

5. *Compute* $K_d(z^{-1}) = \alpha_0(z^{-1})$

6. *For* $j = 1$ *to* N_f

    (a) *Solve for the spectral factors* D_{cpj} *and* D_{dpj}*, and the Diophantine equations for* G^c_{1pj}*,* H^c_{1pj}*,* F^c_{1pj} *and* G^f_{1pj}*,* H^f_{1pj}*,* F^f_{1pj}*.*

    (b) *Create* L_{1j}*,* L_{2j}*,* L_{3j}*,* L_{4j}*,* L_{n1j}*, and* L_{n2j}*.*

    (c) *For all chosen frequencies, calculate* $F_j(e^{-j\omega T})$*,* $L_j(e^{-j\omega T})$*.*

(d) *Assemble* $A_j = \begin{bmatrix} F_j(e^{-j\omega_1 T}) \\ \vdots \\ F_j(e^{-j\omega_N T}) \end{bmatrix}$ *and* $b_j = \begin{bmatrix} L_j(e^{-j\omega_1 T}) \\ \vdots \\ L_j(e^{-j\omega_N T}) \end{bmatrix}$

7. *Estimate current* $\bar{A}_p$, $\bar{B}_p$, $\bar{A}_{1p}$, $\bar{B}_{1p}$, A_{dp} *and* C_{dp} *polynomials by calculation from the identified parameters of a recursive least squares algorithm.*

8. *Repeat steps 6(a) to (d) for the identified polynomials.*

9. *Stack the* $N_f + 1$ A *and* b *matrices to form* $\underline{A}$ *and* $\underline{b}$

10. *Calculate the restricted-structure controller gains,* $x = (\underline{A}^T P \underline{A})^{-1} \underline{A}^T P \underline{b}$

11. *If the cost is lower than the previous cost, repeat steps 8 to 10 using the new* C_{0n}. *Otherwise, use previous controller gains to compute the feedback controller* $K_{nr}(z^{-1}) = K_p\alpha_0(z^{-1}) + K_i\alpha_1(z^{-1})$ *and* $K(z^{-1}) = K_{nr}(z^{-1})/K_{dr}(z^{-1})$.

12. *Implement controller in feedback loop and go back to step 7.*

5.5 A remark on numerical difficulties

The algorithm above also suffers from numerical errors caused by double precision arithmetic in Matlab, as explained earlier in Section 4.5. Thus, it is prudent to repeat the example of Chapter 4, where efforts were made to reduce the order of polynomials and make simplifications where this could be done with limited effect on the outcome.

5.6 Application to dynamic ship positioning

In Section 4.6, a discrete-time polynomial description was derived for the surge axis of the Chapter 2 ship model. There now follows a repeat of that example using the restricted-structure adaptive predictive controller for comparison and contrast.

5.6.1 Single model ship example

The linear ship model was originally derived in continuous-time state-space, see equations (2.47) to (2.50), so it is straightforward to state the necessary system matrices for this example. At an operating point where $u_c = u_0 = 2m/s$:

$$A = -0.00752, B = 7.60 \times 10^{-5}, C = 1, E = 7.60 \times 10^{-5} \tag{5.44}$$

Discretising for a sampling period of 1 second:

$$A_m = e^A = 0.9925, B_m = \frac{1}{A}(e^A - 1)B = 7.57 \times 10^{-5}, C_m = C = 1, D_m = 7.57 \times 10^{-5} \tag{5.45}$$

yielding:

$$g_V(z^{-1}) = W(z^{-1}) = \frac{7.57 \times 10^{-5} z^{-1}}{1 - 0.993z^{-1}} \tag{5.46}$$

The predictive control formulation does not naturally include the integral action which is desired in this example, so one solution is to introduce a fictitious integrator to the plant, carry out the control design, then take the integrator out and put it into the controller. Furthermore, recall from Section 4.6 that the wind and wave disturbances are approximated by unit variance white noise passing through a filter, W_d:

$$W_d(z^{-1}) = \frac{0.01(1 + z^{-1})}{1 - 0.993z^{-1}} \tag{5.47}$$

The state space description with input disturbance does not permit this transfer function, but a one-step delayed version can be implemented so that:

$$A_m = \begin{bmatrix} 0.9925 & 0 \\ 1 & 1 \end{bmatrix}, B_m = \begin{bmatrix} 0.9925(7.57 \times 10^{-5}) \\ 7.57 \times 10^{-5} \end{bmatrix}$$

$$C_m = \begin{bmatrix} 0 & 1 \end{bmatrix}, D_m = \begin{bmatrix} 0.01(0.9925 + 1) \\ 0.01 \end{bmatrix} = \begin{bmatrix} 0.0199 \\ 0.0100 \end{bmatrix} \tag{5.48}$$

Thus:

$$W(z^{-1}) = \frac{7.57 \times 10^{-5} z^{-1}}{(1-0.993z^{-1})(1-z^{-1})}, W_d(z^{-1}) = \frac{0.01(1+z^{-1})z^{-1}}{(1-0.993z^{-1})(1-z^{-1})} \quad (5.49)$$

The introduction of a fictitious integrator is equivalent to weighting Δu in the cost function rather than u, although it would still be necessary to include an integrator in the disturbance model to achieve integral action in the controller. In Chapter 4, time dependent weightings were convolved with the error and input signals to create dynamic weights. However, applying this idea to Q_j and R_j in equation (5.1) would remove the simplicity of the solution, so it is not clear that the restricted structure problem is tractable with such an approach.

Matlab suffers with the numerical problems highlighted in Section 4.5 when W_r has a pole at $z = 1$, hence a near integrator, $A_r = 0.990$, is used instead of the integrator in Section 4.6:

$$A_R = \begin{bmatrix} 0.990 & 0 \\ 1 & 0 \end{bmatrix}, D_R = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_R = \begin{bmatrix} 1 & 0 \end{bmatrix}, H_R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (5.50)$$

$$W_r(z^{-1}) = \frac{z^{-1}}{1-0.990z^{-1}} \quad (5.51)$$

Hence, when $H_p = 2, H_u = 2$:

$$\bar{A}_p = \begin{bmatrix} 1-0.990z^{-1} & 0 & 0 & 0 \\ -z^{-1} & 1 & 0 & 0 \\ 0 & 0 & 1-1.993z^{-1} & 0 \\ 0 & 0 & -z^{-1} & 1-z^{-1} \end{bmatrix}, \bar{B}_p = \begin{bmatrix} 0 \\ 0 \\ 7.51 \times 10^{-5} z^{-1} \\ 7.57 \times 10^{-5} z^{-1} \end{bmatrix} \quad (5.52)$$

$$\begin{aligned}\bar{A}_{1p} &= 1 - 2.983z^{-1} + 2.97z^{-2} - 0.983z^{-3} \\ \bar{B}_{1p} &= \begin{bmatrix} 0 \\ 0 \\ 7.51 \times 10^{-5}z^{-1} - 0.000150z^{-2} + 7.43 \times 10^{-5}z^{-3} \\ 7.57 \times 10^{-5}z^{-1} - 7.49 \times 10^{-5}z^{-2} \end{bmatrix}\end{aligned} \tag{5.53}$$

$$A_{dp} = 1{-}2.983z^{-1}{+}2.97z^{-2}{-}0.983z^{-3},\ C_{dp} = 0.0100z^{-1}{+}0.000100z^{-2}{-}0.0099z^{-3} \tag{5.54}$$

Let the error, control weightings and output noise covariance be:

$$Q_1 = 1000, Q_2 = 1000\ ,\ R_0 = 0.01, R_1 = 0, R_{f1} = 10^{-4} \tag{5.55}$$

Also let $H_m = C_m$, so that the usual plant output is penalised. Increasing Q_1 and decreasing R_0 tends to increase the controller gain throughout the frequency spectrum and vice versa, as expected. However, it is interesting to note that Q_2 and R_1 have virtually no effect at all on the solution, even with extreme values. An increase in plant output noise, R_{f1}, tends to decrease the gain throughout the frequency spectrum and vice versa, but particularly at high frequencies.

Solving equations (5.34) and (5.35) yields:

$$\begin{aligned} D_{cp} &= 0.113 - 0.314z^{-1} + 0.290z^{-2} - 0.0899z^{-3} \\ D_{dp} &= 0.0266 - 0.0364z^{-1} + 0.0139z^{-2} - 0.00383z^{-3} \end{aligned} \tag{5.56}$$

The Diophantine equations give:

$$\begin{aligned}
F^f_{1p} &= \begin{bmatrix} 0 \\ 0 \\ 0.000125+0.000395z^{-1}-0.000527z^{-2} \\ -0.0000605+0.000648z^{-1}-0.000593z^{-2} \end{bmatrix}^T, G^f_{1p} = \begin{bmatrix} 0 \\ 0 \\ -0.0197 \\ -0.0421 \end{bmatrix} \\
F^c_{1p} &= \begin{bmatrix} 2.59-5.73z^{-1}+3.15z^{-2} \\ 0.0612-0.198z^{-1}+0.137z^{-2} \\ -25.2+53.6z^{-1}-28.4z^{-2} \\ -2.84+6.34z^{-1}-3.50z^{-2} \end{bmatrix}^T, G^c_{1p} = \begin{bmatrix} 28.7 \\ 0.681 \\ -281 \\ -31.6 \end{bmatrix}^T \\
H^c_{1p} &= -0.111, H^f_{1p} = -0.0261 \qquad (5.57)
\end{aligned}$$

Therefore the controller is:

$$\begin{aligned}
K &= K_c(z\bar{A}_p+K_{f1}C_{21}+BK_c)^{-1}K_{f1} \\
&= \frac{20100-18200z^{-1}}{8.50-1.43z^{-1}+1.00z^{-2}} \qquad (5.58)
\end{aligned}$$

or, when the fictitious integrator is taken out of the plant and moved into the controller:

$$K = \frac{20100-18200z^{-1}}{8.50-9.94z^{-1}+2.43z^{-2}-1.00z^{-3}} \qquad (5.59)$$

Before executing the restricted-structure algorithm, it is important to note that a difference term, $\Delta = (1-z^{-1})$, must multiply the statement of the PI restricted structure so that the effect of the fictitious integrator is removed from the plant. In practice, this involves multiplying K_n in L_{n1} and L_{n2} and also in α_0 and α_1 by Δ, when stepping through Algorithm 5.3.1. The controller structure therefore becomes PD, but the P term becomes I and the D term becomes P when the integrator is removed from the plant and placed into the controller.

The restricted-structure algorithm is executed for a PI controller where $N = 15$, $\omega_1 = 10^{-4}$, $\omega_N = \pi$ and the frequency points are logarithmically spaced. The iteration begins with $x = (K_p, K_i) = (1, 1)$ as the initial guess, and Table 5.1 shows the convergence of the algorithm to a minimum after four iterations, where the cost, J_0, is calculated using the previous value of x in the denominator of L_{n1} and L_{n2}.

Table 5.1: Restricted-structure algorithm iterations

Iteration Number	$x = (K_p\ ,\ K_i)$	Cost, J_0
0	(1 , 1)	-
1	(2444.06016 , 229.86171)	2.69907×10^6
2	(2284.68162 , 229.24553)	3.25219
3	(2284.58918 , 229.23029)	3.23177
4	(2284.58874 , 229.23031)	3.23171
5	(2284.58870 , 229.23028)	3.23173

The cost increases on the fifth iteration, hence the algorithm ceases. Clearly, the second iteration almost gives the final answer, because only the decimal places change thenceforth. In practice, the algorithm can produce greater fluctuations in x from one iteration to the next and take more or less steps to converge dependent on the system and the restricted controller structure. However, given a range of frequency points that cover pertinent features of the full-order controller and a restricted structure which is sufficiently 'rich' in degrees of freedom, the iterative algorithm always seems to converge to a solution.

The restricted-structure controller in this case is:

$$K_r = 2285 + \frac{229}{1 - z^{-1}} = \frac{2514 - 2285z^{-1}}{1 - z^{-1}} \tag{5.60}$$

Figure 5.2 shows a comparison between the Bode plots of K and K_r, where the magnitude and phase plots are identical up to $0.5rad/s$. Above this frequency, the full-order plots "roll on" slightly then "roll off" with respect to the restricted structure, but the closed-loop performance is almost identical in the two cases.

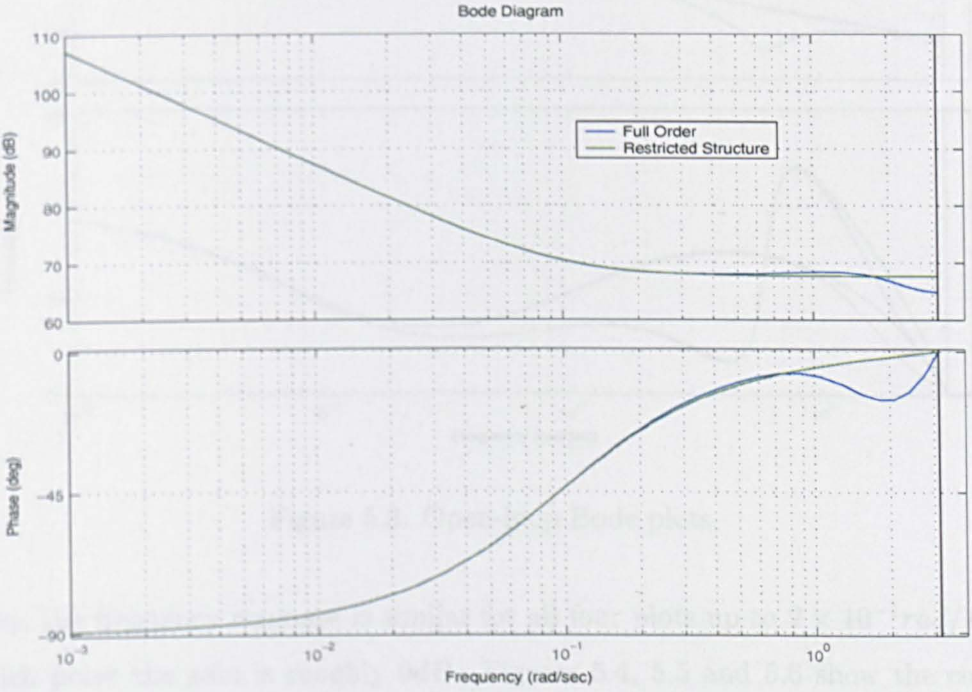

Figure 5.2: Bode plots of full order and PI structure controllers

Recall that the notch filter was neglected in the plant description, as the controller was to be designed so that the cross-over frequency occurred at below the notch frequency. Figure 5.3 shows a comparison between the Bode plots of $K(z^{-1})(Wn)(z^{-1})$, $K_r(z^{-1})(Wn)(z^{-1})$, $K(z^{-1})W(z^{-1})$ and $K_r(z^{-1})W(z^{-1})$.

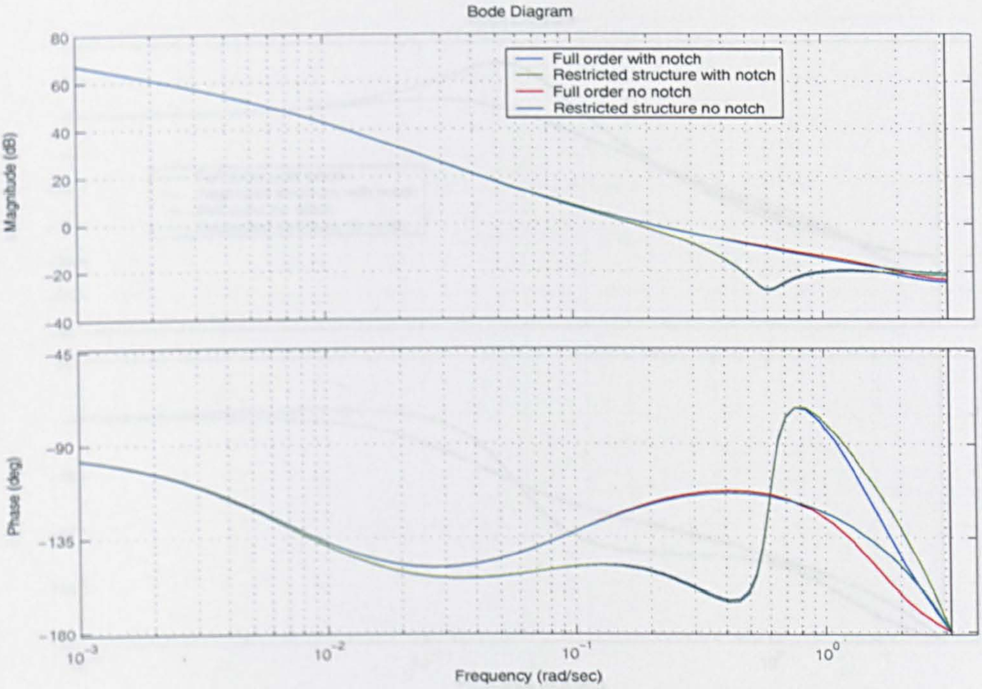

Figure 5.3: Open-loop Bode plots

Here, the frequency response is similar for all four plots up to $2 \times 10^{-1} rad/s$, at which point the gain is roughly $0dB$. Figures 5.4, 5.5 and 5.6 show the corresponding comparisons between the Bode plots of $t_V(z^{-1})$ and the step responses of $t_V(z^{-1})$ and $u_V(z^{-1})$.

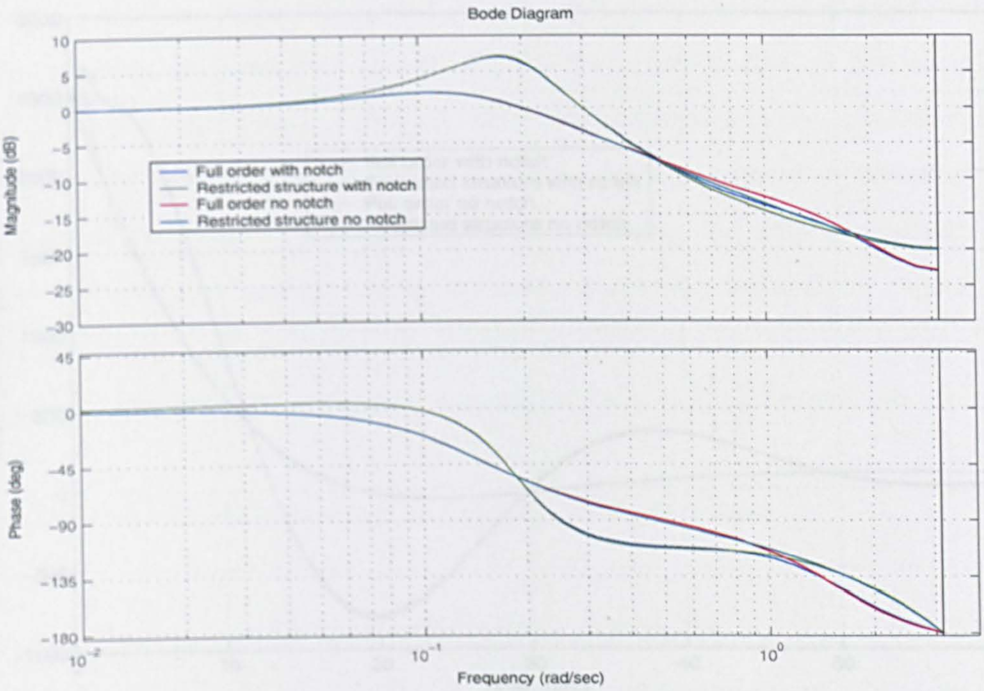

Figure 5.4: Closed-loop Bode plots

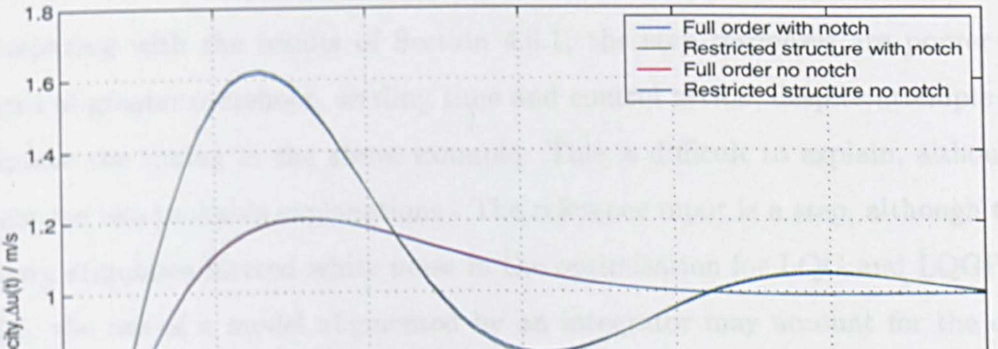

Figure 5.5: Closed-loop step responses

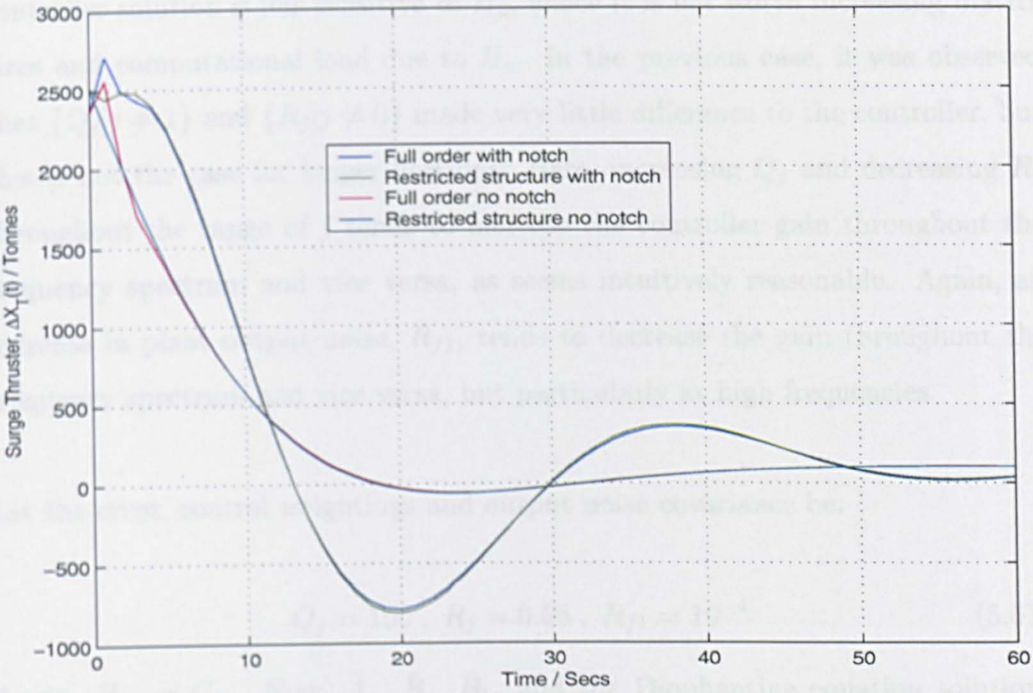

Figure 5.6: Closed-loop u_V step responses

Comparing with the results of Section 4.6.1, the step responses are poorer in terms of greater overshoot, settling time and control action, despite attempts to improve the tuning in the above example. This is difficult to explain, although there are two probable explanations - The reference input is a step, although the theory stipulates filtered white noise in the optimisation for LQG and LQGPC. Also, the use of a model augmented by an integrator may account for the difference because, in effect, the input costing is on Δu rather than u. Thus, in both cases, the controller performance is not strictly comparable between the two Chapters.

Despite this, it is worth noting that the prediction and control horizons selected are both only 2 steps, hence the predictive nature of the controller is barely exploited. To see how H_p and H_u influence the controller, the same example is carried out for $H_p = 30$ and $H_u = 5$. H_p is lengthened much more than H_u as the controller solution is less sensitive to H_u, hence it is not worth increasing matrix sizes and computational load due to H_u. In the previous case, it was observed that $\{Q_j | j \neq 1\}$ and $\{R_j | j \neq 0\}$ made very little difference to the controller, but this is not the case for longer horizons. Here, increasing Q_j and decreasing R_j throughout the range of j tends to increase the controller gain throughout the frequency spectrum and vice versa, as seems intuitively reasonable. Again, an increase in plant output noise, R_{f1}, tends to decrease the gain throughout the frequency spectrum and vice versa, but particularly at high frequencies.

Let the error, control weightings and output noise covariance be:

$$Q_j = 100 \; , \; R_j = 0.05 \; , \; R_{f1} = 10^{-4} \tag{5.61}$$

Again, $H_m = C_m$. Now, $\bar{A}_p$, $\bar{B}_p$, $\bar{B}_{1p}$ and the Diophantine equation solutions have grown in dimension with the prediction horizon and are too large to sensibly display here. However, the controller order is unchanged, as stated earlier. The controller is now:

$$K = \frac{25400 - 24100z^{-1}}{9.15 - 10.2z^{-1} + 2.09z^{-2} - 1.00z^{-3}} \tag{5.62}$$

when the fictitious integrator is taken out of the plant and moved into the controller. The restricted-structure algorithm yields:

$$K_r = 2690 + \frac{142}{1 - z^{-1}} = \frac{2830 - 2690z^{-1}}{1 - z^{-1}} \tag{5.63}$$

after six iterations. Figure 5.7 shows a comparison between the Bode plots of K and K_r. Comparing with Figure 5.2, it is clear that the controller with longer horizons exhibits lower low frequency gain and higher high frequency gain. This translates to a larger proportional gain and smaller integral gain in the restricted-structure controller.

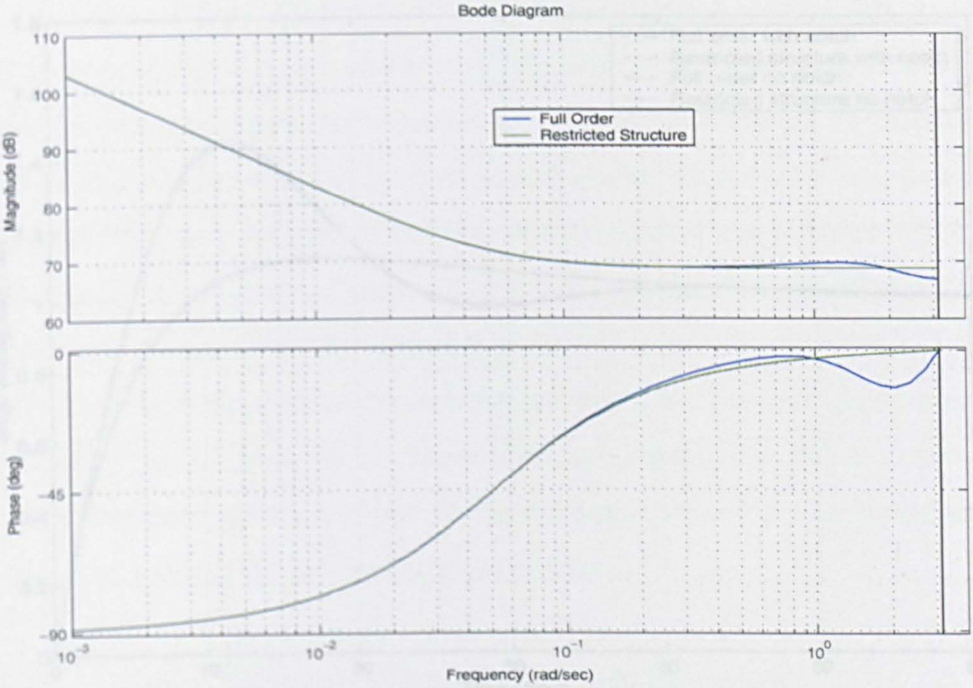

Figure 5.7: Bode plots of full order and PI structure controllers

Figures 5.8 and 5.9 show the corresponding step responses of $t_V(z^{-1})$ and $u_V(z^{-1})$.

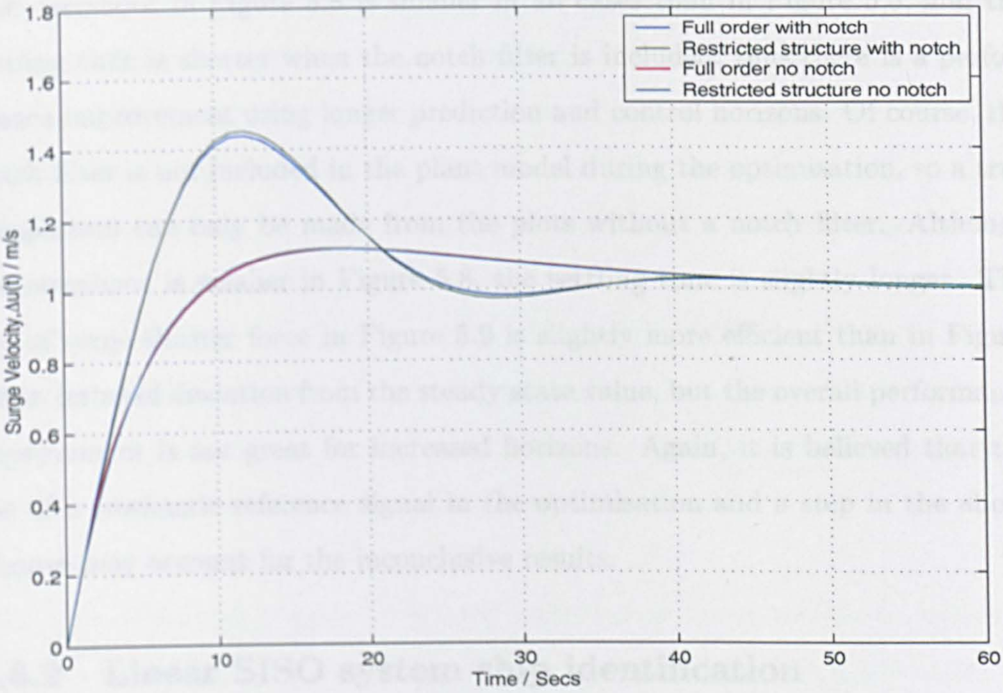

Figure 5.8: Closed-loop step responses

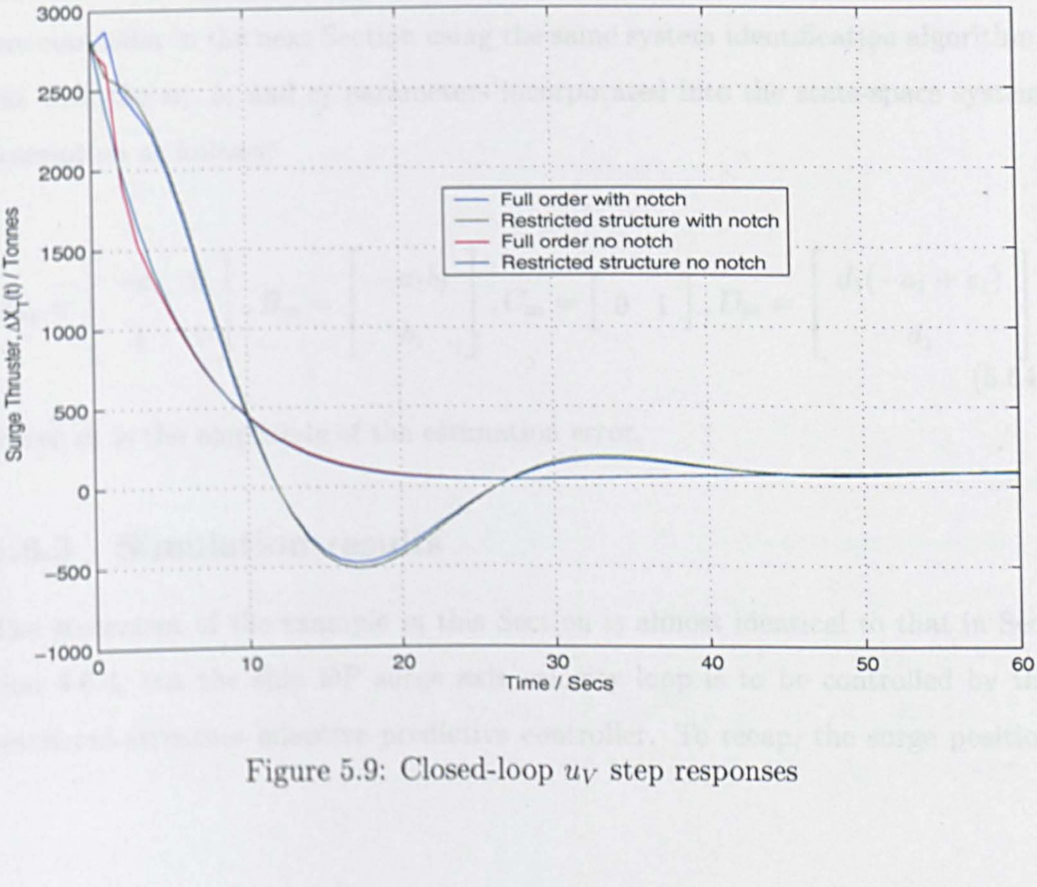

Figure 5.9: Closed-loop u_V step responses

The overshoot in Figure 5.8 is smaller in all cases than in Figure 5.5, and the settling time is shorter when the notch filter is included, thus there is a performance improvement using longer prediction and control horizons. Of course, the notch filter is not included in the plant model during the optimisation, so a true comparison can only be made from the plots without a notch filter. Although the overshoot is smaller in Figure 5.8, the settling time is slightly longer. The use of surge thuster force in Figure 5.9 is slightly more efficient than in Figure 5.6 in terms of deviation from the steady state value, but the overall performance improvement is not great for increased horizons. Again, it is believed that the use of a stochastic reference signal in the optimisation and a step in the above Figures may account for the inconclusive results.

5.6.2 Linear SISO system ship identification

In Section 4.6.3, a recursive least squares algorithm was detailed for use in the multiple-model adaptive example. That example will be repeated for the predictive controller in the next Section using the same system identification algorithm, but with the a_1, b_1 and c_1 parameters incorporated into the state-space system description as follows:

$$A_m = \begin{bmatrix} -a_1 & 0 \\ 1 & 1 \end{bmatrix}, B_m = \begin{bmatrix} -a_1 b_1 \\ b_1 \end{bmatrix}, C_m = \begin{bmatrix} 0 & 1 \end{bmatrix}, D_m = \begin{bmatrix} d_1(-a_1 + c_1) \\ d_1 \end{bmatrix} \tag{5.64}$$

where d_1 is the amplitude of the estimation error.

5.6.3 Simulation results

The statement of the example in this Section is almost identical to that in Section 4.6.4, but the ship DP surge axis velocity loop is to be controlled by the restricted-structure adaptive predictive controller. To recap, the surge position loop, and sway and yaw loops, have fixed controllers designed using the techniques in Chapter 2. The simulation runs for 1600 seconds, with initial reference signals of $(X, Y, \Psi) = (0, 0, 60^o)$ and current condition $(U_C, \beta_C) = (1, 0^o)$. The ship is held at this reference for 400 seconds before turning into the current, so that $(X, Y, \Psi) = (0, 0, 0^o)$ after 800 seconds. This manoeuvre produces a change in operating point for the surge axis from $u_0 = U_C cos(\pi/3) = 0.5$ to $u_0 = U_C cos(0) = 1$. The reference then changes to $(X, Y, \Psi) = (1, 0, 0^o)$ to test the reference following capability at this operating point. After 1000 seconds, the current velocity increases linearly up to $(U_C, \beta_C) = (3, 0^o)$ on 1400 seconds, then remains constant until the end of the simulation.

The multiple-model adaptive controller utilises four linear models to represent the ship at various operating points, where $j \in \{1, 2, 3, 4\}$ is the model index for the matrices given in equations (5.2), (5.3), (5.6) and (5.7), restated below:

$$A_{mj} = \begin{bmatrix} -a_{1j} & 0 \\ 1 & 1 \end{bmatrix}, B_{mj} = \begin{bmatrix} -a_{1j}b_{1j} \\ b_{1j} \end{bmatrix}, C_{mj} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$D_{mj} = \begin{bmatrix} d_{1j}(-a_{1j} + c_{1j}) \\ d_{1j} \end{bmatrix}, A_{rj} = 0.990, C_{rj} = 1, D_{rj} = 1 \tag{5.65}$$

The corresponding transfer functions are:

$$W_j = \frac{b_{1j}z^{-1}}{(1 + a_{1j}z^{-1})(1 - z^{-1})}, W_{dj} = \frac{d_{1j}(1 + c_{1j}z^{-1})z^{-1}}{(1 + a_{1j}z^{-1})(1 - z^{-1})}, W_{rj} = \frac{z^{-1}}{(1 - A_{rj}z^{-1})} \tag{5.66}$$

The matrices are time-invariant for $j \in \{1, 2, 3\}$, and the parameters are obtained as in the single model case of Section 5.6.1, for linearised $g_V(z^{-1})$ at $u_0 = 0m/s$, $u_0 = 1.5m/s$, and $u_0 = 3m/s$. These operating points span the range of velocities to be used in the simulation. Again, the sampling period is 1 second. The identified parameters, a_{14}, b_{14}, c_{14} and d_{14}, are used to complete the set of models in the optimisation.

The parameter c_1 is set to 1 and d_1 is set to 0.0175 for the time-invariant models, in order to match the disturbances encountered in the simulation. Table 5.2 shows the values of the parameters for each j.

Table 5.2: Polynomial parameters

j	u_0	a_{1j}	b_{1j}	c_{1j}	d_{1j}
1	0	-1.00	7.601×10^{-5}	1	0.0175
2	1.5	-0.9944	7.580×10^{-5}	1	0.0175
3	3	-0.9889	7.559×10^{-5}	1	0.0175
4	0.5-3	RLS ID	RLS ID	RLS ID	RLS ID

The error, control weightings and output noise covariance are:

$$Q_j = 100 \, , \; R_j = 0.05 \, , \; R_{f1} = 10^{-4} \tag{5.67}$$

where j is the time step in the prediction and control horizons, not the model index. H_{mj} equals C_{mj}, so the standard output is penalised.

Figure 4.10 portrays the operation of the adaptive predictive controller. The recursive least squares algorithm uses a forgetting factor of $\lambda = 0.95$ and the changes in k_V are made every 4 samples, for the same reasons as given in Section 4.6.4. The wind and wave disturbances in the simulation are the same as in Section 4.6.4, as is the dither signal. To establish the properties of the control scheme under investigation, it is necessary to run the simulation with an assortment of weighting values for probability p_j in equation (4.46). Starting with zero weight on the identified model and increasing by 0.1 for each simulation run, with the remaining weight shared equally among the fixed models, it is possible to see the effect of the adaptation. By evaluating equation (5.1) for each probability weighting, where the average over infinity is ignored, the graph in Figure 5.10 is produced, depicting total cost versus identified-model weighting. Figure 5.10 shows that the cost remains roughly equal for the pure multiple-model case, $p_4 = 0$, up to $p_4 = 0.8$. The lowest cost occurs at $p_4 = 0.9$, and interestingly is highest for the pure adaptive solution, $p_4 = 1$.

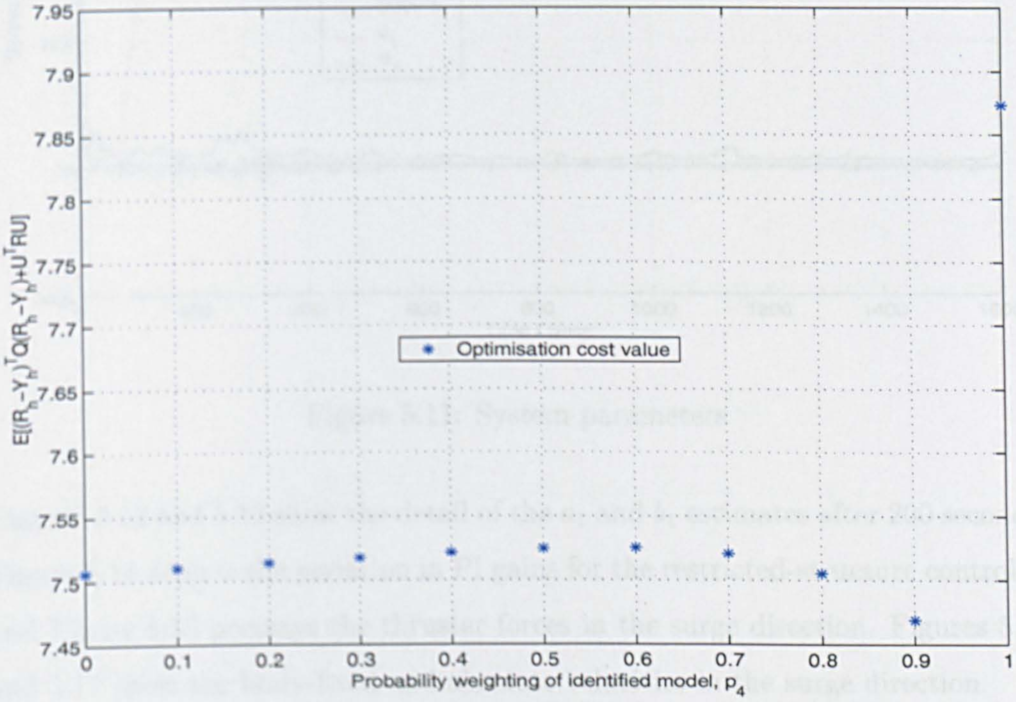

Figure 5.10: Cost versus identified model weight

The following Figures are for the $p_4 = 0.9$ case. The first 200 seconds of the simulation are with a fixed controller from the $p_4 = 0$ case. This is because the recursive least squares algorithm begins with estimated $(a_1, b_1, c_1) = (0, 0, 0)$ and, whilst the estimated a_1 and b_1 converge fairly quickly to sensible values, the c_1 parameter is much slower, taking over 150 seconds. Figure 5.11 shows how the parameter estimates evolve during the simulation.

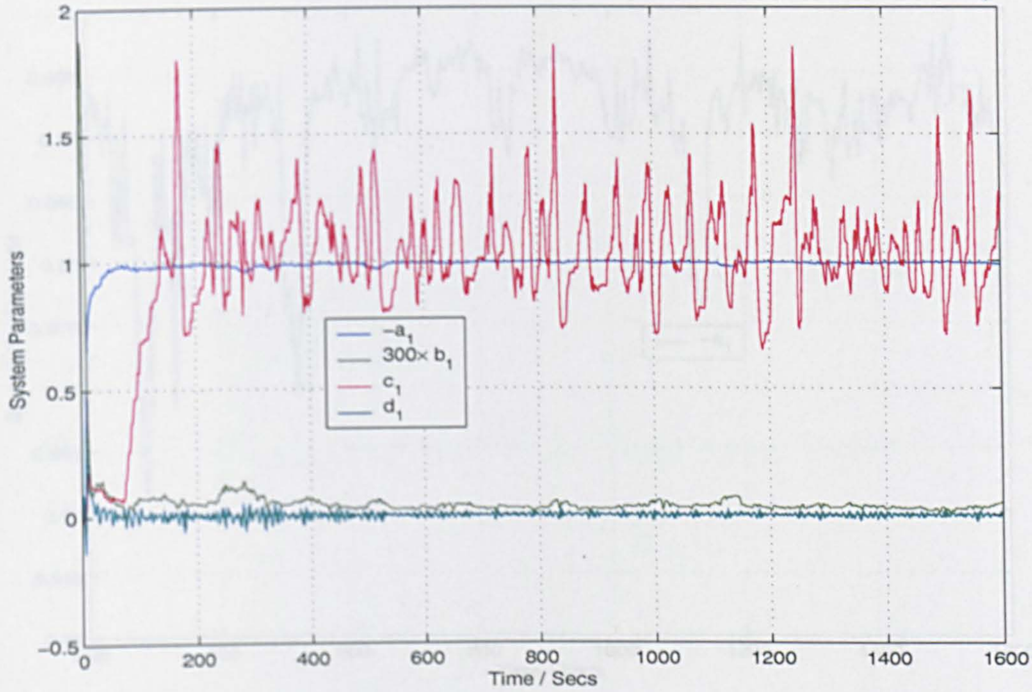

Figure 5.11: System parameters

Figures 5.12 and 5.13 show the detail of the a_1 and b_1 estimates after 200 seconds. Figure 5.14 depicts the variation in PI gains for the restricted-structure controller and Figure 5.15 portrays the thruster forces in the surge direction. Figures 5.16 and 5.17 show the body-fixed and absolute velocities in the surge direction.

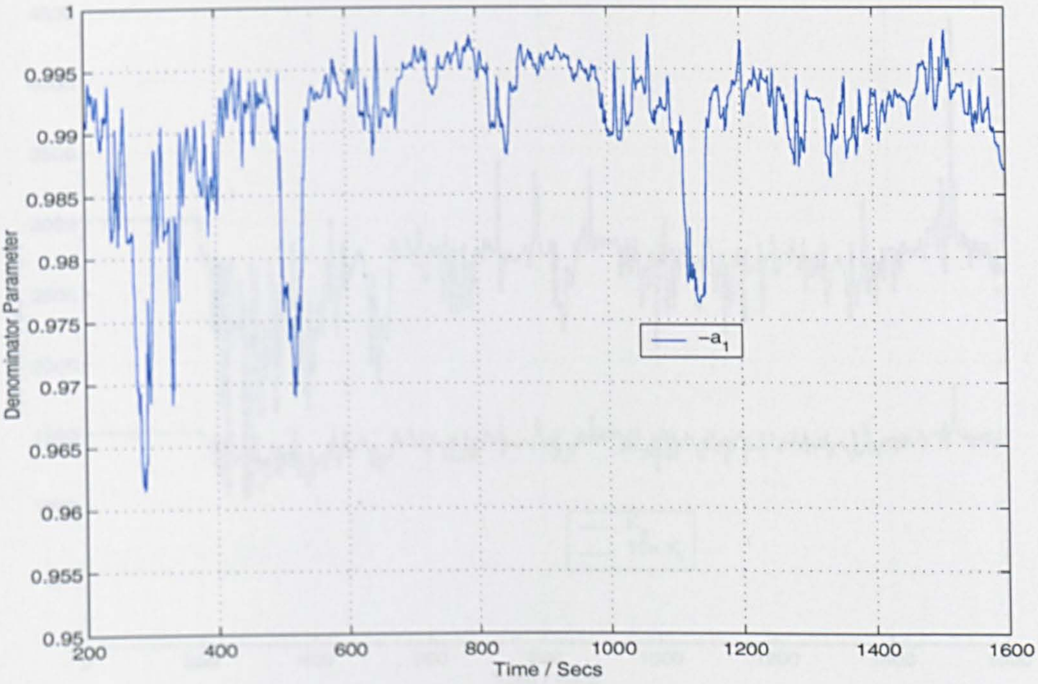

Figure 5.12: Denominator parameter

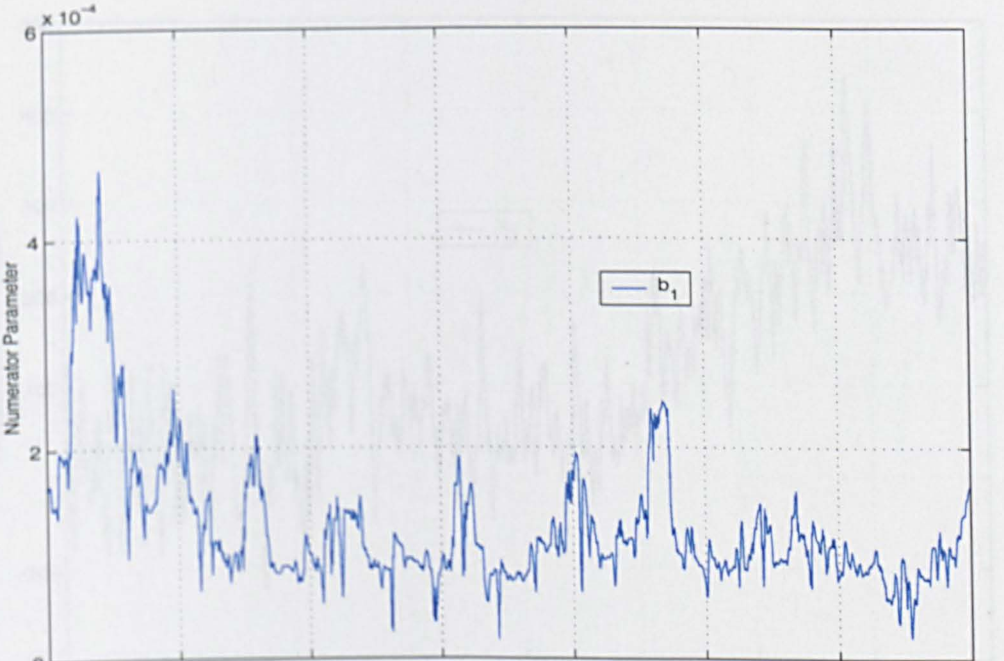

Figure 5.13: Numerator parameter

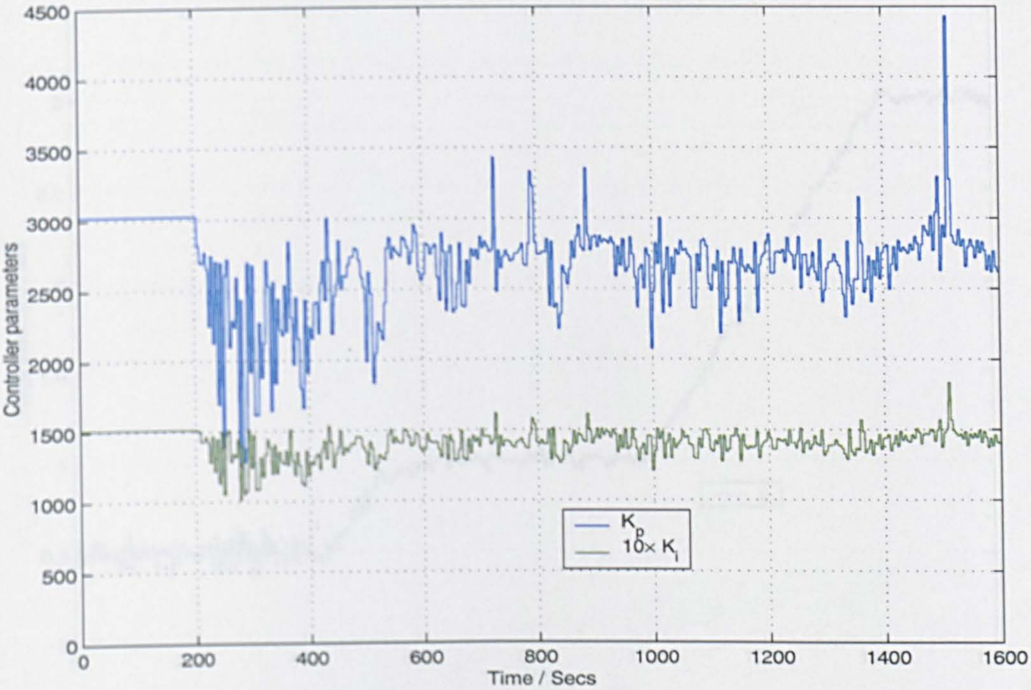

Figure 5.14: Control parameters

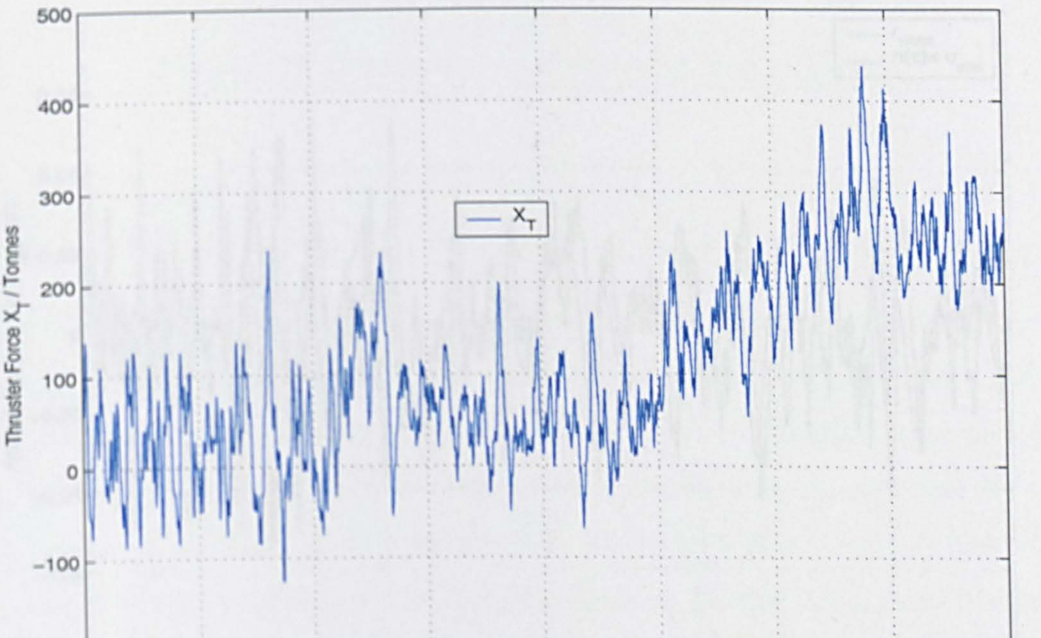

Figure 5.15: Surge thruster force

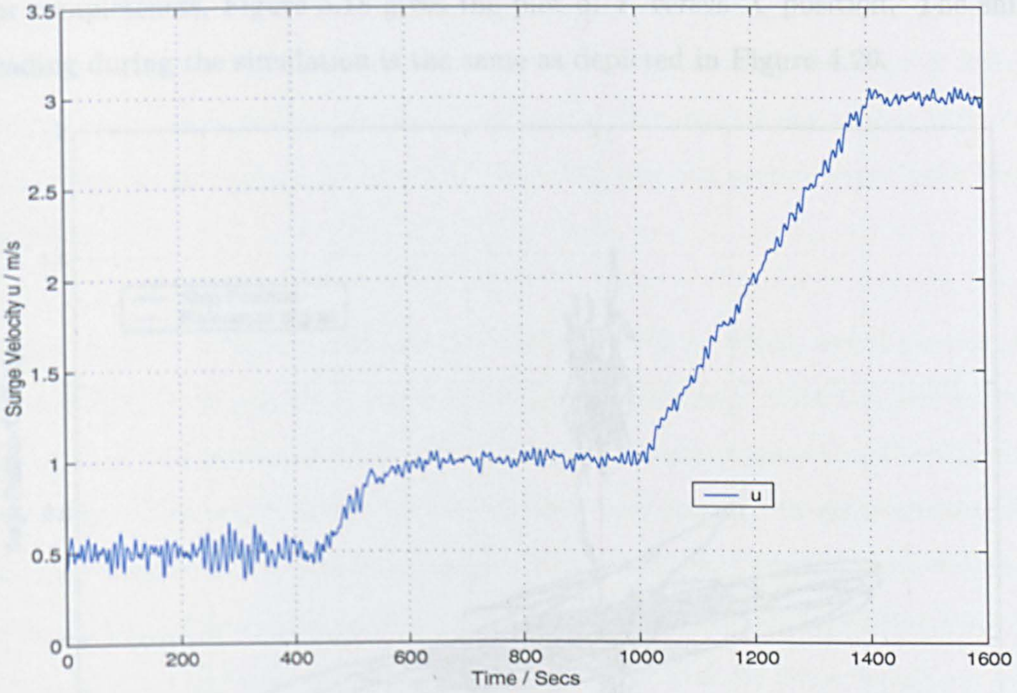

Figure 5.16: Surge velocity

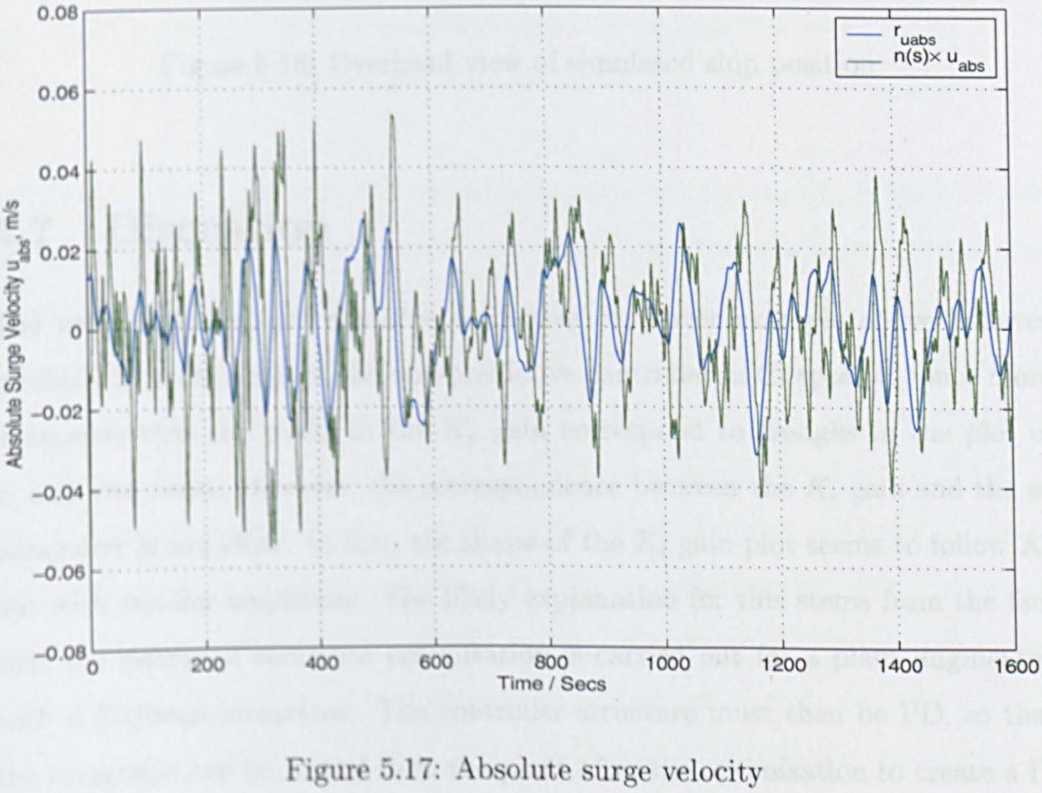

Figure 5.17: Absolute surge velocity

For completeness, Figure 5.18 gives the plot of Y versus X position. The ship heading during the simulation is the same as depicted in Figure 4.20.

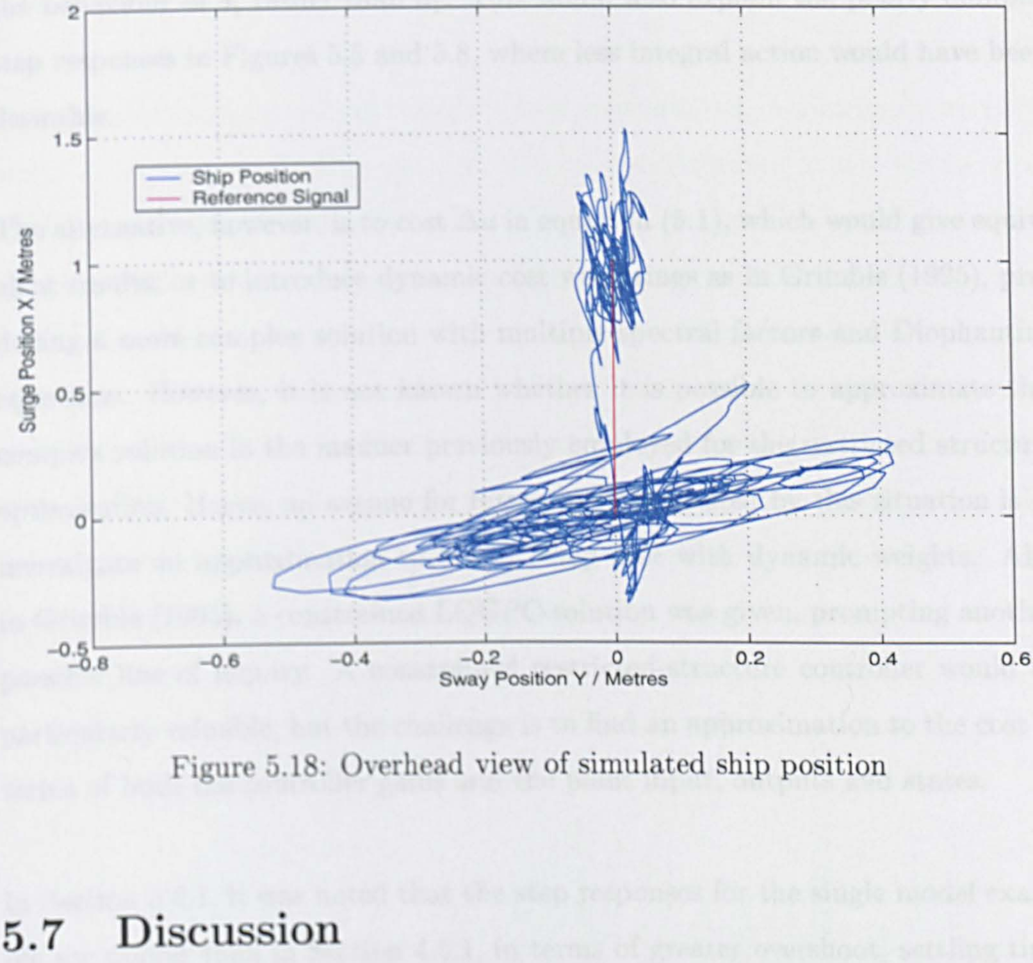

Figure 5.18: Overhead view of simulated ship position

5.7 Discussion

The restricted-structure adaptive predictive controller example above behaves similarly in many ways to the non-predictive controller in Chapter 4. Once more it appears that the peaks in the K_p gain correspond to troughs in the plot of b_1 and vice versa. However, the correspondence between the K_i gain and the a_1 parameter is not clear. In fact, the shape of the K_i gain plot seems to follow K_p but with smaller amplitude. The likely explanation for this stems from the fact that the restricted structure optimisation is carried out for a plant augmented with a fictitious integrator. The controller structure must then be PD, so that the integrator can be moved from the plant after the optimisation to create a PI controller. Hence, when the numerator gain, b_1, is small, larger P and D gains are required to preserve performance and vice versa. In this way, both gains follow the behaviour of b_1 rather than a_1. This would also explain the poorly damped step responses in Figures 5.5 and 5.8, where less integral action would have been desirable.

The alternative, however, is to cost Δu in equation (5.1), which would give equivalent results, or to introduce dynamic cost weightings as in Grimble (1995), producing a more complex solution with multiple spectral factors and Diophantine equations. However, it is not known whether it is possible to approximate this complex solution in the manner previously employed for the restricted structure optimisation. Hence, an avenue for future work suggested by this situation is to investigate an approximation to a multi-step cost with dynamic weights. Also in Grimble (1995), a constrained LQGPC solution was given, prompting another possible line of inquiry. A constrained restricted-structure controller would be particularly valuable, but the challenge is to find an approximation to the cost in terms of both the controller gains and the plant input, outputs and states.

In Section 5.6.1, it was noted that the step responses for the single model example are poorer than in Section 4.6.1, in terms of greater overshoot, settling time and control action. One explanation offered for this was that the theory actually stipulates a filtered white noise reference in both cases, hence the results are not strictly comparable. Therefore, it is very interesting to note that the error between r_{uabs} and $n(s) \times u_{abs}$ in Figure 5.17 has a standard deviation of $0.0196m/s$, yet the the corresponding error standard deviation of Figure 4.18 is $0.0224m/s$, an increase of 14.3%. This is achieved with X_T standard deviation in Figure 5.15 being $130Tonnes$, and the corresponding value for Figure 4.16 being $121Tonnes$, a decrease of only 6.9%. Furthermore, the mean radial error in Figure 5.18 is $0.0381m$, but $0.0579m$ in Figure 4.19. The best explanation for this is that the

r_{uabs} signal more closely approximates the filtered white noise in the optimisation, hence the predictive controller is now able to outperform the standard LQG controller.

Figure 5.10 raises interesting issues regarding robustness, performance and stability, as in Chapter 4. The fact that the best performance is seen for $p_4 = 0.9$ reinforces the idea that the multiple-model adaptive controller gives a combination of the benefits of adaptive and multiple-model optimal control designs in one scheme. However, the poorest performing controller is for the pure adaptive case, hence it is believed that the predictive version is more sensitive to parameter variations and plant-model mismatch than the non-predictive version. Figure 5.19 shows the plot of controller gains when $p_4 = 1$. The variation is clearly much more "violent" than in Figure 5.14, for $p_4 = 0.9$, hence it is reasonable to conclude that this rapid fluctuation actually worsens performance.

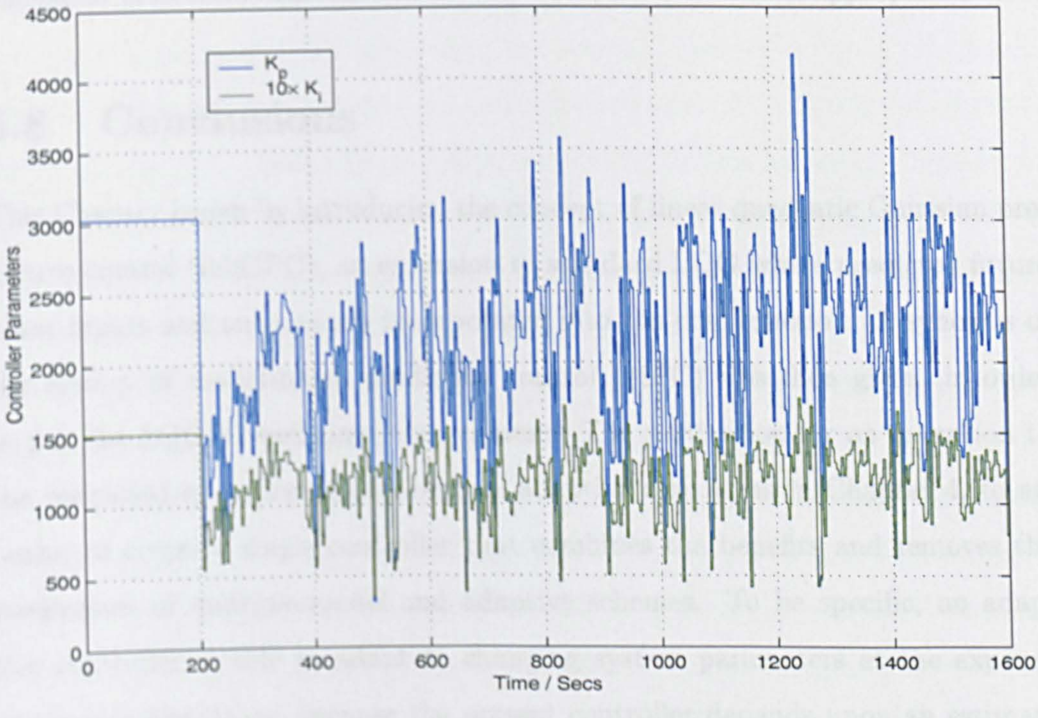

Figure 5.19: Control parameters when $p_4 = 1$

Furthermore, the variation in controller gains for $p_4 = 0.9$ and $p_4 = 1$ in the non-predictive case is considerably smoother than in this Chapter, supporting the belief that the predictive controller is more sensitive to parameter variations. The reason for this is not known, although a possible explanation is that any error in system identification is propagated forward through the prediction horizon. Hence, the controller is trying to control a plant which shows mismatch with the model not only at present, but several steps into the future.

With regard to robustness and stability, the discussion of simultaneous stabilisation in Chapter 4 is also applicable here, as is the suggestion that a proof of convergence of the restricted-structure algorithm would be desirable. Furthermore, a method of allocating realistic probabilities to each of the linear models would be beneficial, discussed in more detail in Chapter 4 also. The idea of comparison with gain scheduling would again be interesting, although in this case, a scheduled predictive controller as in Yu et al. (1992) would be appropriate.

5.8 Conclusions

This Chapter began by introducing the concept of linear quadratic Gaussian predictive control (LQGPC), an extension to standard LQG where predicted future plant inputs and outputs are incorporated into the cost function. A synopsis of the history of model-based predictive control (MPC) was then given, in order to put the LQGPC technique into context. The motivation for an extension to the restricted-structure multiple-model adaptive case is, as in Chapter 4, to attempt to create a single controller that combines the benefits and removes the weaknesses of multiple-model and adaptive schemes. To be specific, an adaptive controller is able to adapt to changing system parameters at the expense of possible instability, because the present controller depends upon an estimate of the current plant model only. However, a multiple-model optimal controller gives greater assurance of stability over a wide range of operating points with the expense of conservative performance. A multiple-model adaptive controller is intended to provide a certain amount of confidence in stability, due to the effect of fixed known models in the optimisation, but with increased performance due to the inclusion of identified system parameters. The fact that the controller is restricted structure and predictive adds to the potentially interesting properties of the solution. The simplicity of the controller provides easy implementation, and the effect of prediction horizons, cost weightings and noise covariance is readily seen in the variation of control parameters. A further motivation for the extension to the restricted-structure multiple-model adaptive case is to compare results with the non-predictive version in Chapter 4.

The LQGPC problem is initially posed in state-space, with the plant and reference models and predictor in this form. An optimisation is carried out over the future input signals, leaving the cost in a form where only the current error and input is penalised. At this point, the system is converted to a polynomial description in order to complete the optimisation using the usual spectral factor and Diophantine equation route. After this, the solution has much in common with the previous Chapter, where the cost is approximated by a matrix-vector form and minimised with respect to the restricted-structure controller parameters. Stacking matrices for each linear model once more gives the multiple model solution, which is augmented with on-line identified parameters to produce the final adaptive algorithm. The example dealt with is the ship DP problem also addressed in Chapter 4, but it is necessary to make some alterations before examining the single model case. LQGPC does not incorporate dynamic cost weights without increasing the complexity of the solution, but it is desirable to include integral action. The plant is therefore augmented with an integrator, so that low frequency errors are penalised. The one-step delay built into the state-space description prevents the disturbance from taking the same form as in Chapter 4, so a delay is tolerated. Additionally, the numerical fragility of Matlab does not allow the reference model to include an integrator, hence the pole is moved to 0.990.

A single model example is detailed where the prediction and input horizons are both two steps long. The restricted structure optimisation terminates after five iterations, giving PI gains to approximate a third-order full-order controller. It is necessary to in fact make the controller structure PD, so that the additional integrator can be moved from the plant description after the optimisation. The closed-loop performance is satisfactory, although not as good as the standard LQG case in Chapter 4. It is noted that increasing the control weighting and decreasing the input weighting increases controller gain across the frequency spectrum and vice versa. The weights for the first time step of the predictions have greatest effect, with very little noticed for the other weightings. The solution is also quite sensitive to output noise, where an increase produces a decrease in controller gain at all frequencies.

This initial single model example is followed by a repeat with longer prediction and input horizons, 30 steps and 5 steps respectively. The main effect on the controller is to reduce low frequency gain and raise high frequency gain. This seems reasonable when it is observed that derivative action gives prediction of future error in a sense and, significantly, acts at high frequencies. The error and control weightings for several steps into the future have more influence on the solution than in the previous example, and the overall performance is slightly better with longer horizons. Notably, the closed-loop step responses are still inferior to the standard LQG case, but it is believed that this is due to the specification of a random reference in the optimisation, but the actual use of a step in the example. Also, for both single model examples, there is very little difference in performance between the full-order and the restricted-structure controller.

The application of the identified plant parameters to the state-space model is detailed, before a full DP example is investigated as in Chapter 4. The 3 degree of freedom simulation from Chapter 2 is used again, but the multiple-model adaptive predictive controller is applied to the surge velocity loop. The other velocity loops take gains from the Penttinen-Koivo method in Chapter 2, and the position loop gains are produced with the Maciejowski technique. Linearised models are taken from three representative operating points, and the fourth model uses parameters identified with recursive least squares. The ship manoeuvre and disturbances are the same as in Chapter 4, and the performance of the restricted-PI-structure controller is evaluated for various probability weightings on each linear model. Unlike in Chapter 4, it is observed that performance is worst for the pure adaptive case, but again best performance is seen at $p_4 = 0.9$. Thus, it is concluded that the adaptive controller is indeed exhibiting better performance than the pure multiple-model case, whilst presumably benefiting from increased robustness due to the fixed linear models. A flaw mentioned in Chapter 4 is a lack of stability or robustness proofs, so at present the robustness benefit is conjectured. The poor performance in the pure adaptive case is apparently due to sensitivity of the predictive controller to parameter variations and plant-model mismatch. This is supported by the great variation in PI gains in the $p_4 = 1$ example.

One very positive outcome from the full DP example is improved performance over the standard LQG controller in Chapter 4. The error standard deviation of the controlled absolute velocity is 14.3% greater with standard LQG, but the control action is only 6.9% smaller. The best explanation for this is that the reference signal approximates the filtered white noise in the optimisation fairly well, in contrast with the single model examples where the reference employed is a step. Hence, the predictive nature of LQGPC is better exploited. A significant contribution of this Chapter is therefore a demonstration that restricted-structure adaptive LQGPC can lead to better performance than equivalent LQG control.

The flaw in this technique, in addition to the lack of stability or robustness proofs for multiple models and unproven convergence of the restricted-structure algorithm, is an inability to address constraints, unlike many other MPC controllers. A suggestion for further work is to look for an approximation to the cost in terms of both the controller gains and the plant input, outputs and states. Hopefully, a QP solver could then be employed to perform the constrained minimisation. Another suggestion is to look for a method of incorporating dynamic weights into the restricted-structure solution, so that the augmented plant is not required and the input weighting is truly on u, rather than effectively on Δu. As with Chapter 4, simultaneous stabilisation theory is suggested as an approach to the stability and robustness problem, and performance improvement may be achieved by allocating realistic probability weightings for the fixed models in some manner. It is also suggested that a scheduled predictive controller could be used for comparison with the results in this Chapter, as there are similarities between the two approaches in that a set of linearised models is required in both cases.

Chapter 6

Multivariable Sandwich Nonlinear System Control from Time-Varying Systems Theory

This Chapter considers the control of multivariable nonlinear "sandwich" systems, consisting of a linear discrete-time dynamic block with input and output nonlinear functions. The solution of a time-varying linear optimal polynomial control problem is presented, which involves time-varying equivalents of spectral factorisation and Diophantine equations. The sandwich nonlinear plant is manipulated into time-varying format by assuming that the nonlinear functions are in a particular rational form. By freezing the time-varying description at each sampling instant, the infinite-time problem becomes tractable, hence a solution is obtained by calculating the controller online using the nonlinearity-dependent polynomials. This controller is simple to implement and a 2×2 system example is simulated, which yields results demonstrating that the achievable performance is superior to a fixed linear controller.

6.1 Introduction

In previous Chapters, plant nonlinearities have been dealt with by designing the controller at a particular linearised operating point (Chapter 2), or by optimising across a set of linearised models (Chapters 4 and 5). In both cases, a Taylor expansion approximation to the true system is used, allowing mature and well-understood linear control techniques to be applied. The pre-eminence of linear analysis and control methods is based on a single principle, that of superposition, now illustrated by a linear operator, f. Given two inputs to the operator, u_1 and u_2, and a real scalar, α, then by superposition:

$$f(u_1) + f(u_2) = f(u_1 + u_2)\ ,\ f(\alpha u_1) = \alpha f(u_1) \tag{6.1}$$

This single principle simplifies a great deal of mathematical systems theory, but when it fails, the system is classified as nonlinear and the supporting theory is fragmented and far from complete. The superposition principle means that it is possible to state a general solution for linear time-invariant systems described by differential or difference equations, but this is not so in the nonlinear case. In general, it is very difficult or impossible to solve the dynamic equations of a nonlinear system, hence numerical methods must be used to give a particular solution. Control engineers would like to have generally applicable design techniques for nonlinear plants, but at present only approximate or restricted approaches have been developed. The techniques available to deal with nonlinear systems range from making simple approximations to nonlinearities, such as the Describing Function method, to quite mathematically intensive methods, such as Geometric Nonlinear Control theory and Lie brackets.

Although it is not possible in general to find the solution of nonlinear dynamic equations, methods do exist for establishing stability. The describing function method was developed simultaneously in different countries during the 1940s, see

Atherton (1982), as a method of approximating the response of nonlinear functions to a sinusoidal input. The output of the function is represented by a Fourier series which is truncated to the fundamental frequency terms. This is justified by the observation that physical systems are always low pass, so high frequencies produce negligible output. For a static (memoryless) nonlinearity, a single fundamental Fourier series coefficient describes an amplitude-dependent gain, the describing function, acting on the input to produce the output sinusoid from the function. If the nonlinearity is dynamic, then both fundamental Fourier series coefficients are used to give a gain dependent on both amplitude and frequency, with attendant phase shift. A Nyquist plot of both describing function and linear plant is then able to predict fairly accurately when limit cycles will occur. The technique is limited to systems described by a cascaded nonlinearity and linear transfer function and is not entirely accurate, but has been used widely in industry due to its simplicity and transparency.

Two other early stability techniques are the Small Gain Theorem, see Zames (1966a), and the related Circle Criterion of Zames (1966b). Essentially, the Small Gain Theorem states that, if the elements of a feedback loop are all stable and the product of the gains of the elements is always less than one, the closed loop will be stable. The Circle Criterion is stated for a linear transfer function cascaded with a nonlinearity in the loop, where the nonlinearity belongs to a particular "sector", $[\alpha, \beta]$. The closed-loop is stable when the Nyquist plot of the linear transfer function does not intersect or encircle the disc centred on the real axis and passing through the points $(-1/\alpha, 0)$ and $(-1/\beta, 0)$.

Perhaps the most powerful approaches to analysing nonlinear system stability are known as Lyapunov's direct and indirect methods, developed from Lyapunov (1892), but unknown in the West until the 1960s. The direct method is generally used in favour of the indirect method and, in essence, depends on showing that the energy in a physical system is always being dissipated. The method can be applied to an abstract mathematical system, where energy is not defined, but this serves to illustrate one of the drawbacks of Lyapunov theory in general. A Lyapunov function, V, of the system state, x, must be selected that is positive-definite, continuous in x, that equals zero when $x = 0$, and so that the change in $V(x)$ is negative-definite. Finding such a function may be difficult, but with a physical system, the energy function is always a good candidate. The method is not so useful for control design, however, as no information on performance can be obtained other than stability. Hence, the Lyapunov function could indicate stability, but the system could take a huge amount of time to settle. Some progress towards addressing this problem via Recursive Lyapunov Design has been made in Freeman and Kokotović (1993), however, and the problem of robustness is addressed using Lyapunov Redesign, as in Khalil (1992).

Another well-known technique, for analysing second-order systems only, is phase-plane analysis, originally developed by Poincaré (1892). This method proceeds by plotting the behaviour of the two system states against one another on a graph and classifying the various "fixed points", where both states are stationary. This classification is achieved by linearising the nonlinear equations at the fixed points and looking at the eigenvalues of the corresponding Jacobian matrix. Knowing whether the system is stable or unstable in a small region around the fixed points is a significant result in the theory of nonlinear systems. The drawback of using the phase-plane is that, while useful for stability analysis, it is not easy to use for actual control design. Hence, a method known as variable structure sliding-mode control has been developed, see Utkin (1992) or Edwards and Spurgeon (1998), so called because the feedback control law varies by switching between gains in order to maintain the state (phase) trajectory on a particular surface. The state trajectory then "slides" along this surface for all time, under the influence of the switching controller. This technique has been quite successful, but the switching can excite undesirable oscillation, and the design is rather involved.

The application of results in differential geometry has led to the powerful but complex Geometric Nonlinear Control theory, exemplified by Sussman and Jurdjevic (1972) and Jurdjevic and Sussman (1972). Unfortunately, the heavy mathematical content of this theory places the results beyond the grasp of the average engineer and is not of great practicality. A rather more pragmatic use of differential-geometric methods is in the use of feedback on a nonlinear system to produce linear behaviour, known as feedback linearisation. This is achieved by a change of coordinates with nonlinear feedback, see Vidyasagar (1993) or Marino and Tomei (1995) for example, but a limitation of this technique is that it is only applicable to a limited class of nonlinear systems. Also, stability is not assured when there are unstable zero dynamics, analogous to linear nonminimum phase systems.

Among the more practical nonlinear control techniques are Neural Control and Fuzzy Logic Control. Chapter 3 has already dealt with the application of a neural network to system identification, and a discussion of Neural Control is found in Section 3.6. It will suffice to say here that Neural Control is based on the ability of a neural network to learn an arbitrary nonlinear characteristic, which can then be used with feedback to compensate a nonlinear plant in some way. Fuzzy Control emerged from the description of human, heuristic knowledge and reasoning with Fuzzy Sets, see Chen and Pham (2001). To elaborate, a normal "crisp" set contains elements with particular characteristics, and every element not within that set does not have those characteristics. In a "fuzzy" set, the elements have gradations of membership of the set, such that concepts such as "fairly cold" or "quite hot" can be described rather than simply "cold" and "hot". Using these human descriptions of system variables, it is then possible to build an "If-Then" rule base and make a decision on the best control signal to apply under the current circumstances. The advantages of Fuzzy Control are that expert knowledge can be incorporated into a controller, the overall operation can be understood in terms of human language, and a complex, interacting plant can be controlled without extensive modelling. The disadvantages are that the standard analysis tools of feedback control, such as frequency response and stability margins, are not available, and that the technique relies on the human to cover every possible system state with a corresponding rule. Fuzzy Control would not be suitable for safety-critical systems such as an aircraft or nuclear power station, for example.

Given the difficulties and limitations associated with nonlinear control, either through the necessary use of approximations, complex mathematics or lack of practicality, the realistic aim of this Chapter is to present a nonlinear control technique that is practical, but has a rigorous basis, with approximations introduced as the theory is developed. Section 6.2 presents the solution of a linear time-varying polynomial control problem. This is an extension to the optimal time-invariant polynomial problems addressed in Kucera (1979). The analysis and synthesis of time-varying linear state-space systems is quite well understood, see D'Angelo (1970) for example, so it is worthwhile looking at the polynomial equivalent. The interesting property of a time-varying linear description is that superposition, as in equation (6.1), still holds, but instead the valuable property of shift-invariance during convolution is lost. As a consequence, the frequency domain results of time-invariant linear systems no longer hold, removing a great deal of useful supporting control theory. The advantage of a time-varying description, however, is that it can be used to describe a nonlinear system as a linear system with time-varying coefficients.

In Section 6.3, the time-varying optimal control solution is adapted to the special case of so-called "sandwich" system models, where a nonlinear plant is represented by a linear system with input and output nonlinearities, the "outer sandwich".

This representation is a natural extension of the Hammerstein model, a linear system with nonlinear input block, and the Wiener model, a linear system with nonlinear output block. A recent book, Taware and Tao (2003), deals exclusively with "sandwich" nonlinear dynamic systems, where the common sensor and actuator nonlinearities of interest include dead-zone, hysteresis, saturation, friction and backlash. The hybrid framework for control used throughout the book consists of a discrete-time inner loop and a continuous-time outer loop, combined with a nonlinearity inverse to cancel the nonlinear effect. The authors work through designs incorporating an exact inverse, an adaptive inverse for an unknown nonlinearity, and a neural controller using networks to represent the nonlinearity and as a compensator. Simulations are utilised to illustrate the effectiveness of the hybrid control designs. Subsequently, friction compensation is addressed for a friction nonlinearity sandwiched between two linear dynamic blocks, an "inner sandwich", where the control methodologies are Model Reference Adaptive Control (MRAC) and feedback linearisation. Control of sandwich systems with actuator failures is examined, before a MIMO example of gun turret control with sandwiched backlash. Again, simulation studies demonstrate successful control when using the developed sandwich system control techniques.

In Section 6.3, each non-linearity is represented by a function of the input multiplied by the input itself, hence the "outer sandwich" representation allows a plant to be expressed as a transfer function matrix that varies with time due to the variation of the non-linearity. The various polynomials in the time-varying theory of Section 6.2 are then substituted with the time-varying polynomials arising from the nonlinear model. The statement of the time-varying solution actually requires the solution of time-varying difference equations over infinite-time, so it is then necessary to "freeze" the plant description at a particular moment in time to yield a tractable solution. Thus, the controller must be computed online to track variations in the nonlinearities, prompting comparison with adaptive or gain-scheduled controllers. The controller here is strictly neither, as the plant model is assumed known and the control law is not selected from a pre-computed set. Instead, the values of the nonlinearities are computed at time, $t-1$, for use in the control computation at time, t.

The theory and an example are given in Grimble (2002) in the SISO case, and the contribution of this Chapter is to extend the theory to the multivariable case and demonstrate a successful MIMO application. Unfortunately, the numerical problems with Matlab highlighted earlier in Chapter 4, Section 4.5, prevent the use of plant order greater than one. Thus, the ship DP control theme cannot be continued and the example is a 2-input 2-output system with deadzone input nonlinearities and backlash output nonlinearities. Simulation results suggest that the controller performance is superior to a fixed design, and directions for further work are given in Section 6.6.

6.2 Time-varying linear control problem formulation

The state-space representation of time-varying systems is familiar, for example in D'Angelo (1970). The equations for the multivariable system shown in Figure 6.1 can be written in polynomial form, however, as:

$$\begin{aligned} y(t) &= (Wu)(t) + d(t) \\ &= W(t, z^{-1})u(t) + d(t) \\ &= A(t, z^{-1})^{-1}B(t, z^{-1})u(t) + d(t) \qquad (6.2) \end{aligned}$$

$$r(t) = W_r(z^{-1})\varsigma(t) = A_e(z^{-1})^{-1}E(z^{-1})\varsigma(t) \qquad (6.3)$$

$$d(t) = W_d(z^{-1})\xi(t) = A_c(z^{-1})^{-1}C_d(z^{-1})\xi(t) \tag{6.4}$$

$$e(t) = r(t) - y(t) \tag{6.5}$$

$$u(t) = (C_0 e)(t) = C_0(t, z^{-1})e(t) \tag{6.6}$$

where z^{-1} is the unit-delay operator with the property $z^{-1}x(t) = x(t-1)$. Note that the signals are vectors, where $u(t) \in \mathbb{R}^l$, $y(t) \in \mathbb{R}^m$, $\zeta(t) \in \mathbb{R}^m$ and $\xi(t) \in \mathbb{R}^m$. $\zeta(t)$ and $\xi(t)$ are assumed, without loss of generality, to be uncorrelated white noise of unity variance and zero mean.

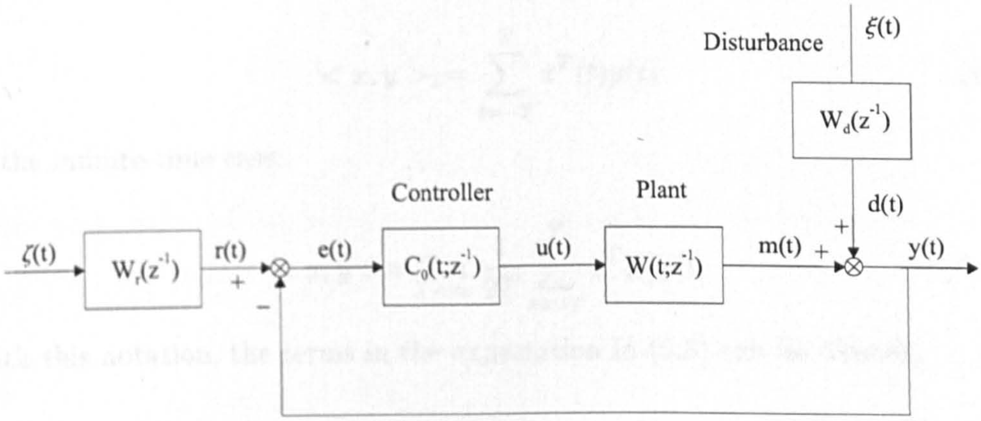

Figure 6.1: Time-varying system regulator

This system description is similar to the time-invariant case, the only difference being that transfer function matrices W and C_0 are functions of time. In the time-invariant case, a suitable cost function would take the form:

$$J = \lim_{T\to\infty}\{\frac{1}{2T}E\{\sum_{t=-T}^{T} e^T(t)H_q^T H_q e(t) + u^T(t)H_r^T H_r u(t)\}\} \tag{6.7}$$

The time-varying analogy to this is provided in Grimble and Johnson (1988):

$$J = \lim_{T\to\infty}\{\frac{1}{2T}E\{\sum_{t=-T}^{T}(H_q e)^T(t)(H_q e)(t) + (H_r u)^T(t)(H_r u)(t)\}\} \tag{6.8}$$

where H_q and H_r are diagonal dynamic weighting operators in z^{-1}. The operator H_q is the weighting on the system output and will often be rational so that an integrator may be included. In this case, $H_q = B_q A_q^{-1}$.

6.3 Time-varying linear control problem solution

To proceed from this point, the notation to be used must be explained. The inner product of two vectors is denoted as:

$$< x, y >_T = \sum_{t=-T}^{T} x^T(t)y(t) \tag{6.9}$$

In the infinite-time case:

$$< x, y > = \lim_{T\to\infty}\frac{1}{2T}\sum_{t=-T}^{T} x^T(t)y(t) \tag{6.10}$$

With this notation, the terms in the expectation in (6.8) can be written:

$$I_T = < H_q e, H_q e >_T + < H_r u, H_r u >_T \tag{6.11}$$

Substituting equations (6.2) and (6.5) into (6.11) gives:

$$\begin{aligned} I_T &= < H_q r - H_q W u - H_q d, H_q r - H_q W u - H_q d >_T + < H_r u, H_r u >_T \\ &= < H_q f - H_q W u, H_q f - H_q W u >_T + < H_r u, H_r u >_T \end{aligned} \tag{6.12}$$

where $f = r - d$. It is shown in Appendix D that:

$$\begin{aligned} < x(t), (Zu)(t) >_T &= < x(t), Z(t, z^{-1})u(t) >_T \\ =< (Z^*x)(t), u(t) >_T &= < Z^*(t, z^{-1})x(t), u(t) >_T \end{aligned} \tag{6.13}$$

where the adjoint operator $Z^*(t, z^{-1}) = Z^T(t, z)$. Therefore:

$$\begin{aligned} I_T &= < u, (W^*H_q^*H_qW + H_r^*H_r)u >_T - < f, H_q^*H_qWu >_T \\ &\quad - < u, W^*H_q^*H_qf >_T + < f, H_q^*H_qf >_T \end{aligned} \tag{6.14}$$

Let $Y_c^*Y_c = W^*H_q^*H_qW + H_r^*H_r = W^*Q_cW + R_c$, where $Q_c = H_q^*H_q$ and $R_c = H_r^*H_r$. Then, defining $Y_c = D_cA_1^{-1}$ and $H_qW = B_1A_1^{-1}$ produces the "operator spectral factor" equation:

$$D_c^*D_c = B_1^*B_1 + A_1^*R_cA_1 \tag{6.15}$$

The quotation marks are used to signify the fact that, although the expression takes the form of a standard time-invariant spectral factor equation, this interpretation is not literal due to the time-varying behaviour.

The cost expression may be restated:

$$I_T =< u, Y_c^*Y_cu >_T - < f, Q_cWu >_T - < u, W^*Q_cf >_T + < f, Q_cf >_T \tag{6.16}$$

which is equivalent to:

$$\begin{aligned} I_T &= < Y_cu - Y_c^{*-1}W^*Q_cf, Y_cu - Y_c^{*-1}W^*Q_cf >_T \\ &\quad + < f, (Q_c - Q_cW(Y_c^*Y_c)^{-1}W^*Q_c)f >_T \end{aligned} \tag{6.17}$$

using a completing-the-squares argument. The second inner product is independent of the control action, u, and is of no further importance to the cost minimisation. The first inner product requires further manipulation, as given below:

$$\begin{aligned} Y_c u - Y_c^{*-1} W^* Q_c f &= Y_c u - Y_c^{*-1} W^* Q_c (Wu + e) \\ &= D_c A_1^{-1} u - D_c^{*-1} A_1^* W^* H_q^* H_q (Wu + e) \\ &= D_c A_1^{-1} u - D_c^{*-1} B_1^* H_q (Wu + e) \qquad (6.18) \end{aligned}$$

Substituting the "operator spectral factor" into the above yields:

$$\begin{aligned} Y_c u - Y_c^{*-1} W^* Q_c f &= D_c^{*-1} (D_c^* D_c - B_1^* B_1) A_1^{-1} u - D_c^{*-1} B_1^* H_q e \\ &= D_c^{*-1} (A_1^* R_c u - B_1^* H_q e) \qquad (6.19) \end{aligned}$$

A true spectral factor is needed in the solution below. To derive it, first define $f = r - d = Y_f \varepsilon = A_c^{-1} D_f \varepsilon$, where ε is zero mean white noise with unity covariance. From Grimble and Johnson (1988), $\Phi_{yy}(z^{-1}) = W(z^{-1}) \Phi_{uu}(z^{-1}) W^*(z^{-1})$ when $y(z^{-1}) = W(z^{-1}) u(z^{-1})$. Therefore:

$$\begin{aligned} Y_f Y_f^* &= W_d W_d^* + W_r W_r^* \\ \Rightarrow D_f D_f^* &= C_d C_d^* + E E^* \qquad (6.20) \end{aligned}$$

Substituting the "operator Diophantine equations":

$$z^{-g} D_c^* G_0 + F_0 A_2 = z^{-g} B_1^* D_{f2} \qquad (6.21)$$

$$z^{-g} D_c^* H_0 - F_0 B_3 = z^{-g} A_1^* R_c D_{f3} \qquad (6.22)$$

into equation (6.19), obtain:

$$\begin{aligned} & Y_c u - Y_c^{*-1} W^* Q_c f \\ = \; & D_c^{*-1}(D_c^* H_0 - z^g F_0 B_3) D_{f3}^{-1} u - D_c^{*-1}(D_c^* G_0 + z^g F_0 A_2) D_{f2}^{-1} H_q e \\ = \; & (H_0 D_{f3}^{-1} u - G_0 D_{f2}^{-1} H_q e) - D_c^{*-1} z^g F_0 (A_2 D_{f2}^{-1} H_q e + B_3 D_{f3}^{-1} u) \\ = \; & (H_0 D_{f3}^{-1} u - G_0 D_{f2}^{-1} H_q e) - D_c^{*-1} z^g F_0 D_f^{-1} (Ae + Bu) \\ = \; & (H_0 D_{f3}^{-1} u - G_0 D_{f2}^{-1} H_q e) - D_c^{*-1} z^g F_0 D_f^{-1} A f \\ = \; & (H_0 D_{f3}^{-1} u - G_0 D_{f2}^{-1} H_q e) - D_c^{*-1} z^g F_0 D_f^{-1} A A_c^{-1} D_f \varepsilon \end{aligned} \tag{6.23}$$

using definitions $D_f^{-1} A = A_2 D_{f2}^{-1} H_q$, $D_f^{-1} B = B_3 D_{f3}^{-1}$.

The signal $\varepsilon(t)$ is white noise, meaning that the above expression can be split into two statistically independent components:

$$\phi_1 = (H_0 D_{f3}^{-1} u - G_0 D_{f2}^{-1} H_q e) \; , \; \phi_2 = D_c^{*-1} z^g F_0 D_f^{-1} A A_c^{-1} D_f \varepsilon \tag{6.24}$$

Returning to equation (6.17), I_T can now be expressed:

$$\begin{aligned} I_T \; = \; & < \phi_1 - \phi_2, \phi_1 - \phi_2 >_T \\ & + < f, (Q_c - Q_c W (Y_c^* Y_c)^{-1} W^* Q_c) f >_T \end{aligned} \tag{6.25}$$

Substituting this back into the cost function, (6.8), and taking advantage of the statistical independence:

$$\begin{aligned} J \; = \; & \lim_{T \to \infty} \{ \frac{1}{2T} E\{I_T\} \} \\ = \; & E\{< \phi_1, \phi_1 > + < \phi_2, \phi_2 >\} + T_1 \end{aligned} \tag{6.26}$$

where $T_1 = E\{< f, (Q_c - Q_c W (Y_c^* Y_c)^{-1} W^* Q_c) f >\}$.

As previously remarked, T_1 does not influence the cost minimisation. This also applies to the second inner product above:

$$\begin{aligned} T_2 &= E\{< \phi_2, \phi_2 >\} \\ &= E\{< D_c^{*-1} z^g F_0 D_f^{-1} A A_c^{-1} D_f \varepsilon, D_c^{*-1} z^g F_0 D_f^{-1} A A_c^{-1} D_f \varepsilon >\} \end{aligned} \tag{6.27}$$

leaving only the first inner product in the cost. Hence, the optimal control law is determined by setting the first inner product to zero. This produces:

$$u = D_{f3} H_0^{-1} G_0 D_{f2}^{-1} H_q e \tag{6.28}$$

In summary, the time-varying control problem involves solving an "operator spectral factor" and a true spectral factor equation:

$$D_c^* D_c = B_1^* B_1 + A_1^* R_c A_1 \tag{6.29}$$

$$D_f D_f^* = C_d C_d^* + E E^* \tag{6.30}$$

and two "operator Diophantine equations":

$$z^{-g} D_c^* G_0 + F_0 A_2 = z^{-g} B_1^* D_{f2} \tag{6.31}$$

$$z^{-g} D_c^* H_0 - F_0 B_3 = z^{-g} A_1^* R_c D_{f3} \tag{6.32}$$

with F_0 of smallest degree, to obtain G_0 and H_0 in (6.28).

6.4 Non-linear problem

The time-varying linear system formulation in Section 6.2 retains a desirable property of time-invariant linear systems, namely superposition at a given instant in time. However, it is also able to describe some nonlinear systems, a particular class of which will now be described, known as "sandwich" systems. Suppose that the nonlinear system is modelled by a linear block, $G(z^{-1})$, "sandwiched" between nonlinear functions acting on the input and output, as in Figure 6.2.

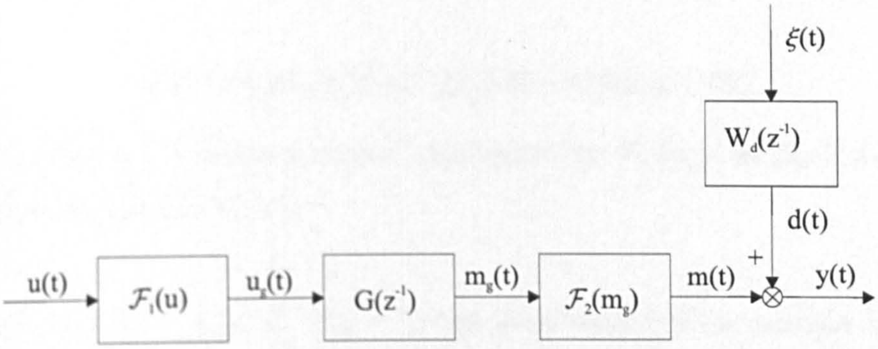

Figure 6.2: "Sandwich" system with input and output non-linearities

A time-varying system description of this, corresponding to equation (6.2), is:

$$\begin{aligned} y(t) &= (Wu)(t) + d(t) = \mathcal{F}_2(G(z^{-1})\mathcal{F}_1(u(t))) + d(t) \\ &= W(t, z^{-1})u(t) + d(t) = [f_2(m_g(t))G(z^{-1})f_1(u(t))]u(t) + d(t) \quad (6.33) \end{aligned}$$

assuming the nonlinear functions to be known and of the form:

$$u_g(t) = \mathcal{F}_1(u(t)) = f_1(u(t))u(t) \quad (6.34)$$

$$m(t) = \mathcal{F}_2(m_g(t)) = f_2(m_g(t))m_g(t) \quad (6.35)$$

where f_2 is diagonal. This representation is able to describe a range of static and dynamic nonlinearities, commonly including deadzone, saturation, backlash, hysteresis, quantisation, relay, friction and piecewise nonlinearities. Some care must be taken, however, as it is possible for the nonlinearity to give a non-zero output to a zero input. In practice, some limits are required on f_1 and f_2, as demonstrated in the simulation later.

The non-linear input element, f_1, is a function of the current control input. However, in an optimal control problem, the $u(t)$ signal is unknown until after the control law is calculated. Thus, the assumption is made that u varies sufficiently slowly that variation in f_1 is negligible between time t and $t-1$, yielding:

$$y(t) = [f_2(m_g(t))G(z^{-1})f_1(u(t-1))]u(t) + d(t) \tag{6.36}$$

It is not necessary to make a similar assumption for $\mathcal{F}_2(m_g)$, as there is a delay of at least one step in $G(z^{-1})$.

Defining $G(z^{-1}) = A_g(z^{-1})^{-1}B_g(z^{-1})$, the time-varying plant transfer function may now be stated:

$$\begin{aligned} W(t, z^{-1}) &= A(t, z^{-1})^{-1}B(t, z^{-1}) \\ &= f_2(m_g(t))A_g(z^{-1})^{-1}B_g(z^{-1})f_1(u(t-1)) \end{aligned} \tag{6.37}$$

where $A(t, z^{-1}) = A_g(z^{-1})f_2^{-1}(m_g(t))$ and $B(t, z^{-1}) = B_g(z^{-1})f_1(u(t-1))$.

Having arrived at the optimal time-varying solution and stated a sandwich non-linear system in time-varying form, it remains to be seen how this technique can be applied to control problems in practice. The "operator spectral factor" and "operator Diophantine equations" in equations (6.29), (6.31) and (6.32) are time-varying, multivariable, nonlinear difference equations derived from an infinite-time cost function. There is no known general method for solving such difference equations, but even if there was, the solution could only be found over a finite time. Hence, to render this solution tractable and practical, it is necessary to assume that the plant is "frozen" at the instant in time when the optimisation is carried out.

Now consider how the spectral factors and Diophantine equations in the earlier optimisation will be affected by the assumptions about the nonlinear plant. In Section 6.3, the statement $H_qW = B_1A_1^{-1}$ was made. Substituting (6.37) into this equation, obtain:

$$H_q(z^{-1})f_2(m_g(t))G(z^{-1})f_1(u(t-1)) = B_1(t,z^{-1})A_1(t,z^{-1})^{-1} \tag{6.38}$$

Recalling that H_q and f_2 are diagonal and "frozen", not unreasonable for slow variations in m_g, H_q and f_2 can commute to obtain:

$$B_1(t,z^{-1})A_1(t,z^{-1})^{-1} = f_2(m_g(t))H_q(z^{-1})G(z^{-1})f_1(u(t-1)) \tag{6.39}$$

At this point, the right coprime form of the weighted linear plant model is introduced:

$$H_q(z^{-1})G(z^{-1}) = B_{1g}(z^{-1})A_{1g}(z^{-1})^{-1} \tag{6.40}$$

and substituted into equation (6.39), producing new definitions of B_1 and A_1:

$$\begin{aligned} B_1(t,z^{-1}) &= f_2(m_g(t))B_{1g}(z^{-1}) && (6.41) \\ A_1(t,z^{-1}) &= f_1(u(t-1))^{-1}A_{1g}(z^{-1}) && (6.42) \end{aligned}$$

Hence the "spectral factor" equation (6.29) follows:

$$\begin{aligned} D_c^*D_c &= B_1^*B_1 + A_1^*R_cA_1 \\ &= B_{1g}^*f_2^*(m_g)f_2(m_g)B_{1g} + A_{1g}^*f_1^*(u)^{-1}R_cf_1(u)^{-1}A_{1g} && (6.43) \end{aligned}$$

Spectral factor (6.30) is unchanged:

$$D_fD_f^* = C_dC_d^* + EE^* \tag{6.44}$$

although the assumption is made that $D_f = C_d = E = I$, in order to simplify the "Diophantine equations" below.

In Section 6.3, it was stated that $D_f^{-1}A = A_2D_{f2}^{-1}H_q$ and $D_f^{-1}B = B_3D_{f3}^{-1}$. Substituting A and B from above and remembering the assumption that H_q and f_2 commute:

$$\begin{aligned} A_2 D_{f2}^{-1} &= D_f^{-1} A H_q^{-1} \\ &= D_f^{-1} A_g f_2^{-1} H_q^{-1} \\ &= D_f^{-1} A_g H_q^{-1} f_2^{-1} \\ &= D_f^{-1} A_g A_q B_q^{-1} f_2^{-1} \qquad (6.45) \\ B_3 D_{f3}^{-1} &= D_f^{-1} B_g f_1 \qquad (6.46) \end{aligned}$$

Assuming that $B_q = I$ and defining $A_2 = A_g A_q$ and $B_3 = B_g$, therefore:

$$D_{f2} = f_2 \ , \ D_{f3} = f_1^{-1} \qquad (6.47)$$

Substituting the new expressions for A_2, B_3, A_1, B_1, D_{f2} and D_{f3} into equations (6.31) and (6.32) produces the "Diophantine equations" for the non-linear system:

$$\begin{aligned} z^{-g} D_c^* G_0 + F_0 A_g A_q &= z^{-g} B_{1g}^* f_2^* f_2 \qquad (6.48) \\ z^{-g} D_c^* H_0 - F_0 B_g &= z^{-g} A_{1g}^* f_1^{*-1} R_c f_1^{-1} \qquad (6.49) \end{aligned}$$

Upon substitution into (6.28), the controller for the non-linear system is:

$$u = C_0 e = f_1^{-1} H_0^{-1} G_0 f_2^{-1} A_q^{-1} e \qquad (6.50)$$

In summary, the non-linear control problem involves solving "spectral factors":

$$D_c^* D_c = B_{1g}^* f_2^*(m_g) f_2(m_g) B_{1g} + A_{1g}^* f_1^*(u)^{-1} R_c f_1(u)^{-1} A_{1g} \qquad (6.51)$$

$$D_f D_f^* = C_d C_d^* + E E^* \qquad (6.52)$$

and two "Diophantine equations"

$$z^{-g} D_c^* G_0 + F_0 A_g A_q = z^{-g} B_{1g}^* f_2^*(m_g) f_2(m_g) \qquad (6.53)$$

$$z^{-g}D_c^*H_0 - F_0B_g = z^{-g}A_{1g}^*f_1^*(u)^{-1}R_cf_1^*(u)^{-1} \tag{6.54}$$

with F_0 of smallest degree, to obtain G_0 and H_0 in (6.50). The assumption that the plant is frozen at a given instant of time allows the "spectral factor" equation and "Diophantine equations" to be solved by the normal method for time-invariant systems. If m_g and u are slowly varying these assumptions are not entirely unreasonable. Furthermore, the solution is simplified by the assumption that $C_d = E = I$ and, letting $A_c = (1 - z^{-1})I$, this is quite reasonable as the reference and disturbance become integrated white noise.

6.4.1 Illustrative SISO example

Before a multivariable simulation example, it is informative to consider the effect on C_0 in the frequency domain of changing f_1 and f_2. For this, a single-input single-output example is considered, where the linear plant transfer function, $G(z^{-1})$, is:

$$G(z^{-1}) \;=\; A_g(z^{-1})^{-1}B_g(z^{-1}) = \frac{z^{-1}(1+0.95z^{-1})}{(1-0.8z^{-1})(1-0.9z^{-1})} \tag{6.55}$$.

The disturbance and reference models are:

$$\begin{aligned} W_d(z^{-1}) &= A_c(z^{-1})^{-1}C_d(z^{-1}) \\ = W_r(z^{-1}) &= A_c(z^{-1})^{-1}E(z^{-1}) = \frac{1}{(1-z^{-1})} \end{aligned} \tag{6.56}$$

The error and control weightings are:

$$Q_c = H_q^*H_q = A_q^{*-1}B_q^*B_qA_q^{-1} = \frac{1}{(1-z^{-1})(1-z)} \tag{6.57}$$

$$R_c = H_r^*H_r = 25(1-z^{-1})(1-z) \tag{6.58}$$

The weighted linear plant model is:

$$\begin{aligned} H_qG &= B_qA_q^{-1}A_g^{-1}B_g = B_{1g}A_{1g}^{-1} \\ &= \frac{1+0.95z^{-1}}{(1-z^{-1})(1-0.8z^{-1})(1-0.9z^{-1})} \end{aligned} \tag{6.59}$$

and the linear controller, from equation (6.50), is:

$$\begin{aligned} C_0 &= H_0^{-1}G_0A_q^{-1} \\ &= \frac{0.581-0.779z^{-1}+0.278z^{-2}}{(1-z^{-1})(1+0.186z^{-1})} \end{aligned} \tag{6.60}$$

The complete expression for C_0 is $C_0 = f_1^{-1}H_0^{-1}G_0f_2^{-1}A_q^{-1}$, which suggests that the nonlinear controller simply consists of a fixed linear part cascaded with an inversion of the nonlinearities, f_1 and f_2. "Freezing" the time-varying linear system description at each sample step to perform the controller optimisation does effectively linearise the nonlinear plant, but note that G_0 and H_0 do not vary linearly with f_1 and f_2. To illustrate this, Figure 6.3 shows the frequency response of C_0, and Figure 6.4 shows the frequency response of WC_0 for different f_1 when $f_2 = 1$. Similar plots can be generated for f_2 when $f_1 = 1$. When one of the f's is small, not only does the overall gain rise, but the controller is more phase lagging at high frequencies. The opposite is seen when one of the f's is large, and the effect is simply accentuated when the f's are simultaneously small or large. From the open-loop Bode plot in Figure 6.4, it can be seen that the stability margins decay with increasing f_1. Therefore, a closed-loop system using the technique in this Chapter will be more prone to oscillation or instability when a nonlinearity is operating in a high gain region. The consequences of this and of unlimited f's will be discussed further later, with reference to the multivariable simulation example.

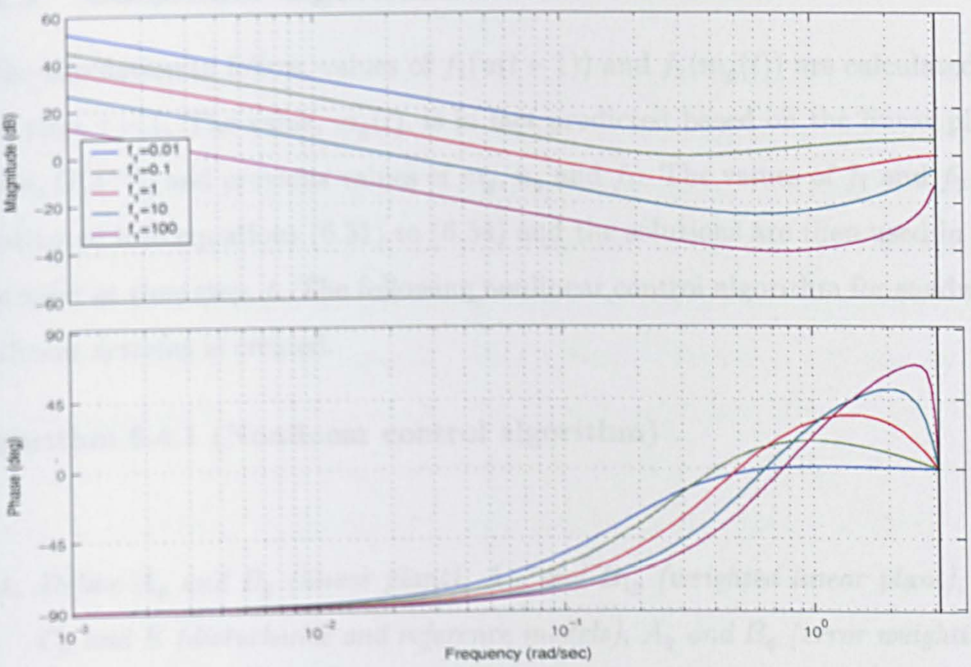

Figure 6.3: Bode plot of C_0 with different f_1 ($f_2 = 1$)

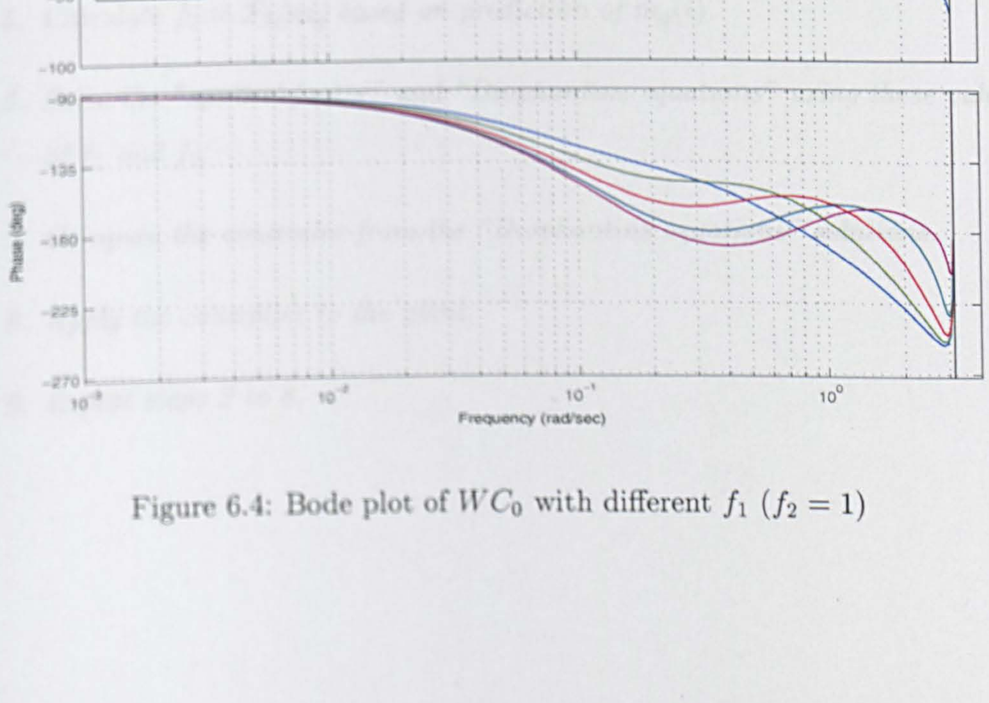

Figure 6.4: Bode plot of WC_0 with different f_1 ($f_2 = 1$)

6.4.2 Controller algorithm

In the simulation to follow, values of $f_1(u(t-1))$ and $f_2(m_g(t))$ are calculated at time step, $t-1$. The value, $m_g(t)$, is in fact predicted based on the linear plant model, $G(z^{-1})$, and previous values of m_g, u_g and f_1. The values of f_1 and f_2 are substituted into equations (6.51) to (6.54) and the solutions are then used in the controller at time step, t. The following nonlinear control algorithm for sandwich nonlinear systems is created.

Algorithm 6.4.1 (Nonlinear control algorithm) .

1. *Define A_g and B_g (linear plant), A_{1g} and B_{1g} (weighted linear plant), A_c, C_d and E (disturbance and reference models), A_q and B_q (error weighting), and H_r (input weighting). Calculate D_f.*

2. *Define the nonlinear functions $\mathcal{F}_1(u)$ and $\mathcal{F}_2(m_g)$.*

3. *At time t, calculate $f_1 = \mathcal{F}_1/u$ based on $u(t-1)$.*

4. *Predict $m_g(t)$ using linear plant model, G, and previous values of m_g, u_g and f_1.*

5. *Calculate $f_2 = \mathcal{F}_2/m_g$ based on prediction of $m_g(t)$.*

6. *Solve the "spectral factor" and "Diophantine equations" using these values of f_1 and f_2.*

7. *Compute the controller from the "Diophantine equations" solutions.*

8. *Apply the controller to the plant.*

9. *Repeat steps 3 to 8.*

6.5 Multivariable simulation example

Due to the numerical problems with Matlab highlighted in Section 4.5, it is not possible to apply the technique in this Chapter to the ship DP problem. Added to the fact that there is no output nonlinearity associated with ship DP either, the simulation example in this Section is therefore chosen to be a significantly cross-coupled 2×2 system with deadzone nonlinearities on both inputs and backlash nonlinearities on both outputs. This system description has applications to mechanical systems. Deadzone is often used to represent friction, and backlash occurs in mechanical gears, as covered in Lewis et al. (2002). A thorough survey of controlling mechanical systems with backlash appears in Nordin and Gutman (2002).

The linear plant transfer function, $G(z^{-1})$, is:

$$G(z^{-1}) \;=\; A_g(z^{-1})^{-1}B_g(z^{-1}) = \begin{bmatrix} \frac{0.5z^{-1}}{(1-0.9z^{-1})} & \frac{0.1z^{-1}}{(1-0.9z^{-1})} \\ \frac{0.1z^{-1}}{(1-0.8z^{-1})} & \frac{0.3z^{-1}}{(1-0.8z^{-1})} \end{bmatrix} \tag{6.61}$$

where

$$A_g(z^{-1}) = \begin{bmatrix} (1-0.9z^{-1}) & 0 \\ 0 & (1-0.8z^{-1}) \end{bmatrix}, B_g(z^{-1}) = \begin{bmatrix} 0.5z^{-1} & 0.1z^{-1} \\ 0.1z^{-1} & 0.3z^{-1} \end{bmatrix}$$

The disturbance and reference models are:

$$\begin{aligned} W_d(z^{-1}) &= A_c(z^{-1})^{-1}C_d(z^{-1}) \\ = W_r(z^{-1}) &= A_c(z^{-1})^{-1}E(z^{-1}) = \begin{bmatrix} \frac{1}{(1-z^{-1})} & 0 \\ 0 & \frac{1}{(1-z^{-1})} \end{bmatrix}^{-1} \end{aligned} \tag{6.62}$$

where

$$A_c(z^{-1}) = \begin{bmatrix} (1-z^{-1}) & 0 \\ 0 & (1-z^{-1}) \end{bmatrix}, C_d = E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The error and control weightings are:

$$Q_c = H_q^* H_q = A_q^{*-1} B_q^* B_q A_q^{-1} = \begin{bmatrix} \frac{1}{(1-z^{-1})(1-z)} & 0 \\ 0 & \frac{1}{(1-z^{-1})(1-z)} \end{bmatrix} \tag{6.63}$$

$$R_c = H_r^* H_r = \begin{bmatrix} 0.25(1-z^{-1})(1-z) & 0 \\ 0 & 0.25(1-z^{-1})(1-z) \end{bmatrix} \tag{6.64}$$

where

$$A_q = \begin{bmatrix} (1-z^{-1}) & 0 \\ 0 & (1-z^{-1}) \end{bmatrix}, B_q = I, H_r = \begin{bmatrix} 0.5(1-z^{-1}) & 0 \\ 0 & 0.5(1-z^{-1}) \end{bmatrix}$$

The weighted linear plant model is:

$$\begin{aligned} H_q G &= B_q A_q^{-1} A_g^{-1} B_g = B_{1g} A_{1g}^{-1} \\ &= \begin{bmatrix} \frac{0.5z^{-1}}{(1-0.9z^{-1})(1-z^{-1})} & \frac{0.1z^{-1}}{(1-0.9z^{-1})(1-z^{-1})} \\ \frac{0.1z^{-1}}{(1-0.8z^{-1})(1-z^{-1})} & \frac{0.3z^{-1}}{(1-0.8z^{-1})(1-z^{-1})} \end{bmatrix} \end{aligned} \tag{6.65}$$

where

$$\begin{aligned} B_{1g} &= \begin{bmatrix} 0.211z^{-1} & 0.0397z^{-1} \\ 0.0474z^{-1} & 0.134z^{-1} \end{bmatrix} \\ A_{1g} &= \begin{bmatrix} 0.418(1-0.908z^{-1})(1-z^{-1}) & -0.016(1-z^{-1}) \\ 0.0188(1-z^{-1}) & 0.450(1-0.908z^{-1})(1-0.794z^{-1}) \end{bmatrix} \end{aligned}$$

The linear controller, from equation (6.50), is:

$$
\begin{aligned}
C_0 &= H_0^{-1} G_0 A_q^{-1} \\
&= \frac{\begin{bmatrix} 1.9(1-0.61z^{-1})(1-0.21z^{-1}) & -0.39(1-0.66z^{-1})(1+0.012z^{-1}) \\ -0.26(1-0.74z^{-1})(1+0.025z^{-1}) & 2.3(1-0.60z^{-1})(1-0.14z^{-1}) \end{bmatrix}}{(1-z^{-1})(1-0.224z^{-1})(1-0.130z^{-1})}
\end{aligned}
\qquad (6.66)
$$

The input nonlinearities are both deadzones with the following equation:

$$
u_g(t) = \mathcal{F}_1(u(t)) = \begin{cases} u(t) - U & : \ u(t) \geq U \\ u(t) + U & : \ u(t) \leq U \\ 0 & : \ |u(t)| < U \end{cases} \qquad (6.67)
$$

where the deadzone widths are $2U = \begin{bmatrix} 0.4 & 0.6 \end{bmatrix}^T$. Figure 6.5 illustrates the deadzone characteristic.

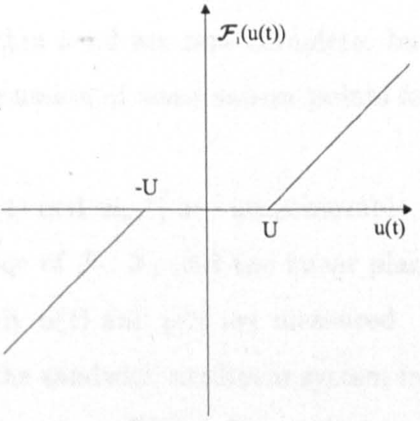

Figure 6.5: Deadzone nonlinearity

The output nonlinearities are both backlash with the following equation:

$$
\begin{aligned}
m(t) &= \mathcal{F}_2(m_g(t)) \\
&= \begin{cases} m_g(t) - V & : \ \Delta m_g(t) > 0 \ and \ m_g(t) - V > m(t-1) \\ m_g(t) + V & : \ \Delta m_g(t) < 0 \ and \ m_g(t) + V < m(t-1) \\ m(t-1) & : \ otherwise \end{cases}
\end{aligned}
\qquad (6.68)
$$

where $\Delta m_g(t) = m_g(t) - m_g(t-1)$ and the backlash widths are $2V = \begin{bmatrix} 0.1 & 0.2 \end{bmatrix}^T$. Figure 6.6 illustrates the backlash characteristic.

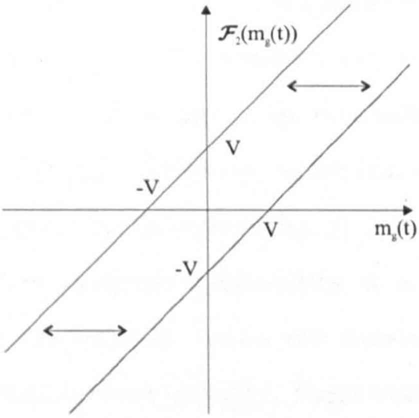

Figure 6.6: Backlash nonlinearity

Steps 1 and 2 of Algorithm 6.4.2 are now complete, but before performing the online calculations, a discussion of some salient points for the simulation.

In reality, the signals $u_g(t)$ and $m_g(t)$ are unmeasurable, hence the algorithm relies on accurate knowledge of $\mathcal{F}_1$, $\mathcal{F}_2$ and the linear plant dynamics, although it may be the case that only $u(t)$ and $y(t)$ are measured. Zhu (2002) has recently described estimation of the sandwich nonlinear system in Figure 6.2 using a least squares algorithm. The functions $\mathcal{F}_1$ and $\mathcal{F}_2$ are approximated with cubic splines, and the problem is to estimate parameters of the splines and the linear block, $G(z^{-1})$, to represent the total nonlinear system. The focus of this Chapter is on the performance of the controller rather than system identification, hence the simulation actually incorporates perfect knowledge of the nonlinearities and the linear block to calculate f_1 and f_2. However, it is assumed that, in principle, it would be possible to find f_1 and f_2 using the algorithm of Zhu (2002) or one similar. This would be an interesting direction for further work.

When the nonlinearity inputs are far removed from zero, for both deadzone and backlash, the nonlinearity functions f_1 and f_2 remain close to unity. This does not produce a challenging control problem, therefore the reference value in the simulation is chosen to be zero, as this involves operating heavily in the nonlinear region. Thus, it is possible for f_1 to be equal to zero or take on very small values, which tends to produce poor control. The controller, "spectral factor" and "Diophantine equations" contain f_1^{-1} terms, which can clearly approach infinity. Hence a limit is imposed given by the inequality, $f_1 \geq 0.7$, which was determined empirically. Examining the backlash nonlinearity, it is possible for f_2 to range from zero to infinity, so the extreme values are constrained by the inequality, $0.7 \leq f_2 \leq 1.3$, again determined empirically. These constraints are quite severe, but it is found that a wider range of values leads to "jittery" control and poor performance.

To further understand why unrestricted f_1 and f_2 produce poor control, it is necessary to recall the approximations made in the nonlinear control algorithm. Each polynomial term in the control "spectral factors" and "Diophantine equations" is a polynomial in the unit-delay operator, z^{-1}, acting on functions of time, f_1 and f_2. However, in order to simplify the solution, it is earlier assumed that f_1 and f_2 are "frozen" at the instant in time of the optimisation. Thus, it is desirable to keep excitation of the plant to a minimum in order to avoid greatly violating the "freezing" assumption. Not unreasonably, a controller gain which is large would be considered exciting for the system, but when this gain oscillates the excitation becomes even more vigorous. Undesirably, the controller in this Chapter is liable to produce exactly that kind of behaviour, for the following reasons.

Inspect Figures 6.5 and 6.6 and note that f_1 and f_2 are gradients of lines from the origin to the position on the nonlinearity, because $f_1 = \mathcal{F}_1/u$ and $f_2 = \mathcal{F}_2/m_g$. Therefore, small u will produce very small f_1 and small m_g may produce either very small or very large f_2. The controller gain in equation (6.50) will grow as f_1 or f_2 decreases, producing exciting plant inputs that are unwanted for this controller. Additionally, the nature of the backlash means that f_2 can very rapidly oscillate between large and small values, producing oscillatory gain and unwanted extra plant excitation. Thus, the solution adopted in this example is to restrict f_1 and f_2 within limits as shown in Figure 6.13. This is not the most elegant of solutions, but nevertheless the nonlinear controller is able to produce higher performance control than a fixed linear controller, as demonstrated in the Figures below.

A better solution to this problem would be to solve the "spectral factors" and "Diophantine equations" in their true difference equation form. An interesting idea for further work would be to investigate whether the limits can be widened or removed altogether using this kind of solution. In Grimble and Martin (2003), these difference equations have been solved for a SISO first order state-space system with input nonlinearity. The results in Grimble and Martin (2003) rely on the fact that the plant is SISO and first order, however, so another challenge for further work would be to extend the approach to high order systems and maybe find a general solution for a particular class of nonlinearity.

The initial condition for the output is $m_g(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ in order to provide a tracking response when the reference is zero. The disturbance inputs, $\xi_1(t)$ and $\xi_2(t)$, are normally distributed random signals of standard deviation, $\sigma = 0.02$. The simulation runs for 70 seconds and the results give a comparison between the method presented in Section 6.4 and a linear controller design based on $G(z^{-1})$ alone using the weightings in equation (6.63) and (6.64). Figures 6.7 and 6.9 contrast the system outputs for the nonlinear and the fixed linear controller, and Figures 6.8 and 6.10 depict the additive disturbances on each output. The same disturbance is used for the linear and nonlinear controller simulations.

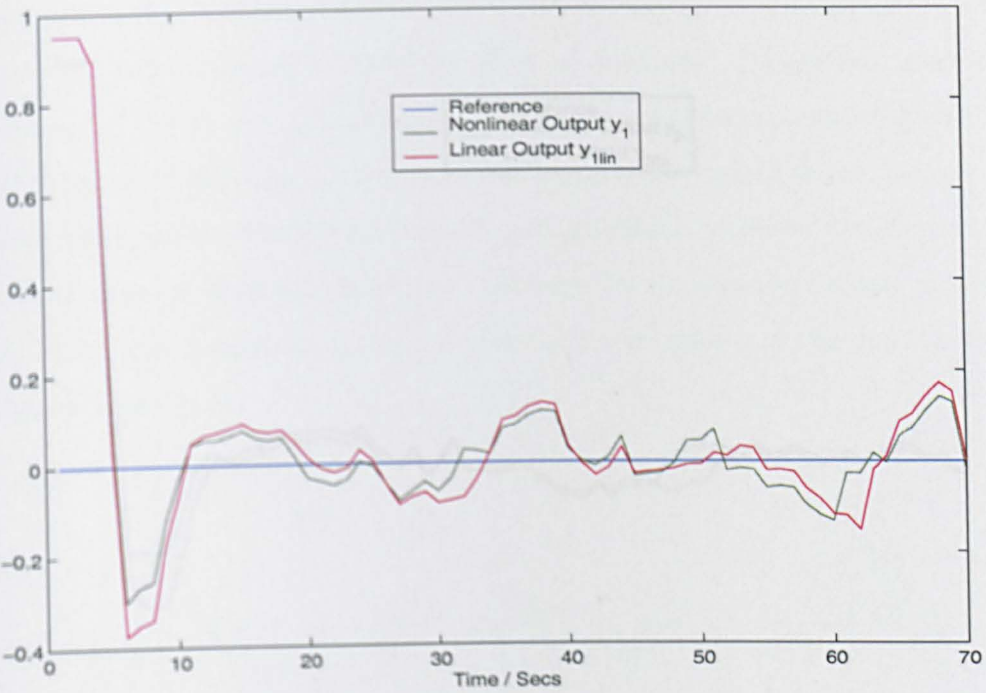

Figure 6.7: 1st system output y_1 - nonlinear controller versus linear

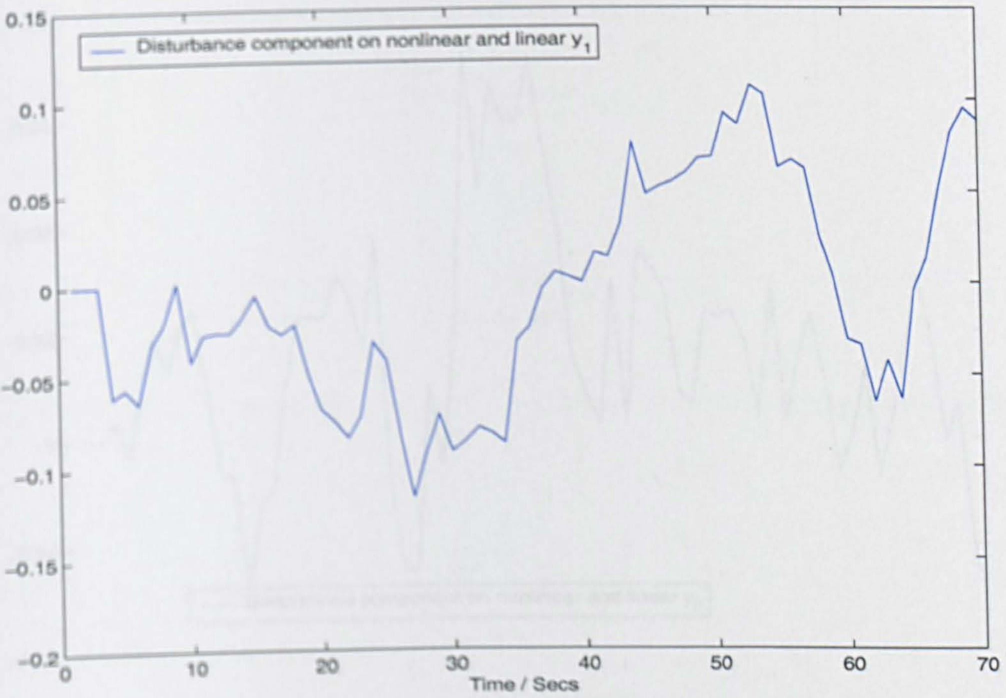

Figure 6.8: Additive disturbance on 1st system output

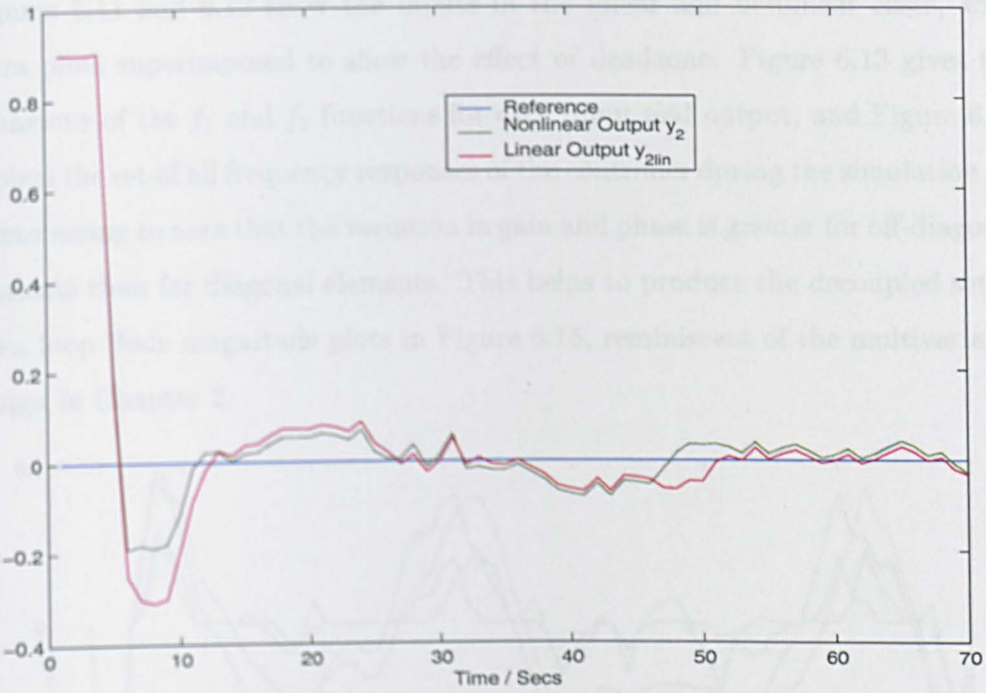

Figure 6.9: 2nd system output y_2 - nonlinear controller versus linear

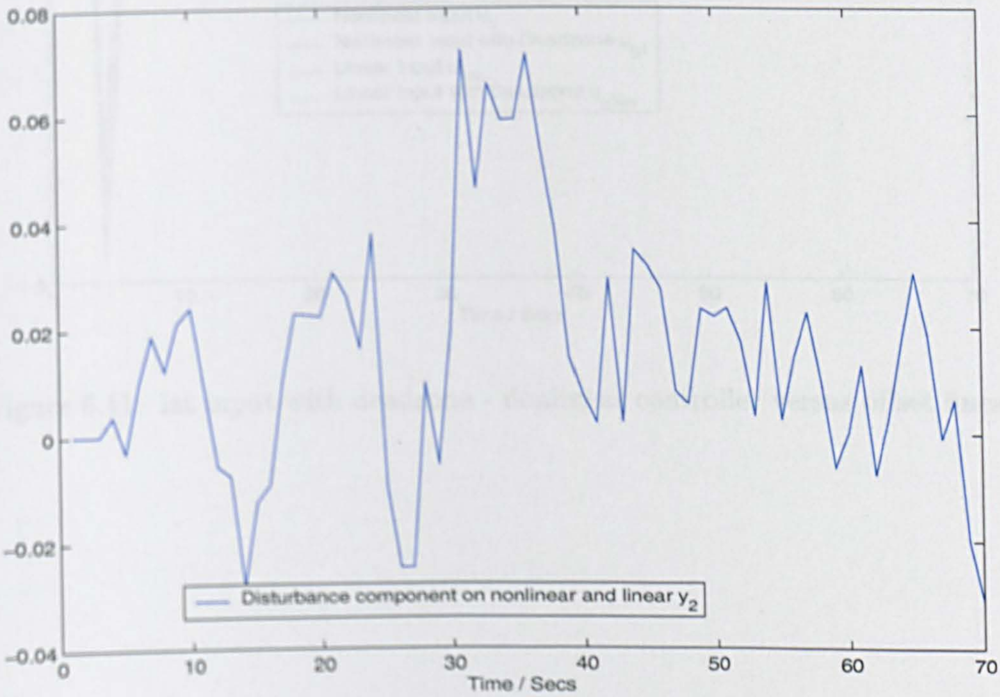

Figure 6.10: Additive disturbance on 2nd system output

Figures 6.11 and 6.12 show the inputs in the linear and nonlinear cases, with extra plots superimposed to show the effect of deadzone. Figure 6.13 gives the behaviour of the f_1 and f_2 functions for each input and output, and Figure 6.14 depicts the set of all frequency responses of the controller during the simulation. It is interesting to note that the variation in gain and phase is greater for off-diagonal elements than for diagonal elements. This helps to produce the decoupled set of open loop Bode magnitude plots in Figure 6.15, reminiscent of the multivariable design in Chapter 2.

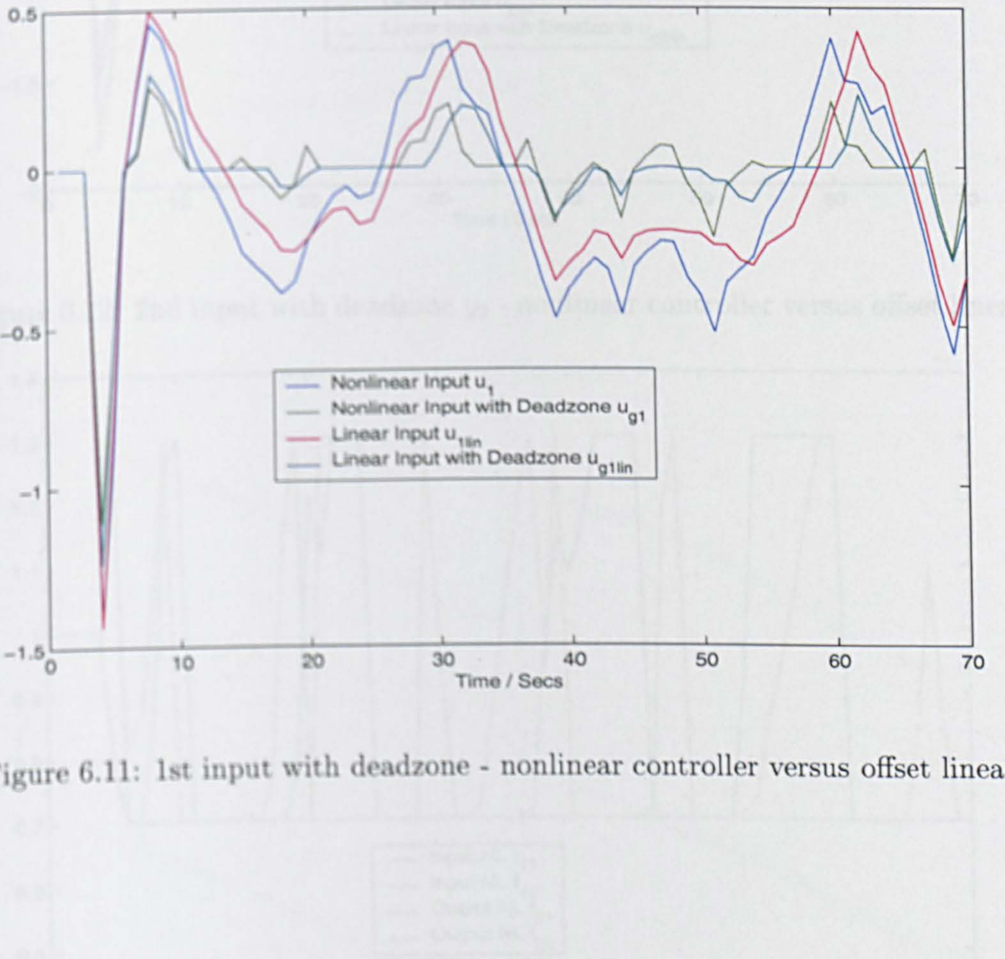

Figure 6.11: 1st input with deadzone - nonlinear controller versus offset linear

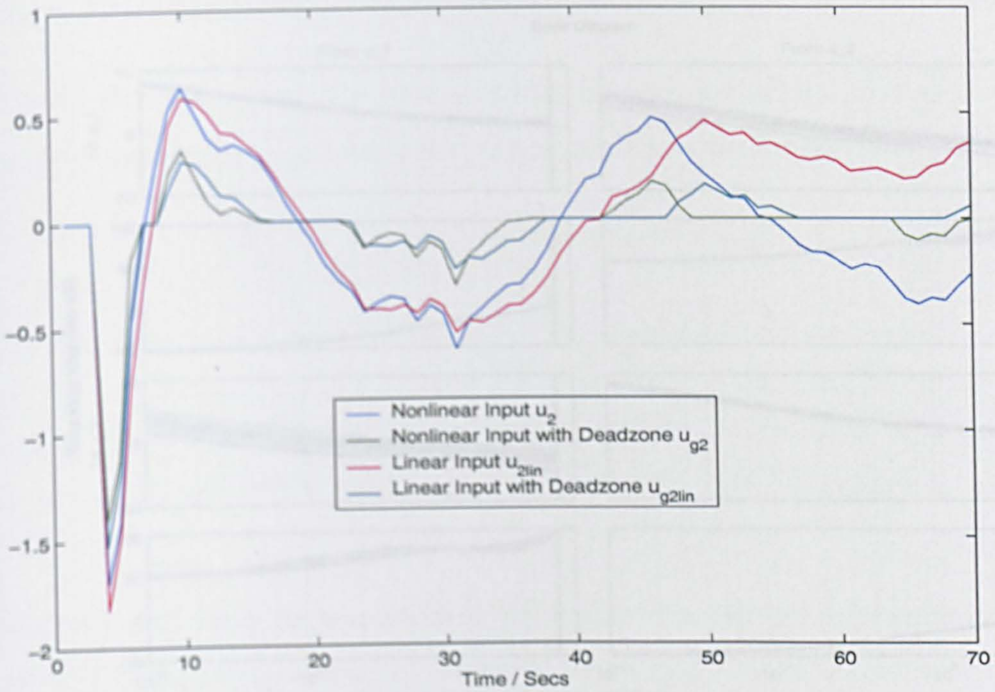

Figure 6.12: 2nd input with deadzone y_2 - nonlinear controller versus offset linear

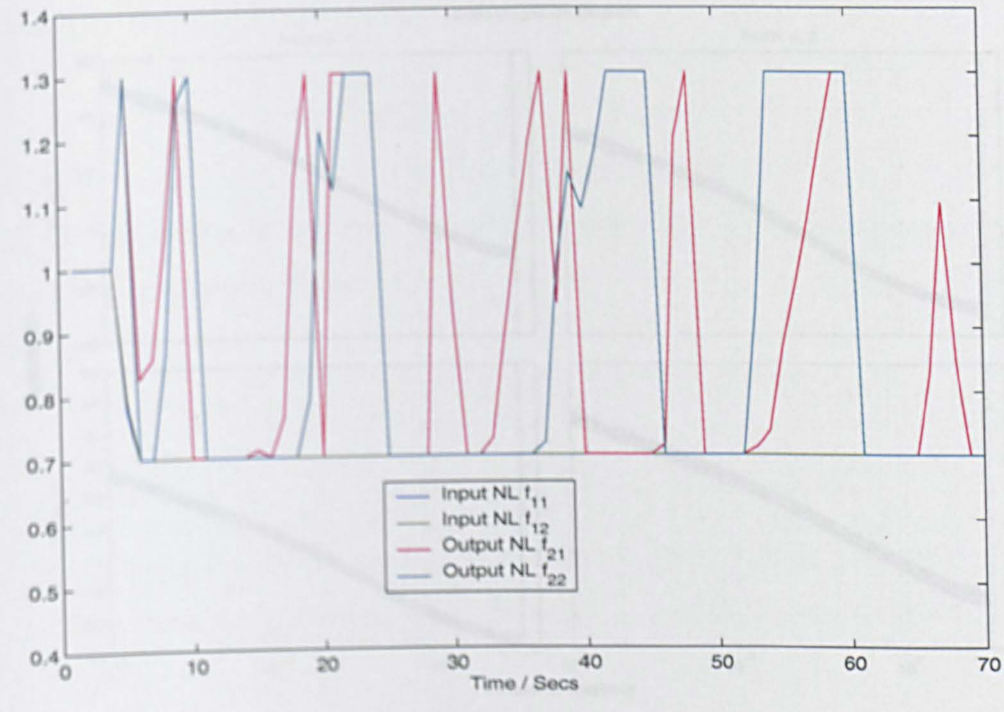

Figure 6.13: Nonlinear functions

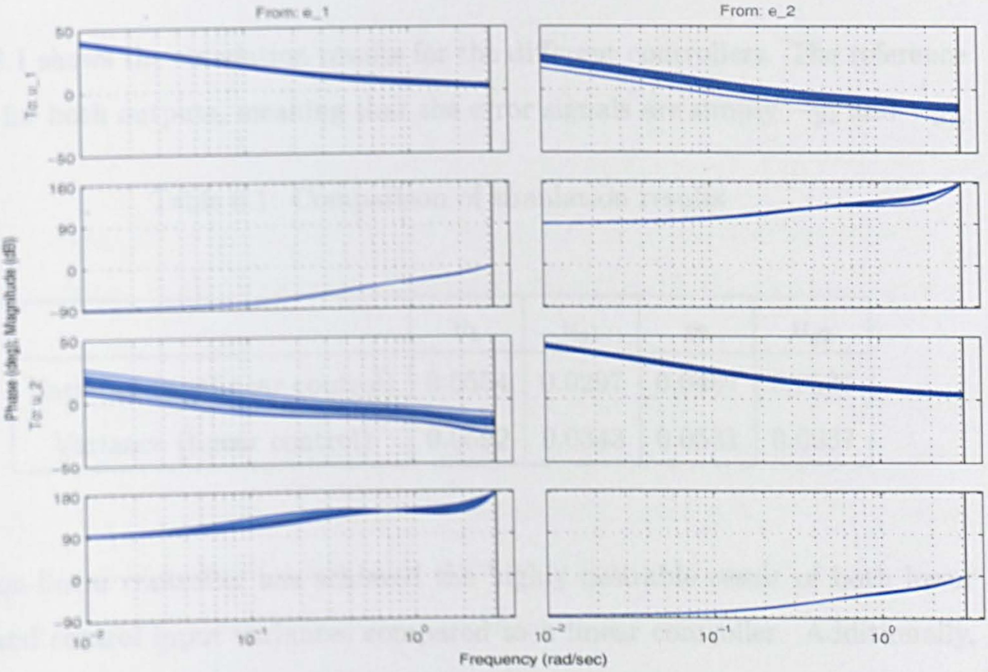

Figure 6.14: Bode plots of C_0

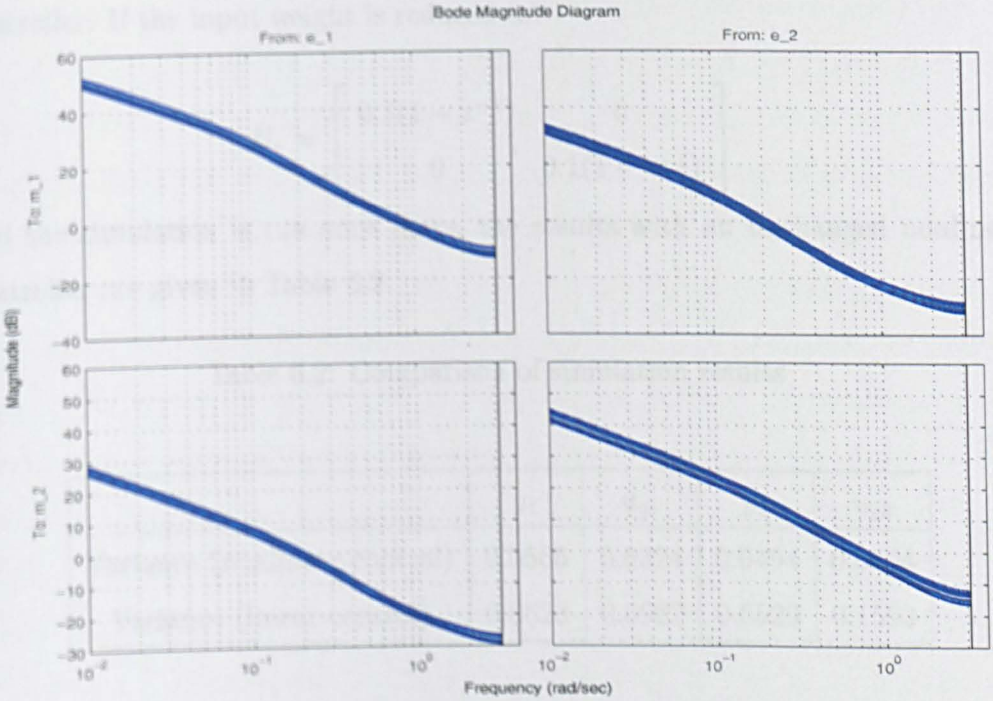

Figure 6.15: Bode magnitude plots of WC_0

6.6 Discussion

Table 6.1 shows the simulation results for the different controllers. The reference is zero for both outputs, meaning that the error signals are simply $-y_1$ and $-y_2$.

Table 6.1: Comparison of simulation results

	y_1	u_{g1}	y_2	u_{g2}
Variance (nonlinear control)	0.0554	0.0297	0.0487	0.0536
Variance (linear control)	0.0592	0.0343	0.0531	0.0637

The non-linear controller has achieved the highly desirable result of both lower error and control input variances compared to a linear controller. Additionally, due to the nonlinear nature of the system, it is not true that the error variance in the linear case can be reduced by simply increasing the gain of the linear controller. If the input weight is reduced to:

$$H_r = \begin{bmatrix} 0.1(1-z^{-1}) & 0 \\ 0 & 0.1(1-z^{-1}) \end{bmatrix}$$

and the simulation is run once more, the results with an unchanged nonlinear controller are given in Table 6.2.

Table 6.2: Comparison of simulation results

	y_1	u_{g1}	y_2	u_{g2}
Variance (nonlinear control)	0.0566	0.0324	0.0494	0.0504
Variance (linear control)	0.0623	0.0962	0.0529	0.1593

The linear control input variance has clearly increased as expected, but there is little change in y_2, and y_1 has actually increased. Conversely, if the input weight is raised to:

$$H_r = \begin{bmatrix} (1 - z^{-1}) & 0 \\ 0 & (1 - z^{-1}) \end{bmatrix}$$

it seems reasonable to predict the input variance to decrease and the error variance to increase. In fact, the system becomes unstable with the linear controller.

These results suggest that the nonlinear control is truly superior and it is not simply a matter of tuning to produce comparable performance. Simulations using nonlinearities other than deadzone and backlash have produced similarly positive results, thus it is concluded that this technique has potential for performance improvements with a variety of multivariable nonlinear plants.

There are disadvantages not seen in these results, however, stemming from the assumption of a "frozen" plant at each sample step. This assumption requires that f_1 and f_2 are slowly varying, so that the solutions of the "spectral factor" and "Diophantine equations" are close to the true solutions - That would require difference equations in the shift operator, z^{-1}, and knowledge of past and future signal variations. Also, the commutation of H_q and f_2 in equations (6.39) and (6.45) relies on slowly varying f_2 to avoid this operation becoming nonsensical. The non-linear results in Figures 6.7, 6.9, 6.11 and 6.12 are known to deteriorate as the system bandwidth or the disturbance variance increases, or as the deadzone and backlash widths decrease. The explanation for this is that, in each case, the non-linearities are excited more. Therefore, f_1 and f_2 vary rapidly and the assumption of constant or slowly-varying $f_2(m_g)$ and $f_1(u)$ is completely violated, leading to control performance degradation. Of course, if the deadzone and backlash nonlinearities are operating with large inputs, their behaviour is relatively linear and f_1 and f_2 are both approximately equal to one. In this case, there is almost no approximation in the theory, but this is to be expected as the plant is behaving almost like a fixed linear system. Thus, in order to realise the potential of the nonlinear controller in this Chapter, the system must be required to operate slowly and predominantly in the nonlinear region. This technique is therefore probably suited to slow chemical processes, or even to ship DP if the numerical issues with Matlab can be overcome.

Figure 6.13 shows that f_2 remains constant at the limits for several seconds at a time and f_1 remains constant at the lower limit for most of the simulation. Obviously this situation is not ideal, as it would be desirable to allow f_1 and f_2 to take whatever value is necessary to describe the nonlinear behaviour. However, widening the range of the f's simply increases the excitation in the system, leading to performance degradation for the reasons given above. Also, as noted in Section 6.4.1, the stability margins of the controller are eroded with increasing f_1 or f_2, giving further reason to restrict the value of the f's. If the system becomes marginally stable or somewhat oscillatory, then the f's will be excited further, leading to more extreme values of f_1 and f_2, rendering the system impossible to stabilise satisfactorily.

An interesting idea for further work, therefore, would be to investigate whether the limits can be widened or removed altogether when the "spectral factor" and "Diophantine equations" are solved as difference equations. In an extension to this Chapter, Grimble and Martin (2003), these difference equations have been solved for a SISO first order state-space system with input nonlinearity. However, saturation was employed in that work and it has been found that limits are not necessary for such a nonlinearity, even with the controller from this Chapter. (With regard to saturation, the technique in this Chapter is not particularly successful. The problem is that, as the nonlinearity becomes increasingly saturated the controller gain increases, which simply pushes further into saturation. With "smooth" saturation, there is a slight performance advantage, but with "hard" saturation, it is better to simply use an anti-windup mechanism). The results in Grimble and Martin (2003) rely on the fact that the plant is SISO and first order, thus a challenge for further work would be to extend the approach to higher order systems and maybe to find a general solution for a particular class of nonlinearity.

Another idea for further work comes from the fact that the simulation incorporates perfect knowledge of the nonlinearities and the linear block to calculate f_1 and f_2. Clearly, this will have considerable bearing on the robustness and performance of a practical algorithm. Thus, it would be of interest to investigate various estimators for f_1 and f_2, including the idea by Zhu (2002) highlighted earlier. It is possible that poor estimation will cancel any advantage gained by use of the nonlinear algorithm, hence placing further importance on the need to develop the difference equation approach mentioned above.

6.7 Conclusions

This Chapter began by drawing attention to the superposition principle of linear systems and the consequence of its failure for nonlinear systems. The lack of general solutions of nonlinear differential and difference equations leads to a paucity of analysis and control design procedures. The existing techniques involve approximations, such as with Describing Functions or Fuzzy Control, complex mathematical theory, such as Geometric Control, or do not yield obvious design techniques, in the case of Lyapunov theory. Hence, the stated aim of the Chapter was to investigate a nonlinear control design technique that is both practical and with a rigorous theoretical foundation, where inevitably some approximations are introduced. The main contribution of the Chapter was to demonstrate that the design technique is applicable to a multivariable nonlinear "sandwich" system, and that an improvement over a fixed linear controller is possible.

The nonlinear controller is built on a foundation of time-varying control, in which the plant and controller are stated as multivariable transfer functions in the delay operator, z^{-1}, and time, t, with fixed reference and disturbance models. The cost function employed is analogous to a normal stochastic LQG cost, except that the weightings and weighted signals are combined into a time-varying form. The solution to the optimal polynomial problem follows, but in the time domain rather than the frequency domain normally employed for polynomial LQG. The frequency domain is not such a useful concept for time-varying systems, but the solution does involves a kind of "spectral factor" and two "Diophantine equations". These expression cannot be interpreted in the usual manner, but it is natural to adopt the terminology from time-invariant control.

After deriving the solution to the time-varying control problem, a particular type of nonlinear system was then defined and manipulated into time-varying form. This nonlinear system is known as a "sandwich" system because it consists of a linear dynamic block sandwiched between input and output nonlinearities. To facilitate manipulation of the sandwich system into time-varying form, the nonlinear functions were expressed as the product of a function of the input, $f(x(t))$ and the input itself, $x(t)$. This form is able to represent many common static and dynamic nonlinearities, with the exception being those with non-zero output for a zero input. In practice this is not a problem, as limits are later placed on the value of f. The polynomial representation of the nonlinear system was then substituted into the solution of the time-varying optimal control problem, with two assumptions being made to render the final solution tractable. Firstly, the input nonlinearity was assumed to be a function of the previous input, $u(t-1)$, so that the current control signal was not required before it had been calculated. This assumption is reasonable for slowly-varying u. Secondly, at any given sample step, all system polynomials are "frozen", so the "spectral factors" and "Diophantine equations" can be solved using standard time-invariant techniques.

A short SISO example was presented to show that the controller does not simply consist of a fixed linear time-invariant controller cascaded with inverses of the nonlinearity, as it first appears. It is seen that large f's cause the controller to be have more phase lead with lower gain and vice versa for small f's. The fact that this behaviour appears from rigorous theory with some reasonable approximations introduced builds confidence in the approach, and supports the assertion that the controller is more sophisticated than simple nonlinearity inversion. As a rule of thumb, phase lead and lower controller gain increase the stability margin, phase lag and higher gain decrease stability margins. Hence, the controller is exhibiting desirable frequency domain properties as a result of this theory. That is, when the nonlinearity gain is high and liable to destabilise the closed-loop, the controller acts in a stabilising manner. When the nonlinearity gain is low, the controller pushes up the gain to improve speed of response without risking instability.

An algorithm for the overall control of a multivariable nonlinear sandwich system is stated, where the values for f_1 and f_2 nonlinearities are inferred from perfect knowledge of the plant behaviour. Such knowledge is unrealistic, but demonstrates the potential of the controller without the complication of any estimation or identification. A simulation example, using a cross-coupled 2-input 2-output system with deadzone on the input and backlash at the output, is presented, where the reference signal is zero in order to excite both nonlinearities. It is necessary to limit the magnitude of the nonlinear functions, f_1 and f_2, however, otherwise the control becomes "jittery" and the performance is poor. This restriction is required so that rapidly changing signals do not appear, thereby violating the assumption of a "frozen" plant. If the signals are sufficiently slowly-changing, then the plant polynomials are close enough to a "frozen" state for the various approximations to be small.

The results of the simulation demonstrated that improved control is possible in comparison with a fixed time-invariant linear controller. The non-linear control yields output error and control input variances that are smaller than with fixed linear control, provided that the signals within the system are slowly-varying. Also, the results were not based on superior tuning in the nonlinear case, as a decrease in input weighting on the linear controller produces larger input variance but no improvement in error variance. An increase in input weighting only results in instability with a linear controller. Hence, the nonlinear technique is genuinely superior.

There are two ideas for further work. The first is to treat the "spectral factors" and "Diophantine equations" as difference equations, rather than making the approximation of a "frozen" plant. This has been attempted in Grimble and Martin (2003) for the first-order SISO case, but clearly would be more useful if high order or multivariable systems could be addressed. This idea would also require reliable prediction of the system variables a few steps in advance, which leads onto the second idea for further work. The simulation works with perfect knowledge of the nonlinearities and linear block when calculating f_1 and f_2. Perfect knowledge is impossible, of course, so an investigation of plant estimators is essential if the nonlinear technique is to be truly practical. Identification of nonlinear sandwich systems has attracted some control engineering interest, however, so it is envisaged that such an estimator may be available or will be developed.

Chapter 7

Conclusions and Further Work

7.1 Summary of the thesis

This thesis has investigated the application of several new and existing techniques to marine and nonlinear systems. The particular marine problem is dynamic positioning (DP), where the aim is to regulate the position and heading of a vessel by using directional thrusters. The environmental disturbances at sea are powerful, hence drilling and offloading ships require effective control to avoid potential accidents. The main difficulties in the DP problem are that ship motion behaves nonlinearly in response to thruster forces, the various degrees of freedom are cross-coupled, the thrusters must not attempt to cancel high power zero mean wave forces and must not saturate.

Part of the research conducted for this thesis has entailed exploring methods for dealing with some of these difficulties, as follows. Multivariable cross-coupling has been dealt with using a set of techniques to quickly produce a PID design for the DP velocity and position loops. In this case, nonlinearities are not explicitly dealt with, as the techniques are applied to a linearisation of the ship at an operating point. Notch filters are used to attenuate the high power zero mean wave forces, requiring the velocity loop to produce enough phase margin to offset the filter phase lag. Thruster saturation is avoided by adjusting the controller gains to fall within the rated thruster force for high sea disturbances.

The nonlinearities are later dealt with using a multiple model optimisation approach. This approach is based on a polynomial LQG control solution using spectral factors and Diophantine equations, that has been approximated to give a suboptimal restricted structure controller. The matrix-based suboptimal solution allows the optimisation to be carried out across several models at once, hence this is exploited in conjunction with linearisations of the nonlinear ship model. Several fixed representations of the ship dynamics are included in the set with an RLS identified model, to produce an online adaptive controller. This is applied to the surge axis velocity loop only, as the algorithm is numerically incapable of dealing with multivariable or high order transfer functions. Notch filters are included, but are not explicit in the ship model in order to again keep the transfer function order low. This is not a problem, however, as the controller bandwidth is designed to be lower than the notch frequency. Thruster saturation is avoided by tuning the weights to give restricted structure PI gains similar to those in the multivariable study.

The basic adaptive multiple model controller is further developed for the LQGPC case, where a stochastic reference generator is included and a multi-step cost function is to be minimised. The problem is initially posed in state-space before an optimisation is performed over future input signals, leaving the cost in a more standard LQG form. Using a polynomial description of the system elements allows the solution to be given in terms of spectral factors and Diophantine equations as above. A sub-optimal approximation is made once again, which permits construction of another adaptive multiple model controller. The ship surge axis velocity loop example is then repeated for comparison with the earlier LQG controller.

One Chapter of this thesis is concerned with an identification problem related to DP. Neural networks have seen little application to marine vessel dynamics identification and have never been considered for describing a coupled ship system. The relevant dynamics are highly nonlinear and coupled, thus a universal approximator such as a neural network seems suitable for the task. In this thesis, a multilayer perceptron network is trained firstly with data from a coupled-ship simulation to test the feasibility of the approach. The results are good, hence further training is carried out using data supplied by Mitsubishi Heavy Industries from a 1/50th scale model. The available data is for several different wave heights, thus network training is performed to examine performance when the data is corrupted by disturbances of differing magnitudes of force.

The remaining component of research in this thesis examines a novel approach to control of multivariable nonlinear systems. The derivation of the controller depends upon the theory of time-varying systems, which permits a nonlinear plant to be described as a succession of linear systems from one sampling instant to the next. A polynomial time-varying optimal control problem is solved, before introducing the "sandwich" form of nonlinear plant description. This consists of a linear transfer function with input and output nonlinearities of a particular input-dependent form. The sandwich system is then stated in polynomial form and substituted into the time-varying optimal control solution. Given the assumption that the system and nonlinearities vary slowly, the result is a suboptimal nonlinear controller. It is simple to calculate online, as the relevant expressions are of similar form to time-invariant spectral factors and Diophantine equations, but with a different interpretation. It is not possible to apply the algorithm to the ship problem, as it is numerically fragile, but a first order 2 × 2 example with deadzone input and backlash output is presented instead.

7.2 Conclusions

The ship DP problem is multivariable in nature due to coupling between surge, sway and yaw directions of movement, whilst the mathematical model may not be known. The problem of rapid preliminary tuning has never previously been addressed whilst taking these factors into account, hence a research contribution is made in Chapter 2 by examining four methods for rapid tuning of a multivariable PID controller.

These model-free methods are intended to straightforwardly produce an initial controller design based on step tests or application of sinusoidal inputs at a single frequency. Although a model was used for simulation, it is not difficult to apply these tests in reality where no model is known. When the four methods were applied to the linearised ship model at an operating point, the Davison method proved to be of little use, as the ship model in question possesses frequency domain characteristics such that the technique cannot be applied. Penttinen and Koivo, Maciejowski and the combined approaches, however, produce control schemes with a few short, simple steps.

The Pentinnen-Koivo technique provides decoupling at very low and very high frequencies, the Maciejowski method provides approximate decoupling at the chosen bandwidth, whereas the combined method decouples at the selected bandwidth and very low frequencies. The control scheme consists of velocity and position feedback loops, tuned in that order. The controllers were easily adjusted to meet design criteria of stability, speed of response, disturbance rejection, decoupling and avoidance of actuator saturation. From simulation results, it was concluded that using the Penttinen-Koivo method in the velocity loop and Maciejowski method in the position loop provides the best performance, by a small margin only, although with slightly more tuning.

The main flaw of these design methods is the dependence on a point where decoupling can take place in the frequency range of interest. The Davison method is of no use where integrators are present in the plant, and the Pentinnen-Koivo method requires that high frequency motions are desirable in a system. Maciejowski's approach can produce decoupling through a large range of frequencies, but selection of the wrong bandwidth can result in strong interaction between one or more of the system modes, and therefore an ineffective control system.

The main advantage of these design approaches is the speed and ease with which an effective multivariable PID controller can be produced. The advantage to industry is that money can be saved on design effort, and also on commissioning time. A benefit for academics is that a MIMO PID controller can easily be obtained when making comparisons with more advanced techniques. Of course, further fine-tuning of individual loops can take place after using these methods, but the initial design can serve as a useful starting point.

Multivariable PID designs in Chapter 2 are applicable at only one operating point and there is potential for instability and poor performance from this controller at other operating points. This is due to the effect of changing plant dynamics combined with an invariant controller. The idea developed in Chapter 4 was to combine the benefits of a multiple-model and adaptive controller into one scheme, where the structure of the controller is restricted to a particular form. Unlike previous multiple-model schemes, the technique in Chapter 4 does not involve designing separate controllers for each model in the set, but instead performs an optimisation across the whole set.

The adaptive multiple-model controller is applied to the ship DP problem from Chapter 2, but first a single model example is detailed to illustrate features of the restricted-structure controller. It is seen that only four iterations are required to produce gains for PI control of the ship's surge velocity loop. The full-order controller is $3rd$ order, whereas the PI controller is only $1st$ order, yet the frequency responses between DC and $0.1rad/s$ are identical. The notch filter is ignored in order to keep the plant model order low. This is desirable to avoid numerical errors due to the use of 64 bit arithmetic, and is justifiable because the notch occurs at above the crossover frequency. Fortuitously, the step response of the restricted-structure controller is actually superior to the optimal controller when the notch filter is included in the plant but not in the model. Without a notch in the plant, the restricted-structure response is only marginally inferior to the full-order control.

A full DP example is described, where the 3 degree of freedom simulation from Chapter 2 is used but the adaptive multiple-model controller is only applied to the surge velocity loop. The other velocity loops take gains from the Penttinen-Koivo method in Chapter 2, and the position loop gains are produced with the Maciejowski technique. The performance of the restricted-PI-structure controller is evaluated for various probability weightings on each linear model. It is observed that performance is worst when the identified model is not included in the optimisation and only linearisations from three representative operating points are taken into account. As the probability weighting on the identified model is increased, the performance gradually improves to a peak when the probability is 0.9. The "pure" adaptive case, where the identified model probability equals 1, exhibits poorer performance than when the probability is 0.7, 0.8 or 0.9.

It is believed that the technique presented in Chapter 4 provides a combination of the benefits of adaptive and multiple-model optimal controller designs in one scheme. An adaptive controller is able to adapt to changing system parameters at the expense of possible instability, as the present controller depends upon an estimate of the current plant model only. A multiple-model optimal controller gives greater assurance of stability over a wide range of operating points with the expense of conservative performance. A multiple-model adaptive controller is intermediate to these two schemes. It provides a certain amount of confidence in stability, due to the weighted effect of fixed known models in the optimisation, plus a performance enhancement due to the incorporation of system identification knowledge from one sample point to the next. The restricted structure of the control law provides simplicity of implementation, and transparency of the solution to those acquainted with much-used classical control laws. Thus, the main contribution of Chapter 4 is to propose a new kind of adaptive controller which combines the benefits of existing control schemes, and also to apply this to the ship DP problem.

The LQGPC problem in Chapter 5 has much in common with the Chapter preceding it, where the cost is approximated by a matrix-vector form and minimised with respect to the restricted-structure controller parameters. Stacking matrices for each linear model once more gives the multiple model solution, which is augmented with on-line identified parameters to produce the final adaptive algorithm. The example dealt with is the ship DP problem also addressed in Chapter 4, but it is necessary to make some alterations before examining the single model case. LQGPC does not incorporate dynamic cost weights without increasing the complexity of the solution, but it is desirable to include integral action. The plant is therefore augmented with an integrator, so that low frequency errors are penalised. The one-step delay built into the state-space description prevents the disturbance from taking the same form as in Chapter 4, so a delay is tolerated. Additionally, the numerical fragility of Matlab does not allow the reference model to include an integrator, hence the pole is moved to 0.990.

A single model example is detailed where the prediction and input horizons are both two steps long. The restricted structure optimisation terminates after five iterations, giving PI gains to approximate a third-order full-order controller. It is necessary to in fact make the controller structure PD, so that the additional integrator can be moved from the plant description after the optimisation. The closed-loop performance is satisfactory, although not as good as the standard LQG case in Chapter 4. It is noted that increasing the control weighting and decreasing the input weighting increases controller gain across the frequency spectrum and vice versa. The weights for the first time step of the predictions have greatest effect, with very little noticed for the other weightings. The solution is also quite sensitive to output noise, where an increase produces a decrease in controller gain at all frequencies.

This initial single model example is followed by a repeat with longer prediction and input horizons, 30 steps and 5 steps respectively. The main effect on the controller is to reduce low frequency gain and raise high frequency gain. This seems reasonable when it is observed that derivative action gives prediction of future error in a sense and, significantly, acts at high frequencies. The error and control weightings for several steps into the future have more influence on the solution than in the previous example, and the overall performance is slightly better with longer horizons. Notably, the closed-loop step responses are still inferior to the standard LQG case, but it is believed that this is due to the specification of a random reference in the optimisation, but the actual use of a step in the example. Also, for both single model examples, there is very little difference in performance between the full-order and the restricted-structure controller.

A full DP example is investigated as in Chapter 4, using the 3 degree of freedom simulation from Chapter 2, where the multiple-model adaptive predictive controller is applied to the surge velocity loop. The other velocity loops take gains from the Penttinen-Koivo method in Chapter 2, and the position loop gains are produced with the Maciejowski technique. Linearised models are taken from three representative operating points, and the fourth model uses parameters identified with recursive least squares. The ship manoeuvre and disturbances are the same as in Chapter 4, and the performance of the restricted-PI-structure controller is evaluated for various probability weightings on each linear model. Unlike in Chapter 4, it is observed that performance is worst for the pure adaptive case, but again best performance is seen at $p_4 = 0.9$. Thus, it is concluded that the adaptive controller is indeed exhibiting better performance than the pure multiple-model case, whilst presumably benefiting from increased robustness due to the fixed linear models. A flaw mentioned in Chapter 4 is a lack of stability or robustness proofs, so at present the robustness benefit is conjectured. The poor performance in the pure adaptive case is apparently due to sensitivity of the predictive controller to parameter variations and plant-model mismatch. This is supported by the great variation in PI gains in the $p_4 = 1$ example.

One very positive outcome from the full DP example is improved performance over the standard LQG controller in Chapter 4. The error standard deviation of the controlled absolute velocity is 14.3% greater with standard LQG, but the control action is only 6.9% smaller. The best explanation for this is that the reference signal approximates the filtered white noise in the optimisation fairly well, in contrast with the single model examples where the reference employed is a step. Hence, the predictive nature of LQGPC is better exploited. A significant contribution of Chapter 5 is therefore a demonstration that restricted-structure adaptive LQGPC can lead to better performance than equivalent LQG control.

In the past, when modelling marine vessels for DP, the problem has been approached by deriving a simplified model and finding the relevant parameters approximately, or with more sophistication by using an extended Kalman filter. Neural networks are a valuable technique for approximating nonlinear, multivariable functions, hence a multilayer perceptron feedforward network has been studied in Chapter 3 for identifying ship dynamics of a coupled FPSO and shuttle tanker. The main contribution is to demonstrate that the network could learn and generalise the dynamics from real data provided by a Mitsubishi Heavy Industries 1/50th scale model.

The coupled ships were modelled in Simulink using the equations from Chapter 2, plus additional forces from the turret, hawser and tug boat. Data from this simulation was then used to check the viability of using a neural network for the problem. With normalised data, it was discovered that the dynamics were learnt well and generalised from the small training set to the complete validation set. Integrating to generate velocity and position signals also demonstrated that the network was successful in learning, although a filter was required to remove steady-state offset. The non-DC performance was otherwise encouraging, so it was possible to progress to using real data from Mitsubishi's scale model.

Here, input and output signals were as in the simulation case, but with the addition of wave height as an input. The first network training procedure was conducted with data from a still environment, that is with no waves applied. Again using normalised data, the network outputs were found to match the data with little error. Further training with a wave height of two metres yielded acceptable results, but with four metre wave height, the errors were not insignificant. The network was also trained using all of the data together from the previous three cases, before being validated with data from three metre wave height. In this case, the results were of poor quality. Indeed, any attempt to train the network with more than one data set at a time resulted in significant errors, even for validation with a data set used in training.

The main advantage of the neural network approach is that it is not necessary to know the precise structure of the equations of motion or to identify particular parameters. The multivariable character of the problem does not present a difficulty either, as the interconnected nature of the network is well-suited to cope with this. The ship is simply an input-output mapping which can behave in a highly nonlinear manner with no loss of accuracy in signal reproduction.

The aim of Chapter 6 was to investigate a nonlinear control design technique that is both practical and with a rigorous theoretical foundation, although where inevitably some approximations are introduced. The main contribution of the Chapter was to demonstrate that the design technique is applicable to a multivariable nonlinear "sandwich" system, and that an improvement over a fixed linear controller is possible.

A short SISO example was presented to show that the controller does not simply consist of a fixed linear time-invariant controller cascaded with inverses of the nonlinearity, as it first appears. It is seen that large f's cause the controller to be have more phase lead with lower gain and vice versa for small f's. The fact that this behaviour appears from rigorous theory with some reasonable approximations introduced builds confidence in the approach, and supports the assertion that the controller is more sophisticated than simple nonlinearity inversion. As a rule of thumb, phase lead and lower controller gain increase the stability margin, phase lag and higher gain decrease stability margins. Hence, the controller is exhibiting desirable frequency domain properties as a result of this theory. That is, when the nonlinearity gain is high and liable to destabilise the closed-loop, the controller acts in a stabilising manner. When the nonlinearity gain is low, the controller pushes up the gain to improve speed of response without risking instability.

A simulation example was examined, using a cross-coupled 2-input 2-output system with deadzone on the input and backlash at the output, where the reference signal is zero in order to excite both nonlinearities. It was necessary to limit the magnitude of the nonlinear functions, f_1 and f_2, however, otherwise the control becomes "jittery" and the performance is poor. This restriction was required so that rapidly changing signals do not appear, thereby violating the assumption of a "frozen" plant. If the signals are sufficiently slowly-changing, then the plant polynomials are close enough to a "frozen" state for the various approximations to be small.

The results of the simulation demonstrated that improved control is possible in comparison with a fixed time-invariant linear controller. The non-linear control yields output error and control input variances that are smaller than with fixed linear control, provided that the signals within the system are slowly-varying. Also, the results were not based on superior tuning in the nonlinear case, as a decrease in input weighting on the linear controller produces larger input variance but no improvement in error variance. An increase in input weighting only results in instability with a linear controller. Hence, the nonlinear technique is genuinely superior.

7.3 Suggestions for further work

In order to extend the work of Chapter 2, it would be interesting to try alternative bandwidths for the velocity loop and to experiment more with the scalar tuning gains. It is conceivable that the control system performance with the ship example could be improved with experimentation. Another extension would be to try a large number of operating points to confirm the wide applicability of the techniques, then to apply gain scheduling as the overall control scheme.

In Chapter 3, it was found that approximating more than one data set simultaneously does not appear to be easily achievable. As further work, several networks could be used, one for each likely operating point, with each network trained extensively using data from each point. Ship motion could be produced by one network corresponding to the present operating condition, or from a weighted sum of the various network outputs. A further, less fundamental flaw is the transformation from acceleration outputs to earth-fixed velocity and position signals. It may be necessary to use a different process in production of the tank test data, and to take care in the use of integrators for velocity and position signals.

Stability and robustness proofs for multiple models and convergence of the restricted-structure algorithm would be valuable directions for advancing the work in Chapter 4. Simultaneous stabilisation theory is suggested as an approach to the first problem, and empirical evidence suggests that convergence is likely to be assured if a suitable procedure can be found. Establishing these two results would also reinforce the perceived advantages of the adaptive controller, which could be further improved by allocating realistic probability weightings for the fixed models in some manner. It is also suggested that a gain scheduling example could be used for comparison with the controller in Chapter 4, as there are similarities between the two approaches in that a set of linearised models is required in both cases.

The suggestions relevant to Chapter 4 also apply to Chapter 5, but it is further noted that the predictive controller does not address constraints, unlike many other MPC controllers. A suggestion for future work is to look for an approximation to the cost in terms of both the controller gains and the plant input, outputs and states. Hopefully, a QP solver could then be employed to perform the constrained minimisation. Another suggestion is to look for a method of incorporating dynamic weights into the restricted-structure solution, so that the augmented plant is not required and the input weighting is truly on u, rather than effectively on Δu.

Regarding Chapter 6, there are two ideas for further work. The first is to treat the "spectral factors" and "Diophantine equations" as difference equations, rather than making the approximation of a "frozen" plant. This has been attempted in Grimble and Martin (2003) for the first-order SISO case, but clearly would be more useful if high order or multivariable systems could be addressed. This idea would also require reliable prediction of the system variables a few steps in advance, which leads onto the second idea for further work. The simulation works with perfect knowledge of the nonlinearities and linear block when calculating f_1 and f_2. Perfect knowledge is impossible, of course, so an investigation of plant estimators is essential if the nonlinear technique is to be truly practical. Identification of nonlinear sandwich systems has attracted some control engineering interest, however, so it is envisaged that such an estimator may be available or will be developed.

Bibliography

Åström, K. J. and Hägglund, T. (1995). *PID controllers: Theory, design and tuning*, 2nd Edition, Instrument Society of America.

Åström, K. J. and Wittenmark B. (1995). *Adaptive Control*, 2nd Edition, Addison-Wesley, Reading, Massachusetts.

Åström, K. J. (1970). *Introduction to stochastic control theory*, Academic Press, New York.

Abdallah, C. T., Dorato, P. and Bredemann, M. (1994). *New sufficient conditions for strong simultaneous stabilisation*, IFAC Symposium on Robust Control Design, Rio de Janeiro.

Anderson, B. D. O. (1993). *Controller design: Moving from theory to practice*, IEEE Control Systems Magazine, Volume 13, pp. 16-25.

Anderson, B. D. O. and Liu, Y. (1989). *Controller reduction: Concepts and approaches*, IEEE Transactions on Automatic Control, Volume 34, Number 8, pp. 802-812.

Athans, M., Castañon, D., Dunn, K., Greene, C. S., Lee, W. H., Sandell Jr., N. R. and Willsky, A. S. (1977). *The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method - Part 1: Equilibrium flight*, IEEE Transactions on Automatic Control, Volume AC-22, Number 5, pp. 768-780.

Atherton, D. P. (1982). *Non Linear Control Engineering*, Student Edition, Van Nostrand Reinhold, New York.

Balchen, J. G., Jenssen, N. A. and Sælid, S. (1976). *Dynamic positioning using Kalman filtering and optimal control theory*, Proceedings of IFAC/IFIP Symposium on Automation in Offshore Oil Field Operation, pp. 183-188.

Bemporad, A., Casavola, A. and Mosca, E. (1997). *Nonlinear control of constrained linear systems via predictive reference management*, IEEE Transactions on Automatic Control, Volume 42, Number 3, pp. 340-349.

Black, H. S. (1934). *Stabilized feedback amplifiers*, Bell System Technical Journal, Volume 13, pp. 1-18.

Blondel, V., Gevers, M., Mortini, R. and Rupp, R. (1991). *Simultaneous stabilisation of three or more plants: Conditions on the positive real axis do not suffice*, Technical Report 91.78, University Catholique de Louvain, Belgium.

Blondel, V. (1994). *Simultaneous stabilisation of linear systems*, Springer-Verlag, London.

Bode, H. W.. (1940). *Relations between attenuation and phase in feedback amplifier design*, Bell System Technical Journal, Volume 19, pp. 421-454.

Burl, J. B. (1999). *Linear optimal control: H_2 and H_∞ methods*, Addison Wesley, Menlo Park, California.

Burns, R. S. (1995). *The use of artificial neural networks for the intelligent optimal control of surface ships*, IEEE Journal of Oceanic Engineering, Volume 20, Number 1, pp. 65-72.

Camacho, E. F. (1993). *Constrained generalized predictive control*, IEEE Transactions on Automatic Control, Volume AC-38, Number 2, pp. 327-332.

van Calcar, H. and Morgan, M.J. (1975). *Dynamic positioning today*, Proceedings of Oceanology International Conference, Brighton, England, pp. 35-40.

Chable, S., Mahieu, S. and Chiappa, C. (2003). *Design and optimisation of restricted complexity controller: A modal approach for reduced-order controllers*, European Journal of Control, Volume 9, Nmber 1, pp. 39-47.

Chen, G. and Pham, T. T. (2001). *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems*, CRC Press, Boca Raton, Florida.

Clarke, D. W. and Gawthrop, P. J. (1975). *Self tuning controller*, Proceedings of the IEE, Volume 122, Number 9, pp. 929-934.

Clarke, D. W., Mohtadi, C. and Tuffs, P. S. (1987). *Generalized predictive control - Part 1, The basic algorithm, Part 2, Extensions and interpretations*, Automatica, Volume 23, Number 2, pp. 137-160.

Cutler, C. R. and Ramaker, B.L. (1980). *Dynamic matrix control - A computer control algorithm*, Proceedings of Joint Automatic Control Conference, San Francisco, WP5-B.

Cybenko, G. (1989). *Approximation by superpositions of a sigmoidal function*, Mathematics of Control, Signals and Systems, Volume 3, pp. 303-314.

D'Angelo, H. (1970). *Linear time-varying systems: Analysis and synthesis*, Allyn and Bacon, Boston.

Davison, E. (1976). *Multivariable tuning regulators: The feedforward and robust control of a general servomechanism problem*, IEEE Transaction on Automatic Control, Volume 21, Number 1, pp. 35-47.

Djaferis, T. E. (1995). *Robust Control Design: A polynomial approach*, Kluwer Academic Publishers, Boston.

Dong, J. and Brosilow, C. B. (1997). *Design of robust multivariable PID controllers via IMC*, Proceedings of American Control Conference, Albuquerque, New Mexico, pp. 3380-3384.

Doyle, F. J., Ogunnaike, B. A. and Pearson, R. K. (1995). *Nonlinear model-based control using second-order Volterra models*, Automatica, Volume 31, Number 5, pp. 697-714.

Edwards, C. and Spurgeon, S. K. (1998). *Sliding mode control: Theory and applications*, Taylor and Francis, London.

Evans, W. R. (1950). *Control system synthesis by root locus method*, Transactions of the American Institute of Electrical Engineers, Volume 69, pp.66-69.

Eikers, B. and Karim, M. N. (1999). *Process identification with multiple neural network models*, International Journal of Control, Volume 72, Numbers 7/8, pp. 576-590.

Fernandez-Anaya, G., Gonzalez-Trejo, J. and Flores-Tlacuahuac, A. (2001). *Stabilisation with interpolated controllers: Application to the tracking control of continuous stirred tank reactor*, Proceedings of IEEE Conference on Control Applications, Mexico City, pp. 1042-1047.

Fortuna, L. and Muscato, G. (1996). *A roll stabilization system for a monohull ship: Modelling, identification, and adaptive control*, IEEE Transactions on Control Systems Technology, Volume 4, Number 1, pp. 18-28.

Fossen, T. I. (1994). *Guidance and control of ocean vehicles*, John Wiley and Sons.

Fossen, T. I. and Grøvlen, Å(1998). *Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping*, IEEE Transactions on Control Systems Technology, Volume 6, Number 1, pp. 121-128.

Fossen, T. I., Sagatun, S. I. and Sørensen, A. J. (1996). *Identification of dynamically positioned ships*, Control Engineering Practice, Volume 4, Number 3, pp. 369-376.

Fotakis, J., Grimble, M. J. and Kouvaritakis, B. (1982). *A comparison of characteristic locus and optimal designs for dynamic ship positioning systems*, IEEE Transactions on Automatic Control, Volume AC-27, Number 6, pp. 1143-1157.

Franklin, G. F., Powell, J. D. and Workman, M. (1998). *Digital control of dynamic systems*, 3rd Edition, Addison Wesley, Menlo Park, California.

Freeman, R. A. and Kokotović, P. V. (1993). *Design of 'softer' robust nonlinear control laws*, Automatica, Volume 29, Number 6, pp. 1425-1437.

Funahashi, K. (1989). *On the approximate realization of continuous mappings by neural networks*, Neural Networks, Volume 2, pp. 183-192.

Fung, P. T. and Grimble, M. J. (1983). *Dynamic ship positioning using a self-tuning Kalman filter*, IEEE Transaction on Automatic Control, Volume AC-28, Number 3, pp. 339-350.

Garcia, C. E. and Morari, M. (1982). *Internal model Control. 1. A unifying review and some new results*, Industrial and Engineering Chemical Process Design and Development, Volume 21, pp. 308-323.

Gattu, G. and Zafiriou, E. (1992). *Nonlinear quadratic dynamic matrix control with state estimation*, Industrial Engineering Chemical Research, Volume 31, Number 4, pp. 1096-1104.

Gawthrop, P. J., Demircioglu, H. and Siller-Alcala, I. I. (1998). *Multivariable continuous-time generalised predictive control: A state-spece approach to linear and nonlinear systems*, IEE Proceedings - Control Theory Applications, Volume 145, Number 3, pp. 241-250.

Gelb, A., Kasper Jr., J. F., Nash Jr., R. A., Price, C. F. and Sutherland Jr., A. A. (1988). *Applied optimal estimation*, MIT Press, Boston, Massachusetts.

Ghosh, B. (1986). *Simultaneous partial pole placement: A new approach to multimode design*, IEEE Transaction on Automatic Control, Volume AC-31, Number 5, pp. 440-443.

Gopinathan, M., Bošković, J. D., Mehra, R. K. and Rago, C. (1998). *A multiple model predictive scheme for fault-tolerant flight control design*, Proceedings of IEEE CDC Conference, Tampa, Florida, USA, pp. 1376-1381.

Gregory, P. C. (Ed.) (1959). Proceedings of Self Adaptive Flight Control Symposium, Wright-Patterson Air Force Base, Ohio:Wright Air Development Centre.

Grimble, M. J., Patton, R. J. and Wise, D. A. (1980a). *Use of Kalman filtering techniques in dynamic ship positioning systems*, IEE Proceedings, Volume 127, Part D, Number 3, pp. 93-102.

Grimble, M. J. (1980b). *A combined state and state estimate feedback solution to the ship positioning control problem*, Optimal Control Applications & Methods, Volume 1, pp. 55-67.

Grimble, M. J., Patton, R. J. and Wise, D. A. (1980c). *The design of dynamic ship positioning control systems using stochastic optimal control theory*, Optimal Control Applications & Methods, Volume 1, pp. 167-202.

Grimble, M. J. (1994). *Robust industrial control. Optimal design approach for polynomial systems*, Prentice Hall International (UK) Limited, Hemel Hempstead.

Grimble, M. J. and Johnson, M. A. (1988). *Optimal control and stochastic estimation. Theory and applications. Volume 2*, John Wiley, Chichester.

Grimble, M. J. (1999a). *Restricted structure feedforward and feedback stochastic optimal control*, Proceedings of IEEE CDC Conference, Phoenix, Arizona.

Grimble, M. J. (1999b). *Robust control of multiple model systems and application to hot strip rolling*, ICC Report 155, University of Strathclyde.

Grimble, M. J. (2000). *Restricted structure adaptive control including series 2 DOF tracking*, ICC Report 167.

Grimble, M. J. (1990). *LQG predictive optimal control for adaptive applications*, Automatica, Volume 26, Number 3, pp. 949-961.

Grimble, M. J. (1995). *Two degree-of-freedom linear quadratic Gaussian predictive control*, IEE Proceedings, Volume 142, Number 4, pp. 295-306.

Grimble, M. J. (2001a). *Restricted structure predictive optimal control*, ICC Report 187, University of Strathclyde.

Grimble, M. J. (2001b). *Industrial control systems design*, John Wiley, Chichester.

Grimble, M. J. (2002). *Nonlinear quadratic control of discrete-time systems for sandwich systems models*, ICC Report 193, University of Strathclyde.

Grimble, M. J. and Martin, P. (2003). *Time-varying optimal control of a nonlinear system*, Proceedings of IEEE CDC Conference, Hawaii, USA.

Gugercin, S. and Antoulas, A. C. (2000). *A comparative study of 7 algorithms for model reduction*, Proceedings of IEEE CDC Conference, Sydney, Australia, Arizona, pp. 2367-2372.

Haddara, M. R. and Xu, J. (1999). *On the identification of ship coupled heave-pitch motions using neural networks*, Ocean Engineering, Volume 26, pp.381-400.

Harbonn, J. (1971). *The Terebel dynamic positioning system - Results of five years of field work and experiment*, Proceedings of Offshore Technology Conference (OTC), Dallas, Texas, pp. 814-825.

Hardier, G. (1995). *Neural architectures for ship modelling and identification*, Proceedings of International Conference on Artifical Neural Networks, Paris, France.

Hardier, G. (1997). *Recurrent neural networks for ship modelling and control identification*, Proceedings of 11th Ship Control Systems Symposium (SCSS), Southampton, UK.

Haykin, S. (1994). *Neural networks - A comprehensive foundation*, Prentice Hall Inc., Upper Saddle River, New Jersey.

Hjalmarsson, H., Gunnarsson, S. and Gevers, M. (1994). *A convergent iterative restricted complexity control design scheme*, Proceedings of IEEE CDC Conference, Lake Buena Vista, Florida, pp. 1735-1740.

Huzmezan, M. and Maciejowski, J. M. (1998). *Reconfiguration and scheduling in flight using quasi-LPV high-fidelity models and MBPC control*, Proceedings of American Control Conference, Philadelphia.

Hyland, D. C. and Bernstein, D. S. (1984). *The optimal projection equations for fixed-order dynamic compensation*, IEEE Transactions on Automatic Control, Volume AC-29, Number 11, pp. 1034-1037.

Ippoliti G. and Longhi, S. (2004). *Multiple models for adaptive control to improve the performance of minimum variance regulators*, IEE Proceedings, Volume 151, Number 2, pp. 210-217.

Johansen, T. A. and Foss, B. A. (Eds.) (2001). *Multiple model approaches to modelling and control. Special issue*, International Journal of Control, Volume 72, Number 7/8, pp. 575-754.

Jurdjevic, V. and Sussman, H. J. (1972). *Control systems on Lie groups*, Journal of Differential Equations, Volume 12, pp. 313-329.

Kalman, R. E. (1958). *Design of self-optimising control systems*, ASME Transactions, Volume 80, pp. 468-478.

Kalman, R. E. (1960a). *Contributions to the theory of optimal control*, Boletin de la Sociedad Matematica Mexicana, Volume 5, pp. 102-119.

Kalman, R. E. (1960b). *A new approach to linear filtering and prediction problems*, Transactions of the ASME, Journal of Basic Engineering, Volume 82D, March, pp. 35-45.

Katebi, M. R., Grimble, M. J. and Zhang, Y. (1997). *H_∞ robust control design for dynamic ship posititioning*, IEE Proceedings, Volume 144, Number 2, pp. 110-120.

Katebi, M. R., Moradi, M. H. and Johnson, M. A. (2000). *A comparison of stability and performance robustness of multivariable PID tuning methods*, Proceedings of PID 2000 Conference, Barcelona, Spain.

de Keyser, R. and van Cauwenberghe, A. R. (1981). *A self-tuning multistep predictor application*, Automatica, Volume 17, pp. 167-174.

Khalil, H. K. (1992). *Nonlinear systems*, Macmillan, New York.

Kiong, T. K., Quing-Guo, W., Chieh, H. C. with Hägglund, T. J. (1999). *Advances in PID control*, Springer-Verlag, London.

Kouvaritakis, B. and Cannon, M. (Eds.) (2001). *Nonlinear predictive control: Theory and practice*, IEE Control Engineering Series, Number 61, Institution of Electrical Engineers, Stevenage, UK.

Kucera, V. (1979). *Discrete linear control*, John Wiley and Sons, Chichester.

Lainiotis, D. G. (1976). *Partitioning: A unifying framework for adaptive systems, Parts 1 and 2*, Proceedings of the IEEE, Volume 64, Part 1, pp. 1126-1143, Part 2, pp. 1182-1198.

Landau, I. D., Karimi, A. and Hjalmarrson, H. (Eds.) (2003). *Design and optimisation of restricted complexity controllers. Special issue*, European Journal of Control, Volume 9, Number 1, pp. 1-132.

Lewis, F. L., Campos, J. and Selmic, R. R. (2002). *Neuro-fuzzy control of industrial systems with actuator nonlinearities*, SIAM Press, Philadelphia.

Li, S., Lim, K. Y. and Fisher, D. G. (1989). *A state space formulation for model predictive control*, AIChE Journal, Volume 35, Number 2, pp. 241-249.

Ljung, L. (1999). *System identification - Theory for the user*, 2nd Edition, Prentice-Hall, Upper Saddle River, New Jersey.

Loh, A. P., Hang, C. C., Quek, C. K, and Vasnani, V. U. (1993). *Autotuning of multiloop proportional-integral controllers using relay feedback*, Industrial Engineering Chemical Research, Volume 32, pp. 1102-1107.

Luenberger, D. G. (1969). *Optimization by vector space methods*, John Wiley, New York.

Luyben, W. L. (1986). *Simple method for tuning SISO controllers in multivariable systems*, Industrial and Engineering Chemical Process Design and Development, Volume 25, pp. 654-660.

Lyapunov, A. M. (1892). *Probléme generale de la stabilité du mouvement*, French Translation in 1907, Photo-reproduced in Annals of Mathematics, Princeton University Press, Princeton, New Jersey, 1949.

MacFarlane, A. G. J. and Kouvaritakis, B. (1977). *A design technique for linear multivariable feedback systems*, International Journal of Control, Volume 25, Number 6, pp. 837-874.

Maciejowski, J. M. (1989). *Multivariable feedback design*, 1st edition, Addison Wesley, Wokingham, England.

Maciejowski, J. M. (2002). *Predictive control with constraints*, Prentice Hall, Pearson Education, Harlow, England.

Marino, R. and Tomei, P. (1995). *Nonlinear control design*, Prentice Hall, Hemel Hempstead, UK.

Martin, P. (1999). *Wave disturbances in dynamic position control of floating vessels*, MSc Dissertation, Control Systems Centre, UMIST, UK.

Mayne, D. Q. (1979). *Sequential design of linear multivariable systems*, IEE Proceedings, Volume 126, pp. 568-572.

Minorsky, N. (1922). *Directional stability of automatic steered bodies*, Journal of the American Society of Naval Engineers, Volume 32, Number 2, pp. 280-309.

Morgan, M. J. (1978). *Dynamic positioning of offshore vessels*, Petroleum Publishing Company (PPC) Books, Tulsa, Oklahoma.

Morishita, H. M., de Souza Jr., J. R. and Fernandes, C. G. (2001). *Nonlinear dynamics of a FPSO and shuttle vessel in tandem configuration*, Proceedings of 11th International Offshore and Polar Engineering Conference, Stavanger, Norway, Volume 1, pp. 336-342.

Muramatsu, E., Ikeda, M. and Hoshii, N. (1999). *An interpolated controller for stabilisation of a plant with variable operating conditions*, IEEE Transaction on Automatic Control, Volume AC-44, Number 1, pp. 76-81.

Narendra, K. S. and Parthasarathy, K. (1990). *Identification and control of dynamical systems using neural networks*, IEEE Transactions on Neural Networks, Volume 1, Number 1, pp. 4-27.

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

Narendra, K. S. and Balakrishnan, J. (1994a). *Improving transient response of adaptive control systems using multiple models and switching*, IEEE Transaction on Automatic Control, Volume AC-39, Number 9, pp. 1861-1866.

Narendra, K. S. and Balakrishnan, J. (1994b). *Intelligent control using fixed and adaptive models*, Proceedings of 33rd IEEE Control and Decision Conference, Lake Buena Vista, Florida.

Narendra, K. S. and Xiang, C. (2000). *Adaptive control of discrete-time systems using multiple models*, IEEE Transactions on Automatic Control, Volume AC-45, Number 9, pp. 1669-1686.

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

Narendra, K. S. and Driollet, O. A. (2001). *Stochastic adaptive control using multiple models for improved performance in the presence of random disturbances*, International Journal of Adaptive Control an Signal Processing, Volume 15, Number 3, pp. 287-317.

Nordin, M. and Gutman, P. O. (2002). *Controlling mechanical systems with backlash - A survey*, Automatica, Volume 38, Number 10, pp. 1633-1649.

Nyquist, H. (1932). *Regeneration theory*, Bell System Technical Journal, Volume 11, pp. 126-147.

Ordys, A. W. and Grimble, M. J. (1996). *A multivariable dynamic performance predictive control with application to power generation plants*, Proceedings of IFAC World Congress, San Francisco.

Ordys, A. W. and Clarke, D. W. (1993). *A state-space description for GPC controllers*, International Journal of Systems Science, Volume 23, Number 2.

Page, G. F., Gomm, J. B. and Williams, D. (1993). *Application of neural networks to modelling and control*, Chapman and Hall, London, United Kingdom.

Palmor, Z. J., Halevi, Y. and Krasney, N. (1995). *Automatic tuning of decentralized PID controllers for TITO processes*, Automatica, Volume 31, Number 7, pp. 1001-1010.

Penttinen, J. and Koivo, N. H. (1980). *Multivariable tuning regulators for unknown systems*, Automatica, Volume 16, pp. 393-398.

Peterka, V. (1984). *Predictor-based self-tuning control*, Automatica, Volume 20, pp. 39-50.

Pham, D. T. and Liu, X. (1995). *Neural networks for identification, predicition and control*, Springer-Verlag, London, United Kingdom.

Piché, S., Sayyar-Rodsari, B., Johnson, D. and Gerules, M. (2000). *Nonlinear model predictive control using neural networks*, US Naval Oceanographic Office Contract 62306-1042.

Pierson, W. J. and Moskowitz, L. (1963). *A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodsku*, US Naval Oceanographic Office Contract 62306-1042.

Poincaré, H. (1892). *Les methodes nouvelles de la méchanique céleste*, Volume 1, Gauthier-Villars.

Richalet, J., Rault, A., Testaud, L. and Papon, J. (1978). *Model predictive heuristic control: Applications to industrial processes*, Automatica, Volume 14, Number 5, pp. 413-428.

Ricker, N. L. (1990). *Model predictive control with state estimation*, Industrial Engineering Chemical Research, Volume 29, Number 3, pp. 374-382.

Rivera, D. E., Morari, M. and Skogestad, S. (1986). *Internal model control. 4. PID controller design*, Industrial and Engineering Chemical Process Design and Development, Volume 25, pp. 252-265.

Roskilly, A. P. and Mesbahi, E. (1996a). *Marine system modelling using artificial neural networks: an introduction to the theory and practice*, Transactions of the Institute of Marine Engineers, Volume 108, Part 3, pp. 185-201.

Roskilly, A. P. and Mesbahi, E. (1996b). *Artificial neural networks for marine system identification and modelling*, Transactions of the Institute of Marine Engineers, Volume 108, Part 3, pp. 203-213.

Rugh, W. J. (1991). *Analytical framework for gain scheduling*, IEEE Control Systems Magazine, Volume 11, Number 1, pp. 79-84.

Sælid, S., Jenssen, N.A. and Balchen, J.G. (1983). *Design and analysis of a dynamic positioning system based on Kalman filtering and optimal control*, IEEE Transactions on Automatic Control, Volume AC-28, Number 3, pp. 331-339.

Schneider, W. P. (1969). *Dynamic positioning systems*, Proceedings of Offshore Technology Conference (OTC), Dallas, Texas, pp. 184-190.

Shaked, U. (1976). *A general transfer-function approach to the steady-state linear quadratic Gaussian stochastic control problem*, International Journal of Control, Volume 24, Number 6, pp. 771-800.

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

Shamma, J. S. and Athans, M. (1990). *Analysis of gain scheduled control for nonlinear plants*, IEEE Transactions on Automatic Control, Volume AC-35, Number 8, pp. 898-907.

Smith, O. J. M. (1959). *A controller to overcome dead-time*, ISA Transactions, Volume 6, Number 2, pp. 28-33.

Sørenson, A. J., Sagatun, S. I. and Fossen, T. I. (1996). *Design of a dynamic positioning system using model-based control*, Journal of Control Engineering Practice, Volume 4, Number 3, pp. 359-368.

Strand, J. P., Sørenson, A. J., Ronæss, M and Fossen, T. I. (2001). *Position control systems for offshore vessels*, Ocean Engineering Handbook, Editor El-Hawary, F., CRC Press, Chapter 3.

Sunan, H., Kiong, T. K. and Heng, L. T. (2002). *Applied predictive control*, Springer-Verlag, London.

Sussman, H. J. and Jurdjevic, V. (1972). *Controllability of nonlinear systems*, Journal of Differential Equations, Volume 12, pp. 95-116.

Sutton, G. J. and Bitmead, R. R. (1998). *Experiences with model predictive control applied to a nonlinear constrained submarine*, Proceedings of IEEE CDC Conference, Tampa, Florida, USA, pp. 1370-1375.

Taware, A. and Tao, G. (2003). *Control of sandwich nonlinear systems*, Springer-Verlag, Berlin.

Tsang, T. T. C. and Clarke, D. W. (1988). *Generalised predictive control with input constraints*, IEE Proceedings, Part D, Volume 135, Number 6, pp. 451-460.

Unar, M. A. and Murray-Smith, D. J. (1999). *Automatic steering of ships using neural networks*, International Journal of Adaptive Control and Signal Processing, Volume 13, pp. 203-218.

Utkin, V. I. (1992). *Sliding modes in control and optimisation*, Springer, Berlin.

Van Overschee, P. and DeMoor, B. (1996). *Subspace identification of linear systems: Theory, Implementation, Applications*, Kluwer Academic Publishers.

Varga, A. (Ed.) (2004). *Numerical awareness in control: Special section*, IEEE Control Systems Magazine, Volume 24, Number 1, pp. 14-76.

Vidyasagar, M. (1985). *Control systems synthesis: A factorisation approach*, The MIT Press, Cambridge, MA, 1st edition.

Vidyasagar, M. (1993). *Nonlinear systems analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 2nd edition.

Vukic, Z., Borovic, B. and Tovornik, B. (2000). *Adaptive neuro controller for a precise manoeuvring of underwater vehicle*, Proceedings of IFAC Conference on Manoeuvring and Control of Marine Craft, Aalborg, Denmark.

Wang, Q., Zou, B., Lee, T. and Bi, Q. (1997). *Auto-tuning of multivariable PID controllers from decentralized relay feedback*, Automatica, Volume 33, Number 3, pp. 319-330.

Warwick, K., Irwin, G. W. and Hunt, K. J. (1992). *Neural networks for control and systems*, IEE Control Engineering Series Number 46, Peter Peregrinus Ltd., London, UK.

Wiener, N. (1949). *The extrapolation, interpolation and smoothing of stationary time series, with engineering applications*, John Wiley, New York.

Xiros, N. I. and Kyrtatos, N. P. (2001). *A non-linear state-space model of diesel propulsion plant operation using neural nets*, Proceedings of IFAC Conference on Control Applications in Marine Systems (CAMS2001), Glasgow, Scotland, Paper WA3.2.

Yamamoto, I., Daigo, K., Matsuura, M., Yamaguchi, Y., Tanabe, A. and Shimazaki, K. (1998). *A dynamic position system for FPSOs with fuzzy control logic based on a nonlinear programming algorithm*, Proceedings of 17th International conference on offshore mechanics and Arctic engineering, ASME.

Ydstie, B. E. (1984). *Extended horizon adaptive control*, Proceedings of the 9th IFAC World Congress, Budapest, Hungary.

Yuh, J. (1990). *A neural net controller for underwater robotic vehicles*, IEEE Journal of Oceanic Engineering, Volume 15, Number 3, pp. 161-166.

Yu, C. (1999). *Autotuning of PID controllers*, Advances in Industrial Control, Springer-Verlag, London.

Yu, C., Roy, R. J., Kaufman, H. and Bequette, B. W. (1992). *Multiple-model adaptive predictive control of mean arterial pressure and cardiac output*, IEEE Transactions on Biomedical Engineering, Volume 39, Number 8, pp. 765-777.

Yukitomo, M., Yutaka, I., Hino, S., Takahashi, K. and Nagata, K. (1998). *A new PID controller tuning system and it's application to a flue gas temperature control in a gas turbine power plant*, Proceedings of IEEE CCA Conference, Trieste, Italy, pp. 1373-1377.

Zames, G. (1981). *Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses*, IEEE Transactions on Automatic Control, Volume AC-26, pp. 301-320.

Zames, G. (1966a). *On the input-output stability of time-varying nonlinear feedback systems - Part 1: Conditions derived using concepts of loop gain, conicity and positivity*, IEEE Transactions on Automatic Control, Volume AC-11, pp. 465-476.

Zames, G. (1966b). *On the input-output stability of time-varying nonlinear feedback systems - Part 2: Conditions involving circles in the frequency plane and sector nonlinearities*, IEEE Transactions on Automatic Control, Volume AC-11, pp. 228-238.

Zhang, Y., Hearn, G. E. and Sen, P. (1996). *A neural network approach to ship track-keeping control*, IEEE Journal of Oceanic Engineering, Volume 21, Number 4, pp. 513-527.

Zhu, Y. (2002). *Estimation of an N-L-N Hammerstein-Wiener model*, Automatica, Volume 38, pp. 1607-1614.

Zhuang, M. and Atherton, D. P. (1994). *PID controller design for a TITO system*, IEE Proceedings, Part D, Volume 141, pp. 111-120.

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

Appendix A

Nomenclature

Abbreviations

Acronym	Denotes
ARMA	Auto Regressive Moving Average
BLT	Biggest Log Tuning
Com	Combined
DMC	Dynamic Matrix Control
DNA	Direct Nyquist Array
DOF	Degrees Of Freedom
DP	Dynamic Positioning
FPSO	Floating Production, Storage and Offloading
GMV	Generalised Minimum Variance
GNSC	Generalised Nyquist Stability Criterion
GPC	Generalised Predictive Control
H_∞	H Infinity
IMC	Internal Model Control
INA	Inverse Nyquist Array
LMN	Local Model Networks
LQG	Linear Quadratic Gaussian

Acronym	Denotes
LQGPC	Linear Quadratic Gaussian Predictive Control
LPV	Linear Parameter Varying
Mac	Maciejowski
MHI	Mitsubishi Heavy Industries
MIMO	Multi-Input Multi-Output
MLP	Multi Layer Perceptron
MMAC	Multiple Model Adaptive Control
MPC	Model-based Predictive Control
MPHC	Model Predictive Heuristic Control
MRAC	Model Reference Adaptive Control
MSE	Mean Square Error
MSO	Mean Square Output
MV	Minimum Variance
NLP	Non Linear Programming
PID	Proportional, Integral, Derivative
P-K	Penttinen-Koivo
QFT	Quantitative Feedback Theory
QP	Quadratic Programming
RBF	Radial Basis Function
RLS	Recursive Least Squares
SISO	Single-Input Single-Output
SS	Simultaneous Stabilisation
TF	Transfer Function
TITO	Two-Input Two-Output
Z-N	Ziegler-Nichols

Symbols

Symbol	Denotes	Value
a	Neuron output	
$\underline{a}$	Vector of neuron outputs from a layer	
a_i	Amplitude of wave component	
a_{Ai}	Amplitude of fluctuating wind component	
a_1	Plant denominator coefficient	
A	State matrix, Jacobian matrix of partial derivatives	
A	Plant denominator polynomial	
A	Total system state matrix	
A_{AL}	Ship lateral wind area	$5040m^2$
A_{AT}	Ship transverse wind area	$880m^2$
A_c	Reference denominator polynomial matrix	
A_{dp}	Disturbance denominator matrix	
A_g	Linear transfer function block denominator matrix	
A_m	Plant model state matrix	
A_p	Plant denominator matrix	
A_q	Error weighting denominator term	
A_r	Input weighting denominator term	
A_r	Reference model state matrix	
A_{rp}	Reference denominator matrix	
A_R	Reference generation state matrix	
A_1	Denominator matrix of weighted plant	
A_{1g}	Denominator matrix of weighted linear TF block	
b	Bias term	
$\underline{b}$	Vector of bias terms	
b_1	Plant numerator coefficient	
B	(Total System) Input matrix	

Symbol	Denotes	Value
B	Plant numerator polynomial	
B_g	Linear transfer function block numerator matrix	
B_m	Plant model input matrix	
B_p	Plant numerator matrix	
B_q	Dynamic error weighting numerator	
B_{rp}	Reference numerator matrix	
B_1	Numerator matrix of weighted plant	
B_{1g}	Numerator matrix of weighted linear TF block	
$\underline{c}$	Vector of current velocites relative to ship body	
c_1	Disturbance numerator coefficient	
C	Output matrix	
C	Total system output matrix	
C_d	Disturbance numerator polynomial	
C_{dp}	Disturbance numerator matrix	
C_m	Plant model output matrix	
C_r	Reference model output matrix	
C_R	Reference generation output matrix	
$C_{WD}(\mu_W)$	Wave drifting force coefficients	
C_0	Controller	
C_{0d}	Controller denominator polynomial	
C_{0f}	Full-order controller	
C_{0n}	Controller numerator polynomial	
C_{0r}	Restricted structure controller	
C_{11}, C_{21}	Total system output matrix partition	
d	Ship draught	14.6m
d	Distance from FPSO stern to shuttler tanker bow	
d	Disturbance signal	
$\underline{d}$	Vector of wind and wave disturbances	

Symbol	Denotes	Value
d_{ta}	Distance from ship turret to anchoring	
d_{to}	Distance from ship turret to body coordinate origin	29.75m
d_1	Amplitude of estimation error	
D	Total system disturbance input matrix	
D_c, D_{cp}	Control spectral factor	
D_c	"Operator spectral factor"	
D_{dp}	Disturbance spectral factor	
D_f	Filter spectral factor	
D_m	Plant model disturbance input matrix	
D_r	Reference model disturbance input matrix	
D_R	Reference generation disturbance input matrix	
D_{11}, D_{12}	Total system disturbance input matrix partition	
e	Error signal	
$\underline{e}$	Vector of position error relative to ship body	
e_x	Surge position error	
e_y	Sway position error	
e_Ψ	Heading position error	
E	Disturbance input matrix	
E	Reference numerator polynomial	
E	Generalised output error	
$\underline{E}$	Vector of position errors relative to earth	
$E[.]$	Expectation	
E_x	Position error in earth X coordinate	
E_y	Position error in earth Y coordinate	
E_Ψ	Heading position error	
f	Activation function	
f	Difference signal between reference and disturbance	
$f(\underline{z})$	Surge acceleration function in terms of $\underline{z}$	

Symbol	Denotes	Value
$f(\mu_W)$	Wave exciting force coefficient	
f_1, f_2	Nonlinear functions	
F_R	Position reference filter	
F_0	(Operator) Diophantine equation solution	
F_{1p}^c	Regulating Diophantine equation solution	
F_{1p}^f	Filtering Diophantine equation solution	
$\mathcal{F}_1, \mathcal{F}_2$	Nonlinear functions	
g	Gravitational constant	$9.8\text{m}/s^2$
$g(\underline{z})$	Sway acceleration function in terms of $\underline{z}$	
g_d	SISO ship TF from disturbances to ship velocity	
g_V	SISO ship TF from thruster force to ship velocity	
G	Plant transfer function	
G	Estimator input matrix	
G	Linear transfer function block	
G_d	MIMO ship TF from disturbances to ship velocities	
G_V	MIMO ship TF from thruster forces to ship velocities	
G_0	(Operator) Diophantine equation solution	
G_{1p}^c	Regulating Diophantine equation solution	
G_{1p}^f	Filtering Diophantine equation solution	
$h(\underline{z})$	Yaw acceleration function in terms of $\underline{z}$	
H	Estimator state matrix	
H_m	Generalised plant output matrix	
H_p	Prediction horizon	
H_q	Dynamic error weighting	
H_r	Generalised reference matrix	
H_r	Dynamic input weighting	
H_u	Input horizon	
H_R	Generalised reference matrix	

Symbol	Denotes	Value
H_0	(Operator) Diophantine equation solution	
H_{1p}^c	Regulating Diophantine equation solution	
H_{1p}^f	Filtering Diophantine equation solution	
i	Wind component index, Row index, Neuron input index	
I	Identity matrix	
I	Summand of infinite-time cost	
I_T	Summand over time $2T$	
I_{zz}	Ship moment of inertia	73MTm
j	Imaginary number	$\sqrt{-1}$
j	Column index, Cable index, Multiple model index	
j	Prediction step index	
J	Value of cost function	
J_{csl}	Cost function causal on controller	
J_k	Cost of kth training error	
J_{min}	Minimum cost	
J_p	Cost dependent on control input signal, u	
J_{p0}	Cost independent of control inputs vector, U	
J_t	Cost at time t	
J_{zz}	Ship added moment of inertia	21MTm
J_0	Approximated cost in restricted structure calculation	
J_{0j}	Approximated cost of jth model	
k	Data set index	
k_d	Scalar derivative gain	
k_i	Scalar integral gain	
k_p	Scalar proportional gain	
K	Controller transfer function	
K_d	Matrix derivative gain	
K_d	Controller denominator	

Symbol	Denotes	Value
K_i	Matrix integral gain	
K_n	Controller numerator	
K_p	Matrix proportional gain	
K_r	Restricted structure controller	
K_P	Controller transfer function matrix in position loop	
K_V	Controller transfer function matrix in velocity loop	
l	Row index	
l	Number of inputs	
L	Ship length	
L_C	Cable length	1714m
L_{FPSO}	Length of FPSO	256m
L_{ST}	Length of shuttle tanker	230m
m	Ship mass	12400T
m	Wave component index	
m	Number of inputs	
m	Disturbance-free output	
m_g	Input to output nonlinearity	
m_x	Surge added mass	727T
m_y	Sway added mass	8290T
M	Constant real gain matrix	
M	Control sensitivity function	
n	Wave component index, Notch filter transfer function	
n	Value after summing junction, Number of neuron inputs	
n	Number of states	
n	Index of iteration steps	
$\underline{n}$	Vector of values after summing junction	
N	Diagonal matrix of notch filter transfer functions	
N	Number of wave components, Yaw moment	

Symbol	Denotes	Value
N	Number of fluctuating wind components	
N	Number of frequency points	
N	Estimator disturbance input matrix	
$N_{hwsFPSO}$	Moment on FPSO due to hawser	
N_{hwsST}	Moment on shuttle tanker due to hawser	
N_r	Hydrodynamic derivative	-0.0339
N_{rr}	Hydrodynamic derivative	-0.0372
N_{trt}	Yaw moment on FPSO due to turret	
N_{tug}	Yaw moment on shuttle tanker due to tugboat	
N_v	Hydrodynamic derivative	-0.1495
N_{vr}	Hydrodynamic derivative	0.1486
N_{vv}	Hydrodynamic derivative	0.0727
N_A	Yaw wind moment	
N_F	Combined moment due to thrusters, turret, hawser and tugboat	
N_H	Yaw hydrodynamic moment	
N_T	Yaw thruster moment	
$N_T^{'}$	Initial guess at N_{T0} for search algorithm	
$N_{T(FPSO)}$	Yaw moment on FPSO	
$N_{T(ST)}$	Yaw moment on shuttle tanker	
N_{T0}	Yaw thruster moment at operating point	
N_W	Yaw wave moment	
p	Scalar tuning parameter, Neuron input	
$\underline{p}$	Vector of neuron inputs to a layer	
$\underline{p}$	Vector of neuron inputs	
p_j	Probability of jth model	
P	Diagonal matrix of model probabilities	
q_c	Error weighting	

Symbol	Denotes	Value
Q	Number of input/output pairs	
Q	State weighting	
$\tilde{Q}$	Diagonal matrix of weights on predicted error signal	
Q_c	Error weighting	
Q_j	Error weighting j steps ahead	
Q_n	Error weighting numerator	
Q_w	Process noise covariance	
r, r_{abs}	Yaw rate	
r	Reference signal	
Δr	Change in r from operating point, r_0	
r_c	Input weighting	
r_h	Generalised reference signal	
r_0	Yaw rate at operating point	
R	Input weighting	
$\underline{R}$	Vector of position references relative to earth	
$\tilde{R}$	Diagonal matrix of weights on predicted input signal	
R_c	Input weighting	
R_{f0}	Reference model output noise covariance	
R_{f1}	Plant model output noise covariance	
R_h	Vector of reference signals	
R_j	Input weighting j steps ahead	
R_n	Input weighting numerator	
R_v	Measurement noise covariance	
$\mathbb{R}$	Set of real numbers	
s	Laplace transform complex number	
s_V	Velocity loop sensitivity function	
S	Sensitivity function	
S_P	Postion loop sensitivity matrix	

Symbol	Denotes	Value
S_V	Velocity loop sensitivity matrix	
S^λ	Number of neurons in λth layer	
t	Time, Sample instant	
$\underline{t}_k$	kth desired target output	
t_V	Velocity loop complementary sensitivity function	
T	Hawser tension	
T	Sample time	
T	Transformation	
T_{BE}	Transformation from ship body to earth coordinates	
T_{EB}	Transformation from earth to ship body coordinates	
T_j	Tension in jth cable	
T_P	Position loop complementary sensitivity matrix	
T_X	Tension X component	
T_Y	Tension Y component	
T_V	Velocity loop complementary sensitivity matrix	
T_1, T_2	Control independent cost elements	
T_1^+, T_d^+	Stable integrand	
T_1^-, T_d^-	Unstable integrand	
u	Surge velocity relative to ship body	
u	Input signal	
$\underline{u}$	Vector of inputs, Vector of thruster forces	
$\underline{u}$	Vector of inputs at different time steps	
Δu	Change in u from operating point, u_0	
u_{abs}	Surge velocity relative to earth	
u_c	Velocity of current in negative surge direction	
u_g	Output from input nonlinearity	
u_A	Surge relative wind velocity	
u_{A0}	Fluctuating wind velocity	

Symbol	Denotes	Value
u_V	Velocity loop control sensitivity function	
u_0	Surge velocity relative to ship body at operating point	
U	Ship translational velocity vector magnitude	
U	Vector of predicted plant inputs	
U	Half-width of deadzone	
U^f	Vector of future input vectors	
U_A	Wind velocity relative to earth	
U_{A0}	Wind velocity average relative to earth	
U_C	Magnitude of current velocity	
U_P	Position loop control sensitivity matrix	
U_V	Velocity loop control sensitivity matrix	
v	Sway velocity relative to ship body	
v	Measurement noise signal	
Δv	Change in v from operating point, v_0	
v_{abs}	Sway velocity relative to earth	
v_c	Velocity of current in negative sway direction	
v_m	Plant model output noise signal	
v_r	Reference model output disturbance signal	
v_A	Sway relative wind velocity	
v_V	Velocity loop control sensitivity function	
v_0	Sway velocity relative to ship body at operating point	
V	Half-width of backlash	
V_A	Magnitude of relative wind velocity vector	
V_P	Position loop control sensitivity matrix	
V_V	Velocity loop control sensitivity matrix	
w	Neuron weighting	
w	Process noise vector	
$\underline{w}$	Vector of neuron weightings	

Symbol	Denotes	Value
W	Matrix of neuron weights	
W	Plant transfer function	
W	Vector of future plant input disturbance signals	
$\bar{W}$	Transfer function from input to total state vector	
W_d	Disturbance transfer function	
$\bar{W}_d$	Transfer function from disturbance to total state vector	
W_r	Reference transfer function	
$\bar{W}_r$	Transfer function from reference to total state vector	
x	State vector	
x	Vector of restricted structure controller gains	
$\underline{x}$	Ship velocity vector (or change in) relative to ship body	
$\underline{x}$	Vector of model data for RLS	
$\hat{x}$	State estimate	
x_m	Plant model state vector	
$x_{r(H_p)}$	Reference model state	
x_R	Vector of reference model states	
X	Position in earth coordinate system, Surge force	
X	Stacked vector of $\underline{x}$'s	
X	Total system state vector	
$X_{hwsFPSO}$	Force on FPSO due to hawser	
X_{hwsST}	Force on shuttle tanker due to hawser	
X_{rr}	Hydrodynamic derivative	-0.0194
X_{trt}	Surge force on FPSO due to turret	
X_{tug}	Surge force on shuttle tanker due to tugboat	
X_{uu}	Hydrodynamic derivative	0.126
X_{vr}	Hydrodynamic derivative	-0.0664
X_{vv}	Hydrodynamic derivative	-0.0057
X_A	Surge wind force	

Symbol	Denotes	Value
X_F	Combined surge force due to thrusters, turret, hawser and tugboat	
X_{FPSO}	FPSO position relative to earth	
X_H	Surge hydrodynamic force	
X_R	Reference position in earth coordinate system	
X_{ST}	Shuttle tanker position relative to earth	
X_T	Surge thruster force	
$X_T^{'}$	Initial guess at X_{T0} for search algorithm	
$X_{T(FPSO)}$	Surge thruster force on FPSO	
$X_{T(ST)}$	Surge thruster force on shuttle tanker	
X_{T0}	Surge thruster force at operating point	
X_W	Surge wave force	
y	Output signal	
$\underline{y}$	Vector of ship velocities relative to earth	
$\underline{Y}$	Vector of position coordinates relative to earth	
y_h	Generalised output signal	
Y	Position in earth coordinate system, Sway force	
Y_h	Vector of predicted generalised plant outputs	
$Y_{hwsFPSO}$	Force on FPSO due to hawser	
Y_{hwsST}	Force on shuttle tanker due to hawser	
Y_r	Hydrodynamic derivative	0.0659
Y_{rr}	Hydrodynamic derivative	0.102
Y_{trt}	Sway force on FPSO due to turret	
Y_{tug}	Sway force on shuttle tanker due to tugboat	
Y_v	Hydrodynamic derivative	-0.218
Y_{vr}	Hydrodynamic derivative	-0.391
Y_{vv}	Hydrodynamic derivative	-0.908
Y_A	Sway wind force	

Symbol	Denotes	Value
Y_F	Combined sway force due to thrusters, turret, hawser and tugboat	
Y_{FPSO}	FPSO position relative to earth	
Y_H	Sway hydrodynamic force	
Y_R	Reference position in earth coordinate system	
Y_{ST}	Shuttle tanker position relative to earth	
Y_T	Sway thruster force	
$Y_T^{'}$	Initial guess at Y_{T0} for search algorithm	
$Y_{T(FPSO)}$	Sway thruster force on FPSO	
$Y_{T(ST)}$	Sway thruster force on shuttle tanker	
Y_{T0}	Sway thruster force at operating point	
Y_W	Sway wave force	
$\underline{z}$	Vector of ship body velocities, heading and thruster forces	
$\underline{z}_0$	$\underline{z}$ vector operating point	
z_m	Plant model observation signal	
z_r	Reference model output signal	
z^{-1}	Unit delay operator, Complex number	
Z	Total system output vector	
$\mathbb{Z}$	Set of integers	
β_A	Incident angle of wind	
β_C	Incident angle of current	
β_W	Incident angle of wave	
δ	Scalar tuning parameter	
δ	Neural network delta term for change in J with n	
$\delta_{t\tau}$	Kronecker delta	
ϵ	Scalar tuning parameter	
ϵ	RLS estimation error based on previous data	
ε	Small positive number	

Symbol	Denotes	Value
ε_{Ai}	Phase angle of fluctuating wind component	
ε_i	Phase angle of wave component	
$\varepsilon_(\mu_W)$	Phase angle of wave exciting force	
ζ	White noise source	
η	Learning rate	
θ	Angle	
θ	Hawser angle	
θ_{ta}	Angle of cable from turret to anchoring	
Θ	Diagonal matrix of angles	
$\underline{\Theta}$	Vector of model coefficients for RLS	
$\underline{\hat{\Theta}}$	Estimate of $\underline{\Theta}$	
κ_i	Wave number of wave component, $\kappa_i = \omega_i^2/g$	
λ	Neural network layer index	
λ	Forgetting factor	
Λ	Number of network layers	
μ_A	Relative wind direction	
μ_C	Angle of ship velocity vector relative to ship body	
μ_W	Angle between incident waves and ship heading	
ξ	White noise source	
ξ	(Normalised) Estimation error	
ξ	Total system disturbance vector	
ξ_m	Plant model disturbance signal	
ξ_r	Reference model disturbance input signal	
ρ	Water density	1.02T$/m^3$
ρ_A	Air density	0.00125T$/m^3$
Φ_{ee}	Error power spectrum	
Φ_{ff}	Reference and disturbance power spectrum	
Φ_{uu}	Input power spectrum	

Symbol	Denotes	Value
Ψ	Ship heading angle	
Ψ_{FPSO}	FPSO heading angle	
Ψ_R	Reference ship heading angle	
Ψ_{ST}	Shuttle tanker heading angle	
ω	Frequency	
ω_{Ai}	Frequency of fluctuating wind component	
ω_b	Bandwidth frequency	
ω_i	Frequency of wave component	
ω_n	Notch frequency	
$\{.\}$	Sequence	
$\{.,.,.\}$	Set	
$*$	Convolution operator	
$\|\|.\|\|$	Norm	
$< .,. >$	Inner product	

Appendix B

Linear Ship Model

To design linear controllers for an operating point of a nonlinear system, a linear model is required which adequately describes the particular operating region. The partial derivatives corresponding to the linearisation of the ship model in Section 2.2.6 are given below.

Surge axis partial derivatives

$$\frac{\partial f}{\partial u} = \frac{-\rho L d X_{uu}|u_0|}{m+m_x} = -0.00372|u_0| \tag{B.1}$$

$$\frac{\partial f}{\partial v} = \frac{\frac{1}{2}\rho L d(2X_{vv}v_0 + LX_{vr}r_0)}{m+m_x} = -0.251r_0 - 0.000169v_0 \tag{B.2}$$

$$\frac{\partial f}{\partial r} = \frac{\frac{1}{2}\rho L^2 d(2LX_{rr}r_0 + X_{vr}v_0)}{m+m_x} = -0.251v_0 - 37.6r_0 \tag{B.3}$$

$$\frac{\partial f}{\partial X_T} = \frac{1}{m+m_x} = 7.60 \times 10^{-5} \tag{B.4}$$

Sway axis partial derivatives ($|\Psi - \beta_C| < \pi/4$)

$$\begin{aligned}\frac{\partial g}{\partial u} &= \frac{\frac{1}{2}\rho L d(Y_v v_0 u_0 + LY_r r_0 u_0)}{(m+m_y)U_0} \\ &= -0.00204 v_0 u_0/U_0 + 0.158 r_0 u_0/U_0\end{aligned} \tag{B.5}$$

$$\frac{\partial g}{\partial v} = \frac{\frac{1}{2}\rho L d(Y_v(2v_0^2 + u_0^2) + 2Y_{vv}|v_0|U_0 + LY_r r_0 v_0 + LY_{vr}|r_0|U_0)}{(m+m_y)U_0}$$
$$= -0.00204(2v_0^2 + u_0^2)/U_0 - 0.0170|v_0| + 0.158 r_0 v_0/U_0 - 0.939|r_0| \quad \text{(B.6)}$$

$$\begin{aligned}\frac{\partial g}{\partial r} &= \frac{\frac{1}{2}\rho L^2 d(Y_r U_0 + 2LY_{rr}|r_0| \pm Y_{vr} v_0)}{m + m_y} \\ &= 0.158U_0 + 126|r_0| \pm 0.939v_0\end{aligned} \quad \text{(B.7)}$$

$$\frac{\partial g}{\partial Y_T} = \frac{1}{m+m_y} = 4.83 \times 10^{-5} \quad \text{(B.8)}$$

Sway axis partial derivatives ($|\Psi - \beta_C| \geq \pi/4$)

$$\begin{aligned}\frac{\partial g}{\partial u} &= \frac{0.35\rho L d U_0(v_0 cos(\mu_C) - 2u_0 sin(\mu_C)) + \frac{1}{2}\rho L^2 d Y_r r_0 u_0}{(m+m_y)U_0} \\ &= 0.00658(v_0 cos(\mu_C) - 2.96u_0 sin(\mu_C)) + 0.158 r_0 u_0/U_0\end{aligned} \quad \text{(B.9)}$$

$$\frac{\partial g}{\partial v} = \frac{-0.35\rho L d U_0(u_0 cos(\mu_C) + 2v_0 sin(\mu_C)) + \frac{1}{2}\rho L^2 d(Y_r r_0 v_0 + Y_{vr}|r_0|U_0)}{(m+m_y)U_0}$$
$$= -0.00658(u_0 cos(\mu_C) + 2.96 v_0 sin(\mu_C)) + 0.158 r_0 v_0/U_0 - 0.939|r_0| \quad \text{(B.10)}$$

$$\begin{aligned}\frac{\partial g}{\partial r} &= \frac{\frac{1}{2}\rho L^2 d(Y_r U_0 + 2LY_{rr}|r_0| \pm Y_{vr} v_0)}{m + m_y} \\ &= 0.158U_0 + 126|r_0| \pm 0.939v_0\end{aligned} \quad \text{(B.11)}$$

$$\frac{\partial g}{\partial Y_T} = \frac{1}{m+m_y} = 4.83 \times 10^{-5} \quad \text{(B.12)}$$

Yaw axis partial derivatives ($|\Psi - \beta_C| < \pi/4$)

$$\begin{aligned}\frac{\partial h}{\partial u} &= \frac{\frac{1}{2}\rho L^2 d(N_v v_0 u_0 + L N_r r_0 u_0)}{(I_{zz}+J_{zz})U_0} \\ &= -0.0000782 v_0 u_0/U_0 - 0.00456 r_0 u_0/U_0\end{aligned} \tag{B.13}$$

$$\begin{aligned}\frac{\partial h}{\partial v} &= \frac{\frac{1}{2}\rho L^2 d(N_v(2v_0^2 + u_0^2) + 2N_{vv}|v_0|U_0 + L N_r r_0 v_0 + L N_{vr}|r_0|U_0)}{(I_{zz}+J_{zz})U_0} \\ &= -0.0000782(2v_0^2 + u_0^2)/U_0 + 0.0000765|v_0| - 0.00456 r_0 v_0/U_0 + 0.0198|r_0|\end{aligned} \tag{B.14}$$

$$\begin{aligned}\frac{\partial h}{\partial r} &= \frac{\frac{1}{2}\rho L^3 d(N_r U_0 + 2L N_{rr}|r_0| \pm N_{vr} v_0)}{I_{zz}+J_{zz}} \\ &= -0.00456 U_0 - 2.57|r_0| \pm 0.0198 v_0\end{aligned} \tag{B.15}$$

$$\frac{\partial h}{\partial N_T} = \frac{1}{I_{zz}+J_{zz}} = 1.06 \times 10^{-8} \tag{B.16}$$

Yaw axis partial derivatives ($|\Psi - \beta_C| \geq \pi/4$)

$$\begin{aligned}\frac{\partial h}{\partial u} &= \frac{0.1\rho L^2 d U_0(v_0 cos(2\mu_C) - u_0 sin(2\mu_C)) + \frac{1}{2}\rho L^3 d N_r r_0 u_0}{(I_{zz}+J_{zz})U_0} \\ &= 0.000105(v_0 cos(2\mu_C) - u_0 sin(2\mu_C)) - 0.00456 r_0 u_0/U_0\end{aligned} \tag{B.17}$$

$$\begin{aligned}\frac{\partial h}{\partial v} &= \frac{-0.1\rho L^2 d U_0(u_0 cos(2\mu_C) + v_0 sin(2\mu_C)) + \frac{1}{2}\rho L^3 d(N_r r_0 v_0 + N_{vr}|r_0|U_0)}{(I_{zz}+J_{zz})U_0} \\ &= -0.000105(u_0 cos(2\mu_C) + v_0 sin(2\mu_C)) - 0.00456 r_0 v_0/U_0 + 0.0198|r_0|\end{aligned} \tag{B.18}$$

$$\begin{aligned}\frac{\partial h}{\partial r} &= \frac{\frac{1}{2}\rho L^3 d(N_r U_0 + 2L N_{rr}|r_0| \pm N_{vr} v_0)}{I_{zz}+J_{zz}} \\ &= -0.00456 U_0 - 2.57|r_0| \pm 0.0198 v_0\end{aligned} \tag{B.19}$$

$$\frac{\partial h}{\partial N_T} = \frac{1}{I_{zz} + J_{zz}} = 1.06 \times 10^{-8} \tag{B.20}$$

Appendix C

Cost Minimisation for Chapter 5

Substituting (5.30) into (5.33):

$$\begin{aligned} u(z^{-1}) &= -K(Wu(z^{-1}) + W_d\xi_m(z^{-1})) - Kv_m(z^{-1}) \\ &= S(-KW_d\xi_m(z^{-1}) - Kv_m(z^{-1})) \qquad (C.1) \end{aligned}$$

where sensitivity function $S = (I + KW)^{-1}$.

Substituting (C.1) into (5.28):

$$\begin{aligned} X(z^{-1}) &= \bar{W}(-SKW_d\xi_m(z^{-1}) - SKv_m(z^{-1})) + \bar{W}_r\xi_r(z^{-1}) + \bar{W}_d\xi_m(z^{-1}) \\ &= \bar{W}_r\xi_r(z^{-1}) + (I - \bar{W}MC_{21})\bar{W}_d\xi_m(z^{-1}) - \bar{W}Mv_m(z^{-1}) \qquad (C.2) \end{aligned}$$

where $M = SK$.

From Grimble and Johnson (1988), $\Phi_{yy}(z^{-1}) = W(z^{-1})\Phi_{uu}(z^{-1})W^*(z^{-1})$ when $y(z^{-1}) = W(z^{-1})u(z^{-1})$. The desired power spectra terms can now be defined as:

$$\begin{aligned} \Phi_{XX} = XX^* &= (\bar{W}MC_{21} - I)\bar{W}_d\bar{W}_d^*(C_{21}^*M^*\bar{W}^* - I) \\ &\quad + \bar{W}_r\bar{W}_r^* + \bar{W}MR_{f1}M^*\bar{W}^* \qquad (C.3) \end{aligned}$$

$$\Phi_{uu} = uu^* = SKW_dW_d^*K^*S^* + SKR_{f1}K^*S^* \tag{C.4}$$

$$\Phi_{uX} = uX^* = SKW_d\bar{W}_d^*(C_{21}^*M^*\bar{W}^* - I) + SKR_{f1}M^*\bar{W}^* \tag{C.5}$$

Expand and simplify to obtain:

$$\begin{aligned} \Phi_{XX} &= \bar{W}MW_dW_d^*M^*\bar{W}^* - \bar{W}_dW_d^*M^*\bar{W}^* - \bar{W}MW_d\bar{W}_d^* \\ &\quad + \bar{W}_d\bar{W}_d^* + \bar{W}_r\bar{W}_r^* + \bar{W}MR_{f1}M^*\bar{W}^* \end{aligned} \tag{C.6}$$

$$\Phi_{uu} = M(W_dW_d^* + R_{f1})M^* \tag{C.7}$$

$$\Phi_{uX} = MW_dW_d^*M^*\bar{W}^* - MW_d\bar{W}_d^* + MR_{f1}M^*\bar{W}^* \tag{C.8}$$

The integrand in (5.27), denoted I_c, becomes:

$$\begin{aligned} I_c &= trace\{\bar{Q}_c\Phi_{XX}(z^{-1}) + 2\bar{G}_c\Phi_{uX}(z^{-1}) + \bar{R}_c\Phi_{uu}(z^{-1})\} \\ &= trace\{\bar{Q}_c(\bar{W}MW_dW_d^*M^*\bar{W}^* - \bar{W}_dW_d^*M^*\bar{W}^* - \bar{W}MW_d\bar{W}_d^* + \bar{W}_d\bar{W}_d^* \\ &\quad + \bar{W}_r\bar{W}_r^* + \bar{W}MR_{f1}M^*\bar{W}^*) + 2\bar{G}_c(MW_dW_d^*M^*\bar{W}^* - MW_d\bar{W}_d^* \\ &\quad + MR_{f1}M^*\bar{W}^*) + \bar{R}_cM(W_dW_d^* + R_{f1})M^*\} \\ &= trace\{MW_dW_d^*M^*(\bar{W}^*\bar{Q}_c\bar{W} + \bar{R}_c + \bar{W}^*\bar{G}_c + \bar{G}_c^*\bar{W}) \\ &\quad + MR_{f1}M^*(\bar{W}^*\bar{Q}_c\bar{W} + \bar{R}_c + \bar{W}^*\bar{G}_c + \bar{G}_c^*\bar{W}) - MW_d\bar{W}_d^*(\bar{Q}_c\bar{W} + \bar{G}_c) \\ &\quad - (\bar{W}^*\bar{Q}_c^* + \bar{G}_c^*)\bar{W}_dW_d^*M^*\} + trace\{\bar{Q}_c(\bar{W}_r\bar{W}_r^* + \bar{W}_d\bar{W}_d^*)\} \end{aligned} \tag{C.9}$$

To simplify this expression, the following spectral factors are defined:

$$Y_c^*Y_c = \bar{A}_{1p}^{*-1}D_{cp}^*D_{cp}\bar{A}_{1p}^{-1} = \bar{W}^*\bar{Q}_c\bar{W} + \bar{R}_c + \bar{W}^*\bar{G}_c + \bar{G}_c^*\bar{W} \tag{C.10}$$

$$Y_d Y_d^* = A_{dp}^{-1} D_{dp} D_{dp}^* A_{dp}^{*-1} = W_d W_d^* + R_{f1} \tag{C.11}$$

so that:

$$\begin{aligned} I_c &= trace\{M Y_d Y_d^* M^* Y_c^* Y_c - M W_d \bar{W}_d^* \bar{Q}_w - \bar{Q}_w^* \bar{W}_d W_d^* M^*\} \\ &\quad + trace\{\bar{Q}_c(\bar{W}_r \bar{W}_r^* + \bar{W}_d \bar{W}_d^*)\} \end{aligned} \tag{C.12}$$

where $\bar{Q}_w = \bar{Q}_c \bar{W} + \bar{G}_c$. Following a conventional completing-the-squares argument, the integrand may be rewritten:

$$\begin{aligned} I_c &= trace\{(Y_c M Y_d - Y_c^{*-1} \bar{Q}_w^* \bar{W}_d W_d^* Y_d^{*-1})(Y_d^* M^* Y_c^* - Y_d^{-1} W_d \bar{W}_d^* \bar{Q}_w Y_c^{-1})\} \\ &\quad - trace\{Y_c^{*-1} \bar{Q}_w^* \bar{W}_d W_d^* Y_d^{*-1} Y_d^{-1} W_d \bar{W}_d^* \bar{Q}_w Y_c^{-1} - \bar{Q}_c(\bar{W}_r \bar{W}_r^* + \bar{W}_d \bar{W}_d^*)\} \\ &= trace\{T_d T_d^*\} - I_{c0} \end{aligned} \tag{C.13}$$

where I_{c0} is independent of the control law. Before the cost can be minimised, $T_d T_d^*$ must be split into stable and unstable parts. This is achieved using two Diophantine equations, but first T_d must be expanded using the polynomials defined earlier.

$$Y_c M Y_d = D_{cp} \bar{A}_{1p}^{-1} M A_{dp}^{-1} D_{dp} \tag{C.14}$$

$$\begin{aligned} Y_c^{*-1} \bar{Q}_w^* \bar{W}_d W_d^* Y_d^{*-1} &= D_{cp}^{*-1} \bar{A}_{1p}^* (\bar{A}_{1p}^{*-1} \bar{B}_{1p}^* \bar{Q}_c + \bar{G}_c^*) \bar{A}_p^{-1} z^{-1} D_{12} C_{dp}^* A_{dp}^{*-1} A_{dp}^* D_{dp}^{*-1} \\ &= D_{cp}^{*-1} (\bar{B}_{1p}^* \bar{Q}_c + \bar{A}_{1p}^* \bar{G}_c^*) \bar{A}_p^{-1} z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1} \end{aligned} \tag{C.15}$$

Substituting the Diophantine equation:

$$z^{-g_1} D_{cp}^* G_{1p}^c + F_{1p}^c \bar{A}_p = (\bar{B}_{1p}^* \bar{Q}_c + \bar{A}_{1p}^* \bar{G}_c^*) z^{-g_1} \tag{C.16}$$

into (C.15), T_d may therefore be denoted as:

$$\begin{aligned} T_d &= Y_c M Y_d - Y_c^{*-1}\bar{Q}_w^* \bar{W}_d W_d^* Y_d^{*-1} \\ &= (D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - D_{cp}^{*-1}(\bar{B}_{1p}^* \bar{Q}_c + \bar{A}_{1p}^* \bar{G}_c^*)\bar{A}_p^{-1} z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1}) \\ &= (D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - (G_{1p}^c \bar{A}_p^{-1} + z^{g_1} D_{cp}^{*-1} F_{1p}^c) z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1}) \\ &= (D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - G_{1p}^c \bar{A}_p^{-1} z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1} \\ &\quad - z^{g_1} D_{cp}^{*-1} F_{1p}^c z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1}) \end{aligned} \tag{C.17}$$

Substituting in a further Diophantine equation:

$$z^{-g_2} G_{1p}^f D_{dp}^* + \bar{A}_p F_{1p}^f = D_{12} C_{dp}^* z^{-g_2} \tag{C.18}$$

produces:

$$\begin{aligned} T_d &= (D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - G_{1p}^c \bar{A}_p^{-1} z^{-1} (G_{1p}^f D_{dp}^* + \bar{A}_p F_{1p}^f z^{g_2}) D_{dp}^{*-1} \\ &\quad - z^{g_1} D_{cp}^{*-1} F_{1p}^c z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1}) \\ &= (D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - G_{1p}^c \bar{A}_p^{-1} z^{-1} G_{1p}^f \\ &\quad - G_{1p}^c z^{-1} F_{1p}^f z^{g_2} D_{dp}^{*-1} - z^{g_1} D_{cp}^{*-1} F_{1p}^c z^{-1} D_{12} C_{dp}^* D_{dp}^{*-1}) \\ &= [D_{cp}\bar{A}_{1p}^{-1} M A_p^{-1} D_{dp} - G_{1p}^c \bar{A}_p^{-1} z^{-1} G_{1p}^f] \\ &\quad - \{D_{cp}^{*-1}(D_{cp}^* G_{1p}^c z^{-1} F_{1p}^f z^{g_2} + z^{g_1} F_{1p}^c z^{-1} D_{12} C_{dp}^*) D_{dp}^{*-1}\} \end{aligned} \tag{C.19}$$

The above expression is hence split into two parts where $T_d = T_d^+ - T_d^-$. The stable part is T_d^+ in square brackets and the unstable part, T_d^-, is in curly brackets. Earlier, M was defined as $M = (I + KW)^{-1}K$. Noting that $(I + KW)K = K(I + WK)$, it is also possible to state $M = K(I + WK)^{-1}$. Hence $M + MWK = K \Rightarrow MA_p^{-1}A_p + MA_p^{-1}B_pK = K$, leading to $MA_p^{-1} = K(A_p + B_pK)^{-1}$. The stable term can now be expressed as:

$$\begin{aligned}
T_d^+ &= [D_{cp}\bar{A}_{1p}^{-1}MA_p^{-1}D_{dp} - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^f] \\
&= D_{cp}\bar{A}_{1p}^{-1}K(A_p + B_pK)^{-1}D_{dp} \\
&\quad - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}(A_p + B_pK)(A_p + B_pK)^{-1}D_{dp} \\
&= (D_{cp}\bar{A}_{1p}^{-1}K - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}(A_p + B_pK))(A_p + B_pK)^{-1}D_{dp} \\
&= ((D_{cp}\bar{A}_{1p}^{-1} - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}B_p)K \\
&\quad - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}A_p)(A_p + B_pK)^{-1}D_{dp} \qquad \text{(C.20)}
\end{aligned}$$

Multiplying Diophantine equation (C.16) by $\bar{B}_{1p}$, and a further Diophantine equation:

$$z^{-g_1}D_{cp}^*H_{1p}^c - F_{1p}^c\bar{B}_p = (\bar{A}_{1p}^*\bar{R}_c + \bar{B}_{1p}^*\bar{G}_c)z^{-g_1} \qquad \text{(C.21)}$$

by $\bar{A}_{1p}$ and adding, obtain the control implied equation:

$$\begin{aligned}
D_{cp}^*G_{1p}^c\bar{B}_{1p} + D_{cp}^*H_{1p}^c\bar{A}_{1p} &= (\bar{B}_{1p}^*\bar{Q}_c + \bar{A}_{1p}^*\bar{G}_c^*)\bar{B}_{1p} + (\bar{A}_{1p}^*\bar{R}_c + \bar{B}_{1p}^*\bar{G}_c)\bar{A}_{1p} \\
\Rightarrow D_{cp}^*(G_{1p}^c\bar{B}_{1p} + H_{1p}^c\bar{A}_{1p}) &= \bar{B}_{1p}^*\bar{Q}_c\bar{B}_{1p} + \bar{A}_{1p}^*\bar{G}_c^*\bar{B}_{1p} + \bar{A}_{1p}^*\bar{R}_c\bar{A}_{1p} + \bar{B}_{1p}^*\bar{G}_c\bar{A}_{1p} \\
\Rightarrow G_{1p}^c\bar{B}_{1p} + H_{1p}^c\bar{A}_{1p} &= D_{cp} \qquad \text{(C.22)}
\end{aligned}$$

Substituting (C.22) into (C.20), the first two terms become:

$$\begin{aligned}
&D_{cp}\bar{A}_{1p}^{-1} - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}B_p \\
&= (G_{1p}^c\bar{B}_{1p}\bar{A}_{1p}^{-1} + H_{1p}^c - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}B_p) \qquad \text{(C.23)}
\end{aligned}$$

Noting $\bar{B}_{1p}\bar{A}_{1p}^{-1} = \bar{A}_p^{-1}\bar{B}_p$, $B_{1p}A_{1p}^{-1} = A_p^{-1}B_p$, $A_{1p} = \bar{A}_{1p}$ and $B_{1p} = C_{21}\bar{B}_{1p}$:

$$\begin{aligned}
&(G_{1p}^c\bar{B}_{1p}\bar{A}_{1p}^{-1} + H_{1p}^c - G_{1p}^c\bar{A}_p^{-1}z^{-1}G_{1p}^fD_{dp}^{-1}B_p) \\
&= G_{1p}^c\bar{A}_p^{-1}(\bar{B}_pA_{1p} - z^{-1}G_{1p}^fD_{dp}^{-1}A_pB_{1p})A_{1p}^{-1} + H_{1p}^c \\
&= G_{1p}^c\bar{A}_p^{-1}(\bar{A}_p - G_{1p}^fD_{dp}^{-1}A_pC_{21}z^{-1})\bar{B}_{1p}\bar{A}_{1p}^{-1} + H_{1p}^c \\
&= G_{1p}^c\bar{A}_p^{-1}(I - G_{1p}^fD_{dp}^{-1}A_pC_{21}z^{-1}\bar{A}_p^{-1})\bar{B}_p + H_{1p}^c \qquad \text{(C.24)}
\end{aligned}$$

Multiplying Diophantine equation (C.18) by $\bar{C}_{2p}$, where $C_{dp} = \bar{C}_{2p}D_{12}$, and a further Diophantine equation:

$$z^{-g_2}H^f_{1p}D^*_{dp} - C_{21}z^{-1}F^f_{1p} = R_{f1}A^*_p z^{-g_2} \tag{C.25}$$

by A_p and adding, obtain the filtering implied equation:

$$\begin{aligned} \bar{C}_{2p}G^f_{1p}D^*_{dp} + A_pH^f_{1p}D^*_{dp} &= \bar{C}_{2p}D_{12}C^*_{dp} + A_pR_{f1}A^*_p \\ \Rightarrow (\bar{C}_{2p}G^f_{1p} + A_pH^f_{1p})D^*_{dp} &= C_{dp}C^*_{dp} + A_{dp}R_{f1}A^*_{dp} \\ \Rightarrow \bar{C}_{2p}G^f_{1p} + A_pH^f_{1p} &= D_{dp} \end{aligned} \tag{C.26}$$

Using (C.26) and the fact that $\bar{C}_{2p} = A_pC_{21}z^{-1}\bar{A}_p^{-1}$, the bracketed term in (C.24) may be written as:

$$\begin{aligned} & (I - G^f_{1p}D^{-1}_{dp}A_pC_{21}z^{-1}\bar{A}_p^{-1}) \\ = \; & (I - G^f_{1p}(\bar{C}_{2p}G^f_{1p} + A_pH^f_{1p})^{-1}A_pC_{21}z^{-1}\bar{A}_p^{-1}) \\ = \; & (I - G^f_{1p}(\bar{C}_{2p}G^f_{1p} + A_pH^f_{1p})^{-1}\bar{C}_{2p}) \\ = \; & (I - G^f_{1p}(H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}G^f_{1p} + I)^{-1}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}) \end{aligned} \tag{C.27}$$

Noting that $(G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p} + I)G^f_{1p} = G^f_{1p}(H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}G^f_{1p} + I)$, hence:

$$\begin{aligned} & (I - G^f_{1p}D^{-1}_{dp}A_pC_{21}z^{-1}\bar{A}_p^{-1}) \\ = \; & (I - G^f_{1p}(H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}G^f_{1p} + I)^{-1}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}) \\ = \; & (I - (G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p} + I)^{-1}G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}) \\ = \; & (G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p} + I)^{-1}((G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p} + I) - G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p}) \\ = \; & (I + G^f_{1p}H^{f-1}_{1p}A_p^{-1}\bar{C}_{2p})^{-1} \end{aligned} \tag{C.28}$$

Substituting back into (C.24) obtain:

$$\begin{aligned} & D_{cp}\bar{A}_{1p}^{-1} - G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1}G_{1p}^{f}D_{dp}^{-1}B_{p} \\ = \; & G_{1p}^{c}\bar{A}_{p}^{-1}(I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}\bar{B}_{p} + H_{1p}^{c} \end{aligned} \tag{C.29}$$

From (C.26):

$$\begin{aligned} G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p}G_{1p}^{f} + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}A_{p}H_{1p}^{f} &= G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}D_{dp} \\ \Rightarrow (G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p} + I)G_{1p}^{f} &= G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}D_{dp} \\ \Rightarrow G_{1p}^{f}D_{dp}^{-1}A_{p} &= (I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}G_{1p}^{f}H_{1p}^{f-1} \end{aligned} \tag{C.30}$$

Substituting both (C.29) and (C.30) into (C.20), the final expression is:

$$\begin{aligned} T_{d}^{+} &= [D_{cp}\bar{A}_{1p}^{-1}MA_{p}^{-1}D_{dp} - G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1}G_{1p}^{f}] \\ &= ((D_{cp}\bar{A}_{1p}^{-1} - G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1}G_{1p}^{f}D_{dp}^{-1}B_{p})K \\ &\quad - G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1}G_{1p}^{f}D_{dp}^{-1}A_{p})(A_{p} + B_{p}K)^{-1}D_{dp} \\ &= ((G_{1p}^{c}\bar{A}_{p}^{-1}(I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}\bar{B}_{p} + H_{1p}^{c})K \\ &\quad - G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1}(I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}G_{1p}^{f}H_{1p}^{f-1})(A_{p} + B_{p}K)^{-1}D_{dp} \\ &= H_{1p}^{c}((I + H_{1p}^{c-1}G_{1p}^{c}\bar{A}_{p}^{-1}(I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}\bar{B}_{p})K - H_{1p}^{c-1}G_{1p}^{c}\bar{A}_{p}^{-1}z^{-1} \\ &\quad (I + G_{1p}^{f}H_{1p}^{f-1}A_{p}^{-1}\bar{C}_{2p})^{-1}G_{1p}^{f}H_{1p}^{f-1})(A_{p} + B_{p}K)^{-1}D_{dp} \end{aligned} \tag{C.31}$$

Recall that the integrand may be written as:

$$I_{c} = trace\{T_{d}T_{d}^{*}\} - I_{c0} = trace\{(T_{d}^{+} - T_{d}^{-})(T_{d}^{+} - T_{d}^{-})^{*}\} - I_{c0} \tag{C.32}$$

To minimise the complex integral of I_c, it must first be noted that the cross-terms integrate to zero, allowing (5.27) to be stated as:

$$J_{p} = \frac{1}{2\pi j} \oint\limits_{|z|=1} trace\{T_{d}^{+}T_{d}^{+*} + T_{d}^{-}T_{d}^{-*}\} - I_{c0}\frac{dz}{z} \tag{C.33}$$

The T_d^- and I_{c0} terms are independent of the controller choice, hence the optimal control problem reduces to minimising:

$$J_d^+ = \frac{1}{2\pi j} \oint\limits_{|z|=1} trace\{T_d^+ T_d^{+*}\} \frac{dz}{z} \tag{C.34}$$

which is achieved when $T_d^+ = 0$. The minimum cost is therefore:

$$J_{min} = \frac{1}{2\pi j} \oint\limits_{|z|=1} trace\{T_d^- T_d^{-*}\} - I_{c0} \frac{dz}{z} \tag{C.35}$$

The expression for the optimal controller is derived by setting (C.31) equal to zero. Therefore:

$$\begin{aligned} & (I + H_{1p}^{c-1} G_{1p}^c \bar{A}_p^{-1} (I + G_{1p}^f H_{1p}^{f-1} A_p^{-1} \bar{C}_{2p})^{-1} \bar{B}_p) K \\ = \; & H_{1p}^{c-1} G_{1p}^c \bar{A}_p^{-1} z^{-1} (I + G_{1p}^f H_{1p}^{f-1} A_p^{-1} \bar{C}_{2p})^{-1} G_{1p}^f H_{1p}^{f-1} \end{aligned} \tag{C.36}$$

Hence:

$$\begin{aligned} K \; = \; & (I + H_{1p}^{c-1} G_{1p}^c \bar{A}_p^{-1} (I + G_{1p}^f H_{1p}^{f-1} A_p^{-1} \bar{C}_{2p})^{-1} \bar{B}_p)^{-1} \\ & H_{1p}^{c-1} G_{1p}^c \bar{A}_p^{-1} z^{-1} (I + G_{1p}^f H_{1p}^{f-1} A_p^{-1} \bar{C}_{2p})^{-1} G_{1p}^f H_{1p}^{f-1} \end{aligned} \tag{C.37}$$

Defining $K_c = H_{1p}^{c-1} G_{1p}^c$, $K_{f1} = G_{1p}^f H_{1p}^{f-1}$, and noting that $\bar{C}_{2p} = A_p \bar{C}_{21} z^{-1} \bar{A}_p^{-1}$, and $B = z\bar{B}_p$:

$$\begin{aligned} K \; & = \; (I + K_c \bar{A}_p^{-1} (I + K_{f1} \bar{C}_{21} z^{-1} \bar{A}_p^{-1})^{-1} \bar{B}_p)^{-1} K_c \bar{A}_p^{-1} z^{-1} (I + K_{f1} \bar{C}_{21} z^{-1} \bar{A}_p^{-1})^{-1} K_{f1} \\ & = \; (I + K_c (z\bar{A}_p + K_{f1} C_{21})^{-1} B)^{-1} K_c (z\bar{A}_p + K_{f1} C_{21})^{-1} K_{f1} \end{aligned} \tag{C.38}$$

Note that $(I + K_c(z\bar{A}_p + K_{f1}C_{21})^{-1}B)K_c = K_c(I + (z\bar{A}_p + K_{f1}C_{21})^{-1}BK_c)$, hence:

$$K = K_c (I + (z\bar{A}_p + K_{f1} C_{21})^{-1} B K_c)^{-1} (z\bar{A}_p + K_{f1} C_{21})^{-1} K_{f1} \tag{C.39}$$

Further, note that $(I + (z\bar{A}_p + K_{f1}C_{21})^{-1}BK_c) = (z\bar{A}_p + K_{f1}C_{21})^{-1}((z\bar{A}_p + K_{f1}C_{21}) + BK_c)$, giving:

$$\begin{aligned} K &= K_c(z\bar{A}_p + K_{f1}C_{21} + BK_c)^{-1}(z\bar{A}_p + K_{f1}C_{21})(z\bar{A}_p + K_{f1}C_{21})^{-1}K_{f1} \\ &= K_c(z\bar{A}_p + K_{f1}C_{21} + BK_c)^{-1}K_{f1} \qquad \text{(C.40)} \end{aligned}$$

Appendix D

Adjoint Operator in Chapter 6

For time-invariant discrete-time systems, the output may be calculated from the convolution sum:

$$y(t) = \sum_{\tau=-\infty}^{t} w(t-\tau)u(\tau) = W(z^{-1})u(t) \tag{D.1}$$

where $y(t)$ is the system output, $w(t-\tau)$ is the system impulse response and $u(\tau)$ is the system input. For a time-varying system, the impulse response does not depend only upon the time elapsed from the input at τ to the present time, t. Hence, the convolution sum becomes:

$$y(t) = \sum_{\tau=-\infty}^{t} w(t,\tau)u(\tau) = (Wu)(t) = W(t,z^{-1})u(t) \tag{D.2}$$

Therefore, the inner product:

$$< x(t), (Wu)(t) >_T = \sum_{t=-T}^{T} x^T(t) \sum_{\tau=-T}^{t} w(t,\tau)u(\tau) \tag{D.3}$$

Interchanging the order of summation:

$$< x(t), (Wu)(t) >_T = \sum_{\tau=-T}^{T} [\sum_{t=\tau}^{T} w^T(t,\tau)x(t)]^T u(\tau) \tag{D.4}$$

then changing the variables of summation:

$$\begin{aligned} < x(t), (Wu)(t) >_T &= \sum_{t=-T}^{T} [\sum_{\tau=t}^{T} w^T(\tau, t) x(\tau)]^T u(t) \\ &= < (W * x)(t), u(t) >_T \end{aligned} \qquad \text{(D.5)}$$

In the limit as $T \rightarrow \infty$, $(W^* x)(t) = \sum_{\tau=t}^{\infty} w^T(\tau, t) x(\tau) = W^T(t, z) x(t)$, thus proving the result in equation (6.13).