DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

University of Strathclyde Engineering

NONLINEAR CONTROL: AN LPV NONLINEAR PREDICTIVE GENERALISED MINIMUM VARIANCE PERSPECTIVE

Petros Savvidis

Industrial Control Centre,

Department of Electronic and Electrical Engineering

University of Strathclyde,

204 George Street,

Glasgow G1 1XW

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Jan 2017

Declaration of Authenticity and Author's Rights

"This thesis is the result of the author's original research. It has been composed by the author and has not previously submitted for examination which has led to the award of a degree."

Petros Savvidis

Signed:

Date:

Abstract

This thesis describes new developments in nonlinear controllers for industrial applications. It first introduces the *Nonlinear Generalised Minimum Variance* (*NGMV*) control algorithm, for *Linear Parameter Varying systems* (*LPV*). This combines the benefits of the basic *NGMV* algorithm in dealing with nonlinearities, where a black box input model can be used, and adds an option to also approximate a nonlinear system with an *LPV* output subsystem. The models can therefore represent *LPV* systems and characteristics including saturation, discontinuities and time-varying dynamics.

The next major contribution is in the nonlinear predictive control algorithms proposed that are also using the *LPV* model structure. The simplest is the *Nonlinear Generalized Predictive Control* (*NGPC*) algorithm that relates to the best known model predictive control law for linear systems. The final predictive control solution is one that may be specialized to either the *NGMV* or *NGPC* cases and is therefore the most general. This is referred to as a *Nonlinear Predictive Generalized Minimum Variance Controller* (*NPGMV*). When the algorithms use only the *LPV* structure to approximate the nonlinear system the solutions are particularly simple in unconstrained and constrained versions, and are relatively light computationally for implementation.

Three representative industrial design examples have been chosen to validate the algorithms for different Bandwidth (*BW*) and nonlinear characteristics. All three examples were based on real application problems with company interest. In the first example (small *BW*) the basic state-space and *LPV* versions of the algorithm are used for the auto-manoeuvring and dynamic positioning of marine vessel. In this application the parameter variations were representative of wave disturbance changes with sea state, rather than due to approximating nonlinear behaviour. Actuator constraints were considered in the design.

In the second industrial example (medium *BW*) the *LPV-NPGMV* was implemented for controlling the blade pitch and generator torque of a 5MW offshore wind turbine. The main objective here was to maintain the power produced at the rated value which requires compensation against wind disturbances, so that wind speed is the varying parameter. The *LPV-NPGMV* controller produced here used a parameterised system model involving the wind speed so that the controller performance changed with wind conditions. Actuator constraints were included and statistical performance assessed.

The third example (fast *BW*) explores the stabilisation of a 2-axis gyroscopic electro-optical turret used in surveillance applications. This application was designed and employed on a real system. Because of the limitations imposed by *BW* requirements and the memory of the digital controller, only the basic state-space version of the algorithm was possible to implement. The main objective in this problem was to improve the tracking performance around the NADIR singularity point (a discontinuity) in the trajectory. In all three examples the *NGMV* controllers showed notable improvement in comparison to the baseline controllers without the need for scheduled gains or re-configuration when moving across different operating points.

Acknowledgements

I would like to express my gratitude and appreciation to my supervisors at Strathclyde University, Prof. Michael Grimble, for his continuous guidance and patience during all the years of research, and Dr. Reza Katebi for his invaluable advice at the most critical part of the thesis progression. I would also like to thank Dr. David Anderson at Glasgow University for sharing his industrial experience and making real systems experimentation possible, Dr. Pawel Majecki for his invaluable assistance with the implementation of the algorithms, Dr. Andy Clegg and all of my colleagues at Industrial Systems & Control Ltd. for their support and understanding. Furthermore I would like to thank Dr. H. Yue and Prof. R. Patton for their comments and corrections which greatly contributed to the final improved version of the Thesis.

A special thanks to my family, Xanthippi Savvidou and Panagiotis Spyridon for believing in me and supporting me all the way through thick and thin during all these years of hard work.

Contents

Nomenclature

ARMAX	Controlled Autoregressive Moving Average with Exogenous Input
CARMA	Controlled Autoregressive Moving Average
DAS	Defensive Aid Suite
DFIG	Doubly-Fed Induction Generator
DIRCM	Directed Infra-Red Countermeasures
DMC	Dynamic Matrix Control
DOF	Degrees of Freedom
EKF	Extended Kalman Filter
FoR	Field of Regard
FPGA	Field Programmable Gate Array
GMV	Generalised Minimum Variance
GPC	Generalised Predictive Control
HIL	Hardware in the Loop
IDCOM	Identification and Command
IMC	Internal Model Control
ISR	Intelligence Surveillance & Reconnaissance
KF	Kalman Filter
LMI	Linear Matrix Inequalities
LMPC	Linear Model Predictive Control
LoS	Line of Sight
LPV	Linear Parameter Varying
LQG	Linear Quadratic Gaussian
LTI	Linear Time Invariant
LV	LabVIEW
MIMO	Multiple-Input-Multiple-Output
MPC	Model Predictive Control
MV	Minimum Variance
NED	North-East-Down
NGMV	Nonlinear Generalised Minimum Variance
NGPC	Nonlinear Generalised Predictive Control
NMPC	Nonlinear Model Predictive Control
NREL	National Renewable Energy Laboratory
PXI	PCI eXtensions for Instrumentation
QDMC	Quadratic Dynamic Matrix Control
qLPV	quasi-Linear Parameter Varying
QP	Quadratic Programming
SCL	Sightline Control Laboratory
WECS	Wind Energy Conversion Systems

Notation

Subscripts

r	: Reference subsystem state-space matrix index
d	: Disturbance subsystem state-space matrix index
0	: Linear or *LPV* subsystem state-space matrix index
p	: Error weighting P_c subsystem state-space matrix index
m	: Measurements signal (*z*) state-space matrix index
t	: Time-varying subsystem state-space matrix index
φ	: Index of combined matrices used within a cost function e.g. E_φ=- E_p E_0

Matrix Notation

A, B, C (Section 2.1)	: Used as CARMA polynomial terms
A, B, C (Section 2.2)	: State, input and output state-space matrix respectively
D	: State disturbance matrix
E	: Input-output state-space matrix (traditional notation is *D*)
Φ (Kalman Filter)	: Impulse response operator representing stochastic disturbance (not to be confused with *Φ* in the cost function i.e. $J = E\{\Phi_0^T \Phi_0\}$)
Λ_N	: Control weighting matrix in *GPC* cost function
d	: Deterministic measurement noise signal used for generality

Chapter 1 Introduction

The most common approaches in classical control for dealing with nonlinearities in systems use linearization of the system around specific points of operation. Unfortunately, there are often limitations and possible unpredictable behaviour in attempting to control very difficult dynamics and nonlinearities in modern systems with classical methodologies. These might involve enhanced simple architectures built around the well-established *PID* controls, like a combination with nonlinear *Feed-Forward* (*FF*) compensation and the use of wind-up protection.

Adaptive *PID* algorithms can also be used to re-define tuning parameters to try to ensure the closed-loop system is stable (Grimble, 2001). These have been shown to be reasonably robust in dealing with relatively smooth nonlinearities. These were developed to replace tuning methods that relied solely on a linear model of the process to calculate the *PID* gains (e.g. Ziegler-Nichols, *IMC* tuning etc.). This can be achieved by various methods like online parameter identification (performing linearization at every iteration), to base tuning upon, and optimisation algorithms to evaluate the best-suited gains with respect to the current operating point (Slotine and Li, 1991).

A very popular technique, that also falls within this category and has been used extensively especially in the aerospace industry (where the dynamics of the aircraft vary dramatically with speed and altitude) is *Gain Scheduling*. This method can employ linear control techniques but requires effort to determine a sufficient set of control parameters to operate the system in a given range. Unfortunately, fast operating point variations can lead to severe nonlinear behaviour in which case guaranteeing stability becomes problematic (Shamma and Athans, 1990).

More advanced strategies that stem from the same approach (employing linearisation) can be seen in basic formulations of model-based control schemes like the *Smith-Predictor* and *Internal Model Control* (*IMC*), but also in optimal control strategies like the *Linear Quadratic Regulator* (*LQR*), *Linear Quadratic Gaussian (LQG*), *the Linear Model Predictive Control* (*MPC*) and of course the linear *Kalman Filter* (*KF*). The latter is required for estimating the system states in the *LQG* and *MPC* methods (Kouvaritakis et al., 1999). These can employ an estimated behaviour of the process for the next iteration, based on a linearised model at the operating point of interest, within the closed-loop solution (the solution of a cost-function in the case of optimal control schemes). Most *MPC* methodologies also rely on linear system models to generate the future predictions vector which are then needed in the optimal control problem solution.

Explicit control of nonlinear systems however has been one of the major areas of both academic and industrial interest over the past two decades (Slotine and Li, 1991) and developments from this area form the building blocks that influenced work on this thesis. Versions of the linear control strategies mentioned above with ad hock changes often fail to guarantee stability. It is intuitive that nonlinear systems which have physically motivated

natural solutions using techniques like feedback linearization should have better performance, be less costly and complex than linear systems involving say scheduling. A general illustration of this type of a direct control approach can be seen in the following figure.

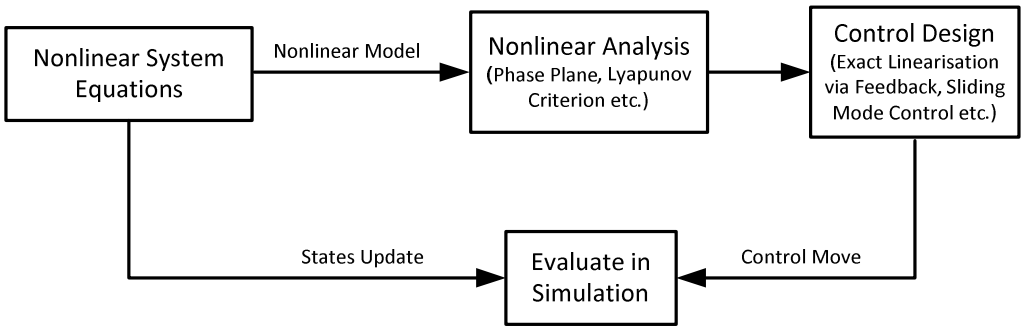


Figure 1:1 Direct nonlinear control methodologies stages diagram

Popular techniques within the direct nonlinear control framework include *Local Feedback Linearisation*, *Variable Structure Control* (*VSS*) and the *Sliding Mode Control* (*SMC*). The first is based on designing a local high gain feedback loop around the nonlinearity isolating it and effectively transforming it into linear for the outer overall control loop. Contrary to other direct nonlinear approaches the nonlinearity needs not be exactly known. Nevertheless, this method requires an additional measurement for the internal loop. A similar approach to this is *Feedback Linearisation* resembling in objective the indirect *Jacobian linearization* approach. Whereas the former considers transforming the system into localised linear approximations the latter aims the same but by exact state transformation and feedback. Both aim at transforming the system such that linear analysis methods can be employed. An immediate limitation springs; the requirement that these derivatives exist which will in turn allow for an input *u* to influence the states making control design possible. Moreover this method as well is susceptible to modelling errors, uncertainty and unmodelled dynamics all of which can greatly impact robustness (Slotine and Li, 1991).

VSS resembles state-dependent or *LPV* models in definition it is more intended however, to describe a design that responds to structural inherent discontinuities of the system hence resulting into a discontinuous feedback control law which involves switching logic to transition between different modes. The key problem in *VSS* control is to determine the switching points (switching or discontinuity surface) in state space. The objective then becomes to bring and keep the system at the switching surface (Utkin et al., 2009). Once the system is on the surface there is no trajectory away from it. Small shifts of the state from the surface causes a motion that brings the state back to the surface, a behaviour known as *sliding mode*. As an extension to *VSS*, *SMC* aims to drive system trajectory to a desired switching state within finite time. This scheme is advantageous in that although individual structures within the discontinuous framework may be unstable, the *VSS* system in whole is asymptotically stable. Moreover, it

results into flexible control implementations (can be combined with *PID* controllers) easy to restrict to certain regions of operation and it is shown to be relatively robust to parameter uncertainty. The key problem with *SMC* is that due to delays there will be no ideal sliding on the sliding surface. This results into chattering (fast switching) around the switching surface due to the discontinuous behaviour (Utkin and Lee, 2006).

The control strategy explored in this thesis provides a framework with very general system description but one that may employ a black box model of the plant. This can be practical for real life applications where access to system information is limited for various reasons. The idea here is to formulate a controller able to accommodate a wide class of nonlinear systems without having to undergo major modifications within its core architecture for different system characteristics. In order to be practical, this controller should be able to use a model of the process where this is available or consider the process using a "*black-box*" model when the physical structure is unknown.

1.1 Direct Nonlinear Control – An NGMV Perspective

The general nonlinear control techniques described in the previous section provide valuable analysis and design tools, however an area that is not well addressed is that of controller synthesis. Optimal control solutions provide a formal synthesis framework and this opens up a very wide range of different possible approaches. For the purposes of this work synthesis approaches which are reasonably easy to understand and implement in industrial applications are of most interest. One of the challenges was implementation of the algorithms, in a manner that allows a fixed range of control parameters to yield reasonable performance throughout a wide operating range without the need of re-tuning.

The theory for the family of optimal nonlinear controllers known as *Nonlinear Generalised Minimum Variance* (*NGMV*) controllers was partly built on together with an extension of the well-known generalized predictive controllers. The *LPV* approach was chosen as the modelling approach to improve the predictive version of the *NGMV* and *NPGMV* controllers, due to its flexibility in approximating a wide range of nonlinear systems.

Much of the previous work on the *NGMV* approach has focussed on state-dependent models but *LPV* models are subtly different, particularly, when trying to establish stability and robustness properties. The basic *LPV* model is of course representative of a linear system with major external parameter variations. However, it is often used to provide an approximation to a nonlinear system model and in this case the parameter variations may be due to states, inputs or outputs (providing a true nonlinear system description).

This thesis concentrates on a simple *LPV* approach for enhancing *NGMV* and *predictive control* solutions (*NGPC* and *NPGMV*), to provide simple controllers for a range of industrial applications. As this research evolved, the *LPV* formulation was described for the basic *NGMV* as well as for the *NPGMV*. The *NGMV* solution is a necessary precursor to *NPGMV* designs, and is not the main theme of the thesis. The main contribution is the exploration of *LPV* methods

and their use in predictive control and in particular the *NPGMV* controller which can be specialized to both of the other designs.

The predictive versions of the *NGMV* seem very suitable for industrial applications when used in conjunction with *LPV* modelling methods. This is mainly due to fact that the current *NPGMV* algorithm relies upon the linear part of the system to generate future error predictions. The latter imposes a restriction in performance when the overall plant cannot be decomposed into a nonlinear and a linear part. If however the plant can be modelled as an *LPV* system this approximation can be utilised to generate more accurate predictions and hence may yield better performance. Recent work on engine control, for example (Majecki et al, 2015) has demonstrated a value of *LPV* methods in torque and emissions control. The *LPV* methods have also been used successfully in aerospace and seem to have great potential in areas such as automotive and wind turbine controls (Do at al., 2013), (Biannic, 2013), (Balas, 2002), (Adegas and Stoustrup, 2012).

Roadmap: In Chapter 2 an introduction to the basic concepts that constitute the *NGMV* is used to provide a preliminary understanding on the structure of the algorithms that follow in detail. In Chapter 3 attention shifts towards *LPV* nonlinear systems approximation and the *Nonlinear Generalised Predictive Control* (*NGPC*) strategy both important proponents for the formulation of the more advanced *NPGMV*. Chapter 4 describes the combination of the above in the derivation of the *LPV-NPGMV* control law. These theoretical concepts are then implemented and explored in different industrial application in Chapters 5, 6 and 7. These applications and the methods considered in this thesis are briefly described in the following sections.

1.1.1 Methods – Generalised Minimum Variance Control

The initial concept was to develop simple controllers using a combination of optimal control theory based on an extension of *Minimum Variance* (*MV*) control (i.e. the *Generalised Minimum Variance* (*GMV*)), combined with the well know technique of *IMC* [56], influenced by structures like that of a strategy well-employed in the industry known as the *Smith Predictor*. *MV* control is an optimal control strategy developed by (Aström, 2012) where the objective is to minimise the variance of the output of a stochastic system *k*-steps ahead (*k* being a measurable system delay) given the information available up to the present time instant. The optimal cost-function here consists of the conditional expectation of the squared output of the system to obtain its variance. It is a model-based optimal scheme which uses a *Controlled Auto-Regressive Moving Average* (*CARMA*) model to derive its control law. To calculate the optimal control the *MV* concept uses this model to generate *k*-steps ahead predictions of the process with the use of the *Diophantine* equation (Wellstead and Zarrop, 1991).

The *MV* scheme naturally provides dead-time compensation and has been very successful in reducing variability of process output while considering measurable (or observable) disturbances within the optimal control solution, being applicable to *MIMO* systems and demonstrating small computational complexity. However, it is unsuitable for non-minimum phase systems, results in excessive control action and is shown to produce a sub-optimal

solution in presence of saturation constraints. The *GMV* design is an extension that includes the weighted control action into the original *MV* cost-function. This provides better results in terms of improving control action variability and can accommodate non-minimum phase systems. Although successfully used in the industry, it has been shown to exhibit poor stability characteristics on non-minimum phase systems (Grimble and Johnson, 1988).

The *NGMV* controller aims to minimise a very similar cost function to the linear *GMV*, the main difference being the model of the process used to generate these signals within the cost function now contains a general nonlinear operator. This is described in detail in (Grimble, 2004) where the assumption is made that the plant model can be decomposed into a set of delay terms, a very general nonlinear subsystem that has to be stable, and a linear subsystem that can be represented in linear polynomial matrix or state equation form and possibly include unstable modes. Moreover, the system output signal within the cost-function is here replaced by a penalty on the error (i.e. difference between the output and the reference). A set of weightings is then selected (error and control weightings) to obtain a stabilising control law. Unlike the error weighting the control weighting can be nonlinear adding to flexibility in nonlinear design. The *LPV* approach is explored here as well as in the *NPGMV* to allow applicability in a wider range of systems and to increase performance over the basic *NGMV* solution.

For equivalent linear system designs, stability is ensured when the combination of a control weighting function and an error weighted plant model is strictly minimum phase (Grimble, 2007). For nonlinear systems it is shown that a related operator equation is required to have a stable inverse. That is, to ensure closed-loop stability an assumption must be made that a certain nonlinear operator has a stable inverse. The cost-function weightings must be chosen to satisfy both performance and stability/robustness requirements.

The inherent optimal control aspect makes the *NGMV* framework a promising candidate for the purpose of this work. The reason for this is that optimal control approach provides a well-proven, intuitive to implement design, yielding often predictable behaviour even in the case of complex systems. Unlike most classical methods, where strong engineering expertise is required to employ, there are several formalised approaches within optimal control design and that makes it attractive for industrial applications. There is rich literature around this family of algorithms with successful results in many realistic applications, ranging from simple architectures like *Feedback-Feedforward* (Grimble, 2007) to more advanced H_∞ (Grimble, 2006), in combination with the internal process nonlinearities considered unknown.

1.1.2 Methods – Nonlinear Generalised Predictive Control and LPV Models

The *NGPC* builds upon a control philosophy successfully applied in the process industries, the *Model Predictive Control* (*MPC*), with proven improvements in performance and profitability. A typical predictive controller benefits by utilising future reference signal or set-point information within the minimisation of a multi-step cost-function. This is very advantageous when dealing with processes that contain long dead times, time-varying parameters and

multivariable interactions. Predictive controllers initially became popular through their application in processes with relatively slow dynamics, representative of process industries like the petrochemical. The evolution of digital controllers along with the development of more practical and less complex algorithmic versions, with respect to computational burden, allowed the application of predictive controllers on faster servo and hydraulic systems as well.

The basis for *NGPC* development is traced back in early work by (Clarke et al., 1987) on the linear *GPC* which was originally derived in a polynomial form but a state-space version, suitable for large linear systems, was obtained by (Ordys and Clarke, 1993) later on. As an extended formulation of the *NMPC* the *NGPC* utilises nonlinear models to generate predictions and also has the capability to solve its optimal cost-function considering nonlinear input, output and state constraints. The application of the *NGPC* typically requires a real-time open-loop optimal control problem solution and its predicted behaviour can be different than the closed-loop. The optimal cost-function used in this work is a modification of the basic *GPC* cost-function, to include a terminal state cost term and a terminal constraint set on the final state. The first can be selected as a positively invariant set of the system under feedback whereas the latter can be selected as a *Lyapunov* function related to the local controller. The advantage of this is that it guarantees closed-loop stability and asymptotic convergence under certain conditions. State estimates necessary for the generation of predictions are obtained via the use of the *Extended Kalman Filter* (*EKF*). The optimal quadratic cost-function, similar to the linear *GPC*, is solved within a specified prediction horizon and according to the *Receding Horizon* principle, where only the first control element from the optimal control moves vector is applied and the procedure repeats at the next iteration.

Model fidelity is a crucial aspect here as well as with most nonlinear model-based strategies. In this regard the *LPV* approach can often improve system approximation and allow a wider range of systems to be accommodated by the algorithm as mentioned above. Another benefit of the use of *LPV* systems is the reduction of computational complexity using off-line optimisation approaches. The latter happens when a series of approximations are precomputed across the operating range and the appropriate controller is chosen based upon the current value of the system state for a system which has a very large operating range.

1.1.3 Methods – Nonlinear Predictive Generalised Minimum Variance Control and LPV Models

The *NPGMV* development was based on the *NGPC* structure and shares some of the advantages of the *Generalised Predictive Control* (*GPC*) algorithms. It shares similar stability assumptions and uses the same subsystems decomposition with the basic *NGMV* controller. Being a predictive control approach it differs in that it utilises a multi-step cost-function that includes future tracking error and control signal weightings in a *GPC* type of problem construction (Grimble, 2005). These weightings can be used with different error and control horizons. As mentioned the *NGMV* and *NGPC* controllers are both special cases of *NPGMV* (e.g.

when the system is linear and the control cost term tends to zero the controller reverts to those for a *GPC* controller).

The adaptation of *NPGMV* to *LPV* systems has very similar plant description to the original *NPGMV,* however it aims to provide a model based fixed-structure controller for time-varying systems through the relevant representation of the nonlinear subsystem. The latter is here further decomposed into input and output nonlinear subsystems. The first black box term can contain hard nonlinearities (i.e. saturation constraints, discontinuities etc.) and the latter dynamic nonlinearities that can be modelled via the *LPV* approach. No structure needs to be assumed for the input nonlinear sub-system, this is however assumed to be open-loop stable. To provide further generality, the internal model of the process is augmented with an "integrator" state as a simple way to introduce integral action. The *NPGMV* is the most general solution but perhaps the *NGPC* is the best compromise between complexity and generality.

1.1.4 Nonlinear Industrial Applications

To implement and test the various algorithms developed in this thesis, three representative nonlinear industrial control applications were selected, imposing different challenges ranging from simple saturation constraints to discontinuities and varying model parameters and structures. An important aspect that was considered upon selecting these applications was the different dynamic responses they exhibit with respect to Bandwidth. A ship dynamic positioning problem was chosen, subject to slowly varying sea state, a power regulation problem for a large scale wind turbine, affected by wind speed variations and finally the stabilisation of an electro-optical device while tracking fast moving targets.

Ship Positioning: The dynamic ship positioning and tracking problem is a well-researched subject with mature industrial applications. However, in the main the nonlinearities in ship positioning systems are not accounted for very directly in the designs. It is therefore likely that performance improvements could be obtained if a true nonlinear control law were applied. The *LPV* modelling and control approach enable a better approximation to nonlinear system behaviour to be made and hence the control laws to be applied should give performance improvements. One question is whether this adds complexity which makes such a control solution impractical. In the following chapter it will be shown that the *NGMV* control law can be applied to subsystems and a reasonably practical solution obtained. Ship simulation results are presented to demonstrate behaviour and it is shown that the implementation is not too much more complicated than that for existing commercial *Kalman Filtering* based ship positioning systems. The basic state-space *NGMV* was successfully employed here and demonstrated consistent performance improvement without the need for retuning across different operating ranges. In the second part the *LPV-NGMV* utilised a time-varying model parameterised by sea state and demonstrated improved disturbance rejection over a standard *NGMV* architecture.

Wind Turbine Control: Regulating wind turbine power production while compensating for wind disturbances is also a popular area of research and a problem that naturally falls within

the *LPV* philosophy. The latter is true as the behaviour of a wind turbine system can be suitably described and parameterised by wind speed through the aerodynamic conversion relations. Challenges imposed by this problem configuration are saturation constraint nonlinearities in pitch angle control and varying system dynamics dependent upon wind speed variations. The *LPV-NPGMV* controller was successfully implemented for various disturbance scenarios (i.e. wind gusts and turbulence effects) as well as for power reference tracking (de-rating) in two alternative configurations; a *Single Input Single Output* (*SISO*) where only the pitch angle is used to control power output and a *Multiple Input Multiple Output* (*MIMO*) where both pitch angle and generator torque are used to control power output and generator speed respectively. Different prediction horizons where found appropriate for different types of disturbances and improvement in regulation ranging from 60-90% was demonstrated in various cases. Regarding the de-rating scenarios, future reference information played a vital role, showing that increasing the prediction window provides anticipatory action and therefore improved tracking of the reference. A *Quadratic Programing* (*QP*) algorithm was used to address constraints adding further benefits in terms of restricting control action compromising slightly however disturbance rejection and tracking performance.

Gyroscopic Turret: The final application was part of a side project considering simple and effective algorithms for the stabilisation of an *Electro-Optical* (*EO*) gyroscopic turret used in surveillance systems. This control problem is characterised by very fast dynamics, trajectory singularities and actuator constraints. Moreover, this was a representative case where access to internal system information was not available and therefore suitable for the "*black-box*" internal model structure of the *NGMV*. In this application the servo commands of the 2-axis gyro are used to control positioning of the sightline of the device upon the moving target (i.e. incoming threat to the host platform) in an incredibly agile manner, to ensure proper tracking, interrogation and neutralisation. The precision demanded in these types of applications is at a level that even, otherwise minor effects due to friction in the gears of the servomechanism, can humper tracking performance significantly ultimately missing the target. The main problem however arises when a certain part of the target trajectory gives rise to a singularity in the kinematics transformation known as "*gimbal lock*". Experimental work was carried out at the Sightline Control Laboratory (*SCL*) of the University of Glasgow which is a bespoke research and teaching facility designed to assist applied research in the areas of pointing, stabilization, tracking and image processing of electro-optic systems, known collectively by the term *Sightline Control*. This provided the opportunity of *Hardware-in-the-Loop* (*HIL*) functional testing and although only the basic state-space *NGMV* algorithm was implemented due to hardware restrictions, it remained an invaluable part of this thesis for being able to tackle this very difficult nonlinearity (discontinuity) and demonstrate tracking improvement. The simple structure for implementation and the "*black-box*" configuration of the algorithm made it an excellent candidate for such a problem that requires minimal algorithmic complexity and a robust design that is able to cope with the entire range of operation without further scheduling.

1.2 Research Goals & Objectives

The main objectives as they evolved throughout this work can be summarised as follows.

- Develop the *Nonlinear Generalised Minimum Variance* (*NGMV*) algorithmic family, aiming to expand the class of nonlinear systems these algorithms can be applied to. The basic state space and predictive versions of the *NGMV* are explored from a *Linear Parameter Varying* (*LPV)* systems formulation standpoint.

- Derive and describe the *LPV* versions for the above algorithms in a manner easy to implement for different system architectures. Program and test the applicability and performance of the algorithms in simulation.

- Explore and implement the use of the *Extended Kalman Filter* (*EKF*) within the *Nonlinear Predictive NGMV* (*NPGMV*) to improve accuracy of state estimation within the controller. Enforce this by exploring the use of a modified Jacobian linearization implementation to improve fidelity of the *LPV* nonlinear system approximation.

1.3 Thesis Contributions

The contributions brought forth out of this work can be outlined as follows.

- Important structural modifications were explored regarding the improvement of the implementation of the basic NGMV algorithm including a *black-box + LPV* structure successfully implemented in simulation.

- A *"black-box" LPV* structure for the basic state space *NGMV* was introduced and successfully implemented in simulation.

- The *LPV* adaptation for the *NPGMV* algorithm was derived mathematically and implemented in simulation.

- The *Modified Jacobian* model derivation was explored and implemented within the *LPV-NPGMV* algorithm along with the integration and use of the *Extended Kalman Filter* (*EKF*) in the algorithm.

- Three, novel *NGMV* and *LPV-NGMV/NPGMV,* application examples were introduced, one of which involved the first real-time NI-Labview based deployment of the algorithm on a real system (cooperation with Thales, Edinburgh).

- Proposal to improve robustness and to add adaptive features with theory described in Future Work.

1.4 Thesis Organisation

Chapter 1 – Provides an introduction survey on general nonlinear systems control methodologies and identifies the place for this work within this framework.

Chapter 2 – Provides an introduction to the theoretical development of the basic *NGMV* formulation (within the state-space framework) and explores a few of the main design aspects of the algorithm. Moreover, provides discussions and ideas to enhance robustness and applicability of the basic algorithm version to systems with uncertainty.

Chapter 3 – Provides an introduction to nonlinear *MPC* (specifically the *NGPC* algorithm) adapted for *LPV* systems. This sets the basis upon which the *LPV-NPGMV* algorithm is derived in the following chapter.

Chapter 4 – Describes the derivation of the *NPGMV* algorithm and its adaptation to *LPV* systems. Moreover, it provides ideas to improve its implementation regarding computational aspects like algorithmic execution speed. At the end of the chapter and continuing from Chapter 2 the derivation of *LPV-NGMV* is described.

Chapter 5 – Utilises a marine vessel auto-piloting and dynamic positioning application to explore performance of the basic state-space *NGMV* design against a baseline controller across various scenarios. In the second part of the chapter the same application is used in different scenarios to assess the performance of the *LPV-NGMV* controller.

Chapter 6 – Implements and assesses quantitative performance of the modified *LPV* Predictive version of the algorithm on a large scale wind turbine control application example, against two baseline controllers for various scenarios.

Chapter 7 – Utilises a real-time electro-optic gyroscopic device platform stabilisation application (benefiting from real experimental results) to explore performance of the basic state-space *NGMV* design against a baseline controller across various scenarios.

Appendix A – Provides the *Matlab* & *SIMULINK* code which was developed for the application case studies in Chapter 5, 6 and 7.

The diagram in Figure 1:2 explains the conceptual flow of the thesis. A core challenge, and at the same time benefit of this work, is that it is the result of multiple applications of different nature. That imposed the main difficulty when drawing a common thread to integrate theoretical and practical concepts together. What comprises of this common thread expands upon the *NGMV* method being a combination of the direct nonlinear control philosophy of internal model control and the *MV* optimal control framework.

Developments in the thesis revolve around two core themes, that of nonlinear (predictive) control and of *NGMV* theory. The linking element between the two is the *LPV* nonlinear systems approximation methodology. After the basic foundation on general nonlinear control methods and the *NGMV* is laid in Chapters 1 and 2 respectively focus turns into setting the fundamental understanding for *LPV* systems and the *Nonlinear Model Predictive Control* paradigm (in this case the *NGPC*) for *LPV* systems in Chapter 3.

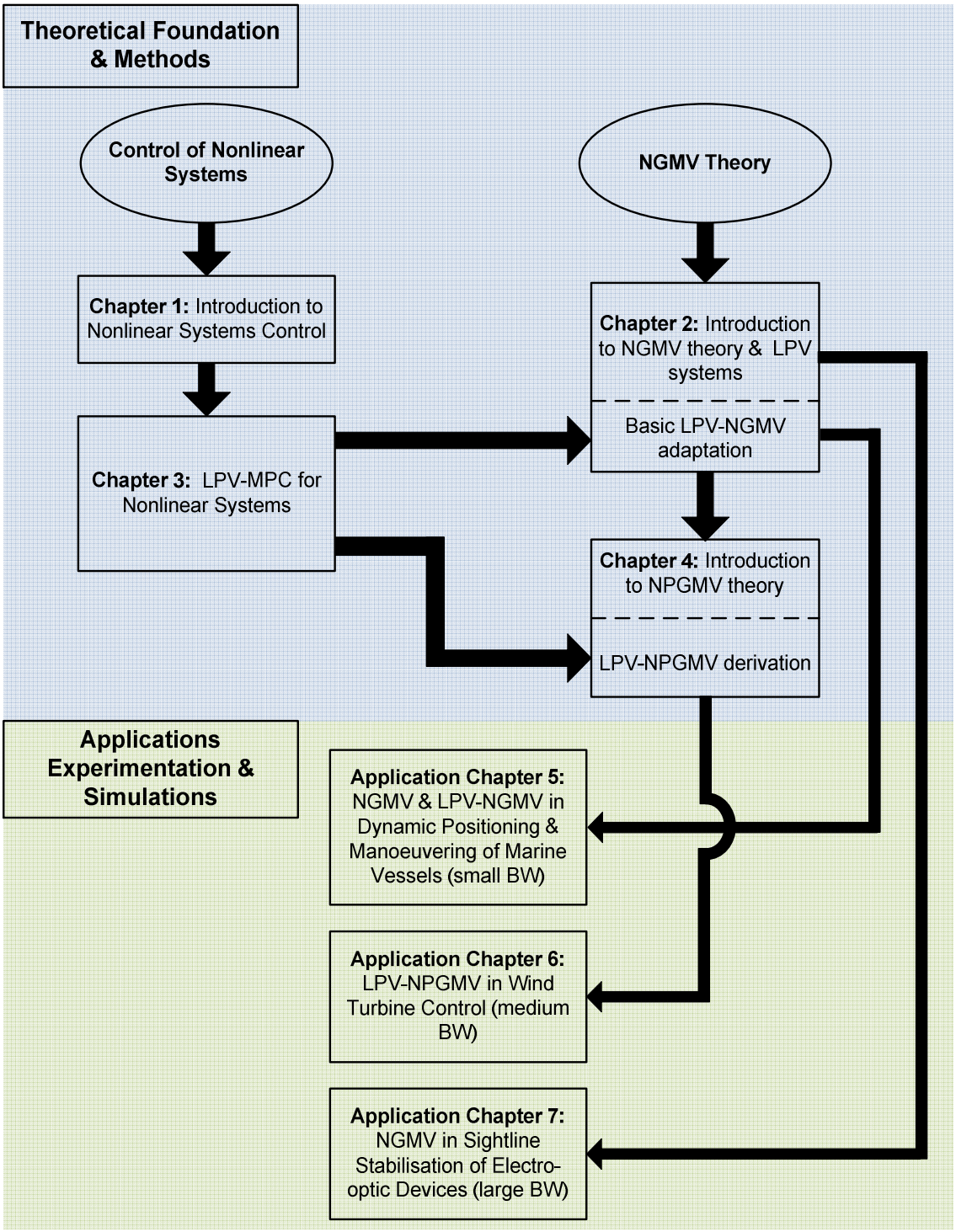


Figure 1:2 Thesis conceptual flow

These two concepts constitute the basis of the main development and contribution of this work which is the *LPV* formulation of the *Nonlinear Predictive Generalised Minimum Variance* (*LPV-NPGMV*) controller which is described in Chapter 4.

A secondary development, which was added at a later stage, was that of the *LPV* algorithm implementation for the basic state-space *NGMV* (*LPV-NGMV*). The main motivation behind this addition was that derivation of the algorithm resulted naturally from the *LPV-NPGMV* formulation with very minimal changes to the core structure of the controller. This is the main reason behind the derivation of the algorithm located at the end of Chapter 4 whereas only a brief mention is found at the end of Chapter 2. In combination with one of the classic *NGMV* formulations, that of the "*black-box*" proved to add flexibility in nonlinear control design for systems subjected to varying factors like disturbances. This addition also improved conceptual uniformity of the thesis.

The applications Chapters 5, 6 and 7 that follow reflect upon the *NGMV/LPV-NGMV* (Chapter 5), *NGMV/LPV-NPGMV* (Chapter 6) and *NGMV* (Chapter 7). As mentioned previously the differentiating factor between these application case studies was the dynamic response speed of the process, starting from slow, medium and fast in Chapters 5, 6 and 7 respectively. The main reason for not implementing a more advanced scheme in application Chapter 7 was restrictions imposed by the experimentation equipment. However, a valuable contribution remains being the first implementation of the *NGMV* controller on a real dynamically demanding nonlinear system.

1.5 List of Publications

1. P. Savvidis, J. Wang, R. Katebi, M. Grimble, 2009, *NGMVC Scheme on marine surface vessels dynamic positioning and manoeuvring*, Fourteenth International Ship Control Systems Symposium, Ottawa, Canada.

2. P. Savvidis, D. Anderson, M. Grimble, 2010, *Application of Nonlinear Generalised Minimum Variance to the nadir problem in 2-axis gimbal pointing & stabilisation*, SPIE Conference, Orlando, US.

3. P. Savvidis, M. Grimble, P. Majecki and Y. Pang, 2016, *Nonlinear Predictive Generalized Minimum Variance LPV Control of Wind Turbines*, IET International Conference on Renewable Power Generation (RPG), London, UK.

4. P. Savvidis, M. Grimble, P. Majecki, 2016, *Nonlinear Generalized Minimum Variance LPV Control for Dynamic Positioning of Marine Vessels*, Maritime International Conference on Safety and Operations, Glasgow, UK.

Chapter 2 Basic Structures

This Chapter briefly outlines the main principles and developments around *Minimum Variance* control which is the foundation upon which the class of *NGMV* controllers have being developed. Starting with the *Minimum Variance* (*MV*) paradigm the evolution via the *GMV* and finally its nonlinear derivative the *NGMV* is described in a fashion that naturally leads to the more advanced predictive methods described in the chapters that follow. A key point to observe is that the internal model architecture as shown in this chapter is preserved throughout the algorithms. The chapter consists of the following sections:

Section 1 – An Introduction to *MV and GMV* control strategies; an overview of the main principles and aspects.

Section 2 – An Introduction to *NGMV*; an overview of the main principles, stability assumptions, internal model structure and weighting selection criteria.

Section 3 – An Introduction to *Linear Parameter Varying Systems*; an overview of the type of *LPV* systems that will be used within the *LPV-NPGMV* formulation.

2.1 Minimum Variance Control

MV control as introduced by (Aström, 2012) also falls within the greater category of optimal control and its main objective, as the name suggest, is to minimise the variance of the output of a stochastic system at $t + k$ given the information available up to time t. In other words at each iteration an optimal control move is produced relative to the minimisation of the following cost function,

$$J = E\{y^2(t+k)\} \tag{2:1}$$

where y is the system output and k is a known time delay. *E* is the conditional expectation on data up to the current instance and the expectation of the squared variable yields its variance. The *MV* concept is a model-based optimal scheme and uses a *CARMA* (*Controlled Auto-Regressive Moving Average*) model to derive its control law.

$$Ay(t) = z^{-k}Bu(t) + C\xi(t) \tag{2:2}$$

Integrating the delay term in the equation yields,

$$y(t+k) = \frac{B}{A}u(t) + \frac{C}{A}\xi(t+k) \tag{2:3}$$

A, B and *C* express polynomials as a function of the inverse *z* operator. The stochastic element appears at the output of the system through *ξ* which represents a random disturbance signal with $E\{\xi(t)\} = 0$ and $E\{\xi(t)^2\} = \sigma^2$. To calculate the optimal control the *MV* concept uses this model to generate *k*-steps ahead predictions of the process. This is possible by further expanding the *C* polynomial with $C = AF + z^{-k}G$, where *F* and *G* are also polynomial terms,

and using the *Diophantine* equation to put generate the *k*-steps ahead prediction relation with respect to future and present output, control and error (through ξ) terms as detailed in (Wellstead and Zarrop, 1991). The optimal prediction is then derived by setting the future error term that cannot be regulated through control to zero as follows.

$$\hat{y}(t+k) = \frac{BF}{C}u(t) + \frac{G}{C}y(t) \tag{2:4}$$

Substituting Equation 2:4 into 2:1 and solving for $u(t)$ that minimises the output variance *k*-steps ahead yields the following *MV* control law,

$$u(t) = -\frac{G}{BF}y(t) \tag{2:5}$$

The *MV* owes its effectiveness in industrial application to a set of useful properties. It is easily seen that the *MV* scheme naturally provides dead-time compensation. Interestingly, in the absence of process noise *MV* turns into a dead-beat controller. It has been proved very successful in reducing variability of process output by also considering measurable (or observable) disturbances within the optimal control solution. Moreover, it is computationally efficient as it does not require a-priori information. It can be extended to *MIMO* systems and accommodate unknown system parameters. There are known limitations however that acted as precursors for the evolution of the algorithm.

The inherent zero-pole cancellation within the control derivation makes the *MV* concept unsuitable for non-minimum phase systems. The optimal solution often results into excessive control action as penalty on control is not included within the *MV* cost function. Moreover the algorithm produces a sub-optimal solution in presence of saturation constraints. It is not suitable for nonlinear systems in general without significant extensions. The next milestone in this family of controllers with the intent to tackle these problems was the Generalised Minimum Variance (*GMV*).

2.1.1 Generalised Minimum Variance Control

The *GMV* is a simple extension to the *MV* controller that aims to minimise the following generalised output $\varphi_0(t)$,

$$\begin{gathered}\varphi_0(t) = P_c e(t) + F_c u(t) \\ J = E\{\varphi_0^2\}\end{gathered} \tag{2:6}$$

The *GMV* cost function includes weighting factors on both the error (P_c) and the control action (F_c). Generation of this signal is illustrated in the following diagram.

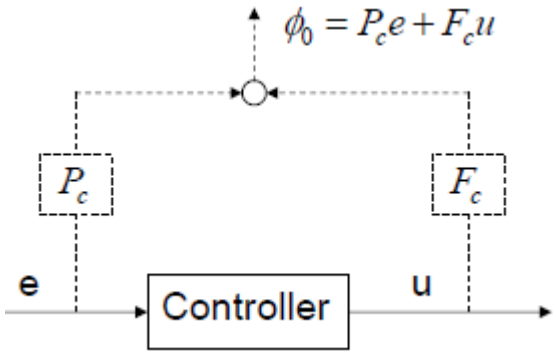


Figure 2:1 *GMV* cost-function signals.

These weighting factors can be defined as dynamic state-space or polynomial transfers e.g.

$$P_c = \frac{P_{cn}}{P_{cd}}, F_c = z^{-k}\frac{F_{cn}}{F_{cd}}$$

where the subscripts *cn* and *cd* denote numerator and denominator respectively. The state-space form of the weightings and the *GMV* derivation in general will become more comprehensible through the *NGMV* description in the following section. Note that the control output only affects the plant *k*-steps ahead. The *GMV* problem can be recast as an *MV* problem for the generalised plant,

$$\begin{aligned}\varphi_0(t) &= P_c\left(-z^{-k}W_k u(t) + Y_f\xi(t)\right) + F_c u(t)\\ &= z^{-k}(F_{ck} - P_c W_k)u(t) + P_c Y_f \xi(t)\end{aligned} \tag{2:7}$$

where W_k is the linear plance, Y_f is a spectral factor used within the Diophantine solution and $\xi(t)$ a stochastic disturbance signal as used in the *MV* formulation. Therefore the *MV* control of the generalised plant will be,

$$\begin{aligned}\varphi_0(t) &= (P_c W_k - F_c)u(t) + P_c Y_f \xi(t)\\ &= \underbrace{F\xi(t)}_{} + \underbrace{(P_c W_k - F_c)u(t-k) + R\xi(t-k)}_{}\end{aligned} \tag{2:8}$$

Statistically independent terms

where *R* and *F* are polynomial terms in the Diophantine equation $P_c Y_f = F + z^{-k}R$. The optimal *GMV* control then becomes,

$$u(t) = -\frac{R}{(P_c W_k - F_c)F}\varphi_0(t) \tag{2:9}$$

The *GMV* algorithm has been successfully employed in industrial applications and especially within self-tuning schemes. It has been shown however to exhibit poor stability characteristics on non-minimum phase systems (Grimble, 1988). The equivalence with its nonlinear formulation will immediately become apparent in the following section.

2.2 Nonlinear Generalised Minimum Variance Control

The adaptation of *GMV* algorithm to nonlinear state-space systems, introduced in (Grimble, 1981 and 2005), is used as a basis for all development across this thesis. In this section the main concept and structure of the *NGMV* is explored but from an *LPV* systems point of view, which will set the foundation for understanding the subsequent *LPV-NPGMV* controller derivation. For this formulation the *GMV* basic structure is retained. The main difference resides on the formulation of the model. In the traditional *NGMV* algorithm the oftentimes nonlinear plant is divided in a Wiener Voltera approach (Schetzen, 1980) into a linear and a nonlinear subsystem.

The plant model may encapsulate a more general black-box type system with unknown nonlinearities but more conveniently input nonlinearities like actuator constraints. Hard nonlinearities like discontinuities however are much more difficult to be dealt with. The latter can be expressed into state-space formulation as seen in the following equations,

$$\begin{aligned} x(t+1) &= A_0 x(t) + B_0 F(x,u)(t) \\ y(t) &= C_0 x(t) \end{aligned} \tag{2:10}$$

x denotes the state of the system, *u* the input (or control loop in the closed loop case) and *y* the output of the system. A_0, B_0 and C_0 are the state-space state, input and output matrices respectively. *F* is a nonlinear function with respect to the system states and inputs. Similar to the advantages state-space formulation holds for other model based schemes, here as well the original plant model can be augmented to include reference and disturbance subsystems but also static or dynamic weightings used for control as described in the following sections. The nature of these subsystems can vary from linear to nonlinear with small loss of generality as explained in (Grimble and Majecki, 2005). For the aims of this work *F* will be used to encapsulate *LPV* systems in Chapter 4.

The basic *NGMV* structure in its most generic form is captured in Figure 2:1. To increase generality disturbance terms can be further divided into deterministic and stochastic components. The latter can be realised with a linear filter driven by white noise (ξ_d).

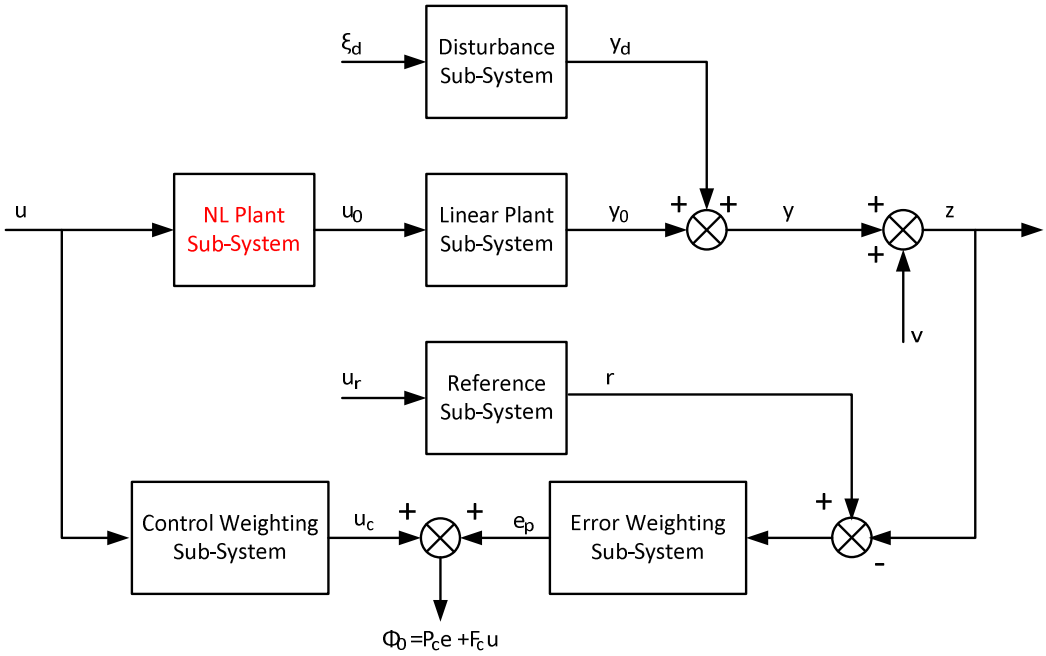


Figure 2:2 *NGMV* controller subsystems layout.

The subsystems that appear in Figure 2:2 contain their equivalent mathematical representations which can vary depending on the problem i.e. can be transfer functions, state-space models etc. For the purpose of this work state-space models are used. Apart from the *NL Plant* subsystem they are assumed to be linear. The former can be a very general nonlinear operator representing anything from input actuator nonlinearities, to *black-box* or as seen in the next chapter *LPV* systems. The full list of options this feature allows is best discussed later in the thesis when all the algorithms and various formulations have been described (Section 4.6). The mathematical representations of these subsystems are described in the following sections.

Control Signals:

u(t) : Vector of control signals applied to the nonlinear subsystem.
$u_0(t)$: Vector of control inputs to the *LPV* subsystem.

Output Signals:

y(t) : Vector of plant output signals.
z(t) : Vector of output measurement signals.

The measurements or observations signal results by adding measurement noise *v(t)* to the measured outputs of the plant; *z(t) = y(t) + v(t).*

Error Signal:

e(t) : Vector of tracking error signals.
This is the signal also used within the *NPGMV* controller cost-function and is the difference between the reference input and the measurements of the plant outputs to be controlled; *e(t)= r(t) – z(t).*

The subsystems in Figure 2:2 are explained in the section below.

Nonlinear Input Sub-System:

This sub-system is described by the following notation,

$$(W_1 u)(t) = z^{-k}(W_{1k} u)(t) \tag{2:11}$$

where z^{-k} is a diagonal matrix that contains all common delay elements in signal paths, assuming these can be extracted out of the system, *u* is the control signal driving the system and W_{1k} is the nonlinear input subsystem. Note that whereas in systems where these delay elements are significantly larger than the time constant of the plant dynamics (e.g. in the process industry) and can be easily allow the separation suggested in Equation 2:11. In more complex processes with strong internal couplings between various channels and with delay terms comparable to fast dynamics (e.g. in the automotive industry) this may not be possible.

The output of W_{1k} is denoted as,

$$u_0(t) = (W_{1k} u)(t) \tag{2:12}$$

where W_{1k} is assumed to be finite gain stable. Note that *k* signifies the explicit delay elements that have been extracted from the full nonlinear plant system.

Nonlinear Output Sub-System:

This sub-system is also nonlinear and is denoted as,

$$(W_0 u_0)(t) = \left(W_{0k} z^{-k} u_0\right)(t) \tag{2:13}$$

where W_{0k} is its delay-free notation (Grimble, 2005).

The state-space representations for all subsystems shown in Figure 2:2 are shown in the following sections as described in (Grimble, 2007). In the following equations the *r*, *d* and *0* indexes denote state-space matrices of the reference, disturbance and linear subsystems respectively.

Reference

$$x_r(t+1) = A_r x_r(t) + B_r u_r(t)$$
$$r(t) = y_r(t) = C_r x_r(t) \tag{2:14}$$

Disturbance

$$x_d(t+1) = A_d x_d(t) + B_d \xi_d(t)$$
$$d(t) = y_d(t) = C_d x_d(t) \tag{2:15}$$

Linear Plant Subsystem

$$x_0(t+1) = A_0 x_0(t) + z^{-k} B_0 u_0(t) + D_0 \xi_0(t)$$
$$y_0(t) = C_0 x_0(t) + z^{-k} E_0 u_0(t) \tag{2:16}$$

Cost-Function Error Weighting

$$x_p(t+1) = A_p x_p(t) + B_p e(t)$$
$$y_p(t) = C_p x_p(t) + E_p e(t) \tag{2:17}$$
$$e(t) = r(t) - d(t) - y_0(t)$$

The index *p* denotes these signals are weighted by the Error Weighting term P_c. Substituting *r, d* and y_0 from Equations 2:14 - 2:16 and expanding yields the following equations showing how all subsystems propagate to the overall system weighted output.

$$x_p(t+1) = A_p x_p(t) + B_p \left(C_r x_r(t) - C_d x_d(t) - C_0 x_0(t) - z^{-k} E_0 u_0(t) \right)$$
$$y_p(t) = C_p x_p(t) + E_p \left(C_r x_r(t) - C_d x_d(t) - C_0 x_0(t) - z^{-k} E_0 u_0(t) \right) \tag{2:18}$$

The above can be integrated in a unified augmented state-space system as shown below. This approach is always effective especially when programming model-based control algorithms in code. The augmented system matrices structure can be seen in detail in Chapter 4 in their *LPV* form.

Augmented State-Space System Model

$$x(t+1) = Ax(t) + Bu_0(t-k) + D\xi(t)$$
$$y(t) = Cx(t) + Eu_0(t-k) \tag{2:19}$$

where $z(t) = y(t) + v(t)$ are the measurements and $y_p(t) = C_\varphi x(t) + E_\varphi u_0(t-k)$ the output signal used within the cost function, with E_φ and C_φ being the combined E and C terms as described in (Grimble, 2007) ($E_\varphi = -E_p E_0$ and $C_\varphi = -C_p C_0$).

As with most model-based control schemes formulated in state-space, states that cannot be measured are obtained via state estimators. For the *NGMV* formulation the *Kalman Filter* and *Extended Kalman Filter* (for the *LPV* formulation) is used for this purpose. The former is put into *Predictor-Corrector* form as introduced in (Grimble and Johnson, 1988) as shown below.

Kalman Filter in Predictor-Corrector Form

$$\hat{x}(t+1|t) = A\hat{x}(t|t) + z^{-k}Bu_0(t) \text{ (Predictor)}$$
$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_f(e_0(t+1) - \hat{e}_0(t+1|t)) \text{ (Corrector)} \quad (2:20)$$
$$\text{where, } \hat{e}_0(t+1|t) = C_e\hat{x}(t+1|t) - z^{-k}E_0u_0(t+1)$$

Here *A*, *B*, and *C* matrices refer to the combined total linear state-space subsystem. *x* and *e* hat denote the state and error estimates respectively and the *e* index denotes the error subsystems expressed as a linear state-space model as well. K_f is the *Kalman* filter correction gain. Substituting from Equation 2:17 and expanding the *Kalman Filter* prediction equation is formulated as follows,

$$\hat{x}(t+1|t+1) = A\hat{x}(t|t) + z^{-k}Bu_0(t) + K_f(e_0(t+1) - (C_e\hat{x}(t+1|t) - z^{-k}E_0u_0(t+1)))$$

$$\begin{aligned}\left(zI - A + K_fC_eA\right)\hat{x}(t|t) \\ = z^{-k}Bu_0(t) + K_f(e_0(t+1) - (C_ez^{-k}Bu_0(t) - z^{-k}E_0u_0(t+1)))\end{aligned} \quad (2:21)$$

$$\hat{x}(t|t) = \left(I - z^{-1}\left(A - K_fC_eA\right)\right)^{-1}\left(K_f\left(e_0(t) + E_0u_0(t-k) - C_ez^{-k}Bu_0(t-k-1)\right) + Bu_0(t-k-1)\right)$$

Using the *Kalman Filter* gain equations (Mayne et al., 2000) the estimated states of the augmented system can be formulated as follows,

$$\hat{x}(t|t) = T_{f1}(z^{-1})e_0(t) + T_{f2}(z^{-1})u_0(t)$$

where the transfer-operators: $T_{f1}(z^{-1}) = \left(I - z^{-1}\left(A - K_fC_eA\right)\right)^{-1}K_f$ (2:22)

and $T_{f2}(z^{-1}) = \left(I - z^{-1}\left(A - K_fC_eA\right)\right)^{-1}z^{-k}\left(K_fE_0 + z^{-1}\left(I - K_fC_e\right)B\right)$

This formulation allows for the separation of the state estimates due to the stochastic input signals from the contribution of the control input as follows,

$$\begin{aligned}\hat{x}(t+k|t) &= A^k\hat{x}_{dist}(t|t) + \Phi(z^{-1})Bu_0(t)\\ &= A^k\big(\hat{x}(t|t) - \Phi(z^{-1})Bu_0(t-k)\big) + \Phi(z^{-1})Bu_0(t)\\ &= A^k\hat{x}(t|t) + \big(I - A^kz^{-1}\big)\Phi(z^{-1})Bu_0(t)\end{aligned} \quad (2:23)$$

Φ here denotes an impulse response operator representing the stochastic disturbance component. The estimates for *k*-steps ahead can then be obtained as follows,

$$\begin{aligned}\hat{x}(t+k|t) = A^k\hat{x}(t|t) + A^{k-1}Bu_0(t-k) + A^{k-2}Bu_0(t-k+1)+\ldots+ABu_0(t-2)\\ + Bu_0(t-1) = A^k\hat{x}(t|t) + T_0(k,z^{-1})Bu_0(t)\end{aligned} \quad (2:24)$$

where $T_0(k,z^{-1})$ is a transfer operator with the following impulse response,

$$T_0(k,z^{-1}) = \big(I - A^kz^{-k}\big)\Phi(z^{-1}) = z^{-1}\big(I + z^{-1}A + z^{-2}A^2+\ldots+z^{-k+1}A^{k-1}\big)$$

2.2.1 NGMV Optimal Control Law Derivation

Revisiting the *GMV* cost-function, its nonlinear version can be defined as follows,

$$J = E\{\Phi_0^2(t)\} = E\big\{\Phi_0^T(t)\Phi_0\ (t)\big\} = E\left\{trace\big\{\Phi_0\ (t)\Phi_0^T(t)\big\}\right\} \quad (2:25)$$

where,

$$\Phi_0\ (t) = P_ce(t) + (F_cu)(t) \quad (2:26)$$

In the *NGMV* case the control weighting can be, but not restricted to, a nonlinear dynamic control operator. It is important to remind here that system Equations 2:14 and 2:18 come into the cost function through the estimated error signal $e(t)$, and are only used in the derivation of the controller.

Figure 2:3 provides a layout of the actual controller location in the control loop relative to the total plant and various subsystems. W_r, W_d, W_{0k}, and W_{1k} represent the reference, disturbance, linear plant and nonlinear plant subsystems respectively. The dotted lines signify that these signals are not real but only used in the theoretical derivation of the control law. As this depicts a general overview of the real implementation it also includes the total rather than the decomposed plant operators (Grimble 2005).

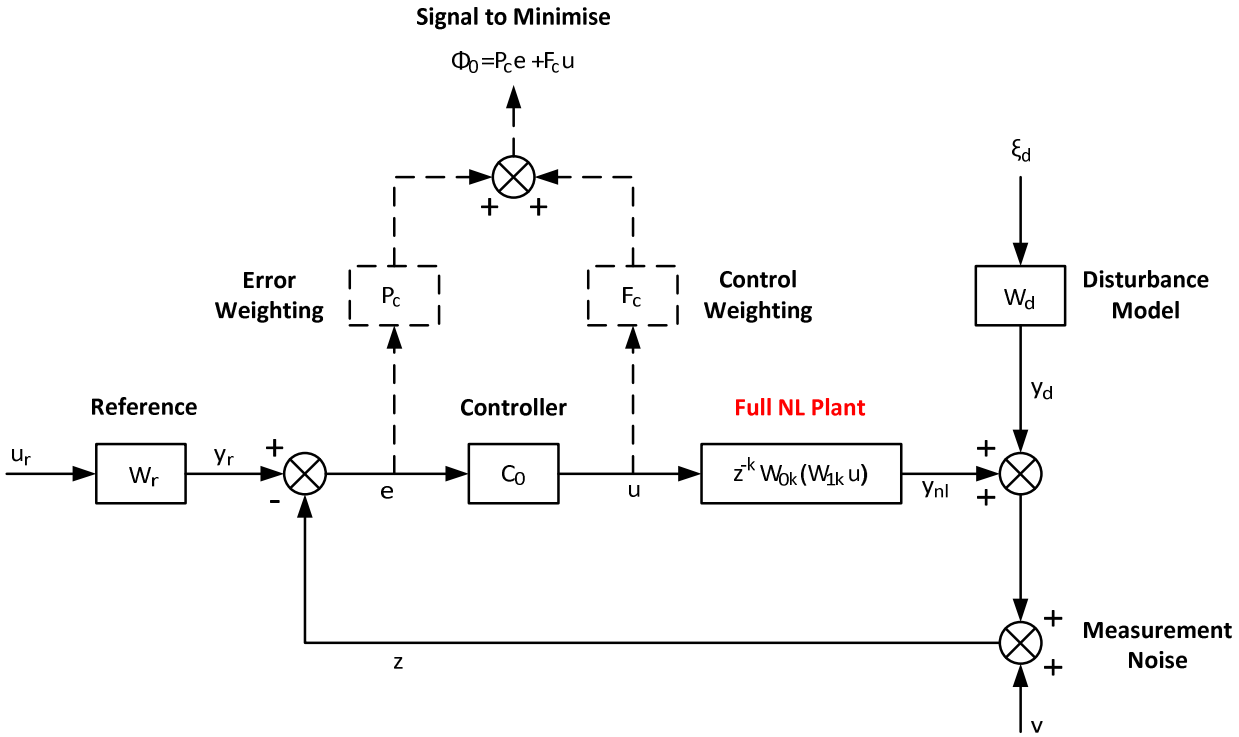


Figure 2:3 State-space *NGMV* controller and cost-function high-level structure.

The starting point for the derivation of the *NGMV* optimal control is the calculation of the Φ_0 signal. The first term $P_c e(t)$ can be substituted with the relation for y_p (recall: $P_c e(t) = C_\varphi x(t) + E_\varphi (W_{1k} u)(t-k)$) derived in Equation 2:18. This yields the following relation recalling that $E_\varphi = -E_p E_0$ and $u_0(t) = (W_{1k} u)(t)$,

$$\Phi_0\ (t) = C_\varphi x(t) + E_\varphi (W_{1k} u)(t-k) + (F_c u)(t) \tag{2:27}$$

Assuming that a common delay factor can be extracted out of the system then $(F_c u)(t)$ can be re-written, to reflect the delayed effect of an input to the output of the system by *k*-steps, as $z^{-k}(F_{ck} u)(t)$. The signal final form and *k-steps* ahead prediction take then the following forms respectively,

$$\Phi_0\ (t) = C_\varphi x(t) + \Big(\big(E_\varphi W_{1k} + F_{ck}\big)u\Big)(t-k) \tag{2:28}$$

$$\widehat{\Phi}_0(t+k|t) = C_\varphi \hat{x}(t+k|t) + \Big(\big(E_\varphi W_{1k} + F_{ck}\big)\Big)u(t) \tag{2:29}$$

The objective of the cost function is to minimise the variance of $\Phi_0\ (t+k)$ or analytically $E\{\Phi_0\ (t+k)^T \Phi_0\ (t+k)\}$. If Φ_0 is put in terms of the prediction and prediction error then the *NGMV* cost-function can be written as follows,

$$J = E\{\hat{\Phi}_0\ (t+k|t)^T \hat{\Phi}_0\ (t+k|t)\} + E\{\tilde{\Phi}_0\ (t+k|t)^T \tilde{\Phi}_0\ (t+k|t)\} \quad (2:30)$$

Given that the term $\tilde{\Phi}_0\ (t+k|t)$ cannot be influenced by control, Equation 2:30 can be minimised by setting $\hat{\Phi}_0$ terms to zero as shown below.

$$\begin{aligned} \hat{\Phi}_0(t+k|t) &= C_\varphi \hat{x}(t+k|t) + \left(E_\varphi W_{1k} + F_{ck}\right)u(t) \\ &= C_\varphi \hat{x}(t+k|t) + E_\varphi (W_{1k}u)(t) + F_{ck}u(t) = 0 \end{aligned} \quad (2:31)$$

Solving Equation 2:31 with respect to $u(t)$ yields the following *NGMV* optimal control signal.

$$u(t) = (-F_{ck})^{-1}\left(C_\varphi \hat{x}(t+k|t) + E_\varphi (W_{1k}u)(t)\right) \quad (2:32)$$

Equation 2:32 provides an easy to implement form of the algorithm as the only guarantee required here is for the F_{ck} term to be invertible. This is reflected in the following diagram.

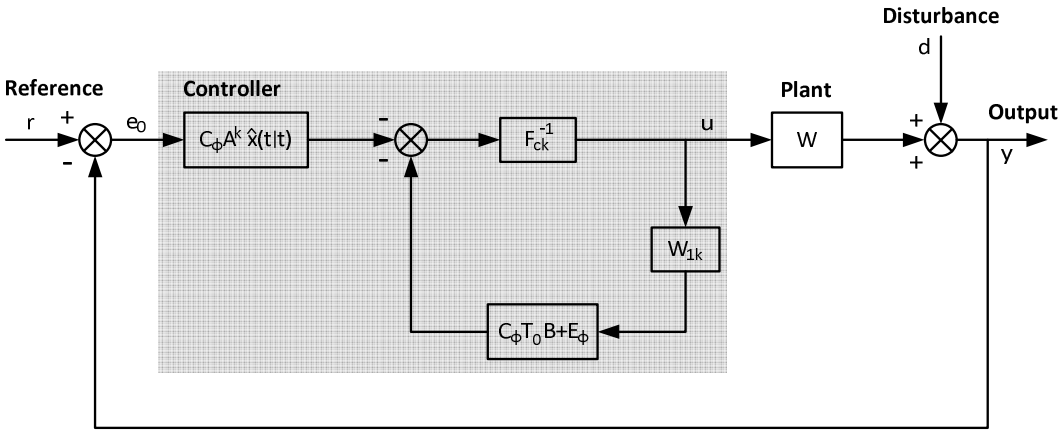


Figure 2:4 *NGMV* controller implementation structure.

The optimal control law can be alternatively written in terms of the current state estimate as shown below, in an alternative form more convenient for implementation.

$$u(t) = -{F_{ck}}^{-1}\left(C_\varphi A^k \hat{x}(t|t) + \left(C_\varphi T_0(k, z^{-1})B + E_\varphi\right)(W_{1k}u)(t)\right) \quad (2:33)$$

This can be implemented as shown in the following diagram.

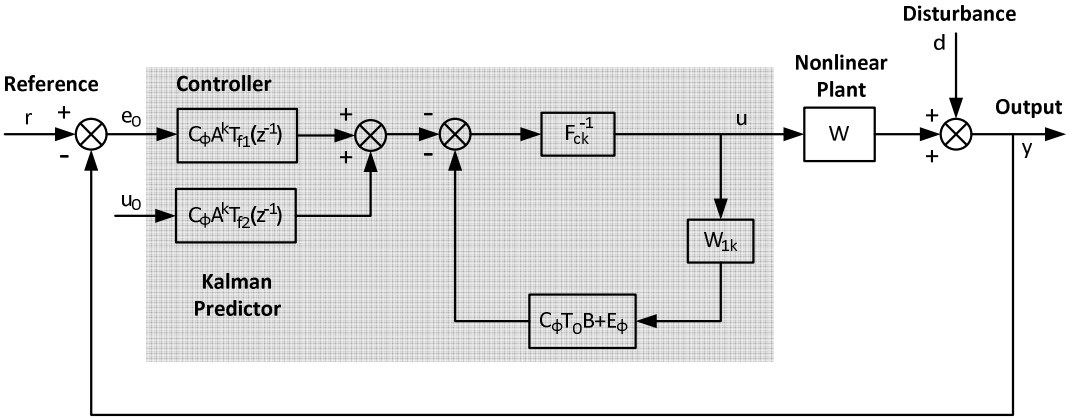


Figure 2:5 *NGMV* controller and *KF* term implementation structure.

2.2.2 NGMV Optimal Control Law Derivation

Control design specifications and required performance, in the case of the *NGMV* strategy, are addressed via the appropriate definition of two weightings, P_c (error weighting) and F_{ck} (control weighting). These weightings can be selected to be either dynamic or scalar. In the simplest case, the algorithm scales down to the *Minimum Variance* controller if the following weightings are specified.

$$P_c = I \text{ and } F_{ck} = 0 \tag{2:34}$$

However, the downside with this option is that it results into aggressive control action and poor robustness properties. Most importantly, as the *Minimum Variance* controller attempts to cancel the plant, the plant model must be stable and invertible. It is useful to mention here that the definition of weightings applies to both *SISO* and *MIMO* systems. The latter is achieved if a diagonal matrix is used for the related weighting. In this case each diagonal element reflects upon a corresponding input-output path in the system. As with classical control the control weightings for the *NGMV* can be selected relative to their impact upon different frequency ranges of the error and control signals.

Although not present in the standard formulation, integral action can be added to the controller by defining the error weighting accordingly. As with classical control the concept here is the presence of high control gain in low frequencies both for providing steady state offset elimination and also good disturbance rejection. Integral action via the error weighting is achieved as follows.

$$P_c = \frac{P_{cn}}{1 - z^{-1}} \tag{2:35}$$

P_{cn} can be defined as a scalar or of the form $P_{cn} = (1 - \alpha z^{-1})$ where α can be selected to provide a faster roll-off for the integral action (this option however is not further explored in

this work). The error weighting can be dynamically tailored to provide high gain within a specific region or frequency. An example of this is seen in Chapter 5 where the error weighting is defined as a 2nd order system to provide accuracy within the band specified by the wave disturbance frequency.

F_{ck} is the control weighting, for a baseline design can be defined as a scalar or linear dynamic component that aims (as a lead term) at sufficient roll-off of the controller in high frequencies to avoid measurement noise amplification. In many cases, the selection of weightings relies upon a good understanding of the system properties and frequency response. A subsequent process of trial and error is usually in order, to achieve the specified performance. A method that can provide (in most cases) reasonable initial results, which relies less heavily on intuition, is suggested in (Grimble and Majecki, 2005) and described in the following section.

2.2.2.1 PID Controller Structure Weighting Selection

If a controller K_0 exists that can stabilise the delay-free plant W_k, then the *NGMV* weightings that result into a stabilising controller can be defined as follows.

$$P_c = K_0 \text{ and } F_{ck} = -I \tag{2:36}$$

This is apparent in the nonlinear operator $(P_c W_k - F_{ck})$ which determines the stability of the closed-loop system. For a linear and negative F_{ck} this operator can be written as follows.

$$(P_c W_k + F_{ck}) = F_{ck}(F_{ck}^{-1} P_c W_k + I) \tag{2:37}$$

The right-hand side term can be considered a return difference operator for the delay-free system with $K_0 = F_{ck}^{-1} P_c$ being the feedback controller. This way the stability of the inverse operator is directly related to the stability of the closed-loop. This provides the option to select the *NGMV* weightings according to the stabilising *PID*. The way this is achieved is by setting the *NGMV* weightings similar to the transfer (or state-space model) of the *PID* controller.

This approach yields a good starting point in design, usually requiring final adjustments to achieve the specified performance. An advantage with this method is that stability is easier to achieve in initial design. To enhance the resulting controller a lead term can be added to provide the high-frequency controller roll-off that was discussed previously.

STATE-SPACE NGMV FORMULATION TUTORIAL EXAMPLE

In this basic *state-space NGMV* tutorial example the required steps to formulate the controller are described in more detail. Here a general *SISO* nonlinear plant is considered under the assumption that it can be effectively broken down into a linear and nonlinear subsystem. Firstly the dynamical equations of the nonlinear subsystem are shown below.

Nonlinear Plant Subsystem W_{1k}:

$$\dot{x}_1 = \frac{x_1 x_2}{1 + x_1^2} + u$$

$$\dot{x}_2 = e^{-(x_1 x_2)^2} + u$$

$$y = x_1$$

This subsystem consists of smooth differentiable nonlinearities and the trajectories of the two states driven by the common input *u* over a period of time can be seen in the following graph. The left axis is scaled with respect to input variation whereas the right axis is scaled with respect to the variation of the states.

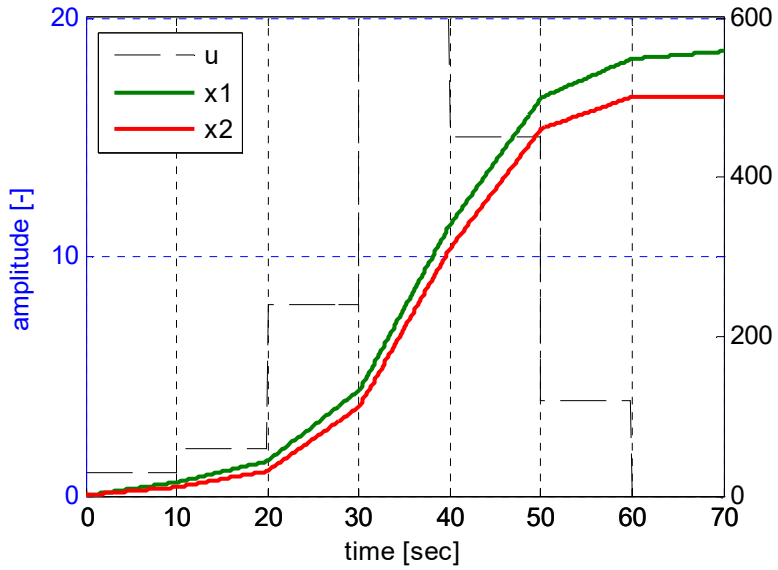


Figure 2:6 System state trajectories for a step sequence input.

Next the linear subsystem is defined. All of the following subsystems are discretised as necessary using the NGMV controller sample time mentioned above.

Linear Plant Subsystem W_0:

$$x_{0_{k+1}} 0.6 x_{0_k} + u_k$$

$$y_{0_k} = 0.6 x_{0_k} + u_k$$

The continuous time step response of W_0 is seen in the following graph.

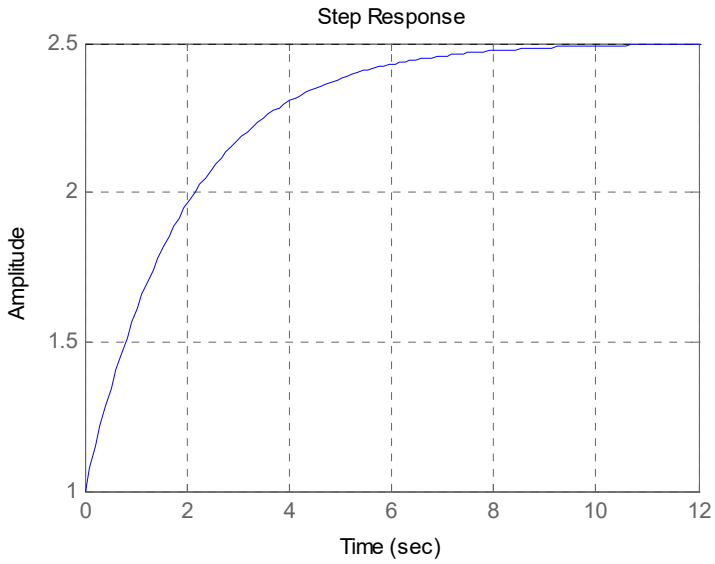


Figure 2:7 Linear plant sub-system step response.

The linear state-space matrices are,

$$A_0 = 0.6, B_0 = 1, C_0 = 0.6, E_0 = 1$$

Next the linear disturbance and reference subsystems need to be defined similar to the *GMV* theory i.e. as the output of linear filters driven by white noise.

Reference Subsystem W_r:

$$x_{r_{k+1}} = 0.9999x_{r_k} + 0.25u_k$$

$$y_{r_k} = 0.2x_{r_k}$$

The linear state-space matrices are,

$$A_r = 0.9999, B_r = 0.25, C_r = 0.2$$

Disturbance Subsystem W_d:

$$x_{d_{k+1}} = 0.8x_{d_k} + 0.25u_k$$

$$y_{d_k} = 0.2x_{d_k}$$

The linear state-space matrices are,

$$A_d = 0.8, B_d = 0.25, C_d = 0.2$$

The next step, important for the *NGMV* weightings selection, is to define an existing stabilising *PID* controller for that system.

Existing *PID* Controller:

The *PID Proportional* gain, *Integral* and *Derivative* time constants are defined as follows.

$$K_p = 0.1, T_i = 4s, T_d = 0s$$

Discretising with a sampling time of $T_s = 1s$ results into the following discrete *PID* transfer function.

$$PID_{TF} = \frac{0.125 - 0.1z^{-1}}{1 - z^{-1}}$$

NGMV Error Weighting P_c Definition:

When using this transfer function as the error weighting P_c it is recommended to slightly modify by adding a near-integrator term which results into a stable weighting. This is shown in the modified transfer function below.

$$P_{c_{near\_int}} = \frac{0.125 - 0.09999z^{-1}}{1 - z^{-1}}$$

This transfer function can easily be translated into state-space formulation as seen below.

$$x_{p_{k+1}} = -0.0001x_{p_k} + 0.125u_k$$

$$y_{p_k} = 0.2x_{p_k} + 0.125u_k$$

The linear state-space matrices are,

$$A_p = 0.9999, B_p = 0.125, C_p = 0.2, E_p = 0.125$$

NGMV Control Loop Gains Calculation:

The next step in the controller formulation is the derivation of the components of the optimal NGMV control law (see reminder below).

$$u(t) = -{F_{ck}}^{-1}\left(C_\varphi A^k \hat{x}(t|t) + \left(C_\varphi T_0(k, z^{-1})B + E_\varphi\right)(W_{1k}u)(t)\right)$$

These components as seen in Figure 2:4 are the following.

- $C_\varphi A^k$, the linear term multiplied by the output of the Kalman Filter.
- $C_\varphi T_0(k, z^{-1})B + E_\varphi$, the linear loop gain.

$C_\varphi A^k$ Definition:

Recall,

$$C_\varphi = [-E_p C_0 \quad -E_p C_d \quad E_p C_r \quad C_p]$$

and

$$E_\varphi = -E_p E_0$$

Substituting the equivalent matrices results into,

$$C_\varphi = [-0.075 \quad -0.025 \quad -0.025 \quad 0.2]$$

and

$$E_\varphi = -0.125$$

A is the state matrix of the augmented system state equation shown as follows.

$$\begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_{k+1} = \begin{bmatrix} A_0 & 0 & 0 & 0 \\ 0 & A_d & 0 & 0 \\ 0 & 0 & A_r & 0 \\ -B_p C_0 & -B_p C_d & B_p C_r & A_p \end{bmatrix} \begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_k + \begin{bmatrix} B_0 \\ 0 \\ 0 \\ -B_p E_0 \end{bmatrix}$$

Substituting the equivalent matrices results into,

$$\begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_{k+1} = \begin{bmatrix} 0.6 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.9999 & 0 \\ -0.075 & -0.025 & 0.025 & 0.9999 \end{bmatrix} \begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0 \\ 0 \\ -0.125 \end{bmatrix}$$

and for an arbitrary time-delay of *k = 10* steps,

$$A^k = \begin{bmatrix} 0.006 & 0 & 0 & 0 \\ 0 & 0.1074 & 0 & 0 \\ 0 & 0 & 0.999 & 0 \\ -0.1862 & -0.1115 & 0.2498 & 0.999 \end{bmatrix}$$

and

$$C_{\varphi}A^{k} = [-0.0377 \quad -0.025 \quad 0.0749 \quad 0.1998]$$

$C_{\varphi}T_0(k, z^{-1})B + E_{\varphi}$ Definition:

Recall the equation for the linear operator $T_0(k, z^{-1})$,

$$T_0(k, z^{-1}) = z^{-1}\left(I + z^{-1}A + z^{-2}A^2 + \ldots + z^{-k+1}A^{k-1}\right)$$

This results into the following discrete polynomial*4x4* transfer function matrix (dimensions of the augmented system *A* matrix).

TF: In 1-> Out 1

$$z^{-1} + 0.6z^{-2} + 0.36z^{-3} + 0.216z^{-4} + 0.129z^{-5} + 0.0776z^{-6} + 0.0466z^{-7} + 0.0279z^{-8} + 0.0168z^{-9} + 0.01z^{-10}$$

TF: In 1-> Out 2, 3 = 0

TF: In 1-> Out 4

$$-0.075z^{-2} - 0.12z^{-3} - 0.147z^{-4} - 0.1632z^{-5} - 0.1729z^{-6} - 0.1787z^{-7} - 0.1822z^{-8} - 0.1842z^{-9} - 0.1855z^{-10}$$

TF: In 2-> Out 1 = 0

TF: In 2-> Out 2

$$z^{-1} + 0.8z^{-2} + 0.64z^{-3} + 0.512z^{-4} + 0.4096z^{-5} + 0.3277z^{-6} + 0.2621z^{-7} + 0.2097z^{-8} + 0.1678z^{-9} + 0.1342z^{-10}$$

TF: In 2-> Out 3 = 0

TF: In 2-> Out 4

$$-0.025z^{-2} - 0.045z^{-3} - 0.06099z^{-4} - 0.07379z^{-5} - 0.08402z^{-6} - 0.0922z^{-7} - 0.09875z^{-8} - 0.104z^{-9} - 0.1082z^{-10}$$

TF: In 3-> Out 1, 2 = 0

TF: In 3-> Out 3

$$z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + 0.9994z^{-7} + 0.9993z^{-8} + 0.9992z^{-9} + 0.9991z^{-10}$$

TF: In 3-> Out 4

$$0.025\,z^{-2} + 0.05z^{-3} + 0.07499z^{-4} + 0.09997z^{-5} + 0.125\,z^{-6} + 0.1499\,z^{-7} + 0.1749\,z^{-8} + 0.1999\,z^{-9} + 0.2248z^{-10}$$

TF: In 4-> Out 1, 2, 3 = 0

TF: In 4-> Out 4

$$z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + 0.9994z^{-7} + 0.9993z^{-8} + 0.9992z^{-9} + 0.9991z^{-10}$$

Substituting into the loop gain equation results in the following gain.

$$\begin{aligned} C_\varphi T_0(k, z^{-1})B + E_\varphi \\ &= -0.125 - 0.1z^{-1} - 0.08499z^{-2} - 0.07599z^{-3} - 0.07058z^{-4} \\ &- 0.06734z^{-5} - 0.06539z^{-6} - 0.06421z^{-7} - 0.06351z^{-8} - 0.06308z^{-9} \\ &- 0.06282z^{-10} \end{aligned}$$

A standard Kalman Filter is utilised to obtain the state estimates within which the augmented plant model is used as described above. Optionally the weighted error state can also be used within the *KF* model. The combined NGMV terms can be seen in the following internal controller diagram. The *Smith Predictor* time-delay compensation similarity becomes clearer in this diagram with the use of a delay-free model of the plant.

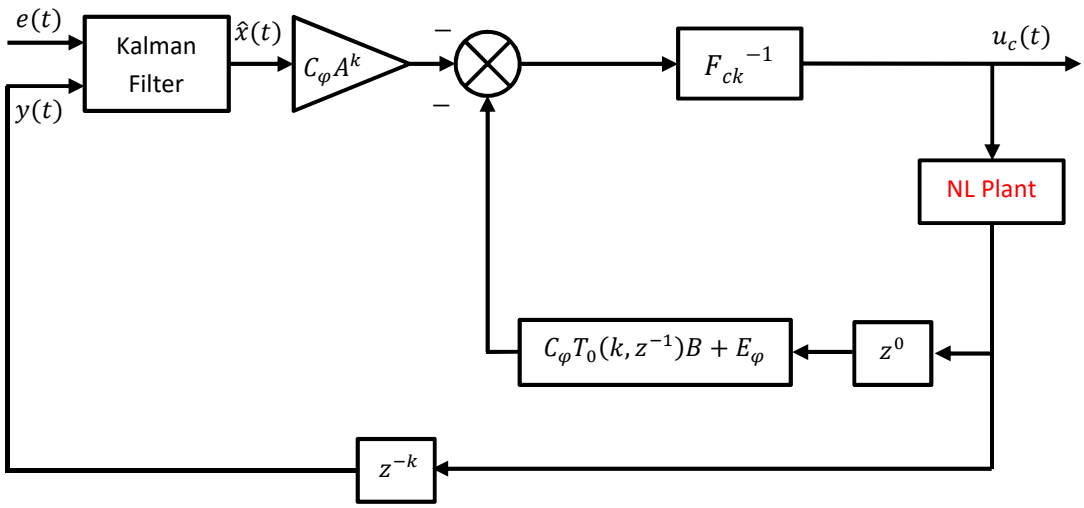


Figure 2:8 *NGMV* internal loop formulation.

A simulation comparison between a baseline *PID* controller and the equivalent state-space *NGMV* using the same gains was formulated. An arbitrary setpoint trajectory was utilised to demonstrate the benefits of the *NGMV* as illustrated in the following graph.

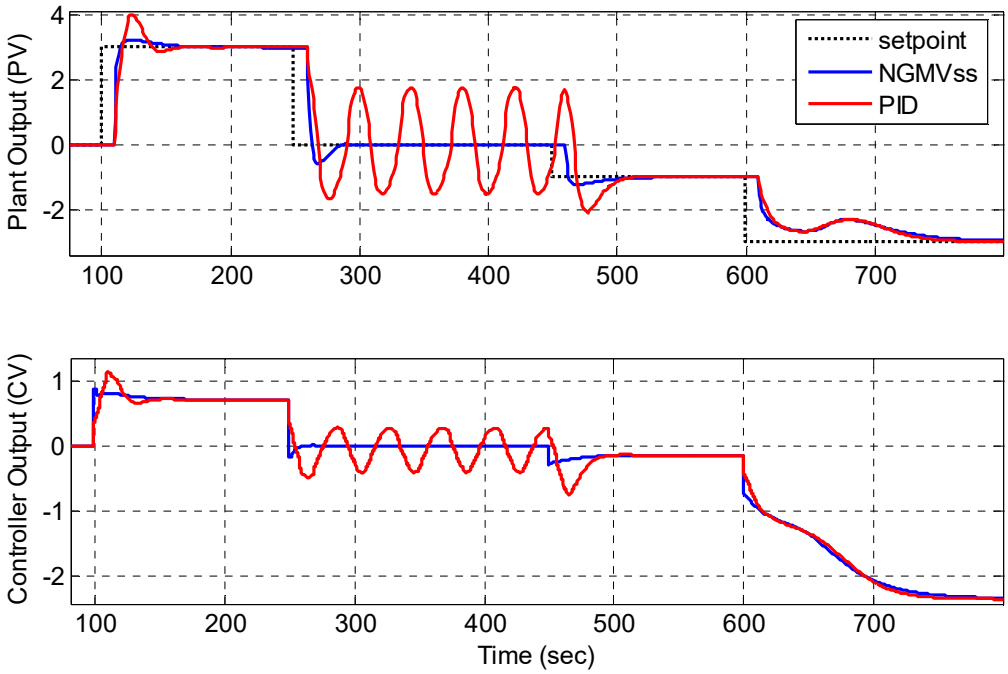


Figure 2:9 *NGMV* internal loop formulation.

In this graph the *NGMV* is seen to maintain its tracking performance as the trajectory is moving across different operating regions whereas the *PID* deteriorates significantly in performance especially around *0*. This is achieved thanks to the information available to the controller by the internal nonlinear model.

2.3 An Introduction to Linear Parameter Varying Systems

LPV systems could fall into the wider range of gain-scheduling methods however they differ, in that they represent dynamical systems and not static approximations around some operating points. The general formulation of an *LPV* system is shown below.

$$\begin{aligned} \dot{x}(t) &= A\big(\rho(t)\big)x(t) + B\big(\rho(t)\big)u(t) \\ y(t) &= C\big(\rho(t)\big)x(t) + E\big(\rho(t)\big)u(t) \end{aligned} \tag{2:38}$$

Where *x, u, y* are the state, input and output system vectors respectively and *A*, *B*, *C* and *E* state-space matrices now depend upon a time-varying parameter *ρ*. These systems are linear regarding their structure being that of an *LTI* system, but nonlinear in nature as the system matrices vary with time and depend on the realisation. Inter-state dependencies may also come through using the varying parameter to represent states.

In literature these parameters are generally considered as bounded within a set of values Δ_ρ ($\{\rho: \mathbb{R} \geq 0 \rightarrow \Delta_p\}$) and have been used to represent the following.

- An exogenous time-varying quantity. The assumption that this quantity can be measured in real-time is typically made here.
- A system state. This usually reference to the case where the parameter is a function of a system state and hence endogenous. This category is more fit to gain-scheduled systems.

The class of *LPV* systems as well as their stability properties depend upon the type of variation of the parameters such as,

- Arbitrarily fast varying parameters.
- Slowly varying parameters (in this case the rate of ρ is bounded within the derivative of the parameter set, $\dot{\rho} \in \dot{\Delta}_\rho$).
- Piecewise constant parameters, where the parameter is a piecewise constant $\rho_i, i = 1,2,3 \ldots$.
- Switched systems with *N* modes where the parameter represents discrete modes in which the system transitions and can be represented as $\left\{\rho: \mathbb{R} \geq 0 \rightarrow \{0,1\}^N : \sum_{i=1}^{N} \rho_i = 1\right\}$.
- Periodic systems in which the parameter, as the name suggests, varies in a period *T* ($\rho(t) = \rho(t+T), t \geq 0$).

LPV systems were initially conceived within the framework of gain-scheduling control design for nonlinear systems (Bruzelius, 2004) (an example in Balas, 2002). The *LPV* approach brought design a step closer to nonlinear systems representation and ahead of the gain-scheduling method of switching via interpolation, between a set of linear controllers produced based on linearization of the system at multiple points across the operating range. The exogenous vs. endogenous distinction becomes clear when classifying *LPV* systems formulations into general *LPV* or *Quasi-LPV* (*qLPV*).

Quasi-LPV Systems:

This category refers to the approximation of nonlinear systems in which the varying parameter is representative of a system state (endogenous). A common example of this representation is illustrated in the following system.

$$\dot{x}(t) = -x(t)^3 \tag{2:39}$$

with a possible *qLPV* shown below,

$$\dot{x}(t) = -\rho(t)^2 x(t), \;\; \rho(t) \coloneqq x(t) \in \mathbb{R} \tag{2:40}$$

The representation in Equation 3:9 is asymptotically stable for every parameter $\rho \neq 0$ providing stability properties identical to the original nonlinear system. This however is not always the case with *LPV* approximation in general. It cannot be automatically assumed that they have equivalence with the original nonlinear system in terms of stability or any control related property. This class of *LPV* systems can also accommodate state-variables as being the varying parameters hence often providing a good approximation to the original nonlinear response. An important observation in the understanding of *LPV* systems is that their structure resembles that of *Linear Time-Varying* (*LTV*) systems. This is particularly useful in predictive control where this structure can be used to increase fidelity of future states and outputs generation. The latter is used within the wind turbine *LPV-NPGMV* example in Chapter 6.

2.3.1 LPV, qLPV and State-Dependent Systems Relationships

Depending upon the specific control group this theory is used, it can either be regarded and related to the state-dependent Riccati equation or the *LPV* design framework. Considering the process of deriving the control laws and design architecture, there is no obvious distinction between *LPV*, *qLPV* or state-dependent formulations. The main difference resides within the analysis of stability properties with *LPV* being a special form of time-varying linear system models, but *qLPV* and state-dependent formulations being true nonlinear system representations.

There are indirect ways to approach the issue of stability avoiding in depth analysis of the system. An example of this can be found within the formulation of the model itself. Given a discrete-time state-space model approximation of a nonlinear system, if the *A* matrix can be derived as a function of the state *x(t)*, then this matrix (denoted *A(x(t))*), can be approximated in a model for control design (referred to as the design model) as *A(x(t-1))*. The design model then becomes an *LPV* system. This simplifies some of the theoretical questions regarding the stability of an optimal control based on this design model. It does not however avoid the problem of showing the system is stable on the true underlying nonlinear system. This work focusses more on control design and this subtle difference in such models can be neglected.

2.3.2 LPV System Derivation

This section describes two methods for obtain an *LPV* formulation from a nonlinear system. Emphasis is put on the second as it is the one used in application Chapter 6.

Jacobian Linearisation:

This method is making use of the *Taylor-Series* expansion, as it was described in Chapter 1 (linearisation) to approximate the nonlinear system at specific operating points. This process can be carried out at multiple equilibrium points (assuming they exist) which then will define the operating region for the *LPV* model. As mentioned previously the transition between different equilibrium points can be achieved by interpolation. The equilibrium points can be determined by solving $\dot{x} = 0$. For the LPV case consider, as previously ρ, as the varying parameter. An equilibrium point $(op(x_{op}, u_{op}) \rightarrow \dot{x}_{op} = 0)$ can be derived as follows,

$$f\big(x_0(t,\rho), u_0(t,\rho)\big) = 0 \tag{2:41}$$

The plant model can then be defined in the following small changes *LPV* form,

$$\begin{bmatrix}\dot{x}_\delta(t)\\ y_\delta(t)\end{bmatrix} = \begin{bmatrix}A(\rho) & B(\rho)\\ C(\rho) & D(\rho)\end{bmatrix}\begin{bmatrix}x_\delta\\ u_\delta\end{bmatrix} \tag{2:42}$$

where,

$$\begin{aligned} A(\rho) &= \frac{\partial f}{\partial x}\big(x_0(t,\rho), u_0(t,\rho)\big), \qquad B(\rho) = \frac{\partial f}{\partial u}\big(x_0(t,\rho), u_0(t,\rho)\big)\\ C(\rho) &= \frac{\partial g}{\partial x}\big(x_0(t,\rho), u_0(t,\rho)\big), \qquad D(\rho) = \frac{\partial g}{\partial u}\big(x_0(t,\rho), u_0(t,\rho)\big) \end{aligned} \tag{2:43}$$

The process then follows the steps mentioned above which is the determination of a set of equilibrium points across a continuous or discrete operating trajectory within which the corresponding *LPV* controllers will be designed and switched via interpolation. Note here the similarity of this approach to that of gain scheduling mentioned previously. This implies similar issues regarding stability properties and performance as well.

Modified Jacobian Linearization:

Unlike the previous approach, modified *Jacobian linearization* is not restricted to equilibrium points. In this approach linearization of the nonlinear system can be performed at any operating point $op(x_{op}, u_{op})$, in other words producing a linear approximation at any point across the operating trajectory. The difference becomes more apparent if put into context of the *Taylor-Series* expansion seen below.

$$f(x,u) = f(x_0,u_0) + \frac{\partial f}{\partial x}(x_0,u_0)(x - x_0) + \frac{\partial f}{\partial u}(x_0,u_0)(u - u_0) \quad (2:44)$$

In this case the first term (free response component) on the right hand side can be non-zero when linearization is not performed at equilibrium. The *LPV* model can be then written as follows,

$$\dot{x}(t) = f(x_0,u_0) + \frac{\partial f}{\partial x}(x_0,u_0)(x - x_0) + \frac{\partial f}{\partial u}(x_0,u_0)(u - u_0) \quad (2:45)$$

The free response term can be expanded to provide more intuitive implementation as follows,

$$\dot{x}(t) = \left(f(x_0,u_0) - \frac{\partial f}{\partial x}(x_0,u_0)x_0 - \frac{\partial f}{\partial u}(x_0,u_0)u_0\right) + \frac{\partial f}{\partial x}(x_0,u_0)x + \frac{\partial f}{\partial u}(x_0,u_0)u \quad (2:46)$$

The previous equation can be put in an *LPV* form as follows,

$$\dot{x}(t) = d_0(\rho) + A_0(\rho)x + B_0(\rho)u \quad (2:47)$$

where $d_0(\rho)$ can be seen as a disturbance term. Improvement in accuracy with this method, lies in that (x_0,u_0) can be derived from the previous sample instant. The main advantage here is the utilisation of the original nonlinear system equations for the derivation of the linear approximation.

2.3.3 NGMV Applied to LPV Systems

The *NGMV* controller for *LPV* systems can be derived following the same type of analysis as will be followed in later sections. However, the solution and structure should be fairly obvious since the later *NPGMV* controller is derived using the *LPV* model and *NGMV* for such systems is just a special case. The formulation of the *LPV-NGMV* control problem follows directly from the *LPV-NPGMV* derivation. This will be discussed in more detail after the *LPV-NPGMV* has been described in Section 4.4.2.

Chapter 3 LPV-NGPC Structure

In the initial part of this chapter the *Nonlinear Predictive Generalised Minimum Variance* (*NPGMV*) algorithm is presented as a combination of the concepts described in Chapter 2 and the *Nonlinear Generalised Predictive Control* scheme. Based on the *NPGMV* derivation, the algorithm is then adapted, at the second part of the chapter, to a further evolution which is designed to work with *Linear Parameter Varying* (*LPV*) systems. A brief overview of this system formulation, which is used to approximate nonlinear systems, is also provided. The chapter consists of the following sections:

Section 1 – An Introduction to *Nonlinear Predictive Control*; an overview of the main principles and aspects of the *Nonlinear Generalised Predictive Control* scheme. An overview of the linear *Generalised Predictive Control* scheme is also provided.

Section 2 – *Nonlinear Generalised Predictive Control* Derivation as the basis for the *LPV-NPGMV* algorithm formulation.

3.1 An Introduction to Nonlinear Predictive Control

This section provides a brief introduction the concept of *Nonlinear Predictive Control* (*NMPC*) and sets the basis on which the *LPV-Predictive* paradigm will be analysed in the last section of the chapter. For consistency with the algorithm structures described so far, only the state-space formulation of the *NL MPC* is explored.

3.1.1 Overview of the MPC Concept

Early developments on the concept of predictive control are traced back to the revolutionary approach on optimal state-feedback control by *Kalman* and following methodologies of *LQR* and *LQG* controllers. In all these approaches a cost function was involved in which output errors and control actions were penalised and the optimal control inputs to the plant where computed with respect to the minimisation of that function. This framework allowed both tracking and regulating processes to be dealt with whilst control effort was maintained within a specified acceptable range at the same time. Various practical extensions to these schemes were addressed over the years, like the addition of integral action and direct control of plant outputs by modifying the cost-function. Practical limitations in real applications however motivated the work behind a predictive control methodology which should be able to address the following issues (Richalet et al., 1978).

- Input, state and output constraints; in the vast majority process variables are only able to vary within certain limits. Apart from physical limitations this may be the result of design as it has been shown that many industrial processes yield greater economic benefits when operating near constraints.

- Process nonlinearities; the internal plant models that are required within these methodologies are linearised across a subset of the full range of operation which limits performance and stability conditions.
- Model uncertainty; deviations between the actual process and its mathematical representation due to model mismatch or unmodelled noise and process disturbances can also add to the deterioration of the control solution.

In early applications and especially in the process industry, *MPC* strategy started as the supervisory control module and was located in the higher level of a hierarchical structure. Within this framework the *MPC* controller functioned more like an optimiser (off-line in cases), that determined the optimal setpoints for the low level control loops (usually simple *PID*, *Lead-Lag* controllers). This setpoint computation was often based on plant economics and other slow varying factors whereas control at the low level dealt with faster dynamic components (valves etc.). An example of this architecture can be seen in the following figure.

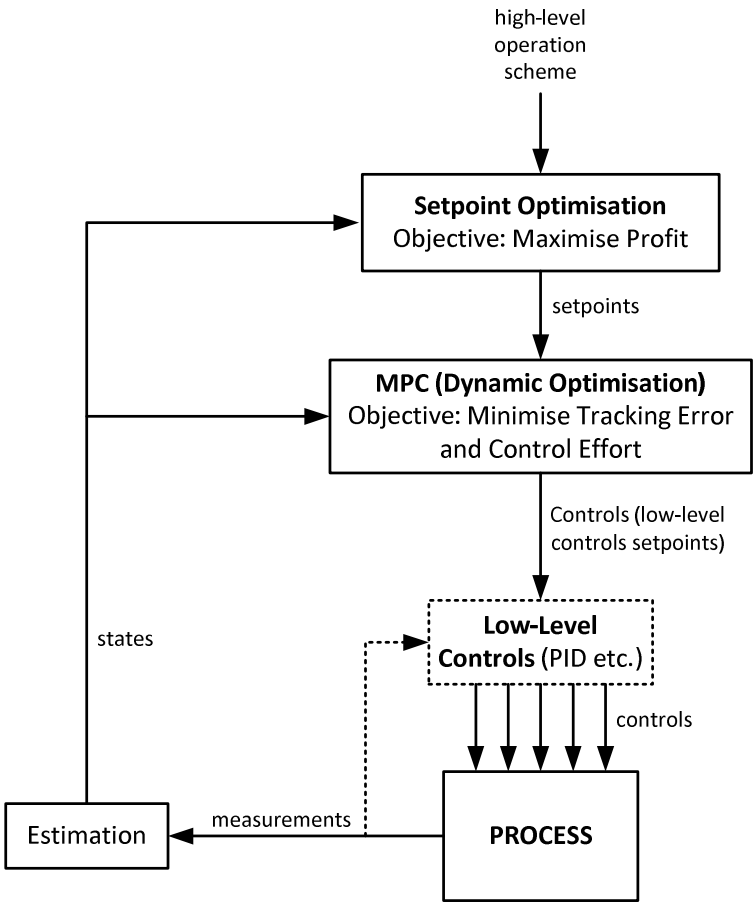


Figure 3:1 MPC within a hierarchical process control structure.

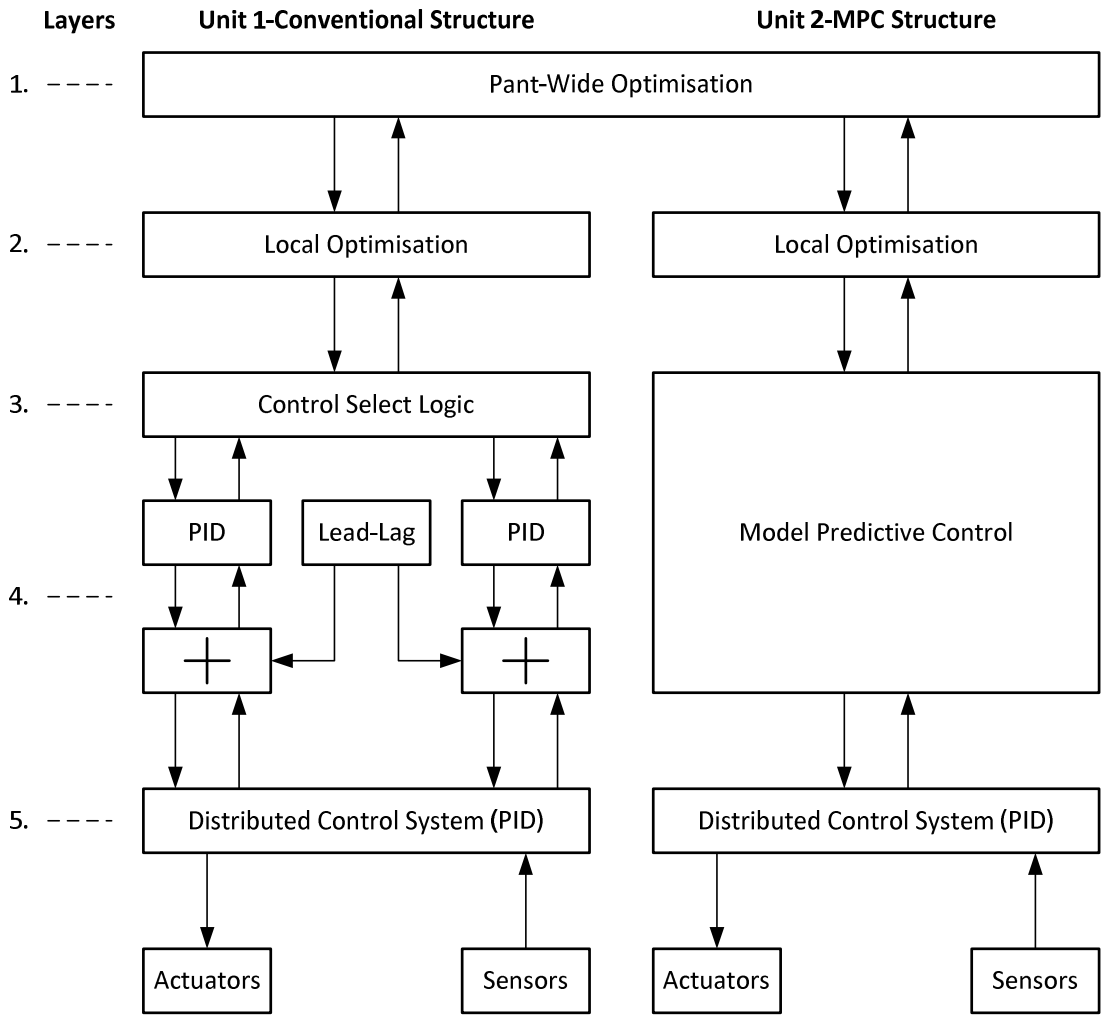


Figure 3:2 MPC and classical control within a hierarchical process control structure.

Initial practical developments on *MPC* methodology were done by groups led by Richalet (*IDCOM*), Cutler and Ramaker (*DMC*) (Cutler and Ramaker, 1980). The first involved impulse response models identified from open-loop tests to generate predictions, considered input and output constrains into the optimal solution which is calculated using an iterative algorithm and noted as the mathematical dual of identification. Richalet's group emphasised the importance of using the *MPC* scheme within a control hierarchy, in this case though in a more dynamical sense between the optimiser and the low level controls (Richalet, 1993). The second approach involved linear step response models for the same purpose, and used a least-squares cost-function solved within a finite horizon. Both schemes could be employed for either *SISO* or *MIMO* systems and were able to provide both tracking of a future trajectory whilst regulating the aggressiveness of control by applying sufficient weighting on control moves. The second however in its initial development did not considered constraints in its least-squares solution. Further development by Shell led to the *QDMC* by reformulating the *DMC* as a

quadratic programing (*QP*) problem which dealt with constraints explicitly (Garcia and Morshedi, 1986).

According to (Qin and Badgwell, 2003) more than 4500 applications are reported to use *MPC* based schemes by 2003 ranging from process to aerospace industries. Theoretical and implementation issues have been thoroughly investigated in (Qin and Badgwell, 2003). Very useful research that adequately covers the subject of linear *MPC* can be found in (Morari and Lee, 1999), (Bitmead et al., 1990), (Camacho and Bordons, 1998) and (Maciejowski, 2002).

A break through the *MPC* approach offered was a good alternative to address the difficulty in solving a stabilising feedback control problem that minimises a cost-function subjected to constraints. This difficulty lies in that oftentimes an analytical closed solution is not possible. The *MPC* methodology, in conjunction with the receding horizon strategy, approaches this by solving an open-loop optimisation problem and implementing the first element of the computed controls sequence repeatedly at each iteration (Kouvaritakis et al., 2000). It is a natural consideration that such an approach will increase the computational burden of the algorithm and this can be especially a problem in the case of systems with fast dynamics and large number of states.

To summarise the *MPC* philosophy involves solving on-line an open-loop finite horizon quadratic optimal control problem subject to input, state and/or output constraints. At each iteration *k* of the algorithm, current inputs sequence and measured variables are used within an internal discrete mathematical model of the process which is used to predict the response of the system at future times. This prediction is computed over a specified time window in discrete steps (*Prediction Horizon Np*) and used within the quadratic *MPC* cost-function (Findeisen and Allgöwer, 2002). The latter is then minimised with respect to the optimal controls vector that achieves this objective subject to constraints. The length of the optimal controls vector is defined as the *Control Horizon* N_u. The *Receding Horizon* principle is then employed according to which only the first element of the computed optimal sequence of controls is applied to the plant and this sequence of steps is repeated at the next iteration, for the updated state. The solution of this dynamic optimisation problem at each iteration, is an advantageous trait when it comes to compensating for uncertainties due to model mismatch, disturbances or noise that may be present in the process. It is actually by this principle that feedback is indirectly implemented in the *MPC* solution. A graphical illustration of this concept is shown in Figure 3:3.

The linear class of *MPC* algorithms has been found to perform adequately mostly in relatively simple slow varying processes and being able to handle both soft and hard constraints where these apply. However, with the increasing complexity of processes and constraints set by more demanding designs and regulations, a control solution is required to be able to operate the system in a wide range of conditions. This, coupled with the inherent nonlinear characteristics of most real-life processes, motivates the development of more generic control schemes able to accommodate these requirements.

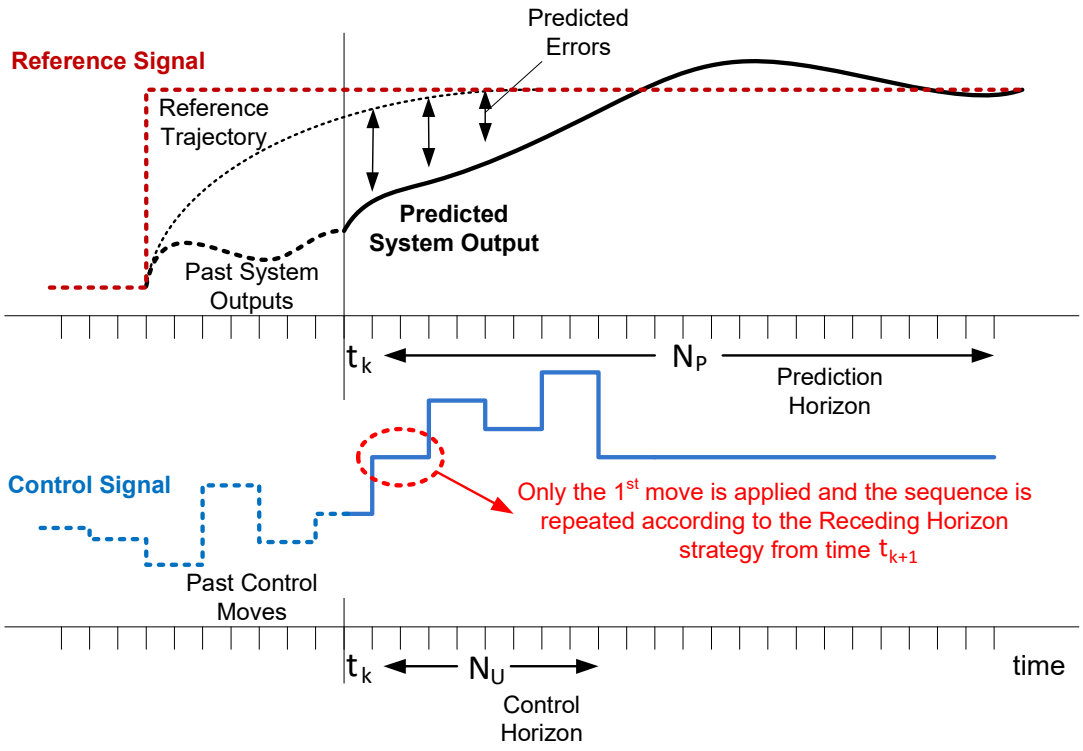


Figure 3:3 The MPC strategy in a nutshell.

3.1.2 Motivation towards Nonlinear MPC (NMPC)

The nonlinear class of *MPC* (*NMPC*) algorithms as the name suggests, unlike linear schemes that use linear models to generate predictions or considers linear constraints etc., uses nonlinear models or even nonlinear constraints in the optimal solution. The main motivation behind *NMPC* is the development of schemes that could deal with either linear systems with input, output and state constraints or systems that are inherently nonlinear. Systems that exhibit non-minimum phase behaviour (i.e. have unstable zeros), introduce another difficulty in control design in which case, regardless of a linear or nonlinear approach, oftentimes a separate closed-loop design to deal with that specific region where the unstable zero lies is inevitable (Cannon and Kouvaritakis, 2002).

Given that the core strength of nonlinear approaches lies on the model fidelity a notable difficulty that immediately springs is that of developing an adequate nonlinear model for control design. These models can be obtained either via *first principle* modelling or other methods (where first principle equations are not available, high order or difficult to implement) like *black-box identification* (Murray-Smith and Shorten, 2005). Oftentimes a combination of both yields efficient models for this purpose and this is known as *grey-box* modelling. There are also approaches that use *Neural Networks*, *Fuzzy Logic* etc. (Qina and Badgwel, 2003).

NMPC approaches include off-line optimisation like neighbouring extremals in which a full solution of the optimal control problem is derived off-line and explicit schemes where a series of approximations are precomputed across the operating range and the appropriate controller is chosen based upon the current position of the system state. This approach poses obvious computational complexity when the system has a very large operating range and/or large number of states.

3.1.3 NMPC Mathematical Formulation

A brief overview of the mathematical basis that all *MPC* schemes derive is provided in this section. Since the development of the *NPGMV* is influenced greatly by the *Nonlinear Generalised Predictive Control* (*GPC*) scheme, this description is intentionally geared towards the latter. The main aspects and requirements behind an *MPC* scheme are as explained in (Findeisen and Allgower, 2002),

- *NMPC* allows the direct use of nonlinear models for prediction.
- *NMPC* allows the explicit consideration of state and input constraints.
- In *NMPC* a specified time domain performance criterion is minimized on-line.
- In *NMPC* the predicted behaviour is in general different from the closed-loop behaviour.
- For the application of *NMPC* typically a real-time solution of an open-loop optimal control problem is necessary.
- To perform the prediction, the system states must be measured or estimated.

Some of the most important consideration behind design of an *MPC* strategy as mentioned in (Findeisen and Allgower, 2002) are summarised below.

- Method by which the predictions are generated.
- Type of cost-function which implements the optimisation problem.
- Method or algorithm used for the optimisation problem solution.
- Initial conditions selection as a starting point for the optimal solution.

The last point is very important in nonlinear processes. The reason is that the cost-function being non-convex can have multiple minima in which case the selection or initial conditions guess is a crucial criterion for convergence of the algorithm.

The general formulation of an *NL MPC* scheme process can be described starting with the generic form of a nonlinear process captured by the following discrete time invariant state and output equations.

$$x(t+1) = f\big(x(t), u(t)\big)$$
$$y(t) = g\big(x(t)\big) \tag{3:1}$$

where $f, g \in C$ are functions that describe nonlinear dynamics under the assumption that $f(0,0) = 0$. The system described by Equations 3:1 is assumed to be causal (i.e. the *feed-through* term is zero). If constraints are present these can be represented with the following vectors relative to control effort *u*, state x and output constraints respectively.

$$U = [u_{min}, u_{max}], X = [x_{min}, x_{max}], Y = [y_{min}, y_{max}]$$

with

$$u_{min} \leq u(t) \leq u_{max}$$
$$x_{min} \leq x(t) \leq x_{max} \tag{3:2}$$
$$y_{min} \leq y(t) \leq y_{max}$$

If the constraints vector occupies the entire space (i.e. $U = R^m$) then the system is considered unconstrained whereas if it is a subspace (i.e. $U \subseteq R^m$) then the system is considered constrained. Similar to the linear *MPC* the objective here as well is to derive the optimal control moves vector U^* that minimises the following cost function (in discrete time) subject to the above constraints within a finite horizon N_p ($t \in [t_k, t_{k+N}]$).

$$\min_u = \sum_{i=0}^{N_p - 1} J(\hat{x}_{t+i}, u_{t+i}) \tag{3:3}$$

where $\hat{x}_{t+i}$ is the vector of future state predictions and u_{t+i} the vector of future control actions. At present time t the controller uses the available measurements to generate predictions of the system states and outputs within a specified prediction horizon N_p. These can be described by the following relations,

$$\hat{x}(t+1|t) = f\big(\hat{x}(t), u(t)\big)$$
$$\hat{y}(t|t) = g\big(\hat{x}(t)\big) \tag{3:4}$$

In an ideal scenario without model mismatch and/or disturbances these predictions would match the output of the real process. In reality this is never the case, therefore the mismatch

manifests as the discrepancy between the model and real plant outputs $e_t = y_t - \hat{y}_{t|t}$. This error is mitigated via the resulting optimal controls vector $[u_t, u_{t+1}, \dots, u_{t+N_p}]$.

In Section 3.3 a form of *Nonlinear GPC* is discussed on which the later *LPV-NPGMV* formulation is based. Firstly, it would be useful to include an overview of *LPV* systems, to allow for a more intuitive progression into the derivation of the *LPV-NPGMV* control law.

3.2 An Introduction to Generalised Predictive Control (GPC)

A brief description of the state-space formulation of the linear *GPC* algorithm, as introduced in (Clarke et al., 1987), is provided in this section. The *GPC* algorithm is available both in polynomial and state-space formulations, only the latter is however considered here as this is also the framework used for the various *NGMV* versions.

As an alternative representation to Transfer Functions a linear time-invariant system can be described by static matrices and a number of integrators depending upon the number of the internal system states. There are a few advantages in using this representation such as being able to seamlessly extend to *MIMO* systems and utilise simple matrix algebra within sophisticated control laws. Another advantage of the state-space formulation and specific to the *GPC* is that it can be conveniently used to generate future predictions of the response of the system and hence the future errors (difference between predicted trajectory and predicted response) which are required for the derivation of the predictive algorithm.

Let us assume a linear state-space system described by the following equations:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Gw(t) \\ y(t) &= Cx(t) + w(t) \end{aligned} \tag{3:5}$$

where $w(t)$ is a general stochastic component which can be used to represent process uncertainties when multiplied by *G* and measurement noise when added to the output of the system. The one step ahead output prediction can be obtain using this equation as follows.

<u>One-step Ahead Prediction</u>:

$$\begin{aligned} \hat{y}(t+1) &= Cx(t+1) + w(t+1) \\ &= CAx(t) + CBu(t) + CGw(t+1) \end{aligned}$$

Note it is *y* hat since it is an estimate of what we think *y* is going to be. Here we assume that we cannot measure or predict *w* terms into the future and therefore the best prediction is to simply set them zero (a disturbance model could be incorporated if available). This equation can now be used to obtain the two step ahead prediction and so on.

Two-step Ahead Prediction:

$$\hat{y}(t+2) = Cx(t+2) + w(t+2)$$
$$= CAx(t+1) + CBu(t+1) + CGw(t+1) + w(t+1)$$
$$= CAAx(t) + CABu(t) + CAGw(t) + CBu(t+1) + CGw(t+1) + w(t+2)$$

This procedure is summarised in the following generalised expression for the *j-step* ahead prediction.

j-step Ahead Prediction:

$$\hat{y}(t+j|t) = \underbrace{CA^j\hat{x}(t|t)}_{\textit{free response}} + \underbrace{\sum_{n=1}^{j} CA^{j-n}Bu(t+n-1) + CA^{j-1}G\big(y(t) - C\hat{x}(t|t)\big)}_{\textit{forced response}}$$

This predictions equation has two parts:

- Free response – the component that can't do anything about (left-most and right most terms).
- Forced response – the component that is determined by inputs to *j*-steps into the future (middle term).

Note that it is *x* hat since states are often not measurable and estimated using a Kalman Filter. This places restrictions on systems to which it can be applied i.e. they must be controllable and observable.

These predictions for steps *j = 1 , … , N+1* in the future are stacked together to form a single matrix equation as shown below.

$$\hat{Y}_{t,N} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix} \cdot (A - CG)\hat{x}(t|t) + \begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix} \cdot Gy(t)}_{F_{t,N}} + \underbrace{\begin{bmatrix} CB & & & \\ CAB & & 0 & \\ \vdots & & & \\ CA^NB & CA^{N-1}B & \cdots & CB \end{bmatrix}}_{S_N} \cdot U_{t,N} \quad (3:6)$$

$$\hat{Y}_{t,N} = F_{t,N} + S_N U_{t,N} \quad (3:7)$$

where

$$\hat{Y}_{t,N} = \begin{bmatrix} \hat{y}(t+1|t) \\ \vdots \\ \hat{y}(t+N+1|t) \end{bmatrix} and\ U_{t,N} = \begin{bmatrix} u(t) \\ \vdots \\ u(t+N) \end{bmatrix}$$

An optimal controller can then be calculated by substituting this matrix equation into the cost function, and solving to minimise *J(t)* as given by the following expression.

$$J_t = \sum_{n=t}^{t+N} \left[\big(y(n+1) - r(n+1)\big)^T \tilde{\Lambda}_e \big(y(n+1) - r(n+1)\big) + \big(u(n)\big)^T \tilde{\Lambda}_u \big(u(n)\big) \right] \quad (3:8)$$

and re-written using vector notation,

$$J_t = \big(\hat{Y}_{t,N} - R_{t,N}\big)^T \Lambda_e \big(\hat{Y}_{t,N} - R_{t,N}\big) + U_{t,N}^T \Lambda_u U_{t,N} \quad (3:9)$$

where Λ_e is the matrix of error weights and Λ_u the matrix of control weights. By setting its derivative to zero the *GPC* optimal future control is obtained as given below.

$$U_{t,N} = (S_N^T \Lambda_e S_N + \Lambda_u)^{-1} S_N^T \Lambda_e \big(R_{t,N} - F_{t,N}\big) \quad (3:10)$$

Observe that the first part of the expression (including the difficult inversion) up to $\big(R_{t,N} - F_{t,N}\big)$ is fixed i.e. S_N is fixed and $F_{t,N}$ is simply a fixed matrix multiplications of states x and y hence a fixed gain controller. Also remember that a Kalman Filter for doing state estimates is simply a fixed gain filter as well. In the following section *GPC* is used as a basis for the nonlinear predictive control scheme proposed here which utilises *LPV* systems.

3.3 Generalised Predictive Control (GPC) for LPV Systems

This formulation for the *NGPC* is based upon previous work in (Sznaier et al., 1998). Here however the decomposition of the plant that was suggested in (Grimble, 2005) is used. The linear part of the plant is used for predictions generation whereas the nonlinear part is used as a nonlinear compensation term. The starting point here is the *GPC* basic quadratic cost-function as shown below in discrete time.

$$J = E\{\sum_{i=0}^{N_p} e_p(t+i+k_0)^T e_p\,(t+i+k_0) + \lambda_i^2 u_0(t+i)^T u_0(t+i) \big| t\} \quad (3:11)$$

E is the conditional expectation on measurements up to time *t* and λ_i is a scalar control weighting. In this case the error signal e_p represents the weighted error signal. The optimisation problem is solved to yield the optimal controls vector within the prediction horizon N_p as discussed in the previous sections. Similar to what was mentioned in Chapter 2 basic state-space *NGMV* formulation the error weighting can be a dynamic weighting function instead (i.e. a transfer function or state-space model instead of a scalar weighting factor). Equation 3:11 can be now put into vector form as follows, including the k_0-steps ahead delay between an input and its effect on the output and hence the error.

$$J = E\{J_t\} = E\left\{E_{Pt+k_0,N}^T E_{Pt+k_0,N} + U_{t,N}^{0T} \Lambda_N^2 U_{t,N}^0 | t\right\} \tag{3:12}$$

where $E_{Pt+k_0,N}$ is the vector of future errors (i.e. within prediction horizon *N*) and $U_{t,N}^0$ the vector of future control actions. The next step is to substitute $E_{Pt+k_0,N} = \hat{E}_{Pt+k_0,N} + \tilde{E}_{Pt+k_0,N}$ (as it will be shown later in Chapter 4) where $\hat{E}_{Pt+k_0,N}$ is the vector of predicted weighted error signals and $\tilde{E}_{Pt+k_0,N}$ error estimates vector obtained by the *Kalman* filter (the detailed derivation for the estimates is found in Chapter 4). The *Kalman* filter is used as here it is assumed that the states are not available and hence need to be estimated.

$$J = E\left\{(\hat{E}_{Pt+k_0,N}\tilde{E}_{Pt+k_0,N})^T(\hat{E}_{Pt+k_0,N}\tilde{E}_{Pt+k_0,N}) + U_{t,N}^{0T}\Lambda_N^2 U_{t,N}^0 | t\right\} \tag{3:13}$$

The weighting factor *Λ* in the above equation is a diagonal matrix of the form,

$$\Lambda_N^2 = diag\{\lambda_0^2, \lambda_1^2, \dots, \lambda_N^2\}$$

Noting that $\hat{E}$ and $\tilde{E}$ are related by orthogonality then Equation 3:13 can be further simplified and written as follows,

$$J = \hat{E}_{Pt+k_0,N}^T \hat{E}_{Pt+k_0,N} + U_{t,N}^{0T}\Lambda_N^2 U_{t,N}^0 + J_0 \tag{3:14}$$

$$\text{where, } J_0 = E\left\{\tilde{E}_{Pt+k_0,N}^T \tilde{E}_{Pt+k_0,N} | t\right\} \tag{3:15}$$

The estimated error $\hat{E}_{Pt+k_0,N}^T$ can now be expanded as follows. This along with the predictions generation procedure is analytically summarised in the following section.

$$\begin{aligned}\hat{E}_{pt+k_0,N} &= D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t) + V_{pt+k_0,N}U_\delta U_{t,N}^0 \\ &= \widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta U_{t,N}^0\end{aligned} \tag{3:16}$$

where

$$\widetilde{D}_{pt+k_0,N} = D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t) \tag{3:17}$$

and

$$V_{pt+k_0,N} = C_{pt+k_0,N}B_{pt+k_0,N} + E_{Pt+k_0,N}$$

U_δ is a control signal shifting term that depends upon the presence of a delay in the loop and will be explained further in Chapter 4.

The cost-function can be now expanded as follows,

$$\begin{aligned} J &= \left(\widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta U^0_{t,N}\right)^T \left(\widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta U^0_{t,N}\right) + U^{0T}_{t,N}\Lambda^2_N U^0_{t,N} + J_0 \\ &= \widetilde{D}^T_{pt+k_0,N}\widetilde{D}_{pt+k_0,N}U^{0T}_{t,N}U^T_\delta V^T_{pt+k_0,N}\widetilde{D}_{pt+k_0,N} + \widetilde{D}^T_{pt+k_0,N}V_{pt+k_0,N}U_\delta U^0_{t,N} \\ &+ U^{0T}_{t,N}\left(U^T_\delta V^T_{pt+k_0,N}V_{pt+k_0,N}U_\delta + \Lambda^2_N\right)U^0_{t,N} + J_0 \end{aligned} \tag{3:18}$$

and the final form will be,

$$\begin{aligned} J &= \widetilde{D}^T_{pt+k_0,N}\widetilde{D}_{pt+k_0,N} + U^{0T}_{t,N}U^T_\delta V^T_{pt+k_0,N}\widetilde{D}_{pt+k_0,N} + \widetilde{D}^T_{pt+k_0,N}V_{pt+k_0,N}U_\delta U^0_{t,N} \\ &+ U^{0T}_{t,N}X_{t+k_0,N}U^0_{t,N} + J_0 \end{aligned} \tag{3:19}$$

where

$$X_{t+k_0,N} = U^T_\delta V^T_{pt+k_0,N}V_{pt+k_0,N}U_\delta + \Lambda^2_N \tag{3:20}$$

Similar to the procedure followed previously the gradient of the cost-function is set to zero in order to solve with respect to the optimal control vector $U^0_{t,N}$. Note that J_0 is independent of the control action.

Optimal Control: The optimal solution is shown in the following equation.

$$U^0_{t,N} = -X^{-1}_{t+k_0,N}U^T_\delta V^T_{pt+k_0,N}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) \tag{3:21}$$

The *receding horizon principle* is then implemented as discussed previously and only the first element form the optimal vector $U^0_{t,N}$ is applied to the plant.

Chapter 4 LPV-NPGMV Control Derivation

As with the *NGMV* controller being a model based control scheme, the structure of the plant plays a significant role in the formulation of the control algorithm and its performance. At this point the reader should be reminded that one of the main scopes of this work is the delivery of a practical and easily implementable control solution in the most generic form, which will be able to accommodate a large class of systems. In this chapter an extension to the *Nonlinear Predictive Generalised Minimum Variance Controller* is introduced, to make the algorithm compatible with *Linear Varying Parameter* systems (*LPV*). This approach is motivated by the intention to enhance the generality of the *NGMV* algorithm by making it applicable to a wider range of nonlinear systems that can be represented as *LPV*.

4.1 Controller and Subsystems Architecture

Following a similar sequence for the derivation, the structure of the overall controller and subsystems needs to be defined at an early stage. This will help understanding of its formulation and will reduce implementation time when adapting into different systems. At this point the reader should be reminded that one of the main scopes of this work is the delivery of a practical and easily implementable control solution, in the most generic form, which will be able to accommodate a large class of system structures.

The basic *NGMV* control structure is used as a common ground for this work and reviewed in Figure 4:1. The structure elements in red are components related to the plant model. The latter, as seen here, is the decomposition of the full nonlinear plant into a general nonlinear operator W_{1k} and an *LPV* approximation W_0. The nonlinear operator can be considered to include unmodelled nonlinearities, represented as input nonlinearities, whereas the *LPV* subsystem can be used to accommodate *LPV* approximations of parts of the plant that can be approximated and also disturbance and reference models, all put into an augmented *LPV* subsystem structure as seen in Equation (4:1. In this equation *A*, *B* and *C* denote the state-space matrices for the *LPV* plant subsystem (subscript: *0*), reference linear subsystem (subscript: *r*) and disturbance linear subsystem (subscript: *d*). An example of the later is seen in Chapter 6 where the augmented wind turbine *LPV* system is formulated.

$$\frac{d}{dt}\begin{bmatrix} x_0 \\ x_d \\ x_r \end{bmatrix} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_d & 0 \\ 0 & 0 & A_r \end{bmatrix}\begin{bmatrix} x_0 \\ x_d \\ x_r \end{bmatrix} + \begin{bmatrix} B_0 & 0 & 0 \\ 0 & B_d & 0 \\ 0 & 0 & B_r \end{bmatrix}\begin{bmatrix} u_0 \\ u_d \\ u_r \end{bmatrix}$$
$$y = \begin{bmatrix} C_0 & C_d & C_r \end{bmatrix}\begin{bmatrix} x_0 \\ x_d \\ x_r \end{bmatrix} \quad (4:1)$$

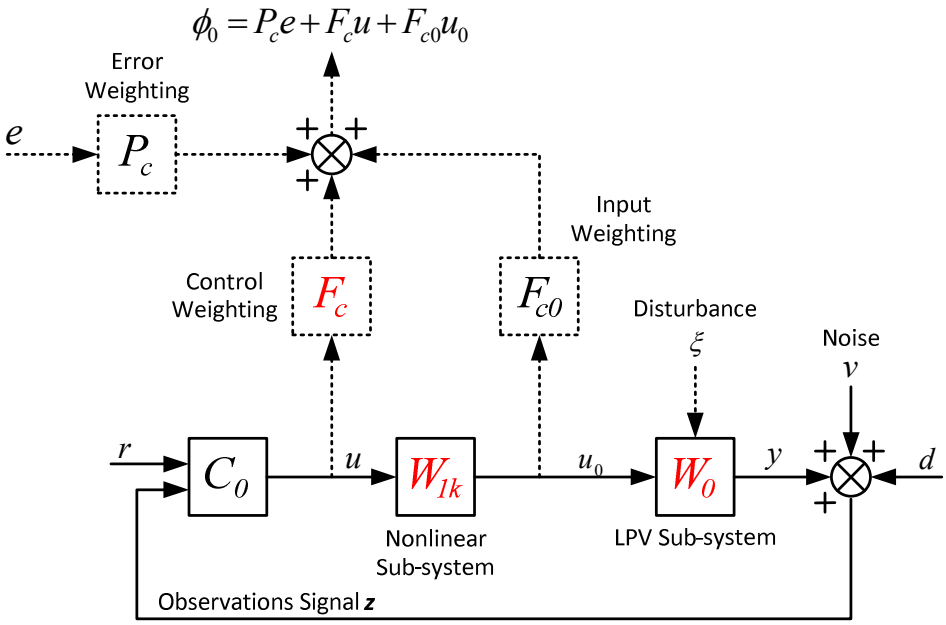


Figure 4:1 Feedback Control Structure the *NPGMV* formulation is based on.

The signals and subsystems in Figure 4:1 are explained in the following sections.

Input Signals:

r(t) : Vector of known reference or setpoint values.
d(t) : Vector of known output disturbance signal values.
v(t) : Vector of measurement noise values.

The reference, disturbance and noise signals can alternatively be implemented as the outputs of linear transfer functions driven by white noise. This is possible in the case the dynamical characteristics of the signals are known and it allows a more generic formulation which can accommodate both stochastic and deterministic components for these signals to improve the fidelity of the model. This is shown more clearly in the next section and in Figure 4:2. As in the previous chapter, here as well it is assumed that these signals will have zero mean and constant covariance matrices without loss of generality.

Control Signals:

u(t) : Vector of control signals applied to the nonlinear subsystem.
$u_0(t)$: Vector of control inputs to the *LPV* subsystem.

Output Signals:

y(t) : Vector of plant output signals.
z(t) : Vector of output measurement signals.

The measurements or observations signal results by adding measurement noise *v(t)* to the measured outputs of the plant; *z(t) = y(t) + v(t).*

Error Signal:

e(t) : Vector of tracking error signals.

This is the signal also used within the *NPGMV* controller cost-function and is the difference between the reference input and the measurements of the plant outputs to be controlled; *e(t)= r(t) – z(t).*

The subsystems in Figure 4:1 are explained in the section below.

Nonlinear Input Sub-System:

This sub-system is described by the following notation,

$$(W_1 u)(t) = z^{-k}(W_{1k} u)(t) \tag{4:2}$$

and the output of W_{1k} is denoted as,

$$u_0(t) = (W_{1k} u)(t) \tag{4:3}$$

as described in Chapter 2.

Nonlinear Output Sub-System:

This sub-system is also nonlinear of an *LPV* form and is denoted as,

$$(W_0 u_0)(t) = \left(W_{0k} z^{-k} u_0\right)(t) \tag{4:4}$$

where W_{0k} is its delay-free notation. Unlike the assumption made in the previous chapter regarding the stability of the nonlinear operator (being open-loop stable), here it is possible that an open-loop unstable system is represented within the *LPV* plant sub-system. The decomposition of the overall plant into its components, including the various signals in more detail, is illustrated in the following figure. Here the disturbance terms are shown to consist of a stochastic $\xi(t)$ and a deterministic component $d_{det}(t)$ influencing the state of the system (process disturbance) and a deterministic component $d_{meas}(t)$ influencing the output of the system (load disturbance). The weighted system output $y_p(t)$ is also shown as it is used in the formulation of the weighted error to be penalised within the controller cost-function as shown in Figure 4:1.

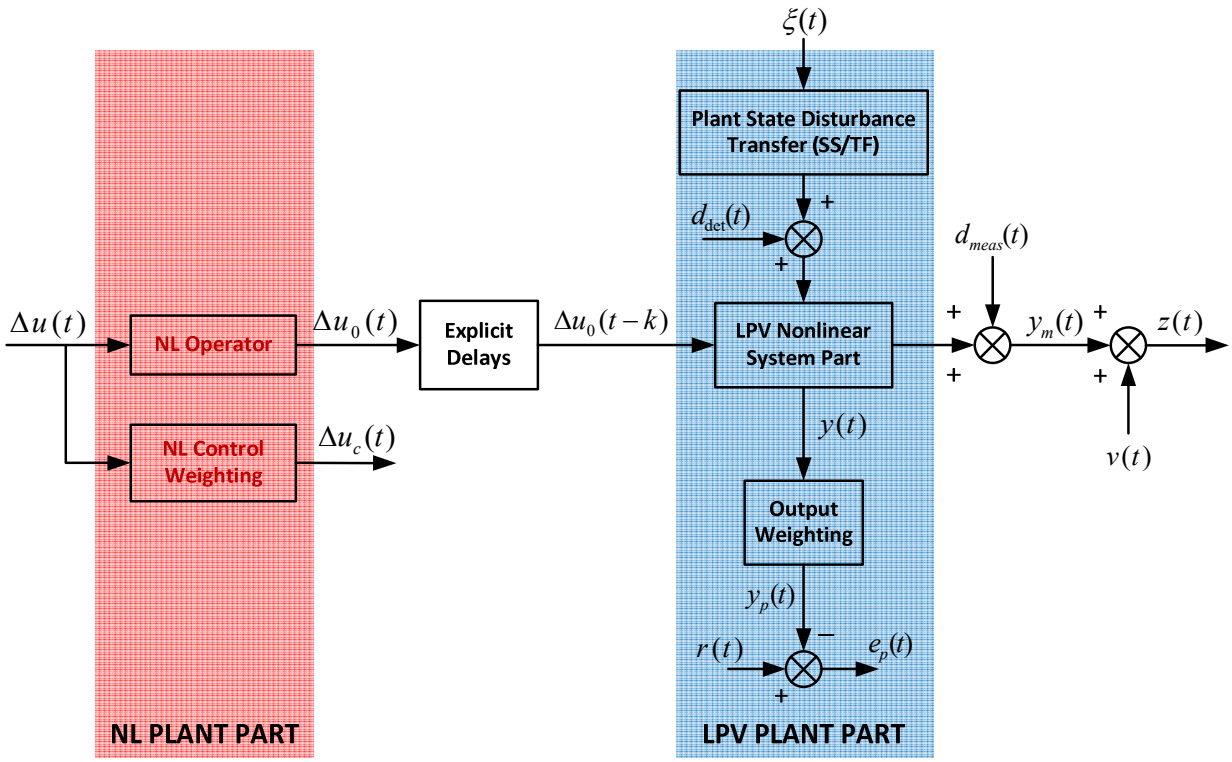


Figure 4:2 Expansion of the plant model, internal to the *LPV-NPGMV* controller, into its nonlinear/LPV and disturbance components.

Nonlinear Output Sub-System (LPV Expansion):

The *LPV* output sub-system is assumed to have the general structure as shown below.

$$x_0(t+1) = A_0(\rho)x_0(t) + B_0(\rho)u_0(t-k) + D_0(\rho)\xi(t) + d_d(t) \quad (4:5)$$

$$y(t) = C_0(\rho)x_0(t) + E_0(\rho)u_0(t-k) + d(t) \quad (4:6)$$

$$z(t) = C_{0m}(\rho)x_0(t) + E_{0m}(\rho)u_0(t-k) + d(t) + v(t) \quad (4:7)$$

d_d and d are used here to denote deterministic disturbance components added to the states and the output of the *LPV* subsystem respectively and z the output measurement (added noise component v). These both stochastic and deterministic disturbance components are used in this part of the formulation to demonstrate the flexibility in model structure used within the control algorithm. The parameter ρ can vary with time and the input of the system (i.e. $\rho\big(t, u_0(t-k)\big)$). We can slightly simplify notation by replacing the state matrices in the following manner.

$$A_t = A_0\left(\rho\big(t, u_0(t-k)\big)\right) \quad (4:8)$$

Equations 4:5 and 4:7 then become,

$$x_0(t+1) = A_t x_0(t) + B_t u_0(t-k) + D_t \xi(t) + d_d(t) \quad (4:9)$$

$$y(t) = C_t x_0(t) + E_t u_0(t-k) + d(t) \quad (4:10)$$

$$z(t) = C_{t_m} x_0(t) + E_{t_m} u_0(t-k) + d(t) + v(t) \quad (4:11)$$

The various components in these equations can be clearly seen by expanding the systems in Figure 4:2 a step further as shown in Figure 4:3. The subscript *t* indicates that the matrices will now vary with time as the parameter they are dependent upon is changing.

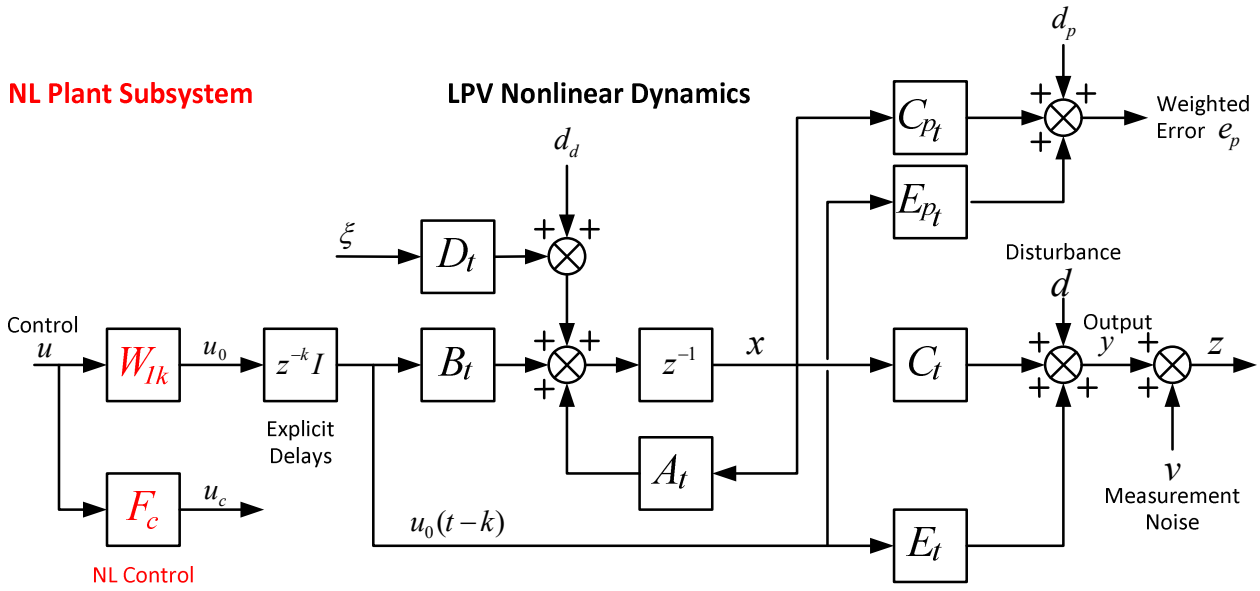


Figure 4:3 Generalised *LPV* Subsystems expansion

The weighted error e_p equation is shown below.

$$e_p(t) = d_p(t) + C_{pt} x(t) + E_{pt} u_0(t-k) \quad (4:12)$$

d_p again here is a deterministic disturbance component added to the error weighting dynamic subsystem to enhance generality in the model. Note in Figure 4:3 that disturbances are broken down into their stochastic and deterministic components. Similar to the derivation that was explained in Chapter 2 each of the reference, disturbance and error weighting subsystems illustrated in the figure can be modelled individually in a state-space manner to compliment the *LPV* nonlinear dynamics. The only difference in this section is the plant linear subsystem

equivalent being replaced by an *LPV* formulation. The derivation in which importance is given here is the augmented model of the later sub-system.

Augmented System Derivation:

The overall sub-system in state-space form will be a multivariable $n \times m$ system (i.e. *n* number of outputs and *m* number of inputs) that consists of the plant *LPV* dynamics, the disturbance and weighted error state-space models, integrated into a complete augmented representation. This model will be a function of control and the varying parameters (considered in the *LPV* formulation $(u_0(t-k), \rho(t))$. The new state vector will be,

$$x = \begin{bmatrix} x_0 \\ x_d \\ x_p \end{bmatrix} \tag{4:13}$$

where x_0, x_d and x_p are the state vectors for the *LPV*, disturbance and error weighting subsystems respectively. For simplicity from this point onwards the *LPV* sub-system matrices will be denoted as shown in Equations 2.15 - 2:17 (A_0, B_0, C_0 etc.). The augmented system matrices are shown below.

$$A = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_d & 0 \\ -B_p C_0 & -B_p C_0 & A_p \end{bmatrix}, B = \begin{bmatrix} B_0 \\ 0 \\ -B_p E_0 \end{bmatrix}, D = \begin{bmatrix} D_0 & 0 \\ 0 & D_d \\ 0 & 0 \end{bmatrix} \tag{4:14}$$

$$R = \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & B_p \end{bmatrix}, C = [C_0 \quad C_d \quad 0], E = E_0 \tag{4:15}$$

$$C_p = [-E_p C_0 \quad -E_p C_d \quad C_p], E_p = -E_p E_0 \tag{4:16}$$

Note here that the reference *r(t)* and disturbance *d(t)* signals are deterministic signals and for the purpose of this formulation can be defined as follows,

$$d_d = \begin{bmatrix} d_{0d} \\ (r-d) \end{bmatrix}, d_p = E_p(r-d) \tag{4:17}$$

Moreover the combined noise vector can be defined as follows,

$$v = \begin{bmatrix} \xi \\ \omega \end{bmatrix} \tag{4:18}$$

The augmented system state equations can be defined by using Equations 4:8 and 4:18 as follows,

$$\begin{bmatrix} x_0(t+1) \\ x_d(t+1) \\ x_p(t+1) \end{bmatrix} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_d & 0 \\ -B_pC_0 & -B_pC_0 & A_p \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_d(t) \\ x_p(t) \end{bmatrix} + \begin{bmatrix} B_0 \\ 0 \\ -B_pE_0 \end{bmatrix} u_0(t-k)$$
$$+ \begin{bmatrix} D_0 & 0 \\ 0 & D_d \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & B_p \end{bmatrix} \begin{bmatrix} d_{0d}(t) \\ (r(t) - d(t)) \end{bmatrix} \quad (4:19)$$

Similarly the error equation can be defined as follows,

$$e_p(t) = [-E_pC_0 \quad -E_pC_d \quad C_p] \begin{bmatrix} x_0(t) \\ x_d(t) \\ x_p(t) \end{bmatrix} - E_pE_0u_0(t-k) + E_p\big(r(t) - d(t)\big)$$
$$= d_p(t) + [-E_pC_0 \quad -E_pC_d \quad C_p] \begin{bmatrix} x_0(t) \\ x_d(t) \\ x_p(t) \end{bmatrix} - E_pE_0u_0(t-k) \quad (4:20)$$

Note here that the output equation is omitted as the system output is ustilised within the controller via the error weighting subsystem.

4.2 Derivation of Predictions Model for Control

An *LPV* model prediction equation is required. The future values of the states and outputs, at times *t+1*, *t+2,...* can be obtained by using the state-equation iteratively as follows:

$$x(t+1) = A_t x(t) + B_t u_0(t-k) + D_t \xi(t) + d_d(t) \quad (4:21)$$

$$\begin{aligned} x(t+2) &= A_{t+1}\big(A_t x(t) + B_t u_0(t-k) + D_t \xi(t) + d_d(t)\big) + B_{t+1} u_0(t-k+1) \\ &\quad + D_{t+1}\xi(t+1) + d_d(t+1) \\ &= A_{t+1}A_t x(t) + A_{t+1}B_t u_0(t-k) + B_{t+1}u_0(t-k+1) \\ &\quad + A_{t+1}D_t\xi(t) + D_{t+1}\xi(t+1) + A_{t+1}d_d(t) + d_d(t+1) \end{aligned} \quad (4:22)$$

$$
\begin{aligned}
x(t+3) = {} & A_{t+2}A_{t+1}\big(A_t x(t) + B_t u_0(t-k) + D_t\xi(t) + d_d(t)\big) \\
& + A_{t+2}B_{t+1}u_0(t-k+1) + B_{t+2}u_0(t-k+2) \\
& + A_{t+2}D_{t+1}\xi(t+1) + D_{t+2}\xi(t+2) + A_{t+2}d_d(t+1) + d_d(t+2) \\
& = A_{t+2}A_{t+1}A_t x(t) + A_{t+2}A_{t+1}B_t u_0(t-k) \\
& + A_{t+2}B_{t+1}u_0(t-k+1) + B_{t+2}u_0(t-k+2) + A_{t+2}A_{t+1}D_t\xi(t) \\
& + A_{t+2}D_{t+1}\xi(t+1) + D_{t+2}\xi(t+2) + A_{t+2}A_{t+1}d_d(t) \\
& + A_{t+2}d_d(t+1) + d_d(t+2)
\end{aligned}
\tag{4:23}
$$

Generalising this result for $i \geq 1$, the state, at any future time $t+i$:

$$
\begin{aligned}
x(t+i) = {} & A_{t+i-1}A_{t+i-2}\dots A_t x(t) + A_{t+i-1}A_{t+i-2}\dots A_{t+1}B_t u_0(t-k) \\
& + A_{t+i-1}A_{t+i-2}\dots A_{t+2}B_{t+1}u_0(t-k+1) + \cdots \\
& + A_{t+i-1}B_{t+i-2}u_0(t+i-k+2) + B_{t+i-1}u_0(t+i-k+1) \\
& + A_{t+i-1}A_{t+i-2}\dots A_{t+1}D_t\xi(t) + \cdots + A_{t+i-1}D_{t+i-2}\xi(t+i-2) \\
& + D_{t+i-1}\xi(t+i-1) + A_{t+i-1}A_{t+i-2}\dots A_{t+1}d_d(t) \\
& + A_{t+i-1}A_{t+i-2}\dots A_{t+2}d_d(t+1) + A_{t+i-1}d_d(t+i-2) \\
& + d_d(t+i-1)
\end{aligned}
\tag{4:24}
$$

This future states equation may be written in more general form by introducing the notation:

$$
A_{t+m}^{i-m} = A_{t+i-1}A_{t+i-2}\dots A_{t+m} \tag{4:25}
$$

where $\mathrm{A}_{\mathrm{t+m}}^{0} = \mathrm{I}$ and

$$
A_t^i = A_{t+i-1}A_{t+i-2}\dots A_t \tag{4:26}
$$

where $\mathrm{A}_{\mathrm{t}}^{0} = \mathrm{I}$.

Thence, write:

$$
\begin{aligned}
x(t+i) = {} & A_{t+i-1}A_{t+i-2}\dots A_t x(t) \\
& + \sum_{j=1}^{i} A_{t+i-1}A_{t+i-2}\dots A_{t+j}\Big(B_{t+j-1}u_0(t+j-1-k) \\
& + D_{t+j-1}\xi(t+j-1) + d_d(t+j-1)\Big)
\end{aligned}
\tag{4:27}
$$

This may be written using the above notation as:

$$x(t+i) = A_t^i(t) + \sum_{j=1}^{i} A_{t+j}^{i-j}\Big(B_{t+j-1}u_0(t+j-1-k) + D_{t+j-1}\xi(t+j-1) + d_d(t+j-1)\Big) \quad (4:28)$$

The weighted error or output signal $e_p(t)$ to be regulated at future times, for the augmented system, has the form (for $i \geq 1$):

$$e_p(t+i) = d_d(t+i) + C_{pt+i}x(t+i) + E_{pt+i}u_0(t+i-k) \quad (4:29)$$

The *i-steps* prediction for the state and the output signals can be summarised in the following manner,

$$\hat{x}(t+i|t) = A_t^i\hat{x}(t|t) + \sum_{j=1}^{i} A_{t+j}^{i-j}\Big(B_{t+j-1}u_0(t+j-1-k) + d_d(t+j-1)\Big) \quad (4:30)$$

$$\hat{y}(t+i|t) = d(t+i) + C(t+i)\hat{x}(t+i|t) + E(t+i)u_0(t+i-k) \quad (4:31)$$

Similarly the estimated weighted error equation is as follows. This is the signal to be regulated at future times ($i \geq 1$),

$$\hat{e}_p(t+i|t) = d_p(t+i) + C_{pt+i}(t+i)\hat{x}(t+i|t) + E_{pt+i}(t+i)u_0(t+i-k) \quad (4:32)$$

Transport Delay Integration:

The *E* matrix denotes the feed-through (instantaneous transmission path) term (usually denoted by the *D* matrix). As it was shown in the previous chapter there are two cases for *E*.

- $E, E_{pt} \neq 0$; *E* is non-zero under the assumption that the common delay terms are possible to extract from the plant and put into the explicit delays block $z^{-k}I$ (normal case). In this case this should also hold for the error weighting block which in turn should have a non-zero feed-through term E_{pt}.
- $E = E_{pt} = 0$; not possible to obtain a model with a non-zero *E* and a delay of at least 1 step. In this case, to counterbalance the absence of delay in the model and obtain a more suitable cost-function, the *k* transport delay term in the controller is set to at least 1 step.

This leads to the following definitions for the *k* transport delay term when formulating the control problem if it also assumed that the plant has a transport delay of k_0.

- For $E, E_{pt} \neq 0$, then $k_0 = k$.
- For $E = E_{pt} = 0$, then $k_0 = k + 1$.

Incorporating the above transport delay definitions into the predictions Equations 4:31 and 4:32, yields the following expressions.

State Predictions Equation (Considering Effective Time Delays):

For $i = k_0$,

$$\hat{x}(t + k_0|t) = A_t^{k_0}\hat{x}(t|t) + \sum_{j=1}^{k_0} A_{t+j}^{k_0-j}\left(B_{t+j-1}u_0(t + j - 1 - k) + d_d(t + j - 1)\right) \quad (4:33)$$

Using the finite pulse response operator that was introduced in the previous chapter (Equation 2:22) the k_0 steps prediction can be reformulated as follows,

$$\hat{x}(t + k_0|t) = A_t^{k_0}\hat{x}(t|t) + T(k_0, z^{-1})u_0(t) + d_{dd}(t + k_0 - 1) \quad (4:34)$$

with the *T* operator defined as follows,

$$T(k_0, z^{-1}) = \sum_{j=1}^{k_0} A_{t+j}^{k_0-j}B_{t+j-1}z^{j-k-1} \quad (4:35)$$

and

$$d_{dd}(t + k_0 - 1) = \sum_{j=1}^{k_0} A_{t+j}^{k_0-j}d_d(t + j - 1) \quad (4:36)$$

For $i + k_0$ and $i > 1$,

$$\begin{aligned}\hat{x}(t + i + k_0|t) = {} & A_{t+k_0}^{i}\hat{x}(t + k_0|t) \\ & + \sum_{j=1}^{i} A_{t+j}^{i-j}\left(B_{t+k_0+j-1}u_0(t + j - 1 + k_0 - k) + d_d(t + j + k_0 - 1)\right)\end{aligned} \quad (4:37)$$

Error Predictions Equation (Considering Effective Time Delays):

The error prediction equation can be similarly derived from Equation 4:32 as follows,

$$\begin{aligned}\hat{e}_p(t+i+k_0|t) &= d_p(t+i+k_0) + E_{pt+i+k_0}u_0(t+i+k_0-k) \\ &+ C_{pt+i+k_0}\hat{x}(t+i+k_0|t)\end{aligned} \tag{4:38}$$

Substituting Equation 4:34 (predicted state) into Equation 4:38 and also introducing $\delta = k_0 - k$ yields the following,

$$\begin{aligned}\hat{e}_p(t+i+k_0|t) &= d_p(t+i+k_0) + E_{pt+i+k_0}u_0(t+i+\delta) \\ &+ C_{pt+i+k_0}A^i_{t+k_0}\hat{x}(t+k_0|t) \\ &+ \sum_{j=1}^{i} C_{pt+i+k_0}A^{i-j}_{t+k_0+j}\Big(B_{t+k_0+j-1}u_0(t+j-1+\delta) \\ &+ d_d(t+j+k_0-1)\Big)\end{aligned} \tag{4:39}$$

To further simplify this equation the deterministic terms can be combined in the following definition,

$$d_{pd}(t+i+k_0) = d_p(t+i+k_0) + \sum_{j=1}^{i} C_{pt+i+k_0}A^{i-j}_{t+k_0+j}d_d(t+j+k_0-1) \tag{4:40}$$

The final error predictions equation will then have the following form,

$$\begin{aligned}\hat{e}_p(t+i+k_0|t) &= d_{pd}(t+i+k_0) + E_{pt+i+k_0}u_0(t+i+\delta) \\ &+ C_{pt+i+k_0}A^i_{t+k_0}\hat{x}(t+k_0|t) \\ &+ \sum_{j=1}^{i} C_{pt+i+k_0}A^{i-j}_{t+k_0+j}B_{t+k_0+j-1}u_0(t+j-1+\delta)\end{aligned} \tag{4:41}$$

4.2.1 Vector-Matrix Notation Derivation

The error and output predictions derived in the previous section can be now put into vector form with respect to the control interval $\tau \in [t, t+N], N > 0$ as follows.

$$\begin{bmatrix} \hat{e}_p(t+k_0) \\ \hat{e}_p(t+1+k_0) \\ \hat{e}_p(t+2+k_0) \\ \vdots \\ \hat{e}_p(t+N+k_0) \end{bmatrix}$$

$$= \begin{bmatrix} d_{pd}(t+k_0) \\ d_{pd}(t+k_0+1) \\ d_{pd}(t+k_0+2) \\ \vdots \\ d_{pd}(t+k_0+N) \end{bmatrix} + \begin{bmatrix} C_{pt+k_0} I \\ C_{pt+k_0+1} A^1_{t+k_0} \\ C_{pt+k_0+2} A^2_{t+k_0} \\ \vdots \\ C_{pt+k_0+N} A^N_{t+k_0} \end{bmatrix} \hat{x}(t+k_0|t) + \begin{bmatrix} E_{pt+k_0} u_0(t+\delta) \\ E_{pt+k_0+1} u_0(t+\delta+1) \\ E_{pt+k_0+2} u_0(t+\delta+2) \\ \vdots \\ E_{pt+k_0+N} u_0(t+\delta+N) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ C_{pt+k_0+1} B_{t+k_0} & 0 & \ddots & 0 & 0 \\ C_{pt+k_0+2} A^1_{t+k_0+1} B_{t+k_0} & C_{pt+k_0+2} B_{t+k_0+1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ C_{pt+k_0+N} A^{N-1}_{t+k_0+1} B_{t+k_0} & C_{pt+k_0+N} A^{N-2}_{t+k_0+2} B_{t+k_0+1} & \cdots & C_{pt+k_0+N} B_{t+k_0+N-1} & 0 \end{bmatrix} \cdot \begin{bmatrix} u_0(t+\delta) \\ u_0(t+\delta+1) \\ u_0(t+\delta+2) \\ \vdots \\ u_0(t+\delta+N) \end{bmatrix} \quad (4:42)$$

Like in the previous section if transport delays are considered in the system then two cases arise. The controls vector will then be re-formulated accordingly as follows using the delay shift vector U_δ.

***δ=0* (Feed–Through Term/ No Extra Delay):**

$$\text{For } U_\delta = I, \; U_\delta U^0_{t,N} = \begin{bmatrix} u_0(t) \\ u_0(t+1) \\ u_0(t+2) \\ \vdots \\ u_0(t+N) \end{bmatrix} \quad (4:43)$$

***δ=1* (Zero Feed–Through Term/ Extra Delay):**

For,

$$U_\delta = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}, \tag{4:44}$$

$$U_\delta U_{t,N}^0 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} u_0(t) \\ u_0(t+1) \\ u_0(t+2) \\ \vdots \\ u_0(t+N) \end{bmatrix} = \begin{bmatrix} u_0(t+1) \\ u_0(t+2) \\ \vdots \\ u_0(t+N) \\ 0 \end{bmatrix}$$

Predicted Errors:

Using the above vector notation predictions Equation 4:41 can be re-formulated as follows.

$$\begin{aligned} \hat{E}_{pt+k_0,N} = D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} \hat{x}(t+k_0|t) \\ + \left(C_{pt+k_0,N} B_{t+k_0,N} + E_{pt+k_0,N}\right) U_\delta U_{t,N}^0 \end{aligned} \tag{4:45}$$

Further simplification can be achieved via introducing the following time-varying matrix.

$$V_{pt+k_0,N} = C_{pt+k_0,N} B_{t+k_0,N} + E_{pt+k_0,N} \tag{4:46}$$

The predicted errors equation then takes the following form.

$$\hat{E}_{pt+k_0,N} = D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} \hat{x}(t+k_0|t) + V_{pt+k_0,N} U_\delta U_{t,N}^0 \tag{4:47}$$

Errors to be minimised at Future Times:

Adding the stochastic disturbance inputs $\Xi_{t+k,N}$ that influence future weighted errors,

$$\begin{aligned} E_{pt+k_0,N} = D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} x(t+k_0|t) + V_{pt+k_0,N} U_\delta U_{t,N}^0 \\ + C_{pt+k_0,N} D_{t+k_0,N} \Xi_{t+k,N} \end{aligned} \tag{4:48}$$

Estimation Error:

The estimation error is now formulated as the difference between the future and estimated (predicted) errors derived previously.

$$\tilde{E}_{pt+k_0,N} = E_{pt+k_0,N} - \hat{E}_{pt+k_0,N} \tag{4:49}$$

$$\begin{aligned}\tilde{E}_{pt+k_0,N} = {} & D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}x(t+k_0|t) + V_{pt+k_0,N}U_\delta U^0_{t,N} \\ & + C_{pt+k_0,N}D_{t+k_0,N}\Xi_{t+k,N} \\ & - \left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t) + V_{pt+k_0,N}U_\delta U^0_{t,N}\right)\end{aligned} \tag{4:50}$$

$$\tilde{E}_{pt+k_0,N} = C_{pt+k_0,N}A_{t+k_0,N}\tilde{x}(t+k_0|t) + C_{pt+k_0,N}D_{t+k_0,N}\Xi_{t+k,N} \tag{4:51}$$

with $\tilde{x}(t+k_0|t)$ being the k_0 steps ahead state estimation error.

Predicted System Matrices:

Similar to the previous sections, the future matrices that constitute Equations 4:43-4:51 are defined as follows,

$$A_{t+k_0,N} = \begin{bmatrix} I \\ A^I_{t+k_0} \\ A^2_{t+k_0} \\ \vdots \\ A^N_{t+k_0} \end{bmatrix},$$

$$B_{t+k_0,N} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ B_{t+k_0} & 0 & \cdots & \vdots & 0 \\ A^I_{t+k_0+1}B_{t+k_0} & B_{t+k_0+1} & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ A^{N-1}_{t+k_0+1}B_{t+k_0} & A^{N-2}_{t+k_0+2}B_{t+k_0+1} & \cdots & B_{t+k_0+N-1} & 0 \end{bmatrix},$$

$$D_{t+k_0,N} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ D_{t+k_0} & 0 & \cdots & \vdots & 0 \\ A^I_{t+k_0+1}D_{t+k_0} & D_{t+k_0+1} & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ A^{N-1}_{t+k_0+1}D_{t+k_0} & A^{N-2}_{t+k_0+2}D_{t+k_0+1} & \cdots & D_{t+k_0+N-1} & 0 \end{bmatrix}, \tag{4:52}$$

$$\Xi_{t,N} = \begin{bmatrix} \xi(t) \\ \xi(t+1) \\ \vdots \\ \xi(t+N-1) \end{bmatrix}$$

4.3 LPV Estimation

To obtain the *k*-steps ahead state estimate $\hat{x}(t+k|t)$, which is used as a starting point within the output and error predictions equations, the *Time-Varying Kalman Filter* (*TVKF*) is employed. For the purpose of this work, the latter suitably fits the *LPV* paradigm. This holds as by using the *LPV* time-varying matrices A_t, B_t, C_t, E_t, results into a time-varying error covariance matrix P_t and hence a *TVKF* gain factor K_{ft} derivation. Similar to most *KF* derivations, the starting point here will be taking the state predictions equation and complimenting with the weighted output prediction error to obtain a good state estimate. This is demonstrated in the following equations.

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_{ft}\big(z(t+1) - \hat{z}(t+1|t)\big) \tag{4:52}$$

where z are the output measurements and $\hat{z}$ the output prediction and K_{ft} is the *Time-Varying Kalman Filter* gain. Here it is worth noting that in the case where a good model is available for the full nonlinear plant, then this can be used to obtain $\hat{z}(t+1|t)$ and $\hat{x}(t+1|t)$ resulting into the *Extended Kalman Filter* (*EKF*). This formulation was actually possible to implement in application Chapter 6. However for the purpose of this formulation $\hat{z}$ is derived by the following expression.

$$\hat{z}(t+1|t) = d(t+1) + C_{t+1}\hat{x}(t+1|t) + E_{t+1}u_0(t+1-k) \tag{4:53}$$

Here *d* denotes a deterministic disturbance component added to the output of the system (e.g. a step disturbance or bias). Notice that the above equation is also adjusted to include delay and through terms. Similar to the derivation in the previous chapter the *k*-steps filter equation can be put into *predictor-estimator* form as follows,

$$\hat{x}(t+k_0|t) = A_t^{k_0}\hat{x}(t|t) + T(k_0, z^{-1})u_0(t) + d_d(t+k_0-1) \tag{4:54}$$

The optimal estimate can be then put in the following form,

$$\hat{x}(t+1|t) = A_t\hat{x}(t|t) + B_t u_0(t-k) + d_d(t) \tag{4:55}$$

$$\hat{x}(t+1|t+1) = A_t\hat{x}(t|t) + B_t u_0(t-k) + d_d(t) + K_{ft}\big(z(t+1) - \hat{z}(t+1|t)\big) \tag{4:56}$$

$$\begin{aligned}(zI - A_t)\hat{x}(t|t) = {} & B_t u_0(t-k) + d_d(t) \\ & + K_{ft}\big(z(t+1) - (d(t+1) + C_{t+1}\hat{x}(t+1|t) + E_{t+1}u_0(t+1-k))\big)\end{aligned} \tag{4:57}$$

Expanding further yields the following equation for the optimal estimate,

$$\begin{aligned}\hat{x}(t|t) = \left(I - z^{-1}\left(I - K_{ft}C_{t+1}\right)A_t\right)^{-1} z^{-1}\big[&B_t u_0(t-k) + d_d(t) \\ &- K_{ft}(C_{t+1}B_t u_0(t-k) + E_{t+1}u_0(t+1-k)) \\ &+ K_{ft}\left(z(t+1) - d(t+1)\right)\big]\end{aligned} \tag{4:58}$$

In the following section the adaptation of predictive control to *LPV* systems is discussed as a preliminary step towards the *LPV-NPGMV* derivation.

4.4 Equivalent Cost-Function Optimisation Problem

In this section another intermediate step is taken that will naturally lead into the *NPGMV* formulation later on. This time the equivalence between the *LPV-GPC* and the *GMV* modified cost-function that will be discussed in the subsequent section is explained. The starting point here will be the factorisation of the positive-definite, real symmetric matrix $X_{t+k_0,N}$ as follows,

$$X_{t+k_0,N} = Y_{t+k_0,N}^T Y_{t+k_0,N} = U_\delta^T V_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta + \Lambda_N^2 \tag{4:59}$$

By completing the squares in Equation 3:19, the cost-function is written as follows,

$$\begin{aligned}J &= \left(\widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta U_{t,N}^0\right)^T \left(\widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta U_{t,N}^0\right) + U_{t,N}^{0T}\Lambda_N^2 U_\delta^T U_{t,N}^0 + J_0 \\ &= \widetilde{D}_{pt+k_0,N}^T \widetilde{D}_{pt+k_0,N} + U_{t,N}^{0T} U_\delta^T V_{pt+k_0,N}^T \widetilde{D}_{pt+k_0,N} \\ &+ \widetilde{D}_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta U_{t,N}^0 + U_{t,N}^{0T}\left(U_\delta^T V_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta + \Lambda_N^2\right) U_{t,N}^0 \\ &+ J_0\end{aligned} \tag{4:60}$$

The final form of the cost-function will be as follows,

$$\begin{aligned}J = \left(\widetilde{D}_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta Y_{t+k_0,N}^{-1} + U_{t,N}^{0T} Y_{t+k_0,N}^T\right)&\left(Y_{t+k_0,N}^{-T} U_\delta^T V_{pt+k_0,N}^T \widetilde{D}_{pt+k_0,N}\right. \\ &\left. + Y_{t+k_0,N} U_{t,N}^0\right) \\ &+ \widetilde{D}_{pt+k_0,N}^T \left(I - V_{pt+k_0,N} U_\delta Y_{t+k_0,N}^{-1} Y_{t+k_0,N}^{-T} U_\delta^T V_{pt+k_0,N}^T\right) \widetilde{D}_{pt+k_0,N} + J_0\end{aligned} \tag{4:61}$$

Comparing the two forms the cost-function can be put in the following equivalent form,

$$J = \Phi_{t+k_0,N}^{0T} \Phi_{t+k_0,N}^0 + J_{10}(t) \tag{4:62}$$

where

$$\widehat{\Psi}^{0}_{t+k_0,N} = Y^{-T}_{t+k_0,N} U^{T}_{\delta} V^{T}_{pt+k_0,N} \left(D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} \hat{x}(t+k_0|t) \right) + Y_{t+k_0,N} U^{0}_{t,N} \tag{4:63}$$

$$\text{and,} \quad J_{10}(t) = J_0 + J_1(t)$$

where

$$J_1(t) = \widetilde{D}^{T}_{pt+k_0,N} \left(I - V_{pt+k_0,N} U_{\delta} Y^{-1}_{t+k_0,N} Y^{-T}_{t+k_0,N} U^{T}_{\delta} V^{T}_{pt+k_0,N} \right) \widetilde{D}_{pt+k_0,N} \tag{4:64}$$

Given that the $J_{10}(t)$ term does not depend upon control action, it can be excluded when setting the gradient of the cost-function to zero to find the optimal solution. Considering this, the following optimal controls vector results out of the minimisation of Equation 4:64.

$$U^{0}_{t,N} = -\left(U^{T}_{\delta} V^{T}_{pt+k_0,N} V_{pt+k_0,N} U_{\delta} + \Lambda^{2}_{N} \right)^{-1} U^{T}_{\delta} V^{T}_{pt+k_0,N} \left(D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} \hat{x}(t+k_0|t) \right) \tag{4:65}$$

4.4.1 Modified GMV Cost-Function

Combining the definitions in Sections 4.2.1 and 2.1 (*GMV* definition), the *GMV* problem can be now modified to include future steps in the cost-function. Recall the *GMV* optimal problem defined as the weighted sum of error and input signals shown below.

$$\Phi(t) = P_c e(t) + F_{c0} u(t) \tag{4:66}$$

Employing a similar strategy as with the use of predictions time-varying vectors in the *LPV-GPC* cost-function, Equation 4:66 can be written as follows,

$$\Phi_{t,N} = P_{cN,t} E_{pt,N} + F^{0}_{cN,t} U^{0}_{t,N} \tag{4:67}$$

where the cost-function weightings are constant matrices of the following form,

$$P_{cN,t} = U^{T}_{\delta} V^{T}_{pt+k_0,N} \text{ and } F^{0}_{cN,t} = \Lambda^{2}_{N} \tag{4:68}$$

A multi-step cost-function can then be defined for the GMV problem as follows,

$$J = E\{J_t\} = E\left\{ \Phi^{T}_{t+k_0,N} \Phi_{t+k_0,N} | t \right\} \tag{4:69}$$

Φ for k_0-steps ahead is shown below.

$$\Phi_{t+k_0,N} = P_{cN,t} E_{pt+k_0,N} + F^{0}_{cN,t} U^{0}_{t,N} \tag{4:70}$$

Similar to the predicted errors expansion used previously, $E_{pt+k_0,N} = \hat{E}_{pt+k_0,N} + \tilde{E}_{pt+k_0,N}$. Substituting this to Equation 4:70 the following is obtained,

$$\begin{aligned} \Phi_{t+k_0,N} &= P_{cN,t}\left(\hat{E}_{pt+k_0,N} + \tilde{E}_{pt+k_0,N}\right) + F^0_{cN,t}U^0_{t,N} \\ &= \left(P_{cN,t}\hat{E}_{pt+k_0,N} + F^0_{cN,t}U^0_{t,N}\right) + P_{cN,t}\tilde{E}_{pt+k_0,N} \end{aligned} \tag{4:71}$$

This expression can be broken down in terms of the estimate and estimation error vectors as follows,

$$\Phi_{t+k_0,N} = \hat{\Phi}_{t+k_0,N} + \tilde{\Phi}_{t+k_0,N} \tag{4:72}$$

where $\hat{\Phi}_{t+k_0,N}$ is the predicted signal and $\tilde{\Phi}_{t+k_0,N}$ the estimation error as shown below.

$$\hat{\Phi}_{t+k_0,N} = \left(P_{cN,t}\hat{E}_{pt+k_0,N} + F^0_{cN,t}U^0_{t,N}\right) \tag{4:73}$$

$$\tilde{\Phi}_{t+k_0,N} = P_{cN,t}\tilde{E}_{pt+k_0,N} \tag{4:74}$$

The multi-step *GMV* cost-function will then have the following form,

$$J = E\{J_t\} = E\left\{\Phi^T_{t+k_0,N}\Phi_{t+k_0,N}|t\right\} = E\left\{\left(\hat{\Phi}_{t+k_0,N} + \tilde{\Phi}_{t+k_0,N}\right)^T\left(\hat{\Phi}_{t+k_0,N} + \tilde{\Phi}_{t+k_0,N}\right)|t\right\} \tag{4:75}$$

Recalling orthogonality between $\hat{E}_{pt+k_0,N}$ and $\tilde{E}_{pt+k_0,N}$ the cost-function can be written as follows,

$$\begin{aligned} J = E\left\{\hat{\Phi}^T_{t+k_0,N}\hat{\Phi}_{t+k_0,N}|t\right\} + E\left\{\hat{\Phi}^T_{t+k_0,N}\tilde{\Phi}_{t+k_0,N}|t\right\} + E\left\{\tilde{\Phi}^T_{t+k_0,N}\hat{\Phi}_{t+k_0,N}|t\right\} \\ + E\left\{\tilde{\Phi}^T_{t+k_0,N}\tilde{\Phi}_{t+k_0,N}|t\right\} = \hat{\Phi}^T_{t+k_0,N}\hat{\Phi}_{t+k_0,N} + E\left\{\tilde{\Phi}^T_{t+k_0,N}\tilde{\Phi}_{t+k_0,N}|t\right\} \end{aligned} \tag{4:76}$$

and therefore,

$$J = \hat{\Phi}^T_{t+k_0,N}\hat{\Phi}_{t+k_0,N} + J_1(t) \tag{4:77}$$

As discussed previously the term J_1 is independent of control action and is defined as follows,

$$J_1 = E\left\{\tilde{\Phi}^T_{t+k_0,N}\tilde{\Phi}_{t+k_0,N}|t\right\} = E\left\{\tilde{E}^T_{pt+k_0,N}P^T_{cN,t}P_{cN,t}\tilde{E}_{pt+k_0,N}|t\right\} \tag{4:78}$$

By substituting $\hat{E}_{pt+k_0,N}$ from Equation 3:28 and further simplifying, $\hat{\Phi}_{t+k_0,N}$ can be written as follows,

$$\widehat{\Phi}_{t+k_0,N} = P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) + \left(U_\delta^T V_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta \ + \Lambda_N^2\right)U_{t,N}^0 \tag{4:71}$$

$$\widehat{\Phi}_{t+k_0,N} = P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) + X_{+k_0,N}U_{t,N}^0 \tag{4:72}$$

The time-varying matrix $X_{t+k_0,N}$ should be non-singular by the appropriate selection of weightings. The optimal control can now be computed by setting the above equation to zero and solving with respect to $U_{t,N}^0$ as shown below.

$$\begin{aligned} U_{t,N}^0 &= -X_{t+k_0,N}^{-1}P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) \\ &= -X_{t+k_0,N}^{-1}U_\delta^T V_{pt+k_0,N}^T\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) \end{aligned} \tag{4:73}$$

Note that the result in the above equation is identical to the optimal control vector in the *GPC* optimal control derivation (Section 3.3). This can be reinforced by the following theorem.

Theorem 4:1 – Equivalent Minimum Variance Optimal Control Problem

Consider a system as defined in Equations 4:2-4:20 and Figure 4:1-4:2 and assume that the nonlinear subsystem $W_{1k} = I$. The optimal *GPC* solution is given by Equation 3:33. If the *GPC* cost-function is modified to have the equivalent *GMV* form then it can be written as follows,

$$J(t) = E\{\Phi_{t+k_0,N}^T\Phi_{t+k_0,N}|t\} \tag{4:74}$$

where

$$\Phi_{t+k_0,N} = P_{cN,t}E_{pt+k_0,N} + F_{cN,t}^0U_{t,N}^0 \tag{4:75}$$

Now define the cost-function weightings according to the original *GPC* cost as shown below.

$$P_{cN,t} = U_\delta^T V_{pt+k_0,N}^T \text{ and } F_{cN,t}^0 = \Lambda_N^2 \tag{4:76}$$

with

$$V_{pt+k_0,N} = \left(C_{pt+k_0,N}B_{t+k_0,N} + E_{t+k_0,N}\right) \tag{4:77}$$

The resulting vector of optimal controls that minimises Equation 4:74 is then as follows and it is identical to the optimal vector resulting out of the GPC solution.

$$U_{t,N}^{0} = -X_{t+k_0,N}^{-1} U_{\delta}^{T} V_{pt+k_0,N}^{T} \left(D_{pt+k_0,N} + C_{pt+k_0,N} A_{t+k_0,N} \hat{x}(t+k_0|t) \right) \quad (4:78)$$

4.4.2 Nonlinear Predictive GMV Control Problem

In this section focus will shift more into the nonlinear aspect of the plant. A good starting point would be to enhance the *GMV* cost function explained in the previous section by an additional control costing term F_c as shown below. Note that in this case as well, k_0-steps ahead delays are considered for the output channels.

$$(F_c u)(t) = \left(F_{ck} z^{-k_0} u \right)(t) \quad (4:79)$$

Selection for this weighting factor can vary from being a linear dynamic operator (transfer-function etc.) to a nonlinear term to cancel the plant inherent nonlinearities if this is possible. Moreover it can be used to introduce anti-windup capability. The main assumption for the F_{ck} operator is that it has to be invertible. The extended cost-function will have the following form.

$$J_p = E\{ \Phi_{t+k_0,N}^{0T} \Phi_{t+k_0,N}^{0} | t \} \quad (4:80)$$

where $\Phi_{t+k_0,N}^{0}$ is here defined to include control signal costing terms as shown below.

$$\Phi_{t+k_0,N}^{0} = P_{cN,t} E_{pt+k_0,N} + F_{cN,t}^{0} U_{t,N}^{0} + F_{ck,N} U_{t,N} \quad (4:81)$$

The nonlinear function $F_{ck,N} U_{t,N}$ is normally defined to have a simple block-diagonal form as shown below.

$$(F_{ck,N} U_{t,N}) = diag\{ (F_{ck} u)(t), (F_{ck} u)(t+1), \dots, (F_{ck} u)(t+N) \} \quad (4:82)$$

$$\text{and} \quad U_{t,N}^{0} = \left(W_{1k,N} U_{t,N} \right) \quad (4:83)$$

with $W_{1k,N}$ being a block diagonal matrix of the following form,

$$\left(W_{1k,N} U_{t,N} \right) = diag\{ W_{1k}, W_{1k}, \dots, W_{1k} \} U_{t,N} = [(W_{1k} u)(t)^T, \dots, (W_{1k} u)(t+N)^T]^T \quad (4:84)$$

4.5 The LPV-NPGMV Optimal Control Solution

In a similar fashion to the previous section the *NPGMV* control solution is briefly described here using the *LPV* model formulation for the nonlinear subsystem as discussed earlier. Recall that the optimal state prediction $\hat{x}(t+k_0|t)$ and state estimation error $\tilde{x}(t+k_0|t)$ are

orthogonal and that the expectation of the product of the future values of the control action (assumed know in deriving the prediction equation), and the zero-mean white noise driving signals is null. Hence the predicted errors vector $\hat{E}_{pt+k_0,N}$ and the prediction errors vector $\tilde{E}_{pt+k_0,N}$ are also orthogonal. From Equation 4:75 Φ estimate and estimation error can be seen in Equations 4:85 and 4:86 respectively,

$$\Phi^0_{t,N} = \Phi_{t,N} + z^{-k_0}\left(F_{ck,N}U_{t,N}\right) \tag{4:85}$$

$$\Phi^0_{t+k,N} = \hat{\Phi}^0_{t+k,N} + \tilde{\Phi}^0_{t+k,N} \tag{4:86}$$

From Equation 4:66 the estimation error can be written as follows,

$$\tilde{\Phi}^0_{t+k,N} = \tilde{\Phi}_{t+k,N} = P_{cN,t}\tilde{E}_{pt+k_0,N} = U^T_{\delta}V^T_{pt+k_0,N}\tilde{E}_{pt+k_0,N} \tag{4:87}$$

The future predicted values of $\hat{\Phi}^0_{t+k,N}$ involve the estimated vector of weighted errors $\hat{E}_{pt+k_0,N}$ which are orthogonal to $\tilde{E}_{pt+k_0,N}$. Moreover, the estimation error is zero-mean and the expected value of the product with any known signal is null, therefore the cost-function can be written as follows,

$$J(t) = \hat{\Phi}^{0T}_{t+k,N}\hat{\Phi}^0_{t+k,N} + J_1(t) \tag{4:88}$$

The optimal control is the one that sets $\hat{\Phi}^0_{t+k,N}$ to zero or as shown in the following equation.

$$P_{cN,t}\hat{E}_{pt+k_0,N} + \left(F_{ck,N} + F^0_{ck,N}W_{1k,N}\right)U_{t,N} = 0 \tag{4:89}$$

The resulting optimal controls vector is shown below.

$$U_{t,N} = -\left(F_{ck,N} + \Lambda^2_N W_{1k,N}\right)^{-1}P_{cN}\hat{E}_{pt+k_0,N} \tag{4:90}$$

Looking at the above equation, it is seen that the resulting control law it involves the nonlinear control weighting $F_{ck,N}$ and the nonlinear plant subsystem $W_{1k,N}$. Further simplification can be achieved by substituting $\hat{E}_{pt+k_0,N}$ from Equation 3:19 into the condition of optimality $\hat{\Phi}^0_{t+k,N} = 0$. The motivation behind this is an easier implementable formulation of the controller.

$$\begin{aligned}\widehat{\Phi}^0_{t+k,N} = \widehat{\Phi}^0_{t+k,N} + \left(F_{ck,N}U_{t,N}\right) = P_{cN,t}\hat{E}_{pt+k_0,N} + F^0_{cN,t}U^0_{t,N} + \left(F_{ck,N}U_{t,N}\right) \\ = P_{cN,t}\left(\widetilde{D}_{pt+k_0,N} + V_{pt+k_0,N}U_\delta\ U^0_{t,N}\right) + F^0_{cN,t}U^0_{t,N} + \left(F_{ck,N}U_{t,N}\right) = 0\end{aligned} \tag{4:91}$$

Further expansion leads to the following form for the optimality condition.

$$\begin{aligned}P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) + \left(U^T_\delta V^T_{pt+k_0,N}V_{pt+k_0,N}U_\delta\ + \Lambda^2_N\right)U^0_{t,N} \\ + F_{ck,N}U_{t,N} = 0\end{aligned} \tag{4:92}$$

The above equation can be summarised as follows,

$$P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) + \left(F_{ck,N} + X_{t+k_0,N}W_{1k,N}\right)U_{t,N} = 0 \tag{4:93}$$

An expression for the optimal controls vector more practical for implementation is shown below.

$$\begin{aligned}U_{t,N} = -F^{-1}_{ck,N}\left(P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\hat{x}(t+k_0|t)\right) + X_{t+k_0,N}W_{1k,N}U_{t,N}\right) \\ = -F^{-1}_{ck,N}\left(P_{cN,t}D_{pt+k_0,N} + C_{\varphi t}\hat{x}(t+k_0|t) + X_{t+k_0,N}W_{1k,N}U_{t,N}\right)\end{aligned} \tag{4:94}$$

where

$$\begin{aligned}P_{cN,t} = U^T_\delta V^T_{pt+k_0,N} \\ \text{and}\ \ C_{\varphi t} = P_{cN,t}C_{pt+k_0,N}A_{t+k_0,N} = U^T_\delta V^T_{pt+k_0,N}C_{pt+k_0,N}A_{t+k_0,N}\end{aligned} \tag{4:95}$$

The inherent model-based nature of the *NPGMV* control law is evident here as it contains an internal model for the nonlinear process. Similar to standard *GPC*, the receding horizon strategy is applied here as well and only the first element of the optimal controls vector is used. When the control costing term tends to zero ($F_{ck,N} \to 0, W_{1k,N} = I$) this becomes identical to a *GPC* controller in the limiting linear case. Expanding $\hat{x}(t+k_0|t)$ from Equation 4:54 and 4:94 becomes as follows,

$$\begin{aligned}P_{cN,t}\left(D_{pt+k_0,N} + C_{pt+k_0,N}A_{t+k_0,N}\left(A^{k_0}_t\hat{x}(t|t) + T(k_0,z^{-1})u_0(t)\right)\right) \\ + \left(F_{ck,N} + X_{t+k_0,N}W_{1k,N}\right)U_{t,N} = 0\end{aligned} \tag{4:96}$$

Substituting $u_0(t) = W_{1k}u(t)$ and using definitions in Equation 4:95 yields the following form.

$$\begin{aligned}\left(P_{cN,t}D_{pt+k_0,N} + C_{\varphi t}A_t^{k_0}\hat{x}(t|t) + C_{\varphi t}T(k_0, z^{-1})u_0(t)\right) \\ + \left(F_{ck,N} + X_{t+k_0,N}W_{1k,N}\right)U_{t,N} = 0\end{aligned} \tag{4:97}$$

A final useful step would be to separate the optimal controls vector into present (applied in time t) and future (derived for $t_{i,N} > t$) elements. This can be done via multiplication of the controls vector with the following matrices.

$$C_{I0} = [I, 0, \dots, 0] \text{ and } C_{0I} = [0 \ I_N] \tag{4:98}$$

The following definitions can then be obtained for the current and future controls.

$$u(t) = C_{I0}U_{t,N} \text{ and } U_{t,N}^f = C_{0I}U_{t,N} \tag{4:99}$$

The vector of future optimal controls can then be reformulated as follows,

$$\begin{aligned}U_{t,N} = \left(-F_{ck,N}\right)^{-1}\left(P_{cN,t}D_{pt+k_0,N} + C_{\varphi t}A_t^{k_0}\hat{x}(t|t)\right. \\ \left. + \left(X_{t+k_0,N} + C_{\varphi t}T(k_0, z^{-1})C_{I0}\right)W_{1k,N}U_{t,N}\right)\end{aligned} \tag{4:100}$$

Implementation of this control structure is shown in the following diagram where W_{0_LPV} denotes now the *LPV* nonlinear plant approximation.

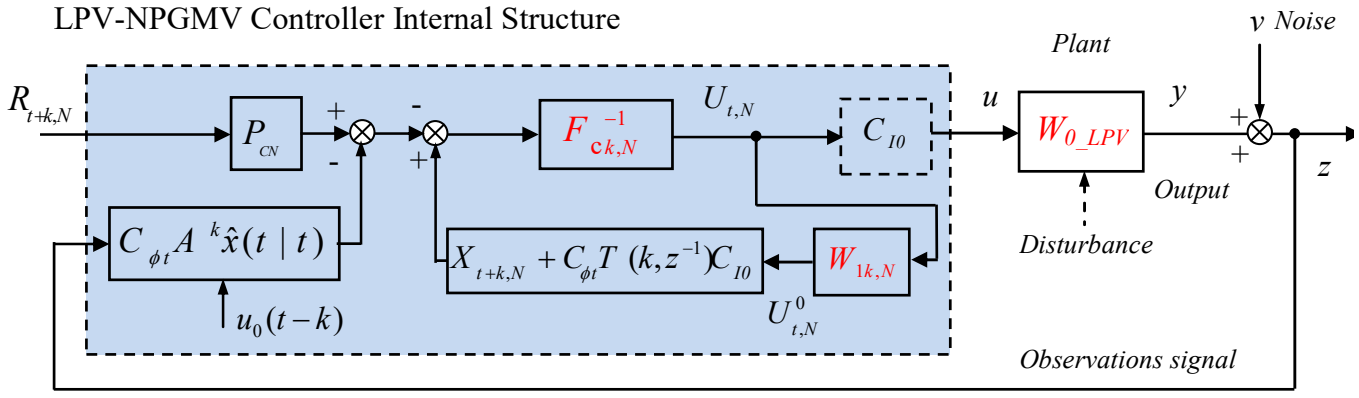


Figure 4:4 *LPV-NPGMV* implementation structure block diagram.

The flexibility in implementation this formulation allows regrading plant decomposition options is shown below and the suitability of each is dependent upon the specific nature of a control problem and the level of information available about the system.

System fully linear and known:

The nonlinear operator W_{1k} can be set to unity and the system is modelled and fully absorbed by the W_0 term.

System unknown:

The system is fully absorbed as a *"black-box"* by the nonlinear operator W_{1k} and the W_0 term is set to unity.

System only contains known smooth differentiable nonlinearities – linear part extractable:

Option 1: the nonlinear part is modelled and absorbed by the W_{1k} operator and the linear part absorbed by W_0.

Option 2: if possible the nonlinear system is approximated as *LPV* and absorbed by the W_{0t} part whereas W_0 (used in cascade) absorbs the linear part or set to unity. The W_{1k} operator is set to unity.

If a linear part cannot be extracted from the overall plant then the W_0 term is simply set to unity.

System only contains known hard nonlinearities (e.g. saturation constraints):

The nonlinear operator W_{1k} is used to absorb the nonlinearity and W_0 term is either set to unity if linear part unavailable or absorbs the full linear part if available.

System only contains known hard and soft nonlinearities:

Option 1: the nonlinear part is modelled and fully absorbed by the W_{1k} operator and W_0 is set to unity.

Option 2: if possible the smooth nonlinear part is approximated as *LPV* and absorbed by the W_{0t} part whereas W_{1k} absorbs all hard nonlinearities.

At this point is important to summarise how one controller results to another under certain conditions as follows.

***LPV-NPGMV → NGMV*:** the *LPV-NPGMV* essentially shares the same cost function with the *NGMV* i.e. $J = E\{\Phi_{t+k,N}^T \Phi_{t+k,N} | t\}$ but in a multi-cost future formulation. Therefore if the prediction horizon is set to 1 and the *LPV* model is fixed it effectively reverts to an *NGMV* controller.

***LPV-NPGMV → NGPC*:** if the general nonlinear operator W_{1k} is set to unity and the plant is absorbed in its entirety by the *LPV* approximation within the term W_{0t} then according to the Equivalent Minimum Variance Theorem (Theorem 4:1) the former reverts to an *NGPC* controller.

LPV-NPGMV FORMULATION TUTORIAL EXAMPLE & CONTROLLERS COMPARISON

In this section a simple tutorial example is given to support understanding of how the *LPV-NPGMV* controller is formulated. This includes definition of all sub-components, definition of weightings and most importantly of the various plant sub-systems into true nonlinear and *LPV* forms. As discussed previously, in this form of the controller the nonlinear operator can be used to contain hard nonlinearities such as saturation constraints in the same manner as in the basic *NGMV*, whereas the linear subsystem is now replaced by an *LPV* approximation that can be used to encapsulate the remaining soft dynamic nonlinearities inherent within the process.

LPV Plant Subsystem W_0:

Here a simple *SISO* two state mass-spring-damper system is considered. The mass of the system is used as a varying parameter which varies subject to an external disturbance input signal. The original system equation is described as follows.

$$m(u)\ddot{y} + c\dot{y} + ky = F(t)$$

Where u is the external disturbance signal, c is the damping ratio, k is the spring stiffness coefficient and $F(t)$ is the force input that drives the system. The output of the system is y which is the position of the mass at a given time t. It is important to observe that the system is linear for a fixed value of the disturbance signal but will structurally vary with time if u varies as well. The state-space equation of the system and related matrices are shown below.

$$A_{lpv} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m(u)} & -\frac{c}{m(u)} \end{bmatrix}, B_{lpv} = \begin{bmatrix} 0 \\ -\frac{1}{m(u)} \end{bmatrix}, C_{lpv} = [1 \quad 0], x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

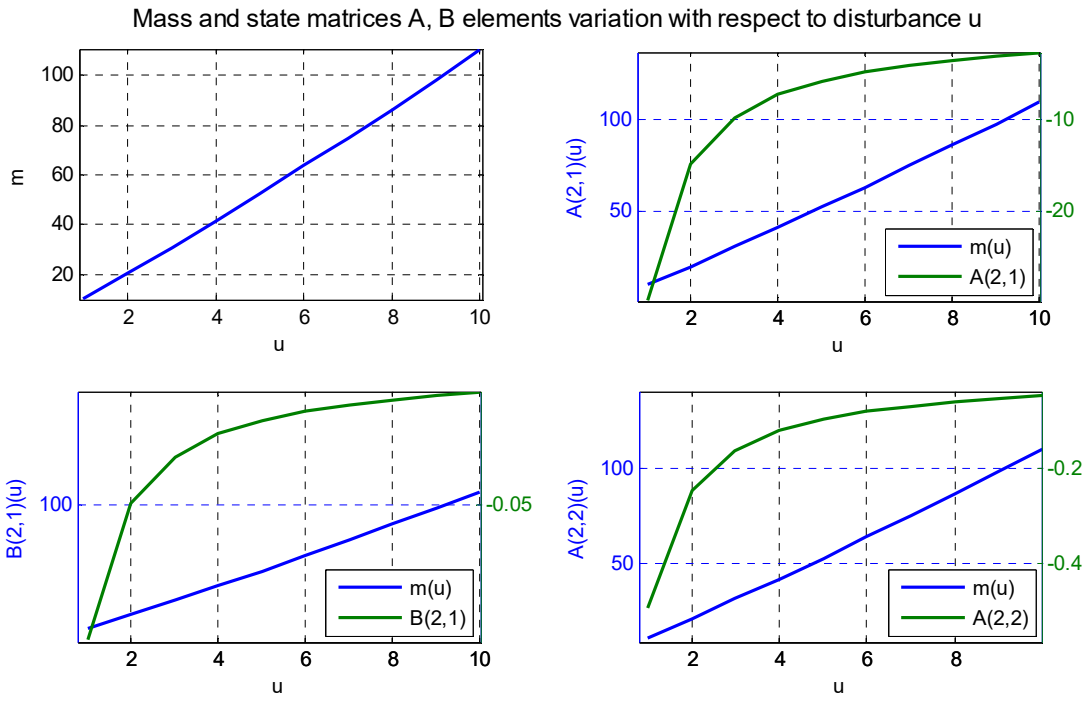
$$\dot{x} = A_{lpv}x + B_{lpv}u$$

$$y = C_{lpv}x$$

Suppose now that the mass varies according to the following equation with the disturbance input signal ranging within 1:10.

$$m(u) = 10u + 0.1u^2$$

This results into variations of the state-space matrices as illustrated in the following graph.



For simplicity in this example let W_r and W_d be defined as unity gains. Recall the *LPV-NPGMV* control law (see below) and using the diagram in Figure 4:4 as a point of reference the control loop components are defined in the following sections.

Weightings P_c, F_{ck} Definition:

For this example the following discrete transfer function was used as the error weighting based on the *PID* weighting selection method. A sample time of *Ts* = 0.1 sec was chosen for the controller.

$$P_c = \frac{25 + 5z^{-1}}{1 - z^{-1}}$$

The control weighting is defined as the following dynamic transfer.

$$F_{ck} = \frac{0.009571 - 0.009476z^{-1}}{1 - 0.9048z^{-1}}$$

Control Loop Gains Calculation:

Recall the *LPV-NPGMV* optimal control,

$$U_{t,N} = \left(-F_{ck,N}\right)^{-1}\left(P_{cN,t}D_{pt+k_0,N} + C_{\varphi t}A_t^{k_0}\hat{x}(t|t) + \left(X_{t+k_0,N} + C_{\varphi t}T(k_0, z^{-1})C_{I0}\right)W_{1k,N}U_{t,N}\right)$$

and also, $P_{cN,t} = C_{pt+k_0,N}B_{t+k_0,N} + E_{t+k_0,N}$ and $F_{cN,t}^0 = \Lambda_N^2$

where $P_{cN,t}D_{pt+k_0,N} + C_{\varphi t}A_t^{k_0}\hat{x}(t|t)$ is the vector form concatenation of the *N* future predictions of the augmented *LPV* system free response part.

This term is equivalent to the free response term $F_{t,N}$ in *GPC* vector form predictions equation $\hat{Y}_{t,N} = F_{t,N} + S_N U_{t,N}$. Also recall that *A B* and *C* matrices (as explained in tutorial example in Section 2.2) refer to the combined (augmented) process model excluding hard constraints such as saturations which are included in the nonlinear plant operator $W_{1k,N}$. Different to the *NGMV* tutorial example in this case these matrices contain the corresponding future terms. To demonstrate this numerical let us first define the augmented *LPV* process part as follows.

$$\begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_{k+1} = \begin{bmatrix} A_{lpv} & 0 & 0 & 0 \\ 0 & A_d & 0 & 0 \\ 0 & 0 & A_r & 0 \\ -B_p C_{lpv} & -B_p C_d & B_p C_r & A_p \end{bmatrix} \begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_k + \begin{bmatrix} B_{lpv} \\ 0 \\ 0 \\ -B_p E_{lpv} \end{bmatrix}$$

Substituting A_{lpv}, B_{lpv} and C_{lpv} from the previous section, neglecting W_r subsystem and substituting $A_d = 1$, $A_p = 1$, $C_d = 1$, $B_p = 4$ results in the following,

$$\begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{300}{m(u)} & -\frac{5}{m(u)} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -4 & 0 & -4 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_d \\ x_r \\ x_p \end{bmatrix}_k + \begin{bmatrix} 0 \\ -\frac{1}{m(u)} \\ 0 \\ 0 \end{bmatrix}$$

Setting $E = 0$ and using Equation 4:52 to calculate the composite future *A*, *B* and *D* results in the following matrices, for *u* = 1 and prediction horizon of $N_p = 2$ and control horizon $N_c = 1$.

$$A_{t+k_0,N} = \begin{bmatrix} 0.8575 & 0.0928 & 0 & 0 \\ -2.7566 & 0.8115 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -4 & 0 & -4 & 1 \\ 0.8575 & 0.0928 & 0 & 0 \\ -2.7566 & 0.8115 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -4 & 0 & -4 & 1 \end{bmatrix}$$

$$B_{t+k_0,N} = \begin{bmatrix} -0.0005 \\ -0.0092 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$C_{pt+k_0,N} = [-105 \quad 0 \quad -105 \quad 2.5 \quad -105 \quad 0 \quad -105 \quad 2.5]$ and $P_{cN,t} = 0.0525$

Observe that the number of rows of the *A* matrix equals the number of states $[x_0, x_d, x_p]$ times the prediction horizon $n = 8$ and the number of columns the number of states times the control horizon $m = 4$. The *D* matrix relates to the stochastic disturbance component which for simplicity in this tutorial can be set to unity.

The $X_{t+k_0,N} + C_{\varphi t}T(k_0, z^{-1})$ term is equivalent to the S_N forced response term in the *GPC* control law however in the *LPV-NPGMV* this signal drives the nonlinear operator W_k.

$$C_{\varphi t} = [-200.075 \quad -19.488 \quad -230 \quad 5]$$

Recall,

$X_{t+k_0,N} = Y_{t+k_0,N}^T Y_{t+k_0,N} = U_\delta^T V_{pt+k_0,N}^T V_{pt+k_0,N} U_\delta + \Lambda_N^2$ and $V_{pt+k_0,N} = \left(C_{pt+k_0,N} B_{t+k_0,N} + E_{t+k_0,N}\right)$ hence, $X_{t+k_0,N} = 0.0028$ and Λ_N^2 is set here as 1.

Simulation and Controller Structure Comparison:

In this section a simple simulation of the aforementioned system is provided, not only to demonstrate performance of the *LPV-NPGMV* but also to portray the equivalence to the *NGPC, NPGMV* and *NGMV* controllers showing that the three latter are but special cases of the first.

In the first simulation artificial actuator constraints where introduced to allow differentiation between the *NGPC* and the two *NGMV* based controllers. Remember the *NGPC* is essentially using the linear *GPC* cost function and an *LPV* approximation of the nonlinear system whereas the *LPV-NPGMV* utilises the *GMV* cost function with a delay-free nonlinear operator (in this

case containing the actuator constraints) within the inner controller loop. The *LPV* system is fixed using a constant value for parameter *u* and different *NGMV* formulations are compared in tracking a varying sinusoidal position setpoint. Results are shown in the following graphs.

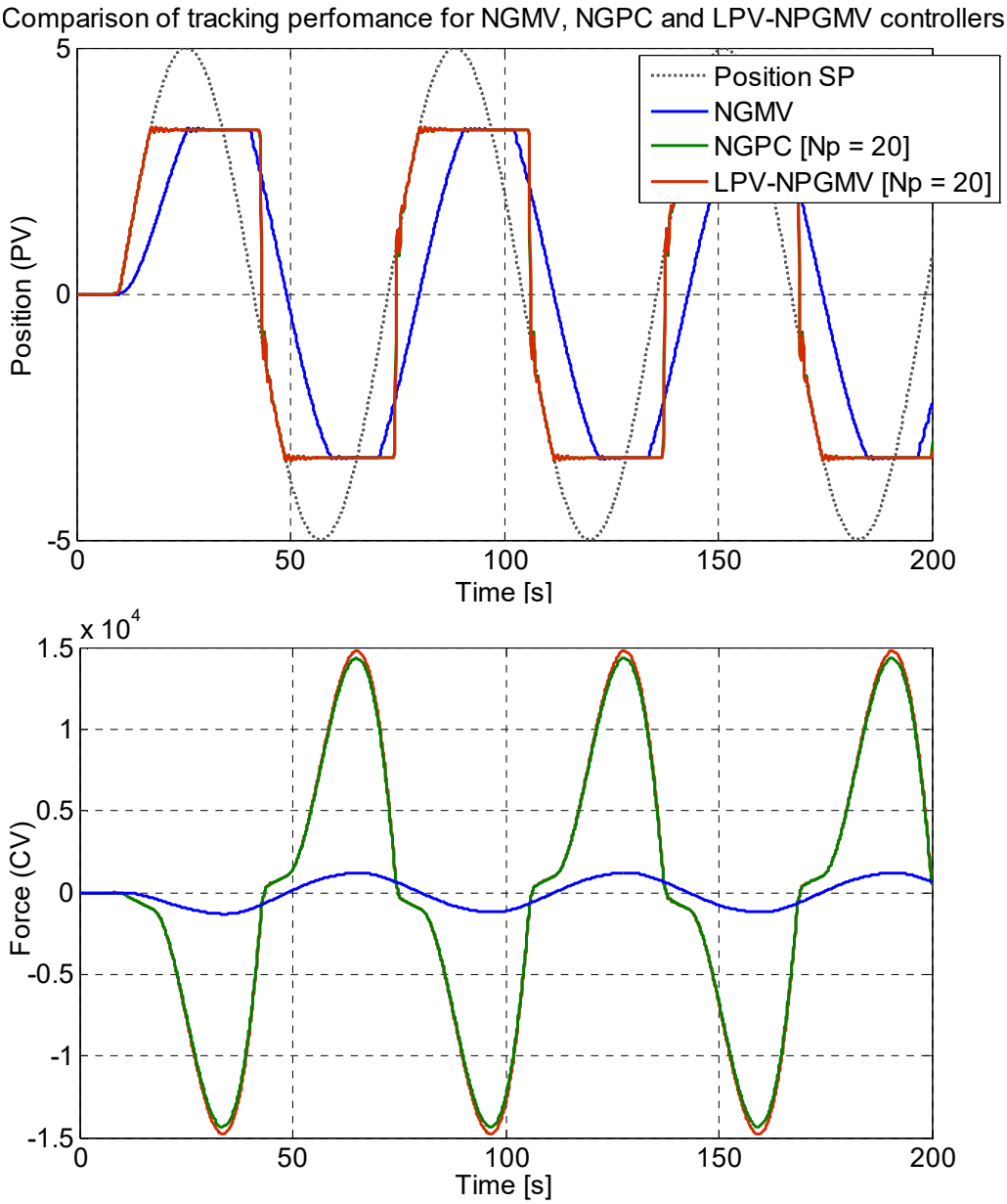


Figure 4:5 Control performance comparison for a sinusoidal trajectory in the presence of constraints.

All three controllers utilise identical definition of weighting factors. As mentioned previously according to Theorem 4.1 will yield similar performance if no constraints are present. In this case there is a small difference which is only observable in the control signals (Force CV) as

saturation effect clamps both controllers identically. Observe the benefit of a long prediction horizon comparing to the more sluggish one step *NGMV* controller.

In the second simulation the Position setpoint is kept constant at 1 and saturation removed to examine the effect of variations in system parameter *u* in regulation performance of the *NGPC, LPV-NPGMV* and the *NPGMV* which essentially utilises a fixed non-updating model of the process. The results are shown below for a sinusoidal variation of *u* between 0-10.

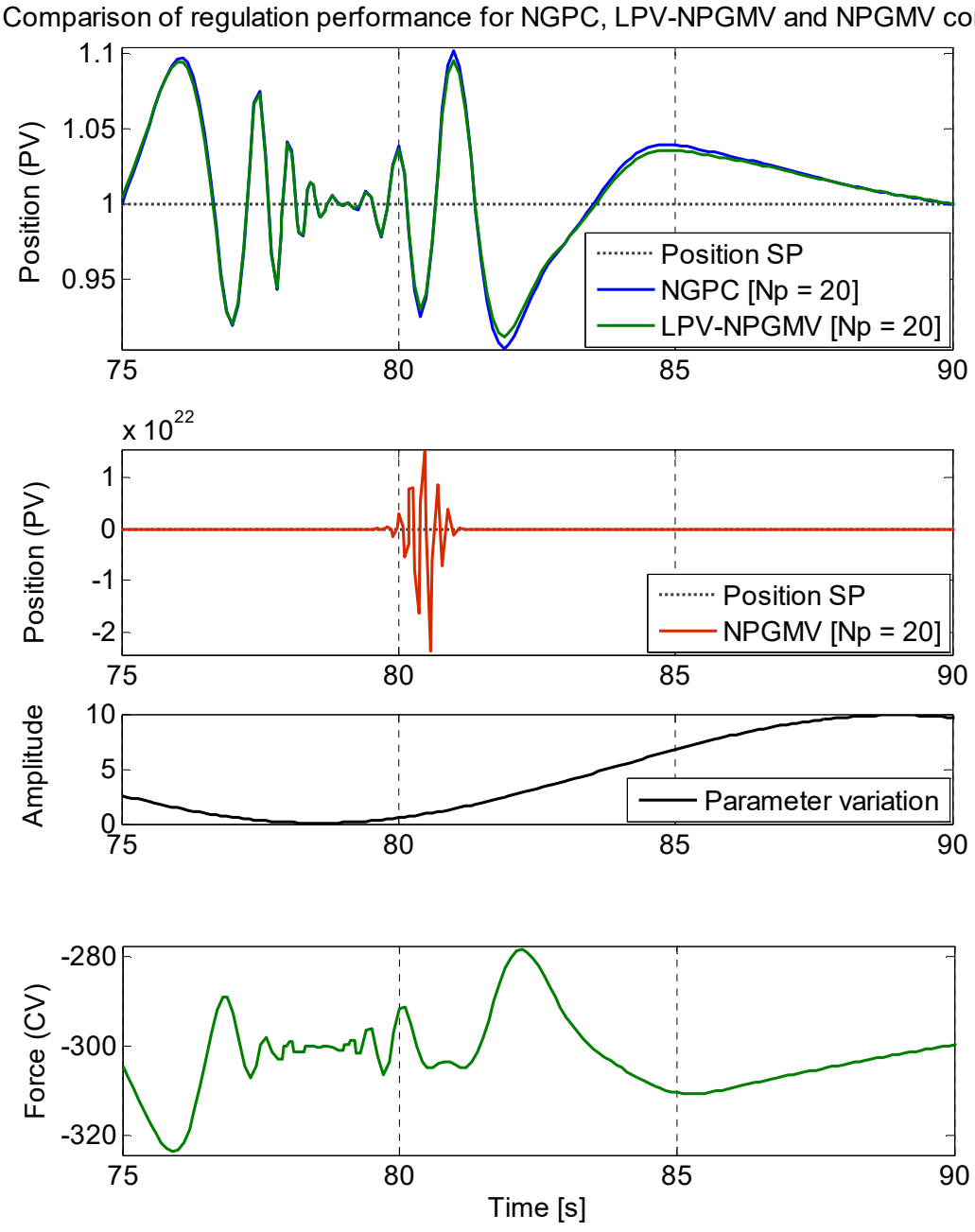


Figure 4:6 Control performance comparison in regulation during model parameter variation.

Since there are no constraints the *NGPC* and *LPV-NPGMV* yield very similar results as expected with the latter giving a small observable improvement. The *NPGMV* goes unstable in various parts of the *u* trajectory due to the mismatch between the actual model and the one included within controller calculations. It is interesting to mention that an artificial increase of the delay term above 1 was found to result in an unstable system for the *NGPC* whereas the *LPV-NPGMV* cost function, i.e. including a delay-free operator of the plant, had a stabilising effect.

4.6 LPV-NGMV Derivation

A simple and easy to implement control formulation is proposed in this section. The structure of the controller remains exactly the same as described by Equation 2:33 and Figure 2:5. The single difference will be that now the output subsystem. Recall Equation 2:13: $(W_0 u_0)(t) = \left(W_{0k} z^{-k} u_0\right)(t)$, is no longer Time-Invariant but an *LPV* subsystem. The control law will remain the same except from the matrices contained in the prediction equations ($\hat{x}$ for *k*-steps ahead, Equation 2:24. The *LPV-NGMV* control law is derived in a very similar way as described in (Grimble and Pang, 2007), Theorem 1 but adapted to *LPV* systems. This is shown in the following section.

Theorem 4.2: *LPV-NGMV* Controller

Let the operator N_0 represent the mapping from the signal $u(t)$ (input to the nonlinear *LPV* subsystem) to the signal $\varphi_0(t)$ to be minimised:

$$(N_0 u)(t) = \left(\left(P_c W_{0k}(t+k_0) - Z_c \widetilde{W}_{0k}(t+k_0)\right) u\right)(t) \tag{4:101}$$

Assume that the weighting operators P_c, Z_0 and F_{ck} are chosen so that the *NL* operator: $(N_0 W_{1k} - F_{ck})$ has a finite-gain m_2 stable causal inverse, to ensure the system is closed-loop stable. The *NGMV* optimal controller to minimize the variance of the weighted error, states and control signals may then be computed. The NGMV optimal control signal may be expressed in the form:

$$\begin{aligned} u(t) = -F_{ck}^{-1} \Big[& C_\varphi(t+k_0) A^{k_0} \hat{x}(t|t) \\ & + \left(C_\varphi(t+k_0) T(k_0, z^{-1}) + E_\varphi(t+k_0)\right) W_{1k} u(t) \Big] \end{aligned} \tag{4:102}$$

where C_φ and E_φ as described in Section 2.2 (Augmented State-Space System Model).

Proof: $J = E\{\varphi_0(t+k_0)^T \varphi_0(t+k_0)\}$ is minimised when $\widehat{\varphi}_0(t+k_0|t) = 0$. This is satisfied by the expression for optimal control described by Equation 4:99. As mentioned previously, stability properties have not been further explored within the scope of this work. However, the main assumptions for the stability of the closed-loop are similar to those included in the remarks section in (Grimble and Majecki, 2015).

Chapter 5 Dynamic Positioning and Manoeuvring of Marine Vessels

Following the theoretical derivation of the state-space *NGMV* law in Chapter 2 and *LPV-NGMV* in Chapters 2 and 4, an application on control of marine vessels is used in this chapter to explore control performance under various scenarios for both controllers. There are two important topics in which automatic control plays a major role in *DP* systems, as distinguished below.

1. *Manoeuvring*
2. *Dynamic Positioning*

The first refers to the guidance of marine vessels along a certain route, for which either prior knowledge is available, or it is generated with respect to certain requirements like obstacle evasion etc. At large, ship manoeuvring is part of a greater family of systems characterised by the trajectory-tracking problem. Originally, it was treated under the assumption that the ship is moving in a controlled environment with insignificant external disturbances. This configuration effectively isolates the tracking problem from its disturbance rejection counterpart and makes control design somewhat easier. This study however, deals with both aspects in unison to fit the *NGMV* framework in a more appropriate way. For consistency, throughout this section manoeuvring describes the task where a vessel is controlled to follow a specific geometric path.

Dynamic positioning, or station-keeping, on the other hand describes the task where the vessel is commanded to maintain a certain position while being subjected to environmental disturbances like current, wave and wind forces. It falls under the general category of setpoint regulation. This subject hosts many real life applications. One example that combines both the manoeuvring and *Dynamic Positioning* (*DP)* tasks is referenced in (Martin, 2004) and describes an open sea oil platform supply operation. In this case a small supply vessel is commissioned to temporarily attain certain points along a path around an oil extraction platform. After its function is over then it needs to navigate to the next point automatically.

Nonlinearities potentially arise in many parts of these systems. The most evident are observed in the kinematics and hydrodynamics of the vessel but they should also be considered in the thrusters. They may appear as couplings between various degrees of freedom (*DOF*) or as range/rate limits in the vessel's actuators. Nonlinearities may also appear on the control side of the problem. Wave disturbances, of a certain frequency range, need to be omitted from the total disturbance input as their variations results into aggressive controls, thus increasing thruster wear and tear. A simple technique to address this problem was to apply individual *PID* controllers, for the surge the sway and the yaw, each in cascade with a *Low Pass* or a *Notch* filter to remove these frequencies from the control loop (Katebi, Morardi, 2000). The disadvantage in this setup is the additional phase lag introduced in the loop by the filters,

which can severely reduce the bandwidth of the system. The main goal in this chapter is to explore control possibilities to tackle both tasks with a single solution.

Two weaknesses previously addressed (Breivik and Fossen, 2004) are the transformation between two coordinate systems and the parameter adaptation on changing environmental conditions, both applied on the error minimization to ensure smoother and realistic convergence to the desired path. Previous approaches include *PID* with Filtering, which was common practice up until the 60s, *LQG* (Shneider et al., 1969) and *H∞ MIMO* designs (Katebi et al., 1997), (Balchen et al, 1980), which became popular after the *Kalman Filter* development, but also nonlinear techniques like *Adaptive Backstepping* (Fossen and Grovlen, 1998), sliding mode, *Fuzzy Logic* and nonlinear *PID* approaches (Yamamoto and Daigo, 1998). The *PID* nevertheless remained in applications for both tracking and dynamic positioning and it wasn't but recently that the prospective of more advanced schemes was explored.

The proposed *NGMV* controller in this chapter is designed to treat the system in full coupling for improved compensation. Furthermore, as discussed in Chapter 2, unlike previous approaches, it takes into account the full nonlinear system rather than a linear approximation. This work is divided into the following sections.

Section 1 – Problem Description; overview of the control objectives and strategy employed in this work.

Section 2 – System Model Description; derivation of a suitable state-space model used within the state-space *NGMV* design and in simulation.

Section 3 – Control System Description; adaptation of the control algorithm as derived in Equation 2:32 and also of a basic *PID* control formulation used here for comparison.

Section 4 – Simulation Results for the Basic *NGMV*; definition of the different scenarios and presentation of control performance results for the different schemes employed throughout the simulations.

Section 5 – Simulation Results for the *LPV-NGMV*; definition of the main *DP* scenario and presentation of control performance results for the basic and *LPV-NGMV* controllers.

5.1 Problem Description

The main objective is to control the axial thruster forces of a fully actuated vessel and to keep it stationary at a point or guide it along a specified 2-axis geometric trajectory. Environmental disturbances are considered throughout the process for performance evaluation and comparison against a classical *PID* design. Simulations are based on the *CyberShip II* model. It has been common for this problem to break into the two following tasks,

1. *Geometric task*

2. *Dynamic task*

The first ensures that the position vector of the ship *y(t)* converges to a desired trajectory $y_d(t)$ parameterised by a path variable ϑ,

$$\lim_{t\to\infty}\left[y(t) - y_d\left(\theta(t)\right)\right] = 0 \tag{5:1}$$

The second ensures that the ships velocity $\dot{\vartheta}$ converges to a desired speed v_d,

$$\lim_{t\to\infty}\left[\dot{\theta}(t) - v_d\left(\theta(t)\right)\right] = 0 \tag{5:2}$$

The primary objective is to use the coordinate transformation used in (Breivik and Fossen, 2004) along with the parameters of *Cybership II*, acquired via identification, and design the *NGMV* algorithm to fulfil both the geometric and the dynamic task with satisfactory performance. Actuator saturations, as part of the ship model nonlinearities, and wave disturbances affecting various parts of the system, have also been considered for realistic simulation results.

5.2 Generalised System Model Description

Unlike the model separation relevant to the application (*Navigation* or *DP*), that is found in traditional theory, a single vessel model is used to cover both cases. That model comprises of kinematics which describe the geometric aspects of motion with respect to the body coordinate frame, hydrodynamics which describe the forces exerted in the vessel and causing these motions and also propulsion and control dynamic characteristics and limitations. Wave disturbances are also modelled. Position commands are given with respect to the *North-East-Down* (*NED*) coordinate frame therefore defining the necessary transformation matrices, from that to the *Body-fixed* frame, is an additional system component (Katebi, Morardi, 2000).

Vessel Body Kinematics Definition:

The geometric aspects of motion and forces in 6 *DOF* are illustrated in Figure 5:1 and a summary of them is shown in Table 5:1. The position and thrust forces vectors shown below are the principal vectors for our design. The *DP* control system acts upon the error between desired and actual position vector and produces an optimal vector of thrust forces to eliminate this error.

$$\eta = [x, y, \psi]^T \in R^3, \qquad \tau = [\tau_x, \tau_y, \tau_N]^T \in R^3 \tag{5:3}$$

Table 5:1: General Notation of Marine Vessel Kinematics

DOF	Definition	Force (1-3)/ Moment (4-6)	Linear (1-3)/ Angular (4-6) Velocity	Positions (1-3)/ Euler (4-6) Angles
1	*surge*	*X*	*u*	*x*
2	*sway*	*Y*	*υ*	*y*
3	*heave*	*Z*	*w*	*z*
4	*roll*	*K*	*p*	*φ*
5	*pitch*	*M*	*q*	*ϑ*
6	*yaw*	*N*	*r*	*ψ*

In ship manoeuvring applications the dominant control modes are the surge, along the fore-aft axis, the sway, along the starboard axis and yaw, which is the rotation about the centre of the vessel. Yaw is the principal mode for navigation whereas roll, pitch and heave are responsible for the safety of the cargo/ comfort of the passengers hence target of stabilisation applications. Note that for this application we assume stability about the pitch and roll modes and we are only concerned on the guidance/*DP* problem hence the model can be reduced into *3-DOF*.

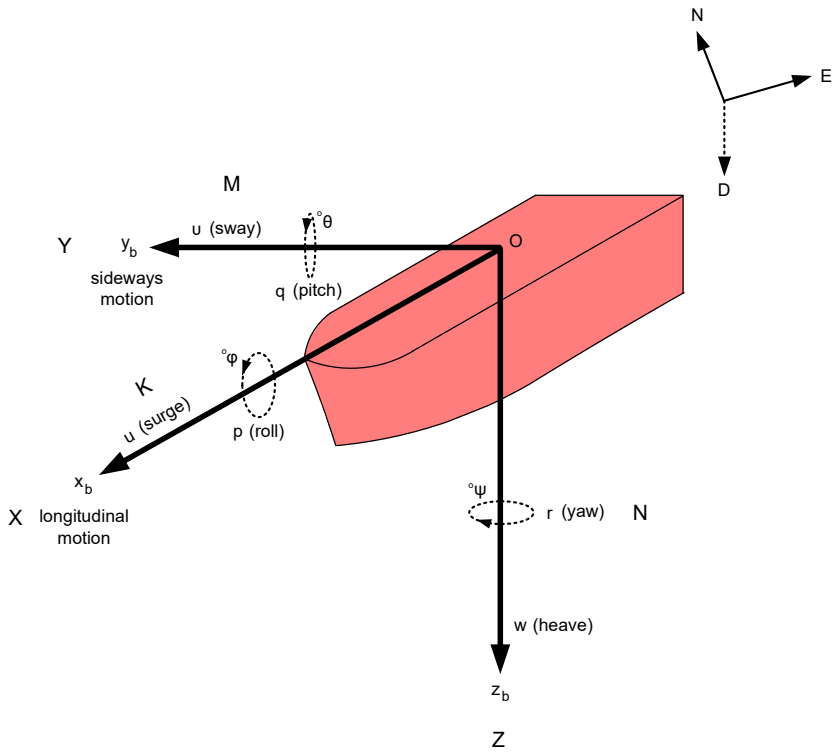


Figure 5:1 Geometric representation of a 6 *DOF* vessel kinematics.

Vessel Body Dynamics Definition:

The hydrodynamic equations that describe a fully actuated marine vessel can be summarised in Equations 5:4 and 5:5. These matrices contain parameters for the hydrodynamic, aerodynamic, wave and control forces in all three directions (Fossen and Grovlen, 1998), (Breivik and Fossen, 2004). A summary of parameter definition can be seen in the following table.

$$\dot{\eta} = R(\Psi)v \tag{5:4}$$

$$M\dot{v} + C(v)v + D(v)v = \tau + R(\Psi)b \tag{5:5}$$

where

$$R(\Psi) = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:6}$$

is the rotation matrix from *NED* (*v* vector) to *BODY* (*η* vector) coordinate frame,

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{5:7}$$

is the rigid body minus the added mass inertial matrix, x_g is the centre of gravity along the body *x*-axis, m is the vessel mass, $X_{\dot{u}}, Y_{\dot{v}}, Y_{\dot{r}}, N_{\dot{v}}, N_{\dot{r}}$ are the hydrodynamic derivatives that define forces due to accelerations in a certain direction and I_z is the moment of inertia about the *z*-axis (yaw rotation),

$$C(v) = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v - (mx_g - Y_{\dot{r}})r \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v + (mx_g - Y_{\dot{r}})r & (-m + X_{\dot{u}})u & 0 \end{bmatrix} \tag{5:8}$$

is the *Coriolis/centrifugal* skew-symmetric matrix,

$$D(v) = \begin{bmatrix} -X_u - |u|X_{uu} - u^2X_{uuu} & 0 & 0 \\ 0 & -Y_v - |v|Y_{vv} - |r|Y_{rv} & -Y_r - |v|Y_{vr} - |r|Y_{rr} \\ 0 & -N_v - |v|N_{vv} - |r|N_{rv} & -N_r - |v|N_{vr} - |r|N_{rr} \end{bmatrix} \tag{5:9}$$

is the hydrodynamic damping matrix. *b* is the *n*-frame bias due to currents. X_u, Y_v, Y_r, N_v, N_r are the forces due to angular velocities similarly. The *τ* vector contains control thrust forces in the surge, sway and yaw directions,

$$\tau = [\tau_X, \tau_Y, \tau_N]^T \in R^3 \tag{5:10}$$

but it can also contain wave and other disturbances.

$$X_{\dot{u}} \triangleq \frac{\partial X}{\partial \dot{u}} \tag{5:11}$$

is the force in the x-direction caused by acceleration in the same direction. Note that these are simplified matrices derived by neglecting the pitch, roll and heave modes. The following table contains the values of the model parameters used in simulation.

$m = 23.8$	$I_z = 1.76$	$x_g = 0.046$	
$X_u = -0.7225$	$X_{uu} = -1.3274$	$X_{uuu} = -5.8664$	$X_{\dot{u}} = -2$
$Y_v = -0.8612$	$Y_{vv} = -36.2823$	$Y_{\dot{v}} = -10$	
$Y_r = 0.1079$	$Y_{rr} = -3.45$	$Y_{\dot{r}} = 0$	
$Y_{rv} = -0.805$	$Y_{vr} = -0.845$		
$N_v = 0.1052$	$N_{vv} = 5.0437$	$N_{\dot{v}} = 0$	
$N_r = -1.9$	$N_{rr} = -0.75$	$N_{\dot{r}} = -1$	
$N_{rv} = 0.13$	$N_{vr} = 0.08$		

Different assumptions apply when dealing with a *DP* or a Manoeuvring application e.g. the *Coriolis* matrix can be neglected due to slow speed operation in *DP*, this work however is an attempt to include most of the nonlinearities for evaluation. A simple diagram which describes the vessel model used in simulation is shown in the following diagram.

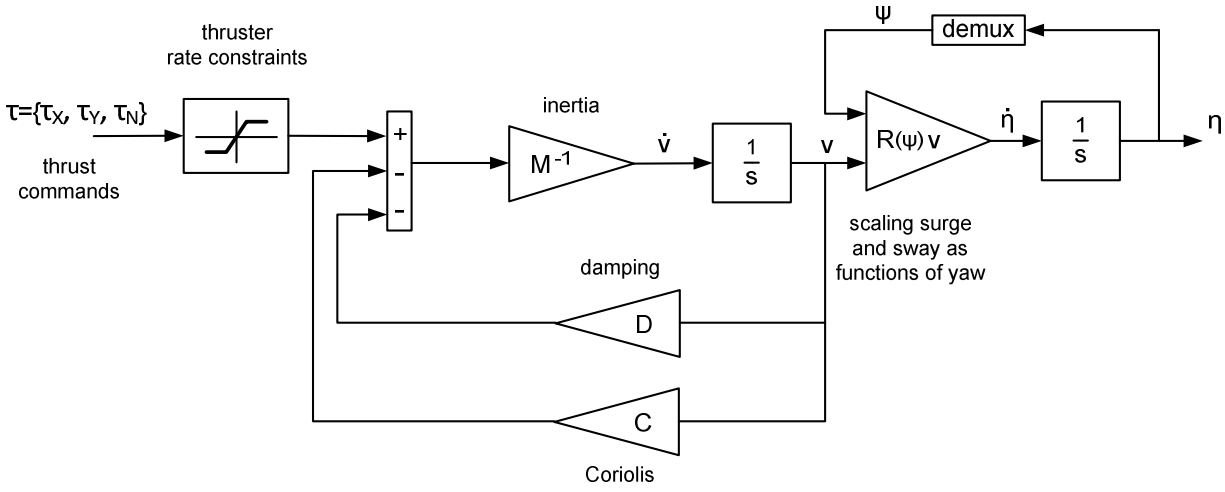


Figure 5:2 Dynamic Marine Vessel system simulation diagram.

Trajectory Analysis:

The position $p \in R^2$ and velocity $v \in R^2$ vectors of the vessel are defined in a 2-dimensional coordinate system; one vessel and one *earth-fixed*. The vessel itself is assumed to be an ideal point mass particle and our objective is to guide it along a geometrical path on the surface (Martin, 2004).

The positioning error:

$$\varepsilon \; = \; p - p_d(\theta^*) \tag{5:12}$$

where $\theta^*$is the angle by which the desired path is parameterised, is complimented by the 2-dimensional rotational matrix as shown in the following equations,

$$R_p(x_t) = \begin{bmatrix} cosx_t & -sinx_t \\ sinx_t & cosx_t \end{bmatrix} \tag{5:13}$$

$$\varepsilon \; = \; R_p(x_t)(p - p_d(\theta^*)) \tag{5:14}$$

Disturbance Model:

The wave disturbance is defined according to the *Pierson-Moskowitz* theory (Pierson and Moskowitz, 1964). This tells us that for "fully developed seas", waves are settling into equilibrium given a steady wind that blows over some time period. We are mainly concerned on the 2nd order waves. They range between 0.5-1.08 rad/sec in frequency and their peak values are: 1600 T (μω≈±π/4), 10000 T (μω≈±π/2), 400000 Tm (μω≈±π/4) in the surge sway and yaw directions respectively and for "heavy seas".

This falls under the category of "rapidly varying wave exiting forces", resulting into high and aggressive control signals, if transmitted within the loop, and can potentially saturate the thrusters. One of the critical design points would be to prevent these signals from feeding directly back to the controller. As mentioned before, in the *PID* control case this was achieved with the using filtering. In the *NGMV* case we will select the frequency range of the weightings such that the controller itself indirectly acts to filter out this component and obtain the same result. For simulation purposes a wider frequency range is considered to fully capture the effect of the wave drifting forces. The disturbance signal is defined as white nose driven into a 2nd order transfer function shown in Equation (5:193). This holds an acceptable approximation for a deterministic signal of this nature.

$$R_{h_i}(\mathrm{s}) \; = \; \frac{a_i s^2 + 2\zeta_{1_i}\omega_{h_i} s + \omega_{h_i}^2}{b_{h_i}(s^2 + 2\zeta_{2_i}\omega_{h_i} s + \omega_{h_i}^2)} \tag{5:15}$$

where *i* = 1, 2, 3 for the surge, sway and yaw respectively. The frequency response we are trying to simulate is shown in Figure 5:3 with the central frequency fixed at 0.5 rad/sec. In simulation $\omega \in [0.1, 2] rad/s$ and $\zeta \in [0.1, 1.2]$ depending upon the roughness of the sea state.

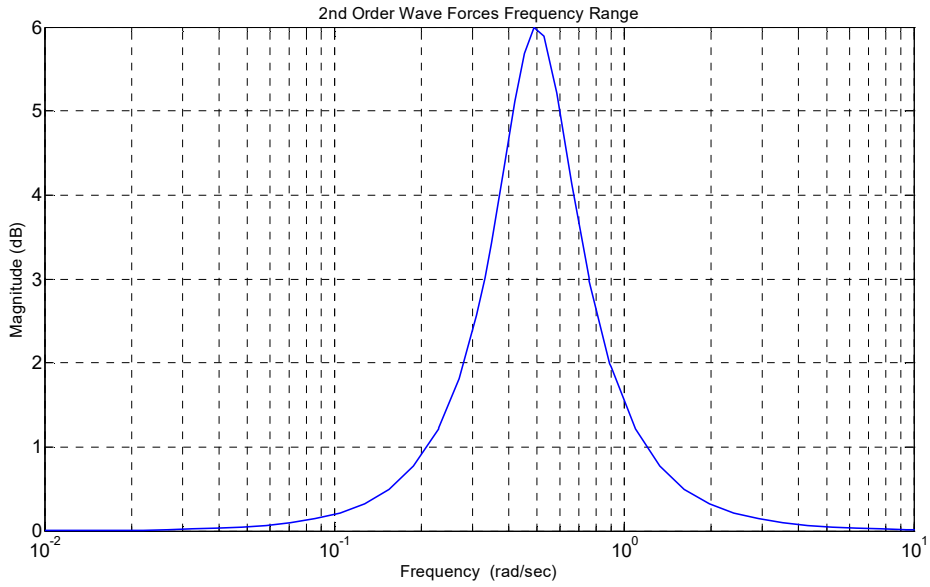


Figure 5:3 Wave disturbance frequency range centred at 0.5 rad/sec.

5.3 Control System Description

The control design is based on the configuration of a basic guidance system as depicted in Figure 5:4. The guidance system is responsible to generate the error between the measured/estimated ship position and the desired one, in relation to the *NED* frame. The error is then converted to body frame coordinates and fed into the controller. The control actions (i.e. thrust force commands) are then computed and drive the actuators (thrusters). Subjected to environmental disturbances the vessel will move towards a new set of coordinates. Velocities are then usually measured by a global navigation satellite system, in combination with motion sensors like accelerometers and gyros and finally out of these measurements the new set of coordinates is estimated relative to the *NED* frame. As the material in this thesis is more concerned about the control problem, we will assume that full measurements are available hence avoiding the estimation problem.

The *PID* and the *NGMV* controllers use slightly different configurations i.e. the *PID* is set in combination with a filter to remove high order waves from the loop, whereas in the *NGMV* case the frequency response combination of the error and the control weightings achieves the same result. Moreover the *NGMV* controller, due to utilisation of an internal model of the process it requires the yaw measurement.

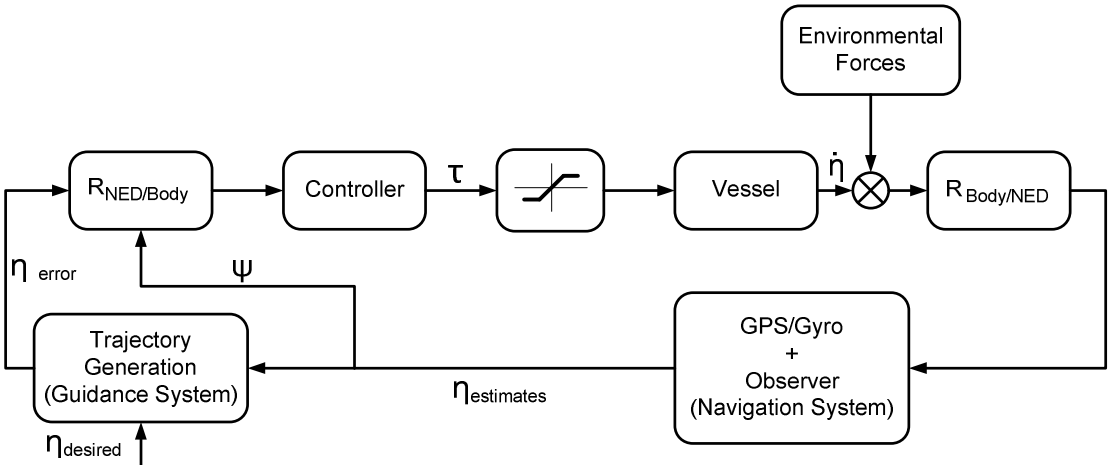


Figure 5:4 Manoeuvring system control loop configuration.

5.3.1 NGMV Controller Design

As a nonlinear control design scheme the *NGMV* uses the plant nonlinearities directly instead of applying linearization techniques as commonly used. Almost similar to the internal model control concept (Grimble, 2005), it uses a delay free exact model of the plant, inside the controller, and it penalises the error and the control signal, using weightings, to eliminate any mismatch with the actual plant and cancel it out. For this application in particular the state-space algorithm formulation was used. Although on the *NGMV* theory the system breaks down into a linear and a nonlinear part, in this case the first was modelled as a scalar component to allow full integration of the second, which is regarded as a black box.

The *DP* controller acts upon the position error vector (in meters) and outputs the optimal vector of thrust forces (in Newton) to the thrusters. The thrusters produce the equivalent velocities in *surge, sway* and *heave* to bring the ship back to the desired position. The disturbance model polynomial is formulated and embedded within the controller. For this experiment the *black-box NGMV* formulation is utilised i.e. the full nonlinear ship model including the thrusters saturation nonlinearities is placed within the general nonlinear operator W_{1k} whereas an unity *3x3* matrix (size equal to the number of *I/O* channels) is used for the linear subsystem. This gives the flexibility to utilise the control in the case where very little is known about the system. Therefore the *NGMV* internal model consists of the following subsystems.

NONLINEAR SUBSYSTEMS

Nonlinear Plant Subsystem (*black-box* operator):

$$W_{1k} \rightarrow M\dot{v} + C(v)v + D(v)v = \tau \tag{5:16}$$

For these experiments the above equation was simplified to include velocities dependency on only one matrix (i.e. the *C* centrifugal matrix). Although at this stage of the problem a varying controller gain formulation is not considered, the *NGMV* holds an advantage for including this nonlinear characteristic within its control law. Examples of the nature of these variations in terms of elements of the *C* matrix can be seen in the following figures.

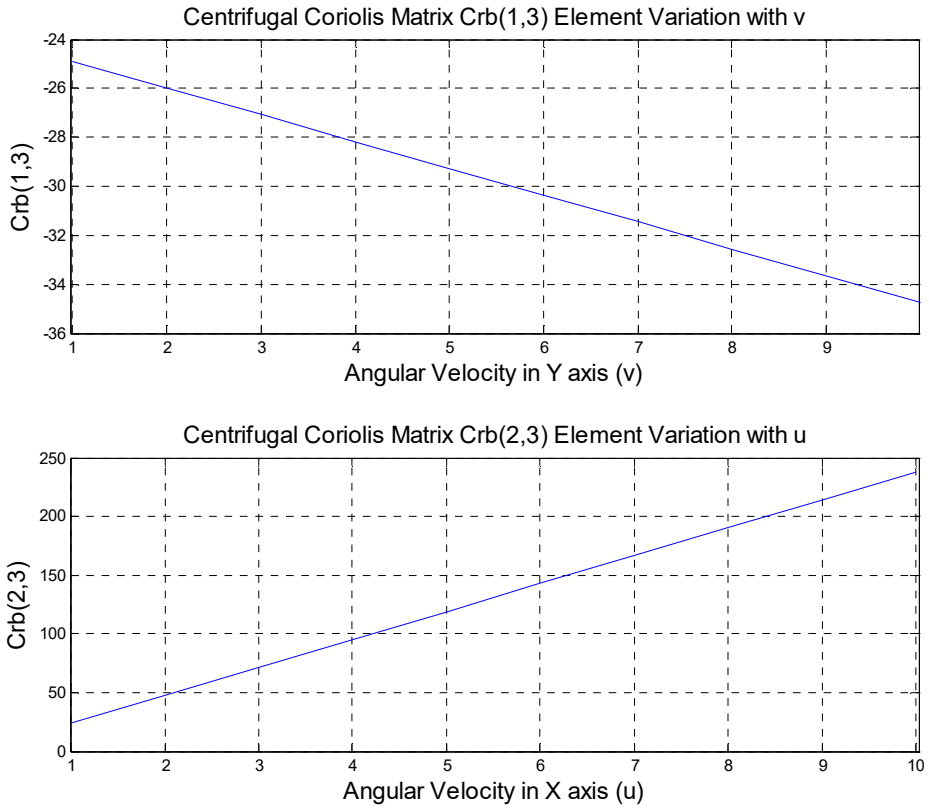


Figure 5:17 Linear variation with angular velocities of Coriolis matrix elements.

The W_{1k} nonlinear operator also includes nonlinear input-output characteristic of an open-water thruster relating input to output thrust force as seen in the following figure and also saturation constraints as seen in the following relations.

$$|\tau_X, \tau_Y| \leq 50N$$
$$|\tau_N| \leq 37.5N \tag{5:17}$$

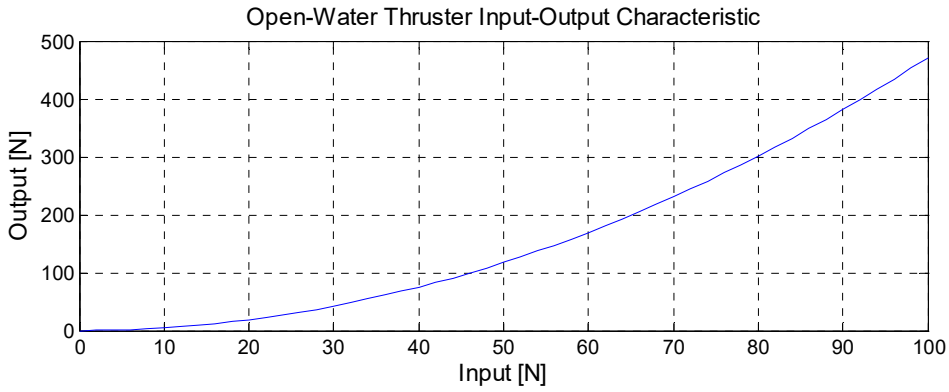


Figure 5:18 Nonlinear thruster input-output characteristic.

LINEAR SUBSYSTEMS

Linear Plant Subsystem:

$$W_0 \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:18}$$

Reference Subsystem:

The reference subsystem was modelled as a *3x3* discrete integrator *TF* matrix, appropriate to approximate the nature of setpoint trajectories.

$$W_r \rightarrow \frac{0.5}{1 - 0.8z^{-1}} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:19}$$

Disturbance Subsystem:

The disturbance subsystem was modelled as a *3x3* discrete *TF* matrix according to the second order process described in the disturbance definition section.

$$W_d \rightarrow \frac{1.002 - 1.995z^{-1} + 0.9925z^{-2}}{1 - 1.995z^{-1} + 0.995z^{-2}} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:20}$$

NGMV Weighting Definition:

The error weighting P_c was selected with respect to the frequency characteristics of the disturbance and has the following discrete *TF.*

$$P_c \rightarrow \frac{1.001 - 1.995z^{-1} + 0.9938z^{-2}}{1 - 1.995z^{-1} + 0.995z^{-2}} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:21}$$

The control weighting F_{ck} is defined as the following discrete lead term.

$$F_{ck} \rightarrow 1.5 - 1.05z^{-1} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5:22}$$

The frequency responses of both the error and control weighting *TF*s against the wave disturbance can be seen in the following figure.

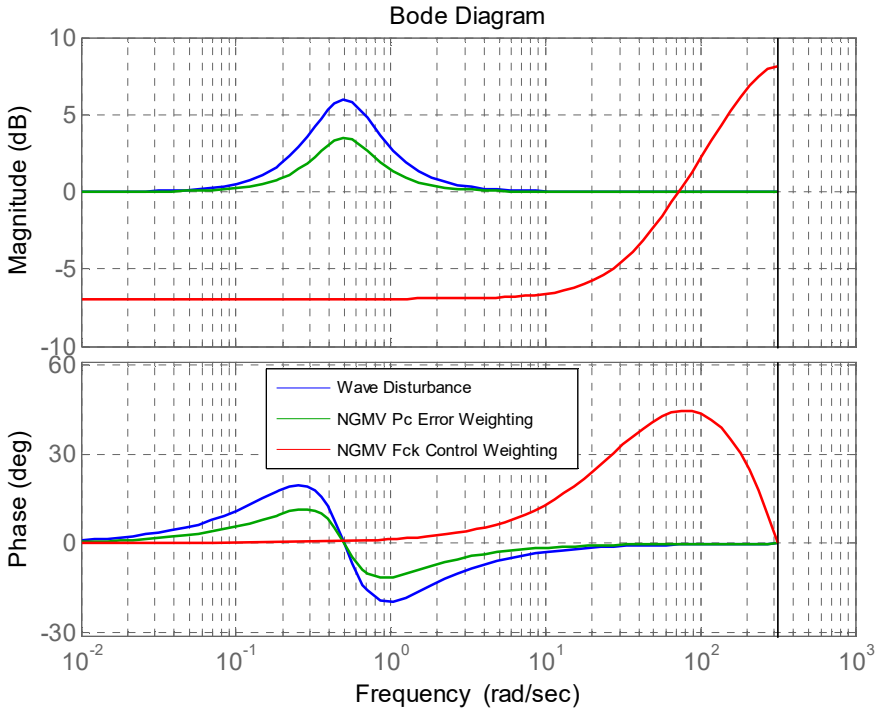


Figure 5:19 Manoeuvring system control loop configuration.

Notch Filtering:

In experiments presented in Figures 31-32 a Notch filter was included in the *PID* feedback measurement path in an attempt to omit higher frequencies than the ones the controller and actuators are intended to act upon which are the main ones responsible for the displacement

of the ship. The Notch filter simply has the reverse characteristics of the unwanted frequencies as shown in the following Bode plot.

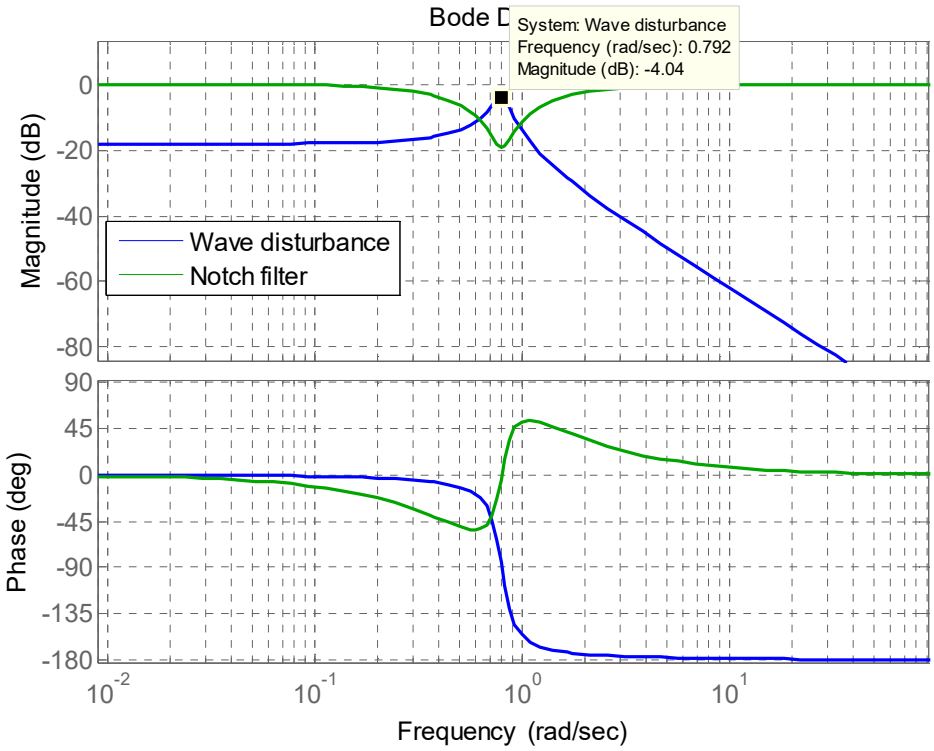


Figure 5:20 Notch filter and disturbance frequency response.

5.4 Simulation Results for the Basic NGMV

5.4.1 Reference Tracking Performance

The reference trajectory consists of position coordinates *x*, *y* and r on the *NED* frame which are given in meters for the surge/ sway and degrees for the yaw mode. The thrusters force commands are given in [N]. For the 1st set of tests, commands were employed only to one of the three modes keeping the rest at zero to examine the equivalent responses in isolation. However there is a strong coupling between the sway/yaw modes and the rest, shown in Figure 5:23 and Figure 5:25 (i.e. varying the sway and/or the yaw effects variations in the other channels as well). In surge control there is no coupling with the other modes so they effectively remained close to zero. Their responses are omitted from the surge control trial. In all experiments importance was given in the trade-off between tracking performance and regulation of control signals. Tuning for both the *NGMV* and different *PID* configurations was carried out with respect to this trade-off.

It can be seen in all results that the *NGMV* explicitly considering the nonlinearities and *MIMO* interactions provide better tracking with more conservative control action. Moreover it can be seen in 5:13 that moving to a different operating point results into performance deterioration for the baseline controller (obviously needing re-tuning) whereas the *NGMV* maintains performance using the same tuning parameters.

Scaling: It is important to note here that as the marine vessel model used in this application was that of a small experimental mock-up of an actual ship the magnitudes of disturbance signals were scaled down accordingly. Hence for example only less than a meter variations are observed in the dynamic positioning results in the presence of disturbances.

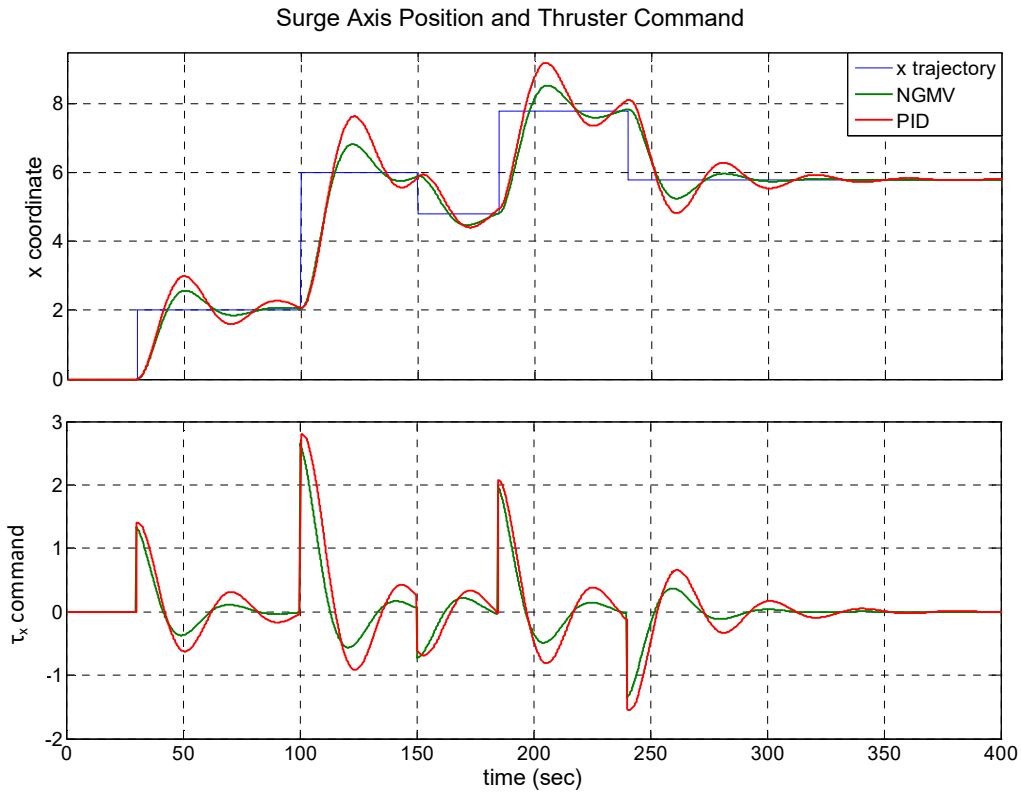


Figure 5:21 Position reference tracking along the Surge axis (no effect on Sway and Yaw motions). Coordinate positions are given in [m] whereas thrust commands in [N].

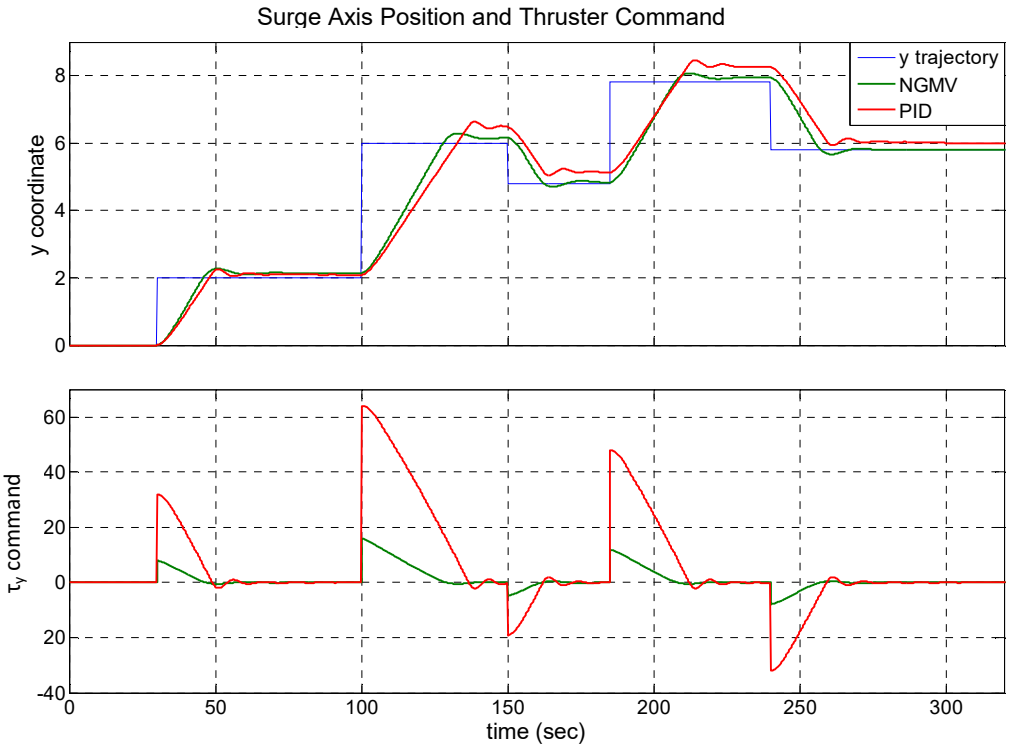


Figure 5:22 Position reference tracking along the Sway axis (has effect on Surge and Yaw motions). Coordinate positions are given in [m] whereas thrust commands in [N].

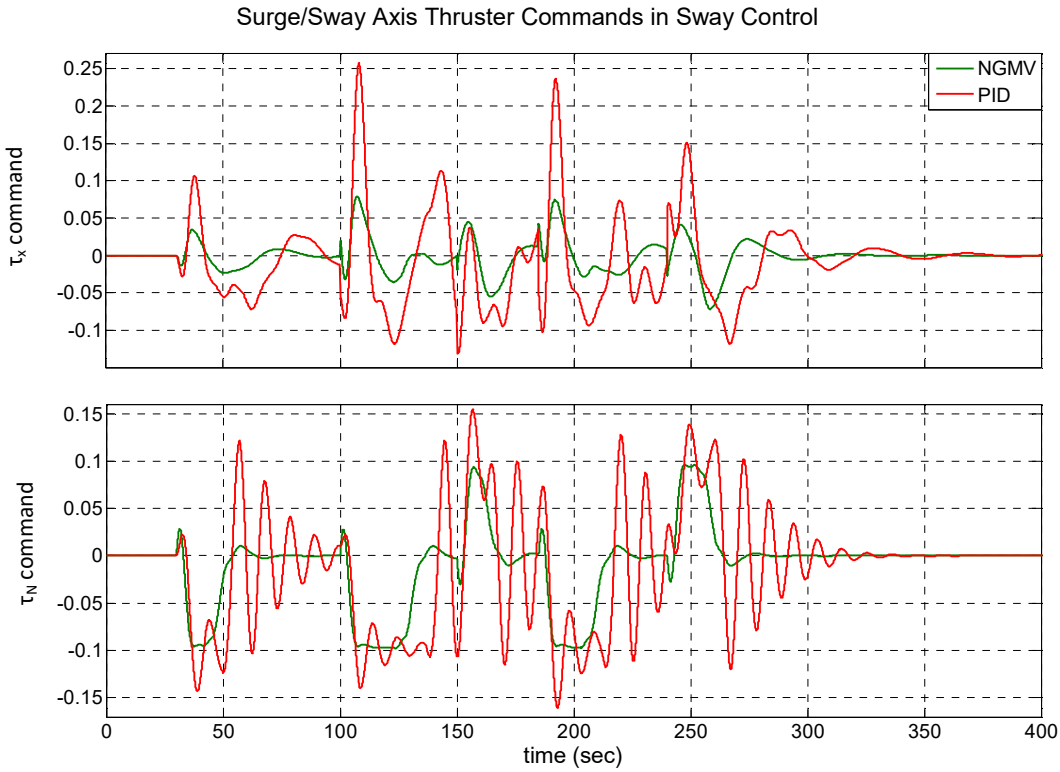


Figure 5:23 Effect of motion along the Sway axis onto Surge and Yaw thruster commands. Thrust commands are given in [N].

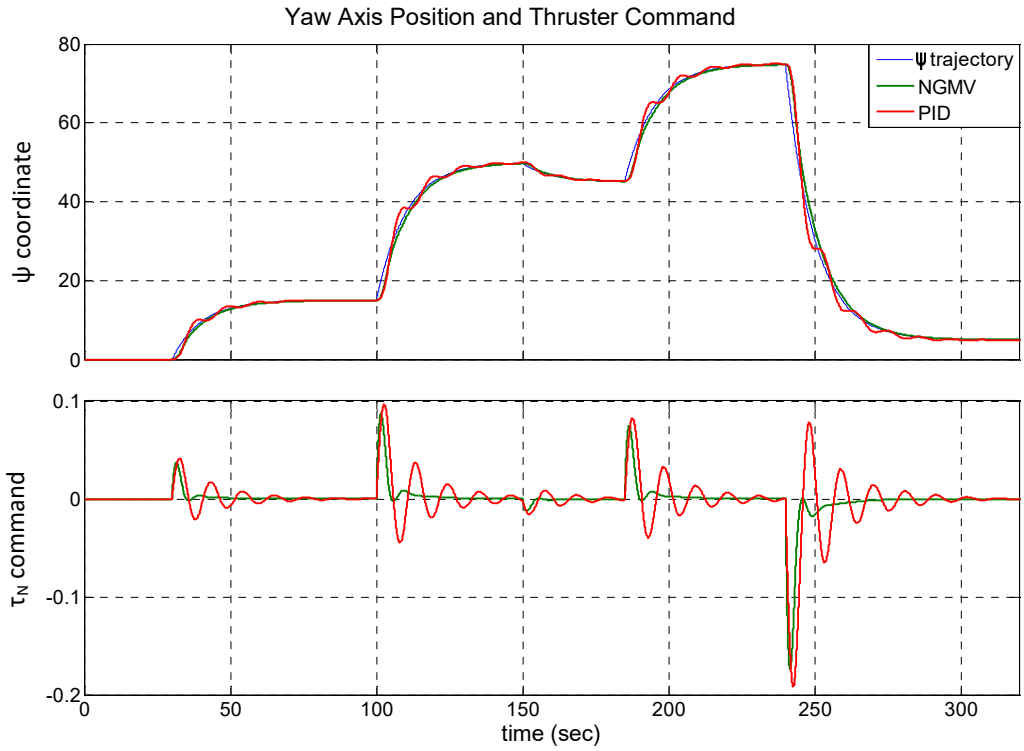


Figure 5:24 Rotation reference tracking around the Yaw axis (has effect on Surge and Sway motions). Coordinate positions are given in [m] whereas thrust commands in [N].

Figure 5:25 Effect of motion around the Yaw axis onto Surge and Sway thruster commands. Thrust commands are given in [N].

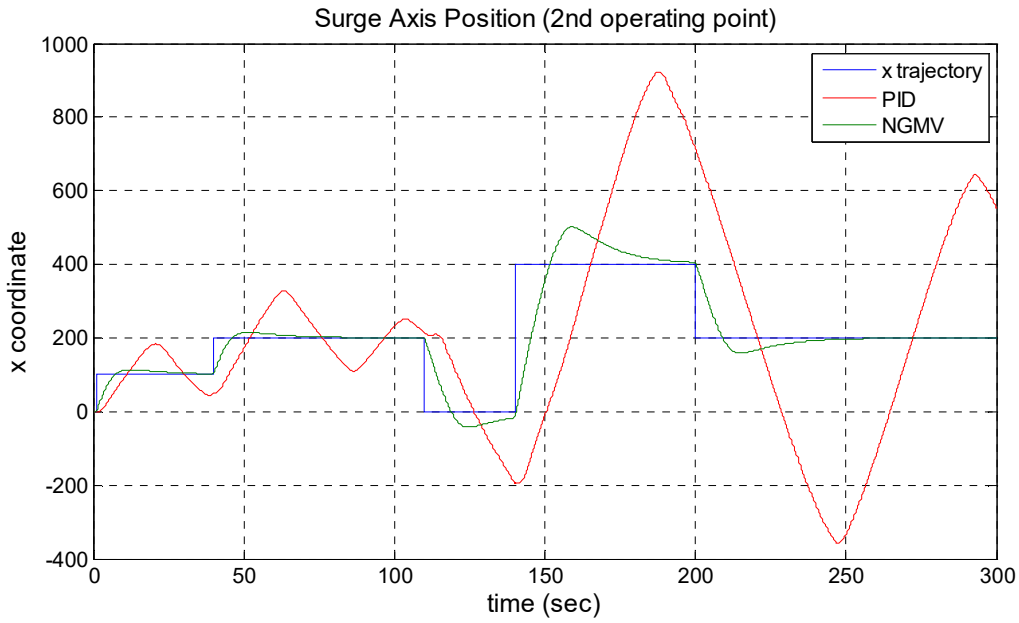


Figure 5:26 Position reference tracking along the Surge axis (2nd operating point). Coordinate positions are given in [m] whereas thrust commands in [N].

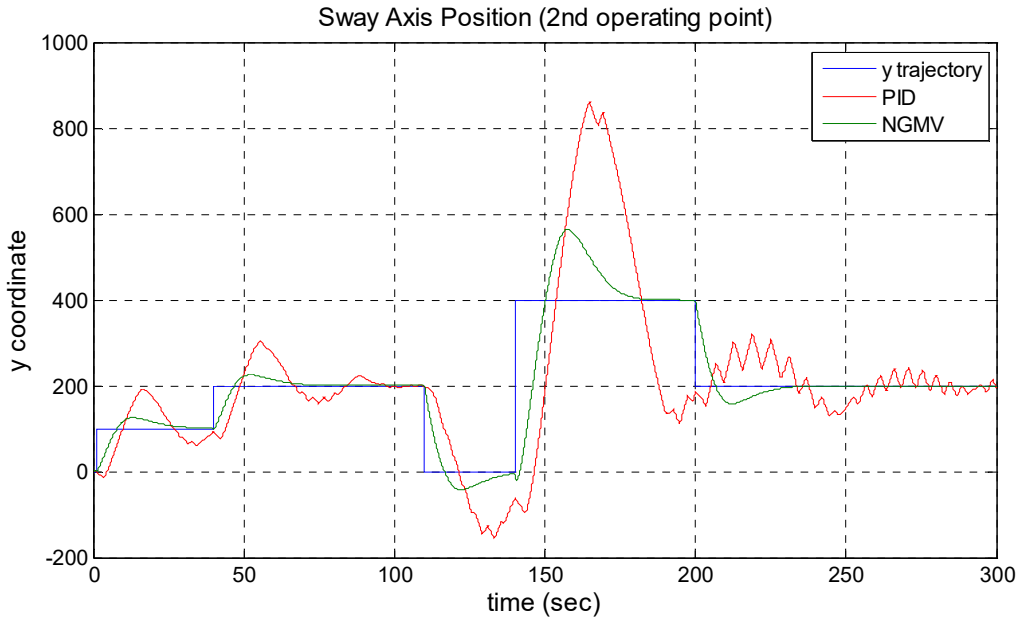


Figure 5:27 Position reference tracking along the Sway axis (2nd operating point). Coordinate positions are given in [m] whereas thrust commands in [N].

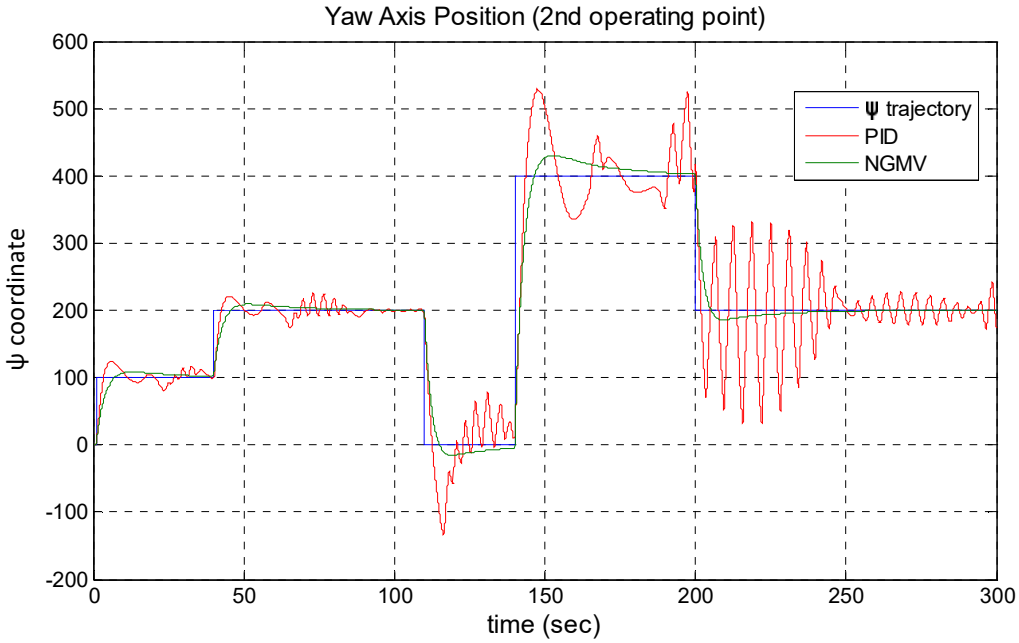


Figure 5:28 Rotation reference tracking around the Yaw axis (2nd operating point). Coordinate positions are given in [m] whereas thrust commands in [N].

5.4.2 Reference Tracking and Disturbance Rejection Performance

In this case, a wave disturbance was applied to the system (as defined in 5.2) along with a series of step setpoint changes to explore the effect upon tracking and disturbance rejection performance of the controllers simultaneously. In the first experiments shown in Figures 5:21-5:30, the baseline *PID* and the *NGMV* were implemented without the application of a *Notch* filter to prevent the unwanted high frequency disturbance component form propagating through the loop. It is observed that the two controllers have similar performance.

The *NGMV* indirectly employs filtering through the appropriate selection of weightings in the frequency domain and therefore yields a slightly smoother response than the baseline controller. In the case that follows, shown in Figures 5:31 and 5:32, *Notch* filtering has been applied to the *PID* loop. Although the unwanted disturbance component has been removed, there is significant lag introduced by the filter adding further complications to tuning and deteriorating tracking performance. The *NGMV* again maintains performance while retaining its initial tuning parameters in the expense however of not completely removing the high frequency component from the loop.

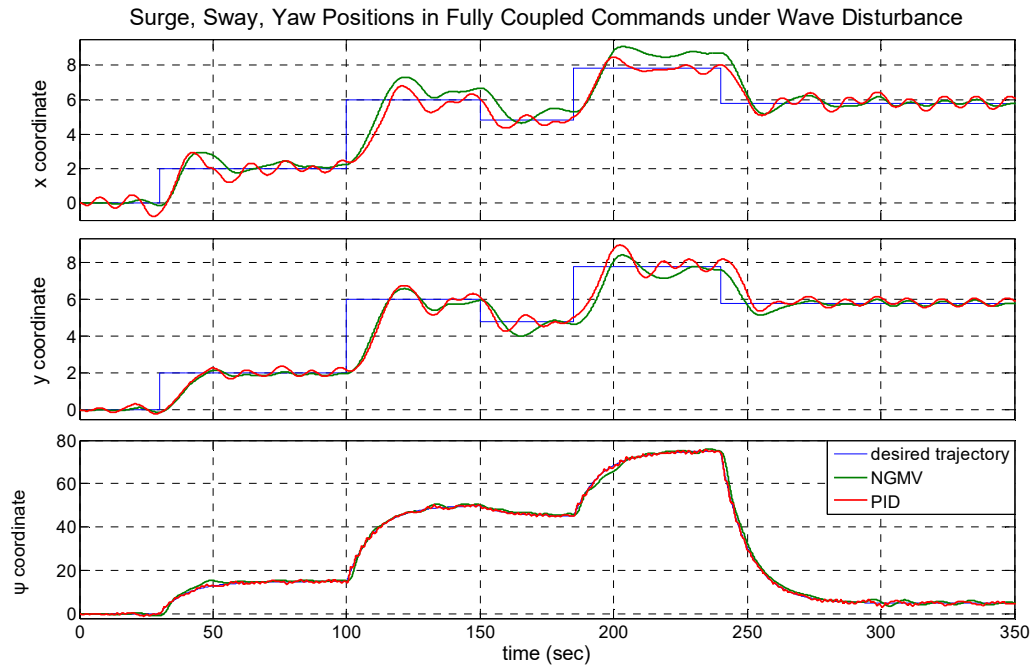


Figure 5:29 Simultaneous reference tracking in all Surge, Sway and Yaw axis under wave disturbance. Coordinate positions are given in [m].

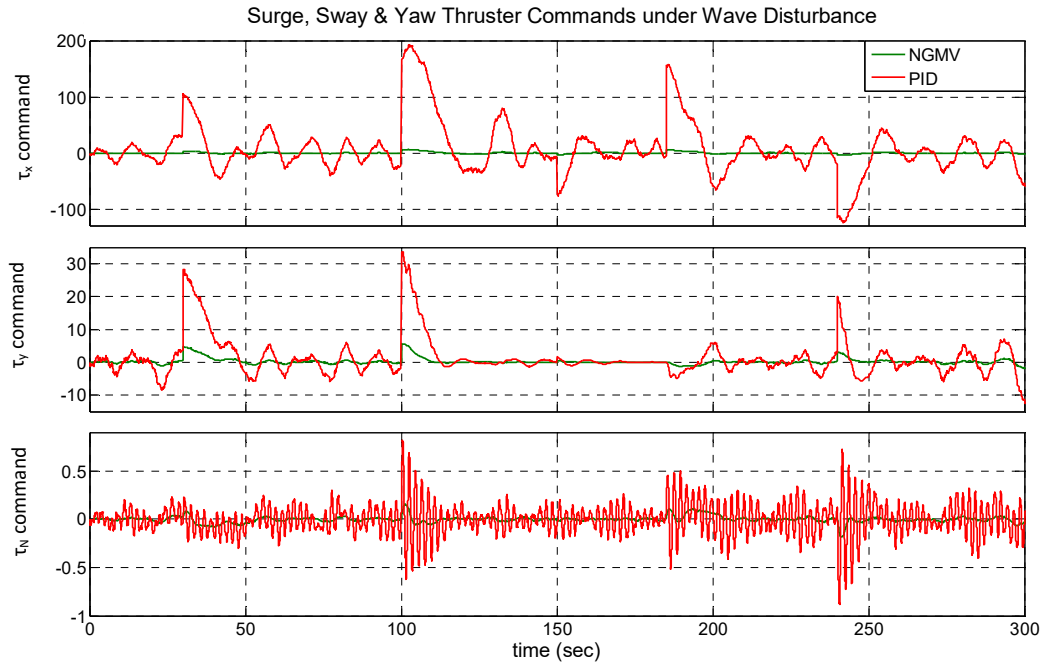


Figure 5:30 Thruster commands in all Surge, Sway and Yaw axis under wave disturbance. Thrust commands are given in [N].

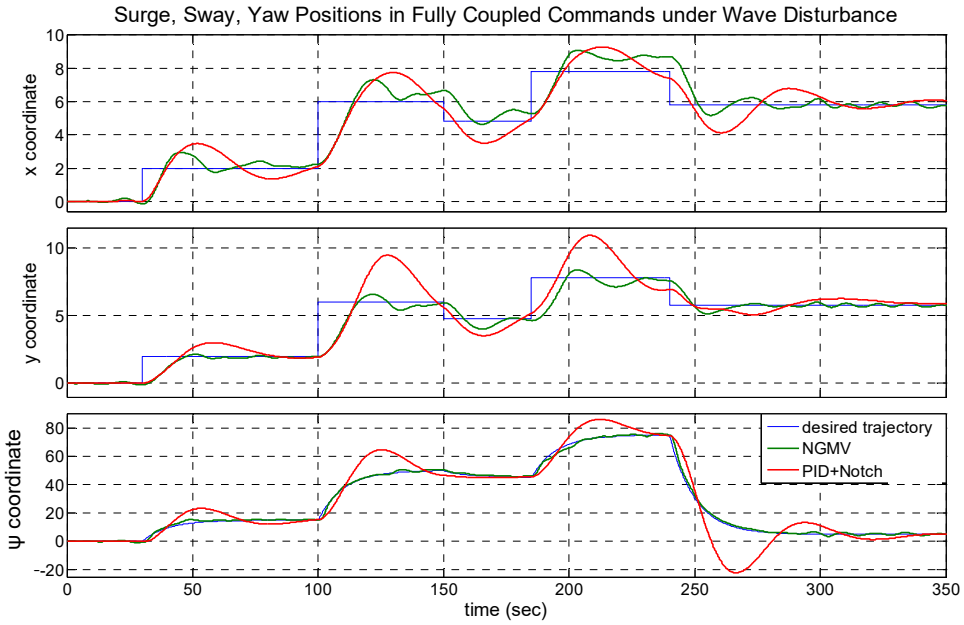


Figure 5:31 Simultaneous reference tracking in all Surge, Sway and Yaw axis under wave disturbance (*PID* combined with *Notch* filtering). Coordinate positions are given in [m].

Figure 5:32 Thruster commands in all Surge, Sway and Yaw axis under wave disturbance (*PID* combined with *Notch* filtering). Thrust commands are given in [N].

5.4.3 Dynamic Positioning – Disturbance Rejection Performance

For this scenario control performance was tested at maintaining the position of the ship at a nominal point under the influence of wave disturbance.

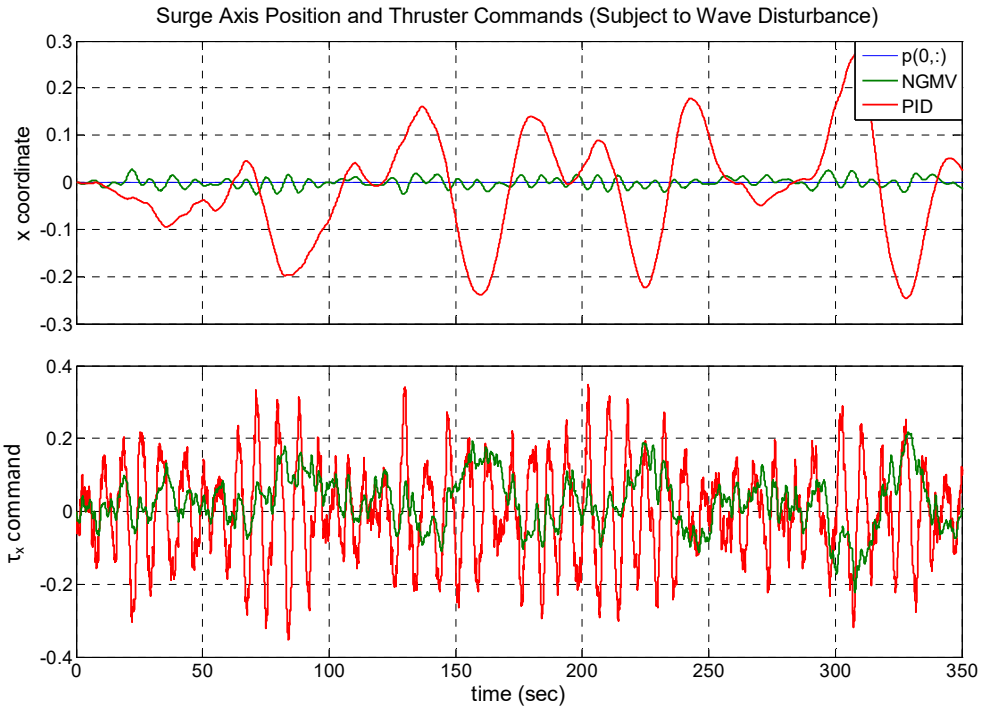


Figure 5:33 Surge position reference at $p(0,0,\psi)$ and resulting thruster command under wave disturbance. Coordinate positions are given in [m] whereas thrust commands in [N].

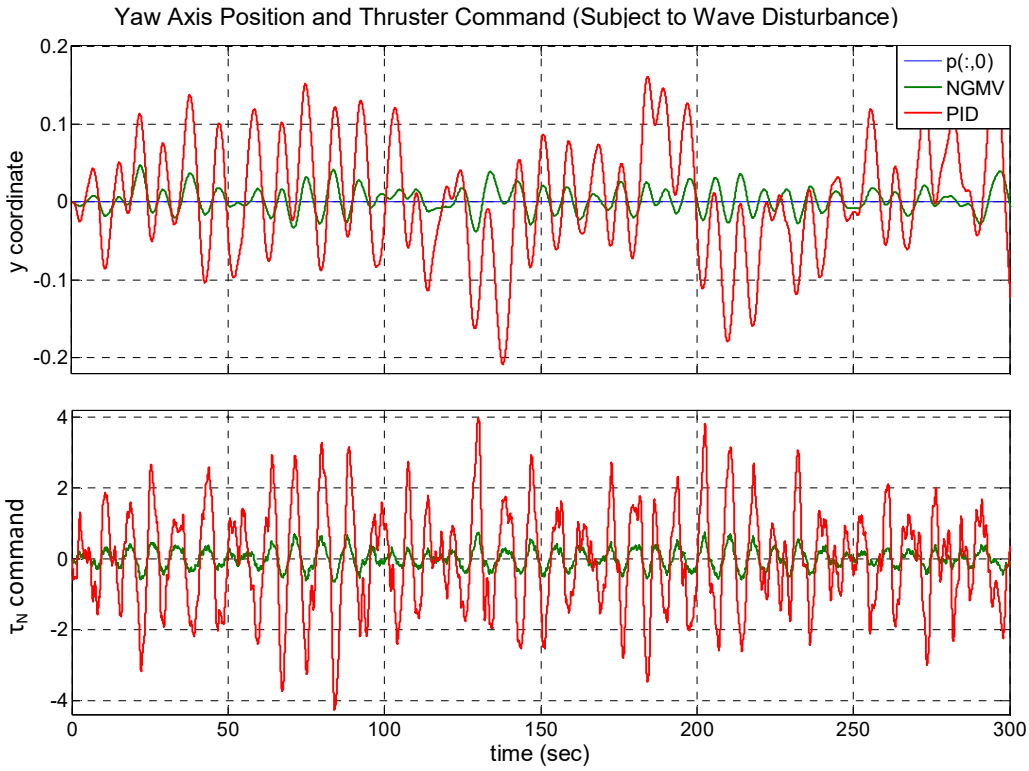


Figure 5:34 Sway position reference at $p(0,0,\psi)$ and resulting thruster command under wave disturbance. Coordinate positions are given in [m] whereas thrust y commands in [N].

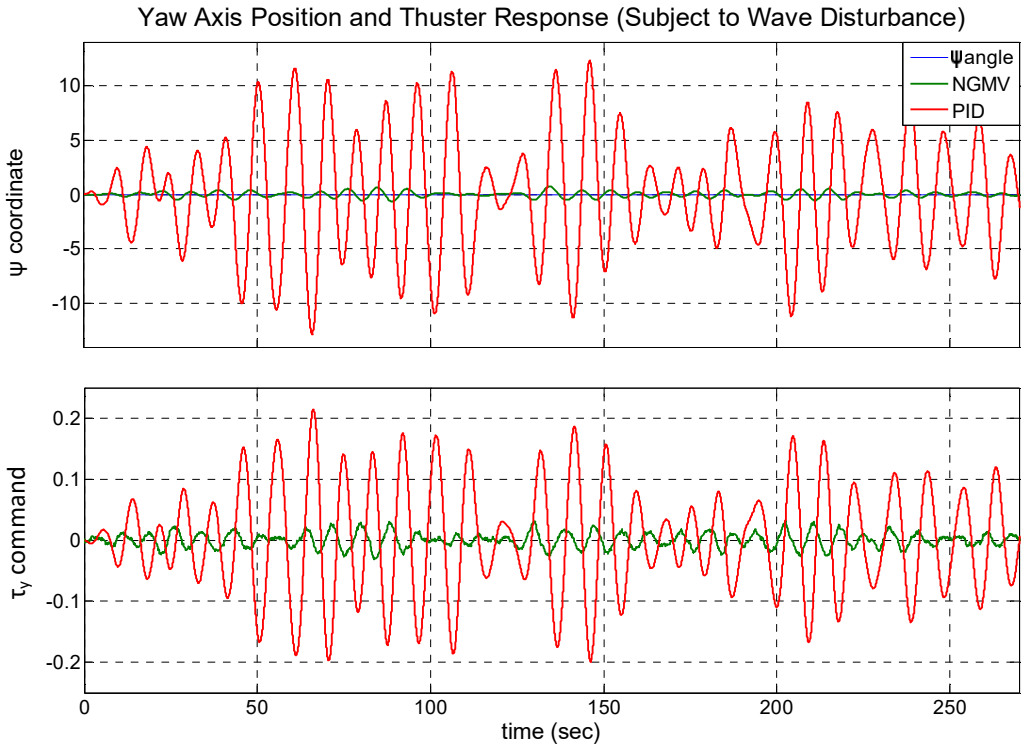


Figure 5:35 Yaw position reference at $p(0,0,\psi)$ and resulting thruster command under wave disturbance. Coordinate positions are given in [m] whereas thrust commands in [N].

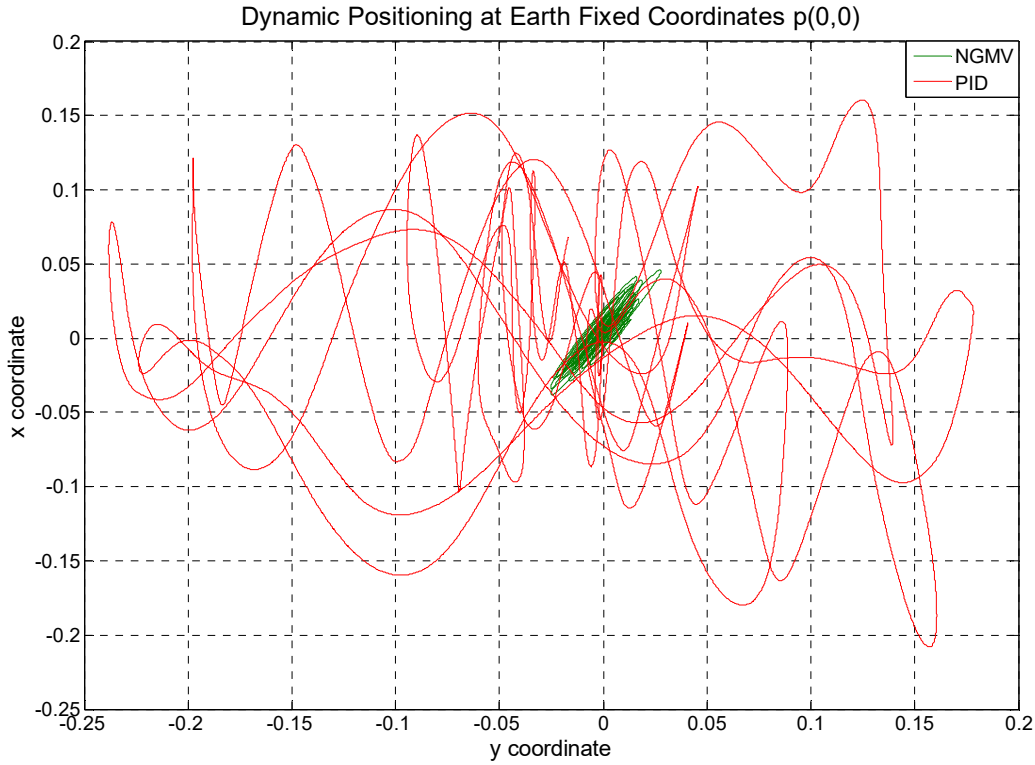


Figure 5:36 *X,Y* axis coordinate response for dynamic positioning of the vessel under wave disturbance for both the *NGMV* and *PID* controller. Coordinate positions are given in [m].

Similar to the case explored in Section 5.4.2, *Notch* filtering has been applied here as well, directly in the *PID* and indirectly (i.e. through weighting selection) in the *NGMV*. The latter demonstrates significant performance improvement in all coordinates and control channels

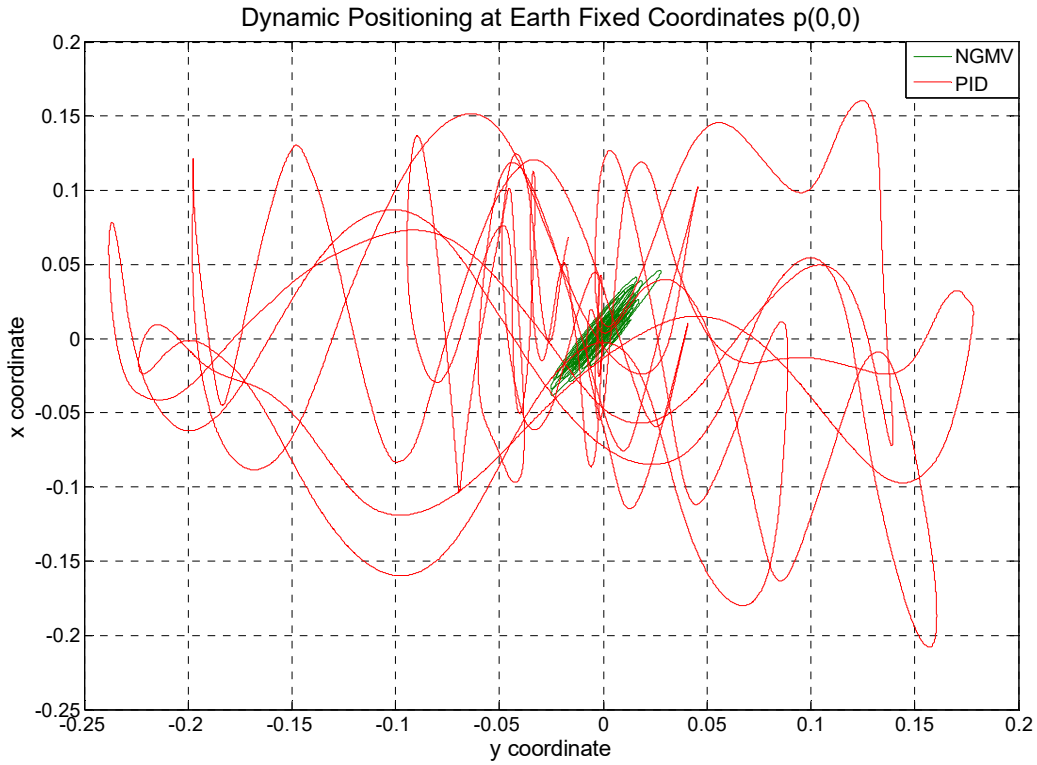


Figure 5:37 shows the actual trajectory of the ship around the nominal point in the Earth coordinate system for both controllers.

5.5 Dynamic Positioning Using LPV-NGMV Control

In this application section the *LPV-NGMV* control approach is used for the dynamic positioning of the experimental marine vessel model (*Cybership II*) described in Section 5.2. The intention here is to use simulated scenarios to explore potential benefits the *LPV* formulation of the algorithm has over the nominal state-space version. The general problem description tested in simulation is summarised in the following diagram.

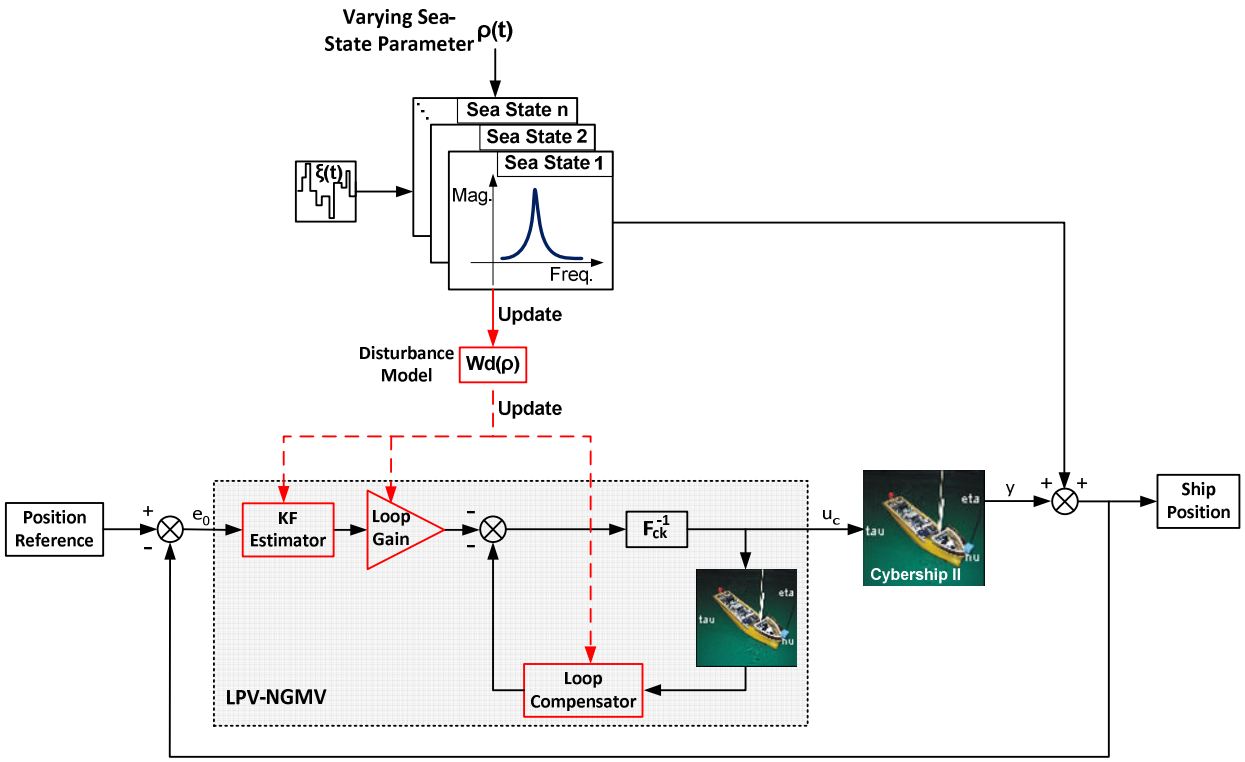


Figure 5:38 Control configuration of the *LPV-NGMV* controller for *DP* under varying wave disturbances. Varying elements within the controller structure are shown in red outline.

The varying parameter *ρ(t)* in this case is the determining factor that affects change in sea state as captured by variations in the parameters of a 2nd order wave model such as the damping ratio, natural frequency, amplitude etc. Similar to previously the wave disturbance is captured by the wave model driven by white noise (*ξ(t)*). For the purpose of this simulation it is realistic to assume that transition between the different sea states happens at a much slower rate than the controller dynamics, therefore it is meaningful to examine and compare performance of the two controllers once the new sea-state has being reached. The main objective here, as with the previous *DP* simulation example, is to maintain the ship position at *p(0,0,0)* earth coordinates in the presence of (this time) varying wave disturbance. Five sea states are defined for this simulation as follows.

$$R_{sea\ state\ 1}(\mathrm{s}) = \frac{0.04s}{s^2 + 0.0004s + 0.04} \tag{5:23}$$

$$R_{sea\ state\ 2}(\mathrm{s}) = \frac{0.16s}{s^2 + 0.008s + 0.16} \tag{5:24}$$

$$R_{sea\ state\ 3}(\mathrm{s}) = \frac{0.64s}{s^2 + 0.16s + 0.64} \tag{5:25}$$

$$R_{sea\ state\ 4}(\mathrm{s}) = \frac{2.56s}{s^2 + 1.6s + 2.56} \tag{5:26}$$

$$R_{sea\ state\ 5}(\mathrm{s}) = \frac{7.84s}{s^2 + 6.72s + 7.84} \tag{5:27}$$

Figure 5:39 shows the frequency responses of all five sea states overlaid and parameterised with respect to natural wave frequency.

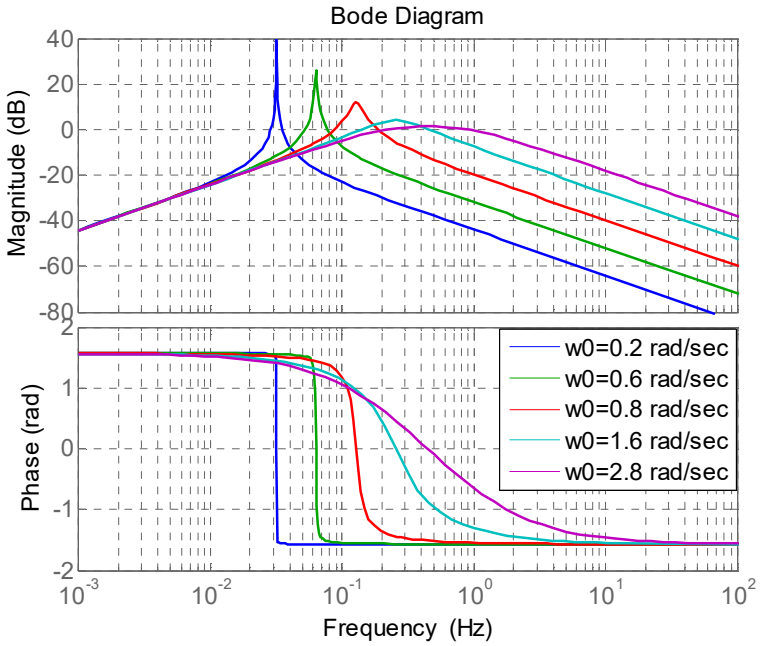


Figure 5:39 Frequency response of the 5 different sea states parameterised by natural frequency ω_0.

5.5.1 Control Design Discussion

The main component within the *LPV-NGMV* controller that varies parameterised by *ρ(t)* is the disturbance model definition W_d whereas the linear plant and reference models remain fixed. The *"black box"* formulation is used here as well where the full nonlinear state-dependent vessel model is contained within the controller and the linear plant subsystem is defined being unity. This is a very interesting formulation, combining nonlinear elements (thruster saturation), state-dependent (varying centrifugal matrix C) within the nonlinear part and at the same time a varying linear component (W_d) to improve knowledge of the changing state within the controller derivation. The following equation shows the continuous time varying state-space structure of W_d which along with the time-invariant W_r linear reference subsystem constitute the augmented *LPV* subsystem of the plant. Figure 5:40 shows the variation range that was used in the following simulation experiments.

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_0^2 & -2\lambda\omega_0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \omega_0^2\end{bmatrix}\Delta u$$
$$y = [0 \quad 1]\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{5:28}$$

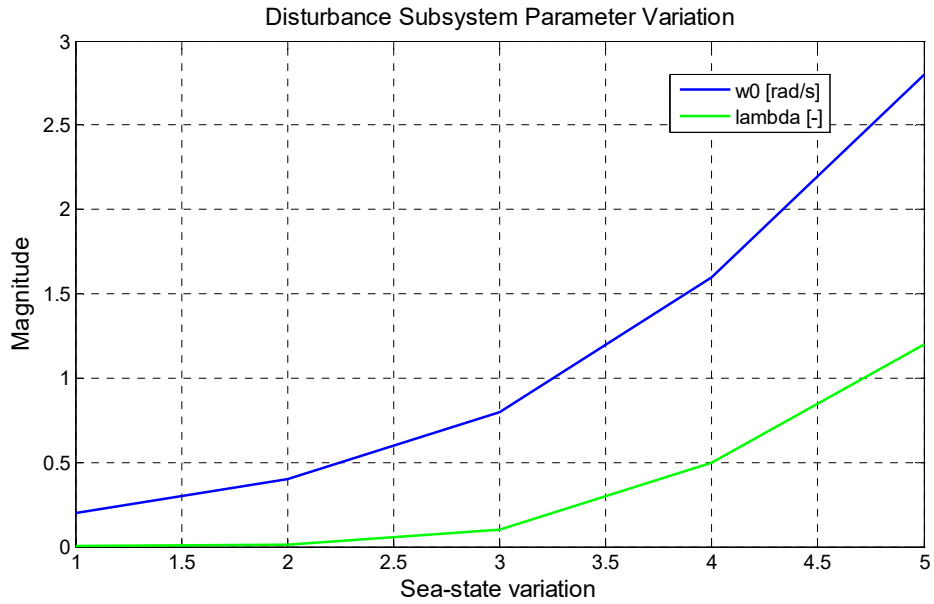


Figure 5:40 Wave natural frequency (ω_0) and damping ratio (λ) variation with respect to sea-state.

The varying $W_d(\rho)$ in turn results into varying controller internal loop gains and *Kalman* filter model as shown in Figure 5:38, where the updated components are signified by the red outline.

The baseline state-space *NGMV* controller used in this example maintains all subsystems fixed throughout the sea states. Another difference between the controllers is that whereas the baseline *NGMV* and *LPV-NGMV* maintain the same P_c weighting based on the *PID* gains definition method, the latter incorporates in the P_c a model of the current wave disturbance to enhance penalty within that particular frequency range. Simulations have shown that by only varying the W_d within the controller formulation yielded a constant improvement in Mean Integrated Squared Error (*MISE*) between 1-2% over the baseline controller. However, scheduling the error weighting in frequency added more to the advantage over the baseline controller as it will be shown in the following results. An example of the Scheduling of P_c with respect to $\rho(t)$ can be seen for sea states 3 and 4 in Figures 5:41 and 5:42 respectively.

Sea State 3 & 4 Controller Weightings:

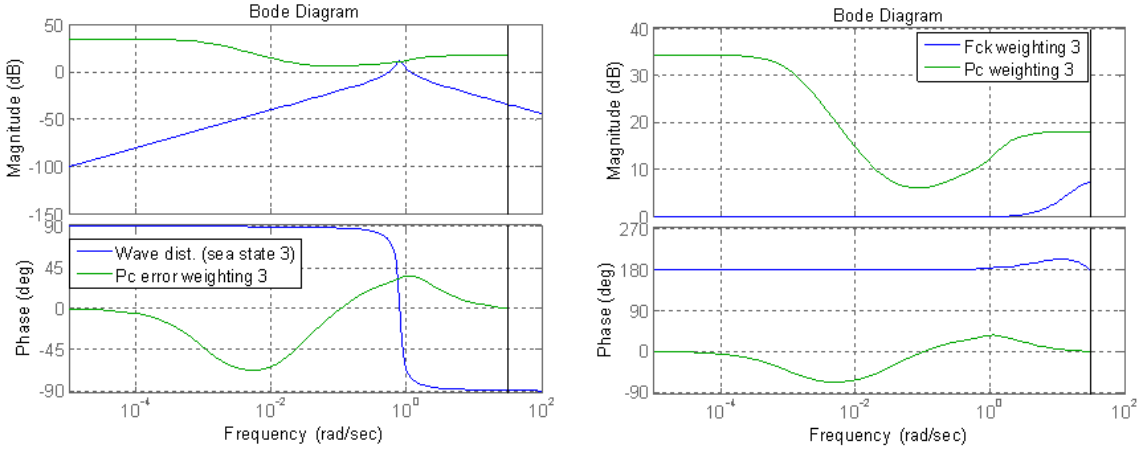


Figure 5:41 Controller error weighting (P_c) against wave disturbance in frequency for Sea State 3 (left) and controller weightings selection in frequency for Sea State 3 (right).

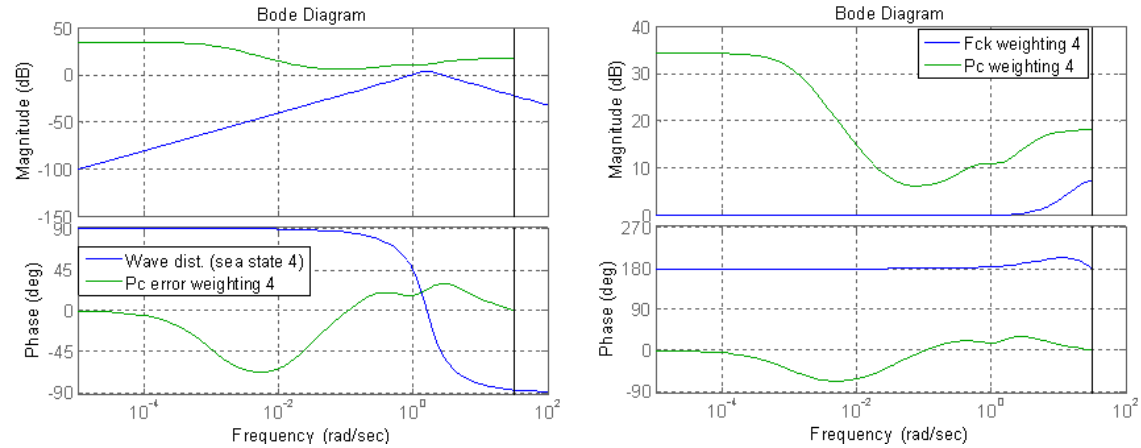


Figure 5:42 Controller error weighting (P_c) against wave disturbance in frequency for Sea State 4 (left) and controller weightings selection in frequency for Sea State 4 (right).

5.6 Simulation Results for the LPV-NGMV

5.6.1 Dynamic Positioning – Disturbance Rejection Performance for Varying Sea State

The two metrics used here to quantify the comparison between the two controllers are the normalised Standard Deviation (*STD*), for the thruster commands, and the Mean Squared Integrated Error (*MISE*) for the three position coordinates.

$$MISE = \frac{1}{N}\sum_{i=0}^{N}(y_{SP\,i} - y_{meas_i})^2 \tag{5:29}$$

where *y* is the positions vector,

$$STD = \left(\frac{1}{N}\sum_{i=0}^{N}(u_i - \bar{u})^2\right)^{1/2} \tag{5:30}$$

where *u* is the thruster force vector.

Ship Earth-frame Coordinates & Thruster Commands Comparison for Sea State 1:

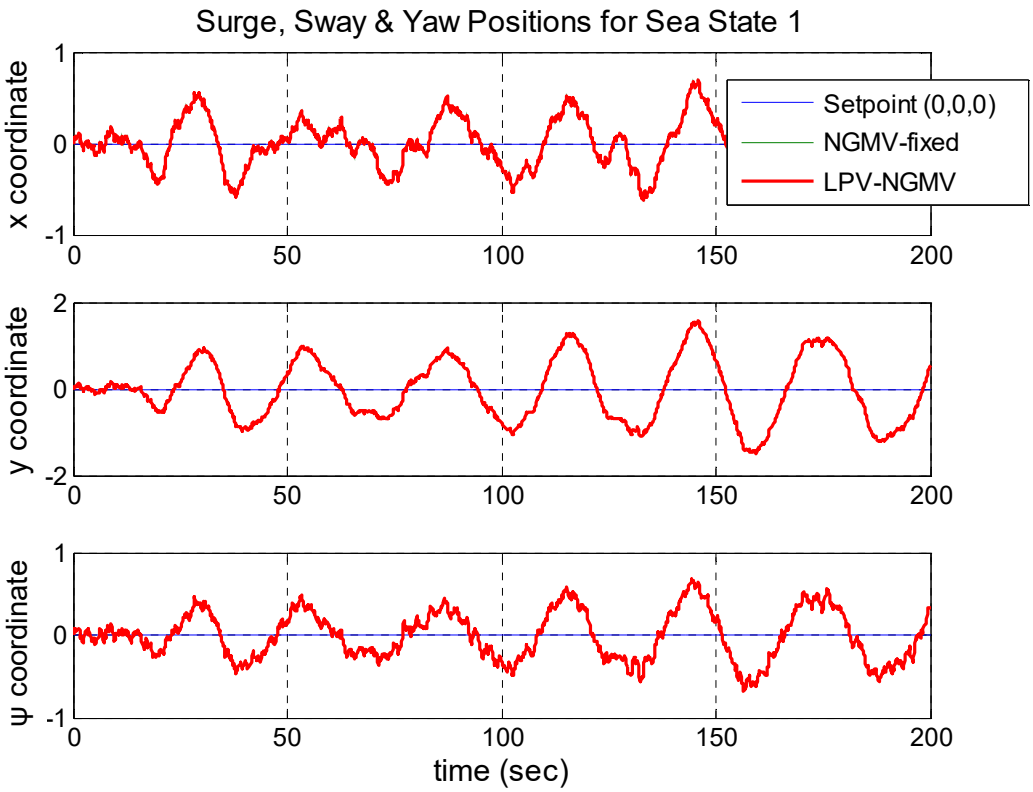


Figure 5:43 Vessel earth coordinate positions in [m] for Sea State 1.

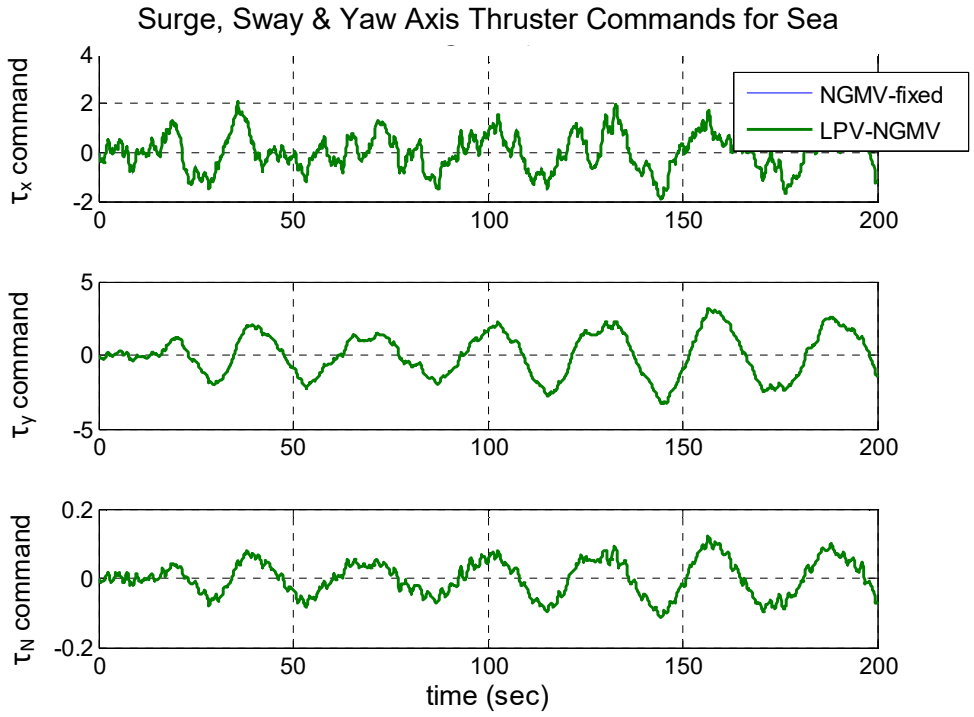


Figure 5:44 Vessel thruster commands in [N] for Sea State 1.

Ship Earth-frame Coordinates & Thruster Commands Comparison for Sea State 2:

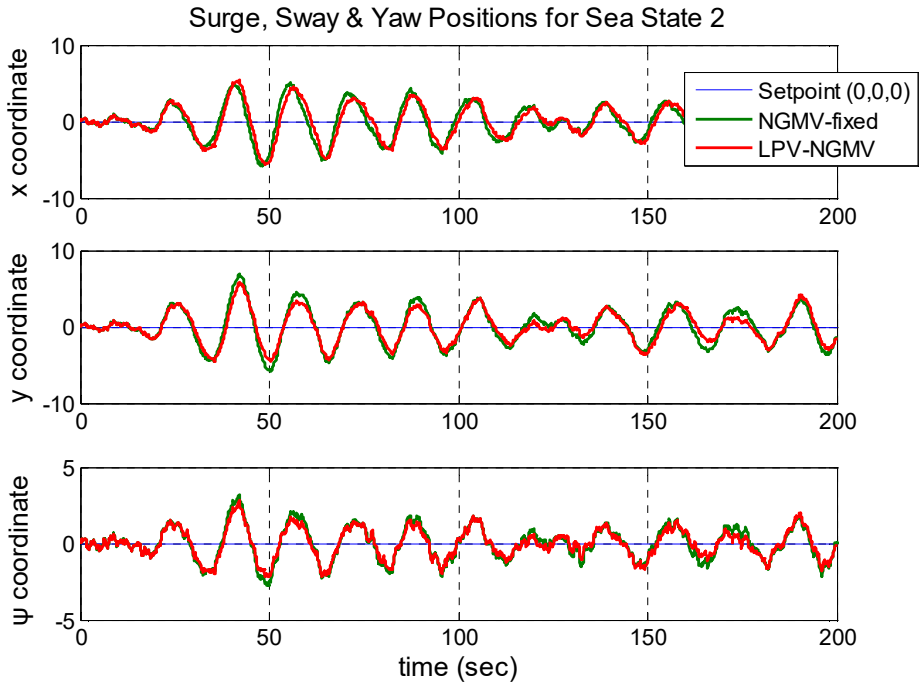


Figure 5:45 Vessel earth coordinate positions in [m] for Sea State 2.

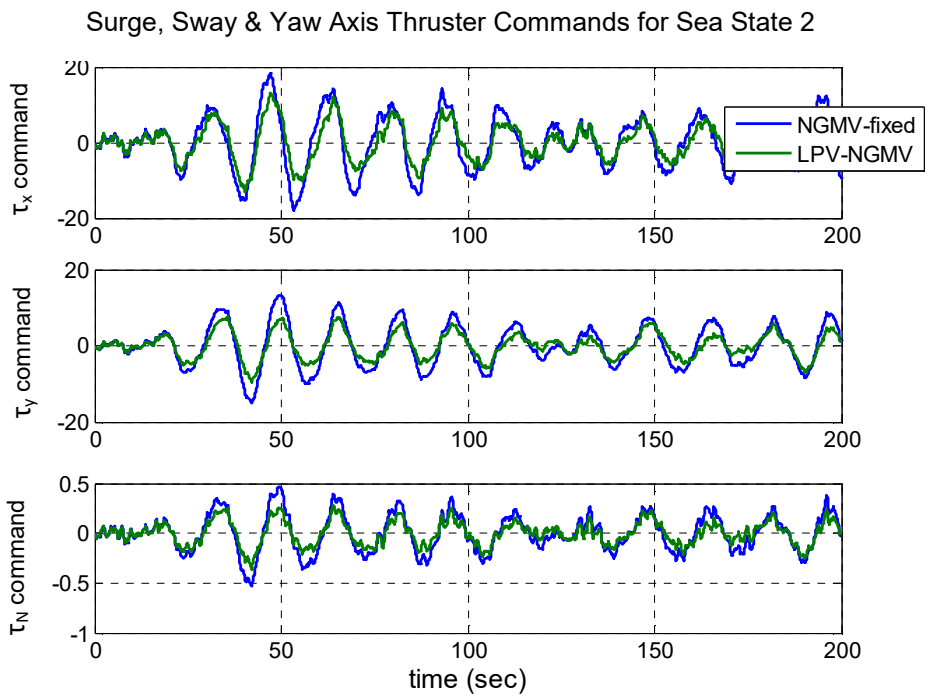


Figure 5:46 Vessel thruster commands in [N] for Sea State 2.

Table 5:2: Sea State 2, quantified controller comparison

Controllers	$\boldsymbol{\theta_{STD}}$ **(norm.)**			$\boldsymbol{Pel_{MISE}}$ **(norm.)**		
	***ux* thrust**	***uy* thrust**	***ur* thrust**	***x* coord.**	***y* coord.**	**ψ coord.**
NGMV	1	1	1	1	1	1
LPV-NGMV	0.6890	0.6152	0.6307	0.9119	0.7533	0.7762

The *LPV-NGMV* yields an improvement in regulation *MISE* of 8.81%, 24.6% and 22.3% for the *x, y* and ψ coordinates respectively relative to the baseline *NGMV*. Furthermore a reduction in control action is achieved, at 31.1%, 38.4% and 36.9% for *x*, *y* and *r* respectively.

Ship Earth-frame Coordinates & Thruster Commands Comparison for Sea State 3:

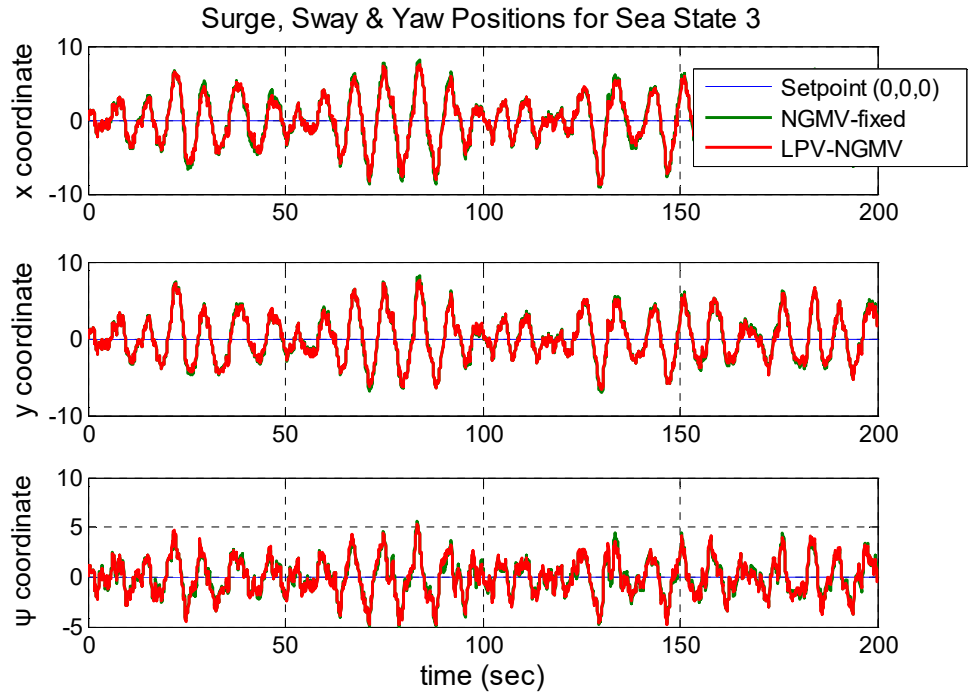


Figure 5:47 Vessel earth coordinate positions in [m] for Sea State 3.

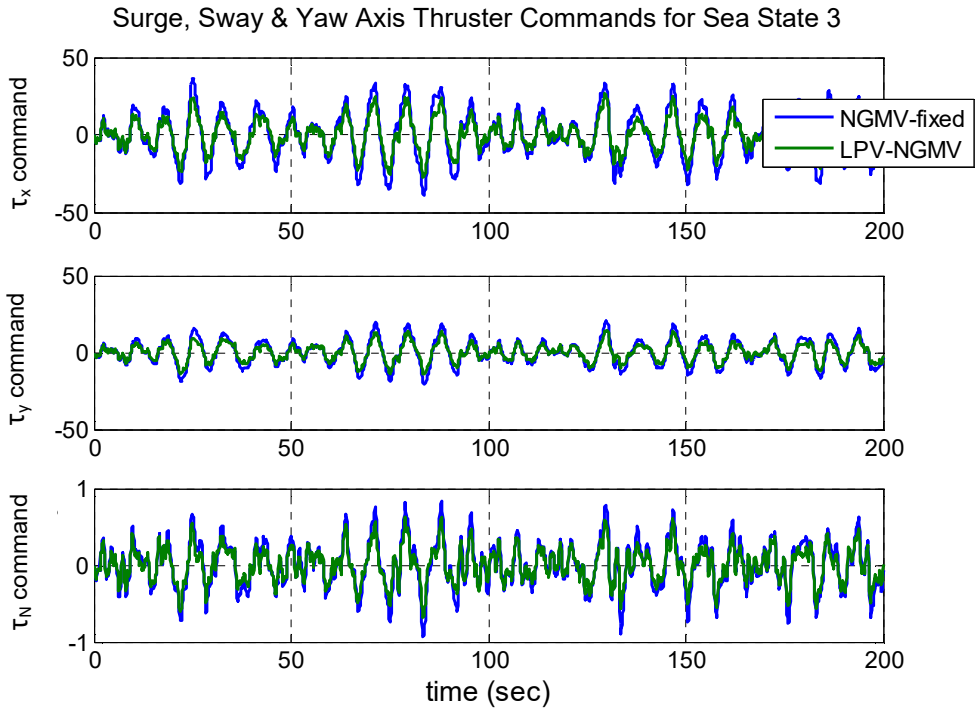


Figure 5:48 Vessel thruster commands in [N] for Sea State 3.

Table 5:3: Sea State 3, quantified controller comparison

Controllers	β_{STD} (norm.)			Pel_{MISE} (norm.)		
	***ux* thrust**	***uy* thrust**	***ur* thrust**	***x* coord.**	***y* coord.**	**ψ coord.**
NGMV	1	1	1	1	1	1
LPV-NGMV	0.6746	0.6976	0.7438	0.8849	0.9282	1.0259

The *LPV-NGMV* yields an improvement in regulation *MISE* of 11.5%, 7.1% for the *x* and *y* coordinates respectively relative to the baseline *NGMV* whilst no improvement is observed for the ψ coordinate. Furthermore a reduction in control action is achieved, at 32.5%, 30.2% and 25.6% for *x*, *y* and *r* respectively.

Ship Earth-frame Coordinates & Thruster Commands Comparison for Sea State 4:

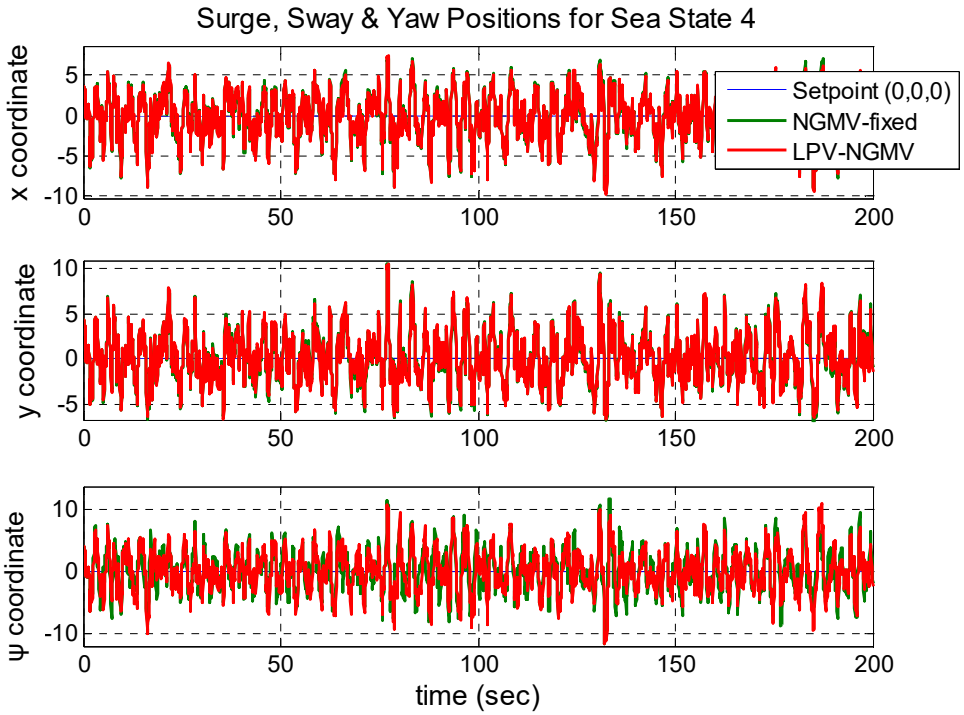


Figure 5:49 Vessel earth coordinate positions in [m] for Sea State 4.

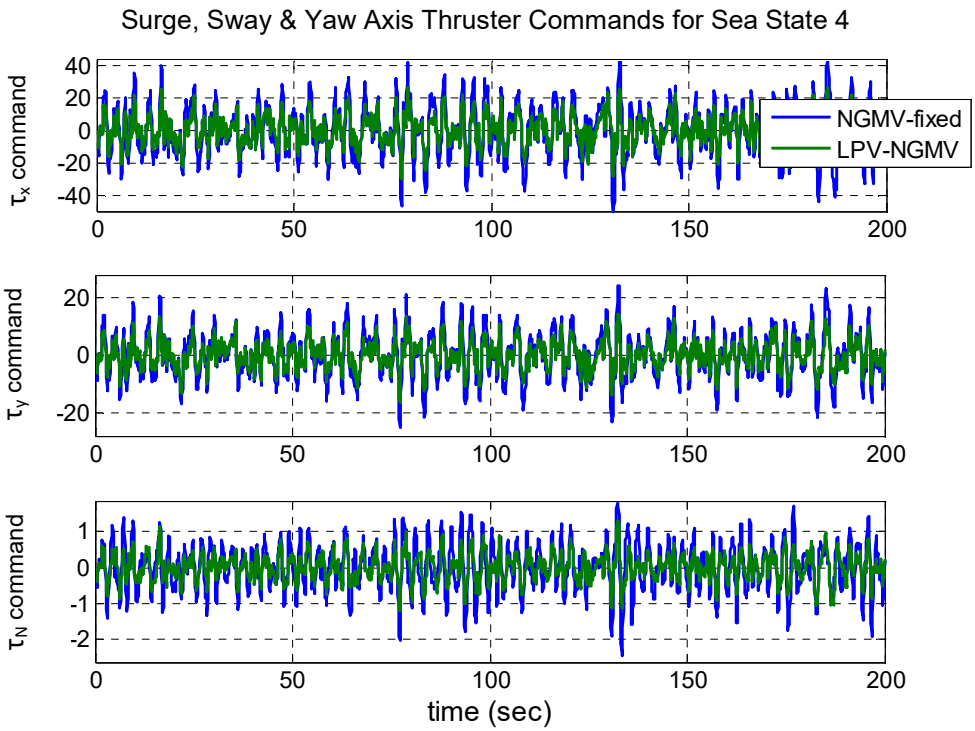


Figure 5:50 Vessel thruster commands in [N] for Sea State 4.

Table 5:4: Sea State 4, quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**			$\boldsymbol{Pel_{MISE}}$ **(norm.)**		
	***ux* thrust**	***uy* thrust**	***ur* thrust**	***x* coord.**	***y* coord.**	**ψ coord.**
NGMV	1	1	1	1	1	1
LPV-NGMV	0.6055	0.6206	0.5754	0.9733	0.9798	0.8947

The *LPV-NGMV* yields an improvement in regulation *MISE* of 2.67%, 2% and 10.5% for the *x, y* and ψ coordinates respectively relative to the baseline *NGMV*. Furthermore a reduction in control action is achieved, at 39.4%, 37.9% and 42.4% for *x*, *y* and *r* respectively.

Ship Earth-frame Coordinates & Thruster Commands Comparison for Sea State 5:

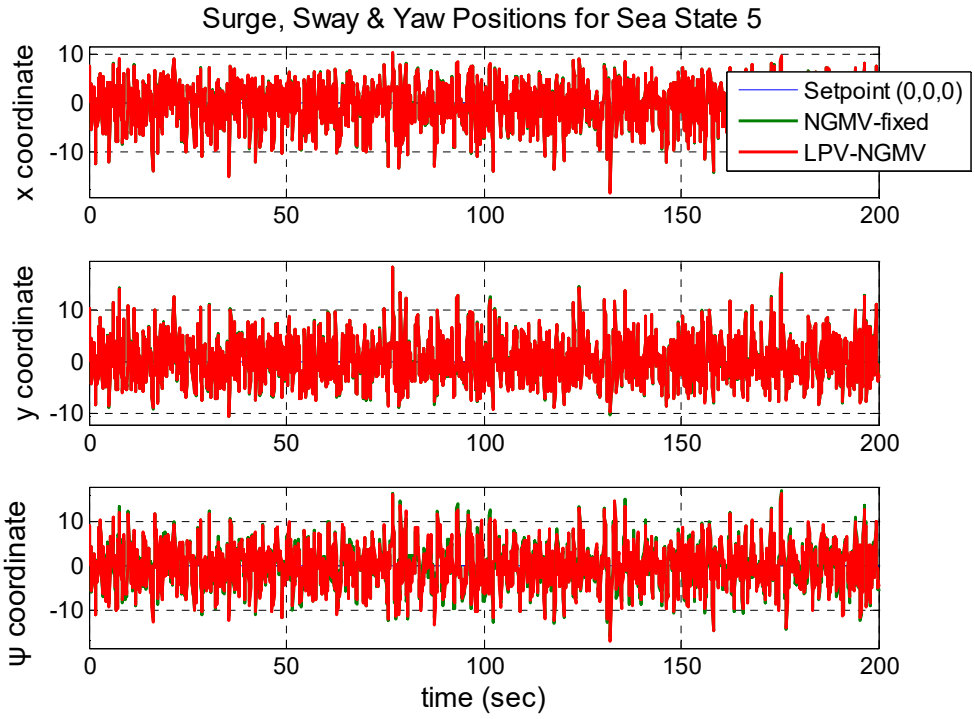


Figure 5:51 Vessel earth coordinate positions in [m] for Sea State 5.

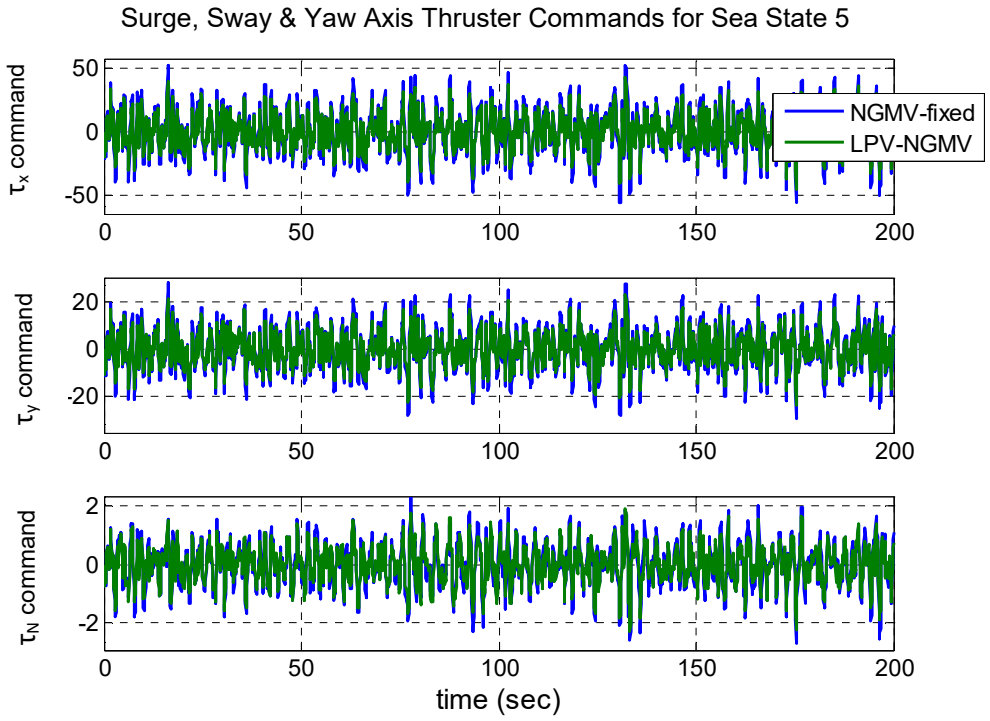


Figure 5:52 Vessel thruster commands in [N] for Sea State 5.

Table 5:5: Sea State 5, quantified controller comparison

Controllers	β_{STD} (norm.)			Pel_{MISE} (norm.)		
	***ux* thrust**	***uy* thrust**	***ur* thrust**	***x* coord.**	***y* coord.**	**ψ coord.**
NGMV	1	1	1	1	1	1
LPV-NGMV	0.7715	0.7781	0.8358	0.9977	0.9987	0.9699

The *LPV-NGMV* yields an improvement in regulation *MISE* of 0.23%, 0.13% and 3% for the *x, y* and ψ coordinates respectively relative to the baseline *NGMV*. Furthermore a reduction in control action is achieved, at 22.8%, 22.1% and 16.4% for the *x*, *y* and *z* axis respectively.

5.7 Final Remarks

In this section the *NGMV* nonlinear control design technique was employed to the navigation and dynamic positioning of an experimental marine vessel. Using an initial *PID* based tuning throughout a set of operating points, simulation showed that the *NGMV* maintains a reasonable tracking performance while the *PID* deteriorates. If not, retuning or gain scheduling may be used for the *PID*, though sometimes their realisation proves to be a very difficult task in nonlinear systems and it usually targets linear approximations of the model. In further simulations, not presented in this section, the *NGMV* seems to maintain performance for an even wider operating range after the *PID* goes unstable. Note that with a small tweaking it proves that it can perform even better in these operating points.

For the disturbance rejection performance, it is shown that the *PID* configuration uses a *Notch* filter, the downside of which is that it reduces the phase margin of the system in critical levels. In simulations the filter was applied and the entire system was re-tuned to reduce the disturbance as much as possible without going unstable. This has set constrains on the specifications of the filter. In the *NGMV* case the high frequency wave components are filtered out by the same weightings to avoid the phase lag that accompanies the *Notch* filter (Martin, 2004), maintaining marginal stability to reasonable levels. The controller without any excessive control signals can compensate the low frequency components such as wave and current induced motion. In further simulations, is shown that when increasing the parameters of the disturbance like power and range the *PID* performance leads to very large control outputs saturating the thrusters or leading to large tracking variations whereas the *NGMV* maintains control output within almost the same range in all cases and without retuning.

Lastly, although not initially in the thesis, a final section was added to the chapter to explore the performance of the *LPV-NGMV* formulation suggested in Section 4.6 when employed for the dynamic positioning of the experimental vessel under varying sea state (wave disturbance). The *LPV* controller was compared against, updating its internal gain calculations parameterised by sea state variations, was compared against a fixed gains standard *NGMV* controller. This "*black box*" and varying disturbance model W_d formulation proved to yield small but yet substantial improvement especially for sea states 2 and 3. This may imply weakness of the algorithm in faster systems, however further research and simulations are suggested for a definitive conclusion.

Chapter 6 Wind Turbine Control

In this chapter the *LPV-NPGMV* algorithm that was derived in Chapter 4 is tested in a control application for a wind energy conversion system (*WECS*). There has been a lot of interest in the application of advanced controls to wind turbine systems due to increase in size and performance requirements. This applies to both individual wind turbine controls and for the coordinated controls for total wind farms. The most successful advanced control method used in other industries is predictive control which has the unique ability to handle hard constraints that limit system behaviour. However, wind turbine systems are particularly difficult in being very nonlinear and dependent upon the external parameter variations which determine behaviour. Many nonlinear approaches are probably too complicated to implement in most situations and the approach proposed here is to use one of the latest predictive control methods which can be used with linear parameter varying models. The use of linear parameter varying (*LPV*) models has also been discussed previously (Østergaard et al. 2009) and (Adegas, and Stoustrup, 2012). However, new controllers have been developed for industrial processes particularly aimed at generating relatively simple designs to understand and implement. The particular features of the design method proposed in this thesis which are valuable for the wind turbine control problem include the very flexible way to model the process and the very general criterion that may be optimised. This criterion can have nonlinear terms and if for example fatigue is being minimised in wind turbines this is an important feature. There are not many control techniques which enable a nonlinear cost function to be minimised using a theoretical solution and an algorithm which is relatively simple to understand and implement. The main feature of the following work is the demonstration of how the controller is used and the benefits that are available. This chapter is divided into four sections which are briefly explained below.

Section 1 – Problem Description; overview of the wind turbine control objectives and strategy employed in this work.

Section 2 – System Model Description; derivation of the *LPV* wind turbine model used within the NPGMV design.

Section 3 – Control System Description; adaptation of the control algorithm as derived in Equation 4:100 and also of a basic *NGMV* and *PID* control formulations used here for comparison.

Section 4 – Simulation Results & Conclusions; definition of the different scenarios and presentation of control performance results for the different schemes employed throughout the simulations.

6.1 Problem Description

The main objective of the control solution proposed in this chapter is the regulation of produced electrical power in a large scale wind turbine. This is achieved at the turbine rated power while compensating for,

- System nonlinearities arising,
    - in the mechanical parts of the turbine (e.g. actuator range limitations),
    - in the aerodynamic conversion between wind energy and electrical power,
- Wind disturbances (e.g. sudden wind gusts and high frequency wind variations).

Using the same control paradigm, a secondary scenario is explored, that of varying the power output demand of the turbine (derating). This is particularly useful in centralised wind farm power production control. In the latter, individual turbines are required to reduce their output so that an optimal global output is reached with respect to various criteria like the minimisation of mechanical loading in turbine tower and nacelle structures, the maximisation of grid power demand etc. (AEOLUS FP7).

It is important to mention here that this was not a full-bore wind turbine control study (considering switching between regions of operation, optimal curve tracking below rated wind speed etc.) but rather one aspect of the system operation was selected to capture characteristics of interest to explore implementability of the introduced scheme such as:

- Reference tracking
- Disturbance rejection
- Nonlinear effects compensation like pitch actuator constraints in that case
- Varying system dynamics like the aerodynamic conversion variables subject to wind speed variations

The wind turbine control strategy described in this section incorporate two separate configurations (a *SISO* and a *MIMO* control system) as explained below (Bianchi et al., 2006).

1. **Fixed-Torque/Variable-Pitch;** the generator torque is kept at the rated value whilst the pitch is manipulated for power regulation at the rated value during wind speed variations.
2. **Variable-Torque/Variable-Pitch;** both the generator torque and pitch are manipulated to regulate the generator speed and power respectively at the rated value, during wind speed variations.

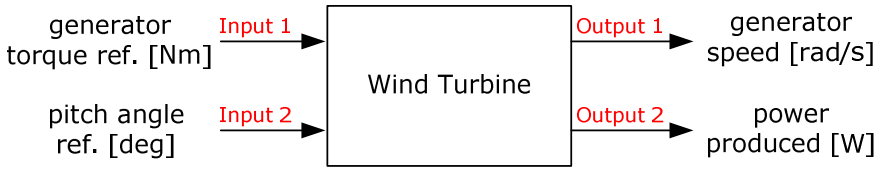


Figure 6:1 *MIMO* System *MV/CV* variables (the *SISO* case is the subset Input 2/Output 2).

When the wind turbine operates in the below rated wind speed region control strategies mostly aim at the maximisation of produced electrical power. Throughout this mode of operation the blade pitch is set to zero to allow full harvest of the energy available in the wind whilst the optimal torque reference to the generator is derived from optimal lookup tables implemented within the controller.

For this application however the focus goes to the above rated operating region (see Mode 3 below) where the main control objective becomes the regulation of the produced electrical power at its rated value, also limited by the generator speed rating. For completion and reference, the operating modes of a *WECS* system are summarised below and also seen in Figure 6:2, with respect to wind speed, for the particular wind turbine system used in this chapter.

5MW NREL Wind Turbine Operating Modes & Control Objectives:

1. Mode 1 (0-4m/s) – Start up/Low wind speed; objective: maximise power constrained by minimum rotor speed.
2. Mode 2 (4-10.9m/s) – Below rated wind speed; objective: maximise power.
3. Mode $2^{1/2}$ (10.9-11.5m/s) – Below rated wind speed and rotor speed; objective: maximise power constrained by nominal/maximum rotor speed.
4. Mode 3 (11.5-25m/s) – Above rated wind speed; objective: maintain rated power and nominal/maximum rotor speed.

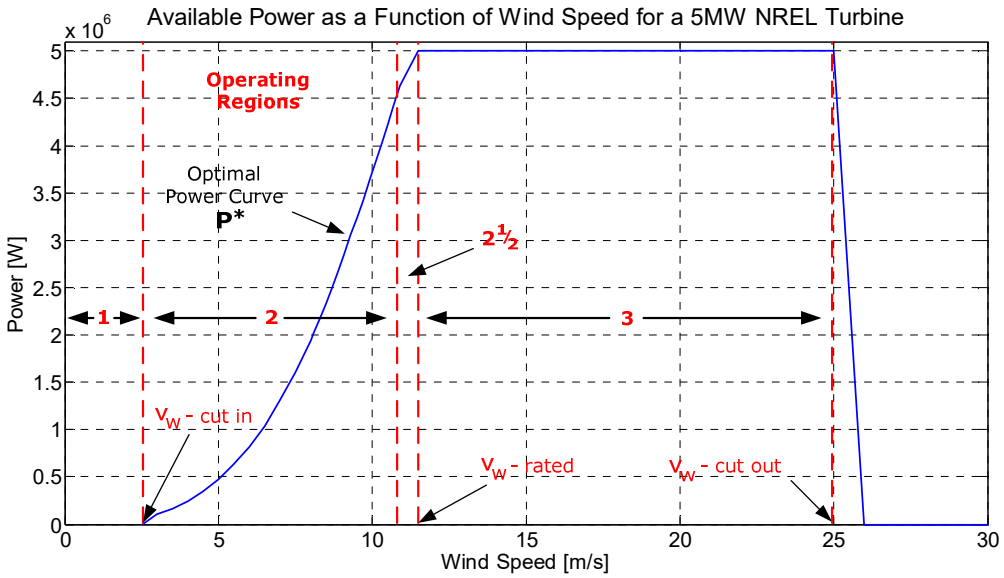


Figure 6:2 Optimal Electric Power curve and modes of operation for the 5MW *NREL* wind turbine system with respect to wind speed.

6.1.2 Controller Structure

The architecture used for all controllers is composed of a feedback and a feedforward component, where feedforward action establishes the nominal operating point at every step and feedback action compensated for deviations around that operating point. This can be summarised in Figures 6:3 and 6:4 for the *SISO* and the *MIMO* case respectively. Note that the wind speed estimation aspect is not a part of this investigation, therefore the assumption of an accurate wind speed measurement is made throughout all simulations.

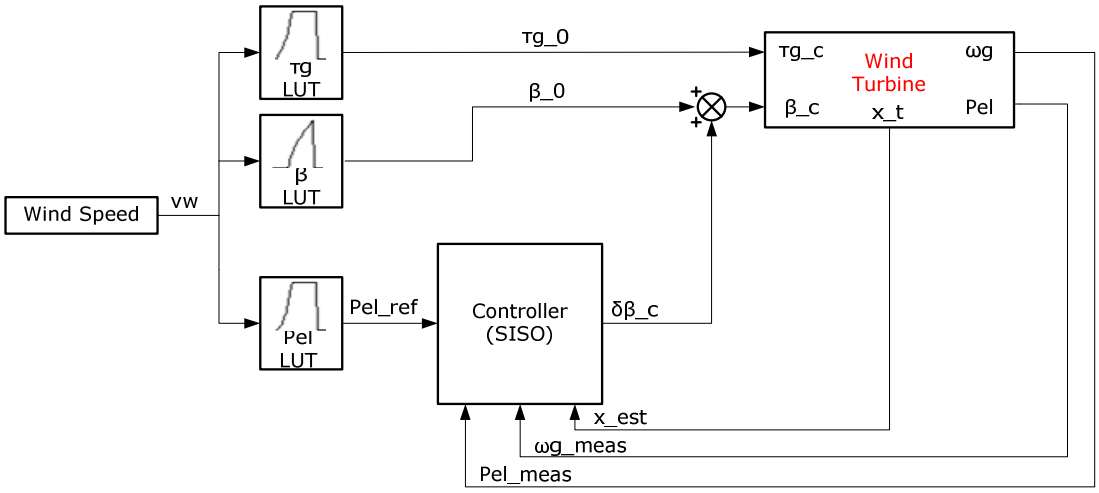


Figure 6:3 *SISO* Control System *FB+FF* structure.

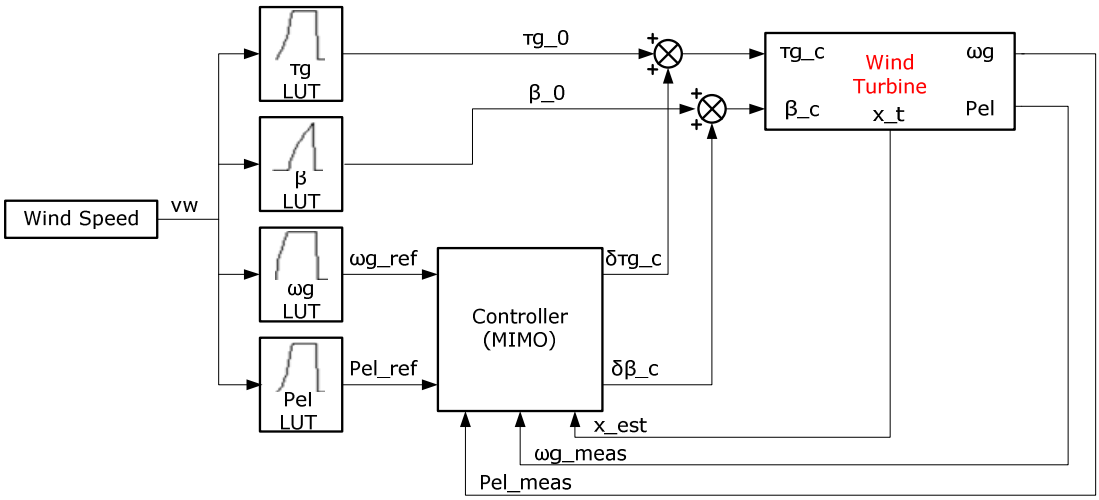


Figure 6:4 *MIMO* Control System *FB+FF* structure.

Feedforward Action; this component is based on the optimal trajectories for the pitch angle, rotor/generator speed, power and generator torque. Its main purpose is to provide control action that will keep power production at the rated value above rated wind speed - in the steady-state sense and assuming no modelling errors. The optimal reference curves, provided by the lookup tables, are also used to generate power and speed reference signals for the feedback controller. Note that for the above rated operation that is examined here only the pitch angle optimal curve varies whereas the power, generator speed and torque curves remain fixed at the corresponding rated values. The optimal curves for this particular wind turbine can be seen in Figure 6:5.

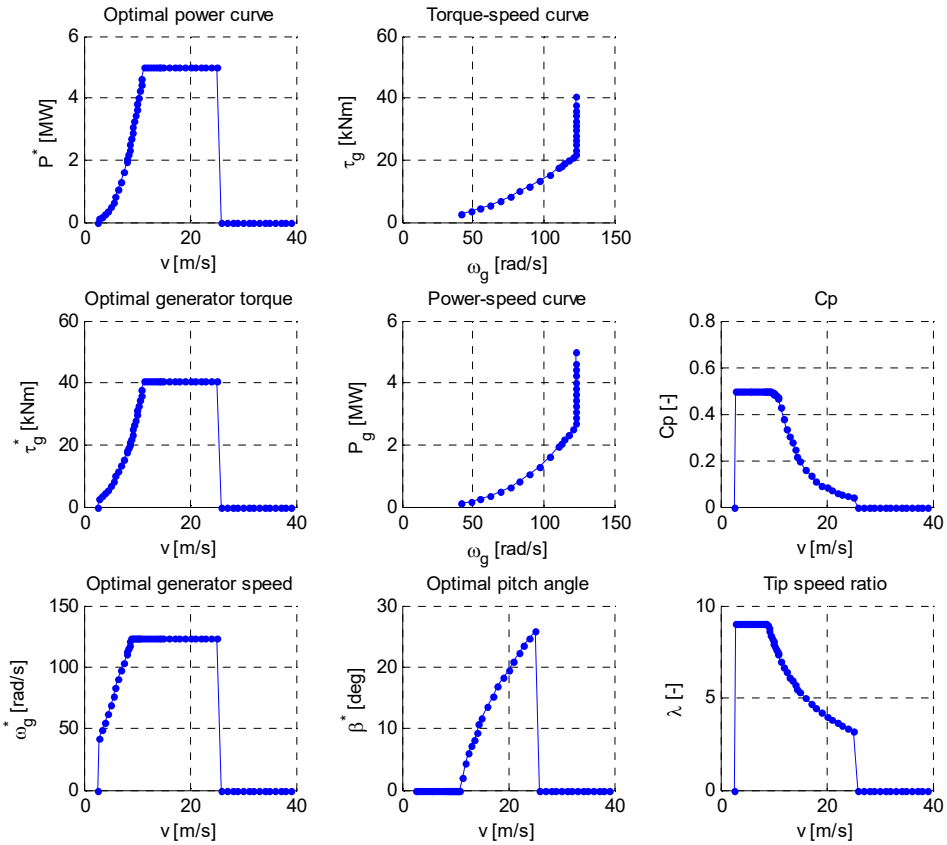


Figure 6:5 Optimal Reference Curves for the 5MW *NREL* wind turbine.

Feedback Action; this component is used to minimise power variations around the rated value and compensate for model uncertainties and nonlinearities.

Here, the generic inputs and outputs of the system in standard control terminology are:

Controlled variables (*CV*):

- Produced (electrical) power reference
- Generator speed reference (for the *MIMO* case only; for the *SISO* only *FF* is employed)

Manipulated variables (*MV*):

- Pitch angle reference
- Generator torque reference (for the *MIMO* case only; for the SISO only *FF* is employed)

Disturbance variables (*DV*):

- Effective wind speed (as experienced in the blades of the turbine; measured minus the tower bending velocity)

The next section covers the derivation of an *LPV* model of the wind turbine system, suitable for the formulation of the *LPV-NPGMV* controller.

6.2 Wind Turbine Model Description

The physical system that was selected for this experimentation is that of an offshore *WECS*. It is a theoretical system, representative of a utility-scale multi-megawatt wind turbine developed by the *National Renewable Energy Laboratory* (*NREL*) and thoroughly validated against real systems (Jonkman et al. 2009). More specifically, it is a three-bladed upwind, 5MW wind turbine with active pitch control and *Doubly-Fed Induction Generator* (*DFIG*) with controllable generator torque.

The overall wind turbine from a systems standpoint is a combination of static and dynamic, linear and nonlinear components. Figure 6:6 illustrates the complete features of the *NREL* system used in this chapter. However the following list only contains those sub-systems that are vital for control design and the modelling process:

1. Pitch Actuator; linear-dynamic.
2. Rotor Aerodynamics; nonlinear-static.
3. Transmission (low/high speed shafts); linear-dynamic.
4. Generator & Converter; linear-dynamic.

The principal objective of a *WECS* is to convert kinetic energy out of the wind into electrical power. The first conversion occurs in the rotor of the turbine where wind power is translated into mechanical power and subsequently translated into electrical through the transmission and generator components. The amount of power which could be extracted by the wind is determined by the area swept by the turbine rotor and is limited by a factor which varies with the tip speed ratio and the pitch angle of the blade.

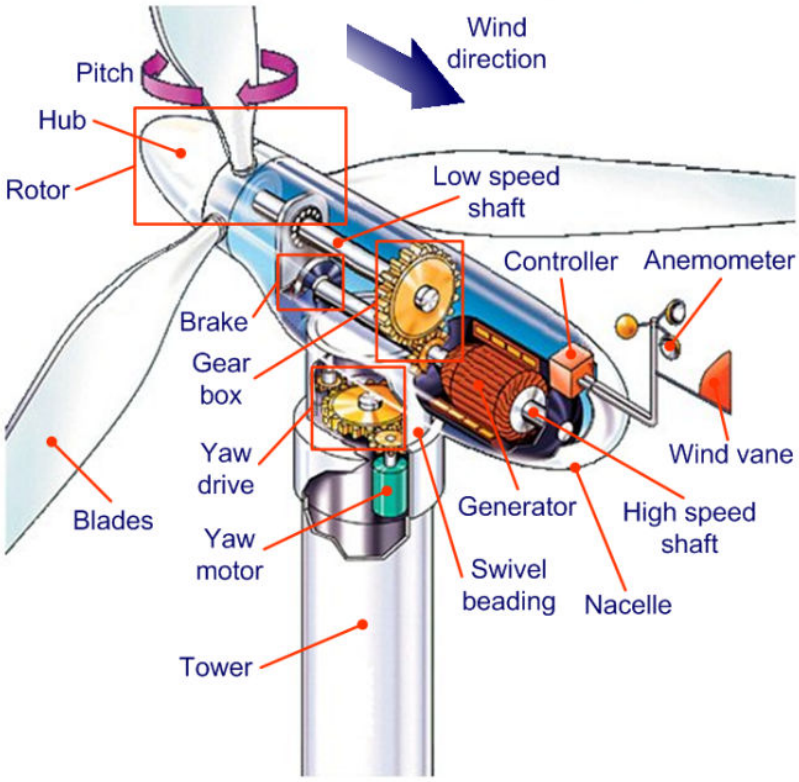


Figure 6:6 Wind Turbine Subsystems. (http://www.alternative-energy-news.info/technology/wind-power/wind-turbines/)

These mathematical representations for each of these subsystems are derived in the following sections (Østergaard et al. 2007).

Extractable Energy in Wind:

The total available power (P_{total}) from the wind at the turbine rotor is given by the following relation.

$$P_{total} = \frac{1}{2}\rho A v^3 = \frac{1}{2}\rho \pi R^2 v^3 \quad (6:1)$$

where

A	area swept by the rotor [m^2]
v	Wind speed [m/s]
ρ	Air density [kg/m^3]
R	Rotor radius [m]

The available power in the wind is related to efficiency and limited by a factor 16/27 known as the *Betz* limit or *Betz* efficiency.

Pitch Actuator:

The pitch actuator is controlled to provide rotation of the blade around the pitch axis to change the angle of attack. In certain wind turbines the blades can be controlled individually here however a common actuator is used for all three blades. The pitch actuator angle can be represented by the following second order system.

$$\beta(s) = \beta_{ref}(s) \cdot \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{6:2}$$

where

β	Collective pitch angle [deg]
β_{ref}	Pitch angle reference
ω_n	Natural frequency [rad/s]
ξ	Damping coefficient

By using the inverse Laplace transform we can translate Equation 6:2 into the time domain. This intermediate step is useful for the conversion of the subsystems into state-space later in the chapter.

$$\begin{gathered}
\beta(s)[s^2 + 2\xi\omega_n s + \omega_n^2] = \beta_{ref}(s)\omega_n^2 \\
\Rightarrow s^2\beta(s) + 2\xi\omega_n s\beta(s) + \omega_n^2\beta(s) = \beta_{ref}(s)\omega_n^2 \\
\overset{\{L^{-1}\}}{\Longrightarrow} \ddot{\beta} + 2\xi\omega_n\dot{\beta} + \omega_n^2\beta = \beta_{ref}\omega_n^2 \\
\Rightarrow \ddot{\beta} = \beta_{ref}\omega_n^2 - 2\xi\omega_n\dot{\beta} - \omega_n^2\beta \\
\Rightarrow \ddot{\beta} = \omega_n^2\left[\beta_{ref} - \frac{2\xi\dot{\beta}}{\omega_n} - \beta\right]
\end{gathered} \tag{6:3}$$

Rotor Aerodynamics:

Rotor aerodynamics is the main nonlinear component of the wind turbine system and it describes the aerodynamic conversion from the wind energy captured by the rotor, to the resulting torque which drives the rotating parts. It can take the form of the mechanical power $P_{extractable}$ produced as shown below.

$$P_{extractable} = C_p \cdot P_{total} \tag{6:4}$$

C_p is the power coefficient and it varies with the blade pitch angle β tip speed ratio λ given by Equation 6:5. In practice C_p has a certain optimum which is approximately 0.4.

$$\lambda = \frac{\omega_r R}{v} \tag{6:5}$$

Substituting and expanding Equations 6:1 and 6:4 results into the following relation for $P_{extractable}$.

$$P_{extractable} = C_p(\lambda, \beta) \cdot \frac{1}{2} \rho \pi R^2 v^3 \tag{6:6}$$

From the extractable power, the aerodynamic (or rotor) torque applied on the rotor shaft can be derived by the following relation.

$$\tau_r = \frac{P_{ext}}{\omega_r} \tag{6:7}$$

Substituting Equation 6:6 yields the following relation.

$$\tau_r = C_q(\lambda, \beta) \cdot \frac{1}{2} \rho \pi R^3 v^2 \tag{6:8}$$

C_q is the torque coefficient and it has an equivalent relation to C_p as shown below.

$$C_q = \frac{C_p}{\lambda} \tag{6:9}$$

Another variable involved in the aerodynamics of the wind turbine is the thrust force F_t. This is the force exerted by the wind on the rotor and produces motion of the turbine tower in the fore-aft direction. The thrust force is given by the following relation.

$$F_t = C_t(\lambda, \beta) \cdot \frac{1}{2} \rho \pi R^2 v^2 \tag{6:10}$$

where C_t is the thrust coefficient.

C_p, C_q, C_t are given by lookup tables parameterised by the pitch angle and the tip speed ratio. A portion of the C_p with respect to either of the two parameters for the *NREL* wind turbine is presented in Figures 6:7 and 6:8.

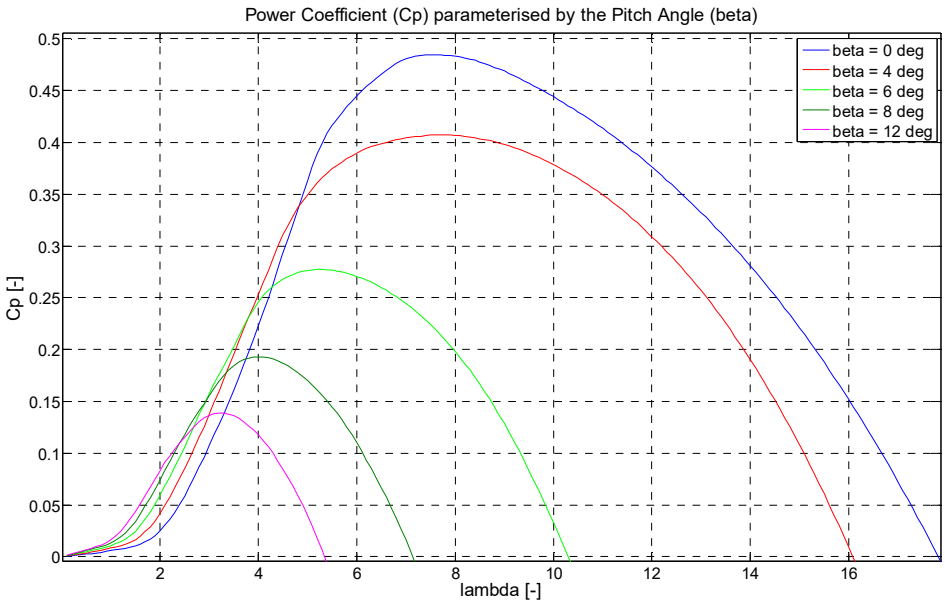


Figure 6:7 5MW NREL C_p curve with respect different values of β.

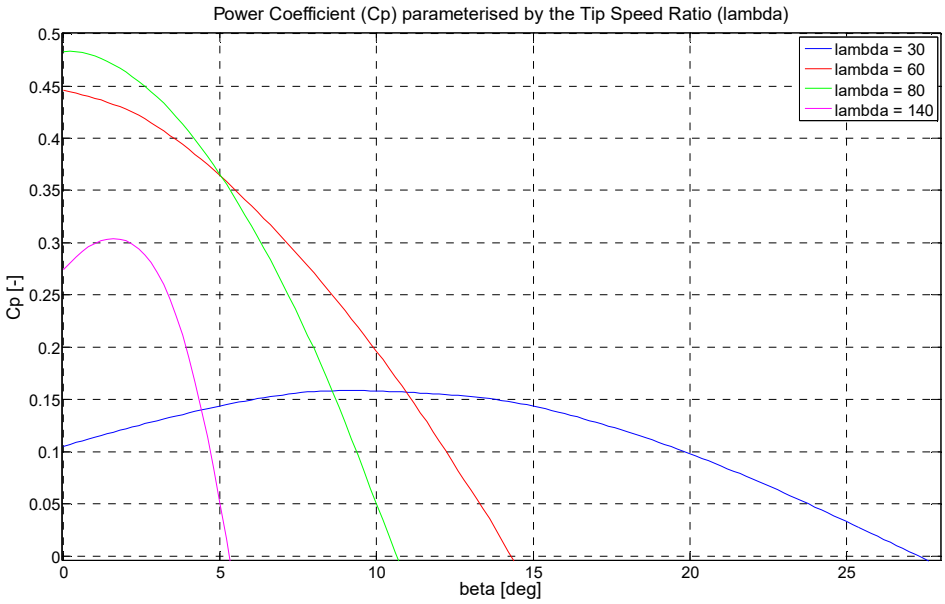


Figure 6:8 5MW NREL C_p curve with respect different values of λ.

Transmission System (Drivetrain):

The transmission system is responsible to transfer the mechanical power, generated at the turbine rotor, to the electrical components (generator). It consists of two shafts (low and high

speed) which are connected via a gearbox. The gearbox introduces an increase of speed from the rotor to the high speed shaft to values suitable to drive the generator which in turn converts the mechanical power to electrical.

For simplicity, in our modelling approach a flexible shaft, two-mass third order system is used which is considered to be a sufficient approximation for control purposes. Figure 6:9 depicts the two-mass system equivalent structure and parameters. The drivetrain configuration shown starts with the aerodynamic torque (τ_r) generated at the rotor (hub and blades) which translates into torque on the generator side (τ_g) via an arrangement of two shafts and a gearbox which amplifies speed from the low speed to the high speed shaft.

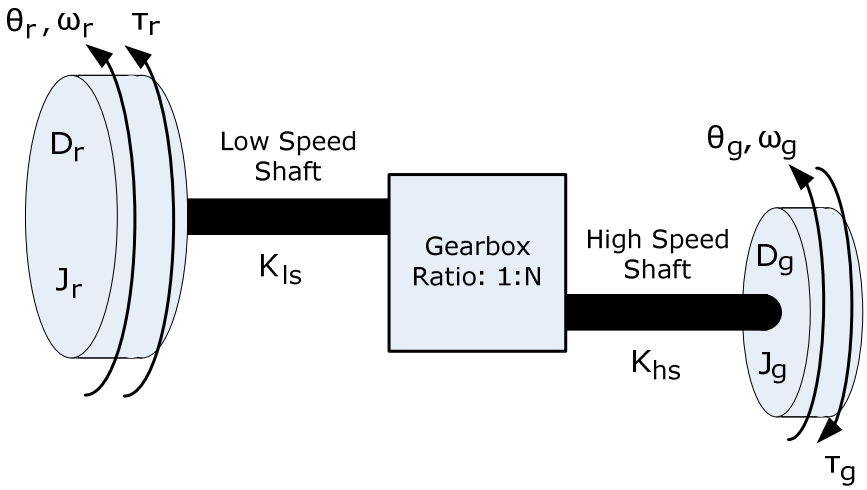


Figure 6:9 Two-Mass System equivalent of the Wind Turbine Drivetrain.

where

D_r, D_g	Rotor and generator friction damping [Nms/rad]
J_r, J_g	Rotor and generator inertia [Nms^2/rad]
K_{ls}, K_{hs}	Low and high speed shaft stiffness [Nm/rad]
ϑ_r, ϑ_g	Rotor and generator angular position [rad]

The torque balance for the system described above is captured by the following equations. Note that the high speed shaft stiffness K_{hs} can be expressed into the low shaft stiffness K_{ls} via the gear ratio N. This similarly goes for the angular positions both expressed in terms of the low shaft torsion angle ϑ_r or just ϑ for simplicity. This can be taken further by expressing D_g in terms of the low shaft friction damping D_r (or D_{ls}). The resulting transmission equations are as follows.

$$\dot{\omega}_r = \frac{1}{J_r}\left[\tau_r - K_{ls}\theta - \frac{1}{N}D_{ls}\omega_r - D_{ls}\omega_g\right]$$
$$\dot{\omega}_g = \frac{1}{J_g}\left[-\tau_g + \frac{1}{N}K_{ls}\theta - \frac{1}{N}D_{ls}\omega_r - \frac{1}{N^2}D_{ls}\omega_g\right] \quad (6:11)$$
$$\dot{\theta} = \omega_r - \frac{1}{N}\omega_g$$

Tower Bending Dynamics:

Tower motion is only considered here with regards to the longitudinal displacement (fore-aft), as a result of the thrust force (F_t) exerted on the turbine structure when the wind passes through the rotor. The following second order state-space system is used to describe this.

$$\begin{bmatrix}\dot{v}_{FA}\\ \dot{x}_{FA}\end{bmatrix} = \begin{bmatrix}-\frac{D_{FA}}{m_{FA}} & -\frac{K_{FA}}{m_{FA}}\\ 1 & 0\end{bmatrix}\begin{bmatrix}v_{FA}\\ x_{FA}\end{bmatrix} + F_t$$
$$v_{FA} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}v_{FA}\\ x_{FA}\end{bmatrix} \quad (6:12)$$

where

v_{FA}	Tower fore-aft velocity [m/s]
x_{FA}	Tower fore-aft displacement [m]
K_{FA}	Tower fore-aft stiffness [N/m]
D_{FA}	Tower fore-aft damping [N/(m/s)]
m_{FA}	Tower fore-aft mass [kg]
F_t	Thrust force [N]

Generator and Converter System:

In this study only the active power control at turbine (mechanical) level is considered. For this reason, the power controller is not described in details. The *NREL* model contains a simpler description of the electrical generator. In this model it is assumed that the power controller provides the electrical generator with torque reference. The generator dynamics is modelled by the first order system. Equation 6:13 describes the generator torque whereas Equation 6:14 the converter power.

Generator Torque

$$\tau_g(s) = \tau_{g,ref}(s) \cdot \frac{\alpha_{gc}}{s + \alpha_{gc}} \quad (6:13)$$

Convertor Power

$$P(t) = \eta_g \omega_g(t) \tau_g(t) \tag{6:14}$$

where

τ_g Generator torque
$\tau_{g,ref}$ Generator torque reference
α_{gc} First order constant
η_g Generator efficiency

Wind Turbine Sub-Systems Integration:

Figure 6:10 shows how all individual subsystems connect to formulate the total system of the wind turbine without the controller.

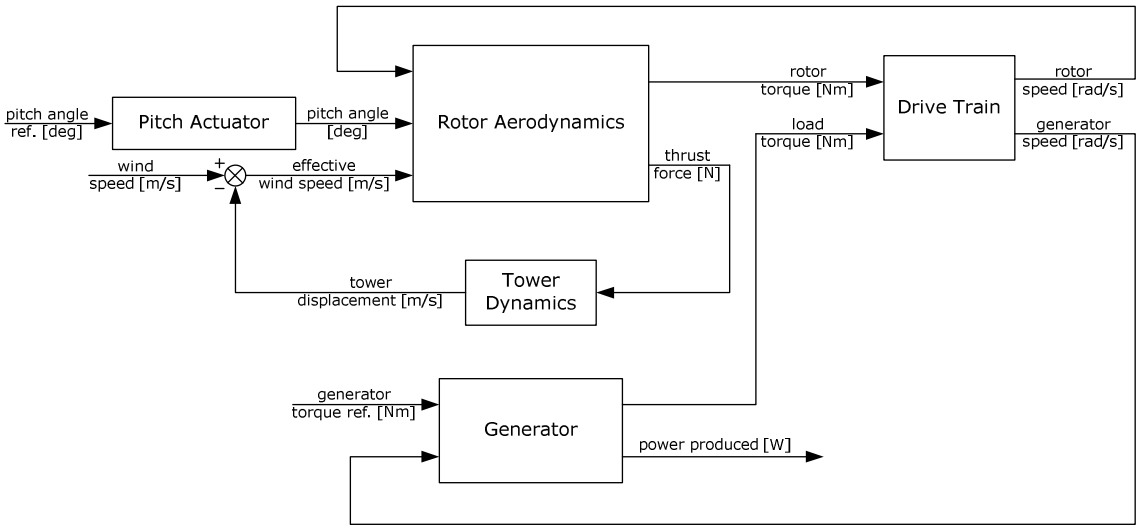


Figure 6:10 Wind Turbine subsystem interconnections.

6.2.1 Wind Turbine LPV Model for Control

This section describes the analytical derivation and the discretisation of the linearised parameter varying (*LPV*) model used within the *NPGMV* controller. The model reflects small deviations along the optimal trajectory is consists of all the sub-systems that were described in the previous section and summarised below (Østergaard et al. 2007).

1. Blade Pitch Actuator
2. Rotor Aerodynamics
3. Transmission System (Drivetrain)

4. Tower Bending Dynamics
5. Generator and Converter System

The wind turbine system structure can be conveniently put into *LPV* formulation as all of its states can be parameterised by wind speed. The *LPV* blade pitch actuator and tower bending models are identical to the ones described in the previous section, since they were already linear. The *LPV* drivetrain model is very similar to the previously described model and it is assumed there are no nonlinear mechanical losses. The *LPV* generator and converter model is also similar to the previous model also under the assumption that there are no electrical power losses. The main difference is that the original torque and thrust coefficient lookup tables of the previous model are replaced by linearised versions.

Blade Pitch Actuator:

State vector • Pitch angle rate, $\dot{\beta}$ [deg/s] • Pitch angle, β [deg]	$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \dot{\beta} \\ \beta \end{bmatrix}$
Input vector • Pitch angle reference (β_r=β_{ref}) [deg]	$u = [u_1] = [\beta_r]$
Output vector • Measured pitch angle, β [deg]	$y = [y_1] = [\beta]$

The state-space model for the pitch actuator can be derived by rearranging Equation 6:2 as follows,

$$\ddot{\beta} = \omega_n^2 \left[\beta_r - \frac{2\xi\dot{\beta}}{\omega_n} - \beta \right] \Rightarrow \dot{x} = Ax + Bu \;\; \& \;\; y = Cx + Du \tag{6:15}$$

This yields the following state-space model,

$$\begin{gathered} u = \beta_r \\ x_2 = \beta \\ x_1 = \dot{x}_2 = \dot{\beta} \\ \dot{x}_1 = \ddot{x}_2 = \ddot{\beta} = \omega_n^2 \left[u - \frac{2\xi x_1}{\omega_n} - x_2 \right] \\ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -2\xi\omega_n & -\omega_n^2 \\ 1 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \omega_n^2 \\ 0 \end{bmatrix}}_{B} u \\ y = x_2 \\ y = \underbrace{[0 \quad 1]}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{[0]}_{D} u \end{gathered} \tag{6:16}$$

Rotor Aerodynamics:

Input vector • Blade effective wind speed, $v_{eff} = v_w - v_{FA}$ (where v_{FA} is the tower displacement speed), [m/s] • Rotor speed, ω_r [rad/s] • Blade pitch angle, β [deg]	$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} v_{eff} \\ \omega_r \\ \beta \end{bmatrix}$
Output vector • Rotor torque, τ_r [Nm] • Thrust force, F_t [N]	$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \tau_r \\ F_t \end{bmatrix}$

The linear deviations model around an operating point for the aerodynamic torque and force exerted on the turbine tower with respect to v_{eff}, ω_r and β are shown below.

$$\lambda_{op} = \frac{R}{v_{eff\_op}} \omega_{r\_op} \tag{6:17}$$

$$\delta\tau_r = \frac{\pi}{2}\rho R^3 \left[-R\omega_{r\_op} C_{q\_\lambda lin}(\lambda_{op}, \beta_{op}),\ Rv_{eff\_op} C_{q\_\lambda lin}(\lambda_{op}, \beta_{op}),\ v_{eff\_op}^2 C_{q\_\beta lin}(\lambda_{op}, \beta_{op})\right] \delta \begin{bmatrix} v_{eff} \\ \omega_r \\ \beta \end{bmatrix} \tag{6:18}$$

$$\delta F_t = \frac{\pi}{2}\rho R^2 \left[-R\omega_{r\_op} C_{t\_\lambda lin}(\lambda_{op}, \beta_{op}),\ Rv_{eff\_op} C_{t\_\lambda lin}(\lambda_{op}, \beta_{op}),\ v_{eff\_op}^2 C_{t\_\beta lin}(\lambda_{op}, \beta_{op})\right] \delta \begin{bmatrix} v_{eff} \\ \omega_r \\ \beta \end{bmatrix} \tag{6:19}$$

Coefficients in the above deviations model correspond to the following partial derivatives for τ_r and F_t respectively as follows,

$$f_{\tau_r\_v_{eff}} = \frac{\delta\tau_r}{\delta v_{eff}}\bigg|_{v_{eff\_op}}, \quad f_{\tau_r\_\omega_r} = \frac{\delta\tau_r}{\delta\omega_r}\bigg|_{\omega_{r\_op}}, \quad f_{\tau_r\_\beta} = \frac{\delta\tau_r}{\delta\beta}\bigg|_{\beta_{op}}$$

$$f_{F_t\_v_{eff}} = \frac{\delta F_t}{\delta v_{eff}}\bigg|_{v_{eff\_op}}, \quad f_{F_t\_\omega_r} = \frac{\delta F_t}{\delta\omega_r}\bigg|_{\omega_{r\_op}}, \quad f_{F_t\_\beta} = \frac{\delta F_t}{\delta\beta}\bigg|_{\beta_{op}}$$

Transmission System (Drivetrain):

State vector • Low speed shaft torsional angle, ϑ [rad] • Rotor speed, ω_r [rad/s] • Generator speed, ω_g [rad/s]	$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \omega_r \\ \omega_g \end{bmatrix}$
Input vector • Rotor torque, τ_r [Nm] • Generator torque, τ_g [Nm]	$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_r \\ \tau_g \end{bmatrix}$
Output vector • Rotor speed, ω_r [rad/s] • Generator speed, ω_g [rad/s]	$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \omega_r \\ \omega_g \end{bmatrix}$

The torque lossless transmission system state-space representation is shown below.

$$\begin{bmatrix}\dot{\theta}\\ \dot{\omega}_r\\ \dot{\omega}_g\end{bmatrix}=\begin{bmatrix}0 & 1 & -\frac{1}{N}\\ -\frac{K_{ls}}{J_r} & -\frac{D_{ls}}{NJ_r} & -\frac{D_{ls}}{J_r}\\ \frac{K_{ls}}{NJ_g} & -\frac{D_{ls}}{NJ_g} & -\frac{D_{ls}}{N^2J_g}\end{bmatrix}\begin{bmatrix}\theta\\ \omega_r\\ \omega_g\end{bmatrix}+\begin{bmatrix}0 & 0\\ \frac{1}{J_r} & 0\\ 0 & -\frac{1}{J_g}\end{bmatrix}\begin{bmatrix}\tau_r\\ \tau_g\end{bmatrix} \quad (6:20)$$

$$\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\theta\\ \omega_r\\ \omega_g\end{bmatrix}$$

Tower Bending Dynamics:

The longitudinal bending of the tower has been described earlier (Equation 6:12). The inputs, outputs and states of the model are defined as follows,

State vector • Tower fore-aft velocity, v_{FA} [m/s] • Tower fore-aft displacement, x_{FA} [m]	$x=\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}v_{FA}\\ x_{FA}\end{bmatrix}$
Input vector • Thrust force, F_t [N]	$u=[u_1]=[F_t]$
Output vector • Tower fore-aft velocity, v_{FA} [m/s]	$y=[y_1]=[v_{FA}]$

Generator and Converter System:

State vector • Generator load torque, τ_g [Nm]	$x=[x_1]=[\tau_g]$
Input vector • Generator load torque reference, $\tau_{g,ref}$ [Nm] • Generator speed, ω_g [rad/s]	$u=\begin{bmatrix}u_1\\ u_2\end{bmatrix}=\begin{bmatrix}\tau_{g\_ref}\\ \omega_g\end{bmatrix}$
Output vector • Generator load torque, τ_g [Nm] • Produced electrical power, P_{el} [W]	$y=\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}\tau_g\\ P_{el}\end{bmatrix}$

The linear state-space model and components for the generator and converter subsystem are defined as follows,

$$\delta\dot{\tau}_g=-\alpha_{gc}\delta\tau_g+[\alpha_{gc}\quad 0]\delta\begin{bmatrix}\tau_{g_{ref}}\\ \omega_g\end{bmatrix}$$

$$\delta\begin{bmatrix}\tau_g\\ P_{el}\end{bmatrix}=\begin{bmatrix}1\\ \omega_{g_{op}}\end{bmatrix}\delta\tau_g+\begin{bmatrix}0 & 0\\ 0 & \tau_{g_{op}}\end{bmatrix}\delta\begin{bmatrix}\tau_{g_{ref}}\\ \omega_g\end{bmatrix} \quad (6:21)$$

$$P_{el\_op}=\tau_{g\_op}\omega_{g\_op}$$

where

ω_{g_op} Generator speed at (nominal) operating point [rad/s]
τ_{g_op} Generator torque at (nominal) operating point [Nm]
P_{el_op} Generator power at (nominal) operating point [W]

The following table contains the values of the parameters used in simulation.

Aerodynamics	**Generator and Converter**
$R = 63$	$a_{gc} = 0.1$
$\rho = 1.225$	
	Tower Fore-Aft-Dynamics
Drive Train	$M_{t\_nac} = 350000$
$K_{ls} = 8.6763 \cdot 10^8$	$M_{t\_tow} = 347460$
$D_{ls} = 6.215 \cdot 10^6$	$\omega_{t\_FA} = 0.321$
$N = 97$	$\zeta_{t\_FA} = 0.8$
$J_g = 534.116$	
$J_r = 35444.067$	

Wind Turbine Subsystem Integration:

Combining the individual subsystems into one integrated *LPV* model for the wind turbine yields the following structure. It is assumed that only three outputs are available for control and these are the generator speed, the pitch angle and the electric power out of the generator.

State vector • Blade pitch angle, β [deg] • Blade pitch angle, $\dot{\beta}$ [deg/s] • Low speed shaft torsional angle, ϑ [rad] • Generator speed, ω_g [rad/s] • Rotor speed, ω_r [rad/s] • Generator load torque, τ_g [Nm] • Tower fore-aft displacement, x_{FA} [m] • Tower fore-aft velocity, v_{FA} [m/s]	$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} \beta \\ \dot{\beta} \\ \theta \\ \omega_g \\ \omega_r \\ \tau_g \\ x_{FA} \\ v_{FA} \end{bmatrix}$
Input vector • Blade effective wind speed, v_{eff} [m/s] • Generator load torque reference, $\tau_{g,ref}$ [Nm] • Pitch angle reference β_r [deg]	$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} v_{eff} \\ \tau_{g\_ref} \\ \beta_{ref} \end{bmatrix}$
Output vector • Generator speed, ω_g [rad/s] • Blade pitch angle, β [deg] • Produced electrical power, P_{el} [W]	$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \omega_g \\ \beta \\ P_{el} \end{bmatrix}$

$$\frac{d}{dt}\begin{bmatrix}\beta\\ \dot{\beta}\\ \theta\\ \omega_g\\ \omega_r\\ \tau_g\\ x_{FA}\\ v_{FA}\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ -\omega_n^2 & -2\xi\omega_n & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -\frac{1}{N} & 1 & 0 & 0 & 0\\ 0 & 0 & \frac{K_{ls}}{NJ_g} & -\frac{D_{ls}}{N^2J_g} & -\frac{D_{ls}}{NJ_g} & -\frac{1}{J_g} & 0 & 0\\ \frac{\boldsymbol{f_{\tau_{r\beta}}}}{J_r} & 0 & -\frac{K_{ls}}{J_r} & -\frac{D_{ls}}{J_r} & -\frac{(D_{ls}+\boldsymbol{f_{\tau_{r\omega_r}}})}{NJ_r} & 0 & 0 & -\frac{\boldsymbol{f_{\tau_{rv_{eff}}}}}{J_r}\\ 0 & 0 & 0 & 0 & 0 & -\alpha_{gc} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ \frac{\boldsymbol{f_{F_{t\beta}}}}{m_{FA}} & 0 & 0 & 0 & -\frac{\boldsymbol{f_{F_{t\omega_r}}}}{m_{FA}} & 0 & -\frac{K_{FA}}{m_{FA}} & -\frac{(D_{FA}+\boldsymbol{f_{F_{tv_{eff}}}})}{m_{FA}}\end{bmatrix}\begin{bmatrix}\beta\\ \dot{\beta}\\ \theta\\ \omega_g\\ \omega_r\\ \tau_g\\ x_{FA}\\ v_{FA}\end{bmatrix} + \begin{bmatrix} 0 & 0 & 0\\ 0 & 0 & \omega_n^2\\ 0 & 0 & 0\\ 0 & 0 & 0\\ \frac{\boldsymbol{f_{\tau_r\_v_{eff}}}}{J_r} & 0 & 0\\ 0 & \alpha_{gc} & 0\\ 0 & 0 & 0\\ \frac{\boldsymbol{f_{F_t\_v_{eff}}}}{m_{FA}} & 0 & 0\end{bmatrix}\begin{bmatrix}v_{eff}\\ \tau_{g_{ref}}\\ \beta_{ref}\end{bmatrix} \quad (6:22)$$

$$\begin{bmatrix}\omega_g\\ \beta\\ P_{el}\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ \frac{\pi}{180} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \tau_{g\_op} & 0 & \omega_{g\_op} & 0 & 0\end{bmatrix}\begin{bmatrix}\beta\\ \dot{\beta}\\ \theta\\ \omega_g\\ \omega_r\\ \tau_g\\ x_{FA}\\ v_{FA}\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}v_{eff}\\ \tau_{g_{ref}}\\ \beta_{ref}\end{bmatrix} \quad (6:23)$$

The terms in red denote the varying partial derivatives with respect to wind speed and these are the varying parameters for the *LPV* wind turbine model.

LPV Model Discretisation:

The combined model is discretised using the Euler method. The discrete state-space matrices are as follows,

$$A_{discrete} = I_{nx} + T_s A_{continuous}$$
$$B_{discrete} = T_s B_{continuous} \quad (6:24)$$

where *A* and *B* are the augmented *LPV* wind turbine state and input matrix respectively.

Discrete LPV Model Validation:

As mentioned in the previous section to derive the *LPV* model of the wind turbine, the nonlinear terms (C_q and C_t coefficients lookup tables in this case) are first linearised, then interpolated and parameterised by wind speed to yield the corresponding deviations model along these optimal curves (see following equation).

$$\begin{aligned} \delta x_{0k+1} &= x_{0k} + A_{0k}(v_{eff})(x_k - x_{0k}) + B_{0k}(v_{eff})(u_k - u_{0k}) \\ \delta y_{0k} &= C_{0k}(v_{eff})(x_k - x_{0k}) \end{aligned} \tag{6:25}$$

In the state of equilibrium these derivative terms will be zero, however this will not be the case in the occasion where there are small discrepancies between the actual system states and the ones derived from the linearised lookup tables. To increase accuracy the model is complimented by maintaining the derivative terms. Finally by adding the nominal (at the corresponding equilibrium point) vectors for the states, inputs and outputs the actual output of the model is obtained in the following equation (recall Modified-Jacobian linearization discussed in Section 3.2.2).

$$\begin{aligned} \delta x_{0k+1} &= \big(f(x_0, u_0) - A_{0k}(v_{eff})x_{0k} - B_{0k}(v_{eff})u_{0k}\big) + A_{0k}(v_{eff})x_k \\ &\quad + B_{0k}(v_{eff})u_k \\ \delta y_{0k} &= \big(g(x_0, u_0) - C_{0k}(v_{eff})x_{0k}\big) + C_{0k}(v_{eff})x_k \end{aligned} \tag{6:26}$$

The model was validated using a series of step changes in wind speed and the results are presented in the following two figures (1st figure: zooming in, 2nd figure: full scale) in the following figures. The blue trends represent responses for the original nonlinear model whereas the green trends represent responses for the *LPV* adaptation.

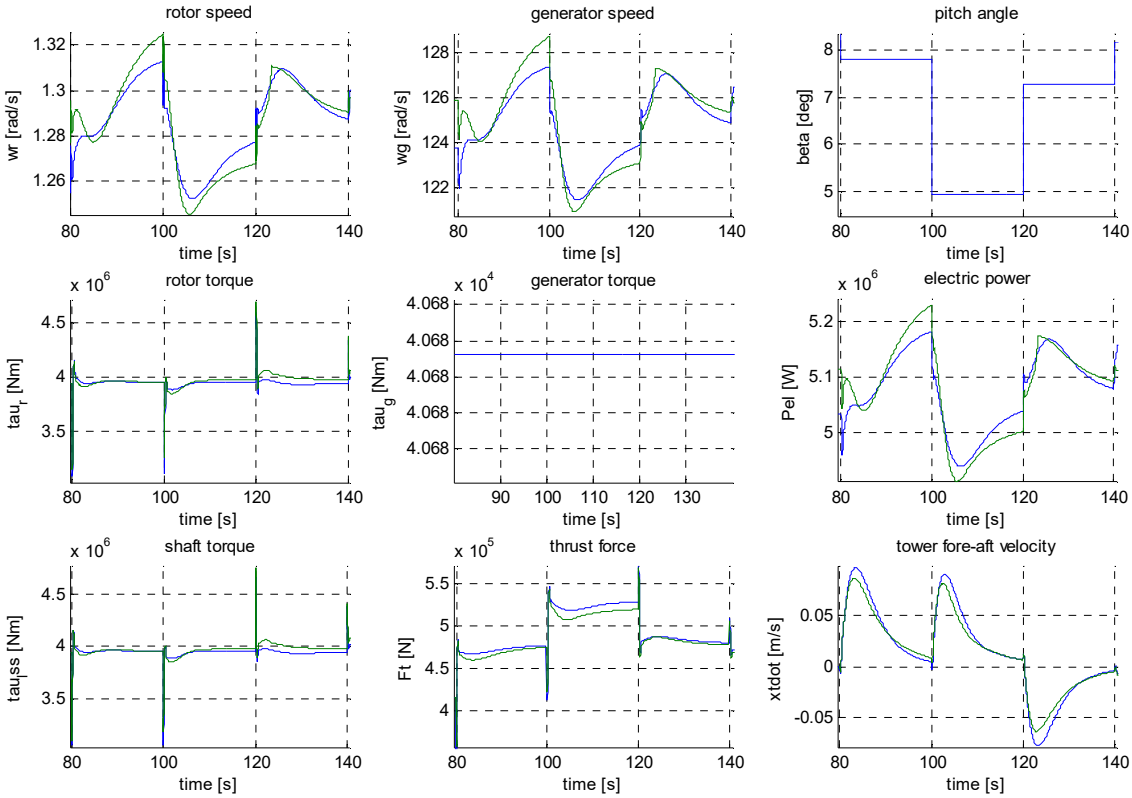


Figure 6:11 Discrete LPV model validation against nonlinear Wind Turbine model (zoom in).

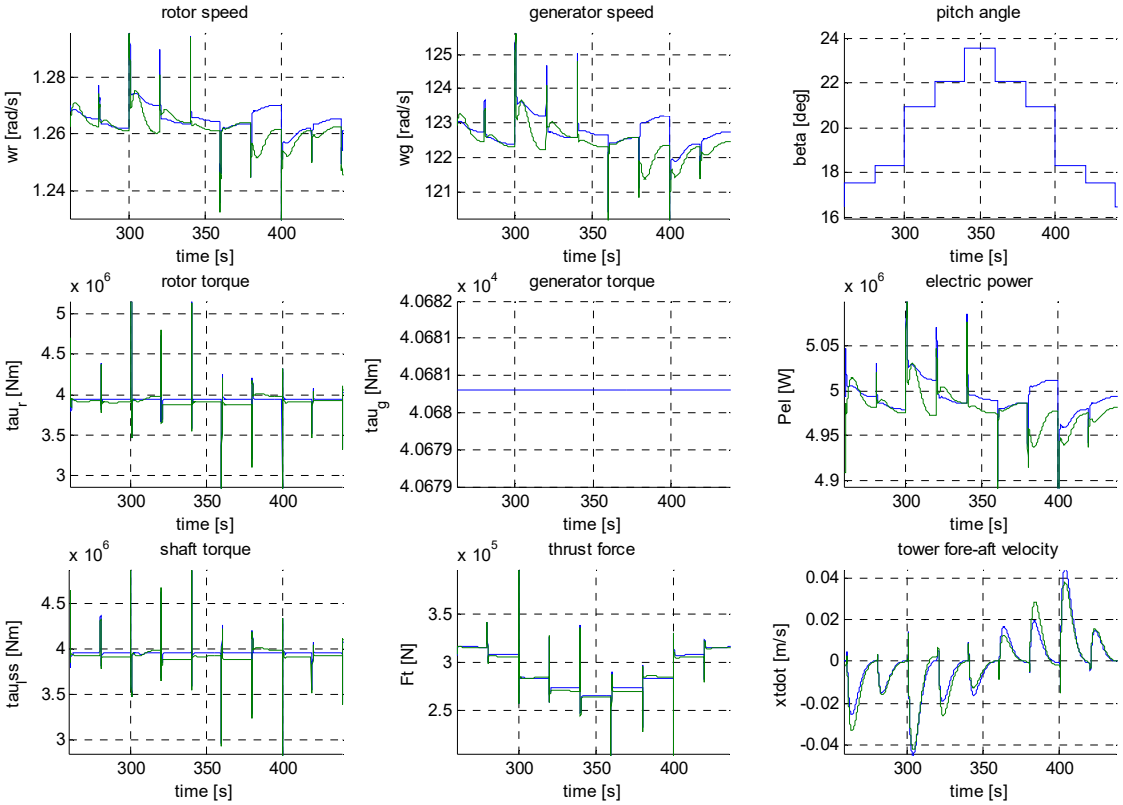


Figure 6:12 Discrete *LPV* model validation against nonlinear Wind Turbine model (full scale).

6.3 Control System Description

6.3.1 Kalman Filter Formulation and Validation

The first right-hand side term in Equation 6:25 (current operating point) is obtained using an *EKF* estimator. The wind turbine original nonlinear model equations are used to provide the current state and output vectors required for the state correction whereas the full *LPV* model is used to provide the state matrices required for the covariance matrix and state predictions. The *EKF* equations are summarised below.

$$P_{k+1} = A_{0k}(v_{eff})P_k A_{0k}^{-1}(v_{eff}) + D_{0k}(v_{eff})Q_N D_{0k}^{-1}(v_{eff}) \tag{6:27}$$

$$K_f = P_k C_{0k}^{-1}(v_{eff})\left(C_{0k}(v_{eff})P_k C_{0k}^{-1}(v_{eff}) + R_N\right)^{-1} \tag{6:28}$$

$$P_k = P_k - K_f C_{0k}(v_{eff})P_k \tag{6:29}$$

$$x_{k+1} = x_k + K_f(y_k - \hat{y}_k)$$
$$\hat{y}_k = g_{NL}(x_k) \tag{6:30}$$

The *EKF* is validated using a series of steps and the results for selected channels are shown in the following graphs. The state estimation problem is not extensively treated here and the EKF is only used to validate its formulation within the *LPV-NPGMV* therefore practically the states are considered available.

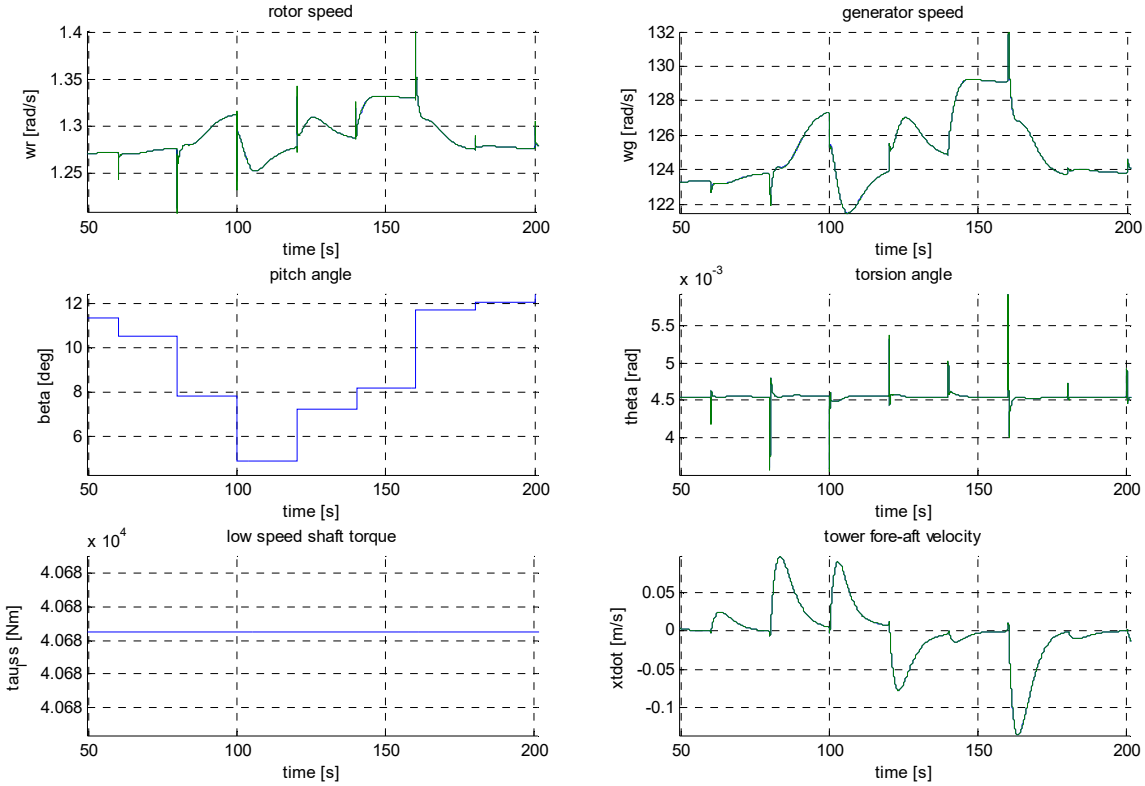


Figure 6:13 *Discrete Extended Kalman Filter* (*EKF*) estimates validation against nonlinear model states.

6.3.2 Controller Formulation

The optimal controller follows the structure that was described in Equation 4:90 where $F_{ck,N}$ is the control weighting matrix, $\hat{E}_{pt+k_0,N}$ the estimated vector of weighted errors, Λ the *NGPC* scalar control as specified within the modified *GMV* cost function, all defined within the specified prediction horizon. Note here, in this formulation of the cost function, the flexibility of three tuning factors becomes available.

This is a full *LPV* formulation and the nonlinear system is approximated in its entirety, by the *LPV* model which is parameterised effectively by wind speed. Therefore $W_{1k,N}$ for the purpose of this implementation is defined as a unity matrix. Note that in the previous application examples $W_{1k,N}$ was used as a *black-box* term to encapsulate the, what was assumed, unknown nonlinearities of the system. The model derived at each iteration is a

deviations model (δu) and therefore the nominal state and output values are internally added to the controller algorithm as biases to produce the actual absolute control signal $U_{t,N}$.

The *LPV* model states are introduced to the controller via the estimated vector of weighted errors. The dynamic error weighting is also included within $\hat{E}_{pt+k_0,N}$ and it was defined using the *PID* based approach described in Section 2.2.2. As with the previous applications, it has been noted that after a satisfactory tuning parameters configuration has been established the *NPGMV* controller performed adequately for most scenarios with minimal requirement for retuning. The only factor that was mostly used for experimentation in the following section was the prediction horizon.

A classical *PID* and basic state-space *NGMV* were used as baseline controllers to assess performance of the *LPV-NPGMV*. For the constrained case the *Hildreth Quadratic Programming* algorithm was used (Wang, 2009) to provide an efficient solution of the predictive controller against absolute constraints on blade pitch angle *β* (*SISO* case only). Being a *Feedback-Feedforward* architecture as shown in Figure 6:3, made the application of rate constraints less meaningful as the action from the *Feedforward* term, added in the output of the *LPV-NPGMV Feedback* controller, will respond to disturbances much faster than the output of the latter.

For this application a slightly different approach is used in the internal architecture of the *NGMV* controller and more specifically in how the overall plant subsystems are formulated. Unlike the *Dynamic Positioning* study, where the *black-box* operator encapsulated the total nonlinear plant dynamics and constraints whereas the linear subsystem was defined as a default unity operator, here the *black-box* operator only contains actuator constraints for the pitch (or pitch and generator torque for the *MIMO* formulation) whereas the linear subsystem is now an *LPV* formulation that includes the plant dynamics in their totality. This can be seen more clearly in the following figure.

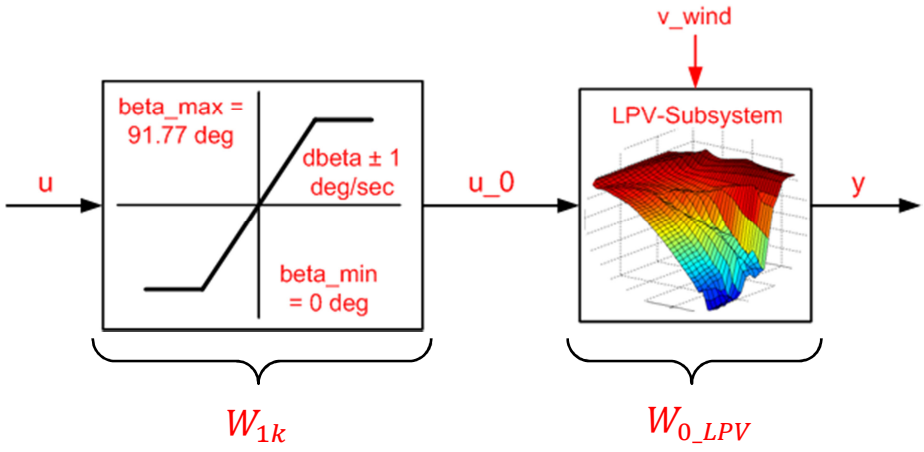


Figure 6:14 Nonlinear pant decomposition into hard nonlinear and *LPV* approximated components.

It is important to observe here that both subsystems are essentially nonlinear. Details of these subsystems are given in the following sections.

Nonlinear Subsystem (*black-box* operator):

W_{1k} is simply defined as the pitch angle and generator torque absolute and rate limits as seen below.

$$0deg \leq \beta \leq 91.77deg$$

$$-1deg/s \leq \dot{\beta} \leq 1deg/s$$

$$0Nm \leq \tau_g \leq 4.3094 \cdot 10^4 Nm$$

$$-100Nm/s \leq \dot{\tau_g} \leq 100Nm/s$$

Note that limits are not implemented in all simulations presented in the following section for experimentation purposes.

***LPV* Subsystem:**

The total *LPV* subsystem is described in Equation xx where terms in red denote the varying parameters which are the partial derivatives $f_{\tau_{r_\beta}}$, $f_{\tau_{r_{\omega_r}}}$, $f_{\tau_{r_{v_{eff}}}}$, $f_{F_{t_\beta}}$, $f_{F_{t_{\omega_r}}}$, $f_{F_{t_{v_{eff}}}}$, $f_{\tau_{r\_v_{eff}}}$ and $f_{F_{t\_v_{eff}}}$ with respect to the varying efficiency coefficients derivatives which in turn are parameterised by the pitch angle and tip-speed ratio i.e. $f_\tau\left(dC_q(\lambda,\beta)\right)$ and $f_{F_t}\left(dC_t(\lambda,\beta)\right)$. Variation in these derivatives with respect to the pitch angle and tip speed ratio is shown in the following figures.

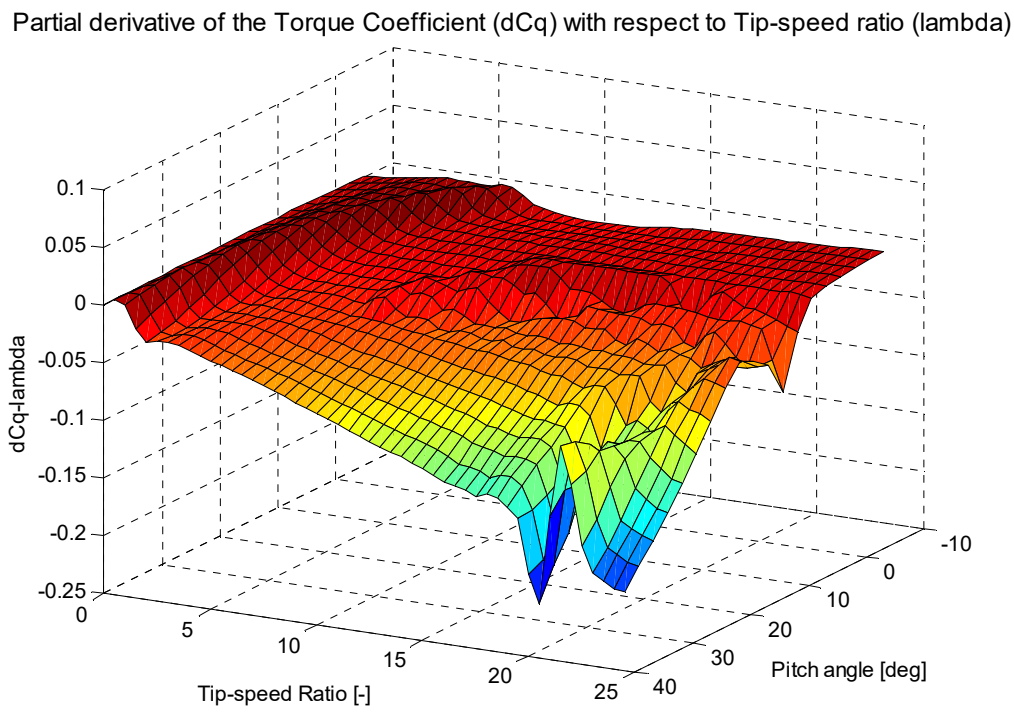


Figure 6:15 Torque coefficient variation with respect to Tip-speed ratio.

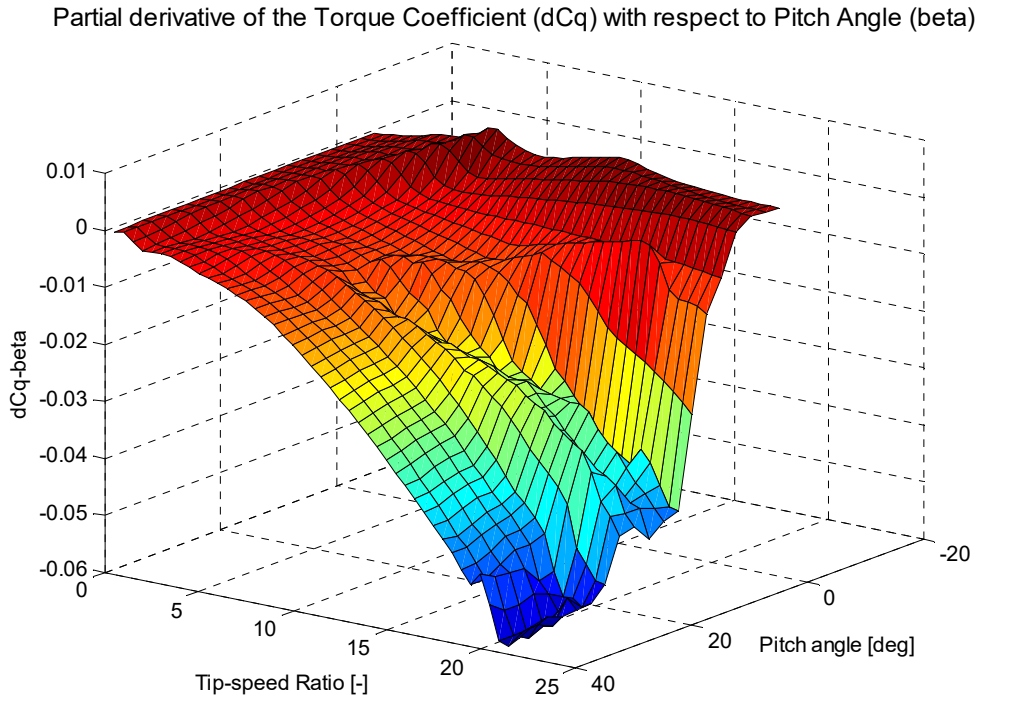


Figure 6:16 Torque coefficient variation with respect to Pitch angle.

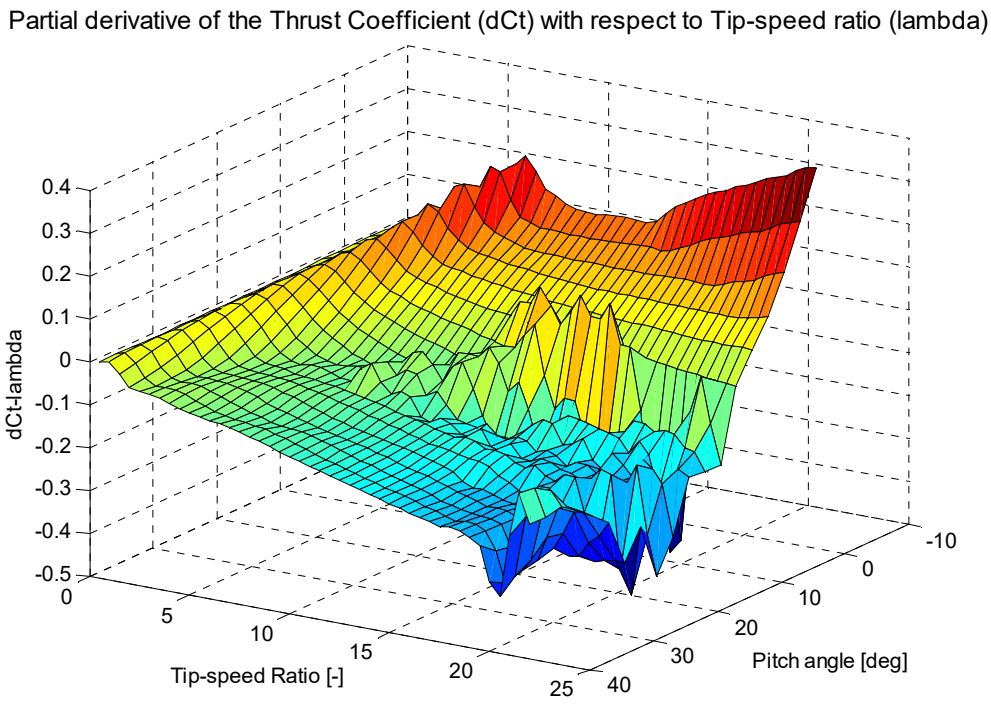


Figure 6:17 Thrust coefficient variation with respect to Tip-speed ratio.

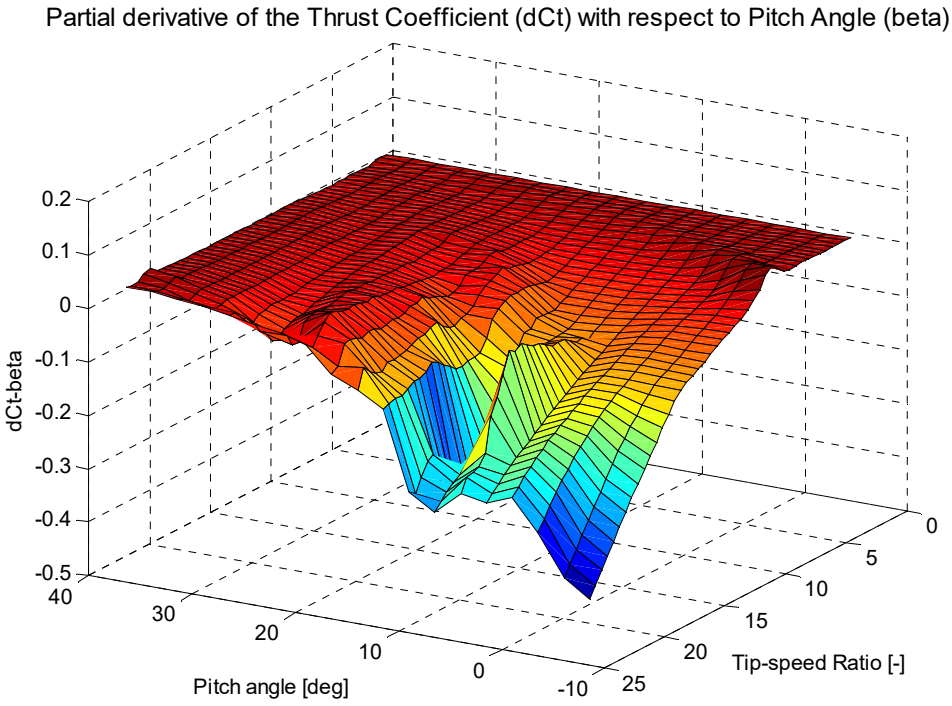


Figure 6:18 Thrust coefficient variation with respect to Pitch angle.

The matrices of the *LPV* subsystem will be recalculated and updated at each iteration of the controller at an experimental sample time of *Ts = 0.02sec.* This is the model used within the *NPGMV* to generate future predictions.

Linear Plant, Reference and Disturbance Subsystems:

Within the *LPV-NPGMV* formulation, the definition of a linear subsystem is maintained for generality in case there are linear components we want to include, however for this experimentation it is defined as a unity operator by default which can either be a scalar or a *2x2* matrix for the *SISO* and the *MIMO* cases respectively. The same applies for the reference and disturbance subsystems as here all dynamical characteristics are fully captured by the *LPV* subsystem.

$$W_{0,r,d} \rightarrow SISO\colon 1, MIMO\colon \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

NPGMV Weighting Definition and Prediction Horizons:

In this case the error weighting P_c was selected with respect to the stabilising *PID* tuning method as a starting point (as described in Section xx) has the following discrete *TF* for the *SISO* and *MIMO* case respectively. The baseline *PID* controllers share the same gains.

$$SISO\colon P_{c\_\beta} \rightarrow \frac{6.05 \cdot 10^{-5} - 6 \cdot 10^{-5} z^{-1}}{1 - z^{-1}}$$

$$MIMO\ (Pitch\ Control\ Channel)\colon P_{c\_\beta} \rightarrow \frac{6 \cdot 10^{-6} - 5 \cdot 10^{-6} z^{-1}}{1 - z^{-1}}$$

$$MIMO\ (Generator\ Torque\ Control\ Channel)\colon P_{c\_\tau_g} \rightarrow \frac{0.061 - 0.06 z^{-1}}{1 - z^{-1}}$$

The general *PID TF* that was used for the P_c definition can be seen below.

$$P_{c\_PID} \rightarrow \frac{K_P + K_I z^{-1}}{1 - a z^{-1}}$$

where K_P and K_I are the *PID* Proportional and Integral gains respectively and a is a near integrator multiplier to ensure stability.

In this study the control weighting F_{ck} was found to give a better performance when defined as a simple scalar gain. Different prediction horizons where found to yield better performance in different scenarios and these can be seen in the following sections were results are presented.

The following section presents the graphical simulation results, accompanied by the corresponding assessment metrics for various scenarios, each for both the *SISO* and *MIMO* cases described in Figures 6:3 and 6:4.

6.4 Simulation Results

Here the three controllers described in the previous section are tested in various scenarios within the above rated operating region. These include different types of wind speed variations (disturbance rejection) such as step changes, gusts and high frequency variations but also power reference variations (tracking). In the later the knowledge of future input signals option within the *NPGMV* controller is used. Two basic metrics are used to quantify these results and assist with assessing performance for each controller. These are the normalised *STD* of the *MV*s and the *MISE* for the *CV*s as shown below.

$$MISE = \frac{1}{N}\sum_{i=0}^{N}(y_{SP\,i} - y_{meas_i})^2 \tag{6:31}$$

where *y* is either electric power *P* or generator speed ω_g.

$$STD = \left(\frac{1}{N}\sum_{i=0}^{N}(u_i - \bar{u})^2\right)^{1/2} \tag{6:32}$$

where *u* is either blade pitch angle β or generator torque τ_g.

In the following section only the set of most successful results is presented. Note here that when the best set of cost weightings has been established for the *LPV-NPGMV* only minor adjustments were required to achieve reasonable performance across all scenarios. It is safe to say that the latter practically remained unaltered whilst the internal system model within the controller varied with speed.

6.4.1 Disturbance Rejection

Scenario 1a: Step of +0.5m/s in wind speed (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is used as a disturbance to examine control compensation by the three controllers. The graphs were focused at the point where the positive step change of 0.5m/s in wind speed occurs for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. Figures 6:14 and 6:15 capture the results for the *SISO* and *MIMO* control structures respectively.

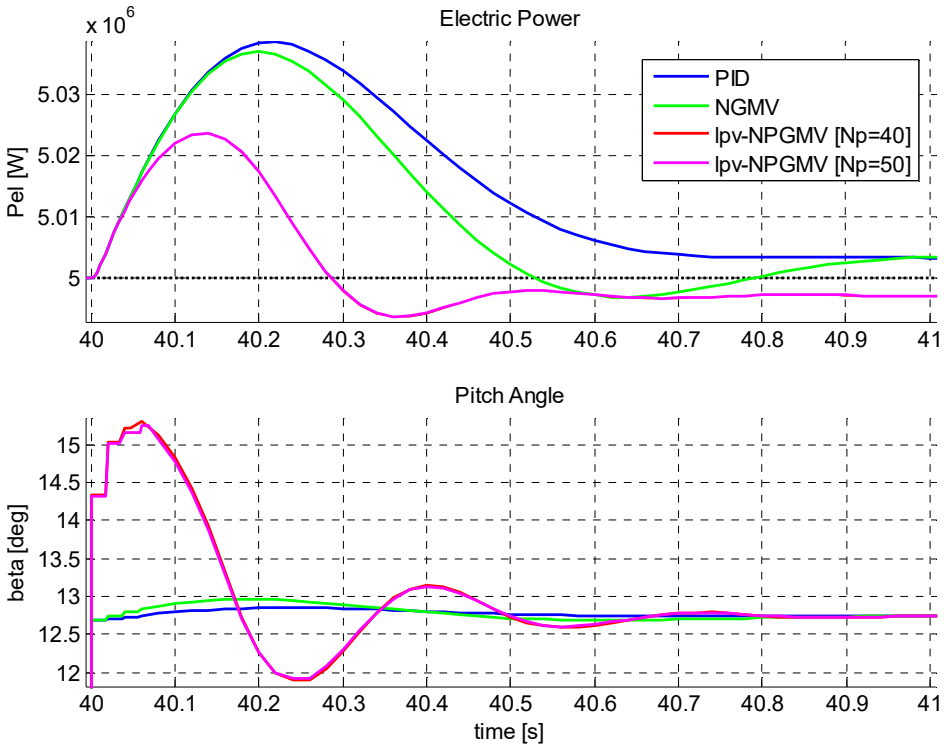


Figure 6:19 Scenario 1a results for the *SISO* control structure.

Table 6:1: Scenario 1a, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	0.8921	1
NGMV	0.8969	0.7950
LPV-NPGMV [*Np*=40]	0.9896	0.2649
LPV-NPGMV [*Np*=50]	1	0.2886

The *LPV-NPGMV* [*Np*=40] provides the highest power regulation benefit with an improvement of 73.5% in *MISE* relative to the baseline *PID* and 66.6% relative to the baseline *NGMV*. However this is achieved at the expense of increase in control action.

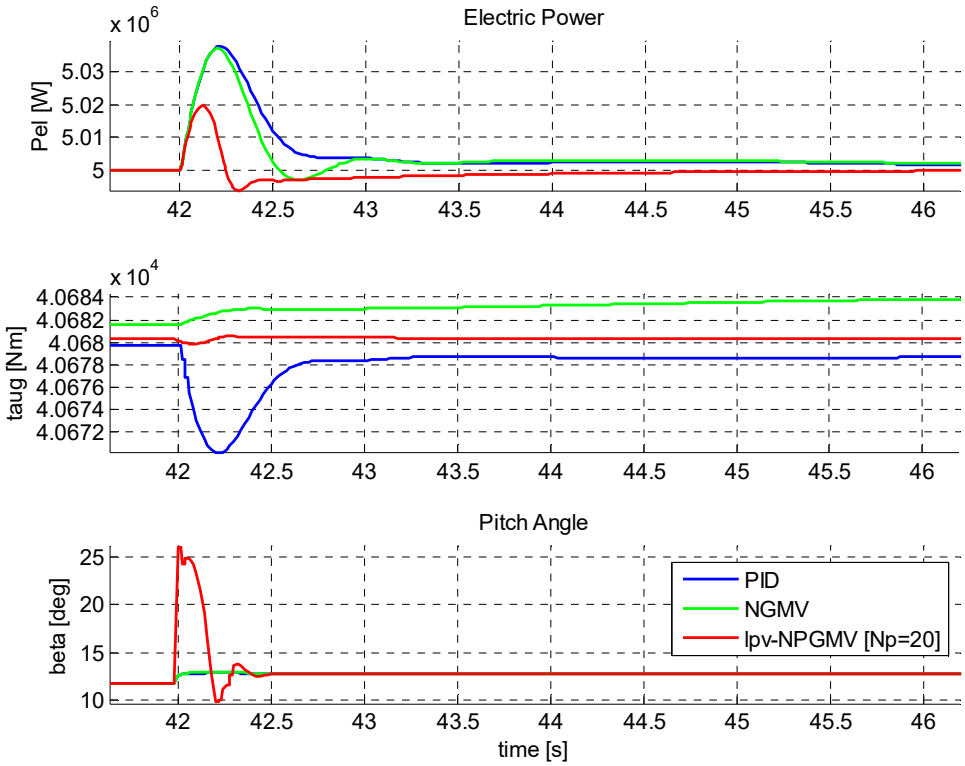


Figure 6:20 Scenario 1a results for the *MIMO* control structure.

Table 6:2: Scenario 1a, MIMO quantified controller comparison

Controllers	**τg_{STD} (norm.)**	**β_{STD} (norm.)**	**ωg_{MISE} (norm.)**	**Pel_{MISE} (norm.)**
PID	0.8330	0.4028	1	1
NGMV	1	0.4046	0.7750	0.8250
LPV-NPGMV [*Np*=20]	0.0471	1	0.1475	0.1544

The *LPV-NPGMV* [*Np*=20] provides the highest power regulation benefit with an improvement of 85.2% in generator speed and 84.5% in electric power *MISE* relative to the baseline *PID* and 80.9% and 80% respectively relative to the baseline *NGMV*. Furthermore with this configuration a 93.7% and 94.8% reduction in generator torque actuation is achieved relative to the *PID* and the *NGMV* respectively.

Scenario 1b: Step of +3m/s in wind speed (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is used as a disturbance to examine control compensation by the three controllers. The graphs were focused at the point where a larger positive step change of 3m/s in wind speed occurs for a clearer examination of

recovery provided by the three controllers, acting to keep the power output at nominal value. Figure 6:21 captures the results for the *SISO* control structure. The *MIMO* structure for the *LPV-NPGV* was found not to yield better performance and for this reason it was omitted.

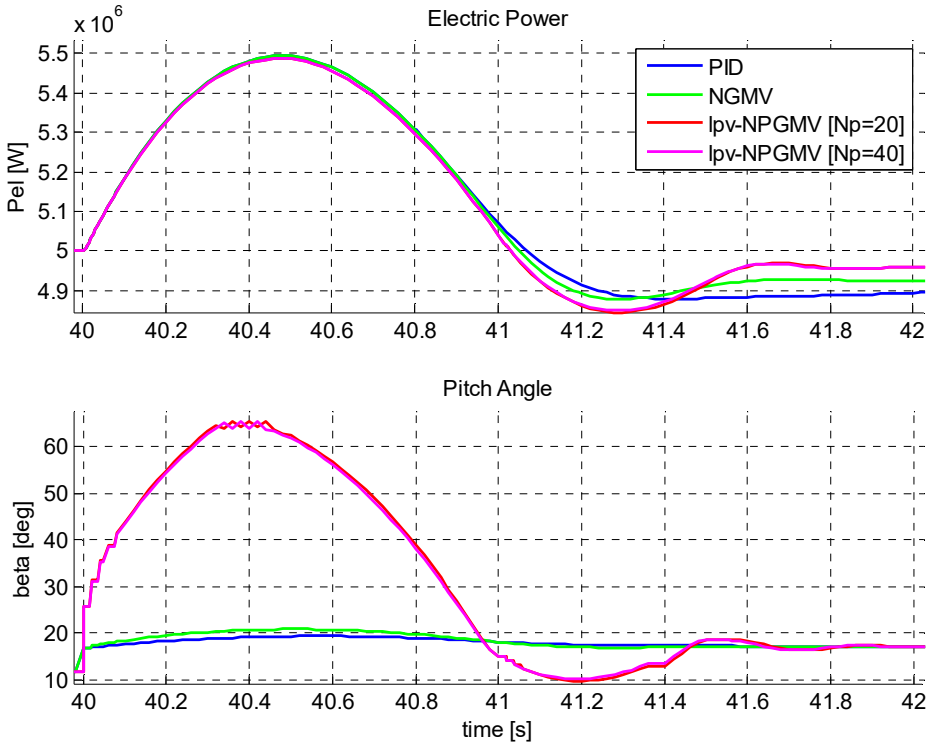


Figure 6:21 Scenario 1b results for the *SISO* control structure.

Table 6:3: Scenario 1b, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	0.3885	1
NGMV	0.3957	0.9423
LPV-NPGMV [*Np*=20]	1	0.8894
LPV-NPGMV [*Np*=40]	0.9897	0.8886

The *LPV-NPGMV* [*Np*=40] provides the highest power regulation benefit with an improvement of 11.1% in *MISE* relative to the baseline *PID* and 5.6% relative to the baseline *NGMV*. However this is achieved at the expense of increase in control action. A decrease in performance is observed here for larger wind variations.

Scenario 2: Step of -0.5m/s in wind speed (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is used as a disturbance to examine control compensation by the three controllers. The graphs were focused at the point where a negative step change of 0.5m/s in wind speed occurs for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. Figure 6:22 captures the results for the *SISO* control structure. The *MIMO* structure for the *LPV-NPGV* was found not to yield better performance and for this reason it was omitted.

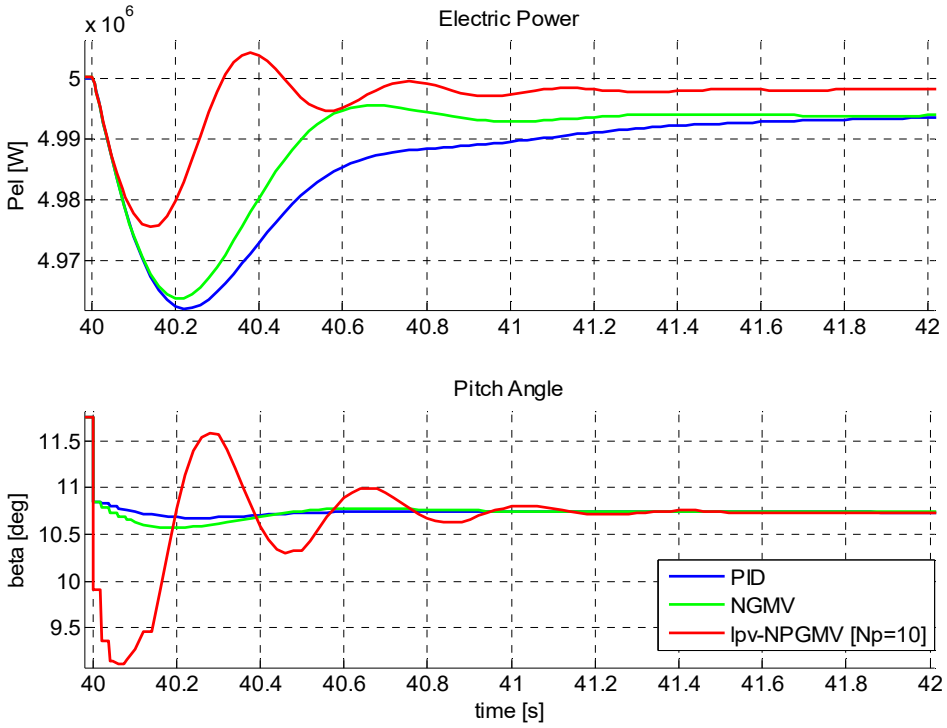


Figure 6:22 Scenario 2 results for the *SISO* control structure.

Table 6:4: Scenario 2, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	0.9444	1
NGMV	0.9490	0.7613
LPV-NPGMV [*Np*=10]	1	0.1891

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 81% in *MISE* relative to the baseline *PID* and 75.1% relative to the baseline *NGMV*. This is achieved at the expense of increase in control action.

Scenario 3b: Large gust variation of 13-19m/s (nominal wind speed at 15m/s)

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) in the form of a gust is used as a disturbance to examine control compensation by the three controllers. The graphs were focused at the window where a larger gust variation of 13-19m/s occurs for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. Figure 6:23 captures the results for the *SISO* structure respectively.

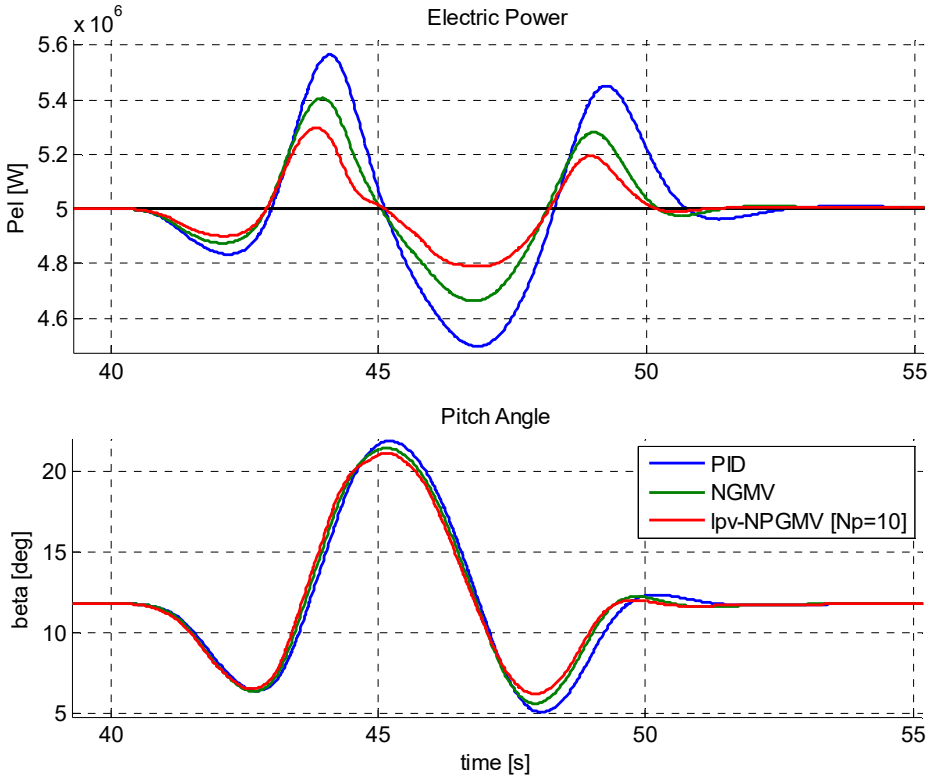


Figure 6:23 Scenario 3b results for the *SISO* control structure.

Table 6:7: Scenario 3b, SISO quantified controller comparison

Controllers	**β_{STD} (norm.)**	**Pel_{MISE} (norm.)**
PID	1	1
NGMV	0.8908	0.3201
LPV-NPGMV [Np=10]	0.9308	0.1984

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 80% in *MISE* relative to the baseline *PID* and 12.17% relative to the baseline *NGMV*. Here a small improvement in control action variability reduction was observed (6% relative to the baseline *PID*).

Scenario 4: Stochastic wind variation between 12-21m/s:

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is disturbed by a stochastic high frequency component to examine control compensation by the three controllers. The graphs were focused at a long enough time window, to allow an overview of the recovery provided by the three controllers, acting to keep the power output at nominal value. Figures 6:24 and 6:25 capture the results for the *SISO* and *MIMO* control structures respectively.

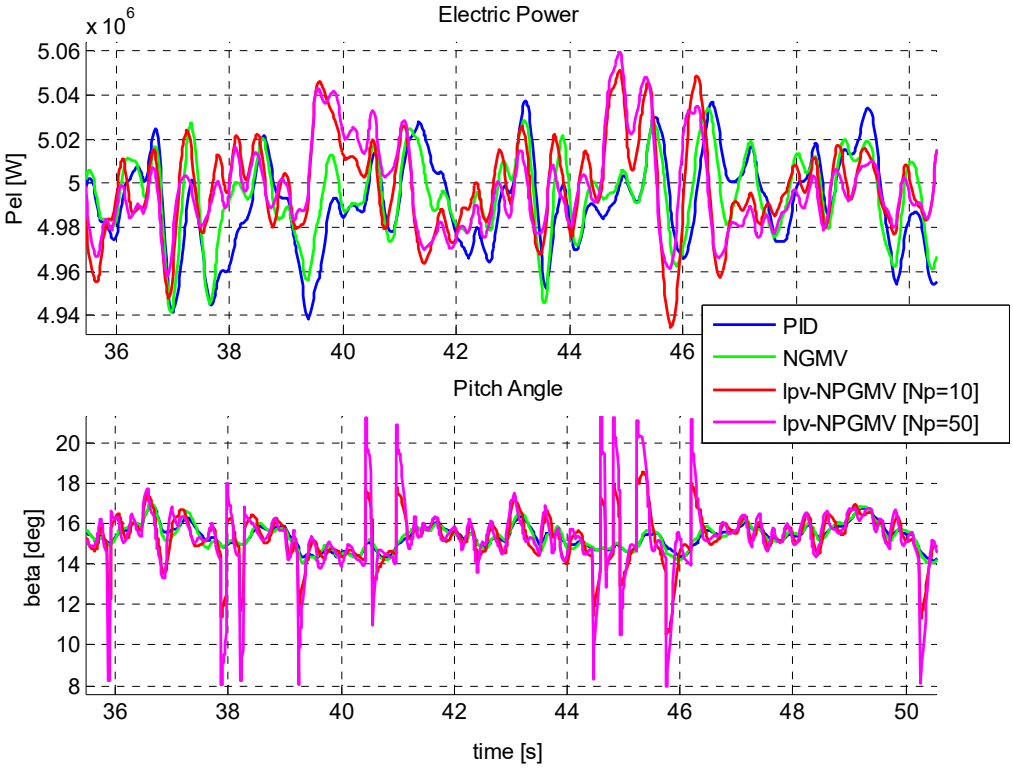


Figure 6:24 Scenario 4 results for the *SISO* control structure.

Table 6:9: Scenario 4, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	0.6005	1
NGMV	0.7799	0.6642
LPV-NPGMV [*Np*=10]	0.8170	0.6291
LPV-NPGMV [*Np*=50]	1	0.3772

The *LPV-NPGMV* [*Np*=50] provides the highest power regulation benefit with an improvement of 62.2% in *MISE* relative to the baseline *PID* and 43.2% relative to the baseline *NGMV*. Longer prediction horizons were found to be more advantageous in this type of disturbance.

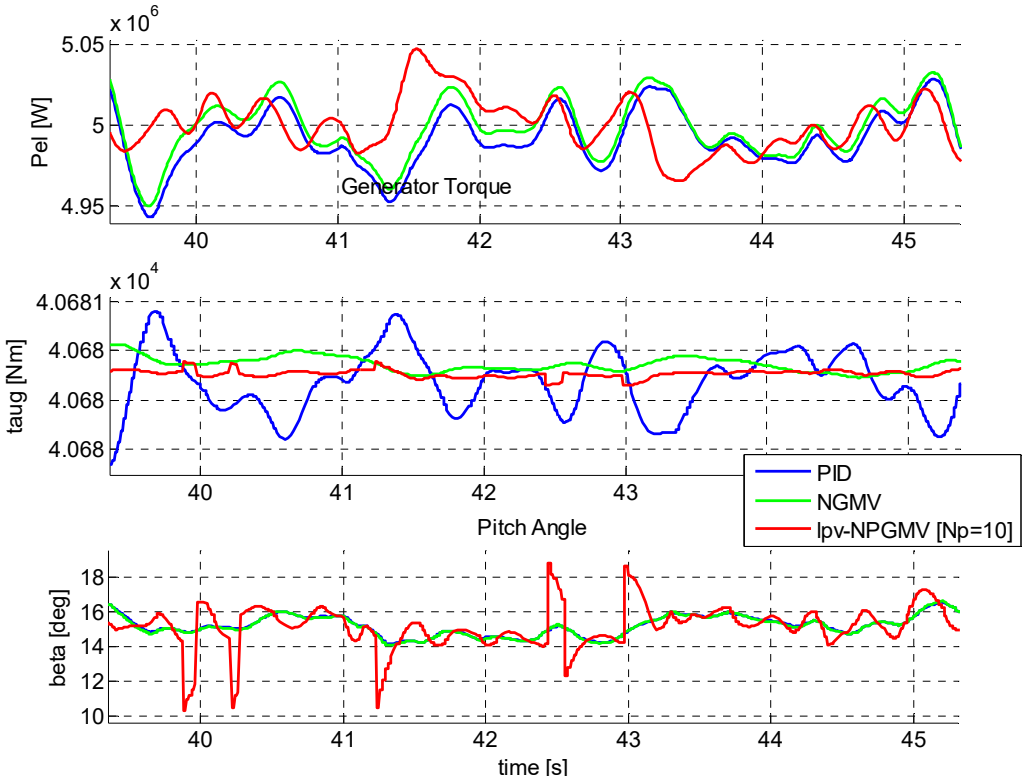


Figure 6:25 Scenario 4 results for the *MIMO* control structure.

Table 6:10: Scenario 4, MIMO quantified controller comparison

Controllers	**τg_{STD} (norm.)**	**β_{STD} (norm.)**	**ωg_{MISE} (norm.)**	**Pel_{MISE} (norm.)**
PID	1	0.9320	1	1
NGMV	0.9985	0.9414	0.7366	0.7418
LPV-NPGMV [*Np*=10]	0.9913	1	0.6446	0.6511

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 35.5% in generator speed and 34.8% in electric power *MISE* relative to the baseline *PID* and 12.4% and 12.2% respectively relative to the baseline *NGMV*. Reduction in control action has been observed at a 0.87% and 0.72% in generator torque relative to the *PID* and the *NGMV* respectively.

6.4.2 Reference Tracking

Scenario 5: Step of -4% of the rated power:

For this scenario wind speed is kept constant at above rated in order to provide rated power availability (5MW). A negative step change is then introduced to the power reference of the turbine (de-rating). The graphs were focused at the point where a negative step change of 4% of the rated power occurs for a clearer examination of tracking provided by the three controllers. Figure 6:26 captures the results for the *SISO* control structure. The *MIMO* structure for the *LPV-NPGV* was found not to yield better performance and for this reason it was omitted.

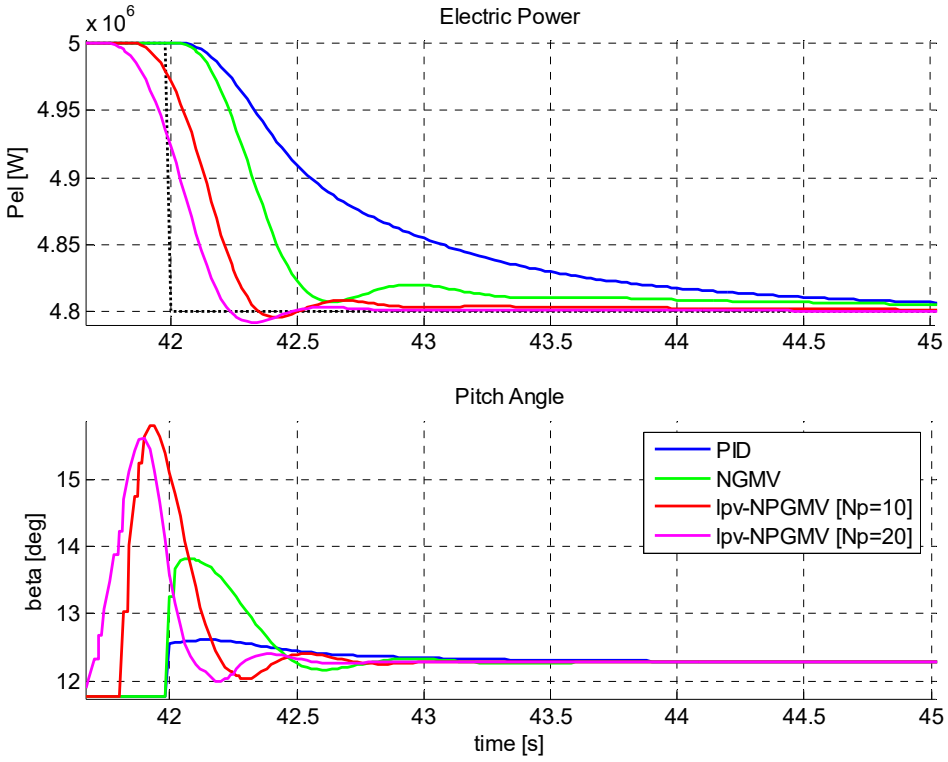


Figure 6:26 Scenario 5 results for the *SISO* control structure.

Table 6:11: Scenario 5, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	0.6800	1
NGMV	0.8075	0.8427
LPV-NPGMV [*Np*=10]	1	0.1885
LPV-NPGMV [*Np*=20]	0.9376	0.0755

Here a consistent correlation has been observed between increase in performance and increase of prediction horizon. This was expected as future setpoint information was assumed

available to fully explore the capabilities of the advanced predictive controller. The *LPV-NPGMV* [*Np*=20] provides the highest power regulation benefit with an improvement of 92.4% in *MISE* relative to the baseline *PID* and 91% relative to the baseline *NGMV*.

Scenario 6: Sequence of steps in power between 0-2MW and stochastic variation:

For this scenario wind speed is kept constant at above rated in order to provide rated power availability (5MW). A series of steps in power reference (between 0-2MW) are then introduced in combination with high frequency stochastic wind variation for disturbance. Figure 6:27 captures the results for the *SISO* control structure across all step changes whereas Figure 6:28 focuses on a specific step change for clarity. This scenario combined both disturbance rejection and tracking experimentation. The *MIMO* structure for the *LPV-NPGV* was found not to yield better performance and for this reason it was omitted.

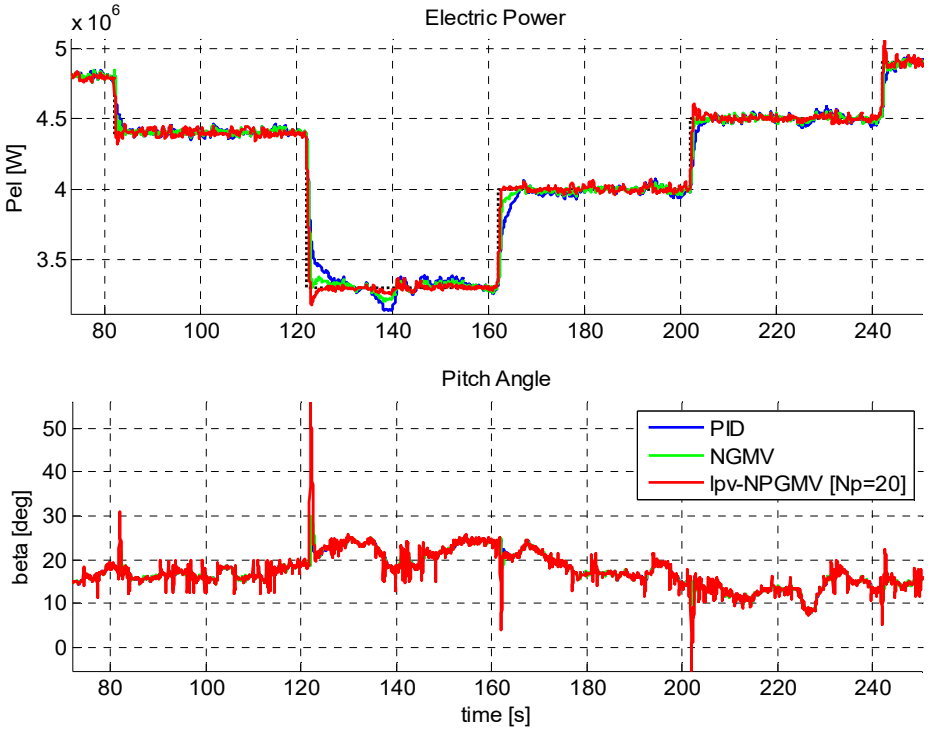


Figure 6:27 Scenario 6 results for the *SISO* control structure.

Table 6:12: Scenario 6, SISO quantified controller comparison

Controllers	β_{STD} **(norm.)**	Pel_{MISE} **(norm.)**
PID	0.8865	1
NGMV	0.9135	0.6397
LPV-NPGMV [*Np*=20]	1	0.3085

The *LPV-NPGMV* [*Np*=20] provides the highest power regulation benefit with an improvement of 69.1% in *MISE* relative to the baseline *PID* and 51.7% relative to the baseline *NGMV*.

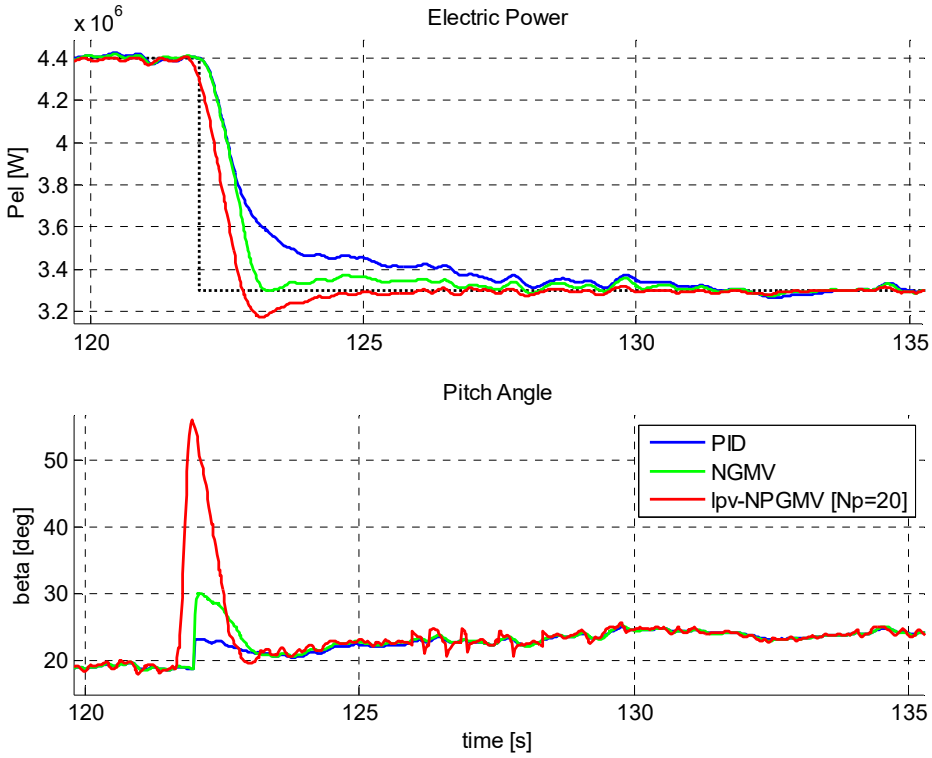


Figure 6:28 Scenario 6 results for the *SISO* control structure (zoom in).

6.4.3 Constraint Handling – Disturbance Rejection

Scenario 7a: Step of +3m/s in wind speed (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is used as a disturbance to examine control compensation by the three controllers only for the *SISO* control structure. The graph were focused at the point where a positive step change of 3m/s in wind speed occurs for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. To explore performance of the *LPV-NPGMV* in the presence of constraints the pitch angle actuator range was limited between 0-20deg.

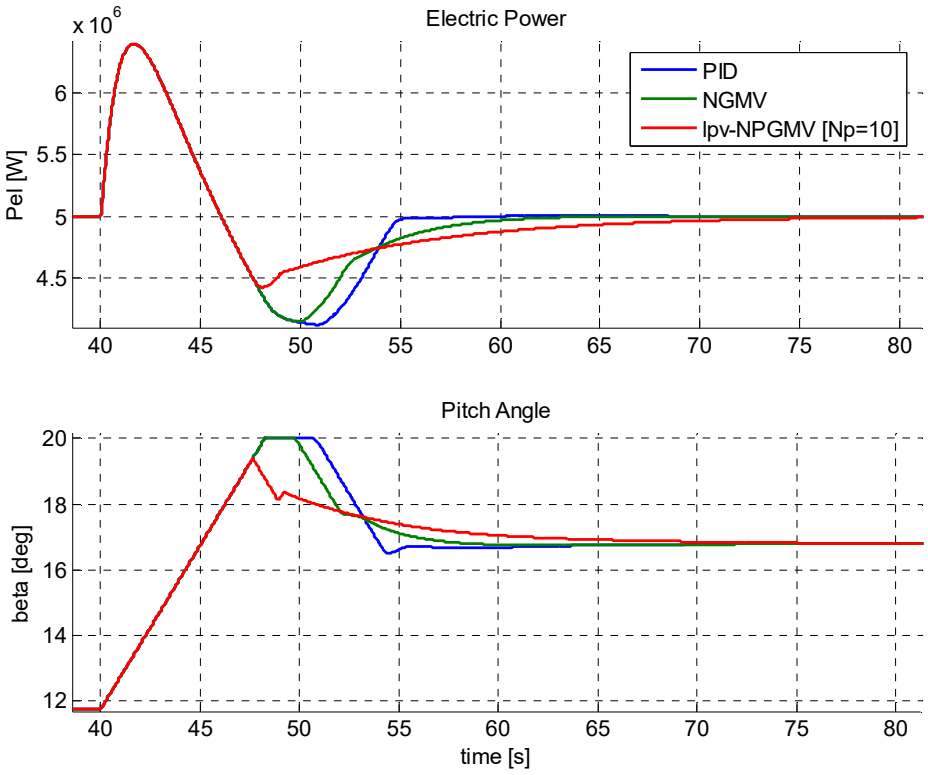


Figure 6:29 Scenario 7a results for the *SISO* control structure.

Table 6:13: Scenario 7a, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	1	1
NGMV	0.9646	0.9386
LPV-NPGMV [*Np*=10]	0.9377	0.7587

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 24.13% in *MISE* relative to the baseline *PID* and 17.9% relative to the baseline *NGMV*. The absolute constraint optimiser is enabled in this case (via quadratic programming) and therefore an improvement of 6.23% and 2.69% in control action is achieved relative to the *PID* and *NGMV* respectively.

Scenario 7b: Step of -2m/s in wind speed (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) is used as a disturbance to examine control compensation by the three controllers only for the *SISO* control structure. The graph were focused at the point where a negative step change of 2m/s in wind speed occurs

for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. To explore performance of the *LPV-NPGMV* in the presence of constraints the pitch angle actuator range was limited between 0-20deg.

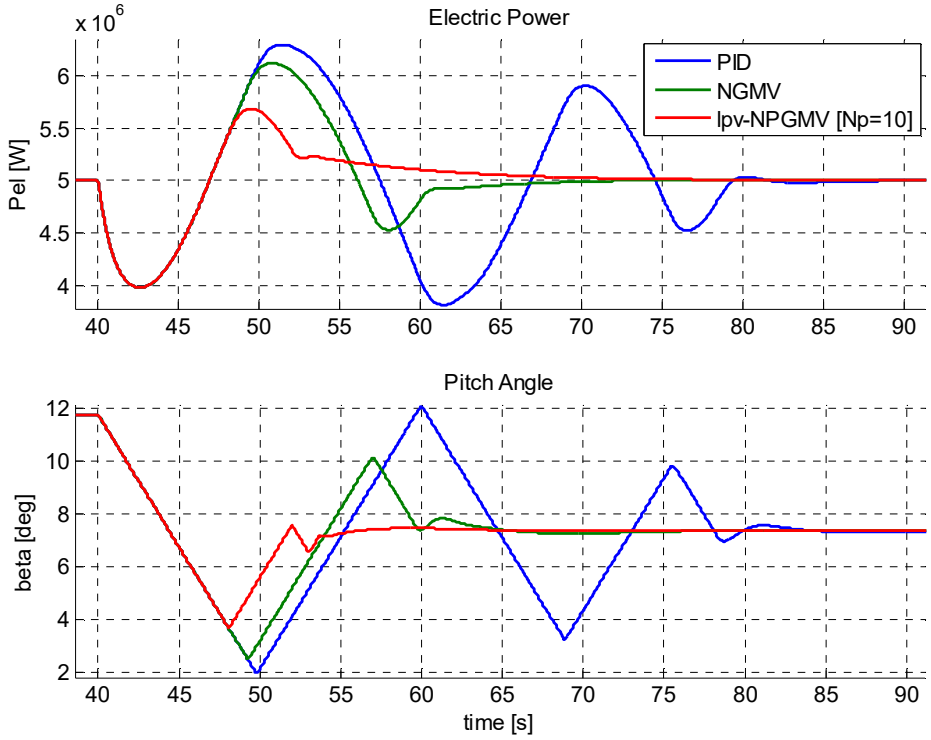


Figure 6:30 Scenario 7b results for the *SISO* control structure.

Table 6:14: Scenario 7b, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	1	1
NGMV	0.8099	0.6454
LPV-NPGMV [*Np*=10]	0.7217	0.2440

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 75.6% in *MISE* relative to the baseline *PID* and 40% relative to the baseline *NGMV*. The absolute constraint optimiser is enabled in this case (via quadratic programming) and therefore an improvement of 27.83% and 8.82% in control action is achieved relative to the *PID* and *NGMV* respectively.

Scenario 7c: Large gust variation of 13-19m/s (nominal wind speed at 15m/s):

For this scenario the power reference to the turbine is kept constant at nominal value (5MW) whereas wind speed variation (around a nominal value of 15m/s) in the form of a gust is used as a disturbance to examine control compensation by the three controllers only for the *SISO* control structure. The graphs were focused at the window where a gust variation of 13-19m/s occurs for a clearer examination of recovery provided by the three controllers, acting to keep the power output at nominal value. To explore performance of the *LPV-NPGMV* in the presence of constraints the pitch angle actuator range was limited between 0-20deg.

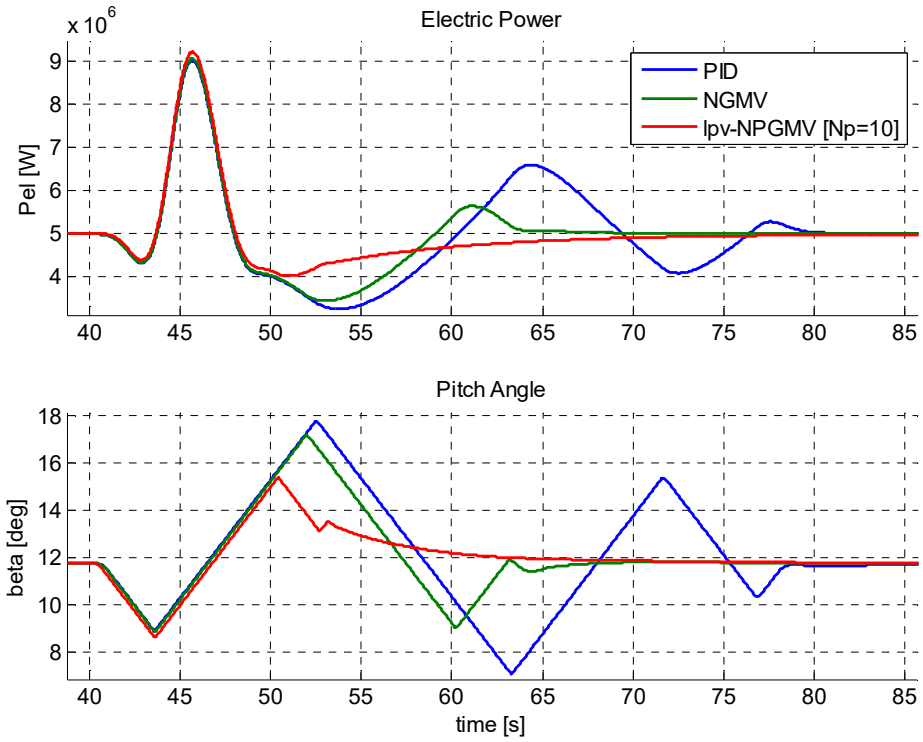


Figure 6:31 Scenario 7c results for the *SISO* control structure.

Table 6:15: Scenario 7c, SISO quantified controller comparison

Controllers	**β_{STD} (norm.)**	**Pel_{MISE} (norm.)**
PID	1	1
NGMV	0.7220	0.7470
LPV-NPGMV [*Np*=10]	0.4714	0.6436

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 35.64% in *MISE* relative to the baseline *PID* and 10.34% relative to the baseline *NGMV*. The absolute constraint optimiser is enabled in this case (via quadratic programming) and therefore an improvement of 52.86% and 25.06% in control action is achieved relative to the *PID* and *NGMV* respectively.

6.4.4 Constraint Handling – Reference Tracking

Scenario 8a: Step of-20% of the rated power:

For this scenario wind speed is kept constant at above rated in order to provide rated power availability (5MW). A negative step change is then introduced to the power reference of the turbine (derating). The graphs were focused at the point where a negative step change of 20% of the rated power occurs for a clearer examination of tracking provided by the three controllers. Here as well, only the *SISO* control structure is considered. To explore performance of the *LPV-NPGMV* in the presence of constraints the pitch angle actuator range was limited between 0-15deg.

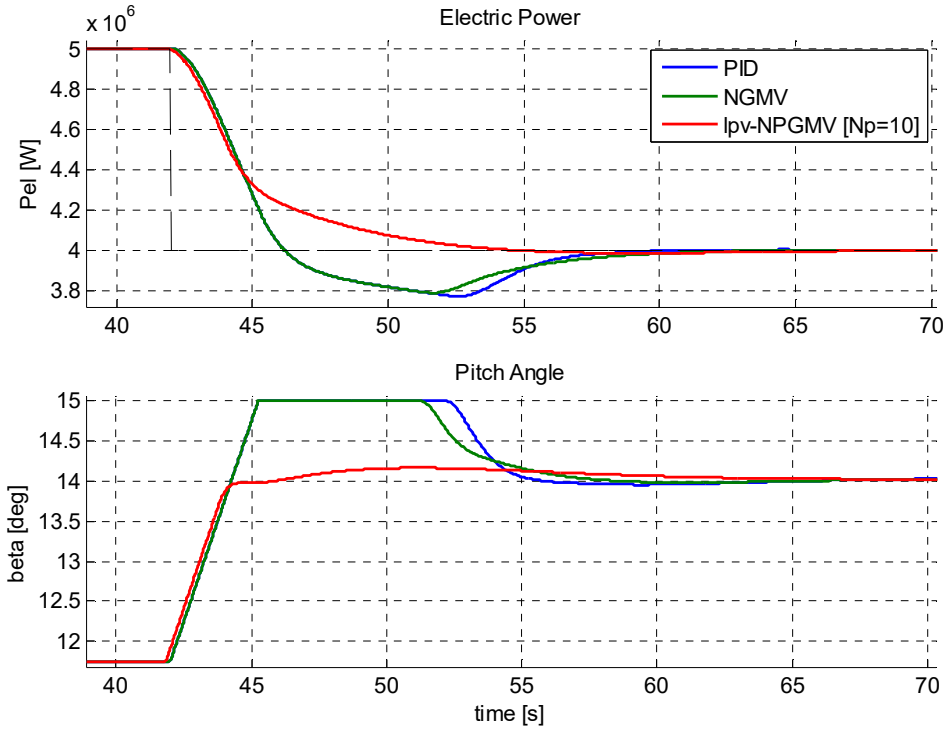


Figure 6:32 Scenario 8a results for the *SISO* control structure.

Table 6:16: Scenario 8a, SISO quantified controller comparison

Controllers	$\boldsymbol{\beta_{STD}}$ **(norm.)**	$\boldsymbol{Pel_{MISE}}$ **(norm.)**
PID	1	1
NGMV	0.9780	0.9891
LPV-NPGMV [*Np*=10]	0.8820	0.8751

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 12.49% in *MISE* relative to the baseline *PID* and 11.4% relative to the baseline *NGMV*. The

absolute constraint optimiser is enabled in this case (via quadratic programming) and therefore an improvement of 11.8% and 9.6% in control action is achieved relative to the *PID* and *NGMV* respectively.

Scenario 8b: Sequence of steps in power between 0-2MW:

Absolute constraints on pitch at 12-17 deg

For this scenario wind speed is kept constant at above rated in order to provide rated power availability (5MW). A series of steps in power reference (between 0-2MW) are then introduced. Figure 6:33 captures the results on reference tracking for the *SISO* control structure across a specific time window for clarity. To explore performance of the *LPV-NPGMV* in the presence of constraints the pitch angle actuator range was limited between 12-17deg.

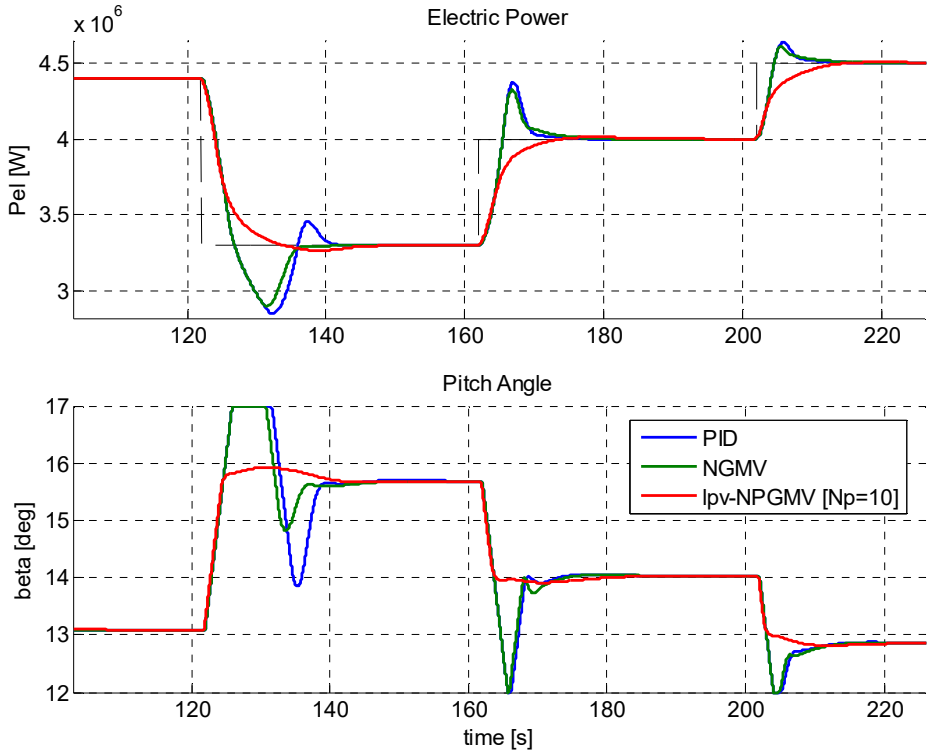


Figure 6:33 Scenario 8b results for the *SISO* control structure.

Table 6:17: Scenario 8b, SISO quantified controller comparison

Controllers	β_{STD} (norm.)	Pel_{MISE} (norm.)
PID	1	1
NGMV	0.9972	0.9113
LPV-NPGMV [*Np*=10]	0.9662	0.5397

The *LPV-NPGMV* [*Np*=10] provides the highest power regulation benefit with an improvement of 46% in *MISE* relative to the baseline *PID* and 37.16% relative to the baseline *NGMV*. The absolute constraint optimiser is enabled in this case (via Quadratic Programming) and therefore an improvement of 3.38% and 3.1% in control action is achieved relative to the *PID* and *NGMV* respectively.

6.5 Final Remarks

In this section the *LPV-NPGMV* nonlinear control design technique was employed to the power regulation and tracking of an experimental (validated) large scale wind turbine system. The turbine model was reformulated in *LPV* form fully parameterised by wind speed through the aerodynamic conversion relationships. Four different types of wind disturbances were used in simulation (step, gust, stochastic variation) to explore performance of the controller in regulating power production at the rated value. Moreover a few scenarios where power tracking was explored (turbine de-rating) were included. A separate section was added to explore performance while using a Quadratic Programing constraint solver in absolute pitch angle limits.

In all scenarios the *LPV-NPGMV* demonstrated substantial improvement in power regulation. In the step disturbance case best results were achieved at small steps peaking at 73.5% improvement over the baseline *PID*. It is worth mentioning here that, whenever it was successfully applied, the *MIMO* configuration provided significant reduction in generator torque variations (e.g. 84.4% for the small step disturbance case), in contrast to most of the *SISO* experiments being negligible. In the gust disturbance case best results were achieved by using smaller prediction horizons peaking at 80% improvement over the baseline *PID* and 12.17% improvement in generator torque for the *MIMO* configuration. The stochastic varying wind case was the only exception where there was no reduction in generator torque for the *MIMO* configuration. Best results in this case were achieved for longer prediction horizons using the *SISO* setup with improvement peaking at 62.2% for power regulation.

In the de-rating scenarios, access to future setpoint information was proven advantageous when using longer prediction horizons with improvement peaking at 92.4% over the baseline *PID*. The application of constraint handling in the controller provided significant restriction in control action (pitch angle) in the expense however of power regulation and tracking performance. Best results were achieved for large step disturbances with improvement peaking at 65% over the baseline *PID* with reduction in pitch angle variability at 40%. Similarly for tracking, performance peaked at 24% over the baseline *PID* with 72% reduction in pitch angle variability. The *MIMO* configuration yielded very similar results to the *SISO* in the constrained case.

A counterintuitive effect is observed in most results and this was the relative improvement in control action while at the same time tighter tracking was achieved. This was the result of this particular *Feedback-Feedforward* configuration that results in an instant reaction on one

channel (*SISO*) or both channels (*MIMO*) during a speed variation driven by the equivalent pitch and/or generator torque lookup tables. This yielded a *Feedforward* term that had a greater impact on performance with the *Feedback* contribution being significantly smaller and only accountable for smaller variations around the operating point. This effect however needs to be further investigated in future work.

Chapter 7 Sightline Stabilisation of Electro-Optical Devices

Based on the same structure as the application that was described in the previous application chapter, the state-space *NGMV* algorithm is now employed to provide stabilisation of an *Electro-Optical* (*EO*) gyroscopic turret used in surveillance applications.

Asymmetric warfare, a large conventional army against much smaller terrorist or 'insurgent' groups has been the defining characteristic of recent military conflicts involving UK and US armed forces. Without clear battlefield lines the identification of and defence against hostile threats has seen a rapid increase in the demand for effective and reliable airborne *electro-optic* (*EO*) systems. *EO* systems fall into two broad classes:

- *Intelligence Surveillance and Reconnaissance* (*ISR*) systems.
- *Defensive Aid Suite* (*DAS*) systems.

Both use similar optical systems technology, but the very real operational differences demand substantially different solutions. For example, an *ISR* system must have long-range imaging capability, which in turn requires high-resolution imagers and extremely precise sightline stabilization systems to minimize jitter. A *DAS* system however needs an incredibly agile sightline to ensure that incoming threats to the host platform are properly tracked and interrogated/defeated. This is especially true for *Directed Infra-Red Countermeasures* (*DIRCM*) systems. This poses a challenging tracking control problem as a singularity arises when the target moves along a certain trajectory essentially resulting into a discontinuity nonlinearity known as "*gimbal lock*". The *NGMV* "*black-box*" model internal model structure was used and shown significant improvement in maintaining engagement to the target along that part of the trajectory. This work is divided into the following sections.

Section 1 – Problem Description; overview of the control objectives and strategy employed in this work.

Section 2 – System Model Description; derivation of a suitable state-space model used within the state-space NGMV design and in simulation.

Section 3 – Control System Description; adaptation of the control algorithm as derived in Equation 2:32 and also of a basic *PID* control formulation used here for comparison.

Section 4 – Simulation Results & Conclusions; definition of the different scenarios and presentation of control performance results for the different schemes employed throughout the simulations.

7.1 Problem Description

During engagement, when the host platform is attacked by a surface-to-air missile, a fast and precise sightline loop is essential to intercept the missile system via jamming of the missile seeker. Moreover, it is imperative for the tracking system to be able to operate over a full hyper-hemispherical field-of-regard (*FoR*) (as shown in Figure 7:1), as any tracking deficiencies are sure to be exploited by future missile guidance systems and tactics (Anderson et al., 2009). Sufficient *FoR* is ensured by a 2-axis gimbal device with one gimbal rotating over the azimuth axis and the other over the elevation.

7.1.1 The Nadir Problem

A significant problem exist with this configuration however, when the target moves such that the line-of-sight (*LoS*) vector approaches the azimuth axis, at around -90 degrees in elevation, the system loses one degree of freedom and is therefore unable to maintain accurate track.

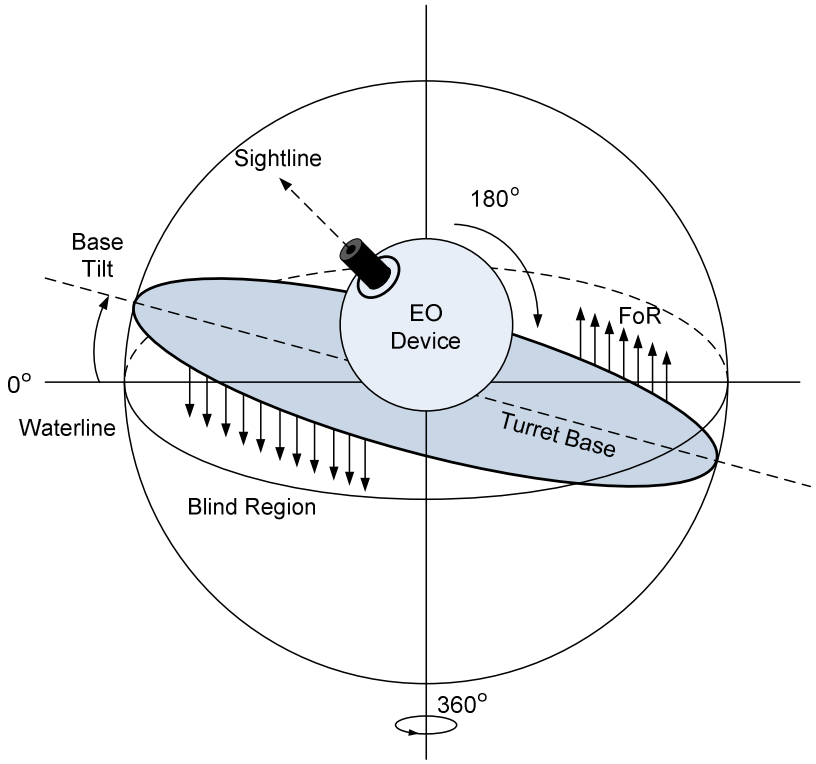


Figure 7:1 Hemispherical coverage of the EO tracking device mounted on a platform.

Often this degree-of-freedom loss is known as one of the following interchangeable terms "*gimbal lock*", "*keyhole singularity*" or "*Nadir cone*", although we shall use the latter for the remainder of this chapter. Practically, the engagement kinematics driving tracking in the

neighbourhood of the nadir require significant agility from the outer, azimuth gimbal axis, to the limit that *LoS* vector tracking through the singularity when the *LoS* and azimuth axis are collinear results in infinite acceleration & rate demands to the *OG* axis. Obviously, the acceleration demands within this range are high enough to saturate the servos, leading to large tracking errors that heavily impair the precision of the system (Rue, 1969).

Various techniques have been used in the past to overcome the *Nadir* problem. The simplest is the orientation of the axes of the turret such that the nadir singularity lies outside the operational region of the system. However, this approach is unsuitable for a *DIRCM* system due to the need for hyper-hemispherical coverage. The second method, also an opto-mechanical solution, is to add additional axes to the pointing and stabilisation system. Unfortunately this approach increases the size, complexity and initial unit price in addition to comparatively reducing mean-time-between-failure and hence raising operational costs. Transition from military to civilian protection *DIRCM* systems using multi-axis technology is therefore not practical. The third approach is to reduce the size of the nadir cone by using a faster actuation mechanism, requiring high-performance motors/sensors etc. with the inevitable financial penalties. The fourth approach and the one investigated in this work is the use of non-conventional sightline controllers to mitigate the effect of nadir singularity on tracking error.

Operation in the neighbourhood of the nadir presents at least two significant nonlinearities to the sightline control engineer. The first is the singularity introduced earlier. The second is a kinematic nonlinearity due to the difference between the cross-elevation axis (as measured in the imager frame) and the outer gimbal (azimuth) axis. Having two significant nonlinearities in the feedback loop would naturally suggest that a nonlinear control design technique should be applied to the problem. Luckily, a number of nonlinear control laws have been proposed in recent years, including *Sliding-Mode* control, *Adaptive Backstepping*, *LMI* optimization, nonlinear *H-infinity* control, feedback linearization and direct *Lyapunov* methods (Utkin et al, 2009), (White et al., 2009), (Lechevin and Rabbath, 2007) and (Henrion et al., 2003). However, most of the nonlinear controller design techniques mentioned are difficult to use, steeped in complex mathematics and often counter-intuitive tuning. As shown in Chapter 2 the *NGMV* provides a framework that attempts to isolate the nonlinearities in the system from the sightline control designer. Once the nonlinearities in the system captured in the modelling process are included into the *NGMV* synthesis, controller tuning is achieved via simple weight selection. The *NGMV* was designed specifically to yield a nonlinear controller which can be easily designed and commissioned. Therefore, it is an ideal candidate technique for sightline control of a two-axis system.

In the next section an overview of the Glasgow University sightline control laboratory is presented. The latter was the basis upon which the derivation and validation of the turret device and the external tracking loop was carried out.

7.2 System Description

7.2.1 Experimental Configuration

All experimental work carried out at the Sightline Control Laboratory (*SCL*) of the University of Glasgow which is a bespoke research and teaching facility designed to assist applied research in the areas of pointing, stabilization, tracking and image processing of electro-optic systems, known collectively by the term *Sightline Control*. The remit of the lab is to provide a *Hardware-in-the-Loop* (*HIL*) functional testing and analysis facility for all aspects of sightline control, irrespective of the particular unit under test.

In the configuration used for this research, an *Aeromech Tiger Eye* 2-axis visible band electro-optic turret was used, (see Figure 7:2 below). The turret is mounted on a tripod about 1.50m height and 2.50m distance from the projector screen. In general, the High-Definition (*HD)* projector is used to display high-definition video of interesting operational vignettes designed to push the tracking loop to its performance limits, in both track creation/association and kinematic prediction. In this set of experiments only the track-loop dynamic response is of interest. Consequently, the projected target is a simple white ball against a blue background, programmed to move along either a circular or vertical trajectory. Each trajectory is specified in a world coordinate system which, for the circular trajectory consists of the y and z axis of the ball defined as sinusoidal functions, the parameters of which define the circle radius and revolution time. For the experiments that follow, the target motion frequency was set at 0.1Hz with a projected image sample rate of 100Hz.

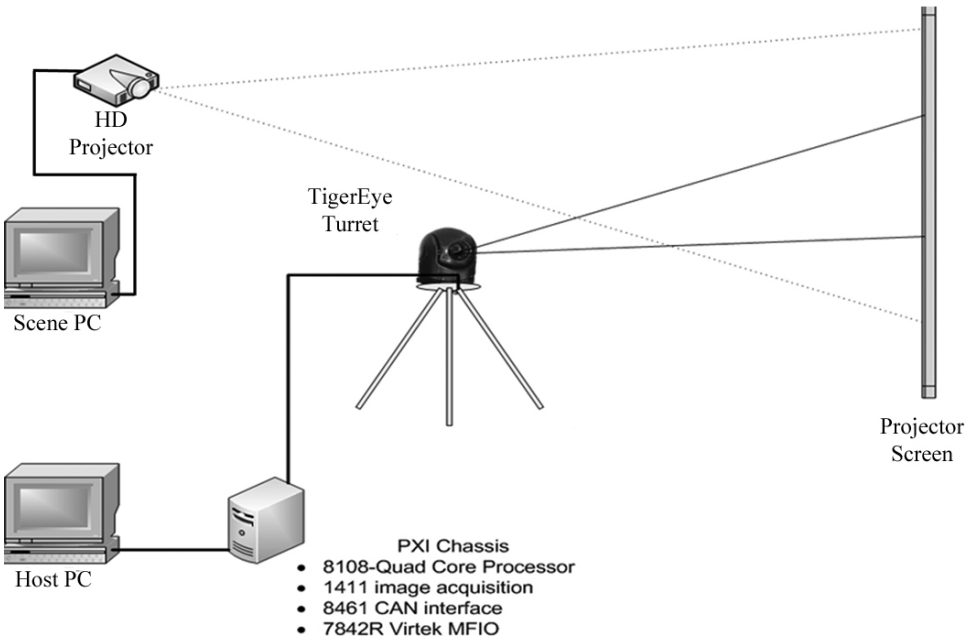


Figure 7:2 Schematic overview of the sightline control laboratory.

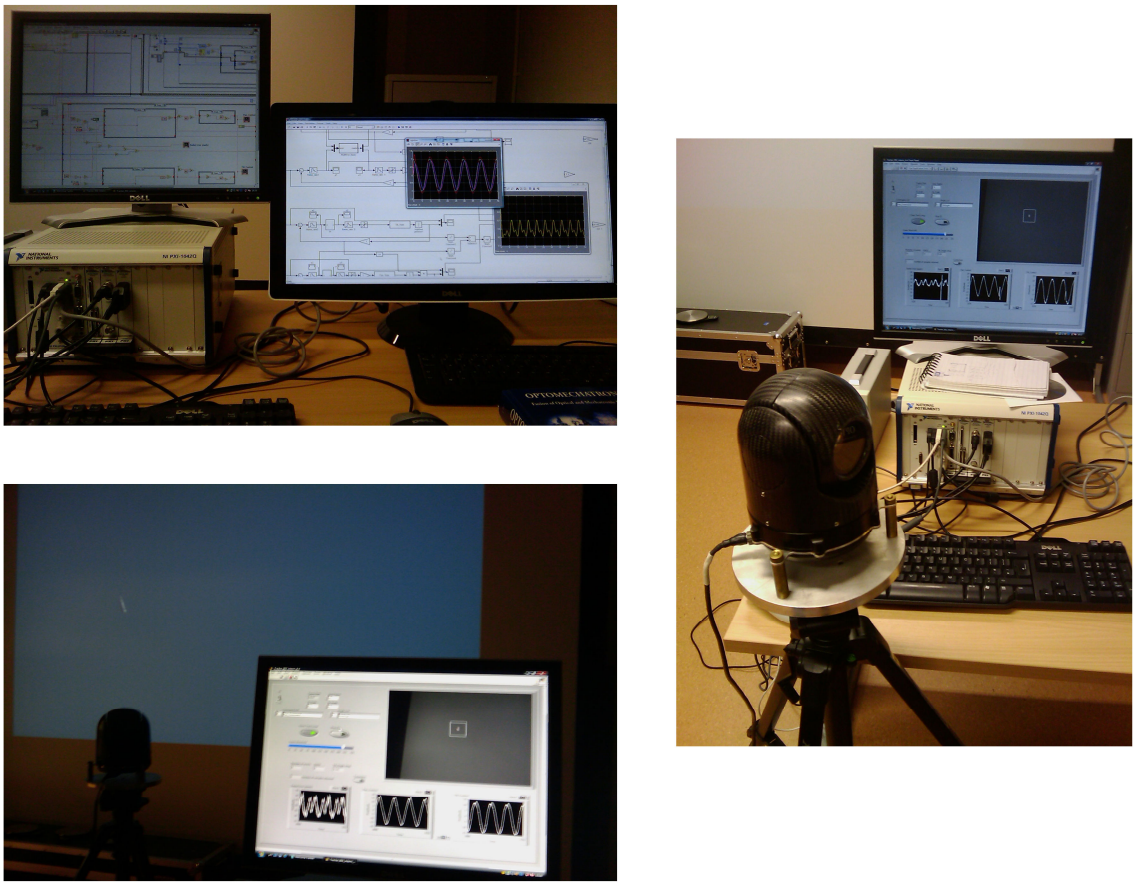


Figure 7:3 Experimentation equipment.

Communication with the *TigerEye* is performed over a high-speed *CAN* bus and an *RS232* video link. These are interfaced to the host *PC* using a *National Instruments PXI* chassis with video frame grabber and *CAN* bus interface cards. The *TigerEye* operates in a number of different command modes – position mode, rate mode, stabilized rate mode, body-referenced etc. To implement an external track loop, the *TigerEye* should be commanded to operate in rate mode, which means that commands sent to the turret over the *CAN* bus will be interpreted as rate loop demands for each axis. The following figure demonstrates a summarizing schematic of the system.

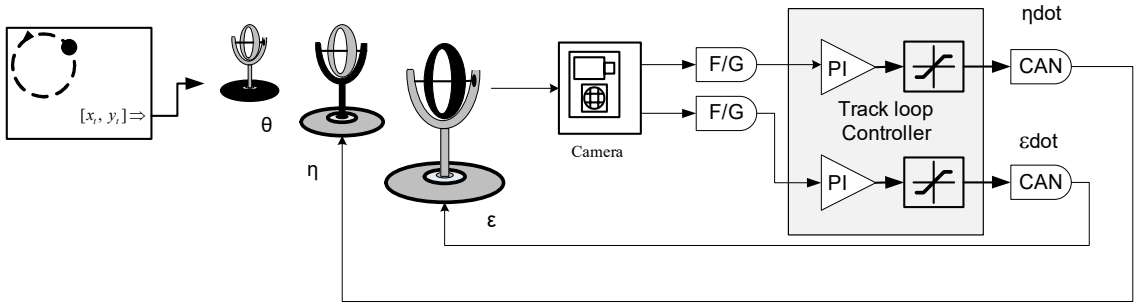


Figure 7:4 Tracking system physical configuration.

The measurement for the elevation displacement is taken directly as the vertical pixel displacement, given the camera is fixed to inner gimbal. This measurement then passes through a scaling factor to convert from pixel error to angle error, inside the processing unit where the comparison is taking place, and converted into rate demand. Likewise, the measurement for the azimuth displacement is taken in terms of horizontal pixel displacement, which is a measurement also obtained from the camera fixed onto the inner gimbal. The second error measurement does not represent the actual azimuth error but the azimuth with respect to the inner gimbal defined as the '*cross-elevation*' (Kennedy, 2003). Using the cross-elevation to control the azimuth axis gimbal can cause problems, as follows.

7.2.2 Simulation Model Definition

A more insightful aspect on the overall tracking system and what was actually used in simulation is shown in Figure 7:5. The various transformations required to condition the target vector from world to azimuth/elevation coordinate frames becomes clearer in this representation. $R_{w/b}$, $R_{b/og}$ and $R_{og/ig}$ are the world to base, base to outer gimbal and outer gimbal to inner gimbal transformation matrices respectively. The *CAN* bus delays and individual rate loops transfer functions were derived via identification. As mentioned earlier each channel comprises of an external position and an internal rate loop. For the duration of the experiments the base declination angle remained constant and it would only be adjusted prior to experiment initialization. Although, in practice base motion will occur i.e. when mounted on an airborne host platform, in these particular experiments we keep the base static in order to isolate and investigate more efficiently the *NADIR* problem.

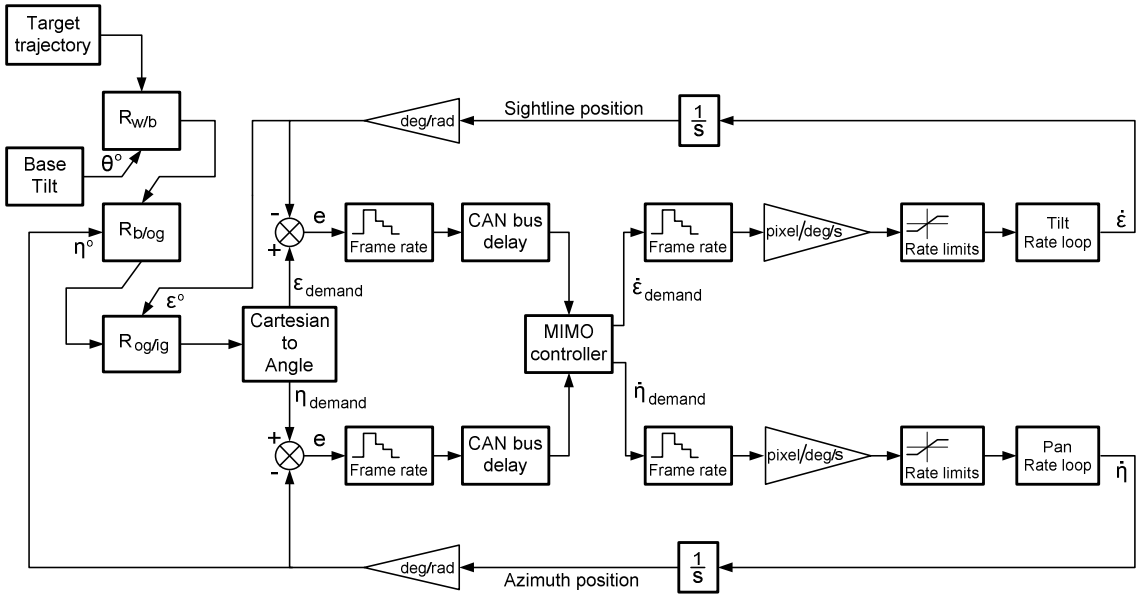


Figure 7:5 Tracking system simulation configuration.

Turret Kinematics Definition:

The following analysis explains how a change in target position relates to demanded gimbal angles for accurate tracking. Say that the line-of-sight rests on the waterline (zero azimuth and elevation angles) prior to tracking a displacement in the target axes. As shown in Figures 7:6 and 7:7, there are three principle rotations to consider; base tilt ϑ, which transforms from world axes to base axes, azimuth (outer gimbal) rotation about the *TigerEye z*-axis (η) and then an elevation (inner gimbal) rotation about the *y*-axis, ε.

As the camera is carried with the inner gimbal, the image and inner gimbal axes are assumed coincident (neglecting alignment tolerances). Therefore the target position in image axes is given by,

$$x_{los\_im} = (T_\varepsilon T_\eta T_\theta) x_{los_{geo}} \tag{7:1}$$

$$T_\varepsilon(\varepsilon) = \begin{bmatrix} \cos(\varepsilon) & 0 & -\sin(\varepsilon) \\ 0 & 1 & 0 \\ \sin(\varepsilon) & 0 & \cos(\varepsilon) \end{bmatrix} \tag{7:2}$$

$$T_\eta(\eta) = \begin{bmatrix} \cos(\eta) & \sin(\eta) & 0 \\ -\sin(\eta) & \cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7:3}$$

$$T_\varepsilon(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{7:4}$$

$$x_{los\_im} = \begin{bmatrix} \cos(\varepsilon)\cos(\eta)\cos(\theta) - \sin(\varepsilon)\sin(\theta) & \cos(\varepsilon)\sin(\eta) & -\cos(\theta)\sin(\varepsilon) - \cos(\varepsilon)\cos \\ \cos(\theta)\sin(\eta) & \cos(\eta) & \sin(\eta)sin(\theta) \\ \cos(\varepsilon)\sin(\eta) + \cos(\eta)\cos(\theta)\sin(\varepsilon) & \sin(\varepsilon)sin(\eta) & \cos(\varepsilon)\cos(\theta) - \cos(\eta)sin \end{bmatrix} \tag{7:5}$$

Where T_ε, T_η, T_θ are the direction cosine transformation matrices for the elevation, azimuth and base tilt angles respectively. To recover the off-boresight angle errors needed to drive each track loop controller, the actual target position is projected onto the surface of a unit sphere centred at the image plane. A simple conversion from *Cartesian* to polar coordinates recovers the actual error angles as measured by the *TigerEye* imager.

In a more analytic way the above can be summarised as the solution to the following problem; suppose we express the *LoS* in the geographical frame when aligned with the initial target position given in geometrical terms as follows,

$$x_{los_{geo}}(\varepsilon, \eta, \theta)_0 \rightarrow x_{target}(x, y, z)_0 \quad (7:6)$$

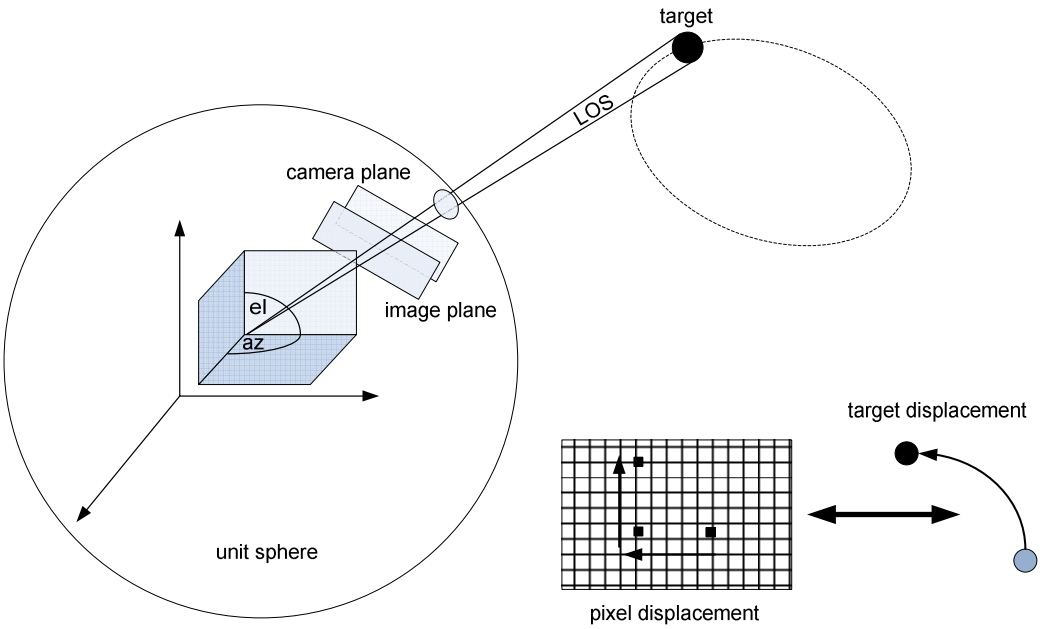


Figure 7:6 Target displacement from global to LoS frame graphical representation.

If the target moves to a different position $x_{target}(x, y, z)_1$ then given the angles *x*, *y*, *z*, ϑ compute the elevation ε and azimuth ϑ angles required by the gimbal to realign its *LoS* with the new position. The main problem this research deals with is an implication occurring in the kinematics, when the turret is tracking a target which moves near or past the -90 degrees elevation angle. Then the sightline axis becomes aligned with the outer gimbal rotation axis and the system loses one degree of freedom (Figure 7:7).

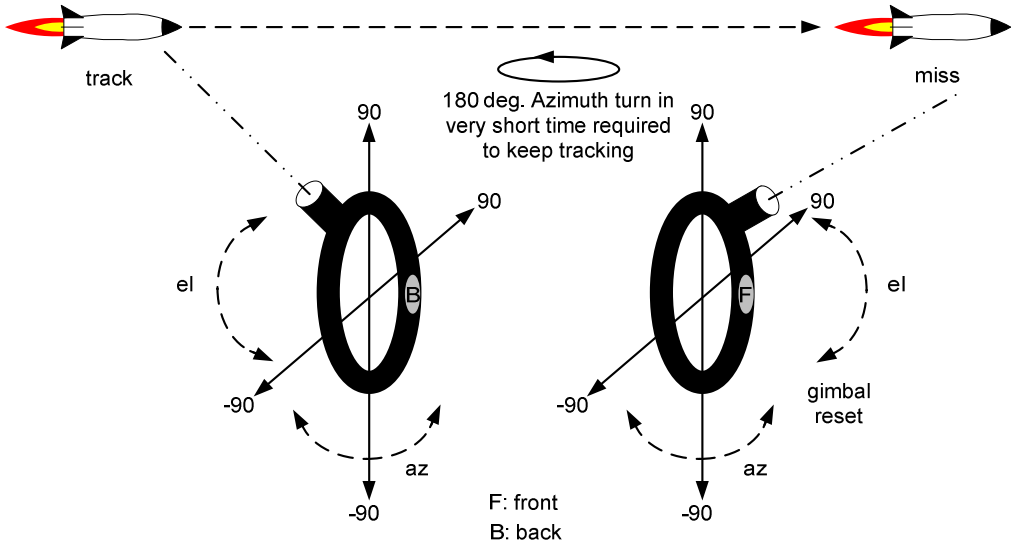


Figure 7:7 Tracking loss through the Nadir.

Operation of the *LoS* near this condition, inside the nadir cone, introduces two significant problems. The first is the mechanical singularity arising from the loss of a degree of freedom at the nadir. Physically this manifests as the inability of the outer gimbal (azimuth) to perform the necessary instantaneous 180 deg. due to drive/motor saturation and tracking is lost. There is little can be done here without preview or predictive '*shooting*' methods (Anderson and Loo, 2009). A second *Nadir* nonlinearity arises from the use of *cross-elevation* error as the measurement driving the azimuth motor. In the transformation required to take the *cross-elevation* back to actual azimuth angle, shown in Equation 7:7, the cross-elevation error δu is multiplied by the cosine of the elevation gimbal angle.

$$\delta\eta_{err} = G_{rad/pix}\begin{bmatrix}\cos(\varepsilon) & 0 & \sin(\varepsilon)\\ 0 & 1 & 0\\ -\sin(\varepsilon) & 0 & \cos(\varepsilon)\end{bmatrix}\begin{bmatrix}0\\0\\\delta u\end{bmatrix} = \begin{bmatrix}0\\0\\\cos(\varepsilon)\delta u\end{bmatrix} \tag{7:7}$$

As the angle ε increases, there is effectively a reduction in the azimuth track loop gain equivalent to $cos(\varepsilon)$. At the nadir, the *cross-elevation* and azimuth axes are orthogonal leading to no effective control in the azimuth track loop. This is the nadir problem this paper will use *NGMV* to mitigate. The following example is set according to the *SCL* configuration and illustrates the previous issue.

Example:

Assume that initially the target rests 2.5 m away from the gimbal in coordinates *x*_target $(x,y,z)\rightarrow(2.5,1,1)$ and that the base is resting at the waterline (0^o base declination). Also assume that the target is moving along a circular trajectory on a 2-dimensional plane perpendicular to the *LoS* of the gimbal. If the target is moving at an angular velocity of

ω=2×π×0.1 then the individual *y* and *z* trajectories are given by the following equations of motion.

$$y_{target} = \sin(\omega t)\,, z_{target} = \cos(\omega t) \tag{7:8}$$

The simplicity of this configuration allows us to proceed in the analytic way of computing the *Inverse Kinematics* (*IK*) problem and obtain the elevation and azimuth angle requirements to track the target. Since the two *DOF* are clearly decoupled we can conveniently inverse the forward kinematics and therefore calculate the angle demands out of the following relations.

$$\varepsilon = \tan^{-1}\left(\frac{\mathrm{y_{target}}}{\mathrm{z_{target}}}\right) \tag{7:9}$$

$$\eta = -\tan^{-1}\left(\frac{\mathrm{x_{target}}}{\sqrt{\mathrm{y_{target}^2} + \mathrm{z_{target}^2}}}\right) \tag{7:10}$$

Figures 7:8-7:11 demonstrates the kinematic responses of the two axes given the previous parameters simulated in Simulink for different base declination angles.

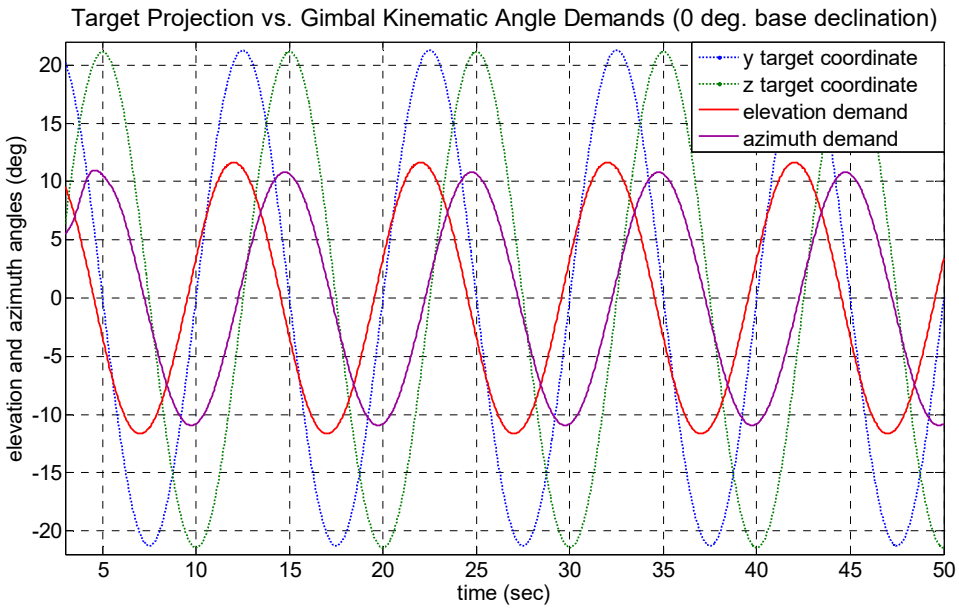


Figure 7:8 Kinematics analytical model validation for 0° base declination.

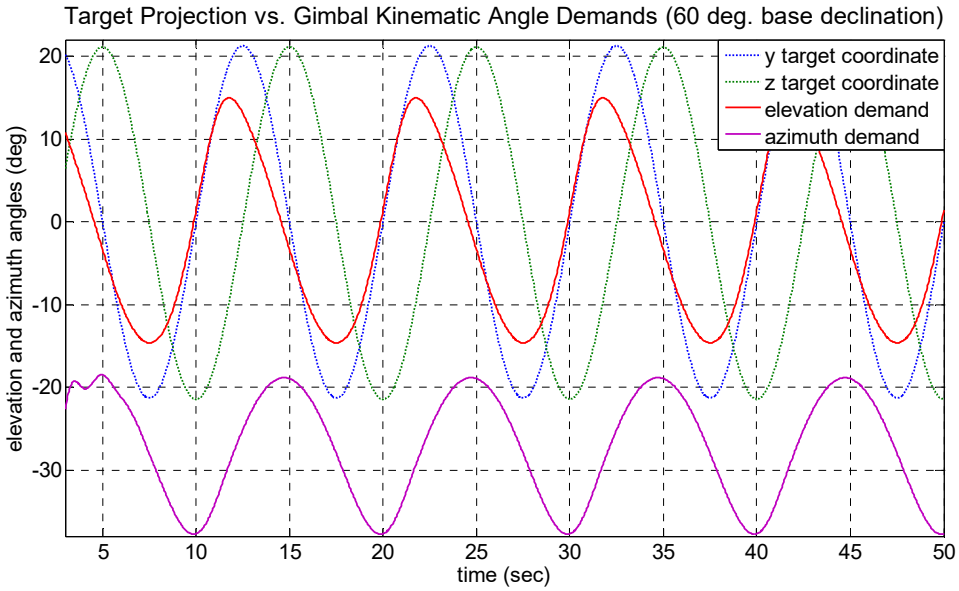


Figure 7:9 Kinematics analytical model validation for 60° base declination.

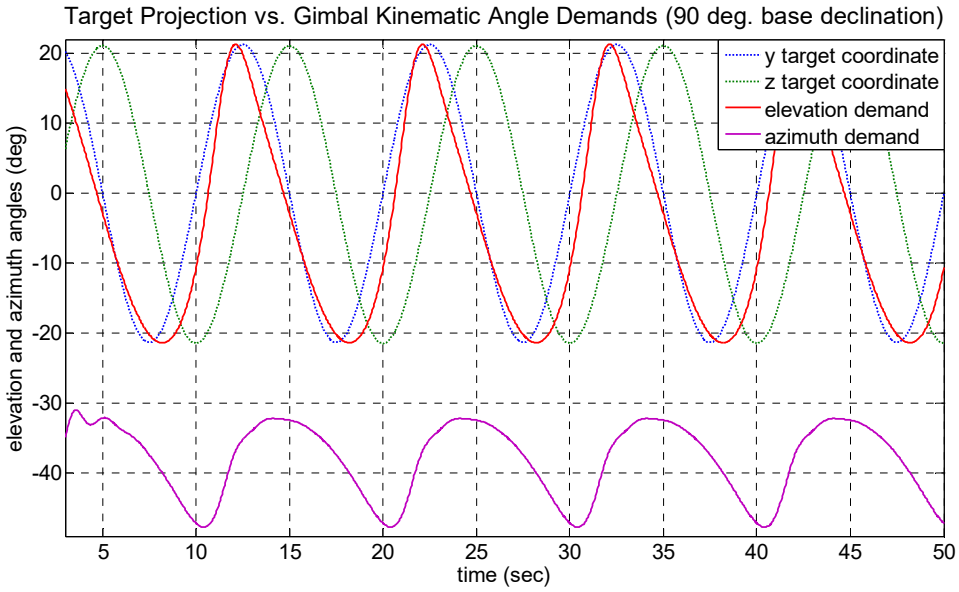


Figure 7:10 Kinematics analytical model validation for 90° base declination.

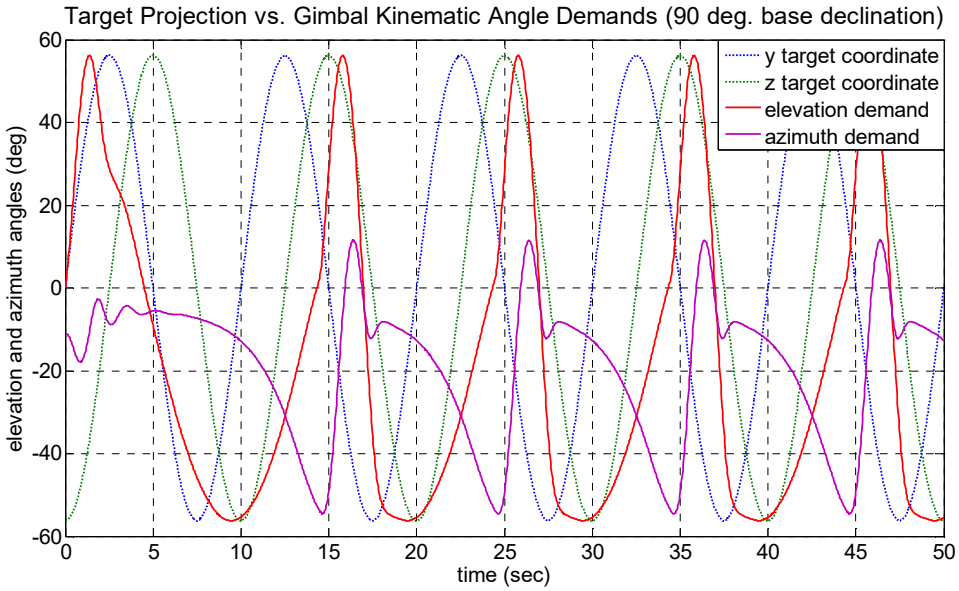


Figure 7:11 Kinematics analytical model validation for 90° base declination and 10cm target plane distance from turret.

Turret Dynamics Definition:

By first principle modelling, the turret system simplicity consists of two servo-motors, one for each gimbal. These rate loops are considered decoupled in this particular application, having no interacting reaction torque effects from one to the other. Consequently the two paths are individually linear and can be easily described by Newton 2nd rotational law of motion as,

$$\bar{J}_I\hat{\dot{\omega}}_I(t) + \left(\hat{\omega}_I(t) \times \bar{J}_I\hat{\omega}_I(t)\right) = \hat{L}_I(t), \tag{7:11}$$

$$\hat{L}_I(t) = [T_{IC}, T_{Iro}] - [T_{IU}] - [T_{If\omega(t)}] \tag{7:12}$$

and

$$\bar{J}_O\hat{\dot{\omega}}_O(t) + \left(\hat{\omega}_O(t) \times \bar{J}_O\hat{\omega}_O(t)\right) + \left(\hat{L}_I(t)\right)_O = \hat{L}_O(t), \tag{7:13}$$

$$\hat{L}_O(t) = [T_{OC}, T_{Oro}] - [T_{OU}] - [T_{Of\omega(t)}] \tag{7:14}$$

where, *J* is the inertial matrix and L the sum of the kinematic torques around the equivalent gimbal which, apart from the control torque T_C, include friction and cable restrain torques $T_{f\omega(t)}$, reaction torques T_{ro} acting on the outer gimbal by the inner, and mass imbalance torques T_U about each gimbal (Kennedy, 2003). The subscripts *I* and *O* refer to the inner and outer gimbal axis respectively.

The Cosecant Correction Definition:

One of the methods used to encounter the *Nadir* tracking consists of directly adjusting the *cross-elevation* error signal with the use of a secant function to cancel out the cosine in the denominator, Figure 7:12. Moreover, alternative transformations, like the quaternion, cannot be employed as the problem itself is a result of the turret's physical attributes rather than just an implication of the modelling kinematics.

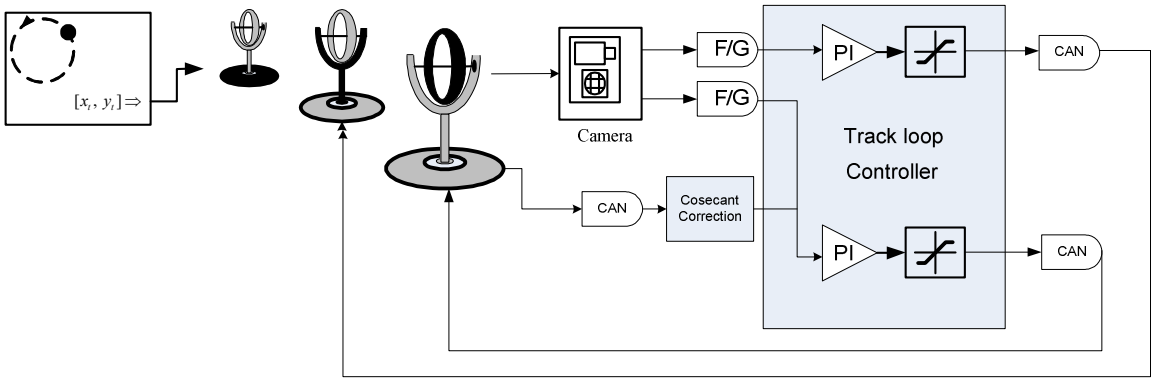


Figure 7:12 Signal flow for the cosecant correction controller.

To implement the cosecant correction controller, the actual inner gimbal angle is obtained from the telemetry message the *TigerEye* periodically transmits over the *CAN* bus. In the results to follow there are some slight synchronization errors between *CAN* bus reporting and the imager frame rate, but not sufficient to disrupt the action of the controller. Finally, the cosecant correction controller should have a roll-off close to the actual *Nadir*, otherwise infinite gain would be injected into the azimuth axis track loop.

7.2.3 Model Identification

To test the performance of the *NGMV* controller, an accurate model of the *TigerEye* is required prior to creating and commissioning a *LabVIEW* version. A frequency domain identification strategy was used to extract the transfer functions of each axis. Each transfer function should relate the rate command to the actual gimbal axis rate, as reported by the stabilization loop gyros over the *CAN* bus. The nominal *CAN* bus trajectory reporting frequency is 10Hz and the imager frame rate is 30Hz. Logarithmically-spaced input sinusoids of frequencies from 0.05 to 4 Hz excited each axis in turn and, using *SCL LabVIEW* code, the system frequency response function for each input frequency were computed. This data was converted into gain and phase plots in *Matlab*, from which accurate transfer functions were obtained. The spectral characteristics for the tilt and pan rate loops are compared against the identified ones in Figure 7:13 upper and lower plot respectively.

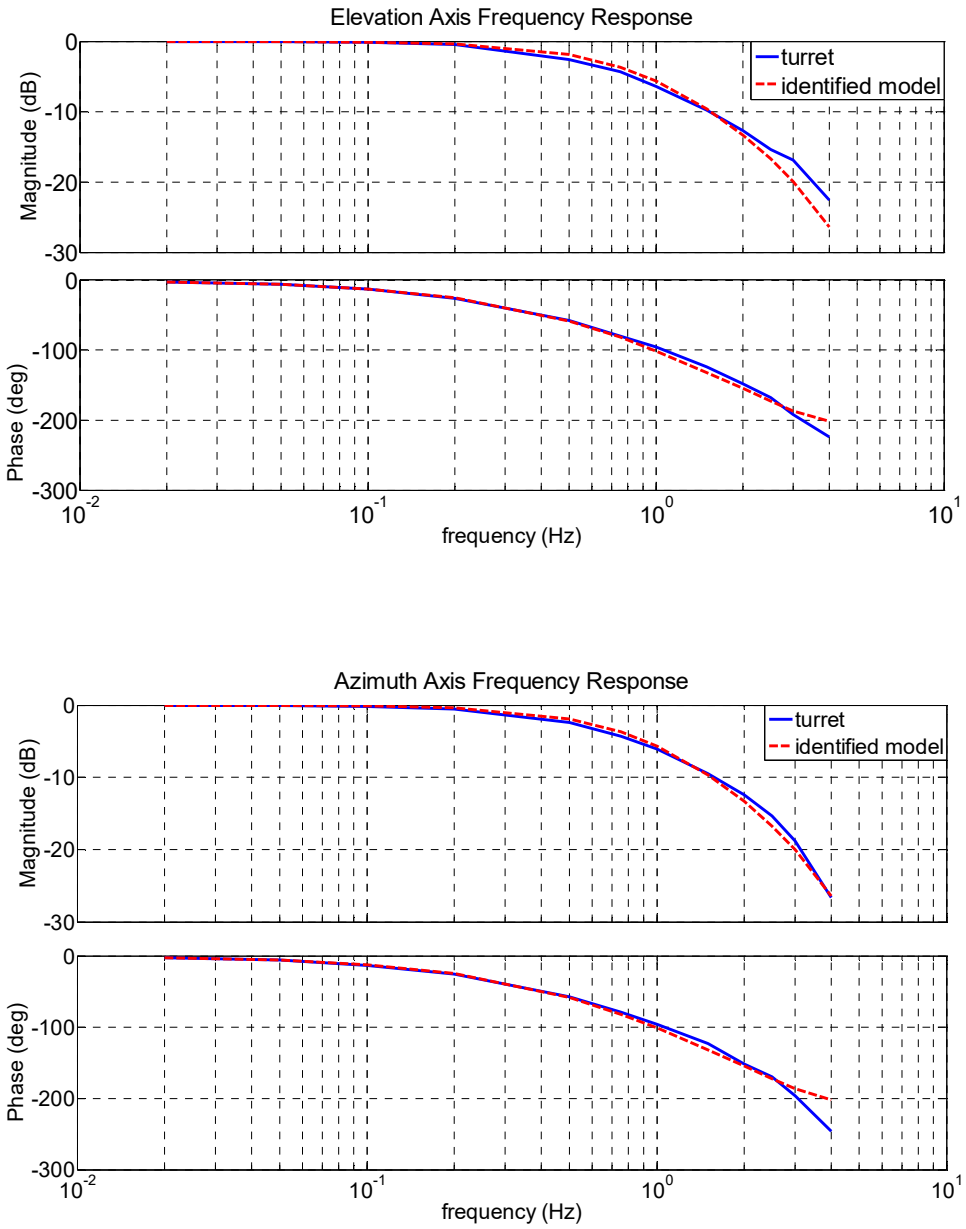


Figure 7:13 Turret elevation (upper plot) and azimuth (lower plot) axis frequency fit.

Model Validation:

To use a mathematical model for control design in the neighbourhood of the nadir, the accuracy of the model in this region has to be determined. The moving target and full gimbal kinematics of the problem were implemented along with the identified turret model in *Matlab*. The system was simulated for the case where the turret tracks a ball moving along a circular trajectory in 0.1 Hz and under various base tilt angles from 0 (waterline) to 90 deg. (*Nadir*). Given the same scenarios, the real turret was tested and the rate commands along with the spectral error compared to those of the simulation (Figures 7:14 and 7:15). On the

pan graph corresponding to -90 degrees base tilt angle the saturation in the azimuth gimbal motor can be seen after which the camera captures the target again but only in the case where the target moves relatively slowly.

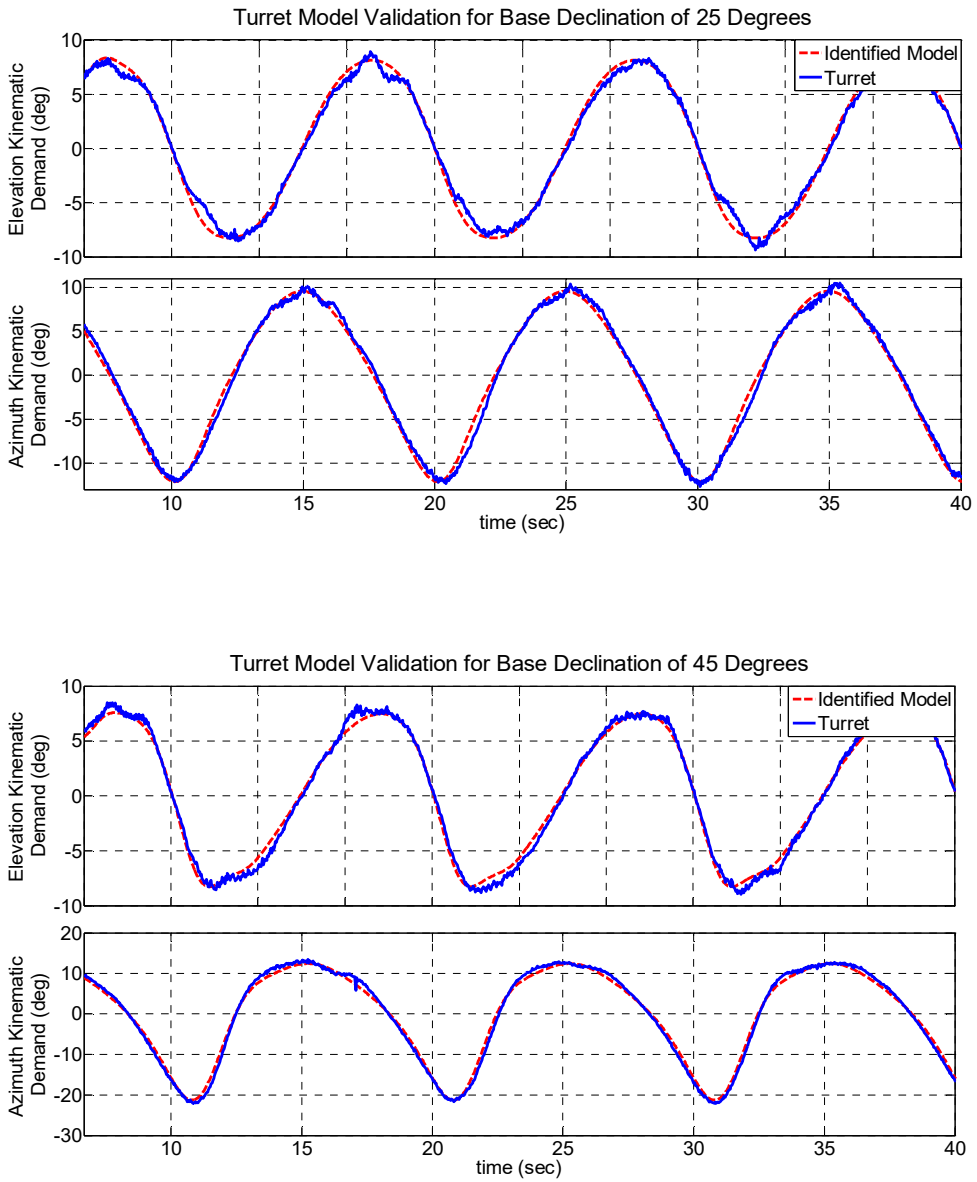


Figure 7:14 Turret identified model validation for 25° (upper plot) 45° (lower plot) base declination.

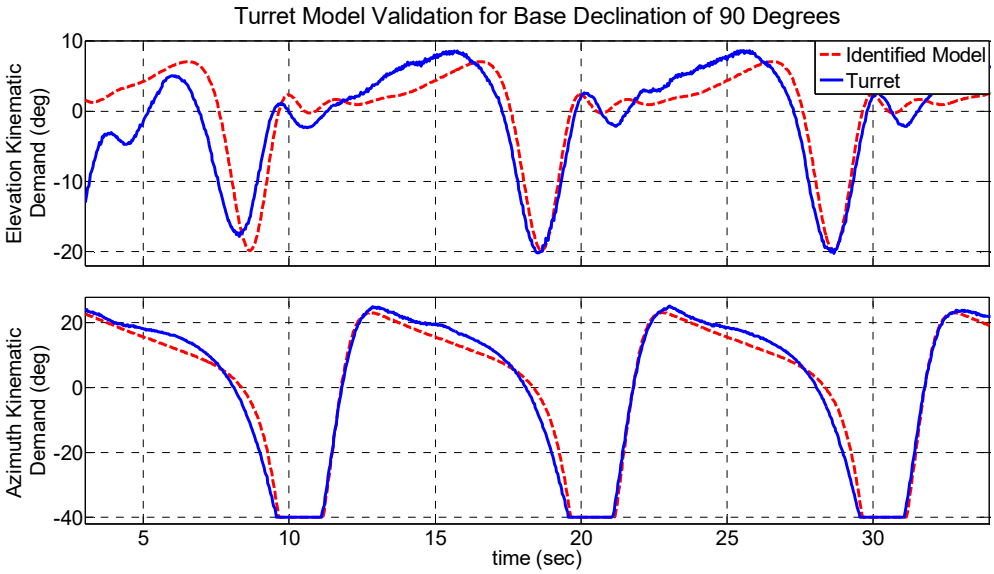


Figure 7:15 Turret identified model validation for 90° base declination.

It is seen from the validation results that regardless of the mismatch occurring due to phase differences and potential jitter (varying sample rate) on the *CAN* bus, the identified data holds a model representation accurate enough to stand as the simulation basis.

7.3 Control System Description

Although the individual rate loops are not nonlinear, the total system demonstrates the inverse cosine nonlinearity which appears as a singularity on the reference signal. That makes the situation peculiarly difficult to be dealt with a conventional nonlinear control scheme. In this section the *NGMV* is formulation as described in Section 2.2.1 is used. This is a very fast control system and a sampling time of 0.0333s is used within the controller.

Nonlinear Subsystem (*black-box* operator):

Here the nonlinear term encapsulates all nonlinearities within the internal plant in a *"black box"* manner. These only comprise the inner and outer axis rate constraints as seen below for both.

$$-40deg/s \leq \dot{\omega} \leq 40deg/s$$

Linear Plant Subsystem:

Here the linear subsystem was used to encapsulate the two gimbal axis pan and tilt linear dynamics as derived in the identification section and discretised as appropriate.

$$W_{0k} \rightarrow \frac{0.02357 + 0.01995z^{-1}}{1 - 1.563z^{-1} - 0.6065z^{-2}} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Reference Subsystems:

For this *NGMV* formulation the reference subsystem is defined as a *2x2* unity operator for the two gimbal axis respectively.

$$W_{0,r} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Disturbance Subsystem:

The disturbance subsystem was modelled as a *2x2* discrete *TF* matrix.

$$W_d \rightarrow \frac{-0.01 + 1.995z^{-1}}{0.7z^{-1} - 0.33z^{-2}} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

NGMV Weighting Definition:

The error weighting P_c was selected using the *PID* gains selection criteria for *PI* proportional and integral gains of $K_p=1$ and $K_i=1$ respectively. This yields the following error weighting equation for both control channels.

$$P_c \rightarrow \frac{2 - z^{-1}}{1 - z^{-1}} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The control weighting F_{ck} is defined as the following discrete lead term.

$$F_{ck} \rightarrow \frac{-1}{1 - z^{-1}} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

7.4 Simulation Results

To quantify the performance of the *NGMV* controller, the track error returned for scenario 2 – target motion in the *z*-axis only – was computed for a basic *PI* controller, *PI+cosecant* correction and *NGMV*. The tracking performance for a close nadir pass scenario (1cm screen coordinates corresponding to a nadir miss angle of 2.3deg), is shown below in Figure 7:16. This close to the nadir singularity the azimuth demand resulting from the kinematics changes rapidly with a large error forming immediately prior to the nadir due to the rate limit on the azimuth gimbal. In all three responses, the underlying *PI* controller is identical, differences in the responses due solely to the additional nonlinearities in the cosecant correction and *NGMV* controllers.

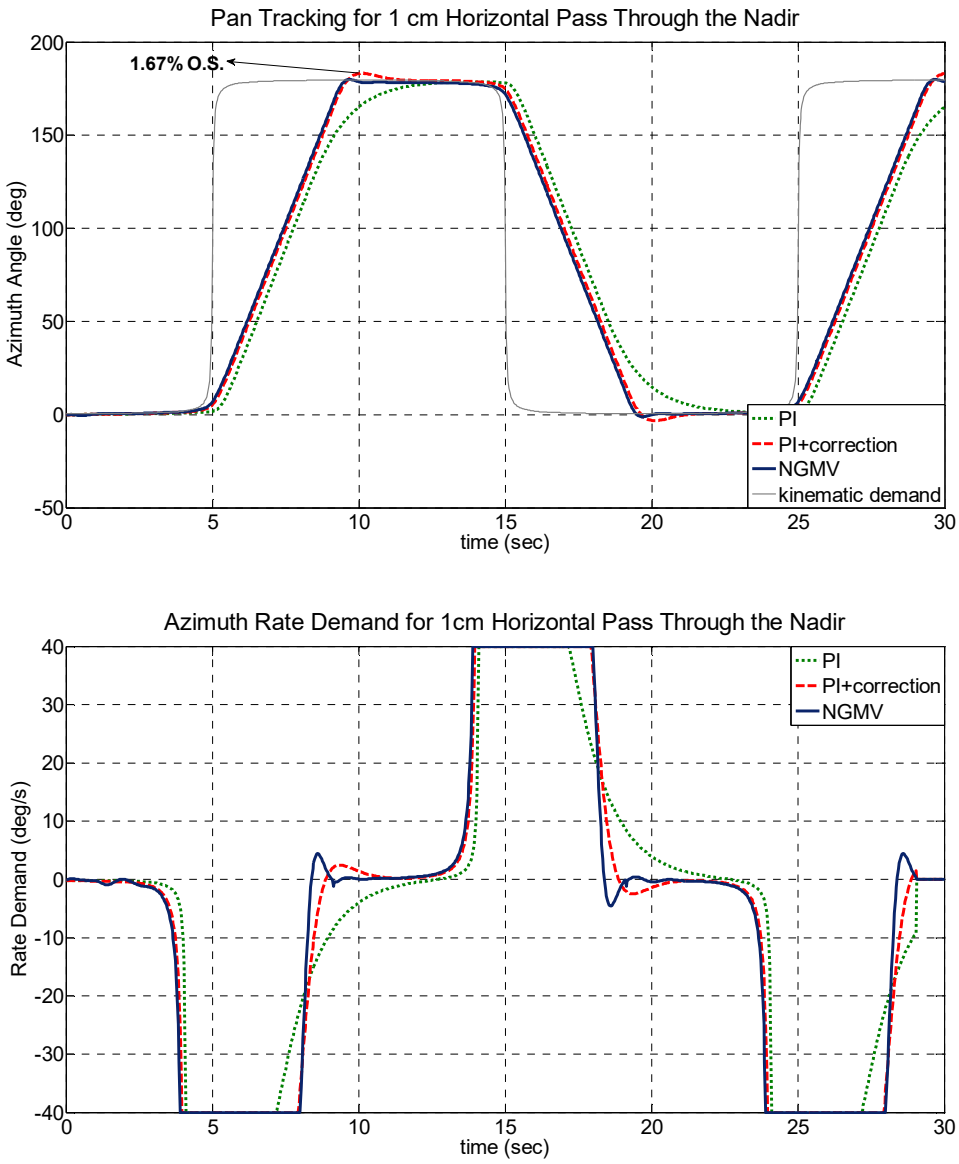


Figure 7:16 Kinematic and achieved azimuth angles for a 1cm *Nadir* pass scenario.

All three controllers are very well damped with minimal overshoot and little or no ringing. The response of the *PI* controller is the slowest of the three, as expected due to the presence of the cos gain term in the measurement path (note: although the gain has reduced, the large error signal forces saturation of the rate loop). Relating this result to the key requirements of a *DIRCM* tracker (maximise energy on target), the *NGMV* controller settles down after the first nadir pass at 10seconds, the *PI+Cosecant* settles after 11sec and the *PI* 12sec. This suggests the *NGMV* controller would provide more time-on-target of the *DIRCM* defeat laser, increasing the probability of missile confusion.

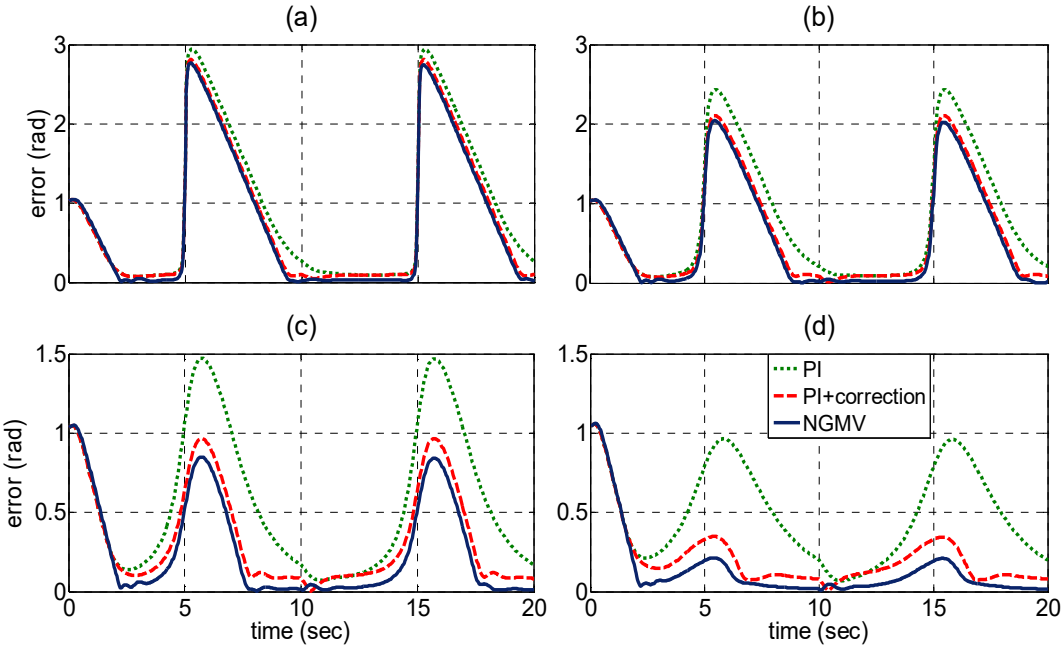


Figure 7:17 Total *LoS* track error for nadir pass geometries of (a) 1cm, (b) 5cm, (c) 20cm and (d) 40cm.

While tracking performance close to the *Nadir* is important, controller performance should be maintained throughout the field-of-regard. Figure 7:17 shows the radial track error (angle between boresight and target vectors) for four nadir pass conditions: 1cm, 5cm, 20cm and 40cm. In all cases the same trend is observed – cosecant correction outperforms the *PI* alone and the *NGMV* consistently provides better tracking performance.

7.5 Final Remarks

A number of contributions to the sightline control literature were presented and discussed in this section. First, the architecture, functionality and intended operation of the *Sightline Control laboratory* were presented. The suitability of the *SCL* hardware for investigation of track loop operation in the neighbourhood of the nadir was demonstrated and the process of system modelling, identification and validation discussed. The mathematical models of both gimbal axes and *SCL* kinematics were validated against experimental results.

Through a detailed derivation and explanation of the key design parameters in *NGMV* controller design, the suitability of an *NGMV* track-loop controller in operation close to the nadir was shown. The primary benefit of this technique is the ability to include the system nonlinearities, the azimuth axis measurement singularity in this case, directly within the controller but shielded from the designer. The *NGMV* controller was consistently the best performing controller with minimal additional tuning. This suggests *NGMV* is a practical synthesis technique for sightline controllers.

Chapter 8 Conclusions

The principal aim of this work was to explore the use of *LPV* models with various nonlinear and predictive control algorithms. In particular the *Nonlinear Predictive Generalised Minimum Variance* (*LPV-NPGMV*) approach is explored and its value demonstrated in three nonlinear control applications. The overall work was based on the *NGMV* method established by Grimble for systems with output subsystems involving *LTI* blocks. The main theme here was to exploit *LPV* system models with *Model Predictive Control* (*MPC*) algorithms. There was also a focus on the *Nonlinear Generalised Predictive Control* (*NGPC*) approach, which is midway in complexity between *NGMV* and *NPGMV* solutions. The work described focussed on design questions and was mainly concerned with the implementation and performance aspects of the various algorithmic formulations.

The key characteristic that differentiates the *NGMV* controller from the traditional approaches mentioned in Chapter 1, is its flexibility in accommodating unknown (or unmodelled) system nonlinearities via its decomposition into a general nonlinear operator (considered as a "*black-box"* in application Chapter 5, 6 and 7). It is proven and demonstrated in various examples in and outside this work that given a small set of assumptions on the stability properties of the open-loop system the controller can be implemented without the use of linearization to cover the entire operating range of a nonlinear system.

The delay-free part of the model is contained in the internal controller loop. This attempts to cancel nonlinearities in the special limiting case of zero control costing. Deviations between the model and the real plant are handled with the appropriate definition of the linear dynamic error weighting term (P_c) and a linear or nonlinear dynamic control weighting (F_{ck}). In the basic controller formulation there are therefore only two major tuning terms. Two application examples were explored to reinforce the applicability and effectiveness of the basic state-space formulation of the *NGMV* control, in both cases demonstrating improvement in performance without involving laborious tuning and reconfiguration. In both designs, the full nonlinear plant was included within the *black box* internal delay-free component whereas the linear part was defined as a unity matrix. In the first example (Chapter 5) the wave disturbance was included in the disturbance model formulation within the algorithm calculation for improved disturbance rejection.

In Chapter 3 the basis for the derivation of a new paradigm in the *NGMV* family of controllers was set, starting with an introduction to *NMPC* which led to the concept of an *NGPC* adaptation for *LPV* systems. This progression in the thesis was chosen as the cost function derivation for the *NPGMV* is shown as a combination of that of the *GMV* and *NGPC* algorithms. The development of the *NPGMV* for *LPV* systems is then straight forward. Furthermore, the *LPV-NGMV* has been produced and its value explored in the additional section of application Chapter 5.

The part of this work which required the most significant amount of effort to complete was the wind turbine application described in Chapter 6. The main challenge was again the derivation

of an appropriate model for analysis and control and subsequently its adaptation to the *LPV* framework. Once the *LPV* model was validated against the full nonlinear system, its coding within the controller algorithm was relatively straightforward. The *Extended Kalman Filter* (*EKF*) was used to provide the initial estimates at each controller scan, based on which the full prediction errors was then produced. For this application minor retuning of the weightings proved to be necessary when the predictions horizon (N_p) factor changed. Similar to the previous examples, however, a relatively limited range of weightings provided reasonable performance across different scenarios and operating points. Ideally, something that could also be within the scope for future improvement, scheduling of the weightings should be provided to tackle this problem more effectively.

In the third example (Chapter 7) the Nadir singularity nonlinear characteristic was included in the control weighting (F_{ck}) and that showed further improvement in performance when tracking across the Nadir. In both cases tuning of the *NGMV* was based on the *PID* initial baseline configuration and was shown to cope nicely at different operating points without further modifications.

A major challenge with all examples was the adaptation of the control architecture to the specific application. Another challenge and especially in the *NPGMV* case was to find solutions that reduce computational burden of the algorithm. End weighting constraints and a control profile selection matrix have been provided to reduce the number of computed states within the algorithm, however other features have not been fully explored and that could be the scope for future work along with an extensive study on the robustness properties of the advanced versions of the *NGMV* range of controllers. That would potentially encourage the transition of these concepts to a realistic real-time easily implementable industrial controller. That being said, in the application described in Chapter 7 the basic algorithm has been implemented for the first time on a real-time *National Instruments PXI FPGA* controller to stabilise the physical Electro-Optic system which was a major step towards the latter.

Overall the *NGMV* controllers and their predictive cousins proved to yield a promising solution for dealing with these particular classes of nonlinear systems with many opportunities for further improvement and real-time implementation.

8.1 Discussion

To be able to judge whether advanced controls provides an improvement some criteria should be established. However, there are many requirements of good industrial controllers and to some extent the decision is subjective. Nevertheless, the following advantages seem clear:

1. Most industries are moving towards using physical models on which to base control designs since this enables more formalised design procedures to be used and the predictive control methods lend themselves to such an approach.

2. To be able to benchmark the performance of the system a cost function is often required and this is available naturally because of the problem of formulation. This can enable performance to be quantified and good control to be judged.
3. Classical controls can provide very adequate and good solutions but when systems are interacting and multivariable they are very much more difficult to control and again the predictive control methods lend themselves to this problem.
4. Classical control methods also do not account for disturbances in a very formal or optimal manner but the predictive control solutions can use the statistical information on disturbances. In the wind energy problem this is of course a central feature of designs.
5. Most classical design methods do not take account of nonlinearities very formally and the same applies to parameter variations rising in systems, the type of solution presented also accounts naturally for these difficulties.

There are of course obvious disadvantages of more advanced methods such as the additional complexity in both implementation and the levels of staff needed in the design office. However, with the increase in computing power over recent years and new formalised design procedures such problems are becoming less significant. It is of course the case that advance controls will not be used if classical methods can be considered adequate, even if they provide some improvements. Moreover, with the cost implications of faults and failures and with the loss in possible outputs that may arise there is real imperative to use more advanced methods in some applications. It seems likely that in future years advanced controls will be considered a necessary evil and companies that do not adopt such philosophies may suffer the economic consequences.

8.2 Suggestions for Future Work

8.2.1 Robust Control NGMV Design

The following idea was very recent and was not explored since the step to Robust Adaptive is significant and well beyond the aims of the thesis. In the practical implementation architecture that was suggested for the *NGMV* algorithms throughout this thesis, the plant model was assumed to be obtainable in a relatively high degree. In this section an extension to the original design is suggested, which could potentially tackle issues that may arise out of the uncertainty most models exhibit in reality. Utilising tools found in robust control theory, this uncertainty or mismatch between the model and the actual process can be fused into our design via the following representation, commonly referred to as additive perturbation [54]. The term $\bar{W}$ in this equation represent the plant operator W including the uncertainty as a ΔW perturbation.

$$\bar{W} = W + \Delta W \tag{8:1}$$

To put it into context with the previous *NGMV* derivation and more specifically with the division of the plant into linear and nonlinear terms, the latter accommodates the additive perturbation as shown in the following relation.

$$\bar{W}_k \to W_k + \Delta W_k = W_{0k}W_{1k} + W_{0k}\Delta W_{1k} = W_{0k}(W_{1k} + \Delta W_{1k}) = \left(W_{0k}z^{-k}u_0\right)(t) \quad (8:2)$$

and including the common delays term, $= z^{-k}W_{0k}(W_{1k} + \Delta W_{1k})$

The control structure shown in Figure 2:4 can be modified to encapsulate this as shown in Figure 8:1. It is useful to remind here, that in this *NGMV* structure as discussed previously, the nonlinear part of the model can accommodate possible unknown black box type sections of the real plant. It can therefore be a convenient structure in which a black box nonlinear part could accommodate the uncertainty term ΔW_{1k}. This term can encapsulate components like varying parameters due to change of operating range, to un-modelled or varying transport delay terms.

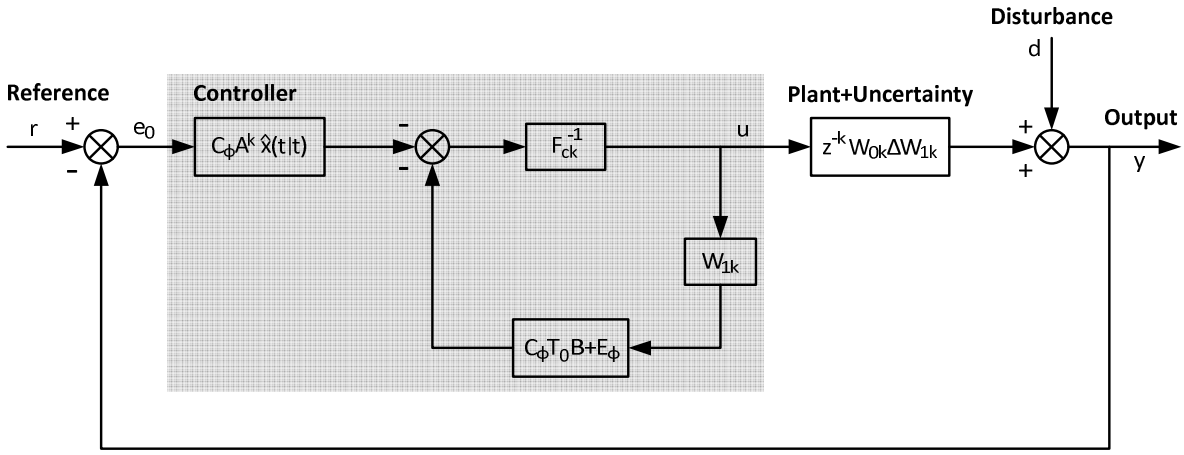


Figure 8:1 *NGMV* and model uncertainty structure.

However, the assumption of a time invariant, possible to model uncertainly factor, is made in this structure. If the latter is holds, then a successful design could be produced which would include a more accurate model of the system. It is proposed here that in the case where the uncertainty cannot be modelled or is time-varying, identification can be employed off-line (thus improving design at an early stage) or online in an adaptive manner, to update the model and improve compensation.

If an initial design is then made using the nominal plant model *W*, then the output of the parallel path will be the sum of the future disturbance estimate *d* (broken down to stochastic and deterministic terms) together with ΔW. The complete output of the plant, including uncertainty and disturbance terms is shown below.

$$y = d_\xi + d_d + \Delta W \bar{W} u \quad (8:3)$$

The control formulation including the alternative parallel loop formulation idea can be seen in the following diagram.

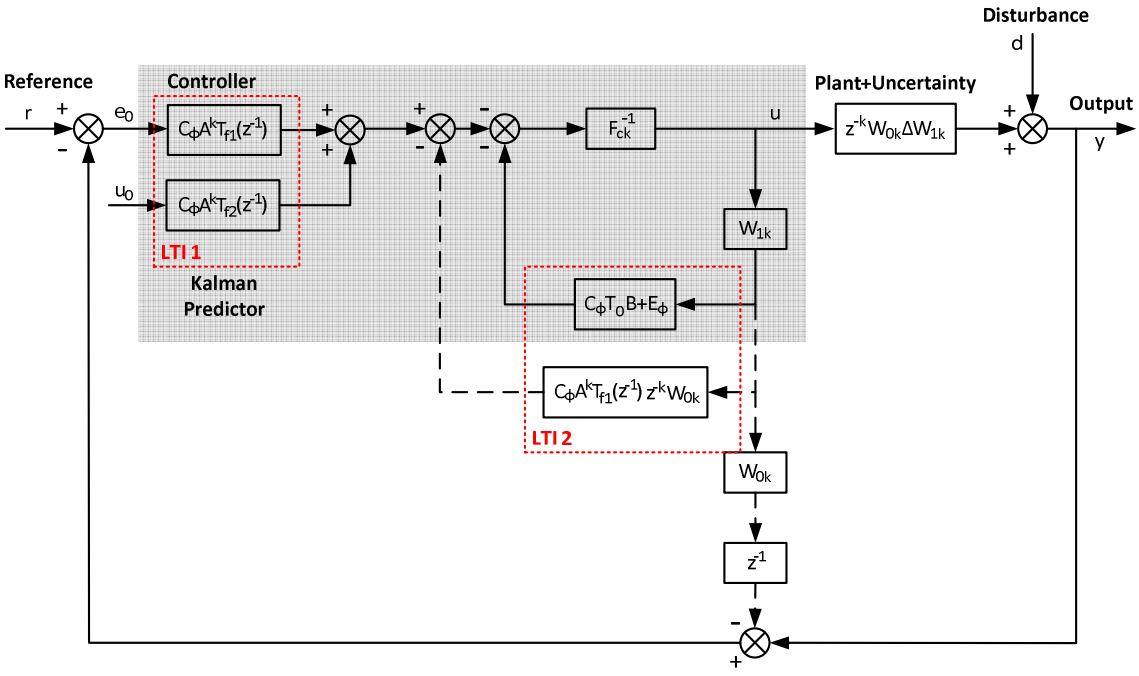


Figure 8:2 *NGMV* and model uncertainty structure expansion.

If there are no changes to the plant *W* (or modelling mismatch) then the parallel loop will remain stable as long as the plant remains open-loop stable. Notice however that $\bar{W}u$ is u_0 which actually represents the known nominal system. Updating the model with knowledge of disturbance can be easily formulated as a Least-Squares system identification problem using *ARMAX* models. The same idea could be used to the error introduced by the outer loop to identify uncertainty in the model as well.

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}\, u_0(k) + \frac{C(q^{-1})}{A(q^{-1})} e(k) \tag{8:4}$$

The adaptation of this idea to the *NGMV* framework is shown in the following figure.

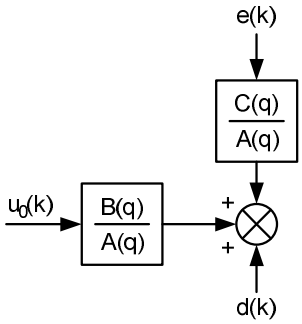


Figure 8:3 *ARMAX* model structure.

Normally this would be impractical as it would imply online identification of the whole plant $\frac{B}{A}$ (problematic for a plant with large number of states). However, in this case this is not necessary as the signal for the model uncertainty is known as explained in the following relations.

$$\widetilde{W} = (1 + \delta W)\overline{W} \tag{8:5}$$

$$e = \overline{W} - \widetilde{W} = \delta W \overline{W} \tag{8:6}$$

There is a bound for inaccuracies in the model and that is what is provided by this formulation; the term to be identified is a low order model (e.g. a 1st order model can be assumed to represent the uncertainty). After being identified, *ΔW* can then be used online to update the model (within the "black box" term). Another advantage with this is that the linear terms *LT1* and *LT2* only depend upon the reference and disturbance signals but not on what is actually within the "black box" hence W_k can be updated without affecting the rest of the controller elements. This can naturally lead to an adaptive, self-tuning formulation for the controller.

This approach provides a method of improving robustness when other techniques do not seem too successful. In a sense it uses system identification on a simple model to identify the uncertainty. The identification stage will just provide the best approximation to the uncertainty in whatever uncertainty description is chosen. It seems either we have to go into a formal design method, which is very complicated, or we have to employ a number of simple approaches which require trial and error optimization, if the best robustness is to be obtained. One of the main reasons is that we have no real link between uncertainly and the controller design. With the adaptive/uncertainty estimation method the form of uncertainty will be defined by the designer but the parameters will be estimated. We might even be able to restrict the variations of the estimated parameters so that unpredictable behaviour is not obtained. Using adaption in this manner was not considered in the original proposal since the focus was more on improving the robustness of the predictive controls. However, the experience from the project suggests off-line, or on-line system identification, be used and could play a useful role.

References

A. Aström, K.J., 2012. Introduction to stochastic control theory. Courier Corporation.

Adegas, F.D., Stoustrup, J., 2012. Structured control of LPV systems with application to wind turbines, in: 2012 American Control Conference (ACC). IEEE, pp. 756–761.

AEOLUS FP7 Project, Distributed Control of Large-scale Offshore Wind Farms, http://www.ict-aeolus.eu/index.html.

Allgower, F., Findeisen, R., 1998. Nonlinear predictive control of a distillation column, International Symposium on Non-linear Model Predictive Control, Switzerland.

Anderson, D., McGookin, M., Brignall, N., 2009. Fast Model Predictive Control of the Nadir Singularity in Electro-Optic Systems. Journal of guidance, control, and dynamics 32, 626–632.

Balas, G., 2002. Linear parameter-varying control and its application to aerospace systems, in: ICAS Congress Proceedings.

Balchen, J.G., Jenssen, N.A., Mathisen, E., Sælid, S., 1980. A Dynamic Positioning System Based on Kalman Filtering and Optimal Control 1, 135–163. doi:10.4173/mic.1980.3.1.

Bianchi, Fernando D., Battista, H. and Mantz, R. J. 2008. Wind turbine control systems. Principles, modelling and gain scheduling design. International Journal of Robust and Nonlinear Control, Volume 18, Issue 7, 10 May 2008, Pages 796–797.

Biannic, J.-M., 2013. Linear Parameter-Varying Control Strategies for Aerospace Applications, in: Sename, O., Gaspar, P., Bokor, J. (Eds.), Robust Control and Linear Parameter Varying Approaches, Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, pp. 347–373. doi:10.1007/978-3-642-36110-4_14.

Bitmead, R.R., Gevers, M., Wertz, V., 1990. Adaptive optimal control: The thinking man's GPC. Elsevier.

Breivik, M., Fossen, T.I., 2004. Path following for marine surface vessels, in: OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04. IEEE, pp. 2282–2289.

Brooms, A.C., Kouvaritakis, B., 2000. Successive constrained optimization and interpolation in non-linear model based predictive control. International Journal of Control 73, 312–316. doi:10.1080/002071700219669.

Bruzelius, F., 2004. Linear parameter-varying systems-an approach to gain scheduling. Chalmers University of Technology.

Burton, T., Sharpe, D., Jenkins, N., Bossanyi, E., 2001. Wind energy handbook. John Wiley & Sons.

Camacho, E., 1993. Constrained generalized predictive control. IEEE Transactions on Automatic Control.

Camacho, E.F., Alba, C.B., 2013. Model predictive control. Springer Science & Business Media.

Cannon, M., Kouvaritakis, B., 2002. Efficient constrained model predictive control with asymptotic optimality. SIAM journal on control and optimization 41, 60–82.

Cannon, M., Kouvaritakis, B., 2001. Open-loop and closed-loop optimality in interpolation MPC. IEE CONTROL ENGINEERING SERIES 131–150.

Casavola, A., Mosca, E., 1998. A predictive reference governor with computational delay. European journal of control 4, 241–248.

Clarke, D.W., Hastings-James, R., 1971. Design of digital controllers for randomly disturbed systems. Electrical Engineers, Proceedings of the Institution of 118, 1503–1506.

Clarke, D.W., Mohtadi, C., 1989. Properties of generalized predictive control. Automatica 25, 859–875.

Clarke, D.W., Mohtadi, C., Tuffs, P.S., 1987. Generalized predictive control—Part I. The basic algorithm. Automatica 23, 137–148.

Cloutier, J., 1997. State-Dependent Riccati Equation Techniques. Proceedings ACC. New Mexico.

Corriou, J.-P., 2013. Process control: theory and applications. Springer Science & Business Media.

Cutler, C.R., Ramaker, B.L., 1980. Dynamic matrix control??A computer control algorithm. Joint Automatic Control Conference 72. doi:10.1109/JACC.1980.4232009.

Deng, H., Krstić, M., 1997. Stochastic nonlinear stabilization—I: a backstepping design. Systems & Control Letters 32, 143–150.

Do, A.-L., Poussot-Vassal, C., Sename, O., Dugard, L., 2013. LPV Control Approaches in View of Comfort Improvement of Automotive Suspensions Equipped with MR Dampers, in: Sename, O., Gaspar, P., Bokor, J. (Eds.), Robust Control and Linear Parameter Varying Approaches, Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, pp. 183–212. doi:10.1007/978-3-642-36110-4_7.

Findeisen, R., Allgöwer, F., 2002. An introduction to nonlinear model predictive control, in: 21st Benelux Meeting on Systems and Control. Technische Universiteit Eindhoven Veldhoven, pp. 119–141.

Fossen, T.I., Grovlen, A., 1998. Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. IEEE transactions on control systems technology 6, 121–128.

Garcia, C.E., Morshedi, A.M., 1986. Quadratic Programming Solution of Dynamic Matrix Control (QDMC). ResearchGate 46, 73–87. doi:10.1080/00986448608911397

Gaspar, P., Szabo, Z., Bokor, J., 2010. LPV design of fault-tolerant control for road vehicles. IEEE, pp. 807–812. doi:10.1109/SYSTOL.2010.5676060

Goodwin, G.C., Rojas, O., Takata, H., 2001. Nonlinear control via generalized feedback linearization using neural networks. Asian Journal of Control 3, 79–88.

Grimble, M.J., 2007. GMV control of non-linear continuous-time systems including common delays and state-space models. International Journal of Control 80, 150–165.

Grimble, M.J., 2006a. Design of generalized minimum variance controllers for nonlinear multivariable systems. International Journal of Control, Automation and Systems 4, 281–292.

Grimble, M.J., 2006b. Robust industrial control systems: optimal design approach for polynomial systems. John Wiley & Sons.

Grimble, M.J., 2005. Non-linear generalized minimum variance feedback, feedforward and tracking control. Automatica 41, 957–969.

Grimble, M.J., 2004. GMV control of nonlinear multivariable systems. Control 2004.

Grimble, M.J., 2001. Industrial control systems design. Wiley Chichester.

Grimble, M.J., 2000. Restricted-structure LQG optimal control for continuous-time systems. IEE Proceedings-Control Theory and Applications 147, 185–195.

Grimble, M.J., 1988. Generalized minimum variance control law revisited. Optimal Control Applications and Methods 9, 63–77.

Grimble, M.J., 1981. A control weighted minimum-variance controller for non-minimum phase systems. International Journal of Control 33, 751–762.

Grimble, M., Johnson, M., 1988. Optimal control and stochastic estimation Vols. I and II. John Wiley, Chichester.

Grimble, M.J., Majecki, P., 2015. Non-linear predictive generalised minimum variance state-dependent control. IET Control Theory & Applications 9, 2438–2450.

Grimble, M.J., Majecki, P., 2010. State-space approach to nonlinear predictive generalised minimum variance control. International Journal of Control 83, 1529–1547.

Grimble, M.J., Majecki, P., 2006. H/spl infin/Control of Nonlinear Systems with Common Multi-channel Delays, in: 2006 American Control Conference. IEEE, pp. 5626–5631.

Grimble, M.J., Majecki, P., 2005. Nonlinear generalized minimum variance control under actuator saturation. IFAC Proceedings Volumes 38, 993–998.

Grimble, M.J., Majecki, P., Giovanini, L., 2007. Polynomial approach to nonlinear predictive GMV control, in: Control Conference (ECC), 2007 European. IEEE, pp. 4546–4553.

Grimble, M.J., Pang, Y., 2007. NGMV control of state dependent multivariable systems, in: 46th IEEE Conference on Decision and Control, 2007. IEEE, pp. 1628–1633.

Grimble, M.J., Patton, R.J., Wise, D.A., 1980. Use of Kalman filtering techniques in dynamic ship-positioning systems, in: IEE Proceedings D-Control Theory and Applications. IET, p. 93.

Grimble, M., Zhang, Y. and Katebi, M.R., 1993. H∞ based ship autopilot design. 10th Ship Symposium, Canada.

Hammett, K.D., 1997. Control of Nonlinear Systems via State Feedback State-Dependent Riccati Equation Techniques. DTIC Document.

Henrion, D., Arzelier, D., Peaucelle, D., 2003. Positive polynomial matrices and improved LMI robustness conditions. Automatica 39, 1479–1485.

Hernandez, E., Arkun, Y., 1993. Control of nonlinear systems using polynomial ARMA models. AIChE Journal 39, 446–460.

Isidori, A., 2013. Nonlinear control systems. Springer Science & Business Media.

Jonkman, J., Butterfield, S., Musial, W., Scott, G., 2009. Definition of a 5-MW reference wind turbine for offshore system development. National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060.

Khalil, H., 1996. Nonlinear Systems, 2nd edition. Prentice-Hall.

Katebi, M.R., Grimble, M.J., Zhang, Y., 1997. H∞ robust control design for dynamic ship positioning. IEE Proceedings-Control Theory and Applications 144, 110–120.

Katebi, M., Morardi, M., 2000. A Comparison of Stability and Performance Robustness of Multivariable PID Tuning Methods. Proceedings of PID Conference, Barcelona, Spain.

Kennedy, P.J., Kennedy, R.L., 2003. Direct versus indirect line of sight (LOS) stabilization. IEEE Transactions on control systems technology 11, 3–15.

Kothare, M.V., Balakrishnan, V., Morari, M., 1996. Robust constrained model predictive control using linear matrix inequalities. Automatica 32, 1361–1379.

Kouvaritakis, B., Cannon, M., Rossiter, J.A., 2000. Stability, feasibility, optimality and the degrees of freedom in constrained predictive control, in: Nonlinear Model Predictive Control. Springer, pp. 99–113.

Kouvaritakis, B., Cannon, M., Rossiter, J.A., 1999. Non-linear model based predictive control. International Journal of Control 72, 919–928.

Kwon, W., Pearson, A., 1978. On feedback stabilization of time-varying discrete linear systems. IEEE Transactions on Automatic Control 23, 479–481.

Kwon, W., Pearson, A., 1977. A modified quadratic cost problem and feedback stabilization of a linear system. IEEE Transactions on Automatic Control 22, 838–842.

Lechevin, N., Rabbath, C.A., 2007. Backstepping guidance for missiles modeled as uncertain time-varying first-order systems, in: 2007 American Control Conference. IEEE, pp. 4582–4587.

Leith, D.J., Leithead, W.E., 1997. Implementation of wind turbine controllers. International Journal of Control 66, 349–380.

Maciejowski, J.M., 2002. Predictive control: with constraints. Pearson education.

Martin, P., Katebi, R., 2005. Multivariable PID tuning of dynamic ship positioning control systems. Journal of Marine Engineering & Technology 4, 11–24.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36, 789–814.

Michalska, H., Mayne, D.Q., 1993. Robust receding horizon control of constrained nonlinear systems. IEEE transactions on automatic control 38, 1623–1633.

Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. Computers & Chemical Engineering 23, 667–682.

Morari, M., Zafiriou, E., 1989. Robust process control. Prentice hall Englewood Cliffs, NJ.

Murray-Smith, R., Shorten, R., 2005. Switching and Learning, LNCS 3355, pp. 185–200, 2005.c. Springer-Verlag, Berlin Heidelberg.

ORDYS, A.W., Clarke, D.W., 1993. A state-space description for GPC controllers. International Journal of Systems Science 24, 1727–1744.

Østergaard, K.Z., Brath, P., Stoustrup, J., 2007. Estimation of effective wind speed, in: Journal of Physics: Conference Series. IOP Publishing, p. 012082.

Østergaard, K.Z., Stoustrup, J., Brath, P., 2009. Linear parameter varying control of wind turbines covering both partial load and full load conditions. International Journal of Robust and Nonlinear Control 19, 92–116.

P. Martin, 2004, The Development of Advanced Multivariable, Linear & Nonlinear Control Design Methods, with Applications to Marine Vessels, PhD thesis, Industrial Control Centre, University of Strathclyde, Glasgow., n.d.

Pierson, W.J., Moskowitz, L., 1964. A proposed spectral form for fully developed wind seas based on the similarity theory of SA Kitaigorodskii. Journal of geophysical research 69, 5181–5190.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control engineering practice 11, 733–764.

Richalet, J., Rault, A., Testud, J., Papon, J., 1976. Algorithmic control of industrial processes. Proceedings of the 4th IFAC symposium on identification and system parameter estimation, (pp. 1119–1167).

Richalet, J., 1993. Industrial applications of model based predictive control. Automatica 29, 1251–1274.

Richalet, J., Rault, A., Testud, J.L., Papon, J., 1978. Model predictive heuristic control: Applications to industrial processes. Automatica 14, 413–428.

Rue, A., 1969. Stabilization of precision electrooptical pointing and tracking systems. IEEE Transactions on Aerospace and Electronic Systems 5, 805–819.

Rugh, W.J., Shamma, J.S., 2000. Research on gain scheduling. Automatica 36, 1401–1425.

Schetzen, M., 1980. The Volterra and Wiener theories of nonlinear systems.

Scokaert, P.O., Mayne, D.Q., Rawlings, J.B., 1999. Suboptimal model predictive control (feasibility implies stability). IEEE Transactions on Automatic Control 44, 648–654.

Shamma, J.S., Athans, M., 1990. Analysis of gain scheduled control for nonlinear plants. IEEE Transactions on Automatic Control 35, 898–907.

Shneider, W.P., others, 1969. Dynamic positioning systems, in: Offshore Technology Conference. Offshore Technology Conference.

Slotine, J.-J.E., Li, W., others, 1991. Applied nonlinear control. prentice-Hall Englewood Cliffs, NJ.

Sznaier, M., Cloutier, J., Hull, R., Jacques, D., Mracek, C., 1998. A receding horizon state dependent Riccati equation approach to suboptimal regulation of nonlinear systems, in: Decision and Control, 1998. Proceedings of the 37th IEEE Conference on. IEEE, pp. 1792–1797.

Taylor, J.H., 1999. Describing functions. Wiley Encyclopedia of Electrical and Electronics Engineering.

Utkin, V., Guldner, J., Shi, J., 2009. Sliding mode control in electro-mechanical systems. CRC press.

Utkin, V., Lee, H., 2006. Chattering problem in sliding mode control systems, in: International Workshop on Variable Structure Systems, 2006. VSS'06. IEEE, pp. 346–350.

Utkin, V.I., 2013. Sliding modes in control and optimization. Springer Science & Business Media.

Vidyasagar, M., 2002. Nonlinear systems analysis. Siam.

Wang, L., 2009. Model predictive control system design and implementation using MATLAB®. Springer Science & Business Media.

Watanabe, K., Ito, M., 1981. A process-model control for linear systems with delay. IEEE Transactions on Automatic control 26, 1261–1269.

Wellstead, P.E.E., Zarrop, M.B., 1991. Self-tuning systems: control and signal processing. John Wiley & Sons, Inc.

White, W.N., Patenaude, J., Foss, M., García, D., 2009. Direct Lyapunov approach for tracking control of underactuated mechanical systems, in: 2009 American Control Conference. IEEE, pp. 1341–1346.

Yamamoto I., Daigo 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., n.d.

Appendix A.1 Application Chapter 5 Simulations Code.

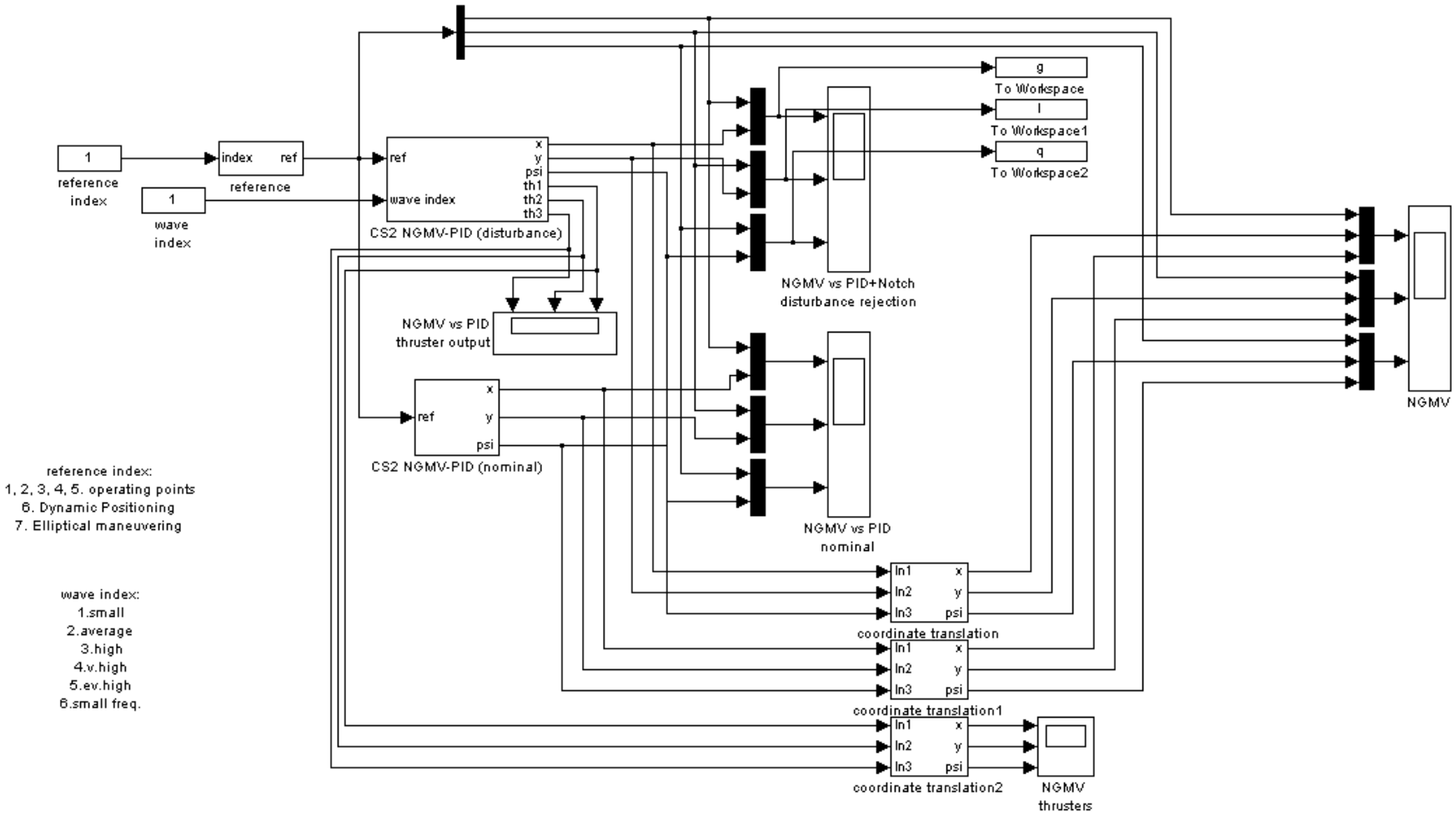
Main Simulation Master File Code (MATLAB):

```
% THE NGMV SCHEME IN SHIP MANEUVERING AND DP (Dynamic Positioniong)

% The Cybership II nonlinear fully actuated model is used in
simulations. Linear models driven
% by white noise are used in the reference and disturbance parts of
the model. NGMV performance
% is compared against a PID-Notch filter in simple step reference
tracking, DP and Manoeuvring
% examples.

% note: thruster nonlinear models are included in the overall
nonlinear model of the ship which
%       is fully simulated in Simulink blocks. All components of the
simulation are in discrete
%       time.

gensym z^-1
clear; clc
d2r=pi/180*eye(3,3);
dw=1*eye(3,3);

%% PLANT DEFINITION
% Cybership II Model

% identification parameters
m=23.8000; Iz=1.7600; xg=0.0460;
Xu=-0.7225; Xuu= -1.3274; Xuuu= -5.8664; Xud=- 2.0;
Yvd=-10.0; Yv= -0.8612; Yvv= -36.2823; Yrd=0.0;
Yr= 0.1079; Yrv=-0.805; Yvr=-0.8450; Yrr=-3.450;
Nrv=0.130; Nr=-1.90; Nvr=.08;Nrr=-0.750; Nvd=0.0;
Nv= 0.1052; Nrd= - 1.0; Nvv=5.0437;

% disturbance vector
b1=-1.5; b2=1.5; b3=0;

% inertial vector
M=[m-Xud 0 0; 0 m-Yvd m*xg-Yrd;0 m*xg-Nvd Iz-Nrd];
Minv=inv(M);

% sampling time & plant delays
Ts=0.01;
Dk=z^-1*eye(3);
stop_time=200;

% ELLIPTIC REFERENCE TRAJECTORY
x=0; y=0; fi=0;
for theta=1:360
    i=theta;
```

```
    [x(i),y(i),fi(i)]=ellip(theta);

end

%% CONTROLLER DESIGN
% PID DESIGN
Kp2=diag([50 100 100]);
Ki2=diag([20 20 20]);
Kd2=diag([50 50 50]);

% PID tuning for Notch-filter configuration
Kpn=1000*diag([1 1 1]);
Kin=0*diag([0.01 10 8]);
Kdn=1*diag([200 60 30]);

% Notch filter design
a=1;
b=10;
wnn=.5;  %natural frequency (rad/sec)
z1=.1;  %damping factor 1
z2=3;  %damping factor 2

notch=1*(10/b)*tf([a 6*z1*wnn wnn^2],[1 1*z2*wnn wnn^2]);
notched=c2d(notch,Ts,'tustin')*eye(3,3);

% DEFINITION OF THE NGMV MODELS
% LINEAR PLANT COMPONENT (W0)
% as an integrator
% (the integrator used to obtain the displacement from the velocities)
sys=filt(1,[1 -0.9],Ts)*eye(3,3);
W0=sys;
[B0k,A0]=lti2lmf(W0);

% REFERENCE LINEAR MODEL (Wr)
Wr=filt(0.5,[1 -0.8],Ts)*eye(3,3);
[Er0,Ar]=lti2lmf(Wr);

% DISTURBANCE DEFINITION (Wd)
% wave disturbance (according to Pierson-Moskowitz spectrum for rough
seas)
% a 2nd order transfer, inverse notch filter look-alike, with
reasonable damping on the sides
% fixed on 0.8 rad/sec.
ad=a;     %denominator coefficient
wnd=.5;   %natural frequency (rad/sec)

dist=tf([1 2*wnd wnd^2],[a 1*wnd wnd^2]);
distd=c2d(dist,Ts,'tustin')*eye(3,3);
 Wd=filt([0.0003786 0.0007573 0.0003786],[1 -1.944
0.9457],Ts)*eye(3,3); %alternative disturbance
% Wd=0.001*distd;
[Cd0,Ad]=lti2lmf(Wd);

% NGMV DESIGN
% PID-based design
Kp=diag([38 12 25]);
Ki=diag([0.02 0.01 0.01]);
```

```
Kd=diag([0.1 0.1 0.1]);

% Kp=diag([80 70 70]);
%
% Ki=1*diag([0.02 0.01 0.01]);
%
% Kd=60*diag([0.1 0.1 0.1]);

% Pc weighting definition
C=dpid2tf(Kp,Ki,Kd,Ts);
Pc=tf(C);  Pc=1*Pc;  Pc.var='z^-1';
[Pcn,Pcd]=lti2rmf(Pc);

% Frequency-based design
% Pc=distd;
%
% Pc=1*Pc;  Pc.var='z^-1';
dil=tf([1 1.5*wnn wnn^2],[1 1*wnn wnn^2]);
% dil=tf([1 1.2*wnn wnn^2],[2 .8*wnn wnn^2]);
dild=c2d(dil,Ts,'tustin')*eye(3,3);
% dil=tf([1 2*wnn wnn^2],[7 1*wnn wnn^2]);
% dild=c2d(dil,Ts,'tustin')*eye(3,3);
Pc1=1*dild;
Pc1=1*Pc1;  Pc1.var='z^-1';
[Pcn1,Pcd1]=lti2rmf(Pc1);

% Fck weighting definition
Fck=-0.08*(filt([1 -0.7],1,Ts))*eye(3,3);

% Common Denominator
 [A,Bk,Cd,Er]=lmf2lcd(B0k,A0,Cd0,Ad,Er0,Ar);

% NGMVC law formulation
[c_cascade,c_loop]=ngmvc(A,Dk,Bk,Cd,Er,Pcn,Pcd,Ts);        %for the
PID-based design

[c_cascade1,c_loop1]=ngmvc(A,Dk,Bk,Cd,Er,Pcn1,Pcd1,Ts);    %for the
freq.-based design
```

Main Simulation Diagram (SIMULINK):



Appendix A.2 Application Chapter 6 Simulations Code.

```
%% Wind Turbine Control Master File
% The following are defined in this file:
% 1. Parameters for a 5MW Wind Turbine model with tower dynamics.
% 2. Parameters for PI, GPC, NGMVss and  lpv-NPGMV controllers.
% 3. Initialisation parameters for different simulation scenarios.

clear all; close all; clc
run('C:\Users\...\Documents\MATLAB\ngmv_toolbox_march_2013\ngmvinit')
addpath isclib

%% Define wind turbine component parameters
% Cut-in and cut-out speeds
v_cutin = 3;                %[m/s] cut-in speed
v_cutout = 25;              %[m/s] cut-out speed
beta_br = 0.0;              %[deg] below-rated pitch angle

% Turbine output at rated
wg_nom = 1173.7*pi/30;      %[rad/s] rated generator speed
taug_nom = 43093.55;        %[Nm] rated generator torque
Pg_nom = 5e6;               %[W] rated power

% Pitch servo (2nd order model)
wn_pitch = 10;              %[rad/s] pitch angle servo natural
frequency
xi_pitch = 0.55;            %[-] pitch angle servo damping
beta_max = 91.77;           %[deg] pitch magnitude constraints
beta_min = 0;
betadot_max = 8;            %[deg/s] pitch rate constraints
betadot_min = -8;

% Aerodynamics
R = 63;                     %[m] rotor radius
rho = 1.225;                %[kg/m3] air density
c = 0.5*rho*pi*(R)^3;
load wtpcoeffs              %load WT performance coefficients (Cq, Ct,
Cp)

% Drive Train (2DOF)
K_LS = 8.67637e8;           %[Nm/rad] LSS stiffness
D_LS = 6.215e6;             %[Nms/rad] LSS damping
N = 97;                     %[-] gearbox ratio
etam = 1;                   %[-] mechanical efficiency
J_g = 534.116;              %[kg*m2] moment of inertia, generator
J_r = 35444.067;            %[kg*m2] moment of inertia, rotor
D_g = 0;                    %[Nms/rad] generator damping    - not used
D_r = 0;                    %[Nms/rad] rotor damping        - not used

% Generator and Converter
tau_gc = 0.1;               %[s] generator torque time constant
etagc = 1;                  %[-] electrical efficiency
taug_max = 474029.1;        %[Nm] torque constraints
```

```
taug_min = 0.0;

% Tower Fore-Aft Dynamics
mt_nacrot = 350000;         %[kg] mass of nacelle and rotor
mt_tow = 347460;            %[kg] mass of tower
mt_FA = mt_nacrot + mt_tow; %[kg] effective moving mass for tower
deflection in fore-aft direction
wn1_t = 0.321;              %[rad/s] 1st tower fore-aft mode: natural
frequency
xi1_t = 0.8;                %[-] 1st tower fore-aft mode: damping
factor
Kt_FA = mt_FA*wn1_t^2;      %[N/m] computed stiffness
Dt_FA = 2*mt_FA*xi1_t*wn1_t;%[N.s/m] computed damping#



%% Optimal trajectories and LUT data generation
% effective wind speed
v = [2.5:0.5:8 8.1:0.2:11 11.5:0.5:15 16:39.5]';
wind_in = v;
plot_style = 'b.-';

% WT structure
par_wt =
struct('v_cutin',v_cutin,'v_cutout',v_cutout,'wind_in',wind_in,...
    'wg_nom',wg_nom,'taug_nom',taug_nom,'Pg_nom',Pg_nom,...
    'wn_beta',wn_pitch,'xi_beta',xi_pitch,'R',R,'rho',rho,'c',c,...

'K_LS',K_LS,'D_LS',D_LS,'N',N,'etam',etam,'J_g',J_g,'J_r',J_r,'D_g',D_
g,'D_r',D_r,...

'tau_gc',tau_gc,'etagc',etagc,'m_FA',mt_FA,'K_FA',Kt_FA,'D_FA',Dt_FA,.
..
    'Lambda',Lambda,'Pitch',Pitch,'Cp',Cp,'Cq',Cq,'Ct',Ct);

% Optimal reference trajectory as a function of wind speed
[Pstar,betastar,wgstar,taugstar] = ORC(v,beta_br,par_wt);
lambda = wgstar/N*R./v;
Cpi = interp2(Pitch,Lambda,Cp,betastar,lambda);

% LUTs
pitch_out = betastar;
torque_out = taugstar;
power_out = Pstar;
wg_out = wgstar;

[dCq_lambda,dCq_beta] = lutder2d(Lambda,Pitch,Cq); % Cq partial
derivatives
[dCp_lambda,dCp_beta] = lutder2d(Lambda,Pitch,Cp); % Cp partial
derivatives
[dCt_lambda,dCt_beta] = lutder2d(Lambda,Pitch,Ct); % Ct partial
derivatives

par_LUT =
struct('dCq_lambda',dCq_lambda,'dCq_beta',dCq_beta,'dCp_lambda',...
```

```
dCp_lambda,'dCp_beta',dCp_beta,'dCt_lambda',dCt_lambda,'dCt_beta',dCt_
beta);

par_wt = catstruct(par_wt,par_LUT);

%% Plot Optimal Trajectories
hfig = figure; set(gcf,'Name','Optimal trajectories')
subplot(331)
plot(v,Pstar*1e-6,plot_style); grid on; box off; hold on
title('Optimal power curve'); xlabel('v [m/s]'); ylabel('P^* [MW]')

subplot(337)
plot(v,wgstar,plot_style); grid on; box off; hold on
title('Optimal generator speed'); xlabel('v [m/s]');
ylabel('\omega_g^* [rad/s]')

subplot(338)
plot(v,betastar,plot_style); grid on; box off; hold on
title('Optimal pitch angle'); xlabel('v [m/s]'); ylabel('\beta^*
[deg]')

subplot(334)
plot(v,taugstar*1e-3,plot_style); grid on; box off; hold on
title('Optimal generator torque'); xlabel('v [m/s]'); ylabel('\tau_g^*
[kNm]')

subplot(336)
plot(v,Cpi,plot_style); grid on; box off; hold on
title('Cp'); xlabel('v [m/s]'); ylabel('Cp [-]')

subplot(339)
plot(v,lambda,plot_style); grid on; box off; hold on
title('Tip speed ratio'); xlabel('v [m/s]'); ylabel('\lambda [-]')

subplot(332)
inds = find(wgstar~=0);
plot(wgstar(inds),taugstar(inds)*1e-3,plot_style); grid on; box off;
hold on
title('Torque-speed curve'); xlabel('\omega_g [rad/s]');
ylabel('\tau_g [kNm]')

subplot(335)
inds = find(wgstar~=0);
plot(wgstar(inds),Pstar(inds)*1e-6,plot_style); grid on; box off; hold
on
title('Power-speed curve'); xlabel('\omega_g [rad/s]'); ylabel('P_g
[MW]')

%% Simulation Parameters
t_sim = 100;  %[s] simulation time
Ts = 0.02;    %[s] controller sample time
v_0 = 15;     %[m/s] wind speed

load turbwind
vmean = vmean15a*0.6+7;
figure
```

```
subplot(211); plot(tx,vmean,'b'); title('Turbulent wind'); ylabel('v
[m/s]')
% Wind data for simulation
avg_wind_speed2 = [tx' vmean']; t_sim = tx(end);

Psteps = 1e6*[zeros(39,1); -0.2*ones(40,1); -0.6*ones(40,1); -
1.7*ones(40,1); -1*ones(40,1); ...
    -0.5*ones(40,1); -0.1*ones(40,1); zeros(39,1);];
Psteps_str = [[1:length(Psteps)]' Psteps];

wg_steps = (1/5)*wg_nom*[zeros(39,1); -0.2*ones(40,1); -
0.6*ones(40,1); -1.7*ones(40,1); -1*ones(40,1); ...
    -0.5*ones(40,1); -0.1*ones(40,1); zeros(39,1);];
wg_steps_str = [[1:length(wg_steps)]' wg_steps];


load gustwind
gust = gust17;
gust = gust(1)-2 + (gust-gust(1))*0.8;  % gust scaling
gust_n =
[gust(1:301),gust(1:301),gust(1:200),gust(302:500),gust(500:698)];
subplot(212); plot(tx,gust_n,'b'); title('Wind gust'); ylabel('v
[m/s]')
avg_wind_speed = [tx' gust_n']; t_sim = tx(end);

% Initial Conditions
[P_0,beta_0,wg_0,taug_0] = ORC(v_0,beta_br,par_wt);
betadot_0 = 0;
wr_0 = wg_0/N;
lambda_0 = wr_0*R/v_0;
Cq_0 = interp2(Lambda,Pitch,Cq',lambda_0,beta_0);
tau_LSS_0 = c*Cq_0*v_0^2;
theta_0 = tau_LSS_0/K_LS;
Ct_0 = interp2(Lambda,Pitch,Ct',lambda_0,beta_0);
Ft_0 = c/R*Ct_0*v_0^2;
yt_0 = Ft_0/Kt_FA;
ytdot_0 = 0;

%% CONTROL DESIGN
% Model Linearisation
% States: [beta,beta_dot,theta_d,omega_g,omega_r,tau_g,xt,xt_dot]
% Inputs: [v_wind,tau_g_ref,beta_ref]
% Outputs:
[omega_r,omega_g,beta_rad,tau_r,tau_g,P_g,v_hub,tau_LS,Ftx,xt_dot]
[A,B,C,D,X0,U0,Y0,inputnames,outputnames,statenames] =
wtlin_nrel(v_0,par_wt);
wt_lin = ss(A,B,C,D);
wt_lind = c2d(wt_lin, Ts);
wt_lind.c(3,1)=1;

% model dimensions - full range
nx0 = 8;     % No of states
nu = 2;      % No of control inputs
nyc = 2;     % No of controlled signals
nym = 3;     % No of measured outputs
nd = 1;      % No of input disturbances (dummy)
Yind = [2,3,6];
```

```
Uind = [2,3];

% system constraints
u_max = [taug_nom;beta_max];
u_min = [0;beta_min];
du_max = [100;betadot_max];
du_min = [-100;betadot_min];

% Scaling Factors
Qe_wg = 1e-2;
Qe_pg = 1e-7;

par_dim =
struct('Ts',Ts,'nx0',nx0,'nu',nu,'nyc',nyc,'nym',nym,'nd',nd,...

'Yind',Yind,'Uind',Uind,'u_max',u_max,'u_min',u_min,'du_max',du_max,..
.
    'du_min',du_min);
par_wt = catstruct(par_wt,par_dim);

s = tf('s');
nint = 0.99999;         % (near)-integrator pole for controller design
zi = filt([0 1],1,Ts);

%% PID
Kp_wg = 30*Qe_wg; Ki_wg = 10*Qe_wg; Kd_wg = 0*Qe_wg; % wg control
% Kp_pg = 40*Qe_pg; Ki_pg = 30*Qe_pg; Kd_pg = 0*Qe_pg; % Pg control
Kp_pg = 100*Qe_pg; Ki_pg = 40*Qe_pg; Kd_pg = 0*Qe_pg; % Pg control

%% NGMVss (state-space)
Wr = eye(nyc)*filt([0 1],[1 -0.9999],Ts);               nxr =
order(Wr);
Wd = 1e0*1*eye(nyc)*filt([0 1],[1 -0.9999],Ts);         nxd =
order(Wd);
Wk = eye(nyc)*filt(1,1,Ts);
% Wk = wt_lind(6,3);
k = 1;              % common time delay in samples

QN = 1*diag(ones(2*nyc,1));
RN = 1*diag([1,1]);

% Define NGMV Dynamic weightings ----------------------------------
% PID-based
% Kp_wg_ngmv = 1*Qe_wg; Ki_wg_ngmv = 0.5*Qe_wg; Kd_wg_ngmv = 0*Qe_wg;
% wg control
Kp_wg_ngmv = 100*Qe_wg; Ki_wg_ngmv = 0.5*Qe_wg; Kd_wg_ngmv = 0*Qe_wg;
% wg control
% Kp_wg_ngmv = 100*Qe_wg; Ki_wg_ngmv = 10*Qe_wg; Kd_wg_ngmv = 0*Qe_wg;
% wg control
Kp_pg_ngmv = 700*Qe_pg; Ki_pg_ngmv = 5*Qe_pg; Kd_pg_ngmv = 0*Qe_pg; %
Pg control
% Kp_pg_ngmv = 400*Qe_pg; Ki_pg_ngmv = 10*Qe_pg; Kd_pg_ngmv = 0*Qe_pg;
% Pg control
Pc = [(Kp_wg_ngmv+Ki_wg_ngmv/(1-nint*zi)), 0; 0,
(Kp_pg_ngmv+Ki_pg_ngmv/(1-nint*zi))];
Fck_dyn = [5*ss(1), 0; 0, 6*ss(1)];
```

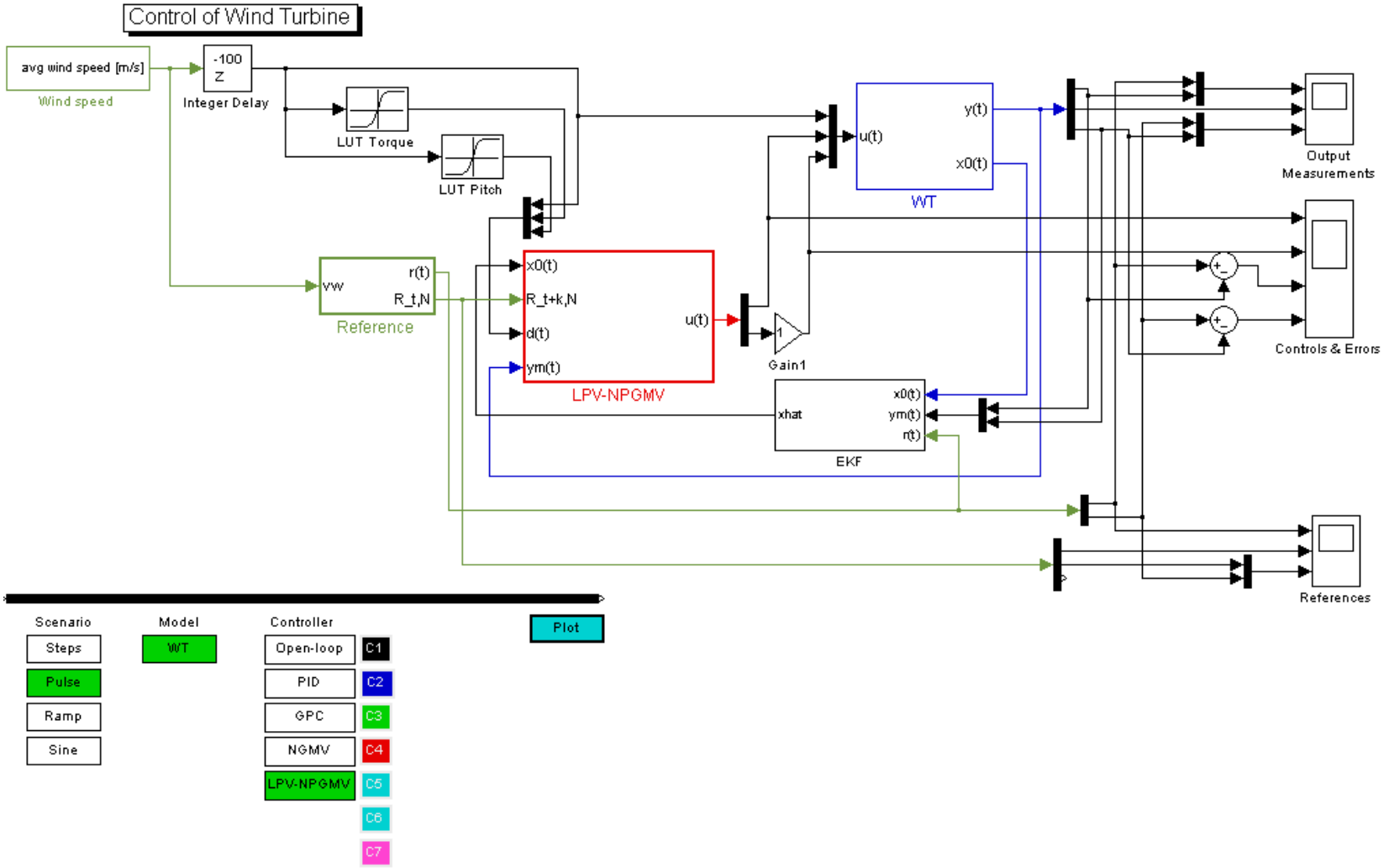
```
nxp = order(Pc);
nx = nxd + nxr + nxp;

[Ap,Bp] = ssdata(Pc);
Pcx = ss(Ap,Bp,eye(nxp),zeros(nxp,nyc),Ts);
xp_0 = zeros(nxp,1);

iFck_dyn = inv(Fck_dyn);
[Af,Bf,Cf,Df] = ssdata(iFck_dyn);

estXp = 'on';
[k_est,css_gain,css_loop] = ngmvss(Wk,k,Wd,Wr,Pc,QN,RN,estXp);
k_est.ts = Ts;
nxi = size(css_gain,2);

% Store parameters in structure
par_ngmv = struct('Ap',Ap,'Bp',Bp,'Pcx',Pcx,...

'k',k,'QN',QN,'RN',RN,'k_est',k_est,'Fck',Fck_dyn,'css_gain',css_gain,
...
    'css_loop',css_loop,'nxp',nxp,'nxi',nxi,'nxd',nxd,'nx',nx,...
    'Af',Af,'Bf',Bf,'Cf',Cf,'Df',Df);
par_ngmv = catstruct(par_ngmv,par_wt);


%% NPGMV-lpv
% Covariances for the Kalman Filter
qn_beta = 0.1;
qn_betadot = 0.01;
qn_theta = 0.01^3;
qn_wg = 0.01^-3;
qn_wr = 0.01;
qn_taug = 0.1^-4;
qn_xt = 1;
qn_xtdot = 0.1;

qn_vw = 0.01;
qn_tauref = 0.01^-4;
qn_xi = 0.01;

% rn_theta = 1;     % theta sensor variance [deg^2]
% rn_wg = 1;        % theta sensor variance [deg^2]

QN = 1*diag([qn_beta qn_betadot qn_theta qn_wg qn_wr qn_taug qn_xt
qn_xtdot...
    qn_vw qn_tauref qn_xi]);
RN = 1*diag([1,1]);

nbuffer = 100;      % state buffer for future reference generation
(max pred. horizon)
% 'u' formulation
c2dflag = 0;         % c2d flag: 0=Euler, 1=direct
constr_flag = 0;     % Constraint handling
freeze_flag = 1;     % 0 = full LPV, 1 = frozen model
futuresp_flag = 1;   % 1 = future reference knowledge ON
futurectr_flag = 0;  % 0 = hold u(t-1), 1 = use U(t-1)
```

```
bta = 0;              % 0='u', 1='delta_u'
Npr = 10;             % prediction horizon
Pu = [1 5; 2 1; 3 1];
 [Tu,Nu] = mpcprof(Pu,Npr,bta);
Kp_wg_npgmv = 50*Qe_wg; Ki_wg_npgmv = 2*Qe_wg; Kd_wg_npgmv = 0*Qe_wg;
Kp_pg_npgmv = 2000*Qe_pg; Ki_pg_npgmv = 80*Qe_pg; Kd_pg_npgmv =
0*Qe_pg;
% Kp_pg_npgmv = 20000*Qe_pg; Ki_pg_npgmv = 1000*Qe_pg; Kd_pg_npgmv =
0*Qe_pg;

L_U = 1*diag([4,10])*diag([1, 1]); % [taug;beta]
% L_U = 1*diag([0.01,1])*diag([1, 1]); % [taug;beta]

% Block-diagonal static control weight
LN_U = kron(eye(Nu),L_U);

% Output disturbances incl. stochastic and "robustness" states
nxd = nym;
Ad = eye(nxd); Dd = eye(nxd); %Dd
Cds = zeros(nxd,nxd);    % stochastic disturbances
Cdm = 0*eye(nxd);        % mismatch on measurements
Cdc = 0;                 % mismatch on controlled variables
Cd = 1*eye(nyc,nxd);     % mismatch on controlled variables

% Integrator switch
nxi = bta*nu;

% Dynamic error weighting
Pc = [(Kp_wg_npgmv+Ki_wg_npgmv/(1-nint*zi)), 0; 0,
(Kp_pg_npgmv+Ki_pg_npgmv/(1-nint*zi))];
[Ae,Be,Ce,Ee] = ssdata(Pc);
nxp = size(Ae,1);
Pcx = ss(Ae,Be,eye(nxp),zeros(nxp,nyc),Ts);
xp_0 = zeros(nxp,1);

% Total number of states
nx = nx0 + nxd + nxi + nxp;


% Separate nonlinear control weighting in linear case into through and
dynamic terms
af = 0.0;
% Fck = 1*filt(1-af,[1 -af],Ts);
Fck = [1*ss(1), 0; 0, 1*ss(1)];
[Af,Bf,Cf,Ef] = ssdata(Fck);
Fck0 = Ef;
Fck1 = ss(Af,Bf,Cf,0,Ts);
nxf = order(Fck1);
Fck_N0 = Fck0;
for i=2:Nu
    Fck_N0 = blkdiag(Fck_N0,Fck0);
end

% Input weighting on u0(t)
a1 = 0.93; b1 = 0.95;
Wu = [(1-a1)/(1-b1)*filt([1 -b1],[1 -a1],Ts),0;0,(1-a1)/(1-b1)*filt([1
-b1],[1 -a1],Ts)];
```

```
[Au,Bu,Cu,Eu] = ssdata(Wu);
nxu = size(Au,1);
Wux = ss(Au,Bu,eye(nxu),zeros(nxu,nu),Ts);
xu_0 = zeros(nxu,0);

% Definition of Model and Control structures
par_npgmv = struct('nxd',nxd,'nxp',nxp,'nx',nx,'nxi',nxi,...

'freeze_flag',freeze_flag,'constr_flag',constr_flag,'c2dflag',c2dflag,
...
    'futuresp_flag',futuresp_flag,'futurectr_flag',futurectr_flag,...
    'Npred',Npr,'Nu',Nu,'Tu',Tu,'QN',QN,'RN',RN,'LN_U',LN_U,...
    'Af',Af,'Bf',Bf,'Cf',Cf,'Ef',Ef,'nxf',nxf,'Fck_N0',Fck_N0,...
    'Ae',Ae,'Be',Be,'Ce',Ce,'Ee',Ee,...
    'Ad',Ad,'Dd',Dd,'Cd',Cd,'Cdm',Cdm,...
    'Au',Au,'Bu',Bu,'Cu',Cu,'Eu',Eu,'nxu',nxu,...
    'k',k,'bta',bta);

par_npgmv = catstruct(par_npgmv,par_wt);




%% Simulink model initialization
open_system('simWT_fr')

% Default simulation scenario
default_scenario = 'Pulse';
tmp_button = ['button_' default_scenario];
hs = find_system('simWT_fr','Tag','button_scenario');
for i=1:length(hs),set_param(hs{i},'BackgroundColor','white'); end
set_param(['simWT_fr/' tmp_button],'BackgroundColor','green')
h = find_system('simWT_fr','Name',tmp_button);
cb_button(h{1})

% Default controller
default_controller = 'PID';
tmp_button = ['button_' default_controller];
hs = find_system('simWT_fr','Tag','button_controller');
for i=1:length(hs),set_param(hs{i},'BackgroundColor','white'); end
set_param(['simWT_fr/' tmp_button],'BackgroundColor','green')
h = find_system('simWT_fr','Name',tmp_button);
cb_button(h{1})

% Default model
default_model = 'WT';
tmp_button = ['button_' default_model];
hs = find_system('simWT_fr','Tag','button_model');
for i=1:length(hs),set_param(hs{i},'BackgroundColor','white'); end
set_param(['simWT_fr/' tmp_button],'BackgroundColor','green')
h = find_system('simWT_fr','Name',tmp_button);
cb_button(h{1})

% Colors
col1='k'; col2='b'; col3=rgb('Green'); col4=rgb('FireBrick');
col5=rgb(23);
```

```
col6 = rgb(109); col7 = 'm';
set_param('simWT_fr/button_C1','BackgroundColor','black')
set_param('simWT_fr/button_C2','BackgroundColor','blue')
set_param('simWT_fr/button_C3','BackgroundColor','green')
set_param('simWT_fr/button_C4','BackgroundColor','red')
set_param('simWT_fr/button_C5','BackgroundColor','gray')
set_param('simWT_fr/button_C6','BackgroundColor','cyan')
set_param('simWT_fr/button_C7','BackgroundColor','magenta')
```

Main Wind Turbine Controllers Simulation Diagram (SIMULINK):



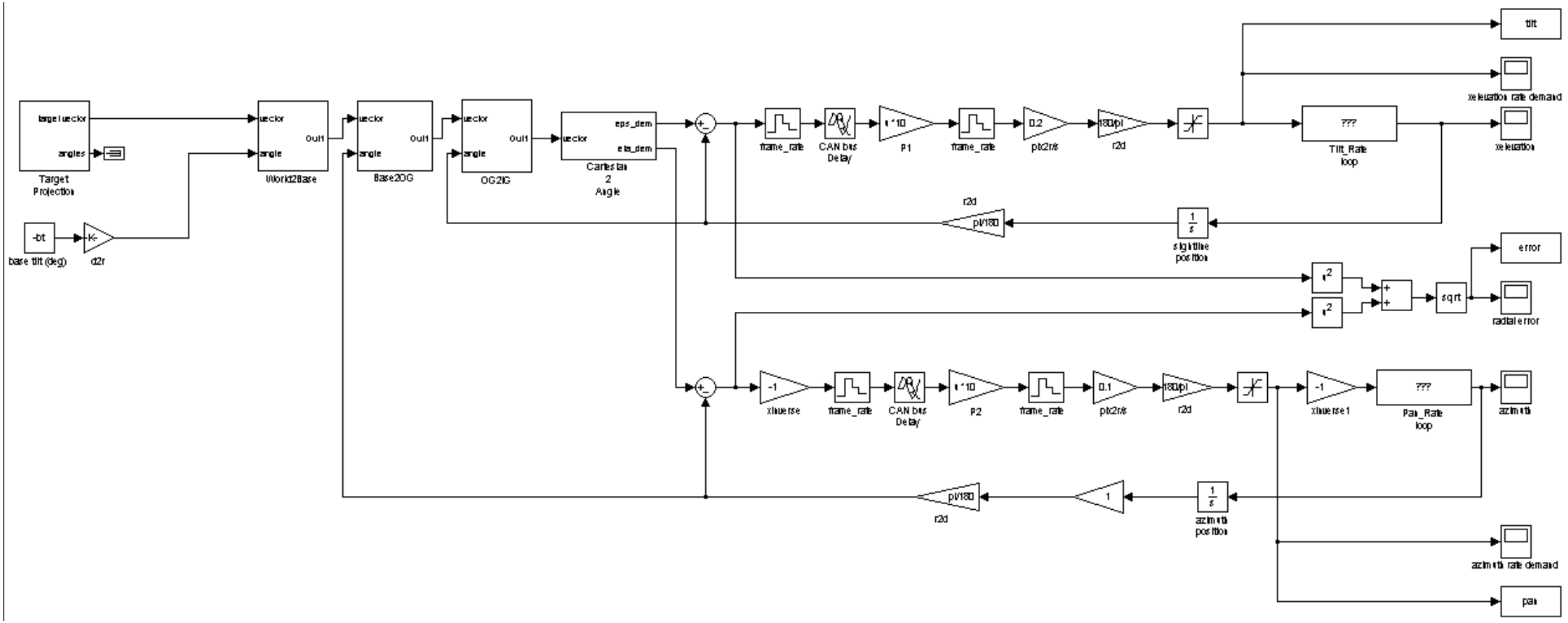
Appendix A.3 Application Chapter 7 Simulations Code.

System Model Validation File Code (MATLAB):

```
%% Tiger Eye electro-optical gyroscopic system validation
% compares the radial error, the tilt & pan control actions of the
gimbal mechanism, to the simulation
% results in various base angles

% identified rate loops
Ts = 1/30;                                          % nominal sampling
time [s]

p1 = -10; p2 = -5; K = (p1*p2);                     %pan rate loop model
from identification
Pan_Rate = zpk([],[p1 p2],K);                       %& discretization
PRd = c2d(Pan_Rate,1/30,'zoh');

p1 = -10; p2 = -5; K = (p1*p2);                     %tilt rate loop model
from identification
Tilt_Rate = zpk([],[p1 p2],K);                      %& discretization
TRd = c2d(Tilt_Rate,1/30,'zoh');

bt=0;                                               %nominal base tilt
r0=1.9;                                             %nominal distance from
screen
r1=.12;                                             %vertical distance
from centre

open_system('replica2.mdl');
sim('replica2.mdl')

n=length(error.signals.values);

e=zeros(n,7); t=zeros(n,7); p=zeros(n,7);
etn=zeros(n,7); ttn=zeros(n,7); ptn=zeros(n,7);
e(:,1)=error.signals.values; t(:,1)=tilt.signals.values;
p(:,1)=pan.signals.values;

bta=[0 25 45 60 70 80 90];                          %base tilt angles


for i=1:length(bta)
    bt=bta(i);

    et=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\raderror' num2str(bt) '.dat']);
    tt=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\tilt' num2str(bt) '.dat']);
    pt=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\pan' num2str(bt) '.dat']);
    etn(1:n,i)=et(1:n); ttn(1:n,i)=tt(1:n); ptn(1:n,i)=pt(1:n);
```

```
    sim('replica2.mdl')
    e(:,i)=error.signals.values; t(:,i)=tilt.signals.values;
p(:,i)=pan.signals.values;
end

a=4;

hold on
plot(e(:,a))
plot(etn(:,a),'g')
title('radial error')
grid

figure
hold on
plot(t(:,a))
plot(ttn(:,a),'g')
title('tilt')
grid

figure
hold on
plot(p(:,a))
plot(ptn(:,a),'g')
title('pan')
hold off
grid
```

Main Simulation Diagram (SIMULINK):



NGMV Control Design Master File Code (MATLAB):

```
%% tiger eye + NGMV
% employs the Nonlinear Generalized Minimum Variance scheme

pinit; gensym z^-1;
clc

%% plant definition
% identified rate loops

Ts = 1/30;                                          % nominal sampling
time [s]
Nu=2; Ny=2;                                         % 2x2 system
k = 1;                                              % plant time-delay
(for the later controller formulation)

p1 = -10; p2 = -5; K = (p1*p2);                     %pan rate loop model
from identification
Pan_Rate = zpk([],[p1 p2],K);                       %& discretization
PRd = c2d(Pan_Rate,1/30,'zoh');

p1 = -10; p2 = -5; K = (p1*p2);                     %tilt rate loop model
from identification
Tilt_Rate = zpk([],[p1 p2],K);                      %& discretization
TRd = c2d(Tilt_Rate,1/30,'zoh');

%Linear part definition
sys=PRd*eye(2);

W0k = ss(sys);
W0 = sys; set(W0,'OutputDelay',[k k])

%% disturbance definition
Nn = 0.01*pol([2 1; 1 2]);
Nd = [1-0.9*z^-1 1-0.3*z^-1; 1-0.4*z^-1 1-0.5*z^-1];          % Add one
delay to the transfer function - doesn't change the disturbance
Wd = rat2lti(z^-1*Nn,Nd);                                     %
spectrum and makes the system strictly proper

%% reference definition
Rn = pol(zeros(2));
Rd = pol(ones(2));
Wr = rat2lti(z^-1*Rn,Rd);

%% initial PID controller
Kp=1; Ki=1; Kd=0; Tau=0.5; nint=0.9999;

Co=dpid2tf(Kp,Ki,Kd,Ts,Tau,nint)*eye(2);

%% NGMV weightings definition
% PID-based design:     Pc = Ko1, Fc = -1
Pc = 1*ss(Co);
Fck = -1*filt(1,[1 -1],Ts)*eye(2);
```

```
%% NGMV controller design
% k_est is the 'full' Kalman filter for x0, xd, xr and xp
QN = eye(2*Ny);                         % model noise covariance for
Kalman filter
RN = diag([0.0 0.0]);                   % measurement noise covariance
[k_est,c_gain,c_loop2,Pcx] = ngmvss(W0k,k,Wd,Wr,Pc,QN,RN);


%% simulation
bt=0;                                           %nominal base tilt
r0=1;                                          %nominal distance from
screen
r1=.12;                                         %vertical distance
from centre

open_system('replicaNGMV.mdl');
sim('replicaNGMV.mdl')

n=length(error.signals.values);

e=zeros(n,7); t=zeros(n,7); p=zeros(n,7);
etn=zeros(n,7); ttn=zeros(n,7); ptn=zeros(n,7);
e(:,1)=error.signals.values; t(:,1)=tilt.signals.values;
p(:,1)=pan.signals.values;

bta=[0 25 45 60 70 80 90];                      %base tilt angles


for i=1:length(bta)

    bt=bta(i);

    et=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\raderror' num2str(bt) '.dat']);
    tt=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\tilt' num2str(bt) '.dat']);
    pt=load (['C:\Users\...\Documents\MATLAB\Gimbal Control\Model
Validation\measurements003\pan' num2str(bt) '.dat']);
    etn(1:n,i)=et(1:n); ttn(1:n,i)=tt(1:n); ptn(1:n,i)=pt(1:n);

    sim('replicaNGMV.mdl')

    e(:,i)=error.signals.values; t(:,i)=tilt.signals.values;
p(:,i)=pan.signals.values;


end

a=7;

hold on
plot(300*e(:,a)+13)
plot(etn(:,a),'g')
title('radial error')
grid
```

```
figure
hold on
plot(t(:,a))
plot(ttn(:,a),'g')
title('tilt')
grid

figure
hold on
plot(p(:,a))
plot(ptn(:,a),'g')
title('pan')
hold off
grid
```

Main NGMV Control Simulation Diagram (SIMULINK):