Model predictive control of a magnetically suspended flywheel energy storage system

---

A dissertation presented to

The school of Electric, Electronic and Computer Engineering

North West University

In partial fulfilment of the requirements for the degree

Magister Ingeneriae

In Computer and Electronic Engineering

by

Christiaan Daniël Aucamp

Supervisor: Prof. G. van Schoor

Co-supervisor: Dr. K. Uren

Date: November 2012

Potchefstroom campus

Abstract

The goal of this dissertation is to evaluate the effectiveness of model predictive control (MPC) for a magnetically suspended flywheel energy storage uninterruptible power supply (FlyUPS). The reason this research topic was selected was to determine if an advanced control technique such as MPC could perform better than a classical control approach such as decentralised Proportional-plus-Differential (PD) control.

Based on a literature study of the FlyUPS system and the MPC strategies available, two MPC strategies were used to design two possible MPC controllers were designed for the FlyUPS, namely a classical MPC algorithm that incorporates optimisation techniques and the MPC algorithm used in the MATLAB® *MPC toolbox*™. In order to take the restrictions of the system into consideration, the model used to derive the controllers was reduced to an order of ten according to the Hankel singular value decomposition of the model.

Simulation results indicated that the first controller based on a classical MPC algorithm and optimisation techniques was not verified as a viable control strategy to be implemented on the physical FlyUPS system due to difficulties obtaining the desired response. The second controller derived using the MATLAB® *MPC toolbox*™ was verified to be a viable control strategy for the FlyUPS by delivering good performance in simulation.

The verified MPC controller was then implemented on the FlyUPS. This implementation was then analysed in order to validate that the controller operates as expected through a comparison of the simulation and implementation results. Further analysis was then done by comparing the performance of MPC with decentralised PD control in order to determine the advantages and limitations of using MPC on the FlyUPS.

The advantages indicated by the evaluation include the simplicity of the design of the controller that follows directly from the specifications of the system and the dynamics of the system, and the good performance of the controller within the parameters of the controller design. The limitations identified during this evaluation include the high computational load that requires a relatively long execution time, and the inability of the MPC controller to adapt to unmodelled system dynamics.

Based on this evaluation MPC can be seen as a viable control strategy for the FlyUPS, however more research is needed to optimise the MPC approach to yield significant advantages over other control techniques such as decentralised PD control.

Keywords: *Model predictive control (MPC)*, *active magnetic bearings (AMBs)*, *flywheel energy storage uninterruptible power supply (FlyUPS)*, *decentralised Proportional-plus-Differential (PD) control*.

Acknowledgements

I would like to acknowledge the following people and institutions who have contributed to this project in some way:

- My supervisors, Prof. Van Schoor and Dr. Uren, for their advice and guidance, and the occasional words of motivation.
- Gert Kruger, Jaco Bean, Dr. Pieter van Vuuren, and the rest of the McTronX group for the help and advice needed to solve problems; and for the interesting conversations during coffee breaks.
- THRIP and Delphius for the financial support during my studies. I would also like to thank the National Research Foundation (NRF) for their contribution.
- Finally, I would like to thank my parents, Chris and Elsa, my sister Rena, and the rest of the Aucamp clan for supporting me at every step of this journey.

Table of contents

Abstract i

Acknowledgements iii

Table of contents iv

List of figures viii

List of Tables x

Nomenclature xi

List of abbreviations xi

List of symbols xii

1. Introduction 1

1.1 Background 1

1.2 Problem statement 3

1.3 Issues to be addressed 3

1.4 Research methodology 4

1.4.1 System model 4

1.4.2 MPC theory and design 4

1.4.3 Controller simulation 4

1.4.4 Implementation of MPC Algorithm 4

1.4.5 Evaluation 5

1.5 Verification and validation 5

1.6 Dissertation layout 5

1.7 Conclusion 6

2. Literature study 7

2.1 System overview 7

2.2 Control strategy: model predictive control 8

2.2.1 Model predictive control as an advanced control strategy 8

2.2.2 Basics of model predictive control 9

2.2.3 Advantages of MPC 13

2.2.4 MPC for flywheel energy storage systems 13

2.3 Active magnetic bearings 14
2.4 Flywheel uninterruptible power supply 16
2.5 System model 17
2.5.1 Linearization of AMB model 19
2.5.2 Radial AMB model 20
2.5.3 Axial AMB model 20
2.5.4 Complete FlyUPS model 21
2.5.5 Model order reduction 22
2.6 Critical literature overview 23
2.7 Conclusion 24
3. Controller design 25
3.1 MPC concept and terminology 25
3.2 Classic MPC controller design 27
3.2.1 Formulation of the model 27
3.2.2 Formulation of predictive control 28
3.2.3 Observers in predictive control design 31
3.2.4 Constraints in the MPC controller design 32
3.3 Optimisation techniques in MPC design 33
3.3.1 Laguerre functions in MPC design 34
3.3.2 Prescribed degree of stability and exponential data weighting 38
3.4 MATLAB® Model Predictive Control toolbox™ 39
3.4.1 Models used to develop the MPC controller 39
3.4.2 Observers in the design of the MPC controller 40
3.4.3 The objective function of the MPC controller 41
3.4.4 Predictive control design 41
3.5 Conclusion 43
4. Controller simulation 44
4.1 Classical MPC using optimization techniques 44
4.1.1 Simulation setup 44

4.1.2 Results for simulation of MPC using optimization techniques 48
4.1.3 Discussion of results for MPC using optimization techniques 49
4.2 MPC using the MATLAB® Model Predictive Control toolbox™ 50
4.2.1 Simulation setup 50
4.2.2 Results for simulation using the MATLAB® MPC toolbox™ 54
4.2.3 Discussion of simulation using the MATLAB® MPC toolbox™ 60
4.3 Conclusion 61
5. Controller implementation 63
5.1 Implementation method 63
5.2 Implementation setup 64
5.3 Benchmark: Decentralised PD control 65
5.4 Considerations 66
5.4.1 Memory usage 66
5.4.2 Execution time 67
5.4.3 Operational limitations 69
5.5 Conclusion 70
6. Controller evaluation 71
6.1 Performance assessment 71
6.1.1 Transient response evaluation 71
6.1.2 Disturbance response evaluation 72
6.1.3 Sensitivity function 72
6.2 Validation 73
6.2.1 Transient response 73
6.2.2 Disturbance response 76
6.2.3 Sensitivity function 79
6.2.4 MPC operating range 83
6.2.5 Discussion 83
6.3 Comparison of MPC and PD 83
6.3.1 Transient response 84

6.3.2 Disturbance response 86
6.3.3 Sensitivity function 88
6.3.4 Discussion 90
6.4 Conclusion 91
7. Conclusions and recommendations 92
7.1 Conclusions 92
7.2 Recommendations for future work 94
7.2.1 System assessment 94
7.2.2 MPC algorithm 95
7.2.3 Control specification 96
7.3 Closure 96
References 98
Appendix A: Program code for MPC using optimisation techniques 100
A.1. MPC main program 100
A.2. Function: *dmpc* 101
A.3. Function: *lagd* 102
A.4. Function: *simucon* 102
Appendix B: Data CD 104
Declaration 105

List of figures

Figure 1-1 Detail representation of the FlyUPS [2] 2
Figure 2-1 Basic overview of the system 7
Figure 2-2 MPC principle [5] 10
Figure 2-3 Basic representation of MPC strategy [5,6,9,12] 10
Figure 2-4 Basic operation of an AMB 15
Figure 2-5 Sketch of FlyUPS indicating six degrees of freedom 18
Figure 2-6 Flow diagram of linearised open-loop AMB [4] 19
Figure 2-7 FlyUPS model including controller 22
Figure 3-1 Typical MPC scheme configuration [9] 25
Figure 3-2 MPC operating principle [5,6,9] 26
Figure 4-1 Program flowchart for MPC using optimization techniques 45
Figure 4-2 Hankel singular values of axial AMB model 46
Figure 4-3 Results for simulation of MPC using optimization techniques 49
Figure 4-4 Hankel singular value decomposition of FlyUPS model 51
Figure 4-5 Simulink model for simulation of MPC developed using MATLAB® *MPC toolbox*™ .. 53
Figure 4-6 Physical characteristics of the FlyUPS [2] 57
Figure 4-7 Simulation results of axial AMB of FlyUPS 58
Figure 4-8 Simulated response of radial AMBs of FlyUPS 59
Figure 4-9 Simulated disturbance response of radial AMBs 60
Figure 4-10 Influence of delays on the system response 60
Figure 5-1 Implementation method of MPC on FlyUPS 64
Figure 5-2 Simulink™ model for 5-DOF control of FlyUPS 65
Figure 6-1 Signals and excitation nodes of a common closed-loop system [7] 71
Figure 6-2 Differential driving mode for linearised AMB 72
Figure 6-3 Step response of the axial AMB at standstill 74
Figure 6-4 step response of the radial AMBs at 0 r/min for a step on *X1* 74
Figure 6-5 Step response of the radial AMBs at 0 r/min for a step on *X2* 75
Figure 6-6 Step response of the axial AMB at 500 r/min 75
Figure 6-7 Step response of the radial AMBs at 500 r/min for a step on *X1* 76
Figure 6-8 Step response of the radial AMBs at 500 r/min for a step on *X2* 76
Figure 6-9 Disturbance response of the axial AMB at standstill 77
Figure 6-10 Disturbance response of the radial AMBs at 0 r/min for a disturbance on *X1* 77
Figure 6-11 Disturbance response of the radial AMBs at 0 r/min for a disturbance on *X2* 78
Figure 6-12 Disturbance response of the axial AMB at 500 r/min 78
Figure 6-13 Disturbance response of the radial AMBs at 500 r/min for a disturbance on *X1* 79

Figure 6-14 Disturbance response of the radial AMBs at 500 r/min for a disturbance on *X2*..... 79
Figure 6-15 Sensitivity function of the axial AMB at standstill 80
Figure 6-16 Sensitivity function of the top radial AMB at standstill 81
Figure 6-17 Sensitivity function of the bottom radial AMB at standstill 81
Figure 6-18 Sensitivity function of the axial AMB at 500 r/min 82
Figure 6-19 Sensitivity function of the top radial AMBs at 500 r/min 82
Figure 6-20 Sensitivity function of the bottom radial AMBs at 500 r/min 82
Figure 6-21 Step response of the axial AMB at standstill..... 84
Figure 6-22 Step response of the radial AMBs at 0 r/min for a step on *X1*..... 85
Figure 6-23 Step response of the radial AMBs at 0 r/min for a step on *X2*..... 85
Figure 6-24 Step response of the radial AMBs at 500 r/min for a step on *X2*..... 86
Figure 6-25 Radial AMB orbital patterns at 500 r/min 87
Figure 6-26 Disturbance response of the axial AMB at standstill 87
Figure 6-27 Sensitivity function of *Z* at standstill..... 88
Figure 6-28 Sensitivity function of *X1* at standstill..... 89
Figure 6-29 Sensitivity function of *X2* at standstill..... 89
Figure 6-30 Sensitivity function of *X1* at 500 r/min..... 90

List of Tables

Table 4-1 Parameter values used in the simulation of MPC using optimization techniques 47
Table 4-2 MATLAB® *MPC toolbox*™ parameters 50
Table 4-3 Simulation results for step response on axial AMB 54
Table 4-4 Simulation results for control effort of MPC for varying N_c and N_p 55
Table 4-5 Results of step response for varying $W_{\Delta u}$ 56
Table 4-6 Results for step response with varying W_y 56
Table 4-7 Parameter values for MPC controller developed with the *MPC toolbox*™ 58
Table 4-8 Response characteristics of coupling between radial AMBs 59
Table 5-1 PD control experiment parameters 66
Table 5-2 Controller size for different combinations of PD and MPC 67
Table 5-3 Execution time for varying N_p 69
Table 5-4 Execution time for varying N_c 69
Table 6-1 Sensitivity ratings according to the ISO CD 14839-3 standard [7] 73

Nomenclature

List of abbreviations

FESS	Flywheel energy storage system
FlyUPS	Flywheel uninterruptible power supply
AMB	Active magnetic bearing
MPC	Model predictive control
DOF	Degrees-of-freedom
NWU	North-West University
PMSM	Permanent magnet synchronous machine
ISO	International Standards Organisation
MIMO	Multiple-input-multiple-output
SISO	Single-input-single-output
GUI	Graphical user interface
PID	Proportional-plus-Integral-plus-Differential
PI	Proportional-plus-Integral
PD	Proportional-plus-Differential
PC	Personal computer
FIR	Finite impulse response
DARMAX	Directional auto-regressive moving-average with exogenous input
COG	Centre-of-gravity
LQG	Linear-quadratic Gaussian
AC	Alternating current
PA	Power amplifier
SVD	Singular value decomposition
SV	Singular value
LTI	Linear time-invariant
P.O.	Percentage overshoot
PPC	Power PC
I/O	Input-output
ADC	Analogue-to-digital-converter
DAC	Digital-to-analogue-converter
PWM	Pulse-width modulation

List of symbols

F_{mag}	Magnetic force
k_e	Equivalency constant
$i_{control}$	Control current
i_0, i_{bias}	Bias current
x_0	Operating point
k_i	Force-current constant
k_s	Force-distance constant
m	Mass
u	Control signal
Δu	Increment of control signal
y	Output signal
v	Measured disturbance signal
r	Set-point signal
d	Unmeasured disturbance signal
$\bar{y}$	Actual output signal
z	Measurement noise signal
N_c	Control horizon
N_p	Prediction horizon
k	Sampling instant
A,B,C	State-space model matrices
ω	Plant noise and disturbance
$\rho(\lambda)$	Eigenvalues of the system
ΔU	Future control trajectories
Y	Predicted output variables
J	Cost function
r_w	Future control trajectory tuning weight
F	Variable related to observability
Φ	Variable related to controllability
K_{mpc}	State feedback control gain vector
K_y	Feedback gain vector
x	State variable
Δx	Increment of state variable
$\hat{x}$	Estimated state variable
$\tilde{x}$	State estimation error
K_{ob}	Observer gain

X^{min}, X_{min}	Minimum limit of X
X^{max}, X_{max}	Maximum limit of X
M	Parameter constraints
γ_c	Constraint values
$\Gamma(z), l(k)$	Laguerre network parameter
a	Scaling factor
N	Number of Laguerre parameters
c_i	Laguerre coefficients
$\phi(m)$	Data matrix for Laguerre network
η	Parameter vector for Laguerre network
Q, R	Performance parameters for LQG control
Ω	Past information of system
Ψ	Present information of system
λ	Prescribed degree of stability
α	Exponential data weight
γ	Ratio between λ and α
W_y	MPC output weight
$W_{\Delta u}$	MPC increment on control weight
W_u	MPC control weight
$K_{\Delta u}$	Hessian matrix
f_s	Sampling frequency
T	Control interval
T_s	Settling time
T_r	Rise time
ζ	Damping ratio
ω_n	Natural frequency
τ	Time constant
k_{eq}	Equivalent stiffness
b_{eq}	Equivalent damping
$K_{out.dist}$	Output disturbance gain
Z	Axial AMB
$X1$	Top radial AMB
$X2$	Bottom radial AMB
K_p	Proportional gain
K_D	Differential gain

1

Chapter

Introduction

This chapter gives a brief introduction to the dissertation as well as a motivation for the study. Firstly, the main elements of the project are introduced and discussed. Then the problem statement is given, followed by the issues that must be addressed as well as the method used to resolve these issues. Finally, the layout of the dissertation is discussed.

1.1 Background

This study is based on the control of the flywheel energy storage system (FESS) developed by the McTronX research group in the Faculty of Engineering at the North-West University. This system was dubbed the flywheel uninterruptible power supply, or FlyUPS for short, and uses active magnetic bearings (AMBs) to suspend the flywheel in five degrees of freedom.

The major reason for the development of the FlyUPS was to improve the knowledge base of AMBs within the McTronX research group. By developing the FlyUPS completely in-house many valuable research opportunities were created that enabled research on the development and optimisation of AMB systems, as well as the evaluation and optimisation of various control strategies. The development of the FlyUPS is thoroughly documented in [1] and [2].

A detail representation of the FlyUPS is given in Figure 1-1. This figure shows the flywheel, permanent magnet synchronous machine (PMSM), Eddy probe sensors and the radial and axial AMBs. The wire diagram around the rotor indicates the vacuum enclosure which consists of several modules to allow the system to be taken apart easily in order to inspect critical parts such as the rotor and the AMBs.

The FlyUPS operates on the basic principle that the FlyUPS is connected between an electrical source and a load. When the source is active, electrical energy is supplied to both the load and the FlyUPS. The FlyUPS stores this electrical energy as kinetic energy in the flywheel. If at any point in time there are dips or sags in the electrical energy supplied by the source, the energy necessary for the operation of the load is then extracted from the FlyUPS. In order to ensure the

efficiency of the FlyUPS the losses of kinetic energy must be restricted to a minimum. For this reason AMBs and a vacuum enclosure are used in the design [3].

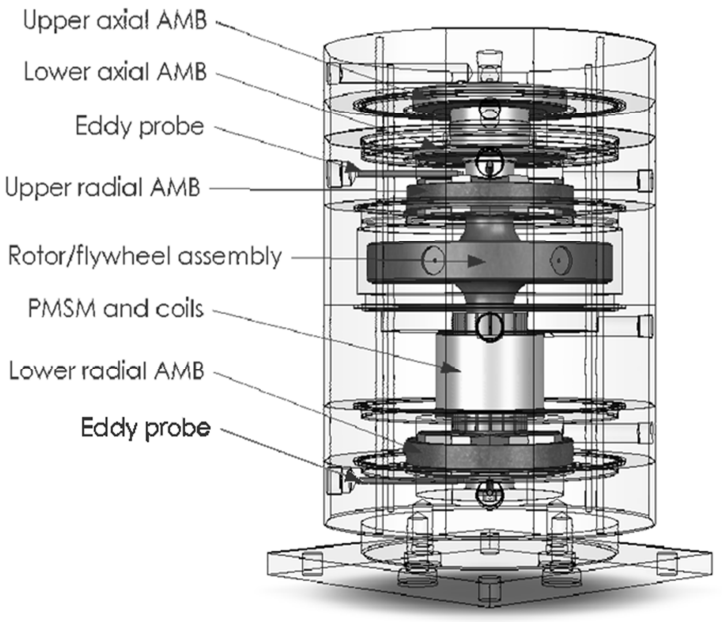

Figure 1-1 Detail representation of the FlyUPS [2]

AMBs offer many advantages over conventional roller element bearings due to the fact that there is no contact between the rotor and stator. The most important advantage for the FlyUPS project is that there is no friction which means that there are far fewer losses. However, AMBs are inherently non-linear which means that complex control strategies are often necessary to control the AMB system optimally [4]. Also, as the FlyUPS has various AMBs a multiple-input-multiple-output (MIMO) control strategy is necessary to ensure the optimal operation of the system.

Figure 1-1 also illustrates the layout of the axial and radial AMBs. This figure shows the axial AMBs controlling the position of the rotor in the z-axis, and the upper and lower radial AMBs controlling the position of the rotor in the x- and y-axes. The PMSM is used to rotate the flywheel rotor about the z-axis, which may cause a complex rotordynamic problem in the x- and y-axes as a disturbance, e.g. a rotor unbalance, may lead to limit cycles, the amplification of external excitations, or even the instability of the system [4]. Also, as the FlyUPS is designed to only operate within the first two rigid modes, a strong cross-coupling exists between the lower and upper radial AMBs [1].

Model predictive control (MPC) is evaluated in this dissertation to determine if MPC is a viable control strategy for such a complex system. MPC is an advanced control technique that uses a dynamic model of the system to make predictions of what the control trajectory of a system

must be in order to reach a set-point value [5,6]. The performance of the controller is also improved by using a cost function to reduce the error of the system, and updating the states of the system during operation by using an estimator in the design of the controller [5].

Some key features of MPC that make it a potential control strategy for the FlyUPS, are that the design of the controller can easily be extended from a single-input-single-output (SISO) controller to a MIMO controller, and that constraints can be incorporated into the design of the controller.

However, in order for MPC to operate optimally various operations such the estimation of states, the prediction of future control signals and the optimisation of the control signals result in a large computational load. Also, as MPC was developed in the process industry where slower processes with larger time constants are usually encountered, another challenge will be to implement real-time control with MPC on the FlyUPS where fast time constants are required for optimal control [5,6].

1.2 Problem statement

The aim of this dissertation is to evaluate the effectiveness of MPC when it is implemented on the FlyUPS where the flywheel is magnetically suspended by AMBs. The results of this project will indicate whether MPC is able to control a complex system such as a high speed FESS.

MPC will be seen as a successful control strategy if real-time control can be implemented while ensuring the stability of the system. The stability of the system must also be kept within zones A and B of the ISO CD 14839-3 standard on magnetic bearings which states that the performance is not expected to deteriorate drastically due to changes in nominal parameters during the machine operation [7].

The non-linearity present in the AMBs and the complexity of implementing MIMO real-time control are some of the complications present in this project. Unmodelled disturbances can also create complications due to the dependency of the MPC controller on an accurate model of the system.

1.3 Issues to be addressed

The following issues are addressed in the dissertation:

- **System model:** A suitable model of the FlyUPS is obtained to develop the MPC controller.

- **MPC theory and design:** Control strategies present in MPC are investigated in order to design the most effective controller.
- **Controller simulation:** Simulations of the designed controller are done in order to theoretically verify that the MPC controller is able to control the FlyUPS.
- **Controller implementation:** After obtaining a controller that works well in simulation, the controller is implemented on the FlyUPS in order to perform an experimental analysis.
- **Controller evaluation:** From the experimental analysis the MPC controller is evaluated in order to validate whether the MPC controller addresses the problem statement.

1.4 Research methodology

This section briefly describes the methodology that is followed in order to resolve the issues discussed in the previous section.

1.4.1 System model

An existing model of the system as derived by the author of [8] is used. This is a linear, continuous-time state-space model of the system and includes the power amplifiers, a rotor/actuator model, the sensors and the anti-aliasing filters. This model has already been verified, and used in the development of an H-infinity (H_∞) controller. This H_∞-controller was successfully implemented on the FlyUPS [8].

1.4.2 MPC theory and design

A literature study is done to obtain a detailed overview of all of the theory applicable to understand every part in the design of the MPC controller. This study focuses on MPC, but sufficient information is also presented regarding other aspects of the dissertation such as AMBs and the FlyUPS.

1.4.3 Controller simulation

From the literature study, system model and controller specification the MPC controller for the FlyUPS is designed. The controller is then implemented in a software simulation package such as MATLAB® or Simulink™ in order to perform a theoretical analysis. The results of this analysis verify that the MPC controller is theoretically able to control the FlyUPS.

1.4.4 Implementation of MPC Algorithm

Following the theoretical analysis, the MPC controller is implemented on the FlyUPS. This implementation is done in MATLAB® and Simulink™, from which it is linked to the ControlDesk®

graphical user interface (GUI). These software instructions are then executed on the FlyUPS using a dSPACE® controller. An experimental analysis of this implementation is then done.

1.4.5 Evaluation

The MPC controller is evaluated according to the results of the experimental analysis. This analysis includes the transient response, disturbance response and sensitivity function of the FlyUPS. The first step is to validate that the MPC controller functions as expected by comparing the simulated and measured responses of the FlyUPS. Then the response of the MPC is compared to decentralised Proportional-plus-Differential (PD) control to investigate any advantages or disadvantages of using MPC on the FlyUPS. The MPC controller is also validated according to the problem statement of this dissertation. Based on these evaluations, a conclusion is then made on the effectiveness of MPC for a complex system such as the FlyUPS.

1.5 Verification and validation

For the purpose of this research topic the verification and validation of MPC is done on the FlyUPS. Simulations are used to verify that the MPC controller is able to control the FlyUPS effectively and efficiently. The MPC controller is only implemented on the FlyUPS when it is verified. This implementation of the MPC controller is then validated by comparing the evaluation results of the implementation and the simulation in order to ensure that the MPC controller operates as expected. A further part of the validation is to evaluate if the MPC controller addresses the problem statement for this dissertation.

1.6 Dissertation layout

This section describes the layout of this dissertation by giving the title of each chapter and a brief discussion of the contents of that chapter.

Chapter 2: Literature study

This chapter gives a detailed study of all of the necessary theory to successfully complete this research project. The literature study contains the theory, trends and techniques that are currently available for MPC, AMBs and the FlyUPS. This chapter also gives a description of the model of the FlyUPS that is used to derive the MPC controller.

Chapter 3: Controller design

This chapter discusses the design of the MPC controller. Two control algorithms used for MPC are considered, namely the use of a classical MPC algorithm using Laguerre functions and

optimisation techniques [6], and the control strategy used in the MATLAB® *Model Predictive Control toolbox*™ [9].

Chapter 4: Controller simulation

In this chapter the two controllers designed in the previous chapter are implemented in simulations. This chapter presents the method that is followed in order to simulate both controllers in MATLAB® and Simulink™, as well as the results of the simulations. These results verify whether the controllers are effective. Only verified controllers are then implemented on the FlyUPS.

Chapter 5: Controller implementation

This chapter gives the necessary steps in the process of implementing the controller on the FlyUPS. This includes a brief background of how the controller is implemented on the system as well as the setup of the MPC controller in Simulink™. The setups of the decentralised PD controllers which are used for comparison are also discussed. Finally, some considerations are discussed based on typical limitations of MPC.

Chapter 6: Controller evaluation

In this chapter the tests used for the evaluation of the MPC and PD controllers are discussed. These tests include the step and disturbance responses of the FlyUPS and the sensitivity function of the FlyUPS according to the ISO CD 14839-3 standard for AMBs according to [4]. The results for each test are then given and used to evaluate the effectiveness of the MPC controller in comparison with the PD controllers. The MPC controller is then validated according to the problem statement of this dissertation.

Chapter 7: Conclusions and recommendations

This chapter draws a conclusion on the effectiveness of MPC on the FlyUPS. Then recommendations are made on how to improve the performance of the system by possible changes in the controller and the system. Finally, a brief paragraph brings closure to this dissertation.

1.7 Conclusion

This chapter gave a brief overview of the elements of this project. This overview serves as an introduction to the implementation of MPC as a control strategy for the FlyUPS. The next chapter discusses the literature elements of the project in more detail.

2

Chapter

Literature study

This chapter discusses the topics introduced in section 1.1 in more detail. This discussion focuses on the literature that pertains to the model predictive control of a magnetically suspended flywheel energy storage system. This literature is divided into the sections of active magnetic bearings, the flywheel uninterruptable power supply, and model predictive control. The model of the FlyUPS from which the MPC controller will be derived, is also discussed. Finally, a critical overview places the relevant literature in context with the topic of this dissertation.

2.1 System overview

An overview of the current system is given in Figure 2-1. This figure indicates the different sections that form part of the system. These parts consist of the user, host personal computer (PC) and the physical FlyUPS. The host PC forms the interface between the user and the physical system and is used by the user to implement the control strategy in MATLAB®/ Simulink™, as well as to monitor the condition of the system through ControlDesk™. As discussed in the previous chapter, the control strategy investigated in this dissertation is MPC.

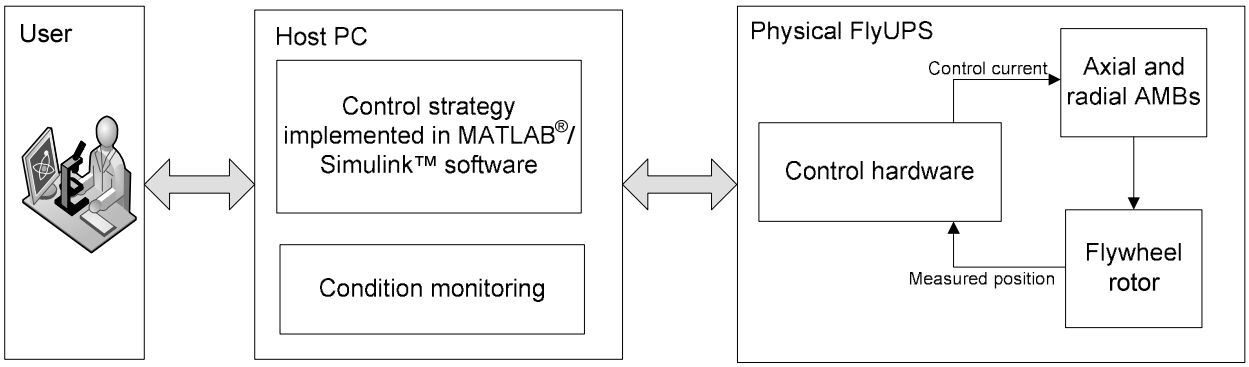

Figure 2-1 Basic overview of the system

The host PC is connected to the physical FlyUPS through a fibre optic bus network. In the physical FlyUPS the most important parts are the control hardware that implements the control strategy on the FlyUPS, the axial and radial AMBs that control the position of the rotor, and the flywheel rotor.

The control hardware contains the dSPACE® controller which determines the control signal for each of the five degrees of freedom based on the position measurement of the rotor. The control hardware also contains the power amplifiers which convert the control signals to control currents and Eddy current sensors that measure the position of the rotor.

Chapter 5 gives a more detailed discussion of the implementation of MPC on the FlyUPS. The concepts that form the basis of this study are MPC, AMBs, the FlyUPS and the model of the FlyUPS. These concepts are discussed in the next sections.

2.2 Control strategy: model predictive control

Model predictive control (MPC) is a control strategy that has been used widely in the process industry since it first appeared in the late 1970's with algorithms such as Model Predictive Heuristic Control [10] which aimed to control systems that were too difficult for conventional proportional-integral-differential (PID) control, and Adaptive Predictive Control [11] which aimed to exploit the adaptive internal model to determine the optimal control signal.

From these beginnings many variations of MPC algorithms were developed, most of which only differ in the model used to represent the process, the cost function to be minimised and the method used to minimise the cost function. This leads to the concept that MPC is not a specific control strategy but rather a wide range of control algorithms with a common nature [12].

This development of MPC was already widely established in the process industry before attracting the attention of academia to truly develop MPC as an advanced control technique [5]. The application of analysis and new ideas by the academia has allowed MPC to be expanded to control engineering problems beyond current industrial practice. The rate of this expansion is further increased by the increase in computing speed and power, as well as the research done on new MPC techniques [5].

The focus for this dissertation is on the general formulation of linear MPC. Some techniques are also discussed that aid in reducing the computational load during optimisation, as well as reducing the time necessary to determine the optimal control.

2.2.1 Model predictive control as an advanced control strategy

As technology progresses and processes become more complex, more advanced control techniques become necessary. Advanced control techniques build on normal PID control but offer much more complex functions. These advanced control techniques include adaptive

control, multivariable control, robust control, neural network or fuzzy-based control, optimal control and model-based control [13].

Advanced control techniques usually have the following features: [13]

- Process modelling and parameter identification.
- Prediction of process behaviour based on a process model.
- Evaluation of process criterion based on process constraints.
- Optimization of process criterion.
- Multivariable control.
- Feedback control.

However, the borders between these control techniques have become blurred as a modern control method, e.g. model predictive control, can contain elements of the other advanced control methods such as adaptive, multivariable, robust and optimal control.

2.2.2 Basics of model predictive control

MPC makes use of a model of the physical system to make predictions of what the output trajectory (or process variable) should be in order to reach a reference at some future time as shown in Figure 2-2. From this information the control trajectory is calculated which will ensure that the system reaches the desired output trajectory. The desired output trajectory is determined by various factors such as error margins, response time required by the process, output accuracy required by the process, or constraints on the physical system [6].

These factors are incorporated in the design of the MPC controller by defining an objective function. The objective function is based on the cost function for the system and the constraints on the system. The cost function is a weighted error function where weights are used to determine the importance of certain factors in the cost function [5]. These factors include the error between the measured output and the set-point, the input, and the rate of change of the input. The optimal control action is determined by minimising this objective function within the moving horizon window [6].

The moving horizon window is a time-dependent window of fixed length that is used to plan the control moves from the current time to a future time for each step in the control process [6]. The moving horizon window has the same length as the prediction horizon (N_p). The prediction horizon is the number of future samples that are predicted from the current time.

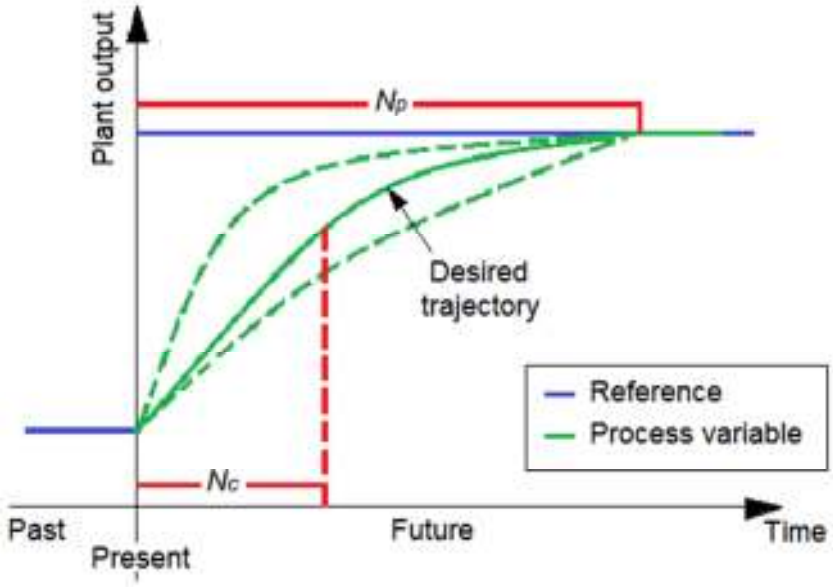

Figure 2-2 MPC principle [5]

Although the moving horizon window contains the optimal trajectory of the future control moves based on the measurements of the output at the current time, only the first sample of the optimal trajectory is given as an input to the plant, while the rest of the trajectory is disregarded. This is known as receding horizon control [5]. The control horizon (N_c) dictates the number of parameters used to represent the future control trajectory [6]. The control horizon has a length shorter than or equal to the prediction horizon.

The MPC control strategy can be represented as in Figure 2-3. This figure indicates the two main parts of the MPC strategy, namely the model and the optimisation algorithm. The process shown in Figure 2-3 is repeated at each control interval to find the optimal control variable.

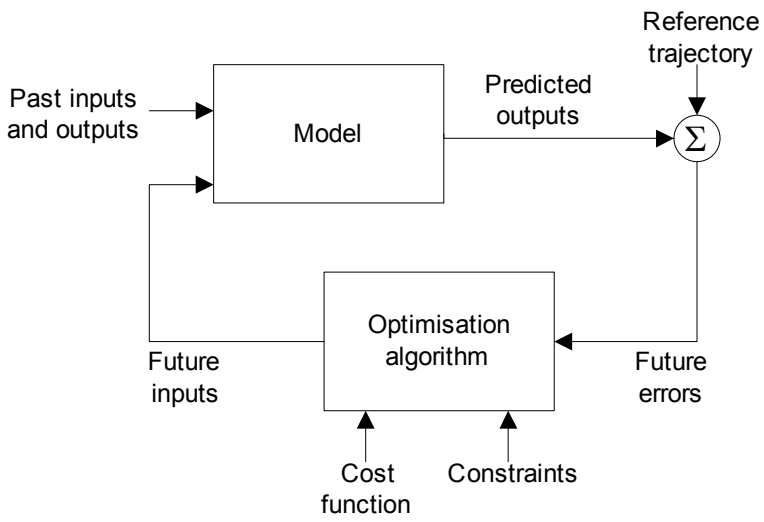

Figure 2-3 Basic representation of MPC strategy [5,6,9,12]

Model:

As the model is used to make predictions of what the future response of the system may be, it forms a critical part of the MPC design framework. The model can be either static or dynamic. A static model remains the same throughout the control process. A dynamic model is updated at each control interval by taking current and past measurements of system responses into account. A dynamic model may be necessary for applications where the dynamics of the system change during operation such as nonlinear applications or applications where disturbances may have a detrimental effect on the system performance [5]. Nonlinear plants can be represented by a linear model, however this model will only be viable for a certain operating range [14].

As mentioned previously, there exist many MPC algorithms that differ in minor ways, e.g. the model that is used in the design of the controller. Some of the models that are used are Finite-Impulse-Response (FIR) models or step response models, transfer function models, state-space models, and Directional Auto-Regressive Moving Average with Exogenous input (DARMAX) models [15].

From these models, state-space models are most commonly used due to the simplicity of the design and the ability to represent multivariable plants [6]. However in order to incorporate the state-space model into the design of the MPC controller, an estimator is needed to determine unknown or immeasurable states from process measurements in order to update the model and increase the accuracy of predictions [5,6].

These models may either be obtained by deriving the model from first principles or by developing an empirical or black-box model [15]. First principle models are derived from axioms or postulates that describe the system. In engineering applications these axioms or postulates are usually physical attributes of the system, which can be described by physical laws, such as Newton's laws. Empirical or black-box models on the other hand are obtained through techniques such as parameter fitting or system identification where the model parameters can be estimated from the measured data of the physical system [16]. "Smart" black-box models can also be obtained by training fuzzy-logic or neural network models from measured data [5].

Even though first principle models are more accurate to capture the dynamics of a system, it can be computationally intensive to derive a first principle model. This also leads to first principle models being more expensive to obtain. For this reason empirical or black-box models are more popular in the industry, even though they are less accurate [15].

Optimisation algorithm:

The optimisation algorithm is used to determine the optimal control trajectory by minimising the objective function. This is primarily done through the concept of predictive control.

Predictive control is used to predict the future process state or output over a finite time horizon [5]. From this information the future input is determined at each time step by minimizing the objective function which represents the cost function and constraints on manipulated or process variables. Of this future control trajectory only the first input to the plant is applied to the physical system. This process is then repeated at each step in the finite time horizon with the new measured input, state and output variables.

For MPC this predictive process is done by using a dynamic model of the system. The prediction errors are reduced by updating the system model after each control move is implemented during plant operation. By minimizing this prediction error the system can be optimized to obtain more stability or a faster closed loop response. However, increased stability and a faster response are often obtained inversely as the one is usually obtained at the cost of the other [5]. The control algorithm is used to control and optimize the system according to an optimal cost function and/or constraints.

In nonlinear systems or systems with multivariable constraints, optimisation usually involves a large computational load. This is due to the fact the objective function represents a quadratic cost function used to obtain the minimum square error, and inequality constraints for the feasible or optimal range of operation. Solving this objective function over a prediction horizon then leads to a quadratic programming problem that must be solved at each step in the prediction horizon [5,6].

For this reason academic and practical research has been done to obtain algorithms that can reduce this computational load. Some of the possible solutions are:

- The linearization of the system model to simplify the model, however this method may exclude some of the dynamics of the system [14].
- Reducing the order of the system model through reduction techniques such as singular value decomposition, Krylov-based approximation methods, or a combination of the two [17].
- Using Laguerre functions to represent the model and constraints of a system in order to reduce the computational load needed to compute the optimal control actions [6].
- Using neural networks or fuzzy networks to obtain “smart” models that can adapt to changes in the system [16].

- The use of convex optimisation algorithms that reduce computation time by making use of special patterns in the formulation of the objective function [5].
- Computing some of the variables that need to be computed only once off-line, and calling these variables during the on-line optimisation of the predictive control [6].

These solutions are merely a small number of possibilities. A wide variety of solutions are available in the literature on this topic.

2.2.3 Advantages of MPC

The advantage of MPC is that the design formulation has a completely multivariable framework where the performance parameters of the control system can be stated as engineering aspects. This means that for a system model where the parameter are known and can be inspected by the user, the operations done by the MPC controller can be monitored and evaluated during operation. The controller is also extended from a SISO controller to a MIMO controller with little increase in the complexity of the controller [5].

MPC also has the ability to implement hard and soft constraints on a multivariable system, which means that a system can operate close to its profit margins by placing constraints on the operation of a system [6]. Hard constraints are constraints that must be adhered to, whereas soft constraints are simply constraints that would optimize the system but are not implemented at the cost of hard constraints.

Another advantage of MPC is that a system can be optimised on-line. The simplicity of the design framework compared to other techniques that are able to control such complex MIMO systems is also a major advantage. This simplicity allows MPC to be implemented on simple systems such as microcontrollers [6].

2.2.4 MPC for flywheel energy storage systems

The concept of using MPC for the control of systems with fast dynamics such a flywheel energy storage system that is suspended by magnetic bearings is not new. Some research has been done to evaluate the use of MPC for such systems. One such study was conducted by [18], where a FESS was proposed with the radial motion was controlled by permanent magnetic bearings, and the axial motion was controlled by AMBs that utilised MPC. This study was only implemented in simulation, and reached a conclusion that MPC yielded superior stability, sensitivity and robustness to PID control for the same system.

Another study conducted by [19] focused on the cross-coupling between the top and bottom AMBs for a FESS with a vertical rotor. This study yielded simulation results that lead to the conclusion that the proposed MPC algorithm was very effective in reducing the effort needed to ensure robustness against external disturbances and model uncertainties.

A study conducted by Nguyen et. al. used a MPC algorithm that pre-computed the system behaviour in order to choose the optimal value of control variables [20]. This study implemented the MPC controller on a FESS and found that MPC was able to control the FESS effectively for fast acceleration and deceleration, that is, fast storing and releasing of energy.

Bleuler et. al. also discuss the effectiveness of different control techniques for AMBs. According to Bleuler et. al., the most effective and simplest control for a coupled system such as the FlyUPS is decentralised or decoupled parallel and conical mode control, such as decentralised PID control and centre-of-gravity (COG) coordinate control [4]. Modern control techniques that give promising results for controlling AMB systems are H_∞ and μ-synthesis, in particular for flexible rotor systems.

Observer or state estimator based control such as linear-quadratic-Gaussian (LQG) control, a technique on which MPC is based, offers no appreciable advantage over decentralised or decoupled control [4]. In fact, according to Bleuler et. al. observer based techniques can have destabilising effects due to uncertain dynamics in AMBs during rotation [4]. In the case where the observer based control is designed for a rotating system the controller may be very effective at the design speed, but may become unstable at other speeds due to non-conservative forces introduced by the controller [4]. This is particularly the case for model based control techniques such as MPC, where the dynamics of the system, and the model, change as the rotational speed changes.

2.3 Active magnetic bearings

Many methods exist for magnetic suspension in order to incorporate contact free support in systems where this feature may be necessary. One of these methods is active magnetic bearings (AMBs) [4]. AMBs were developed primarily due to the limitations of conventional journal or ball bearings such as load, size, stiffness, temperature, precision, speed, losses, dynamics, reliability and re-configurability [21].

The basic principle of operation of AMBs (see Figure 2-4) is that the position of the rotor is measured, and the error between this measured position and the reference position of the rotor is determined. According to this error the control unit determines the necessary control action

that will reduce this error in order to track the reference. The control action is then converted to a magnetic force by the power amplifier by applying a current in the coil of the electromagnet. It is due to this ability to alter the magnetic force that the term “active” is used. “Passive” magnetic bearings use permanent magnets to control the position of the rotor; however this gives a fixed control force which cannot be altered according to the application. [4]

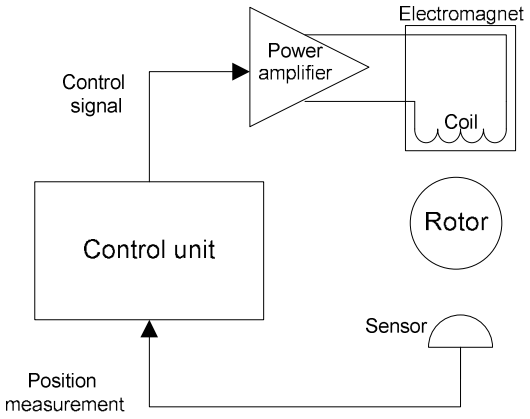

Figure 2-4 Basic operation of an AMB

The main advantage of AMBs over conventional bearings is that AMBs can support moving machinery without any physical contact [4,22]. This means that there is no friction, which in turn requires no lubrication and very little maintenance. Because friction is eliminated, the speed to which the rotor can be accelerated is limited only by the strength of the materials used in the design of the rotor and the capabilities of the power amplifiers to keep the rotor in position. AMBs are also ideal for applications such as vacuum systems, clean and sterile systems, systems used in the transport of aggressive or very pure media, or systems that operate at high temperatures [22].

Another advantage of AMBs is that they can be used in dynamic or “smart” systems where the control law can be used to determine the characteristics of the system such as the stiffness and damping, and the specific load capacity of the system [4]. Unbalance compensation and force-free rotation can also be managed by the control law. As the state of the rotor must be measured continually, online diagnostics can also be done to verify operating conditions and performance.

The main disadvantage of AMBs is that they are often difficult to control, depending on the application. This is because AMB systems are inherently non-linear and usually require multiple-input-multiple-output (MIMO) control methods to be implemented efficiently [22]. The inherent non-linearity is due to the force-distance and force-current relationships which can be explained by the following equation:

$$F_{mag} = k_e \frac{i_{control}^{\ 2}}{s^2} \tag{2-1}$$

where F_{mag} is the magnetic force, k_e is an equivalency constant, $i_{control}$ is the control current, and s is the distance between the rotor and the electromagnet. This means that the control force is proportional to the square of the control current and inversely proportional to the square of the air gap [21].

The design of AMB systems for a specific application can be very demanding as this requires knowledge of mechanical and electrical engineering and information processing, as well as knowledge about the specific application. The cost of developing an AMB system for a specific application can also be very expensive and time consuming [4].

In many applications the advantages of AMBs far outweigh the disadvantages and AMBs have become a necessity in these fields. These fields include: [4]

- Vacuum and cleanroom systems where contamination due to mechanical wear or lubrication must be avoided.
- High speed machinery such as turbo-machinery and high speed machining tools where very high speeds are necessary without the problems of heat and wear.

2.4 Flywheel uninterruptible power supply

As stated in Chapter 1, the major reason for the development of the flywheel uninterruptible power supply (FlyUPS) by the McTronX group of the Faculty of Engineering at the North-West University (NWU) was to improve the knowledge base of AMBs within the group. This project also aimed to develop the energy storage system completely in-house and to develop a product that could be supplied to industries that make use of the technology. Such a project could also be used as a valuable research tool as it would enable research on the development and optimization of AMB systems, the optimization of control algorithms and the implementation of redundancy within AMB systems. [1,2]

The FlyUPS is designed to deliver 2 kW of electrical energy for 3 minutes during power sags or power outages. The FlyUPS is a fully magnetically suspended AMB system meaning it has five degrees of freedom controlled by two radial AMBs and one axial AMB. The motor/generator mechanism of the FlyUPS contains a high speed permanent magnet synchronous motor (PMSM) that was also developed by the McTronX research group. This PMSM can rotate the flywheel to speeds of up to 30 000 r/min which enables the FlyUPS to mechanically store 527 kJ of energy. [1,2]

The principle of operation of the FlyUPS is that electrical energy is stored as kinetic energy in the flywheel rotor. As the energy is stored the speed of the flywheel is increased, and alternately it slows down as the energy is discharged [3]. In a typical application the FlyUPS is connected between an alternating current (AC) line and a load. A motor/generator configuration is used to convert the electrical energy to kinetic energy, and vice versa. A control unit, which in this application is a dSPACE® controller, is used to regulate the flow of electricity to and from the flywheel, implement the control and communicate status information between the control unit and user [1].

According to [1] the sensitivity of the AMBs used in the FlyUPS system was characterised as being in zone A for newly commissioned machines in accordance to the ISO CD 14839–3 standard on magnetic bearings. This indicates that the stiffness and damping characteristics of the AMBs have been verified and that the electrical system is fully functional. The control strategy implemented in [1] was decentralised PD control. It should be noted that this performance was measured in 2007. Various factors such as component aging, damage to components and the replacement of components can have an effect on the performance of the system [4].

Another important aspect of the FlyUPS is the rotor dynamics. The dynamics of the rotor pertain to the natural frequencies experienced during operation, and the effects of disturbances on the operation of the rotor. During a rotor dynamic analysis of the FlyUPS by [1,2] the natural frequencies of the flywheel rotor were determined to be at 48 Hz, 174 Hz and 732 Hz for the radial AMBs, and at 42 Hz for the axial AMB based on the design stiffness and damping. However, these natural frequencies can move up or down depending on the stiffness and damping which can be actively managed in AMB systems by the control strategy.

From this analysis the FlyUPS is expected to transverse only the first two rigid modes during operation up to 30 000 r/min.

2.5 System model

Dynamic models of a system are important in two aspects of controller design: simulation and control. As MPC uses of a model of the physical system to derive a controller for the physical system the model of the FlyUPS must also be investigated.

This development of a model of the FlyUPS system is documented in [8]. The model was developed by firstly deriving a model of the rotor and stator. Then the effects of the sensors,

filters and power amplifiers were included to give an accurate model of the FlyUPS. The goal was to obtain a linear state-space model of the FlyUPS. This model was verified and used to implement H_∞-control on the FlyUPS [8].

As illustrated in Figure 2-5 the model includes five degrees of freedom: the translations in the x-, y- and z-directions, as well as the rotations about the x- and y-axes. The rotation about the z-axis is controlled by the PMSM.

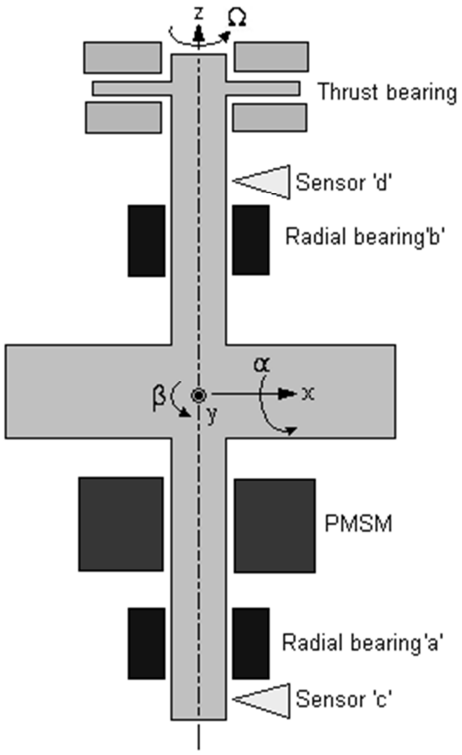

Figure 2-5 Sketch of FlyUPS indicating six degrees of freedom

The model is derived with the assumption that the rotor that is used is rigid. This means that the effects of bending modes are not modelled. Some other effects such as rotor touchdown and rotor whirl are also excluded from the model. However, these rotor dynamics can be represented as LTI differential equations by making the following assumptions [4]:

- Small rotor displacements.
- Rotationally symmetric rotor.
- Constant rotational speed.
- Decoupled radial and axial displacements.
- External forces and measurements are taken from discrete locations.

2.5.1 Linearization of AMB model

The difficulty of incorporating AMBs into applications is that AMBs are nonlinear. This is due to the force-current and force-distance relationships inherent in AMBs [4]. However, this system can be linearised within a linear interval around a working point by:

- Placing two electro-magnets opposite each other.
- Adding a pre-magnetization current (i_0) to the control current (i).
- Assigning an operating point (x_0).

This means that the force F applied by the AMBs to the rotor can be expressed as a linear function of current (i) and displacement (x), assuming small deviations from the operating point, i_0 and x_0:

$$F(i,x) = k_i i + k_s x \tag{2-2}$$

In (2-2), k_i is a constant representing the force-current relationship, and k_s is a constant representing the force-distance relationship of the AMB.

Figure 2-6 shows a flow diagram of a linearised open-loop AMB. In Figure 2-6 the input and output of the AMB are indicated as *u(s)* the control current, and *y(s)* the position of the rotor, respectively. The two opposing electromagnets are represented by the top and bottom combinations of the variables k_i and k_s. For each of the opposing electromagnets the control current is used to obtain a control force. The resulting force on the rotor is the difference between the forces generated by the opposing electromagnets. This resulting force is divided by the mass (m) of the rotor and then integrated twice to yield the resulting position of the rotor.

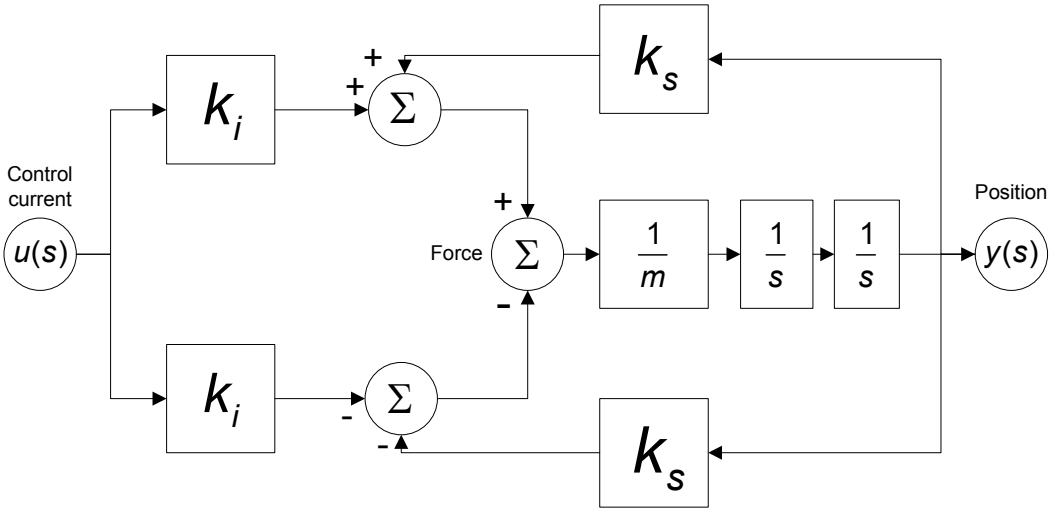

Figure 2-6 Flow diagram of linearised open-loop AMB [4]

From the flow diagram in Figure 2-6 a SISO, linearised, open-loop model for an AMB can be derived. In the case of the FlyUPS an AMB model will be used for each of the five degrees of freedom.

2.5.2 Radial AMB model

From the arrangement of the AMBs illustrated in Figure 2-5, the radial and axial motions can be controlled independently, or decoupled. Because of the translations and rotations about the x- and y-axes, there exists a cross coupling between the control of the radial bearings, *a* and *b*. However, this coupling is low for slow rotation because of small gyroscopic coupling [4].

The model is firstly represented by Newton-Euler equations of motion. Then by taking the moment of inertia, the mass and the gyroscopic coupling at each radial AMB into account and connecting this representation to the force-current and force-displacement constants of the system, a linear state-space model of the radial AMBs is obtained in the form:

$$\begin{aligned} \dot{x}_s &= A_s x_s + B_s u \\ y_s &= C x_s \end{aligned} \tag{2-3}$$

with states

$$\mathrm{x}_s = [x_c \; x_d \; y_c \; y_d \; \dot{x}_c \; \dot{x}_d \; \dot{y}_c \; \dot{y}_d]^T \tag{2-4}$$

The subscript *s* indicates that the state-space model is derived with regard to a sensor coordinate system. This leads to a fully coupled radial state-space model with four inputs (current to radial AMBs, *a* and *b*) and eight outputs (four displacement variables and their derivatives).

2.5.3 Axial AMB model

As the radial and axial displacements are decoupled and the PMSM controls the rotation about the z-axis, the thrust bearing will only need to control the translation in the z-direction. This means that the axial state-space model can be seen as a simple one degree of freedom point mass system. This also means that there is no need for coordinate transformations.

The state-space formulation for the axial AMB model is in the following form:

$$\begin{aligned} \dot{x}_z &= A_z x_z + B_z u_z \\ \dot{y}_z &= C_z x_z \end{aligned} \tag{2-5}$$

with states

$$\mathbf{x}_z = [z\ \dot{z}]^T \tag{2-6}$$

This representation of the axial AMB model has one input (current) and two outputs (the displacement variable and its derivative).

2.5.4 Complete FlyUPS model

In order to obtain a complete model of the FlyUPS the effects of the sensors, power amplifiers and anti-aliasing filters must also be taken into consideration. This is done for each of the five degrees of freedom.

The FlyUPS uses eddy-current sensors. This means that the sensors can be modelled by second order low-pass filters. From this transfer function (T_{sensor}), for experimentally measured displacements about the five possible axes (K_{sensor}), critical damping (ζ = 0.707) and a bandwidth of ω_s = 10 000 x 2π rad/s, the state-space sensor model is obtained by using a modified version of the controllable canonical form with numerically rescaled state vectors. This sensor model is then appended in series to the displacements of the AMB model. The sensor model for each degree of freedom can be written as:

$$T_{sensor} = K_{sensor} \cdot \frac{\omega_s^{\ 2}}{s^2 + 2\zeta\omega_s s + \omega_s^{\ 2}} \tag{2-7}$$

The power amplifiers (PAs) used by the FlyUPS are bi-state, PWM switching amplifiers. For this application these amplifiers are assumed to be identical and each PA is modelled as a closed-loop Partial-plus-Integral (PI) controller system with a bandwidth of 2.5 kHz. The transfer function of each PA is calculated as:

$$T_{PA} = \frac{2V_{bus}(K_I + K_P s)}{Ls^2 + (2K_P V_{bus} + R)s + 2K_I V_{bus}} \tag{2-8}$$

where V_{bus} is the bus voltage, R is the coil resistance, L is the coil inductance, K_P is the proportional constant, and K_I is the integral constant. The model of each PA is then converted to state-space form in a similar fashion to the sensors. The PA models are then cascaded to each current input of the system model.

The final component which must be added to the FlyUPS model is the anti-aliasing filter. This filter is modelled as a first order low-pass filter with a cut off frequency of 4.82 kHz. This filter model is then cascaded after the displacement sensors to complete the system model.

The final model of the system in the x- and y-directions is a continuous-time, linear state-space model of the form:

$$\begin{aligned} \dot{x}_{xy} &= A_{xy}x_{xy} + B_{xy}u_{xy} \\ y_{xy} &= C_{xy}x_{xy} \end{aligned} \tag{2-9}$$

with the dimensions: A_{xy}(28x28), B_{xy}(28x4) and C_{xy}(8x28). This relates to four inputs, twenty eight states and eight outputs.

The final model of the system in the z-direction is a continuous-time linear state-space model of the form:

$$\begin{aligned} \dot{x}_z &= A_z x_z + B_z u_z \\ y_z &= C_z x_z \end{aligned} \tag{2-10}$$

with the dimensions: A_z(7x7), B_z(7x1) and C_z(2x7). This relates to one input, seven states and two outputs.

By appending the axial model (z-direction) to the radial model (x- and y-directions) a MIMO model of the FlyUPS is formed to represent the five degree of freedom AMB system. Figure 2-7 indicates the model as derived by [8] and where the control strategy will be connected during simulations.

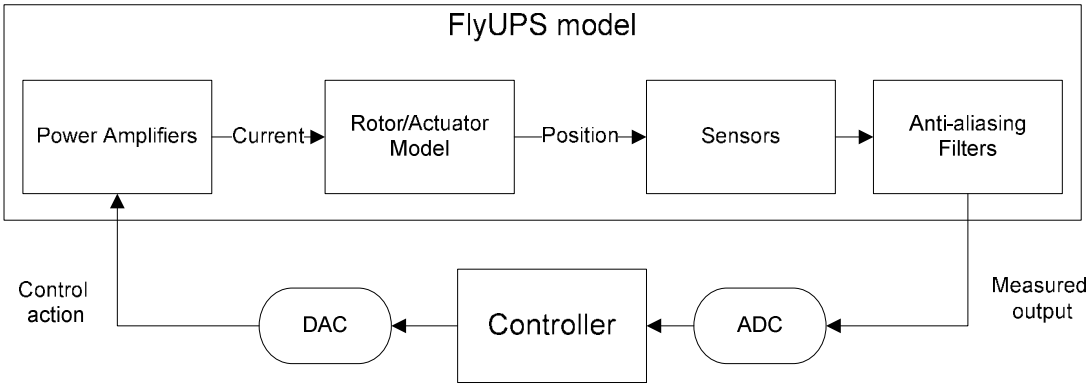

Figure 2-7 FlyUPS model including controller

2.5.5 Model order reduction

From the model of the FlyUPS discussed in the previous section, it is seen that the model for the full five degree of freedom FlyUPS results in a thirty fifth order model. As the MPC controller will be designed based on the model of the system, the size and complexity of the controller may cause complications. For this reason model reduction is investigated.

Within the MATLAB® framework there exists three basic approximation categories: singular value decomposition (SVD) based methods, Krylov-based methods, and iterative methods that combine the strengths of both SVD and Krylov based methods [23]. Each category contains some techniques to deal with dynamic linear and non-linear systems. However, this discussion

will focus on SVD as this technique is simple yet effective in greatly reducing the order of a model while considering the most important dynamics.

SVD is a model reduction method used to approximate matrices with matrices of lower rank where the optimal solution is found in unitarily invariant norms [23]. The singular values (SVs) of a system are the square roots of the eigenvalues of the product of the system model and its adjoint. These SVs indicate by how many states a finite-dimensional system model can be reduced. The model order reduction is done according to a predetermined error bound which determines how much of the system dynamics are included in the reduced model.

An important set of SVs are the Hankel singular values that are determined from the square roots of the eigenvalues of the product of the controllability Gramian and the observability Gramian [23]. In order to obtain the controllability and observability Gramians, two Lyupanov equations need to be solved. This requires dense computations, but does not involve large computations, even for large systems [23].

The Hankel SVs of a system model give an indication of the amount of energy contained in each state. From the Hankel SVs a system model can be reduces through various methods, such as balanced model reduction where the states that contain large amounts of energy are maintained while states with less energy are discarded. In many applications the Hankel SVs decay very rapidly, which indicates that low order approximations are possible.

2.6 Critical literature overview

This section aims to bring the literature discussed in this chapter in context with the topic of this dissertation, namely the model predictive control of a magnetically suspended flywheel energy storage system. From the problem statement this topic can be understood as the performance assessment of MPC for the FlyUPS.

A major aspect of this study is that the rotor is magnetically suspended by AMBs. This holds various advantages such as contact-free and thus frictionless support, the ability to control the characteristics of the bearings, and the online monitoring of the condition of the system. However, AMBs often require complex control strategies to manage the inherent nonlinearity of the AMBs and to implement MIMO control to handle the coupling between multiple AMBs in a system. [4]

The AMB system that is used to assess the performance of MPC for a high-speed flywheel application is the FlyUPS. The development of the FlyUPS is fully documented in [1,2]. These

sources discuss the hardware and software of which the FlyUPS system consists. [1] also discusses the implementation and evaluation of PD control on the FlyUPS. In [2] a rotor dynamic analyses of the flywheel rotor is performed to find the critical frequencies of the FlyUPS at the design stiffness and damping.

The model of the FlyUPS that is used to derive the MPC controller and to perform simulations was discussed in section 2.5. This is a linearised, continuous-time model of the FlyUPS containing the rotor dynamics that are expected within the rated operating speed of 30 000 r/min. These rotordynamic effects include the gyroscopic effects, the effects of inertia and the critical frequencies corresponding to the first two rigid modes of the FlyUPS. The model is analytically derived to include the AMBs, sensors, filters and power amplifiers for the five degrees of freedom at a fixed operating speed [8]. Model order reduction techniques that can be used to reduce the computational load of model-based control techniques are also discussed in section 2.5.5. From these techniques Hankel SVD was chosen as this technique is simple yet effective in reducing the model order.

Based on this background of the system MPC was determined as a possible control strategy. MPC uses the previously discussed model of the system to implement optimal MIMO control. The optimal control actions are calculated based on a cost function that minimises the error between the predicted optimal outputs and the measured outputs of the system, and constraints that specify the optimal operating conditions. This means that modelled dynamics such as cross-coupling can be anticipated in the control action. [5,6,9]

Some common disadvantages of MPC that can determine the success of this control technique on the FlyUPS are the large computational load resulting in long execution times that may make real-time control impossible, and uncertain dynamics of the FlyUPS that can have destabilising effects outside of the parameters of the controller design [5,6,9]. The use of estimators can also cause the system to become unstable during operation due to non-conservative forces introduced by the controller [4].

2.7 Conclusion

This chapter has presented a literature study of the most important topics present in this dissertation. As AMBs are inherently open-loop unstable, an accurate model is required to ensure the effective operation of MPC for the FlyUPS. For a linear MPC control strategy to be implemented on the FlyUPS a linear model of the system is used, which means that a certain degree of robustness is required from the MPC control algorithm. The next chapter discusses the design of the MPC controller for the FlyUPS.

3

Chapter

Controller design

This chapter discusses the design of an MPC controller for the FlyUPS. Based on the model of the FlyUPS discussed in the previous chapter an MPC controller is now derived. A background of the terminology is given, followed by a discussion of a classical algorithm for MPC. Then two MPC algorithms for the FlyUPS are discussed, namely an extension of the classical algorithm to incorporate optimisation strategies, and the algorithm used in the MATLAB® Model Predictive Control toolbox™.

This discussion focuses on the design of a discrete time MPC controller. Also, as the complete state-space model of the FlyUPS is a multi-input-multi-output (MIMO) model with five inputs and ten outputs (four radial position outputs, one axial position output, and the speed derivative of each of these outputs), the controller is also designed to be MIMO.

3.1 MPC concept and terminology

Before the theory behind the derivation of a discrete-time MPC controller is discussed, some concepts and terminology must first be introduced. Figure 3-1 indicates the signals that are present in the control and operation of a plant.

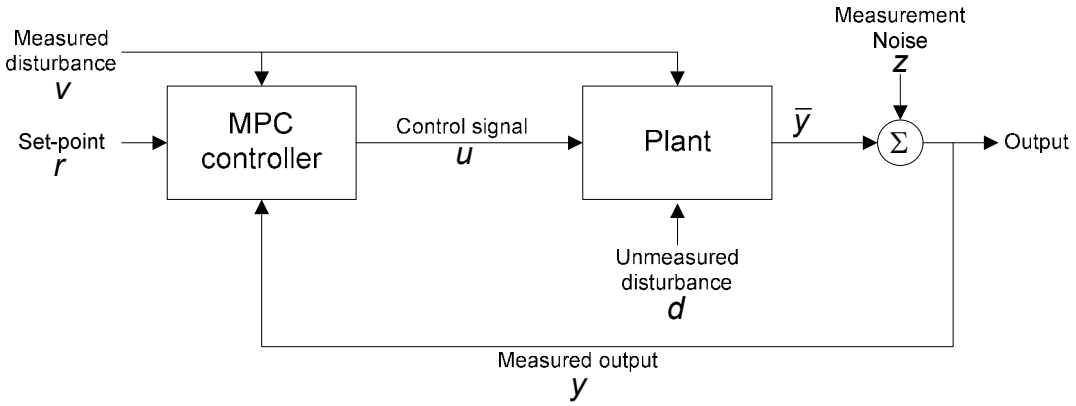

Figure 3-1 Typical MPC scheme configuration [9]

As illustrated in Figure 3-1 the MPC controller calculates the control signal (*u)* from the set-point (*r*), measured output (*y*) and measured disturbance signals (*v*). As the MPC controller makes use of a model of the system to make predictions, provision can also be made for unmeasured

disturbances (*d*). The measurement noise (*z*) on the actual output signal ($\bar{y}$) can also be incorporated into the design.

Figure 3-2 indicates the operating principle of an MPC controller for a SISO example. It is assumed that the plant has been operating for some time. The objective is to hold the output at a set-point or reference value. Figure 3-2 shows the plant state at the current sampling instant, *k*. At this sampling time the set-point changes and controller starts to compensate for the error between the set-point and the output of the system.

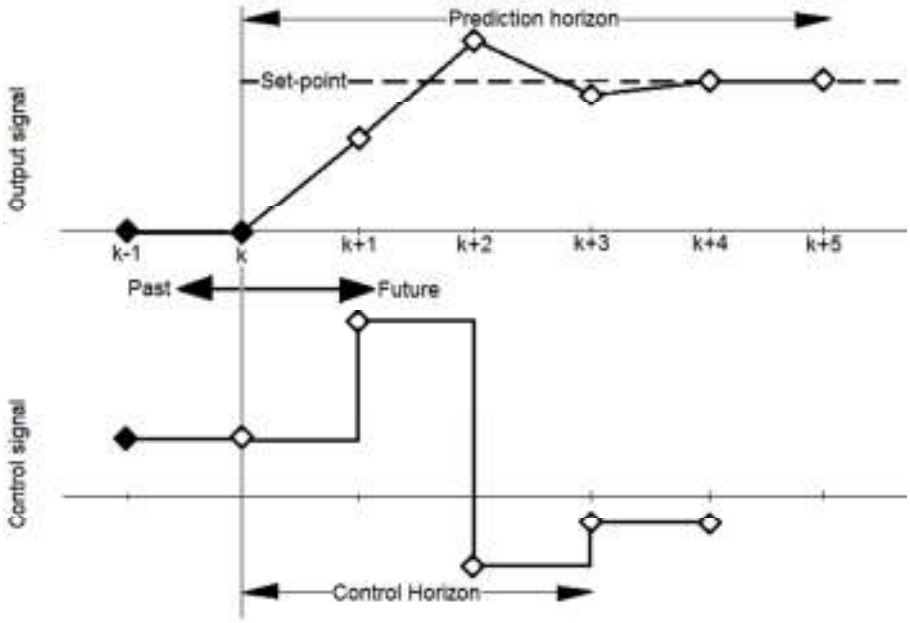

Figure 3-2 MPC operating principle [5,6,9]

In Figure 3-2 the measured outputs and the past control moves of the system are indicated as filled diamonds. The empty diamonds indicate the predicted outputs and future control moves, respectively. The step-wise variations in the control signals can be seen as the control actions implemented by the controller, and held at this value by the zero order hold function until the next sampling instant. The output of the plant is only known at the sampling instants, however the plant is assumed to be continuous. As the controller is discrete the sampling periods are assumed to have a fixed duration (Δt) from the current sample to the next.

Before the MPC controller can calculate the optimal control signal the current state of the system must be known. This includes any states that affect the actual output signal ($\bar{y}$) as well as the prediction of the future output signal ($\bar{y}_{k+1}, \bar{y}_{k+1}, \ldots, \bar{y}_{k+Np}$). If the state of the system cannot be measured directly, it can be estimated from the past and current measurements of the system and the models of the plant, disturbance and noise.

When the current state of the system is known the optimal control signal is calculated over the control horizon ($N_c \geq 1$) based on the set-points, constraints, and disturbances as defined over the prediction horizon ($N_p \geq 1$; $N_p \geq N_c$). From the optimal control signal the control action for the current sampling instant (u_k) is then applied to the system. At the next sampling instant, k+1, this process is repeated based on the new measurements.

The length of the prediction and control horizons play an important role in the design of the MPC controller. Firstly, for a large prediction horizon, constraints that will become active can be anticipated and avoided [9]. Secondly, in systems with delays, the prediction horizon must be long enough to take the effect of the delay into consideration, and the control horizon must be short enough to carefully consider the effect of the control move [9]. Lastly, for a non-minimum phase system with a short term response in one direction and a long term response in the other direction, the prediction horizon must be long enough to take the long term response into consideration [9].

3.2 Classic MPC controller design

As various methods for the design of an MPC controller exist, this section focuses on the design of a classic MPC algorithm as discussed in [5,6]. This discussion is divided into the formulation of the model for MPC design followed by the formulation of the predictive control strategy. Then use of observers in the design of the predictive controller is discussed. Finally the implementation of constraints in the design of the MPC controller is discussed.

3.2.1 Formulation of the model

For a plant with n_u inputs, n_y outputs and n_x states, the general formulation of the MIMO model can be represented by

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + B_d \omega(k) \\ y(k) &= C_m x_m(k) \end{aligned} \quad . \tag{3-1}$$

This formulation incorporates plant noise and disturbance ($\omega(k)$) which is assumed to be a sequence of integrated white noise. It should be noted that for plants where the number of outputs is larger than the number of inputs, each measured output cannot be controlled independently with zero steady-state error [5].

This model in (3-1) is augmented in order to embed the model with integrators [6]. This will ensure that the steady-state error will be minimized for each output. This results in the augmented state-space model [6]:

$$\overbrace{\begin{bmatrix}\Delta x_m(k+1)\\ y(k+1)\end{bmatrix}}^{x(k)} = \overbrace{\begin{bmatrix}A_m & o_m^T\\ C_mA_m & \mathrm{I}_{n_y\times n_y}\end{bmatrix}}^{A}\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix} + \overbrace{\begin{bmatrix}B_m\\ C_mB_m\end{bmatrix}}^{B}\Delta u(k) + \begin{bmatrix}B_d\\ C_mB_d\end{bmatrix}\varepsilon(k)$$
$$y(k) = \underbrace{\begin{bmatrix}o_m & \mathrm{I}_{n_y\times n_y}\end{bmatrix}}_{C}\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix} \tag{3-2}$$

where $\varepsilon(k)$ represents the zero mean, white noise sequence. I_{nyxny} is a $n_y \times n_y$ identity matrix, and o_m is a $n_y \times n_x$ null matrix. The matrices A_m, B_m and C_m have dimensions $n_x \times n_x$, $n_x \times n_u$ and $n_y \times n_x$, respectively. Thus the dimension of the augmented state transition matrix will be $n = n_x + n_y$.

The eigenvalues of the system can be determined from the characteristic equation of the augmented model [6]:

$$\begin{aligned}\rho(\lambda) &= \det\begin{bmatrix}\lambda\mathrm{I} - A_m & o_m^T\\ -C_mA_m & (\lambda-1)\mathrm{I}_{n_y\times n_y}\end{bmatrix}\\ &= (\lambda-1)^{n_y}\det(\lambda\mathrm{I} - A_m)\\ &= 0 \qquad .\end{aligned} \tag{3-3}$$

This indicates that the eigenvalues of the augmented model contain the eigenvalues of the plant, as well as n_y eigenvalues having the values λ=1. These n_y eigenvalues indicate that there are n_y integrators embedded into the augmented model.

When designing the controller based on the augmented state-space model, one must ensure that the augmented model is not uncontrollable or unobservable. Controllability is necessary to achieve a desired closed-loop control performance. Observability is necessary to implement an observer in the controller design. However, if the system only requires closed-loop stability, it may only be necessary to ensure stabilisability and detectability. [24]

3.2.2 Formulation of predictive control

From the model discussed in the previous section, the predicted output of the system is now calculated. The future control signal will be used to optimize the system within an optimization window of length N_p. For MIMO systems it is important to note the dimensions of the state, control and output vectors.

The first step is to define the vectors ΔU and Y as [6]:

$$\Delta U = \begin{bmatrix}\Delta u(k)^T & \Delta u(k+1)^T & \cdots & \Delta u(k+N_c-1)^T\end{bmatrix}^T \tag{3-4}$$

$$Y = \begin{bmatrix}y(k+1\,|\,k)^T & y(k+2\,|\,k)^T & \cdots & y(k+N_p\,|\,k)^T\end{bmatrix}^T \tag{3-5}$$

where ΔU contains the future control trajectories ($\Delta u(k+n)^T$) for all inputs over the control horizon (N_c) and Y contains the predicted output variables ($y(k+i|k)^T$) over the prediction horizon (N_p). The notation $(k+i|k)$ indicates the predicted output at time $k+i$ based on the measured output at the current time k.

The future state variables can now be calculated from the state-space formulation (A, B, C) and the future control moves as [6]:

$$\begin{aligned}
x(k+1\,|\,k) &= A\,x(k)+B\,\Delta u(k)+B_d\,\varepsilon(k)\\
x(k+2\,|\,k) &= A\,x(k+1)+B\,\Delta u(k+1)+B_d\,\varepsilon(k+1\,|\,k)\\
&= A^2\,x(k)+AB\,\Delta u(k)+B\,\Delta u(k+1)\\
&\quad +AB_d\,\varepsilon(k)+B_d\,\varepsilon(k+1\,|\,k)\\
&\vdots\\
x(k+N_p\,|\,k) &= A^{N_p}\,x(k)+A^{N_p-1}B\,\Delta u(k)+A^{N_p-2}B\,\Delta u(k+1)\\
&\quad +A^{N_p-N_c}B\,\Delta u(k+N_c-1)+A^{N_p-1}B_d\,\varepsilon(k)\\
&\quad +A^{N_p-2}B_d\,\varepsilon(k+1\,|\,k)+\ldots+B_d\,\varepsilon(k+N_p-1\,|\,k)
\end{aligned} \tag{3-6}$$

under the assumptions that $\varepsilon(k)$ is a zero mean white noise sequence. This means that the predicted noise effects on state variables and output variables will be negligible, thus $\varepsilon(k)$ can be eliminated from the calculations of the state variables in the equations above.

Commonly, the predictive control of MIMO systems can is then written in terms of (3-4) – (3-6) as [6]:

$$Y = F\,x(k)+\Phi\,\Delta U\,, \tag{3-7}$$

where F is related to the observability and Φ is related to the controllability. These variables are calculated as:

$$F=\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix};\ \Phi=\begin{bmatrix} CA & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}.$$

In order to minimize the difference between the predicted output and a set-point signal, a cost function must be defined. This cost function or objective function is defined as [6]:

$$J=(R_s-Y)^T(R_s-Y)+\Delta U^T\bar{R}\,\Delta U \tag{3-8}$$

where R_s is a vector of all of the set-points, and $\bar{R}$ is a diagonal matrix of the form:

$$\bar{\dot{R}} = r_w \, \mathrm{I}_{N_c \times N_c} . \tag{3-9}$$

In (3-9), r_w can be seen as a tuning parameter for the weight on the future control trajectory. As r_w is increased more caution is exercised when reducing the prediction error with respect to the magnitude of the control signal.

In order to minimize the cost function, the derivative of the cost function is firstly determined after which it is set equal to zero. As a result, the optimal incremental control after one optimization window can be calculated by [6]:

$$\Delta U = (\Phi^T \Phi + \bar{\dot{R}})^{-1} (\Phi^T \bar{\dot{R}}_s \, r(k) - \Phi^T F \, x(k)) . \tag{3-10}$$

Equation (3-10) holds true if and only if the Hessian matrix $(\Phi^T\Phi + \bar{\dot{R}})^{-1}$ exists. In this equation $\bar{\dot{R}}$ is known as a weight matrix containing the tuning parameters for the closed-loop response. $\bar{\dot{R}}$ has the form of a block matrix with q blocks and has the same dimension as $\Phi^T\Phi$. $\bar{\dot{R}}_s$ is a data matrix filled with ones and has the same dimension as Φ^T. $r(k)$ contains the n_y set-point signals for the multi-output system.

To implement receding horizon control only the first m elements of ΔU are considered when calculating the incremental optimal control. This leads to the following equation [6]:

$$\begin{aligned} \Delta u(k) &= \begin{bmatrix} I_m & o_m & \cdots & o_m \end{bmatrix} (\Phi^T \Phi + \bar{\dot{R}})^{-1} (\Phi^T R_s \, r(k) - \Phi^T F \, x(k)) \\ &= K_y \, r(k) - K_{mpc} \, x(k) \end{aligned} . \tag{3-11}$$

Equation (3-11) is a standard form for linear time-invariant state feedback control. K_{mpc} represents the state feedback control gain vector and K_y represents the feedback gain vector. By applying (3-11) to the augmented model of (3-2), the following equation is obtained for the closed-loop system [6]:

$$x(k+1) = (A - BK_{mpc}) \, x(k) + BK_y \, r(k) . \tag{3-12}$$

From (3-12) the closed-loop eigenvalues of the system can be calculated from the closed-loop characteristic equation as:

$$\det[\lambda \mathrm{I} - (A - BK_{mpc})] = 0 . \tag{3-13}$$

The structure of this formulation causes K_y to be identical to the last element of K_{mpc}. This means that K_{mpc} can be written as

$$K_{mpc} = [K_x \quad K_y] , \tag{3-14}$$

where K_y represents the feedback gain vector for $\Delta x_m(k)$, and K_y represents the feedback gain vector for $y(k)$.

3.2.3 Observers in predictive control design

In most applications, not all of the state variables of a system are measurable. For this purpose an observer is implemented to estimate the unknown state variables from available process measurements. The observer can also be used to reduce the effect of noise on measured state variables [5].

Firstly, in order for the state variables of any system to be estimated, the system must be observable. A dynamic system is observable if for any future time there exists sufficient knowledge about the input and output to the system at the present time to determine the initial state of the system and consequently any future states based on the initial state. The observability of a linear discrete-time system can be verified by ensuring that the observability matrix given by (3-15) has rank *n*, when the state variable model is *n* dimensional [24].

$$L_0 = \begin{bmatrix} C_m \\ C_m A_m \\ C_m A_m^2 \\ \vdots \\ C_m A_m^{n-1} \end{bmatrix}. \tag{3-15}$$

In order to obtain an accurate estimate of the state variable of a system, the estimation error ($\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k)$) must be kept to a minimum. The error will decrease as the estimated state $\hat{x}_m(k)$ converges to the actual state $x_m(k)$. This is only possible if A_m is stable and has poles inside the unit circle. The location of the system poles will determine the convergence rate. The closer the poles are to the origin of the complex plane, the faster the estimation error will converge to zero [24].

For the augmented system model the next state can be estimated with an observer by [6]:

$$\hat{x}(k+1) = A\,\hat{x}(k) + B\,\Delta u(k) + K_{ob}(y(k) - C\,\hat{x}(k)). \tag{3-16}$$

In (3-16) the term K_{ob} is the observer gain matrix and is used together with a correction term based on the error between the measured output $y(k)$ and the predicted output $C\hat{x}(k)$, to reduce the error in the predicted model. When considering the receding horizon control of the system, the optimal control move $\Delta u(k)$ at time k can be written as a standard feedback control law as in (3-11), with estimated states $\hat{x}(k)$. This results in the following equation:

$$\Delta u(k) = K_y\, r(k) - K_{mpc}\,\hat{x}(k). \tag{3-17}$$

The closed-loop observer error can be calculated as [6]:

$$\tilde{x}(k+1) = (A - K_{ob}\,C)\,\tilde{x}(k). \tag{3-18}$$

From (3-17) and (3-18) it can easily be shown that the closed-loop MPC system containing an observer has two independent characteristic equations: [6]

$$\begin{aligned} \det\left[\lambda \mathrm{I} - (A - K_{ob} C)\right] &= 0 \\ \det\left[\lambda \mathrm{I} - (A - BK_{mpc})\right] &= 0 . \end{aligned} \tag{3-19}$$

This indicates that the closed-loop system eigenvalues consists of the observer eigenvalues and the predictive control eigenvalues. As there are two independent characteristic equations, the design of the observer and the predictive control law can be carried out separately and then combined without affecting the stability of the system [6].

3.2.4 Constraints in the MPC controller design

By incorporating constraints into the design of the MPC controller, potential control saturation during plant operation can be anticipated and avoided. This control saturation can be due to a variety of causes such as physical constraints on the position of the rotor, or the amount of current that a power amplifier can supply. Constraints can also be set for the operation of the system in order to ensure optimal operation.

These constraints can be either *hard* or *soft*. Hard constraints are constraints that have to be adhered to whereas soft constraints will only be adhered to if they do not cause conflict with any other constraints. In the case of soft constraints, a slack variable can be used to determine to what extent the soft constraint may be violated, and how constraint violations are penalised [5].

The constraints that are placed on a plant can be classified as constraints on the input of the system, on the rate of change of the input of the system, and on the output of the system. In order to use these constraints to find the optimal control signal, it is necessary to parameterize each constraint to use the same parameters as used in the design of predictive control. The parameter that will be used is the decision variable ΔU as described in (3-4).

With ΔU defined as the incremental control for each input, the constraints on the incremental change in the control signal for each input can be defined as:

$$\Delta U^{\min} \le \Delta U \le \Delta U^{\max} , \tag{3-20}$$

where ΔU^{min} is the vector containing the lower limits, and ΔU^{max} is the vector containing the upper limits for each input. In matrix form these inequalities can be written as

$$\begin{bmatrix} -\mathrm{I} \\ \mathrm{I} \end{bmatrix} \Delta U \le \begin{bmatrix} -\Delta U^{\min} \\ \Delta U^{\max} \end{bmatrix} . \tag{3-21}$$

By following the same approach as with the incremental change in the control, the constraints on the amplitude of the control signal can be written in terms of ΔU as [6]:

$$\begin{aligned} -\left(C_1 u(k-1) + C_2 \Delta U\right) &\le -U^{\min} \\ \left(C_1 u(k-1) + C_2 \Delta U\right) &\le U^{\max} \end{aligned} \tag{3-22}$$

with C_1 and C_2 relating the constraints to the past control actions and the decision variable ΔU, respectively.

The output constraints can be written in terms of the decision variable ΔU as

$$Y^{\min} \le Fx(k) + \Phi \Delta U \le Y^{\max}, \tag{3-23}$$

which corresponds to (3-7).

The control problem now becomes finding the decision variable ΔU that minimises the cost function [6]:

$$J = \left(R_s - F\,x(k)\right)^T \left(R_s - Fx(k)\right) - 2\,\Delta U^T \Phi^T \left(R_s - F\,x(k)\right) + \Delta U^T \left(\Phi^T \Phi + \bar{R}\right) \Delta U, \tag{3-24}$$

subject to the inequality constraints:

$$\begin{bmatrix} M_u \\ M_{\Delta u} \\ M_y \end{bmatrix} \Delta U \le \begin{bmatrix} N_u \\ N_{\Delta u} \\ N_y \end{bmatrix} \tag{3-25}$$

with $M_u = \begin{bmatrix} -C_1 \\ C_2 \end{bmatrix}$; $N_u = \begin{bmatrix} -U^{\min} + C_1 u(k-1) \\ U^{\max} - C_1 u(k-1) \end{bmatrix}$; $M_{\Delta u} = \begin{bmatrix} -\mathrm{I} \\ \mathrm{I} \end{bmatrix}$; $N_{\Delta u} = \begin{bmatrix} -\Delta U^{\min} \\ \Delta U^{\max} \end{bmatrix}$; $M_y = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}$;

$N_y = \begin{bmatrix} -Y^{\min} + Fx(k) \\ Y^{\min} - Fx(k) \end{bmatrix}$.

For simplicity these inequality constraints can be written as [5]:

$$M \Delta U \le \gamma_c \tag{3-26}$$

where M indicates which parameters have constraints, and γ_c indicates the value of each constraint. This formulation of inequality constraints together with the quadratic cost function leads to a quadratic programming problem [5].

3.3 Optimisation techniques in MPC design

As discussed in the previous chapter some techniques can be used to reduce the computational load needed to implement effective MPC. One such technique, as discussed by the author of [6], incorporates Laguerre functions in the design of a classical MPC algorithm. Other

optimisation functions such as a prescribed degree of stability and exponential data weighting are also added to the design.

3.3.1 Laguerre functions in MPC design

Laguerre functions are orthonormal basis functions used to approximate the model and the constraints in the design of the MPC controller [6]. The use of Laguerre functions helps to reduce the number of parameters needed for long control horizons. This means that the number of computations needed to obtain accurate responses is reduced. The use of Laguerre functions also allows tuning parameters to be incorporated into the design of the MPC controller [6]. These tuning parameters can be used to determine the performance of the controller.

Any function can be approximated by a discrete polynomial function [6]. Under this premise any element within the control vector can be represented as a function of the Dirac δ-operator and the decision variable as:

$$\Delta u(k+i) = \begin{bmatrix} \delta(i) & \delta(i-1) & \cdots & \delta(i-N_c-1) \end{bmatrix} \Delta U \,, \tag{3-27}$$

where $\delta = \begin{cases} = 1, \textit{if } i = 0 \\ = 0, \textit{if } i \neq 0 \end{cases}$ is the pulse operator and the centre of the pulse moves forward as the index increases. In this way the control vector can also be approximated as a set of discrete Laguerre functions.

The set of discrete-time Laguerre networks can be written as [6]:

$$\begin{aligned} \Gamma_1(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \\ \Gamma_2(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \frac{z^{-1}-a}{1-az^{-1}} \\ &\vdots \\ \Gamma_N(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \left(\frac{z^{-1}-a}{1-az^{-1}} \right)^{N-1} . \end{aligned} \tag{3-28}$$

The scaling factor ($0 \leq a < 1$) is used to select the position of the poles in a Laguerre network describing the system. From the representation in (3-28) it can be seen that the Laguerre function at the current time can be written as a function of previous Laguerre functions as:

$$\Gamma_k(z) = \Gamma_{k-1}(z) \frac{z^{-1}-a}{1-az^{-1}} \,. \tag{3-29}$$

By taking the inverse $\mathcal{Z}$-transform of (3-29) the set of discrete Laguerre functions can be written as [6]:

$$L(k) = \begin{bmatrix} l_1(k) & l_2(k) & \cdots & l_N(k) \end{bmatrix}^T . \tag{3-30}$$

Following the same method as in (3-29) these discrete Laguerre functions can be written as the following difference equation:

$$L(k+1) = A_l L(k) \tag{3-31}$$

where A_l is an $N \times N$ matrix relating to the previous Laguerre functions, and is a function of a and $\beta = 1 - a^2$. The initial condition for this definition of discrete Laguerre functions is:

$$L(0) = \sqrt{\beta}\begin{bmatrix} 1 & -a & a^2 & \cdots & (-1)^{N-1}a^{N-1} \end{bmatrix}. \tag{3-32}$$

As the Laguerre functions are orthonormal basis functions, the orthonormality for this definition of discrete Laguerre functions also holds in the time domain and can be written as [6]:

$$\begin{aligned} &\sum_{k=0}^{\infty} l_i(k) l_j(k) = 0 \ \textit{for} \ i \neq j \\ &\sum_{k=0}^{\infty} l_i(k) l_j(k) = 1 \ \textit{for} \ i = j \ . \end{aligned} \tag{3-33}$$

For a stable system ($H(k)$) the impulse response can be given as [6]:

$$H(k) = c_1 l_1(k) + c_2 l_2(k) + \cdots + c_N l_N(k) \tag{3-34}$$

for a given number of terms N used to approximate the system, and the coefficients $c_1,\ldots,c_N$ determined from system data. Based on the orthonormality of the Laguerre functions, the coefficients of the equivalent Laguerre networks, for $i = 1,2,\ldots,N$, can be written as [6]:

$$c_i = \sum_{k=0}^{\infty} H(k) l_i(k) . \tag{3-35}$$

Due to the orthonormality of these Laguerre networks, these coefficients minimize the sum of squared errors between the actual system and the Laguerre networks [6]:

$$J_{SE} = \sum_{k=1}^{\infty} \left(H(k) - \sum_{i=1}^{N} c_i l_i(k) \right)^2 . \tag{3-36}$$

This approximation becomes more accurate as N increases.

Now, in order to approximate the model of the FlyUPS, each input must be assigned its own pole location a. By doing this the decay rate of the incremental control of each input can be controlled. For the system, define the incremental control signal for each input as:

$$\Delta u(k) = \begin{bmatrix} \Delta u_1(k) & \Delta u_2(k) & \cdots & \Delta u_{n_u}(k) \end{bmatrix}^T \tag{3-37}$$

and define the input matrix for each input as:

$$B = \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_u} \end{bmatrix}. \tag{3-38}$$

Each input of the system can then be represented in terms of Laguerre functions, with a specific a_i and N_i for each input, as:

$$\Delta u_i(k) = L_i(k)^T \eta_i \tag{3-39}$$

where $L_i(k)$ and η_i describe the Laguerre network for the *i*-th input.

Based on (3-39) and assuming that the current state of the system is known, the future state at sample *p* can then be calculated as [6]:

$$x(k+p \mid k) = A^p x(k) + \phi(p)^T \eta \tag{3-40}$$

with the data matrix $\phi(p)^T$ and the parameter vector η defined as:

$$\begin{aligned} \phi(p)^T &= \sum_{j=0}^{p-1} A^{p-j-1} \left[B_1 L_1(j)^T \quad B_2 L_2(j)^T \quad \cdots \quad B_{n_u} L_{n_u}(j)^T \right] \\ \eta^T &= \left[\eta_1^T \quad \eta_2^T \quad \cdots \quad \eta_{n_u}^T \right] \end{aligned} \tag{3-41}$$

The cost function of the system is defined in a similar way to the cost used in the design of linear quadratic Gaussian (LQG) control [5]:

$$J = \sum_{j=0}^{\infty} \left(x(k+j-1)^T Q\, x(k+j-1) + \Delta u(k+1)^T R\, \Delta u(k+j) \right). \tag{3-42}$$

This formulation presents the cost function as the sum of squared errors, with the performance parameters *Q* and *R*. *Q* is a positive semi-definite matrix representing the weight on the outputs and *R* is a positive definite matrix representing the weights on the incremental control. Larger values of *Q* indicate that faster responses are required from the corresponding outputs, and larger values of *R* indicate that greater consideration is necessary when calculating control signals.

The cost function for the design using Laguerre functions can be defined as the sum of squared errors, similar to (3-42), as [6]:

$$J = \eta^T \Omega \eta + 2\eta^T \Psi\, x(k) + \sum_{m=1}^{N_p} x(k)^T \left(A^T \right)^p Q A^p x(k), \tag{3-43}$$

where Ω and Ψ are matrices corresponding to the past and present information available for the system, respectively. These parameters are defined as:

$$\Omega = \sum_{m=1}^{N_p} \phi(p) Q \phi(p)^T + R_L \;; \quad \Psi = \sum_{m=1}^{N_p} \phi(p) Q A^p .$$

Without constraints, the optimal solution of the parameter vector is then calculated as:

$$\eta = -\Omega^{-1} \Psi\, x(k). \tag{3-44}$$

In the case of receding horizon control, the optimal incremental control is calculated as:

$$\Delta u(k) = \begin{bmatrix} L_1(0)^T & o_2^T & \cdots & o_p^T \\ o_1^T & L_2(0)^T & \cdots & o_p^T \\ \vdots & \vdots & \ddots & \vdots \\ o_1^T & o_2^T & \cdots & L_p(0)^T \end{bmatrix} \eta \tag{3-45}$$

where o_k^T is a zero-valued row vector with the same dimension as $L_k(0)^T$.

From this discussion it is important to note that the use of a control horizon no longer forms a part of the design [6]. This means that the optimal control signal at each sampling instance is calculated over the prediction horizon, and not for a control horizon. This greatly reduces the computational load of implementing the MPC controller.

In the case where constraints need to be incorporated into the design of the MPC controller, the same method can be followed to obtain the constraints in the same form as in (3-25). The constraints are approximated with the Laguerre functions and used in conjunction with the cost function to solve the optimal control problem. As discussed previously this formulation of the problem will result in a quadratic programming problem.

In the case of constraints on the incremental control, the Laguerre networks can be used to approximate the constraints as [6]:

$$\Delta u^{\min} \le \begin{bmatrix} L_1(p)^T & o_2^T & \cdots & o_p^T \\ o_1^T & L_2(p)^T & \cdots & o_p^T \\ \vdots & \vdots & \ddots & \vdots \\ o_1^T & o_2^T & \cdots & L_p(p)^T \end{bmatrix} \eta \le \Delta u^{\max}, \tag{3-46}$$

which is directly related to (3-45).

In the case of constraints on the magnitude of the control signal, the Laguerre networks can be used to approximate the constraints as [6]:

$$u^{\min} \le \begin{bmatrix} \sum_{i=0}^{k-1} L_1(i)^T & o_2^T & \cdots & o_p^T \\ o_1^T & \sum_{i=0}^{k-1} L_2(i)^T & \cdots & o_p^T \\ \vdots & \vdots & \ddots & \vdots \\ o_1^T & o_2^T & \cdots & \sum_{i=0}^{k-1} L_p(i)^T \end{bmatrix} \eta + u(k-1) \le u^{\max}, \tag{3-47}$$

where the constraints are expressed as the results of the current control action and the control action at the previous sampling instant.

3.3.2 Prescribed degree of stability and exponential data weighting

Two more functions that can be added to the design of the MPC controller to optimize the effectiveness of the controller are: adding a prescribed degree of stability, and exponential data weighting.

The principle of adding a prescribed degree of stability to the design is that a circle with radius $0<\lambda<1$ is defined within the unit circle. If all closed-loop system poles are located within the radius of λ then the system will have a prescribed degree of stability [6]. The principle of adding exponential data weighting works by multiplying the cost function with an exponential function that decreases exponentially as the horizon increases. This leads to better numerical conditioning in systems with large prediction horizons [6].

These functions are added to the design of MPC using Laguerre functions by altering the performance parameters Q and R in the cost function of (3-42) before the Laguerre functions are incorporated into the cost function. This is done by defining the ratio γ between the prescribed degree of stability λ and the exponential data weight α as:

$$\gamma = \frac{\lambda}{\alpha}. \tag{3-48}$$

The performance parameters Q and R are then altered as [6]:

$$\begin{aligned} Q_\alpha &= \gamma^2 Q + (1-\gamma^2)P_\infty \\ R_\alpha &= \gamma^2 R \end{aligned} \tag{3-49}$$

where P_∞ is the solution of the Riccati equation. The new performance parameters Q_α and R_α are now used in the cost function to determine the optimal control.

In the case where constraints are used in the design of the MPC controller the exponential data weights are also implemented in the constraints as [6]:

$$\alpha^{-m}\Delta u^{\min} \le L(m)^T \eta \le \alpha^{-m}\Delta u^{\max} \tag{3-50}$$

$$u^{\min} \le u(k-1) + \sum_{j=1}^{m} \alpha^j L(j)^T \eta \le u^{\max}. \tag{3-51}$$

From this design of MPC using Laguerre functions with a prescribed degree of stability and exponential data weighting, it is clear that many tuning parameters have been incorporated into the controller. For the control of the FlyUPS these tuning parameters can be used to obtain a desired system response.

3.4 MATLAB® Model Predictive Control toolbox™

This section covers the theory of MPC controllers for MIMO systems, as used in the *MATLAB® Model Predictive Control toolbox™* (*MPC toolbox™*). It should be noted that this section will not discuss all of the functions of the *MPC toolbox™*; only the functions applicable to the design of the MPC controller for the FlyUPS will be discussed.

An advantage of using the *MPC toolbox™* is that MPC controllers can be designed and implemented with relative ease. Also, with insight into the working of MPC, the parameters of the controller can be tuned to obtain a desirable response from the system [9]. The development of the MPC controller using the *MPC toolbox™* is discussed according to the model, observers, objective function and predictive control used in the design of the MPC controller for the FlyUPS.

3.4.1 Models used to develop the MPC controller

The model of the FlyUPS discussed in section 2.5 is used to design the MPC controller. However, in order to improve the accuracy of the predictions, the effects of disturbances must also be included. This is done by including a model of the unmeasured disturbances and measurement noise in the design.

The model of the FlyUPS can be expressed as a linear time-invariant model in state-space form as [9]:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) B_v v(k) + B_d d(k) \\ y(k) &= C_m x(k) + D_v v(k) + D_d d(k) \end{aligned} \tag{3-52}$$

where $x(k)$ is an n_x-dimensional vector of states, $u(k)$ is an n_u-dimensional vector of control variables, $v(k)$ is an n_v-dimensional vector of measured disturbances, $d(k)$ is an n_d-dimensional vector of unmeasured disturbances. $y(k)$ represents an n_y-dimensional vector of measured outputs.

In (3-52) the unmeasured disturbance $d(k)$ includes the state and output disturbances. These disturbances can be modeled as [9]:

$$\begin{aligned} x_d(k+1) &= \bar{A}x(k) + \bar{B}n_d(k) \\ d(k) &= \bar{C}x_d(k) + \bar{D}n_d(k) \end{aligned} \tag{3-53}$$

where $n_d(k)$ is assumed to be random zero-mean unit-variance white Gaussian noise. The plant model is augmented with a model of the output disturbances in order to reject constant disturbances on measured outputs. This adds the integration effect to the design of the MPC controller.

The final disturbance that can be included in the design is the effect of measurement noise $n_m(k)$. The measurement noise can be modeled as the linear time-invariant (LTI) system driven by random zero-mean unit-variance white Gaussian noise as:

$$\begin{aligned} x_m(k+1) &= \tilde{A}x_m(k) + \tilde{B}n_m(k) \\ m(k) &= \tilde{C}x_m(k) + \tilde{D}n_m(k). \end{aligned} \tag{3-54}$$

By appending the models of the disturbances and noise to the plant model an extended model of the system is obtained. The immeasurable states $x(k)$, $x_d(k)$ and $x_m(k)$ are obtained from the state estimator at each sampling instant.

3.4.2 Observers in the design of the MPC controller

In order to reduce the prediction error the states $x(k)$, $x_d(k)$ and $x_m(k)$ are updated at each sampling instant by using a state observer. A condition for the use of state estimation is that the models used in the design of the MPC controller are observable. In order to check the observability the models are firstly converted to discrete-time models without delays. These models are then combined and numerically checked for observability [9].

Before the states of the system can be estimated, the output of the system must first be estimated from the known parameters as [9]:

$$\hat{y}(k) = C\hat{x}(k \mid k-1) + D_v v(k) + D_d \bar{C}\hat{x}_d(k \mid k-1) + \tilde{C}\hat{x}_m(k \mid k-1). \tag{3-55}$$

As can be seen from (3-55) the estimated output $\hat{y}(k)$ is calculated from the models discussed in the previous section, as well as the states estimated at the previous sampling instant ($\hat{x}(k|k-1)$).

From $\hat{y}(k)$ the states at the current sampling instant can then be calculated as [9]:

$$\begin{bmatrix} \hat{x}(k \mid k) \\ \hat{x}_d(k \mid k) \\ \hat{x}_m(k \mid k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k \mid k-1) \\ \hat{x}_d(k \mid k-1) \\ \hat{x}_m(k \mid k-1) \end{bmatrix} + K_{ob}\left(y(k) - \hat{y}(k)\right), \tag{3-56}$$

where K_{ob} is the observer gain obtained from Kalman filtering techniques. This indicates that the current states are estimated from the states estimated at the previous sampling instant and a correction term. The correction term updates the estimated states based on the error between the measured output $y(k)$ and the estimated output $\hat{y}(k)$. The estimated states from (3-56) are used in the calculation of the optimal control.

The estimated states are also used to estimate the state at the next sampling instant as [9]:

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}_d(k+1|k) \\ \hat{x}_m(k+1|k) \end{bmatrix} = \begin{bmatrix} A\hat{x}(k|k) + B_u u(k) + B_v v(k) + B_d \bar{C}\hat{x}_d(k|k) \\ \bar{A}\hat{x}_d(k|k) \\ \tilde{A}\hat{x}_m(k|k) \end{bmatrix}. \quad (3\text{-}57)$$

The estimates obtained from (3-57) are used at the next sampling period as the estimates of the previous states (*x̂(k|k-1)*) as indicated in (3-56).

3.4.3 The objective function of the MPC controller

Based on the estimates *x(k)*, $x_d(k)$ and $x_m(k)$ as calculated in (3-56), the optimal control action $u_{optimal}$ can be calculated. This is done by solving the optimisation problem [9]:

$$J = \min_{\Delta u(k|k),\ldots,\Delta u(k|k+N_c-1)} \left[\sum_{i=0}^{N_p-1} \left[\begin{array}{c} \sum_{j=1}^{n_y} \left| w_{i+1,j}^{y} \left(y_j(k+i+1) - r_j(k+i+1) \right) \right|^2 \\ + \sum_{j=1}^{n_u} \left| w_{i,j}^{\Delta u} \, \Delta u_j \left(k+1|k \right) \right|^2 \end{array} \right] \right], \quad (3\text{-}58)$$

subject to the constraints

$$\begin{aligned} u_j^{\min}(i) \le u_j(k+i|k) \le u_j^{\max}(i) \\ \Delta u_j^{\min}(i) \le \Delta u_j(k+i|k) \le \Delta u_j^{\max}(i) \\ y_j^{\min}(i) \le y_j(k+i+1|k) \le y_j^{\max}(i) \end{aligned} \quad (3\text{-}59)$$

with $i = 0,1,2,\ldots,(N_p\text{-}1)$ and assuming that the increment in the control from the end of the control horizon to the end of the prediction horizon is $\Delta u(k+N_c|k),\ldots,\Delta u(k+N_p|k) = 0$. This assumption is made in order to reduce the computational load.

As mentioned previously the use of linear inequality constraints with the quadratic cost function leads to a quadratic programming problem. The solution of the quadratic programming problem is the optimal control move $u_{optimal}$. In the unconstrained MPC case $u_{optimal}$ is calculated analytically.

3.4.4 Predictive control design

The cost function given in (3-58) can be written as [9]:

$$J = \left\{ \begin{bmatrix} y(1) \\ \vdots \\ y(N_p-1) \end{bmatrix} - \begin{bmatrix} r(1) \\ \vdots \\ r(N_p-1) \end{bmatrix} \right\}^T W_y^2 \left\{ \begin{bmatrix} y(1) \\ \vdots \\ y(N_p-1) \end{bmatrix} - \begin{bmatrix} r(1) \\ \vdots \\ r(N_p-1) \end{bmatrix} \right\} + \begin{bmatrix} \Delta u(0) \\ \vdots \\ \Delta u(N_p-1) \end{bmatrix}^T W_{\Delta u}^2 \begin{bmatrix} \Delta u(0) \\ \vdots \\ \Delta u(N_p-1) \end{bmatrix} \quad (3\text{-}60)$$

with the weights W_y and $W_{\Delta u}$ representing the weights on the output and the incremental control, respectively. These weights are used to determine the speed and the robustness of the response. For larger values of W_y any deviations between the output and the set-point are penalized to a greater extent. This means that set-points are tracked more stringently. For

larger values of $W_{\Delta u}$ greater care is taken when calculating the incremental control signal. This increases the robustness, but also has the effect that the response is slower.

By substituting *u(k), Δu(k)* and *y(k)* in (3-60) the cost function can be written in terms of the extended system model as [9]:

$$J = \Delta u^T K_{\Delta u} \Delta u + 2\left\{ \begin{bmatrix} r(1) \\ \vdots \\ r(N_p) \end{bmatrix}^T K_r + \begin{bmatrix} v(0) \\ \vdots \\ v(N_p) \end{bmatrix}^T K_v + u(-1)^T K_u + x(0)^T K_x \right\} \Delta u + const. \qquad (3\text{-}61)$$

In (3-61) $K_{\Delta u}$ represents the Hessian matrix. K_r, K_v, K_u and K_x are parameters relating the cost to the set-point over the prediction horizon, the measured disturbance over the prediction horizon, previous control actions and the initial states of the system, respectively.

In the case of unconstrained MPC the optimal incremental control action is calculated analytically using the following equation [9]:

$$\Delta u_{optimal} = -K_{\Delta u}^{-1} \left\{ \begin{bmatrix} r(1) \\ \vdots \\ r(N_p) \end{bmatrix}^T K_r + \begin{bmatrix} v(0) \\ \vdots \\ v(N_p) \end{bmatrix}^T K_v + u(-1)^T K_u + x(0)^T K_x \right\}^T . \qquad (3\text{-}62)$$

From (3-62) the optimal control action $u_{optimal}$ can then be calculated as:

$$u_{optimal}(k) = u(k-1) + \Delta u_{optimal}(k) . \qquad (3\text{-}63)$$

In the case of constrained MPC the constraints are written in a similar form as (3-25). By relating the constraints on *u(k)*, *Δu(k)* and *y(k)* to the optimal control signal, the following inequality is obtained [9]:

$$M_{\Delta u} \Delta u \le M_{\lim} + M_v \begin{bmatrix} v(0) \\ \vdots \\ v(N_p) \end{bmatrix} + M_u u(-1) + M_x x(0) . \qquad (3\text{-}64)$$

The parameters of the inequality presented in (3-64) are then used in the quadratic programming problem to solve the optimal control action. The gains used to relate the constraints to the optimal control are $M_{\Delta u}$ that relates to the incremental control, M_{lim} that relates to the values of the constraints, M_v that relates to the modeled noise, M_u that relates to the past control actions, and M_x that relates to the past responses of the plant.

3.5 Conclusion

This chapter has discussed the design of MPC controllers for the FlyUPS. Two algorithms were discussed, namely the use of a classical MPC algorithm that incorporates optimisation techniques, and the MPC algorithm that is used by the MATLAB® *Model Predictive Control toolbox*™.

In the next chapter the above mentioned MPC controllers are implemented in simulation. These simulations give a better understanding of the parameters used in the design of the MPC controller such as the lengths of horizons, the selection of weights, and the effects of adding constraints to the design of the controller.

4

Chapter

Controller simulation

From the discussion on the design of the MPC controllers in the previous chapter MPC is now implemented in simulation in order to verify the effectiveness of the controller. The MPC controllers that are simulated are the MPC algorithm using optimization techniques, and the MPC controller derived using the MATLAB® Model Predictive Control toolbox™. Each MPC controller is discussed according to the setup of the simulation, the results of the simulations and a discussion of the results.

4.1 Classical MPC using optimization techniques

In section 3.3 the use of some optimization techniques in the design of an MPC controller were discussed. These techniques involve incorporating Laguerre functions, a prescribed degree of stability, and exponential data weighting into the design of a classical MPC algorithm. In this section the design of a classical MPC controller using these optimization techniques is implemented in simulation in order to verify this design for use on the FlyUPS.

4.1.1 Simulation setup

In order to simulate a classical MPC controller using optimization techniques the MPC algorithm, as discussed in section 3.3, is implemented in the MATLAB® simulation package. For this discussion the implementation of the MPC controller is done according to the program flowchart presented in Figure 4-1.

As illustrated in Figure 4-1, at the start of the program the plant model and constraints are approximated using the Laguerre functions. Then, based on these Laguerre functions and the desired set-point signal, the cost function parameters are calculated. At this step the exponential data weighting and the prescribed degree of stability are also used to calculate the cost function parameters. The parameters calculated at this step remain the same for the duration of the simulation as the model is assumed to be static for a predetermined operating speed.

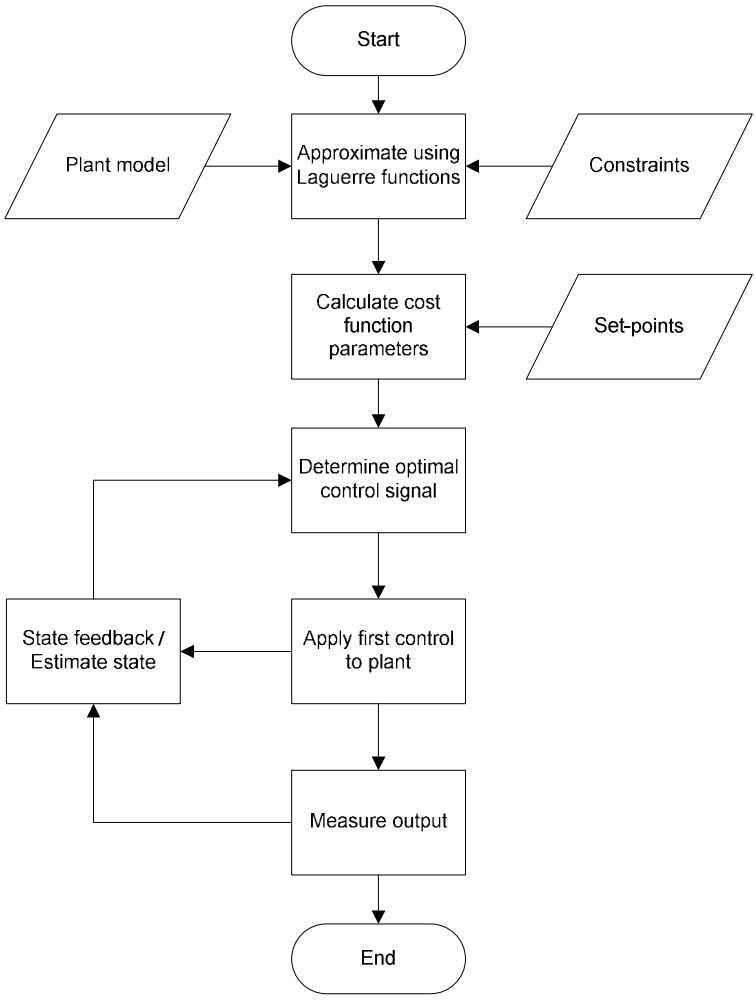

Figure 4-1 Program flowchart for MPC using optimization techniques

The cost function parameters are then used to determine the optimal control signal. For the first iteration of the control process the optimal control signal is determined over the prediction horizon using the cost function parameters and initial values for the states of this system. From this optimal control signal the first value is then applied to the plant. The response of the plant to the control value (the measured output) is then used to update the states of the system. These estimated states along with the cost function parameters are then used to determine the optimal control signal at the next sampling instance. This process is repeated at each sampling instance for the duration of the simulation.

The code according to the program flowchart in Figure 4-1 is implemented in MATLAB® and is available in *Appendix A*. This program is based on the implementation of MPC as discussed by Wang [6]. The model of the FlyUPS used for this simulation is discussed in section 2.5. As this model can be decoupled into the axial and radial models of the FlyUPS, this simulation focuses on the axial model. If the simulation is successful for the axial AMB the simulation will be expanded to include the radial AMBs.

In order to reduce the computational load and the execution time this model is reduced using the MATLAB® *hankelmr*™ command. This command determines the Hankel singular values of the system as given in Figure 4-2. This figure shows the seven states of the axial AMB model, as well as the corresponding state energy. From Figure 4-2 the model of the axial AMB of the FlyUPS is reduced to three states as most of the state energy and thus the dynamics of the system are described by these three states.

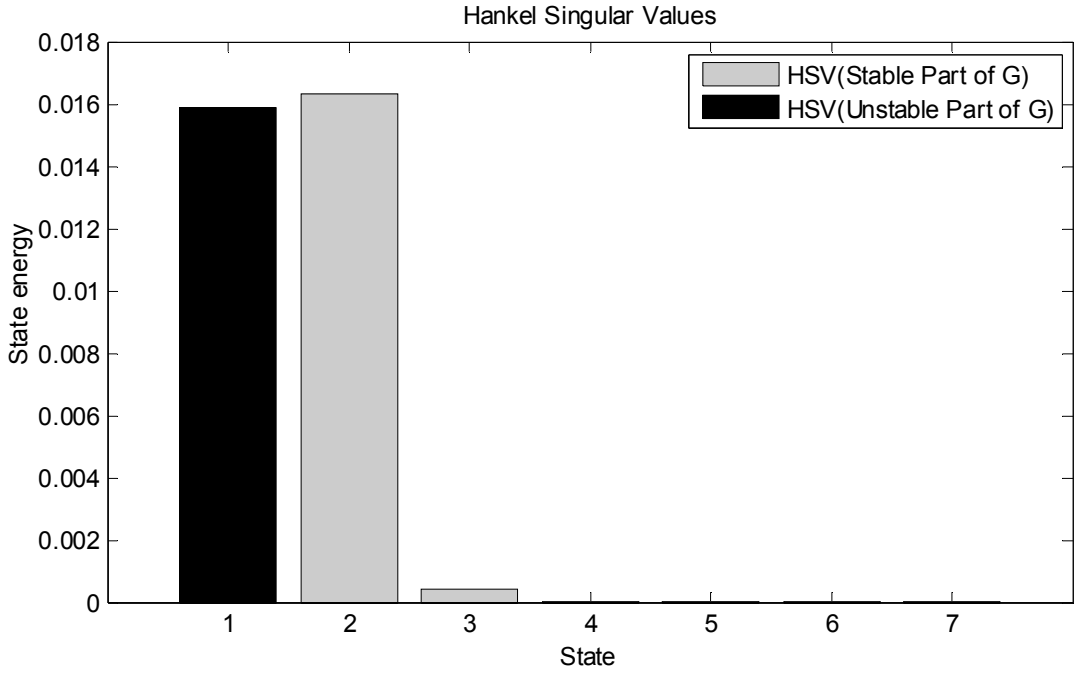

Figure 4-2 Hankel singular values of axial AMB model

The values of the parameters used for this simulation are given in Table 4-1. These values are chosen according to guidelines proposed by Wang [6], as well as trial-and-error during simulations. The reason for this is that the above mentioned guidelines only provide a means to reach an approximate desired response. Also, the response of the system changes depending on combinations of parameter values. This means that trial-and-error methods may be necessary to tune the controller parameters to obtain the optimal system response.

The process of selecting the parameter values as given in Table 4-1 can be summarised as follows. The prediction horizon is initially chosen as a large value, e.g. $N_p = 50$, however as this value has a large effect on the execution time of the control, N_p is decreased until the response of the system starts to deteriorate.

As mentioned previously, the values of the Laguerre parameters are chosen according to the guidelines given by Wang [6]. These guidelines state that for a smaller value of the pole position (with $0<a<1$) the response of the system is slower, but more accurate. For a larger value of *a*

the response of the system is faster, but results in a larger overshoot. This effect is similar to that of the integral term in conventional PID control.

Table 4-1 Parameter values used in the simulation of MPC using optimization techniques

Parameter	**Value**
Sampling frequency (f_s)	10 kHz
Prediction horizon (N_p)	2 samples
Laguerre parameters	
Position of pole (a)	0.9
Number of approximation terms (N)	3
Performance parameters	
Q	$0.85(C^TC)$
R	0.15
Optimization parameters	
Prescribed degree of stability (λ)	0.999
Exponential weighting (α)	1.3

The number of terms used to approximate the model determines the accuracy of the response. This means that a larger number of terms will yield more accurate results. However, as the Laguerre approximation of the model can be written as an expansion series of Laguerre functions and coefficients, with the coefficients becoming exponentially smaller as the series progresses, the approximation becomes fairly accurate after four to five terms, depending on the complexity of the model [6].

According to Wang [6] and Maciejowski [5] the choice of the performance parameters also takes some consideration. This is due to the fact that any choice of Q or R as well as the combination of the performance parameters can change the response of the system. For the purpose of set-point tracking the value of Q is usually chosen to be C^TC and the value of R is chosen to be 0.1 [6,5]. During simulation the values of Q and R are seen to be most effective when chosen as given in Table 4-1. However, these parameters are also seen to affect the settling time and stead-state error. For this simulation the values of Q and R are chosen to ensure a fast response.

For the choice of optimization parameters given in Table 4-1 careful consideration is also necessary. According to Wang [6] the choice of the prescribed degree of stability (where $\lambda < 1$) is chosen to ensure the stability of the response. However, it is seen that the value of λ cannot be chosen smaller than 0.9 as this causes the control problem to become unstable.

Exponential weighting ($\alpha \geq 1$) determines how fast the response converges toward the set-point, i.e. the larger α is, the faster the set-point can be reached. However, in simulation it is seen that for $1 \leq \alpha \leq 1.1$ the response of the system is unstable to marginally stable, while for $\alpha > 1.3$ no appreciable gain in performance can be seen.

For this simulation only hard constraints are incorporated into the design of the MPC controller. This is done by setting limits for the control (u) and incremental control (Δu) as can be seen in *appendix A.4*. If the calculated u or Δu is greater or less than these limits, then the value is taken to be equal to the limit. By setting these hard constraints and excluding the soft constraints on the output the need to implement a quadratic programming strategy is avoided. Also shown in *appendix A.4* is the use of a state feedback variable which is updated from the current calculation of the optimal control signal at each sampling instance.

4.1.2 Results for simulation of MPC using optimization techniques

The results for this simulation of the classic MPC controller using optimization techniques are presented in Figure 4-3.

Figure 4-3 shows the position output of the axial AMB, as well as the control and incremental control signals as determined by the simulation program discussed in the previous section. These results are obtained for a step reference change of 10 μm. The set-point change is applied at the first sampling instance. The simulation is then run for 2,000 samples.

From the output signal displayed in Figure 4-3 it is seen that the position of the rotor settles at a value of 6 μm after 819 samples. This relates to a steady state error of 40 percent. From the output it is also seen that the response has no overshoot, which means that the response is over damped. The rise time is determined as approximately 656 samples (or 0.0656 s at a sampling frequency of 10 kHz). A delay in the response is seen where the output changes by no more than 0.5 μm for the first 40 samples.

From the control and incremental control signals in Figure 4-3 it is seen that there are no constraints that become active at any point in the simulation. At the beginning of the simulation the control signal reacts rapidly to the change in the set-point signal, after which this control action is regulated to achieve a steady state value in the output with no overshoot. This result corresponds to the choice of the performance parameters that are chosen specifically to facilitate set-point tracking.

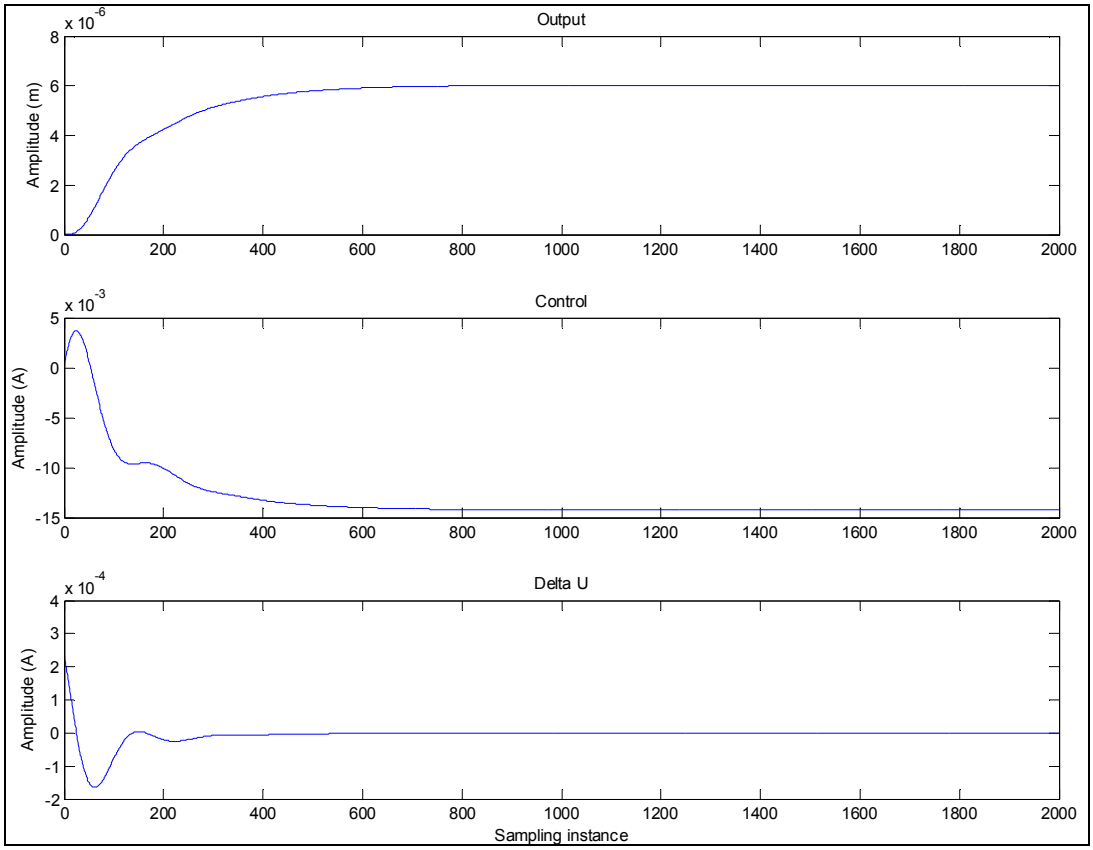

Figure 4-3 Results for simulation of MPC using optimization techniques

4.1.3 Discussion of results for MPC using optimization techniques

From the results presented in the previous section it is seen that a classical MPC algorithm using optimization techniques may be effective in controlling a single degree of freedom AMB. However, these results also raise some concerns when considering the implementation of this MPC algorithm on the physical FlyUPS system.

The first concern is the steady-state error of the system. In this simulation this error is determined as 40 percent. During simulations this error alters greatly as the performance parameters are changed. This means that choosing the values of parameters becomes more difficult than following some guidelines and using trial-and-error to tune the response. Careful consideration must be given to the effect of each parameter as well as the effect of the parameters when altering more than one parameter.

The second concern is the rise time and settling time of this MPC algorithm. For this simulation a sampling time of 10 kHz was chosen which would be sufficient for implementation in the physical FlyUPS system. From this simulation the rise time and settling time of this MPC algorithm are 0.0656 s and 0.0819 s, respectively. Considering the application of this algorithm

on the FlyUPS system this rise time and settling time may be too slow to implement real-time control.

Based on these issues the classical MPC algorithm using optimization techniques is not considered as a viable control strategy for implementation on the FlyUPS.

4.2 MPC using the MATLAB® Model Predictive Control toolbox™

This section determines the effectiveness of MPC for the FlyUPS system when the MPC algorithm used in the MATLAB® *Model Predictive Control toolbox*™ is implemented in simulation. This discussion follows the same method as the previous section in order to reach a conclusion on the validity of this MPC algorithm.

4.2.1 Simulation setup

From the discussion of the MPC algorithm in section 3.4 an MPC controller for the FlyUPS is developed and implemented in simulation. The parameters that are necessary for the design of an MPC controller using the MATLAB® *MPC toolbox*™ are presented in Table 4-2.

Table 4-2 MATLAB® *MPC toolbox*™ parameters

Parameter class	**Parameter**
Model and horizons	Plant model
	Control interval (T)
	Prediction horizon (N_p)
	Control horizon (N_c)
Constraints	Constraints on control (u_{min}, u_{max})
	Constraints on incremental control (Δu_{min}, Δu_{max})
	Constraints on output (y_{min}, y_{max})
Weight tuning	Gain (K_w)
	Control weight (W_u)
	Incremental control weight ($W_{\Delta u}$)
	Output weight (W_y)
Estimation	Overall estimator gain (K_{ob})
	Output disturbance gain ($K_{out.dist}$)
	Measurement noise gain ($K_{meas.noise}$)

In order to design the MPC controller, the model of the system is specified first. For this simulation the model of the FlyUPS discussed in section 2.5 is used. As this model is a thirty-fifth order model, order reduction of the model is done before it is used in the design of the MPC

controller in order to reduce the computational load and the execution time. The model order reduction is done based on the Hankel singular value decomposition given in Figure 4-4. From Figure 4-4 it is evident that the model of the FlyUPS can be reduced to a tenth order model.

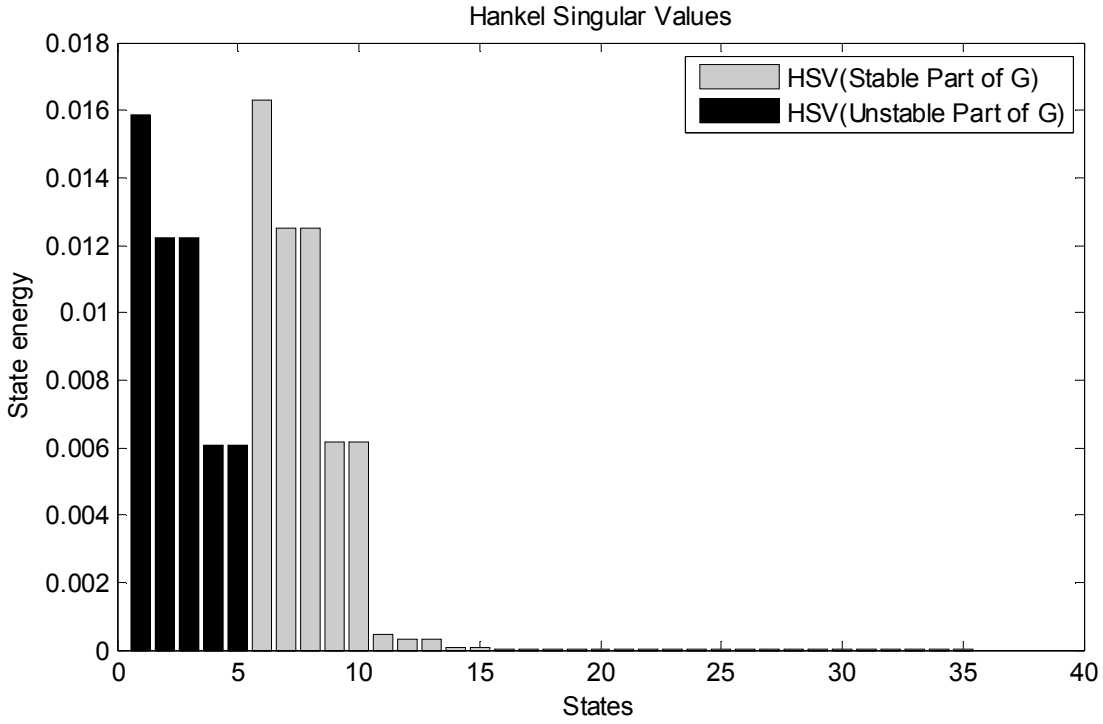

Figure 4-4 Hankel singular value decomposition of FlyUPS model

As the goal of this project is to implement discrete-time model predictive control on the FlyUPS, a sample time or control interval must be specified. For this application T_s is chosen to be 200 µs. This relates to a sampling frequency of 5 kHz which is sufficient for implementation on the FlyUPS.

The choices of the other parameters needed in the design of an MPC controller are chosen according to the guidelines presented in [5,6,9]. As noted previously these guidelines are only a starting point for the design of an MPC controller. The desired system response is obtained by tuning these parameters and investigating the effects of each parameter. The guidelines for choosing the parameter values are discussed in the following paragraphs.

For a non-minimum phase system such as the FlyUPS that uses AMBs, the prediction horizon must be chosen large enough to focus on the long-term response of the system [9]. If the prediction horizon is chosen too small, the control action is determined according to the short-term response of the system which, for a non-minimum phase system, may be in the wrong direction. The control horizon is then chosen as approximately an order lower than the prediction horizon.

Constraints related to the physical system can be identified and applied to the simulation. Control and incremental control constraints are usually hard constraints [5], e.g. the maximum current that can be applied by the power amplifiers and the slew rate of the power amplifiers. Output constraints are usually soft constraints such as the maximum desired position error of the backup bearings [5].

The use of constraints in the design of the MPC controller is one of the major advantages of using MPC. However, by incorporating constraints in the design of the controller the computational load is increased as the optimal control action is determined by solving a quadratic programming problem at each sampling instance. This will be investigated further during the implementation of the MPC controller.

The weight tuning is done in order to achieve either more robustness or a faster response. The gain weight ($0 \leq K_w \leq 1$) primarily determines the robustness and speed of the response, where a smaller K_w results in a more robust response and a larger K_w results in a faster response [9]. The control weight is used to determine the importance of input set-point tracking. However for this application this function will not be used.

The incremental control weight is used to determine how much consideration is used to determine the incremental control signal [9], e.g. a larger value of $W_{\Delta u}$ results in a longer execution time but reduces the control effort needed. The output weight is used to determine how important it is for a certain output of the system to accurately track a set-point [9], e.g. for a larger W_y the rise time and overshoot may be reduced at the cost of a large control effort.

An observer based on Kalman filtering techniques is used to determine the states of the plant model and disturbance models [9]. The estimator gain is used to determine the gain of the observer. The output disturbance gain is used to determine the affect of the integrator mode on the output of the system. As the measurement noise model is not used in the prediction model needed for optimization it will not be included in the design of this MPC controller.

From this discussion on the parameters of the MPC controller developed using the MATLAB® *MPC toolbox*™, a controller for the FlyUPS is derived and implemented in a Simulink™ simulation. This simulation model is presented in Figure 4-5 and shows the closed-loop simulation of the FlyUPS with the full thirty-fifth order state-space model as discussed in section 2.5.

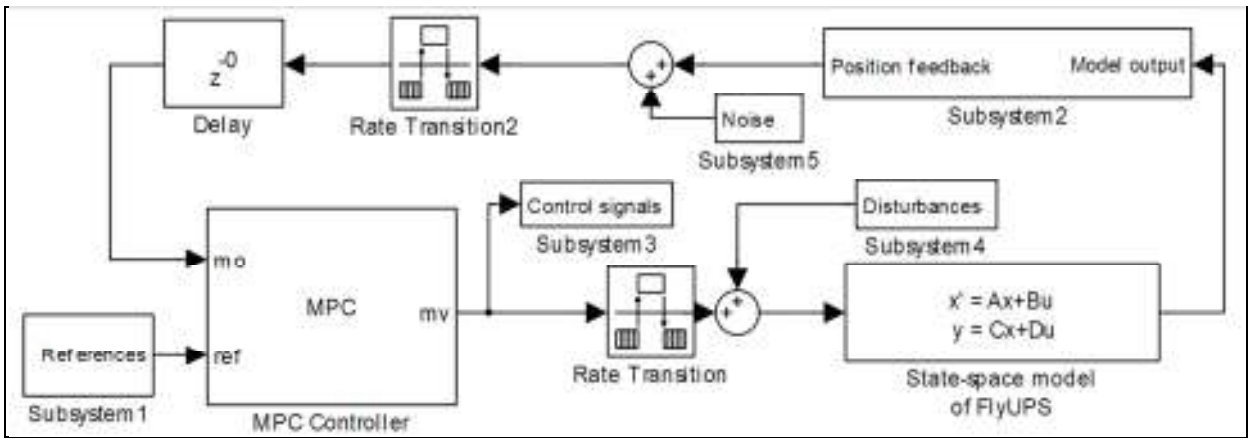

Figure 4-5 Simulink model for simulation of MPC developed using MATLAB® *MPC toolbox*™

For simulation purposes the simulation model in Figure 4-5 contains subsystems which are used contain important parts of the simulation. Subsystem1 contains references for each input. Subsystem2 is used to log data and to select the position outputs for feedback purposes. Similarly, Subsystem3 is used to log the control signal data.

The effects of disturbances and measurement noise are included in the simulation with Subsystem4 and Subsystem5, respectively. The disturbances represent external disturbance forces on the rotor. The measurement noise is assumed to be random white Gaussian noise with unit-variance. In the feedback path a delay can also be included in order to simulate the effects of delays on the response of the system.

As each AMB, i.e. input-output pair, of the system can be approximated as a second order transfer function some performance characteristics of each AMB can be determined from the step response of each AMB. These performance characteristics are the percentage overshoot (*P.O.*), settling time (T_s) and rise time (T_r) [25]. From these measurable characteristics the damping ratio (ζ), natural frequency (ω_n) and time constant (τ) are calculated as [25]:

$$P.O. = 100e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{4-1}$$

$$\omega_n = \frac{4}{\zeta T_s} \tag{4-2}$$

$$\tau = \frac{1}{\zeta\,\omega_n}. \tag{4-3}$$

From these variables the equivalent stiffness (k_{eq}) and damping (b_{eq}) are then calculated as [1]:

$$k_{eq} = \omega_n{}^2 m \tag{4-4}$$

$$b_{eq} = \zeta\left(2\sqrt{k_{eq}\,m}\right) \tag{4-5}$$

with m the equivalent mass at the position of the AMB.

4.2.2 Results for simulation using the MATLAB® MPC toolbox™

In order to determine the values for the parameters, the behaviour of the system must first be determined for different values and combinations of the parameters. This is done by varying the parameters under investigation and keeping all other values constant. In order to avoid redundancy, only the results of this investigation for the axial AMB will be given. However, these results can easily be extended to the radial AMBs.

Table 4-3 shows the simulation results when a step reference change is applied to the axial AMB of the FlyUPS. The results given in Table 4-3 focus on the changes in N_c and N_p, with the values of the weights and estimation parameters held constant.

Table 4-3 Simulation results for step response on axial AMB

N_c (samples)	N_p (samples)	*P.O.* (%)	T_s (s)	T_r (s)	k_{eq} (N/m)	b_{eq} (N.s/m)	τ(s)
2	20	Unstable step response					
	30	4.4	0.0148	0.0096	2593958.524	9540.541	0.0037
	40	3.2	0.0164	0.0112	1933045.251	8609.756	0.0041
	50	3.1	0.016	0.011	2014472.243	8825	0.0040
	60	2.5	0.014	0.0104	2501065.521	10085.714	0.0035
	70	0.8	0.0095	0.0102	4361648.477	14863.158	0.0024
	80	Over damped	0.0101	0.0057			
3	20	Unstable step response					
	30	8.9	0.0201	0.0075	1878507.388	7024.876	0.0050
	40	8.6	0.0199	0.0076	1879304.334	7095.477	0.0050
	50	6.1	0.0177	0.007	2038331.749	7977.401	0.0044
	60	1.6	0.0154	0.0072	1884035.117	9168.831	0.0039
	70	Over damped	0.0144	0.0042			
4	20	Unstable step response					
	30	13.2	0.0185	0.0064	2808811.939	7632.432	0.0046
	40	11.5	0.0171	0.006	3004044.445	8257.310	0.0043
	50	5.7	0.0149	0.0058	2800092.398	9476.510	0.0037
	60	Over damped	0.0135	0.0036			
5	20	Unstable step response					
	30	16.7	0.0171	0.0056	3941504.927	8257.309942	0.0043
	40	12.7	0.015	0.0054	4164256.625	9413.333333	0.0038
	50	4.7	0.0146	0.0054	2727052.034	9671.232877	0.0037
	60	Over damped	0.0129	0.0034			

From the results in Table 4-3 it is seen that as N_c increases the equivalent stiffness of the system also increases. It is also clear that as N_p increases the equivalent damping also increases, along with a decrease in τ. For larger values of both N_c and N_p the overall performance of the system is improved with lower *P.O.* and faster T_r and T_s. As damping increases the effect of noise on the response of the system the case where the response becomes overdamped will be taken as the maximum allowable values [1]. For larger values of N_c and N_p than presented in Table 4-3 no appreciable gains in performance are seen.

However, as given in Table 4-4 the increase in the performance when using larger parameters results in a larger control effort during the natural response of the system. From the results in Table 4-4 it is clear how the MPC controller responds to the non-minimum phase characteristics of the FlyUPS with a large control effort in the one direction at the beginning of the response and a control effort in the opposite direction for the rest of the natural and transient response.

Table 4-4 Simulation results for control effort of MPC for varying N_c and N_p

N_c (samples)	N_p (samples)	Control effort {max; min; transient} (A)
2	30	{0.0375; -0.03474; -0.02367}
	40	{0.0291; -0.03106; -0.02367}
	50	{0.03029; -0.03142; -0.02367}
	60	{0.03612; -0.0329; -0.02367}
	70	{0.04443; -0.03439; -0.02367}
3	30	{0.04548; -0.04339; -0.02367}
	40	{0.0439; -0.04363; -0.02367}
	50	{0.05674; -0.04638; -0.02367}
	60	{0.06978; -.0.4919; -0.02367}
4	30	{0.05444; -0.05387; -0.02367}
	40	{0.06258; -0.05779; -0.02367}
	50	{0.07821; -0.06136; -0.02367}
5	30	{0.06133; -0.06417; -0.02367}
	40	{0.07554; -0.06981; -0.02367}
	50	{0.0914; -0.07078; -0.02367}

By using the same method as described above, the effects of the weighting parameters are also investigated. This is done by keeping $N_c = 5$, $N_p = 40$ and $W_y = 8$, and altering $W_{\Delta u}$ in order to determine the effect of $W_{\Delta u}$ on the response of the system. The result for the step response of the system with a varying $W_{\Delta u}$ is given in Table 4-5. As the fastest response is required only $K_w = 1$ is considered during simulations.

Table 4-5 Results of step response for varying $W_{\Delta u}$

$W_{\Delta u}$	*P.O.* (%)	T_s (s)	T_r (s)	k_{eq} (N/m)	b_{eq} (N.s/m)	τ(s)	Control effort {max; min; transient} (A)
0.1	2.1	0.0138	0.0148	2475281.513	10231.884	0.0035	{0.01416; -0.02852; -0.02367}
0.05	5	0.0109	0.0043	4992447.597	12954.128	0.0027	{0.1549; -0.104; -0.02367}
0.01	12.7	0.015	0.0054	4164256.625	9413.333	0.0038	{0.07554; -0.06981; -0.02367}
0.005	5	0.0109	0.0043	4992447.597	12954.128	0.0027	{0.1476; -0.104; -0.02367}
0.001	0.8	0.0084	0.004	5578780.826	16809.524	0.0021	{0.3264; -0.1425; -0.02367}
0.0005	9.3	0.0073	0.0026	14574187.34	19342.466	0.0018	{0.4715; -0.236; -0.02367}
0.0001	43.2	0.0068	0.0013	88294849.37	20764.706	0.0017	{2.422; -1.207; -0.02367}

From the results in Table 4-5 it can be seen that as $W_{\Delta u}$ decreases, the equivalent stiffness and damping of the system show an increase. However, the damping does not increase as rapidly as the stiffness. Also, for a decrease in $W_{\Delta u}$ a decrease in T_s, T_r and τ can be seen at the expense of larger *P.O.* and control effort during the natural response.

The effects of varying W_y are investigated by choosing a constant value for $W_{\Delta u}$, with N_c and N_p as chosen above. The results for this scenario can be seen in Table 4-6. These results indicate that as W_y increases the damping of the system also increases, however no fixed pattern can be deduced for the stiffness. The overall performance the system is better with an increase in W_y, with lower *P.O.*, faster T_s and T_r, and a decrease in τ. However, this increase in performance can once again only be gained at the expense of an increase in the control effort during the natural response.

Table 4-6 Results for step response with varying W_y

W_y	*P.O.* (%)	T_s (s)	T_r (s)	k_{eq} (N/m)	b_{eq} (N.s/m)	τ(s)	Control effort {max; min; transient} (A)
1	1.7	0.0138	0.0149	2370028.413	10231.884	0.0035	{0.01488; -0.0283; -0.02367}
2	3.9	0.018	0.0122	1690716.51	7844.444	0.0045	{0.01922; -0.0308; -0.02367}
4	10.2	0.0222	0.0082	1657311.299	6360.36	0.0056	{0.03384; -0.0418; -0.02367}
8	12.7	0.015	0.0054	4164256.625	9413.333	0.0038	{0.07554; -0.06981; -0.02367}
10	11	0.0135	0.0049	4686638.609	10459.259	0.0034	{0.09806; -0.08179; -0.02367}
12	8.8	0.0123	0.0046	4983696.944	11479.675	0.0031	{0.1183; -0.09146; -0.02367}
14	6.8	0.0115	0.0044	505408216.9	12278.261	0.0029	{0.1369; -0.09871; -0.02367}
18	3.6	0.0105	0.0043	4859737.964	13447.6191	0.0026	{0.17; -0.1078; -0.02367}
20	2.5	0.0102	0.0043	4711734.354	13843.1371	0.0026	{0.1824; -0.1106; -0.02367}

In the case of the estimator gains the effects of the parameters are not clear for implementation of a step response. As a fast response is required the value of K_{ob} is chosen as one. As discussed previously, $K_{out.dist}$ adds an integration mode to the controller. However, the effect of $K_{out.dist}$ cannot be indicated clearly when simulating the step response of the FlyUPS as the simulation does not have a steady-state error when $K_{ob} = 1$.

Thus far this discussion has been focused on the effect of the *MPC toolbox*™ parameters on the response of a SISO AMB, or the axial AMB of the FlyUPS. An advantage of using MPC is that this discussion can also be extended to the MIMO FlyUPS system including the radial AMBs. The effects discussed previously are also applicable to the radial AMBs of the FlyUPS. However, an adjustment must be made.

This adjustment is due to the equivalent weight at each AMB. Figure 4-6 shows the physical characteristics of the FlyUPS as determined in [2]. From these characteristics it is clear that the top radial AMB is closer to the centre of mass than the bottom radial AMB. Thus, the equivalent mass at the top radial AMB is much larger than at the bottom radial AMB during rotation. For this reason the top radial AMB will require much larger control effort to maintain stability during rotation [1]. This is achieved by changing the weights of the cost function.

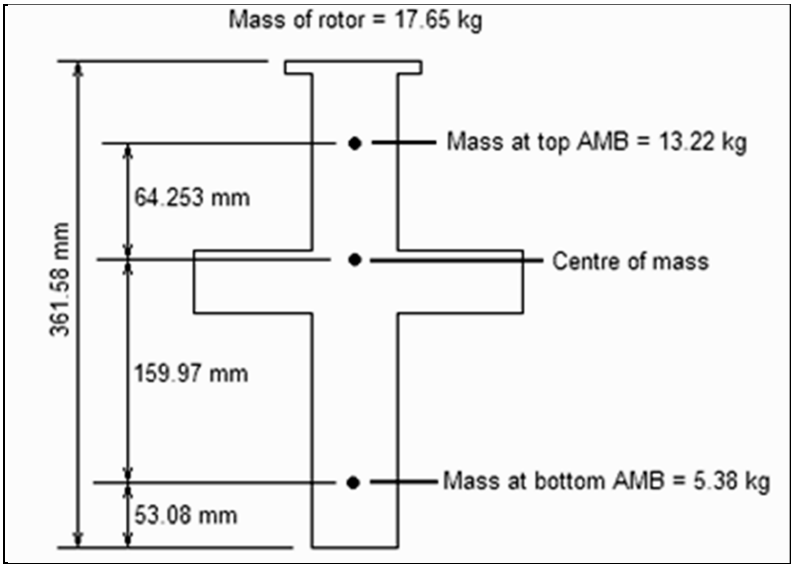

Figure 4-6 Physical characteristics of the FlyUPS [2]

From this investigation into the effects of the parameters, an MPC controller is developed for the FlyUPS. The values for the parameters used in the simulation are given in Table 4-7. These values are chosen to give the best possible performance while keeping the control effort low. The above mentioned adjustment due to the equivalent mass at each AMB can be seen in the

values of W_y where the weight at the axial AMB is eight, the weights at the top radial AMBs are both five, and the weights at the bottom radial AMBs are both three.

Table 4-7 Parameter values for MPC controller developed with the *MPC toolbox*™

Parameter	Value
N_p	50
N_c	5
$W_{\Delta u}$	{0.001; 0.01; 0.01; 0.01; 0.01}
W_y	{8; 5; 5; 3; 3}
$K_{out.dist}$	{1; 1; 1; 1; 1}

The results for the simulation of a step response in the axial AMB are given in Figure 4-7. The characteristics of this response for the axial AMB of the FlyUPS are already indicated in Table 4-3 and Table 4-4. As seen in Figure 4-7 a large control effort is required to overcome the non-minimum phase effects of the AMB, which corresponds to a short delay in the output.

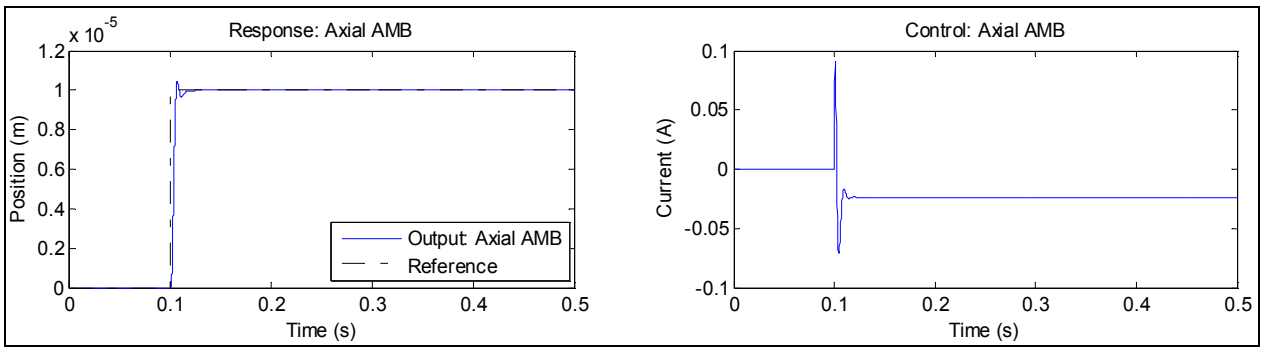

Figure 4-7 Simulation results of axial AMB of FlyUPS

In the case of the radial AMBs, the results for a step response on the top radial AMB and the corresponding control effort are given in Figure 4-8. Figure 4-8 also shows the corresponding response at the bottom radial AMB. As the response in the x- and y-axes are very similar, only the results for the x-axis are presented. From Figure 4-8 it is seen that the MPC controller expects a certain amount of cross-coupling between the top and bottom radial AMBs and then acts in order to eliminate the effects of the disturbance.

In order to investigate the effects of the cross-coupling between the top and bottom radial AMBs, a step reference is placed on one AMB, with a zero reference on the other AMB. Table 4-8 shows the response characteristics of the radial AMBs when a step reference is placed on *X1* (the top radial AMB) and the reference for *X2* (the bottom radial AMB) is kept at zero.

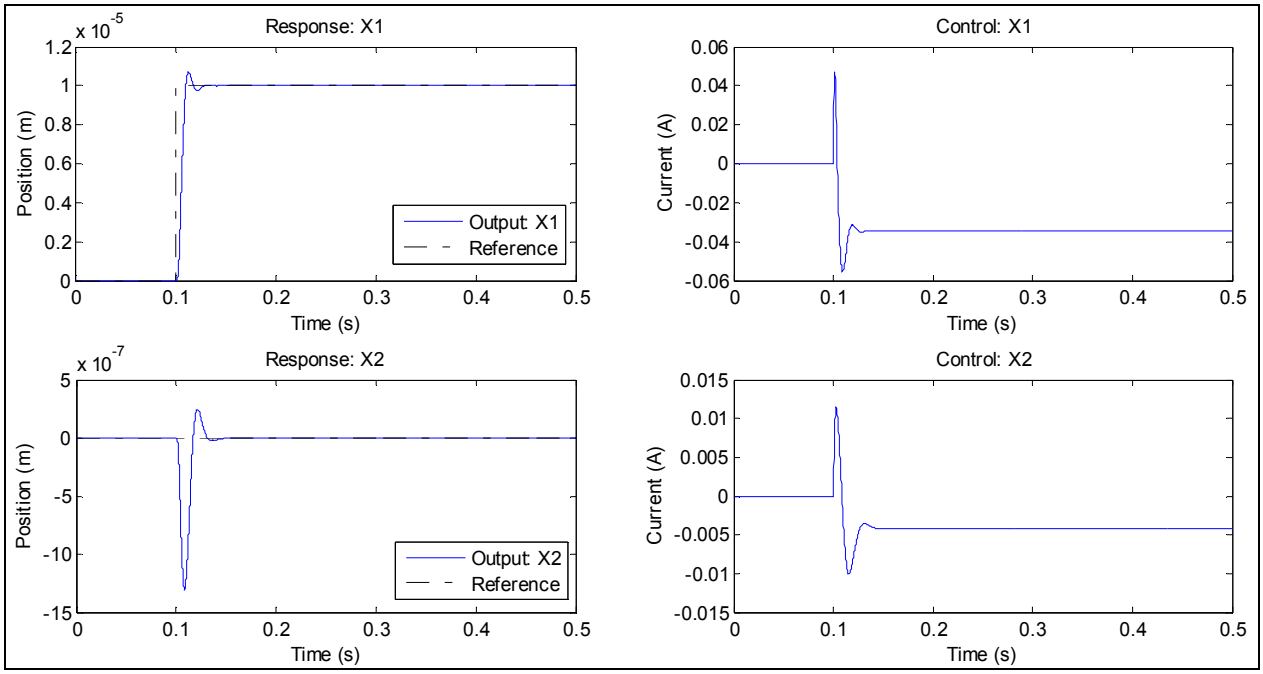

Figure 4-8 Simulated response of radial AMBs of FlyUPS

Table 4-8 Response characteristics of coupling between radial AMBs

Response: *X1*					
***P.O.* (%)**	**T_s (s)**	**T_r (s)**	**k_{eq} (N/m)**	**b_{eq} (N.s/m)**	**Control effort {max; min; transient} (A)**
7	0.024	0.0098	1.225×10^6	6.009×10^3	{0.047; -0.055; -0.034}
Response: *X2*					
Disturbance magnitude(m)	**T_s (s)**	**T_r (s)**	**Control effort {max; min; transient} (A)**		
1.309×10^{-6}	0.024	0.017	{0.012; -0.010; -0.004}		

The set-point change of the previous scenario can be seen as an expected disturbance which the MPC controller can predict in order to take action. However for an unexpected disturbance on the rotor this is not the case. Figure 4-9 shows the response of the radial AMBs when an unexpected step disturbance with a magnitude of 0.25 A is applied to *X1*. From this figure a position disturbance of 60 μm in *X1* is seen, with the cross-coupling causing a corresponding disturbance of 15 μm on X2.

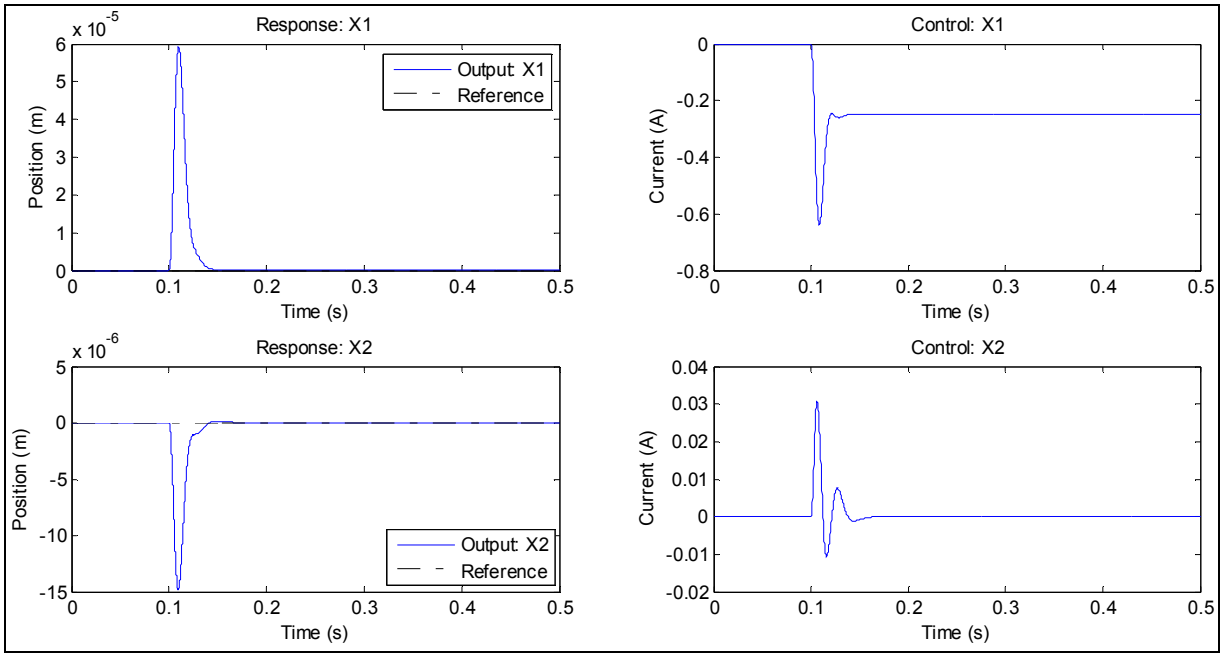

Figure 4-9 Simulated disturbance response of radial AMBs

Another phenomenon that is important in many modern systems is the effect of delays on the response of the system. Figure 4-10 shows the response of the axial AMB when a delay is added to the feedback path. From this figure it is clear that there is a significant influence on the percentage overshoot, settling time and rise time for minor differences between the actual system model and the model used to derive the controller.

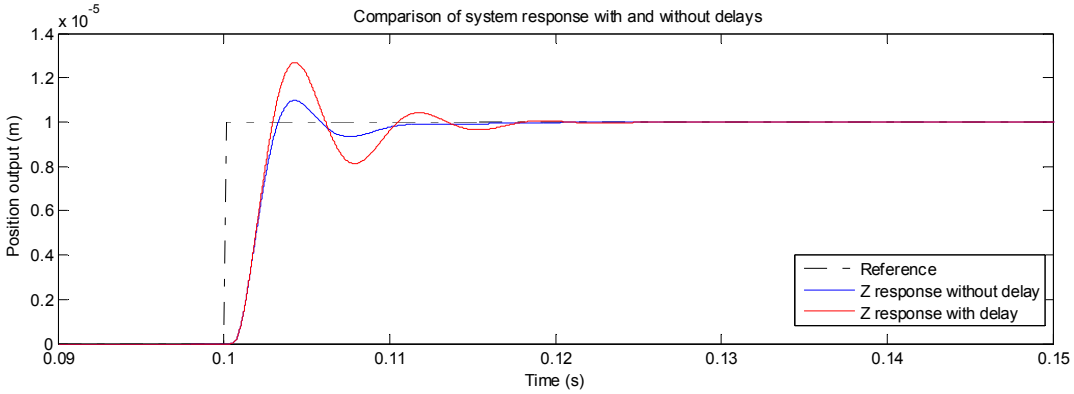

Figure 4-10 Influence of delays on the system response

4.2.3 Discussion of simulation using the MATLAB® MPC toolbox™

From these results it is seen that the MPC controller derived using the MATLAB® *MPC toolbox*™ is effective in controlling the FlyUPS. The results also indicate that the response can be tuned to obtain a desirable performance. The results pertaining to the effects of the MPC controller parameters are particularly useful for giving insight into the working of the MPC controller on the physical system.

The control strategy implemented by the *MPC toolbox*™ gives fast and accurate results. In every simulation the rise time is less than 20 ms, with a settling time that is comparably fast. Also, in each simulation the steady-state error is reduced to zero after a short period of time.

In the case of cross-coupling between the radial AMBs, the control strategy is also effective in following a set-point on one AMB, while keeping the other AMB at a fixed set-point. Also, a larger control effort is required manipulate the position of the rotor at the top radial AMB than at the bottom radial AMB. These results correspond to what is expected in the physical FlyUPS system. For an unexpected disturbance the system experiences large disturbances on the radial AMBs. These disturbances are eliminated with no steady-state error within a short period of time by the MPC algorithm.

When a delay is added to the system a large difference is seen in the response of the system. However, the simulation results indicate that the MPC algorithm is robust enough to handle such unexpected changes in the parameters of the physical FlyUPS system. The deterioration in the performance of MPC when a delay is added to the system also indicates that the performance of the MPC controller depends on an accurate model of the system.

From these points of view the MPC controller derived using the *MPC toolbox*™ is verified as a viable controller to be implemented on the physical FlyUPS system. In chapter 6 the simulation results for this controller are compared to the measured results of the FlyUPS in order to validate that the controller operates as expected. The validation consists of a transient response and disturbance response, as well as the sensitivity function as prescribed by the ISO CD 14839-3 standard for AMBs.

4.3 Conclusion

This chapter investigated the viability of two MPC control strategies for implementation on the FlyUPS. Firstly, a classic MPC algorithm using optimization techniques was simulated. This strategy proved too slow in determining the optimal control signal for a single-degree-of-freedom AMB and did not yield accurate set-point tracking for any choice of the tuning parameters. For this reason the classical MPC algorithm using optimization techniques was not verified as a viable control strategy for the FlyUPS.

Then the MPC strategy used by the MATLAB® *MPC toolbox*™ was investigated. This strategy yielded fast and accurate results. The MPC controller was also able to counteract the effect of cross-coupling for set-point changes on the radial AMBs, as well as unexpected disturbances.

From these results the MPC controller derived using the MATLAB® *MPC toolbox*™ was verified as a viable control strategy for the FlyUPS.

Some considerations that are made before the implementation of the verified MPC controller on the FlyUPS are discussed in the following chapter.

5

Chapter

Controller implementation

This chapter discusses the implementation of the MPC controller derived using the MATLAB® MPC toolbox™ which was verified in the previous chapter. This discussion gives a brief background on the implementation method followed by the setup of the MPC controller for the implementation. Then the benchmark for the implementation of an effective controller for the FlyUPS, namely PD control, is discussed. Finally some considerations regarding the memory usage and controller complexity are discussed.

5.1 Implementation method

As described in [6] and [9], an MPC controller can be implemented on a wide range of hardware platforms. For this project this implementation is represented as in Figure 5-1. Firstly the model containing the MPC controller is set up in Simulink™. With the use of Real-Time Workshop™, this model is then converted to *C* code and compiled as an executable file. The executable file is then linked and embedded to the dSPACE® target.

When this executable file is embedded in the dSPACE® target, the FlyUPS is controlled using the MPC strategy. The user can monitor and control the FlyUPS with the host computer through dSPACE® ControlDesk™ software. ControlDesk™ is used for condition monitoring such as observing the response of the rotor in real-time, and for changing experimental parameters such as activating the PMSM drive and changing the reference signals.

The dSPACE® hardware used in this application features an expansion box with a DS1005 power PC (PPC) connected to various input-output (I/O) modules. The memory of the DS1005 PPC is limited to 16 MB, with the maximum sampling frequency determined by the amount of operations needed to implement the application during each control cycle [26].

With the hardware configuration already in place, applying and tuning the control strategy is done at the top level of the system. This means that the controller is set up in the Simulink™ model and connected to specific I/O modules by using dSPACE® libraries in the model. These tasks are done in software and changes to the control strategy are easily applied to the system.

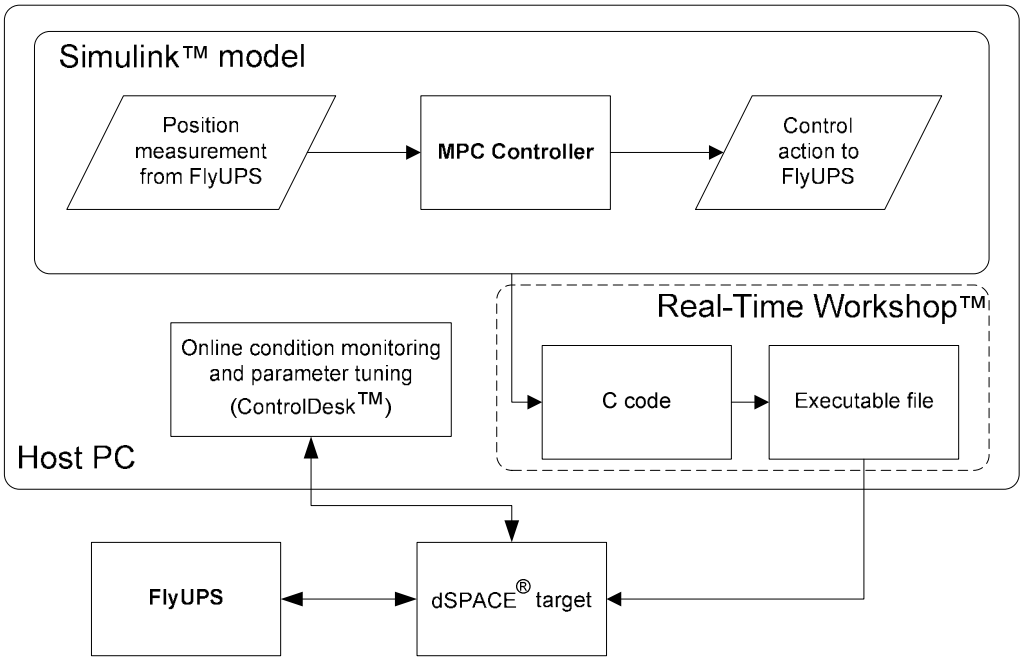

Figure 5-1 Implementation method of MPC on FlyUPS

5.2 Implementation setup

Based on the implementation method discussed in the previous section, the MPC control algorithm is implemented in a Simulink™ model as presented in Figure 5-2. This figure shows the Simulink™ model with the MIMO MPC controller connected to the axial and radial AMBs of the FlyUPS using the dSPACE® libraries for the I/O modules. In order to perform a comparison of the characteristics of MPC with one of the previous control strategies implemented on the FlyUPS, the aim is to keep the basic model of the control the same, and replacing the MPC controller with the alternate controller.

Starting on the left side of Figure 5-2 the position measurements from the analogue-to-digital converter (ADC) modules are read and then calibrated according to the characteristics of the Eddy-probe sensors and the architecture of the FlyUPS. These values together with the reference signals are then taken as the input to the MIMO MPC controller. For this MPC controller block in the Simulink™ model the switch variable is used to switch between different MPC controller setups to be implemented at various operating conditions of the FlyUPS.

The outputs of the MPC controller are the optimal control signals for the FlyUPS. Before the optimal control signals are passed to the power amplifiers, a disturbance is added for analysis purposes. The final blocks are used to set up the control of the differentially driven power amplifiers. The bias blocks specify the base control of the two AMBs used for the linearised control of each degree of freedom with a bias current added to ensure acceptable system

dynamics. The switching blocks are used to switch individual or combinations of AMBs on or off. Finally the control signals are routed to the specific ports of the digital-to-analogue converter (DAC) module according to the design of the FlyUPS.

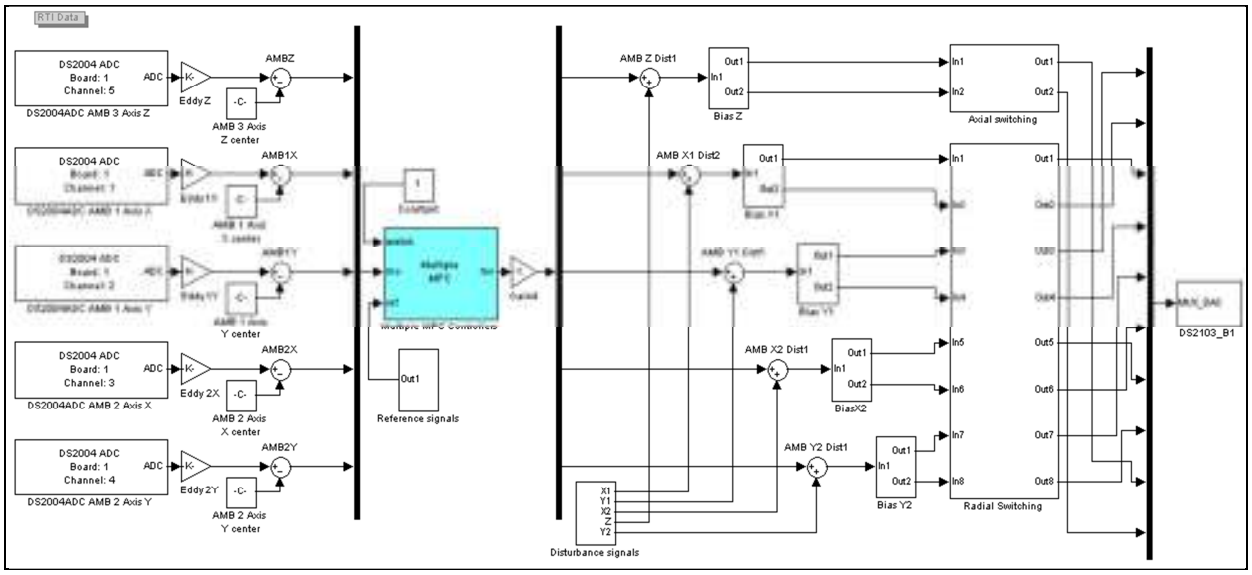

Figure 5-2 Simulink™ model for 5-DOF control of FlyUPS

As the control of the PMSM is beyond the scope of this dissertation, the code for an open-loop three phase PWM control strategy is re-used from previous applications on the FlyUPS. This strategy only marginally increases the execution time of the control. However, as there is no feedback on the actual operating speed of the system, the rate at which the operating speed is increased must be kept low in order to ensure that the control PWM does not lose synchronism with the actual position in the PMSM. In case of a loss of synchronism the rotation must be stopped and restarted.

5.3 Benchmark: Decentralised PD control

For this discussion MPC is compared to decentralised PD control as designed and implemented on the FlyUPS in [1]. However, the setup for the experiment is updated to incorporate some of the recommendations made by the author of [1]. The parameters for the PD control experiment are as given in Table 5-1. The values of the parameters in Table 5-1 are chosen to operate the system at the designed stiffness and damping [1].

The premise for comparing MPC to PD comes from [4], which states that advanced control techniques do not offer appreciable advantages of decentralised control strategies such as PID control. For this application various factors are investigated in order to determine the advantages and disadvantages of using MPC control instead of PD control.

Table 5-1 PD control experiment parameters

Parameter	Value
Proportional gain (K_P) of top radial AMBs	14213
Derivative gain (K_D) of top radial AMBs	10
K_P of bottom radial AMBs	13000
K_D of bottom radial AMBs	13
K_P of axial AMBs	8000
K_D of axial AMBs	25
Bias current (i_{bias}) of radial AMBs	2.5 A
i_{bias} of axial AMBs	2.5 A (top); 1.5 A (bottom)

5.4 Considerations

Before the MPC controller is implemented on the FlyUPS, some consideration regarding the memory usage and execution time is given. These considerations depend on the specifications of the dSPACE® hardware. These considerations determine the performance of the MPC controller e.g. a more complex controller results in more accurate control, but also requires more memory and a longer execution time.

5.4.1 Memory usage

For this discussion the memory usage of the controller is described as the size of the executable file that is embedded in the dSPACE® hardware. The memory of the controller is determined using a function that is available in the command prompt of the host PC from which the FlyUPS is controlled and monitored. This function returns the size of the executable file, including the size of the program code, the size of initialised constant variables and the size of uninitialized variables. However, only the total size of the controller is of interest for this discussion.

As given in Table 5-2 the decentralised PD controller uses the least amount of memory. As more elements of the MPC controller are added, i.e. single-degree-of-freedom MPC controller for the axial AMB, four-degree-of-freedom MPC controller for the radial AMBs, and the full five-degree-of-freedom MPC controller for the FlyUPS, the size of the controller increases. However, at no point does the size of the controller exceed the maximum storage limit of the dSPACE controller, which is 16 MB [26].

Another observation is that the lengths of the control and prediction horizons have no effect on the memory usage of the controller. These measurements are taken for unconstrained MPC. In

the case where constraints are added to the design the memory usage only increases marginally.

Table 5-2 Controller size for different combinations of PD and MPC

Controller setup	Controller size [bits]
Decentralised PD control	1071999
Axial MPC with radial PD control	1088775
Radial MPC with axial PD control	1230771
Full MIMO MPC	1270743

5.4.2 Execution time

The execution time of the controller indicates the period of time it takes to implement the control strategy during one control cycle. This includes reading the measured output of the system, calculating the optimal control signal, and implementing the optimal control action. In order to determine the execution time of the MPC controller the dSPACE® Profiler™ is used. The profiler logs the period of time that passes from the start of the control cycle until the control action is implemented [27].

The baseline for this test is decentralised PD control where each of the five degrees of freedom has an independent PD controller. This PD control along with the PMSM control result in an execution time of 11.25 µs.

For this application various factors influence the execution time of the FlyUPS when the MPC controller is implemented. These factors include the order of the model from which the controller is derived, the use of constraints in the design, and the lengths of N_p and N_c. This is expected as these factors increase the complexity and number of operations that are performed during each control cycle.

By systematically incorporating elements of the MPC controller into the control of the system, e.g. adding a SISO MPC controller for the axial AMB, then including the MIMO MPC controller for the radial AMBs, etc. the effects of the model order and constraints are investigated.

Incorporating a SISO MPC controller for the axial AMB with the radial AMBs controlled by the decentralised PD controllers is investigated first. When an unconstrained MPC controller is derived from the full seventh order model of the axial AMB of the FlyUPS the execution time is determined as 127 µs. By including constraints into this design the execution time increases to 152 µs. By incorporating more elements of the MPC controller, such as the control of the radial

AMBs where the MPC controller is derived from the full twenty-eighth order model of the radial AMBs of the FlyUPS, into the design the execution time become far greater than 200 μs. The sampling period of 200 μs corresponds to a sampling frequency of 5 kHz which is determined experimentally as the lowest possible sampling rate for the FlyUPS to implement stable control.

When adding an unconstrained single-degree-of-freedom MPC controller to replace the axial PD control, the execution time is increased to 31 μs. This unconstrained MPC control is derived based on the reduced second order model of the axial AMB of the FlyUPS. In the case where simple hard constraints such as the minimum and maximum control actions are added to the design of the MPC controller the execution time increases to 44 μs.

For an unconstrained four-degree-of-freedom MIMO MPC controller is used to replace the four radial PD controllers the execution time increases to 125 μs. This unconstrained MIMO MPC controller is derived based on a reduced eighth order model of the FlyUPS. In the case where simple constraints are added to the design of the MPC controller, the controller becomes unable to control the system as the amount of time it would take to complete all of the calculations becomes too large to be completed within a sampling period of 200 μs.

For the unconstrained five-degree-of-freedom MIMO MPC implemented as the control strategy for the FlyUPS, the execution time is 174 μs. This unconstrained controller is derived from a reduced tenth order model of the FlyUPS. As with the previous case, the addition of simple constraints to the design of the MPC controller also causes the control problem to become unstable.

Based on the full 5-DOF MIMO MPC controller for the FlyUPS the effects of N_p and N_c are then determined. Firstly, the effect of N_p on the execution time is investigated by setting N_c = 5 and varying N_p. From the results in Table 5-3 it is seen that the execution time decreases as N_p decreases.

Following the same method the effect of N_c on the execution time is determined. By keeping N_p = 60 the results for varying N_c are determined as in Table 5-4. From these results it is clear that the execution time decreases as N_c is decreased.

In addition to the execution time discussed in this section some overhead operations result in a slight increase in the total time used to implement the control during each control cycle. These overhead operations include initialising and clearing variables at the start and end of the sampling period, reading and writing variables to and from memory, and communication

between the dSPACE hardware and the host PC [28]. As the design of an MPC controller uses more and often larger variables than PD control this overhead will be larger for MPC.

Table 5-3 Execution time for varying N_p

N_p (samples)	Cycle time (µs)
80	174
70	160
60	140
50	120
45	110

Table 5-4 Execution time for varying N_c

N_c (samples)	Cycle time (µs)
5	140
4	117
3	92
2	69

Based on the considerations with regard to the execution time the maximum sampling rate for each control strategy is determined. For the PD control with an execution time of 11.25 µs and considering the overhead discussed previously, the maximum sampling rate is determined as 20 kHz. For MPC (assuming N_c = 5 and N_p = 50) with an execution time of 120 µs including the necessary overhead results in a maximum viable sampling rate of 5 kHz.

5.4.3 Operational limitations

In Figure 5-2 the MPC controller block allows multiple MPC controllers to be used in the control of the FlyUPS. These controllers correspond to different operating conditions, i.e. the changes in the dynamics of the FlyUPS corresponding to changes in the operating speed. For the FlyUPS these controllers are derived based on the models discussed in section 2.5, derived at various operating speeds.

The first controller is derived for operation at standstill and the following controllers are derived at predetermined intervals of the operating speed. These intervals become shorter as the operating speed increases due to the fact that the dynamics of the system change more rapidly with increasing speed. This indicates that the controller is able to update itself only to a point where the model differs too much from the actual system model. Each controller is derived not

only with a different model, but also with different parameters depending on the dynamics of the system.

The MPC toolbox™ implements these multiple by implementing the selected controller and updating this controller at the end of the control cycle. The other controllers that are not selected are also updated at the end of the control cycle. This ensures smooth switching between the controllers.

However when the application of multiple controllers is investigated it is seen that implementing any more than a single MPC controller results in an increase in the execution time of the controller beyond the maximum sampling period of 200 µs. At most two MPC controllers can be implemented when reducing N_c and N_p, however this deteriorates the performance of the controllers which decreases the effectiveness of MPC. It is also noted that taking the necessary overhead into account causes the maximum sampling period to be exceeded occasionally during operation. While not resulting in system instability at standstill, this does raise some concern regarding the application of multiple controllers at other operating speeds where instability can cause damage to the system.

Based on this consideration, only one MPC controller is implemented on the FlyUPS during operation. This controller is derived at standstill and the effects of changing the operating speed are determined.

5.5 Conclusion

This chapter discussed the setup for implementing MPC on the FlyUPS. The aim of this implementation is to evaluate MPC for the FlyUPS by firstly validating the MPC controller and then forming a comparison between MPC and decentralised PD control. Some considerations that were previously identified as possible limitations regarding the memory usage and execution time were also investigated.

In the next chapter the validation and evaluation of the MPC controller implemented on the FlyUPS is discussed.

6

Chapter

Controller evaluation

This chapter discusses the evaluation of the MPC controller on the FlyUPS. Firstly the tests used in the analysis of the MPC controller, namely the transient response, disturbance response and sensitivity function, are discussed. Then the MPC controller is validated by comparing the implementation results with the simulation results in order to determine if the controller functions as expected. Then the MPC controller is compared to decentralised PD control in order to determine the advantages and disadvantages of using MPC.

6.1 Performance assessment

In this discussion three tests are performed to assess the performance of the MPC controller. These tests are the transient response evaluation, the disturbance response evaluation, and the sensitivity function. As the dynamics of the system depend on the operating speed of the FlyUPS, the tests are performed at standstill and 500 r/min. The operating speed of 500 r/min is chosen to show the performance deterioration of the MPC controller derived from a model of the system at standstill, without allowing the system to become unstable.

The transient response, disturbance response and sensitivity function tests are discussed according to the common representation of a closed-loop system as given in Figure 6-1.

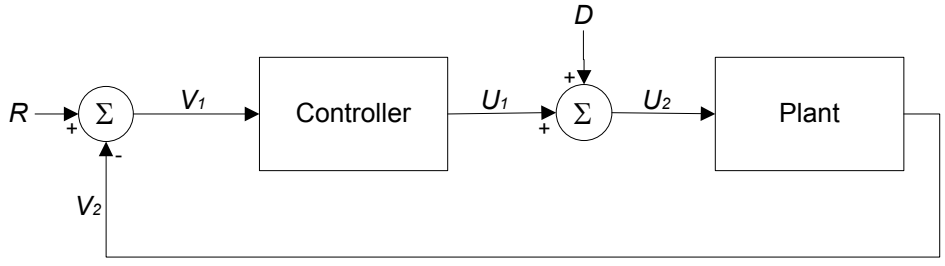

Figure 6-1 Signals and excitation nodes of a common closed-loop system [7]

6.1.1 Transient response evaluation

In order to perform a transient response evaluation a step reference is applied at R and the response of the closed-loop system is measured at V_2. The transient response gives an indication of the percentage overshoot, settling time and rise time of the system. The results of

this evaluation are compared to the results of the simulation, to determine the correlation between the physical and simulated systems.

The magnitude of the step is chosen as 15 µm in order to ensure a good signal to noise ratio, while not operating the FlyUPS in a non-linear area. As the rotor is assumed to only operate within the first two rigid modes, the transient response also gives an indication of how a controller handles the cross-coupling between the radial AMBs.

6.1.2 Disturbance response evaluation

The disturbance response of the closed-loop system is obtained by applying a disturbance to the system at *D* and measuring the response at V_2. The disturbance represents a disturbance force acting on the rotor. The result of this test indicates the ability of a controller to handle unexpected disturbances. For a system such as the FlyUPS that uses linearised AMBs where the control currents are applied in a differential driving mode as presented in Figure 6-2 the disturbance is added to $i_{control}$.

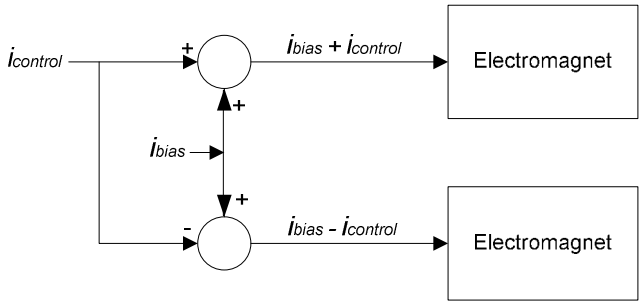

Figure 6-2 Differential driving mode for linearised AMB

The disturbance is chosen as a step with a magnitude of 0.2 A. This corresponds to a force of approximately 3.5 N and 4.7 N applied to the radial and axial AMBs of the FlyUPS, respectively. This disturbance is large enough to ensure a good signal-to-noise ratio while ensuring that the FlyUPS is not forced into a non-linear or unstable condition.

6.1.3 Sensitivity function

For AMB systems the performance robustness of the system is analysed using the sensitivity function. The sensitivity function is evaluated according to the ISO CD 14839-3 standard for AMBs [7]. This function forms the standard due to the fact that it shows the sensitivity of an AMB to disturbances of different frequencies, and it is simple to measure [4].

The sensitivity function is evaluated by injecting a harmonic or sinusoidal excitation signal at *R* and measuring the response at V_1 for each input channel. The sensitivity function is then calculated as [7]:

$$S = 20\log\left[\frac{V_1}{R}\right] \tag{6-1}$$

According to the ISO CD 14839-3 standard this sensitivity function must be characterised up to a frequency of three times the rated speed of the AMB system or 2 kHz, depending on which is the larger value. As the FlyUPS was designed for a maximum operating speed of 25 000 r/min, the sensitivity function is performed up to 2 kHz.

The peak or worst-case sensitivity among the axes determines the sensitivity of the system. The peak sensitivity of the system is classified according to the ISO CD 14839-3 standard as being in one of the four zones described in Table 6-1 [7]:

Table 6-1 Sensitivity ratings according to the ISO CD 14839-3 standard [7]

Zone	Rating
A/B	8 dB
B/C	12 dB
C/D	14 dB

Zone A: Common sensitivity for newly commissioned machines.

Zone B: Machines are cleared for unrestricted long-term operation.

Zone C: Machines are not cleared for unrestricted long-term operation; machines in this zone may require maintenance.

Zone D: Machine sensitivity is severe; machines in this zone may be damaged.

The excitation signal chosen for this test is a chirp, i.e. sinusoidal signal with increasing frequency, with a magnitude of 15 µm. This excitation signal is chosen as it offers a good method of exciting all of the necessary frequencies. The magnitude also ensures a good signal-to-noise ratio without operating the FlyUPS in a non-linear area. The frequency of the chirp is increased by 1 Hz every second to ensure the integrity of the test by allowing the system to reach a steady-state response before the frequency is increased further.

6.2 Validation

In order to determine whether the MPC controller functions as expected the simulated results of the performance assessment discussed previously are compared to the implementation results.

6.2.1 Transient response

The transient response of the FlyUPS is investigated by firstly applying a step reference change to the axial AMB (*Z*) at standstill. Figure 6-3 shows the results of the simulated and measured response, and reference signal for this step change on *Z* at standstill. It is clear from this figure

that the measured response has a much larger *P.O.* and settling time, as well as a longer rise time than in the simulated case. A slight oscillation is also noted in the measured response. This 80 Hz oscillation corresponds to the inertia of the rotor when a force is applied to correct the position of the rotor using the relatively low sampling frequency of 5 kHz.

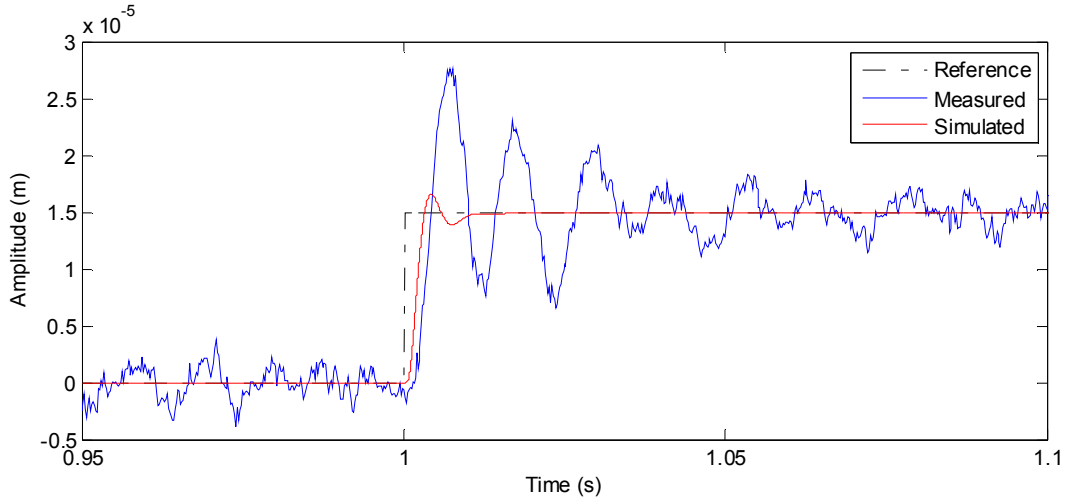

Figure 6-3 Step response of the axial AMB at standstill

Figure 6-4 and Figure 6-5 show the response of the radial AMBs when the step reference change is applied to the top radial AMB (*X1*) and the bottom radial AMB (*X2*), respectively. This indicates that the measured response is slower than predicted. However the measured response follows the simulated response closely in both cases. The cross-coupling between the top and bottom radial AMBs also correlate closely to the simulated values at standstill.

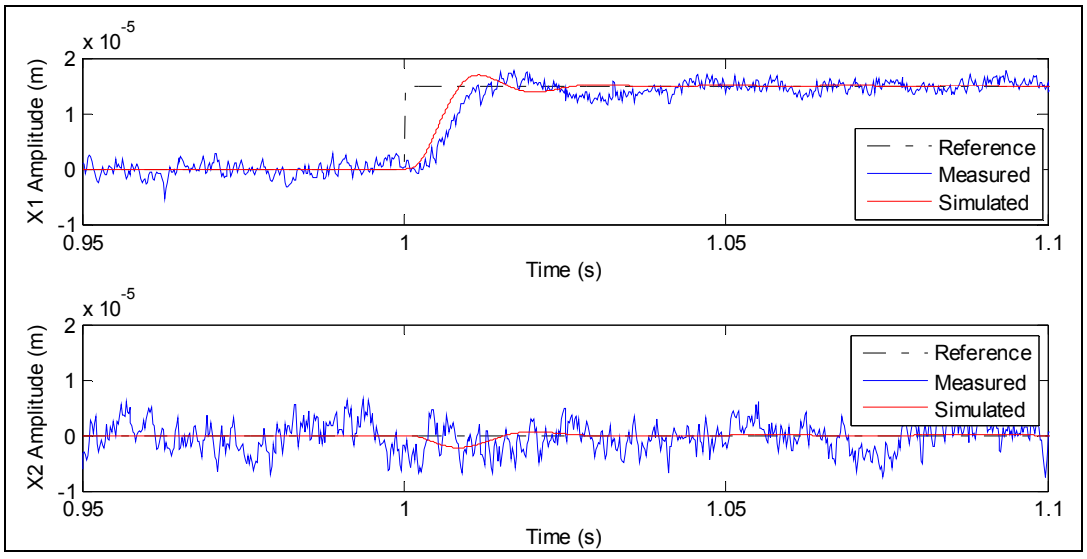

Figure 6-4 step response of the radial AMBs at 0 r/min for a step on *X1*

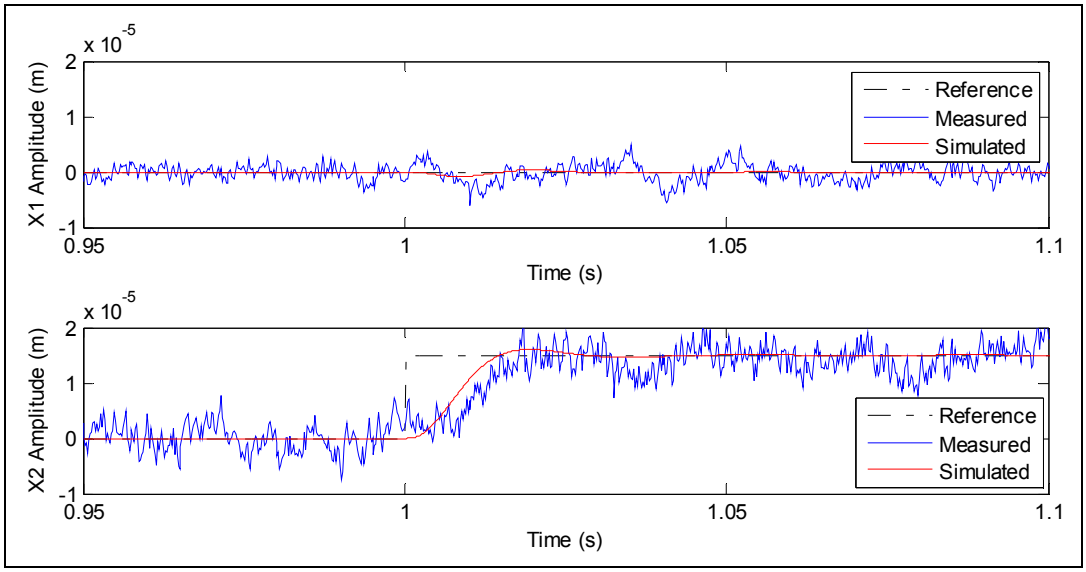

Figure 6-5 Step response of the radial AMBs at 0 r/min for a step on *X2*

At 500 r/min the simulated response of the axial AMB does not change. However as seen in Figure 6-6 the measured response shows some difference as the second harmonic of the operating speed (at 16.7 Hz) influences the response of the axial AMB. This component at 16.7 Hz is apparent in the frequency response of the axial AMB in Figure 6-18.

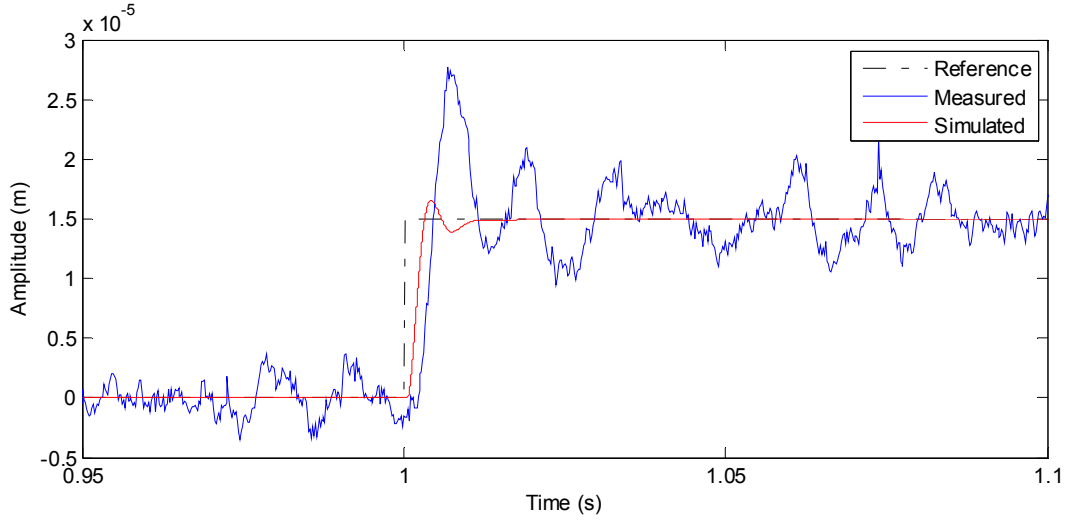

Figure 6-6 Step response of the axial AMB at 500 r/min

The transient responses of the radial AMBs at 500 r/min when a step reference change is applied to *X1* and *X2* are given in Figure 6-7 and Figure 6-8, respectively. In Figure 6-7 a change in the mean of *X1* corresponding to the step reference change is clear, despite the presence of an oscillation with a magnitude of 25 µm. This oscillation corresponds to an unbalance force acting on the rotor which can be identified as the integer multiples-of-two harmonics and is clear from the frequency response of the top and bottom radial AMBs in Figure 6-19 and Figure 6-20, respectively. The reason that only half of the resonance peaks are seen in the frequency response is due to a characteristic of unbalance excitations that only

excite vibrations that whirl in the same direction as the rotor is spinning [4]. In this case this case these vibrations correspond to the integer multiples-of-two harmonics of the operating speed. This unbalance disturbance makes it difficult to see the effects of cross-coupling on the top radial AMB. However, Figure 6-7 and Figure 6-8 indicate that MPC is able to implement the step reference change in the presence of the unbalance.

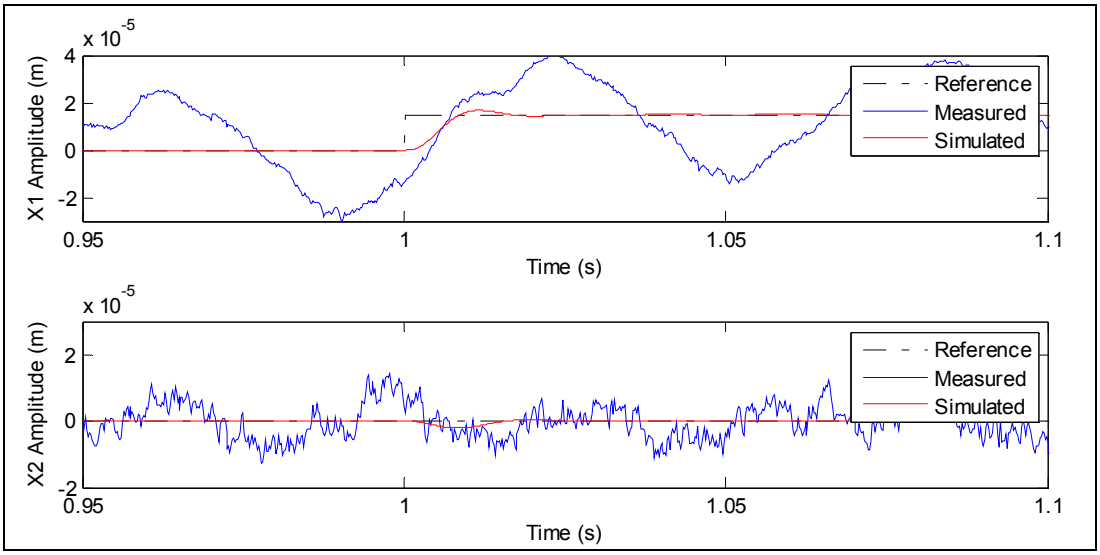

Figure 6-7 Step response of the radial AMBs at 500 r/min for a step on *X1*

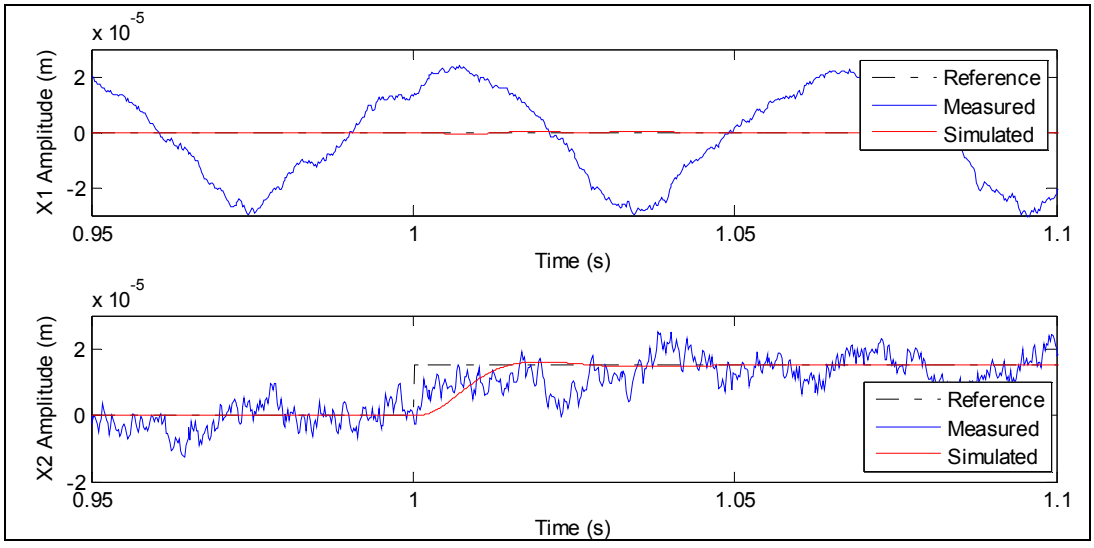

Figure 6-8 Step response of the radial AMBs at 500 r/min for a step on *X2*

6.2.2 Disturbance response

The disturbance response determines the response of the system to unexpected forces acting on the rotor. One such unexpected force is the unbalance disturbance measured on the top radial AMB at an operating speed of 500 r/min. This disturbance causes an oscillation corresponding to the integer multiples-of-two harmonics of the operating speed. The MPC controller is able to keep this disturbance from becoming unstable, but is not able to eliminate this unbalance disturbance.

Further investigations of the effects of disturbances on the system are conducted by firstly applying a disturbance to the axial AMB of the FlyUPS at standstill. Figure 6-9 shows that the rotor experiences a larger displacement than expected, but eliminates the disturbance without a steady-state error within approximately the same amount of time as expected from simulation.

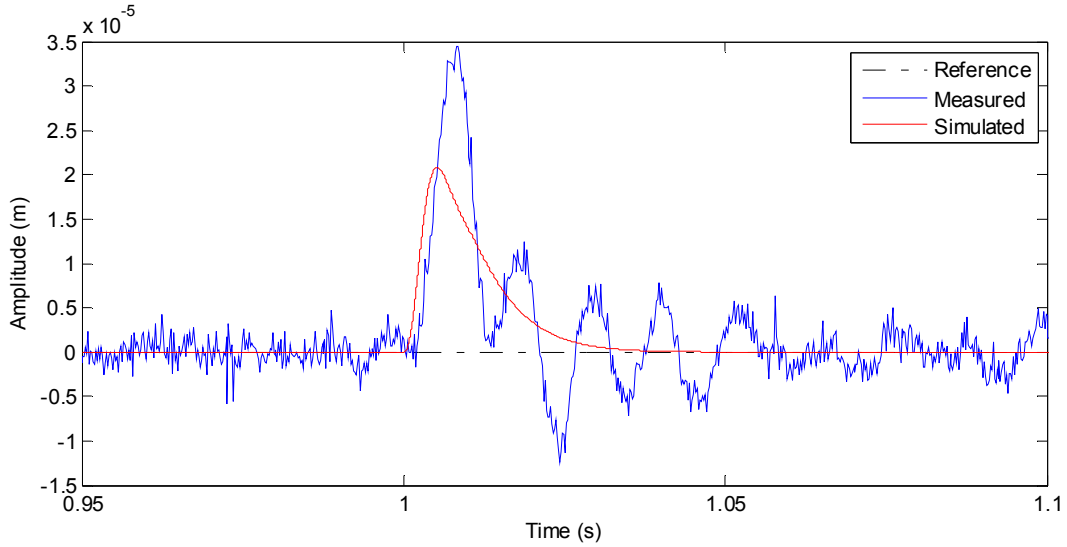

Figure 6-9 Disturbance response of the axial AMB at standstill

Figure 6-10 and Figure 6-11 show the measured and simulated responses of the system at standstill to a disturbance on *X1* and *X2*, respectively. These results indicate that the measured results have a large correlation. As expected the disturbance causes a larger displacement in *X1* than *X2*. However *X1* also eliminates this error faster than *X2*. The amount of cross-coupling between *X1* and *X2* due to the disturbance force in the simulated and measured results also coincide closely.

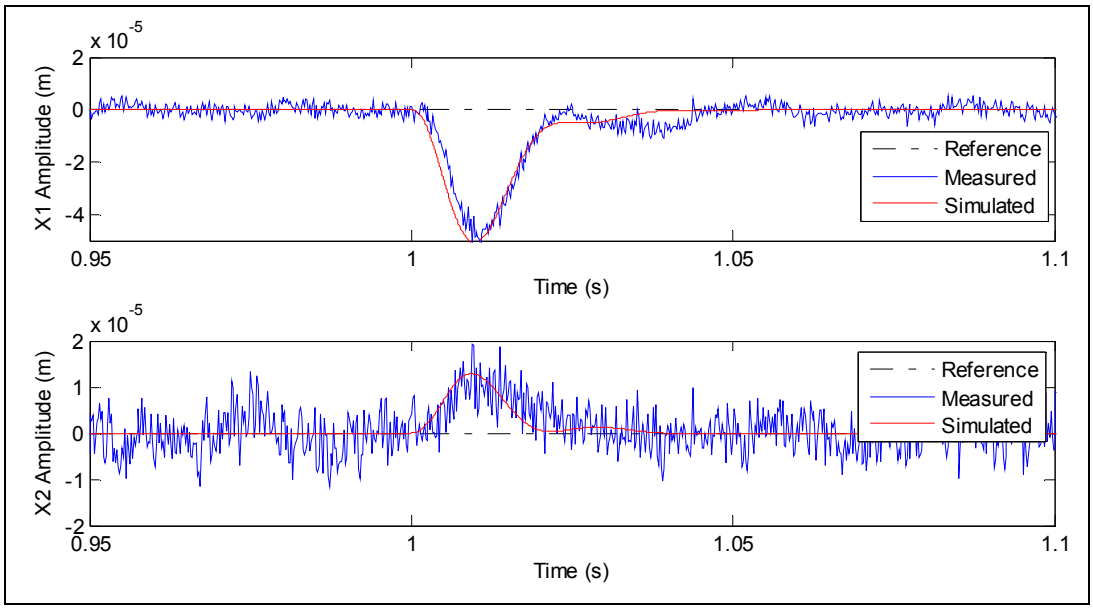

Figure 6-10 Disturbance response of the radial AMBs at 0 r/min for a disturbance on *X1*

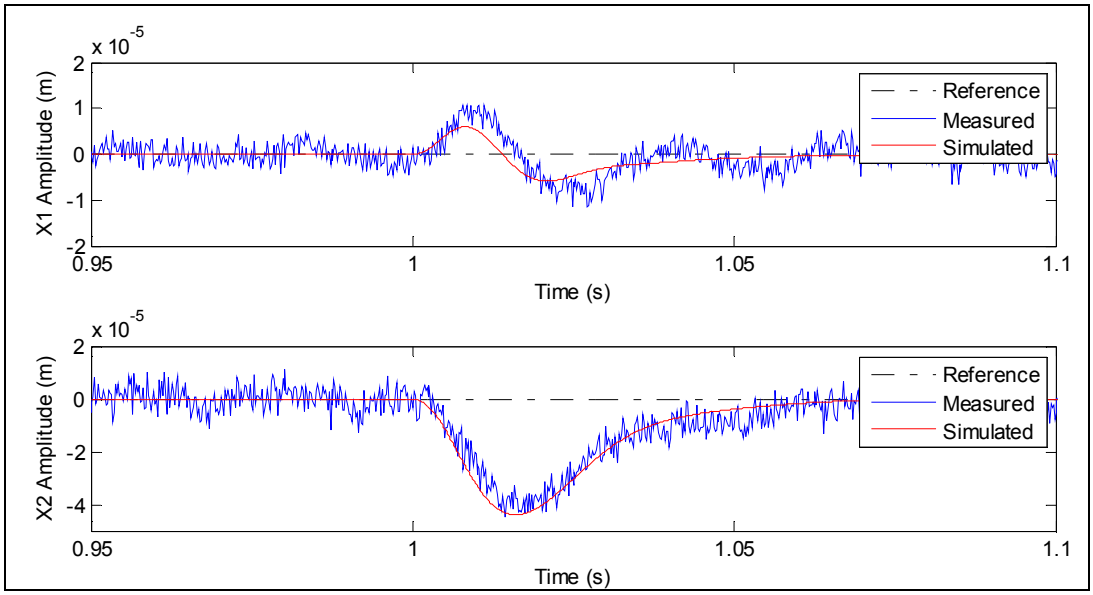

Figure 6-11 Disturbance response of the radial AMBs at 0 r/min for a disturbance on *X2*

The disturbance response of the axial AMB at 500 r/min differs marginally from the response at standstill in both simulation and implementation. Figure 6-12 shows the disturbance response of the axial AMB at an operating speed of 500 r/min. As previously, the measured response shows a larger overshoot than expected, but this error eliminated in accordance with the simulated results.

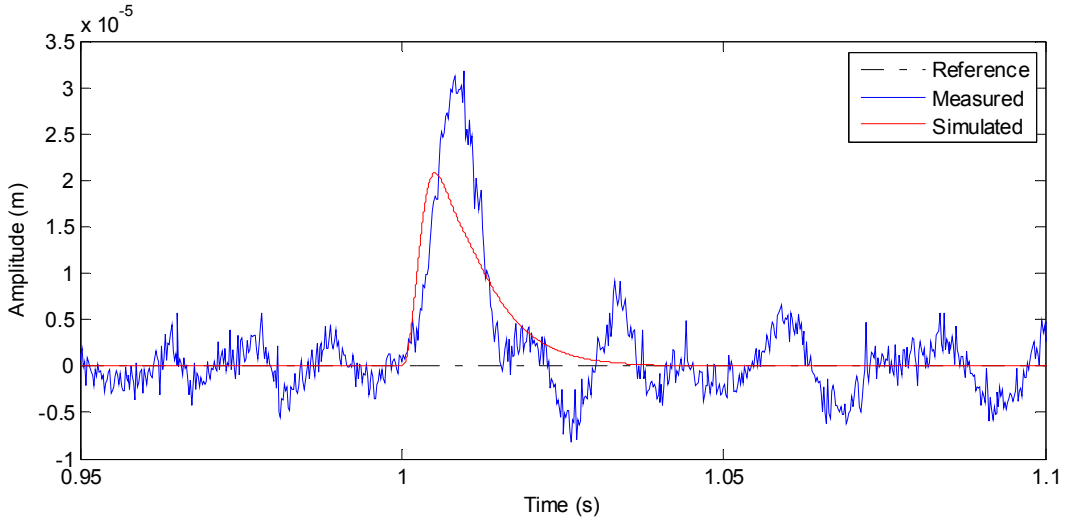

Figure 6-12 Disturbance response of the axial AMB at 500 r/min

At 500 r/min the measured disturbance response of the radial AMBs also follows the simulated response as indicated by Figure 6-13 and Figure 6-14, with the exception of the unbalance seen in the response of *X1*. In Figure 6-13 the response of *X1* shows a larger displacement in the position of the rotor than simulated, however the MPC controller is able to keep the system stable. In Figure 6-14 the effect of the cross-coupling on *X1* is also unclear amidst the unbalance disturbance.

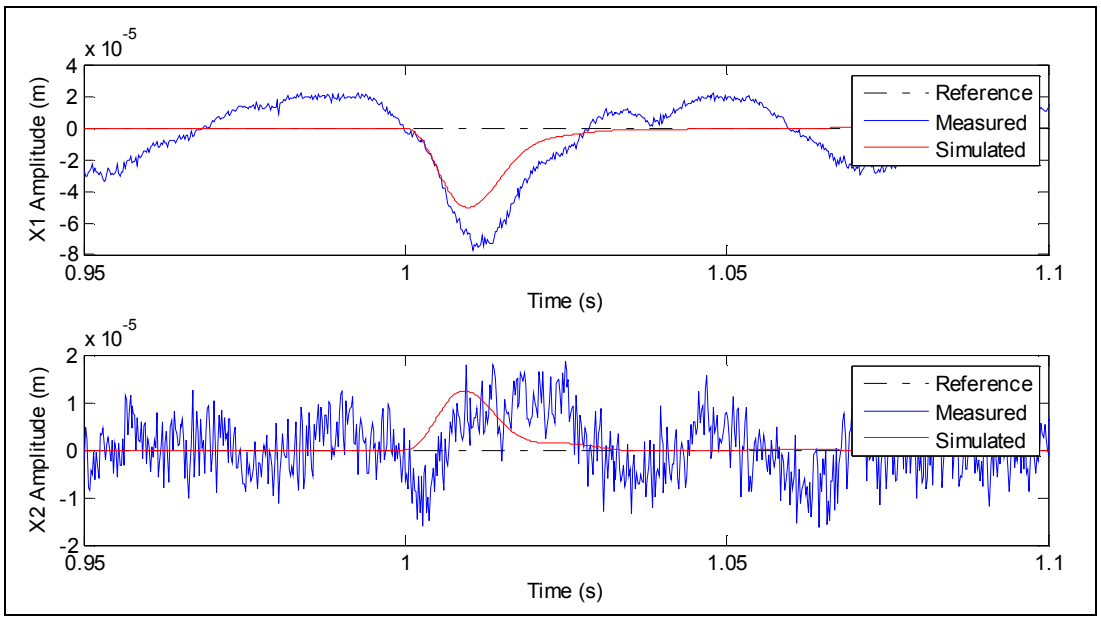

Figure 6-13 Disturbance response of the radial AMBs at 500 r/min for a disturbance on *X1*

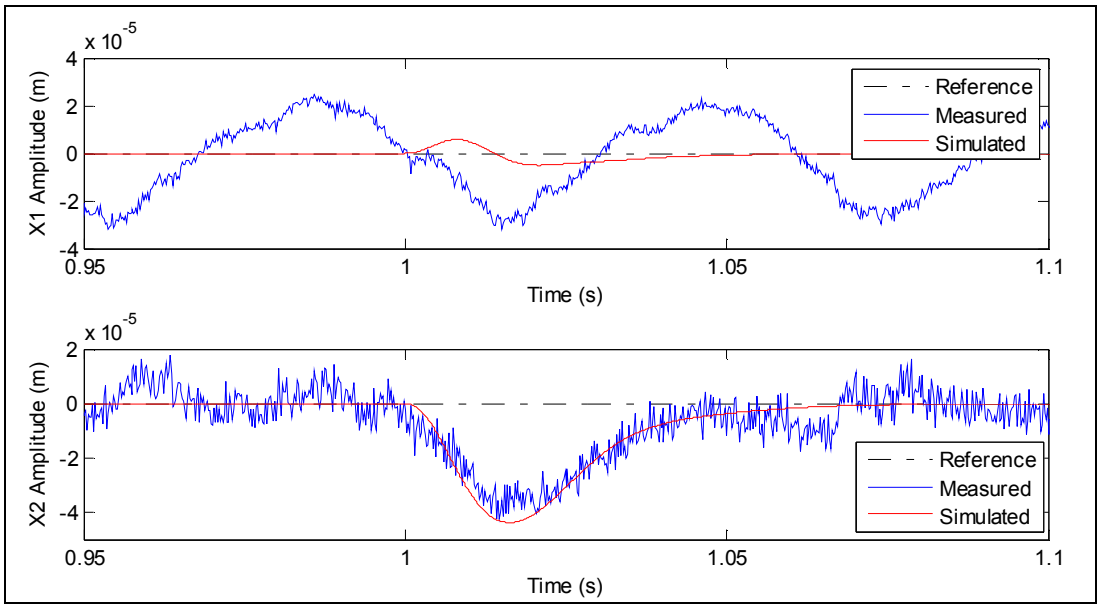

Figure 6-14 Disturbance response of the radial AMBs at 500 r/min for a disturbance on *X2*

6.2.3 Sensitivity function

A comparison of the measured and simulated sensitivity functions for *Z*, *X1* and *X2* are given in Figure 6-15, Figure 6-16 and Figure 6-17, respectively. These results indicate that the actual system contains some dynamics that are not expected from the simulations.

From Figure 6-15 the peak measured sensitivities for *Z* are 4.15 dB at 44.21 Hz and 16.22 dB at 81.76 Hz corresponding to the first two rigid modes of the axial AMB. The peak of 16.22 dB places the axial AMB of the FlyUPS in class D of the ISO CD 14839-3 standard. This measured

result indicates that the axial AMB of the FlyUPS is very sensitive to parameter changes from 82 Hz to 96.5 Hz where *Z* operates in class C and D of the ISO CD 14839-3 standard. The simulated sensitivity function clearly does not predict these dynamics. The simulation result indicates a single peak sensitivity of 4 dB at 160 Hz placing the axial AMB in zone A of the ISO CD 14839-3 standard. This difference indicates that the model of the axial AMB may not be accurate enough to meet the requirements of MPC.

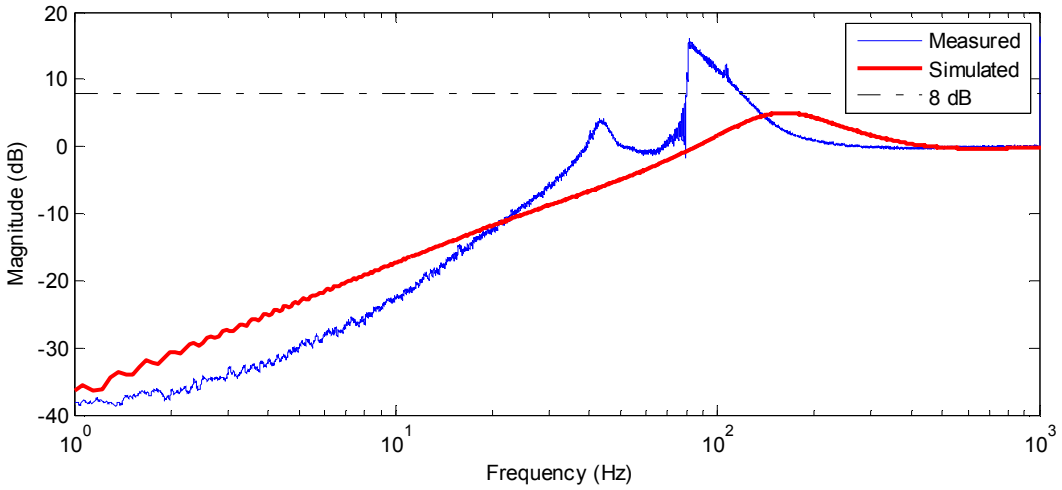

Figure 6-15 Sensitivity function of the axial AMB at standstill

Figure 6-16 and Figure 6-17 give the results for the sensitivity functions of *X1* and *X2*, respectively, at standstill. These results indicate a larger correlation between the simulated and measured results. However, it is also clear from these results that the actual system has some inherent dynamics that are not contained in the model and thus do not appear on the simulation results.

Figure 6-16 shows three peak sensitivities of - 4.82 dB, 4.728 dB and 4.14 dB at 15.38 Hz, 42.16 Hz and 57.63 Hz, respectively, in the measured sensitivity of *X1* at standstill. This places the top radial AMBs within zone A of the ISO CD 14839-3 standard for AMBs. The simulated sensitivity expects a single peak of 5 dB at 58 Hz corresponding to zone A of the ISO CD 14839-3 standard.

At standstill, the measured sensitivity function of *X2* given in Figure 6-17 shows peaks in the sensitivity of 3.18 dB and 4.13 dB, at 15.13 Hz and 34.93 HZ, respectively. This places the bottom radial AMBs within class A of the ISO CD 14839-3 standard on AMBs. The simulated sensitivity of *X2* indicates a single peak of 4.1 dB at 35 Hz corresponding to zone A of the ISO CD 14839-3 standard as well.

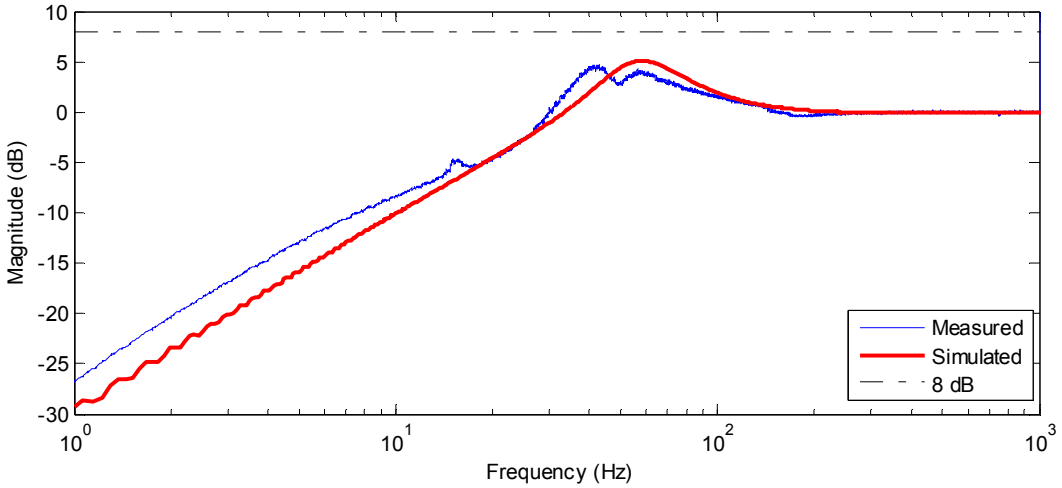

Figure 6-16 Sensitivity function of the top radial AMB at standstill

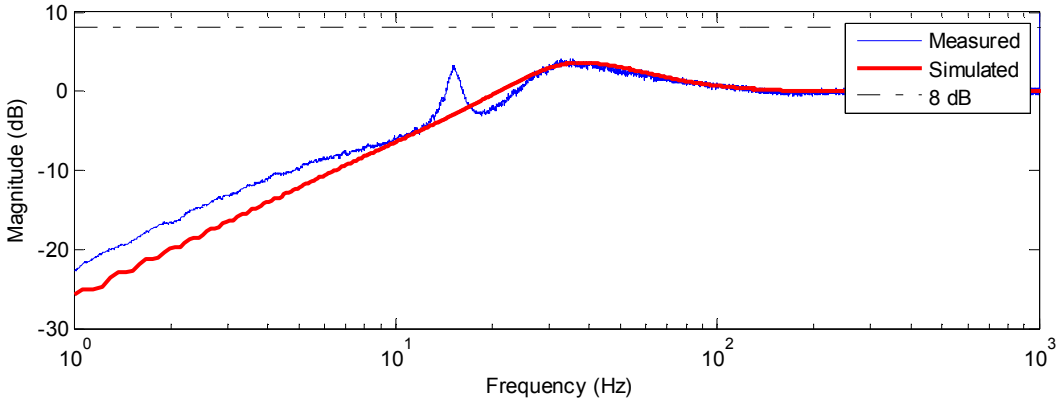

Figure 6-17 Sensitivity function of the bottom radial AMB at standstill

Despite the low sensitivities of the radial AMBs to parameter changes, the system is classified according to the worst case measured sensitivity. This means that the FlyUPS falls within class D of the ISO CD 14839-3 standard due to the peak sensitivity of the axial AMB. For MPC the shape of the sensitivity function is expected as the controller is derived from the model of the FlyUPS at 0 r/min. This means that low sensitivities are expected at low frequencies, with the gain of the sensitivity function increasing as the operating speed increases.

The sensitivity functions of *Z*, *X1* and *X2* only change marginally from operation at standstill to an operating speed of 500 r/min in both simulation and implementation. These results are given in Figure 6-18, Figure 6-19 and Figure 6-20 for *Z*, *X1* and *X2*, respectively. The major change that is noted in these results is the increase in the frequency components corresponding to integer multiples-of-two harmonics of the operating speed, indicating the presence of an unbalance [4]. These harmonics do not appear to have a large effect on the sensitivity of the

axial AMB. However, in the case of the radial AMBs the sensitivity at these harmonic frequencies is driven into zone D of the ISO CD 14839-3 standard for AMBs.

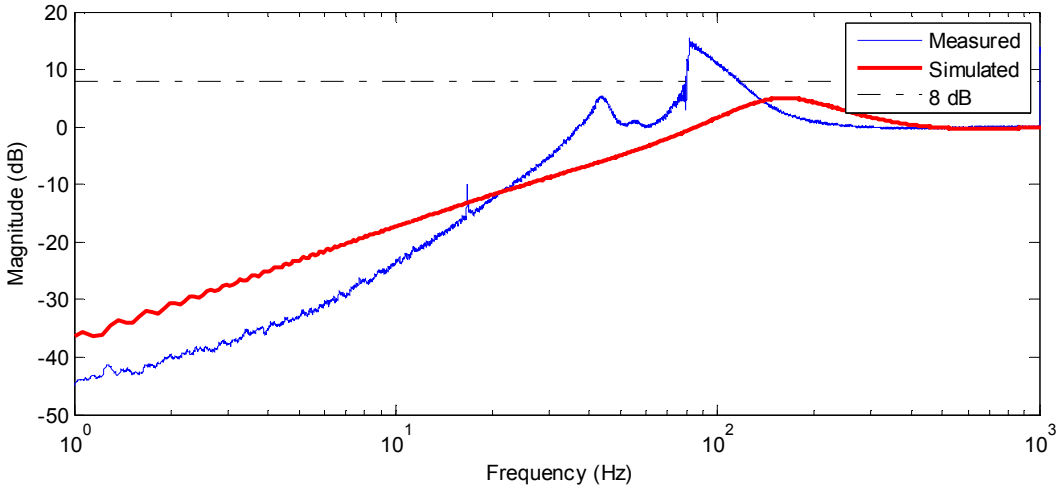

Figure 6-18 Sensitivity function of the axial AMB at 500 r/min

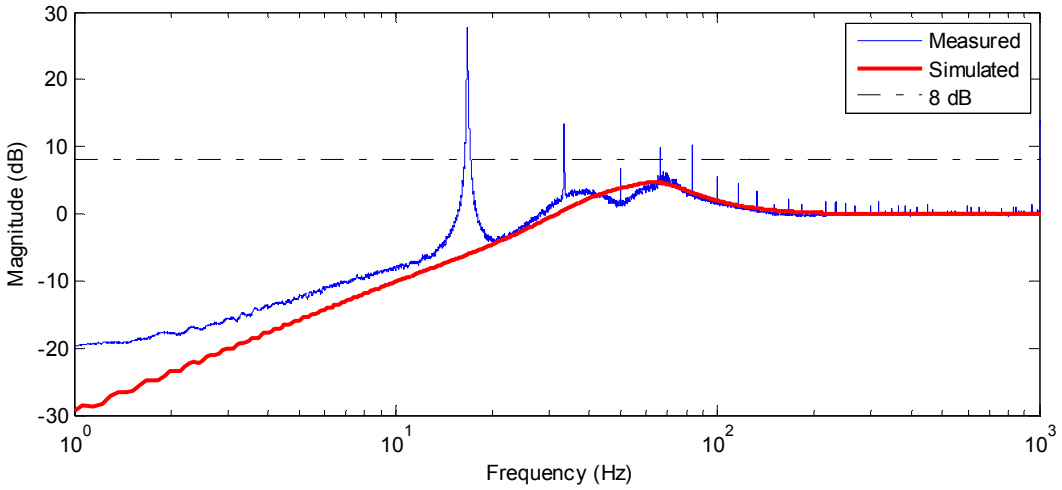

Figure 6-19 Sensitivity function of the top radial AMBs at 500 r/min

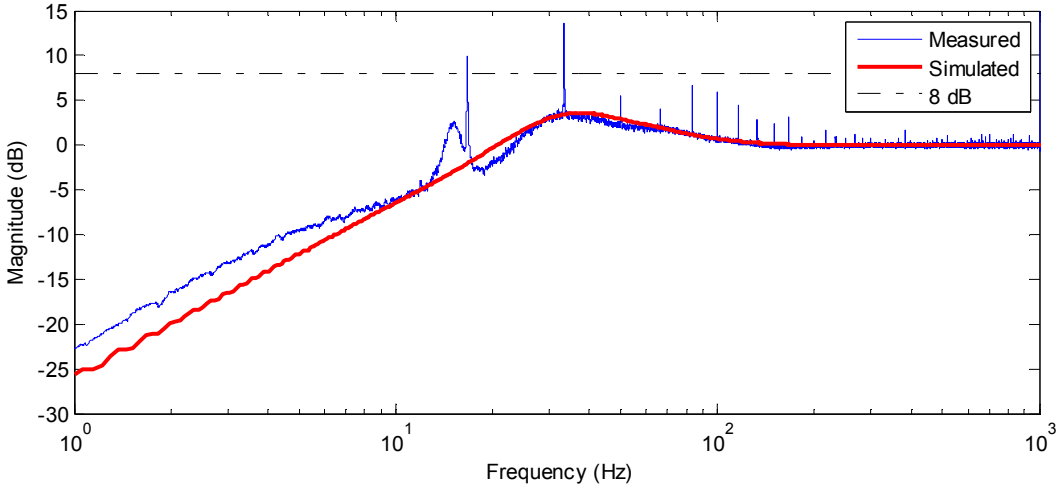

Figure 6-20 Sensitivity function of the bottom radial AMBs at 500 r/min

6.2.4 MPC operating range

For this paragraph the operating range is defined as the amount by which the operating speed can be adjusted about the design speed of the MPC controller before the system response becomes unstable. This is done by implementing an MPC controller derived for the model of the FlyUPS at 0 r/min and increasing the operating speed until the maximum operating speed is attained where the radius of the orbital pattern about the origin reaches 120 µm.

An increase in the operating speed introduces a visible change in the performance of the MPC controller in simulation and implementation where the radius of the orbital pattern increases nearly exponentially as the operating speed is increased. In the case of simulations the maximum operating speed reached is 2440 r/min for an orbital radius of 120 µm in *X1*. In the case of implementation this orbital radius in *X1* is already reached at 1500 r/min. Any further increase in the operating speed of the implementation causes the open-loop PMSM control to lose synchronism with the rotor, and ends the test.

This difference between the measured and expected behaviour can be ascribed to two factors. Firstly the unmodelled dynamics have a larger influence on the system as the operating speed is increased. These unmodelled dynamics include unbalances on the rotor and delays commonly experienced in systems with ADCs and DACs. Secondly the model of the noise that is used in the design of the MPC controller assumes that the noise is wide-sense-stationary white Gaussian noise with unit-variance. As this part of the MPC controller incorporates the integration effect into the design of the controller to eliminate steady-state errors, a more accurate model of the noise may increase the operating range of the MPC controller.

6.2.5 Discussion

Based on the results discussed in this section it is clear that the performance of the MPC controller implemented on the FlyUPS has a large correlation to what is expected in simulation. This means that MPC is validated as a viable control strategy for the FlyUPS within the design parameters of the MPC controller. However, some limitations of MPC such as the effects of unmodelled dynamics and the long execution time raise the question of whether MPC can perform better than a more classical approach such as decentralised PD control.

6.3 Comparison of MPC and PD

In order to further evaluate the effectiveness of MPC as a control strategy for the FlyUPS it is compared to PD control using the same methods as in the previous section.

6.3.1 Transient response

Firstly the step response for the MPC and PD controllers are compared by implementing a step reference change on the axial AMB of the FlyUPS at standstill. From the response shown in Figure 6-21 the PD has a much larger *P.O.*, T_s and T_r than the MPC controller. The MPC controller is also able to follow the reference without any steady-state error due to the integrator effect embedded in the design of the controller. For PD control this steady-state error indicates the high negative stiffness (k_s) encountered in the FlyUPS [4]. This error can be reduced by redesigning the control strategy to accommodate smaller values of k_s or implementing integrating feedback in the control scheme (PID control). The 80 Hz oscillation is not present in the measured response of the PD controller. This is indicative of the fast execution time of the decentralised PD control strategy.

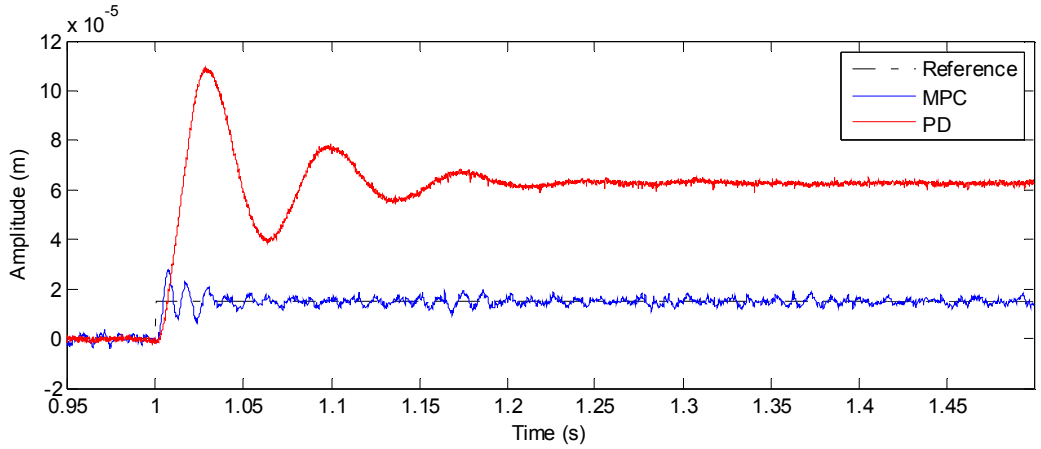

Figure 6-21 Step response of the axial AMB at standstill

The transient responses of the radial AMBs at standstill are given in Figure 6-22 and Figure 6-23 for a step reference change in *X1* and *X2*, respectively. From these figures it is clear that the PD controller manages a faster T_r, however this results in a larger *P.O.* and T_s than the MPC controller. The transient responses of the decentralised PD control also settle with a steady-state error due to high values of k_s for *X1* and *X2*, whereas the MPC follows the reference accurately due to the embedded integrators in the controller design.

The magnitude of the disturbance on the secondary AMB when the step is applied to the primary AMB is also larger for the decentralised PD control. This is expected as the cross-coupling is included in the design of the MPC controller.

When comparing Figure 6-22 and Figure 6-23 it is noted that the bottom radial AMB has a longer T_r and T_s than the top radial AMB for both MPC and PD. In the case of the PD control the setup of the decentralised controllers are the same for the top and bottom radial AMBs

indicating that the equivalent stiffness at the bottom radial AMB is slightly less than the top radial AMB at standstill.

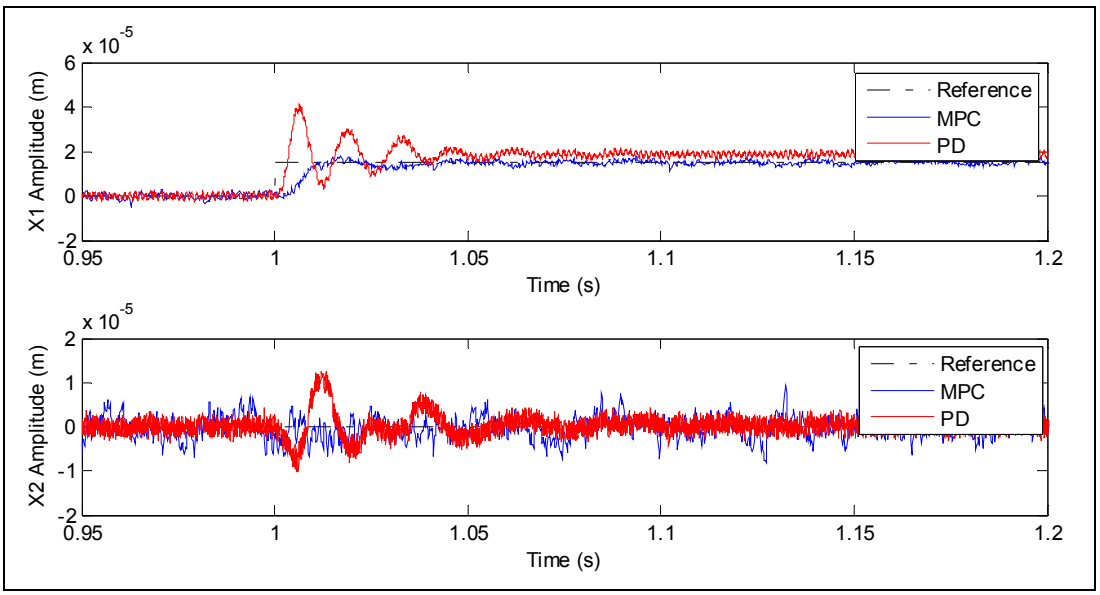

Figure 6-22 Step response of the radial AMBs at 0 r/min for a step on *X1*

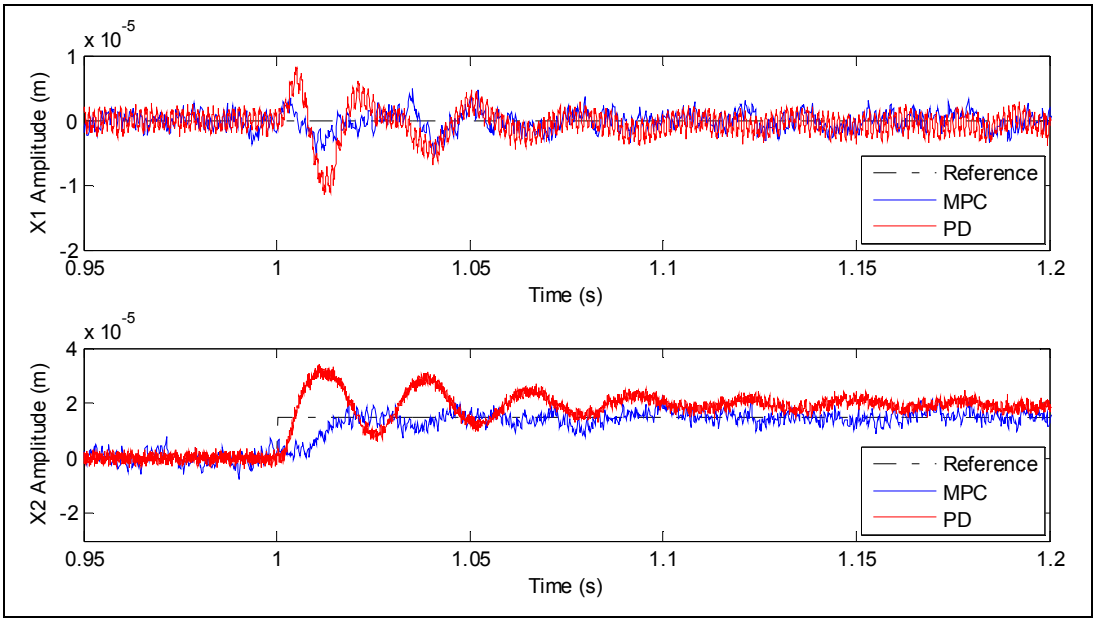

Figure 6-23 Step response of the radial AMBs at 0 r/min for a step on *X2*

When this transient response evaluation is repeated at an operating speed of 500 r/min the performance of the axial AMB only shows marginal changes such as the addition of the 80 Hz oscillation in the response of the MPC controller. This oscillation is not present in the response of the PD controller which is indicative of the faster execution time of the decentralised PD control strategy.

As shown in Figure 6-24 a large oscillation caused by an unbalance is seen in the responses of the radial AMBs for both MPC and PD at 500 r/min. The magnitude of this oscillation for decentralised PD control is less than half that of MPC in the upper radial AMB. The step responses of the radial AMBs at 500 r/min indicate that the performance of both control strategies degrade with an increase in the operating speed, however decentralised PD control is more robust in terms of unbalance rejection than MPC.

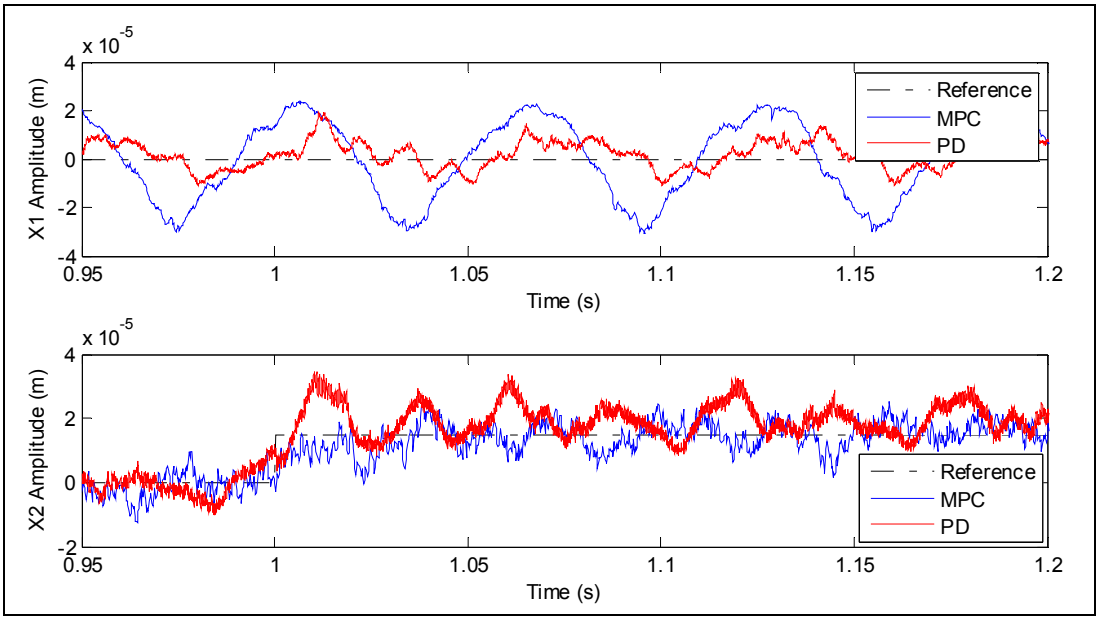

Figure 6-24 Step response of the radial AMBs at 500 r/min for a step on *X2*

6.3.2 Disturbance response

The first disturbance response noted in the FlyUPS is the disturbance force on the rotor due to an unbalance force acting on the rotor at an operating speed of 500 r/min. In Figure 6-25 a comparison of the rotor orbital patterns are given for the top and bottom radial AMBs of the FlyUPS when MPC and PD control are implemented. From these results it is clear that decentralised PD control manages this unexpected unbalance much better than MPC.

The reason for this is that the weights on u, Δu and y used in the design of the MPC controller were chosen to achieve a desirable system response at standstill, i.e. the required stiffness and damping necessary for stable operation. This means that the MPC controller implements a limited amount of stiffness and damping which at 500 r/min is enough to keep the system stable but is not sufficient to eliminate the unbalance. Also, the design of the MPC only incorporates a certain degree of adaptability in the estimation of the system states. At 500 r/min the parameters of the FlyUPS differ too much from the parameters of the reduced order FlyUPS model used to derive the MPC controller. The MPC controller is then unable to adapt enough to take the unbalance into consideration.

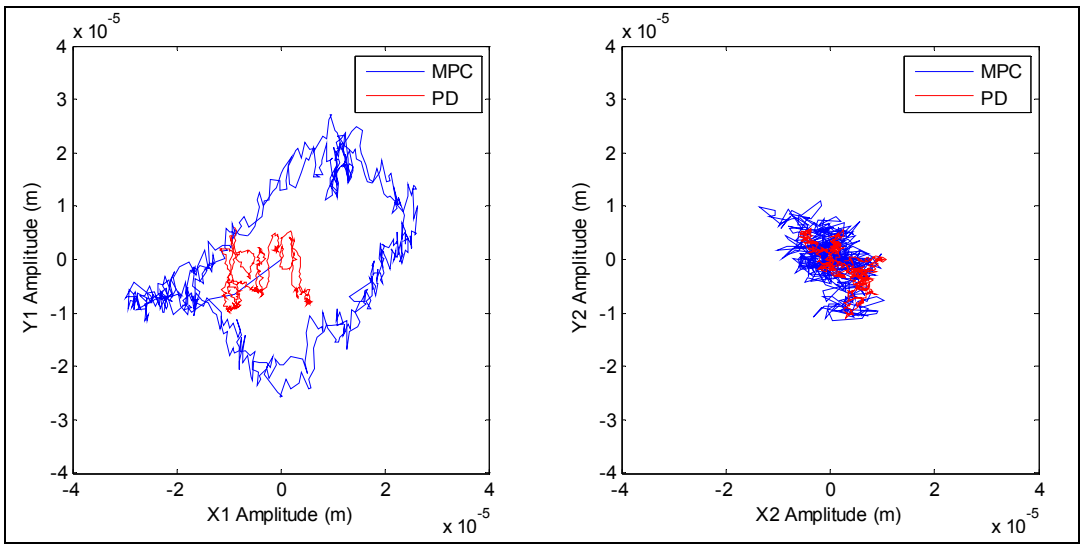

Figure 6-25 Radial AMB orbital patterns at 500 r/min

When determining the disturbance response of the axial AMB of the FlyUPS at standstill using the methods described in section 6.1.2, the results in Figure 6-26 are obtained for the axial AMB at standstill. The PD control seems to implement a step response rather than disturbance rejection. The MPC on the other hand identifies the disturbance and works to eliminate the disturbance with no steady-state error.

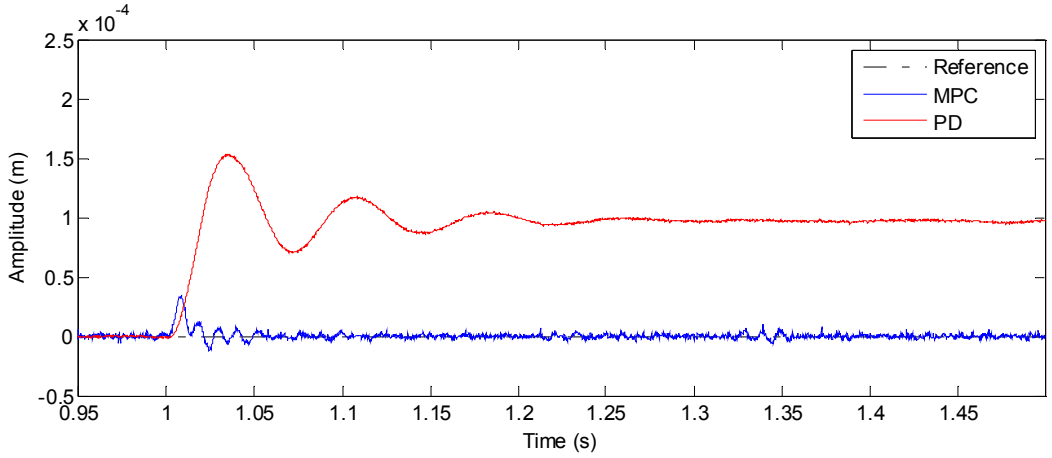

Figure 6-26 Disturbance response of the axial AMB at standstill

The implementation of the disturbance as discussed in section 6.1.2 for the evaluation of the PD control can be predicted by taking into account that the values of K_P and K_D are calculated based on a predetermined stiffness and damping required for the stable operation of the FlyUPS. By applying the disturbance using the method discussed in section 6.1.2 the stiffness and damping characteristics of the AMBs are changed causing the transient position of the rotor to change. This effect can further deteriorate the performance of the PD controller with any uncertainty in k_s. To reduce this error for PD control the disturbance force must be known to calculate the necessary reference signal that must be implemented to cancel the effects of the

disturbance on the position of the rotor. Alternatively, an integrator can be incorporated into the control scheme (PID control).

In the case of MPC a model specifying the operation of the FlyUPS is used to determine the performance of the system. This means that the disturbance applied as discussed in section 6.1.2 is recognised as a disturbance and the MPC controller works to eliminate the error. An integration effect is also embedded in the MPC controller design to eliminate steady-state errors.

From this discussion it is clear that using PD control requires a lot of insight to successfully implement disturbance rejection. For MPC an accurate model of the system ensures that disturbances are identified and eliminated within the operating limits of the controller.

6.3.3 Sensitivity function

As stated previously the sensitivity function of an AMB system represents the robustness of the system to parameter changes. The sensitivity function of the axial AMB of the FlyUPS at standstill is given in Figure 6-27. The peak sensitivity for PD control on *Z* is 19.67 dB at 13.46 Hz, which falls within zone D of the ISO CD 14839-3 standard. This is a worse sensitivity than MPC, however both MPC and PD are categorised as being in zone D. From these it is also noted that MPC has a lower sensitivity to parameter changes for frequencies up to 30 Hz. After this mark the PD control shows better robustness.

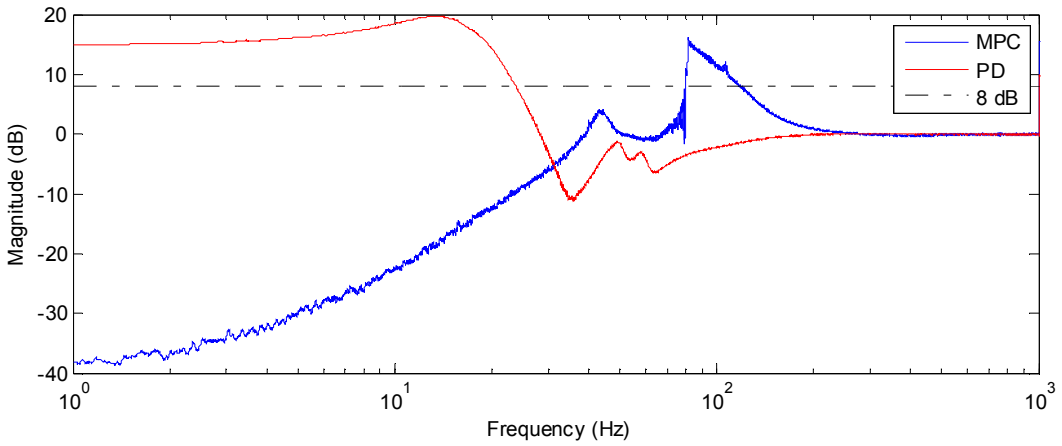

Figure 6-27 Sensitivity function of *Z* at standstill

In the case of the radial AMBs at 0 r/min shown in Figure 6-28 and Figure 6-29 for *X1* and *X2*, respectively, the overall robustness of the MPC controller is better than the decentralised PD controllers with peak sensitivities placing *X1* and *X2* within zone A of the ISO CD 14839-3 standard for AMBs. For the decentralised PD controllers peaks in the sensitivity of *X1* and *X2*

place both the top and bottom radial AMBs in zone D of the ISO CD 14839-3 standard for AMBs. These peaks are 16.02 dB at 79.14 Hz for *X1* and 17.2 dB at 39.84 Hz for X2.

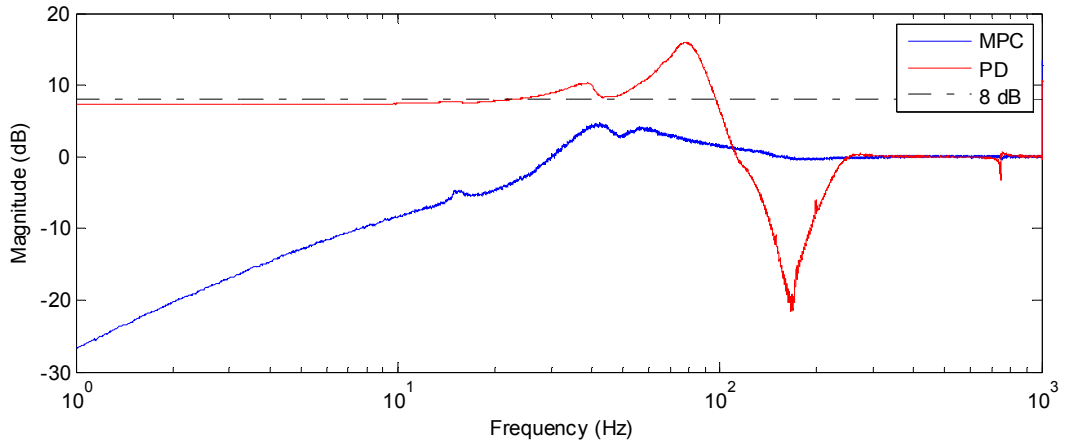

Figure 6-28 Sensitivity function of *X1* at standstill

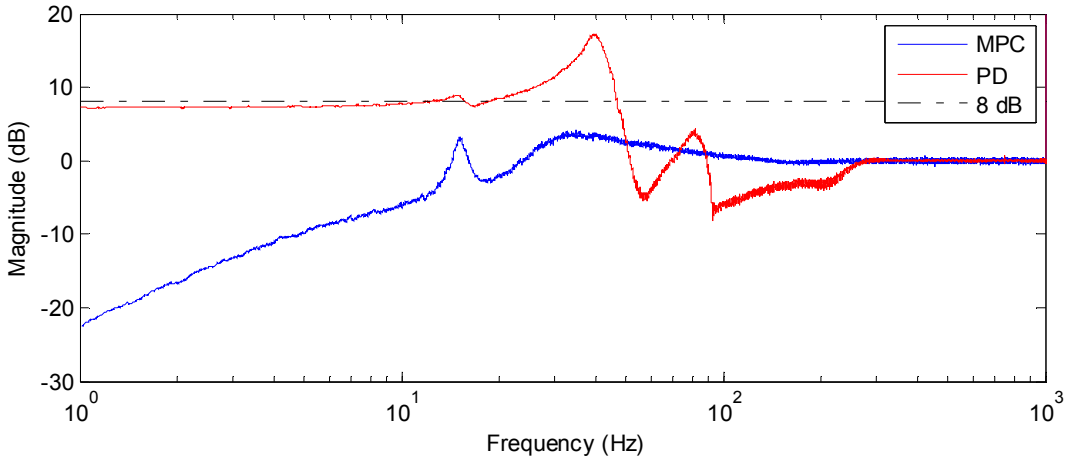

Figure 6-29 Sensitivity function of *X2* at standstill

At an operating speed of 500 r/min the frequency response shows the integer multiples-of-two harmonic components introduced by an unbalance on the rotor and the influence of these harmonics on the sensitivity functions of the FlyUPS for MPC and PD control. Figure 6-30 shows the sensitivity function for *X1* which indicates that the integer multiples-of-two harmonics have a much larger effect on the sensitivities of the radial AMBs at lower frequencies. For the radial AMBs the peak sensitivities of both controllers at 500 r/min fall within zone D of the ISO CD 14839-3 standard.

The sensitivity of the system is classified by the worst-case sensitivity of any of the components according to the ISO CD 14839-3 standard for AMBs. This means that the FlyUPS falls within zone D for MPC and decentralised PD control. However, it can be noted that MPC offers superior performance to PD control at the design frequency of the controller.

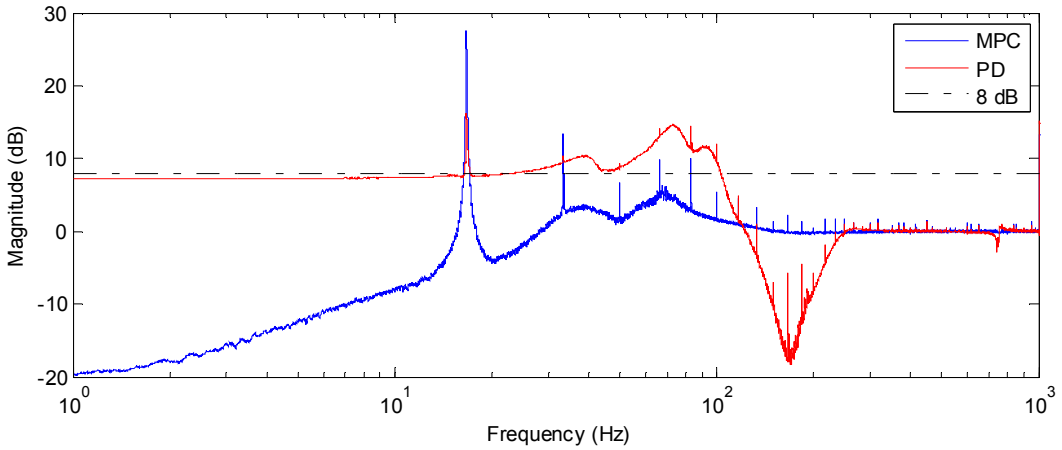

Figure 6-30 Sensitivity function of *X1* at 500 r/min

6.3.4 Discussion

From the comparison of MPC and PD several observations are made regarding the advantages and disadvantages of implementing MPC on the FlyUPS. By using a modern control technique such as MPC the optimal operation of the system can be guaranteed. However, this optimal control is limited to the design parameters of the MPC controller. The most important parameter in the design of the MPC controller is an accurate model of the system dynamics. A certain degree of adaptability is also incorporated in the design of the MPC controller to obtain a fair operating range for the MPC controller about the design parameters.

As discussed previously the operating range of the MPC controller derived at standstill can operate with a relative degree of stability to an operating speed of 1500 r/min. In the case of the decentralised PD control the FlyUPS can operate safely up to an operating speed of 6600 r/min without any changes to the controllers [1]. This indicates that the decentralised PD control is more robust to changes in the dynamics of the FlyUPS due to the relatively short execution time and high stiffness that are achieved.

The derivation of the MPC controller is simple yet effective when using a tool such as the *MPC toolbox*™. Some insight of the system is needed to determine the parameters of the controller for a desired system response. Tuning these parameters also requires some time. On the other hand, the design and implementation of successful PD controllers require a high level of skill and insight into classical control theory and the system [4].

6.4 Conclusion

In this chapter the performance of MPC on the FlyUPS was investigated. This analysis included the validation of the MPC controller and a comparison between MPC and decentralised PD control. The validation results indicate that the MPC controller performs as expected in simulation at standstill with good responses in set-point tracking, disturbance rejection and robustness in the output sensitivity. However, as the operating speed is increased to 500 r/min an unbalance is observed on the radial AMBs.

This unbalance is identified by the sinusoidal disturbance in the position output of *X1* and introduces integer multiples-of-two harmonics that are clearly visible in the frequency response of the FlyUPS for MPC and PD control [4]. For decentralised PD control the unbalance has a smaller effect than for MPC due to the faster sampling frequency and higher effective stiffness of the PD controllers.

When considering the performance of the controllers with regard to the effort required to design the controllers, it is clear that decentralised PD control requires more insight and skill to obtain desirable results such as accurate set-point tracking, disturbance rejection and output sensitivity [4]. The good performance of the MPC controller is inherent in the design, even when implementing MIMO control. This design of the MPC controller is simplified by using a tool such the *MPC toolbox*™ where some insight into the system and the effects of the controller parameters is required. However, this generic design of the MPC controller increases the computational load, leading to a long execution time.

The good performance of the MPC controller is limited to a certain operating range, which is seen to be much shorter than the operating range of the decentralised PD control. This indicates that the MPC controller is limited in adapting to unmodelled dynamics in the system. Another limitation in the MPC design is that fixed parameters restrict the stiffness and damping applied in the AMBs.

In the next chapter conclusions and recommendations are made based on this analysis of MPC implemented on the FlyUPS.

7

Chapter

Conclusions and recommendations

This chapter reports the conclusions and recommendations for this dissertation. Firstly, a conclusion is drawn on the use of MPC as a viable control strategy for the FlyUPS. Then recommendations for the future improvement of MPC for the FlyUPS are made. Finally, a closing statement is made regarding the success of this study.

7.1 Conclusions

Based on the evaluation of MPC for the FlyUPS in this dissertation, it is clear that there are some advantages and limitations in the design and implementation of MPC.

From the verification of the MPC controller in chapter 4 it is clear that various control algorithms exist in the class of MPC. However, not all algorithms provide the same level of performance. As seen in chapter 4 the derivation of a MIMO MPC controller using a classical approach and incorporating some optimisation techniques offers good insight into the theory of MPC, but was not verified as an effective control scheme for the FlyUPS. When using a software tool such as the MATLAB® *MPC toolbox*™ a generalised approach to deriving the MIMO MPC controller was used along with some guidelines on parameter selection, and parameter tuning to obtain the desired response. This approach yielded good results and required much less effort to derive than the MPC controller derived based on the classical approach. Thus the MPC controller derived using the MPC toolbox™ was verified as an effective control scheme for the FlyUPS.

The implementation on the FlyUPS provided a further advantage of AMBs. This advantage is that the existing system can be used for the excitation and measurement of the MIMO dynamics of the FlyUPS. This approach reduces the equipment and cost of the physical implementation as no extra equipment such as signal generators or oscilloscopes are needed. The condition of the system can also be monitored without removing the rotor from the enclosure to perform a rotordynamic analysis.

From the implementation of the MPC controller on the FlyUPS further observations can be made regarding common limitations of MPC. The model order used for MPC and control

constraints implemented heavily impact computational intensity and the resulting controller execution time. This implied limiting the order of the model to ten and omitting constraints from the controller design in order to achieve a minimum 5 kHz sampling frequency required for high speed AMB control The relatively long execution time of a single MPC controller also meant that only one controller could be implemented during operation so that no gain-scheduling could be done to switch between controllers derived for certain operating conditions.

In order to evaluate the implementation of MPC on the FlyUPS the MPC controller was firstly validated by comparing the performance of the implementation to the simulated performance. From this validation it was seen that the performance of the implementation correlated closely to the performance expected in simulation with regard to the transient response, disturbance rejection and the robustness to parameter changes. The design of the MPC controller based on the model of the FlyUPS shows good performance about the working point of the controller design with the elimination of steady-state errors in the transient response due to the integrator effect incorporated in the design. This controller also effectively eliminates disturbances resulting from the cross-coupling between the radial AMBs.

However, at an operating speed of 500 r/min an unbalance disturbance is noted in the response of the radial AMBs. This unbalance is seen as the sinusoidal disturbance in the position output of the radial AMBs and can be identified by the integer multiples-of-two harmonic components in the frequency response that are introduced by the unbalance at the operating speed. From these results it is clear that the MPC controller is limited to the design parameters derived at standstill which cannot implement enough stiffness in the AMBs to eliminate the unbalance disturbance. The MPC algorithm updates the states at each control cycle to update the controller for optimal control. However this adaptability is limited as the dynamics of the system change.

A further analysis was performed by comparing the performance of a MIMO MPC controller with the five SISO controllers used in decentralised PD control. This analysis was performed to investigate whether MPC would have any advantages over this classical control strategy. The results indicated that MPC offers better performance than PD control within the design parameters of the MPC controller for the step response, disturbance rejection and sensitivity function. Further advantages of MPC were that the disturbances as a result of the cross-coupling had a smaller effect on the performance of the system, and that the MPC controller was able to eliminate steady-state errors due to the integrator effect embedded in the design of the controller. In order to achieve the same performance with PD control various approaches based on deep insight and past experience would be required.

As the operating speed was increased the advantage in the performance of MPC deteriorated. At 500 r/min an unbalance was noted in the responses of both MPC and PD control. The decentralised PD control was able to reject this disturbance more effectively than the MPC controller. This was due to the limitations in the adaptability of the MPC controller to unmodelled dynamics. The PD controller on the other hand is designed to achieve a predetermined stiffness and damping that should be sufficient for the entire operating range of the FlyUPS making decentralised PD control more robust to parameter changes at higher frequencies.

As stated previously the design of the MPC controller is based on specifications from which the controller design directly follows, with some insight required for choosing and tuning the design parameters. The design of a sufficient PD controller for the FlyUPS is motivated by a physical approach and gives good insight into the control problem. However this requires a deeper knowledge of classical control theory and skill rooted in experience to achieve good performance. A major advantage of the PD control is that the design of the decentralised PD controllers ensures a relatively short execution time of 11.25 μs compared to the execution time of 140 μs needed by the MPC controller.

A final consideration in the validation of MPC for the FlyUPS is to ensure that the problem statement as described in section 1.2 has been resolved. From the discussion of this research topic in this dissertation it is clear that MPC is able to implement real-time control of a complex AMB system. However some limitations regarding the operating range due to unmodelled dynamics, as well as a long execution time due the computational load, have been noted. With regard to the sensitivity of the FlyUPS to parameter changes, MPC was able to operate the radial AMBs of the FlyUPS within zone A of the ISO CD 14839-3 standard for AMBs; however even at standstill the total system sensitivity was rated as being in zone D. This indicates that while MPC is able to control the FlyUPS, some improvements to the MPC strategy must first be made to ensure the optimal performance of the FlyUPS.

7.2 Recommendations for future work

Through this investigation it was seen that MPC is a viable control strategy for the FlyUPS. However, some limitations of MPC must be addressed to improve the performance. These limitations include the execution time and operating range of the MPC controller.

7.2.1 System assessment

Prior to further implementations on the FlyUPS, an assessment of the system should be performed to determine the status of the FlyUPS. Of primary interest is the rotor dynamics such as the unbalance that is clearly visible in the frequency response of the FlyUPS. Other factors to

investigate are the effects of aging on the system since its commissioning in 2007. This includes the aging of the flywheel rotor, stator, sensors and other electrical components. Any changes in the system e.g. the replacement of electrical components and sensors are also of interest as this can have minor effects on the performance of the system.

Other assessments of the system can be done with the focus of improving the performance of the system by expanding the hardware capabilities of the FlyUPS. Currently, the dSPACE® controller uses a single Power PC (PPC) as the central processing unit to implement the control strategy on the FlyUPS. This limits the abilities of the system as discussed previously. By incorporating a second PPC into the dSPACE® controller, the computational load can be distributed yielding faster execution times for complex control strategies such as MPC [26]. This will also allow more features of MPC such as constraints and a gain-scheduling strategy to be included in the controller design.

7.2.2 MPC algorithm

As discussed in section 2.2 the class of MPC is derived from various control strategies that are model-based and employ predictive elements. In chapter 3 and 4 of this dissertation two such techniques were investigated for implementation on the FlyUPS. However, there exist various other MPC strategies involving different models or optimisation techniques in the design of the MPC controller.

With regard to alternative models used in the design of the MPC controller “smart” models such as neural- or fuzzy network models can be incorporated that can be updated on-line to adapt to changes in the system [5]. System identification can also be used to create an accurate model of the system that includes the dynamics [5]. Alternatively system identification can be done to create a model of the uncertainty in the system which, along with the existing model described in section 2.5, can be taken into account when determining the optimal control action [4]. This may also give a better representation of the noise used in the design of the MPC controller, as opposed to the assumption of wide-sense-stationary unit-variance white Gaussian noise that is generally used. Another alternative is to use numerical structuring methods to reduce the large amount of numerical redundancy often inherent to state-space models [4]. These redundancies occur as a result of the number of control parameters and multiply-accumulate (MAC) operations, and lead to an increase in the computational load and longer execution times.

For the optimisation algorithm various techniques have been proposed for the reduction of the computational load when calculating the optimal control action for a MIMO system. The most prominent technique involves convex optimisation algorithms that reduce the computational

load by exploiting special patterns in the formulation of the objective function [5]. This technique is common in applications involving quadratic programming problems such the formulation of the MPC cost function when constraints are implemented on the output and increment of the control signals. Alternatively the parameters that do not change during the plant operation can be identified and calculated offline to reduce the computational load [5]. These parameters can then be implemented in a gain-scheduling algorithm during operation.

7.2.3 Control specification

From the results of the evaluation it is clear that MPC does not respond well to unmodelled disturbances; in this case an unbalance. Unbalance disturbances are technically the most important source of exciting natural vibrations in rotating machinery [4].

Apart from balancing the rotor, incorporating a model representing the effects of unbalances in the MPC design, or incorporating feed-forward control for unbalance compensation, this limitation in the MPC design can be minimised or overcome by changing the control specification to let the rotor spin about its principle axis of inertia rather than forcing the rotor to spin about its axis of geometry [4]. This specification will allow the rotor to spin within the air gap and without touching the backup bearings with the advantage that no disturbance forces act on the rotor.

A further advantage of using this specification is that a smaller control current is required to maintain the position of the rotor during operation without touching the backup bearings. This control current increases until the rotor operates about the inertial axis. As no disturbance forces act on the rotor the control current does not increase significantly when the operating speed is increased, which is the case for rotation about the axis of geometry [4].

7.3 Closure

The goal of this dissertation was to evaluate the effectiveness of MPC for the FlyUPS. Various advantages and disadvantages of MPC were investigated during the design and implementation of the MPC controller according to the problem statement in section 1.2. Based on this evaluation MPC can be seen as a viable control strategy for the FlyUPS, however more research is needed to optimise the MPC approach to yield significant advantages over other control techniques such as decentralised PD control.

The findings of this dissertation have yielded good insight and experience in the development of a model predictive control strategy for the FlyUPS. With the methods and results discussed, this

dissertation can serve as a basis for future research of advanced control strategies on AMB systems.

References

[1] S. Myburgh, "The development of a fully suspended AMB system for a high-speed flywheel application," North West University, Potchefstroom, M Dissertation 2007.

[2] J.J. Janse van Rensburg, "Development of a flywheel energy storage system - Uninterruptable power supply," North West University, Potchefstroom, M Dissertation 2007.

[3] R. Hebner, J. Beno, and A. Walls, "Flywheel Batteries Come Around Again," *IEEE SPECTRUM*, pp. 46-51, April 2002.

[4] H. Bleuler et al., *Magnetic Bearings - Theory, Design and Application to Rotating Machinery*, Gerhard Schweitzer and Eric H. Maslen, Eds. Berlin, Germany: Springer, 2009.

[5] J.M. Maciejowski, *Predictive Control with Constraints*. Essex, England: Pearson Education Limited, 2002.

[6] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*. Melbourne, Australia: Springer-Verlag London Ltd, 2009.

[7] ISO 14839-3, "Mechanical vibration - Vibration of rotating machinery equipped with active magnetic bearings - Part 3: Evaluation of stability margin.," International Organization for Standardization, ISO, 2006.

[8] SJM Steyn, "Multivariable H_infinity control for an active magnetic bearing flywheel system," North West University, Potchefstroom, M Dissertation 2010.

[9] A. Bemporad, M. Morari, and N. L. Ricker, *Model Predictive Control Toolbox (TM) User's Guide*, 3rd ed. Massachussets, United States of America: The Mathworks, Inc., 2011.

[10] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: applications to industrial processes," *Automatica*, no. 14, pp. 413-428, 1978.

[11] J.M. Martin Sanchez, "Adaptive predictive control systems," 4197576, 1976.

[12] E. F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. London, United Kingdom: Springer-Verlag, 1995.

[13] P. Airkka, "Advanced Control Methods for Industrial Process Control," *IEE Computing and Control Engineering*, pp. 18-23, July 2004.

[14] B. Gu and Y. P. Gupta, "Control of nonlinear processes by using linear model predictive control algorithms," *ISA Transactions*, vol. 47, pp. 211-216, 2008.

[15] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, no. 11, pp. 733-764, 2002.

[16] B. Codrons, *Process modelling for control: a unified framework using standard black-box techniques*. United States of America: Springer-Verlag, 2005.

[17] The Mathworks, Inc. (2011) Product documentation: modred. [Online]. http://www.mathworks.com/help/toolbox/control/ref/modred.html

[18] C. Zhang, K. J. Tseng, Y. Xiao, and K.Y. Zhu, "Model-Based Predictive Control for a Compact and Efficient Flywheel Energy Storage System with Magnetically Assisted Bearings," in *35th Annual IEEE Power Electronics Specialists Conference*, Aachen, Germany, 2004, pp. 3573-3579.

[19] K. Y. Zhu, Y. Xiao, and A. U. Rajendra, "Optimal control of the magnetic bearings for a flywheel energy storage system," *Mechatronics*, no. 19, pp. 1221-1235, 2009.

[20] T. D. Nguyen, K. S. Tseng, C. Zhang, S. Zhang, and H. T. Nguyen, "Model Predicitve Control of a Novel Axial Flux Permanent Magnet Machine for Flywheel Energy Storage System," in *IPEC, 2010 Conference Proceedings*, Singapore, 2010, pp. 519-524.

[21] G. Schweitzer, "Active Magnetic Bearings - Chances and Limitations," in *Proc. IFTOMM 6th Int. Conf. Rotor Dyn, Vol. 1, p.1*, 2002.

[22] Gerhard Schweitzer, Hannes Bleuler, and Alfons Traxler, *Active Magnetic Bearings*.: Authors Working Group, 2003.

[23] The Mathworks, Inc. (2011) Mathworks Product Documentation - Robust Control Toolbox. [Online]. http://www.mathworks.com/help/toolbox/robust/ref/hankelmr.html

[24] C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed. Upper Saddle River, N.J.: Pearson Education International, 1998.

[25] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 10th ed. Upper Saddle River, NJ.: Pearson Prentice Hall, 2005.

[26] dSPACE GmbH, "DS1005 PPC board," dSPACE GmbH, Paderborn, User documentation 2011.

[27] dSPACE GmbH, "dSPACE Profiler 1.1.1," dSPACE GmbH, Paderborn, User documentation 2007.

[28] S. Van der Linde, "Bulk energy storage potential in the USA, current developments and future prospects," *ENERGY*, no. 31, pp. 3446-3457, 2006.

[29] R. G. Lawrence, K. L. Craven, and G. D. Nichols, "FLYWHEEL UPS," *IEEE INDUSTRY APPLICATIONS MAGAZINE*, pp. 44-50, May | June 2003.

[30] A. Emadi, A. Nasiri, and S. B. Bekiarov, *Uninterruptible Power Supplies and Active Filters*. Boca Raton, United States of America: CRC PRESS, 2004.

[31] dSPACE GmbH, "FAQ 269 - Determine the size of real-time application," dSPACE GmbH, Paderborn, FAQ 2011.

Appendix A: Program code for MPC using optimisation techniques

A.1. MPC main program

```
% MPC algorithm for SDOF AMB
% adapted from "Model Predictive Control System Design and Implementation
% using MATLAB(R)" by L. Wang

%% Model
clear all;
clc;

load('AMBSFPAZ_0rpm.mat')       %load axial model
modelc = AMBSFPAZ;
h = 1e-6;                       %sample time
modeld = c2d(modelc,h,'zoh');   %discretize model
modeld = ss(modeld,'minimal');  %ensure minimal realization
[Ap,Bp,Cp,Dp] = ssdata(modeld);
[m1,n1] = size(Cp);
[n1,n_in] = size(Bp);

%% Augment model
A = eye(n1+m1);
A(1:n1,1:n1) = Ap;
A(n1+1:n1+m1,1:n1) = Cp*Ap;
B = zeros(n1+m1,n_in);
B(1:n1,:) = Bp;
b(n1+1:n1+m1,:) = Cp*Bp;
C = zeros(m1,n1+m1);
C(:,n1+1:n1+m1) = eye(m1,m1);
D = zeros(m1,n_in);

%% Laguerre parameters
a = 0.85;                      %pole positions
N = 3;                         %number of approximation terms

Hp = 1;                        %prediction horizon

%% Weights for cost
Q = 0.9*(C'*C);                %performance parameters
R = 0.1 * eye(n_in,n_in);
lambda = 0.999;                %prescribed degree of stability
alpha = 1.3;                   %exponential weighting
[Klqr,P,E] = dlqr(A/lambda,B/lambda,Q,R);
gamma = lambda/alpha;
Q_alpha = gamma^2*Q + (1-gamma^2)*P;
R_alpha = gamma^2*R;
Ahat = A/alpha;
Bhat = B/alpha;

%% Cost function
% J = eta'*Q*eta + 2*eta'*Psi*eta
[Omega,Psi] = dmpc(Ahat,Bhat,C,a,N,Hp,Q_alpha,R_alpha);
L_m = zeros(n_in, sum(N));
[Al,L0] = lagd(a(1),N(1));
L_m(1,1:N(1)) = L0';
In_s = 1;
for jj = 2:n_in
    [Al,L0] = lagd(a(jj),N(jj));
    In_s = N(jj-1) + In_s;
    In_e = In_s + N(jj) - 1;
    L_m(jj,In_s:In_e) = L0';
```

```
end
K = L_m*(Omega\Psi);            %MPC controller gain
Acl = A - B*K;                  %closed-loop eigenvalues

%% Initialise simulation data
y = zeros(m1,1);                %output
u = zeros(n_in,1);              %control signal
xm = zeros(n1,1);               %states
N_sim = 10000;                  %number of samples in simulation
sp = ones(1,N_sim+10)*10e-6;    %set-point signal for simulation

%% Simulation
[M,Lzerot] = Mdu(a,N,n_in,1);
[u1,y1,deltau1,k] = simucon(xm,u,y,sp,Ap,Bp,Cp,N_sim,Omega,Psi,Lzerot,alpha,lambda);

%% Results
yfin = y1';
ufin = u1';
deltaufin = deltau1';

figure
subplot(3,1,1)
plot(y1(1,:))
title('Output')
%xlabel('samples')
subplot(3,1,2)
plot(u1(1,:))
title('Control')
%xlabel('time (ns)')
subplot(3,1,3)
plot(deltau1(1,:))
title('Delta U')
xlabel('samples')
```

A.2. Function: *dmpc*

```
function [Omega,Psi,Setwei] = dmpc(A,B,C,a,N,Np,Q,R)
% discrete time MPC cost function parameter calculation
% adapted from "Model Predictive Control System Design and Implementation
% using MATLAB(R)" by L. Wang

[n,n_in] = size(B);
N_pa = sum(N);
Omega = zeros(N_pa,N_pa);
Psi = zeros(N_pa,n);
R_para = zeros(N_pa,N_pa);
n0 = 1;
ne = N(1);      %parameterize R matrix for number of inputs
for i = 1:(n_in-1)
    R_para(n0:ne,n0:ne) = R(i,i)*eye(N(i),N(i));
    n0 = n0 + N(i);
    ne = ne + N(i+1);
end
R_para(n0:N_pa,n0:N_pa) = R(n_in,n_in)*eye(N(n_in),N(n_in));

S_in = zeros(n,N_pa);
[Al,L0] = lagd(a(1),N(1));
S_in(:,1:N(1)) = B(:,1)*L0';
In_s = 1;     %calculate first cost function parameter for each input
for jj = 2:n_in
    [Al,L0] = lagd(a(jj),N(jj));
    In_s = N(jj-1) + In_s;
    In_e = In_s + N(jj) - 1;
    S_in(:,In_s:In_e) = B(:,jj)*L0';
end
```

```
S_sum = S_in;
phi = S_in;
Omega = (phi)'*Q*(phi);
Psi = phi'*Q*A;
Setwei = (C*phi)';
              %calculate rest of cost function parameters over prediction
              %horizon
for i = 2:Np
    Eae = A^i;
    [Al,L0] = lagd(a(1),N(1));
    S_sum(:,1:N(1)) = A*S_sum(:,1:N(1)) + S_in(:,1:N(1))*(Al^(i-1))';
    In_s = 1;
    for kk = 2:n_in
        [Al,L0] = lagd(a(kk),N(kk));
        In_s = N(kk-1) + In_s;
        In_e = In_s + N(kk) - 1;
        S_sum(:,In_s:In_e) = A*S_sum(:,In_s:In_e) + S_in(:,In_s:In_e)*(Al^(i-1))';
    end
    phi = S_sum;
    Omega = Omega + phi'*Q*phi;
    Psi = Psi + phi'*Q*Eae;
    Setwei = Setwei + (C*phi)';
end
Omega = Omega + R_para;
```

A.3. Function: *lagd*

```
function [A,L0] = lagd(a,N)
%Laguerre parameter calculation
% adapted from "Model Predictive Control System Design and Implementation
% using MATLAB(R)" by L. Wang

v(1,1) = a;
L0(1,1) = 1;
for k = 2:N
    v(k,1) = (-a).^(k-2)*(1-a*a);
    L0(k,1) = (-a).^(k-1);
end
L0 = sqrt(1-a*a)*L0;
A(:,1) = v;
for i = 2:N
    A(:,i) = [zeros(i-1,1);v(1:N-i+1,1)];
end
```

A.4. Function: *simucon*

```
function [u1,y1,deltau1,k] =
simucon(xm,u_old,y,sp,Ap,Bp,Cp,N_sim,Omega,Psi,Lzerot,alpha,lambda)
% closed-loop simulation of constrained MPC
% adapted from "Model Predictive Control System Design and Implementation
% using MATLAB(R)" by L. Wang

[m1,n1] = size(Cp);
[~,n_in] = size(Bp);
Xf = [xm; (y-sp(:,1))];  %state feedback variable

u_max = 5;               %limits for control and incremental control
u_min = -5;
deltau_max = 1;
deltau_min = -1;

%% simulation
for kk = 1:N_sim
```

```
    eta = -(Omega\Psi)*Xf;
    deltau = Lzerot*eta;        %incremental control: check for constraints
    if (deltau>deltau_max)
        deltau = alpha*deltau_max;
    end
    if (deltau<deltau_min)
        deltau = alpha*deltau_min;
    end
    u = u_old + alpha*deltau;   %control: check for constraints
    if (u>u_max)
        u = u_max;
    end
    if (u<u_min)
        u = u_min;
    end
    deltau1(:,kk) = deltau;
    u1(1:n_in,kk) = u;
    y1(1:m1,kk) = y;

    xm_old = xm;
    xm = Ap*xm + Bp*u;          %update states
    y = Cp*xm;

    Xf = [(xm - xm_old); (y - sp(:,kk))]; %update state feedback
    u_old = u;
end
k = 0:(N_sim-1);
```

Appendix B: Data CD

The following materials are contained in the accompanying data CD:

- Copy of the dissertation
- Simulation files: MATLAB® scripts and Simulink™ models
- MPC controller objects
- Measurement results

Declaration

I hereby declare that all the material incorporated in this dissertation is my own original unaided work except where specific reference is made by name or in the form of a numbered reference. This dissertation, or any part thereof, has not been submitted for a degree at any other university.

CD Aucamp Place Date