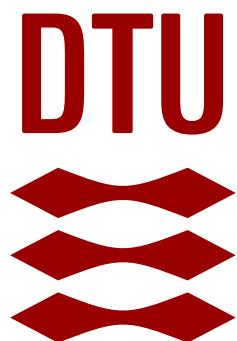


Modelling and Optimal Control
of a Flywheel Energy Storage System:
Theory and Experiments

Master Thesis

Eduard Maximilian Fiedler



January 2023

**Modelling and Optimal Control
of a Flywheel Energy Storage System:
Theory and Experiments**

Master Thesis
January 2023

By
Eduard Maximilian Fiedler

Supervisors
Associate Professor, Roberto Galeazzi PhD
Professor, Dr.-Ing. dr.techn. livre-docente Ilmar Santos

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Approval

This thesis has been prepared over a five-month period under the supervision of Roberto Galeazzi and Ilmar Santos at the Department of Electrical Engineering, with laboratory facilities at the Department of Mechanical Engineering, at the Technical University of Denmark, DTU, in partial fulfilment for the degree of Master of Science in Engineering, MSc Eng.

Eduard Maximilian Fiedler – s210134


Signature

2023-01-15

Date

Abstract

Advancements in material technology, magnetic bearings, and power electronics have turned flywheel energy storage systems (FESSs) into an interesting alternative to traditional battery systems. The ability to store energy in the rotation of the rotor offers effective and efficient storage where fast charging and discharging is required. One of the challenges that presents itself in these systems is the stabilisation of the rotor with the least amount of energy possible.

An FESS test rig has been designed and constructed by the Department of Engineering at the Technical University of Denmark. Its use in a previous project allowed its rotor to only be stabilised through the control of active magnetic bearings, using a PID-PI cascade control scheme.

This project assesses the state of the test rig and mathematically models the dynamics of the system. Optimal methods were applied based on the model to design a linear-quadratic-Gaussian (LQG) controller that was implemented in the experimental setup and to investigate the use of an input-constrained model predictive control (MPC) in simulation. The LQG showed, within the limitations of the setup, that better rotor stability can be achieved with less energy than with the PID-PI controller. For the MPC in simulation, although stability was achieved, it was found that for practical use, a more sophisticated scheme would be required that could accommodate for the real-time constraints.

Acknowledgements

I would like to express my thanks to Roberto for his guidance and willingness to help with his extensive knowledge in control theory. It was extremely refreshing to be able to have very open but also technical discussions on even the very basics.

I would also like to thank Ilmar for providing me with the opportunity of investigating this topic and providing me with the necessary contacts whenever I needed help. I am also grateful for the small project that I was able to undertake which allowed me to familiarise myself with the laboratory environment.

I am extremely grateful to Bruno for the suggestions he has provided to me throughout this project. The discussions we had were very helpful and his willingness to spend time helping me with the difficulties I had is greatly appreciated.

I would also like to extend my thanks to Leonardo for helping me with some of the model derivations, to Thomas and Claus for helping me troubleshoot issues with LabVIEW, and to Alessio for taking the time to fix the VFD in the best way possible.

Throughout my time at DTU I am grateful for having met Tommy, Karol, Szilárd, and everyone from the A0 kitchen in my final semester.

Lastly, I could not have undertaken this journey without the support of my family. Their encouragement and the support I received to pursue my further education provided me with the motivation I needed to do my best.

Contents

1 Introduction	1
1.1 Background information	1
1.2 Problem statement	2
1.3 Relation to sustainable development	3
1.4 Thesis overview and contribution	3
2 Materials	5
2.1 Risk assessment	5
2.2 Test rig	6
2.2.1 Mechanical system	7
2.2.2 Electrical system	9
2.2.3 Control system	9
2.2.4 LabVIEW overview	11
2.2.5 Operation	11
3 Mathematical Modelling	15
3.1 System modelling	15
3.1.1 Relations to the centre of mass	15
3.1.2 Rotor dynamics	19
3.1.3 AMB model	21
3.1.4 PMB model	25
3.1.5 Butterworth filter	29
3.1.6 PID-PI controller	29
3.2 Test rig emulation	31
3.2.1 Differential equation solver	31
3.2.2 Filter implementation	32
3.2.3 Controller implementation	33
4 Optimal Control Methods	35
4.1 LQG control method	35
4.1.1 State-space formulation	36
4.1.2 LQR design	37
4.1.3 Kalman filter design	39

4.1.4 Formulation	40
4.1.5 LabVIEW implementation	43
4.2 MPC method	46
4.2.1 Formulation	46
4.2.2 MATLAB implementation	48
5 Test Procedures	49
5.1 PID model validation	49
5.1.1 Gathering experimental data	49
5.1.2 Experimental data processing	50
5.1.3 Test emulation	51
5.1.4 Comparison	51
5.2 Kalman filter tuning	51
5.2.1 LabVIEW implementation	52
5.2.2 Sensor noise identification	53
5.2.3 Tuning procedure	54
5.3 Kalman filter performance	54
5.4 LQG tuning	55
5.4.1 Obtaining a base controller	56
5.4.2 Tuning procedure	56
5.5 LQG emulation comparison	57
5.6 Step response comparison	57
5.7 Spin response comparison	58
5.8 Spin-up response comparison	58
5.9 Touch-recovery comparison	59
5.10 MPC Simulation	60
6 Results	63
6.1 PID-controlled test rig emulation	63
6.2 Kalman filter performance	66
6.3 LQG-Controlled test rig	70
6.3.1 LQG-Controlled test rig emulation	70
6.3.2 Step response comparison	72
6.3.3 Spin response comparison	75
6.3.4 Spin-up response comparison	77
6.3.5 Touch-recovery comparison	77
6.4 MPC Simulation	81
7 Discussion	83
7.1 PID-controlled system	83
7.2 System emulation	84
7.3 LQG implementation in LabVIEW	85
7.4 Kalman filtering	86
7.5 LQG performance	87

<i>CONTENTS</i>	ix
7.6 Application of MPC	89
8 Conclusion	91
8.1 Limitations	91
8.2 Project summary	92
8.3 Future work	92
A Input parameters	99

List of Figures

2.1 View of the foundation, frame, gimbal, and flywheel housing which make up the test rig as seen in the laboratory.	6
2.2 Schematic of the rotor cross section.	7
2.3 Schematic of housing cross section and the arrangement of the components inside it.	8
2.4 Components inside the electrical cabinet. 1) Test rig power switch, 2) CompactRIO (cRIO), 3) Power supply for cRIO, 4) Power supply for proximity sensors, 5) Power supply for AMB control current, 6) Power amplifiers for control signal to AMBs, 7) Proximity sensor vibration monitor.	9
2.5 AMB power supplies on the left, VFD on the right.	10
2.6 Schematic of the communication between the cRIO and FESS.	10
2.7 AMB power supply. 1) Main power switch, 2) ON/OFF switch.	12
2.8 VFD interface on the left and its power switch on the right, located on the bench next to the VFD.	12
2.9 <i>RealTime_V1</i> front panel for PID-controlled operation.	13
3.1 Rigid rotor definition in its inertial and moving reference frame.	16
3.2 Cross section of rotor and magnet pole pair generating an attraction force.	22
3.3 Opposing bearing magnets producing a differential in attraction force to position the rotor.	23
3.4 Manner of discretisation of one of the ring magnets into wiring and definition of parameters.	25
3.5 Calculated vectors at various positions from the cross-section of the stator magnet.	27
3.6 Analysis of changing the air gap between PMB rotor and stator and the effect on the axial force.	28
3.7 Analysis of changing alignment between PMB rotor and stator with respect to the change in radial force.	28
3.8 PID-PI cascade controller schematic as implemented in the test rig.	30

4.1	LQG control method made up by the LQR and Kalman filter, optimally calculating the reference current for the existing PI controller to follow.	35
4.2	Block diagram of LQG controller showing integration of LQR and Kalman filter components.	41
4.3	Energy of states in the balanced realisation. The first four states had	42
4.4	LabVIEW block diagram communicating matrices from <i>Real-Time V1</i> to <i>FPGA V1</i>	43
4.5	LabVIEW block diagram of the LQG structure used to calculate the control input.	44
4.6	LabVIEW block diagram of the controller selector between PID and LQG.	45
4.7	LabVIEW front panel controls to engage the LQG.	45
5.1	Program for the file transfer between cRIO and local computer.	50
5.2	LabVIEW block diagram of the Kalman filter estimating for the PID-controlled rotor.	52
5.3	FFT on data from position sensor for x_{s1} and current sensor for i_{a1x} . The same characteristics were found in the other position and current sensors.	54
6.1	Step in x_1 performed experimentally and in simulation.	64
6.2	Step in y_1 performed experimentally and in simulation.	64
6.3	Step in x_2 performed experimentally and in simulation.	65
6.4	Step in y_2 performed experimentally and in simulation.	65
6.5	PID-stabilised rotor outputs at 0 rpm estimated using a Kalman filter based on 0 rpm.	67
6.6	PID-stabilised rotor outputs at 2000 rpm estimated using a Kalman filter based on 2000 rpm.	68
6.7	Sum of standard deviations in position and current over a 1 min period at different speeds for different set models for Kalman filtering.	69
6.8	Autocorrelation function on the innovation signal for x_{s1} and i_{a1x} obtained for the Kalman filter based on the 0 rpm model.	69
6.9	Step in x_1 performed experimentally and in simulation.	70
6.10	Step in y_1 performed experimentally and in simulation.	71
6.11	Step in x_2 performed experimentally and in simulation.	71
6.12	Step in y_2 performed experimentally and in simulation.	72
6.13	Differences in response between the LQG and PID-controlled systems to a step in x_1	73
6.14	Differences in response between the LQG and PID-controlled systems to a step in y_1	73

6.15 Differences in response between the LQG and PID-controlled systems to a step in x_2	74
6.16 Differences in response between the LQG and PID-controlled systems to a step in y_2	74
6.17 Rotor spinning at 2000 rpm controlled by the PID controller.	76
6.18 Rotor spinning at 2000 rpm controlled by LQG controller set for 2000 rpm.	76
6.19 Total standard deviation of the combined positions and currents across the available speed range for different controllers.	77
6.20 Position of rotor for LQG set for 2000 rpm at spin-up from 0 to 1000 rpm. The red line indicates the bounds between which the signal oscillated when spinning at steady state.	78
6.21 Current in system for LQG set for 2000 rpm at spin-up from 0 to 1000 rpm. The red line indicates the bounds between which the signal oscillated when spinning at steady state.	78
6.22 PID-controlled rotor recovering after touching boundary.	79
6.23 LQG controller set for 0 rpm compensating for the same impulse that caused the PID to touch the boundary.	80
6.24 LQG controller set for 0 rpm subjected to an impulse causing the rotor to touch the boundary.	80
6.25 Estimation of states in the PID controlled system when the rotor is brought to the boundary.	81
6.26 Positioning of the rotor to the same initial conditions by the MPC, LQG, and PID-controlled systems.	82
6.27 Changing current for the stabilisation of the rotor by the MPC, LQG, and PID controllers to the same initial conditions.	82

List of Tables

3.1 PID-PI controller gains as implemented in the FESS test rig	33
5.1 Overview of the standard deviations calculated in the sensor tests	53
5.2 Overview of the initialised parameters for the PID, LQG, and MPC comparison	60
6.1 Overview of discrepancies between PID-controlled experimen- tal and simulation results in the primary response in position	66
6.2 Summary of discrepancies between LQG-controlled experimen- tal and simulation results in the primary response in position	72
6.3 Improvement of the LQG control strategy over the PID	75
A.1 Parameters used for the calculation of the linearised PMB forces	99
A.2 Input parameters used for modelling the dynamics of the FESS	100
A.3 LQR weights for the tuned LQG controller placed in corre- sponding diagonal matrices	101

Nomenclature

Abbreviations

AMB	Active magnetic bearing
cRIO	CompactRIO
ESS	Energy storage systems
FESS	Flywheel energy storage systems
FFT	Fast Fourier transform
FPGA	Field-programmable gate array
FTP	File-transfer protocol
LQG	Linear-quadratic-Gaussian
MPC	Model predictive control
PI	Proportional-integral
PID	Proportional-integral-derivative
PID-PI	PID-PI operation in cascade
PMB	Passive magnetic bearing
PMSM	Permanent magnet synchronous motor
PWM	Pulse-width modulation
TDMS	Technical data management streaming
VFD	Variable frequency drive
VI	Virtual instrument

Mathematical Symbols

$\dot{()}$	First-derivative in time
$\ddot{()}$	Second-derivative in time
$\mathbf{0}_{n \times m}$	n by m 0 matrix
\mathbf{A}	State matrix of full model description
\mathbf{A}_a	State matrix of model with integral states
A_a	Projected area of pole face onto object
\mathbf{A}_c	State matrix of LQG augmented system
$\mathbf{A}_{r,c}$	State matrix of reduced LQG augmented system
\mathbf{A}_{sys}	State matrix for continuous-time rotor dynamics

B	Input matrix of full model description	
B_a	Magnetic flux density	T
B_a	Input matrix of model with integral states	
B_c	Input matrix of LQG augmented system	
B_r	Radial magnetic flux density by PMB	T
$B_{r,c}$	Input matrix of reduced LQG augmented system	
B_{sys}	Input matrix for continuous-time rotor dynamics	
B_v	Process noise matrix of full model description	
B_z	Axial magnetic flux density by PMB	T
C	Output matrix of full model description	
C_a	Output matrix of model with integral states	
C_c	Output matrix of LQG augmented system	
$C_{r,c}$	Output matrix of reduced LQG augmented system	
C_{sys}	Output matrix for continuous-time rotor dynamics	
\mathcal{C}	Controllability matrix	
E	Energy	J
E	Output matrix for discrete-time rotor dynamics	
\hat{e}	Innovation of augmented Kalman filter system	
F	State matrix for discrete-time rotor dynamics	
F_{a1x}	Force in the x -direction by the upper AMB	N
F_{a2x}	Force in the x -direction by the lower AMB	N
F_{a1y}	Force in the y -direction by the upper AMB	N
F_{a2y}	Force in the y -direction by the lower AMB	N
F_{px}	Force in the x -direction by the PMB	N
F_{py}	Force in the y -direction by the PMB	N
F_{pz}	Force in the z -direction by the PMB	N
G	Input matrix for discrete-time rotor dynamics	
H_a	Magnetic field strength	A m^{-1}
H_s	Input weighting matrix based on depth in MPC	
h	Height of ring magnet in PMB	m
I	Moment of inertia	kg m^2
$I_{n \times m}$	n by m identity matrix	
I_f	Measured and filtered currents from sensors	A
\hat{I}_f	Estimated currents	μm
I_r	x and y -axis moment of inertia	kg m^2
I_{ref}	Reference currents for PI controller	A
I_{xx}	x -axis moment of inertia	kg m^2
I_{yy}	y -axis moment of inertia	kg m^2
I_{zz}	z -axis moment of inertia	kg m^2
i	Current in coil windings	A
i_0	Bias current for differential drive	A
$i_{a,1/2,x/y}$	Current in AMB pole pair	A s^{-1}
i_f	Measured and filtered current in AMB	A

i_{ref}	Reference current for PI controller	A
i_x	Control current for differential drive	A
J	Cost associated with LQR performance index	
\mathbf{K}	Optimal control gain	
$K_{d,PID}$	Derivative gain of PID	$\text{A } \mu\text{m}^{-1}$
\mathbf{K}_f	Kalman filter gain	
$\mathbf{K}_{f,Y}$	Kalman filter gain onto measured positions	
$\mathbf{K}_{f,I}$	Kalman filter gain onto measured currents	
\mathbf{K}_i	Control gain associated with integral states	
$K_{i,PI}$	Integral gain of PI	A^{-1}
$K_{i,PID}$	Integral gain of PID	$\text{A } \mu\text{m}^{-1}$
$K_{p,PI}$	Proportional gain of PI	A^{-1}
$K_{p,PID}$	Proportional gain of PID	$\text{A } \mu\text{m}^{-1}$
\mathbf{K}_x	Control gain associated with system states	
$k_{i,1/2,x/y}$	Force-current factor in AMB	N A^{-1}
k_{pr}	Force-radial-displacement factor in PMB	N m^{-1}
k_{pa}	Force-axial-displacement factor in PMB	N m^{-1}
$k_{s,1/2,x/y}$	Force-displacement factor in AMB	N m^{-1}
$k_{u,1/2,x/y}$	Voltage-velocity factor in AMB	V s m^{-1}
$L_{1/2,x/y}$	Voltage-Change in current with time factor in AMB	V s A^{-1}
\mathcal{L}	Lagrangian	
l_{a1}	Length from c.m. to upper AMB plane of rotor	m
l_{a2}	Length from c.m. to lower AMB plane of rotor	m
l_{nr}	Length of wiring in section n_r	m
l_p	Length from c.m. to PMB plane of rotor	m
l_{s1}	Length from c.m. to upper sensor plane of rotor	m
l_{s2}	Length from c.m. to lower sensor plane of rotor	m
M	Magnetic moment of PMB	A m^{-1}
\mathbf{M}_u	Input weighting vector based on depth in MPC	
N_z	Number of discretisations of PMB in z -direction	
n	Number of coil windings	
n_r	Radial position of assumed wiring in PMB	
n_z	Axial position of assumed wiring in PMB	
n_α	Angular section in ring magnet of PMB	
\mathcal{O}	Observability matrix	
\mathbf{P}	Infinite horizon solution	
PWM	PWM signal generated by PI controller	
\mathbf{Q}	Discrete-algebraic Riccati equation solution	
Q_z	Output weighting matrix in MPC	
Q_z	Output weighting matrix based on depth in MPC	
R	Resistance in AMB coils	Ω
S	Input weighting matrix based on depth in MPC	
\mathbf{r}	Position vector	m

\bar{r}	Reference output in MPC calculation	
$\bar{\mathbf{r}}$	Reference output vector in MPC calculation	
s	Displacement	m
s_0	Initial rotor displacement from magnet	m
\bar{s}	Laplace operator	
\mathbf{T}	Transformation matrix	
T	Similarity transform to balanced realisation	
t_s	System sampling time	s
U	Input voltages to AMBs	V
\mathcal{U}	Input sequence calculated by the MPC	V
\mathcal{U}_{\max}	Maximum input constraint sequence in MPC	V
\mathcal{U}_{\min}	Minimum input constraint sequence in MPC	V
\mathbf{u}	Input currents for full model description	A
u	Input voltage	V
$u_{1/2/3/4}$	Input voltage to respective AMB pole pairs	V
\mathbf{u}_c	Inputs to LQG augmented system	
u_{EMF}	Input voltage as a result of EMF	V
u_{Ohm}	Input voltage as a result of Ohm's law	V
u_{\max}	Maximum input constraint in MPC	V
u_{\min}	Minimum input constraint in MPC	V
\mathbf{u}_{PID}	Inputs calculated by PID	V
\mathbf{u}_{sys}	Input for rotor dynamics	V
\mathbf{v}	Velocity vector	m
\mathbf{v}_1	Process noise vector	
\mathbf{v}_2	Measurement noise vector	
W_a	Field energy	V m^{-1}
W_c	LQG augmented system controllability Gramian	
\mathbf{W}_o	Observability Gramian	
W_o	LQG augmented system observability Gramian	
\mathbf{w}	State vector for Lagrangian	
w	State in Lagrangian	
\mathbf{w}_1	Weighting matrix on states	
\mathbf{w}_2	Weighting matrix on inputs	
x	x -position of rotor in inertial frame	m
\mathbf{x}	States of full model description	
$\hat{\mathbf{x}}$	Estimated states of full model description	
$\bar{\mathbf{x}}$	States of balanced realisation of LQG system	
$x_{a,1/2}$	x -position of rotor in AMB plane	m
\mathbf{x}_a	States of model with integral states	
$\hat{\mathbf{x}}_a$	Estimated states of augmented Kalman filter	
\mathbf{x}_c	States of LQG augmented system	
$\mathbf{x}_{\text{cur,fil}}$	States of filtered currents for full-s.s. formulation	
x_I	x -axis of the inertial frame	

\boldsymbol{x}_i	Integral states	
$\boldsymbol{x}_{\text{PI}}$	States of PI controller for full-s.s. formulation	
x_p	x -position of rotor in PMB plane	m
$\boldsymbol{x}_{\text{pos,fil}}$	States of filtered positions for full-s.s. formulation	
x_R	x -axis of the rotor-fixed frame	
$\bar{\boldsymbol{x}}_r$	States of reduced LQG augmented system	
$\boldsymbol{x}_{\text{sys}}$	States for rotor dynamics	
Y_f	Measured and filtered positions from sensors	μm
\hat{Y}_f	Estimated positions	μm
Y_{ref}	Reference positions in sensor planes for rotor	μm
y	y -position of rotor in inertial frame	m
\boldsymbol{y}	Measured and filtered outputs in vector	
$y_{a,1/2}$	y -position of rotor in AMB plane	m
\boldsymbol{y}_c	Outputs of LQG augmented system	
y_I	y -axis of the inertial frame	
y_p	y -position of rotor in PMB plane	m
y_R	y -axis of the rotor-fixed frame	
$\boldsymbol{y}_{\text{sys}}$	Output of unfiltered measured	
\boldsymbol{Z}	Predicted output for MPC	
z	z -position of rotor in inertial frame	m
\bar{z}	Z-transform domain operator	
\hat{z}	Output in MPC calculation	
z_I	z -axis of the inertial frame	
z_R	z -axis of the rotor-fixed frame	
α	Angular offset of pole face to force direction	rad
$\boldsymbol{\Gamma}$	Matrix for zero-state solution given depth	
γ	Cost function for MPC	
δ	Percentage change between data	
ε	Characteristic obtained from recorded data	
ζ	Arbitrary small angle	rad
η	Arbitrary small angle	rad
θ	Angle about z_I axis	rad
θ_{n_r}	Angle based on n_r	rad
κ	Prediction depth of MPC	
μ_0	Magnetic permeability of free space	H m ⁻¹
σ	Standard deviation of signal in subscript	
$\boldsymbol{\Phi}$	Matrix for zero-input solution given depth	
Φ	Magnetic flux	V s
ϕ	Angle about x_I axis	rad
ψ	Angle about y_I axis	rad
ω	Angular velocity	rad s ⁻¹
$\boldsymbol{\omega}$	Angular velocity vector w.r.t. inertial frame	rad s ⁻¹

Chapter 1

Introduction

1.1 Background information

The global drive towards decarbonising the energy economy has steadily increased the demand for renewable energy resources. However, given their inherent power fluctuations, the mass integration of these sources into current global power grids can destabilise the grid system [1]. To facilitate the integration of intermittent energy sources, energy storage systems (ESSs) have become an increasingly important topic of research [1, 2, 3, 4].

ESSs exist to convert generated electrical energy such that it can be stored for conversion back to electricity when required [3]. This serves to improve the efficiency of intermittent energy sources, as times of low generation cost will not have to coincide with times of high demand. The forms of energy form of energies that have been found viable for storage are chemical, mechanical, thermal, and magnetic [4].

The use of flywheels to store energy mechanically dates back thousands of years in the form of a potter's wheel [5]. More recent advances in material technology, bearings, and power electronics, however, have heightened the viability of flywheel energy storage systems (FESSs) [4]. State-of-the-art flywheels are able to rotate upwards of 100 000 rpm using composite materials of high tensile strength in a vacuum with magnetic bearings to create a near frictionless environment [4, 6]. This enables FESSs to offer a high cycle life, reliability, round-trip efficiency, and high power density, at a low environmental impact. The charge/discharge rates are high in comparison to ESS alternatives [4]. Despite their shortcomings, such as their inability to produce high voltages above 36 kV and a high initial capital cost relative to ESS alternatives as well as the increase in frictional losses with the size of the flywheel [1], FESSs have found use in a number of applications, such as maintenance of power quality, frequency regulation, voltage sag control, transportation and spacecraft [4, 7, 8].

The basic principle of a flywheel is to store energy in the form of kinetic

energy in its rotation. It follows then, that the amount of energy stored, E , is directly proportional to the flywheel's mass, m , to the square of its angular velocity, ω . This is described by the equation:

$$E = \frac{I\omega^2}{2}, \quad (1.1)$$

where I is the moment of inertia of the flywheel. Thus, it is more lucrative to increase the speed of rotation over the mass of the flywheel. In turn, this means that energy density is maximised and that the tensile strength of the rotor becomes one of the main limiting factors in the amount of energy that can be stored [6].

Another limiting factor to the energy capacity is the ability to control the rotating flywheel at speed [9]. For high-performance FESSs, this is achieved through magnetic levitation and stabilisation using magnetic bearings [10, 11, 12]. A simple, robust control system that can be applied is a PID-based controller which acts decentrally [11, 12], however, high rotational speeds and the associated gyroscopic effects can impact the stability significantly [13]. To compensate for the gyroscopic effects, optimal controllers can be used to improve performance [10].

Optimal controllers rely on the definition of a performance index, based on the designated system, which is minimised (or maximised) to best achieve the control requirements set by this index [14]. Linear-quadratic-Gaussian (LQG) control and model-predictive control (MPC) are both optimal control methods that rely on the minimisation of a cost function that is based on the derived dynamics of a system [14, 15]. In LQG control, a linear model of a system is defined in order to obtain gains that observe the system through a Kalman filter and control the system via a linear-quadratic regulator (LQR). As the performance index in this formulation is based on an infinite horizon, the optimal control gains are constant with time [14]. An MPC optimises, like an LQG, but does so at every step in time based on a receding time horizon [15]. As a result, additional constraints to the input and output of the system can be integrated in the control [16], at an increased computational cost.

In terms of FESSs, optimal controllers have been implemented and have shown performance benefits over other methods [10, 17, 18, 19]. However, the MPC implementations were related to power regulation rather than the explicit stabilisation of the flywheel rotor.

1.2 Problem statement

Within the scope of this project, the idea was to implement optimal control strategies on an experimental setup. For this, the setup provided was the FESS test rig investigated in [20], which was originally used to examine the

viability of mobile FESSs in offshore applications. As the rotation of the flywheel creates high gyroscopic effects [13], the use of magnetically levitated flywheels in mobile applications is more complex than in fixed applications. Therefore, the flywheel was mounted on a gimbal to counteract perturbations typically present on a ship.

Further work on the test rig was conducted in [21] with regards to the stabilisation of the flywheel using optimal control methods and in [22] with regards to the energy extraction. As the latter study affected the construction of the system, the dynamics relating to the basis of the optimal controller implemented in [21] were changed. Therefore, the rotor in the test rig could only be stabilised using the original PID-based controller from [20] that was hand-tuned in [22].

The goal of this thesis was, firstly, to reinstate an LQG controller that stabilises the rotor and, secondly, to investigate the viability of an MPC. This combined a thorough examination of the system and a study of the relevant control theory as well as its application with the implementation of the viable controllers to the system for evaluation against the PID controller.

1.3 Relation to sustainable development

The advancement in energy storage systems plays a quintessential role in the viability of renewable power. As the availability of renewable energy sources generally fluctuates, the ability to store this energy when it is available is extremely important [23]. Thus, by investigating the application of methods to make FESSs more effective, the following United Nations Sustainable Development Goals are supported:

- Goal 7: Affordable and clean energy,
- Goal 11: Sustainable cities and communities,
- Goal 12: Responsible consumption and production,
- Goal 13: Climate action.

1.4 Thesis overview and contribution

A brief overview of the following chapters in the thesis is provided to give context to the structure of the project. An outline of each of the chapters indicates the personal contributions to the thesis and the contributions to the project based on previous studies on the test rig.

In Chapter 2, the risks involved during the completion of the project are explained and the test rig on which the experimental tests were carried out on is introduced. The test rig, its components, and its operation, are

explained based on the state in which it was received from the previous project [22], at the start of this project.

In Chapter 3, the modelling process pertaining to the components of the test rig that affect the rotor dynamics is explained. This was carried out using the principles discussed in [20] and applied in a manner similar to [21], based on changes to the system made in [22]. It is then described how the derived model was used to set up a simulation environment to evaluate the model against experimental results.

In Chapter 4, the application of the LQG and MPC methods is explained based on the derived model of the test rig. The LQG method is elucidated more thoroughly, as, in addition to its application in simulation, the required steps to implement the control scheme in the test rig are presented.

In Chapter 5, using the derived models and controllers, the methods used to gather data for an evaluation of the performance are presented. This includes how the simulated original test rig was evaluated, how the LQG was tuned, how the simulated LQG-controlled test rig was evaluated, various benchmarks regarding the LQG performance against the PID, and how the MPC simulation was set up.

In Chapter 6, based on the methods shown, the gathered results are presented and commented.

In Chapter 7, the results are discussed using the knowledge gained throughout the project to provide an explanation for what was found.

In Chapter 8, limitations to the project are outlined, the thesis is summarised, and recommendations for future work on the test rig are discussed.

Chapter 2

Materials

2.1 Risk assessment

Although the flywheel could spin very fast, it remained within a rigid housing that was supported in a stiff frame, rendering the operation of the equipment safe for the operator. Additionally, the moving parts of the test rig were situated behind a transparent barrier, further protecting the operator and anyone in the laboratory from the moving parts.

The equipment could fail in critical ways which were acknowledged before operating the system. Firstly, the method of turning the test rig on and off was very particular in order to prevent damage to the system to the extent possible. This will be elucidated in Sec. 2.2.5. Secondly, the implemented variable frequency drive (VFD) at the time of testing required a lot of current to spin up the flywheel and was only passively cooled. Therefore, at currents in excess of 8 A, the motor was at risk of burning inside the housing, causing permanent damage to the motor. Additionally, over longer periods of operation, the motor would become hot. Thus, it was important to monitor the temperature of the motor in the LabVIEW program so the system was shut down if the temperature exceeded 75 °C. Thirdly, the rotor had the potential to detach itself from its coupled parts, as high acceleration using the permanent magnetic synchronous motor (PMSM) can induce enough shear force for this to occur. Lastly, as the test rig was used in parallel for other research purposes, the disassembly and reassembly of the system as well as the tests being performed had the possibility of causing the system to behave differently.

When evaluating the LQG controller, parameters were initially tuned based on a simulation of the system. However, as the simulation could not emulate the real system closely enough, the effect of the tuning had to be validated experimentally. This had the possibility of causing unintentional behaviours in the control scheme that would lead to high instability and could permanently damage the system. Therefore, the LQG controller was

always initiated carefully as described in Sec. [4.1.5].

2.2 Test rig

As mentioned, the test rig (see Fig. [2.1]) was initially designed to investigate the dynamics of gimbal-mounted flywheels and consequently to improve the viability of FESSs in offshore applications. For this project, however, the pneumatically articulated housing allowing for rotational and translational perturbations as well as the passive gimbal were not utilised. Instead, the test rig was used in its static state with the housing in a locked vertical position.

The test rig is explained in its three main components: the mechanical system (Sec. [2.2.1]), the electrical system (Sec. [2.2.2]), and its the control system (Sec. [2.2.3]). Furthermore, the LabVIEW program associated with the test rig is briefly explained (Sec. [2.2.4]) and the operation of the test rig, as it was initially received, is demonstrated (Sec. [2.2.5]).

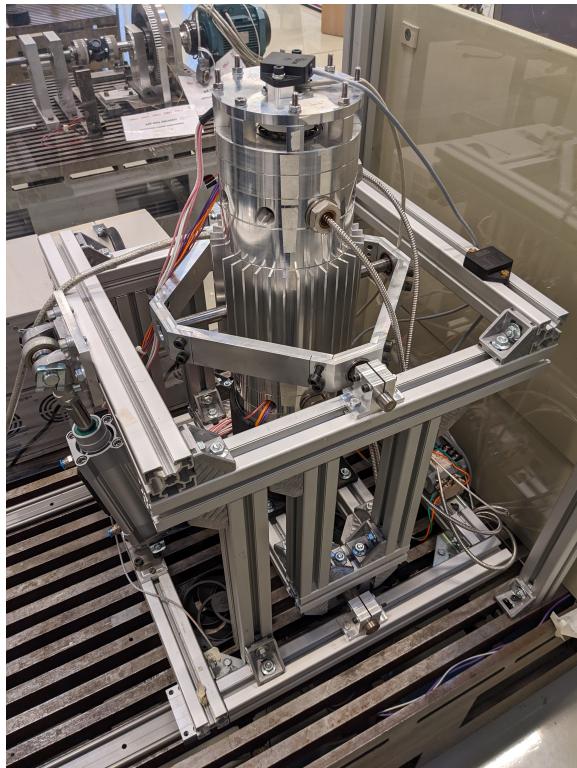


Figure 2.1: View of the foundation, frame, gimbal, and flywheel housing which make up the test rig as seen in the laboratory.

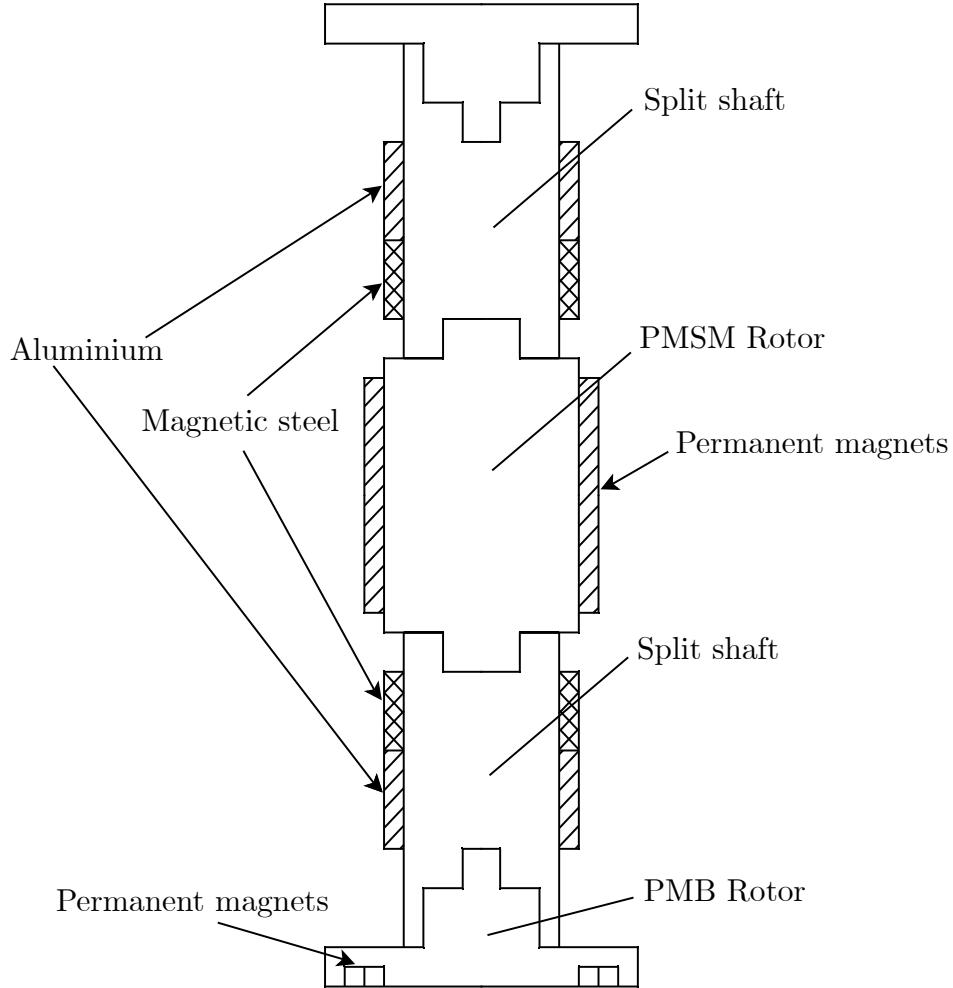


Figure 2.2: Schematic of the rotor cross section.

2.2.1 Mechanical system

The rotor within the housing was made up of a split steel shaft, from the original rotor in [20, 21], which was press fitted onto either end of the PMSM rotor [22] (Fig. 2.2). On either end of the shaft, magnetic steel sheets and aluminium hubs were press fitted onto the surface. The former allowed the active magnetic bearings (AMBs) to produce an attraction force onto the rotor, while the latter provided the position sensors with a suitable surface that avoided electrical runout [20]. Additionally, the rotor part of the passive magnetic bearing (PMB) was integrated into the bottom of the entire rotor, which interacted with the stator part of the PMB at the base of the housing.

The rotor housing (Fig. 2.3) enclosed the rotor and protected the surroundings from the fast spinning flywheel and vice versa. It also housed

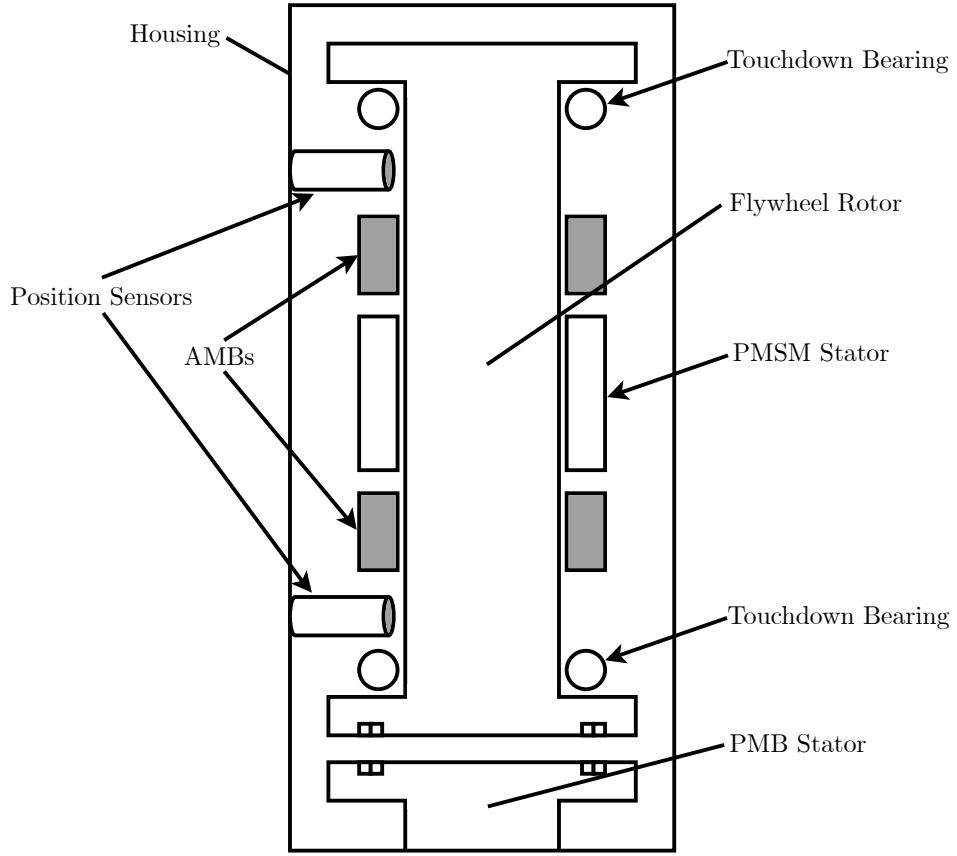


Figure 2.3: Schematic of housing cross section and the arrangement of the components inside it.

the PMB stator at the base, enabling the passive levitation of the rotor. Furthermore, the housing held the AMBs that actuated around the ends of the flywheel to keep the levitation stable, position sensors to track the rotor, the PMSM stator that facilitated the charge and discharge, and touchdown bearings to prevent excessive damage in the event of the rotor stabilisation being compromised.

The PMB between the base of the rotor and housing operated by using two rows of magnets on each side, forming two pairs of ring magnets (bottom of Fig. 2.3). They were arranged on either side such that the outer ring exposed magnets of south polarity, while the inner ring exposed magnets of north polarity. Therefore, both sides would repel each other, imparting an upwards force onto the unrestrained rotor which enabled the passive levitation.

The PMSM used to charge and discharge the flywheel was implemented in [22] because of its high power density and its ability to obtain a very high speed of rotation. This necessitated the attachment of permanent magnets

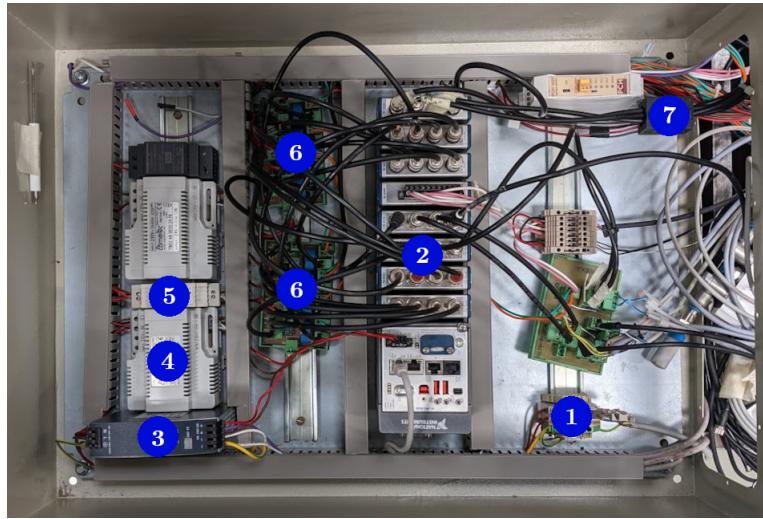


Figure 2.4: Components inside the electrical cabinet. 1) Test rig power switch, 2) CompactRIO (cRIO), 3) Power supply for cRIO, 4) Power supply for proximity sensors, 5) Power supply for AMB control current, 6) Power amplifiers for control signal to AMBs, 7) Proximity sensor vibration monitor.

placed at the centre of the rotor and a VFD to drive the motor. The VFD allowed the manipulation of a magnetic field to alter the angular velocity of the rotor, which required a braking resistor to dissipate energy when decelerating the rotor, and a magnetic encoder to monitor the speed of rotation.

2.2.2 Electrical system

The electrical system of the test rig governed the connectivity between components and their purpose within the implemented control system. Next to the moving parts of the test rig, an electrical cabinet was situated which enclosed the connections between all components of the system. The relevant components for this project are shown in Fig. 2.4

Also located next to the moving parts were the power supplies for the AMBs and the VFD which are shown in Fig. 2.5. Their operation will be elaborated in Sec. 2.2.5.

2.2.3 Control system

The existing control system utilised two controllers working in cascade. The type of controllers used were a PID controller to determine the current needed to keep the rotor stable and a PI controller to match the actual current with the desired one. A detailed explanation of the control strategy is given in Sec. 3.1.0.



Figure 2.5: AMB power supplies on the left, VFD on the right.

The gains for the PID controllers were originally determined in [20] and implemented in the existing LabVIEW program pertaining to the test rig. Some slight adjustments to improve the performance with the updated system were made in [22]. A general schematic of the system and communication of the signals is shown in Fig. 2.6.

It can be seen that the signals from the proximity sensors along with the measured input current were digitised by the CompactRIO (cRIO) and used to calculate an appropriate pulse-width modulated (PWM) signal. This calculation was based on the program created on the LabVIEW Virtual Instrument (VI) that was downloaded onto the field-programmable array (FPGA). The PWM output that resulted from the control strategy was then used by the amplifiers to generate the correct signal to stabilise the rotor. In

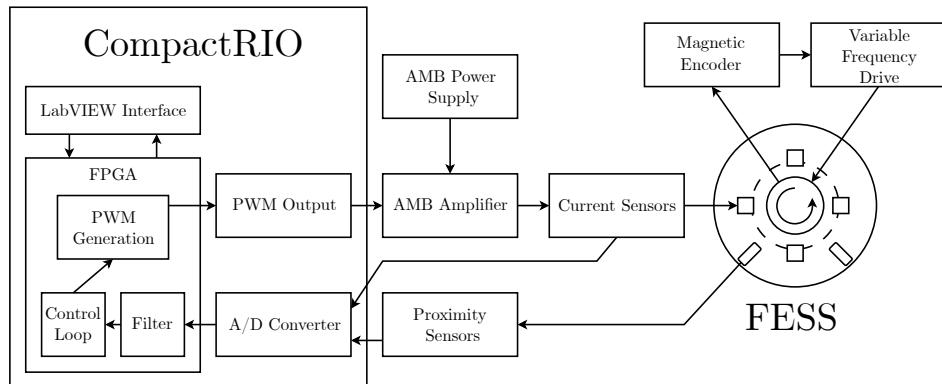


Figure 2.6: Schematic of the communication between the cRIO and FESS.

addition to this part of the system, the VFD, PMSM, braking resistor, and magnetic encoder were used to spin the flywheel up or down.

2.2.4 LabVIEW overview

A LabVIEW program was used to interface with the real-time computer. This program consisted of two major VIs that communicated with each other and with the test rig. These VIs were programmed on a laptop that was able to communicate with the cRIO real-time embedded controller.

The first major VI named *RealTime_V1.vi* was the VI with which the user interacted directly. The purpose of this VI was to accept user input, communicate this to the control system, and to receive information about the test rig. In this VI, the sensors were calibrated, controller gains were set, test signals (chirp, impulse, step, etc.) were initiated, and data recordings were started. Additionally, an interface was provided to track the state of the system in terms of the position of the rotor, the current being applied, the speed of rotation, the temperature of the motor, etc.

The other major VI named *FPGA_V1.vi* was the VI that contained the data processing and control of the test rig. Based on the input provided by the user through the *RealTime_V1* program, the *FPGA_V1* program would filter sensor data and apply the selected control strategy to the test rig. Furthermore, all the collected data was arranged and communicated to the *RealTime_V1* program for the user to interface with.

When working with these programs, it must be stressed that changes made to *FPGA_V1.vi* required a compilation of the program onto the cRIO which would take upwards of 1 hour. Changes to *RealTime_V1* were not did not need to be compiled onto the cRIO, as it was only used to communicate with the real-time computer.

2.2.5 Operation

The stabilisation of the rotor based on the previously programmed PID control strategy was achieved by first turning on all the necessary equipment. Firstly, the power switch from Fig. 2.4 was flipped from the open to the closed position. The hinged mechanism, which was detachable, contained a fuse that protected the equipment. It was found that flipping the switch very slowly, led to the burning of the fuse and made the system fail to function. Therefore, with a functioning fuse, the switch had to be closed quickly and deliberately. Next, the AMB power supplies were switched on by pressing all 4 main power switches and then all 8 ON/OFF switches (Fig. 2.7). The displays would all show 15 V and 1 A. Lastly, the VFD was switched on by turning the red switch to the right of the box clockwise into the ON position (Fig. 2.8). The display on the VFD would show a speed in rpm that could be set by the blinking cursor using the arrow buttons.

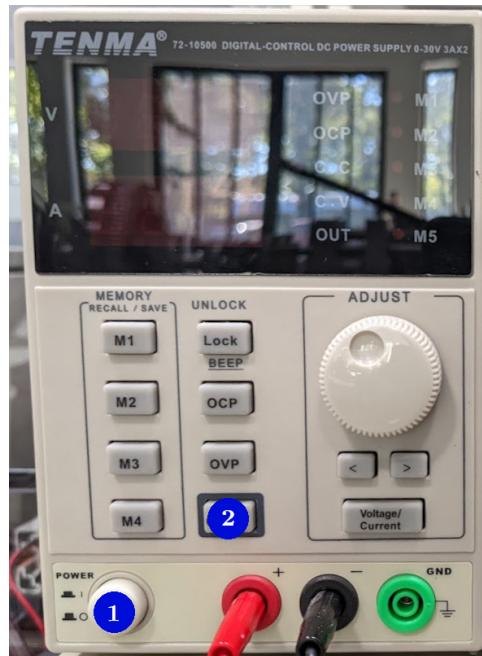


Figure 2.7: AMB power supply. 1) Main power switch, 2) ON/OFF switch.



Figure 2.8: VFD interface on the left and its power switch on the right, located on the bench next to the VFD.

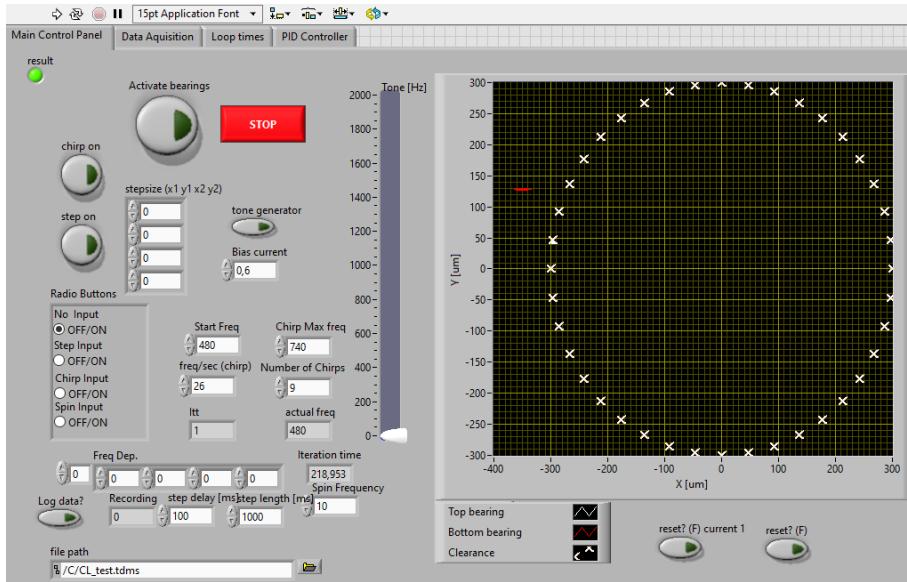


Figure 2.9: *RealTime_V1* front panel for PID-controlled operation.

At this point, the LabVIEW program was initiated to stabilise the rotor using the PID control strategy that was previously implemented in [20] and adjusted by [22]. To do this, the folder containing the LabVIEW program was located, then, *MAIN.lvproj* was opened and *RealTime_V1.vi* started. The front panel then appeared and the program was run by first pressing the arrow in the top left (see Fig. 2.9). Next, the two reset buttons, seen in the bottom-right of Fig. 2.9, were pressed twice to reset the state of the PID and PI controllers. After this, the *activate bearings* button was pressed which initiated the stabilisation of the rotor. Near the centre of the front panel (right of Fig. 2.9), the position of the rotor in the sensor planes was shown in a real-time plot. With the controller working correctly, the position indicated by the top and bottom of the rotor would move to the centre of the plot.

As the rotor was now being stabilised towards the centre, it could be spun. When spinning the rotor, it was important that the speed did not exceed 2000 rpm as the controller for in the VFD was drawing high amounts of current due to a fault during the tests in [22]. The VFD indicated the amount of current being used on the display, and prolonged usage at over 8 A would lead to a burn of components pertaining to the motor (top of display in Fig. 2.8). To spin the rotor, the *hand-on* button was pressed, then, the cursor was moved to the desired position, using the left and right arrow buttons, to set the rpm. Steps of 100 rpm were used to alter the speed, by pressing the up and down arrow buttons, but only once the speed of the rotor matched the desired speed (top-right of display in Fig. 2.8). However,

it was possible to go from complete standstill to 1000 rpm. Bigger changes in speed, up or down, could lead to the motor exceeding the limited range or to the VFD losing track of the rotation which compromised the operation of the system. When the rotor was not spinning, the *off* button was pressed, as this prevented accidental changes to the desired rpm from being carried out.

When shutting down the system, with the rotor at 0 rpm, the *activate bearings* button was deactivated and the *stop* button was pressed to halt the program (Fig. 2.9). Next, the power switches on all 4 AMB power supplies were turned off and the main power switch in the electrical cabinet was opened. Finally, the switch for the VFD was turned anticlockwise to the off position. Due to the way the VFD was wired, it was possible for the breaker to trip when the VFD was turned off. In this case, the breaker would have to be directed back into the active position for any subsequent tests.

Chapter 3

Mathematical Modelling

3.1 System modelling

To model the test rig, the centre of mass was first related to points of interest (Sec. 3.1.1), and, using this, the rotor dynamics were derived (Sec. 3.1.2). Next, the interaction between active and passive magnetic bearings with the rotor were described mathematically (see Sec. 3.1.3 & 3.1.4). Finally, the filter on the system outputs (Sec. 3.1.5) and the cascade control scheme (Sec. 3.1.6), which were implemented in LabVIEW were described in equation form.

3.1.1 Relations to the centre of mass

The rotor as shown in Fig. 3.1 was being manipulated or sensed at 5 planes that did not coincide with its centre of mass. As the rotor and electrical dynamics derived in Sec. 3.1.2 & 3.1.3 were to be based around the centre of mass, a relation of the position and velocity of the rotor in these planes had to be established.

With respect to the inertial frame described by the x_I , y_I , z_I -axes (refer to Fig. 3.1), rotation matrices were described by the following transformation matrices:

$$\mathbf{T}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}, \quad (3.1)$$

$$\mathbf{T}_\psi = \begin{bmatrix} \cos \psi & 0 & -\sin \psi \\ 0 & 1 & 0 \\ \sin \psi & 0 & \cos \psi \end{bmatrix}, \quad (3.2)$$

$$\mathbf{T}_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

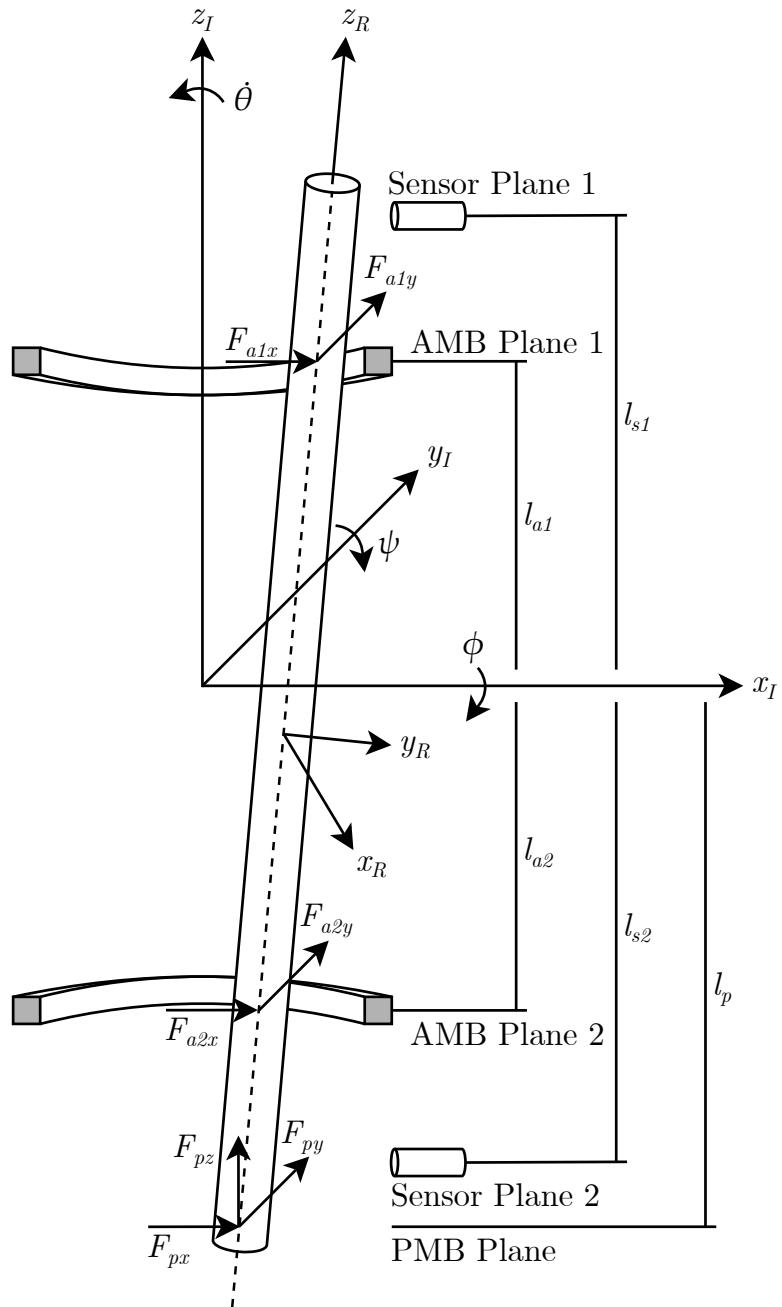


Figure 3.1: Rigid rotor definition in its inertial and moving reference frame.

Next, angular velocities about their axes were described as:

$$\boldsymbol{\omega}_\phi = [\dot{\phi} \ 0 \ 0]^\top, \quad (3.4)$$

$$\boldsymbol{\omega}_\psi = [0 \ \dot{\psi} \ 0]^\top, \quad (3.5)$$

$$\boldsymbol{\omega}_\theta = [0 \ 0 \ \dot{\theta}]^\top. \quad (3.6)$$

Then, by sequentially relating rotating frames of reference to a single one, the angular velocities in the rotor-fixed frame were obtained as follows:

$$\boldsymbol{\omega} = \mathbf{T}_\theta \boldsymbol{\omega}_\theta + \mathbf{T}_\theta \mathbf{T}_\psi \boldsymbol{\omega}_\psi + \mathbf{T}_\theta \mathbf{T}_\psi \mathbf{T}_\phi \boldsymbol{\omega}_\phi, \quad (3.7)$$

$$= \begin{bmatrix} \dot{\psi} \sin \theta + \dot{\phi} \cos \psi \cos \theta \\ \dot{\psi} \cos \theta - \dot{\phi} \cos \psi \sin \theta \\ \dot{\theta} + \dot{\phi} \sin \psi \end{bmatrix}. \quad (3.8)$$

The position of the rotor in the planes of interest was derived by relating the centre of mass in the inertial frame:

$${}_I \mathbf{r}_{cm} = [x \ y \ z]^\top, \quad (3.9)$$

to the approximate position of the planes in the rotor-fixed frame:

$$\mathbf{r}_{s1} = [0 \ 0 \ l_{s1}]^\top, \quad (3.10)$$

$$\mathbf{r}_{a1} = [0 \ 0 \ l_{a1}]^\top, \quad (3.11)$$

$$\mathbf{r}_{a2} = [0 \ 0 \ -l_{a2}]^\top, \quad (3.12)$$

$$\mathbf{r}_{s2} = [0 \ 0 \ -l_{s2}]^\top, \quad (3.13)$$

$$\mathbf{r}_p = [0 \ 0 \ -l_p]^\top. \quad (3.14)$$

These positions were then related to the inertial frame as follows:

$$I\mathbf{r}_i = I\mathbf{r}_{cm} + \mathbf{T}_\phi^\top \mathbf{T}_\psi^\top \mathbf{T}_\theta^\top \mathbf{r}_i. \quad (3.15)$$

By substituting the position of the planes for \mathbf{r}_i , the position vectors were derived as:

$$I\mathbf{r}_{s1} = \begin{bmatrix} x + l_{s1} \sin \psi \\ y - l_{s1} \sin \phi \cos \psi \\ z + l_{s1} \cos \phi \cos \psi \end{bmatrix}, \quad (3.16)$$

$$I\mathbf{r}_{a1} = \begin{bmatrix} x + l_{a1} \sin \psi \\ y - l_{a1} \sin \phi \cos \psi \\ z + l_{a1} \cos \phi \cos \psi \end{bmatrix}, \quad (3.17)$$

$$I\mathbf{r}_{a2} = \begin{bmatrix} x - l_{a2} \sin \psi \\ y + l_{a2} \sin \phi \cos \psi \\ z - l_{a2} \cos \phi \cos \psi \end{bmatrix}, \quad (3.18)$$

$$I\mathbf{r}_{s2} = \begin{bmatrix} x - l_{s2} \sin \psi \\ y + l_{s2} \sin \phi \cos \psi \\ z - l_{s2} \cos \phi \cos \psi \end{bmatrix}, \quad (3.19)$$

$$I\mathbf{r}_p = \begin{bmatrix} x - l_p \sin \psi \\ y + l_p \sin \phi \cos \psi \\ z - l_p \cos \phi \cos \psi \end{bmatrix}. \quad (3.20)$$

Note that these expressions do not exactly define the part of the rotor within the plane, but the position of the rotor that intersects the planes when steady in the inertial frame. This simplification, as a result, becomes invalid when the tilting angles ϕ and ψ are very high, which was not the case as deviations from the centre were limited to 300 µm by the touchdown bearings.

The position vectors were then differentiated with respect to time in order to obtain the approximate velocity of the rotor in the respective planes:

$$\mathbf{v}_{s1} = \begin{bmatrix} \dot{x} + \dot{\psi}l_{s1} \cos \psi \\ \dot{y} + l_{s1} (\dot{\psi} \sin \phi \sin \psi - \dot{\phi} \cos \phi \cos \psi) \\ \dot{z} - l_{s1} (\dot{\psi} \cos \phi \sin \psi + \dot{\phi} \sin \phi \cos \psi) \end{bmatrix}, \quad (3.21)$$

$$\mathbf{v}_{a1} = \begin{bmatrix} \dot{x} + \dot{\psi}l_{a1} \cos \psi \\ \dot{y} + l_{a1} (\dot{\psi} \sin \phi \sin \psi - \dot{\phi} \cos \phi \cos \psi) \\ \dot{z} - l_{a1} (\dot{\psi} \cos \phi \sin \psi + \dot{\phi} \sin \phi \cos \psi) \end{bmatrix}, \quad (3.22)$$

$$\mathbf{v}_{a2} = \begin{bmatrix} \dot{x} - \dot{\psi}l_{a2} \cos \psi \\ \dot{y} - l_{a2} (\dot{\psi} \sin \phi \sin \psi - \dot{\phi} \cos \phi \cos \psi) \\ \dot{z} + l_{a2} (\dot{\psi} \cos \phi \sin \psi + \dot{\phi} \sin \phi \cos \psi) \end{bmatrix}, \quad (3.23)$$

$$\mathbf{v}_{s2} = \begin{bmatrix} \dot{x} - \dot{\psi}l_{s2} \cos \psi \\ \dot{y} - l_{s2} (\dot{\psi} \sin \phi \sin \psi - \dot{\phi} \cos \phi \cos \psi) \\ \dot{z} + l_{s2} (\dot{\psi} \cos \phi \sin \psi + \dot{\phi} \sin \phi \cos \psi) \end{bmatrix}, \quad (3.24)$$

$$\mathbf{v}_p = \begin{bmatrix} \dot{x} - \dot{\psi}l_p \cos \psi \\ \dot{y} - l_p (\dot{\psi} \sin \phi \sin \psi - \dot{\phi} \cos \phi \cos \psi) \\ \dot{z} + l_p (\dot{\psi} \cos \phi \sin \psi + \dot{\phi} \sin \phi \cos \psi) \end{bmatrix}. \quad (3.25)$$

3.1.2 Rotor dynamics

The flywheel in Fig. 3.1 was controlled radially by AMBs at both ends and axially by the PMB at the bottom. As the rotor was being levitated, it had 6 degrees of freedom, one of which was its rotation about the z -axis, which was controlled by the motor.

When deriving the equations of motion, the following assumptions were made:

- The rotor was considered symmetric and rigid.
- The initial starting position of the rotor was where the final moving reference frame coincided with the inertial reference frame.
- Displacements experienced by the rotor from the starting position were small compared to the rotor dimensions, making the linearisation of the dynamics more valid.
- The rotational speed of the rotor was assumed to be constant.
- The effect of the motor on the dynamics was considered negligible.

Displacements of the rotor were described by x, y, z with respect to the centre of mass, while tilting of the rotor was described by the angles ϕ and ψ . The equations of motion were derived to describe these states:

$$\mathbf{w} = [x \ y \ z \ \phi \ \psi]^\top, \quad (3.26)$$

using Lagrange's equations given by:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{w}_i} \right) - \frac{\partial \mathcal{L}}{\partial w_i} = F_i, \quad (3.27)$$

where \mathcal{L} describes the difference between the kinetic and potential energy, w_i as one of the states in \mathbf{w} , and F_i the generalised forces present for that state. The potential energy of the rotor was neglected, as the rotor position did not change significantly along the z -axis. Therefore, \mathcal{L} was only described by the kinetic energy and was given by:

$$\mathcal{L} = \frac{m}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2} (I_{xx}\omega_x^2 + I_{yy}\omega_y^2 + I_{zz}\omega_z^2), \quad (3.28)$$

$$= \frac{m}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2} \left[I_r (\dot{\psi}^2 + \dot{\phi}^2 \cos^2 \psi) + I_{zz} (\dot{\theta}^2 + 2\dot{\theta}\dot{\phi} \sin \psi + \dot{\phi}^2 \sin^2 \psi) \right], \quad (3.29)$$

where the rotational speed components $\omega_x, \omega_y, \omega_z$ came from the derived absolute angular velocity vector (Eq. 3.8) and $I_r = I_{xx} = I_{yy}$. The generalised forces were obtained in the following way:

$$F_i = F_j \frac{\partial}{\partial w_i} ({}_I \mathbf{r}_j). \quad (3.30)$$

Here, a given force F_j acting on the rotor, as seen in Fig. 3.1, was part of the generalised forces for state w_i , if the partial derivative with respect to this state in its position ${}_I \mathbf{r}_j$ in the inertial frame (refer to Eq. 3.16–3.20), was non-zero.

Applying the Lagrangian equation (Eq. 3.27) to the Lagrangian obtained (Eq. 3.29), the equations of motion for the rotor dynamics were obtained as:

$$\ddot{x} = \frac{1}{m} (F_{a1x} + F_{a2x} + F_{px}), \quad (3.31)$$

$$\ddot{y} = \frac{1}{m} (F_{a1y} + F_{a2y} + F_{py}), \quad (3.32)$$

$$\ddot{z} = \frac{1}{m} F_{pz}, \quad (3.33)$$

$$\ddot{\phi} = \frac{1}{I_r} (-I_{zz}\dot{\theta}\dot{\psi} - F_{a1y}l_{a1} + F_{a2y}l_{a2} + F_{py}l_p), \quad (3.34)$$

$$\ddot{\psi} = \frac{1}{I_r} (I_{zz}\dot{\theta}\dot{\phi} + F_{a1x}l_{a1} - F_{a2x}l_{a2} - F_{px}l_p). \quad (3.35)$$

To obtain this, the following small angle approximations were used:

$$\sin \eta = \eta, \quad (3.36)$$

$$\cos \eta = 1, \quad (3.37)$$

$$\eta\zeta = 0, \quad (3.38)$$

$$\dot{\eta}\zeta = 0, \quad (3.39)$$

where the representative angles η and ζ may be replaced with either tilting angle ϕ or ψ . The speed of rotation $\dot{\theta}$ was not subjected to these simplifications, as it was possible for it to be very high in comparison to the tilting angles.

3.1.3 AMB model

The AMBs were used to stabilise the positioning of the flywheel by applying radial forces to the rotor. This was derived under the assumption that the effect of the iron on the magnetic forces was negligible, as the permeability of iron is considered significantly larger than that of the air [24, pg. 74].

The attraction force of the AMB was generated based on a change of field energy W_a , with respect to a change in the displacement s :

$$f = -\frac{\partial W_a}{\partial s}. \quad (3.40)$$

This field energy may be described as follows:

$$W_a = \frac{1}{2} B_a H_a (2A_a s), \quad (3.41)$$

$$= \frac{1}{\mu_0} B_a^2 A_a s, \quad (3.42)$$

where B_a is the magnetic flux density, H_a is the magnetic field strength, μ_0 is the permeability of the free space, and A_a is the cross-sectional area of the air gap [24, pg. 77]. Thus, $2A_a s$ described the volume of space in which the field energy is stored. The negative sign in Eq. 3.40 arose due to the magnet being assumed stationary and due to the assumption that increases in the air gap s corresponded to movements of the rotor away from the magnet (see Fig. 3.2).

In the AMB, the amount of attractive force was regulated based on the current sent to the pole pair. To represent this, the flux density was written in terms of the number of coil windings n , the current sent to the electromagnet i , and the change in the air gap s :

$$B_a = \mu_0 \frac{ni}{2s}, \quad (3.43)$$

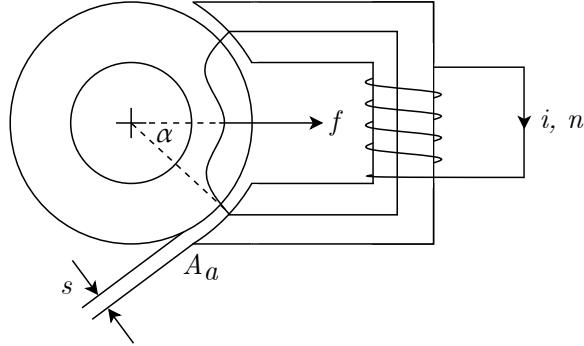


Figure 3.2: Cross section of rotor and magnet pole pair generating an attraction force.

resulting in the representation of the force in the form:

$$f = \frac{1}{4} \mu_0 n^2 A_a \frac{i^2}{s^2} \cos \alpha, \quad (3.44)$$

$$= k \frac{i^2}{s^2} \cos \alpha, \quad (3.45)$$

where A_a is the assumed projected area of the pole face onto the ferromagnetic body and the term $\cos \alpha$ characterising the offset of the pole faces to the generated force direction (see Fig. 3.2) [24, pg. 78]. For the AMBs used, the 8-pole heteropolar configuration meant that the angular offset was considered to be $\alpha = 22.5^\circ$.

Knowing the expression for the attraction force of one pole pair, Eq. 3.45 was expanded, as the rotor was positioned in the AMB plane using an attraction force differential between oppositely positioned pole pairs (see Fig. 3.3). The basis for the differential in attraction force was the use of a bias current i_0 which was manipulated by the control current i_x by the same magnitude but in opposite directions for the respective pole pairs. Therefore, the resultant force generated was expressed as:

$$f_x = k \left(\frac{(i_0 + i_x)^2}{(s_0 - x)^2} - \frac{(i_0 - i_x)^2}{(s_0 + x)^2} \right) \cos \alpha, \quad (3.46)$$

$$\text{where, } k = \frac{1}{4} \mu_0 n^2 A_a. \quad (3.47)$$

Here, s_0 is the initial air gap between the rotor and pole, assumed to be the same on both sides, and x , the position of the rotor along the direction of the force [24, pg. 80]. Eq. 3.46 was then linearised around the operating point where x and i_x were 0, using a Taylor's expansion up to the first-order partial derivative with respect to x and i , giving:

$$f_x = \frac{4ki_0}{s_0^2} (\cos \alpha) i_x + \frac{4ki_0^2}{s_0^3} (\cos \alpha) x, \quad (3.48)$$

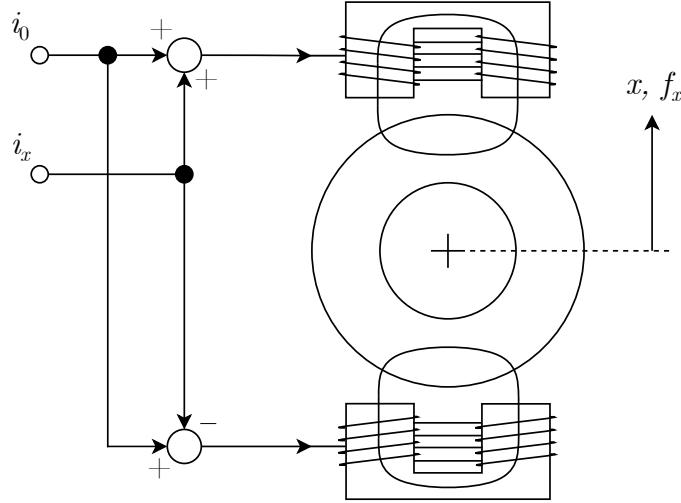


Figure 3.3: Opposing bearing magnets producing a differential in attraction force to position the rotor.

which was rewritten as:

$$f_x = k_i i_x - k_s x, \quad (3.49)$$

$$\text{where, } k_i = \frac{4k i_0}{s_0^2} \cos \alpha, \quad (3.50)$$

$$k_s = -\frac{4k i_0^2}{s_0^3} \cos \alpha. \quad (3.51)$$

With the resulting force from the differential attraction force in an orthogonal direction derived, the radial AMB forces in the two AMB planes were defined as:

$$F_{a1x} = k_{i1x} i_{a1x} - k_{s1x} x_{a1}, \quad (3.52)$$

$$F_{a1y} = k_{i1y} i_{a1y} - k_{s1y} y_{a1}, \quad (3.53)$$

$$F_{a2x} = k_{i2x} i_{a2x} - k_{s2x} x_{a2}, \quad (3.54)$$

$$F_{a2y} = k_{i2y} i_{a2y} - k_{s2y} y_{a2}, \quad (3.55)$$

where the factors k_i and k_s can be unique to the respective pole pairs, the currents i described the control current manipulating the force, and the positions x and y were based on the previously derived position relations regarding the AMB plane from Eq. 3.17 & 3.18.

Next, as the AMB was being actuated via the manipulation of the control current through an input voltage, the electrical dynamics had to be considered. Additionally, the movement of the rotor in the AMB plane would also induce a current in the wiring which was also included. Thus:

$$u = u_{\text{Ohm}} + u_{\text{EMF}}, \quad (3.56)$$

representing that the voltage in the system u was changing based on Ohm's law u_{Ohm} , and the changing electromagnetic field u_{EMF} .

The voltage based on Ohm's law is defined by:

$$u_{\text{Ohm}} = iR, \quad (3.57)$$

where R was the resistance present in the AMB coils, and i the current that passed through the wiring. The induced voltage by the changing magnetic field was given by:

$$u_{\text{EMF}} = n \frac{d\Phi}{dt}, \quad (3.58)$$

which is caused by the change in flux Φ over time [24, pg. 72]. Neglecting the magnetisation of the iron in the AMB [24, pg. 74–75], this may be rewritten as:

$$u_{\text{EMF}} = n \frac{d}{dt} (B_a A_a), \quad (3.59)$$

$$= \frac{\mu_0 n^2 A_a}{2} \frac{d}{dt} \left(\frac{i}{s} \right). \quad (3.60)$$

Eq. 3.60 was then linearised via a Taylor's expansion up to the first-order derivative with respect to i and \dot{s} around the operation point at the initial air gap s_0 , and the current equal to the bias current i_0 , arriving at:

$$u_{\text{EMF}} = \frac{\mu_0 n^2 A_a}{2s_0} i - \frac{\mu_0 n^2 A_a i_0}{2s_0^2} \dot{s}. \quad (3.61)$$

The expression was rewritten in the form:

$$u_{\text{EMF}} = L \dot{i} + k_u \dot{s}, \quad (3.62)$$

$$\text{where, } L = \frac{\mu_0 n^2 A_a}{2s_0}, \quad (3.63)$$

$$k_u = -\frac{\mu_0 n^2 A_a i_0}{2s_0^2}. \quad (3.64)$$

Combining Eq. 3.57 & 3.62, the voltage in the AMB was defined as:

$$u = Ri + L \dot{i} + k_u \dot{s}, \quad (3.65)$$

which, when rearranged in terms of the change in current and expressed in terms of the AMB planes, gave:

$$\dot{i}_{a1x} = \frac{1}{L_{x1}} (u_1 - Ri_{a1x} - k_{u1x} \dot{x}_{a1}), \quad (3.66)$$

$$\dot{i}_{a1y} = \frac{1}{L_{y1}} (u_2 - Ri_{a1y} - k_{u1y} \dot{y}_{a1}), \quad (3.67)$$

$$\dot{i}_{a2x} = \frac{1}{L_{x2}} (u_3 - Ri_{a2x} - k_{u2x} \dot{x}_{a2}), \quad (3.68)$$

$$\dot{i}_{a2y} = \frac{1}{L_{y2}} (u_4 - Ri_{a2y} - k_{u2y} \dot{y}_{a2}), \quad (3.69)$$

where the velocity terms were based on the previously derived relations regarding the AMB plane from Eq. 3.22 & 3.23, and L & k_u can be considered unique for each pole pair. These equations described the electrical dynamics of the FESS by expressing the evolution of the *control* current present in the AMB pole pairs with time.

3.1.4 PMB model

To model the physics of the PMB, the ring magnets at both the bottom of the rotor, PMB rotor (Fig. 2.2), and the base of the housing, PMB stator (Fig. 2.3), were approximated as rings of wiring (see Fig. 3.4). Using this method, the magnetisation in the magnets was approximated with a current through the lower assumed wirings, according to the Biot-Savart law. Then, the force onto the bottom of the rotor in the PMB plane (Fig. 3.1) was calculated by virtue of the Lorentz force law.

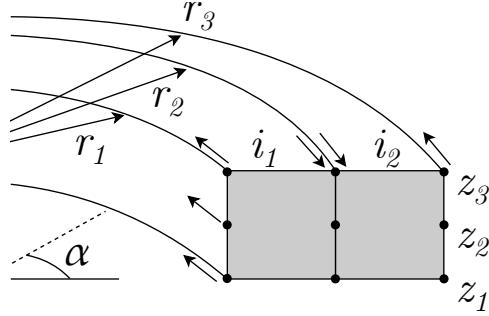


Figure 3.4: Manner of discretisation of one of the ring magnets into wiring and definition of parameters.

To calculate the magnetic flux density caused by the stator magnet, the following equations for the radial and axial directions [20, 25], respectively, were used:

$$B_r(r, z) = 2M \left(\frac{a_1}{2r} \left[\left(1 + k'_1 \right)^2 K(k_1^2) - 2E(k_1^2) \right] - \frac{a_0}{2r} \left[\left(1 + k'_0 \right)^2 K(k_0^2) - 2E(k_0^2) \right] \right) \cdot 10^{-7}, \quad (3.70)$$

$$B_z(r, z) = 2M \left(\frac{\gamma_1}{a_1} \left[K(k_1^2) + \frac{r' - r}{r' + r} \Pi(\nu^2, k_1^2) \right] - \frac{\gamma_0}{a_0} \left[K(k_0^2) + \frac{r' - r}{r' + r} \Pi(\nu^2, k_0^2) \right] \right) \cdot 10^{-7}. \quad (3.71)$$

In these equations, K , E , and Π are the complete elliptic integrals of the first, second, and third kind, respectively. The subscripts 0 and 1 indicate whether the parameter was in reference to the upper or lower current wiring, respectively, in the approximated stator magnet (Fig. 3.4). Additionally, the

parameters used were defined as follows:

$$\gamma = z' - z, \quad (3.72)$$

$$a^2 = \gamma^2 + (r' + r)^2, \quad (3.73)$$

$$k^2 = \frac{4rr'}{a^2}, \quad (3.74)$$

$$k'^2 = 1 - k^2, \quad (3.75)$$

$$\nu^2 = \frac{4rr'}{(r' + r)^2}, \quad (3.76)$$

where r and z were the positions in space for which the vector of the magnetic field was being calculated, while r' and z' were the positions of the wiring in the discretised stator magnet.

In this case, based on the assumptions made for the arrangement of the magnets, the radial and axial components of the magnetic field were calculated by:

$$B_r(r, z) = \sum_{n_r=1}^3 \begin{cases} -B_{r,n_r}(r, z), & n_r = 1, 3 \\ 2B_{r,n_r}(r, z), & n_r = 2 \end{cases}, \quad (3.77)$$

$$B_z(r, z) = \sum_{n_r=1}^3 \begin{cases} -B_{z,n_r}(r, z), & n_r = 1, 3 \\ 2B_{z,n_r}(r, z), & n_r = 2 \end{cases}, \quad (3.78)$$

where n_r denotes the position of r' from which the magnetic field was being calculated from in the inner (1), middle (2), or outer (3) radial position of the stator magnet. By applying these assumptions, a vector field was generated to show the application of these equations (see Fig. 3.5).

Having calculated the magnetic field density, the Lorentz force law was applied based on a discretisation of the rotor magnet into current sheets like in Fig. 3.4, but at a height relative to the centre of the PMB stator magnets. The force was then described by:

$$\mathbf{F} = \int \mathbf{M} \times \mathbf{B} dA, \quad (3.79)$$

which simplified to:

$$\mathbf{F} = I \int dl \times \mathbf{B}, \quad (3.80)$$

given the wiring assumption being applied. The current I was based on the magnetic moment \mathbf{M} , the height h of the magnet, and number of wires used to discretise in the z -direction N_z . The integral was based on an infinitesimal

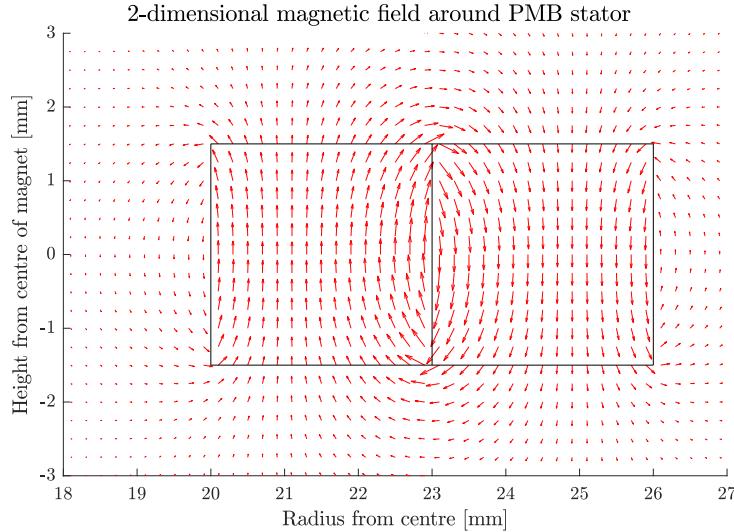


Figure 3.5: Calculated vectors at various positions from the cross-section of the stator magnet.

discretisation of the assumed wiring dl , and depended on the amount of sections into which the ring magnet was split. As a result, the force vector in Cartesian coordinates was found using the following equations:

$$\mathbf{F} = M \frac{h}{N_z} \sum_{n_\alpha=1}^6 \sum_{n_z=1}^3 \sum_{n_r=1}^3 \begin{cases} f & n_r = 1, 3 \\ -2f & n_r = 2 \end{cases}, \quad (3.81)$$

$$\text{where, } f = \begin{bmatrix} -(\sin \theta_{n_\alpha}) l_{n_r} \\ (\cos \theta_{n_\alpha}) l_{n_r} \\ 0 \end{bmatrix} \times \begin{bmatrix} (\cos \theta_{n_\alpha}) B_{r,n_z} \\ (\sin \theta_{n_\alpha}) B_{r,n_z} \\ B_{z,n_z} \end{bmatrix}. \quad (3.82)$$

The upper ring magnet was discretised into 3 radial lengths, at 3 different heights, and 6 angular sections. Therefore, θ_{n_α} was defined based on the angular section being analysed n_α , and l_{n_r} being based on the number of angular sections and radial length. Essentially, the equation identifies the field density at the discretised points, based on the wiring assumptions made in the upper and lower ring magnets, to obtain the radial and axial forces applied by the PMB onto the rotor.

To determine how much force the PMB was exerting on the rotor, it was assumed that the ring magnets were symmetric. The process of defining a relation between position and force was determined by the knowledge that the PMB was supporting the weight of the rotor. Therefore, by first assuming perfect alignment between the rotor and stator, the axial force was determined by calculating the force at different axial separations between them (see Fig. 3.6). This gave data points for the force which were interpolated using a

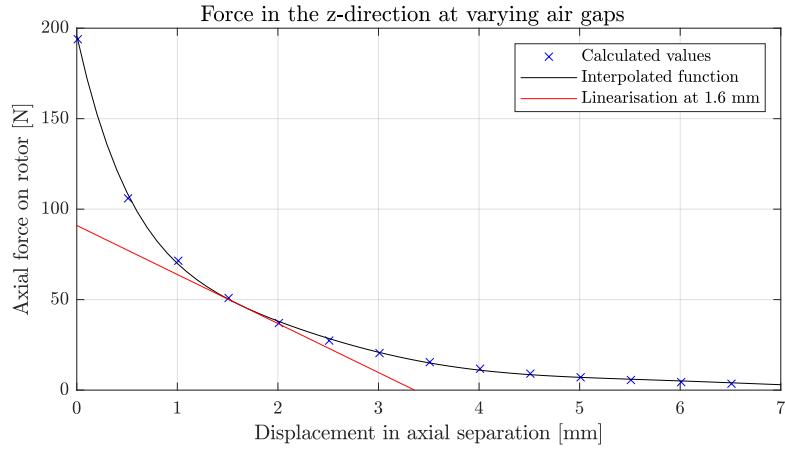


Figure 3.6: Analysis of changing the air gap between PMB rotor and stator and the effect on the axial force.

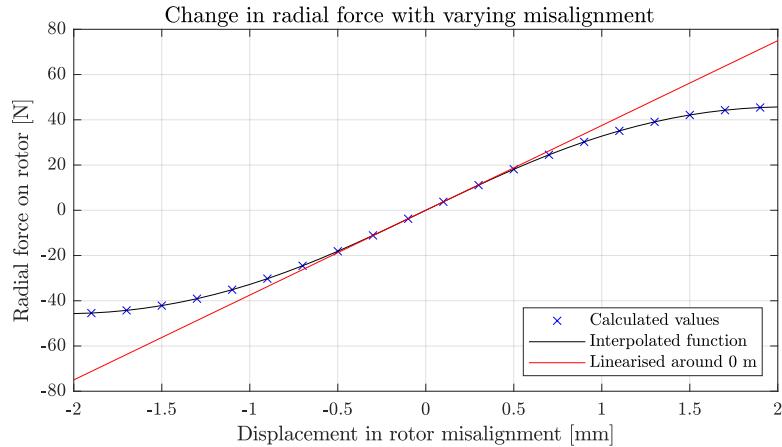


Figure 3.7: Analysis of changing alignment between PMB rotor and stator with respect to the change in radial force.

tenth-order polynomial and linearised at the force correlated to the weight of the rotor. Next, having assumed the vertical separation between the magnets, the force at varying radial misalignments was determined (see Fig. 3.7). The linearisation of the resulting tenth-order-polynomial interpolated data was then carried out around the point of perfect alignment. This analysis was carried out in MATLAB using the parameters outlined in Tab. A.1

The result of this analysis provided a linearised relation between the forces the PMB generated, based on changes in the rotor position given by:

$$F_{px} = k_{pr}x_p, \quad (3.83)$$

$$F_{py} = k_{pr}y_p, \quad (3.84)$$

$$F_{pz} = k_{pa}(z_p + l_p), \quad (3.85)$$

where the constants k_{pr} and k_{pa} characterise the slope determined by the linearised analyses. The position terms were based on the previously derived relations regarding the PMB plane from Eq. 3.20.

3.1.5 Butterworth filter

To minimise the noise from both the current and the position sensors, a digital filter was implemented on their raw output [20]. The filter that was implemented on every output signal was a 2nd-order Butterworth filter with a cutoff frequency at 4 kHz. The normalised Butterworth polynomial for this filter was given by:

$$B(\bar{s}) = \bar{s}^2 + \frac{2}{\sqrt{2}}\bar{s} + 1, \quad (3.86)$$

based on taking the position of complementary poles at $\pm 135^\circ$ [26, pg. 54].

The implemented filter was then described in continuous time as:

$$G(\bar{s}) = \frac{1}{B\left(\frac{\bar{s}}{2\pi \cdot 4000}\right)}, \quad (3.87)$$

$$= \frac{6.4\pi^2 \cdot 10^7}{\bar{s}^2 + 8000\pi\sqrt{2}\bar{s} + 6.4\pi^2 \cdot 10^7}, \quad (3.88)$$

which was converted to discrete time using the zero-order hold method, mimicking the test rig, based on the system sampling time of 50 µs:

$$G(\bar{z}) = \frac{0.4215\bar{z} + 0.229}{\bar{z}^2 - 0.5186\bar{z} + 0.1691}. \quad (3.89)$$

3.1.6 PID-PI controller

The existing control scheme implemented on the test rig was a cascade PID-PI control as shown in Fig. 3.8. Using this method, the desired positions Y_{ref} were compared to the filtered actual positions Y_f of the system. The error was then fed to PID controllers that determined the reference currents I_{ref} which were compared to the filtered actual currents I_f and used by PI controllers to produce the control inputs U . This occurred at every time step k , sampled at 20 kHz.

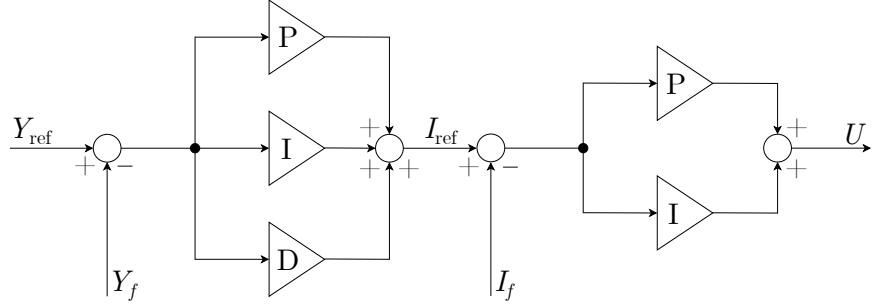


Figure 3.8: PID-PI cascade controller schematic as implemented in the test rig.

The PID controllers operated together as follows:

$$I_{\text{ref},k} = K_{p,\text{PID}} \cdot e_k + K_{i,\text{PID}} \sum_0^k \frac{e_k + e_{k-1}}{2} + K_{d,\text{PID}} (e_k - e_{k-1}) , \quad (3.90)$$

$$\text{where, } -0.6 \leq I_{\text{ref},k} \leq 0.6 , \quad (3.91)$$

$$e_k = Y_{\text{ref},k} - Y_{f,k} . \quad (3.92)$$

The gains $K_{p,\text{PID}}$, $K_{i,\text{PID}}$, $K_{d,\text{PID}}$ were used to manipulate the desired response of the controllers and to multiply the error, approximation of the integral of the error, and approximation of the derivative of the error, respectively.

The calculated reference currents I_{ref} then had to be generated in the AMBs. Therefore, the PI controllers were implemented as a method of ensuring a low error between the desired and actual currents, by means of applying the appropriate voltage. This was achieved as follows:

$$\text{PWM}_k = K_{p,\text{PI}} \cdot e_k + K_{i,\text{PI}} \sum_0^k \frac{e_k + e_{k-1}}{2} , \quad (3.93)$$

$$\text{where, } -950 \leq \text{PWM}_k \leq 950 , \quad (3.94)$$

$$U_k = \frac{15}{1000} \text{PWM}_k , \quad (3.95)$$

$$e_k = I_{\text{ref},k} - I_{f,k} , \quad (3.96)$$

where I_f is the filtered measured current and the gains $K_{p,\text{PI}}$ and $K_{i,\text{PI}}$ manipulate the desired response from the controllers. The implemented controllers calculated the required PWM duty cycles out of 1000, which was then communicated to the power supplies to generate the appropriate voltages U_k , which had a maximum of 15 V. Thus, by applying a factor to the calculated PWM result, the input voltages for the four pole pairs in the two AMBs were obtained.

3.2 Test rig emulation

To emulate the test rig, the derived mathematical equations from Sec. 3.1 were used to construct a simulation. In Sec. 3.2.1, the manner is outlined in which the differential equations were solved. Then, the implementation of the filter (Sec. 3.2.2) and controller (Sec. 3.2.3) in the simulation is explained.

3.2.1 Differential equation solver

The equations of motion in the mechanical system as shown in Eq. 3.31–3.35 with the descriptions for the forces in Eq. 3.52–3.55 and Eq. 3.83–3.85, as well as the electrical system as shown in Eq. 3.66–3.69 were used directly in MATLAB as functions. For the parameters used in these equations, refer to Tab. A.2.

These functions were solved by using the Runge-Kutta fourth-order (RK4) method. This method offered a fast solution of the differential equations with outputs provided strictly based on the set time step, which is not the case with faster-solving variable-time-step solvers. As a result, by implementing the time step that was set in the test rig, the manner in which the flywheel was controlled was emulated more easily in the simulation loop.

The RK4 method is a fixed-step differential equation solver that iteratively approximates the value of a state from its derivative at the next time step by taking a weighted average of four increments. This is solved for a given initial value problem:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0, \quad (3.97)$$

and works as follows:

$$y_{k+1} = y_k + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) t_s, \quad (3.98)$$

$$t_{k+1} = t_k + t_s, \quad (3.99)$$

where the subscript k indicates the value at the time step and t_s is the chosen sampling time. The four increments are calculated in the following way:

$$k_1 = f(t_k, y_k), \quad (3.100)$$

$$k_2 = f\left(t_k + \frac{t_s}{2}, y_k + t_s \frac{k_1}{2}\right), \quad (3.101)$$

$$k_3 = f\left(t_k + \frac{t_s}{2}, y_k + t_s \frac{k_2}{2}\right), \quad (3.102)$$

$$k_4 = f(t_k + t_s, y_k + t_s k_3), \quad (3.103)$$

giving an approximation of the system that is accurate based on how fast it evolves with time and the relation to this with the chosen sampling time [27, pg. 423].

The shown method was directly applied to the equations of motion of the first order (Eq. 3.66–3.69), as the RK4 method can only be applied to first-order differential equations directly. For the second-order differential equations of the mechanical system (Eq. 3.31–3.35), the equations had to be expressed in terms of two first-order differential equations as follows:

$$\frac{dy}{dt} = \dot{y}, \quad \frac{d\dot{y}}{dt} = \ddot{y}. \quad (3.104)$$

The RK4 method was then applied sequentially as for a first-order differential equation, but – as they were dependent on each other – simultaneously for both of these descriptions.

To implement this solver in MATLAB, a function had to be written separately for the second-order differential equations and for the first-order differential equations present in the equations of motion. First, however, the equations of motion, as they were obtained, were declared as functions with the system states declared as variables within them. When defining the states in the functions, it was important to order them such that the states being differentiated were always in the same index, as the function applying the RK4 method could then be generalised.

Next, the RK4 method was implemented in a function that could continuously be called in the simulation loop. When calling this function, it was important that the output of the RK4 approximation was used to redefine the state variables at the end of each iteration. This was because the RK4 method approximates the value of the states at the current time step, using the states from the previous time step.

After obtaining the states at every time step, the output of the simulated system could be defined. This was based on the position of the rotor in the sensor planes (Eq. 3.16 & 3.19) and the control currents in the AMBs.

3.2.2 Filter implementation

The filter in discrete time (Eq. 3.89) was most conveniently implemented in its state-space form. This gave the following description for any of the filtered states:

$$x_{f,k+1} = \begin{bmatrix} -0.0599 & -0.4514 \\ 0.4514 & 0.5785 \end{bmatrix} x_{f,k} + \begin{bmatrix} 0.4514 \\ 0.4215 \end{bmatrix} y_k, \quad (3.105)$$

$$y_{f,k} = [0 \ 1] x_{f,k}, \quad (3.106)$$

where $x_{f,k}$ describes the filter state at the given time step k , y_k is the read calibrated state from the position or current sensors, and $y_{f,k}$ is the filtered state.

In the simulation loop, this was implemented as a function that used the filtered states from the previous iteration and the calculated output from the solved differential equations. The output of the filter was then used, like in the test rig, as an input to the control law.

3.2.3 Controller implementation

The controllers, as implemented in the test rig, were explained in Sec. 3.1.6. Defining the gains for the controllers took place in the *RealTime_V1* program which converted the values to the gains for the *FPGA_V1* program that ultimately controlled the system in real time. The implemented gains gathered from the previous studies [20, 22] are displayed in Tab. 3.1.

Table 3.1: PID-PI controller gains as implemented in the FESS test rig

	Value	Unit
$K_{p,PID}$	0.032	A μm^{-1}
$K_{i,PID}$	$5 \cdot 10^{-7}$	A μm^{-1}
$K_{d,PID}$	0.1	A μm^{-1}
$K_{p,PI}$	10000	A $^{-1}$
$K_{i,PI}$	5	A $^{-1}$

The LabVIEW program defined the position in micrometers, therefore, as the simulation would execute using SI units, the output in position had to be scaled by a factor of 10^6 . This allowed the direct implementation of the controller gains, as they were defined in the experimental system, to apply the control law as shown in Sec. 3.1.6.

Chapter 4

Optimal Control Methods

4.1 LQG control method

Given the existing implementation of the PID-PI cascade controller, it was decided that the LQG would be formulated based on the schematic shown in Fig. 4.1. In this manner, the LQG would generate a reference current for the PI controller to follow, based on the difference between a reference position and a measured position.

To obtain the LQG controller, the model on which it was based was first arranged in state-space form (Sec. 4.1.1). Using this model, the LQR (Sec. 4.1.2) and the Kalman filter (Sec. 4.1.3) were designed separately. LQR and Kalman filter were then combined to formulate the LQG (Sec. 4.1.4) as it was then programmed in LabVIEW (Sec. 4.1.5).

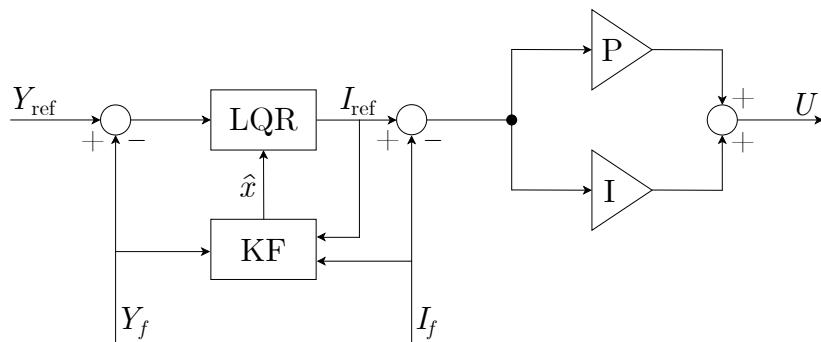


Figure 4.1: LQG control method made up by the LQR and Kalman filter, optimally calculating the reference current for the existing PI controller to follow.

4.1.1 State-space formulation

To apply a state-space based control, the equations of motion of the controlled system were linearised and brought into state-space form. This allowed the application of established routines to calculate the optimal gains required for the LQG.

Given the application of the LQG in the system (Fig. 4.1), the model it was based on had to consider the PI controller, the mechanical and the electrical dynamics of the flywheel, and the filtered output of the system. These components were first individually defined in state-space form and then combined into a single state-space formulation, where the input was the reference current I_{ref} and the output was the filtered outputs of the FESS Y_f and I_f . The following analysis describes how the individual state-space systems were identified and presents the resulting states that were then included in the state vector of the full state-space system.

The PI controller was based on Eq. 3.93–3.96 without the limitation on the PWM output. The controller took $I_{\text{ref},k}$ as input and outputted U_k to the electrical dynamics of the flywheel. Described for one of the four PI controllers, adapted from Eq. 3.93–3.96, the state-space formulation was based on the following equations:

$$\begin{aligned} u_{k+1} = & \frac{15}{1000} \left[i_{\text{ref},k} \left(K_{p,\text{PI}} + \frac{K_{i,\text{PI}}}{2} \right) - i_{f,k} \left(K_{p,\text{PI}} + \frac{K_{i,\text{PI}}}{2} \right) \right. \\ & \left. + e_{p,k} \frac{K_{i,\text{PI}}}{2} + e_{i,k} \right], \end{aligned} \quad (4.1)$$

$$e_{p,k+1} = i_{\text{ref},k} - i_{f,k}, \quad (4.2)$$

$$e_{i,k+1} = \frac{K_{i,\text{PI}}}{2} (i_{\text{ref},k} - i_{f,k} + e_{p,k}) + e_{i,k}. \quad (4.3)$$

Therefore, three states were required for each of the four PI controllers in the full system. As a result, the full system state vector included from the PI controllers:

$$\boldsymbol{x}_{\text{PI}} = [u_{1-4} \quad e_{p,1-4} \quad e_{i,1-4}]_k^\top, \quad (4.4)$$

where u_{1-4} represents the four individual voltage inputs to the electrical system, $e_{p,1-4}$ represents the four error signals required to be used in the next time step to calculate the integral term, and $e_{i,1-4}$ represents the four integral terms on which the integral gain acts on.

Next, the mechanical and electrical dynamics of the flywheel were brought into state-space form based on Eq. 3.31–3.35 and Eq. 3.66–3.69. This system took the generated voltages $u_{1-4,k}$ as input and outputted the position of the rotor sensed at the sensor planes ($x_{s1}, y_{s1}, x_{s2}, y_{s2}$) in micrometers and the control currents ($i_{a1x}, i_{a1y}, i_{a2x}, i_{a2y}$) in amperes. This added the following

states to the state vector of the full system:

$$\mathbf{x}_{\text{sys}} = [x \ \dot{x} \ y \ \dot{y} \ \phi \ \dot{\phi} \ \psi \ \dot{\psi} \ i_{a1x} \ i_{a1y} \ i_{a2x} \ i_{a2y}]^\top, \quad (4.5)$$

where the states associated with z were omitted, as they could not be controlled using the horizontal actuation of the AMBs.

The final state-space component belonged to the Butterworth filter being applied. This required two states for each of the eight filters as described in Sec. 3.2.2, and added the following states to the state vector of the full system:

$$\mathbf{x}_{\text{pos,fil}} = [x_{s1,f} \ \dot{x}_{s1,f} \ y_{s1,f} \ \dot{y}_{s1,f} \ x_{s2,f} \ \dot{x}_{s2,f} \ y_{s2,f} \ \dot{y}_{s2,f}]_k^\top, \quad (4.6)$$

$$\mathbf{x}_{\text{cur,fil}} = [i_{a1x,f} \ \dot{i}_{a1x,f} \ i_{a1y,f} \ \dot{i}_{a1y,f} \ i_{a2x,f} \ \dot{i}_{a2x,f} \ i_{a2y,f} \ \dot{i}_{a2y,f}]_k^\top, \quad (4.7)$$

where the filtering of each individual FESS output required the use of two states as the Butterworth filter was of second order.

With all three components described in state-space form, they were combined into a single state-space system. However, first, the system pertaining to the FESS was discretised using the zero-order hold method, based on the 50 µs sampling time of the LabVIEW program. The choice of the sampling time was validated based on the results obtained from the procedure outlined in Sec. 5.1. Then, the state-space systems were combined to describe the dynamics to be controlled by the LQG. The full system contained 40 states based on:

$$\mathbf{x}_k = [\mathbf{x}_{\text{PI}} \ \mathbf{x}_{\text{sys}} \ \mathbf{x}_{\text{pos,fil}} \ \mathbf{x}_{\text{cur,fil}}]_k^\top, \quad (4.8)$$

with the input vector:

$$\mathbf{u}_k = [i_{\text{ref},a1x} \ i_{\text{ref},a1y} \ i_{\text{ref},a2x} \ i_{\text{ref},a2y}]_k^\top, \quad (4.9)$$

to describe the system from the PI controller to the filtered output of the system in the form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \quad (4.10)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \quad (4.11)$$

where the filtered output was:

$$\mathbf{y}_k = [x_{f,s1} \ y_{f,s1} \ x_{f,s2} \ y_{f,s2} \ i_{f,a1x} \ i_{f,a1y} \ i_{f,a2x} \ i_{f,a2y}]^\top. \quad (4.12)$$

4.1.2 LQR design

In the design of classical controllers, the performance of the method is generally based on time and frequency domain measures as well as on relative

stability. With optimal control, as is the case with the LQR, the quality of the controller is based on a formulated performance index [14]. This is to calculate the optimal input:

$$u = -Kx \quad (4.13)$$

where K is the optimal gain used onto the full states of the system x .

For the FESS, integral action was implemented to the design of the LQR in order to eliminate any potential offsets in position. This required the inclusion of integral states \mathbf{x}_i in an augmented system:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_i \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{40 \times 4} \\ -t_s \mathbf{C} & \mathbf{I}_{4 \times 4} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_i \end{bmatrix}_k + \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{4 \times 4} \end{bmatrix} \mathbf{u}_k, \quad (4.14)$$

rewritten as:

$$\mathbf{x}_{a,k+1} = \mathbf{A}_a \mathbf{x}_{a,k} + \mathbf{B}_a \mathbf{u}_k, \quad (4.15)$$

where the matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} were based on the full state-space system obtained in Sec. 4.1.1, $\mathbf{0}$ was an appropriate 0 matrix, \mathbf{I} was an identity matrix, and t_s the 50 µs sampling time on which the controller was based.

For the augmented 44-state discrete-time system, the control gain was calculated using the performance index defined as:

$$J = \sum_{k=0}^{\infty} (\mathbf{x}_{a,k}^\top \mathbf{w}_1 \mathbf{x}_{a,k} + \mathbf{u}_k^\top \mathbf{w}_2 \mathbf{u}_k), \quad (4.16)$$

where \mathbf{w}_1 and \mathbf{w}_2 were the weighting matrices determining how much the deviations from 0 in the states and inputs, respectively, influenced the overall cost function. Additionally, the cost was calculated based on an infinite time horizon, therefore, the optimal control gain on the states was the same at every time step.

Although the augmented system described by Eq. 4.15 was found not to be fully controllable, which was verified by finding that the controllability matrix defined by:

$$\mathcal{C} = [\mathbf{B}_a \ \mathbf{A}_a \mathbf{B}_a \ \mathbf{A}_a^2 \mathbf{B}_a \ \dots \ \mathbf{A}_a^{44-1} \mathbf{B}_a], \quad (4.17)$$

had a rank lower than 44. However, the optimum value for Eq. 4.16 could still be found as the system was stabilisable [14, pg. 317]. This meant that even though the system was not fully controllable, a stabilising solution could be found within finite time.

The performance index was then minimised with the following control gain matrix calculated by:

$$\mathbf{K} = (\mathbf{B}_a^\top \mathbf{P} \mathbf{B}_a + \mathbf{w}_2)^{-1} (\mathbf{B}_a^\top \mathbf{P} \mathbf{A}_a), \quad (4.18)$$

where the infinite horizon solution \mathbf{P} was associated with the solution of the discrete-time Riccati equation:

$$\mathbf{A}_a^\top \mathbf{P} \mathbf{A}_a - \mathbf{P} - (\mathbf{A}_a^\top \mathbf{P} \mathbf{B}_a) (\mathbf{B}_a^\top \mathbf{P} \mathbf{B}_a + \mathbf{w}_2)^{-1} (\mathbf{B}_a^\top \mathbf{P} \mathbf{A}_a) + \mathbf{w}_1 = 0, \quad (4.19)$$

which was solved via the MATLAB function `dlqr`.

For clarification, the control gain matrix was separated into the control gain onto the system states and the integral action:

$$\mathbf{K}_{4 \times 44} = [\mathbf{K}_{\mathbf{x}, 4 \times 40} \quad -\mathbf{K}_{\mathbf{i}, 4 \times 4}], \quad (4.20)$$

where $\mathbf{K}_{\mathbf{x}}$ was the control gain onto the states of the full state-space system described in Sec. 4.1.1 and $\mathbf{K}_{\mathbf{i}}$ was the control gain onto the integral states introduced in the augmented system (Eq. 4.14). As a result, the optimal input to the system took on the form:

$$\mathbf{u}_k = \mathbf{K}_{\mathbf{i}} \mathbf{x}_{\mathbf{i}, k} - \mathbf{K}_{\mathbf{x}} \mathbf{x}_{\mathbf{x}, k}. \quad (4.21)$$

4.1.3 Kalman filter design

The Kalman filter is an optimal observer that is able to obtain an estimation of all observable states, based on the system input and measured output. As an LQR regulates a system based on the internal states, the information on them is required despite the fact that it is not always measured directly. Therefore, the Kalman filter was employed to provide this information.

In this case, the measured outputs were the position of the rotor in the horizontal plane at either end and the control current of the AMBs. Although the full system (Eq. 4.10 & 4.11) was not fully observable, given that the observability matrix:

$$\mathcal{O} = [\mathbf{C} \quad \mathbf{CA} \quad \mathbf{CA}^2 \quad \dots \quad \mathbf{CA}^{40-1}]^\top, \quad (4.22)$$

had a rank lower than 40 and the observability Gramian \mathbf{W}_o , calculated from:

$$\mathbf{A}^\top \mathbf{W}_o \mathbf{A} - \mathbf{W}_o + \mathbf{C}^\top \mathbf{C} = 0, \quad (4.23)$$

was not positive definite, it was found that a Kalman gain could still be identified, as a finite stabilising solution to the discrete-time algebraic Riccati solution could be found.

Therefore, Kalman filter could be calculated and was designed as stationary to simplify the implementation in the real-time computer. To do this, the Kalman observer gain was calculated for the following system:

$$\mathbf{x}_{k+1} = \mathbf{Ax}_k + \mathbf{Bu}_k + \mathbf{B}_v \mathbf{v}_1, \quad (4.24)$$

$$\mathbf{y}_k = \mathbf{Cx}_k + \mathbf{v}_2, \quad (4.25)$$

which was based on the full state-space system obtained in Sec. 4.1.1, where \mathbf{v}_1 and \mathbf{v}_2 represent the process and measurement noise, respectively. The process noise was modelled to represent the input noise, thus $\mathbf{B} = \mathbf{B}_v$, as the input affected all states related to the rotor dynamics and simplified the amount of tunable parameters.

Next the discrete algebraic Riccati equation was solved implicitly for \mathbf{Q} to calculate the optimal observer gain in the equation:

$$\mathbf{A}\mathbf{Q}\mathbf{A}^\top - \mathbf{Q} - (\mathbf{A}\mathbf{Q}\mathbf{C}^\top)(\mathbf{C}\mathbf{Q}\mathbf{C}^\top + \mathbf{v}_2)^{-1}(\mathbf{A}\mathbf{Q}\mathbf{C}^\top)^\top + \mathbf{B}\mathbf{v}_1\mathbf{B}^\top = 0, \quad (4.26)$$

using the MATLAB function `idare`. From this, the Kalman gain \mathbf{K}_f was derived as:

$$\mathbf{K}_f^\top = (\mathbf{C}\mathbf{Q}\mathbf{C}^\top + \mathbf{v}_2)^{-1} \mathbf{C}\mathbf{Q}\mathbf{A}. \quad (4.27)$$

Thus, the full-state observer for the system took on the form:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}_f(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k), \quad (4.28)$$

giving an optimal estimation of the internal states $\hat{\mathbf{x}}_k$ using a time-invariant observer.

4.1.4 Formulation

An LQR control method becomes an LQG control method when the states onto which the optimal control gain is applied are inferred using a Kalman filter observer. As a result, the implementation of the LQR control does not require the measurement of all internal states, rather having this information estimated via the system output.

Based on the LQR and Kalman filter designs from Sec. 4.1.2 & 4.1.3, respectively, by combining the designs, the implementation as an LQG can be seen in the block diagram shown in Fig. 4.2. In this scheme, it is clearly shown how the reference position Y_{ref} is compared to the filtered measured positions Y_f . This is then used to generate the integral states, which are minimised by the integral control gain \mathbf{K}_i . The filtered measured position and the current output Y_f & I_f , respectively, are used to generate the innovation which is multiplied by the Kalman observer gain \mathbf{K}_f . This is used to generate the estimated states that multiply with the state control gain \mathbf{K}_x . As a result, the input to the system based on the LQG structure was as follows:

$$\mathbf{u}_k = \mathbf{K}_i \mathbf{x}_{i,k} + \mathbf{K}_x \hat{\mathbf{x}}_k. \quad (4.29)$$

However, the block diagram in Fig. 4.2 could not be implemented directly due to the limited memory of the real-time computer. To apply the control

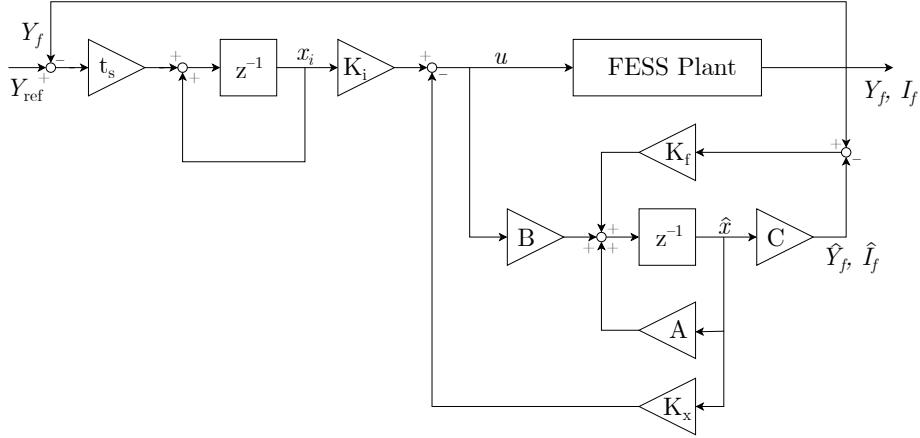


Figure 4.2: Block diagram of LQG controller showing integration of LQR and Kalman filter components.

scheme, the LQG had to be described in a more compact and simple manner. This was achieved by first describing the workings of the LQG in state-space form:

$$\begin{bmatrix} \mathbf{x}_i \\ \hat{\mathbf{x}} \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} - \mathbf{K}_f \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \hat{\mathbf{x}} \end{bmatrix}_{k+1} + \begin{bmatrix} t_s \mathbf{I} & -t_s \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,Y} & \mathbf{K}_{f,I} & \mathbf{B} \end{bmatrix} \begin{bmatrix} Y_{\text{ref}} \\ Y_f \\ I_f \\ \mathbf{u} \end{bmatrix}_k, \quad (4.30)$$

$$\mathbf{u}_k = [\mathbf{K}_i \quad -\mathbf{K}_x] \begin{bmatrix} \mathbf{x}_i \\ \hat{\mathbf{x}} \end{bmatrix}_{k+1}, \quad (4.31)$$

rewritten as:

$$\mathbf{x}_{c,k+1} = \mathbf{A}_c \mathbf{x}_{c,k} + \mathbf{B}_c \mathbf{u}_{c,k}, \quad (4.32)$$

$$\mathbf{y}_{c,k} = \mathbf{C}_c \mathbf{x}_{c,k}, \quad (4.33)$$

where the subscripts \mathbf{Y} and \mathbf{I} for \mathbf{K}_f indicate the selection of the correct columns in \mathbf{K}_f that observe for Y_f and I_f , respectively, corresponding to the arrangement in the output vector (Eq. 4.12).

Next, a balanced realisation was applied to this LQG representation. This reduced the difference between the maximum and minimum elements in the matrices to between 1 and -1, which benefitted the fixed-point implementation in LabVIEW. The transformation into a balanced realisation is based on Hankel singular values g which describe the energy of each state.

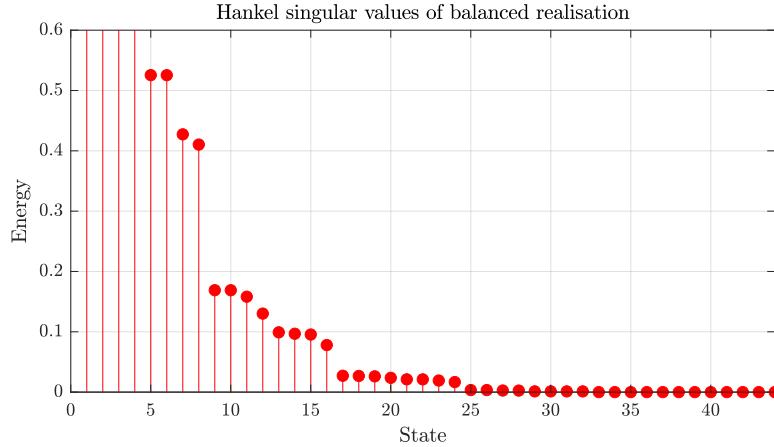


Figure 4.3: Energy of states in the balanced realisation. The first four states had .

This was used to obtain the similarity transform T as follows:

$$\text{diag}(g) = \bar{W}_c = \bar{W}_o, \quad (4.34)$$

$$\bar{W}_c = TW_cT^\top, \quad (4.35)$$

$$\bar{W}_o = T^{-\top}W_oT^{-1}, \quad (4.36)$$

where W_c and W_o are the controllability and observability Gramians of the augmented system. Thus, the balanced realisation was given by:

$$\bar{x}_{k+1} = T A_c T^{-1} \bar{x}_k + T B_c u_{c,k}, \quad (4.37)$$

$$y_k = C_c T^{-1} \bar{x}_k. \quad (4.38)$$

This routine was calculated using the MATLAB function `balreal`.

At this point, the realisation of the LQG still contained 44 states, while the real-time computer could handle 30 states at most. Therefore, the model had to be reduced to no more than 30 states. By using the MATLAB function `balreal`, the balanced realisation of the system had its states ordered from highest to lowest energy (Fig. 4.3). As the contribution of a state to the dynamics increased relative to its energy, the last 14 states were discarded from the LQG representation. This was applied using the MATLAB function `modred`, which ultimately gave a reduced-order balanced realisation of the LQG controller shown in Fig. 4.2, retaining as much of the characteristics as possible in a simplified form. The representation of this system will be referred to as:

$$\bar{x}_{r,k+1} = A_{r,c} \bar{x}_{r,k} + B_{r,c} u_{c,k}, \quad (4.39)$$

$$y_k = C_{r,c} \bar{x}_{r,k}, \quad (4.40)$$

Note how input and output remained the same as in Eq. 4.32 & 4.33 through the various transformations, as the aim was to preserve the relation throughout the simplification of the LQG formulation.

4.1.5 LabVIEW implementation

With the simplified form of the LQG, the matrices $A_{r,c}$, $B_{r,c}$, and $C_{r,c}$ had to be implemented in the existing LabVIEW program such that a reference current was calculated from the measured filtered output of the test rig, to be followed by the existing PI controller.

First, the calculated matrices were communicated from the *RealTime_V1* interface to the *FPGA_V1* program (Fig. 4.4). This required the elements to be arranged in a delimited .txt file which was uploaded onto /c on the real-time computer (refer to Sec. 5.1 to see how files were accessed on the cRIO computer). The elements of the files were then arranged in a single column vector in LabVIEW, based on the known number of elements in the matrices, and converted into a signed fixed-point representation with a 32-bit word length and 2-bit integer word length. This represented the numbers within the range from -2 to 2 with a precision of $9.313 \cdot 10^{-10}$.

Next, the transferred matrices were used in the *FPGA_V1* program to calculate the control input as shown in Fig. 4.5. In the program, a while loop was used that executed based on a sampling time of $50\text{ }\mu\text{s}$. Then, by utilising the *shift register* function, the state vector $\bar{x}_{r,k}$ was iteratively calculated based on its value in the previous iteration. The input to this structure was the reference for all four measurable positions, the filtered and measured position and the control currents of the system, as well as the calculated input. The matrix times vector multiplication was executed in a subVI, which calculated the output vector based on the known size of the matrix and the length of the output vector. Finally, the *reset states* button was implemented such that states of the LQG could be cleared through the front panel of the *RealTime_V1* program.

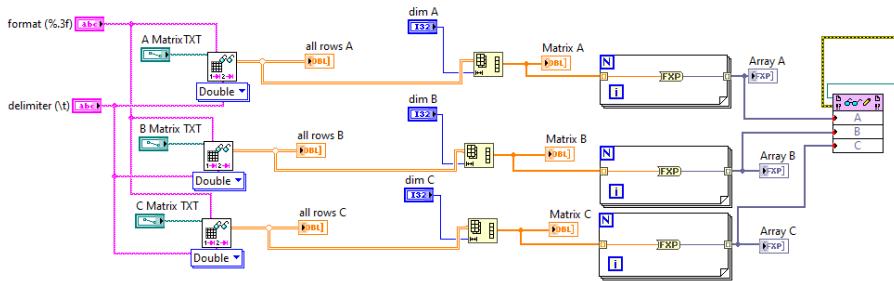


Figure 4.4: LabVIEW block diagram communicating matrices from *RealTime_V1* to *FPGA_V1*.

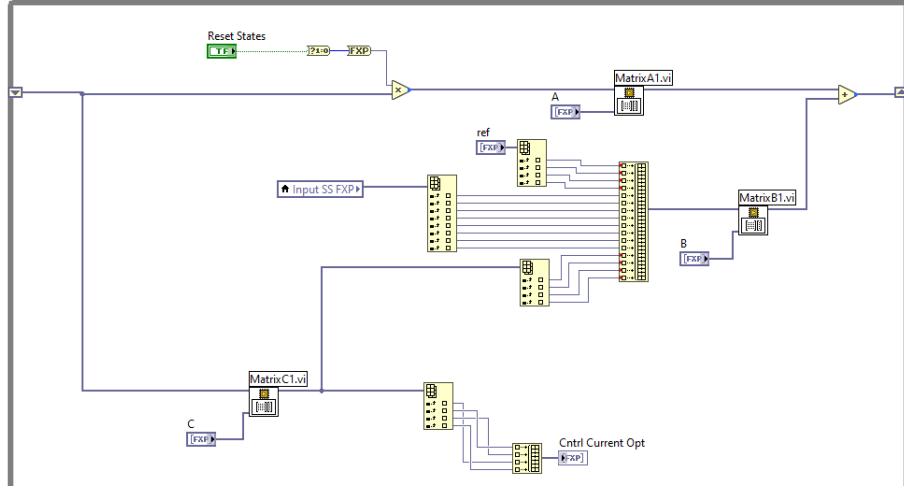


Figure 4.5: LabVIEW block diagram of the LQG structure used to calculate the control input.

In the *FPGA_V1* program, it was very important to adequately define the precision of each potential calculation and the length of all vectors. For most calculations, the output was chosen to have a fixed-point representation with a 32-bit word length and a 15-bit integer word length. The only differing calculation output was from the output of $\mathbf{C}_{r,c}\bar{x}_r$, which used a representation with a 20-bit word length and a 2-bit integer word length, as the control currents could not exceed $\pm 0.6\text{A}$. In either of these cases, the bit usage was not optimal. However, through MATLAB testing it was found to guarantee a comparable output to the default floating-point double representation used in simulation and it was possible to build the implementation in LabVIEW onto the cRIO.

The final aspect of the implementation was the controller selector in Fig. 4.6. It allowed the manipulation of the contribution of the PID and LQG controllers to the PI controller input, based on a value that was set in the front panel of the *RealTime_V1* program.

To engage the LQG, the rotor was first stabilised using the PID as described in Sec. 2.2.5 until the *activate bearings* button was pressed for the first time. At this point, the terminals required to activate the LQG could be used on the right side of the front panel in *RealTime_V1*, in the *control* panel (Fig. 4.7). As the LQG was dependent on its point of operation, the sensors had to be calibrated using the *offset position* control to guarantee that the defined centred position in the program coincided with the physical position of the rotor. The calibration was carried out by assuming, based on the physics described in Sec. 3.1.3, that the rotor was centred when the control current of the PID indicated in *current PID* was equal to 0. After calibrating, the *opt On* button was pressed, which allowed the states to build

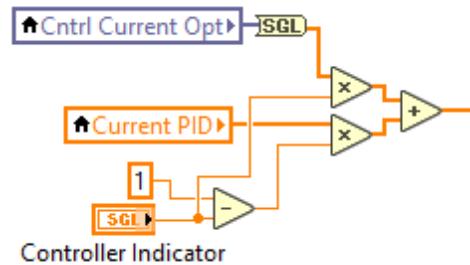


Figure 4.6: LabVIEW block diagram of the controller selector between PID and LQG.

up in the LQG, such that the *cntrl indicator* could be increased steadily from 0 to 1. When the value reached 1, the LQG was fully active, and, if desired, the reference position given by *ref* could be changed. Importantly, when the program was stopped, the *opt on* button had to be deactivated and the *cntrl indicator* set back to 0. The outlined process had to be repeated in order to stabilise the rotor via the LQG again.

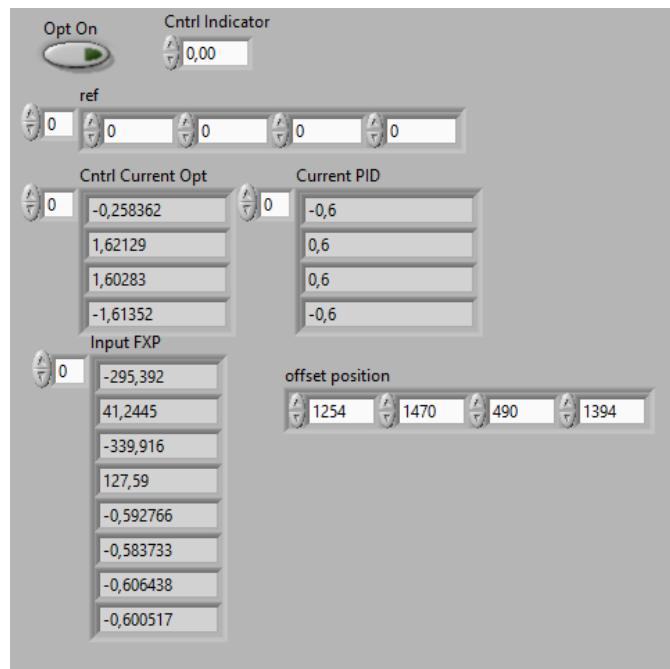


Figure 4.7: LabVIEW front panel controls to engage the LQG.

4.2 MPC method

An MPC is based on the optimisation of the control of a plant within a finite time horizon [15]. This optimisation is calculated at every time step and allows for constraints on the input and output of the system to be applied and considered in the performance index [16].

For this project, it was decided to investigate the application of an input-constrained MPC based on the methods shown in [15]. The formulation of this controller is explained in Sec. 4.2.1.

Given the limited experience with LabVIEW and the time constraints of the project, the implementation of the presented algorithm on limited memory for a high sampling rate system (20 kHz) was deemed overly complex. Therefore, it was decided to investigate the MPC only in simulation, using the derived models, and to refrain from implementing it onto the test rig like the LQG.

4.2.1 Formulation

The MPC was applied directly on the FESS dynamics given by Eq. 3.31–3.35 and Eq. 3.66–3.69, described in state-space form as:

$$\dot{\mathbf{x}}_{\text{sys}} = \mathbf{A}_{\text{sys}} \mathbf{x}_{\text{sys}} + \mathbf{B}_{\text{sys}} \mathbf{u}_{\text{sys}}, \quad (4.41)$$

$$\mathbf{y}_{\text{sys}} = \mathbf{C}_{\text{sys}} \mathbf{x}_{\text{sys}}. \quad (4.42)$$

This system was discretised using the sampling time of 50 µs, given the validation of this value from Sec. 5.1, and the zero-order hold method. It was described by:

$$\mathbf{x}_{\text{sys},k+1} = \mathbf{F} \mathbf{x}_{\text{sys},k} + \mathbf{G} \mathbf{u}_{\text{sys},k}, \quad (4.43)$$

$$\mathbf{y}_{\text{sys},k} = \mathbf{E} \mathbf{x}_{\text{sys},k}. \quad (4.44)$$

Given the requirement of using a Kalman filter and the ability to set hard input constraints, it was deemed unnecessary to apply the MPC in a similar manner to the LQG, with the existing additional PI controller and filter. The inclusion of these additional components in the system would furthermore have required a more complex model, making the computation involved in the control scheme more demanding.

An MPC uses the given state and the input into the system to calculate the optimal input sequence, based on a prediction of the evolution of the output. This is obtained using information on the zero-input and zero-state solutions for the discretised system described by:

$$\mathbf{Z} = \Phi \hat{\mathbf{x}}_{\text{sys},k} + \Gamma \mathcal{U}, \quad (4.45)$$

where \mathbf{Z} is the predicted output and:

$$\Phi = \begin{bmatrix} \mathbf{E}\mathbf{F} \\ \mathbf{E}\mathbf{F}^2 \\ \vdots \\ \mathbf{E}\mathbf{F}^\kappa \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \mathbf{E}\mathbf{G} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{E}\mathbf{F}\mathbf{G} & \mathbf{E}\mathbf{G} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{E}\mathbf{F}^2\mathbf{G} & \mathbf{E}\mathbf{F}\mathbf{G} & \mathbf{E}\mathbf{G} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{E}\mathbf{F}^\kappa\mathbf{G} & \mathbf{E}\mathbf{F}^{\kappa-1}\mathbf{G} & \mathbf{E}\mathbf{F}^{\kappa-2}\mathbf{G} & \dots & \mathbf{E}\mathbf{G} \end{bmatrix}, \quad (4.46)$$

are the matrices used to premultiply the estimated states in order to obtain the zero-input solution and the calculated input sequence for the zero-state solution based on the Markov parameters of the system, respectively, for the given depth κ .

The objective function solved by the MPC is based on a quadratic objective function that is obtained from a weighted-least-squares problem. In discrete form, the objective function is described by:

$$\min_{\mathcal{U}} \gamma = \frac{1}{2} \sum_{k=0}^{\kappa} \|\hat{z}_k - \bar{r}_k\|_{Q_z}^2 + \frac{1}{2} \sum_{k=0}^{\kappa-1} \|\Delta u_k\|_S^2, \quad (4.47)$$

where the function γ is minimised for the depth κ , based on the input sequence \mathcal{U} used in terms of the difference in the output \hat{z}_k and the reference \bar{r}_k , as well as the input effort Δu_k . The contributions of these metrics were weighted in their cost by the weighting matrices Q_z and S , and for the applied input-constrained MPC, the following additional condition was imposed:

$$\begin{bmatrix} u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix} \leq \begin{bmatrix} u_0 \\ \vdots \\ u_{\kappa-1} \end{bmatrix} \leq \begin{bmatrix} u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix}, \quad (4.48)$$

to define the minimum and maximum boundaries for the input sequence.

It followed, then, that the objective function could be described as a quadratic objective function [15]:

$$\min_{\mathcal{U}} \gamma = \frac{1}{2} \mathcal{U}^\top H \mathcal{U} + f^\top \mathcal{U}, \text{ where } \mathcal{U}_{\min} \leq \mathcal{U} \leq \mathcal{U}_{\max}, \quad (4.49)$$

where:

$$H = \Gamma^\top Q_z \Gamma + H_s, \quad (4.50)$$

$$f = (\Gamma^\top Q_z \Phi) \hat{x}_k + (-\Gamma^\top Q_z) \bar{r}_k + M_u u_{k-1}. \quad (4.51)$$

The additional matrices describe the weightings on the output and input Q_z and H_s , respectively, and the input weighting on the input M_u from the

previous step. They are defined as:

$$\mathbf{Q}_z = \mathbf{I}_{n \times n} \otimes Q_z, \quad (4.52)$$

$$\mathbf{H}_s = \begin{bmatrix} 2S & -S & \mathbf{0} & \dots & \mathbf{0} \\ -S & 2S & -S & \ddots & \vdots \\ \mathbf{0} & -S & 2S & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & -S \\ \mathbf{0} & \dots & \mathbf{0} & -S & S \end{bmatrix}, \quad (4.53)$$

$$\mathbf{M}_u = -[S \ \mathbf{0} \ \dots \ \mathbf{0}]^\top, \quad (4.54)$$

with their sizes dependent on the depth of the horizon κ .

4.2.2 MATLAB implementation

The MPC was implemented in simulation by first precomputing all the matrices outlined in Sec. 4.2.1 which were required for the quadratic objective function, based on a chosen depth. Additionally, the matrices for calculating the stationary Kalman gain were also defined such that the states could be estimated. The dynamics of the rotor were then represented in a simulation loop as described in Sec. 3.2.1, and the sensor outputs were used to infer the states with the Kalman filter. Using this, the MATLAB function `quadprog` was applied based on Eq. 4.50 & 4.51 and the input constraint (Eq. 4.48) to calculate the optimal input sequence for the given depth at every step. Thus, for the next step in the simulation, only the first instance in the input sequence \mathcal{U} was applied to the system.

As the MPC was only applied in simulation, sensor noise (see Tab. 5.1) and input noise (based on the power supply data sheet [28]) were added to the simulated output to emulate the real system.

Chapter 5

Test Procedures

5.1 PID model validation

The mathematical model obtained in Sec. 3.1 was validated based on its ability to reproduce the experimental step response, as the spinning of the rotor was not effectively modelled in simulation. To induce a step in the test rig, a constant offset was applied to the reference current that was fed to the PI controller. The result of this was an abrupt movement of the rotor in this direction, and the compensation and correction of this action by the applied control law.

5.1.1 Gathering experimental data

The experimental data was obtained by following the steps mentioned in Sec. 2.2.5 until the *activate bearings* button was pressed in the front panel to stabilise the stationary rotor. From this point on, the step size was defined as 0.1 A in one of the *stepsize* boxes (refer to Fig. 2.9) while the others remained at 0 A. This defined the magnitude of the offset being applied.

Afterwards, the *step input* button was selected in *radio buttons*, which meant that a step would be applied when the *log data* button was pressed. Then, the *step delay [ms]* and the *step length [ms]* controls were defined as 100 ms and 3000 ms, respectively, which set the desired characteristics of the offset being applied. Lastly, in the *data acquisition* panel (top left in Fig. 2.9), the *logging time [ms]* was set to 3000 ms.

Now that the test parameters were set, the *file path* (bottom left in Fig. 2.9) was given a unique title every time before pressing the *log data* button. The *log data* button was pressed every time the *recording* indicator to the right showed 0. The naming and logging was done 10 times for each step and, after the tenth recording, the *activate bearings* button was pressed to deactivate the AMBs. Subsequently, the *stop* button was pressed.

The recording files were accessed through the *FileZilla* program, an FTP client able to communicate with the cRIO (Fig. 5.1). As *host*

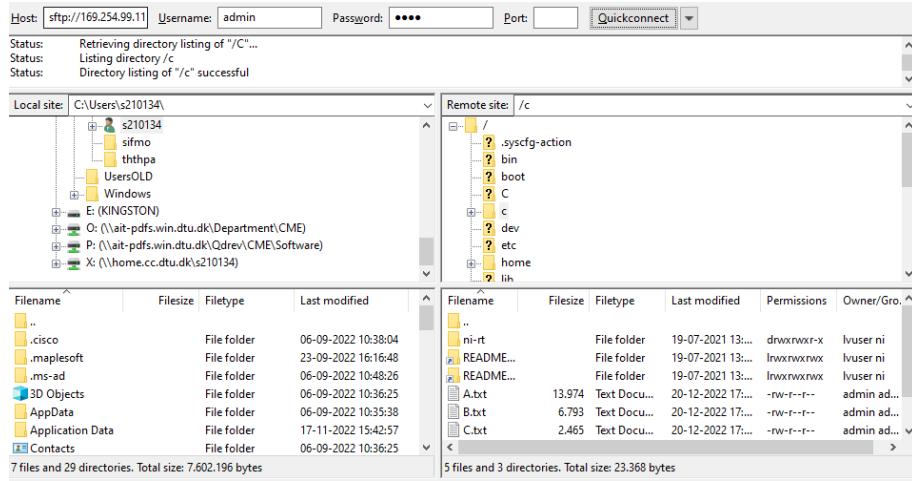


Figure 5.1: Program for the file transfer between cRIO and local computer.

`sftp://169.254.99.113` was inserted, with *username* `admin` and *password* `1234`. The *quickconnect* button was pressed, and the folder `/c`, which included all the recorded TDMS files, was located on the *remote site*. When the files did not appear directly, which was the case from time to time, the file list had to be refreshed. The selected files were then dragged to the local computer and used for later analysis.

Without shutting down the whole system, the rotor was again stabilised by following the instructions from when the LabVIEW program was started to when *activate bearings* was pressed for the first time. From this point, the step size was applied in a different direction, and 10 recordings were made for each of the four directions.

5.1.2 Experimental data processing

By obtaining 10 recordings for each step, the data was combined to generate a mean response for each of the four different steps. However, each recording was subjected to the inherent lag of the system which had to be taken into account.

The recorded TDMS files were read into MATLAB and the contents of each file was ordered based on the manner it was recorded in the LabVIEW program. The occurrence of the step size was recorded, and all of the data was truncated from the start of the recording such that the step occurred at exactly 100 ms. Afterwards, the mean of each of the 10 recordings that belonged to all four different steps was taken.

In the recordings, the current for each individual pole in the test rig was made (eight in total) and included the measured and the reference current. However, as the current was in pairs based on opposing poles, the difference

between the paired currents from the bias were the same but in opposite directions. Therefore, only the measured current that changed in the same direction as the modelled control current was used and adjusted such that in the stationary period before 100 ms, the average current was 0 A.

5.1.3 Test emulation

The experimental test was emulated using the simulation setup where the control law was based on the PID-PI cascade control. To simulate the test, the reference current calculated by the PID was altered by the same 0.1 A offset as in the experiment. The offset was applied at the step that coincided with 100 ms and a simulation length that corresponded to 3 s was conducted.

From the simulation, the position and current data was recorded for comparison.

5.1.4 Comparison

To compare the step-response data, three different metrics were used. All these metrics were based on the primary step response, which was based on the change in position over time in the direction that was excited by the step.

The first metric was the maximum deviation. This was found by taking the maximum value of the entire response. Next, the rise time, that is the time window where the rising position, before reaching the maximum deviation, was between 10 % and 90 % of the maximum deviation, was compared. Finally, the settling time, that is the time taken from the initiation of the step at 100 ms to the time after the maximum deviation where the position remained below 5 μm , was compared.

The differences in these characteristics were quantified as a percentage δ , as follows:

$$\frac{\varepsilon_{\text{sim}} - \varepsilon_{\text{exp}}}{\varepsilon_{\text{exp}}} \times 100\% = \delta\%, \quad (5.1)$$

where ε represents the identified value for the three metrics compared between the simulated (sim) and the experimental (exp) data.

5.2 Kalman filter tuning

The Kalman filter was tuned individually based on its ability to produce an estimated output that followed the measured output, using the design discussed in Sec. 4.1.3. Thus, the innovation was compared with the rotor being controlled by the PID in a separate program, given the memory limitations. The filter was still valid when it was implemented with the LQG based on the separation theorem [24, pg. 466]. This was done by first

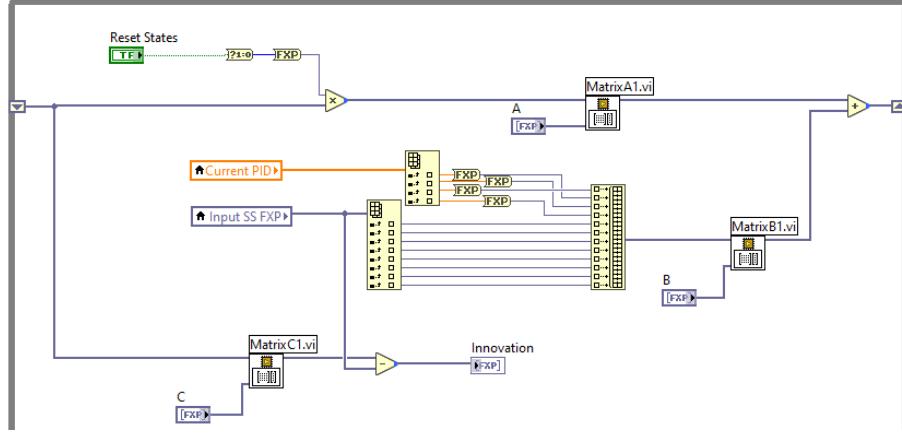


Figure 5.2: LabVIEW block diagram of the Kalman filter estimating for the PID-controlled rotor.

creating a LabVIEW program that used the Kalman filter to estimate the output of the PID controlled rotor (Sec. 5.2.1). Then, the measured output noise present in the system was identified (Sec. 5.2.2). Finally, the process and measurement noise components from Eq. 4.24 & 4.25, were tuned to obtain the best performance (Sec. 5.2.3).

5.2.1 LabVIEW implementation

To implement the Kalman filter in the PID-controlled system, the same principles from Sec. 4.1.4 and 4.1.5 were applied. However, the augmented system was based on the following system augmentation:

$$\hat{x}_{a,k+1} = (\mathbf{A} - \mathbf{K}_f \mathbf{C}) \hat{x}_{a,k} + [\mathbf{B} \quad \mathbf{K}_{f,Y} \quad \mathbf{K}_{f,I}] \begin{bmatrix} \mathbf{u}_{\text{PID}} \\ Y_f \\ I_f \end{bmatrix}_k , \quad (5.2)$$

$$\hat{e}_k = \mathbf{C} \hat{x}_{a,k} + [\mathbf{0}_{8 \times 4} \quad -\mathbf{I}_{8 \times 8}] \begin{bmatrix} \mathbf{u}_{\text{PID}} \\ Y_f \\ I_f \end{bmatrix}_k , \quad (5.3)$$

where the model the filter was based on assumed a stationary rotor, \mathbf{u}_{PID} was the calculated output from the PID and \hat{e}_k the innovation. Similar to the presented LQG in Sec. 4.1.4, a balanced realisation of this system was made and subsequently reduced to 30 states.

Next, the program was implemented in a copy of the LabVIEW project, where the LQG block diagram structure shown in Fig. 4.4 was replaced with Fig. 5.2. This allowed the innovation to be tracked with the stabilisation of the rotor in conjunction with the measured output of the system. Additionally, the recorded signals in the *FPGA_V1* program were altered such that the innovation was being recorded instead of the PWM signals.

The operation of this program was the same as for the LQG described in Sec. 4.1.5. However, instead of engaging the LQG, the estimation of the states ultimately leading to the innovation was calculated.

5.2.2 Sensor noise identification

As a starting point for the Kalman filter tuning, the measurement output noise had to be identified. This was done by reading the sensor data for 1 min and identifying the respective standard deviations. For the noise from the position sensors, the standard deviation was taken from when the AMBs were not active. For the current sensors, the standard deviation was taken from when the AMBs were active, as there would be current flowing through the wiring then. The results are summarised in Tab. 5.1

Table 5.1: Overview of the standard deviations calculated in the sensor tests.

Sensor	σ	Unit
x_{s1}	0.8499	μm
y_{s1}	0.8484	μm
x_{s2}	2.4438	μm
y_{s2}	0.9763	μm
i_{a1x}	0.0142	A
i_{a1y}	0.0168	A
i_{a2x}	0.0182	A
i_{a2y}	0.0163	A

As was found in [21], the sensor measuring the x -position in the lower part of the rotor was subject to far more noise than the other position sensors. As the sensor data seemed to indicate some periodicity in the position, a fast Fourier transform (FFT) was applied to the data to investigate for any frequency contributions to the signal (Fig. 5.3). It was found that in frequencies below 2000 Hz, the position sensors all had noticeable frequency contributions to the signal. The current sensors on the other hand did not display this behaviour.

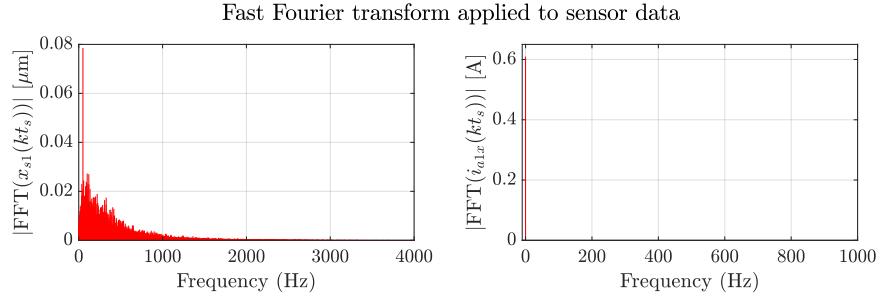


Figure 5.3: FFT on data from position sensor for x_{s1} and current sensor for i_{a1x} . The same characteristics were found in the other position and current sensors.

Based on the analysis, as an initial estimate of the covariance of the measurement noise, the following was used:

$$\mathbf{v}_2 = \text{diag}([0.9 \ 0.9 \ 2.5 \ 0.9 \ 0.015 \ 0.015 \ 0.015 \ 0.015]) . \quad (5.4)$$

5.2.3 Tuning procedure

To tune the Kalman filter, the initial measurement noise covariance from Eq. 5.4 was used and the process noise covariance \mathbf{v}_1 was set to the identity matrix. Then, data was gathered from when the rotor was stationary and rotating at 2000 rpm. The closeness of the estimated output was compared with the measured output.

The tuning of the covariances was carried out by trial and error, where the matrices with different tunings were calculated in MATLAB and then transferred onto the cRIO (refer to Sec. 5.1.1). First, only the process noise covariance was changed, but a significant lag was observed between the measured and estimated outputs. However, this was reduced when the measurement noise covariance was increased regarding the current. The best performing covariances from this process were:

$$\mathbf{v}_1 = 10 \cdot \mathbf{I}_{4 \times 4} , \quad (5.5)$$

$$\mathbf{v}_2 = \text{diag}([0.9 \ 0.9 \ 2.5 \ 0.9 \ 0.1 \ 0.1 \ 0.1 \ 0.1]) . \quad (5.6)$$

5.3 Kalman filter performance

The performance of the Kalman filter was evaluated using the identified weights. As the model on which the filter was based could be changed by incorporating the spin velocity in Eq. 3.34 & 3.35, the difference in the ability to estimate the system was evaluated at different speeds.

First, the Kalman filter matrices as described in Sec. 5.2.1 were identified for the speeds 0, 1000, 2000, and 15 000 rpm using the same weights as

shown in Sec. 5.2.3. This provided a range of models for the speed range available and an additional model that was outside of this range to observe a potentially very different behaviour.

For each of the identified Kalman filters, the matrices were transferred to the system as described in Sec. 4.1.5 using the same operation guidelines. This engaged the Kalman filter, but as the implementation only estimated the system, the rotor was being controlled by the PID.

Next, for each filter, data was recorded for a 1 min duration and saved as outlined in Sec. 5.1.1 using the VFD to manipulate the speed of the rotation as described in Sec. 2.2.5. The logging was initiated with *no input* selected in the front panel (Fig. 2.9) for the speeds 0, 500, 1000, 1500, and 2000 rpm. This was to evaluate each of the filters within the speed range available.

The recorded innovation of the state estimation was then used to compare the different filters against each other at the different speeds. First, the measured output was compared against the estimated output in a time series, which allowed a visual comparison of the two signals.

Next, the innovation signal was used to quantify the performance of the filters against each other. As a lower standard deviation in this signal corresponded to a better estimation, the estimation quality in position and current was evaluated as follows:

$$\sigma_{\text{pos}} = \sigma_{xs1} + \sigma_{ys1} + \sigma_{xs2} + \sigma_{ys2}, \quad (5.7)$$

$$\sigma_{\text{cur}} = \sigma_{i,a1x} + \sigma_{i,a1y} + \sigma_{i,a2x} + \sigma_{i,a2y}, \quad (5.8)$$

which was carried out for the innovation signal corresponding to the measurement indicated in the subscript. This was calculated for each filter at all recorded speeds.

Finally, by applying an autocorrelation on the innovation signal of one of the positions and currents, the signal was tested on whether only white noise was present or not. This was tested for the stationary case for the filter set for 0 rotation using the MATLAB function `autocorr`, as spinning the rotor would introduce periodicity in the signal which would have to be accounted for. If white noise is identified in the autocorrelation function, the model used for the Kalman filter is such that it incorporated the dynamics of the real system well. Otherwise, the signal indicates that there were unmodelled dynamics present, which the Kalman filter could not use to estimate the states based on the measured output.

5.4 LQG tuning

Having verified that the Kalman filter was operating at an acceptable level, the LQR portion of the LQG was tuned, knowing that the state estimation was working. This was done by first creating a base tuning using a MATLAB

simulation of the system (Sec. 5.4.1) and then fine-tuning the weights to a desired performance (Sec. 5.4.2).

5.4.1 Obtaining a base controller

For the LQR weights, it was decided that weightings would be defined for all states and inputs. To do this, the emulated PID-controlled rotor was leveraged as it was found to be in acceptable accordance with the experimental system (Sec. 6.1). From the previously gathered step-response tests (Sec. 5.1.3), additional data on all the states and inputs relevant to the LQR were also recorded.

From the recorded signals relevant to the LQR, the maximum values were identified to define the initial weights as:

$$[\mathbf{w}_1]_i = \frac{1}{\max([\mathbf{x}_{a,k}]_i)}, \quad (5.9)$$

$$[\mathbf{w}_2]_j = \frac{1}{\max([\mathbf{u}_k]_j)}, \quad (5.10)$$

where the state or input weighting was based on its corresponding maximum value in the recorded simulation [14, pg. 307]. The states pertaining to the integral action were initially set to 1.

The performance was then verified in simulation by replacing the control law in the PID-emulated system with the LQG structure from Sec. 4.1.1–4.1.4. The simulation was used verify that the base tuning was able to stabilise the system similarly to the PID in the procedure outlined in Sec. 5.1.3.

5.4.2 Tuning procedure

The base controller defined by the initial weighting matrices was used to generate the matrices in MATLAB for the LabVIEW implementation of the LQG (Sec. 4.1.5). Then, using the operation guidelines outlined in Sec. 4.1.5 to engage the controller, it was tested visually for its ability to compensate for a step and for how centred the rotor was while spinning.

The tuning of the weights was carried out by changing the weights that defined the LQR, building the matrices, formulating the LQG with the previously identified Kalman filter, and running a simulation. When a good performance was found in the step response, the matrices were ported to the test rig. From there, the response of the rotor spinning was observed, as this was not emulated sufficiently well in simulation. Through trial and error, it was found that the greatest influence on the performance lay in the specification of the input weighting \mathbf{w}_2 .

When the rotor was spinning at 1000 rpm, the upper and lower part of the rotor would whirl around the centre. This was most likely a result of

machining and material inaccuracies contributing to an unbalanced mass in the rotor. It was found that when all input weightings were equal, the lower part of the rotor would whirl with a greater distance from the centre. However, when the weighting was increased for the lower AMB inputs, increasing the constraint on this actuation, the whirl would reduce to being inside the upper rotor's whirl. Overall, the whirl of the entire rotor was then reduced and the rotor was kept closer to the centre when rotating.

The input weighting was then further investigated through trial and error in order to find a setting in which the rotor was controlled well in the step and spin response. The weights implemented for the definition of the LQR are outlined in Tab. A.3.

5.5 LQG emulation comparison

As the LQG was found to be working experimentally, the simulation of the LQG-controlled system was validated against experimental data to show that the tuning of the weights in simulation translated to the experimental response of the system. This was only verified for the step-response tests at 0 rpm, with the model on which the LQG was based set at 0 rpm, as the uncertainties leading to the experimental response of the spinning rotor were not emulated effectively in simulation.

The process for this test was the same as in Sec. 5.1, though during the experiments, the LQG was engaged as instructed in Sec. 4.1.5. The manner in which the experimental data was then processed was the same, but the emulation occurred using an LQG-controlled model in simulation instead. The comparison was then carried out using the same metrics (Eq. 5.1).

5.6 Step response comparison

Having collected step-response data on both the PID and LQG-controlled flywheel experimentally (from Sec. 5.1 & 5.5), the gathered mean responses of the step tests were compared.

The comparison was made based on the already gathered information on the maximum deviation, rise time, and settling time. Additionally, the total position and current error was compared. For the different controllers, this was calculated by summing the absolute difference of all four positions or currents, at every time step between when the excited direction increased above 10% from its maximum deviation, to when it remained below 5 μm , where it was considered settled. The percentage difference was then obtained as follows:

$$\frac{\varepsilon_{\text{LQG}} - \varepsilon_{\text{PID}}}{\varepsilon_{\text{PID}}} \times 100\% = \delta\%, \quad (5.11)$$

where ε represents the value of the characteristic determined from the data and δ the corresponding percentage change going from the PID to the LQG-controlled system.

5.7 Spin response comparison

The performance of the PID and LQG controllers was then compared with the rotor spinning. To do this, either PID or LQG was engaged. For the LQG controller, different controllers were trialled based on the incorporation of the gyroscopic effects. The difference between these models was the speed at which the rotor was assumed to be spinning, set in Eq. 3.34 & 3.35 with $\dot{\theta}$, which altered the state-space systems on which the LQG was then formulated (Sec. 4.1.4). This allowed to investigate whether changing this parameter would lead to a better or worse performance in the control.

The different LQG controllers were based on 0, 1000, 2000, and 15 000 rpm, resulting in a range based on the speed range of the motor and an additional controller set for a much higher speed. The five controllers were then engaged using the operation guidelines outlined in Sec. 2.2.5 and additionally Sec. 4.1.5 for the LQG controllers. Then, as outlined in Sec. 2.2.5, the VFD was used to set the speed of the rotor. For each controller, data was recorded and saved as outlined in Sec. 5.1.1, but for a logging time of 1 min with *no input* selected in the front panel (Fig. 2.9). The 1 min recordings were made for the speeds 0, 500, 1000, 1500, and 2000 rpm after the rotor was stabilised at that speed.

The ability to centre the rotor was first visually compared by plotting the x and y positions of the sensed top and bottom of the rotor for a given time period. This allowed the comparison of the amount of deviation from the centre and the paths that were taken by the rotor as it spun.

The ability to stabilise the rotor was then quantified by summing the standard deviations of the positions together for the recorded data and comparing with regards to the sum of the standard deviations of the currents. This was done for each controller at each speed as follows:

$$\sigma_{\text{pos}} = \sigma_{xs1} + \sigma_{ys1} + \sigma_{xs2} + \sigma_{ys2}, \quad (5.12)$$

$$\sigma_{\text{cur}} = \sigma_{i,a1x} + \sigma_{i,a1y} + \sigma_{i,a2x} + \sigma_{i,a2y}, \quad (5.13)$$

and served as metrics to demonstrate the ability to keep the rotor centred σ_{pos} for a given control effort σ_{cur} at each recorded speed.

5.8 Spin-up response comparison

With the five different controllers used in Sec. 5.7, the stability of the rotor was investigated during its acceleration. This was to see whether or not there were specific characteristics that would arise during rotor acceleration.

To carry out the test, the controller first had to be engaged as outlined in Sec. 2.2.5 and additionally Sec. 4.1.5 for the LQG controllers. From this point on, with the VFD set to 0 in the *off* setting, the desired speed was set to 1000 rpm (Fig. 2.8). Then, a 10 s recording was started where within this time frame, the *hand on* button on the VFD was pressed and the rotor was made to accelerate as fast as possible to 1000 rpm. This was the fastest possible acceleration achievable for the biggest difference in speed with the limited operation of the VFD available.

The recording of each controller test was analysed visually, by creating a time series for the positions and currents to identify any unusual behaviour from the time where the rotor was stationary to when it was steadily rotating at 1000 rpm. This was aided by identifying the maximum and minimum bounds in which the positions and currents would oscillate when the rotor was spinning.

5.9 Touch-recovery comparison

Another characteristic which was compared between the controllers was the ability to recover stability from when the rotor touched the boundary. This was tested between the PID and LQG-controlled systems, where the LQG was generated based on a model with a stationary rotor.

First, the PID-controlled rotor was engaged as outlined in Sec. 2.2.5 until *activate bearings* was pressed for the first time. Then, the *stepsize* value for $x1$ was increased in steps of 0.01 A, until the rotor would audibly and visually touch the bearing when *step on* was pressed with *no input* activated in the *radio buttons*. This occurred when the step size was defined as 0.32 A. A recording was then made based on the guidelines outlined in Sec. 5.1.1, where the touch and recovery of the rotor was captured when *step on* was pressed.

Next, the LQG was engaged as outlined in Sec. 4.1.5 and a step of the same magnitude which caused the PID-controlled rotor to touch the bearing was recorded. This was to compare the responses to the same disturbance. Then the *stepsize* was again gradually increased to the point where the rotor touched the bearing. This occurred with a step size of 0.37 A, and again a recording of the response was made.

Due to the found behaviour, the Kalman filter performance with the PID-controlled system and the rotor touching the bearing was evaluated as in Sec. 5.3. This was to investigate if the measured output was estimated equally well when the rotor touched the bearing as when this did not occur.

Table 5.2: Overview of the initialised parameters for the PID, LQG, and MPC comparison.

	Value	Unit
x	$1 \cdot 10^{-5}$	m
\dot{x}	0.003	m s^{-1}
y	$-1 \cdot 10^{-5}$	m
\dot{y}	-0.003	m s^{-1}
ϕ	$1 \cdot 10^{-4}$	rad
$\dot{\phi}$	0.022	rad s^{-1}
ψ	$1 \cdot 10^{-4}$	rad
$\dot{\psi}$	0.022	rad s^{-1}
i_{a1x}	0.006	A
i_{a1y}	0.006	A
i_{a2x}	0.0024	A
i_{a2y}	0.0024	A

5.10 MPC Simulation

In the course of the project, a working simulation of the test rig, as controlled by the PID, the LQG and the MPC was obtained. The likeness of the PID and LQG-controlled simulations to the real system was investigated and good similarity was found in the step response with a stationary rotor. Thus, the MPC was implemented in simulation knowing that the model could represent the real system to a certain extent. For the comparison, the LQG and the MPC were based on models where the rotor was considered stationary.

The MATLAB implementation of the MPC had its weights defined such that it could stabilise the rotor. Through trial and error, it was found that for a depth of $\kappa = 100$ steps, the weights:

$$Q_s = \text{diag}([1 \ 1 \ 1 \ 1 \ 0.01 \ 0.01 \ 0.01 \ 0.01]) \cdot 10^{11}, \quad (5.14)$$

$$S = \text{diag}([1 \ 1 \ 1 \ 1]), \quad (5.15)$$

stabilised the rotor. Additionally, the reference was set to 0 μm for the outputs at every step and the input was constrained to $\pm 15 \text{ V}$, in accordance with the real system AMB power supplies. Using these definitions, the matrices in Sec. 4.2.1 were calculated.

As the MPC implementation did not produce a step like the PID and LQG systems, due to its direct implementation to the FESS dynamics, the control of the rotor from an arbitrary initial state of the system was investigated. A simulation of the systems stabilising from this initialisation was carried out for a step count that corresponded to 0.02 s, as the simulation of the MPC-controlled system for this timeframe already took over 20 min. Thus, for each of the three simulated control systems, the rotor was initialised with

the the parameters outlined in Tab. 5.2, the measurement noise given in Tab. 5.1, and input noise according to [28].

Chapter 6

Results

6.1 PID-controlled test rig emulation

The ability to emulate the physical test rig in simulation was quantified by comparing the time series of the simulated and experimental outputs for the same step response tests. This allowed the analysis of the maximum deviation, rise time, damping, settling time, and differences in dynamical behaviour. From this analysis, it was deemed that the mathematical model was in accordance with the experimental tests in the primary response.

The step responses in the upper magnetic bearing can be seen in Fig. 6.1 & 6.2. The simulation followed the primary response, the response based on the direction being stepped in, quite well. The maximum deviation in the simulation was greater in comparison to the experiment by less than 15% and occurred in the direction that was stepped in. After reaching the peak, the experimental result showed some additional oscillations, as the step in the reference current was being compensated for. In either case, the change in position towards the maximum happened slightly faster experimentally, but settled slower compared to the simulation. The response in the current, due to the coupled nature, was very similar for the primary direction being stepped in. The complimentary response, the response in the same direction but in the unexcited bearing, differed more. In each case, the maximum deviation was emulated in simulation and occurred in the same direction, but experimentally, this was damped slightly better as the resulting oscillations from this had a lower amplitude. The same differences were reflected in the current for the complimentary response.

The step responses in the lower magnetic bearing are presented in Fig. 6.3 & 6.4. In these cases, the simulation followed the primary response slightly worse than for the upper magnetic bearing, however, the emulation was still good. Similarly, in the primary response, the difference in maximum deviation was below 15% and the response was damped better in simulation, which resulted in a faster settling time. The biggest difference appeared

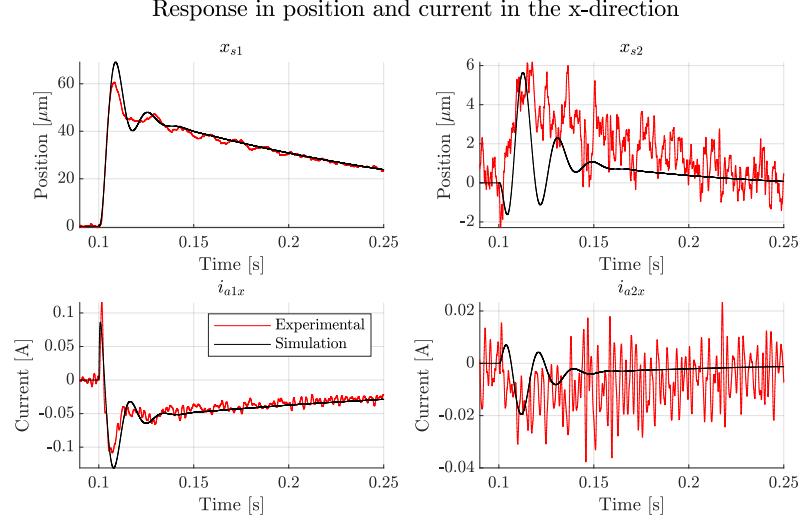


Figure 6.1: Step in x_1 performed experimentally and in simulation.

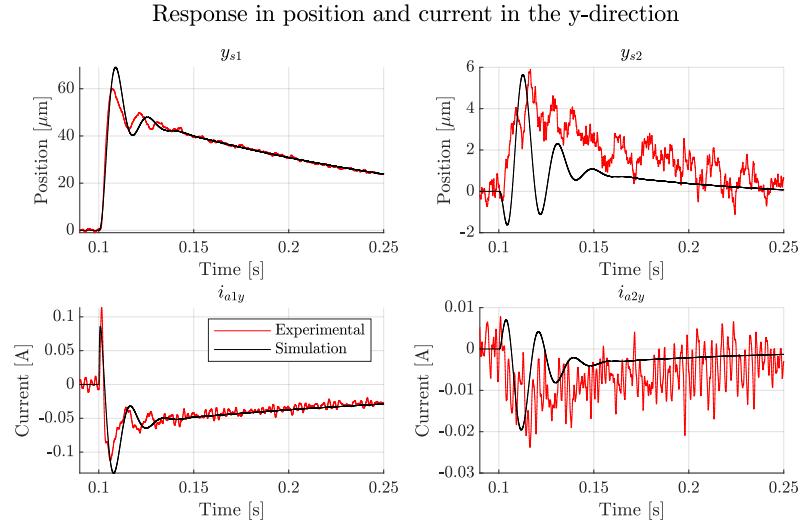
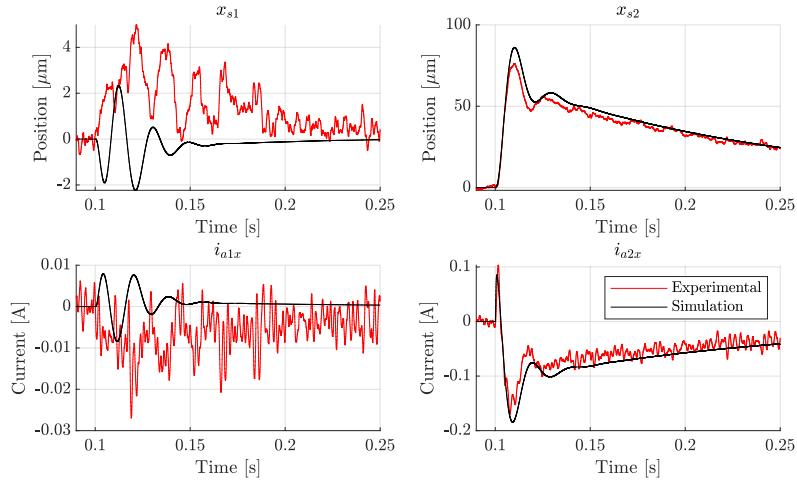


Figure 6.2: Step in y_1 performed experimentally and in simulation.

in the persistent offset after reaching the peak for a short period of time, which was not apparent in the responses from the upper magnetic bearing. The biggest discrepancy was found in the complimentary response for these steps. In these responses, neither the maximum deviation nor the settling times were similar. Experimentally, the top of the rotor moved more towards the direction in which the bottom part was being stepped into, while in simulation, an oscillation around 0 took place.

Response in position and current in the x-direction

Figure 6.3: Step in x_2 performed experimentally and in simulation.

Response in position and current in the y-direction

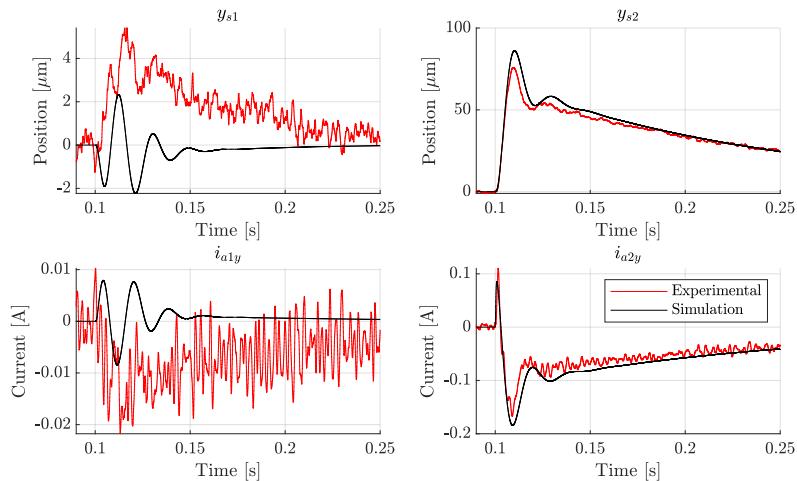
Figure 6.4: Step in y_2 performed experimentally and in simulation.

Table 6.1: Overview of discrepancies between PID-controlled experimental and simulation results in the primary response in position.

Characteristic	x_1 step	y_1 step	x_2 step	y_2 step
Max. deviation	13.95%	14.99%	11.54%	13.64%
Rise time	19.28%	23.89%	9.35%	0.87%
Settling time	-7.74%	-3.79%	-25.22%	-12.57%

The discrepancies between the experimental and simulation results for the primary responses are summarised in Tab. 6.1. All simulations deviated similarly from the experimental result in terms of the maximum deviation. However, as noticed in the time series, steps in the upper magnetic bearing had a more similar settling time but a higher disparity in the rise time. The opposite was true for the responses in the lower magnetic bearing.

6.2 Kalman filter performance

The Kalman filter was evaluated based on its ability to estimate the output of the real system. This could be seen in the output of the estimated states compared to the measured ones, the standard deviation of the innovations, and in the autocorrelation of the innovations. The results from this demonstrated the viability of the model for state estimation and the suitability of the selected weights.

When the rotor was stationary, the chosen Kalman filter was very capable at estimating both the position and the current in the system output. The difference between the measured output from the PID-stabilised system and the estimated output from the Kalman filter can be seen in Fig. 6.5. In the position estimates, the peaks of the measured output were followed well, but at a slight delay. In the current estimates, the larger peaks were not always estimated by the filter, additionally, oscillations were occasionally inferred although they were not present. The tuning of this filter based on the procedure outlined in Sec. 5.2, however, had optimised the ability to estimate the changes in position of the rotor at the lowest possible amount of delay.

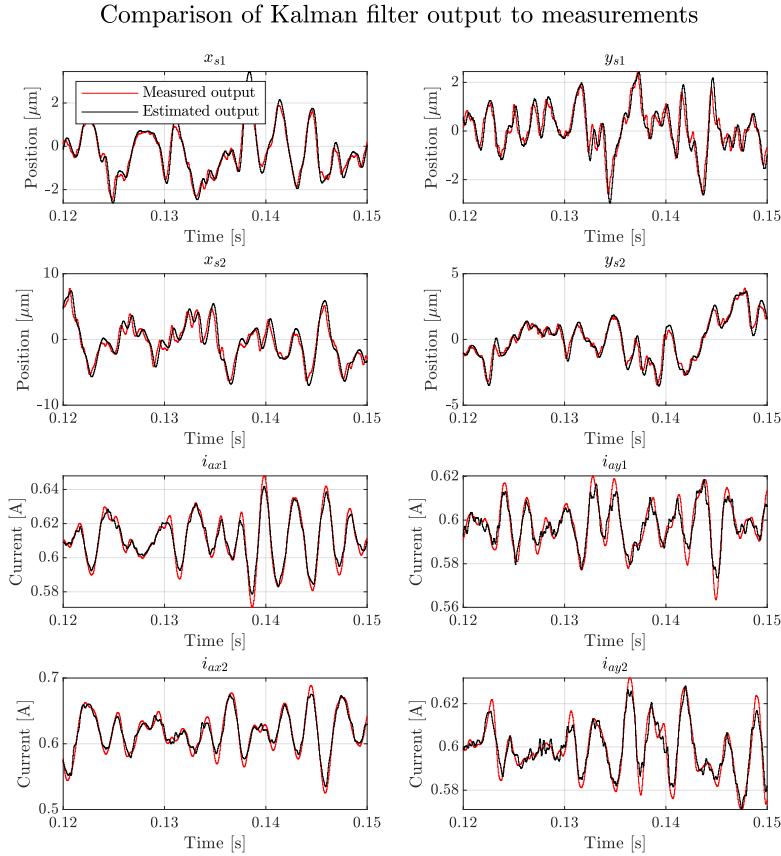


Figure 6.5: PID-stabilised rotor outputs at 0 rpm estimated using a Kalman filter based on 0 rpm.

At the highest set speed of the rotor, the Kalman filter was based on a model incorporating the additional gyroscopic effects at this rotation. The Kalman filter remained capable at estimating the output despite this rotation as seen in Fig. 6.6. Similar to when the rotor was stationary, the position was estimated much more closely than the current.

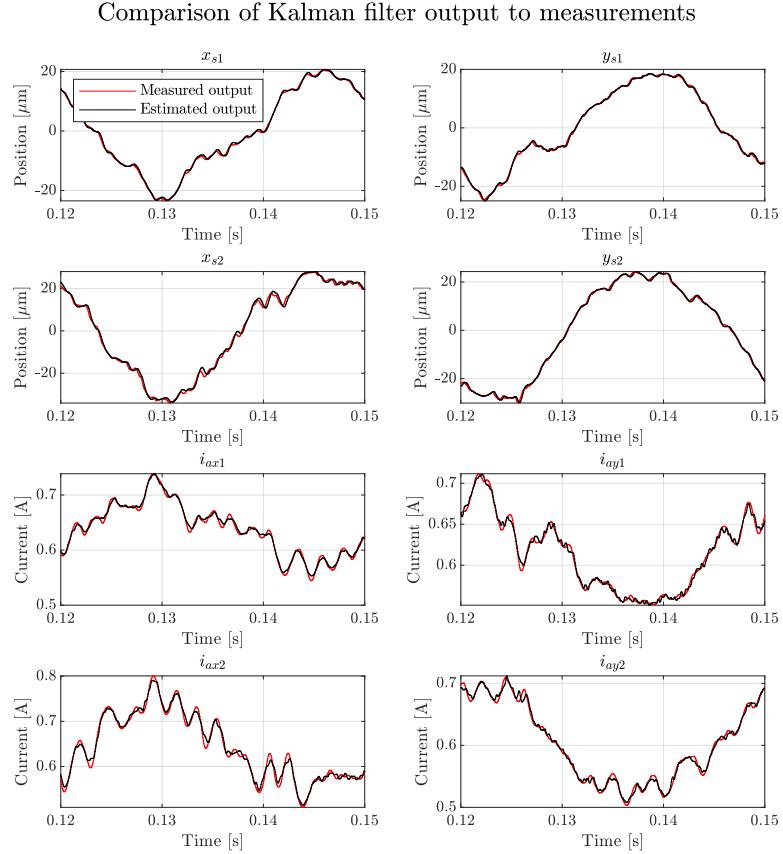


Figure 6.6: PID-stabilised rotor outputs at 2000 rpm estimated using a Kalman filter based on 2000 rpm.

From examining the time series in either case, the estimated output was deemed sufficiently similar to the measured output, indicating that the inferred states were accurate enough based on the applied model-based approach. Additional analyses were made to see the effect of the incorporated gyroscopic effects in the model on the ability to estimate the system. The result is summarised in Fig. 6.7, based on the sum of the standard deviations of all positions and currents at a range of speeds from 0 to 2000 rpm.

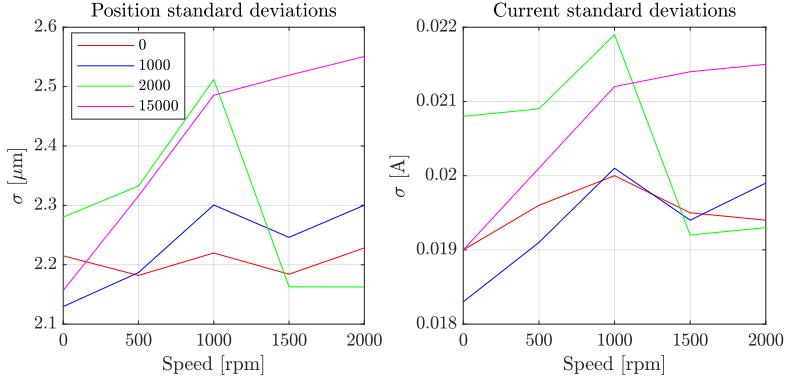


Figure 6.7: Sum of standard deviations in position and current over a 1 min period at different speeds for different set models for Kalman filtering.

It can be seen that, at 2000 rpm, the definitively best performing Kalman filter was the one incorporating the gyroscopic effects at this speed. This was not observed when stationary or at 1000 rpm. The filter that included the excessively high speed performed, with the model for 2000 rpm, the worst at 1000 rpm and the filter for 15 000 rpm started performing significantly worse than the others at higher speeds. The models for 0 rpm and 1000 rpm performed fairly consistently over the speed range with the least amount of deviation over this range. In all filters, a noticeable decay in performance was found going from 500 to 1000 rpm, which in 3 of the 4 cases subsided at 1500 rpm.

An additional test that was applied to the innovation signal was the autocorrelation function. The autocorrelation of the innovation is shown for one of the position and current signals in Fig. 6.8, as the other signals showed the same behaviour. From this test, it is clear that, as the autocorrelation was not predominantly within the bounds at low lag, the difference between the estimated and measured output was not dominated by white noise.

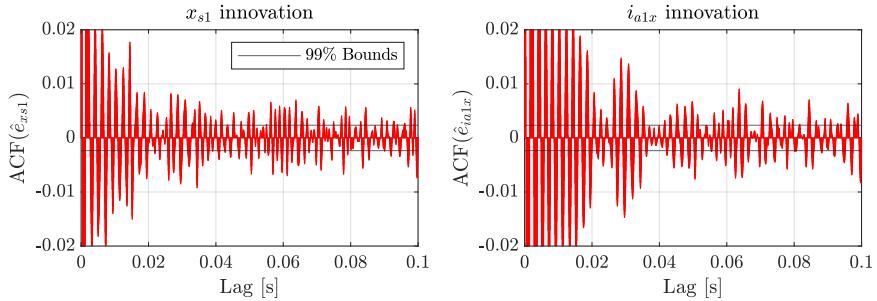


Figure 6.8: Autocorrelation function on the innovation signal for x_{s1} and i_{a1x} obtained for the Kalman filter based on the 0 rpm model.

6.3 LQG-Controlled test rig

The performance of the LQG controlled system was compared against the performance of the PID as well as LQGs based on varying incorporations of the gyroscopic effect. Firstly, the simulation of the LQG was verified against the experimental output (Sec. 6.3.1). Secondly, the recorded step responses were compared to the PID compensating for the same offset (Sec. 6.3.2). Thirdly, the ability to keep the rotor centred when it rotated was observed (Sec. 6.3.3). Then, the differences in the responses when the rotor was accelerating were assessed (Sec. 6.3.4). Lastly, the ability to compensate for a deliberate touch of the boundary was examined (Sec. 6.3.5).

6.3.1 LQG-Controlled test rig emulation

Previously, the simulation of the PID-controlled system was verified against the experimental results as a way to validate the mathematical model. The comparison of the simulated LQG model with the experimental LQG results was performed to validate the possibility of gauging the performance of the controller before its implementation. This analysis showed that the simulated LQG-controlled system was similar enough to provide an idea of the performance to be expected in the real system in a step response test.

For steps in the upper magnetic bearing, the comparison is displayed in Fig. 6.9 & 6.10. It was observed that the maximum deviation was much lower in simulation in the primary response of the position. While the compensation for the step decayed at a similar rate, the discrepancy in the maximum deviation meant that the settling time was lower in the simulated

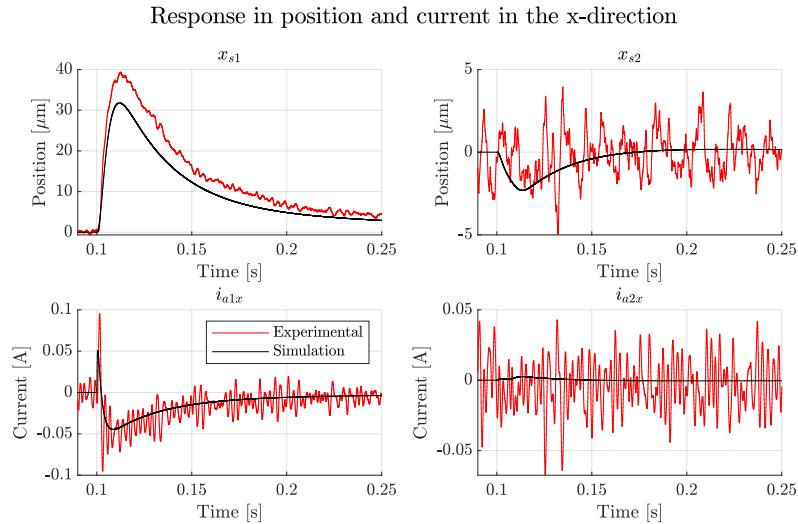


Figure 6.9: Step in x_1 performed experimentally and in simulation.

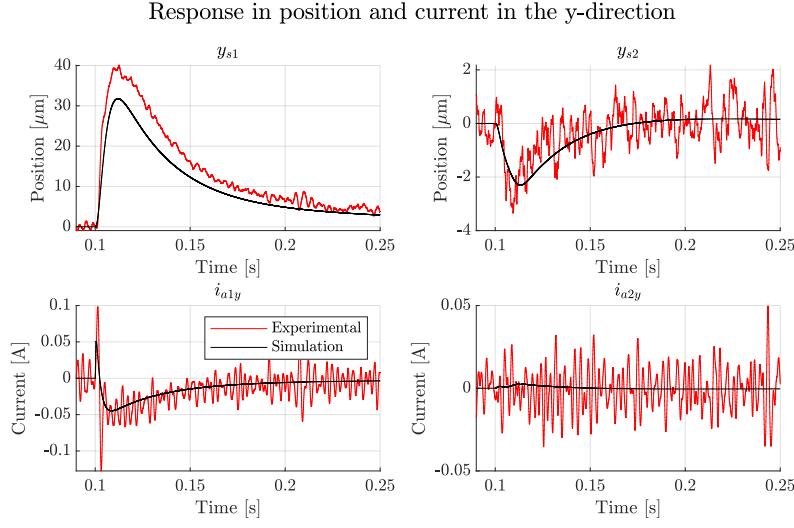


Figure 6.10: Step in y_1 performed experimentally and in simulation.

response. Interestingly, the change in current was very similar, barring the noise as well as the change in the complementary responses.

The similarity in the observed responses for the lower magnetic bearing is shown in Fig. 6.11 & 6.12. Similar to what was found in the upper magnetic bearing in the primary response, the model underestimated the experimental output. As a result, the maximum deviation, rise time, and settling time were overall lower in comparison. Additionally, as with the upper bearing

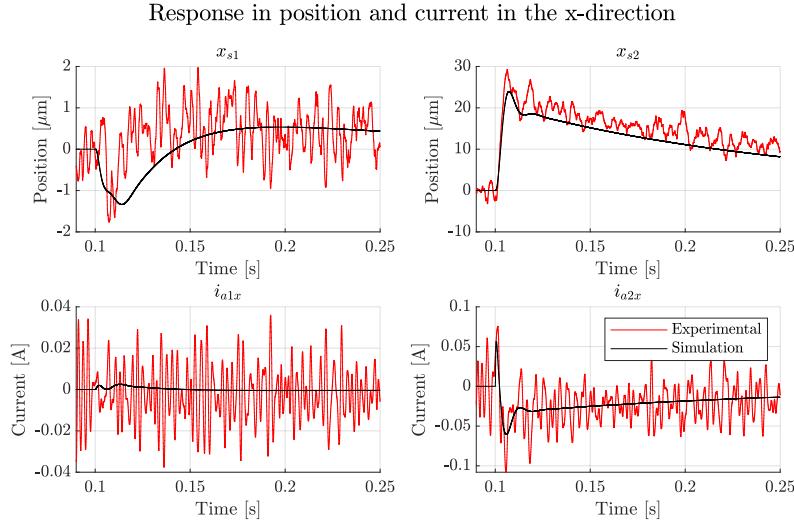


Figure 6.11: Step in x_2 performed experimentally and in simulation.

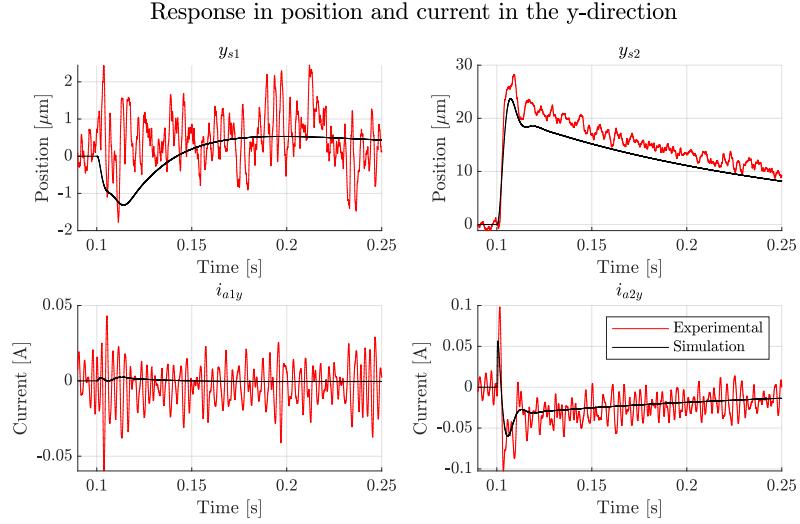


Figure 6.12: Step in y_2 performed experimentally and in simulation.

responses, the simulated current in the dominant response was also similar to the measured output. The complementary response, however, was not found to be recreated accurately in simulation, especially in the y_2 -step test (Fig. 6.12).

The differences between the experimental and the simulated results are summarised in Tab. 6.2. In general, it can be seen that the simulated response underestimated the main characteristics in the dominant response. On the other hand, the simulated PID responses overestimated what would occur in practice. Both, however, had underestimated the settling time of the real system, the LQG-simulated responses more so.

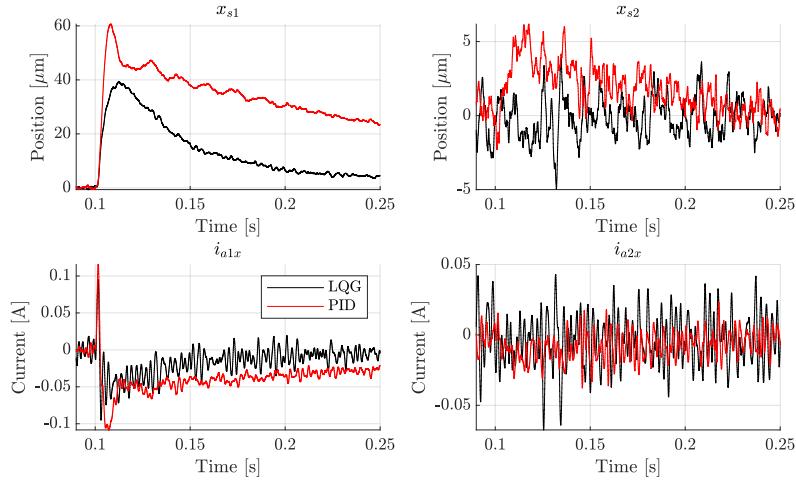
Table 6.2: Summary of discrepancies between LQG-controlled experimental and simulation results in the primary response in position.

Characteristic	x_1 step	y_1 step	x_2 step	y_2 step
Max. deviation	-19.22%	-20.79%	-18.14%	-16.08%
Rise time	4.35%	1.56%	-11.59%	-10.39%
Settling time	-26.28%	-31.51%	-47.88%	-29.40%

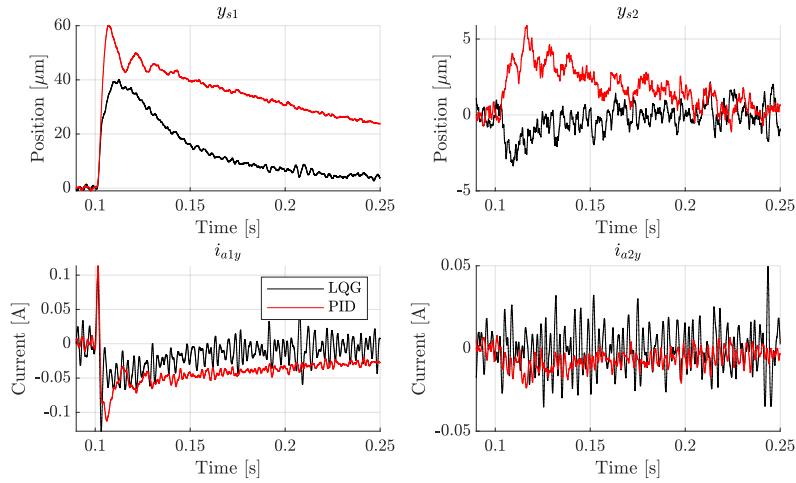
6.3.2 Step response comparison

On the basis of the experimental step-response tests for both the LQG and the PID-controlled systems, the control strategies were compared to observe any improvements garnered through the application of the model-based control. It was found that the implementation of the LQG improved the

Comparison between LQG and PID-controlled experimental responses

Figure 6.13: Differences in response between the LQG and PID-controlled systems to a step in x_1 .

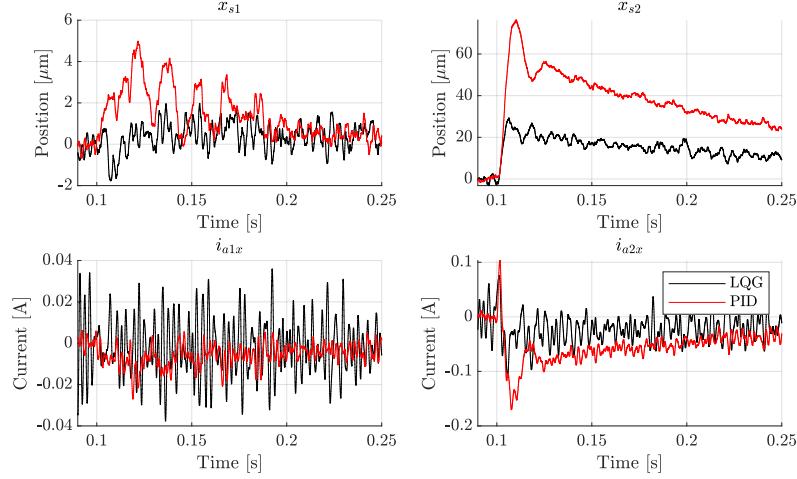
Comparison between LQG and PID-controlled experimental responses

Figure 6.14: Differences in response between the LQG and PID-controlled systems to a step in y_1 .

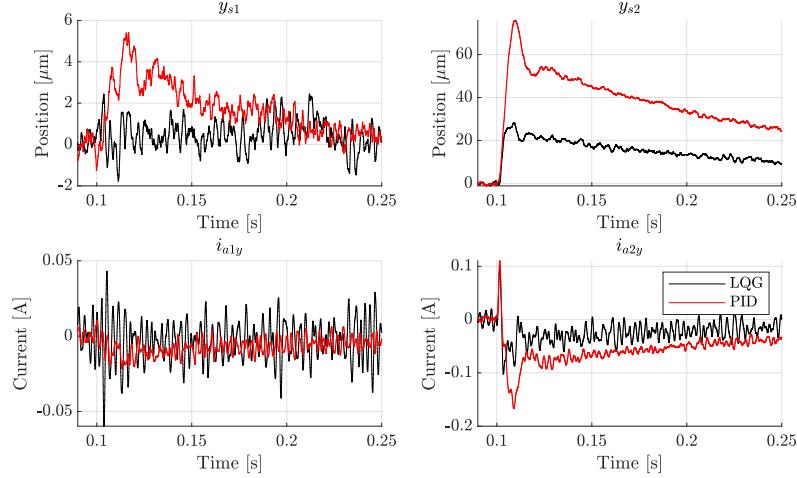
control of the rotor in all metrics, allowing for a more centred position of the rotor despite the stepped offset.

The differences between the control strategies can be seen in Fig. 6.13–6.16. Overall, it was observed that the LQG-controlled system showed a much lower deviation from the centre than the PID-controlled system. Likewise,

Comparison between LQG and PID-controlled experimental responses

Figure 6.15: Differences in response between the LQG and PID-controlled systems to a step in x_2 .

Comparison between LQG and PID-controlled experimental responses

Figure 6.16: Differences in response between the LQG and PID-controlled systems to a step in y_2 .

the step disturbance to the current was compensated for much faster in comparison.

The improvement in the response of the LQG over the PID control strategy is summarised in Tab. 6.3. In addition to the previously applied characteristics, the total positional and current error while the system settled

Table 6.3: Improvement of the LQG control strategy over the PID.

Characteristic	x_1 step	y_1 step	x_2 step	y_2 step
Max. deviation	-35.15%	-33.31%	-61.61%	-62.76%
Rise time	56.63%	80.56%	11.40%	-34.95%
Settling time	-71.57%	-74.38%	-44.52%	-39.66%
Total position error	-76.87%	-77.63%	-65.99%	-64.52%
Total current error	-62.47%	-61.36%	-28.29%	-30.18%

to 0 from the step, was calculated. In comparison to the PID-controlled system, the LQG showed lower maximum deviation, settling time, and errors, indicating a better control of the system in the step response. For three of the four step responses, the rise time increased which indicates a better damping of the rotor to the abruptly applied step. This attribute, however, is not as impactful on the response, as though it was found to be lower in the y_2 step, the dataset still showed comparable metrics in all other characteristics with the x_2 step, which occurred in the same magnetic bearing.

6.3.3 Spin response comparison

The aim of the flywheel is to store energy in its rotation. When rotating, however, the flywheel should stay as close as possible to the centre for the lowest amount of energy. This makes it more robust to unknown disturbances that might bring the flywheel to the boundary and wastes the least amount of energy for the storage. It was found that for the speed range available, the PID-controlled system used the least amount of energy for much higher deviations from the centre. The deviations from the centre as well as the energy used for this, however, trended upwards much faster as the speed increased than for the LQG-controlled system.

The nature of the PID and LQG control strategies when the rotor was spinning at the highest possible speed can be seen in Fig. 6.17 & 6.18, respectively. Depicted are the positions across the upper and lower sensing planes over a 5 s period. Within this time frame, it is clear that the LQG-controlled system kept the rotor much closer to the centre than the PID system. In either case, there was an observed, preferred, non-elliptical path being taken by the rotor across the top plane. The lower part of the rotor followed a more elliptical path in the PID system further from the centre than the upper part, while in the LQG-controlled system, the deviation of this part of the rotor was kept similar to that of the top of the rotor.

Additional analyses were made on the PID and different LQG settings at different speeds. This is summarised in Fig. 6.19. It can be seen that across the range of speeds, the PID exerted, overall, the least amount of effort which increased more rapidly with speed than the LQG controllers. However, it

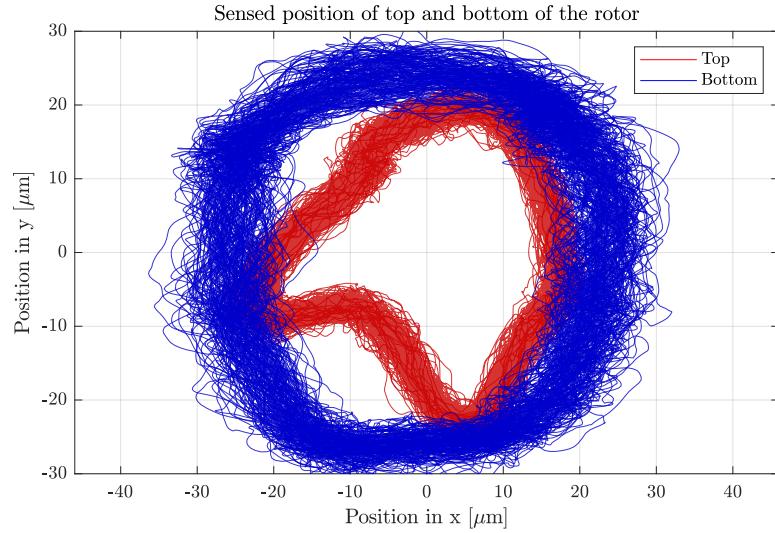


Figure 6.17: Rotor spinning at 2000 rpm controlled by the PID controller.

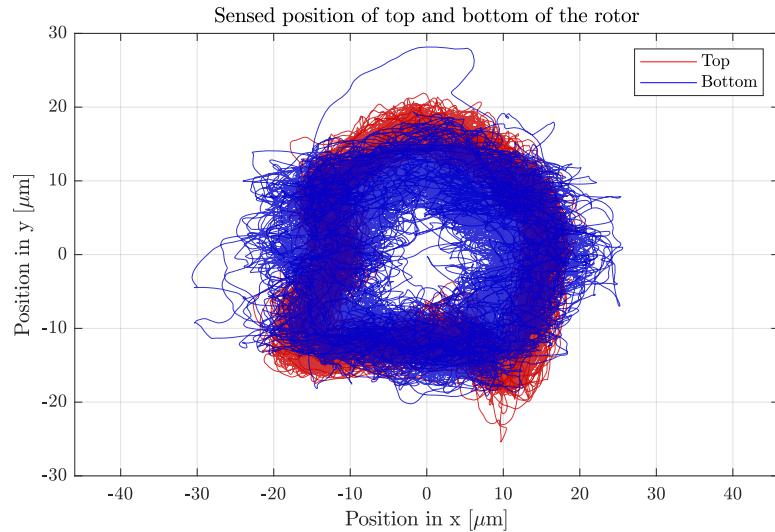


Figure 6.18: Rotor spinning at 2000 rpm controlled by LQG controller set for 2000 rpm.

performed significantly worse at keeping the rotor centred above 1000 rpm. For the LQG controllers, the controller for a set speed did not always necessarily perform the best. Moreover, the performance was only different in position for the LQG set for 15 000 rpm, otherwise, the performance between the LQGs were very similar over the tested range. Interestingly, all controllers both PID and LQG, experienced a decrease in performance at

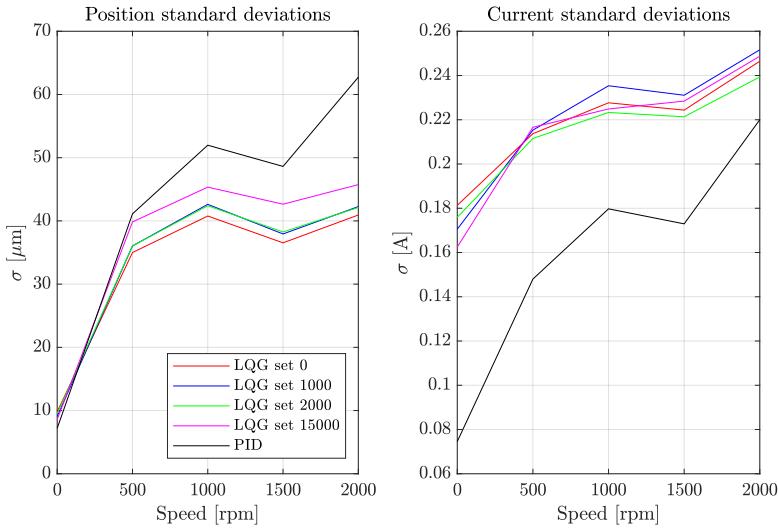


Figure 6.19: Total standard deviation of the combined positions and currents across the available speed range for different controllers.

1000 rpm, followed by a better performance at 1500 rpm.

6.3.4 Spin-up response comparison

In this test, the aim was to observe whether or not any of the controllers performed significantly worse when the rotor was accelerated as fast as possible. However, given the capabilities of the system, which was only able to accelerate the rotor from 0 to 1000 rpm, a difference in the responses was not observable.

When the rotor was spinning, the position of the rotor as well as the current to control it showed a distinct periodicity according to the speed of rotation. The change from stationary to this periodic state controlled by one of the controllers can be seen in Fig. 6.20 & 6.21. It was observed, however, that neither the position nor the current experienced a deviation significantly larger than the amplitude that followed when it was constantly spinning. This was the case for all the controllers tested (PID, LQG at 0, 1000, 2000, 15 000 rpm), thus, this test showed that all controllers were similarly capable of controlling the acceleration of the rotor.

6.3.5 Touch-recovery comparison

The touch-recovery test aimed to evaluate how capable the PID and LQG controllers were at compensating for an impulse that caused the rotor to touch the boundary. It was found that, although the LQG controller was able to compensate for an impulse of the same step as the PID without the

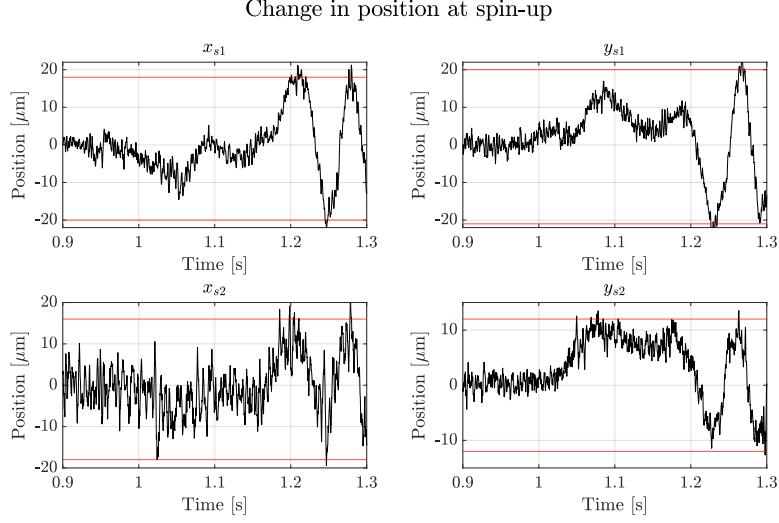


Figure 6.20: Position of rotor for LQG set for 2000 rpm at spin-up from 0 to 1000 rpm. The red line indicates the bounds between which the signal oscillated when spinning at steady state.

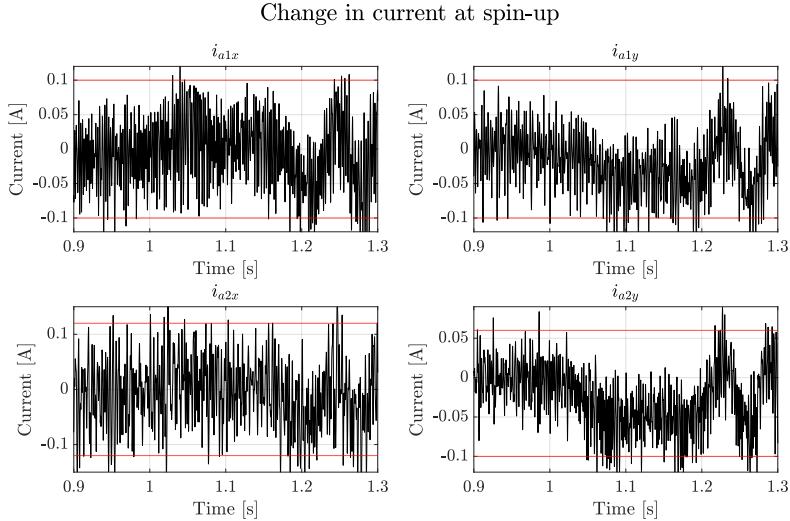


Figure 6.21: Current in system for LQG set for 2000 rpm at spin-up from 0 to 1000 rpm. The red line indicates the bounds between which the signal oscillated when spinning at steady state.

rotor touching the boundary, it was unable to bring the rotor back to a stable position once it had touched the boundary.

The PID stabilising the rotor after it had touched the boundary is displayed in Fig. 6.22. It can be seen that the direction being stepped in,

x_1 , experienced a significantly higher change in position than the rest of the directions. This occurred until the boundary was hit at below 400 μm . At this point, the offset in the current was still active, however, the PID controller compensated for it as seen in the other measurements tending towards 0. After the impulse was removed, the rotor was brought back to its fully vertical position once again.

Applying an impulse of the same magnitude to the LQG-controlled system, resulted in the response seen in Fig. 6.23. With this controller, the maximum deviation in x_{s1} was 250.79 μm from the boundary and measurements from the other directions also remained close to the centre.

However, when the impulse magnitude was increased by 0.05 A in the LQG-controlled system, the boundary was struck and the response in Fig. 6.24 was observed. In this case, the boundary was touched by the rotor, which then overshot the centre and touched the boundary on the opposite side after the impulse was lifted. This caused the rotor to oscillate in the x -direction, as the rotor was periodically hitting the boundaries in the upper and lower portion. Eventually, this caused the rotor to oscillate in the y -directions as well after an approximately 10 μs delay, which ultimately ended in an unstable behaviour with the rotor ends rotating about the centre at 5000 rpm along the boundaries.

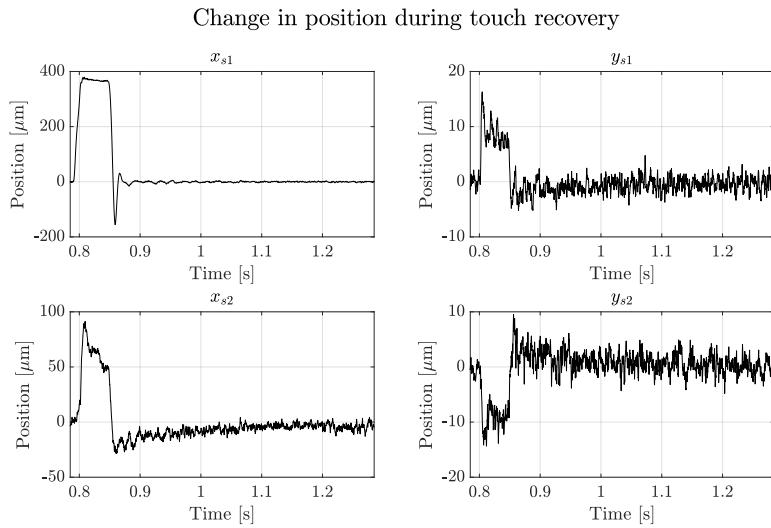


Figure 6.22: PID-controlled rotor recovering after touching boundary.

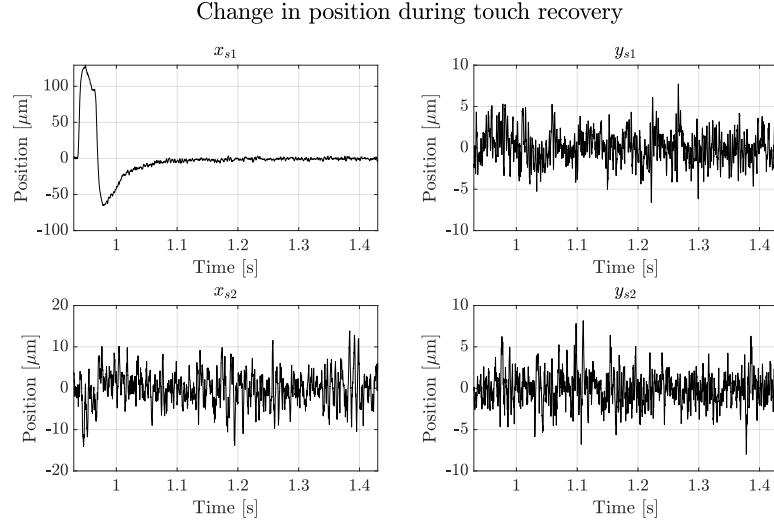


Figure 6.23: LQG controller set for 0 rpm compensating for the same impulse that caused the PID to touch the boundary.

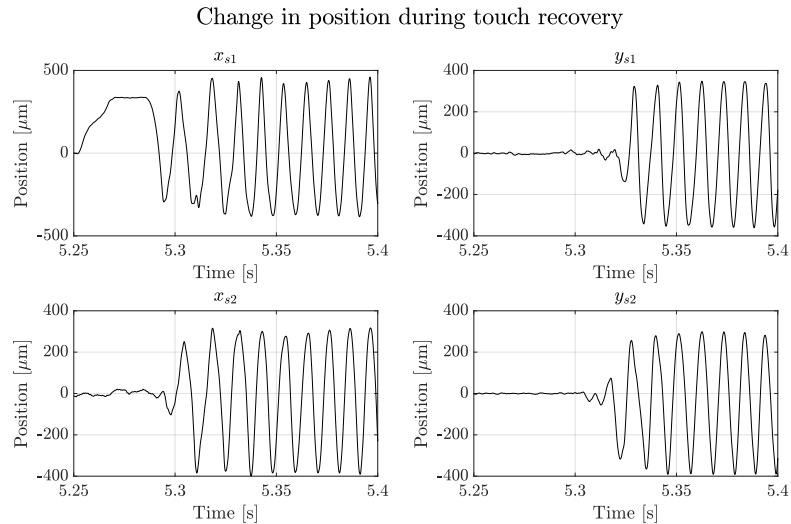


Figure 6.24: LQG controller set for 0 rpm subjected to an impulse causing the rotor to touch the boundary.

The cause of this behaviour was investigated by checking whether the estimation of the states was a potential culprit. The result is shown in Fig. 6.25. It can be seen that, despite the rotor going into the boundary, the Kalman filter was still able to estimate the output of the system well, as the measured output was being followed.

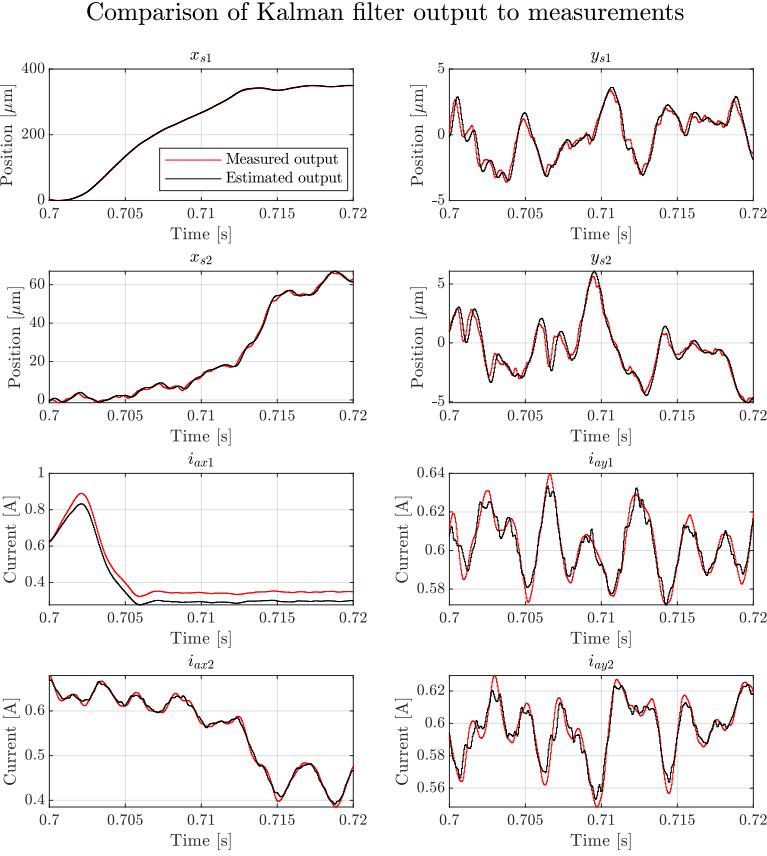


Figure 6.25: Estimation of states in the PID controlled system when the rotor is brought to the boundary.

6.4 MPC Simulation

The validity of an MPC controlled system was confirmed by implementing an input-constrained MPC to the derived model of the flywheel system. The result, carried out in simulation, was compared to the PID and LQG controlled systems subjected to the same conditions with noise implemented on the inputs and outputs to emulate the real system in an optimal fashion. The comparison is depicted in Fig. 6.26 & 6.27.

In terms of the change in position, the MPC was able to rapidly stabilise the position of the rotor towards the vertical position. The LQG took somewhat longer and the PID took the longest amount of time. The control effort, on the other hand, was highest in the MPC, which also displayed an oscillating current signal. The LQG used a lower control effort, while the PID used the lowest control effort.

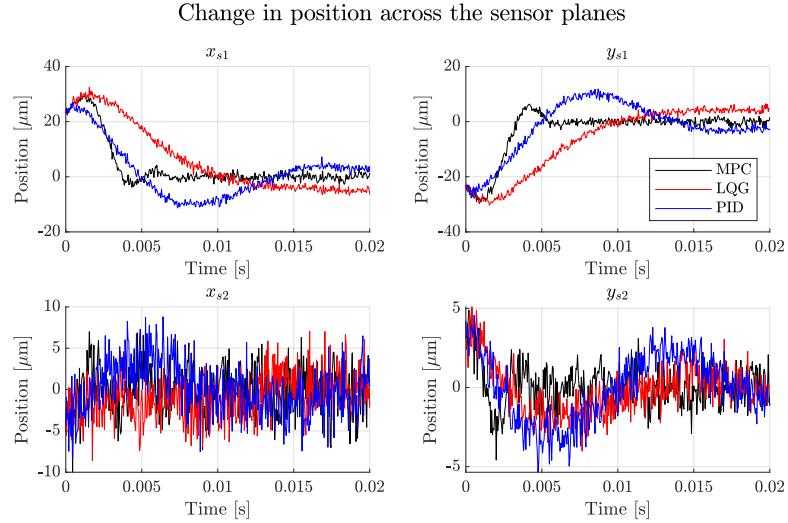


Figure 6.26: Positioning of the rotor to the same initial conditions by the MPC, LQG, and PID-controlled systems.

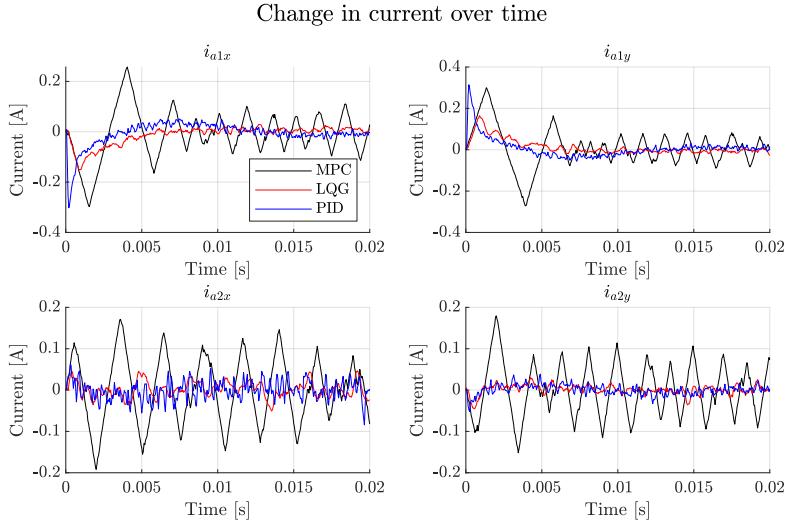


Figure 6.27: Changing current for the stabilisation of the rotor by the MPC, LQG, and PID controllers to the same initial conditions.

Chapter 7

Discussion

7.1 PID-controlled system

The only working control system on the test rig before the start of this project was the PID-based system. The tuning of the PID was such that the performance was optimised when the rotor was stationary.

It was observed that in comparison to the LQG implemented later, the PID controller was very steady when the rotor was stationary, at a low control effort (Fig. 6.19). Additionally, despite the addition of the motor in the middle of the shaft and introducing changes in the system dynamics, the PID-controlled system remained able to adequately stabilise the rotor. This was a result of the PID relying solely on the positional changes being sensed in both planes, using a decentralised control scheme, and the gains remaining suitable for the new system.

This approach, however, showed drawbacks in the step-response and spinning tests. As control effort was only evoked by a positional change, when a step was induced in an orthogonal direction in one of the AMBs, a large deviation was observed in this direction as the step was mostly being compensated for by the electromagnets responsible for this specific direction. Intuitively, a deviation in one direction in one of the AMBs can also be minimised by forcing the rotor in the opposite direction in the other AMB. By knowing the rotor dynamics, as with the LQG, this principle would be applied.

When the rotor was spinning, gyroscopic effects and the inherent uneven distribution of mass around the axis of rotation made it difficult to keep the rotor centred. As the PID reacted solely to the sensed position, the deviation from the centre increased with the spinning speed of the rotor. This behaviour can be seen in Fig. 6.19, where the standard deviation in the recorded position measurements increased almost linearly within this range of speed. It was not possible to verify whether this trend would continue to persist at speeds higher than 2000 rpm.

Interestingly, when the rotor was spinning, the PID-controlled system controlled the rotor along a very consistent path (Fig. 6.17) with less variance than seen in the LQG (Fig. 6.18). It was found that the bottom part of the rotor followed a more elliptical path, whereas the top part of the rotor followed a non-elliptical unique shape. This unique shape was also observed at speeds below 2000 rpm. This was not the case in [21], though shown at 600 rpm, therefore, the behaviour was thought to be a result of changes made to the system in [22]. These changes included alterations to the housing, the addition of an electrical motor that influences the dynamics and changed the manner in which the rotor was being spun, and the alteration of the rotor to accommodate the motor, all of which may have led to the described behaviour.

7.2 System emulation

The mathematical model of the system that was used throughout the project was a system based on measured parameters in [20] and [22]. Additionally, linearisations and simplifications were made to the characteristics of the magnet and, ultimately, to the system dynamics. The controller and filter parameters were known and applied based on a known sampling time.

Therefore, the discrepancies in the output shown in Sec. 6.1 & 6.3.1 were largely rooted in the differences between the mathematical model of the flywheel and the real flywheel system. When deriving the model, the rotor was assumed to be symmetric, the contributions to the dynamics by the motor to be negligible, the structure holding the housing and rotor to be rigid, the system characteristics to be invariant with time, and the forces by both the AMB and PMB to be linear. Over the course of the project, it was found that the mass of the rotor was non-uniform, that the magnets of the PMSM generated a noticeable force, that the structure would rattle at certain spin frequencies, that temperature can greatly affect the performance of the system, and that the nonlinearity of the magnetic forces required the LQG-controlled system to operate as close as possible to the assumed point of operation.

The unbalanced rotor meant that when the rotor was spinning, an unmodelled moment would generate a movement characteristic that would generate wobbling. This may be a cause to the non-elliptical path taken by the top of the rotor seen in Fig. 6.17 and, in general, is the reason for active control to keep the rotor as centred as possible in FESSs. It also may have accentuated the loosened structure holding the housing which at resonance, observed at around 1000 rpm, would noticeably vibrate the entire test rig. This resonance frequency also explains why a general increase in deviation from the centre was observed at this speed (Fig. 6.19).

The effect the PMSM had on the rotor dynamics was noticeable when

stationary. When turning the rotor slightly by hand while it was simply levitated, a slight resistance to the turn was experienced. This is known as a cogging torque that is generated as a result of an interaction between the magnetic fields of the rotor and stator parts of the PMSM [29]. Thus, the addition of the cogging torque and PMSM overall may have contributed greatly to the change in the dynamics from the older system, which was not included in the modelling for this project.

Over time, it was often experienced that for maladjusted LQG weighting matrices, the performance of the stabilisation would deteriorate. This would present itself in the form of high frequency oscillations where the rotor was initially near stationary. This would most likely occur as a result of increased temperature affecting the electrical components of the system, but also due to a drift in positional calibration of the system. The drift was highly noticeable after the test rig was subject to high vibrations such as when the LQG failed to stabilise the rotor after it had touched the boundary (Fig. 6.24). Therefore, this behaviour was most likely due to the compromised rigidity of the test rig.

Despite omitting the mathematical description of these characteristics in the model, the simulation of the controlled systems was fairly similar in the primary responses (Sec. 6.1 & 6.3.1). One outstanding difference between the PID and LQG comparisons was that in the LQG simulations, the response undershot the experimental findings, whereas with the PID, the simulation overshot the recording. This was most likely a result of the LQG being applied to the model it closely resembled in simulation, and being applied to the real system which it only approximated.

An attempt was made at performing a parameter tuning routine based on a least-squares approach. This involved using experimental recordings of the step response tests to adjust linearised parameters, in order to improve the emulation of the experimental data by the simulation of the system. However, unlike the results obtained in [21], most parameters would have had to be adjusted in excess of 50% from the original, which would indicate an incorrect measurement of many of the system parameters by [20]. This was deemed implausible, especially as the initial model found emulated the experimental result better than in [21], but was likely a result of not incorporating the dynamics caused by the additions to the system in [22]. Thus, as the model was already closely emulating the experimental responses, the derived model was deemed sufficient to apply an LQG on the real system.

7.3 LQG implementation in LabVIEW

To implement the LQG in LabVIEW, the augmented system had to be arranged in the program. As memory was limited in the real-time computer and the timing of the calculations is of utmost importance, the maximum

degrees of freedom available and the precision of the values used had to be preallocated.

The effect of these measures on the control performance was difficult to quantify, as an exact emulation was not easily implemented in MATLAB. As a result, they were minimised by using a balanced realisation of the system to reduce the difference between the highest and lowest values of the augmented system. This would allow more bits to be allocated in the length of the fraction rather than integer. Next, this realisation was reduced to 30 states, which was determined as a limit in the amount of degrees of freedom that could be represented in the program with the available memory. Based on simulation results, the LQG should have been able to stabilise the system with as few as 20 states, however, this was not verified experimentally given that the inclusion of more states also improved the control performance. Finally, the numbers used in the LQG control loop were all defined as fixed-point numbers with an appropriate range and precision to ensure sufficiently fast execution.

The more precisely the LQG was represented in the program, the better the control strategy would be executed. However, this was limited by the memory available in the cRIO, any unmodelled dynamics of the system neglected in the LQG, and ultimately personal inexperience with LabVIEW programming. This may also be a reason for the experimental result overshooting the performance in simulation (Sec. 6.3.1).

7.4 Kalman filtering

Based on the separation theorem, the design of the observer and controller of the system may occur separately. Therefore, due to the limitation in memory in the real-time computer, a separate program was created to test the Kalman filter.

Ideally, the Kalman filter should be tuned based on weights relating to the process and measurement noise. The measurement noise should be determined based on the standard deviations of the sensors while the process noise should be estimated and tuned to give the best performance.

Experimentally, however, the issue with this method was that defining the measurement noise solely based on the recorded signals was inaccurate. When the rotor was levitated, the errors in the measurement and the inter-dependency between position and current resulted in a higher measurement error than expected, which would also depend on the applied controller. Therefore, the exact strength of the noise in the position and current sensors could not be verified by the system output, and instead the method outlined in Sec. 5.2.2 was applied. Additionally, for simplicity, the process noise was arranged such that it correlated with the input noise, which reduced the amount of parameters to be tuned but also compromised the ability to

individually adjust the process noise found in any of the 40 states.

By trial and error, it was then found that an increase in process noise and measurement noise related to the current, significantly improved the ability to track the measured output when the rotor was controlled by the PID (Fig. 6.5 & 6.6). Consequently, this also improved the performance of LQG controllers using the same Kalman filter.

When evaluating the Kalman filter, a number of filters were tested that differed based on the model used to construct the gains. To obtain the Kalman filters, the models used differed only in the speed that was implemented. It was found that, despite a filter being modelled for a specific speed, it was not necessarily the best performing filter (Fig. 6.7). In fact, the best performing Kalman filter over the entire speed range was based on the model incorporating 0 rpm. This may indicate that the gyroscopic effects within this speed range are insignificant, but also that the weights chosen were best adjusted for this specific model used.

From Fig. 6.7, it was also gathered that there was an overall decrease in performance across all filters at 1000 rpm. At this speed, it was noticed that the test rig setup would vibrate, indicating a natural resonance of the system near this frequency, resulting in an additional disturbance that most likely caused the observed decrease in filter performance.

Finally, from the autocorrelation function (Fig. 6.8) it was evident that unmodelled dynamics were affecting the estimation, as the innovation signal was not purely affected by white noise. A mostly white noise signal would be 1 at 0 lag and clearly within the bounds as lag increases. In this case, the autocorrelation of the innovation was not displaying white noise behaviour, but despite this, the estimated output was still quite comparable to the measured output (Fig. 6.5 & 6.6).

7.5 LQG performance

From the base controller obtained through the procedure in Sec. 5.4.1, it was found that the best spinning performance, which kept the rotor closest to the centre, was obtained by weighting the input on the upper AMB lower than that of the lower AMBs. In doing so, the lower part of the rotor, which was more affected by the radial forces generated by the PMB, was excited less, reducing the amount of movement caused by compensating for the PMB forces. This caused the lower part of the rotor to wobble closer to the centre within the deviations of the wobble experienced in the upper part (Fig. 6.18). Thus, the actuation by the upper AMB allowed for compensation of higher deviations experienced in the lower part of the rotor. This explains the difference between what is seen in the PID-controlled spinning (Fig. 6.17) and LQG-controlled spinning (Fig. 6.18).

The improved control of the rotor position also coincided with better

performance of the LQG in comparison to the PID in the step-response tests (Sec. 6.3.2). In these tests, it was observed how the lower part of the rotor was controlled for differently than the upper portion. Interestingly, despite the control effort being punished more harshly in the lower AMB, maximum deviation was lower than in the upper AMB step tests, due to the action of the upper AMBs. However, the effect of the higher control weighting was apparent in the settling time from this deviation.

The main advantage in the use of a model-based controller is best visualised in Fig. 6.10. Here, it can be seen that a deviation in y_{s1} was not only compensated for by its actuation in this direction, i_{a1y} , but that the lower part of the rotor was forced in the opposite direction of this movement to minimise the deviation and bring it back to 0 as fast as possible. This is the opposite of what occurred in the PID-controlled system, where the deviation in the upper part also pulled the lower part in the same direction from the centre, ultimately adding to the maximum deviation from the centre.

With the LQG controller in its provided configuration, it was not possible to initiate the stabilisation of the rotor from its lying state. This would have required an appropriate initialisation of the states pertaining to the LQG system. Instead, the rotor stabilisation was always initiated using the PID controller and gradually transitioned into the LQG from 0% contribution to the reference current for the PI to 100% contribution (Sec. 4.1.4). This gradual transition was helpful when testing unknowingly unstable LQG configurations without risking excessive damage to the test rig.

Another shortcoming of this LQG controller was the demonstrated inability to recover from touching the boundary (Fig. 6.24). Unlike in the system from [21], the rotor would fall into a highly unstable behaviour from touching the boundary. When the rotor touched the boundary, it introduced a disruption in the expected dynamics that were provided in the mathematical model for the LQG. However, as inspected by the Kalman filter individually, the act of hitting the boundary did not compromise the ability to estimate the states (Fig. 6.25). Therefore, it is assumed that the unstable behaviour is elicited by the LQR portion of the LQG generating an overcompensation for the high deviation from the centre.

An aspect of the LQG which was investigated was how the inclusion of gyroscopic effects in the model would improve the control of the rotor when it was spinning. This was marginally seen in Fig. 6.19, where the LQG set for 2000 rpm gave the same deviation from the centre as the one set at 1000 rpm for a lower control effort. Overall, however, the controller that performed the best at keeping the rotor centred was the one set for 0 rpm and, in general, the difference between the LQGs was not very significant. This may indicate that the gyroscopic effects experienced by the rotor within this speed range were not very evident. However, this may also be in part due to the overall best observer being the Kalman filter set for 0 rpm, as shown in Fig. 6.7. As expected, out of the LQGs, the one set for 15 000 rpm

performed the worst. However, it was not possible to see if it would perform better for its set speed, as the motor was limited to 2000 rpm.

In general, the analysis of the different LQGs at speed showed that when the rotor was spinning, the LQG would, based on the trend displayed in the available speed range, eventually outperform the PID in terms of the ability to keep the rotor centred for a lower control effort. This could not be shown definitively, as the available range of speed did not suffice to demonstrate this. However, based on robustness to unknown disturbances such as touching the boundary (Sec. 6.3.5) and changes to the FESS dynamics overall (stabilising the updated setup by [22] using gains defined in [20]), the nature of the PID allowed it to perform better.

7.6 Application of MPC

The application of an MPC was investigated only in simulation, as various shortcomings were realised during the programming of the LQG in LabVIEW. Given that it was found that memory was already quite limited when implementing the LQG, the more demanding quadratic programming problems to solve the optimisation problem at every step using high-order matrices would not have made a suitable control approach in real time using the implementation provided in Sec. 4.2. Thus, given that a better method of implementation was not found within the limited time, only an elementary simulation of its working was presented.

The MPC shown was able to settle the set initial conditions faster than both the PID and LQG (Sec. 6.4). However, the control effort required was much higher and the oscillation found was very peculiar. The pattern in the current was a result of the MPC determining it as optimal within the set depth, to have the input voltages to vary only between the constrained maximum and minimum.

More importantly, however, it was shown that the implemented MPC was able to stabilise the system. Optimising the tuning of the controller to reach a certain performance was not pursued, as the control method was not implemented in practice. Additionally, simulations for a time interval of 0.02 s took more than 20 min to complete, which made the tuning of the weights very tedious. As a result, based on this preliminary investigation into the application of MPC for the optimal control of flywheel stability, the application into a real system would require the ability for faster optimal calculations or approximations of optimal calculations [30].

Chapter 8

Conclusion

8.1 Limitations

Throughout this project, various shortcomings were found that hindered the research process. Firstly, the test rig was limited to spinning only up to 2000 rpm, which was much lower than what was anticipated based on the past work done in [21] & [22]. Therefore, it was not possible to make a definitive statement on the performance of the LQG in comparison to the PID, as the results only indicated that the improvement in the stability for the control effort at higher speeds would increase along with the speed.

Secondly, the addition of the LQG controller to the *FPGA_V1* LabVIEW program had already pushed the capabilities of the cRIO its limit. Therefore, the controller had to be formulated into a simplified version, which could not simultaneously track estimated states and sacrificed some precision. As a result, the design of the Kalman filter using the PID-controlled rotor may have adversely affected its performance when implemented in the LQG, despite the separation theorem.

Thirdly, as the test rig was also being used in parallel for other projects, the state of the system was potentially fluctuating. Thus, although the effect on the recorded data was limited by collecting everything in a single session, the reproducibility of the results is not guaranteed if even a slight change occurs to the condition of the test rig.

Fourthly, the test rig in general had some noticeable characteristics which were not included in the mathematical model. In practice, it was noticed how the PMSM introduced a cogging torque and how the test rig would vibrate when spinning at certain speeds. As these dynamics were not taken into account in the model, the performance of the LQG was limited.

Finally, there were significant challenges to programming in LabVIEW with only limited prior knowledge. Through experience, basics in LabVIEW were gathered and ultimately, it was possible to implement a working controller. However, this process was very time-consuming which reduced the

time available for more intricate analysis given the limited time available for the project. A more experienced user might well have been able to optimise the implementations and apply more complex methods.

8.2 Project summary

This thesis has shown how the altered characteristics in the FESS test rig introduced in [22] were taken into account to model some of the new dynamics of the system. Using this model, an LQG controller was reinstated and the application of MPC was investigated in simulation.

Through various tests it was then verified that the implemented LQG performed better than the PID to step disturbances with the rotor stationary and kept the rotor closer to the centre when spinning. Furthermore, within the spin velocity range available, the data indicated that at a greater speed, the rotor would not only be more centred, but that this would be achieved at a lower control effort. The LQG implementation, however, was not able to stabilise the rotor initially without the PID bringing it to the vertical position first. Also, an impulse that would bring the rotor to touch the boundary would also destabilise the rotor. In addition, the viability of an input-constrained MPC was investigated and stability was achieved in simulation. However, a real system implementation would require a more elaborate formulation which is able to calculate the inputs to the system in a more efficient manner.

For the speed range available, the indication of lower control effort for better stabilisation at potentially higher spin velocity using the LQG proves the viability of optimal methods for rotor stabilisation in FESSs. As a result, less energy would be wasted on maintaining the stability of the flywheel, thus energy would be stored more efficiently in the system.

8.3 Future work

For future work on the same test rig, the replacement of the VFD would be necessary to test the stabilisation capabilities of the LQG at high speed. This may require an investigation into the application of gain scheduling to compensate for the changing gyroscopic effects with the speed of the rotor. Additionally, the inclusion of any unmodelled dynamics into the system model and the effect on the LQG performance would be interesting to observe. Dynamics that were not included in the model used in this project included, but were not limited to, the vibration characteristics of the test rig, effects from the PMSM, temperature effects and the unbalanced mass of the shaft. Further investigations should also be made experimentally on the viability of MPC methods in the application for the stabilising control of the rotor.

For the test rig in general, based on the experience gathered, some suggestions to its construction can be made that would enhance its capabilities. Firstly, more effective cooling measures should be implemented for the PMSM to reduce the effect of the increasing temperature on the system. Secondly, the suspension of the rotor in a vacuum environment would greatly reduce any remaining frictional losses, improving its energy storage capabilities. Thirdly, the construction of the FESS should be assessed for its practical limitations and, lastly, the advantages of a hollow cylinder flywheel [31] should be investigated.

Bibliography

- [1] Turgut M Gür. Review of Electrical Energy Storage Technologies, Materials and Systems: Challenges and Prospects for Large-Scale Grid Storage. *Energy & Environmental Science*, 11(10):2696–2767, 2018.
- [2] Hussein Ibrahim, Adrian Ilinca, and Jean Perron. Energy Storage Systems — Characteristics and Comparisons. *Renewable and Sustainable Energy Reviews*, 12(5):1221–1250, 2008.
- [3] Haisheng Chen, Thang Ngoc Cong, Wei Yang, Chunqing Tan, Yongliang Li, and Yulong Ding. Progress in electrical energy storage system: A critical review. *Progress in Natural Science*, 19(3):291–312, 2009.
- [4] Mustafa E Amiryar and Keith R Pullen. A Review of Flywheel Energy Storage System Technologies and their Applications. *Applied Sciences*, 7(3):286, 2017.
- [5] Valentine Roux and Pierre de Miroshedji. Revisiting the History of the Potter’s Wheel in the Southern Levant. *Levant*, 41(2):155–173, 2009.
- [6] Haichang Liu and Jihai Jiang. Flywheel Energy Storage — An Upswing Technology for Energy Sustainability. *Energy and Buildings*, 39(5):599–604, 2007.
- [7] Rion Takahashi and Junji Tamura. Frequency Stabilization of Small Power System with Wind Farm by Using Flywheel Energy Storage System. In *2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, pages 393–398. IEEE, 2007.
- [8] Ning Lu, Mark R Weimar, Yuri V Makarov, FJ Rudolph, SN Murthy, Jim Arseneaux, and Clyde Loutan. Evaluation of the Flywheel Potential for Providing Regulation Service in California. In *IEEE PES General Meeting*, pages 1–6. IEEE, 2010.
- [9] Markus Ahrens, Ladislav Kucera, and Rene Larssonneur. Performance of a Magnetically Suspended Flywheel Energy Storage Device. *IEEE Transactions on Control Systems Technology*, 4(5):494–502, 1996.

- [10] KY Zhu, Y Xiao, and Acharya U Rajendra. Optimal Control of the Magnetic Bearings for a Flywheel Energy Storage System. *Mechatronics*, 19(8):1221–1235, 2009.
- [11] B Polajžer, J Ritonja, G Štumberger, D Dolinar, and J-P Lecointe. Decentralized PI/PD Position Control for Active Magnetic Bearings. *Electrical Engineering*, 89(1):53–59, 2006.
- [12] Arunvel Kailasan, Tim Dimond, Paul Allaire, and David Sheffler. Design and Analysis of a Unique Energy Storage Flywheel System — An Integrated Flywheel, Motor/Generator, and Magnetic Bearing Configuration. *Journal of Engineering for Gas Turbines and Power*, 137(4), 2015.
- [13] Y Okada, B Nagai, and T Shimane. Cross-Feedback Stabilization of the Digitally Controlled Magnetic Bearing. *Journal of Vibration and Acoustics*, 114(1):54–59, 01 1992.
- [14] Elbert Hendricks, Ole Jannerup, and Paul Haase Sørensen. *Linear systems control: deterministic and stochastic methods*. Springer, 2008.
- [15] Kenneth R Muske and James B Rawlings. Model Predictive Control with Linear Models. *AIChE Journal*, 39(2):262–287, 1993.
- [16] Jacob Mattingley, Yang Wang, and Stephen Boyd. Receding Horizon Control. *IEEE Control Systems Magazine*, 31(3):52–65, 2011.
- [17] Jr-Yi Shen and Brian C Fabien. Optimal Control of a Flywheel Energy Storage System with a Radial Flux Hybrid Magnetic Bearing. *Journal of the Franklin Institute*, 339(2):189–210, 2002.
- [18] Jun Hou, Jing Sun, and Heath Hofmann. Control Development and Performance Evaluation for Battery/Flywheel Hybrid Energy Storage Solutions to Mitigate Load Fluctuations in All-Electric Ship Propulsion Systems. *Applied Energy*, 212:919–930, 2018.
- [19] Mohammad Ghanaatian and Saeed Lotfifard. Control of Flywheel Energy Storage Systems in the Presence of Uncertainties. *IEEE Transactions on Sustainable Energy*, 10(1):36–45, 2018.
- [20] Nikolaj A. Dagnæs-Hansen. *Magnetic Bearings for Offshore Flywheel Energy Storage Systems*. PhD thesis, Technical University of Denmark, 2018.
- [21] Josephine Skov Villefrance. *Minimization of Energy Consumption in Active Magnetic Bearings of Flywheel Energy Storage Systems based on Multiphysics Modeling – Theory & Experiment*. Master’s thesis, Technical University of Denmark, 2021.

- [22] Alessio Dallabona. *Design and Implementation of an electrical Drive Train for a Flywheel Energy Storage System*. Master's thesis, Technical University of Denmark, 2022.
- [23] European Environment Agency. Energy Prosumers in Europe: Citizen Participation in the Energy Transition. *EEA Report*, November 2022.
- [24] Gerhard Schweitzer, Eric H Maslen, et al. *Magnetic Bearings*. Springer, 2009.
- [25] L Urankar. Vector Potential and Magnetic Field of Current-Carrying Finite Arc Segment in Analytical Form, Part II: Thin Sheet Approximation. *IEEE Transactions on Magnetics*, 18(3):911–917, 1982.
- [26] Kendall L Su. *Analog Filters*. Kluwer Academic Publishers, 2003.
- [27] Kendall Atkinson. *An Introduction to Numerical Analysis*. Wiley, 2nd edition, 1989. ISBN 0471624896; 9780471624899.
- [28] *Single Channel Adjustable DC Power Supply*. TENMA, March 2019.
- [29] Touzhu Li and Gordon Slemon. Reduction of Cogging Torque in Permanent Magnet Motors. *IEEE Transactions on magnetics*, 24(6):2901–2903, 1988.
- [30] Yuning Jiang, Juraj Oravec, Boris Houska, and Michal Kvasnica. Parallel MPC for Linear Systems with Input Constraints. *IEEE Transactions on Automatic Control*, 66(7):3401–3408, 2020.
- [31] Lukas Quurck, Michael Richter, Maximilian Schneider, Daniel Franz, and S Rinderknecht. Design and Practical Realization of an Innovative Flywheel Concept for Industrial Applications. *Technische Mechanik – European Journal of Engineering Mechanics*, 37(2-5):151–160, 2017.

Appendix A

Input parameters

This appendix contains the input parameters used to obtain the results shown in this thesis. Tab. A.1 presents the values used for the calculation of the linearised PMB forces (Sec. 3.1.4). Tab. A.2 displays the values used to simulate the dynamics of the FESS based on measurements made in [20] & [22] (Sec. 3.2.1). Tab. A.3 lists the weights placed in the diagonal matrices based on the respective state or input on which it was applied (Sec. 5.4.2).

Table A.1: Parameters used for the calculation of the linearised PMB forces.

	Description	Value	Unit
z'	Heights for PMB stator	[−1.6, 0, 1.6]	mm
r'	Radial lengths for PMB stator	[20, 23, 26]	mm
M	Magnetic moment	883.31	kA m ^{−1}
h	Height of the PMB stator magnet	1.6	mm
N_z	Number of heights for PMB rotor	3	
θ_{n_α}	Angle at given n_α	[0 : $\pi/3$: 2π] + δ	rad
l_{n_r}	Length of wiring in section	[20.9, 24.1, 27.2]	mm

Table A.2: Input parameters used for modelling the dynamics of the FESS.

	Description	Value	Unit
$l_{a1,2}$	Centre of mass to AMB planes	101.9	mm
l_p	Centre of mass to PMB plane	212.9	mm
$l_{s1,2}$	Centre of mass to sensor planes	133.9	mm
$I_{xx,yy}$	Moment of inertia along x and y axis	56.146	g m^2
I_{zz}	Moment of inertia along the z -axis	1.378	g m^2
m	Mass of flywheel rotor	4.82	kg
μ_0	Permeability of free space	1.27e-6	H m^{-1}
n	Number of windings around magnet	350	
s_0	Initial air gap	0.45	mm
i_0	Bias current	0.6	A
A_a	Cross-sectional area of magnet onto rotor	3.64e-4	m^2
R	Resistance in wiring	4.5	Ω
k_{pa}	Linearised stiffness by PMB axially	-2.7091e4	N m^{-1}
k_{pr}	Linearised stiffness by PMB radially	3.7496e4	N m^{-1}
t_s	Sampling time	50	μs

Table A.3: LQR weights for the tuned LQG controller placed in corresponding diagonal matrices.

		Value	Unit
States	u_{1-4}	150	V
	$e_{p,1-4}$	10	μm
	$e_{i,1-4}$	2.5	μm
	x	100	μm
	\dot{x}	0.1	$\mu\text{m s}^{-1}$
	y	100	μm
	\dot{y}	0.1	$\mu\text{m s}^{-1}$
	ϕ	5e-4	rad
	$\dot{\phi}$	0.1	rad s^{-1}
	ψ	5e-4	rad
	$\dot{\psi}$	0.1	rad s^{-1}
	i_{a1x}	0.3	A
	i_{a1y}	0.3	A
	i_{a2x}	0.3	A
	i_{a2y}	0.3	A
	$x_{f,s1}$	0.7	$\mu\text{m s}$
	$\dot{x}_{f,s1}$	100	μm
	$y_{f,s1}$	0.7	$\mu\text{m s}$
	$\dot{y}_{f,s1}$	100	μm
	$x_{f,s2}$	0.7	$\mu\text{m s}$
	$\dot{x}_{f,s2}$	100	μm
	$y_{f,s2}$	0.7	$\mu\text{m s}$
	$\dot{y}_{f,s2}$	100	μm
	$i_{f,a1x}$	0.1	A s
	$i_{f,a1x}$	0.3	A
	$i_{f,a1y}$	0.1	A s
	$i_{f,a1y}$	0.3	A
	$i_{f,a2x}$	0.1	A s
	$i_{f,a2x}$	0.3	A
	$i_{f,a2y}$	0.1	A s
	$i_{f,a2y}$	0.3	A
Input	$i_{\text{ref},a1x}$	0.05^{-2}	A
	$i_{\text{ref},a1y}$	0.05^{-2}	A
	$i_{\text{ref},a2x}$	0.031^{-2}	A
	$i_{\text{ref},a2y}$	0.031^{-2}	A
Integral	$x_{i,xs1}$	1	μm
	$x_{i,ys1}$	1	μm
	$x_{i,xs2}$	1	μm
	$x_{i,ys2}$	1	μm