María Beneyto Gómez-Polo

Classical Control Design Theory Applied to Mitigate Rotor-Blade Vibrations

Master Thesis, 24th February 2019

Classical Control Design Theory Applied to Mitigate Rotor-Blade Vibrations, Master Thesis

Author(s):

María Beneyto Gómez-Polo

Supervisor(s):

Ilmar Ferreira Santos, Professor Dr.-Ing. Dr. Tech., MEK, DTU

Release date:	24th February 2019
Class:	Public
Comments:	This report is a part of the requirements to achieve the MSc in Electrical Engineering at Technical University of Denmark. The report represents 30 ECTS points.
Rights:	María Beneyto Gómez-Polo, 2019

PREFACE

This thesis is the final written submission of the Electrical M.Sc. programme at the Technical University of Denmark, it has however been carried out in conjunction with the DTU Mechanical department.

The following thesis deals with the study, design and implementation of classical control strategies in order to perform active control of vibrations in rotor-blade structures. The control will be designed for a specific rotor-blade test-rig arrangement that was lent by the DTU Mechanical Department. In order to build the mathematical model of the plant on Matlab, previous information from a Ph.D thesis written by R.H Christensen [1] will be used, he was able to obtain a theoretical and numerical description of the movement and behaviour of a four blade rotor-blade structure.

The work here presented, is part of a bigger project carried out throughout the semester on the test-rig. I have been working closely with two other DTU students who also had to study and address different issues regarding the test-rig, Alvaro Brández (s171801) and Ignacio Escudero (s171798). Throughout his thesis Alvaro had to physically update the test-rig's software (control interface) and hardware (sensors and actuators) to obtain the correct functioning that allows for the possibility of actively controlling vibrations. My thesis deals with the mathematical theoretical modelling of the test-rig plant and the design of classical controllers to dampen vibrations. While Ignacio's thesis deals with the numerical identification of the test-rig plant and the design of an adaptive controller to dampen vibrations. The whole project was therefore designed with the objective of experimentally reducing vibrations in a four blade rotor-blade test-rig using both classical and more refined control strategies. This whole project was carried out with the encouragement of our supervisor Ilmar Santos.

ABSTRACT

This paper studies the design and implementation of active control strategies to reduce vibrations in rotor-blade systems. Rotating machinery are present in mostly every industrial application, some examples include, compressors, turbines, pumps or motors. The fast development of engineering applications has lead to the demand of highly efficient and fast rotating machinery. Vibrations occur naturally in rotating components, they introduce various issues, like working performance degradation and machinery lifetime reduction.

This project will study the application of control strategies to attenuate both hub and blade vibrations in a four-blade rotor system. This thesis was devised as a theoretically driven investigation of the design of classical control techniques. However, it has been carried out with the fundamental idea of ensuring the applicability of the designed control algorithms in a physical four-blade rotor test-rig. Previous studies carried out on the test-rig investigated the design and application of complex time-varying control structures. The main disadvantage of these solutions was the difficult application of the control algorithm in the test-rig control unit, due to heavy computation. This thesis therefore studies the design of simpler controllers with a suboptimal solution but with still good performance at vibrations reduction.

The first part of the thesis will deal with the obtention of an accurate mathematical model of the four-blade rotor test-rig, that is able to explain the dynamic behaviour of the system. The second part will deal with the design of classical controllers for active vibration control in the frozen version model, with no rotation. A simple structure of decoupled SISO PID controllers will designed. The advantage of the PID controller is its simple design and easy tuning. State space control structure will also be used to study the model as a MIMO system. A full-state feedback approach will be used to design the controllers. The controller gains are found using two methods. First one being pole placement assignment, where the gain is chosen to obtain a desired close loop behaviour. The other method is optimal control design with linear quadratic regulation. All of the controller designs will be tested on the system, analyzed and compared. These controllers will be designed both for shaft and individual blade actuation as well as for shaft based actuation control only, the possibility of reducing blade vibrations with shaft control only will be studied.

The last part of the thesis deals with the study of the adaptation of designed static controllers to the rotating model of the system using classical control methods. A feedforward control will be added to the PID control structure to cancel the effect of disturbances that appear during rotation.

SYMBOLS

$A(t)$	Model state space system matrix
A_0	Static Model state space system matrix
$B(t)$	Model state space input matrix
B_0	Static Model state space input matrix
$B_v(t)$	Model state space Disturbance input matrix
B_{v0}	Static Model state space Disturbance input matrix
B_i	Auxiliary rotating frame of blade i
C	Model state space output matrix
C_i	Auxiliary rotating frame of tip mass i
$d_{1,2,3,4}$	Blades measured displacement
D_{hx}	Damping of hub suspension in x-direction
D_{hy}	Damping of hub suspension in y-direction
E_{bi}	Youngs modulus of blade i
$E_{kin,(h,r,b,t)}$	Kinetic energy of hub, rotor, blade and blade tip respectively
$E_{pot,(h,b,c)}$	Potential energy of hub, blade and centrifugal forces respectively
f_{int}	Known disturbances due to internal forces in Newtons
f_{ext}	Input actuation forces in Newtons
$Fh_{x,y}$	Actuation force applied in hub's x and y direction (N)
F_i	Generalized force in the i'th degree of freedom in Lagrange Equation
F_0	Static Discrete Model state space system matrix
G_0	Static Discrete Model state space input matrix
G_{v0}	Static Discrete Model state space disturbance input matrix
I	Inertial frame of reference
I_{bi}	Area moment of inertia of blade i
J_{ti}	Mass moment of inertia of tip mass i
K_{hx}	Stiffness of hub suspension in x-direction
K_{hy}	Stiffness of hub suspension in y-direction
L_{bi}	Length of blade i
L_{ti}	Length of tip mass i
m	Number of mode shapes included in the mode shape expansion
m_r	Mass of rotor
m_{hx}	Mass of hub in x-direction
m_{hy}	Mass of hub in y-direction
m_{ti}	Mass moment of tip mass i
N	Number of degrees of freedom in the mechanical part of the model

N_a	Number actuators
N_s	Number sensors
$N_{bi}(x)$	Normal force in blade i as a function of position x
$q_{i,j}(t)$	Generalized coordinate of blade i, mode j
R_r	Rotor Radius
$R1$	LQR State weighting matrix
$R2$	LQR Actuation weighting matrix
$r^I_{o-h}(t)$	Position vector of center of rotation with respect to the inertial frame
$r^{B_i}_{h-p_i}(t, x_i)$	Position vector of point p_i on blade i with respect to the center of rotation
$r^{C_i}_{o_i t_i}$	Position of tip mass center i with respect to the frame C_i
T_a	Actuators transformation matrix
T_s	Sensors transformation matrix
T	Sampling time
T_{B_i}	Transformation matrix from B_i to I
T_{C_i}	Transformation matrix from C_i to B_i
$u(t)$	Electromagnets actuation input signal in Volts
$v^I_{o-t_i}(t)$	Absolute linear velocity of tip mass i with respect to the center of the inertial frame
$v^{B_i}_{h-p_i}(t, x_i)$	Absolute velocity of point p_i with respect to the center of rotation
$v^I_{h-p_i}(t, x_i)$	Absolute linear velocity of point p_i with respect to the center of rotation
$v^I_{o-p_i}(t, x_i)$	Absolute linear velocity of point p_i with respect to the center of the inertial frame
x_h	Hub output displacement in x direction
x	State vector
$\hat{x}$	Estimated state vector
y	Measured output vector
y_h	Hub output displacement in y direction
$z(t)$	Generalized coordinate of mechanical model
Ω	Rotor Angular velocity
$\dot{\Omega}$	Rotor Angular acceleration
α_i	Angle offset of blade i with respect to blade 1
ϵ	Displacement of rotor centre of gravity
κ	Angle of rotor unbalance with respect to blade 1
$\phi_{i,j}(x)$	Mode shape of blade i, mode j
ρ_{bi}	Density of blade i
θ	Angular position rotor in rad
$\omega^I_{B_i}$	Angular velocity of frame B_i with respect to the inertial frame I
$\omega^I_{C_i}$	Absolute velocity of frame C_i with respect to the inertial frame I

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This paper studies the design and implementation of active control strategies used to reduce vibrations in industrial rotor-blade systems with the objective of extending their lifetime. Rotating machinery, like compressors, turbines, pumps or motors are nowadays used in a variety of industrial applications, these may include the power, process or oil industries. Vibration naturally occurs in rotating machinery, however the incorrect suppression of these vibrations can lead to serious physical problems, it accelerates the rate of wear of mechanical components, creates uncomfortable noise, leads to the degradation of the machine's working conditions and performance and more importantly can cause safety problems [2].

The fast development of engineering applications has lead to the demand of highly efficient and fast rotating machinery. An example of this is the robotic industry which needs lightweight, versatile, energy efficient machine components with high performance accuracy. Light components usually come with the addition of higher vibrations that need to be addressed. These rotating components are usually really expensive, therefore degradation of its functionality due to excessive vibration is inadmissible.

In the past, passive vibration control has been commonly used as a method to reduce machine vibrations. This strategy has proven to be a very effective and cheap option when reducing vibrations in a given working range. It however comes with disadvantages which may not be practical for the high demanding industry. Passive vibration strategies have poor low frequency performance, they also have a very narrow effective frequency suppression range, which makes them poorly adjustable. They are not capable of adapting to fit different working conditions, changing disturbances or to adapt to structural changes caused by natural wear of components.

Active vibration control strategies are being introduced as viable way to fulfill the deficiencies of passive strategies, including the low frequency gap and the lack of flexibility and adaptability. Moreover, they do not add excessive weight to the structure.

1.1 Active Vibration Control Background

Active vibration control strategies function by applying an artificial vibration input to a structure in order to modify its response in a desired manner. To reduce vibrations and dampen the response of the structure, an artificial cancelling vibration source must be generated, this source will destructively interfere with the naturally occurring vibration and will therefore lead to a reduction in the level of vibrations at a localized area in the structure [4].

Every active vibration control system is composed of three main subsystems, the plant,

actuators and sensors, and the designed controller.
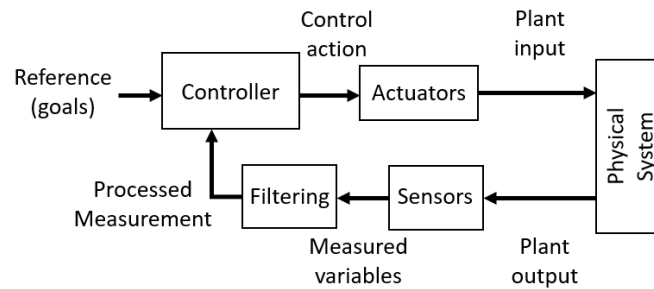
Figure 1.1: Active control of vibrations scheme.

The plant is a mathematical representation of the physical system to be controlled. An approximate model of the real system has to be derived, this model must explain with accuracy the real behaviour of the system, however most systems are complex and difficult to model, continuous flexible structures have infinite order vibration characteristics. In order to reduce computational complexity a model order reduction is usually performed, this also allows for the easier design and implementation of the controller [5]. However, model errors or poor accuracy can lead to spill-over problems. Spill-over issues appear when unmodelled (residual) dynamics of the system become unstable when adding the control loop. To avoid this issue, a suitable reduced model of the system has to be obtained, this model needs to correctly explain the behaviour of the system around the desired frequency bandwidth.

Sensors are needed to measure the system behaviour and response. Actuators provide the input needed to modify the plant's response in a desired manner. The location of sensors and actuators is a very important decision, as it will affect the design and modelling of the controller and its efficiency. The location of sensors and actuators affects the controllability and observability of the plant, two very important aspects in classical control theory that need to be studied before designing the controller. Controllability studies the possibility of physically taking the system to a desired state by applying a control input in the actuators. Observability studies whether the system states needed to describe the behaviour of the system can be observed or known from the sensed signals.

The controller implements the feedback control algorithm needed to obtain the desired behaviour of the real system. In classical control theory the real system paired with the controller is known as the closed-loop system. In the closed-loop system the open-loop system dynamics are combined with the controller dynamics, modifying its overall behaviour. The control mechanism must be able to cope with the frequency dependent changes on the characteristics of the system [3]. Time varying model structures are common in flexible vibration structures, controllers must therefore be complex and robust to adapt to these changes.

1.2   Active Vibration Control in Rotor-Blade Systems

This thesis focuses in the active control of vibrations in rotating rotor-blade systems, more specifically in a four blade rotor-blade test-rig.

Rotor-blade systems are present in most industrial processes, they have numerous applications, these may include, steam turbines, gas turbines, generators, motors, pumps, compressors, aircraft engines, fans, or drive trains. Common to all rotating machinery is blade vibration during operation, this mainly happens due to unavoidable unbalances in the rotor, misalignment along the rotor shaft, wear throughout the years and looseness [7]. Vibrations will propagate between the rotor and blades due to the mechanical coupling present between both parts. Vibrations are undesirable, they can damage the equipment and reduce efficiency. Rotating machines are expensive, therefore, failure due to vibrations is inadmissible.

In rotating rotor-blade systems, the system's characteristics vary periodically with time [8], complex dynamic dependent changes in the system's characteristics like centrifugal stiffening occur, where an increase in the blade's natural frequencies with the rotation speed can be seen [12]. Classical techniques for model analysis and controller design and implementation are usually inadequate for time varying systems. Controller design for generic time varying systems is challenging. Even though it has theoretically been approached in various ways, experimental application is however limited due to the great computational load these methods require. Studies assessing the performance of time varying controlled systems acknowledge the limited experimental application of the designed time varying controller due to tedious amounts of complex mathematical operations and numerical computations which can be very difficult to apply in complex model structures [9].

However, for the specific subgroup of periodic time-varying systems, many research has already been carried out focusing on the idea of using the periodicity of the system's behaviour to reduce the complexity of the time-varying model and controller design. This field of study has gained lots of attention during the past years due to its various applications in the engineering industry, the efficient and reliable model and control of periodic systems can be applied to flexible systems, rotating machinery or helicopter blades. Nevertheless, the experimental application of these methods has not been fully researched yet.

Research on the mathematical modelling of a rotating four rotor-blade system with periodic motion has already been carried out, models have been obtained for different physical constrains and motion assumptions. In [8] a model is obtained for a rotor-blade system with only allowed lateral horizontal displacement in the rotor foundation, here the blades are modelled as Euler–Bernoulli beams, by using different approaches they obtain a linear beam model, a linear beam model with second order terms and a non-linear one. The article concludes with the analysis of the different models, being the non-linear the most accurate one at describing the motion of the blade. In the Ph.D thesis by R.H. Christensen [1] a mathematical model describing the motion of a rigid rotor mounted in a hub, which can perform planar movement in the x-y plane, with four attached flexible blades with tip point masses is described. Throughout the thesis a modal periodic decomposition is

also performed, this method decouples the system dynamics, it allows to easily reduce the model complexity by separating the dominant dynamics from the redundant ones. The mathematical model derived in this thesis will be used as a basis for the mathematical plant that will be used in this thesis, all controllers will be designed for this plant.

Active control of vibrations in flexible beams, rotating flexible beams and rotating rotor-blade systems has been widely studied, usually with a theoretical approach. In [13] a feedback control strategy for a flexible mechanical systems is obtained. The general equation of motion with damping is used to describe the model. A modal feedback control is used, the control of spillover due to the residual modes is also treated. In [14] an electromagnetic damper control is designed for a one degree of freedom mechanical flexible structure. A state space full-state feedback control is designed using pole placement assignment. In [6] a reduced modal model of a beam using modal filters is obtained and a feedback control loop is designed using optimal control strategies. Moreover, a feedforward control loop is added to cancel unknown disturbances in the system. In [15] a rotating beam is modeled using finite element analysis and state space representation, a state estimator and an active optimal controller are designed. However, all of these studies deal with linear time-invariant systems and classical methods for controller design, they neglect some of the dynamics of the real system and simplify the model. Throughout the past decade, more and more articles deal with the periodic or time varying active control of rotor-blade systems. In [3] an adaptive active control system is designed to suppress vibrations in a flexible beam using two methods recursive least squares optimization and genetic algorithm optimization. In [16] a robust controller using gain-scheduled periodic control design is used to reduce helicopter's rotor vibrations. Gain-scheduled control is performed by designing a controller whose gains are adjusted as a function of a scheduling variable that describes the current operating point. The helicopter rotor is modeled as a time varying system affected by a known periodic disturbance that accounts for the periodic nature of the vibrations that appear on a rotating rotor.

Previous students from DTU Mechanical Department have aslo studied the active control of vibrations in four blades rotor-blade systems. C.S. Jakobsen [17] studied the design of a gain-scheduled periodic controller, the different controller gains were designed off-line using optimal control design for different rotor angle and velocity arrangements, only shaft base actuation was taken into account. J.B. Hansen [18] studied the design of robust periodic control algorithms using H_∞ and H_2 methods which transform the controller design into a mathematical optimization problem which assures stability while maintaining performance, only shaft based actuation was taken into account. Both of these methods require very advanced control algorithms, this makes them very difficult to design for a complex mathematical model like a four blade rotor-blade system with coupled vibrations, it is even more complex to experimentally apply in the rotor-blade test rig due to heavy mathematical computations. In [18] Jesper concludes by stating that the proposed robust control algorithms could not be resolved for the four blade rotor-blade model due to the complexity of the numerical calculations required.

1.3 Project Focus

The objective of this thesis is to study the active control of vibrations in a rotor-blade system using classic control approaches. As stated in the Preface the global aim is to experimentally apply the designed controllers on the test-rig. As seen from the conclusions of C.S. Jakobsen [17] and J.B. Hansen [18], previous Master Theses, advanced control algorithms are too complex to effectively be applied on the test-rig control interface. Therefore a classical approach will be used throughout the thesis, this will allow for the design of simpler controllers that can be experimentally implemented obtaining a suboptimal solution with still good performance at damping vibrations. Classical design methods also have the advantage of being easier to understand meaning that anyone could afterwards tune or adapt the controller for future research.

Throughout the semester another student, Alvaro Brández (s171801), has been working on the test-rig's software and hardware control interface. He was able to update the actuators so that individual blade actuation is also possible. This thesis will therefore also study the impact of only shaft actuation or shaft and blade actuation and how it affects the control of vibrations.

The mathematical model of the plant will be derived using R.H. Christensen's [1] Ph.D. Thesis as a basis. He was able to numerically obtain the equations of motion that describe the test-rig's behaviour. His findings will be used to model the plant needed for the controller design. However some significant changes will be made to fit the model to the current physical test-rig facility. The previous model assumed the controlled input to the system was the force that the actuator magnets directly apply to the physical system's components. However, we cannot measure and control the force the magnets apply to the system, but the voltage applied to the magnets. Therefore the updated model will include this change.

The first part of the thesis will deal with the design of controllers for the frozen model of the rotor-blade system, this means zero angular velocity or acceleration are applied to the rotor. The final part of the thesis will study how rotor rotation affects the modeled system, how the previously designed controllers perform on the rotating system and what changes can be made on the controllers to adjust to rotation.

Various classical control algorithms will be designed. Starting with the most basic one, a PID controller will be introduced, here the system will be treated as the sum of different single input single output uncoupled systems. The advantage of the PID controller is its simple design and easy tuning. State space control structure will also be used to study the plant as a multiple input multiple output system. A full-state feedback approach will be used to design the controllers. The controller gains are found using two methods. First one being pole placement assignment, where the gain is chosen to obtain a desired close loop behaviour, this technique is useful when the designer has an understanding of the system dynamics and wants to obtain a specific behaviour. The other method is optimal control design with linear quadratic regulation, this method allows you to design the controller gains by adjusting the performance and quality of the controller, this way you can obtain the best possible controller taking into account important factors, like input limitations. All of the controller designs will be tested on the system, analyzed and compared.

2 PHYSICAL SYSTEM

This chapter will focus on the description of the test-rig structure I have based my work on. This test-rig was arranged by R.H. Christensen, however some components have been updated throughout this semester by another DTU student, Alvaro Brández who had to change part the test-rig's hardware and software to allow for the possibility of performing active control of vibrations with actuation both in the shaft and individual blades.

2.1 Mechanical Structure

The test-rig was designed to study vibrations in a rotor-blade system. Descriptive photographs of the structure can be seen on page 10, which will be addressed in the following explanation. The whole rotor-blade system is installed inside a static external rigid frame structure (14). The rotating rotor-blade system main components include a rotating shaft (16), a mounted rigid disk, and the attached blades. The shaft is supported at both ends by two bearings, these are at the same time mounted into a flexible suspended cage or hub (1). The shaft and hub move together. The hub has a flexible suspension made out of thin flexible beams which are attached to the external frame. Mounted at one end of the shaft is the rigid disk (2) which holds the blades (3) and blade components. There are four flexible blades with identical tip masses, all blades are attached symmetrically to the disk. The flexible hub was designed with a high stiffness suspension, therefore, it only allows movement of the shaft and rigid disk in the planar x-y direction.

Magnetic actuators have now been updated to actuate both in the shaft and in each individual blade. The external magnets actuate on the flexible suspended hub which supports the shaft, both in the x (4) and y (5) direction. There are individual magnets acting on each blade (6), these magnets are attached to the rigid disk.

Sensors have also been updated to measure displacements in the hub's x and y direction and displacement of each individual blade. Hub sensor's (7) are attached to the external stationary frame and measure x and y displacement of the flexible hub which holds the shaft and moves with it. Individual blade sensors (8) are attached to the rigid disk.

To introduce rotation into the rotor-blade system a DC-motor (9) will be used. A timing belt system with two gears is used to attach the motor to the shaft. The DC-motor is controlled with a voltage power supply, different angular velocities can be obtained. An eddy current sensor is mounted in the static structure to measure the rotor's angular velocity (13), this information will be used in the control interface.
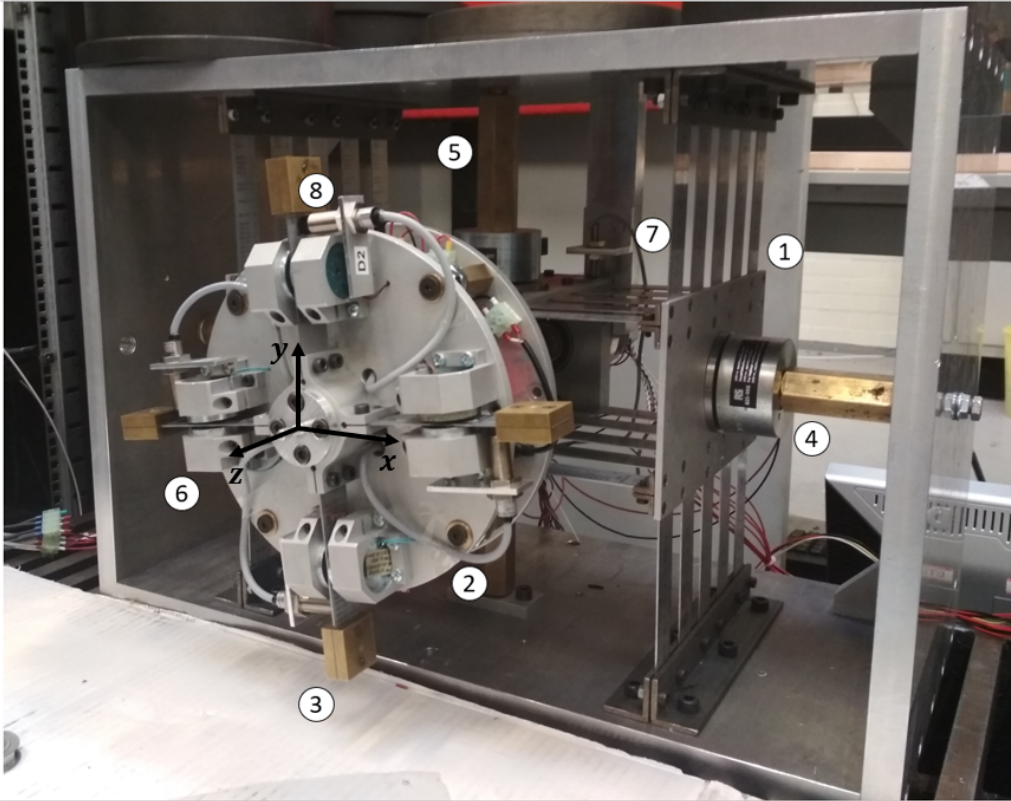
Figure 2.1: Test-rig Description (I).
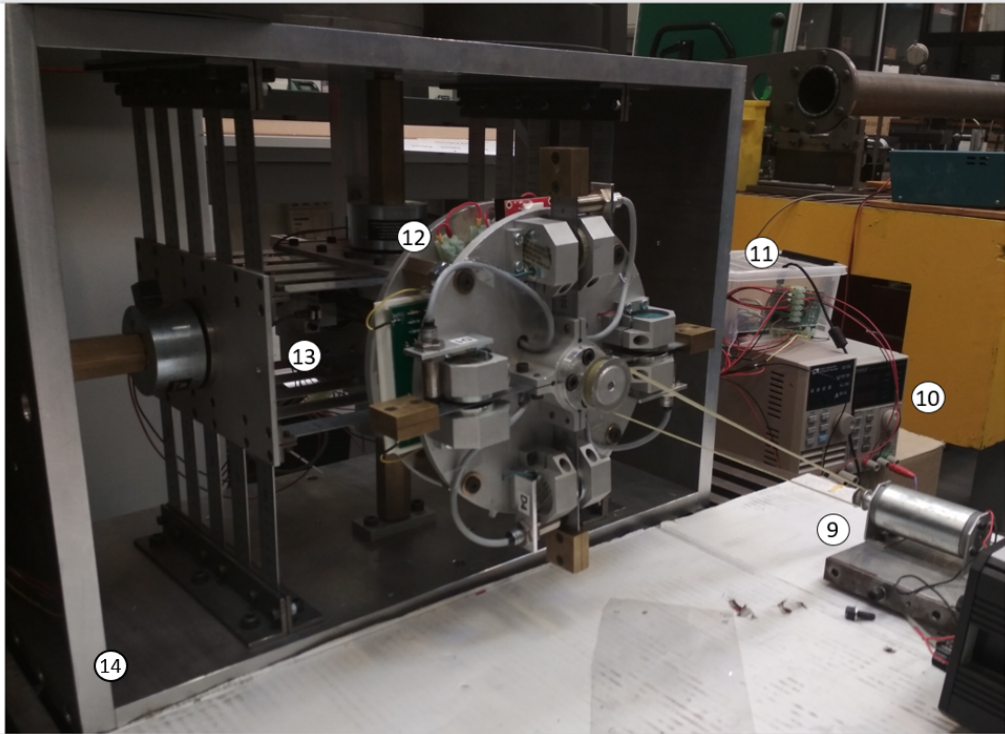
Figure 2.2: Test-rig Description (II).

Micro-controllers will be used for the control interface. The electronic control hardware for the hub (11) and for the blades (12) is powered by an external voltage supply (10). The hub's control electronics are static, as sensors and actuators are only attached to the external stationary structure. However, the blade's actuators and sensors are located in the rigid disk which rotates with the shaft. Therefore, to power the blade control electronics a slip ring (15) is used, it is a electro-mechanical system which allows the transmission of power and electrical signals from a stationary structure to a rotating one. The slip ring is located in the rotor's end opposite to the rigid disk.
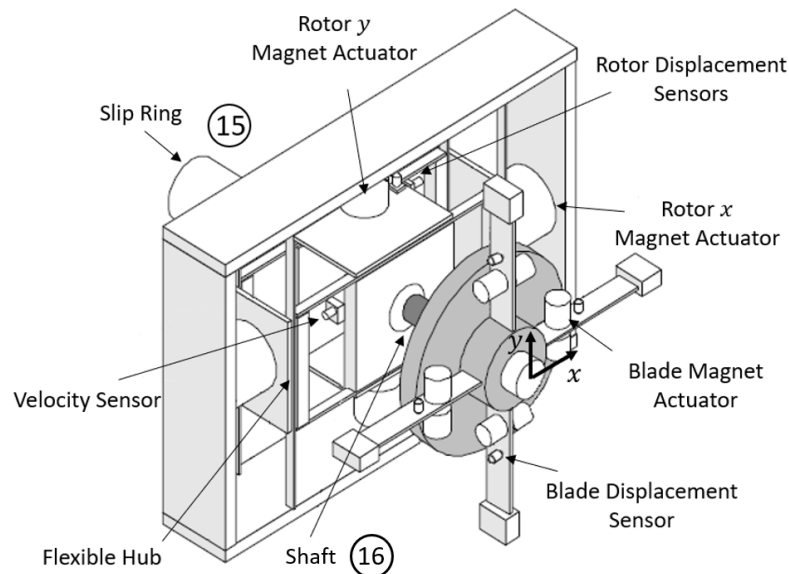
Figure 2.3: Test-rig Diagram.

2.2 Sensors

In the current slip-ring arrangement eddy-current sensors are used to measure the rotor's x and y displacement, as well as to measure individual blade-tip displacement. The sensor model is the Pulsotronic kj4-m12mn50-anu [19]. There are two sensors measuring hub displacement, and one sensor per blade.

These eddy-current sensors have a linear operation range of 4 mm, this is the range we will be working on as the blades are physically not allowed to move more than this distance and the hub's displacement amplitude is small due to the hub's suspension high stiffness. Assuming a linear measurement range will afterwards make controller design easier. Each sensor is located at a specific distance from the blade-tip and from the hub, they have been calibrated so that they output an analog signal of 4.07 V when they are at static position. This voltage will vary from 0.3 V to 8.7 V as the system vibrates in the allowed working range. This signal is afterwards conditioned and amplified to a voltage range of 0.15 V to 4.5 V and digitized with an ADC to make it suitable for the micro-processor acquisition system. The sensibility of the measurement system is of 0.3750 mm/V in the linear operation range, this value will be needed in the software control algorithm to convert from input voltage to real distance.

There is an extra eddy-current sensor installed perpendicular to the shaft, at the position where the sensor is placed the shaft has an indentation mark. This sensor is used to measure rotor's angular velocity, it will produce a pulse whenever it detects the mark, by measuring the time between pulses the angular velocity can be obtained. With knowledge about the real time data of the system one can also calculate the position of the rotor system.

2.3 Actuators

There are six pairs of electromagnetic actuators installed in the test-rig, they will be used to attenuate vibrations in rotor-blade system. An electromagnet can only produce an attracting force, therefore, to produce forces in both directions, a pair of electromagnets is installed at each point of actuation. There is a pair of magnets acting on each blade, and two pairs acting on the hub, in x and y direction.

The blade electro-magnets are $\phi32$ mm holding magnets, while the hub magnets are $\phi65$ mm holding magnets, both models have a maximum voltage input of 20 V DC. Electrical data of the magnets can be seen in the table underneath. The magnet characterization was carried out during this semester by another student, Alvaro Brández, he determined that after a 50 Hz sinusoidal input, their power reduces significantly.

Magnet Data		
	$\phi32$ mm	$\phi65$ mm
Resistance (Ohms)	170	40
Inductance (mH)	3	1.5

Table 2.1: Electro-magnet specifications

The force produced by an electromagnet is non-linear, it varies with the control current going through the coil of the magnet squared and with the air gap between magnet and controlled surface squared. However in our system, due to the really the small amplitude of vibrations the magnet actuation system can be approximated to a linear model, this will be however explained in the chapter Mathematical Model of the System on page 14.

The current test-rig uses power drivers to actuate the electromagnets, they are able to regulate the power supplied to each magnet. There are two micro-processors that control these drivers, one used for blade vibrations control, the second one for hub control. They generate the PWM (Pulse-width Modulation) control signals that modulate the amount of power or voltage the drivers supply to each magnet. All drivers are externally powered with +20V, meaning they can output up to 20 V. The PWM control signal generated by the micro-processors is really high in frequency, therefore the driver's output voltage signal will be seen as a time-varying DC signal actuating on each magnet. To change the effective voltage input to a magnet, the duty cycle of the PWM signal that controls the driver of the given magnet must be changed. The duty cycle is the fraction of one period in which the signal is active, at 100% of duty cycle the driver will output the maximum, 20V. The duty cycle percentage will be used in the control algorithm to change the supplied voltage to the magnets, 1%-100% in duty cycle translates to 0V-20V.

2.4 Control Unit

The previous test-rig used dSpace 2004 as software to implement the control algorithm and DSPACE DS1103 PPC controller board to execute the control action. The control board has 16 A/D converter input channels and 8 D/A converter output channels. The maximum input and output voltage is of ± 10 V. Due to the limited amount of output power (10V) and the insufficient amount of D/A channels and slip-ring channels that are needed for a complete control of the whole system actuators and sensors, the control interface has been changed.

The current test-rig uses a distributed control system. There is a centralized processor, the computer, and two autonomous micro-processors, Raspberry Pi 3B+ [20], one for the hub vibration control, the second one for the blades vibration control . A simplified scheme of the system can be seen on figure 2.4 underneath, the scheme only shows the interaction of the system with one micro-processor, the other one carries out the same function. The blades control micro-processor is installed in the rigid disk and therefore must be powered through the slip ring.
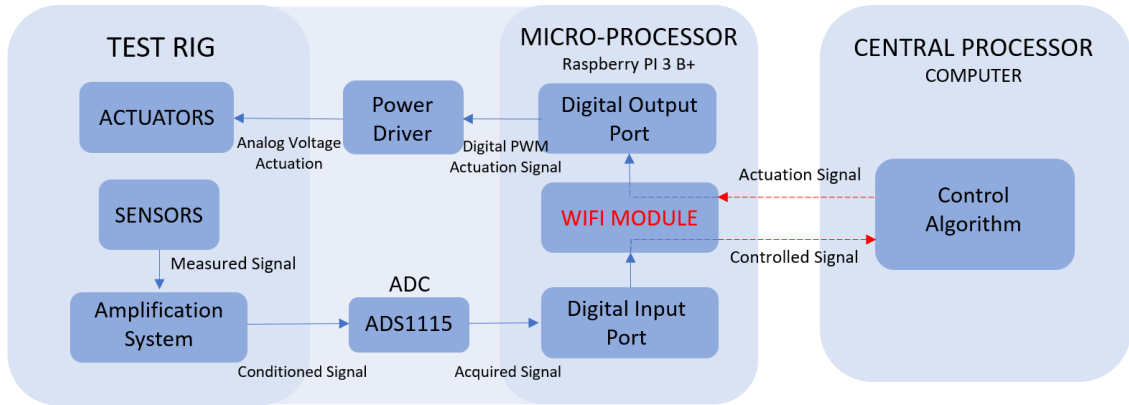
Figure 2.4: Control Unit Scheme

In the current setup, the micro-processor is the component which physically interacts with the physical plant, it is equipped with a DAQ interface capable of reading and sending digital signals. Its main function is to read the digitized measurement signal coming from the external ADC and to send the digital actuation signal, coming from the central processor, to the drivers that power the actuators. It also contains a WIFI module which interacts with the central processor. By using WIFI connection all of the measurement and actuation signals needed for a full control of the system can be transmitted without the use of the slip-ring.

The central processor carries out all of the computations, it implements the control algorithm. Currently a computer running on Linux is installed, it uses C as programming language. Its main function is to receive and synchronize the measurement signals coming from the two Raspberry Pis, compute the control action and send it back. It is important to coordinate both micro-processors so that they both work at the same sampling rate, they need to measure and actuate in the same time sample for optimal control.

3 MATHEMATICAL MODEL OF THE SYSTEM

As stated before, the mathematical model is derived using R.H. Christensen's [1] Ph.D. Thesis as a basis. The model must explain with accuracy the real behaviour of the system to avoid spill over problems related to unmodelled dynamics, however it still needs to be simple enough so that classical system analysis tools can be implemented.

The model needs to correctly account for the x and y planar vibrations of the rotor hub as well as for the low order dominant vibrations that appear on the blades. It must also explain the rotor-blade coupling dynamics as this is important when designing controllers with rotor actuation only. The model needs to recreate the periodic behaviour of the mechanical system dynamics once the rotor is subjected to rotation. This updated model must also account for the physical changes that have been carried on the test-rig's actuators and sensors. The model will be augmented to include the actuators dynamics.

3.1 Model Assumptions

To avoid the increase of the model's complexity some assumptions regarding the behaviour and movement of the mechanical system where taken into account in R.H. Christensen's Thesis.

As seen in the Physical System chapter on page 9, the rotor system is supported by a flexible suspended cage known as the hub, they move together. The hub's high stiffness allows us to assume there wont be hub foundation's angular movement. We can therefore make some model simplifications.

- The shaf-hub system can only perform lateral movements along the x-y plane.
- The rigid disk mounted on the shaft can also only move in the x-y plane. This reduces disk's gyroscopic effects, we don't need to include them in the modelling.
- Blades are attached non-twisted to the disk. They will therefore only bend in the planar x-y direction. This also suppresses the axial motion coupling between blades and shaft.
- Blades are assumed to be non flexible in their longitudinal direction.
- The hub 's suspension is assumed to be massless. Its high stiffness restricts the hub's movement to a very linear motion in the x and y direction. The hub behaviour can therefore be approximated to a lumped (discrete mass) system with harmonic oscillating motion.

3.2 Mathematical Description of the Test-Rig

The first step in modelling the test-rig is to identify the most important constants, variables and features that characterize the behaviour of the real system. On figure 3.1 underneath a simplified diagram of what the mathematical model must explain can be seen. The numerical values of the system parameters are given on table 3.1 on page 17.
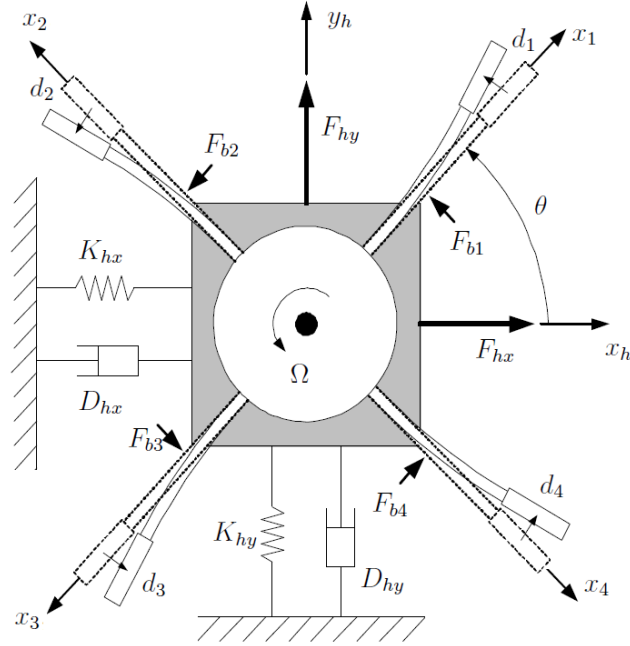
Figure 3.1: Mathematical Description of the System

To accomplish the active control of vibrations the test rig is equipped with six sensors that measure displacement and 6 pairs of electromagnetic actuators to dampen the response of the system. On the figure above this signals can be seen. The sensors measure the local displacement of the four blades tips d_1, d_2, d_3, d_4 as well as the rotor/hub displacement in x and y direction x_h and y_h. There is a pair of electromagnets actuating on each blade F_{b1}, F_{b2}, F_{b3}, F_{b4}, as well as in the hub's x and y direction, F_{hx}, F_{hy}. The blade electromagnets actuate at a distance of 40 mm from the base of the blade. The hub electromagnets are attached to the external static structure and actuate the hub on the x and y direction. Every pair of electromagnets is modeled as a single magnetic force perpendicularly acting on a singular point in the structure, this force can take positive or negative values.

The angular velocity of the rotor will be defined by Ω in rad/s, while θ defines the angular position of the rotor in rad. As the shaft-disk subsystem is rigid, the angular position of each blade will automatically be defined in $(\theta + 0)$, $(\theta + \frac{\pi}{2})$, $(\theta + \pi)$, $(\theta + \frac{3\pi}{2})$.

As explained in the previous chapter, Model Assumptions, the hub suspension will be modeled as a massless linear system with the following descriptive variables, damping coefficient D_{hx} and D_{hy}, and stiffness constant K_{hx} and K_{hy}. This simplification makes the modelling of the hub really easy as its behaviour is explained by the classical mechanics

one degree of freedom harmonic oscillator motion.

However the modelling of the blades is quite complex, mainly because they cannot be considered a lumped discrete system, but a continuous system with an infinite number of degrees of freedom. Figure 3.2 underneath shows the important variables that characterize the blade.
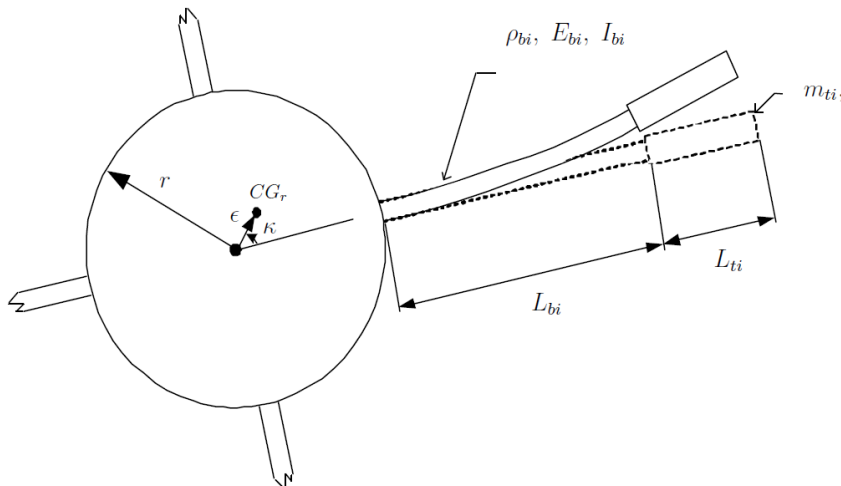
Figure 3.2: Blade's Mathematical Description from R.H. Christensen's [1] Thesis

L_{bi} is the longitudinal length of the flexible blade from its base up to the start of the tip mass. As the blade is a continuous system its parameters are distributed along its length, some of these include, δ_{bi}, E_{bi}, I_{bi}, which name and value are given on table 3.1. The weighted tip attached at the end of each blade is characterized by its length L_{ti} and mass m_{ti} and is considered to be rigid.

It is also important to define the masses of the components in the rotor-blade system. The rotor (rotating part of the system) is composed of the shaft, rigid disk mounted on it, and blades attached to the disk. However, as the blades have a completely different and more complex movement than the rigid shaft-disk system, its masses and mass distribution are defined separately. The shaft and disk are supported by the hub, due to the rigid structure we assume they all move together, they are supported by the hub's suspension system. The inertial mass acting on the hub's suspension is defined for the two allowed directions of motion x and y, named m_{hx} and m_{hy}, which are different due to the asymmetrical construction of the rotor-hub system.

Moreover, the rotor-blade system has a mass unbalance, it was constructed with an eccentricity that displaces the centre of gravity from O to CG_r defined by the distance ϵ and angle k, this eccentricity increases vibrations when rotating.

Rotor-Hub Sub-System Parameters			
Mass	m_{hx}	10.5	$[kg]$
	m_{hy}	8.6	$[kg]$
Stiffness	K_{hx}	$66 \cdot 10^3$	$[N/m]$
	K_{hy}	$77 \cdot 10^3$	$[N/m]$
Damping	D_{hx}	1.2	$[N \cdot s/m]$
	D_{hy}	1.5	$[N \cdot s/m]$
Eccentricity	ϵ	$1 \cdot 10^{-3}$	$[m]$
	k	0	$[rad]$
Rotating Disk Diameter	r	0.04	$[m]$
Rotating Disk Mass	m_r	3.0	$[kg]$

Blade and Tip Parameters			
Blade Length	L_{bi}	$80 \cdot 10^{-3}$	$[m]$
Distributed Mass	ρ_{bi}	0.195	$[kg/m]$
Elasticity	E_{bi}	$2 \cdot 10^{11}$	$[N/m^2]$
Moment of Intertia (Area)	I_{bi}	$2.08 \cdot 10^{-12}$	$[m^4]$
Damping	D_{bi}	0.8	$[N \cdot s/m]$
Tip Mass	m_{ti}	0.109	$[kg]$
Tip Mass Moment of Inertia	J_{ti}	$3.35 \cdot 10^{-5}$	$[kg/m^2]$
Tip Mass Length	L_{ti}	$30 \cdot 10^{-3}$	$[m]$
Blade Angular Location (i=1,2,3,4)	α_i	$(i-1) \cdot \frac{\pi}{2}$	$[rad]$

Table 3.1: Test Rig Mathematical Model Parameters

3.3 Frame of Reference

Now that we have a mathematical model description of the appearance of the test-rig, we need to obtain the equations of motion that describe the real system's behaviour. However, we firstly need to define a coordinate system with reference frames where we can uniquely specify the position of the system as it moves. On figure 3.3 underneath the chosen reference system can be seen.
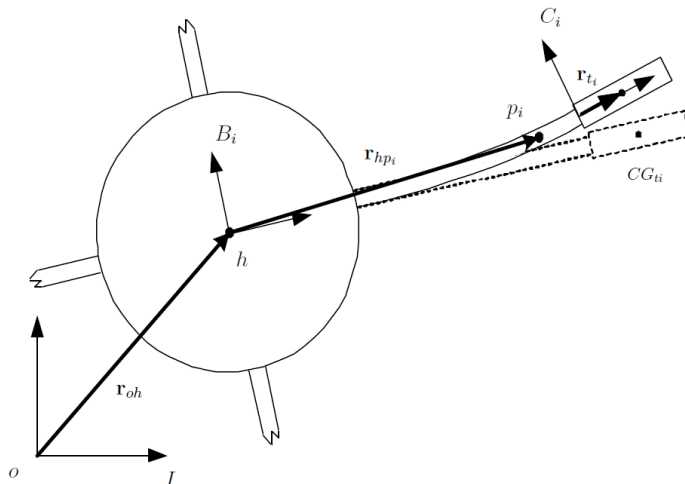
Figure 3.3: Reference frames from R.H. Christensen's [1] Thesis

The inertial stationary reference frame is placed in the external static structure that holds the rotor-hub system and is denoted by I. Lateral rotor-hub movements in x and y direction are measured relatively to this reference. However the individual blades will have local reference frames. B_i will be used as an auxiliary reference frame for the blade movement, while C_i will be used as a reference for the blade's tip movement. Both B_i and C_i are rotate with the rotor. There will be a unique frame for each blade and its tip for $i = 1, 2, 3, 4$. Positive angular direction is defined counter-clockwise.

Underneath, the coordinate system transformation matrices from the static to local blade reference frames are stated.

$$T_{B_i} = \begin{bmatrix} cos(\alpha_i + \theta) & sin(\alpha_i + \theta) & 0 \\ -sin(\alpha_i + \theta) & cos(\alpha_i + \theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$T_{C_i} = \begin{bmatrix} 1 & \frac{\partial y_i(t, L_{bi})}{\partial x_i} & 0 \\ -\frac{\partial y_i(t, L_{bi})}{\partial x_i} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

In the previous chapter α_i and θ were defined, the first one being the angular position of each blade in the rigid disk and the second one the angular position of the rotor. The

18

rotation of the tip is assumed to be very small, therefore, $sin\left(\frac{\partial y_i(t,L_{bi})}{\partial x_i}\right) \approx \frac{\partial y_i(t,L_{bi})}{\partial x_i}$ and $cos\left(\frac{\partial y_i(t,L_{bi})}{\partial x_i}\right) \approx 1$.

To translate a vector r between both auxiliary coordinate systems and to the inertial reference frame one must use the transformation matrices in the following way:

$$r^{B_i} = (T_{C_i})^T \cdot r^{C_i} \tag{3.3}$$

$$r^I = (T_{B_i})^T \cdot r^{B_i} \tag{3.4}$$

Where $\cdot^T$ indicates the matrix transpose operation. The vector super-index indicates the reference frame that is being used as coordinates reference.

3.3.1 Auxiliary Frames Angular Velocity

As stated before both auxiliary frames rotate. B_i rotates together with the rotor's angular rotation Ω, its rotation is defined relative to the inertial frame of reference I

$$\Omega_{B_i}^I = \left\{0 \quad ; \quad 0 \quad ; \quad \Omega\right\}^T \tag{3.5}$$

C_i rotation can also be defined relative to the inertial frame I:

$$\Omega_{C_i}^I = \left\{0 \quad ; \quad 0 \quad ; \quad \Omega + \frac{\partial}{\partial x_i}\left(\frac{\partial y_i(t,L_{bi})}{\partial t}\right)\right\}^T \tag{3.6}$$

3.3.2 Position Vectors

The coordinates that define the position of the rotor's origin h are given relative to the inertial frame I with origin o using the following vector:

$$r_{o-h}^I(t) = \left\{x_h(t) \quad ; \quad y_h(t) \quad ; \quad 0\right\}^T \tag{3.7}$$

The coordinates that define the position of a point p_i on the i-th blade are given relative to the auxiliary coordinate system B_i with origin h using following vector:

$$r_{h-p_i}^{B_i}(t,x_i) = \left\{r + x_i \quad ; \quad y_i(t,x_i) \quad ; \quad 0\right\}^T \tag{3.8}$$

The vector coordinate in x direction does not change with time, this is because, as explained in chapter Model Assumptions, the blades are assumed to be non extensible in the longitudinal direction. As the blade is a distributed system, its vertical displacement y is both a function of x and t.

The coordinates that define the position of the i-th blade tip mass centre of gravity CG_{ti} are given relative to the auxiliary coordinate system C_i using the following vector:

$$r_{CG_{ti}}^{C_i} = \left\{\tfrac{1}{2}L_{ti}; \quad 0 \quad ; \quad 0\right\}^T \tag{3.9}$$

3.3.3 Linear Velocities

The blades are considered a continuous system with infinite points. It is therefore important to define the linear velocity of any point in the blades and in the blade mass tips, to later obtain the blade's equations of motion.

The absolute linear velocity of a point p_i on the i-th blade is described relative to the inertial frame I:

$$v^I_{o-p_i}(t, x_i) = (T_{B_i})^T \cdot \frac{\partial}{\partial t} \left(r^{B_i}_{h-p_i}(t, x_i) \right) + \Omega^I_{B_i} \cdot \left((T_{B_i})^T \cdot r^{B_i}_{h-p_i}(t, x_i) \right) + \frac{d}{dt} \left(r^I_{o-h}(t) \right) \quad (3.10)$$

The first term accounts for the linear velocity of the point i relative to the auxiliary rotating frame B_i, that is translated to the inertial frame I. The second term accounts for the linear velocity in the position of the point i due to the angular rotation of frame B_i, translated to the inertial frame. The last term accounts for the added linear velocity that appears on point i due to the lateral displacement of frame B_i measured relative to inertial frame.

The absolute linear velocity of the tip mass on the i-th blade is described relative to the inertial frame I:

$$v^I_{o-CG_{ti}}(t) = v^I_{o-p_i}(t, L_{bi}) + \Omega^I_{C_i} \cdot \left((T_{B_i})^T \cdot (T_{C_i})^T \cdot r^{C_i}_{CG_{ti}} \right) \quad (3.11)$$

As the tip mass is considered to be rigid, the linear velocity at its centre of gravity CG_{ti} is the sum of the blade's linear velocity at its end (in $x_i = L_{bi}$) plus linear velocity at the position of the point CG_{ti} due to the angular rotation of frame C_i, translated to the inertial frame I.

3.4 Blade Modelling

As stated before, the blade is considered a continuous system and therefore has infinite degrees of freedom. This issue becomes a problem when trying to obtain the equation of motion that explains its behaviour. The blade's motion is governed by partial differential equations, it is impossible to solve these equations by evaluating every point in the blade. Therefore, a discretization of the blade system in finite and simpler one degree of freedom systems must be carried out. There are different methods that can be used to discretize the blade, two commonly used are the Assumed Modes Method and the Finite Element Method.

The assumed modes method [21] is easy to use and produces low order models which are easy to control. It assumes the user has previous information of the system's fundamental motion. With this method, the blade's distributed transverse displacement $y_i(x_i, t)$ is discretized by a finite series of functions.

$$y_i(x_i, t) = \sum_{j=1}^{m} \phi_{i,j}(x) q_{i,j}(t) \quad (3.12)$$

Where $\phi_{i,j}(x)$ are known trial mode shape functions and $q_{i,j}(t)$ are generalized coordinates of the system. The trial functions are chosen with previous knowledge of the modal behaviour

of the system, they are simpler functions the user thinks can describe the blade's behaviour over their entire domain. Generalized coordinates are the minimum amount of independent coordinates that completely describe the motion of the system. The summation coefficient m limits the number of trial mode shape functions the blade is discretized in. The next step in this method is to obtain the discretized kinetic energy, potential energy and virtual work equations and then introduce them in Lagrange's equations, this will result in the obtention of a discrete set of equations of motion. One disadvantage of this method is the difficulty of deciding which trial functions to use for complex continuous systems.

The finite element method [22] produces high order models but can be used in systems with complex motion without previous information. The method consists in discretizing the blade into a finite amount of elements. For each element a very simple local function (low order polynomial) describing its deflection is defined. The boundary coordinates between elements are the generalized coordinates of the discretized system. All local models are then assembled using interpolating functions and boundary conditions to form a global motion model.

The blade modelling has already been carried out by R.H. Christensen [1] and C.S. Jakobsen [17], therefore here I will just present a summary of the obtained result. R.H. Christensen used a combination of both methods explained before to model the blade. Finite element method was used to derive the trial mode shape functions that are later used in the assumed modes method. The trial shape functions are obtained by discretizing the blade into 20 local elements and applying finite element method and eigen-analysis to obtain the most dominant trial mode shape functions, this functions are a very accurate estimate of the blade real mode shapes. C.S. Jakobsen [17] obtains a diagram with the three most dominant estimated mode shape functions.
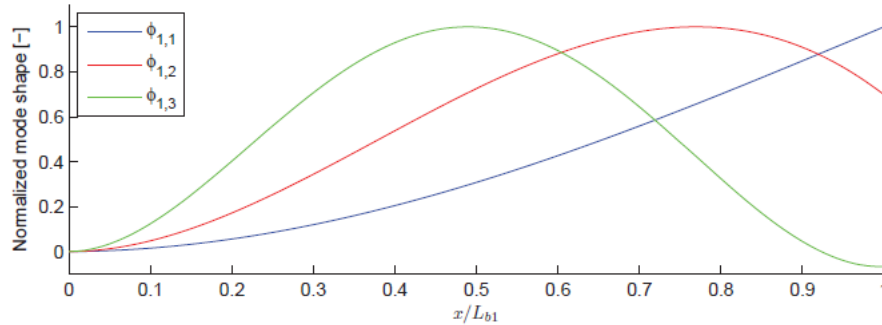
Figure 2.4: First 3 mode shapes of blade 1

Figure 3.4: Blade Mode Shapes from C.S. Jakobsen [17] Thesis

These estimated blade mode shapes are then used as shape functions in the assumed modes method, with m=2 (only the first two estimated modes will be used to describe the system). They have the following structure:

$$\phi_i(x_i) = \left\{ \phi_{i,1}(x_i) \quad \phi_{i,2}(x_i) \right\} \text{ for i} = 1 \ , \ 2 \ , \ 3 \ , \ 4 \tag{3.13}$$

The numerical value of the estimated mode shape functions obtained by R.H. Christensen can be seen on Appendix A. The assumed modes method will be then used to discretize the partial differential equations that appear when defining Lagrange's equations for the whole rotor-blade system. Equation 3.12 will be used to substitute $y_i(x_i, t)$ when necessary.

For $N_a = 6$:

3.5   System's Kinematics

Once the parameters and variables needed to characterize the system are defined, the equations needed to describe the motion of the system have to be determined. These equations have already been obtained in R.H. Christensen's [1] Thesis, therefore here a summary will be presented just to help with the understanding of the derived model.

To derive the equation of motion of the system, Lagrange's approach will be used. This method evaluates the energy present in the system and uses it to derive the classical equation of motion.

Lagrange's equation is defined as:

$$\frac{\delta}{\delta t}\left(\frac{\delta E_{kin}}{\delta \dot{z}_i}\right) - \left(\frac{\delta E_{kin}}{\delta z_i}\right) + \left(\frac{\delta E_{pot}}{\delta z_i}\right) = F_i \tag{3.14}$$

Where E_{kin} is the kinetic energy of the system, E_{pot} the potential energy. z_i are the generalized coordinates of the whole system. The rotor-hub system lateral motion in x and y is modelled by two independent discrete lumped mass systems, therefore their coordinates x_h and y_h are already generalized coordinates of the system. As stated on the previous chapter the blades bending motion, defined by the transverse displacement $y_i(x_i, t)$ will be approximated using the assumed modes method to:

$$y_i(x_i, t) = \sum_{j=1}^{m} \phi_{i,j}(x_i) q_{i,j}(t) \text{ for i=1,2,3,4} \tag{3.15}$$

Where $q_i(t) = \left\{q_{i,1}(t), ..., q_{i,m}(t)\right\}^T$ are the generalized coordinates of the mode shape functions used in the discretization of each i-th blade. We are using m=2 for the discretization, each blade will have two mode shape functions with two generalized coordinates to describe its motion. The total number of generalized coordinates in the whole rotor-blade system will be of N=2+4m=10, the first two ones z_1 and z_2 describe the rotor-hub lateral movement, the other ones are modal coordinates of each blade, with two per blade. F_i is the generalized forces (acting in the direction of each generalized coordinate).

The kinetic energy of the system comes from three contributions, kinetic energy in the rotor-hub subsystem E_{kin}^h, kinetic energy in the blades E_{kin}^b and kinetic energy coming from the blade tip mass E_{kin}^t. The potential energy of the system comes from three contributions, potential strain energy from the elastic deformation of blades E_{pot}^e, gravity potential energy of the system E_{pot}^g and potential energy coming from longitudinal deformation due to

centrifugal force, this term is added to take into account the stiffening effect of blades as they rotate E_{pot}^c. The generalized forces acting on the system include, external control forces that the electromagnets apply on the structure as well as forces that internally act on the structure, including forces that appear during rotation due to the rotor's eccentricity and gravity.

All of the mathematical expressions describing the above mentioned energy terms can be seen in Appendix A. By introducing the different energy terms into Lagrange's equation 3.14 and rearranging the equations of motion of the whole rotor-blade system can be obtained based on the previously defined system's generalized coordinates z.

3.6   System's Equations of Motion

The application of Lagrange's theory resulted in the obtention of the equations of motion of the system. The equations are time varying, they depend on the angular position θ, angular velocity Ω and angular acceleration $\dot{\Omega}$ of the rotor.

The matrix formulation of the equation is:

$$M(\theta)\ddot{z}(t) + H(\theta,\Omega)\dot{z}(t) + K(\theta,\Omega,\dot{\Omega})z(t) = f_{int}(\theta,\Omega,\dot{\Omega}) + T_a f_{ext}(t) \qquad (3.16)$$

Where $z(t)_{\text{(N x 1)}} = \left\{x_h, y_h, q_1, q_2, q_3, q_4\right\}^T$ is the vector of generalized coordinates of the whole system, the first two coordinates describe the lateral motion of the rotor, the other ones describe the deflection of each blade, every q is a two vector coordinate. The total order of the system is of N= 2+4m=10.

The equation of motion derived has a classical structure, $M_{\text{(N x N)}}$ is the mass matrix, $H_{\text{(N x N)}}$ is the damping matrix and $K_{\text{(N x N)}}$ is the stiffness matrix. The symbolic and numerical description of these matrices can be seen in R.H. Christensen's [1] Thesis Appendix A.

The generalized forces acting on the mechanical structure have been divided into two terms, internal and external forces. Internal forces $f_{int}(\theta,\Omega,\dot{\Omega})_{\text{(N x 1)}}$ appear due to gravity and rotor eccentricity, they act as disturbances on the system, their value is however known with information about the rotor's angular velocity, the symbolic and numerical equation defining this forces already in generalized coordinates can be seen in R.H. Christensen's [1] Thesis Appendix A. The external forces are the controlled actuation forces $f_{ext}(t)_{\text{(Na x 1)}}$ the electro-magnets apply on the structure (in Newtons). The external force's $f_{ext}(t)$ vector dimension depends on the number of actuator pairs used, for shaft-based actuation only $f_{ext}(t) = \left\{F_{hx}, F_{hy}\right\}^T$ $(N_a = 2)$, for both shaft and blade actuation $f_{ext}(t) = \left\{F_{hx}, F_{hy}, F_{b1}, F_{b2}, F_{b3}, F_{b4}\right\}^T$ $(N_a = 6)$. T_a is a transformation matrix of dimensions (N x N_a) used on $f_{ext}(t)$ to change from the actuation forces applied on the real system to the generalized forces applied when using generalized coordinates in the motion equation. This matrix is obtained using the mode shape function $\phi_i(x_i) = \left\{\phi_{i,1}(x_i) \quad \phi_{i,2}(x_i)\right\}$ for each i-th blade, evaluated at $x_i = x_{act} = 4cm$, distance where the actuators are placed, same for every blade.

For $N_a = 6$:

$$
T_a =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \phi_{1,1}(x_{act}) & 0 & 0 & 0 \\
0 & 0 & \phi_{1,2}(x_{act}) & 0 & 0 & 0 \\
0 & 0 & 0 & \phi_{2,1}(x_{act}) & 0 & 0 \\
0 & 0 & 0 & \phi_{2,2}(x_{act}) & 0 & 0 \\
0 & 0 & 0 & 0 & \phi_{3,1}(x_{act}) & 0 \\
0 & 0 & 0 & 0 & \phi_{3,2}(x_{act}) & 0 \\
0 & 0 & 0 & 0 & 0 & \phi_{4,1}(x_{act}) \\
0 & 0 & 0 & 0 & 0 & \phi_{4,2}(x_{act})
\end{bmatrix}
\tag{3.17}
$$

For $N_a = 2$ only the first two columns are used.

Throughout the thesis angular acceleration of the rotor $\dot{\Omega}$ will be considered zero. For constant angular acceleration Ω the equation of motion 3.16 becomes periodic time varying depending only the rotor's angular position $\theta = \theta_0 + \Omega t$ which only depends on the elapsed time. Therefore:

$$
M(t)\ddot{z}(t) + H(t)\dot{z}(t) + K(t)z(t) = f_{int}(t) + T_a f_{ext}(t)
\tag{3.18}
$$

The system matrices have a periodicity of $2\pi/\Omega$, meaning they will repeat their values every $(t + 2\pi/\Omega)$ time period.

The equation of motion can be expressed in state space representation, which is very appropriate for controller design. The system is transformed into a set of first order differential equations in matrix form. Because the equation of motion equations are linear, there is no need to linearize at any point, the state space matrices can be directly obtained.

Defining the new state variable as $x(t)_{(2N \text{ x } 1)} = \left\{ z(t) \quad \dot{z}(t) \right\}^T$ the equation of motion can be represented as:

$$
\dot{x}(t) = A(t)x(t) + B(t)f_{ext}(t) + B_v(t)f_{int}(t)
\tag{3.19}
$$

$$
y(t) = Cx(t)
\tag{3.20}
$$

$A(t)$ is the system matrix, $B(t)$ the input matrix, $B_v(t)$ the input disturbance matrix and $C(t)$ the output matrix. $F_{ext}(t)$ are the external actuation forces electromagnets apply to the structure. $F_{int}(t)$ are the disturbance internal forces that act in the system, they appear due to internal forces of gravity and eccentricity. y(t) are the measured signals in the system, in this case, the $N_s = 6$ sensor signals that measure displacement $(x_h, y_h, d_1, d_2, d_3, d_4)$. The rotation parameters of the system $\theta, \Omega, \dot{\Omega}$ are not state variables, this is because they are external parameters that are imposed on the system, but we do not have any control power on their value.

Where:

$$A(t)_{(2N \text{ x } 2N)} = \begin{bmatrix} 0 & I \\ -M(t)^{-1}K(t) & -M(t)^{-1}H(t) \end{bmatrix} \qquad (3.21)$$

$$B(t)_{(2N \text{ x } N_a)} = \begin{bmatrix} 0 \\ -M(t)^{-1}T_a \end{bmatrix} \qquad (3.22)$$

$$B_v(t)_{(2N \text{ x } N)} = \begin{bmatrix} 0 \\ -M(t)^{-1} \end{bmatrix} \qquad (3.23)$$

$$C_{(N_s \text{ x } 2N)} = \begin{bmatrix} T_s & 0 \end{bmatrix} \qquad (3.24)$$

T_s is a transformation matrix of dimensions (N_s x N) used on $x(t)$ to change from state variables of the system defined using generalized coordinates to the real displacement measurements of the system. This matrix is obtained using the mode shape function $\phi_i(x_i) = \left\{ \phi_{i,1}(x_i) \quad \phi_{i,2}(x_i) \right\}$ for each i-th blade. This function has to be evaluated at $x_i = x_{sen} = 6.25cm$, distance where the sensors are placed, same for every blade.

$$T_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_{1,1} & \phi_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \phi_{2,1} & \phi_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi_{3,1} & \phi_{3,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi_{4,1} & \phi_{4,2} \end{bmatrix} \qquad (3.25)$$

3.7 Electro-Mechanical Model of the Actuators

The previous state-space model assumes the input to the system is the force in Newtons the actuators apply on the structure. However, we cannot measure or control this force. As stated on chapter 2, the variable we can control is the effective applied voltage on the electro-magnets. The electro-magnets are electro-mechanical systems with their own dynamics. For a good approximation of the system's real behavior it is important to include the actuator system dynamics, not only to match the input of the system with the real one, but also because they affect the real structure's behaviour.

The force produced by an electromagnet F_m is non-linear, it varies with the control current i_m going through the coil of the magnet squared and with the air gap z between magnet and the controlled surface squared. However in our system, due to the really the small amplitude of vibrations the magnet actuation system can be approximated to a linear model with good accuracy. Each pair of electro-magnets acting on a surface will be modelled as a single one capable of producing positive and negative forces.

$$F_m = \frac{n_m^2 \mu_a A_a}{4} \left(\frac{i_m}{z_0 + z} \right)^2 \qquad (3.26)$$

Due to the small amplitude of vibrations, the above force equation can be linearized using two parameters K_i and K_s. K_i indicates the magnitude of the linearized force dependent on current change when the air gap $s = s_0$ mm (air gap between electromagnet and controlled surface in static conditions). K_s indicates the magnitude of the linearized force dependent on air gap change during displacement of the controlled surface when the current $i = i_0$ mm (current going through coil in static conditions). K_i and K_s are both experimentally obtained with the use of a force transducer.

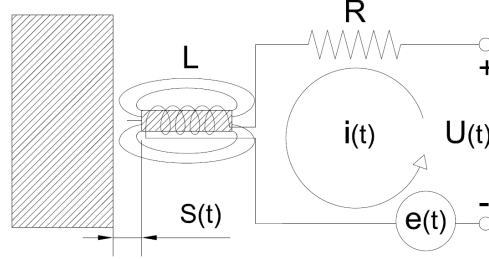A schematic model of the electro-magnet can be seen underneath:

Figure 3.5: Electro-magnet model

The electromagnet circuit is on the right, the controlled surface on the left. R and L are electrical parameters of the electro-magnet, which are obtained experimentally on table 4.1. $U(t)$ is the control voltage input to the system, e(t) is the induced electromotive force in the magnet circuit due to the motion of the controller surface.

The linearized electromagnet force is given by:

$$F_m(t) = -(K_i i(t) + K_s y(t)) \tag{3.27}$$

Where $y(t)$ is the measured displacement from static position of the controlled surface.

The electrical equation describing the electro-magnet's circuit is:

$$u(t) + u_R(t) + u_L(t) + e(t) = 0 \tag{3.28}$$

Substituting:

$$u(t) - Ri(t) - L\frac{di}{dt} + K_i \dot{y}(t) = 0 \tag{3.29}$$

The induced electromotive force is also modeled with K_i because this phenomenon induces a current in the coil.

Rearranging the previous equation:

$$\frac{di}{dt} = -\frac{R}{L}i(t) + \frac{K_i}{L}\dot{y}(t) + \frac{1}{L}u(t) \tag{3.30}$$

26

With equations 3.27 and 3.30 an actuator model can be obtained. There will be two of these equations per pair of electromagnets acting on the system. The system is linear therefore, all of the equations can be arranged into matrix based state space representation easily. The input to the system is $u(t)_{(Na \times 1)}$, the state variable is $i(t)_{(Na \times 1)}$, the output of the system is $F_m(t)_{(Na \times 1)}$. $y(t)$ is the measured displacement of the controlled surfaces, this is the output from the rotor-blade mechanical system given in the previous chapter. The order of the actuator state space system is of $N_a = 6$, as there are six pairs of actuators acting on the system.

$$\dot{i}(t) = A_i(t)i(t) + B1_i(t)\dot{y}(t) + B2_i(t)u(t) \tag{3.31}$$

$$F_m(t) = -C1_i i(t) - C2_i y(t) \tag{3.32}$$

The symbolic and numerical description of these matrices can be seen on Appendix A

To test the actuator model a 20 V step is introduced in u_1. This will generate an output force in actuator one F_{m1} which corresponds with F_{xh}.
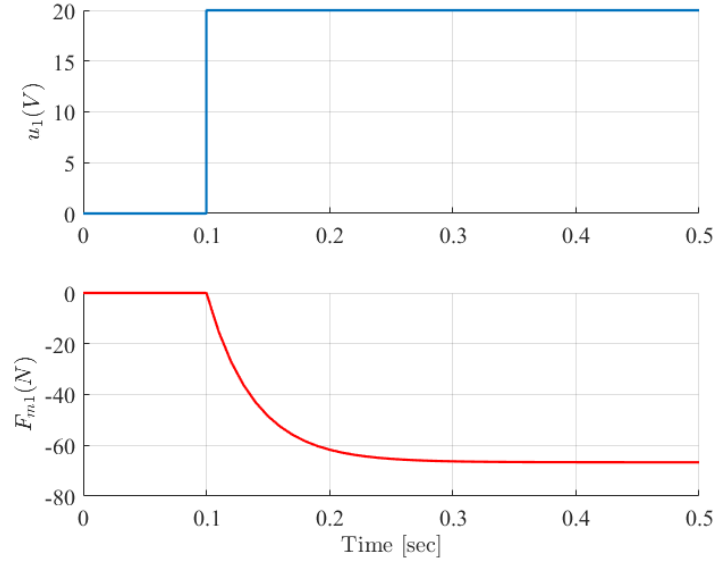

Figure 3.6: Electro-magnet model

As it can be seen from the figure above, the electromagnet has the dynamics of a first order system, with a settling time of 0,2 seconds. This dynamics are quite dominant and therefore should be included in the model of the system as it will change its behaviour.

3.8   Final State Space Representation of the system

By combining the mechanical model of the system derived from the equations of motion with the actuator model, the complete model is obtained. It is a linear periodic time varying system.

The new augmented state variables are $x(t)_{((2N+Na) \text{ x } 1)} = \left\{ z(t) \quad i(t) \quad \dot{z}(t) \right\}^T$.

The state space system has the following structure:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + Bv(t)f_{int}(t) \tag{3.33}$$

$$y(t) = Cx(t) \tag{3.34}$$

Where $u(t)$ are the external voltage actuation signals applied to the electro-magnets mounted in the system, with dimensions of $(Na \text{ x } 1)$. $F_{int}(t)$ are the disturbance forces that act on the system due to internal forces of gravity and eccentricity, their magnitude is known, equations defining their value are stated in R.H. Christensen's [1] Thesis Appendix A, they depend on the value of $\dot{\Omega}, \Omega, \theta$. $Y(t) = \left\{ x_h, y_h, d_1, d_2, d_3, d_4 \right\}^T$ are the measured displacement signals.

The new augmented system matrices are defined as:

$$A(t)_{((2N+N_a) \text{ x } (2N+N_a))} = \begin{bmatrix} 0 & 0 & I \\ 0 & A_i & B1_iT_s \\ -M(t)^{-1}(T_aC2_iT_s + K(t)) & -M(t)^{-1}T_aC1_i & -M(t)^{-1}H(t) \end{bmatrix} \tag{3.35}$$

$$B_{((2N+N_a) \text{ x } N_a)} = \begin{bmatrix} 0 \\ B2_i \\ 0 \end{bmatrix} \tag{3.36}$$

$$B_v(t)_{((2N+N_a) \text{ x } N)} = \begin{bmatrix} 0 \\ 0 \\ -M(t)^{-1} \end{bmatrix} \tag{3.37}$$

$$C_{(N_s \text{ x } (2N+N_a))} = \begin{bmatrix} T_s & 0 & 0 \end{bmatrix} \tag{3.38}$$

3.9   Matlab Simulator

To analyze the model and design controllers Matlab will be used. A Simulink model of the system has been built with the state space model matrices derived in the previous sections.

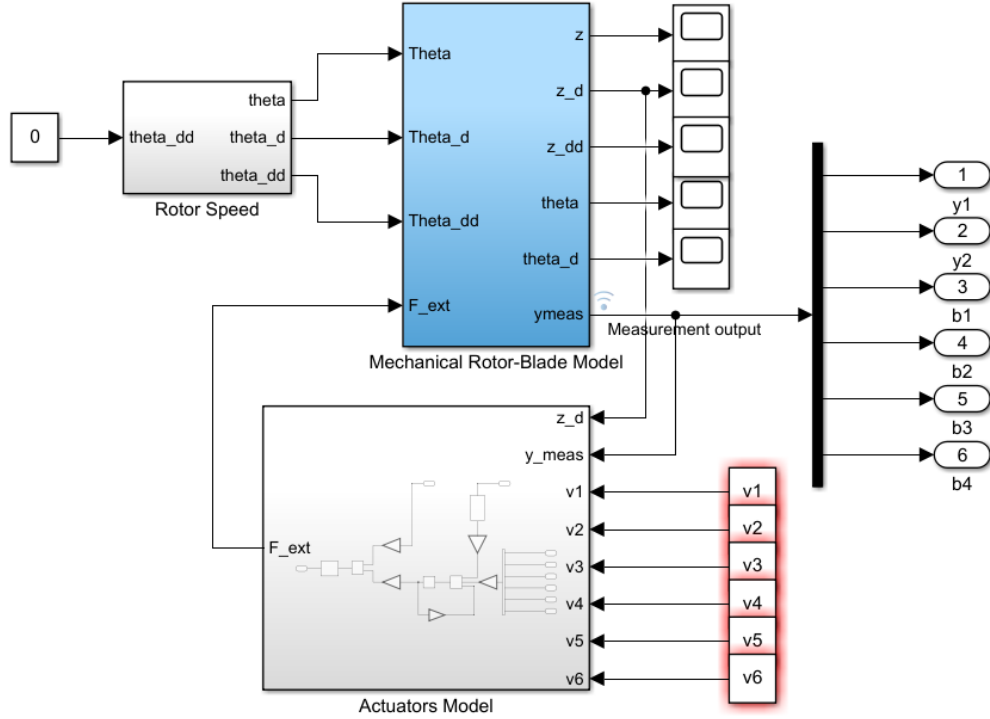This model will be used as the open-loop simulator of the rotor-blade system.

Figure 3.7: Rotor-Blade System Open-loop Simulink Simulator

The input to the modelled system is the control voltage signal $u(t) = \left\{ v_1, v_2, v_3, v_4, v_5, v_6 \right\}^T = \left\{ V_{xh}, V_{yh}, V_{d1}, V_{d2}, V_{d3}, V_{d4} \right\}^T$ applied to each pair of electromagnets acting on the structure. For shaft based actuation only $v_3 = v_4 = v_5 = v_6 = 0$. The output of the system is the sensor measured signal $y(x) = \left\{ x_h, y_h, d_1, d_2, d_3, d_4 \right\}^T$.

The angular velocity Ω and acceleration $\dot{\Omega}$ of the rotor can be selected in the Rotor Speed double integrator sub-system. $\dot{\Omega}$ is selected by changing the input constant (in rad/s^2) to the sub-system, it will be zero for every simulation in this thesis. Ω (in rad/s) is selected by changing the initial condition of the velocity integrator inside the sub-system.

The Actuators Model sub-system is built using actuator state space model in equation 3.31:

29
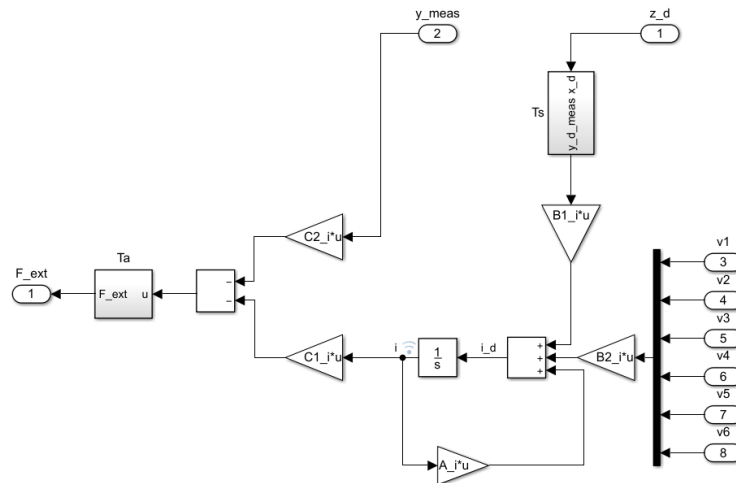
Figure 3.8: Actuation Sub-System Simulink Simulator

The output of this sub-system $f_{ext}(t)$ are the actuation forces in Newtons the electromagnets apply on the mechanical rotor-blade structure. This is the input to the mechanical sub-system.

The Mechanical Rotor-Blade Sub-system is built using the equation of motion of the mechanical system 3.16:

Figure 3.9: Mechanical Rotor-Blade Sub-System Simulink Simulator

The disturbance internal forces f_{int} are calculated inside $F-matrix$ block, with information on the rotor's $\dot{\Omega}$, Ω and θ.

To test the accuracy of the simulator its open-loop response to a certain voltage signal is compared to the experimental one. Underneath, a 20 V step signal in the hub x direction pair of actuators is introduced both in the simulat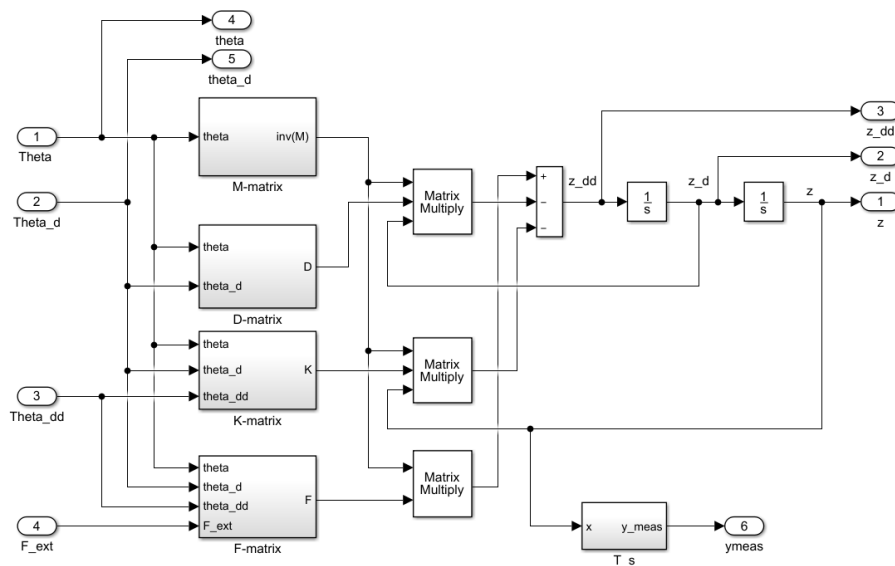or and in the test-rig. The simulated output displacement x_h is obtained and compared with the test-rig experimental sensor output. This test has been carried out in static conditions, $\Omega = 0$.
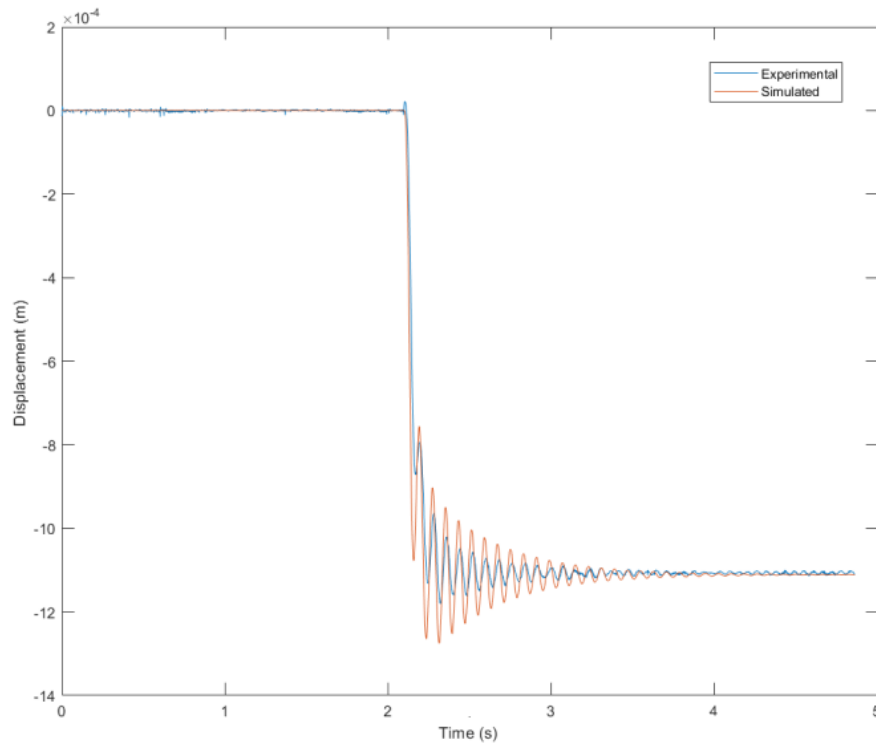

Figure 3.10: Simulated and Experimental x_h displacement with a 20 V step

As it can be seen the model is an accurate approximation. Important characteristics from the experimental data are replicated by the modelled system, these include the vibration frequency and settling time.

4 SYSTEM ANALYSIS

Once an appropriate and accurate mathematical model of the system is built, it can be analyzed to understand how it responds both in the frequency and time domain. It is important to understand how the system behaves in open-loop to be able to later decide on the specifications the designed controllers need to fulfill, as well as to check their performance.

Controllers are designed to improve different open-loop behaviours, these may include, improving stability, speed response, steady-state error or dampening of oscillations. As stated on the previous Chapter, in equation 3.33, the state space model of the system is linear periodic time varying, the matrices that explain the behaviour of the system change with time. However, most analysis tools are defined for time invariant systems. Therefore, to use classical analysis and controller design tools the system matrices will be evaluated at the static point of operation $\dot{\Omega} = \Omega = \theta = 0$. Chapter 7 will study how this static controllers adapt to the rotation and what can be changed to improve their performance.

The time invariant state space model of the system in static conditions is:

$$\dot{x}(t) = A_0 x(t) + B_0 u(t) + B_{v0} f_{int}(t) \tag{4.1}$$

$$y(t) = C x(t) \tag{4.2}$$

With state variables $x(t)_{((2N+Na) \times 1)} = \left\{ z(t) \quad i(t) \quad \dot{z}(t) \right\}^T$. There are two inputs to the system the external actuation voltage signal $u_t(t)$ and the vector of internal force disturbances $f_{int}(t)$, the disturbance forces magnitude is known, their value is constant for static conditions. The output of the system are the measured sensor signals $y(t) = \left\{ x_h \quad y_h \quad d_1 \quad d_2 \quad d_3 \quad d_4 \right\}^T$.

We now have a linear time invariant system we can analyze using classical tools.

4.1 Stability

Stability is a very important property of dynamic systems. Stability studies how the system evolves to a small perturbation in the system. Stability of the system is a mayor requirement for dynamic systems, it is most of the times a necessary condition for the practical applicability of the model.

Stability can be addressed in various ways. For linear time invariant systems one can analyze internal and external stability. If a system is internally stable, given an equilibrium point of the system states, every combination of state variables starting nearby the equilibrium point will stay nearby. If a system is internally asymptotically stable, the state variables

will always go back to the exact equilibrium point. When the system state variables are at equilibrium, they will remain in that point in the absence of external inputs or disturbances. If a system is externally stable given a bounded (finite) input the output will also be bounded, this is also known as BIBO stable (bounded input, bounded output). Internal stability is easily studied with the state space representation of the system, while external stability is uses the input output transfer function representation.

For linear time invariant systems internal stability can be checked by analyzing the system matrix A eigenvalues. This theorem can be found in the Linear System Control book [23]. For asymptotically internal stability: $real\left\{\lambda_i(A)\right\} < 0$. Using the Matlab command $eig()$ with the static system matrix A_0 the eigenvalues can be obtained. They all satisfy the theorem, the system is internally stable.

To check for external stability the state space representation of the system should be expressed in input output transfer functions. This can be done by using the Matlab command $ss2tf()$ with the system matrices, as our system is a MIMO system (multiple input - multiple output) we will have various transfer functions. For the system to be BIBO stable [23] the poles of every transfer function must be in the left-half plane. Transfer function poles can be checked using the $pole()$ command. They all fulfill the requirement. The system is externally stable.

After the analysis we can conclude the system has no stability issues that can affect the controller design or model application.

The equilibrium point of the system can be found using the Simulink model of the plant. The zero input response of the system with the state variables initialized to zero is performed, the system state variables and output will evolve to its equilibrium point. Underneath the output zero input response can be seen:
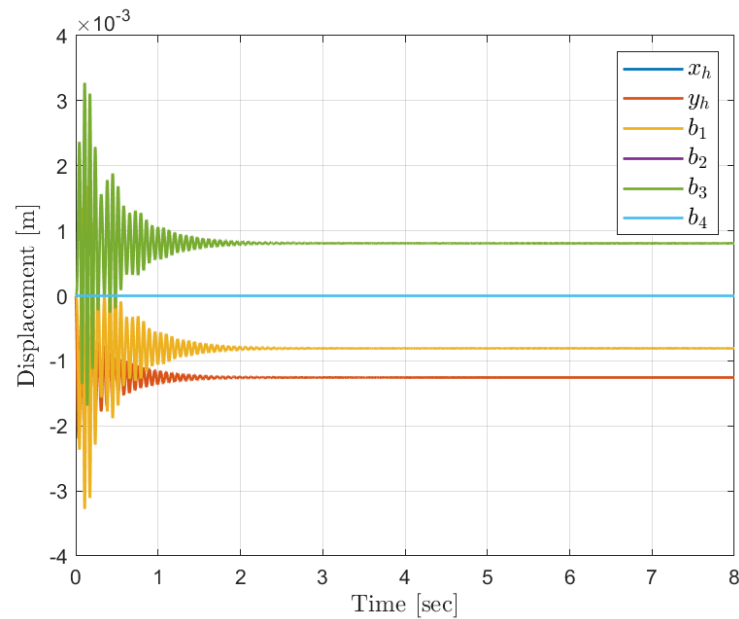

Figure 4.1: Zero input Response Output

We can see the output displacements with only x component motion, like x_h and b_2, b_4 (vertical blades) stay in zero, while components with motion in the y direction like y_h and b_1, b_3 (horizontal blades) move, this is due to the effects of gravity forces acting on the static forces. The state variables of the system will be initialized to their equilibrium position (static configuration), this will eliminate the effect of initial vibrations in the model.

4.2 Controllability and Observability

Controllability and observability are two important properties of dynamic systems which need to be studied before the controller design. They are easily studied by analyzing the structural properties of the state space system.

Controllability studies the possibility of forcing the system to a specific point by the application of a control input. If a given state of the system is non controllable then no input will be able to control the state.

Observability studies whether the states can be observed from the output. If a state is non observable we are not able to determine its behaviour and therefore its information cannot be used by the controller to stabilize the system.

There are various theorems that study observability and controllability, here I will use the most common ones.

From [23] a system is controllable if its controllability matrix,

$$M_c = \left\{ B \quad AB \quad A^2 B \quad .... \quad A^{N-1} B \right\},$$ (4.3)

has full rank (rank=N). Matlab command $ctrbf()$ computes this equation for you. Controllability has to be checked both for shaft actuation control and shaft and blade actuation control. The B matrix changes with the different control options, for shaft based actuation, only two columns are used instead of six. The system fulfills controllability in both situations.

From [23] a system is observable if its observability matrix,

$$M_o = \left\{ \begin{array}{c} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{array} \right\},$$ (4.4)

has full rank (rank=N). Matlab command $obsvf()$ computes this equation for you. With the six sensor measurements the system is observable.

After the analysis we can conclude the system is observable and controllable and therefore we have information of the whole system behaviour, and we have the ability of changing this behaviour.

4.3   Dynamic Response of the System

It is important to analyze the open-loop behaviour of the system. The system's response to a given input can be analyzed both in the time and frequency domain.

The time response of a linear dynamic system consists of the sum of to terms, the zero state solution which depends on the input to the system and the zero input solution which depends on the initial conditions. As seen on figure 4.1 the system evolves to the resting equilibrium position, this equilibrium position is selected to initialize the states, this removes the initial vibrations on the system.

To study the zero state solution, different inputs have been tested in the simulator. It is important to analyze the transient response of the system as well as the steady state response.
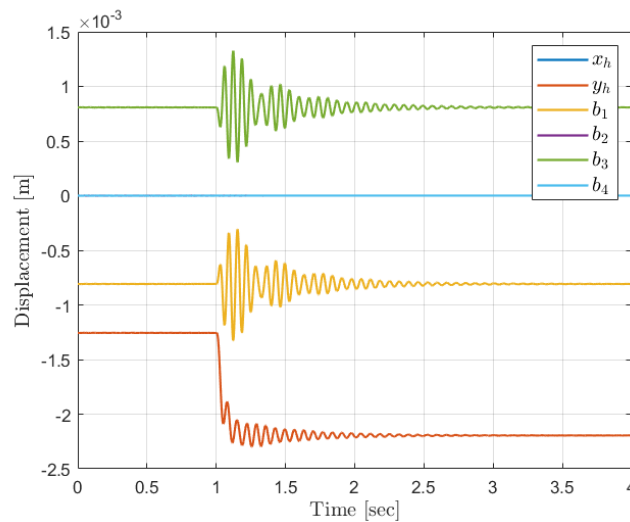

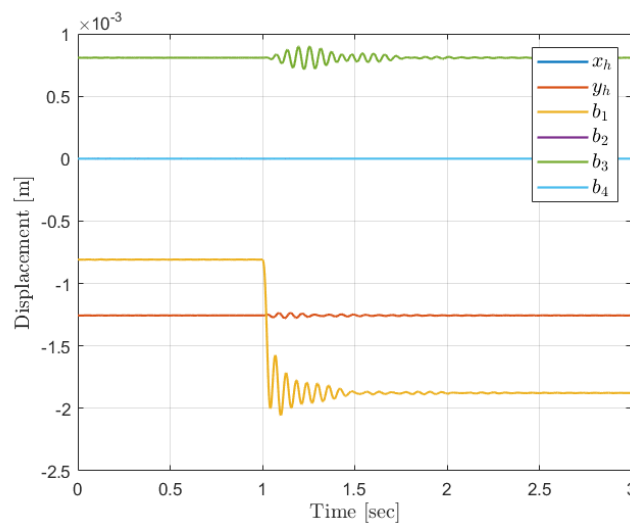Figure 4.2: 20 V step in hub y direction actuators

Figure 4.3: 20 V step in blade 1 actuators

Figure 4.2 shows a 20V step in the y direction hub pair of actuators. We can see how this step alters the steady state value of the directly affected output y_h hub displacement in 1 mm. We can see the coupling between the hub and the blades with motion in the y direction (horizontal blades 1 and 2), both blades vibrate around their equilibrium point, but their steady state value does not change. Both the hub and blades motion have a settling time of 2 seconds. Figure 4.3 shows a 20V step in the blade 1 pair of actuators. We can still see the coupling between the y direction components of motion, however, the coupling from blades to hub is much smaller. The settling time of the transient response of the blades displacement due to the blade actuators is smaller, of approximately 1 second.

The system's time response to a step is very similar to that of a first order in series with a second order under-damped system. The first order system comes from the actuator dynamics, the second order from the mechanical vibration of the system. It is important to study the natural frequency of vibrations and damping ratio of the system. This parameters can be obtained for every state variable using Matlab *damp*() command.

Natural Frequency (Hz)	
x_h	12
y_h	14
Blades $1rst$ mode Component $q_{i,1}$	17
Blades 2_{nd} mode Component $q_{i,2}$	285

Table 4.1: Natural frequencies states of the system

On the table above only the states related to displacement are displayed, this is because their respective velocity states have the same natural frequencies (only integration operation to go from the velocity to displacement state). Intensity states nearly have no oscillation, they are dominated by their first order behaviour. It is already obvious that the blades second mode with natural frequency of 285 Hz is cannot be effectively controlled by our actuation system. As stated in Chapter 2, the electromagnets used as actuators have a limited actuation frequency of 50 Hz. It is however still important to observe those states, to check their behaviour and avoid stability spill-over problems. The damping coefficient of the system states is also derived, it is really small for all displacement and velocity states, in the order of 10^{-2}, the controller should increase the damping of these states to reduce vibrations.

The system's frequency response studies how the system behaves when a sinusoidal input is introduced. It is important to study this response as once the designed controller is introduced the control signal will have a sinusoidal behaviour. The frequency response can be studied in the input output transfer function of the system. Underneath the bode plot of the transfer function from hub x direction actuators and output displacement x_h can be seen.
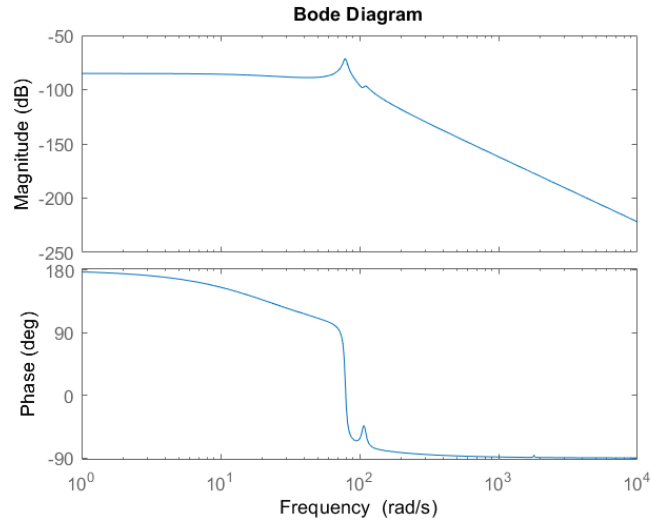
Figure 4.4: Transfer function of x direction actuator input to measurement output x_h bode plot

We can see the magnitude in dB is never positive, the system is therefore stable, it never amplifies the input signal. We can see the phase starts at 180 in low frequencies, this is due to frame of reference considerations, a positive voltage in V_{xh} produces a negative sign displacement x_h. We will be working at a maximum of 50 Hz as the magnets power is extremely reduced after this. We can see that in this working frequency region the system transfer function does not apply noticeable changes in the response magnitude or phase.

After the dynamic study of the system important features needed to establish the controller design specifications have been determined, including settling time of system vibrations, their natural frequency and damping.

5 STATIC CONTROLLER DESIGN

The controllers designed in this section all use feedback control as a basis for their control algorithm. In a feedback controlled system, its output is controlled using information of its measured value or the measured state variable as feedback. The measured signal is compared with a reference signal to generate an error which is fed into the control algorithm, this operation will output a control actuation signal that is applied to the system, correcting its behaviour. The system response with the feedback controller included is know as the close-loop response.
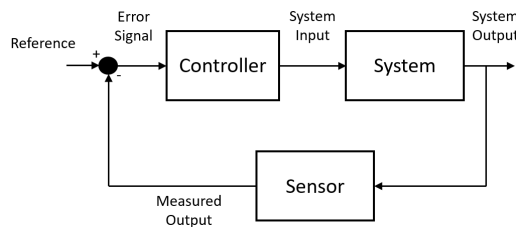

Figure 5.1: Simple Feedback Control Loop

Feedback control is used to ensure a certain behaviour in the system's variables, usually their ability to track reference changes (reference tracking), or maintaining their set-point value even with external disturbances (regulation). We are interested in the second application, we want to reduce vibrations of the displacement state variables around their selected set-point.

There are many reasons why we benefit from feedback control. We can counteract unpredictable and unmodelled disturbances, because their effects will appear in the system output. We can also protect ourselves from model uncertainty issues which lead to discrepancies between the model predicted output of the system to the real one.

All controllers in this section will be designed using the linearized static state space model from the previous chapter, with matrices A_0, B_0 and C. All controllers will be tested on the Simulink simulator model in static conditions, with $\dot{\Omega}, \Omega, \theta = 0$.

5.1 Operation Point

The first step when designing a controller is to decide on the state set-points we want to be working in.

When analyzing stability of the model in chapter 4, the equilibrium point of the system was determined. We could see how the output displacement signals with motion in y

direction had an equilibrium point different from zero due to the effect of gravity.

The rotor-hub system is quite heavy and having to constantly beat the effects of gravity, which are constant in the hub's y direction will consume a great amount of power. It was therefore decided to, correct the horizontal blades position, but leave the hub in its resting position.

Using Matlab command $trim()$ one can obtain the equilibrium point of a system close to a given input, output and state position. For a measurement output of $y_{op} = \left\{0 \quad -0.0013 \quad 0 \quad 0 \quad 0 \quad 0\right\}^T$ in mm, an input of $u_{op} = \left\{0 \quad 0 \quad -15.1 \quad 0 \quad +15.1 \quad 0\right\}^T$ (V) is needed. Both u_{op} and y_{op} point of operation will be used as linearization point of the static state space states used in the controller design and of the input output signals of the system. This implies that we need to subtract the output operation point from the output measurements coming from the simulator, this measurements will be used in the control algorithm which will create an actuation signal, the input operation point must be added to this computed signal before entering the simulator plant.

5.2 System Discretization

To simulate the behaviour of the real acquisition system we will transform the continuous system matrices defined in equation 3.33 to a discrete time with the following structure:

$$x(k+1) = F_0 x(k) + G_0 u(k) + G_{v0} f_{int}(k) \tag{5.1}$$
$$y(k) = C x(k) \tag{5.2}$$

The next step is to decide on the sampling rate T to use in the discretization. According to Nyquist for a correct recreation of the system's behaviour, T has to be at least two times smaller than the time constant of the fastest pole in the system. However for a correct control of the system, T should be at least 10 times smaller.

In our system the fastest poles come from the blade's second mode shape states, which vibrate at approximately 300 Hz, for a theoretically correct control of the system we need a sampling frequency of 3000Hz , $T = 0.0003s$. However, as stated in Chapter 2, our actuators have a limited frequency working range that only allows as to control the blade's first mode at 20 Hz. The control unit developed during the semester has a maximum sampling frequency of 200 Hz. To give veracity to the designed controllers, the selected sampling rate $T = 0.005s$.

We will assume we have a ZOH (zero order hold) acquisition system, all signals in the system are constant between the sample instants, therefore they can only change value at these sample instants.

To discretize the continuous state space system, Matlab command $c2d()$ with ZOH will be used. The discrete system is stable if the absolute value of the eigenvalues of F_0 lie inside the unit circle. This is true for $T_s = 0.005s$. Controllability and observability are also checked for the discrete system, this is tested the same way as in continuous system,

it is both observable and controllable with both shaft and individual blade actuation , and only shaft base actuation.

5.3   PID Controller

PID Controller uses a very simple yet effective control algorithm, it uses a feedback compensator structure, like the one in figure 5.1. One of the advantages of PID's its their simplicity, they are easy to understand and tune.

PID controllers are applied to SISO systems (single input-single output). They are applied to the input-output transfer function of the system.

The controllers output actuation signal is computed in the time domain with the following equation:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \tag{5.3}$$

K_p=proportional gain, K_i=integral gain, K_d=derivative gain are controller parameters that have to be tuned. The PID control algorithm calculates both the derivative and integral of the error signal, multiplies each term by its corresponding parameter and computes the control signal.

Increasing K_p reduces the output settling time and reduces steady state error, it however increases output overshoot. K_p anticipates the error trajectory, increasing its value, dampens the system reducing overshoot. K_i tends to eliminate steady state error, as it integrates its value, it can however increase overshoot.

Taking Laplace on equation 5.3, the controller transfer function can be obtained as:

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + Ki}{s} \tag{5.4}$$

In Laplace domain the PID can be seen as a transfer function that multiplies the system's transfer function, the overall close-loop response of the system can be changed by changing the poles and zeros of the controller transfer function. This is accomplished by changing the controller parameters. With knowledge about the systems zero-pole location one can change its behaviour in a desired manner adjusting the PID transfer function.

In our case we have a MIMO (multiple input-multiple output) system with coupling between output signals. However for the PID design the system will be treated as a decoupled system with six individual transfer functions relating each input (pair of actuators) to the corresponding output displacement. This transfer function can be obtained using Matlab command $ss2tf()$ and selecting the desired input-output relations.

PID controllers will be designed with transfer functions from the continuous system, once the parameters are obtained, the controllers will be discretized with $c2d()$.

5.3.1 Model Order Reduction

The obtained individual input-output transfer functions have a very high order, up to 26 poles. Analyzing the individual system's dynamics due to its zero-pole location is quite complex for high order models. Understanding the dynamics of these systems is important for the PID parameter design, therefore the transfer functions order will be reduced.

To reduce the transfer function model we first need to obtain its zeros and poles, this can be done with the Matlab command $zpk()$. Once we can see all poles and zeros we need to analyzing. All poles and zeros in every transfer function are stable (lay on the LHP), we therefore have the possibility of cancelling them without compromising stability. We will start by cancelling those poles and zeros with same value, then those poles and zeros with similar value, lastly we will cancel poles with very high negative value, which correspond to very high dynamic components of the system and therefore don't have a lot of influence in the visible response of the system. The last step is to adjust the DC-gain (value at s=0) of the new reduced transfer function. We have removed roots of the system and this alter the DC-gain of the system, to correct it, we need to divide the new transfer function by its DC-gain and multiply by the original transfer function DC-gain, this makes sure both transfer functions have the same value at steady state.

All individual transfer functions have been reduced from a high order system to a third order system (one real pole and a complex pair).

Underneath a comparison between the initial model and the reduced model response to an impulse and a step can be seen for the hub's x component direction, and for blade 2.
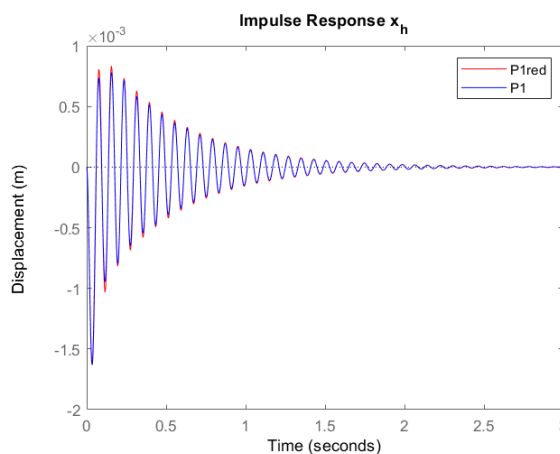

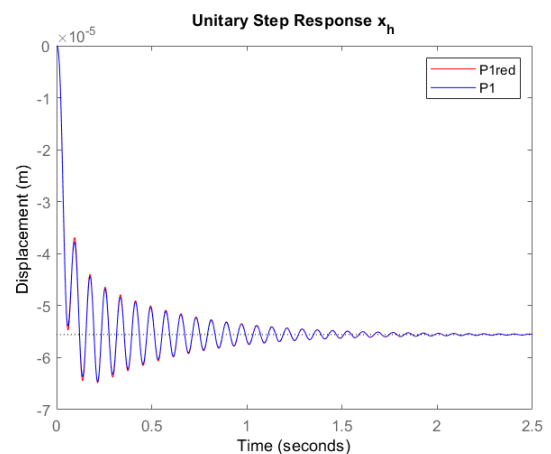Figure 5.2: Impulse response hub x
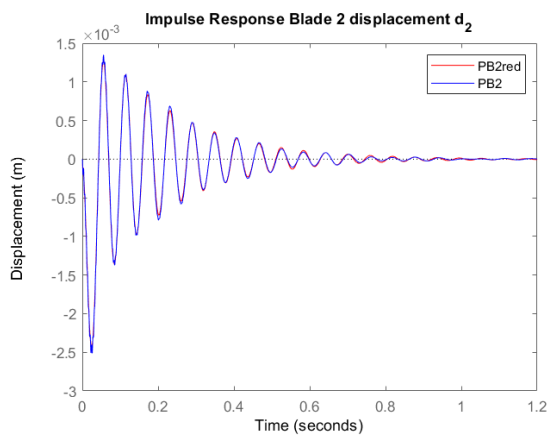
Figure 5.3: Step response hub x
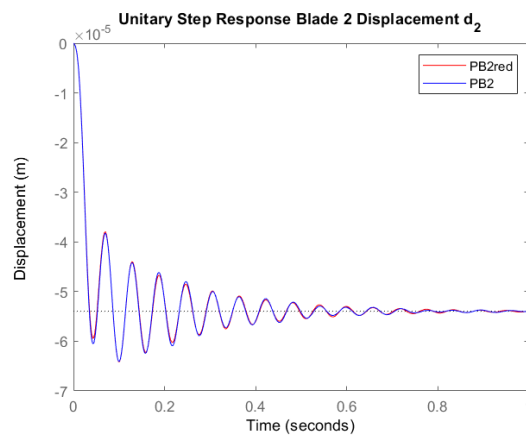
Figure 5.4: Impulse response blade 2

Figure 5.5: Step response blade 2

We can see how both for the hub and blades the reduced models have a nearly identical response both for to an impulse and a step.

5.3.2 PID design with Root Locus Analysis

Once the reduced transfer functions are determined, Root Locus Analysis will be used as a design tool for PID design. With this method controllers that satisfy specific dynamic performances can be designed.

Root Locus is a graphical tool that allows us to analyze how the roots (poles and zeroes) of the system change with the addition of a feedback controller. Root Locus uses the s-domain map to plot the system's open loop poles and transfer functions, it also plots all of the possible trajectories of the system's poles that appear with the addition of a gain with feedback, by increasing this gain we move the original system poles along the trajectory . We can add to the gain feedback controller poles and zeros, this will modify the possible trajectory the original open loop system can follow in the close-loop case.

Root Locus Analysis can be done with the help of rltool Toolbox from Matlab. This will be used for the PID parameter design. As an example the root locus analysis for hub x direction motion transfer function will be explained.

Figure 5.17 shows the root locus with only a K_p gain feedback controller. The blue crosses are the starting open loop poles, the blue circles are the open loop zeroes, the blue lines are the possible trajectories, the pink squares are the new poles that can move along the trajectories, when controller gain is increased. We can see there is an issue with the original system poles that the controller needs to address, the system has two really slow and high oscillatory complex poles (near the vertical origin line), this produces the vibrations in the system. To reduce vibrations we need to lower the poles (less imaginary component) and make them faster (move them to the left). We can see that with only a proportional gain feedback controller, the pole trajectories allow as to make them faster, but not to decrease their oscillatory value, also it requires a big increase in K_p just to move the poles a little bit to the left, big K_p gains produce overshoots which we do not want.
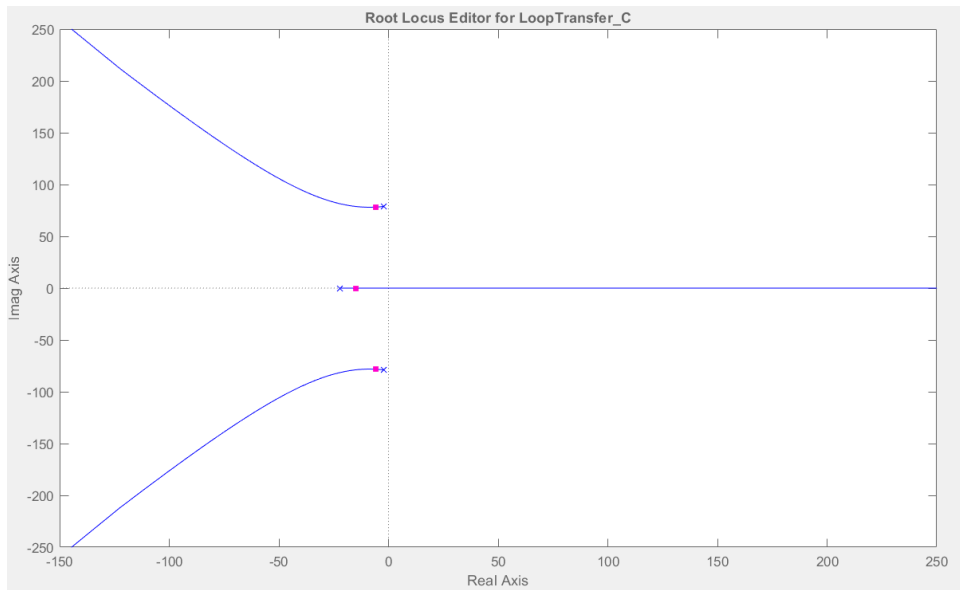
42

Figure 5.6: Root Locus Analysis Hub x direction motion transfer function with P.

We therefore add derivative action to the feedback controller, we add a zero in the root locus. With a PD (K_p, K_d) we alter the possible pole trajectories of the system. By adding a real zero in -55.57, the system's trajectory is changed in a way where with an increase in the gain K_p we are able to reduce oscillatory component of the poles and increase their velocity, we dampen the response of the system, with a reasonable gain we can move the system's original poles from the blue crosses to the pink squares position. At this position the poles have a damping of 0.1, much bigger than the original of one of 0.0025.
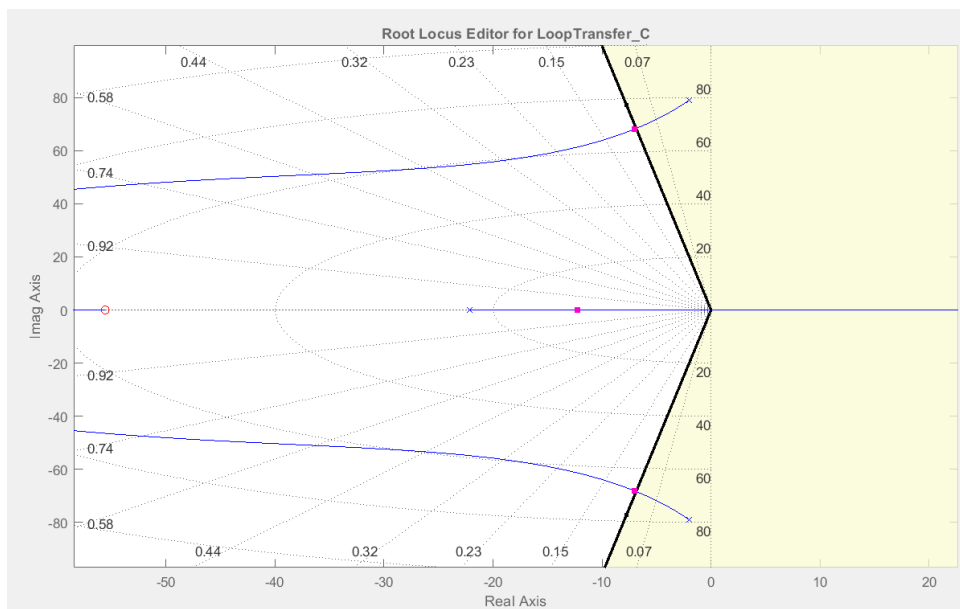

Figure 5.7: Root Locus Analysis Hub x direction motion transfer function with PD.

Adding integral K_i control in the system is not possible, the system becomes unstable. This

is due to the mathematical modelling of the system, the addition of the actuator model, with a pure integrator just after the input is not compatible with PID controller. This is however not an issue as our objective is to reduce vibrations due to punctual disturbances, we do not need to correct steady state error.

This method is followed for all 6 individual transfer functions, PD controller parameters K_p, K_d, are defined for every single input-single output system. PD transfer functions can be checked in Appendix B.

5.3.3   Shaft and Individual Blade Actuation Results

System's response with PD control is now tested with shaft and individual blade actuation control. The controller performance is tested by introducing momentary force disturbances (impulse type) at different points in the structure. At 0.5 seconds a disturbance force is introduced in the hub's x direction, at 3 seconds a force disturbance is introduced in the fourth blade. Underneath the system's output response in open-loop and close-loop can be seen.
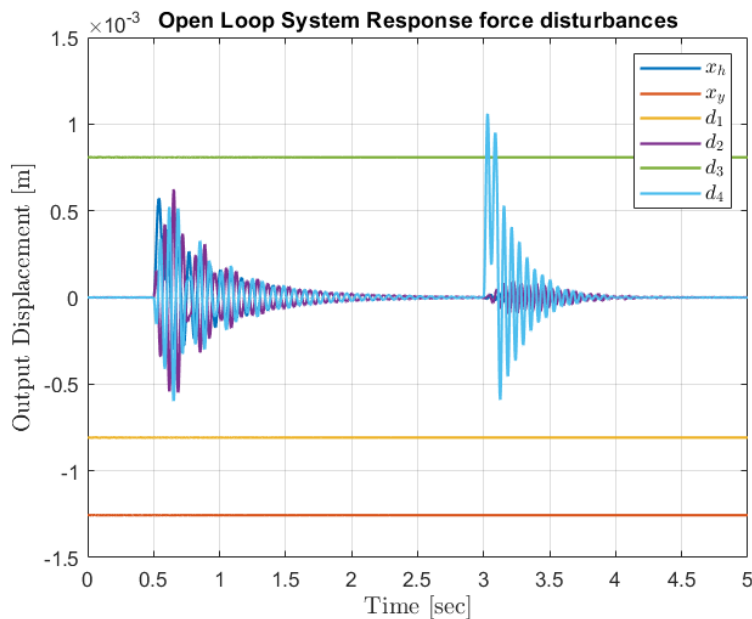

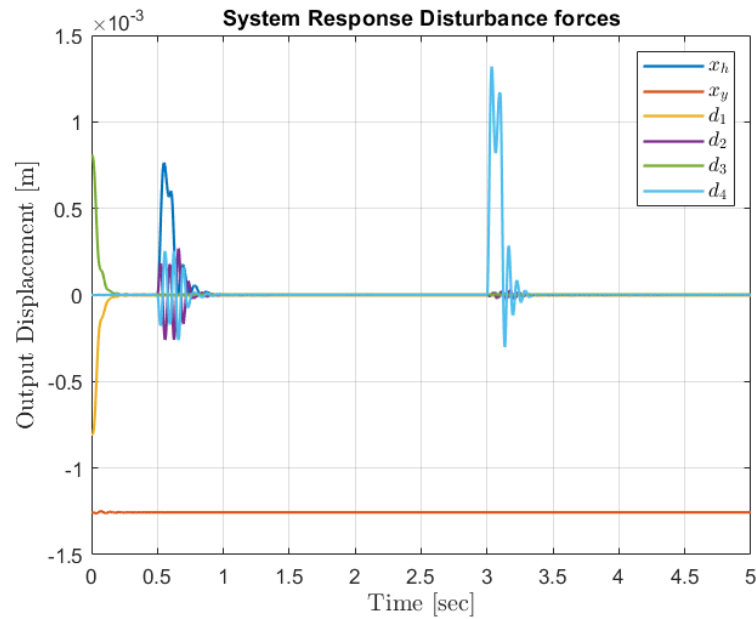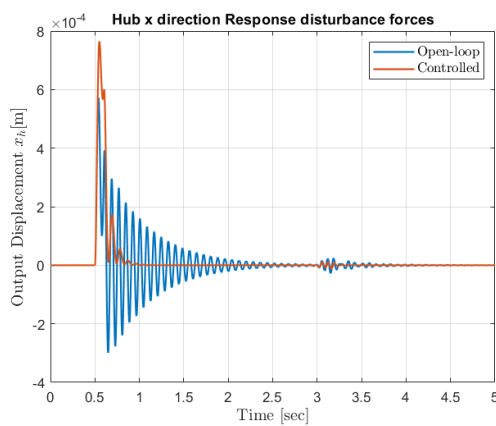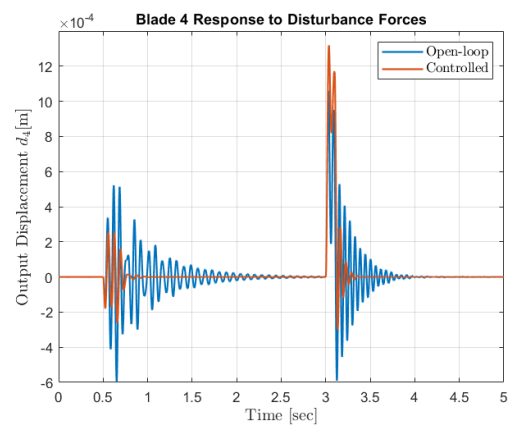Figure 5.8: System Open-loop Response to Disturbance Forces

Figure 5.9: System Controlled Response to Disturbance Forces

First thing we notice is how blade's 1 and 3 PD controllers by default take the blades from their resting position to the operation point (zero displacement). The controlled system also keeps the hub's vertical position y_h in the operating point. Underneath more detailed figures of the directly affected output measurements, x_h and d_4, can be seen.

Figure 5.10: Displacement x_h Response to Disturbance Forces

Figure 5.11: Displacement b_4 Response to Disturbance Forces

We can see how the disturbance force in x has a big impact in the open-loop response of both output displacements. A disturbance force in the individual blade 4 has however much more impact in the open-loop response of the blade itself. The individual PD controllers are able to reduce vibrations that appear due to direct force disturbances, as well as vibrations due to mechanical couplings. There is a small overshoot in both signals, however after that vibrations fade quickly. The hub vibration settling time due to a disturbance in x goes from 2 seconds to 0.5s, the blade vibration settling time due to a direct disturbance goes

45

from 1 s to 0.25s. They are successful at reducing vibrations, they reduce settling time in a 75 %, however they produce initial overshoots in the outputs of 33% when cancelling force disturbances.

The control action signals that were required during this test were never bigger than 15V, this is positive, as we know the controller gain is not too big, it does not force the actuation signals to saturation for regular disturbance forces.
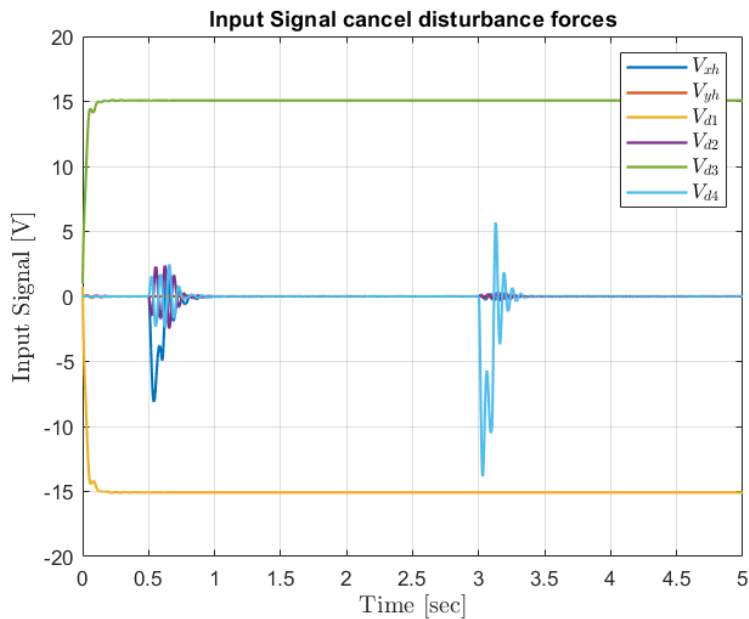
Figure 5.12: Input Signal Response to Disturbance Forces

Overall the individual PD controllers have a good performance for momentary disturbances which is our main objective. I will however recall that these PD controllers cannot keep up with constant disturbances affecting the system, without integral action the controller cannot correct steady state error.

5.3.4 Shaft Based Actuation Control Results

It is also of interest to study the performance of active vibration control with shaft-based actuation control only. This is due to the fact that for many rotor-blade systems, individual blade actuation can become very expensive.

To test performance of only shaft based actuation same simulation as before has been carried out.

Figure 5.13: Displacement x_h Response to Disturbance Forces

Figure 5.14: Displacement b_4 Response to Disturbance Forces

We can see performance in the output response of x_h is nearly identical as before. However the response of the blade output measurement b_4 changes significantly. Although the hub x direction PD is able to compensate blade coupled disturbances c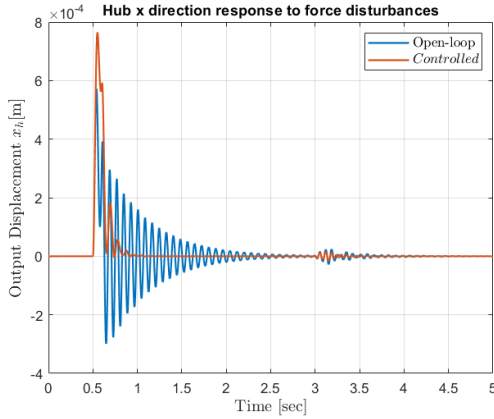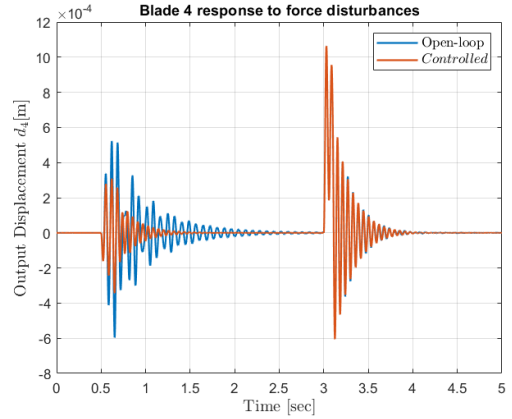oming from the hub, it is not able to compensate direct force disturbances applied on the blade. This is due to the small coupling that exists from blades to hub.

5.4 Full State Feedback

In the previous section PID controllers were designed for the six individual uncoupled SISO transfer functions. However, as seen on figure 5.14, individual controllers cannot cope with indirect coupled vibrations. To take into account the couplings of our MIMO system, multi-variable feedback controllers will be designed. To design these controllers the static state space representation of the system linearized in the before mentioned operation point will be used.

Using state space representation is particularly useful for MIMO systems with a high degree of interaction between the system's variables. It can be applied to complex systems where a great amount of internal variables (states) shape the behaviour of the output.

In full-state feedback, instead of the system output, the internal states values are used to compute the actuation signal. Using the system states gives as more information and therefore allows as to design a better performance controller. Also, in state space the mathematical formulation of controllers can be developed in discrete.

Starting with the discrete state space representation of the system:

$$x(k+1) = F_0 x(k) + G_0 u(k) + G_{v0} f_{int}(k) \tag{5.5}$$

$$y(k) = Cx(k) \tag{5.6}$$

The actuation signal is defined as:

$$u(k) = -Kx(k) + r(k) \tag{5.7}$$

Where K is the feedback matrix, and $r(k)$ is the reference input vector.

The close-loop equation can be determined by inserting equation 5.7 into 5.5:

$$\dot{x}(k+1) = (F_0 - G_0 K)x(k) + G_0 r(k) + G_{v0} f_{int}(k) \tag{5.8}$$

The close-loop system is stable if the absolute value of the eigenvalues of the close-loop system matrix $F_K = (F_0 - G_0 K)$ are inside the unitary circle.

The feedback gain K can be designed with various methods, in the next section we will used the most common ones, Robust Eigenstructure Assignment and LQR (Linear Quadratic Regulator).

As seen in equation 5.7, to use full-state feedback we need information about every state $x(k)$.

5.4.1 Observer Design

Full-state feedback relies on the knowledge of the system's state variables. However, state variables are usually internal variables of the system that are not directly measurable externally. Therefore these state variables need to be obtained using the known signals, measured outputs and computed actuation signals.

If the system to be controlled is observable, then it is possible to use a modified model of the system to estimate the states which are not measured with the input, output signal information.

The observer can be designed as a Kalman filter [24] if the system's signal-to-noise ratios are not very high. However, if the signal-to-noise ratios are sufficiently high, the system can be treated as deterministic and the estimator can be designed as a Luenberger observer, we will use this type of observer.

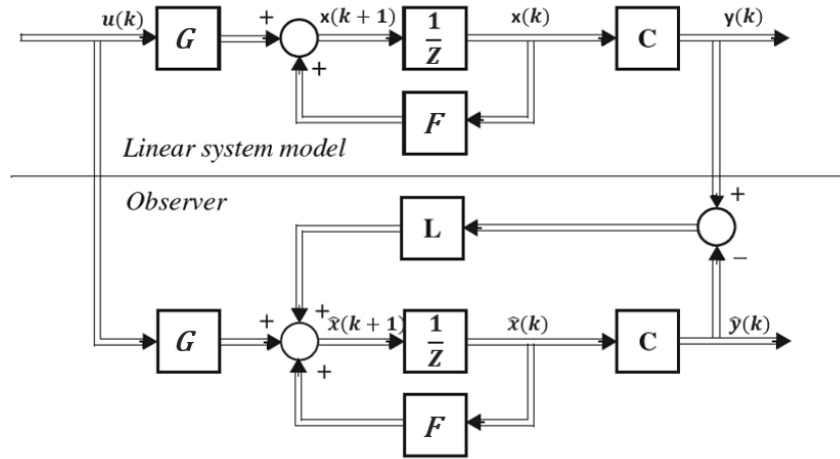The observer has the following structure [23]:

Figure 5.15: Luenberger Discrete Observer Structure.

The observer is characterized with the obtention of gain L. The observer states $\hat{x}(k)$ are the estimated states.

The estimation error $e_e(k) = x(k) - \hat{x}(k)$ is defined as the difference between the real states and the estimated one. The estimation error dynamics are defined by the following equation:

$$e_e(k+1) = (F_0 - LC)e_e(k) \tag{5.9}$$

L has to be therefore chosen so that the error dynamics are fast enough for the observer states to track the original ones.

Our system is affected by internal known disturbances that change with the system's rotation f_int. For the observer to take into account the impact of these forces as the system rotates, they will be fed as another input to the system with their corresponding input matrix G_{V0}.

The equation determining the observer behaviour is:

$$\hat{x}(k+1) = F_0\hat{x}(k) + G_0 u(k) + G_{v0} f_{int}(k) + L(y(k) - \hat{y}(k)) \tag{5.10}$$

For MIMO systems, to obtain the value of L, eigenstructure assignment can be used. This method computes L based on information of the desired close-loop eigenvalues for error dynamics $(F_0 - LC)$. Matlab function $place()$, computes L choosing the eigenvalues close to the desired ones, but promoting orthogonality of eigenvectors which improves robustness.

$$L = place(F_0^T, C^T, [\text{desired } \lambda])^T \tag{5.11}$$

The desired close-loop eigenvalues are chosen to be 10 times faster than the system's continuous open-loop eigenvalues $eig(A_0)$ for a correct state tracking. To translate from

the desired continuous eigenvalues to discrete specifications, the following conversion is used:

$$eig_{discrete} = e^{(eig_{cont}T_s)} \tag{5.12}$$

The whole estimated state vector $\hat{x}$ will be used in the feedback loop.

5.4.2 Robust Eigenstructure Assignment

This method is used to compute the full-state feedback gain K. This method is commonly used for MIMO systems. This method allows the designer to decide the specific location of the feedback close-loop $(F_0 - G_0K)$ eigenvalues. This method is therefore useful when the designer understands the dynamics of the open loop system and knows how to change the dynamics to obtain the desired behaviour.

The matlab function *place*() computes an algorithm that places the close-loop eigenvalues close to the desired location, promoting orthogonality of the eigenvectors solution, which increases robustness.

$$K = place(F_0, G_0, [\text{desired } \lambda]) \tag{5.13}$$

The desired eigenvalue location will be selected using information of the system's continuous open-loop eigenvalues $eig(A_0)$. We want the close-loop system to be faster and to have less oscillations, we therefore need to augment the real part of all close-loop poles and to reduce the imaginary part of the complex poles. The desired close-loop eigenvalues are designed with a real part 3 times faster than the open-loop eigenvalues, but with 0.75 the complex part.

Using robust eigenstructure assigment it is very difficult to design a good performance gain K while at the same time limiting the magnitude on actuation signal needed. With the obtained K for common disturbances the controller at some points generates input signals of up to 30 V. From Chapter 2 we now electromagnets can stand up to 20 V, therefore a saturation block $(\pm 20V)$ is introduced in the simulator.

Shaft and individual Blade Actuation Control Results

System's response with Full-State Feedback designed with Robust Eigenstructure Assigment is now tested with shaft and individual blade actuation control. The controller performance is tested by introducing momentary force disturbances (impulse type) at different points in the structure. At 1 seconds a disturbance force is introduced in the hub's y direction, at 3 seconds a force disturbance is introduced in the first blade (horizontal blade). Underneath the whole system's output response in open-loop and close-loop can be seen. The input operation point was added in the open-loop response to make the comparison between both responses easier.
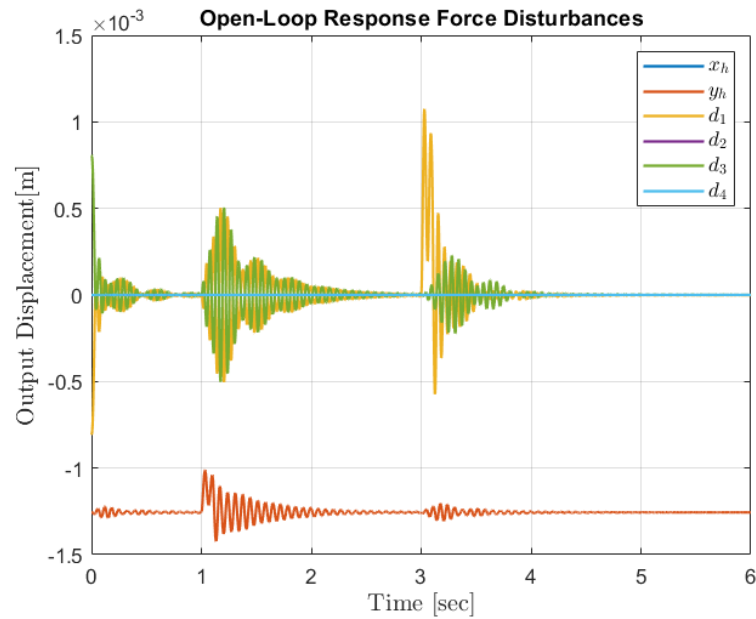
Figure 5.16: Open Loop System response to disturbance forces
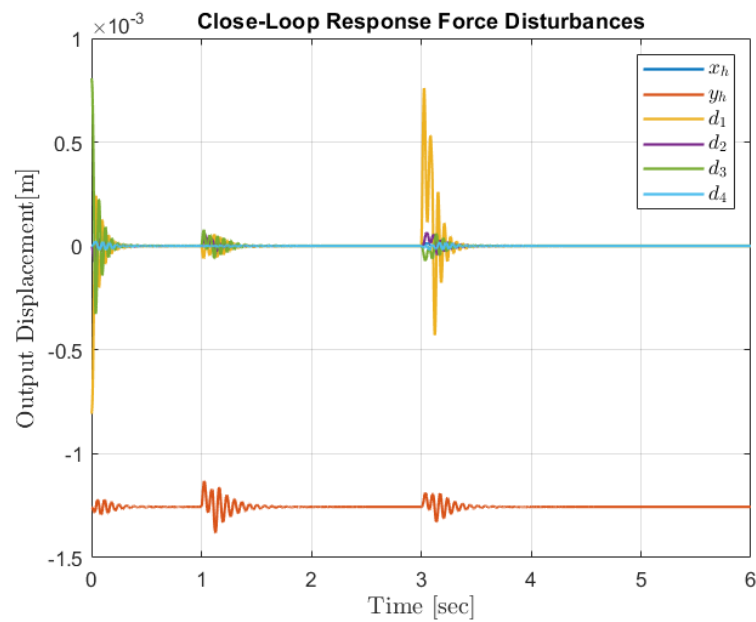
Figure 5.17: Close Loop System response to disturbance forces

We can see how in overall vibrations in the close-loop system are reduced, both in magnitude of oscillations and in settling time. One disadvantage compared with individual PD controllers from previous chapter is how this controller produces small vibrations in those outputs that are not excited by the disturbances in open-loop (states in x direction vibrate), however this vibration is really small. Underneath more detailed figures of the directly affected output measurements y_h and d_1, can be seen.

Figure 5.18: Displacement y_h Response to Disturbance Forces
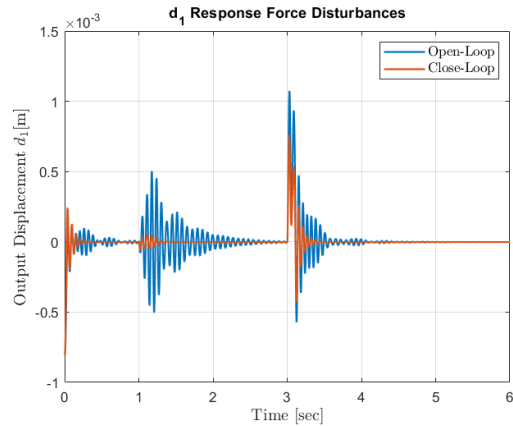
Figure 5.19: Displacement d_1 Response to Disturbance Forces

The controller is able to reduce vibrations that appear due to direct force disturbances, as well as vibrations due to mechanical couplings. Compared with PD's an advantage is how this controller does not produce overshoots, however due to the saturation, vibrations take a little bit longer to settle. The hub vibration settling time due to a disturbance in y goes from 2 s to 0.85 s, the blade vibration settling time due to a direct disturbance goes from 1 s to 0.4 s.

This controller cannot cope with constant disturbances, cannot accomplish zero steady state error.

Shaft Based Actuation Control Results

We are now going to study the performance of the controller with shaft-based actuation control only.

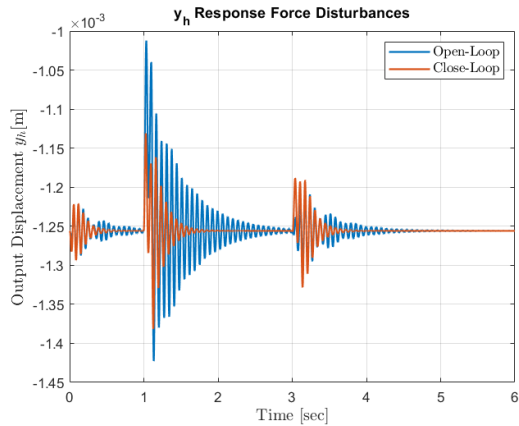Figure 5.20: Displacement y_h Response to Disturbance Forces
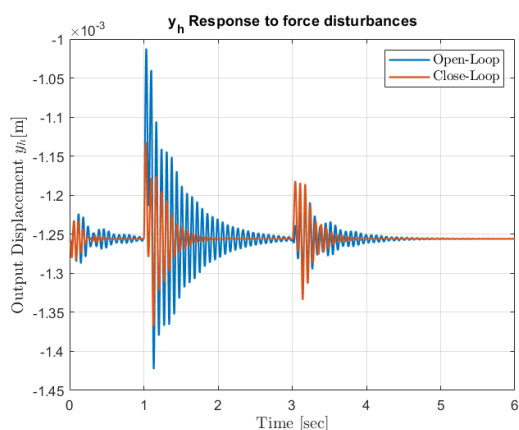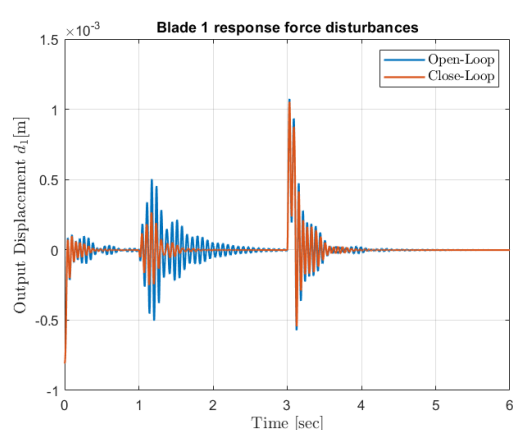
Figure 5.21: Displacement d_1 Response to Disturbance Forces

We can see performance in the output response of y_h is nearly identical as before. With shaft based actuation only the controller is still able to compensate direct and indirect

forces in the shaft. However the response of the blade output d_4 changes. With hub base actuation only the system reduces blade vibrations due to direct disturbances just in a small amount.

5.4.3 Linear Quadratic Regulator

In this section optimal control will be used to compute the full-state feedback gain K. In previous sections we used system's dynamic performance specifications (like settling time, overshoot, resonance damping) to design the controllers. This, sometimes leads to impossible or conflicting actuation signals. With optimal control the controller is designed by minimizing a performance index which takes into account limiting conditions, like actuator input power. The best possible control signal which satisfies boundary conditions will be generated.

The system's quadratic performance index J to minimize [23] is defined as:

$$J = \frac{1}{2} x^T(t_1) S(t_1) x(t_1) + \frac{1}{2} \sum_{k=0}^{N-1} \left[x^T(k) R_1 x(k) + u^T(k) R_2 u(k) \right] T_s \qquad (5.14)$$

Where $S, R_1 T$ and $R_2 T_s$ are weighting matrices, which must be positive definite. This matrices determine how much deviation is allowed in $x(t_1), x(k)$ and $u(k)$ signals. A big R_1 means only a small deviation of the states from their zero positions is allowed. A big R_2 penalizes the amount of input that can be used in the control law. S penalizes state deviations in the final state.

The feedback gain K that optimizes the performance index is found as:

$$K(k) = \left[R_2 + G_0^T P(k+1) G_0 \right]^{-1} G_0^T P(k+1) F \qquad (5.15)$$

Where F_0 and G_0 are the system's discrete state space matrices. P is the solution to the Riccati equation. However the solution of this equation is time varying. A suboptimal constant gain can be obtained by using the steady state converging solution of the Riccati equation P_∞. According to the LQR regulator theorem [23] if the system to be controlled is stabilizable, there is a constant solution to the Riccati equation. Matlab command $dlqr()$ calculates the suboptimal feedback constant gain K_∞. The final control law is given by:

$$u(k) = -K_\infty \hat{x}(k) \qquad (5.16)$$

Where $\hat{x}(k)$ is the observer estimated states.

The Matlab command $dlqr()$ uses weighting matrices R_1 and R_2 to compute the constant feedback gain.

The weighting matrices are tuned to obtain a good active vibration control performance (reduce oscillation amplitude, and settling time) but keeping the control inputs under 20V for common magnitude disturbances.

Shaft and individual Blade Actuation Control Results

System's response with Full-State Feedback designed with Optimal Control is now tested with shaft and individual blade actuation control. The controller performance is tested by introducing momentary force disturbances (impulse type) at different points in the structure. At 1 second a disturbance force is introduced in the hub's y direction, at 3 seconds a force disturbance is introduced in the first blade (horizontal blade). Underneath the whole system's output response in open-loop and close-loop can be seen. The input operation point was added in the open-loop response to make the comparison between both responses easier.
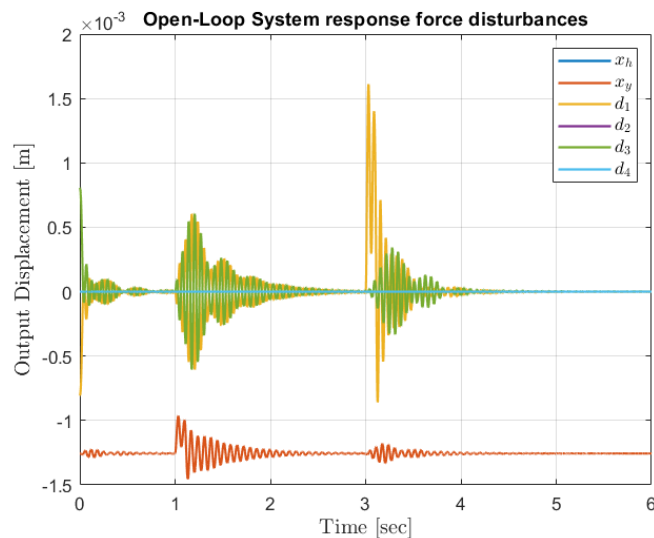
Figure 5.22: Open Loop System response to disturbance forces
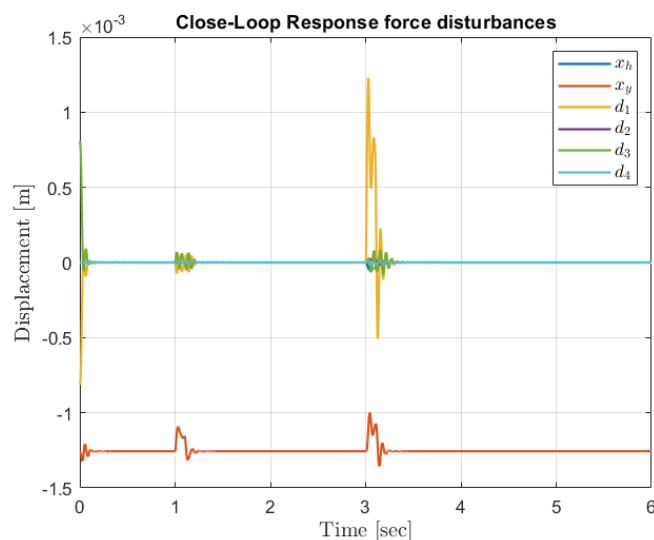
Figure 5.23: Close Loop System response to disturbance forces

The above control action was achieved with the control signals underneath, all inside 20V.
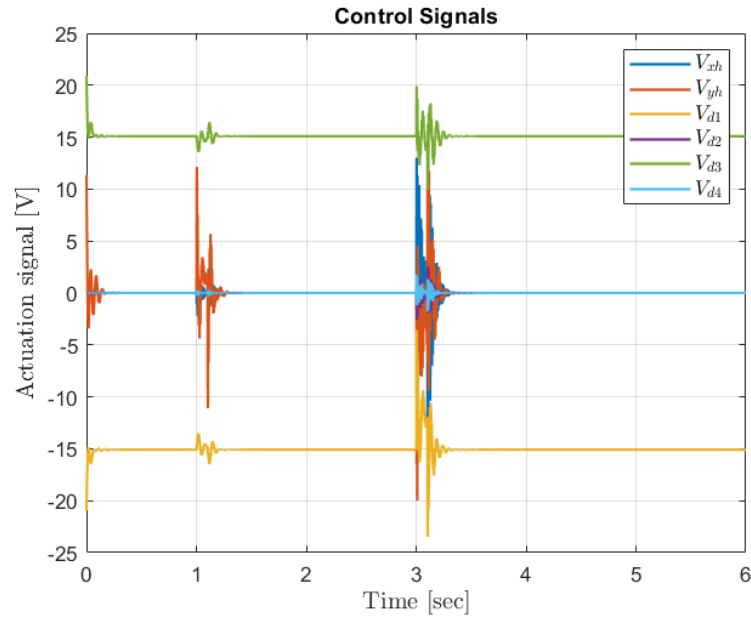
Figure 5.24: Control signals

We can see how with limited control signals the response of the controlled system is really quick compared with the previously designed controllers, settling time is really small. The amplitude of oscillations is also drastically reduced.

Underneath more detailed figures of the directly affected output measurements y_h and d_1, can be seen.
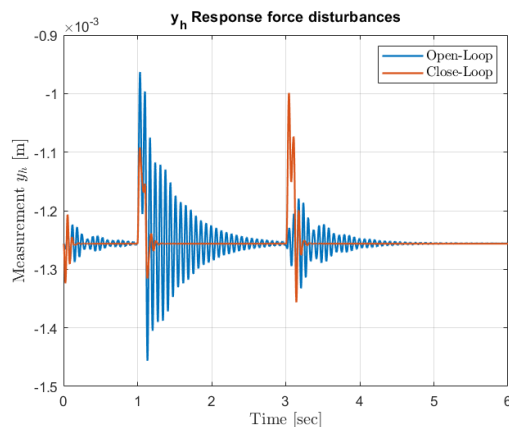
Figure 5.25: Displacement y_h Response to Disturbance Forces with DLQR.
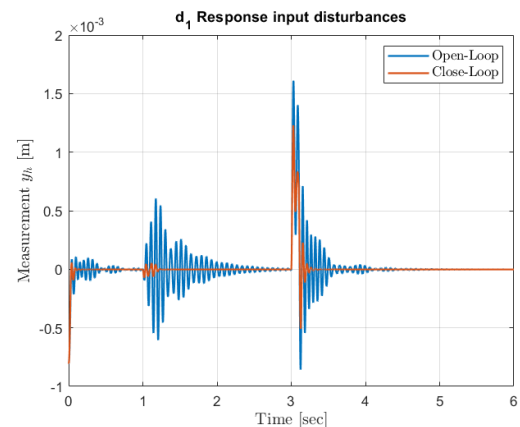
Figure 5.26: Displacement d_1 Response to Disturbance Forces with DLQR.

The controller is able to reduce vibrations that appear due to direct force disturbances, as well as vibrations due to mechanical couplings. This controller reduces settling time of vibrations by a great amount. One issue however, is the over-shoot it creates in y_h when compensating indirect disturbances coming from mechanical coupling from blades.

Shaft Based Actuation Control Results

We are now going to study the performance of the controller with shaft-based actuation control only.
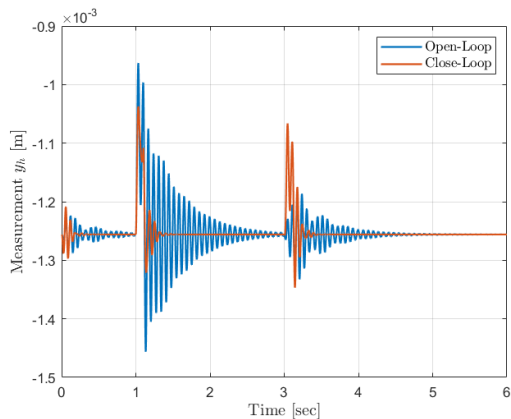
Figure 5.27: Displacement y_h Response to Disturbance Forces with DLQR
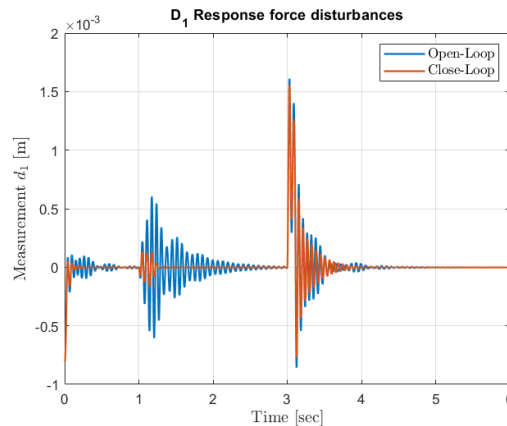
Figure 5.28: Displacement d_1 Response to Disturbance Forces with DLQR

We can see performance in the output response of y_h is nearly identical as before. With shaft based actuation only the controller is still able to compensate direct and indirect forces in the shaft. However the response of the blade output d_4 changes. With this controller we are able to reduce settling time and amplitude of vibrations in blades due to blade disturbances, however the result is not that noticeable.

The Optimal Controllers designed in this section cannot cope with constant disturbances, they cannot accomplish zero steady state error. However with the addition of integral action this can be achieved.

5.4.4   LQR with Integral Action

The full-state feedback controllers design in the previous sections (both with Robust eigenvalue assignment and LQR) cannot cannot cope with constant disturbances, they cannot accomplish zero steady state error. This however can be changed with the addition of integral action. This change can be applied to both controllers, we will apply it however only to the LQR controller as it already achieves a better performance.

Integral control is achieved by converting the desired signals into new integrated states. In this case, we want to control the output signals, we are therefore going to augment the original discrete state space system 5.5 with the new integrated states.

$$\begin{bmatrix} x(k+1) \\ x_i(k+1) \end{bmatrix} = \begin{bmatrix} F_0 & 0 \\ -C & I \end{bmatrix} \begin{bmatrix} x(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} G_0 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ I \end{bmatrix} r(k) + \begin{bmatrix} G_{0v} \\ 0 \end{bmatrix} f_{int}(k) \tag{5.17}$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_i(k) \end{bmatrix} \tag{5.18}$$

Figure 5.29: DLQR + Integral action Structure

Now the DLQR feedback gain K_a can be computed for the new augmented system using Matlab $ldqr()$ command like in the previous system.

The obtained gain K_a has to be divided in two terms, the state feedback gain K and the integral states gain K_i:

$$u(k) = -K_a x_a(k) = -\begin{bmatrix} K & K_i \end{bmatrix} \begin{bmatrix} x(k) \\ x_i(k) \end{bmatrix} \tag{5.19}$$

System's response with the addition of integral action is now tested. The controller performance is tested by introducing constant force disturbances (step type) at different points in the structure. At 1 second a disturbance force is introduced in the hub's x direction, at 3 seconds a force disturbance is introduced in the 4th blade (vertical blade). Underneath the comparison between the response with the system both with and without integral control can be seen.

Figure 5.30: Systems Response to constant force disturbances with DLQR

57

In the figure above we can see how DLQR controller reduces vibrations but has a big steady state error.
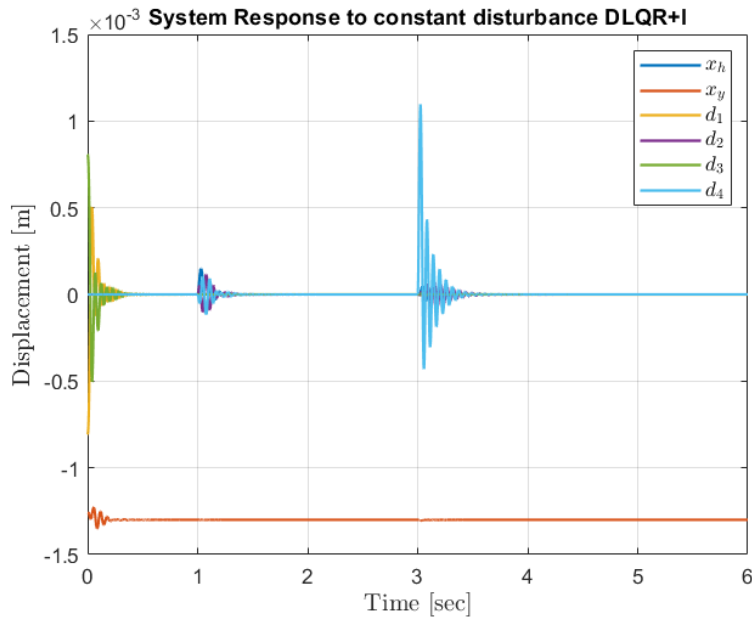
Figure 5.31: Systems Response to constant force disturbances with DLQR + Integral Action

With the addition of integral action the system's steady state error is eliminated. This is not however the main objective of our control goal, as we want to eliminate momentary disturbances that appear in the rotor when rotating.

6 ROTATING CONTROLLER DESIGN

The previous Chapter resulted in the design of four different controllers, individual PD controllers, full state feedback with robust eigenvector assignment, full state feedback with LQR, full state feedback with LQR + integral action. These controllers are designed for the static model.

This section studies the adaptation of the designed controllers with a constant rotor angular velocity Ω.

6.1 Rotating System Analysis

Firstly, the open-loop response of the system with a constant angular velocity of $\Omega = 300$ rpm (operating point) is studied.

The figures underneath shows the effects of rotation in the measured output signals.
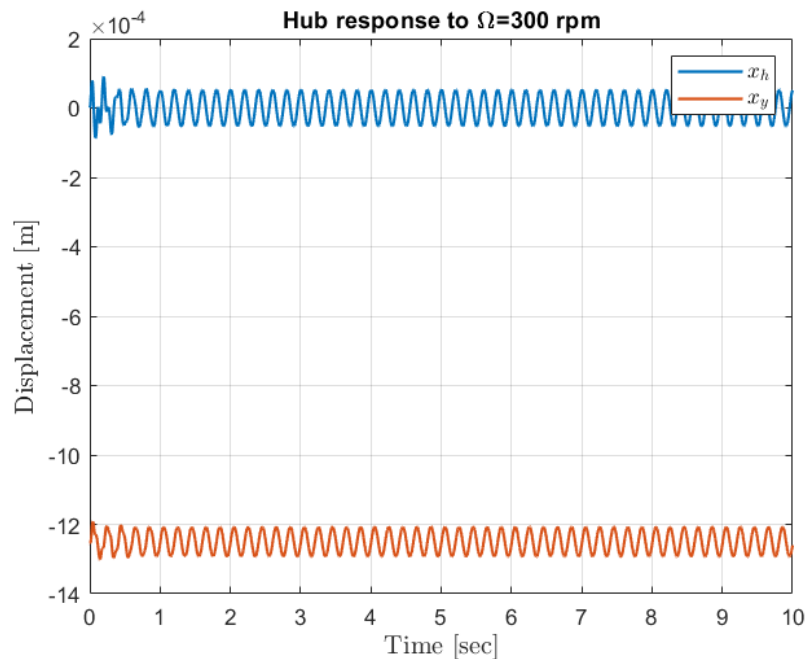
Figure 6.1: Hub displacement Response to Rotation
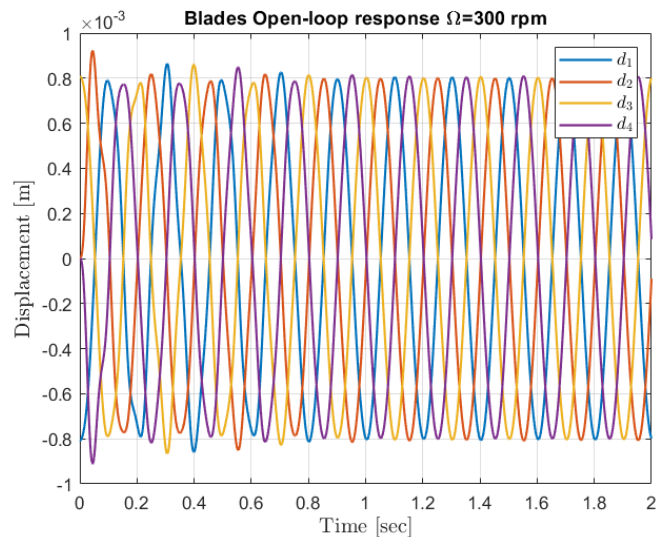
Figure 6.2: Blade displacement Response to Rotation

The figures show that all signals rotate at a frequency of approximately 5 Hz, which corresponds to the frequency of Ω in rpm. The hub displacement in x direction vibrates with an amplitude of 0.050 mm, in the y direction 0.04 2mm. However blade's amplitude vibration is much higher, of 0.8 mm.

In Chapter 3 Mathematical Model, the periodic time varying behaviour of the rotor-blade system described in equation 3.33 was obtained. That equation shows that the system's matrices change with time when there is rotation. The system is also affected by a vector of disturbance forces f_{int} that appear due to internal forces of the system like rotor eccentricity and gravity. The value of these disturbances is known and changes with values of Ω and θ.

Figure underneath shows the variation in internal disturbance forces with rotation.
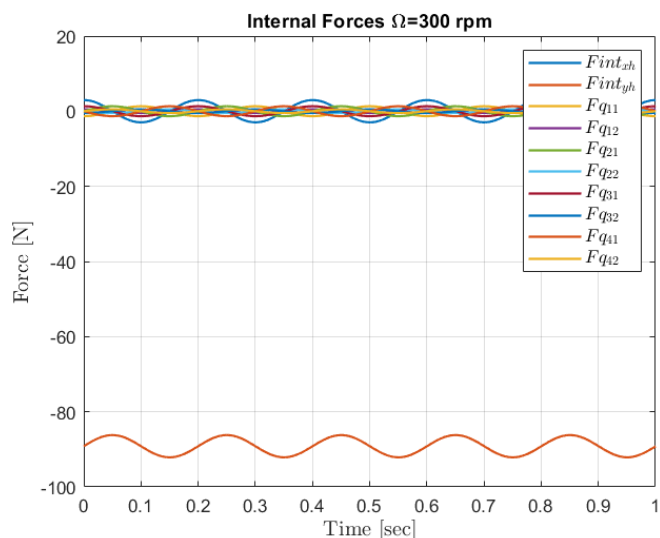
Figure 6.3: Internal Forces Variation with rotation

60

Force variation amplitude is of 3 N in the hub y direction, and around 1 N in the blades and hub x direction. This variations are quite big and have a big impact in the output displacements.

6.2 Controller Adaptation Strategy

The previously designed static controllers performance was tested on the rotating plant simulator. These tests concluded with the rejection of the full-state feedback controllers. With rotation, states vary constantly, this resulted in very aggressive actuation signals. Therefore, the control strategy to follow in this section will be based only on the PD controllers.

With rotation both the system matrices and disturbance forces vary. To compensate the changes in the system properties it would be necessary to implement complex control structures with time varying parameters. As seen from the conclusions of C.S. Jakobsen [17] and J.B. Hansen [18], previous Master Theses, advanced control algorithms are too complex to effectively be applied on the test-rig control interface. Therefore, we will focus our strategy in compensating disturbance forces. As seen on the previous section, the variation in disturbance forces has a great impact in the system's behaviour, making this strategy a good option.

As the disturbance forces are known, they depend on Ω and θ, a feedforward control will be implemented in conjunction with the PD controllers.

6.3 Feed-forward Control Design

Feed-forward control is used to compensate known disturbances. Here it will be used in conjunction to the PD controllers, as we still need to compensate the unknown force disturbances acting on the system. The controlled system has the following structure:
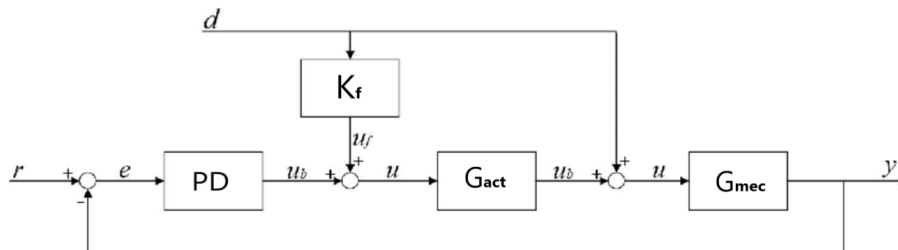
Figure 6.4: Feed-forward Structure

The known internal disturbance is a force directly applied on the mechanical structure. Our objective is to produce an opposite force with the electromagnets that cancels the disturbances. To produce these voltage actuation signals, the dynamics of the actuators need to be taken into account.

Feed-forward gains K_f are determined:

$$K_f = -\left(\frac{1}{G_{act}}\right)H_f \tag{6.1}$$

G_{act} are the individual actuator transfer functions from input voltage to output force. H_f is a filter added to the gain transfer function to assure causality, a fast pole is added to each function.

6.4 Results

System's response with the addition of feedforward control is tested. The controller performance is tested in the simulator with $\Omega = 300$ rpm. Additionally an unknown disturbance force is added at 0.5 s in the hub's x direction.
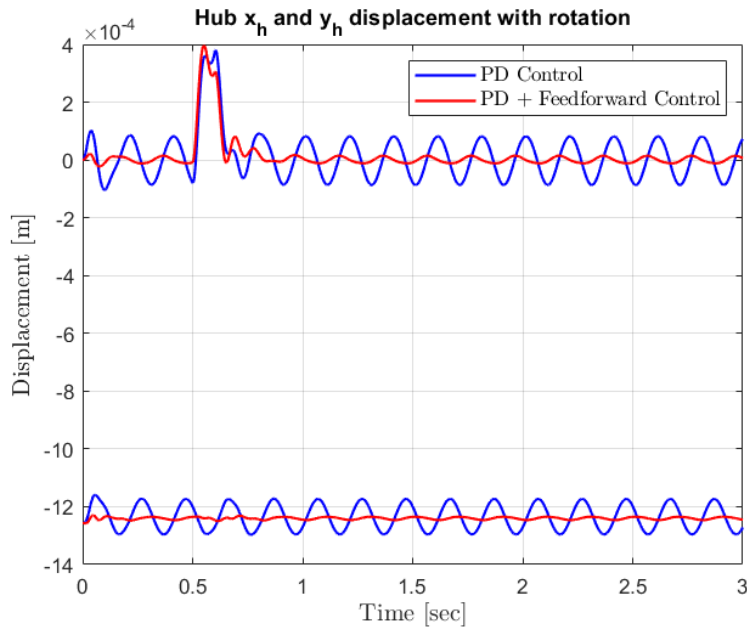
Figure 6.5: Hub Displacement Control with Rotation
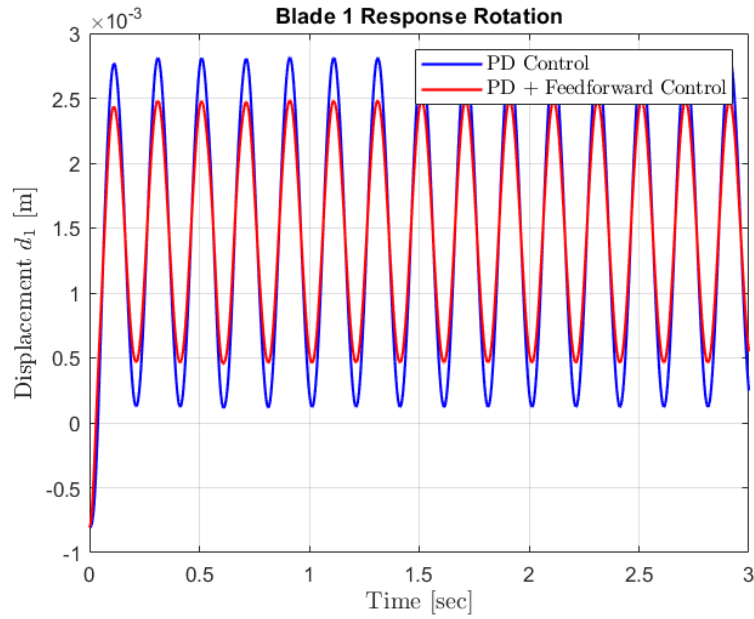
Figure 6.6: Blade 1 Displacement Control with Rotation

We can see the addition of feedforward control drastically reduces vibrations in the hub, it also compensates unknown disturbances. It has less impact in the blades vibration. This is due to the fact that the blade's behaviour in rotation is more affected by system's matrices parameters change, not only by force disturbances primarily. The control signal magnitude is under 20 V.
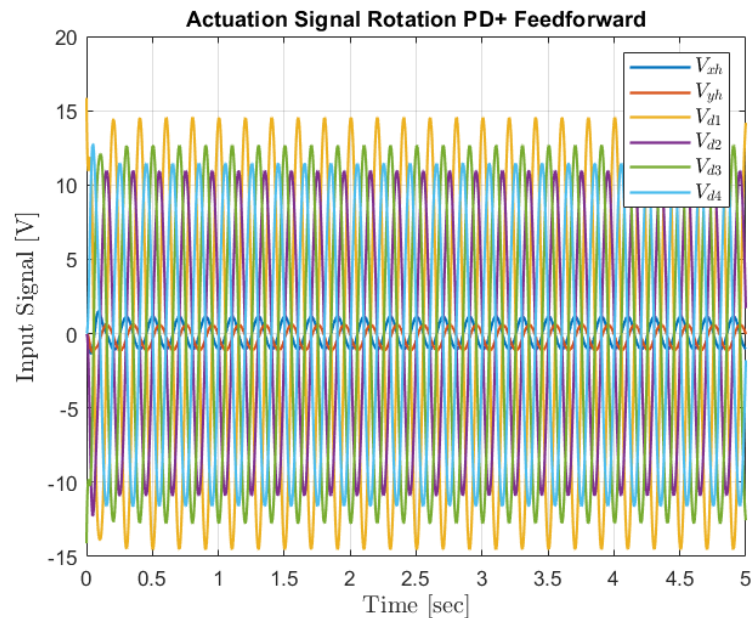
Figure 6.7: Actuation Signal with Rotation

7   Conclusion

This thesis was devised as a theoretically driven investigation of the application of classical control techniques to mitigate rotor blade vibrations. However, it has been carried out with the fundamental idea of ensuring applicability of the designed control algorithms. The main objective was to design simple controllers with suboptimal, but still good performances, that can be physically implemented in the current test-rig control unit.

The starting point of the thesis was to update the mathematical model of the mechanical rotor-blade system plant to take into account the actuation dynamics, which have a great impact in the test-rig behaviour. By including these dynamics an accurate model describing the test-rig's behaviour was obtained.

The next step was to design simple classical control algorithms to reduce system's output measurements vibrations for the static plant (without rotation) with shaft and individual blade actuation. Individual, uncoupled PD controllers were designed to mitigate disturbances, the advantage of these controllers is their simple design which make them easy to tune. They are successful at reducing vibrations, they reduce settling time in a 75 %, however they produce initial overshoots in the outputs of 33% when cancelling force disturbances. To take into account mechanical coupling between the states in the rotor-blade systems two full-state feedback controllers were designed. The first one was designed with Robust Eigenstructure Assigment. In this controller, the feedback gain was difficult to tune to avoid big input signals, saturation had to be added. This controller is able to reduce settling time in 50 %, moreover, it does not produce overshoot. The second controller was designed with optimal control (LQR). With this control algorithm control input magnitude was easy to tune to avoid saturation. This controller is able to reduce settling time in 80% without creating too much overshoot.

The above designed static controllers were modified and tested with shaft based actuation control only. All controllers were able to reduce vibrations in blades with disturbances applied in the hub. PD controllers, due to their decoupling are not able to reduce direct disturbances acting on blades. Full-state feedback controllers, specially LQR, are able to reduce vibrations slightly, their effect is however limited. This is due to the fact that mechanical couplings from blades to shaft is not very noticeable, with an incrementation in the blade's tip mass this effect may be increased, favouring the cancellation of blade direct disturbances. All controllers have good performance when cancelling momentary force disturbances, however an integral action is added to the LQR controller to compensate constant force disturbances.

The study of the adaptation of the static designed controllers was then carried out. Full-state feedback controllers had to be discarded as they introduced too aggressive control signals. The simpler uncoupled PD's produced better results at rotating vibrations

cancellation. To adapt the controller, feedforward control of the known internal force disturbance was added. This controller drastically reduces vibrations amplitude (in a 93%) due to rotation and unknown disturbances in the hub, in both x and y directions. In blades vibrations is reduced in 22%. This is a positive result as hub vibrations produce the most mechanical damage, as it is in contact with the mechanical bearings. This solution produces a meaningful impact, without heavy computation, it is possible to apply on the test rig's control unit.

The different control algorithms designed in the thesis using Matlab can be consulted in Appendix B.

7.1 Application and Future Work

After the completion of the thesis we achieve the design of classical control algorithms that can be experimentally applied on the test-rig. The control structures here designed need to be transformed into C language, that can be interpreted by the test-rig's control unit. The experimental application of the controllers could not be implemented during this time period due to time constrains.

During this semester another DTU student working on the test-rig Ignacio Escudero, was able to experimentally identify process and measurement noise of the model. A possible upgrade that could be to update the deterministic observer for a statistical one, Kalman Filter, using this data. This observer will increase estimation accuracy.

REFERENCES

[1] Christensen R.H. *Active vibration control of rotor-blade systems, theory and experiment.* PhD thesis, Technical University of Denmark, 2004.

[2] Osami Matsushita, Masato Tanaka, Hiroshi Kanki, Masao Kobayashi Patrick Keogh. *Vibrations of Rotating Machinery.* Springer Japan, (2015).

[3] Hossain, M.A., and M.O. Tokhi. *"Evolutionary Adaptive Active Vibration Control."*Proceedings of the Institution of Mechanical Engineers. Part I, Journal of Systems and Control Engineering 211.13 (1997): 183–193. Print.

[4] Fuller, Christopher. *"Active Vibration Control."* Handbook of Noise and Vibration Control, 770–784. Web, 2008.

[5] Kazuto Seto. *"Active Vibration Control in Machinery."* Journal of the Acoustical Society of Japan (e) 12.6 (1991): 263–72, 263–272. Web.

[6] Wang, DA, and YM Huang. *"Modal Space Vibration Control of a Beam by Using the Feedforward and Feedback Control Loops."* International Journal of Mechanical Sciences 44.1 (2002): 1–19. Web.

[7] F. R. Gomaa, K. M. Khader, M. A. Eissa. *"Fault Diagnosis of Rotating Machinery based on vibration analysis."* International Journal of Advanced Engineering and Global Technology I Vol-04, Issue-01, January 2016.

[8] Santos, IF, CM Saracho, JT Smith, and J Eiland. *"Contribution to Experimental Validation of Linear and Non-Linear Dynamic Models for Representing Rotor-Blade Parametric Coupled Vibrations."* Journal of Sound and Vibration 271.3-5 (2004): 883–904. Web.

[9] Zhang, JX, and CS Zhang. *"The Performance of Discrete Linear Time Varying Control of Linear Periodic Plants."* Asian Journal of Control 6.1 (2004): 9–20. Web.

[10] Christensen, Rene Hardam, and Ilmar Santos. *"Design of Active Controlled Rotor-Blade Systems Based on Time-Variant Modal Analysis."* Journal of Sound and Vibration 280.3-5 (2005): 863–882. Web.

[11] Hardam Christensen, René, and Ilmar Ferreira Santos. *"Modal Vibration Control in Periodic Time-Varying Structures with focus on Rotor-Blade Systems"* (2008): n. pag. Web.

[12] Anderson, G. L. *"On the Extensional and Flexural Vibrations of Rotating Bars."* International Journal of Non-linear Mechanics 10.5 (1975): 223–36, 223–236. Web.

[13] Balas, Mark J. *"Feedback Control of Flexible Systems"*Ieee Trans Autom Control AC-23.4 (1978): 673–679. Print.

[14] Bonfitto, Angelo et al. *"Self-Sensing Active Magnetic Dampers for Vibration Control."* Journal of Dynamic Systems Measurement and Control-transactions of the Asme 131.6 (2009): 1–7. Web.

[15] Khulief, Y. A. *"Vibration Suppression in Rotating Beams Using Active Modal Control."* Journal of Sound and Vibration 242.4 (2001): 681–699. Web.

[16] Bittanti, S, and FA Cuzzola. *"Periodic Active Control of Vibrations in Helicopters: a Gain-Scheduled Multi-Objective Approach."* Control Engineering Practice 10.10 (2002): 1043–1057. Web.

[17] Jakobsen, Christian Sidelmann. *"Shaft Based Attenuation of Coupled Rotor-Blade Vibrations via Gain-Scheduled Controlled. "* Master's Thesis, Mechanical Department, Technical University of Denmark, 2012.

[18] Hansen, Jesper Berg. *"Contiuous-time Periodic H_2 and H_∞ Control of a Rotor-Blade System."* Master's Thesis, Mechanical Department, Technical University of Denmark, 2013.

[19] Pulsotronic GmbH & Co. *Sensor KJ4-M12MN50-ANU* $https : //www.pulsotronic.de/index.php?option = com_j proofpulsotronicview = articlecategory_i d = 18art_i d = 1095lang = enprint = 1$

[20] Raspberry Pi Foundation *Raspberry Pi 3 Model B+* $https : //static.raspberrypi.org/files/product - briefs/Raspberry - Pi - Model - Bplus - Product - Brief.pdf$

[21] Meirovitch, L. *Elements of Vibration Analysis.* McGraw-Hill, 1975. Print.

[22] Junkins, J.L., Y. Kim, and American institute of aeronautics. *Introduction to Dynamics and Control of Flexible Structures.* 1993. Print.

[23] Hendricks, Elbert, Ole Jannerup, and Paul Haase SÃ¸rensen. *Linear Systems Control: Deterministic and Stochastic Methods.* (2008): 1–555. Web.

[24] R. E. Kalman. *"A new approach to linear filtering and prediction problems".* Journal of Fluids Engineering vol. 82 no. 1 pp. 35-45 1960.

A MATHEMATICAL MODELLING

A.1 Blade Numerical Mode Shape Equations

1rst Mode Shape Equation

$$\phi_{i,1}(x_i) = a_1 x_i + a_2 x_i^2 + a_3 x_i^3 \tag{A.1}$$

Where :

$$a_1 = \quad 8.653867 \cdot 10^{-3}$$
$$a_2 = \quad 2.160229 \cdot 10^2$$
$$a_3 = \quad -7.595265 \cdot 10^2$$

2nd Mode Shape Equation

$$\phi_{i,2}(x_i) = b_1 x_i + b_2 x_i^2 + b_3 x_i^3 + b_4 x_i^4 \tag{A.2}$$

Where :

$$b_1 = \quad 1.567485$$
$$b_2 = \quad -1.766271 \cdot 10^3$$
$$b_3 = \quad 2.524826 \cdot 10^4$$
$$b_4 = \quad -6.698016 \cdot 10^4$$

A.2 Lagrange Energy Terms

Kinetic Energy

Kinetic Energy Hub:

$$E_{kin,h}(t) = \frac{1}{2} m_{hx} \dot{q}_1^2(t) + \frac{1}{2} m_{hy} \dot{q}_2^2(t) \tag{A.3}$$

Kinetic Energy rotor:

$$E_{kin,r}(t) = \frac{1}{2} J_r (w_{B_1}^I)^T \cdot w_{B_1}^I \tag{A.4}$$

Blades Kinetic Energy:

$$E_{kin,b} = \sum_{i=1}^{4} \frac{1}{2} \rho_i \int_0^{L_{bi}} (V_{op_i}^I(x_i,t))^T \cdot V_{op_i}^I(x_i,t) dx_i \tag{A.5}$$

Blade's Tip Masses Kinetic Energy:

$$E_{kin,t} = \sum_{i=1}^{T} \left(\frac{1}{2} m_{ti} (V_{ot_i}^I)^T \cdot V_{ot_i}^I + \frac{1}{2} J_{ti} (w_{C_i}^I)^T \cdot w_{C_i}^I \right) \tag{A.6}$$

Total kinetic energy is given by the sum of all terms.

Potential Energy

Hub potential energy:

$$E_{pot,h}(t) = \frac{1}{2} k_{hx} q_1^2(t) + \frac{1}{2} k_{hy} q_2^2(t) \tag{A.7}$$

Potential Energy in blade deformation:

$$E_{pot,b}(t) = \sum_{i=1}^{4} \int_0^{L_{bi}} E_{bi} I_{bi} \left(\frac{\delta^2 y_i(x_i,t)}{\delta x_i^2} \right)^2 dx_i \tag{A.8}$$

Potential Energy due to centrifugal effects:

$$E_{pot,c} = \sum_{i=1}^{4} E_{bi} I_{bi} \int_0^{L_{bi}} N_{bi}(x_i) \left(\frac{\delta y_i(x_i,t)}{\delta x_i} \right)^2 dx_i \tag{A.9}$$

Where N_{bi} is the normal force acting on the blades:

$$N_{bi}(x_i) = \left(m_{ti} \left(R + L_{bi} + \frac{1}{2} L_{ti} \right) + \rho_{bi}(L_{bi} - x_i) \left(R + \frac{1}{2}(L_{bi} + x_i) \right) \right) \Omega^2 \tag{A.10}$$

The total potential energy is the sum of all terms.

A.3 Mathematical Actuator Model

$$A_i = \begin{bmatrix} -R_s/L_s & 0 & 0 & 0 & 0 & 0 \\ 0 & -R_s/L_s & 0 & 0 & 0 & 0 \\ 0 & 0 & -R_r/L_r & 0 & 0 & 0 \\ 0 & 0 & 0 & -R_r/L_r & 0 & 0 \\ 0 & 0 & 0 & 0 & -R_r/L_r & 0 \\ 0 & 0 & 0 & 0 & 0 & -R_r/L_r \end{bmatrix} \tag{A.11}$$

$$B1_i = \begin{bmatrix} ki_s/L_s & 0 & 0 & 0 & 0 & 0 \\ 0 & ki_s/L_s & 0 & 0 & 0 & 0 \\ 0 & 0 & ki_r/L_r & 0 & 0 & 0 \\ 0 & 0 & 0 & ki_r/L_r & 0 & 0 \\ 0 & 0 & 0 & 0 & ki_r/L_r & 0 \\ 0 & 0 & 0 & 0 & 0 & ki_r/L_r \end{bmatrix} \tag{A.12}$$

$$B2_i = \begin{bmatrix} 1/L_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/L_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/L_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/L_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/L_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/L_r \end{bmatrix} \tag{A.13}$$

$$C1_i = \begin{bmatrix} ki_s & 0 & 0 & 0 & 0 & 0 \\ 0 & ki_s & 0 & 0 & 0 & 0 \\ 0 & 0 & ki_r & 0 & 0 & 0 \\ 0 & 0 & 0 & ki_r & 0 & 0 \\ 0 & 0 & 0 & 0 & ki_r & 0 \\ 0 & 0 & 0 & 0 & 0 & ki_r \end{bmatrix} \tag{A.14}$$

$$C2_i = \begin{bmatrix} ks_s & 0 & 0 & 0 & 0 & 0 \\ 0 & ks_s & 0 & 0 & 0 & 0 \\ 0 & 0 & ks_r & 0 & 0 & 0 \\ 0 & 0 & 0 & ks_r & 0 & 0 \\ 0 & 0 & 0 & 0 & ks_r & 0 \\ 0 & 0 & 0 & 0 & 0 & ks_r \end{bmatrix} \tag{A.15}$$

B CONTROLLER DESIGN

B.1 PD transfer functions

Individual PD transfer functions in continuous time:

$$PD_1 = 189.29(s + 55.57) \tag{B.1}$$
$$PD_2 = 149.65(s + 82.47) \tag{B.2}$$
$$PD_3 = 101.12(s + 94) \tag{B.3}$$
$$PD_4 = 80(s + 100) \tag{B.4}$$
$$PD_5 = 85.4(s + 104.5) \tag{B.5}$$
$$PD_6 = 82.3(s + 101) \tag{B.6}$$

B.2 LQR Weighting Matrices

States weighting matrix:

$$R_1 = (C_d^T C_d)3 \cdot 10^4 \tag{B.7}$$

Where C_d is the state space discrete output matrix.

Input weighting matrix:

$$R_2 = diag(0.2, 0.2, 0.7, 0.7, 0.7, 0.7) \cdot 1.2^{-4} \tag{B.8}$$

C Matlab Code

Here the Matlab code used to design the controllers can be consulted.

Plant Model:

```matlab
1  %%%%%%%%%%%%%%%%%%%% Main file to obtain the plant %%%%%%%%%%%%%%%%%%%%
2  %Global variables
3  global alpha1 alpha2 alpha3 alpha4 e mr ephi Phis Phim simPlant statesCon
        theta_ini
4
5  %Time step
6  Ts=0.0003; %s
7  % Ts=0.005; %s
8  %Number of states for the internal Simulink block
9  nostates = 10;
10
11 %Numerical matrices for the model
12 NumericalModelValues
13
14 %Linearized magnetic constants
15 R_r = 180; % [Ohm]
16 L_r = 4000e-3; % [H]
17 R_s = 39; % [Ohm]
18 L_s = 1500e-3; % [H]
19
20 ki_r = 50;
21 % ki_r = 10;
22 ks_r = -1e3;
23 % ks_r = -2e3;
24 % ki_s = 112;
25 ki_s =130;
26 % ks_s = -7.84e3;
27 ks_s = -6e3;
28
29 A_i=[-R_s/L_s 0 0 0 0 0; 0 -R_s/L_s 0 0 0 0; 0 0 -R_r/L_r 0 0 0; 0 0 0 -R_r/L_r
        0 0; 0 0 0 0 -R_r/L_r 0;
30     0 0 0 0 0 -R_r/L_r];
31 B1_i=[ki_s/L_s 0 0 0 0 0; 0 ki_s/L_s 0 0 0 0; 0 0 ki_r/L_r 0 0 0; 0 0 0 ki_r/L_r
        0 0; 0 0 0 0 ki_r/L_r 0;
32     0 0 0 0 0 ki_r/L_r];
33 B2_i=[1/L_s 0 0 0 0 0; 0 1/L_s 0 0 0 0; 0 0 1/L_r 0 0 0; 0 0 0 1/L_r 0 0; 0 0 0
        0 1/L_r 0;
34     0 0 0 0 0 1/L_r];
35 C1_i=[ki_s 0 0 0 0 0; 0 ki_s 0 0 0 0; 0 0 ki_r 0 0 0; 0 0 0 ki_r 0 0; 0 0 0 0
        ki_r 0;
```

72

```matlab
36       0 0 0 0 0 ki_r];
37  C2_i=[ks_s 0 0 0 0 0; 0 ks_s 0 0 0 0; 0 0 ks_r 0 0 0; 0 0 0 ks_r 0 0; 0 0 0 0
         ks_r 0;
38       0 0 0 0 0 ks_r];
39
40  %Values of the angles of every blade for the mode shape matrices
41  alpha1 = 0; %rad
42  alpha2 = pi/2; %rad
43  alpha3 = pi; %rad
44  alpha4 = 3*pi/2; %rad
45
46  %Eccentricity values for the displaced center of gravity of the rotor
47  e = 1e-03; %m
48  mr = 3; %kg
49  ephi = 0; %rad
50
51  %Mode shapes vectors to transform
52  a3=-7.595265031870808e+002;
53  a2=2.169022899342803e+002;
54  a1=8.653867145542855e-003;
55  b4=-6.698016635431316e+004;
56  b3=2.524826002857015e+004;
57  b2=-1.766271326563018e+003;
58  b1=1.567485111255101e+000;
59
60  %Mode shapes of the sensors, to convert from modal to real system
61  xs = 0.08; %Distance of the sensor to the rotor, in cm
62  phi1s = a1*xs + a2*xs^2 + a3*xs^3;
63  phi2s = b1*xs + b2*xs^2 + b3*xs^3 + b4*xs^4;
64  Phis = [phi1s phi2s];
65  Mphis=[1 0 0 0 0 0 0 0 0 0;
66       0 1 0 0 0 0 0 0 0 0;
67       0 0 Phis 0 0 0 0 0;
68       0 0 0 0 Phis 0 0 0;
69       0 0 0 0 0 0 Phis 0 0;
70       0 0 0 0 0 0 0 0 Phis];
71  %Mode shapes of the actuators, to convert from real to modal system
72  xm = 0.040; %Distance of the actuator to the rotor, in cm
73  phi1m = a1*xm + a2*xm^2 + a3*xm^3;
74  phi2m = b1*xm + b2*xm^2 + b3*xm^3 + b4*xm^4;
75  Phim = [phi1m phi2m];
76  Mphim=[1 0 0 0 0 0 0 0 0 0;
77       0 1 0 0 0 0 0 0 0 0;
78       0 0 Phim 0 0 0 0 0;
79       0 0 0 0 Phim 0 0 0;
80       0 0 0 0 0 0 Phim 0 0;
81       0 0 0 0 0 0 0 0 Phim];
82  Mphim=Mphim';
83
84  %Modeling the linear sensor and the ADC conversion
85  K_adc = 32768/4.096; %bits/V
86  K_distance = 2/(1.5*10^-3); %V/m
```

```matlab
87  K_sensor = K_adc*K_distance; %bits/m
88
89  %Initial conditions for modal states
90  xxd_ini = zeros(10,1);
91  xx_ini = InitialCond(theta_ini);
92
93  %Simulating the plant in open loop for the actuator forces specified in
94  %Simulink model
95  if simPlant==1
96      sim('PlantSim');
97      figure
98      plot(tout,x)
99      title('Displacement of hub in direction x')
100     xlabel('t(s)')
101     ylabel('xh(m)')
102     figure
103     plot(tout,y)
104     title('Displacement of hub in direction y')
105     xlabel('t(s)')
106     ylabel('yh(m)')
107     figure
108     plot(tout,d1)
109     title('Displacement blade 1')
110     xlabel('t(s)')
111     ylabel('d1(m)')
112     figure
113     plot(tout,d2)
114     title('Displacement blade 2')
115     xlabel('t(s)')
116     ylabel('d2(m)')
117     figure
118     plot(tout,d3)
119     title('Displacement blade 3')
120     xlabel('t(s)')
121     ylabel('d3(m)')
122     figure
123     plot(tout,d4)
124     title('Displacement blade 4')
125     xlabel('t(s)')
126     ylabel('d4(m)')
127 end
128
129 %Obtaining the transfer functions of the plant and the state space model
130 if statesCon==6
131
132     %Find steady state output and state values with zero input
133     simopts = simset('FinalStateName','xFinal'); % set the sim function options
                to obtain the final state values as xFinal
134     [t,x,y] = sim('SteadyState',50,simopts); % simulate for 5000 seconds to
                obtain states steady state
135     xf = xFinal;
136     xx_ini=xf(1:10)';
```

```matlab
137        xxd_ini=xf(17:26)';
138        xxi_ini=xf(11:16)';
139        xx_ini_disp=[xx_ini(1:2);Phis*xx_ini(3:4);;Phis*xx_ini(5:6);Phis*xx_ini(7:8);Phis*xx_ini(9:
140
141        %Internal forces with no rotation
142        f_ini=Fmatrix10([0 0 0]);
143
144        %Find trim point for operating point
145        y_trim = [xx_ini(1:2);zeros(4,1)]; %Desired output operating point
146        u_trim = [0,0,-15.1,0,15.1,0,f_ini]';
147
148        [x_0,u_0,y_0,dx] = trim('PlantCon6',[],u_trim,y_trim,[],[],[1:1:6]);
149
150     % Linearize at operating point
151
152        sys = linmod('PlantCon6',x_0,u_0);
153
154        Ac = sys.a;
155        Bc = sys.b;
156        Cc = sys.c;
157        Dc = sys.d;
158        sys_ss = c2d(ss(Ac,Bc,Cc,Dc),Ts,'zoh');
159        sys_tf= tf(sys_ss);
160
161  end
162
163  if statesCon==10
164        %Find steady state output and state values with zero input
165        simopts = simset('FinalStateName','xFinal'); % set the sim function options
               to obtain the final state values as xFinal
166        [t,x,y] = sim('SteadyState',50,simopts); % simulate for 5000 seconds to
               obtain states steady state
167        xf = xFinal;
168        xx_ini=xf(1:10)';
169        xxd_ini=xf(17:26)';
170        xxi_ini=xf(11:16)';
171        %xx_ini_disp=[xx_ini(1:2);Phis*xx_ini(3:4);;Phis*xx_ini(5:6);Phis*xx_ini(7:8);Phis*xx_ini(9
172
173        %Internal forces with no rotation
174        f_ini=Fmatrix10([0 0 0]);
175
176        %Find trim point for operating point
177        y_trim = [xx_ini(1:2);zeros(4,1)]; %Desired output operating point
178        u_trim = [0,0,-11.14,0,11.14,0,f_ini]';
179
180        [x_0,u_0,y_0,dx] = trim('PlantCon10',[],u_trim,y_trim,[],[],[1:1:6]);
181
182     % Linearize at operating point
183
184        sys = linmod('PlantCon10',x_0,u_0);
185
186        Ac = sys.a;
```

```matlab
187     Bc = sys.b;
188     Cc = sys.c;
189     Dc = sys.d;
190     sys_ss = c2d(ss(Ac,Bc,Cc,Dc),Ts,'zoh');
191     sys_tf= tf(sys_ss);
192 end
```

Input-Output Transfer Function Order Reduction:

```matlab
1  %% Continuous System
2  sys_ss_c=ss(Ac,Bc,Cc,Dc);
3  sys_tf_c=tf(sys_ss_c);
4  %% Continuous transfer functions
5
6  P1 = sys_tf_c(1,1);
7  P2 = sys_tf_c(2,2);
8  P3 = sys_tf_c(3,3);
9  P4 = sys_tf_c(4,4);
10 P5 = sys_tf_c(5,5);
11 P6 = sys_tf_c(6,6);
12
13 %% Transfer Function 1
14 s=tf('s');
15 P1zpk=zpk(P1)
16 KP1=dcgain(P1)
17 %%
18 gain_1=-8.0587;
19 z1_1=(s+40.69)^4;
20 z2_1=(s+22.63);
21 z3_1=(s^2 + 4.35*s + 8377);
22 z4_1=(s^2 + 9.103*s + 1.122e04);
23 z5_1=(s^2 + 9.069*s + 1.133e04);
24 z6_1=(s^2 + 9.035*s + 1.143e04);
25 z7_1=(s^2 + 8.388*s + 1.244e04);
26 z8_1=(s^2 + 22*s + 3.24e06)^4;
27 p1_1=(s+40.69)^4 ;
28 p2_1=(s+22.63);
29 p3_1=(s+22.18);
30 p4_1=(s^2 + 4.058*s + 6238);
31 p5_1=(s^2 + 4.35*s + 8377);
32 p6_1=(s^2 + 9.079*s + 1.131e04) ;
33 p7_1=(s^2 + 9.069*s + 1.133e04) ;
34 p8_1=(s^2 + 9.04*s + 1.189e04);
35 p9_1=(s^2 + 8.388*s + 1.244e04);
36 p10_1=(s^2 + 22*s + 3.24e06)^4;
37 %%
38 P1red=((gain_1))/(p3_1*p4_1);
39 Kp1red=dcgain(P1red);
40 P1red=P1red*KP1/Kp1red; %Reduced Model
41
42 step(P1red,'r')
43 hold on
```

```matlab
44  step(P1,'b')
45  hold off
46  figure;
47  impulse(P1red,'r')
48  hold on
49  impulse(P1,'b')
50  hold off
51  %% Transfer Function 2
52  s=tf('s');
53  P2zpk=zpk(P2)
54  KP2=dcgain(P2); %DC gain
55  %%
56  gain_2=-9.7883 ;
57  z1_2=(s+40.69)^4;
58  z2_2=(s+22.18);
59  z3_2=(s^2 + 4.058*s + 6238);
60  z4_2=(s^2 + 9.072*s + 1.132e04)^3 ;
61  z5_2=(s^2 + 9.04*s + 1.189e04);
62  z6_2= (s^2 + 22*s + 3.24e06)^4;
63  p1_2=(s+40.69)^4 ;
64  p2_2=(s+22.63) ;
65  p3_2=(s+22.18);
66  p4_2=(s^2 + 4.058*s + 6238);
67  p5_2=(s^2 + 4.35*s + 8377);
68  p6_2=(s^2 + 9.079*s + 1.131e04);
69  p7_2=(s^2 + 9.069*s + 1.133e04);
70  p8_2=(s^2 + 9.04*s + 1.189e04);
71  p9_2=(s^2 + 8.388*s + 1.244e04);
72  p10_2=(s^2 + 22*s + 3.24e06)^4;
73  %%
74  P2red=((gain_2))/(p3_2*p5_2);
75  Kp2red=dcgain(P2red);
76  P2red=P2red*KP2/Kp2red; %Reduced Model
77
78  step(P2red,'r')
79  hold on
80  step(P2,'b')
81  hold off
82  figure;
83  impulse(P2red,'r')
84  hold on
85  impulse(P2,'b')
86  hold off
87  %% Transfer Function 3
88  s=tf('s');
89  P3zpk=zpk(P3)
90  KP3=dcgain(P3) %DC gain
91
92  opt = balredOptions('Offset',.001,'StateElimMethod','Truncate');
93  r3sys = balred(P3,3,opt)
94  dcgain3=dcgain(r3sys)
95  r3sys=r3sys*KP3/dcgain3 %Reduced Model
```

```matlab
96
97   step(r3sys,'r')
98   hold on
99   step(P3,'b')
100  hold off
101  figure;
102  impulse(r3sys,'r')
103  hold on
104  impulse(P3,'b')
105  hold off
106
107
108  %% Transfer Function 4
109
110  s=tf('s');
111  P4zpk=zpk(P4)
112  KP4=dcgain(P4) %DC gain
113
114  opt = balredOptions('Offset',.001,'StateElimMethod','Truncate');
115  r4sys = balred(P4,3,opt)
116  dcgain4=dcgain(r4sys)
117  r4sys=r4sys*KP4/dcgain4 %Reduced Model
118
119  step(r4sys,'r')
120  hold on
121  step(P4,'b')
122  hold off
123  figure;
124  impulse(r4sys,'r')
125  hold on
126  impulse(P4,'b')
127  hold off
128
129  %% Transfer Function 5
130  s=tf('s');
131  P5zpk=zpk(P5)
132  KP5=dcgain(P5)%DC gain
133
134  opt = balredOptions('Offset',.001,'StateElimMethod','Truncate');
135  r5sys = balred(P5,3,opt)
136  dcgain5=dcgain(r5sys)
137  r5sys=r5sys*KP5/dcgain5 %Reduced Model
138
139  step(r5sys,'r')
140  hold on
141  step(P5,'b')
142  hold off
143  figure;
144  impulse(r5sys,'r')
145  hold on
146  impulse(P5,'b')
147  hold off
```

```matlab
148
149 %% Transfer Function 6
150 s=tf('s');
151 P6zpk=zpk(P6)
152 KP6=dcgain(P6) %DC gain
153
154 opt = balredOptions('Offset',.001,'StateElimMethod','Truncate');
155 r6sys = balred(P6,3,opt)
156 dcgain6=dcgain(r6sys)
157 r6sys=r6sys*KP6/dcgain6 %Reduced Model
158
159 step(r6sys,'r')
160 hold on
161 step(P6,'b')
162 hold off
163 figure;
164 impulse(r6sys,'r')
165 hold on
166 impulse(P6,'b')
167 hold off
```

Full-state Feedback Robust Eigenstructure Assigment:

```matlab
1 %%%%%%%%%% Main file for the State Space Feedback controller%%%%%%%%%%%%%%%
2 clear
3 clc
4
5 global simPlant statesCon theta_ini theta_d_ini
6
7 %Simulating the plant for the state space feedback controller
8 simPlant = 0;
9 simSS = 0;
10 statesCon = 6; %number of states controlled
11 theta_ini = 0; %rad
12 theta_d_ini = 0; %rad/s
13 if statesCon== 10
14     switch_no=-1;
15 else switch_no=1;
16 end
17
18 %Obtaining the plant state space description, parameters specified in Plant.m
19 Plant
20
21 Ad = sys_ss.A;
22 Bd = sys_ss.B;
23 Cd = sys_ss.C;
24 Dd = sys_ss.D;
25
26 %Check stability
27 eig_cont=eig(Ac);
28 eig_cont=[eig_cont(9);eig_cont(16);eig_cont(12);eig_cont(1);eig_cont(14);eig_cont(3);eig_cont(1
29         eig_cont(11);eig_cont(24);eig_cont(22);eig_cont(23);eig_cont(25);eig_cont(26);
```

```matlab
30            eig_cont(10);eig_cont(17);eig_cont(13);eig_cont(2);eig_cont(15);eig_cont(4);eig_cont(19);eig
31 eig_dis=eig(Ad);
32
33 %% OBSERVER
34 %Since we do not have the full state information, observer
35 %design is needed to estimate the speed of the modal displacements
36
37 obs_eig = 10*eig_cont;
38 obs_eig_d = exp(Ts*obs_eig);
39 L_trans = place(Ad',Cd',obs_eig_d);
40 L = L_trans';
41
42 %% OBSERVER INITIAL STATE
43  % set the sim function options to obtain the final state values as xFinal
44 [to,xo,yo] = sim('Observer_Steady',10,simopts); % simulate for 5000 seconds to
        optain steady state
45 xobs=yo(end,:)';
46
47 %% PROPORTIONAL REGULATOR
48 %Design of the proportional regulator for the desired poles
49 %Stator and rotor control
50
51 eig_desired=[real(eig_cont)*1.5 + imag(eig_cont)*1*i]; %rot
52 poles_con_d=exp(Ts*eig_desired);
53 K_cp = place(Ad,Bd(:,1:2),poles_con_d);
54
55 K_cp=[K_cp(1:2,:);zeros(4,26)];
56 %% PROPORTIONAL REGULATOR + INTEGRAL ACTION
57 %With the whole state, the control is designed, an integral action is
58 %needed since the disturbance force (F_int, dependent of theta, theta_d and
59 %theta_dd) needs to be taken into account
60 A_int = [Ac zeros(26,6);-Cc eye(6)];
61 B_int = [Bc(:,1:6);zeros(6,6)];
62 poles_int = eig(A_int);
63 con_eig_int = [real(poles_int)*3 + imag(poles_int)/2*1i];
64 con_eig_d_int =exp(Ts*con_eig_int);
65 A_int_d = [Ad zeros(26,6);-Cd eye(6)];
66
67
68 B_int_d = [Bd(:,1:6);zeros(6,6)];
69 K_tot = place(A_int_d,B_int_d,con_eig_d_int);
70 K_ci = K_tot(:,1:26);
71 K_hi = K_tot(:,27:32);
```

Full-state Feedback LQR:

```matlab
1 %%%%%%%%%% Main file for the State Space Feedback controller%%%%%%%%%%%%%%
2 clear all
3 clc
4
5 global simPlant statesCon theta_ini theta_d_ini
6
```

```matlab
 7  %Simulating the plant for the state space feedback controller
 8  simPlant = 0;
 9  simSS = 0;
10  statesCon = 6; %number of states controlled
11  theta_ini = 0; %rad
12  theta_d_ini = 0; %rad/s
13
14  %Obtaining the plant state space description, parameters specified in Plant.m
15  Plant
16
17  Ad = sys_ss.A;
18  Bd = sys_ss.B;
19  Cd = sys_ss.C;
20  Dd = sys_ss.D;
21
22  %Check stability
23  eig_cont=eig(Ac);
24  eig_dis=eig(Ad);
25  %% OBSERVER
26  %Since we do not have the full state information, observer
27  %design is needed to estimate the speed of the modal displacements
28
29  obs_eig = 10*eig_cont;
30  obs_eig_d = exp(Ts*obs_eig);
31  L_trans = place(Ad',Cd',obs_eig_d);
32  L = L_trans';
33
34  %% OBSERVER INITIAL STATE
35   % set the sim function options to obtain the final state values as xFinal
36  [to,xo,yo] = sim('Observer_Steady',10,simopts); % simulate for 5000 seconds to
        optain steady state
37  xobs=yo(end,:)';
38
39  %% DLQR
40
41  %Design of the proportional regulator for the LQR formula
42  Q_con = Cd'*Cd*100000000;
43  R_con =diag([0.2 , 0.2, 0.7, 0.7, 0.7,0.7])*0.4;
44  K_cp = dlqr(Ad,Bd(:,1:6),Q_con*Ts,R_con*Ts);
45  % R_con =diag([1 , 1])*0.4;
46  % K_cp = dlqr(Ad,Bd(:,1:2),Q_con*Ts,R_con*Ts);
47
48  %% DLQR + Integral
49  %With the whole state, the control is designed, an integral action is
50  %needed since the disturbance force (F_int, dependent of theta, theta_d and
51  %theta_dd) needs to be taken into account
52  A_int_d = [Ad zeros(26,6);-Cd eye(6)];
53  B_int_d = [Bd(:,1:6);zeros(6,6)];
54
55  % Q_int = [Cd'*Cd ,zeros(26,6); zeros(6,26), eye(6)]*100;
56  Q_int = [Cd'*Cd ,zeros(26,6); zeros(6,26), eye(6,6)]*100000;
57  R_int = diag([0.5 , 0.5, 1, 1, 1,1])*2;
```

```matlab
58  K_tot = dlqr(A_int_d,B_int_d,Q_int*Ts,R_int*Ts);
59  K_ci = K_tot(:,1:26);
60  K_hi = -K_tot(:,27:32);
61
62  %% Shaft Actuation Only
63
64  A_int_d = [Ad zeros(26,6);-Cd eye(6)];
65  B_int_d = [Bd(:,1:2);zeros(6,2)];
66
67  % Q_int = [Cd'*Cd ,zeros(26,6); zeros(6,26), eye(6)]*100;
68  Q_int = [Cd'*Cd ,zeros(26,6); zeros(6,26), eye(6,6)]*100000;
69  R_int = diag([0.5 , 0.5])*2;
70  K_tot = dlqr(A_int_d,B_int_d,Q_int*Ts,R_int*Ts);
71  K_ci = K_tot(:,1:26);
72  K_hi = -K_tot(:,27:32);
```

Electro-Magnet Actuator Model Transfer Functions

```matlab
1   sys_magnets = linmod('ImanesPRUEBA');
2   Ac_mag = sys_magnets.a;
3   Bc_mag = sys_magnets.b;
4   Cc_mag = sys_magnets.c;
5   Dc_mag = sys_magnets.d;
6   sys_ss_c_mag=ss(Ac_mag,Bc_mag,Cc_mag,Dc_mag);
7   sys_tf_c_mag=tf(sys_ss_c_mag);
8   sys_ss_mag = c2d(ss(Ac_mag,Bc_mag,Cc_mag,Dc_mag),Ts,'zoh');
9   sys_tf_mag= tf(sys_ss_mag);
10
11  % Transfer functions
12
13  P1mag=sys_tf_c_mag(1,1);
14  P2mag=sys_tf_c_mag(2,2);
15  P31mag=sys_tf_c_mag(3,3);
16  P32mag=sys_tf_c_mag(4,3);
17  P41mag=sys_tf_c_mag(5,4);
18  P42mag=sys_tf_c_mag(6,4);
19  P51mag=sys_tf_c_mag(7,5);
20  P52mag=sys_tf_c_mag(8,5);
21  P61mag=sys_tf_c_mag(9,6);
22  P62mag=sys_tf_c_mag(10,6);
```

Feed-forward Control Gain Transfer Function

```matlab
1   s=tf('s');
2
3   %Inverse Magnet Model
4   invermag1=inv(P1mag);
5   invermag2=inv(P2mag);
6   invermag31=inv(P31mag);
7   invermag32=inv(P32mag);
8   invermag41=inv(P41mag);
```

```matlab
9    invermag42=inv(P42mag);
10   invermag51=inv(P51mag);
11   invermag52=inv(P52mag);
12   invermag61=inv(P61mag);
13   invermag62=inv(P62mag);
14
15   %Adding Filter
16   invermag1=invermag1/(1+0.001*s);
17   invermag2=invermag2/(1+0.001*s);
18   invermag31=invermag31/(1+0.001*s);
19   invermag32=invermag32/(1+0.001*s);
20   invermag41=invermag41/(1+0.001*s);
21   invermag42=invermag42/(1+0.001*s);
22   invermag51=invermag51/(1+0.001*s);
23   invermag52=invermag52/(1+0.001*s);
24   invermag61=invermag61/(1+0.001*s);
25   invermag62=invermag62/(1+0.001*s);
26
27   %Obtaining Transfer Function Num+Den
28   [P1magnum,P1magden]=tfdata(invermag1);
29   [P2magnum,P2magden]=tfdata(invermag2);
30   [P31magnum,P31magden]=tfdata(invermag31);
31   [P32magnum,P32magden]=tfdata(invermag32);
32   [P41magnum,P41magden]=tfdata(invermag41);
33   [P42magnum,P42magden]=tfdata(invermag42);
34   [P51magnum,P51magden]=tfdata(invermag51);
35   [P52magnum,P52magden]=tfdata(invermag52);
36   [P61magnum,P61magden]=tfdata(invermag61);
37   [P62magnum,P62magden]=tfdata(invermag62);
38
39
40   %DC gain s=5
41   %
42   % K1mag=evalfr(P1mag,5);
43   % K2mag=evalfr(P2mag,5);
44   % K31mag=evalfr(P31mag,5);
45   % K32mag=evalfr(P32mag,5);
46   % K41mag=evalfr(P41mag,5);
47   % K42mag=evalfr(P42mag,5);
48   % K51mag=evalfr(P51mag,5);
49   % K52mag=evalfr(P52mag,5);
50   % K61mag=evalfr(P61mag,5);
51   % K62mag=evalfr(P62mag,5);
```