UNIVERSITY OF CALIFORNIA, IRVINE

Practical control design using constrained $\mathcal{H}_{\infty}$

THESIS

submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in Mechanical and Aerospace Engineering

by

Sergio Lopez Lopez

Thesis Committee:
Professor Athanasios Sideris, Chair
Professor Faryar Jabbari
Professor James E. Bobrow

2007
The thesis of Sergio Lopez Lopez is approved:

________________________________________

________________________________________

________________________________________

Committee Chair

University of California, Irvine

2007
to Laura
4.1 Methods ................................................................. 22
  4.1.1 Splitting the controller ........................................... 23
  4.1.2 Lifting technique .................................................. 25
4.2 Controller design ..................................................... 31
  4.2.1 High rate design and controller decomposition ............... 31
  4.2.2 Low rate design of the lifted system ......................... 32
  4.2.3 Low rate design .................................................. 34
  4.2.4 Controllers obtained ............................................ 34
4.3 Testing the controllers ............................................... 36

5 Time domain and $\mathcal{H}_2$ norm constrained $\mathcal{H}_\infty$ 38
  5.1 Methods .............................................................. 39
    5.1.1 Problem formulation ......................................... 39
    5.1.2 Problem parametrization .................................... 40
    5.1.3 Solving the constrained convex optimization problem .... 42
  5.2 Constrained $\mathcal{H}_\infty$ design example ....................... 43

6 Conclusions ............................................................ 47

Bibliography ............................................................. 49
List of Tables

5.1 Results of several designs. ........................................ 45
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Bode plots of all the data sets</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Input and output data of the ANC study</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Frequency response data of the ANC study</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Bode plots of data set $F_1$ and corresponding fits</td>
<td>11</td>
</tr>
<tr>
<td>2.5</td>
<td>Bode plots of data set $F_2$ and corresponding fits</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Bode plots of data set $F_3$ and corresponding fits</td>
<td>12</td>
</tr>
<tr>
<td>2.7</td>
<td>Bode plots of data set $F_4$ and corresponding fits</td>
<td>12</td>
</tr>
<tr>
<td>2.8</td>
<td>Bode plots of the bounds, fits and the mean data</td>
<td>13</td>
</tr>
<tr>
<td>2.9</td>
<td>Bode plots of the middle plant and its corresponding accurate fit.</td>
<td>14</td>
</tr>
<tr>
<td>2.10</td>
<td>Pole zero map for the accurate fit of the middle plant.</td>
<td>14</td>
</tr>
<tr>
<td>2.11</td>
<td>Bode plot of data and fit of the ANC data</td>
<td>15</td>
</tr>
<tr>
<td>2.12</td>
<td>Pole zero map of fit obtained</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>(a) Framework used for the disturbance rejection problem formulation. (b) Obtained framework for the design.</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Bode plot of the results obtained at the $H_2$ design.</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Framework used to formulate the mixed sensitivity $H_\infty$ problem.</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Results from the $H_\infty$ controller design</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Framework used to formulate the mixed sensitivity $H_\infty$ problem</td>
<td>23</td>
</tr>
<tr>
<td>4.2</td>
<td>Series decomposition of the controller.</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Parallel decomposition of the controller.</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>System representation</td>
<td>25</td>
</tr>
<tr>
<td>4.5</td>
<td>Lifted system representation</td>
<td>26</td>
</tr>
</tbody>
</table>
4.6  Bode plot of the $\mathcal{H}_\infty$ controller obtained at high rate. .......... 32
4.7  Bode plot of the high frequency part of the high rate $\mathcal{H}_\infty$ controller obtained. ................................................................. 33
4.8  Bode plot of the low frequency controller for the lifted system. .... 34
4.9  Bode plot of the $\mathcal{H}_\infty$ controller obtained at low rate. .......... 35
4.10 Magnitude plot of the $\mathcal{H}_\infty$ controllers ............................. 35
4.11 Time response of the plant after an impulse disturbance in the measurement has been applied. ................................................................. 36
4.12 Time response of the all the plants ............................................. 37

5.1  Control system configuration. .................................................... 39
5.2  Control system configuration for the constrained problem. ........ 40
5.3  General behavior of the optimal norms. ....................................... 44
5.4  Formulation for the constrained $\mathcal{H}_\infty$ problem. ................... 44
5.5  Disturbance time response of the different optimal controllers. .. 46
Acknowledgments

I would like to express special thanks to my advisor Dr. Athanasios Sideris, for his advise, guidance and help in the robust control field.

In addition, my sincere appreciation to the Balsells Fellowship and Catalonia-California Engineering Innovation Programs for offering me the opportunity to come to the University of California, Irvine. Specially thanks to Professor Roger H. Rangel, for his support and great advice.

I also would like to thank Professor Faryar Jabbari and Professor James E. Bobrow for satisfying the computer needs during along the last two years, and to Dr. Jie Yu and Dr. Ricardo S. Sánchez Peña for their help and advice.

And finally, special thanks to my family and friends, specially to my partner Laura. Thanks for being always there for me. This thesis would not have been possible without your support and patience.
Abstract of the Thesis

Practical control design using constrained $\mathcal{H}_{\infty}$

by

Sergio Lopez Lopez
Master of Science in Mechanical and Aerospace Engineering
University of California, Irvine, 2007
Professor Athanasios Sideris, Chair

A framework for the implementation and design of practical controllers is presented. The methodology used is divided and presented in two stages, identification and design. The results demonstrate efficiency in the identification stage. The controller design is presented from basic optimal control problems to multirate and constrained optimal control problems. The results validate the methodology and implementation proposed.
Chapter 1

Introduction

The implementation and design of practical controllers is a task that can be divided in two stages, identification and design. There exists a need for obtaining a practical framework that make possible to solve faster and more accurately the control design problems found in many industrial, mechanical and aerospace engineering applications.

During the first stage, a model of the system to control is calculated. The desired mathematical model of the system to be controlled can be described and obtained directly by developing the equations governing its dynamics, or more frequently, by curve fitting sampled data. The identification of the model is crucial to advance to the design stage. A well formulated problem needs a well conditioned model to give the desired results. Even an easy control problem design can become impossible to solve if the mathematical model of our system is not the correct one. It is necessary, for example, to check the stability, and complexity (number of states) of the obtained model.

Many techniques can be used to identify the model using sampled data, for instance, robust identification [1]. It has the advantage that its output is a set of models, nominal plus bounded uncertainty that can be used directly in a robust control formulation, but this can result in a high order model that will lead to high order controller.
We propose an algorithm that fits the sampled data by using Chebyshev polynomials, and that has the capability to calculate models with low complexity compared to other identification methods.

Once we have the desired model, we have to start with the design stage. During the design, we need to formulate the problem accommodating the control objectives by adding, if necessary, weights in the inputs and/or outputs. Some of these objectives are formulated in time-domain while others are in the frequency-domain. It is necessary then to translate them to either the time-domain or frequency-domain. This task can lead to lose performance due to the conservatism applied in the translation process.

The control design methodology proposed in this thesis, called Constrained $\mathcal{H}_\infty$, can treat the time-domain constraints explicitly within a frequency-domain problem formulation. The formulation and design of multirate controllers by using $\mathcal{H}_\infty$ controllers is also explored.

The main goals of this Thesis are to develop a unified framework for modeling, analysis, and control design that provides more accurate and faster results than current ones and to apply the previous results in mechanical and aerospace engineering problems. The methodology developed has been tested in two different examples, a Hard Disk Drive (HDD) model, and an Active (acoustic) Noise Control (ANC) design problem.

1.1 Literature Review

The identification of sampled data systems is a hard problem to solve. First of all, it is necessary to find the equilibrium point between complexity of the model and an accurate identification. It is also necessary to check for stability and study the data before starting the fit. Since having a flat magnitude/phase at low frequencies makes the fit easier, it is very useful identify if the data has zeros/poles at the origin (for continuous time systems).

The curve fitting method using Chebyshev polynomials is presented in the chap-
The method was introduced for the SISO case in [2] and extended for the MIMO case in [3]. Other methods for system identification, as instance, robust identification have been used with success, as in [1]. As said before, this method has the advantage of computing two models, the nominal and the bounded uncertainty. This fact is very useful in a robust control formulation. The identified system though can be a high order model.

The proposed framework for the practical controller design studies basically three aspects in the $\mathcal{H}_\infty$ control. These aspects are reflected in chapter 3, 4 and 5 of this thesis. A $\mathcal{H}_2$ optimal controller and a $\mathcal{H}_\infty$ optimal controller are first obtained in chapter 3. We can find synthesis control methods for general problems in [4], a state-space solution in [5], and in [6] we can find detailed information about the Youla parametrization method used to solve the unconstrained $\mathcal{H}_\infty$ problem.

The results obtained in Chapter 3 are analyzed and used to design a multirate $\mathcal{H}_\infty$ controller in Chapter 4 for a Hard Disk Drive (HDD) example. The main goal of a multirate controller is to satisfy the performance/robustness specifications while satisfying also hardware/computational constraints. A HDD is also used in [7] and in [8]. In [7] the design a multirate controller seeks as main goal saving computation effort. The multirate controller is obtained by interlacing (split the original controller in $n$ parts and use only one to compute the control at each sample time) at slow sampling rate. Another approach is used in [8] by implementing a multirate controller that uses an estimator when the measurement is not available.

Chapter 5 focuses the effort in the time domain and $\mathcal{H}_2$ norm constrained $\mathcal{H}_\infty$ controller design. The approach used varies from the usual mixed $\mathcal{H}_2/\mathcal{H}_\infty$ that has been used extensively since multiobjective control problems were first introduced and studied.

The most common control problem design is to find a stabilizing controller that minimizes $\mathcal{H}_2$ norm of the system transfer function while having the $\mathcal{H}_\infty$ norm bounded.

Different solutions and approaches to solve this problem have been studied and proposed. In [9], a state space solution is parameterized and presented. A different approach is used in [10], where the problem is studied for the state-feedback case.
and for the output-feedback case. In both cases, the problem is converted to a convex optimization problem, but without using any convex optimization algorithm to solve any example. In [11], the problem is also solved as a convex optimization problem.

On the other hand, [12] compares the controllers obtained using mixed $\mathcal{H}_2/\mathcal{H}_\infty$ with other synthesis methods such as mixed $\mathcal{H}_2/\mu$ or robust $\mathcal{H}_2$ in a dual-stage Hard Disk Drive. For time-domain $\mathcal{H}_2/\mathcal{H}_\infty$ an exact solution was proposed in [13] by solving an equivalent discrete-time problem.

We will impose time domain and $\mathcal{H}_2$ norm constraints to the $\mathcal{H}_\infty$ norm optimal problem. The algorithm and procedure is the same as used in [6], and in [14]. The main idea is to use the Youla parametrization to reformulate the problem into a convex constrained optimization problem. This problem is then solved using the ellipsoid algorithm as explained in [15], and [16].
Chapter 2

Transfer function curve fitting from data

Modern control theory is based on the knowledge of the mathematical model of the plant to control. For example, in Robust control theory (e.g. $\mathcal{H}_\infty$, $\mu$-synthesis, etc) the mathematical model of the plant is used for the design of a feedback and/or feedforward controller while making the whole system (i.e. plant and controller) to satisfy certain specifications of performance and robustness.

Obtaining a mathematical model of a real plant is never an easy task. In particular, in simplified systems, the model can be obtained by developing the equations of its dynamics. Generally, for complex dynamics (e.g. involving airflows, temperature dependency, etc) finding the equations becomes a hard problem to solve.

Usually, experimental data is used to obtain better and more realistic model of the plant to control. The experimental data is usually transformed into a set of complex numbers that represents the frequency response data of the plant studied. The mathematical model of the plant is obtained by fitting this data.

Two different examples are presented in this chapter; data from a Hard disk drive (HDD) and data from an active noise control (ANC) experiment as used in [1].

Our goal is to obtain a transfer function model that provides a good match with
the data used. In order to achieve this objective, we use an iterative algorithm that uses Chebyshev polynomials and the solution of least squares problems to construct a transfer function that best fits the data. The algorithm for the transfer function curve fitting using Chebyshev polynomials was studied in for the SISO case in [2] and extended for the MIMO case in [3]. Our approach uses the same principle as in [2] and [3], but adds the capacity to construct a state-space realization numerically well conditioned.

2.1 Methods

The problem to solve is to find a transfer function that fits a given frequency response data represented by complex numbers.

Let \( g(s) \) be the transfer function that we are looking for and \( n(s) \) and \( d(s) \) be the numerator and denominator polynomials respectively,

\[
g(s) = \frac{n(s)}{d(s)} = \frac{n_0 + n_1 s + n_2 s^2 + \ldots + n_M s^M}{d_0 + d_1 s + d_2 s^2 + \ldots + d_N s^N}
\]  

(2.1)

The nonlinear equation 2.1 can be linearized by multiplying by the denominator \( d(s) \), obtaining \( g(s)d(s) - n(s) = 0 \) or equivalently \( g(j\omega)d(j\omega) - n(j\omega) = 0 \) for each frequency \( \omega \). Then, the curve fitting problem is transformed into finding the coefficients \( n_i \) and \( d_i \) in the numerator and denominator polynomials that solve the Linear Least Squares problem represented in 2.2.

\[
\min_{n_i,d_i} \sum_{k=1}^{N} W(\omega) \cdot [\text{real}^2 (g(j\omega_k)d(j\omega_k) - n(j\omega_k)) + \text{imag}^2 (g(j\omega_k)d(j\omega_k) - n(j\omega_k))]
\]  

(2.2)

Since we have a considerably large number of frequency points, the problem to solve will involve large matrices. We use an iterative algorithm that solves the equations using Chebyshev polynomials, and calculates the coefficients of the numerator and denominator in equation 2.2. For the first iteration, a weight \( W_0 \) is introduced.
in order to emphasize the frequencies in which we want a more precise fitting. After this first iteration, the weight changes automatically due to the relative error in the fitting. It is clear that the initial weight $W_0$ determines the result and the evolution of the next weights $W_i$ of the $i$th iteration, so a good initial weight $W_0$ is crucial for a good fitting. The criteria for choosing the initial weight $W_0$ is based in a ad hoc basis, meaning that it needs to be tuned by checking the solution after the fitting is finished. On the other hand, the stability of the fitting is also checked. We know that the plant must be stable, and we checked it by looking at the real part of the poles (i.e. stable if $\text{Re}(p) < 0 \ \forall p$). The stability constraint complicates the fitting process a lot since there is not a straightforward relationship between the stability of the plant and the initial weight $W_0$. This problem is solved, or at least minimized, by imposing constraints at the weights $W_i$ of the $i$th iteration. Basically, once the pole that makes the system unstable is known, the weight at that frequency is reduced. Reducing this weight increases the possibilities of having a stable system. In any case, the weights $W_0$ and $W_i$ need to be tuned and constrained independently for each data set.

For the HDD example, two different fits are calculated, a simple one and another more accurate. On the other hand, only one fit is calculated for the ANC data.

### 2.2 Sampled data

#### 2.2.1 Hard Disk drive data

We have 16 sets of data that correspond to two different HDD designs with two head types and tested at different operating temperatures. That gives 4 different families ($F_1, F_2, F_3,$ and $F_4$) of data to model. For each one of our 16 data sets, each of them requires a different $W_0$ such that the main desired resonances for each set are fitted precisely. On the other hand, each set within the same family of data (i.e. same design and same head) use a similar weight $W_0$.

The prior information about the plant tells us that it should be stable and that it has two integrators. Also, we know that the data contains an integrator, and the
other should be added later. In order to improve the fitting the integrator is removed form the data by multiplying each \( g(j\omega) \) by \( j\omega \) at each frequency \( \omega \), so the result \( g_2(j\omega) = g(j\omega) \cdot j\omega \) has no integrator on it making the fitting easier to solve and more precise. Once the fitting for \( g_2(j\omega) \) is finished, two integrators will be added to the fitted transfer function.

The figure 2.1 shows the bode plots of all 16 sets of data\(^1\), one can see that the plant has two integrators.

![Bode plots of all the data sets](image)

Figure 2.1: Bode plots of all the data sets

### 2.2.2 Active (acoustic) noise control data

The original data corresponds to an experiment in a noise control case study for robust identification and a feedback controller design, the interested reader can found more detailed information in [1].

The system to generate the data consists of a 4 meter long square tub connected to a semi anechoic\(^2\) room. The other end of the tube is connected to a noise generator (either a fan or a speaker). The set up has also an omnidirectional microphone used to measure the error between the noise source and the control signal, and a speaker

---

\(^{1}\)Due to confidentiality requirements, the grid and units have been removed from the HDD plots

\(^{2}\)Not having or producing echoes.
working as the controller actuator (i.e. generates the control signal to cancel the noise).

Our data, corresponds to the circuit that has the feedback controller (called secondary acoustic circuit). This secondary circuit has as input the signal produced by the control speaker and output the error captured by the microphone.

On the other hand, the sampling time $T_s$ is 0.04 ms, and 500Hz is the maximum input frequency to the system. We have then input and output data in time-domain. We can see one set of the input/output data in figure 2.2.

![Figure 2.2: Input and output data of the ANC study](image)

We have 20 sets of input/output data. The frequency response is obtained by dividing the Fast Fourier Transform (FFT) of the output signal by the FFT of the input signal. Above 500Hz, the input and output signals are almost zero, so that the quotient turns out to be really large. Then, the resulting frequency response is filtered above 500Hz to avoid this numerical error. Figure 2.3 shows the bode plot of the resulting frequency response data after filtering.
2.3 Fitting results

The results of the fittings are presented in this section. As we will see, the transfer functions obtained represent the data with the desired accuracy for both of the examples studied.

2.3.1 Hard Disk drive data

For the HDD data, the simple fit only considers data up to 14 or 15 kHz., and the accurate fit considers all the data. The number of zero and poles used for each fitting is also different. The simple fit considers a transfer function with 8 zeros and 12 poles. On the other hand, 18 zeros and 20 poles are used for the accurate fit. Also, since we are modeling the same HDD, we also calculate a middle data set that captures all the main resonances for all the families. The transfer function obtained from this data will represent the HDD plant more generally because it captures resonances from all the families.

We fit the 16 plants and obtain a mathematical model for each family of data (i.e. a transfer function $g(s)$). As expected, for each family the curve fitting transfer
function obtained is the same.

The Figure 2.4 shows the bode plots of the 4 sets of data in the $F_1$ family (blue), the simple fit (dashed magenta), and the accurate fit (dashed black).

![Figure 2.4: Bode plots of data set $F_1$ and corresponding fits](image)

The Figure 2.5 shows the bode plots of the 4 sets of data in the $F_2$ family (green), the simple fit (dashed magenta), and the accurate fit (dashed black).

![Figure 2.5: Bode plots of data set $F_2$ and corresponding fits](image)
The Figure 2.6 shows the bode plots of the 4 sets of data in the $F_3$ family (blue), the simple fit (dashed magenta), and the accurate fit (dashed black).

![Bode plots of $F_3$ family](image)

Figure 2.6: Bode plots of data set $F_3$ and corresponding fits

The Figure 2.7 shows the bode plots of the 4 sets of data in the $F_4$ family (green), the simple fit (dashed magenta), and the accurate fit (dashed black).

![Bode plots of $F_4$ family](image)

Figure 2.7: Bode plots of data set $F_4$ and corresponding fits

After all the families are curve fitted separately, the upper bound (ub) and the lower bound (lb) of the data is calculated, as well as a middle data. In the figure 2.8 we can see the upper and lower bounds (dashed black), the accurate fit for each family set (in red), and the mean plant data (in blue).
In Figure 2.8 we can see how different are the 4 accurate fits that we have obtained. Since our main purpose is obtaining a transfer function that represents all the plants, we calculate a middle plant to curve fit. This middle plant is calculated by weighting mean data (i.e. $0.5 \cdot (lb + ub)$) and the average plant for all the 16 data sets.

For the middle plant only an accurate fit is performed. In Figure 2.9 we can see the middle plant (blue) and the result of the curve fit (red). As before, we are using a model with 18 zeros and 20 poles. Also, two integrators are added to follow the prior information that we have for the HDD plant.

As for all the fitted plants (accurate or simple fit), stability is checked by looking the real part of the poles. In Figure 2.10 we can see the pole zero map (zeros marked as o, and poles marked as x) of the fit for the middle plant. We can see that all the poles are located in the left hand side, so they have negative real part. Also we can see the integrators are also plotted.

The curve fit of the middle plant will be used as a model of the HDD plant for the controller design. During this design, the difference between the 4 families will be treated as parametric uncertainty, so the variation in the resonance frequencies and in the damping ratios will capture the behavior for all the different families.
2.3.2 Active acoustic noise control data

In the case of the active acoustic noise control data only one fit has been performed. As in [1], the fit is focused in the range [90,120] Hz. The error in the fit will be treated as a uncertainty if a controller is designed. When formulating the problem for the controller design, a weight will be added to ensure that error in the frequency response is covered. The bode plot of obtained fit is shown in Figure 2.11.

We have obtained a stable model for this data that has 14 poles and 10 zeros. We can verify the stability of our fit by looking into the pole zero map in Figure 2.10.
2.12.

Figure 2.12: Pole zero map of fit obtained.

2.4 Conclusion

In this chapter, we have presented an iterative method to curve fit data corresponding to the frequency response of a plant. A transfer function is calculated by approximating the data using Chebyshev polynomials. The method used needs to be tuned by adapting an initial weight for the precise fit in the desired frequencies.

The algorithm is used to curve fit 16 sets of data for a Hard Disk Drive and data from a Active Noise Control experiment. In the HDD example, the data has been
separated in 4 families. For each family, a simple model and an accurate model has been calculated obtaining a total of 8 transfer functions.

The results show that the 8 models capture well the behavior for each family at the desired frequency range. The differences between the families lead us to calculate a middle model data that can capture all the resonances in all the families. This data has been curve fitted and a general model has been obtained.

We have obtained an accurate and stable plant model that capture the behavior of all the HDD plants studied and that can be used later for a controller design by treating the differences in the plants as parametric uncertainty.

In the case of the active acoustic noise control data, the fit obtained has very few modes compared to the model obtained in [1] by using robust identification. The fit obtained captures the behavior in the desired range of performance as well as in most of the frequency range.

The curve fitting method used in this chapter has been demonstrated to work well for two very different applications. Only by changing the number of poles and zeros in the desired model and adjusting the weight to capture the behavior in a desired frequency range, one can obtain a good fit relatively fast.
Chapter 3

Mixed sensitivity design

The transfer function for the HDD data obtained in the Chapter 2 by curve fitting the average plant has been used as the nominal model in the controller design. During the design, the difference between the 4 families and the nominal plant is treated as parametric uncertainty, thus the variation in the resonance frequencies and in the damping ratios will capture the behavior of the different families. By modeling the data as described above, we obtained a stable transfer function that is used as the nominal plant model in our designs. This chapter presents two designs using two different methodologies.

The first design looks for a controller that minimizes the $\mathcal{H}_2$ norm while the second controller minimizes the $\mathcal{H}_\infty$ norm.

3.1 $\mathcal{H}_2$ design

As introduced, our first goal is to design a $\mathcal{H}_2$ optimal controller $K$. The problem is formulated as a disturbance rejection problem and it is shown in Figure 3.1.

In Figure 3.1(a), $d$ represents the disturbance, $u$ is the control action (controller output), and $y$ is the measurement (controller input). The weights $W_d$ and $W_y$ are low pass filters, and $W_u$ is a high pass filter that is used to penalize the control
Figure 3.1: (a) Framework used for the disturbance rejection problem formulation. (b) Obtained framework for the design.

action $u$. This problem is equivalently formulated in Figure 3.1(b) by considering the transfer matrix $M$ between the input signals $d$ and $u$ and the output signals $z$ and $y$, where we define $z = [z_1; z_2]$. The objective is to design a controller $K$ such that the $H_2$ norm of the weighted ($W_d$) sensitivity transfer function (i.e. $S = (I + PK)^{-1}$) from $d$ (disturbance) to $z$ (output) is minimized. That means that we are looking for the controller $K$ that minimizes $\|W_d(I + PK)^{-1}\|_2$.

Figure 3.2: Bode plot of the results obtained at the $H_2$ design.

Figure 3.2 shows the results of the $H_2$ design. More specifically, it shows the bode plots of the controller $K$, the nominal HDD plant $P$, the open loop system $L = PK$.
the sensitivity transfer function \( S = (I + PK)^{-1} \), and the complementary transfer function \( T = PK(I + PK)^{-1} \). We can see that the controller \( K \) is a plant-inverting controller for the most part, meaning that the controller obtained tries to invert the plant in the high frequency range. This controller behavior, although it may result in good nominal performance is usually characterized by poor robustness properties.

### 3.2 \( \mathcal{H}_\infty \) design

The next goal is to formulate a mixed sensitivity \( \mathcal{H}_\infty \) problem in order to design a controller \( K \) that ensures robust performance. The problem is again formulated as a disturbance rejection problem, where now we want to minimize \( \| T_{wz} \|_\infty \).

![Figure 3.3: Framework used to formulate the mixed sensitivity \( \mathcal{H}_\infty \) problem.](image)

Figure 3.3(a) shows the formulation used to design the controller and (b) its equivalent form actually use in the design. In the schematics shown \( d \) represents the disturbance, \( u \) is the control action (controller output), \( \text{int} \) represents an integrator, \( n \) is measurement noise, \( w = [d; \xi; n] \) and \( z = [z_1; z_2; z_3] \) are the input and output respectively to the plant \( M \), and \( y \) is the measurement signal used as the input to the controller. The weights \( W_d \) and \( W_y \) are low pass filters that are used to model the desired performance specifications. The weights \( W_2 \) and \( W_u \) are high pass filters that are used respectively to robustify the design against unmodeled dynamics and keep the control action \( u \) in an acceptable range. The formulation used leads to a
system that is described in equation 3.1 as follows.

\[
\begin{bmatrix}
  z_1 \\
  z_2 \\
  z_3 \\
  y
\end{bmatrix} =
\begin{bmatrix}
  0 & 0 & 0 & W_{2\text{int}} \\
  0 & 0 & 0 & W_{u\text{int}} \\
  W_yP & W_yW_d & 0 & W_yP\text{int} \\
  P & W_d & 1 & P\text{int}
\end{bmatrix}
\begin{bmatrix}
  \xi \\
  d \\
  n \\
  u
\end{bmatrix}
\] (3.1)

Figure 3.4 shows the results obtained by the $\mathcal{H}_\infty$ controller design when using a high sampling rate. We can see that in this design the controller does not invert the plant as much as the $\mathcal{H}_2$ controller designed previously. This behavior can be explained by the fact that uncertainty has been introduced in the formulation to ensure robust performance.

![Figure 3.4: Results from the $\mathcal{H}_\infty$ controller design.](image)

The $\mathcal{H}_\infty$ controller designed has been tested on the 4 models obtained in Chapter 2 by fitting the data for each family. It was found to stabilize all 4 plants obtained in Chapter 2, so our controller is robust.

After obtaining an optimal $\mathcal{H}_\infty$ controller two aspects concerning implementation and performance constraints during the design are studied in the following chapters.
The first aspect is related to the implementation of the controller in a real HDD. The controller obtained has been designed at high rate (4x)\(^1\) but not all the controller modes can be implemented at this rate. In order to solve this problem a multirate controller should be designed. The reader can find in Chapter 4 the methodology and results obtained solving this problem.

The second aspect is related to performance constraints. Note that the \(\mathcal{H}_\infty\) controller has been constrained in the frequency domain by using specific weights in the formulation. It may seem that more constraints could be added. As for example time domain constrains that specify, for example, impulse or step response characteristics. On the other hand, we first designed a \(\mathcal{H}_2\) controller to ensure nominal performance but we also would like robustness. So a \(\mathcal{H}_2\) norm constraint could be added to the \(\mathcal{H}_\infty\) controller during its design so robust and nominal performance are assured. The \(\mathcal{H}_\infty\) design with time domain and \(\mathcal{H}_2\) constraints is explained in Chapter 5.

\(^1\)It means that high rate is 4 times the base rate
Chapter 4

Multirate design

Due to hardware and computational constraints, the implementation of the theoretical results obtained is a hard problem in most of the cases. In our case, the design has been implemented at high rare, which means a ratio 4x using a base (low) rate. Our specifications tell us that no more that 10 modes can be implemented at this 4x rate, and 20 more modes can be used at the base rate. The optimal $\mathcal{H}_\infty$ controller obtained in Chapter 3 has 77 states. Even though the order of the controller can be reduced using model reduction techniques, achieving a controller at high rate with 10 modes while keeping $\mathcal{H}_\infty$ optimality is an impossible task.

One can think that designing directly at the base rate could solve the problem, at least after model reduce the controller obtained. The problem of this approach arises when one analyzes the plant model obtained in Chapter 2. Since the plant has modes at frequencies higher than the Nyquist frequency of the base rate, aliasing will appear in the discrete plant. The aliasing will affect the controller since the optimal $\mathcal{H}_\infty$ will approximate a plant inverting controller.

4.1 Methods

In order to address the implementation problem stated above, we propose a multirate controller. This multirate controller is obtained by separating the high frequency
part of the high rate $\mathcal{H}_\infty$ controller (at 4x rate). This high rate part is added to the plant, so the high frequency modes are canceled so, in case of a posterior base rate rediscetization, the aliasing will not appear.

After the high frequency part of the controller is added to the plant, the plant is lifted to keep the high rate performance constraints. Notice that we are using the same disturbance rejection formulation as before (as shown in Figure 4.1), but now the uncertainty weight $W_2$ and the input $\xi$ are located after the HDD plant. As seen in figure 4.1, the formulation used leads to a system that is described in equation 4.1 as follows.

$$
\begin{bmatrix}
z_1 \\
z_2 \\
z_3 \\
y
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & W_2 \text{Pint} \\
0 & 0 & 0 & W_u \text{int} \\
W_y & W_y W_d & 0 & W_y \text{Pint} \\
1 & W_d & 1 & \text{Pint}
\end{bmatrix}
\begin{bmatrix}
\xi \\
d \\
n \\
u
\end{bmatrix}
$$

(4.1)

Before we proceed to the results, we need to explain first how we split the controller and the theory of the lifting technique cited above.

### 4.1.1 Splitting the controller

Any system (in continuous or discrete time) can be defined as a product or summation of modes, where each mode has an associated frequency. Without loss of generality we will study the discrete time case. The difference between continuous
and discrete time is how we calculate the frequency using the poles/zeros associated to each mode.

Let’s have a discrete time system $K(z)$. As said before we can define $K(z)$ as the product of $m$ modes (i.e. $K(z) = \prod_{i=1}^{m} K_i(z)$) or as a summation of $m$ modes (i.e. $K(z) = \Sigma_{i=1}^{m} \tilde{K}_i(z)$). The first one is a series decomposition (Figure 4.2), and the second one is a parallel decomposition (Figure 4.3).

![Figure 4.2: Series decomposition of the controller.](image)

In our design we will use a series decomposition so we divide our high rate controller ($K_{HR}$) in a low frequency part and a high frequency part ($K_{HR} = K_{lf} \cdot K_{hf}$). After we split the controller, we add $K_{hf}$ to the plant. Then, we formulate the problem (as disturbance rejection), and lift the resulting system. With the lifted system (at the base rate), we design again to obtain a new optimal controller $K'_{lf}$.

One of the advantages of the method proposed is that the low frequency part of the controller (i.e. $K'_{lf}$) is calculated at the base rate and at high frequency part of the controller (i.e. $K_{hf}$ is calculated at the 4x rate. The combination in series of both gives us the multirate controller we were looking for.
4.1.2 Lifting technique

Consider a continuous time system $G(s)$ with $n_i$ inputs $w$, $n_o$ outputs $z$, $n_c$ controls $u$, $n_m$ measurements $y$. We can see in Figure 4.4 the representation for our system.

![System representation](image)

Figure 4.4: System representation

Our purpose is design a discrete time controller $K_d(z)$ by sampling/holding the measurements and controls at a slow sampling rate ($T_s$). On the other hand, the inputs/outputs of the system are held/sampled at a fast rate ($T_f$). In our case, $T_f = T_s/n$ where $n$ is a positive integer. As shown in Figure 4.4, we transform our continuous time system $G(s)$ to a discrete time system $P(z)$ by including the ideal sampler (S) and holder (H). Then,

\[
P(z) = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} S_s & 0 \\ 0 & S_f \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} H_s & 0 \\ 0 & H_f \end{bmatrix} = \begin{bmatrix} S_f G_{11} H_f & S_f G_{12} H_s \\ S_s G_{21} H_f & S_s G_{22} H_s \end{bmatrix}
\]

As we can see in equation 4.2, the resulting system $P(z)$ is a multirate system. The lifting ($L$) and inverse lifting ($L^{-1}$) operators are introduced in order to obtain a system with one rate. The lifting operator $L$ maps a discrete time signal $v(k)$ of
period $h$ to a vector $v(k)$ of period $h/n$, where $n$ is a positive integer. Then,

$$
v = \begin{cases} 
\begin{bmatrix} v(0) \\
v(1) \\
\vdots \\
v(n-1) 
\end{bmatrix}, & \begin{bmatrix} v(n) \\
v(n+1) \\
\vdots \\
v(2n-1) 
\end{bmatrix}, \ldots 
\end{cases}
$$

(4.3)

Note that the underlined signal is the lifted signal. The inverse lifting operator $(L^{-1})$ maps $v(k)$ into $v(k)$. Then the lifted system can be represented as

![Lifted system representation](image)

Figure 4.5: Lifted system representation

From Figure 4.5 we can calculate the lifted system $P$.

$$
P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} LS_f \bar{G}_{11} \bar{H}_f L^{-1} & LS_f \bar{G}_{12} \bar{H}_s \\ S_{s} \bar{G}_{21} \bar{H}_f L^{-1} & S_{s} \bar{G}_{22} \bar{H}_s \end{bmatrix}
$$

(4.4)

From the equation 4.4 we can calculate the lifted system. The lifting technique used in [17] will be used to calculate $P_{11}$, $P_{12}$, $P_{21}$ and $P_{22}$.

We would like to clarify the notation that will be used in this chapter. We consider that the continuous time system can be represented as:

$$
G(s) = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}
$$

(4.5)

The discrete representation of the system will be denoted with a subindex $f$ and $s$ for the discretization at fast and slow rate, respectively. For example, the fast
discretization of $G_{12}$ can be represented as follows.

$$P_{12} = \begin{bmatrix} A_f & B_{2f} \\ C_{1f} & D_{12f} \end{bmatrix} \quad (4.6)$$

In order to clarify the procedure used later in this chapter is necessary first to define the formulas used for the discretization. The fast discretization (using the sample time $T_f$) of the system $G(s)$ represented in equation 4.5 is defined as follows.

$$P_f = \begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix} = \begin{bmatrix} A_f & B_{1f} & B_{2f} \\ C_{1f} & D_{11f} & D_{12f} \\ C_{2f} & D_{21f} & D_{22f} \end{bmatrix} \quad (4.7)$$

where

$$A_f = e^{T_f A}$$
$$B_f = \begin{bmatrix} B_{1f} & B_{2f} \end{bmatrix} = \int_0^{T_f} e^{\tau A} d\tau B$$
$$C_f = \begin{bmatrix} C_{1f} \\ C_{2f} \end{bmatrix} = C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$
$$D_f = \begin{bmatrix} D_{11f} & D_{12f} \\ D_{21f} & D_{22f} \end{bmatrix} = D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \quad (4.8)$$

Note also that the underlined systems or signals represent the systems or signals that they have been lifted.

In order to calculate $P_{11}$, we have to discretize $G_{11}$ first at the fast rate to obtain $P_{11}$ (defined in equation 4.9) and then use the lifting technique [17] to obtain $\underline{P}_{11}$ (defined in equation 4.10).

$$P_{11} = \begin{bmatrix} A_f & B_{1f} \\ C_{1f} & D_{11f} \end{bmatrix} \quad (4.9)$$

Then, following the same procedure as in [17] we can calculate the lifted system $\underline{P}_{11}$ using the equations in 4.10.
\[
\begin{bmatrix}
\frac{A}{C_1} & \frac{B_1}{D_{11}}
\end{bmatrix} = \begin{bmatrix}
A^n_f & A^{n-1}B_{1f} & A^{n-2}B_{1f} & \cdots & B_{1f} \\
C_{1f} & D_{11f} & 0 & \cdots & 0 \\
C_{1f}A_f & C_{1f}B_{1f} & D_{11f} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{1f}A_f^{n-1} & C_{1f}A_f^{n-2}B_{1f} & C_{1f}A_f^{n-3}B_{1f} & \cdots & D_{11f}
\end{bmatrix}
\]
(4.10)

In order to calculate \( P_{12} \) and \( P_{21} \) we have to manipulate the expressions from equation 4.4. For the first one, we have that
\[
P_{12} = LS_fG_{12}H_s = L(S_fG_{12}H_f)S_fH_s = L(S_fG_{12}H_f)L^{-1}LS_fH_s.
\]
It follows that \( L(S_fG_{12}H_f)L^{-1} \) can be obtained by lifting the discretization at fast rate of \( G_{12} \), and we can obtain the value of \( LS_fH_s \) as is calculated in 4.11.

\[
\begin{bmatrix}
I & 0 \\
\vdots & \vdots \\
I & 0 \\
0 & I \\
\vdots & \vdots \\
0 & I
\end{bmatrix}^n = \begin{bmatrix}
I \\
\vdots \\
I
\end{bmatrix}, \text{ (n blocks)} \quad (4.11)
\]

We can obtain \( P_{12} \) by multiplying the lifted system and the result from equation 4.11. The equations 4.12 and 4.13 show the resulting expressions for \( P_{12} \).

\[
P_{12} = \begin{bmatrix}
A^n_f & A^{n-1}B_{2f} & A^{n-2}B_{2f} & \cdots & B_{2f} \\
C_{1f} & D_{12f} & 0 & \cdots & 0 \\
C_{1f}A_f & C_{1f}B_{2f} & D_{12f} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{1f}A_f^{n-1} & C_{1f}A_f^{n-2}B_{2f} & C_{1f}A_f^{n-3}B_{2f} & \cdots & D_{12f}
\end{bmatrix} \begin{bmatrix}
I \\
\vdots \\
I
\end{bmatrix} \quad (4.12)
\]
\[
P_{12} = \begin{bmatrix}
A & B_2 \\
C_1 & D_{12}
\end{bmatrix} = \begin{bmatrix}
A_f^n & (A_f^{n-1} + A_f^{n-2} + \cdots + I)B_{2f} \\
C_1f & D_{12f} \\
C_1fA_f & C_1fB_{2f} + D_{12f} \\
\vdots & \vdots \\
C_1fA_f^{n-1} & C_1fA_f^{n-2}B_{2f} + C_1fA_f^{n-3}B_{2f} + \cdots + D_{12f}
\end{bmatrix}
\] (4.13)

Similarly, \(P_{21} = S_sG_{21}H_fL^{-1} = S_sH_f(S_fG_{21}H_f)L^{-1} = S_sH_fL^{-1}L(S_fG_{21}H_f)L^{-1}\). It follows that \(L(S_fG_{21}H_f)L^{-1}\) can be obtained by lifting the discretization at fast rate of \(G_{21}\), and we can obtain the value of \(S_sH_fL^{-1}\) as is calculated in equation 4.14.

\[
S_fH_sL^{-1} = \begin{bmatrix}
I & 0 & \cdots & 0 \\
I & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
I & 0 & \cdots & 0
\end{bmatrix}
\]

\[
L^{-1} = \begin{bmatrix}
I & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
I & 0 & \cdots & 0
\end{bmatrix}
\] (4.14)

We can obtain \(P_{21}\) by multiplying the lifted system and the result from equation 4.14. The equations 4.15 and 4.16 show the resulting expressions for \(P_{21}\).

\[
P_{21} = \begin{bmatrix}
A_f^n & A_f^{n-1}B_{1f} & A_f^{n-2}B_{1f} & \cdots & B_{1f} \\
C_2f & D_{21f} & 0 & \cdots & 0 \\
C_2fA_f & C_1fB_{1f} & D_{21f} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_2fA_f^{n-1} & C_2fA_f^{n-2}B_{1f} & C_2fA_f^{n-3}B_{1f} & \cdots & D_{21f}
\end{bmatrix}
\] (4.15)

\[
P_{21} = \begin{bmatrix}
A_s & B_1 \\
C_2 & D_{21}
\end{bmatrix} = \begin{bmatrix}
A_f^n & A_f^{n-1}B_{1f} & A_f^{n-2}B_{1f} & \cdots & B_{1f} \\
C_2f & D_{21f} & 0 & \cdots & 0
\end{bmatrix}
\] (4.16)

On the other hand, from equation 4.4, we can see that \(P_{22}\) is just the slow rate discretization of \(G_{22}\). Note that there is no lifting in this part of the system, we just discretize at low rate.

\[
P_{22} = P_{22} = \begin{bmatrix}
A_s & B_{2s} \\
C_2s & D_{22s}
\end{bmatrix}
\] (4.17)
Note that from equations 4.10 and 4.17 it follows that \( A^n_f = A_s \). This can be proved easily by using the formulas from equation 4.8 and the fact that \( T_s = nT_f \).

\[
A^n_f = (e^{T_f A})^n = e^{nT_f A} = e^{T_s A} = A_s
\]

(4.18)

It is necessary also to prove that \( C_{2f} = C_{2s} \) and that \( B_{2s} = (A_f^{n-1} + A_f^{n-2} + \cdots + A_f I)B_{2f} \). The first one can be directly derived from equation 4.8, since \( C_2 \) does not change in the discretization it follows that \( C_2 = C_{2f} = C_{2s} \). The second equality can be proved using the equation 4.8 and the fact that \( T_f = T_s / n \).

\[
B_{2s} = \int_0^{T_s} e^{\tau A} d\tau B = \left( \int_0^{T_s/n} e^{\tau A} d\tau + \cdots + \int_{(n-1)T_s/n}^{T_s} e^{\tau A} d\tau \right) B
= (I + A_f + \cdots + A_f^{n-2} + A_f^{n-1}) B_{2f}
\]

(4.19)

After calculating all the subsystems separately and using the equivalences proved above, we can add them so that we obtain the low rate lifted system \( P \). The equations 4.20 and 4.21 show the final result.

\[
P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}
\]

(4.20)
Then,

\[
A = [A^n_f]
\]

\[
B = \left[ A_f^{n-1} B_{1f} \ A_f^{n-2} B_{1f} \ \cdot \cdot \cdot \ B_{1f} \ \left( A_f^{n-1} + A_f^{n-2} + \cdots + A_f + I \right) B_{2f} \right]
\]

\[
C = \begin{bmatrix}
C_{1f} \\
C_{1f} A_f \\
\vdots \\
C_{1f} A_f^{n-1} \\
C_{2f}
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
D_{11f} & 0 & \cdots & 0 & D_{12f} \\
C_{1f} B_{1f} & D_{11f} & \cdots & 0 & C_{1f} B_{2f} + D_{12f} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
C_{1f} A_f^{n-2} B_{1f} & C_{1f} A_f^{n-3} B_{1f} & \cdots & D_{11f} & C_{1f} A_f^{n-2} B_{2f} + \cdots + D_{12f} \\
D_{21f} & 0 & \cdots & 0 & D_{22s}
\end{bmatrix}
\]  
\hspace{8cm} (4.21)

4.2 Controller design

The first step in the multirate controller design is to obtain a high rate (4x) $\mathcal{H}_\infty$ controller. We will take the high frequency part of the high rate controller $K_{HR}$ and add it to the plant. The new filtered plant is used to formulate again the disturbance rejection problem. The resulting system is lifted by using the methodology explained above.

4.2.1 High rate design and controller decomposition

Using the disturbance rejection formulation we obtain a optimal controller $K_{HR}$ that minimizes the $\mathcal{H}_\infty$ norm from the inputs to the outputs defined in Figure 4.1.
We can see the bode plot of the controller obtained in the figure 4.2.1. Having in mind the shape of our plant (obtained in chapter 2) this controller inverts the plant almost completely.

![Bode plot of H∞ controller](image)

Figure 4.6: Bode plot of the $\mathcal{H}_\infty$ controller obtained at high rate.

Using a series decomposition, we split the controller in low and high frequency modes. For our design we just keep high frequency modes within a range of frequencies. This range is selected to capture the modes that will cancel the modes in the plant causing aliasing if discretized at low rate. The value of the frequency rage is not specified due to data confidentiality. We can see in Figure 4.7 the high frequency controller $K_{hf}$ that will be used in our design. This controller has 10 modes, so the hardware constraints are satisfied.

Comparing the bode plot of the controllers $K_{HR}$ and $K_{hf}$ (figures and 4.7 respectively), one can see that, within a frequency range, $K_{hf}$ is very similar to $K_{HR}$.

### 4.2.2 Low rate design of the lifted system

After obtaining the controller $K_{hf}$, it is added to the plant and the disturbance rejection problem is reformulated again. The resulting system is lifted using the
The controller obtained \( (K_{lf}) \) has 20 modes, so the hardware constraints are satisfied. The bode plot of the controller obtained is shown in Figure 4.8.

An obvious question that the reader can think of is why the system is lifted. It seems that once we have the filtered plant \( (P_{filt}) \) we can discretize at low rate and avoid aliasing. If we take a look into 4.1 and we change \( P \) by \( (P_{filt}) \), we obtain equation 4.22.

\[
\begin{bmatrix}
  z_1 \\
  z_2 \\
  z_3 \\
  y
\end{bmatrix} =
\begin{bmatrix}
  0 & 0 & 0 & W_2 P_{filt int} \\
  0 & 0 & 0 & W_x int \\
  W_y & W_y W_d & 0 & W_y P_{filt int} \\
  1 & W_d & 1 & P_{filt int}
\end{bmatrix}
\begin{bmatrix}
  \xi \\
  d \\
  n \\
  u
\end{bmatrix}
\]

If we discretize the system at low rate, we avoid aliasing in the plant, but we will lose performance and robustness since we are using low rate weights. In other words, the information that the weights are giving us at high rate is not used.
Using the lifting technique explained above, the only part in the system that is discretized at low rate is $P_{filt\,int}$. The rest of the system is split (lifted) so we do not lose performance and robustness because we are not changing the weights by rediscretizing them at low rate.

### 4.2.3 Low rate design

As we stated at the beginning of this chapter, the methodology used in this is used to avoid the aliasing due to discretization at low rate. In order to proof that aliasing affects the resulting $\mathcal{H}_\infty$ controller, we design also at low rate using the same formulation. The low rate weights used were chosen to follow the same premises as the high rate weights. We can see this fact in the figure 4.9.

### 4.2.4 Controllers obtained

In order to clarify if the obtained controllers have the expected shape, it is necessary to plot them together. In Figure 4.10 we can see the magnitude plot of the
controllers. The represented controllers $K_{HR}$, $K_{hf}$, $K_{lf}$, and $K_{LR}$ are respectively the high rate controller, the high frequency part of the high rate controller, the low frequency controller (obtained from the lifted system), and the low rate controller. For plotting and comparing purposes the controller $K_{hf}$ has been scaled. Note that after splitting the high rate controller $K_{HR}$, its high frequency part $K_{hf}$ is calculated to have unitary gain, so it does not modify the gain of the HDD plant.

![Figure 4.9: Bode plot of the $\mathcal{H}_\infty$ controller obtained at low rate.](image)

We can see in Figure 4.10 that $K_{hf}$ and $K_{lf}$ have the same shape as $K_{HR}$ for a determined frequency range, so we have achieved the expected result. On the other hand, we can see that $K_{LR}$ has three small picks (marked with a red *) that the rest

![Figure 4.10: Magnitude plot of the $\mathcal{H}_\infty$ controllers](image)
of the controllers do not have. Since the $\mathcal{H}_\infty$ controller tries to invert the plant as much as possible, these picks appear because of the aliasing caused by the low rate discretization of the HDD plant.

4.3 Testing the controllers

Even though we have seen that the controllers obtained have the expected shape, this does not ensure that they perform as they should. Using Simulink© we have tested the controllers. A impulse is introduced as a disturbance in the measurement of the closed loop system. Since the system has been formulated as a disturbance rejection problem, the controllers should be able to handle the measurement disturbance.

![Figure 4.11: Time response of the plant after an impulse disturbance in the measurement has been applied.](image)

Figure 4.11 shows the time response of the plant for the three controllers obtained ($K_{HR}$, $K_{LR}$, and $K_{hf}K_{lf}$). As we can see in the figure 4.11, the $K_{HR}$ provides the best time response (fastest error rejection) and $K_{LR}$ the worst time response. This fact was expected since at high rate the controller has more information and it can cancel the disturbance faster. We expect that the controller $K_{hf}K_{lf}$ provides a
response that will be between the time response at high rate and the low rate. The figure 4.11 shows that this controller gives a response very similar to the high rate controller.

The advantage of this multirate controller \((K_{MR} = K_{lf}K_{hf})\) is that it satisfies the hardware constraints (i.e. number of modes) while the loss in performance in the disturbance rejection is very small compared to the performance of a low rate controller.

On the other hand we also checked the error rejection in the plant model for the four families obtained in Chapter 2. We can see in Figure 4.12 that the same behavior seen previously. The multirate controller \(K_{MR}\) works better than the low rate controller \(K_{LR}\) and almost as well as the high rate controller \(K_{HR}\). Note also that some of the low rate responses are not stable, making the low rate controller not as robust as the high rate and multirate controllers. This fact comes from the fact that the low rate weights miss some of the robustness information (i.e. the robustness constraints at high frequency).

![Figure 4.12: Time response of the all the plants.](image-url)
Chapter 5

Time domain and $H_2$ norm constrained $H_\infty$

In previous Chapters we have been able to identify sampled data and curve fit it in order to obtain a model of our system. This model has been used to design a mixed sensitivity $H_2$ and a $H_\infty$ optimal controller by formulating a disturbance rejection problem. In this chapter we present an algorithm (implemented in Matlab\textsuperscript{\textregistered}) that computes and $H_\infty$ optimal controller subject to time domain and $H_2$ performance constraints.

Although $H_\infty$ computes a controller that ensures robust stability, when we impose time domain constraints in our design, robust stability may not be enough. Time domain constrains were added to $H_\infty$ optimization [6] so robust stability is achieved as well as the desired (time domain) performance. The time domain optimization is done for a finite horizon, but long enough to avoid problems after the horizon (for example, instability).

On the other hand, mixed $H_2/H_\infty$ has also been studied by minimizing $H_2$ norm while having $H_\infty$ norm bounded ( [9] , [12] and [10]). The purpose of the mixed $H_2/H_\infty$ is to keep the optimal nominal performance subject to a robust stability constraint.
In our case, we add $\mathcal{H}_2$ constraints to the time domain constrained $\mathcal{H}_\infty$ framework. Then, the controller guarantees robust stability while the time domain/$\mathcal{H}_2$ constraints ensure to obtain the desired performance.

The methods implemented in Matlab® are explained in the next section. The last section in this chapter presents an example where we apply the time domain and $\mathcal{H}_2$ constrained $\mathcal{H}_\infty$ algorithm.

5.1 Methods

The algorithm used in [6] was initially implemented in Fortran® to optimize calculation effort and time. This base algorithm has been translated to Matlab® and the $\mathcal{H}_2$ constraint has been added.

In order to understand better how the algorithm has been implemented it is necessary to review some theory about robust control. The interested reader can find more detailed information about robust control theory in [4] and more detailed information about the theory explained below in [6].

5.1.1 Problem formulation

Let us consider a discrete time system $P$ with inputs $w_f$, control $u$, and outputs $z_f$ and measurement $y$. Note that the subindex $f$ means that the input/outputs are in frequency domain. We can see this general system configuration in Figure 5.1.

![Figure 5.1: Control system configuration.](image)

The most common problem in $\mathcal{H}_\infty$ control is to find a stabilizing controller $K$
that minimizes the infinity norm of transfer function from $w_f$ to $z_f$ (i.e. $\|T_{w_f,z_f}\|_\infty$).

For the time domain and $H_2$ constrained $H_\infty$ control problem it is necessary to add more input and outputs to the control system configuration shown in Figure 5.2. We can see that we have added $w_t/z_t$, and $w_2/z_2$. The subindexes $t$ and 2 refer to the inputs/outputs with time domain and $H_2$ norm constrains respectively.

![Control system configuration](image)

Figure 5.2: Control system configuration for the constrained problem.

With this configuration, the problem to solve is to find a stabilizing controller $K$ that minimizes $\|T_{w_f,z_f}\|_\infty$ while $T_{w_t,z_t}$ satisfies certain time domain performance specifications and $\|T_{w_2,z_2}\|_2 \leq \beta$ where $\beta$ is specified depending on the problem. Note that the time domain constraint are specified for a finite horizon.

### 5.1.2 Problem parametrization

The idea behind this methodology relies in reformulating the problem into a constrained optimization over a finite horizon, that can be demonstrated to be a convex optimization. The time domain and $H_2$ constraints are transformed and parameterized to depend on a parameter transfer matrix $Q$. The $H_\infty$ norm minimization problem is also reformulated in terms of $Q$. The parameter $Q$ has a value for each sample of the horizon and leads to calculate the optimal controller $K$ that solves the original problem formulated above.
Considering the system represented in Figure 5.2, we can define it as follows.

\[
\begin{bmatrix}
    z_t \\
    z_2 \\
    z_f \\
    y
\end{bmatrix} =
\begin{bmatrix}
    P_{11} & P_{12} & P_{13} & P_{14} \\
    P_{21} & P_{22} & P_{23} & P_{24} \\
    P_{31} & P_{32} & P_{33} & P_{34} \\
    P_{41} & P_{42} & P_{43} & P_{44}
\end{bmatrix}
\begin{bmatrix}
    w_t \\
    w_2 \\
    w_f \\
    u
\end{bmatrix}
\] (5.1)

Using Equation 5.1, we can close the loop using the controller \( K \). Then, we can obtain the transfer functions for each input/output as follows.

\[
T_{w_t,z_t} = P_{11} + P_{14}K(I - P_{44}K)^{-1}P_{41}
\] (5.2)

\[
T_{w_2,z_2} = P_{22} + P_{24}K(I - P_{44}K)^{-1}P_{42}
\] (5.3)

\[
T_{w_f,z_f} = P_{33} + P_{34}K(I - P_{44}K)^{-1}P_{43}
\] (5.4)

In order to find the controller \( K \), we can parameterize all stabilizing controllers by using the so-called Youla parametrization \(^1\). Using the Youla parametrization, we can calculate \( T_{ij} \) from the problem data, so we obtain the set of stable closed-loop transfer functions as follows.

\[
T_{w_t,z_t} = T_{11} + T_{14} QT_{41}
\] (5.5)

\[
T_{w_2,z_2} = T_{22} + T_{24} QT_{42}
\] (5.6)

\[
T_{w_f,z_f} = T_{33} + T_{34} QT_{43}
\] (5.7)

The Youla parametrization gives us the opportunity to separate the constraints and minimization and treat them separately to obtain a convex optimization problem.

By using the Youla parametrization, expressed in equation 5.5, the time domain constraints can be transformed into a convex (linear) constraints on the \( Q \)'s. The interested reader is referred to [6] where more detailed information about the Youla parametrization can be found.

\(^1\) The interested reader is referred to [6] where more detailed information about the Youla parametrization can be found.
ested readed is referred to [6] for more detailed information about this topic.

On the other hand, the $\mathcal{H}_2$ norm constraints are treated similarly to the time domain constraints. In this case, our lower bound is zero and the upper bound is $\beta$ for all the horizon. Then, using again the Youla parametrization expressed in equation 5.6, we can calculate the $\mathcal{H}_2$ norm depending on $Q$.

Once we have defined the constraints in terms of $Q$, we need to reformulate the problem finding the controller that minimizes the $\mathcal{H}_\infty$ norm. Using again the Youla parametrization we can reformulate our optimization problem as follows.

\[
\|T_{w_f,z_f}\|_\infty = \left\| \begin{array}{cc} G_{11} & G_{12} \\ G_{21} & G_{22} - Q^\sim \end{array} \right\|_\infty \leq \gamma \tag{5.8}
\]

where the value of $G_{ij}$ depends on the plant $P$ and can be calculated from the Youla parametrization ans inner-outer factorizations.

The problem stated in 5.8 is the so-called four-block problem. We have parameterized our problem in terms of $Q$, obtaining a constrained convex optimization problem.

### 5.1.3 Solving the constrained convex optimization problem

In order to solve the constrained convex optimization problem, we use the Ellipsoid Algorithm (EA). The interested reader can find more detailed information about the Ellipsoid Algorithm in [16] and [15].

The Ellipsoid Algorithm calculates a sequence of ellipsoids $(E_0, E_1, \ldots)$ with decreasing volume. Each of these $E_i$ ellipsoids will contain the optimal solution for our problem if the first $E_0$ ellipsoid contained it.

At each iteration the algorithm check if the center of the corresponding ellipsoid is a feasible solution (i.e. satisfies all the constraints).

If it is not feasible, then one of the violated constraints is selected and a new ellipsoid is calculated. This new ellipsoid is the ellipsoid of minimum volume con-
tained in the half ellipsoid defined by the violated constraint. This new minimum volume ellipsoid become the ellipsoid used in the next iteration.

If the point is feasible, the objective function and its generalized gradient is computed. Then, using the generalized gradient, the half ellipsoid is selected ensuring that contains the optimal solution of the problem. Then, we compute the minimum volume ellipsoid containing the half ellipsoid previously selected. Again, this new minimum volume ellipsoid become the ellipsoid used in the next iteration. Modified versions of the ellipsoid algorithm used the so-called *Deep-cuts*, that are characterized by using less than half of the ellipsoid (but still containing the optimal solution). The *Deep-cuts* modified algorithm speed up the iterations to convergence.

On the other hand, the generalized gradient is not calculated directly. The maximum singular value of the objective function at the current point is calculated by using the Lanczos algorithm. We will not explore in detail the basis of the Lanczos algorithm since is not the purpose of the Thesis. The interested reader is referred to [18] for more detailed information.

### 5.2 Constrained $\mathcal{H}_\infty$ design example

In order to validate the algorithm and the theory behind it, one example of $\mathcal{H}_2$ norm constrained $\mathcal{H}_\infty$ are presented.

The model used is the ANC model obtained in Chapter 2. The time domain constraints are not used in these examples because they are not specified in the problem statement [1]. On the other hand, time domain constrained $\mathcal{H}_\infty$ control has already been proved to give positive results ([14], [6]).

The process of optimization explained above needs that we specify the $\mathcal{H}_2$ norm bound. The specification of robustness is ensure if the $\mathcal{H}_\infty$ norm is less than one. In order to define this bound, straights $\mathcal{H}_2$ design and $\mathcal{H}_\infty$ designs are performed.

The the optimal $\mathcal{H}_2$ controller will be the desired solution if the corresponding $\|T_{w_f,z_f}\|_\infty$ is less than one. If this happens, optimization is not needed and the straight $\mathcal{H}_2$ design guarantees robust stability.
Same would happen if for the optimal $H_\infty$ controller, the corresponding $\|T_{w_2,z_2}\|_2$ would be the optimal.

This rarely happens, the trade off between optimal $H_\infty$ and $H_2$ norms allow the optimization process. We can see in the Figure 5.3 , the usual behavior of the optimal norms.

![Figure 5.3: General behavior of the optimal norms.](image)

As we can see for the optimal $H_2$ norm the $H_\infty$ norm is greater than one. So in order to achieve robust stability is necessary to sacrifice nominal performance.

We can see in the Figure 5.4 the formulation used. The $\|T_{w_f,z_f}\|_\infty$ is minimized while keeping the $\|T_{w_2,z_2}\|_2$ bounded.

![Figure 5.4: Formulation for the constrained $H_\infty$ problem.](image)

First of all, the optimal $H_2$ and $H_\infty$ have been designed in order to define the boundaries of optimization. The optimal $H_2$ controller achieves a $H_2$ norm of 0.1397, but the corresponding $H_\infty$ norm is 364.25. We are then, far away from robustness but with a good nominal performance. on the other hand, the optimal
$\mathcal{H}_\infty$ controller achieves a $\mathcal{H}_\infty$ norm of $1e-15$ (zero), but the corresponding $\mathcal{H}_2$ norm is 0.4092.

Once we have our limits of performance and robustness, several controllers are obtaining by increasing gradually the $\mathcal{H}_2$ constraint. The optimal controller obtained has a $\mathcal{H}_2$ norm of 0.1909 and a $\mathcal{H}_\infty$ norm of 0.3225. This controller satisfies the robustness specification ($\|T_{w_f,z_f}\|_\infty < 1$) and has a $\mathcal{H}_2$ norm close to the pure $\mathcal{H}_2$ optimal norm.

On the other hand, several designs have been done to see if the expected behavior is confirmed. We can see in Table 5.1 that the behavior expected is confirmed. The increment of the optimal $\mathcal{H}_2$ norm produces a decrement in the $\mathcal{H}_\infty$ norm.

<table>
<thead>
<tr>
<th>Design</th>
<th>$|T_{w_2,z_2}|_2$</th>
<th>$|T_{w_f,z_f}|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt $\mathcal{H}_2$</td>
<td>0.1397</td>
<td>364.25</td>
</tr>
<tr>
<td>1</td>
<td>0.1419</td>
<td>1.8820</td>
</tr>
<tr>
<td>2</td>
<td>0.1674</td>
<td>1.7492</td>
</tr>
<tr>
<td>3</td>
<td>0.1766</td>
<td>1.2300</td>
</tr>
<tr>
<td>4</td>
<td>0.1909</td>
<td>0.3425</td>
</tr>
<tr>
<td>5</td>
<td>0.2024</td>
<td>0.3221</td>
</tr>
<tr>
<td>6</td>
<td>0.2119</td>
<td>0.3002</td>
</tr>
<tr>
<td>7</td>
<td>0.2257</td>
<td>0.2569</td>
</tr>
<tr>
<td>8</td>
<td>0.3728</td>
<td>0.0370</td>
</tr>
<tr>
<td>Opt $\mathcal{H}_\infty$</td>
<td>0.4092</td>
<td>1e-15</td>
</tr>
</tbody>
</table>

Table 5.1: Results of several designs.

Our optimal $\mathcal{H}_n$ norm constrained $\mathcal{H}_\infty$ design is then design number 4. The achieved $\mathcal{H}_2$ norm is 0.1909 and the $\mathcal{H}_\infty$ norm is 0.3425 < 1. A design that guarantees robustness and is near the optimal $\mathcal{H}_2$ has been designed.
The disturbance rejection time responses of the closed loop systems (computed using the different optimal controllers) are shown in Figure 5.5.

![Figure 5.5: Disturbance time response of the different optimal controllers.](image)

In Figure 5.5 we can see that the optimal $\mathcal{H}_2$ controllers (red and green) have the best time response to an impulse disturbance. On the other hand, the $\mathcal{H}_\infty$ (cian) controller has the worst response. The $\mathcal{H}_2$ constrained $\mathcal{H}_\infty$ controller (blue) has a response located between them.

Then, as shown in the data from table 5.1 and in Figure 5.5, we lose performance but we are able to obtain robust stability (i.e. $\mathcal{H}_\infty$ norm is smaller than one).

It has been shown then, that the methodology and algorithm used are valid for $\mathcal{H}_2$ norm constrained $\mathcal{H}_\infty$ design.
Chapter 6

Conclusions

The framework presented in this Thesis for a practical control design has been shown to work.

In Chapter 2, a transfer function is obtained by curve fitting sampled data. By adjusting adequately the weights used to capture the behavior of the data in a specific range of frequency, the obtained transfer function fits very well the sampled data.

Two examples are shown. The first one is data from a Hard Disk Drive where 16 sets of data are used. In the second one, the data come from an Active Acoustic Noise Control Experiment. In both examples the results are very good, obtaining stable models with relatively low complexity.

In Chapter 3, the transfer function calculated for the Hard Disk Drive is used to solve a mixed sensitivity problem using $H_2$ and $H_\infty$ optimal control. The problem is formulated as a disturbance rejection problem and solved. The results help us to understand the optimal controller behavior and lead to the multirate controller.

In Chapter 4, a multirate controller is designed using $H_\infty$ optimal control. Again the mixed sensitivity problem is formulated as a disturbance rejection system.

The multirate controller design is performed in three steps. First, an optimal controller is designed at high rate (4x in our case) and the high frequency part of
this controller is saved for later. In the second step, the high frequency controller saved is added to the plant and the problem is formulated again. At the last step, the new formulation is lifted to low rate and a new controller is designed.

The results show that the multirate controller has less modes that a high rate controller with very few loss of performance and robustness, and better than a optimal low rate controller in error rejection. We obtain then, a multirate controller that satisfies the performance and robustness specification as well as the implementation and computational constraints.

In Chapter 5, an algorithm that allows to compute optimal $\mathcal{H}_\infty/\mathcal{H}_2$ controllers is presented. The computation algorithm is explained and an example using the Active Acoustic Noise Control transfer function obtained in Chapter 2. The controller obtained minimizes the effect of a disturbance (noise), so the noise cancelation control works as desired. The controller obtained in this chapter will be implemented on a DSP and tested with the experimental set up use to obtain the data for the curve fitting. The simulation results obtained here give reasons to expect also good experimental results.

A future research topic will be be the development of multirate optimal controllers satisfying time domain and $\mathcal{H}_2$ constraints.
Bibliography


