TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Making spatial trade-offs using multiobjective H2 synthesis

Citation for published version (APA):
Donkers, M. C. F. (2006). *Making spatial trade-offs using multiobjective H2 synthesis*. (DCT rapporten; Vol. 2006.109). Technische Universiteit Eindhoven.

Document status and date:
Published: 01/01/2006

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 22. Oct. 2021

Making Spatial Trade-Offs Using Multiobjective $\mathcal{H}_2$ Synthesis

Tijs Donkers

DCT 2006-109

Internship report

Supervisor: Matthijs Boerlage

Technische Universiteit Eindhoven
Department Mechanical Engineering
Dynamics and Control Technology Group

Eindhoven, September 2006

Abstract

In feedback control design, the Bode sensitivity integral dictates that trade-offs must be made frequency-wise. For multiple-in-multiple-out (MIMO) systems, trade-offs can also be made spatially (element-wise). To do this using 'norm based' control methods, such as $\mathcal{H}_2$ or $\mathcal{H}_\infty$, one can only change weights on the inputs and the outputs of the systems. This way, it is not possible to exploit all spatial design freedom, e.g. it is not possible the put a lot of weight on one particular element of the transfer function matrix. To exploit all spatial design freedom, multiobjective methods can be employed.

In the first part of this report, we discuss a multiobjective $\mathcal{H}_2$ synthesis method, which uses a vector valued performance criterion. This criterion is handled using the notion of Pareto optimality. Using this, the multiobjective problem is converted into a 'simultaneous model matching problem' by using the Youla parametrisation. Finally, a closed form expression is formulated to calculate one (out of many) Pareto optimal compensator.

Second, we show that a $\mathcal{H}_2$ optimal control problem can be formulated by giving a frequency domain interpretation to the 'classical' LQG optimal control problem. Then, given a SISO system, it is shown that this method of multiobjective control synthesis is capable of trading off competing objectives.

Then, this method is extended to MIMO systems, so spatial design freedom can be studied. First, we show that it is possible for a system with a non-minimum-phase zero (NMPhZ) to shift the effect of the NMPhZ from one output to another output. Then, the multiobjective problem is formulated for a 2×2 system, with one (real) NMPhZ. Four objectives can be distinguished: from each input to each output, which can be traded off against each other.

Spatial design freedom is depicted by a curve of all Pareto optimal compensators. This curve shows that the output in which the zero is most present, is more restricted in bandwidth (resulting in a higher minimal 2-norm). Moreover, it can be seen that decoupling of the system decreases performance of the other loops mostly in the direction which the NMPhZ is present.

Subsequently, a similar multiobjective problem is formulated for a 2×2 modal MIMO system, which is is not fundamentally restricted in bandwidth, but restricted due to robustness specifications. The curve of Pareto optimal compensators shows that both outputs are equally restricted and that the cost of decoupling is equal in both outputs.

Conclusively, it can be said, that multiobjective control synthesis can be used to make spatial trade-offs. However, we were unable to apply this methodology to more realistic systems. Thus, the implementation need some improvements.

Table Of Contents

Nomenclature

$\mathcal{L}_2$	Hilbert space of matrix-valued functions which are square integrable on the imaginary axis, including at ∞. The inner product, and the norm induced by this inner product are as follows: $\langle F, G\rangle = \int_{-\infty}^{\infty} \text{trace}[F(j\omega)G^*(j\omega)]d\omega$ (* denotes complex conjugate transposed) $\|F\|_2 = \sqrt{\langle F, F\rangle}$
$\mathcal{H}_2$	Subspace of $\mathcal{L}_2$, with functions that are analytic in the open right-half plane.
$\mathcal{H}_2^\perp$	Subspace of $\mathcal{L}_2$, with functions that are analytic in the open left-half plane. (Orthogonal to $\mathcal{H}_2$.)
$\mathcal{H}_\infty$	Banach space of matrix-valued functions that are bounded on the imaginary axis and analytic in the open right-half plane.
$\mathcal{R}$	Used as a prefix; it denotes real rational.
$P_{\mathcal{H}_2}$	Orthogonal projection from $\mathcal{L}_2$ to $\mathcal{H}_2$.
$G^\sim(s)$	$= G^T(-s)$
$\left[\begin{array}{c\|c} A & B \\ \hline C & D \end{array}\right]$	shorthand for state-space realisation $= C(sI - A)^{-1}B + D$.
$A \otimes B$	Kronecker product: $[a_{ij}B]$
vec(A)	Vector formed by stacking columns of A.
$\mathcal{F}_l$	Lower linear fractional transformation.

Chapter 1

Introduction

In feedback control design, the Bode Sensitivity Integral dictates that analytical (frequency wise) trade-offs must be made (e.g. see [5]). This means that reducing sensitivity at one frequency implies that it is amplified at other frequencies. This holds for single-in-single-out (SISO) systems, but also for multiple-in-multiple-out (MIMO) systems. This sensitivity integral can be generalised to MIMO systems by using the determinant or the singular values (see e.g. [11]). For a stable system, the sensitivity integral is as follows:

$$\int_0^\infty \ln |\det\left(S(j\omega)\right)| d\omega = \sum_j \int_0^\infty \ln \sigma_j\left(S(j\omega)\right) d\omega = 0 \tag{1.1}$$

This integral shows that for MIMO systems another kind of trade-off can be made, namely: spatially (element wise). By reducing one singular value of S, the others are amplified. Because singular values are associated with an input and output direction, this results in a spatial trade-off. In addition, in [6] is illustrated that if the system has unstable poles or non-minimum phase zeros (NMPhZ) in non-canonical directions, these effects can be shifted toward a certain input or output, respectively. Again, a spatial trade-off.

Some conventional control design methods make use of decoupling, where a $n \times n$ MIMO system is converted into n SISO systems. For some systems, this decoupling does not decline optimality. In many systems, the decoupling reduces spatial design freedom. Control design methods that do not require decoupling can be 'norm based' methods, such as $\mathcal{H}_2$ or $\mathcal{H}_\infty$. However, with these norm based techniques it is not transparent how trade-offs are made spatially for they map multivariable system properties in scalar measures. In this report, we will further study $\mathcal{H}_2$ control, and try to overcome this limitation induced by using a scalar performance measure.

1.1 Limitations of regular $\mathcal{H}_2$ control

Like we already mentioned above, in 'regular' $\mathcal{H}_2$ control, a multivariable system is mapped into a scalar norm. Within this norm, some elements can be made more significant by applying input and output weights. However, by applying these weights, it is not possible to influence a single element individually. In this section, we will therefore show that it is in principle not possible to exploit all spatial freedom using regular $\mathcal{H}_2$ optimal control.

First consider the case were diagonal input and output weights are used. For simplicity, we consider a 2×2 system $G(s)$, with input weight $W_i = \text{diag}(1, a)$ and output weight $W_o = \text{diag}(1, b)$. The 2-norm of this system would be:

$$\|W_i G W_o\|_2 = \left\| \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & b \end{pmatrix} \right\|_2 = \sqrt{G_{11}^2 + (bG_{12})^2 + (aG_{21})^2 + (abG_{22})^2} \tag{1.2}$$

From (1.2) can be seen that it is e.g. not possible to put a lot of weight on element G_{12} without influencing either G_{22} or G_{21}. If we use full weighing matrices instead of diagonal ones, the same still applies, although less transparent. First consider that the input and output weights have a singular value decomposition $W_i = U_i \Sigma_i V_i^T$ and $W_o = U_o \Sigma_o V_o^T$. Note that Σ_i and Σ_o are again diagonal, and U_i, U_o, V_i, and V_o are unitary matrices, thus have norm 1. Then, the 2-norm is:

$$\|W_i G W_o\|_2 = \|U_i \Sigma_i \underbrace{V_i^T G U_o}_{\tilde{G}} \Sigma_i V_o^T\|_2 = \left\| \begin{pmatrix} \sigma_{i1} & 0 \\ 0 & \sigma_{i2} \end{pmatrix} \begin{pmatrix} \tilde{G}_{11} & \tilde{G}_{12} \\ \tilde{G}_{21} & \tilde{G}_{22} \end{pmatrix} \begin{pmatrix} \sigma_{o1} & 0 \\ 0 & \sigma_{o2} \end{pmatrix} \right\|_2 \tag{1.3}$$

The result is equivalent to (1.2). Again, it is not possible to stress a single element without influencing the others. If we want to be able to do so, other methods need to be employed, in which the performance criterion is intrinsic vector valued, instead of a scalar norm.

1.2 Problem formulation

A solution to this can be found in the field of multiobjective control synthesis, where indeed vector valued performance criteria are used. These vector valued performance criteria could be solved using Linear Matrix Inequalities (LMIs). However, solving LMIs has proved to be difficult when applied to realistic systems (see e.g. [3]). An alternative to this can be found in [7], where a method for solving a multiobjective $\mathcal{H}_2$ synthesis problem is discussed. This particular method uses the Youla parametrisation and a projection in a Hilbert space.

This leads to the following problem formulation:

> **How can spatial trade-offs be made in a transparent, straightforward way, using the multiobjective $\mathcal{H}_2$ synthesis proposed in [7]?**

In order to study this problem, the following questions need to be answered:

- How does the method discussed in [7] work, and (how) can it be implemented in a numerically stable way?
- How is a $\mathcal{H}_2$ optimal control problem formulated?
- How can multiobjective control be used to make trade-offs in a transparent way?
- How do we visualise information about spatial design freedom in a easy to interpret way?
- What spatial design freedom can we expect from systems, using the theory from [6]?

1.3 Outline of this report

This report is organised as follows: In chapter 2, the aforementioned multiobjective $\mathcal{H}_2$ synthesis will be discussed. We elaborate on both principle and implementation in state-space. Next, a SISO system will be used to show how a $\mathcal{H}_2$ problem is formulated, and how a multiobjective method enables us to make trade-offs, however, not spatially. In chapter 4, the multiobjective problem will be put into a framework, that can be used to make spatial trade-offs for MIMO systems. A system with a NMPhZ will be used to demonstrate the spatial design freedom. In the subsequent chapter, the same will be done for a modal system. In the final chapter, some conclusions will be drawn and some recommendations will be given for future research.

Chapter 2

Multiobjective $\mathcal{H}_2$ Synthesis

In this chapter, the algorithm for solving the multiobjective optimal control is discussed. The outline of this chapter is more or less the same as [7], but more elaborately. In order to solve a multiobjective optimal control problem, we need a suitable notion of optimality. This next section will be devoted to that. Subsequently, we can formulate the control synthesis problem as a convex optimisation problem. Then, we are able to derive an expression for the optimum (or optima as it will soon be clear). Subsequently, we make use of some Kronecker algebra (see [2]) and the inner/outer factorisation to obtain a closed form expression for the optimum. Methods for implementing this in state-space are given in the latter section of this chapter.

2.1 Pareto optimality

In optimisation theory, the concept op Pareto optimality can be used to handle multiple objective problems (see [1]). It is defined as follows: let $\mathbb{X}$ be a arbitrary nonempty set, and $f_i : \mathbb{X} \to \mathbb{R}^+, i \in s$ be s nonnegative functionals defined on $\mathbb{X}$. On this set, a point $x^o \in \mathbb{X}$ is said to be Pareto optimal with respect to the vector valued criterion $f = (f_1, f_2, \ldots, f_s)$ if there does not exist an $x \in \mathbb{X}$ such that:

$$f_i(x) \leq f_i(x^o), \qquad \text{for all } i \in s, \text{and}$$

$$f_k(x) < f_k(x^o), \qquad \text{for some } k \in s$$

To illustrate this, we consider a criterion $f(x) = (f_1(x), f_2(x))$. Point x^o is Pareto optimal, if it minimises e.g. $f_1(x^o)$, and, for that x^o, makes $f_2(x^o)$ as minimal as possible. This is shown in figure 2.1 (taken from [9]). Note that there are in principle an infinite number of optimal solutions.

One approach to solve the optimisation problem is to parameterise this optimal set by the solution to a parametric scalar-valued optimisation problem. This results into the following (see [7]): suppose that $\mathbb{X}$ is a normed, linear space and each component of $f = (f_1, f_2, \ldots, f_s)$ is convex in $\mathbb{X}$. Let:

$$\Lambda = \left\{ \lambda \in \mathbb{R}^s \ : \ \lambda_i \leq 0 \ : \ \sum_{i=1}^{s} = 1 \right\}, \qquad s = 1, 2, \ldots, k \tag{2.1}$$

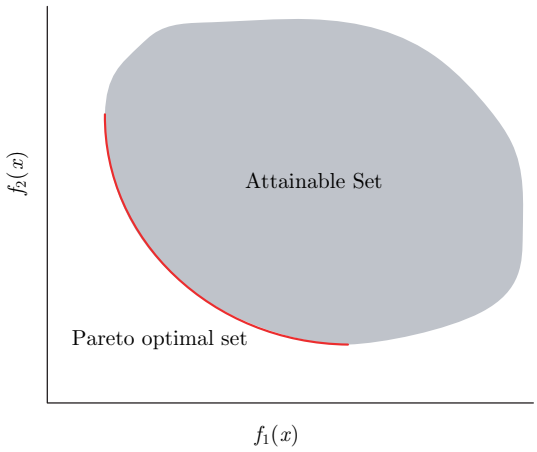

Figure 2.1: Pareto optimality

Then, for each $\lambda \in \Lambda$ the following scalar valued optimisation problem results into a Pareto optimal point:

$$\inf \left\{ \lambda^T f(x) : x \in \mathbb{X} \right\} \tag{2.2}$$

2.2 Formulation of the optimisation problem

In (post)modern control, the standard plant paradigm is used to formulate the control synthesis problem (see figure 2.2). The signals $w_1, w_2, \ldots, w_s, u, y, z_1, z_2, \ldots, z_s$ are vector valued functions of time (or frequency). For each $k \in s$, w_k denotes an exogenous input vector (e.g. setpoints, disturbances, measurement noise), and z_k are the outputs to be regulated. The vector signals u and y are the controlled inputs and outputs, respectively. The transfer matrices P and K denote the given standard plant and compensator. It is assumed that P can be stabilised and that the open loop from u to y is strictly proper.

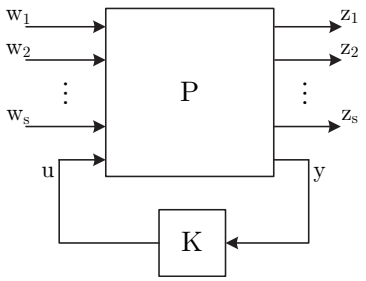

Figure 2.2: The standard plant

For each $k \in s$, T_k is the closed loop transfer function from w_k to z_k, the control goal is to find a compensator K, such that K internally stabilises P, and K is Pareto optimal with respect to the vector valued cost function:

$$\min_K J(K), \qquad \text{with:} \qquad J(K) = (\|T_1(K)\|_2, \|T_2(K)\|_2, \ldots, \|T_s(K)\|_2) \tag{2.3}$$

2.2.1 Conversion to a 'simultaneous model matching problem'

So far, we defined the control synthesis problem as a multiple objective optimisation problem. Now, we link the aforementioned optimisation problem to the notion of Pareto optimality. First, the compensator should internally stabilise the given plant. (In terms of section 2.1: it should be a in the linear space $\mathbb{X}$.) Furthermore, the optimisation problem should be convex in $\mathbb{X}$. A way of fulfilling both conditions is to introduce the Youla parametrisation of all internally stabilising compensators, (e.g. see [12]). By doing so, the closed loop transfer functions turn out to be affine functions of the free parameter Q, that is:

$$T_k = R_k - U_k Q V_k, \qquad k \in s \tag{2.4}$$

where R_k, U_K, and V_k are transfer function matrices in $\mathcal{RH}_\infty$, and depend only on P. It will be shown later how these transfer function matrices are obtained, and how the compensator K can be derived from this Q.

After applying (2.4) to (2.3), the problem turns into a 'simultaneous model matching problem', because ideally it would be the goal to match the product $U_k Q V_k$ such that it equals R. In that case, the result would be zero, hence, the norm would be zero. The simultaneous model matching problem is defined as follows:

$$\min_Q I(Q), \qquad \text{with:} \qquad I(Q) = \left(\|T_1(Q)\|_2^2, \|T_2(Q)\|_2^2, \ldots, \|T_s(Q)\|_2^2\right) \tag{2.5}$$

The algorithm for solving the 'simultaneous model matching problem' is obtained by directly applying (2.2). First fix $(\lambda_1, \lambda_2, \ldots, \lambda_s) \in \Lambda$ and obtain transfer function matrices $R_{k\lambda}$ and $U_{k\lambda}$ according to:

$$R_{k\lambda} = \sqrt{\lambda_k} R_k \qquad \text{and:} \qquad U_{k\lambda} = \sqrt{\lambda_k} U_k \tag{2.6}$$

Then find a $Q^o \in \mathcal{RH}_\infty$ such that:

$$\sum_{k=1}^{s} \|R_{k\lambda} - U_{k\lambda} Q^o V_k\|_2^2 = \inf \left\{ \sum_{k=1}^{s} \|R_{k\lambda} - U_{k\lambda} Q V_k\|_2^2 \ : \ Q \in \mathcal{RH}_\infty \right\} \tag{2.7}$$

If this Q^o exist, the Pareto optimal controller can be obtained.

2.3 Achieving a Pareto optimal point

After we formulated the control problem as a optimisation problem, and linked it to a proper definition of optimality, we can discuss finding a Pareto optimum point for a particular choice of $(\lambda_1, \lambda_2, \ldots, \lambda_s) \in \Lambda$. Note that changing $\lambda \in \Lambda$ yield different compensators. Before actually giving an expression for the optimum, a necessary condition is discussed.

2.3.1 Necessary conditions for existence of an optimum

The solution to the model matching problem had a finite $\mathcal{H}_2$-norm, if $U_{k\lambda}$ and V_k have full column and row rank on the extended imaginary axis, respectively, and if $R_{k\lambda}$ and $U_{k\lambda}QV_k$ match perfectly at the point $j\omega = \infty$. In order to achieve this, Q is partitioned into $Q = \hat{Q} + Q_\infty$. Then, Q_∞ is chosen such that:

$$R_{k\lambda}(\infty) = U_{k\lambda}(\infty)Q_\infty V_k(\infty), \quad \text{for all } k \in s \tag{2.8}$$

When this Q_∞ is not unique for all $k \in s$, the Pareto optimal solution does not exist. If it does, let: $Q^o = \hat{Q}^o + Q_\infty$, then:

$$\hat{R}_{k\lambda} = R_{k\lambda} - U_{k\lambda}Q_\infty V_k, \qquad \text{with:} \quad \hat{R}_{k\lambda}, \hat{Q} \in \mathcal{RH}_2 \tag{2.9}$$

2.3.2 Achieving the infinum

Now, we can write (2.7) more conveniently by substituting (2.9) herein:

$$\inf_Q \left\{ \sum_{k=1}^{s} \|R_{k\lambda} - U_{k\lambda}QV_k\|_2^2 \right\} = \inf_Q \left\{ \sum_{k=1}^{s} \|\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}V_k\|_2^2 : \hat{Q} \in \mathcal{H}_2^{m\times p} \right\} \tag{2.10}$$

To solve this optimisation problem, an analogy is used from the Euclidian space. The minimisation of $\|\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}V_k\|_2^2$ with $U_{k\lambda}$ and V_k having full column and row rank on the extended imaginary axis, respectively, is analogous to a common least-square problem: $\min \|b - Ax\|_2^2$ in the Euclidian space, where it is the goal to minimise the distance between $b - Ax$ and Ax. The minimum 'distance' in this case is where $\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}V_k$ and $U_{k\lambda}\hat{Q}V_k$ are orthogonal [10]:

$$\sum_{k=1}^{s} \langle \hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k, U_{k\lambda}\hat{Q}V_k \rangle = 0, \quad \text{for all } \hat{Q} \in \mathcal{H}_2^{m\times p} \tag{2.11}$$

This can be rewritten as:

$$\begin{aligned}\sum_{k=1}^{s}\langle \hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k, U_{k\lambda}\hat{Q}V_k\rangle &= \sum_{k=1}^{s}\frac{1}{2\pi}\int_{-\infty}^{\infty}\text{trace}\left[(\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k)(U_{k\lambda}\hat{Q}V_k)^{\sim}\right]d\omega \quad \Leftrightarrow \\ &= \sum_{k=1}^{s}\frac{1}{2\pi}\int_{-\infty}^{\infty}\text{trace}\left[U_{k\lambda}^{\sim}(\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k)V_k^{\sim}\hat{Q}^{\sim}\right]d\omega \quad \Leftrightarrow \\ &= \sum_{k=1}^{s}\langle U_{k\lambda}^{\sim}\left(\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k\right)V_k^{\sim}, \hat{Q}\rangle = 0 \end{aligned} \tag{2.12}$$

In section A.1 of appendix A it is shown that this latter expression is equivalent to the following projection on the $\mathcal{H}_2$ space:

$$P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}U_{k\lambda}^{\sim}U_{k\lambda}\hat{Q}V_kV_k^{\sim}\right) = P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}U_{k\lambda}^{\sim}\hat{R}_{k\lambda}V_k^{\sim}\right) \tag{2.13}$$

where $P_{\mathcal{H}_2}$ denotes an orthogonal projection on the $\mathcal{H}_2$ space. An orthogonal projection on the $\mathcal{H}_2$ space is nothing more than taking the stable part.

2.4 Obtaining a closed form expression for the infinum

So far, we have found an expression for the infinum for the 'simultaneous model matching problem' via a projection on the $\mathcal{H}_2$ space. In this section, a basis for constructing a state-space solution to the problem is presented by first resorting to Kronecker algebra and then using the inner/outer factorisation.

2.4.1 Solutions via Kronecker algebra

The first step in obtaining a closed form expression is to apply some Kronecker algebra (see [2]). The idea is that the matrix equation can be converted to a problem in which the unknown is a vector. We define:

$$W_{k\lambda} = V_k^T \otimes U_{k\lambda} \tag{2.14}$$

$$W = \begin{pmatrix} W_{1\lambda}^T & \dots & W_{s\lambda}^T \end{pmatrix}^T \tag{2.15}$$

$$\text{vec}(\hat{R}) = \begin{pmatrix} \text{vec}(\hat{R})_{1\lambda}^T & \dots & \text{vec}(\hat{R})_{s\lambda}^T \end{pmatrix}^T \tag{2.16}$$

Then, as derived in section A.2 of appendix A, (2.13) can be written as follows:

$$P_{\mathcal{H}_2}\left(W^{\sim}W\,\text{vec}(\hat{Q})\right) = P_{\mathcal{H}_2}\left(W^{\sim}\,\text{vec}(\hat{R})\right) \tag{2.17}$$

Because for all $k \in s$, $U_{k\lambda}$ and $V_k^{\sim}$ have full column rank on the extended imaginary axis, so does W_k. Thus, W does have full column rank on the extended imaginary axis.

2.4.2 The inner/outer factorisation

The last step in the derivation is applying an inner/outer factorisation. Before doing so, we first need to define inner and outer transfer functions. According to [7], a transfer function P is inner if: $P^{\sim}P = I$, and a transfer function is outer if it is proper and stable, and that its inverse is also proper and stable. In [12] is shown that a transfer function matrix can be written uniquely as:

$$W = W_i W_o \tag{2.18}$$

if W has full column rank on the extended imaginary axis. This, we had already concluded. Because $W_i^{\sim} W_i = I$, the following holds:

$$W_o^{\sim} W_o = W^{\sim} W \tag{2.19}$$

When applying the inner/outerfactorisation to (2.17):

$$\begin{aligned} P_{\mathcal{H}_2}\left(W_o^{\sim} W_o \operatorname{vec}(\hat{Q})\right) &= P_{\mathcal{H}_2}\left(W_o^{\sim} W_i^{\sim} \operatorname{vec}(\hat{R})\right) \quad \Leftrightarrow \\ P_{\mathcal{H}_2}\left(W_o \operatorname{vec}(\hat{Q})\right) &= P_{\mathcal{H}_2}\left(W_i^{\sim} \operatorname{vec}(\hat{R})\right) \end{aligned} \tag{2.20}$$

From this expression, and the fact that $\hat{Q} \in \mathcal{RH}_2$ follows:

$$\operatorname{vec}(\hat{Q}) = W_o^{-1} P_{\mathcal{H}_2}\left(W_i^{\sim} \operatorname{vec}(\hat{R})\right) \tag{2.21}$$

This latter expression is the desired closed-form expression for finding an Pareto optimal point. How this can be implemented in state-space, and how a compensator can be calculated from this expression, will be discussed next.

2.5 Implementation in state-space

In the previous section is shown that a closed-form expression can be used to calculate a Pareto optimal compensator. In this section, state-space realisations required to implement this procedure in MATLAB are discussed. The required realisations are: $\operatorname{vec}(R_k)$ and $W_{k\lambda}$, and they can be derived from the state-space formulae of the Youla parametrisation. Then, we discuss a method of implementing the inner/outer factorisation in state-space. After that, it should be straightforward to implement the procedure in MATLAB.

2.5.1 Obtaining a compensator K from the Youla parameter Q

In section 2.2.1, it was mentioned that the Youla parametrisation of all internally stabilising compensators could be obtained from the standard plant P. Now, we show how this can be implemented in state-space. As a starting point, we consider the standard plant once again:

$$P = \left[\begin{array}{c|ccc} A & B_1 & \dots & B_s & B_u \\ \hline C_1 & D_{11} & \dots & D_{1s} & D_{1u} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_y & D_{y1} & \dots & D_{ys} & D_{yu} \end{array}\right] \tag{2.22}$$

where A, B_k, B_u, C_k, C_y, D_{kk}, D_{ku}, D_{yk}, with $k \in s$, are correctly partitioned matrices from the realisation of the plant P. Subscripts $1, \dots, s$ denote the exogenous input channels w_k and output channels z_k, and subscript u and y denote the controlled input and measured output, respectively. According to [12], the parametrisation of all internally stabilising compensators can be parameterised as depicted in figure 2.3.

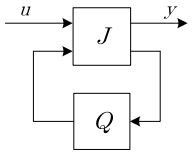

Figure 2.3: The Youla Parametrisation as a linear fractional transformation

It can be seen that an initial stabilising compensator J is used to parameterise all stabilising compensators. According to [12], that this initial controller is given by the following realisation:

$$J = \left[\begin{array}{c|cc} A + B_u F + H C_y & -H & B_u \\ \hline F & 0 & I \\ -C_y & I & 0 \end{array}\right] \tag{2.23}$$

Where F and H are such that $A + B_u F$ and $A + H C_y$ are stable. F and H can be calculated by solving two Riccati equations (see [12]). Note that this initial controller J only depends on A, B_u, and C_y. Thus, for every $k \in s$, this initial compensator can be taken the same. The final compensator can then obtained by taking a Lower Linear Fractional Transformation:

$$K = \mathcal{F}_l(J, Q) = J_{11} + J_{12} Q (I - J_{22} Q)^{-1} J_{21}$$

As introduced in 2.2.1, realisations R_k, U_k and V_k can be obtained using this initial compensator. They will not be directly implemented in MATLAB. For completeness, they are given in appendix B.

2.5.2 Realisations $\mathrm{vec}(R_{k\lambda})$ and $W_{k\lambda} = V_k^T \otimes U_k$

The realisations given in section 1 of appendix B could be implemented in MATLAB, and then perform the Kronecker algebraic manipulations described in section 2.4.1. This will result into a time consuming, sensitive algorithm, because Kronecker products result in large matrices, which are sensitive to numerical errors. In [4], an alternative is proposed. Namely, we can calculate both $\mathrm{vec}(R_k)$ and W_k in a single realisation, which can then be partitioned:

$$\begin{array}{l}\big(\mathrm{vec}(R_k) \quad W_k\big) = \\ \left[\begin{array}{cc|cc} (A+HC_y)^T \otimes I_{p1} & C_y^T \otimes (C_k + D_{ku}F) & \mathrm{vec}(C_1) & -C_y^T \otimes D_{ku} \\ 0 & I_{p2} \otimes (A + B_u F) & \mathrm{vec}(-H) & -I_{p2} \otimes B_u \\ \hline \big((B_k + HD_{yk})^T \otimes I_{p1}\big) & \Big(D_{yk}^T \otimes (C_k + D_{ku}F)\Big) & \mathrm{vec}(D_{kk}) & \Big(-D_{yk}^T \otimes D_{ku}\Big) \end{array}\right] \end{array} \tag{2.24}$$

where I_{p1} and I_{p2} are identity matrices with the sizes according to the vector length of k and y, respectively. A full derivation of these realisations are given in appendix B. Now, we can easily calculate $\mathrm{vec}(R_{k\lambda})$ and $W_{k\lambda}$ as defined in section 2.4.1:

$$\mathrm{vec}(R_{k\lambda}) = \sqrt{\lambda_k}\,\mathrm{vec}(R_k), \qquad \text{and} \qquad W_{k\lambda} = \sqrt{\lambda_k} W_k \tag{2.25}$$

2.5.3 Implementation of the inner/outer factorisation

In this section, the implementation of the inner/outer factorisation is discussed. We already defined the notion of inner and outer, but so far, we did not mention that an inner/outer factorisation is just a special form of a coprime factorisation. In [12], efficient state-space formulae are derived to make an inner/outer factorisation. Given a state-space realisation of W and $R^{1/2}R^{1/2} = R$, the inner and outer factors are obtained as follows:

$$\begin{pmatrix} W_i \\ W_o^{-1} \end{pmatrix} = \left[\begin{array}{c|c} A - BX & BR^{-1/2} \\ \hline C - DX & DR^{-1/2} \\ -X & R^{-1/2} \end{array}\right] \tag{2.26}$$

where $R = D^*D \geq 0$, and $X = -R^{-1}(B^*Y + D^*C)$. Y is obtained by finding the positive definite solution of the following algebraic Riccati equation:

$$\big(A - BR^{-1}D^*C\big)^* Y + Y\big(A - BR^{-1}D^*C\big) + Y(-BR^{-1}B^*)Y + (C^*(I - DR^{-1}D^*)C) = 0$$

2.6 Summary

In order to get complete picture of the algorithm, we now summarise the steps that have to be taken:

- Fix $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_s) \in \Lambda$, according to (2.1);
- Correctly partition the standard plant P and calculate and initially stabilising compensator J, using (2.23);
- Calculate $\text{vec}(R_{k\lambda})$ and $W_{k\lambda}$, for each $k \in s$;
- Form W and $\text{vec}(R)$ according to (2.15) and (2.16), respectively;
- Apply an inner/outer factorisation to W;
- Calculate the Pareto optimal Q from (2.21), and the compensator K from this Q.

By changing $\lambda \in \Lambda$, all Pareto optimal compensators are obtained.

2.7 Conclusions

In this chapter, a method for multiobjective control has been discussed. Using this method, it is possible to trade off various objectives. Its implementation can be done in state-space, requiring algebraic Riccati equations to be solved.

Chapter 3

Making Trade-offs in a SISO System

In this chapter, multiobjective control synthesis is applied to a single-in-single-out (SISO) system. Obviously, there is no spatial freedom in a SISO system, but there is design freedom is trading off competing requirements, (e.g. low control sensitivity versus low process sensitivity). First, we obtain a formulation for the standard plant. Subsequently, the multiobjective control problem can then be formulated. Results of the synthesis, using the method discussed in chapter 2, are shown in the last section of this chapter. These results will turn out to be trivial, but this chapter only acts as a steppingstone to more complicated systems.

3.1 Obtaining a standard plant formulation

3.1.1 LQG control as a $\mathcal{H}_2$ optimal control problem

The generalised plant in a $\mathcal{H}_2$ optimal control problem, can be obtained by giving a frequency domain interpretation to the LQG problem [11]. The LQG problem is as follows: given a stochastic system

$$\begin{array}{rl} \dot{x} = & Ax + Bu + w_d \\ y = & Cx + w_n \end{array}, \qquad \text{where:} \quad E\left\{ \begin{bmatrix} w_d(t) \\ w_n(t) \end{bmatrix} \begin{bmatrix} w_d^T(\tau) & w_n^T(\tau) \end{bmatrix} \right\} = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \delta(t-\tau)$$

The LQG problem is to find a $u = -K(s)y$, such that the following cost function is minimised:

$$J = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(x^T Q x + u^T R u \right) dt \right\}, \qquad \text{with: } Q = Q^T > 0, R = R^T > 0$$

This can be put in an $\mathcal{H}_2$ optimisation frame by defining an output signal z and the input signal w as:

$$z = \begin{bmatrix} Q^{1/2} & 0 \\ 0 & R^{1/2} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \qquad \text{and:} \qquad \begin{bmatrix} w_b \\ w_n \end{bmatrix} = \begin{bmatrix} W^{1/2} & 0 \\ 0 & V^{1/2} \end{bmatrix} w$$

where w is a white noise process of unit intensity. Using this and Parseval's theorem, the LQG cost function becomes:

$$J = E\left\{ \tfrac{1}{2\pi} \int_{-\infty}^{\infty} z^T(j\omega) z(j\omega) d\omega \right\} = \|\mathcal{F}_l(P,K)\|_2^2, \qquad \text{where:} \qquad z(s) = \mathcal{F}_l(P,K) w(s)$$

The standard plant P, belonging to the above stated problem is:

$$P = \left[\begin{array}{c|cc:c} A & W^{1/2} & 0 & B \\ \hline Q^{1/2} & 0 & 0 & 0 \\ 0 & 0 & 0 & R^{1/2} \\ \hdashline -C & 0 & V^{1/2} & 0 \end{array}\right] \tag{3.1}$$

The formulation of the LQG problem given in terms of the standard plant, is shown in figure 3.1. In this case, all weights Q, R, W, and V are independent of frequency. In principle the could be frequency depended. A way of selecting frequency depended weights is proposed in [8].

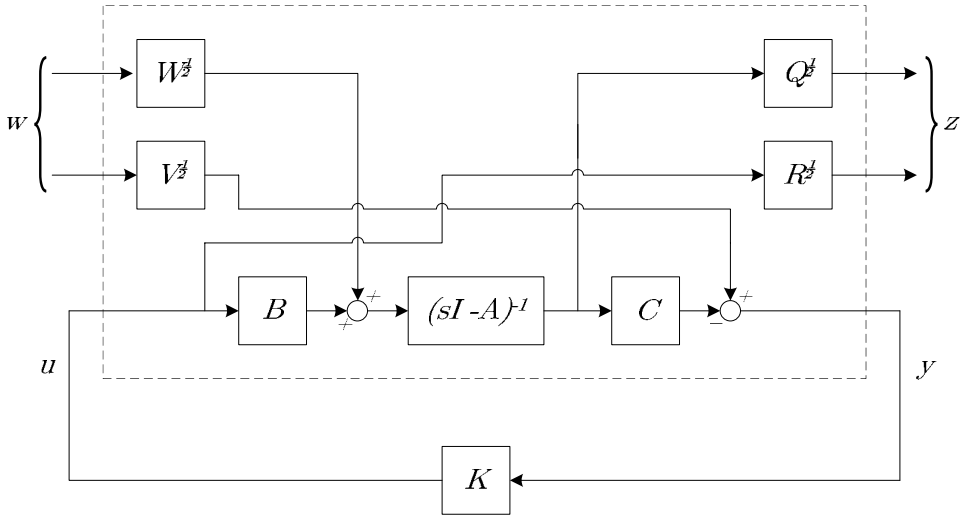

Figure 3.1: The LQG problem formulated in a standard plant paradigm

3.1.2 Standard plant of a mass-spring-damper system

After seeing that $\mathcal{H}_2$ control is basically a frequency domain interpretation of the LQG control, we try to formulate a standard plant for a mass-spring-damper system, given by the following transfer function:

$$G(s) = \frac{10}{s^2 + 0.2\pi s + (2\pi)^2} \tag{3.2}$$

A state-space realisation of this transfer function can be obtained by automated algorithms. A few other adjustments are made to the standard plant, discussed in the previous section: instead of state noise, input noise is assumed, therefore, in terms of LQG control, $W^{1/2} = B$. Furthermore, the measured output is also the controlled output, thus: $Q^{1/2} = C$. Finally, for simplicity: $R^{1/2} = \rho$, and $V^{1/2} = v$. Both are taken $v = \rho = 5 \cdot 10^{-4}$, because standard $\mathcal{H}_2$ control would yield a bandwidth (first 0dB crossing of the open-loop GK) of around 20Hz. By making these adjustments, figure 3.1 simplifies to figure 3.2. The state-space realisation of the standard plant is given in appendix C.

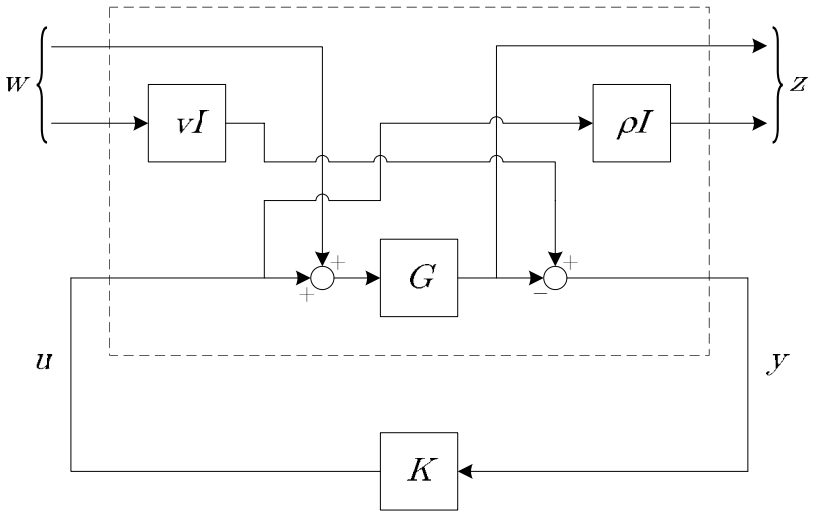

Figure 3.2: The standard plant used for multichannel synthesis

3.2 Formulation of the multiobjective control problem

After obtaining a standard plant for this problem, the multiobjective problem can be formulated. First, we need to define some objectives by channelling inputs and outputs. We choose T_1 to be the closed loop transfer from w_1 to z_1 and T_2 be the closed loop transfer from w_2 to z_2. It can be verified that T_1 represents the (weighted) process sensitivity, given by $(I+GK)^{-1}G$, and that T_2 represents the (weighted) control sensitivity, given by $K(I+GK)^{-1}$. These are obviously competing objectives; pursuing low process sensitivity yields a high gain compensator, while pursuing low control sensitivity yields a low gain compensator.

Now we are ready to solve the multiobjective control problem using the algorithm described in chapter 2. We take $\Lambda = \{(\alpha, 1-\alpha) : \alpha \in [0,1]\}$. Note that $\alpha = 1$ is corresponds to minimising $\|T_1\|_2$, and therefore minimising the process sensitivity, while $\alpha = 0$ corresponds to minimising $\|T_2\|_2$, which is equivalent to minimising the control sensitivity. The synthesis of the compensator K is done by using the results of chapter 2.

3.3 Results

In figure 3.3, the process sensitivity and control sensitivity are shown for three (out of many) Pareto optimal compensators. It can be seen that choosing α to be large, results in a low process sensitivity and, thus, in a large bandwidth. The converse is also true: choosing α to be small, results in a low control sensitivity and, thus, in a small bandwidth. This trivial example shows that it is indeed possible to use multiobjective control for making trade-offs.

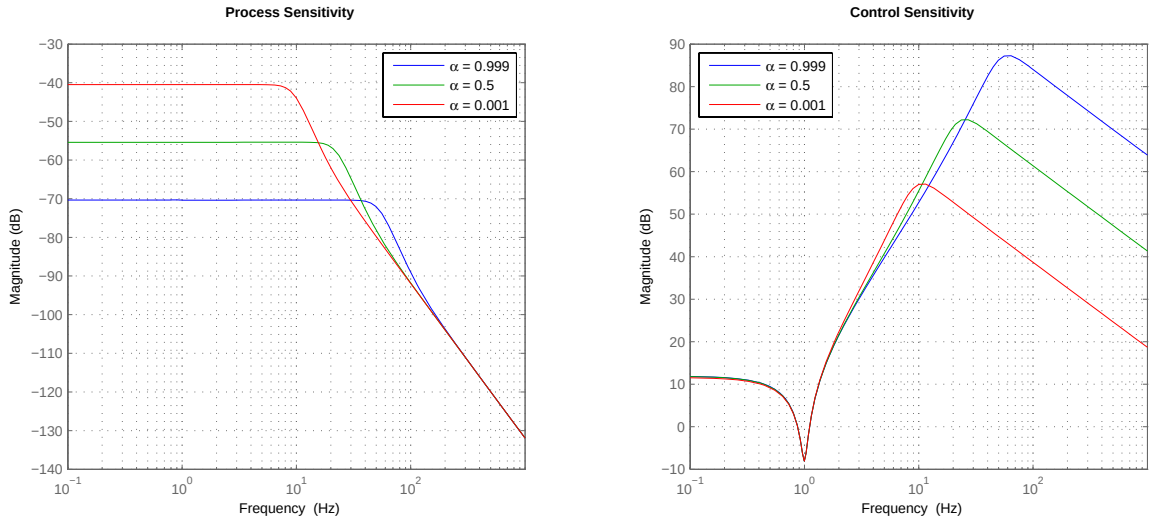

Figure 3.3: Resulting process sensitivity and control sensitivity for various α's.

3.4 Conclusions

This chapter demonstrated how multiobjective $\mathcal{H}_2$ synthesis can be applied to SISO systems. We formulated a $\mathcal{H}_2$ control problem by giving a frequency domain interpretation to the LQG problem. The resulting standard plant can then be transformed to a multiobjective problem by selecting competing objectives. Although the results of this synthesis can be obtained using a conventional $\mathcal{H}_2$ synthesis, the SISO example provides valuable insights in formulating multiobjective control problems.

Chapter 4

Making Spatial Trade-Offs in a MIMO System with a NMPhZ

After showing that it is possible to make trade-offs using multiobjective control synthesis, an extension is made to MIMO systems. In the following two chapters, multiobjective control synthesis, and the notion of Pareto optimality, are employed to make spatial trade-offs. In this particular chapter: for a MIMO system with a non-minimum-phase zero (NMPhZ), a zero in the open right half plane. But before doing so, we show what a NMPhZ imposes on the Bode sensitivity integral, especially for a MIMO systems. Then, such a system is given, and the multiobjective problem is formulated, in a similar way of the previous chapter. Because of this NMPhZ, the system is fundamentally restricted in bandwidth. In the latter section of this chapter, the curve of all Pareto optimal compensators is plotted. In this plot, the spatial freedom, and the effect of the NMPhZ, can be observed.

4.1 Spatial design freedom in a MIMO system

In [5], an extension to the Bode sensitivity integral is made for SISO systems with unstable poles and NMPhZ. For example, given a system, with a single (real) NMPhZ at $s = z$, and two more poles than zeros, the Bode sensitivity integral is given as follows:

$$\int_{-\infty}^{\infty} \ln|S(j\omega)| \frac{z}{z^2+\omega^2} d\omega = 0 \tag{4.1}$$

This shows that a NMPhZ causes larger peaks when the bandwidth approaches z. In [6], this result is further extended to MIMO systems, where the sensitivity integrals are in vector form. Thus, it shows how directionality of either NMPhZ or unstable poles can be exploited in making trade-offs. We will demonstrate the results of this paper by means of a 2×2 system with a single (real) NMPhZ zero at $s = z$ along the vector $h = \begin{pmatrix} h_1 & h_2 \end{pmatrix}$. Let:

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \tag{4.2}$$

be the closed loop sensitivity function. Applying the theory developed in [6] gives the following inequalities:

$$\tfrac{1}{\pi} \int_{-\infty}^{\infty} \ln|h_1 S_{11}(j\omega) + h_2 S_{21}(j\omega)| \frac{z}{z^2+\omega^2} d\omega \geq \ln h_1 \tag{4.3}$$

$$\tfrac{1}{\pi} \int_{-\infty}^{\infty} \ln|h_1 S_{12}(j\omega) + h_2 S_{22}(j\omega)| \frac{z}{z^2+\omega^2} d\omega \geq \ln h_2 \tag{4.4}$$

This shows that there is indeed spatial design freedom in a system with a NMPhZ, if this NMPhZ has a non-canonical direction. Therefore, it is possible shift the effect of the NMPhZ to a certain output.

4.2 Obtaining a standard plant formulation

The problem under consideration is formulated similar to that of chapter 3. Now, we consider a (physically not realisable) system, consisting of a sum of a diagonal mass-spring-damper system, and a rotated second order system with a NMPhZ at $s \approx 2\pi \cdot 100$ and poles at $s = -2\pi(1 \pm 500)$. The poles are present just to make the whole system strictly proper. The system is described as follows:

$$G(s) = \begin{bmatrix} \frac{1000}{s^2+0.8\pi s+(2\pi)^2} & 0 \\ 0 & \frac{1000}{s^2+0.8\pi s+(2\pi)^2} \end{bmatrix} + \begin{bmatrix} \sqrt{k} & -\sqrt{1-k} \\ \sqrt{1-k} & \sqrt{k} \end{bmatrix} \begin{bmatrix} \frac{-5\cdot 10^{-6} s}{\frac{1}{(2\pi\cdot 500)^2}s^2+\frac{2}{2\pi\cdot 500}s+1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{k} & -\sqrt{1-k} \\ \sqrt{1-k} & \sqrt{k} \end{bmatrix}^{-1} \tag{4.5}$$

the pre- and post multiplication with an unimodular matrix with $0 \leq k \leq 1$ enables us to study the behaviour of the NMPhZ in various non canonical directions, without influencing the eigenvalues of the system. The input and output direction of the NMPhZ are both along the vector $h = \begin{pmatrix} \sqrt{k} & \sqrt{1-k} \end{pmatrix}$.

Again, like in chapter 3, the standard plant is obtained by giving a frequency domain interpretation to the LQG problem. We use the same standard plant, and again $W^{1/2} = B$ and $Q^{1/2} = C$. Then, we choose $R^{1/2} = 1 \cdot 10^{-3} I$ and $V^{1/2} = 1 \cdot 10^{-2} I$. The standard plant is given in appendix C.

4.3 Formulation of the multiobjective control problem

Like in chapter 3, we formulate the multiobjective problem by channelling inputs and outputs and selecting weights Λ. Here, we can define four channels:

- T_1 is from w_1 and w_3 to z_1 and z_3
- T_2 is from w_2 and w_4 to z_2 and z_4
- T_3 is from w_1 and w_3 to z_2 and z_4
- T_4 is from w_2 and w_4 to z_1 and z_3

This way, each channels contains a (weighted) combination of process, control and complementary sensitivities for each physical input and output of the system G. Again, we use the algorithm discussed in chapter 2 to solve the multiobjective problem. We take:

$$\Lambda = \left\{(\alpha, 1-\beta-\alpha, \tfrac{1}{2}\beta, \tfrac{1}{2}\beta) \ : \ \alpha \in \langle 0, \beta\rangle \ : \ \beta \in \langle 0, 1\rangle\right\} \tag{4.6}$$

This way, the parameter β can be used to enforce closed loop decoupling as $\beta \to 1$, while the parameter α can be used to shift the weight between the diagonal terms.

4.4 Results

In this section, some results obtained using the multiobjective synthesis are shown. These results will be discussed in the certain order. First, we make some statements based on the results discussed in section 4.1 and verify them by looking at the output sensitivity function. It will become clear that this is not the most transparent way of showing spatial trade-offs. That is why we show this trade-off by showing curves of all Pareto optimal compensators (see section 2.1). Finally, using this Pareto curve, we can show how spatial design freedom changes when the input and output direction of the NMPhZ changes.

4.4.1 Spatial freedom using Pareto optimal compensators

To show how spatial freedom can be exploited, we first fix $k = 0.2$. This gives the system a NMPhZ in the input and output direction $h \approx \begin{pmatrix}0.45 & 0.89\end{pmatrix}$. Then, we first reconsider (4.3) and (4.4). From these inequalities, some statements can be made, which are verified by looking at the output sensitivity (from reference r to error e), shown in figure 4.1. By considering the multivariable Bode sensitivity integral, one could say that:

1. the transfer from reference input r_2 to error e_2, denoted $r_2 \to e_2$, shows the NMPhZ behaviour most clearly;
2. the bandwidth in loop $r_2 \to y_2$ will be lower than loop $r_1 \to y_1$;
3. decoupling will 'cost' more performance at input r_2 than at input r_1, especially when demanding high performance from either output y_2 or y_1, respectively;

In figure 4.1, a bode magnitude diagram is shown of the output sensitivities for several values for α and β. It can be seen in the lower-left quadrant of this figure that loop $r_2 \to e_2$ has the highest peak, and therefore shows more NMPhZ behaviour. This confirms statement 1. Moreover, for a fixed value of β, the bandwidth (first 0dB crossing of the sensitivity function) of loop $r_1 \to e_1$ is higher than loop $r_2 \to e_2$. To confirm this, compare the green line in the upper left quadrant (41Hz) with blue line of the bottom-right quadrant (34Hz), and the cyan line of the upper left quadrant (37.5Hz) with the red line of the lower left quadrant (31Hz). To confirm the third statement, compare the blue and the red line from the upper right quadrant with the green and cyan line from the lower left quadrant of the figure.

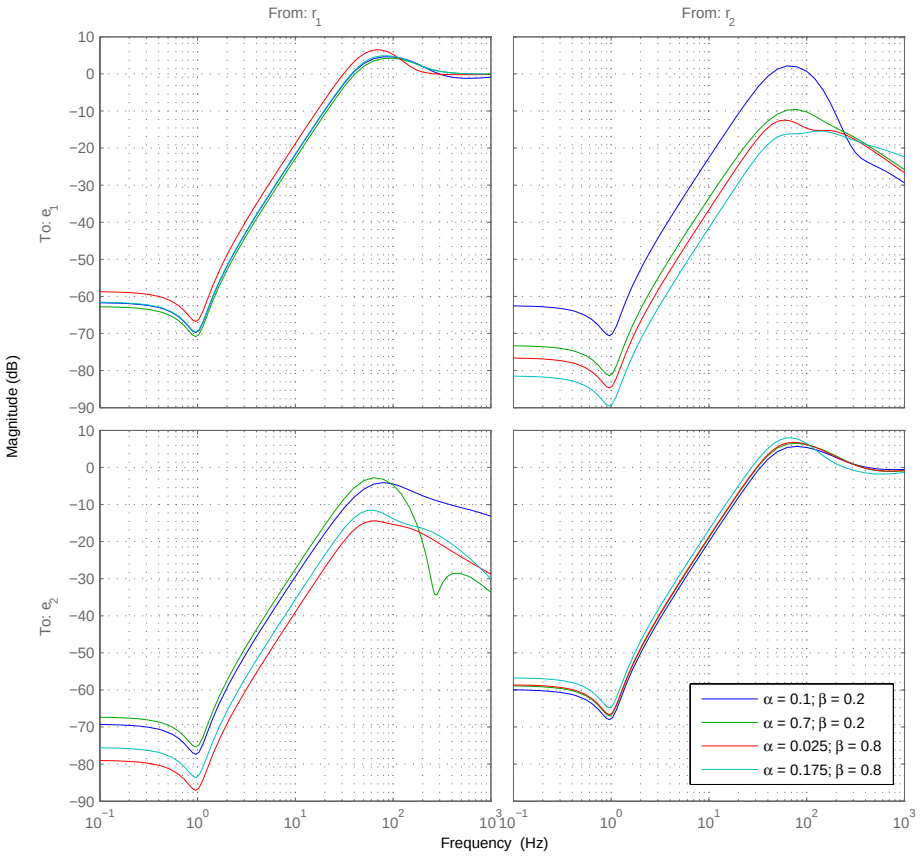

Figure 4.1: Bode (mag) plot of the (output) sensitivity for various α and β.

It can be concluded by looking at the bandwidths, that the statements, made from considering the Bode sensitivity integral, are difficult to distinguish by looking at the output sensitivity. This is mostly due to limited interaction of the loops, and therefore differences are not really significant. Yet, there is a way of showing the spatial design freedom more clearly. This will be discussed next.

4.4.2 Curve of all Pareto optimal compensators for $k = 0.2$

Another way of showing spatial design freedom is by looking at the achieved minimal norm in each channel. This shows achieved performance by a number, rather than a bode diagram. In figure 4.2 such a Pareto curve is shown for the system discussed before ($k = 0.2$). The four lines contain the minimal norm of channel T_1 and T_2 for a fixed value of β. By going along the line, the parameter α is changed, and therefore trading off channel T_1 and T_2. Although it makes the notion of performance more difficult to interpret, it makes spatial design freedom more clearer to see.

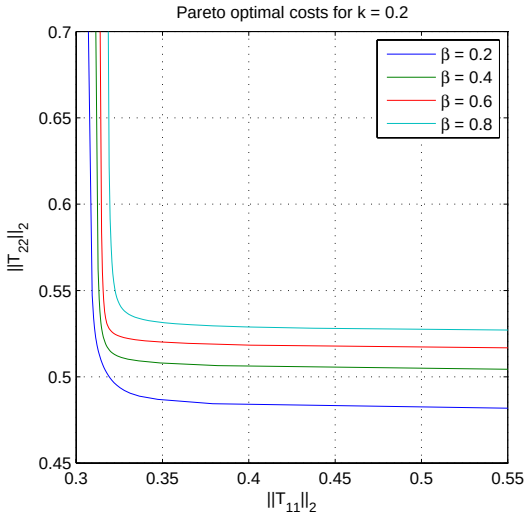

Figure 4.2: Pareto optimal costs

It can be seen much more clearly that the statements made in the previous section hold. In the T_2 channel, where the NMPhZ is more dominant, the lowest achievable 2-norm is higher that in channel T_1. Moreover, the 'cost' of decoupling is lower in channel T_1, where the NMPhZ is present less dominantly. Conclusively, using the notion of Pareto optimality, we can show the spatial design freedom in a transparent way.

4.4.3 Curve of all Pareto optimal compensators for various k's

After this, we can also investigate what happens when the direction of the NMPhZ is changed. In figure 4.3, Pareto curves are shown for various k's. Recall that the direction of the NMPhZ is along the vector $h = \begin{pmatrix}\sqrt{k} & \sqrt{1-k}\end{pmatrix}$. By changing $0 < k \leq 0.5$, the NMPhZ becomes less dominant in channel T_2, and the minimal achievable 2-norm decreases. As a result the minimal achievable 2-norm of channel T_1 increases. Also, the 'cost' of decoupling becomes smaller in channel T_2 and larger in T_1 when k increases.

4.5 Conclusions

In this chapter, we applied the framework provided in chapter 3 to a MIMO system with a NMPhZ. We showed that it is possible to shift the effect of a NMPhZ for one output to another. However, due to limited interaction in our system under study, this effect of is not very significant when looking at the output sensitivity function. The curve of Pareto optimal compensators shows the limitations imposed by the NMPhZ more clearly. Using Pareto optimality, it is possible to make trade-offs spatially, which would not be possible using conventional 'norm based' methods.

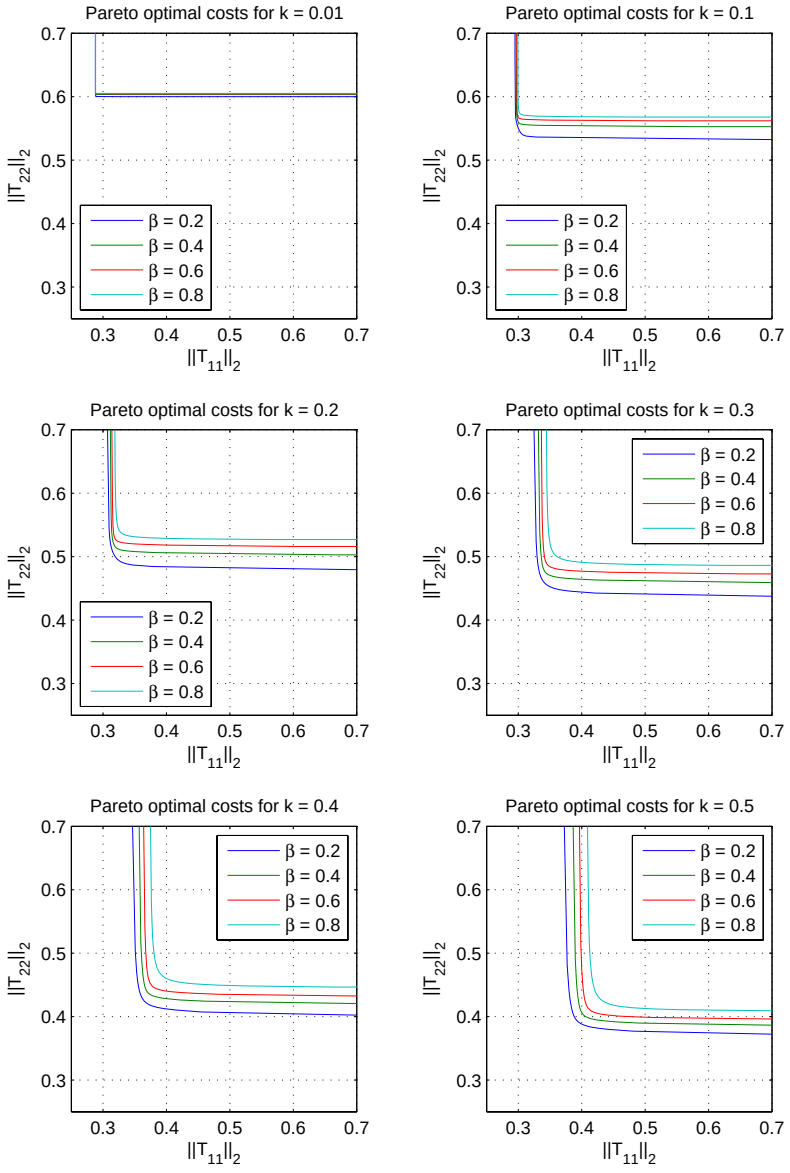

Figure 4.3: Pareto optimal costs for various k's

Chapter 5

Making Spatial Trade-Offs in a Modal MIMO System

In this chapter, multiobjective control synthesis is employed to show spatial design freedom in a modal system. In this system, all poles and zeros are in the open left plane, and therefore theory from [6] cannot applied. There is, however, spatial design freedom, like we introduced in the introduction of this report. However, it is not possible to shift bandwidth limitations to a certain input or output. The methodology used in this chapter is equal to the previous, and therefore, treated more quickly.

5.1 Obtaining a standard plant formulation

The system under consideration is shown in figure 5.1 (blue line). It consists of a mass-spring damper system, with two flexible modes in (both the same) non-canonical direction. This system is, like the one in chapter 4, physically not realisable. The flexible modes put constraints on the bandwidth. However, these constraints are no fundamental ones like the NMPhZ, but constraints due to robustness requirements. To study the influence of coupling, the system is approximately decoupled as follows:

$$\tilde{G} = \begin{pmatrix} 0.998 & 0.071 \\ 0.071 & -0.998 \end{pmatrix} G \begin{pmatrix} 0.655 & 0.756 \\ 0.756 & -0.655 \end{pmatrix} \tag{5.1}$$

These (unimodular) decoupling matrices are obtained from making a singular value decomposition (SVD) of the system evaluated at $s = 1$. This decoupled systems is also shown in figure 5.1. Again, the standard plants are formulated in the same way as in the previous chapter. However, this time $R = 1 \cdot 10^{-3} I$ and $V = 9.5 \cdot 10^{-3}$, for both the coupled and decoupled system. A state-space realisation for both systems can be found in appendix C.

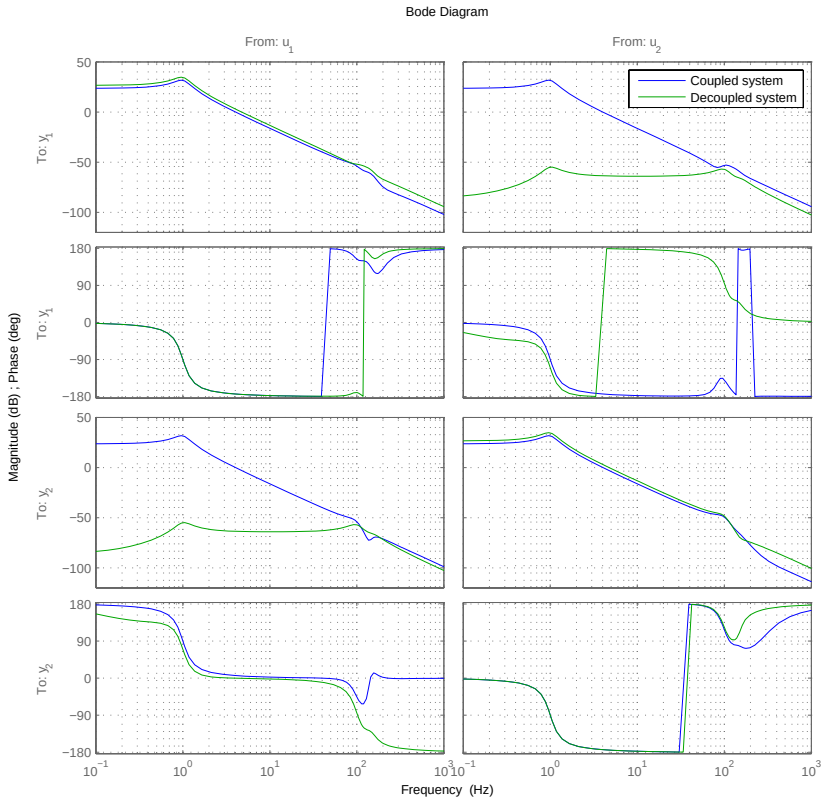

Figure 5.1: Bode diagram

5.2 Results

The multiobjective problem is formulated in the same fashion as in chapter 4. Again, we take:

$$\Lambda = \left\{(\alpha, 1-\beta-\alpha, \tfrac{1}{2}\beta, \tfrac{1}{2}\beta) \;:\; \alpha \in \langle 0, \beta\rangle \;:\; \beta \in \langle 0, 1\rangle\right\} \tag{5.2}$$

Moreover, we again employ the curve of all Pareto optimal compensators for showing the spatial design freedom. These Pareto curves for the coupled and decoupled system are shown in figure 5.2. First, it can be seen for both systems that the minimal attainable 2-norm is equal in both directions. This can be explained by the fact that these systems, unlike a system with a NMPhZ, are not fundamentally limited in performance. The limitations are imposed by robustness specifications. The second fact that can be observed is that there is more spatial design freedom in the coupled case. This can be seen from the shape of the curve.

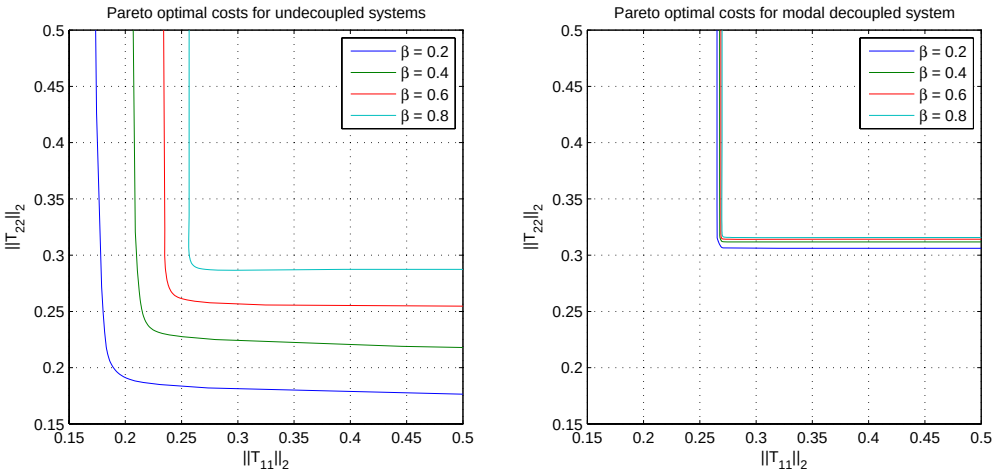

Figure 5.2: Pareto curve for both coupled and decoupled systems

5.3 Conclusions

In this final chapter, a modal MIMO system was put into a multiobjective framework. The fact that this system has no fundamental limitations in achievable bandwidth can be observed by looking at the Pareto curve. We also showed by comparing a coupled and a decoupled system, that the 'amount of spatial freedom' can be observed from the Pareto curve. Unfortunately, when trying to study more realistic systems, it was found out that numerically the algorithm needs to be improved.

Chapter 6

Concluding Remarks and Recommendations

This report has presented a method for making spatial trade-offs in a more transparent way. In this final chapter, conclusions are drawn from the work discussed in this report and some recommendations for future research are made.

6.1 Conclusions

The algorithm, discussed throughout this report is able to trade-off competing requirements. It can be implemented in state-space by solving algebraic Riccati equations. We also elaborated on formulating a $\mathcal{H}_2$ optimal control problem. This is done by giving a frequency domain interpretation to the LQG problem. This optimal control problem can then be transformed into a multiobjective optimal control problem by defining 'channels' containing competing requirements.

Next, we showed how multiobjective control synthesis can be used to make trade-offs in a SISO system. Then, multiobjective $\mathcal{H}_2$ was successfully applied to a MIMO system with a NMPhZ and a modal MIMO system. Both of these systems were not physically realisable. Using the Pareto curve, spatial design freedom can be studied and traded-off in a transparent way.

From the Pareto curve, some fundamental properties of the systems can be visualised. If a system has a NMPhZ in a non canonical direction, the direction in which the zero is most present is most restricted in achievable bandwidth. This is shown by the higher minimum 2-norm in this loop. Also, the cost of decoupling is higher in this loop. By studying the coupled and the approximately decoupled system (coupling gives more spatial design freedom), it is illustrated how the 'amount of spatial freedom' is shown in the Pareto curve. A heavily coupled systems shows smoothly curved lines, while a decoupled systems shows a kneed line.

Conclusively, the following can be said:

> **Multiobjective $\mathcal{H}_2$ synthesis, discussed throughout this report, can be used to make spatial trade-offs in a more transparent, straightforward way. However, future research is required to make a more stable implementation, in order to make it applicable to real systems.**

6.2 Recommendations

Although, the algorithm works well on the systems discussed om this report, it is still impossible to apply it to more realistic systems, (e.g. an 8^{th} order model of a wafer stage.) The current implementation of the algorithm is numerically not very stable. Because Kronecker products result in large matrices, small numerical errors are unavoidable. This causes the realisations of W and vec(R) to be of high order, with lots of repetitive poles, all in slightly different directions. This again causes the projection on the $\mathcal{H}_2$ space to fail. The implementation of this projection (the standard MATLAB routine called `stabsep`), gives an error message and the outcome does not make sense.

One possible solution is as follows: thus far, we developed realisation for W_k and vec(R_k) analytically. It might be possible to derive an analytical expression for the inner/outer factorisation of W in terms of the partitioned matrices of the standard plant P. Maybe, it is even possible to go even further towards the final closed form expression analytically. By doing this, the algorithm would become more stable, and (maybe) applicable to more realistic systems.

Another recommendation that can be made is more general about $\mathcal{H}_2$ control. Although sufficient literature is available on solving a $\mathcal{H}_2$ optimal control problem, little is written about formulating a correct standard plant. The frequency domain interpretation of the LQG problem is still signal based, requiring a lot to be known from the system to be controlled (e.g. disturbances and its directions). Mixed sensitivity $\mathcal{H}_\infty$ optimal control is more user-friendly because it requires desired closed loop transfers to be specified. Moreover, more literature is available on that subject. Thus, more focus should be on setting up a $\mathcal{H}_2$ control problem. This because of in $\mathcal{H}_2$ not just worst case gains in worst case directions are dealt with, like in $\mathcal{H}_\infty$, but all gains in all directions.

Appendix A

Some Extended Derivations

A.1 The infinum as a projection on the $\mathcal{H}_2$ space

By definition, $\mathcal{H}_2$ and $\mathcal{H}_2^\perp$ are both closed subspaces of $\mathcal{L}_2$, such that $\mathcal{L}_2 = \mathcal{H}_2 \oplus \mathcal{H}_2^\perp$, where $\oplus$ stands for direct external sum (see [10]). If we define:

$$X = U_{k\lambda}^{\sim}(\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}^o V_k)V_k^{\sim} \;\in\; \mathcal{L}_2 \tag{A.1}$$

Then, X can be written as $X = X_{\mathcal{H}_2} + X_{\mathcal{H}_2^\perp}$, with $X_{\mathcal{H}_2} \in \mathcal{H}_2$ and $X_{\mathcal{H}_2^\perp} \in \mathcal{H}_2^\perp$. For simplicity, we can now rewrite (2.12) as:

$$\langle X, \hat{Q}\rangle = 0 \tag{A.2}$$

Because the inner product is a linear operation:

$$\langle X_{\mathcal{H}_2} + X_{\mathcal{H}_2^\perp}, Q\rangle = \langle X_{\mathcal{H}_2}, Q\rangle + \langle X_{\mathcal{H}_2^\perp}, Q\rangle = 0 \tag{A.3}$$

The latter term of this expression $\langle X_{\mathcal{H}_2^\perp}, Q\rangle = 0$, because $X_{\mathcal{H}_2^\perp} \notin \mathcal{H}_2$. This means that $\langle X, Q\rangle = 0$ is equivalent to the following orthogonal projection:

$$P_{\mathcal{H}_2}(X) = \sum_i \{\langle X_{\mathcal{H}_2}, Q_i\rangle Q_i\} = 0 \tag{A.4}$$

where $\{Q_i\}$ denotes an orthonormal basis for Q. This proves that (2.12) is equivalent to the following projection on the $\mathcal{H}_2$ space:

$$P_{\mathcal{H}_2}\left(\sum_{k=1}^{s} U_{k\lambda}^{\sim}\left(\hat{R}_{k\lambda} - U_{k\lambda}\hat{Q}V_k\right)V_k^{\sim}\right) \;=\; 0 \tag{A.5}$$

where $P_{\mathcal{H}_2}$ denotes an orthogonal projection on the $\mathcal{H}_2$ space.

A.2 Kronecker algebraic manipulations

If the rule '$\mathrm{vec}(ADB) = B^T \otimes A\,\mathrm{vec}(D)$' is applied to (2.13), we get:

$$P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}(V_k V_k^{\sim})^T \otimes (U_{k\lambda}^{\sim}U_{k\lambda})\,\mathrm{vec}(\hat{Q})\right) = P_{\mathcal{H}_2}\left(\sum_{k=1}^{s} V_k^{\sim T} \otimes U_{k\lambda}^{\sim}\,\mathrm{vec}(\hat{R}_{k\lambda})\right) \tag{A.6}$$

Using the so-called mixed-product rule, we get:

$$P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}(V_k^{\sim T}\otimes U_{k\lambda}^{\sim})(V_k^T\otimes U_{k\lambda})\,\mathrm{vec}(\hat{Q})\right)=P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}V_k^{\sim T}\otimes U_{k\lambda}^{\sim}\,\mathrm{vec}(\hat{R}_{k\lambda})\right)\tag{A.7}$$

Then, we define $W_{k\lambda}=V_k^T\otimes U_{k\lambda}$. This results in:

$$P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}W_{k\lambda}^{\sim}W_{k\lambda}\,\mathrm{vec}(\hat{Q})\right)=P_{\mathcal{H}_2}\left(\sum_{k=1}^{s}W_{k\lambda}^{\sim}\,\mathrm{vec}(\hat{R}_{k\lambda})\right)\tag{A.8}$$

This can be written more conveniently using matrix multiplications:

$$\begin{aligned}
P_{\mathcal{H}_2}\left((W_{1\lambda}^{\sim}W_{2\lambda}+\ldots+W_{s\lambda}^{\sim}W_{s\lambda})\,\mathrm{vec}(\hat{Q})\right)&=P_{\mathcal{H}_2}\left(W_{1\lambda}^{\sim}\,\mathrm{vec}(\hat{R}_{1\lambda})+\ldots+W_{s\lambda}^{\sim}\,\mathrm{vec}(\hat{R}_{s\lambda})\right)\quad\Leftrightarrow\\
P_{\mathcal{H}_2}(\begin{pmatrix}W_{1\lambda}\\ \vdots\\ W_{s\lambda}\end{pmatrix}^{\sim}\begin{pmatrix}W_{1\lambda}\\ \vdots\\ W_{s\lambda}\end{pmatrix}\mathrm{vec}(\hat{Q}))&=P_{\mathcal{H}_2}(\begin{pmatrix}W_{1\lambda}\\ \vdots\\ W_{s\lambda}\end{pmatrix}^{\sim}\mathrm{vec}\begin{pmatrix}R_{1\lambda}\\ \vdots\\ R_{s\lambda}\end{pmatrix})\quad\Leftrightarrow\\
P_{\mathcal{H}_2}\left(W^{\sim}W\,\mathrm{vec}(\hat{Q})\right)&=P_{\mathcal{H}_2}\left(W^{\sim}\,\mathrm{vec}(\hat{R})\right)
\end{aligned}\tag{A.9}$$

Appendix B

State-Space Formulae

B.1 Realisations for R_k, V_k, and U_k

According to [12], realisations of R_k, V_k, and U_k are as follows:

$$\begin{pmatrix} R_k & U_k \\ V_k & 0 \end{pmatrix} = \left[\begin{array}{cc|cc} A+B_uF & -B_uF & B_k & -B_u \\ 0 & A+HC_y & B_k+HD_{yk} & 0 \\ \hline C_k+D_{ku}F & -D_{ku}F & D_{kk} & -D_{ku} \\ 0 & C_y & D_{yk} & 0 \end{array}\right] \tag{B.1}$$

where A, B_k, B_u, C_k, C_y, D_{kk}, D_{ku}, D_{yk} are correctly partitioned matrices from the realisation of the plant P. Subscript k denotes the exogenous input and output (channel), and subscript u and y denote the controlled input and measured output, respectively. Matrices F and H are stabilising solutions to state-feedback and observer Ricatti equations, respectively.

B.2 Realisations for $\mathrm{vec}(R_{k\lambda})$ and $W_{k\lambda}$

Under similarity, the R_k part of (B.1) can be transformed:

$$T = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \Rightarrow R_k = \left[\begin{array}{c|c} TAT^{-1} & TB \\ \hline CT^{-1} & D \end{array}\right] = \left[\begin{array}{cc|c} A+B_uF & -HC_y & -HD_{yk} \\ 0 & A+HC_y & B_k+HD_{yk} \\ \hline C_k+D_{ku}F & C_k & D_{kk} \end{array}\right]$$

This system can be partitioned in the following factors:

$$R_k = R_\alpha R_\beta = \left[\begin{array}{c|cc} A+B_uF & 0 & -H \\ \hline C_k+D_{ku}F & I & 0 \end{array}\right] \left[\begin{array}{c|c} A+HC_y & B_k+H_Dyk \\ \hline C_k & D_{kk} \\ C_y & D_{yk} \end{array}\right]$$

If we then apply some Kronecker algebra (see [2]):

$$\begin{aligned} \mathrm{vec}(R_k) &= \mathrm{vec}(R_\alpha R_\beta) = \mathrm{vec}(I_{p1} R_\alpha R_\beta) = (R_\beta^T \otimes I_{p1})\,\mathrm{vec}(R_\alpha) \Leftrightarrow \\ &= \left[\begin{array}{c|cc} (A+HC_y)^T \otimes I_{p1} & C_k^T \otimes I_{p1} & C_y^T \otimes I_{p1} \\ \hline (B_k+HD_{yk})^T \otimes I_{p1} & D_{kk}^T \otimes I_{p1} & D_{yk} \otimes I_{p1} \end{array}\right] \left[\begin{array}{c|c} I_{p2} \otimes (A+B_uF) & \mathrm{vec}(-H) \\ \hline I_{p2} \otimes (C_k+D_{ku}F) & \mathrm{vec}(I_{p1}) \end{array}\right] \\ &= \left[\begin{array}{cc|c} (A+HC_y)^T \otimes I_{p1} & C_y^T \otimes (C_k+D_{ku}F) & \mathrm{vec}(C_1) \\ 0 & I_{p2} \otimes (A+B_uF) & \mathrm{vec}(-H) \\ \hline (B_k+HD_{yk})^T \otimes I_{p1} & D_{yk}^T \otimes (C_k+D_{ku}F) & \mathrm{vec}(D_{kk} \end{array}\right] \end{aligned} \tag{B.2}$$

The realisation of $W_k = V_k^T \otimes U_k$ is obtained using the mixed product rule from Kronecker algebra [2]:

$$\begin{aligned} W_k &= V_k^T \otimes U_k = (V_k^T I_{p2}) \otimes (I_{p1} U_k) = (V_k^T \otimes I_{p1})(I_{p2} \otimes U_k) \Leftrightarrow \\ &= \left[\begin{array}{c|c} (A+HC_y)^T \otimes I_{p1} & C_y^T \otimes I_{p1} \\ \hline (B_k+HD_{yk})^T \otimes I_{p1} & D_{yk}^T \otimes I_{p1} \end{array}\right] \left[\begin{array}{c|c} I_{p2} \otimes (A+B_uF) & -I_{p2} \otimes B_u \\ \hline I_{p2} \otimes (C_k+D_{ku}F) & -I_{p2} \otimes D_{ku} \end{array}\right] \Leftrightarrow \\ &= \left[\begin{array}{cc|c} (A+HC_y)^T \otimes I_{p1} & C_y^T \otimes (C_k+D_{ku}F) & -C_y^T \otimes D_{ku} \\ 0 & I_{p2} \otimes (A+B_uF) & -I_{p2} \otimes B_u \\ \hline (B_k+HD_{yk})^T \otimes I_{p1} & D_{yk}^T \otimes (C_k+D_{ku}F) & -D_{yk}^T \otimes D_{ku} \end{array}\right] \end{aligned} \tag{B.3}$$

If we now put the two realisations of $\mathrm{vec}(R_k)$ and W_k together, and apply (2.6), we get:

$$\begin{array}{l} \begin{pmatrix} \mathrm{vec}(R_{k\lambda}) & W_{k\lambda} \end{pmatrix} = \\ \left[\begin{array}{cc|cc} (A+HC_y)^T \otimes I_{p1} & C_y^T \otimes (C_k+D_{ku}F) & \mathrm{vec}(C_1) & -C_y^T \otimes D_{ku} \\ 0 & I_{p2} \otimes (A+B_uF) & \mathrm{vec}(-H) & -I_{p2} \otimes B_u \\ \hline \sqrt{\lambda_k}\left((B_k+HD_{yk})^T \otimes I_{p1}\right) & \sqrt{\lambda_k}\left(D_{yk}^T \otimes (C_k+D_{ku}F)\right) & \sqrt{\lambda_k}\,\mathrm{vec}(D_{kk}) & \sqrt{\lambda_k}\left(-D_{yk}^T \otimes D_{ku}\right) \end{array}\right] \end{array}$$

where I_{p1} and I_{p2} are identity matrices with the sizes according to the vector length of k and y, respectively.

Appendix C

State Space Realisations of Systems Under Study

System of chapter 3:

$$P = \left[\begin{array}{cc|ccc} -0.628 & -4.935 & 1 & 0 & 1 \\ 8 & 0 & 0 & 0 & 0 \\ \hline 0 & 1.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^{-4} \\ 0 & -1.25 & 0 & 5 \cdot 10^{-4} & 0 \end{array}\right] \tag{C.1}$$

System of chapter 4:

$$P = \left[\begin{array}{cccc|cccccc} -2.51 & -4.94 & 0 & 0 & 8 & 0 & 0 & 0 & 8 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.51 & -4.94 & 0 & 8 & 0 & 0 & 0 & 8 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 15.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} \\ 0 & -15.6 & 0 & 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -15.6 & 0 & 0 & 0 & 10^{-2} & 0 & 0 \end{array}\right] +$$

$$\left[\begin{array}{cc|cccccc} -6.28 \cdot 10^{3} & -1.54 \cdot 10^{5} & 8\sqrt{k} & 8\sqrt{1-k} & 0 & 0 & 8\sqrt{k} & 8\sqrt{1-k} \\ 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -6.19\sqrt{k} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6.19\sqrt{1-k} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} \\ 6.19\sqrt{k} & 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 \\ 6.19\sqrt{1-k} & 0 & 0 & 0 & 0 & 10^{-2} & 0 & 0 \end{array}\right]$$

System of chapter 5:

Coupled system:

$P = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, with:

$$A = \begin{pmatrix} -2.51 & -4.94 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.51 & -4.94 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.51 \cdot 10^2 & -7.71 \cdot 10^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.12 \cdot 10^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.77 \cdot 10^2 & -8.67 \cdot 10^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.02 \cdot 10^3 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 2.50 \cdot 10^{-1} & 0 & 0 & 02.50 \cdot 10^{-1} & 0 \\ 0 & 0 & 0 & 00 & 0 \\ 0 & 2.5 \cdot 10^{-1} & 0 & 00 & 2.5 \cdot 10^{-1} \\ 0 & 0 & 0 & 00 & 0 \\ 1.40 \cdot 10^{-2} & -2.80 \cdot 10^{-2} & 0 & 01.40 \cdot 10^{-2} & -2.80 \cdot 10^{-2} \\ 0 & 0 & 0 & 00 & 0 \\ 1.40 \cdot 10^{-2} & 6.99 \cdot 10^{-3} & 0 & 01.40 \cdot 10^{-2} & 6.99 \cdot 10^{-3} \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 3.01 \cdot 10^2 & 0 & 3.01 \cdot 10^2 & 0 & -16.6 & 0 & -11.88 \\ 0 & -3.01 \cdot 10^2 & 0 & 3.01 \cdot 10^2 & 0 & 33.3 & 0 & -5.94 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3.01 \cdot 10^2 & 0 & -3.01 \cdot 10^2 & 0 & 16.6 & 0 & 11.88 \\ 0 & 3.01 \cdot 10^2 & 0 & -3.01 \cdot 10^2 & 0 & -33.3 & 0 & 5.94 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-3} \\ 0 & 0 & 9.5 \cdot 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.5 \cdot 10^{-3} & 0 & 0 \end{pmatrix}$$

Decoupled system:

$$\tilde{P} = \left[\begin{array}{c|c} A & \tilde{B} \\ \hline \tilde{C} & D \end{array}\right]$$

$$\tilde{B} = \begin{pmatrix} 1.64 \cdot 10^{-1} & 1.89 \cdot 10 & 0 & 0 & 1.64 \cdot 10 & 1.89 \cdot 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.89 \cdot 10^{-1} & -1.64 \cdot 10 & 0 & 0 & 1.89 \cdot 10 & -1.64 \cdot 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1.20 \cdot 10^{-2} & 2.89 \cdot 10^{2} & 0 & 0 & -1.20 \cdot 10^{2} & 2.89 \cdot 10^{2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.44 \cdot 10^{-2} & 5.98 \cdot 10^{3} & 0 & 0 & 1.44 \cdot 10^{2} & 5.98 \cdot 10^{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\tilde{C} = \begin{pmatrix} 0 & 2.79 \cdot 10^{2} & 0 & 3.21 \cdot 10^{2} & 0 & -1.42 \cdot 10 & 0 & -1.23 \cdot 10 \\ 0 & 3.21 \cdot 10^{2} & 0 & -2.79 \cdot 10^{2} & 0 & -3.44 \cdot 10 & 0 & 5.08 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2.79 \cdot 10^{2} & 0 & -3.21 \cdot 10^{2} & 0 & 1.42 \cdot 10 & 0 & 1.23 \cdot 10 \\ 0 & -3.21 \cdot 10^{2} & 0 & 2.79 \cdot 10^{2} & 0 & 3.44 \cdot 10 & 0 & -5.083 \end{pmatrix}$$

A and D matrix are equal to those of system P.

Bibliography

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, chapter 4, pages 174–187. Cambridge University Press, 2004.

[2] John W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, CAS-25(9):772–781, 1978.

[3] Marco Dettori and Carsten W. Scherer. Mimo control design for a compact disc player with multiple norm specifications. *IEEE Transactions on Control Systems Technology*, 10(5):635–645, September 2002.

[4] Frank Stephen Tromp van Diggelen. *Hadamard Weighting in Robust Control*. PhD thesis, Wolfson College, Cambridge, 1992.

[5] James S. Freudenberg and Douglas P. Looze. Right half plane poles and zeros and design tradeoffs in feedback systems. *IEEE Transactions on Automatic Control*, AC-30(6):555–565, 1985.

[6] Guillermo I. Gómez and Graham C. Goodwin. Intergal constraints on sensitivity vectors for multivariable linear systems. *Automatica*, 32(4):499–518, 1996.

[7] Pramod P. Khargonekar and Mario A. Rotea. Multiple objective optimal control of linear systems: The quadratic norm case. *IEEE Transactions on Automatic Control*, 36(1):14–24, 1991.

[8] Leonard Lublin, Simon Grocott, and Michael Athans. *The Control Handbook*, chapter 40. CRC Press, 1996.

[9] Panos Y. Papalambros and Douglass J. Wilde. *Principles Of Optimal Design*, chapter 1, pages 15–16. Cambridge University Press, 2000.

[10] Phil Schniter. *Hilbert Spaces, http://cnx.org/content/m10434/2.11/.*

[11] Sigur Skogestad and Ian Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons, Ltd., 2005.

[12] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1996.