TUHH
Technische Universität Hamburg-Harburg

Arbeitsbereich Regelungstechnik

Project work

Andrey Popov

Less Conservative Mixed H_2/H_∞ Controller Design Using Multi-Objective Optimization

Supervisors:
Dr. A. Farag
Prof.Dr. H. Werner

July 2005

Report 2005.15

Declaration

Hereby I declare that I produced the present work myself only with the help of the indicated aids and sources.

Hamburg, 28th July 2005 Andrey Popov

This file was updated June 6, 2009.

Contents

1 Introduction

In the last years the H_2 and H_∞ controller design techniques have gained a lot of research attention. Both have strong theoretical basis and are efficient algorithms for synthesizing optimal controllers. Their combination, the mixed H_2/H_∞ allows combining intuitive quadratic performance specifications of the H_2 synthesis with robust stability requirements specifications expressed by the H_∞ synthesis.

Despite the popularity of the above methods among researchers and the many successfully solved control problems, the actual application of those approaches is quite limited. This is mainly due to two reasons: (i) The order of the synthesized controllers is the same as the order of the generalized plant, which in many cases is very high for practical applications; (ii) Often it is convenient or necessary to use a certain controller structure (e.g., decentralized control, PID control, etc.), which is impossible with the standard methods.

Additional hindrance exists with the mixed H_2/H_∞ controller design, where the straightforward combination of the H_2 with the H_∞ synthesis results in a conservative solution, i.e. the algorithm may fail to find a controller even if one exists, or it may be possible to find another controller, which achieves better values for the two norms.

In this project work a mixed H_2/H_∞ controller synthesis technique based on multi-objective optimization is used, where the optimized criteria are the H_2 and H_∞ norms. The method is compared with the existing methods for solving linear matrix inequalities (LMIs) and bilinear matrix inequalities (BMIs).

The multi-objective method is shown to have several advantages over the standard ones. The first one is, that it doesn't use a common Lyapunov matrix P in the LMI representation of the mixed H_2/H_∞ design problem. This makes it less conservative than the standard method and allows it to achieve smaller values of the H_2 and H_∞ norms both for the state feedback and output feedback cases. Next, because a genetic algorithm is used for the optimization, one can easily impose controller structure and controller order. It is also possible to impose constrains on the values of the controller coefficients. It will be shown that thanks to these advantages a better values of the norms even with lower order controller than the full-order controller are achievable.

The report is structured as follows. In Section 2 the LMI representations of the state-feedback and output-feedback design problems are shown. It is shown how the BMI problem formulation of the output-feedback case can be converted to LMI one by appropriate change of variables. Next, in Section 3 the genetic algorithms and more specifically multi-objective genetic algorithms are briefly presented. The algorithm used here - the Strengthen Pareto Evolutionary Algorithm (SPEA) is also reviewed. Section 4 presents a two mass-spring system control problem, known as the ACC benchmark problem, used in the rest of the work for all numerical implementations and simulations.

Section 5 presents the results for the state feedback design problem - the generalized plant with uncertainty in the parameters is modeled and the results obtained by solving the mixed H_2/H_∞ LMI problem are compared with the ones from the multi-objective

optimization. A comparison is done also with a less-conservative iterative approach, proposed in [13].

The output feedback problem is considered in Section 6 - the model of the generalized plant with uncertainty in all coefficients is derived and the solutions from the existing LMI representation and compared with the ones from the multi-objective approach. Designs are done for different controller order and the result are compared with the full-order result from the standard LMI approach. A low order controller is selected and attempt is made to satisfy the design requirements for the benchmark problem. As a result a controller achieving better scores than some previously designed high order controllers is obtained.

In Section 7 the results are summarized and conclusions are drawn. Based on those conclusions prepositions for further research are made.

2 Mixed H_2/H_∞ Controller Design via LMIs

A linear matrix inequality (LMI) [1, 2, 3] can be expressed in the canonical form (2.1), where L_0 to L_N are given symmetric matrixes and $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$ is a vector of with the scalar variables to be determined. It is important to note, that the set of all solutions x is convex.

$$L(x) = L_0 + x_1 L_1 + \cdots + x_N L_N < 0 \tag{2.1}$$

Many control engineering problems can be expressed in this form, but numerical problem solving via LMI gained a lot of attention only in the last 20 years, since interior-point methods for their solving have been developed [1].

LMIs are well suited for solving multi-objective problems, because it is easy to combine them to a single LMI. For example $L_1(x) < 0$ and $L_2(x) < 0$ can be combined to:

$$\begin{bmatrix} L_1(x) & 0 \\ 0 & L_2(x) \end{bmatrix} < 0 \tag{2.2}$$

LMIs can also be further used as constraints to minimization problems

$$\min_x c^T x \quad \text{subject to} \quad L(x) < 0 \tag{2.3}$$

where the elements of vector c are weights on the decision variables.

Before describing how the H_2 and H_∞ design requirements can be represented as LMIs, a short overview of the generalized plant concept and the norms will be made.

2.1 Generalized Plant

In the modern control techniques most controller design problems are represented in the form, shown in Figure 2.1, where:

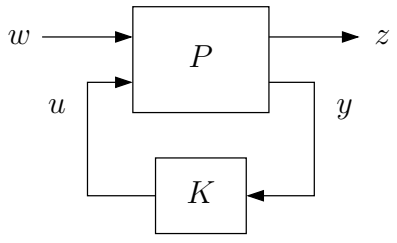

Figure 2.1: Closed-loop with generalized plant

P - generalized plant

K - controller

w - external inputs, such as reference signal, disturbances, noise, etc

u - control signal

z - fictitious output, used to represent design specifications

y - measured signals, used for feedback.

Without loss of generality, we can assume for the moment, that the reference signal is zero $r = 0$. Expanding the w and z channels to ones connected to the performance (w_2 and z_2) and the ones connected to the robustness (or H_∞ performance) (w_∞ and z_∞) we get to the generalized plant representation, shown in Figure 2.2.

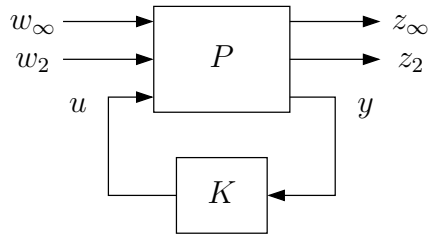

Figure 2.2: Closed-loop with generalized plant containing performance and robustness channels

The state-space representation of the system is

$$\left|\begin{array}{rcrcrcrcr} \dot{x} & = & Ax & + & B_\infty w_\infty & + & B_2 w_2 & + & B_u u \\ z_\infty & = & C_\infty x & + & D_{\infty\infty} w_\infty & + & D_{\infty 2} w_2 & + & D_{\infty u} u \\ z_2 & = & C_2 x & + & D_{2\infty} w_\infty & + & D_{22} w_2 & + & D_{2u} u \\ y & = & C_y x & + & D_{y\infty} w_\infty & + & D_{y2} w_2 & + & D_{yu} u \end{array}\right. \tag{2.4}$$

This can be written in a compact form as

$$P(s) = \left[\begin{array}{c|c} A_p & B_p \\ \hline C_p & D_p \end{array}\right] = \left[\begin{array}{c|ccc} A & B_\infty & B_2 & B_u \\ \hline C_\infty & D_{\infty\infty} & D_{\infty 2} & D_{\infty u} \\ C_2 & D_{2\infty} & D_{22} & D_{2u} \\ C_y & D_{y\infty} & D_{y2} & D_{yu} \end{array}\right] \tag{2.5}$$

Since all real systems are strictly proper $D_{yu} = 0$ can be assumed without loss of generality. To be able to compute a finite value of the H_2 norm $D_{22} = 0$ should be fulfilled. On the other hand the H_2 and H_∞ performance indexes do not depend on one another in the general case, so $D_{\infty 2} = 0$ and $D_{2\infty} = 0$.

Assuming that d and ν are correspondingly process noise and measurement noise, with $E[dd^T] = Q_e$, $E[\nu\nu^T] = R_e$, $E[d\nu^T] = 0$, the H_2 norm can be used to represent the LQG performance index

$$V_{LQG} = \lim_{T\to\infty} \mathrm{E}\left[\frac{1}{T}\int_0^T (x^T Q x + u^T R u)\, dt\right] \tag{2.6}$$

To do so, the following definitions are made

$$B_2 = \begin{bmatrix} Q_e^{1/2} & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix}, \quad D_{2u} = \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix}, \quad D_{y2} = \begin{bmatrix} 0 & R_e^{1/2} \end{bmatrix} \tag{2.7}$$

where:

R penalty matrix for the control signal

Q penalty matrix for the system states

0 denotes matrixes with appropriate dimensions, so that:

The number of w_2 inputs is equal to the number of system states plus the number of system outputs

$$D_{y2}\begin{bmatrix} B_2 \\ D_{y2} \end{bmatrix} = \begin{bmatrix} 0 & R_e \end{bmatrix}$$

The number of z_2 outputs is equal to the number of system states plus the number of system inputs

$$D_{2u}^T \begin{bmatrix} C_2 & D_{2u} \end{bmatrix} = \begin{bmatrix} 0 & R \end{bmatrix}$$

The four matrixes Q, Q_e, R and R_e are used later as tuning parameters to meet the design requirements.

Here the H_2 norm of the system is used as a measure of its performance, whereas the H_∞ norm as a measure of the system robustness. For completeness the definitions of the two norms are given below.

H_2 norm

Given the system

$$\left| \begin{array}{l} \dot{x} = Ax + Bw \\ z = Cx \end{array} \right.$$

The H_2 norm of the system is defined as

$$\left\|T\right\|_2^2 = \text{ trace } CP_0C^T = \frac{1}{2\pi}\int_{-\infty}^{\infty} \left\|T(j\omega)\right\|_F^2 d\omega \tag{2.8}$$

where P_0 satisfies: $P_0 > 0$ and $AP_0 + P_0A^T + BB^T = 0$, and $\left\|T\right\|_F = \sqrt{\text{trace } (T^HT)}$ denotes the Frebenius norm of T.

H_∞ norm

Given the system

$$\left| \begin{array}{l} \dot{x} = Ax + Bw \\ z = Cx + Dw \end{array} \right.$$

The H_∞ norm of the system is defined as the maximal singular value over all frequencies

$$\left\|T(s)\right\|_\infty = \sup_\omega \bar{\sigma}(T(j\omega)) = \max_{\omega \neq 0} \frac{\int_0^\infty z^T(t)z(t)dt}{\int_0^\infty \omega^T(t)\omega(t)dt} \tag{2.9}$$

2.2 Controller Design Using LMIs

The full derivation of the LMIs presented here can be found in [3] and [4]. Here only the final LMI terms are presented. By $\star$ are represented symmetric terms in the LMIs.

State Feedback Controller

Assume we want to design controller of the form

$$u = Ky \tag{2.10}$$

where K is a matrix with appropriate dimension and $C_y = I$, thus we can measure all the system states.

*H_2 **performance***

Remembering, that $D_{22} = 0$ and $D_{y2} = 0$ the closed-loop system will be

$$\left| \begin{array}{ll} \dot{x} & = (A + B_u K)x + B_2 w_2 = A_{cl}x + B_{cl}w_2 \\ z_2 & = (C_2 + D_{2u}K)x = C_{cl}x \end{array} \right. \tag{2.11}$$

The H_2 norm of the closed-loop is finite and smaller than ν if a Lyapunov matrix $P = P^T > 0$ exists that satisfies

$$\begin{bmatrix} A_{cl}^T P + P A_{cl} & P B_{cl} \\ \star & -I \end{bmatrix} < 0$$
$$\begin{bmatrix} P & C_{cl} \\ \star & W \end{bmatrix} > 0 \tag{2.12}$$
$$\text{trace } W < \nu^2$$

for some symmetric matrix W of an appropriate dimension.

As one can easily see, some of the matrix terms involve multiplication of design variables (A_{cl} and C_{cl} depend on the state feedback matrix K) and thus make the inequality bilinear (Bilinear Matrix Inequality - BMI). To make it again linear a new variable is introduced: $Y = KP$. Than the H_2 performance requirement can be formulated as:

The H_2 norm of the closed-loop system is less then ν, if and only if matrixes $P = P^T > 0$, W and $Y = KP$ exist, that satisfy

$$\begin{bmatrix} AP + PA^T + B_u Y + Y^T B_u^T & B_2 \\ \star & -I \end{bmatrix} < 0$$
$$\begin{bmatrix} W & C_2 P + D_{2u} Y \\ \star & P \end{bmatrix} > 0 \tag{2.13}$$
$$\text{trace } W < \nu^2$$

In this sense, minimizing the trace of W and using the two LMIs as constraints, can be used to find the minimal H_2 norm of the closed-loop system. Than the feedback matrix can be calculated as

$$K = Y P^{-1} \tag{2.14}$$

H_∞ performance

The closed-loop system with the H_∞ channels and $D_{y\infty} = 0$ yields the closed-loop system

$$\left| \begin{array}{rl} \dot{x} & = (A + B_u K)x + B_\infty w_\infty \\ z_\infty & = (C_\infty + D_{\infty u} K)x + D_{\infty\infty} w_\infty \end{array} \right. \tag{2.15}$$

. Directly writing the H_∞ performance, leads again to a BMI. Using again the variable $Y = KP$ it can be reformulated as:

The H_∞ norm to the closed-loop system is smaller than γ, if matrixes $P = P^T > 0$ and $Y = KP$ exist, that satisfy:

$$\begin{bmatrix} AP + PA^T + B_u Y + Y^T B_u^T & \star & \star \\ B_\infty^T & -\gamma I & \star \\ C_\infty P + D_{\infty u} Y & D_{\infty\infty} & -\gamma I \end{bmatrix} < 0 \tag{2.16}$$

Here the weight vector c can be set to represent only the value of γ, so by solving the constrained problem $\min_x c^T x$ directly the H_∞ norm can be minimized. The controller is calculated back, as in the H_2 case via $K = YP^{-1}$.

Mixed H_2/H_∞ design

The combination of the H_2 and H_∞ synthesis is done by combining (2.13), (2.16) and $P > 0$ as shown in the way, shown at (2.2). However a problem arises. There are, let us call them P_2 and Y_2, matrixes from the H_2 LMIs and P_∞ and Y_∞ matrixes from the H_∞ LMI, which in the general case are different. This means, that at the end we will have two controllers, minimizing correspondingly the H_2 and H_∞ norm, but we can use only one with the system. To get an unique solution, the same set of matrixes needs to be used: $P = P_2 = P_\infty$ and $Y = Y_2 = Y_\infty$. This is known as the *Lyapunov shaping paradigm.* The drawback is that this method is conservative and does not achieve the best possible solution of the problem, or sometimes cannot even find a feasible (stabilizing) solution. The advantage, on the other hand is, that the problem remains convex and easy to solve with the LMI solvers.

Having combined the LMIs one can set γ to some desired value and solve the LMIs to find minimal trace(W), corresponding to the value of the H_2^2 norm. If the results are unsatisfactory γ is set to new value and the synthesis is repeated.

Output Feedback Controller

For the case of output-feedback, a dynamic controller is assumed

$$\left|\begin{array}{ll} \dot{\xi} & = A_k\xi + B_k y \\ u & = C_k\xi + D_k y \end{array}\right. \tag{2.17}$$

By denoting

$$B_j = B_w R_j, \quad C_j = L_j C_z, \quad D_j = L_j D_{zw} R_j, \quad E_j = L_j D_{zu}, \quad F_j = D_{yw} R_j \tag{2.18}$$

where

> w and z represent either w_2 and z_2 for the H_2 case or w_∞ and z_∞ for the H_∞ case. The same notation is used for the B_w, D_{zu} and D_{yw} matrixes.
>
> R_j and L_j can be used to select appropriate input/output combination, and thus represent the closed-loop system as: $T_j(s) = L_j T(s) R_j$. Here, however they are both set to unity.

The closed-loop system is

$$\left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array}\right] = \left[\begin{array}{cc|c} A+B_uD_kC_y & B_uC_k & B_j+B_uD_kF_j \\ B_kC_y & A_k & B_kF_j \\ \hline C_j+E_jD_kC_y & E_jC_k & D_j+E_jD_kF_j \end{array}\right] \tag{2.19}$$

As in the state-feedback case, the direct writing of the matrix inequalities for H_2 and H_∞ leads to terms involving products of design variables and thus BMIs. Whereas for the state-feedback, this problem is solved by the simple change of variables $Y = KP$, in the case of output feedback a more complicated change of variables has to be done [3]. The new design variables are: Y, X, $\hat{A}_k$, $\hat{B}_k$, $\hat{C}_k$ and $\hat{D}_k$, which fulfill

$$\begin{bmatrix} Y & I \\ I & X \end{bmatrix} > 0 \tag{2.20}$$

$$\begin{array}{ll} \hat{A}_k & = NA_kM^T + NB_kC_yY + XB_uC_kM^T + X(A + B_uD_kC_y)Y \\ \hat{B}_k & = NB_k + XB_uD_k \\ \hat{C}_k & = C_kM^T + D_kC_yY \\ \hat{D}_k & = D_k \end{array} \tag{2.21}$$

where M and N are not a design parameters, but are instead calculated after the solution of the LMI is obtained, such that fulfill

$$MN^T = I - YX \tag{2.22}$$

By (2.20) it is inferred that $X > 0$, $Y - X^{-1} > 0$, such that if $I - YX$ is nonsingular, so it is always possible to find M and N satisfying (2.22), via singular value decomposition.

The controller matrixes, can be computed back from (2.21) as

$$\begin{aligned} D_k &= \hat{D}_k \\ C_k &= (\hat{C}_k - D_k C_y Y) M^{-T} \\ B_k &= N^{-1}(\hat{B}_k - X B_u D_k) \\ A_k &= N^{-1}(\hat{A}_k - N B_k C_y Y - X B_u C_k M^T - X(A + B_u D_k C_y) Y) M^{-T} \end{aligned} \tag{2.23}$$

Because those calculations include the inverse of M and N, an ill-conditioned $I - YX$, will lead to ill-conditioned inversion. Because $I - YX$ is nearly singular if the constraint (2.20) is saturated at the optimum one needs to deal with this problem [3].

H_2 performance

The LMIs for the H_2 controller synthesis problem are

$$\begin{bmatrix} AY + YA^T + B_u\hat{C}_k + (B_u\hat{C}_k)^T & \star & \star \\ \hat{A}_k + (A + B_u\hat{D}_k C_y)^T & A^TX + XA + \hat{B}_k C_y + (\hat{B}_k C_y)^T & \star \\ C_j Y + E_j \hat{C}_k & C_j + E_j \hat{D}_k C_y & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} Y & I & B_j + B_u\hat{D}_k F_j \\ \star & X & XB_j + \hat{B}_k F_j \\ \star & \star & W \end{bmatrix} > 0 \tag{2.24}$$

$$\text{trace}\,(W) < \nu$$

and inequality (2.20), with the additional requirement $D_j + E_j\hat{D}_k F_j = 0$. To minimize the H_2 norm, one needs to minimize trace(W).

H_∞ performance

For the H_∞ synthesis, the needed LMIs are (2.20) and

$$\begin{bmatrix} AY + YA^T + B_u\hat{C}_k + (B_u\hat{C}_k)^T & \star & \star & \star \\ \hat{A}_k + (A + B_u\hat{D}_k C_y)^T & A^TX + XA + \hat{B}_k C_y + (\hat{B}_k C_y)^T & \star & \star \\ (B_j + B_u\hat{D}_k F_j)^T & (XB_j + \hat{B}_k F_j)^T & -\gamma I & \star \\ C_j Y + E_j\hat{C}_k & C_j + E_j\hat{D}_k C & D_j + E_j\hat{D}_k F_j & -\gamma I \end{bmatrix} < 0 \tag{2.25}$$

The minimization of the H_∞ norm is equivalent, to the minimization of γ.

Mixed H_2/H_∞ design

As in the case for state-feedback, the combination of the H_2 and H_∞ synthesis is done by combining (2.20), (2.24) and (2.25) to a single LMI. Conservatism is added to the design again by the Lyapunov shaping paradigm. To be able to keep the problem convex and obtain an unique solution the postulation $X = X_2 = X_\infty$ and $Y = Y_2 = Y_\infty$ is made, where X_2 and Y_2 are from the H_2 performance LMIs and X_∞ and Y_∞ from the H_∞ design requirements. A solution can be found again by setting γ to a desired, achievable value and solving a trace(W) minimization problem, using the LMI as a constraint.

3 Multi-Objective Genetic Algorithms

Genetic Algorithms (GA) are direct, parallel, stochastic search method for optimization[1], which imitates the evolution of the living beings, described by Charles Darwin. GA are particularly good in solving

- global optimization problems
- multi-variable problems
- multi-objective problems
- non-convex optimization problems.

The parallel character of the search comes from the fact, that GA are dealing with population (set) of solutions at each iteration. In terms of GA language those solutions are called individuals and the iterations are called populations. The fact that GA deals not with a single solution, but with a set of such, allows them to explore many points in the variable space. To help finding the global extremum a selection process and set of genetic operators are applied to the population.

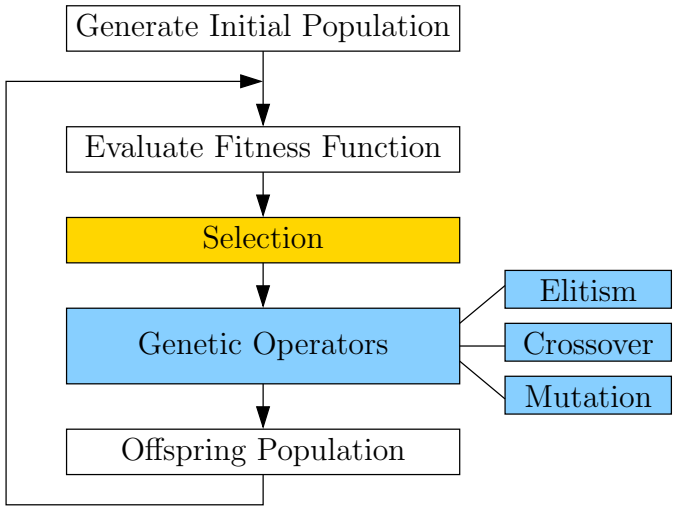

Figure 3.1: Block diagram of GA algorithm

The general structure of GA is shown at Figure 3.1. The algorithm starts by generating initial population, which is in most cases a random one. Then the cost function is calculated. Based on this cost function, a fitness function is assigned to each individual, corresponding to how good the individual 'fits' to the requirements. This means, that the individual with the smallest cost function (in case of minimization) gets the highest fitness value; the second best the second highest, etc.

[1]Here only minimization problems are considered

Next, the individuals are applied to the Selection process, which aims to chose the 'best' ones, which will continue and be involved in the production of offspring population, as well as to keep the diversity in the population and avoid convergence to local minima. The selected individuals are applied further to genetic operations, the three most common from which are

- Elitism - the best individuals are directly copied into the next generation, so the best achieved result is preserved
- Crossover - the individuals recombine with one another. The idea is to exchange the variables of the individuals in such a way, that only the good features are used, so the offspring individuals have better fitness function than the parent ones;
- Mutation - in general random change of the value of some of the optimized parameters of one or several individuals. The individuals are also randomly chosen, within the population.

The result of the genetic operators is a next generation of individuals. Their cost function values are evaluated and the process repeated until some stopping criteria is met.

3.1 Multi-Objective Optimization

Quite often, in the engineering tasks, there are several criterion that need to be simultaneously satisfied. Often those criterion are contradicting and cannot have optimum at the same time, thus improving the value of one-criteria means getting worst values for another. In these cases the optimization goal can be represented as vector of objective functions

$$F(x) = \begin{bmatrix} f_1(x) & f_2(x) & \dots & f_n(x) \end{bmatrix}^T \tag{3.1}$$

One widely known method of solving multi-objective optimization problems is by reducing the problem to a single-objective one by the use of weights. In this case every cost function is scaled by a weight and they are summed all together to form a single-objective function

$$f(x) = \sum_{i=1}^{n} w_i f_i(x) = WF(x)$$

The later one can be solved with a standard optimization algorithms.

However a new problem arises - How to chose the weights? Setting large value to one of the weights will make the optimization algorithm minimize mainly that objective and allow *bad* values for the others. The iterative try-and-error approach of setting the weights can finally lead to a good solution, but it is difficult to get an overview about the trade-off between the objective functions.

Genetic Algorithms on the other hand work with population of individuals, which appears to be favorable for multi-objective optimization, because in the same population there could exist individuals fitted best to different objective functions. This can be used to solve

the problem directly as multi-objective and get a set of possible solutions (individuals) at the end. It should be noted that with multi-objective optimization, one makes the choice of solution a-posteriori (after the optimization run) and not a-priori (before the optimization run). In other words the user doesn't have to care about defining weights, but instead has to chose a solution from the obtained set.

The genetic operators in the multi-objective genetic algorithms (MOGA) are the same as in the single-objective algorithms, but another type of selection is needed. The most widely used class of the selection algorithms is based on the proposed by Vilfredo Pareto set of non-dominated solutions and Pareto-optimality.

Definition 3.1 (Domination) *Vector v is said to dominate vector u ($v \succ u$), if and only if v is partially less then u:*

$$\forall i \in \{1, 2, \ldots, n\} : v_i \leq u_i \text{ and } \exists j \in \{1, 2, \ldots, n\} : v_j < u_j \tag{3.2}$$

In other words the values of vector v should not be bigger than the corresponding ones from vector u and should be smaller for at least one of them.

Definition 3.2 (Pareto optimality) *A solution $x \in S$ is a said to be Pareto-optimal solution, if and only if there is no $y \in S$, for which $F(y) = \big(f_1(y), \ldots, f_n(y)\big)$ dominates $F(x) = \big(f_1(x), \ldots, f_n(x)\big)$.*

where S is the permitted parameter space.

The above two concepts are illustrated in Figure 3.2, where

Y - solution space

P - Pareto surface - consists only from Pareto-optimal solutions

$f_1(x)$, $f_2(x)$ - objective functions

f_1^*, f_2^* - optimal solutions for the corresponding objective function

F^* - imaginary optimal solution (does not exist).

Overview of the variety of multi-objective algorithms can be found in [5, 6, 7]. The underlying idea of the methods using Pareto-optimality is the same: for the current generation find the non-dominated individuals and assign them with rank 0 (it is important to note, that the fitness is to be minimized here). From the remaining individuals find again the non-dominated ones (those that form the second Pareto-surface) and assign them with rank = 1. Continue until all individuals have rank assigned. Then the fitness values are calculated, so the individuals with smaller rank gets the biggest fitness function (i.e. the highest survival probability). Than roulette wheel, uniform or other selection method can be applied, followed by the genetic operators.

At the end of the optimization run, a Pareto-surface of solutions is obtained. Based on this information the user has to chose the result that suits her/him the best. Furthermore

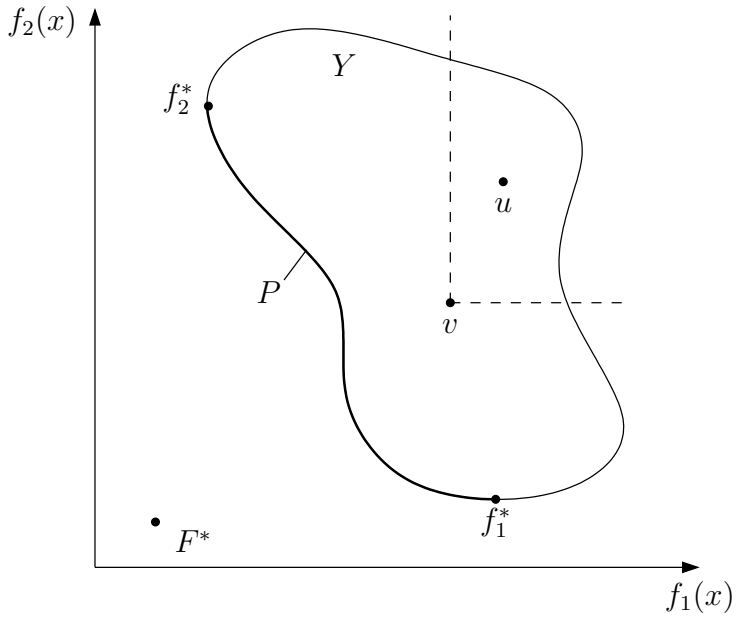

Figure 3.2: Pareto-surface

this information can give deeper insight of the tradeoff between the objective functions - from the Pareto-surface one can easily see how much the improvement of one objective function will influence on the others.

There are modifications and improvements of the basic concept, which modify the rank of the individuals depending on how many individuals are dominated by the current one or what is the population density around the individual. One of them is the applied here - SPEA 2.

3.2 Strengthen Pareto Evolutionary Algorithm 2

The multi-objective algorithm used in this work is Strengthen Pareto Evolutionary Algorithm 2 (SPEA2), selected due to its outstanding performance, shown in [8, 9]. SPEA2 is a second version of SPEA method, an in contrast to its predecessor incorporates fine-grained fitness assignment strategy and a density estimation.

The method uses Pareto-dominant based selection and elitism. This means that the best N individuals are copied into an archive A_t, which together with the current population is used for generating the offspring population. The algorithm is as follows

1. Initialization: Generate an initial population P_0 and create the empty archive (external set) $A_t = \emptyset$. Set the population counter $t = 0$;

2. Fitness assignment: Calculate fitness values of individuals in P_t and A_t;

3. Environmental selection: Copy all non-dominated individuals from P_t and A_t to A_{t+1}. If size of A_{t+1} exceeds N then reduce A_{t+1} by means of the truncation op-

erator[2], otherwise if the size of A_{t+1} is less than N then fill A_{t+1} with dominated individuals from A_t and P_t.

4. Termination: If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in A_{t+1} and Stop.

5. Mating selection: Perform binary tournament selection with replacement on A_{t+1} in order to fill the mating pool.

6. Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to step 2.

The important concept in SPEA2 is the fitness assignment. To avoid the situation that individuals dominated by the same archive members have identical fitness values, for each individual both dominating and dominated solutions are taken into account. First a strength value $s(i)$ is assigned to each individual, giving the number of individuals dominated by it:

$$s(i) = \{j | j \in (P_t \cup A_t) \wedge i \succ j\} \tag{3.3}$$

Based on this the raw fitness $R(i)$ of the individual i is calculated as a sum of the strength values of the individuals dominating it:

$$R(i) = \sum_{j \in (P_t \cup A_t) \wedge j \succ i} s(j) \tag{3.4}$$

More detailed information about SPEA2 can be obtained from [9, 7].

The available on-line code of SPEA2 is in C and follows the structure of PISA [10]. In order to conveniently use the optimization algorithm with MATLAB, here are used the MEX functions derived from the C code by the Institute of Control Systems, Hamburg University of Technology (http://www.tu-harburg.de/rts).

[2]Estimate the population density for each point, and remove those with highest one.

4 Robust Control Design Benchmark Problem

The Robust Control Design benchmark problem is presented in [11] and is also famous as the "ACC Benchmark Problem". Here it will be referred to also as the *cart system* control problem.

4.1 System description

The generic model of an uncertain two-mass-spring dynamical system is concerned - Figure 4.1.

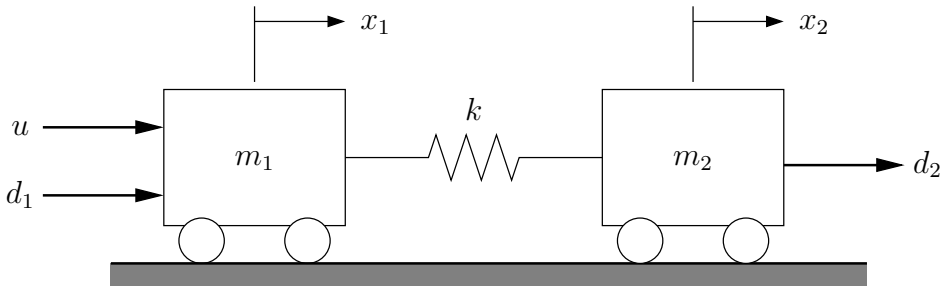

Figure 4.1: Two-mass-spring system with uncertain parameters

It is assumed that the nominal system is with cart masses $m_1 = m_2 = 1$ and spring constant $k = 1$, where appropriate units are used and the time is in seconds.

The state-space representation of the system is

$$\left|\begin{array}{rcl} \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} &=& \begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -k/m_1 & k/m_1 & 0 & 0\\ k/m_2 & k/m_2 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1/m_1\\ 0\end{bmatrix}(u+d_1) + \begin{bmatrix}0\\ 0\\ 0\\ 1/m_2\end{bmatrix} d_2 \\ y &=& \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} \end{array}\right.$$

where

x_1, x_2 - the positions of body 1 and 2 respectively

x_3, x_4 - correspondingly the velocities of body 1 and 2

u - the control force applied to body 1

d_1, d_2 - the disturbances acting on body 1 and 2 correspondingly

y the measured position of the second body, which has to be controlled.

The following compact notation will be used as well

$$\left|\begin{array}{ll} \dot{x} & = Ax + B_u u + B_d d \\ y & = C_y x + D_u u + D_d d \end{array}\right.$$

where

A, B_u and C_y are the corresponding matrixes from (4.1)

d is vector containing both disturbances $d = \begin{bmatrix} d_1 & d_2 \end{bmatrix}^T$

B_d is defined as

$$B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_1 & 0 \\ 0 & 1/m_2 \end{bmatrix}$$

$D_u = 0$ has a dimension 1×1

$D_d = 0$ has a dimension 1×2.

4.2 Design Requirements

The entire set of 4 benchmark problems can be found in [11]. Here the design requirements from [12] are used for the output-feedback case. For the state-feedback case a combination between existing requirements and new is made, because the original problem formulation does not concern the state-feedback case.

Problem A

Design a constant-gain linear feedback controller of the form

$$u = Ky \tag{4.1}$$

assuming, that $C_y = I$, i.e. all system states are measured, satisfying the following requirements.

1. For a unit impulse disturbance exerted on body 1 or body 2, the controlled output ($z = x_2$) has a settling time of about 15 sec for the nominal system with $m_1 = m_2 = k = 1$.
2. The closed-loop system is stable for $0.5 \leq k \leq 2.0$ and $m_1 = m_2 = 1$.
3. Reasonable performance/stability robustness are achieved.
4. Reasonable control effort (e.g. peak control input) is used.

Problem B

The problem formulation for the output-feedback case is taken from [12]:

1. For a unit impulse disturbance exerted on body 1 and/or body 2, the controlled output of the nominal system shall not exceed 0.1 after 15 sec.
2. For the same disturbance the peak control level of the nominal system should not exceed 1.
3. The gain margin shall be 6 dB or greater and the phase margin shall be at least 30 deg.
4. The closed-loop system shall be stable for $0.5 \leq k \leq 2.0$ and $m_1 = m_2 = 1$.
5. The closed-loop system shall be stable for simultaneous change $0.7 \leq k, m_1, m_2 \leq 1.3$.
6. There shall be reasonable high-frequency sensor noise rejection, performance robustness and controller complexity.

5 State Feedback Controller Design

5.1 Modeling the Uncertainty

In the case of state feedback $m_1 = m_2 = 1$ and $0.5 \leq k \leq 2.0$. The variable k can be represented as $k = k_0 + \delta k$, where $k_0 = 1.25$ and $-0.75 \leq \delta k \leq 0.75$. Substituting m_1 and m_2 with their values, the A system matrix can be written as:

$$A = A_0 + \delta A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_0 & k_0 & 0 & 0 \\ k_0 & k_0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\delta k & \delta k & 0 & 0 \\ \delta k & -\delta k & 0 & 0 \end{bmatrix}$$

Further δA can be represented as:

$$\delta A = \delta k \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \delta k \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} = B_w \delta k C_z = B_w \Delta C_z$$

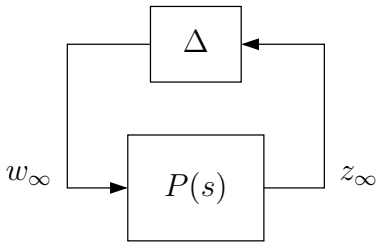

Figure 5.1: Uncertainty representation as upper LFT

The uncertainty can be represented in upper linear fraction transformation (LFT) form, as shown in Figure 5.1, where $-1 \leq \Delta \leq 1$. From $w_\infty = \Delta z_\infty$ and the autonomous system

$$\left| \begin{array}{rl} \dot{x} & = A_0 x + B_\infty w_\infty \\ z_\infty & = C_\infty x \end{array} \right.$$

can be represented as

$$\dot{x} = (A_0 + B_\infty \Delta C_\infty) x$$

where

$$B_\infty = \begin{bmatrix} 0 \\ 0 \\ -0.75 \\ 0.75 \end{bmatrix} \text{ and } C_\infty = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}$$

Another way to obtain the B_∞ and C_∞ matrixes is to use singular value decomposition (SVD), as shown in [4], but since here it is easy to make the factorization it is not used.

From the small-gain theorem it follows, that the closed-loop system, will be robustly stable upon any parameter variation, if its $H_\infty < 1$.

5.2 LMI Solutions

For 50 different values of γ the combined LMI of (2.13), (2.16) and $P > 0$ is solved and a Pareto-set, between the H_2 and H_∞ norm is obtained (Figure 5.2). This set is later compared with the one from the multi-objective optimization.

The H_2 connected matrixes are calculated according (2.7), where

$$R = B_u^T B_u = 1; \quad Q = C_y^T C_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad R_e = 0; \quad Q_e = 0;$$

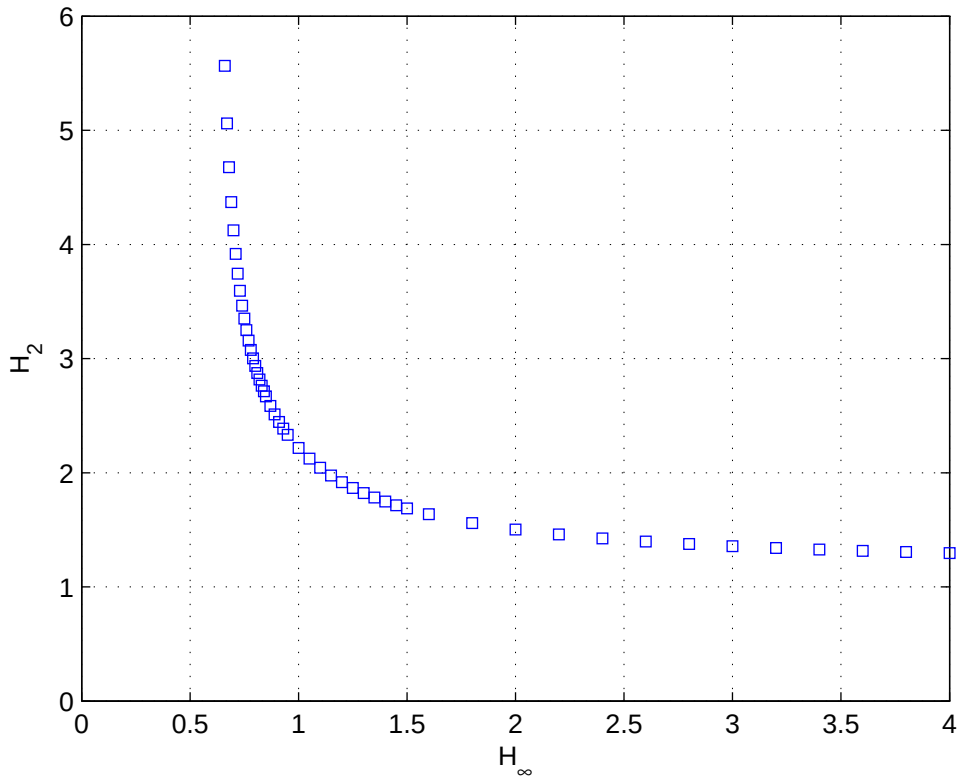

Figure 5.2: Results for solving the mixed H_2/H_∞ LMIs for 50 different γ

5.3 MOGA solution

The optimization parameters of MOGA are the coefficients of the feedback matrix K. Since K is 1×4 matrix, there are 4 coefficients to be found.

To be able to explore bigger part of the search space, the design variables are represented in fraction-exponential coding: $p = a10^b$. The advantages come from the fact, that a

small change of b will affect p a lot and will help exploring both very small and very large values of p, while a is used for fine tuning of the value.

By encoding all variables as fragment-exponential, the total number of decision variables becomes 8. The following boundaries are used for the variables: $-1 \leq a_i \leq 1$ and $-6 \leq b_i \leq 1$, where $i = 1..4$.

From separate optimization runs it was found that population size of 50 and 100 generations are sufficient to give good reproducibility of the results and Pareto-set, close to the optimal one (obtained with population size of 200 and 500 generations) Figure 5.3.

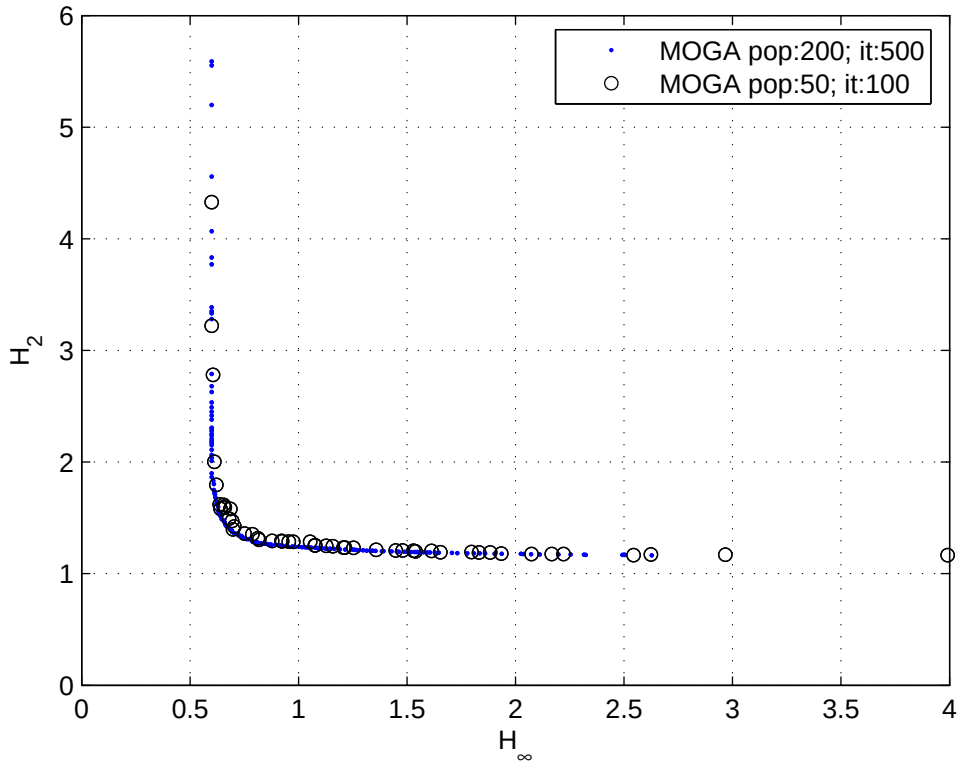

Figure 5.3: Comparing the result from a MOGA run with the *best achievable* Pareto-set

5.4 Comparison of the LMI and MOGA Results

The results from LMI and MOGA are plotted together in Figure 5.4.

From the figure it is obvious that MOGA can achieve better values for the two norms. This, as already explained, is die to the usage of common Lyapunov matrixes in the LMI representation, which bring conservatism. One can see, that the biggest improvement with MOGA, compared to the LMI case, is in the area close to the origin. As the two trade-off sets go to higher values of H_2 or H_∞ norms their distance gets smaller, because

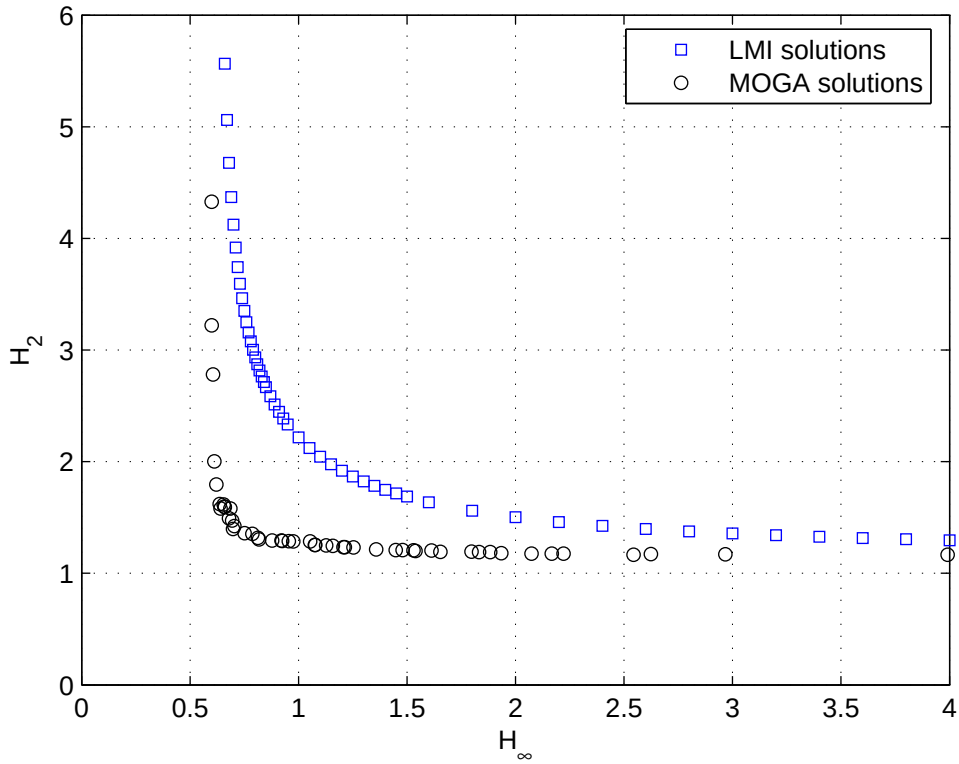

Figure 5.4: Comparing the result from a LMI and MOGA

the LMI representation emphasizes more on H_2 or H_∞ norm and the constrained brought by the common P has not such a big influence on the solution.

For $\gamma = 0.7$ a state-feedback calculated with MOGA is compared to the one computed by the standard LMI method. The coefficients of the feedback matrix are given below and the closed-loop system responses in Figure 5.5 for the LMI and in Figure 5.6 for MOGA. The upper plots are the output of the system (system states) and the lower are the control signals applied. The figures to the left are for the case of impulse disturbance on the d_1 channel and the ones on the right side - for the d_2 channel.

$$K_{LMI} = \begin{bmatrix} -6.2160 & 5.2160 & -3.5259 & -3.3237 \end{bmatrix}$$

$$K_{MOGA} = \begin{bmatrix} -2.5631 & 1.2138 & -1.9607 & -0.6582 \end{bmatrix}$$

From the impulse responses one can see that the feedback, calculated by MOGA achieves settling time under 10 seconds, whereas for the one calculated with the LMIs it is about 15 seconds. Furthermore the peak control value with the standard approach reaches 3.5 units, whereas 1.9 is the maximal for the genetic approach. Choosing a state-feedback designed with MOGA for a higher value for the $H\infty$ norm can reduce further this value. In conclusion it can be said, that the selected K_{MOGA} satisfies the design requirements

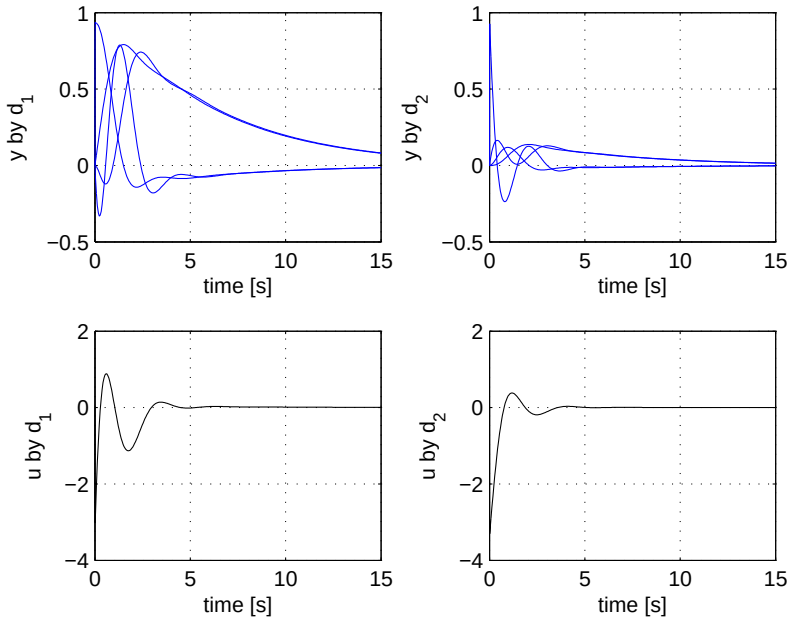

Figure 5.5: Impulse responses with the LMI calculated feedback matrix

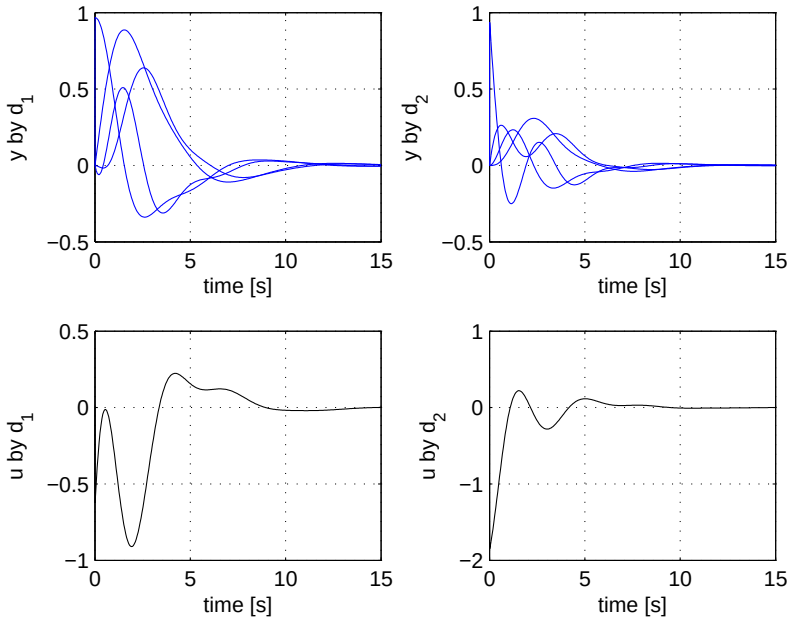

Figure 5.6: Impulse responses with the MOGA calculated feedback matrix

for Problem 4.2, but as described in Section 3 the designer can chose from the obtained solutions (in this case 50) that one that suits his/her needs the most.

5.5 Comparing MOGA with an Iterative, Less Conservative Approach

Here the multi-objective state-feedback controller design is compared with the proposed in [13] iterative procedure, which breaks the difficult non-convex problem into a series of convex subproblems. The algorithm is as follows

1. *Initialization:* Given an achievable γ find the central controllers K_c and use it as a starting point for the algorithm.

2. *Compute X_0 and Y_{k_0}:* For a given K_0, obtain a positive define matrix X_0 by solving maximal determinant LMI problem. Compute the observability gramian Y_{k_o}.

3. *H_2 cost reduction:* Solve constraint minimization LMI problem and obtain new controller with improved value of H_2 cost function.

4. *Stopping criteria:* Repeat step 2 and 3 until either the boundary of the set of static feedback gain matrixes is reached ($\|T_{\infty\infty}\|_\infty = \gamma$) or K_0 is the global minimum of $\|T_{22}\|_2^2$.

That method is shown to give controllers with better performance than the central controllers.

For the means of comparison the second example from [13] is used. The lower performance boundaries are $\gamma = 5.0728$ and $\nu = 10.1278$. The controllers obtained with the algorithm for state-feedback controllers are shown to achieve the minimal H_2 norm for a set γ, meaning that the same performance values are obtained by recomputing the norms for the closed-loop system.

In Figure 5.7 the performance of the controllers obtained with MOGA are compared to those obtained by the iterative approach. One can see, that the results from MOGA are very close to the boundary from the iterative approach. Furthermore the iterative approach is computationally expensive, because to find a single controller about 30 iterations are necessary and on each min_cx and *mindet* problems are solved, where in a single run MOGA computes 50 controllers. The other two big advantages of using the genetic algorithms approach are that it doesn't require advanced knowledge on LMIs and as it will be shown it can be extended to the output-feedback dynamic controller case, where as the iterative mixed H_2/H_∞ approach cannot.

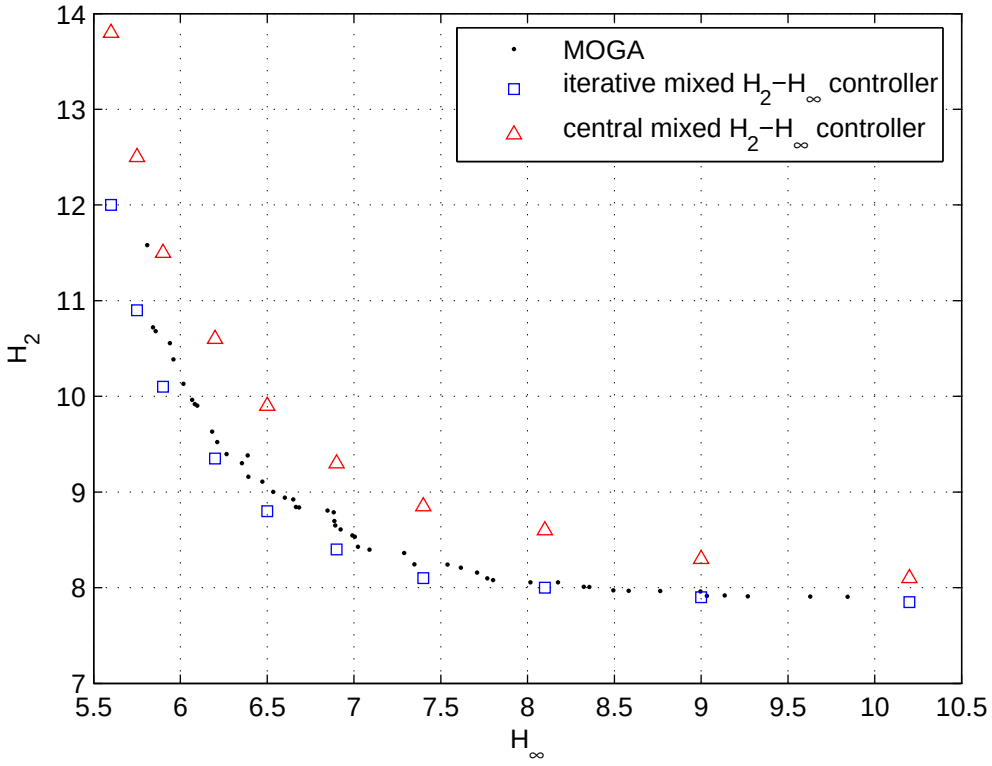

Figure 5.7: Comparing the result from MOGA with the iterative procedure

6 Output Feedback Controller Design

For the case of output feedback a dynamic controller is used

$$\left|\begin{array}{ll}\dot{\xi} & = A_k\xi + B_k y \\ u & = C_k\xi + D_k y\end{array}\right.$$

First, the generalized plant model with uncertainty in all three of the parameters is derived. The model is used later in the LMI mixed H_2/H_∞ problem formulation and central controllers are computed for different values of γ. The so constructed Pareto-surface is compared with the one obtained with the MOGA approach. Finally to meat the design requirements the Q, Q_e, R and R_e matrixes are adjusted and a model with only varying k is used. Based on the results conclusions are made.

6.1 Model of the Uncertainty

Using the cart system state space model (2.4) one can represent the system in the form, shown in Figure 6.1

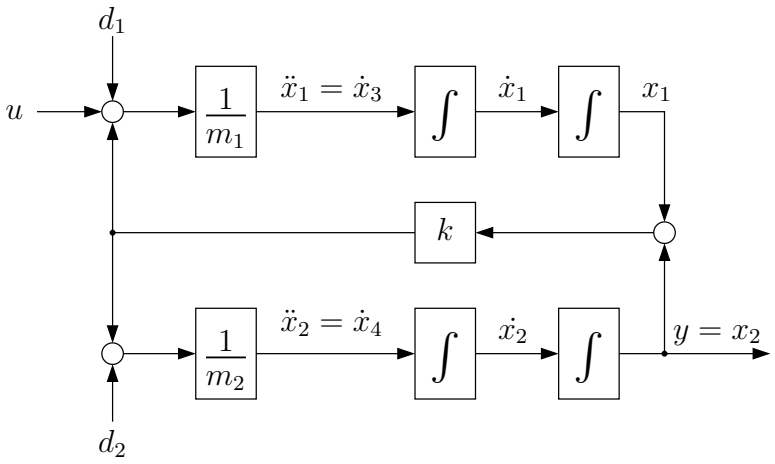

Figure 6.1: Plant representation

The range of the system parameters is $0.5 \leq k, m_1, m_2 \leq 1.3$. The variations can be represented as

$$z_k = \bar{k} + \tilde{k}\delta_k \tag{6.1}$$

$$z_{m_i} = (\bar{m}_i + \tilde{m}_i\delta_{m_i})^{-1} \tag{6.2}$$

where $i = 1, 2$; $\bar{k} = \bar{m}_i = 1$; $\tilde{k} = \tilde{m}_i = 0.3$ and $-1 \leq \delta_{k,m_i} \leq 1$.

This, can be converted to corresponding upper and lower LFT form, as shown in Figure 6.2, where

$$M_k = \begin{bmatrix}\bar{k} & \tilde{k} \\ 1 & 0\end{bmatrix}$$

$$M_{m_i} = \begin{bmatrix} -\frac{\tilde{m}_i}{\bar{m}} & \frac{1}{\bar{m}} \\ -\frac{\tilde{m}_i}{\bar{m}} & \frac{1}{\bar{m}} \end{bmatrix}$$

The signals w_k, w_{m_1} and w_{m_2} are referred as external inputs and z_k, z_{m_1} and z_{m_2} as

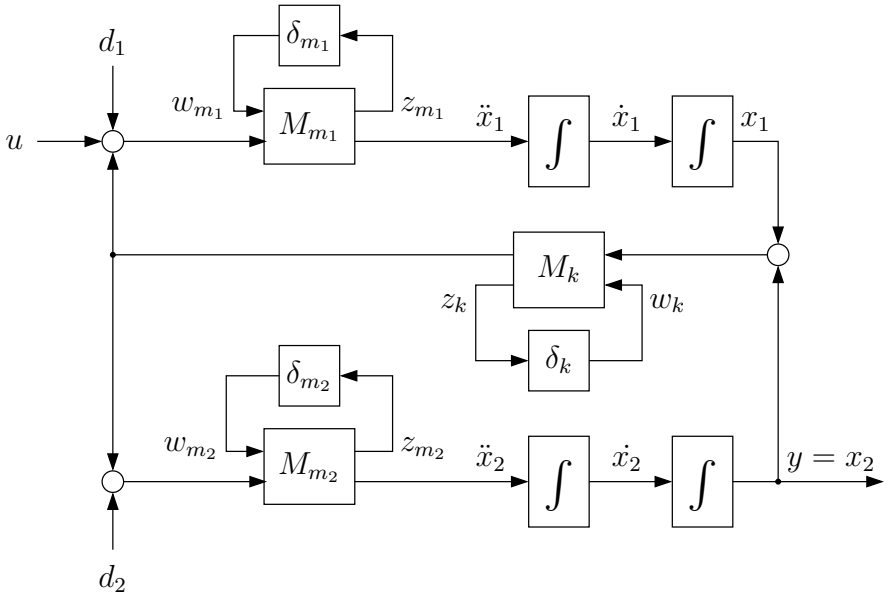

Figure 6.2: Plant representation with parameter uncertainty in LFT form

additional outputs. The state-space description of the cart system than is

$$\left|\begin{array}{l} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\bar{k}\frac{1}{\bar{m}_1} & \bar{k}\frac{1}{\bar{m}_1} & 0 & 0 \\ \bar{k}\frac{1}{\bar{m}_2} & -\bar{k}\frac{1}{\bar{m}_2} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{\tilde{m}_1}{\bar{m}_1} & 0 & -\tilde{k}\frac{1}{\bar{m}_1} & \frac{1}{\bar{m}_1} & 0 & \frac{1}{\bar{m}_1} \\ 0 & -\frac{\tilde{m}_2}{\bar{m}_2} & \tilde{k}\frac{1}{\bar{m}_2} & 0 & \frac{1}{\bar{m}_2} & 0 \end{bmatrix} \begin{bmatrix} w_{m_1} \\ w_{m_2} \\ w_k \\ d_1 \\ d_2 \\ u \end{bmatrix} \\ \\ \begin{bmatrix} z_{m_1} \\ z_{m_2} \\ z_k \\ y \end{bmatrix} = \begin{bmatrix} -\bar{k}\frac{1}{\bar{m}_1} & \bar{k}\frac{1}{\bar{m}_1} & 0 & 0 \\ \bar{k}\frac{1}{\bar{m}_2} & -\bar{k}\frac{1}{\bar{m}_2} & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} -\frac{\tilde{m}_1}{\bar{m}_1} & 0 & -\tilde{k}\frac{1}{\bar{m}_1} & \frac{1}{\bar{m}_1} & 0 & \frac{1}{\bar{m}_1} \\ 0 & -\frac{\tilde{m}_2}{\bar{m}_2} & \tilde{k}\frac{1}{\bar{m}_2} & 0 & \frac{1}{\bar{m}_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_{m_1} \\ w_{m_2} \\ w_k \\ d_1 \\ d_2 \\ u \end{bmatrix} \end{array}\right| \tag{6.3}$$

Now the three uncertainty parameters δ_k, δ_{m_1} and δ_{m_2}, can be grouped to a single matrix uncertainty Δ and the represented as upper LFT, as shown in Figure 6.3, where

$$\begin{bmatrix} w_k \\ w_{m_1} \\ w_{m_2} \end{bmatrix} = \begin{bmatrix} \delta_k & 0 & 0 \\ 0 & \delta_{m_1} & 0 \\ 0 & 0 & \delta_{m_2} \end{bmatrix} \begin{bmatrix} z_k \\ z_{m_1} \\ z_{m_2} \end{bmatrix} \iff w_\infty = \Delta z_\infty$$

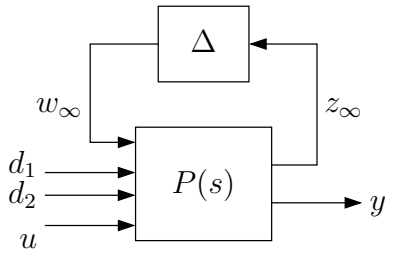

Figure 6.3: Uncertainty representation of the generalized plant in upper LFT form

From the small-gain theorem it follows, that the closed-loop system will be robustly stable for any parameter uncertainty $-1 \leq \|\Delta\| \leq 1$, if $T_{z_\infty w_\infty} \leq 1$.

During the design, the d_1 and d_2 disturbance inputs are not used. For the comparison of the LMI based solution with the multi-objective optimization one, the following matrixes are used:

$$R = 1; \quad Q = C_y^T C_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad R_e = 0; \quad Q_e = B_u B_u^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The matrixes B_2, C_2, D_{2u} and D_{y2} are created according equation (2.7)

Separating the plant outputs to measured plant output y and robust outputs z_∞ and by adding the performance input and output channels w_2 and z_2 the generalized plant

$$P(s) = \left[\begin{array}{c|ccc} A & B_\infty & B_2 & B_u \\ \hline C_\infty & D_{\infty\infty} & D_{\infty 2} & D_{\infty u} \\ C_2 & D_{2\infty} & D_{22} & D_{2u} \\ C_y & D_{y\infty} & D_{y2} & D_{yu} \end{array}\right] \tag{6.4}$$

can be rewritten as

$$P(s) = \left[\begin{array}{ccc|ccc|ccc|c} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\bar{k}\frac{1}{\bar{m}_1} & \bar{k}\frac{1}{\bar{m}_1} & 0 & 0 & -\frac{\tilde{m}_1}{\bar{m}_1} & 0 & -\tilde{k}\frac{1}{\bar{m}_1} & 0 & 0 & \frac{1}{\bar{m}_1} \\ \bar{k}\frac{1}{\bar{m}_2} & -\bar{k}\frac{1}{\bar{m}_2} & 0 & 0 & 0 & -\frac{\tilde{m}_2}{\bar{m}_2} & \tilde{k}\frac{1}{\bar{m}_2} & 0 & 0 & 0 \\ \hline -\bar{k}\frac{1}{\bar{m}_1} & \bar{k}\frac{1}{\bar{m}_1} & 0 & 0 & -\frac{\tilde{m}_1}{\bar{m}_1} & 0 & -\tilde{k}\frac{1}{\bar{m}_1} & 0 & 0 & \frac{1}{\bar{m}_1} \\ \bar{k}\frac{1}{\bar{m}_2} & -\bar{k}\frac{1}{\bar{m}_2} & 0 & 0 & 0 & -\frac{\tilde{m}_2}{\bar{m}_2} & \tilde{k}\frac{1}{\bar{m}_2} & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{6.5}$$

6.2 LMI-based Solution

The mixed H_2/H_∞ problem can be solved by LMIs as described in Section 2, i.e. changing the design variables according equation (2.21) and solving a constrained LMI derived from the combination of inequalities (2.24), (2.25) and (2.20). This mixed H_2/H_∞ controller design procedure work by setting achievable value for γ and computing a controller, that has minimal H_2 norm (by minimizing trace(W)).

By solving the robust design problem, the minimal achievable values for the H_∞ norm is calculated: $\gamma_{opt} = 0.9660$.

The values for the H_2 norm of the closed-loop system for 53 set values for γ as well as the lines, corresponding to the minimal achievable norms are displayed in Figure 6.4.

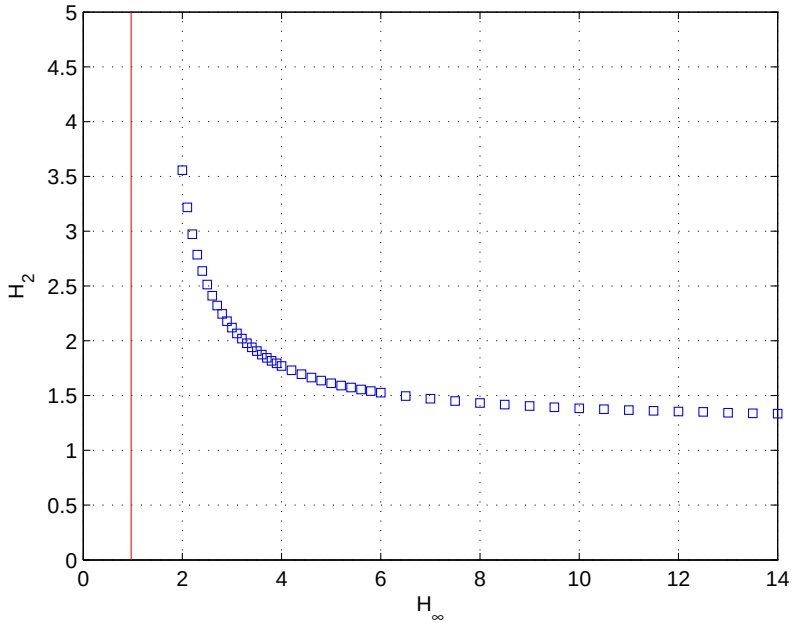

Figure 6.4: LMI solution to the mixed H_2/H_∞ design problem

From that plot one can see the trade off between H_2 and H_∞ norm - reducing the value of γ increases the H_2 norm. Further more for $\gamma < 1.99$ it is not possible to find solution of the LMI problem, which is the reason why only 49 points are plotted. The total computation time is 78.5 seconds on 735 MHz P3 processor.

6.3 MOGA-based Solution

4th-order Controller

Here we start first with full order controller (4th order). The controller can be written in the form of normalized ($a_0 = 1$) transfer function

$$K(s) = \frac{b_0 s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4}{s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4} \tag{6.6}$$

This form is preferred than the state-space, because it has only 9 coefficients to be adjusted. If a state space form is used, the number of coefficients in the general case is 16 for A matrix plus 4 for B and C matrixes, or totaly 24 coefficients. One can, however replace the transfer function representation with a canonical state-space one, with the same number of optimization coefficients.

The 9 design variables are represented in standard coding. The allowed ranges for the variables are: $b_{0,1,2,3,4} \in [-5\,10^5,\, 0]$; $a_{1,2,3,4} \in [0,\, 5\,10^5]$. The optimization runs are done with population size of 50 and 200 number of generations. In Figure 6.5 the results from such run are compared with the ones from run with population size of 100 and 500 generations, where the minimal achieved γ is 1.04.

Because for the run with 200 generations the minimal value of 1.48 for γ was achieved it was found necessary to increase the population size to 100, which brought down γ to 1.24. The reason for this bad convergence is the relatively large search space and number of design variables. Tests were made with fragment-exponential coding, but no improvement was recognized.

In Figure 6.6 the results from the multi-objective technique run increased population size are plotted together with the results from the standard LMI based one. One can easily see that with MOGA a Pareto-surface dominating the surface form the standard approach is achieved. The minimal γ achieved with MOGA is 1.24 versus 1.99 with LMI. This however was done on the expense of almost 16 times longer computation time (1100 s) or taking in account that, there are actually 100 controllers on the Pareto-surface - only 8 times longer time.

To take use of the other advantage of the MOGA approach, i.e. the possibility to fix the controller order, the design is repeated for 3rd and 2nd order controllers. It should be noted, that the cart-system cannot be stabilized with static gain or 1st order controller (verifiable via a root-locus plot).

3rd-order Controller

In Figure 6.7 the Pareto-surface for the third order controller is compared to the one computed for the full controllers with the standard LMI approach. One can clearly see, that despite the lower controller order the MOGA approach has successfully found controllers with better values for the two cost functions. This is possible, due to the smaller conservatism with this approach. The needed computation time is 540 seconds for population

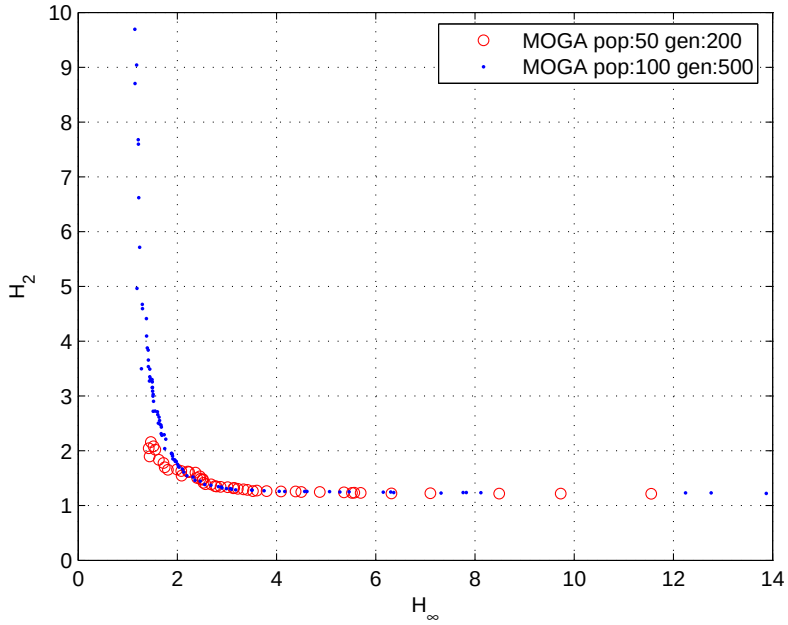

Figure 6.5: MOGA results for 4th order controllers with different number of generations

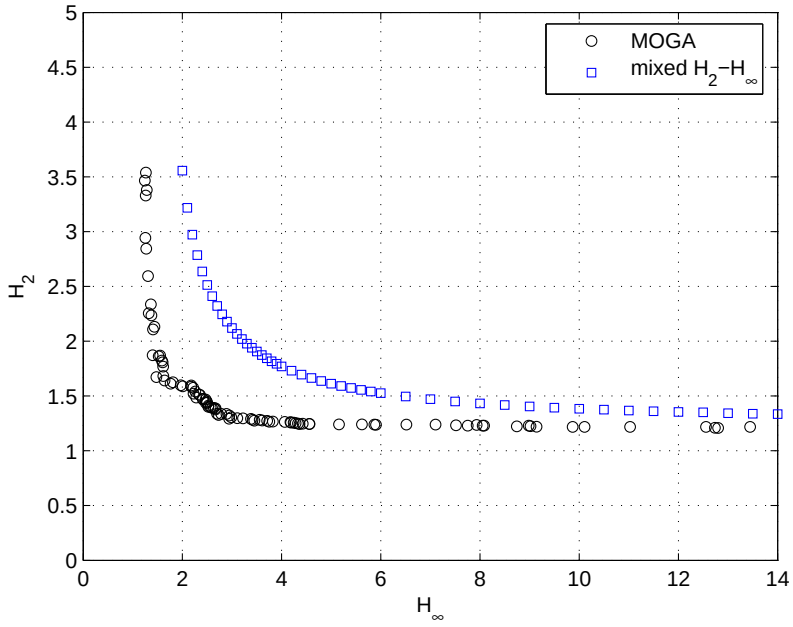

Figure 6.6: Comparison between 4th order controllers computed with MOGA and with LMI approach

size of 50 and 200 generations, or only 7 times slower than the LMI based approach. The

minimal achieved γ is 1.1498 with H_2 norm of 11.58. The controller transfer function is

$$K(s) = \frac{b_0 s^3 + b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3} \tag{6.7}$$

The permitted coefficient ranges are as follows: $b_{0,2,3} \in [-10^5, 0]$, $b_1 \in [-10^5, 10^5]$, and $a_{1,2,3} \in [0, 10^5]$.

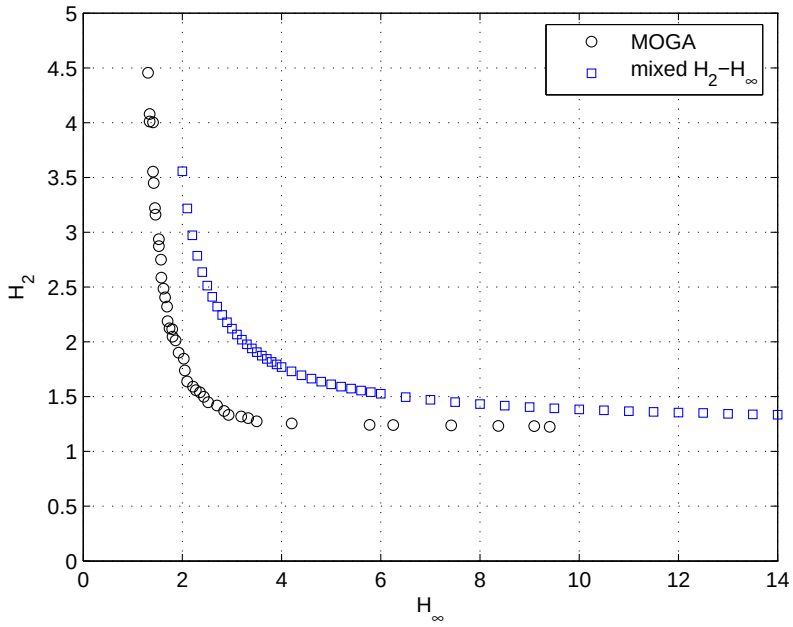

Figure 6.7: 3rd order controllers computed with MOGA and the full order controllers computed with the standard LMI approach

2nd-order Controller

Finally the second order controllers computed with the MOGA approach are compared towards the full order LMI controllers - Figure 6.8. The controller transfer function is

$$K(s) = \frac{b_0 s^2 + b_1 s + b_2}{s^2 + a_1 s + a_2} \tag{6.8}$$

The coefficient are limited within ranges giving stable closed loop system: $b_0 \in [0, M]$; $b_{1,2} \in [-M, 0]$ and $a_{1,2} \in [0, M]$, where M was set 10^4. Among the controllers there is one, achieving $\gamma = 1.1396$, but on the cost of very high H_2 norm: $\nu = 220.45$. By setting $M = 10^6$ controller achieving $\gamma = 1.06$ with $\nu = 66.38$ was achieved, but on the expense of very high values for the controller coefficients. The optimization run time is 492 seconds.

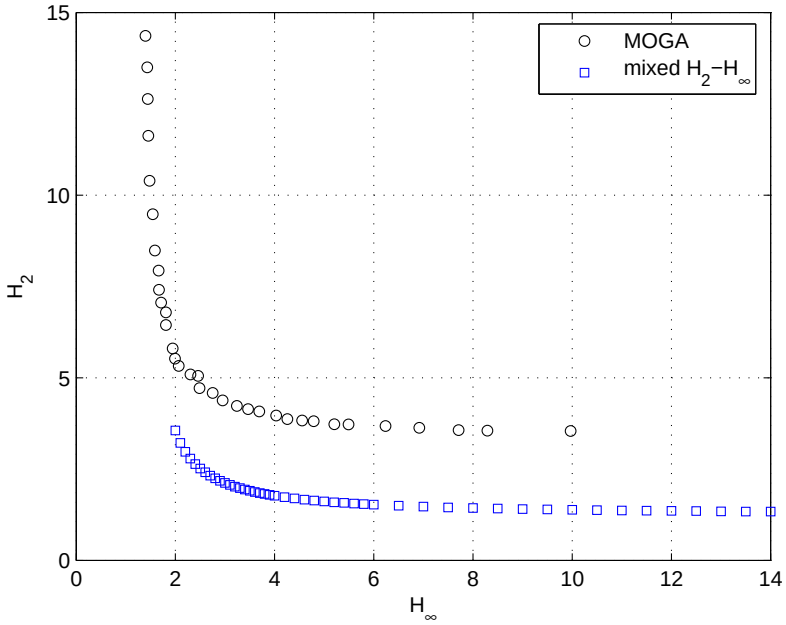

Figure 6.8: 2nd order controllers computed with MOGA and the full order controllers computed with the standard LMI approach

Comparison between the different order controllers

In Figure 6.9 the obtained Pareto-surfaces for the 4th, 3rd and 2nd order controller are plotted together. One can see that the 3rd order controller, achieves about the same values for the cost functions as the 4th order controller achieved with MOGA. However, when the controller order is further reduced to 2, there is a clear distinguish between the Pareto-surfaces. The second order controller cannot achieve as good performance and robustness measures as the third and forth order controllers. This can be expected, because for the low order controller there are fewer coefficients to adjust and thus fewer degrees of freedom that can be used to achieve small values of the design criterion.

6.4 Comparison Between the LMI and MOGA Results

From the achieved results one can clearly see that the MOGA mixed H_2/H_∞ approach is less conservative than the standard LMI one in term of achieving Pareto-surface with lower values for the H_2 and H_∞ norms for the same controller order.

Furthermore the genetic approach was shown to achieve better solutions than the standard LMI approach even for low order controllers. It should be noted that for any controller order smaller than the system order the problem is not convex and there is no analytical solution. Furthermore there is no less conservative approach, like the one in [13] working for the output feedback case even for full controller order.

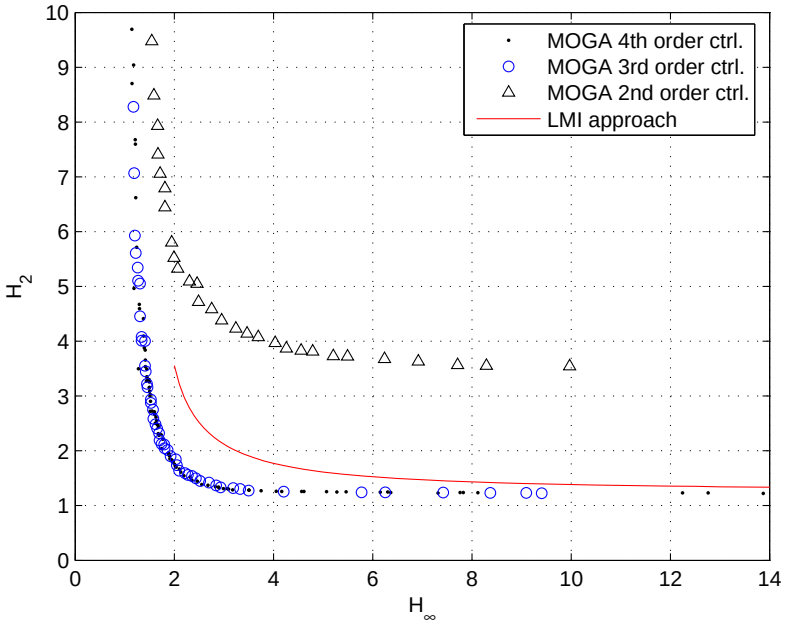

Figure 6.9: Comparison between 2nd, 3rd and 4th order controllers computed with MOGA

6.5 Meeting the Design Requirements

Based on the results achieved in section 6.3, it can be concluded that third order controller is capable of achieving almost the same values for the cost functions as the 4th, full order controller. Since the performance of the second order controllers is significantly worst, third order controller was found to be reasonable trade-off between controller complexity and performance and it is used in the rest of the work to meet the design requirements of problem B, described in section 4 on page 20.

The model used for the design incorporates only the uncertainty in k in the range $0.5 \leq k \leq 2.0$ ($m_1 = m_2 = 1$), which can be represented as $k = \bar{k} + \delta_k \tilde{k} = 1.25 + \delta_k 0.75$, where $-1 \leq \delta_k \leq 1$. Using (6.3) the state space representation of the plant becomes

$$\left|\begin{array}{l} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\bar{k}\frac{1}{m_1} & \bar{k}\frac{1}{m_1} & 0 & 0 \\ \bar{k}\frac{1}{m_2} & -\bar{k}\frac{1}{m_2} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\tilde{k}\frac{1}{m_1} & \frac{1}{m_1} \\ \tilde{k}\frac{1}{m_2} & 0 \end{bmatrix} \begin{bmatrix} w_k \\ u \end{bmatrix} \\ \\ \begin{bmatrix} z_k \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_k \\ u \end{bmatrix} \end{array}\right. \tag{6.9}$$

where $w_\infty = \Delta z_\infty$, $w_\infty = w_k$, $\Delta = \delta_k$ and $z_\infty = z_k$.

The minimal achievable γ for this system is 0.6017, which makes the problem easier to solve than the case with all parameter varying.

The results from an optimization run with the already defined matrixes R,Q,R_e and Q_e gives the Pareto-surface shown in Figure 6.10.

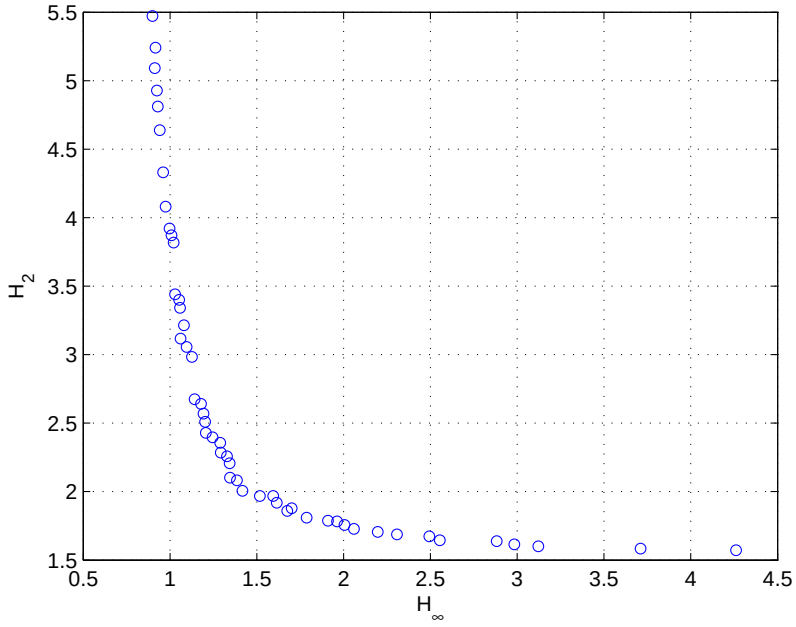

Figure 6.10: Pareto-surface between H_∞ and H_2 norm for 3rd order controllers

As a first step, the settling time and the maximal control values of the already computed controllers are checked. To do this, impulse disturbance first on channel d_1 and than on d_2 are applied and from the simulation results the maximal form the two settling times (within ± 0.1 zone) and the maximal from the two control signals are taken. The way of computing the impulse responses is discussed shortly in Appendix A.

In Figure 6.11 the settling time (T_{set}) and the maximal control values (U_{max}) of the all 50 controllers forming the Pareto-surface are plotted.

From Figures 6.10 and 6.11 one can see that there are controllers in the final population that satisfy the robust requirements (i.e. have $\gamma \leq 1$) but there is none satisfying the performance requirements (have peak control value under 1 and settling time under 15 seconds).

Now the R, Q, R_e and Q_e matrixes can be used to weight the system states, control input and w_2 inputs and help meeting the design requirements. If, however R_e is set to a value different than 0 and a proper controller is used ($b_0 \neq 0$), than the closed-loop system will not be strictly proper (D matrix will be different than zero matrix) and the H_2 norm will be infinity. In order to allow R_e to be different from zero, one has to impose a strictly proper controller structure. To be able to compare the achievable performance of a strictly proper controller with the one of proper controller, MOGA run is performed

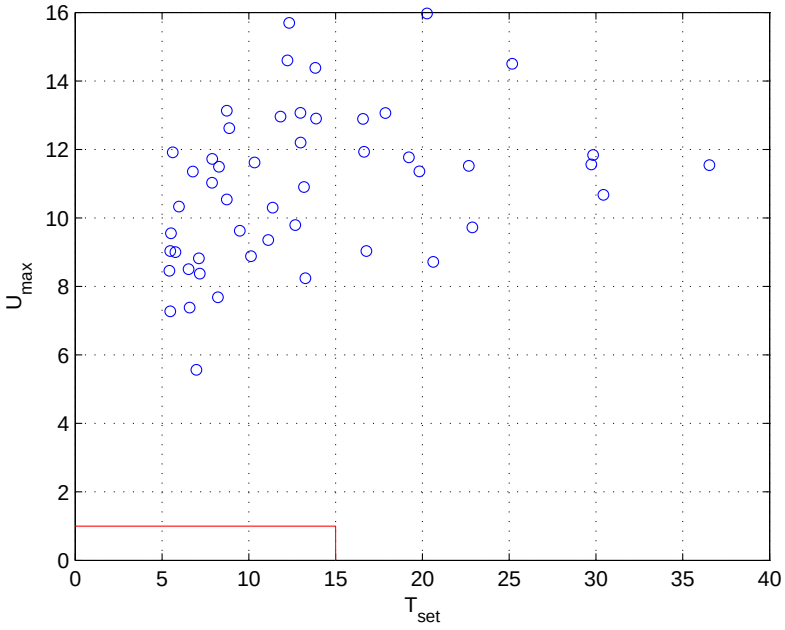

Figure 6.11: Settling time and maximal control value for the 3rd order controllers

and the results are shown in Figures 6.12 and 6.13. On both figures with dots are given the results for the proper controllers shown above. Among the population of strictly proper controllers there are several meeting the performance requirements, but on the expense of large value for γ (over 3).

In order to easily determine proper values for the matrixes, Q and Q_e are set to be equal to $Q = qI$ and $Q_e = q_e I$, where I is an unit matrix, and q and q_e are weights. R is set to be 1 and $R_e = r_e$ is also tunable. A logarithmic step gridding for the three tunable parameters is done in the range $0.0001 \leq r_e, q, q_e \leq 1000$. For each combination a LQG problem is solved (with the standard `lqg` solver in MATLAB) and the settling time and maximal control values for the calculated full order controller are computed. Based on those results, appropriate combinations for the tuning parameters is chosen. After several optimization runs and manual adjustment, with the combination $R = 4$, $r_e = 0.0001$, $q = 0.1$ and $q_e = 10$ a controller satisfying the settling time and maximal control values is derived (negative feedback is assumed)

$$K(s) = \frac{-99.7337s^2 + 26.0268s + 3.8147}{s^3 + 26.1626s^2 + 65.5720s + 90.6388} \tag{6.10}$$

However this controller doesn't meet the robustness requirements and has $\gamma = 1.1150$.

The resulting Pareto-surface for the population between the H_∞ and H_2 norm is shown in Figure 6.14. The controller is marked with filled circle. In Figure 6.15 the settling times and maximal control values for the population are visualized and the numbers are ordered in increasing H_∞ of the controllers (the selected controller is number 33).

One can clearly see that there is not a clear dependence between the H_2 norm and the

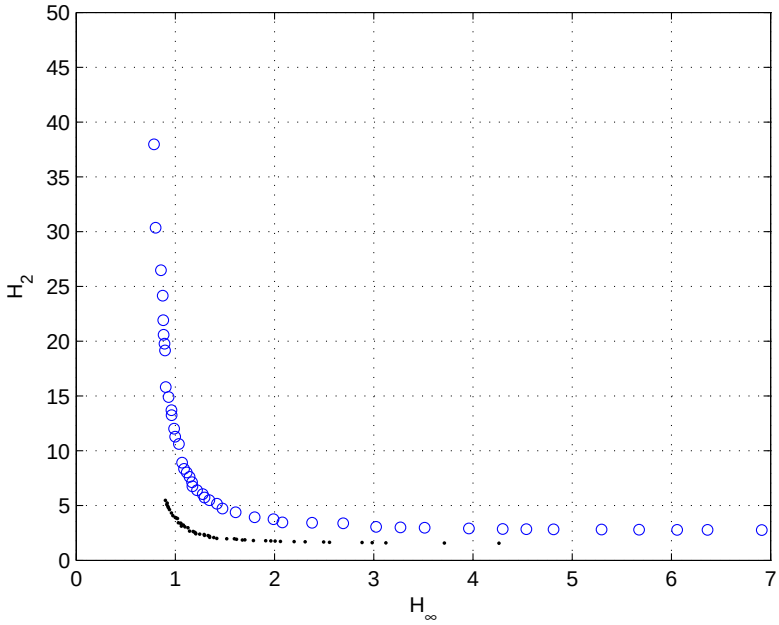

Figure 6.12: Pareto surface for 3rd order proper and strictly proper controllers

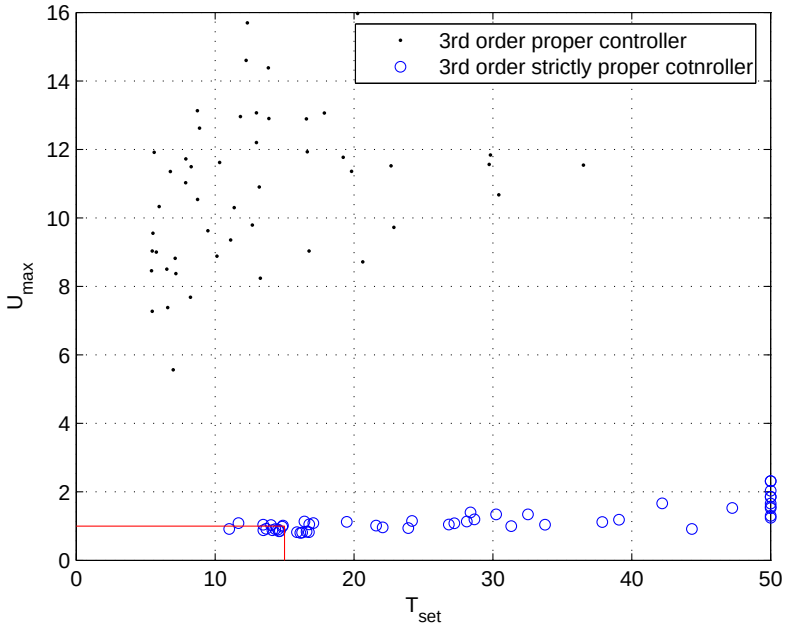

Figure 6.13: Settling time and maximal control for 3rd order proper and strictly proper controllers

settling time and maximal control value. This is due to the fact, that the H_2 norm is a weighted sum of the integral control signal and system states, thus it is possible controllers

to achieve the same value for the norm by having different characteristics, e.g. very large control signal but very small change of the states and vice versa. Furthermore the maximal control signal is a non-linear measure. Attempt was made to solve this problem and find a better solution, by removing the H_2 criteria and adding two separate ones: settling time and maximal control value. However from the performed several optimization runs a controller meeting all design requirements was not achieved.

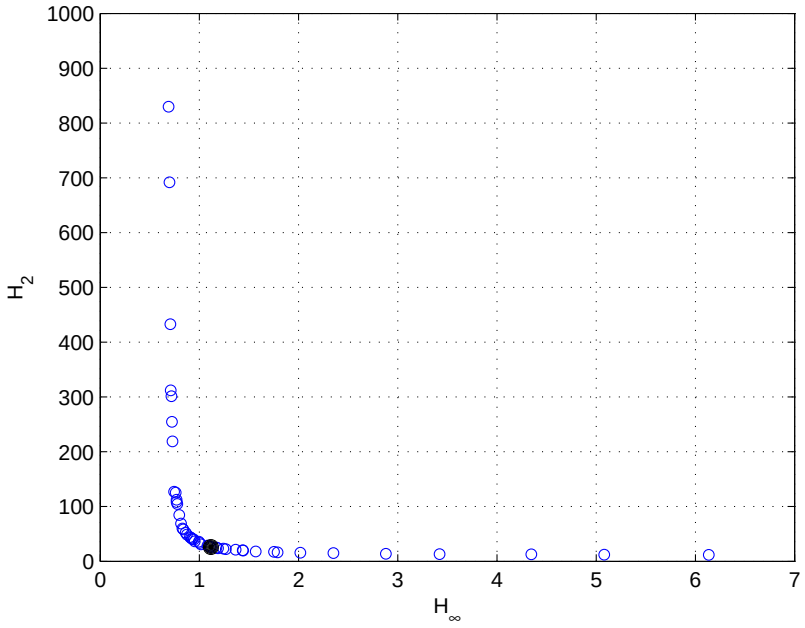

Figure 6.14: H_∞/H_2 trade off for the last population with $R = 4$,$r_e = 0.0001$,$q = 0.1$ and $q_e = 10$

The overall score of the controller (6.10) is calculated using the method presented in [13] and resented in Table 6.5

Design	PM [deg]	GM [dB]	T_{set} [s]	U_{max}	$k_{min} \rightarrow k_{max}$	p_m	Meets all	Score
Requirements	30	6.0	15	1	0.5 $\rightarrow$ 2.0	0.30		
Controller from [13]	35	6.1	14.5	0.75	0.44$\rightarrow$ 3.9	0.45	Yes	7.4
3rd order MOGA	20	1.97	14.4	0.86	0.46 $\rightarrow$ 35.05	0.22	No	-3.32

Table 1: Scores for the controller

Comparing the scores, one should keep in mind, that the computed with MOGA controller is 3rd (lower) order, derived by optimization of non-convex problem and still it

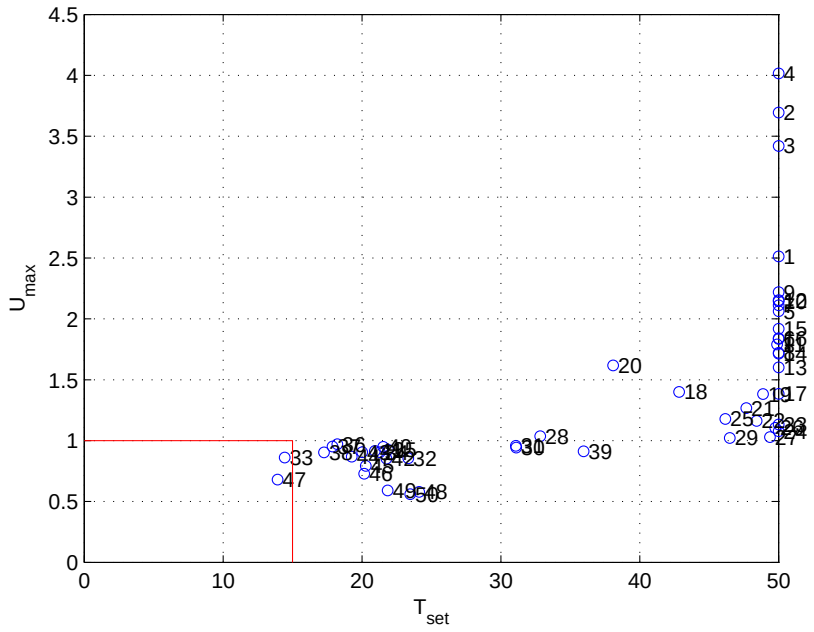

Figure 6.15: Settling time and maximal control value for the last population with $R = 4$,$r_e = 0.0001$,$q = 0.1$ and $q_e = 10$

achieves better score than 5 other designs published in "Journal for Guidance, Control and Dynamics" and compared in [13].

The impulse responses of the closed-loop to disturbances on both carts are shown in Figure 6.16. The upper plots are the output of the system and the lower are the control signal applied. The left two figures are for the case of impulse disturbance on the d_1 channel and on the right for the d_2 channel.

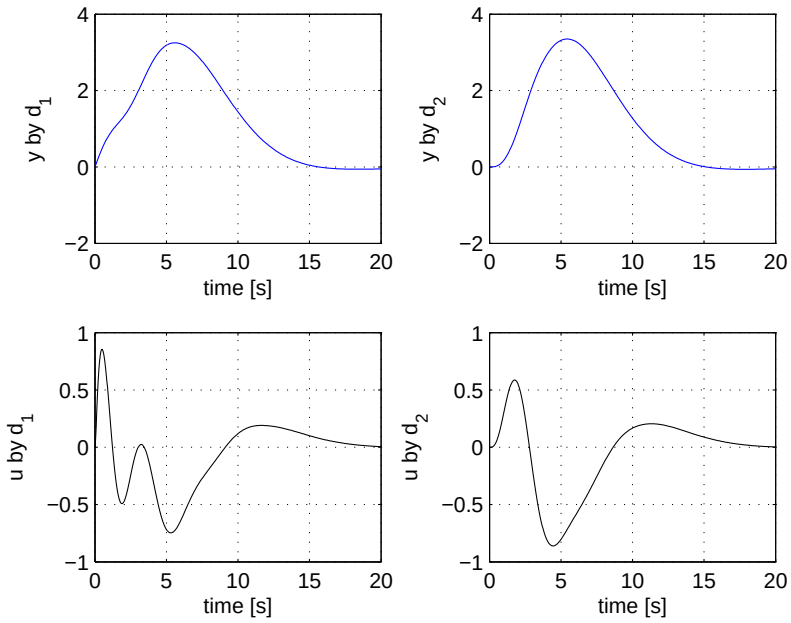

Figure 6.16: Responses of the closed-loop system to impulse disturbances

7 Conclusions

In this project work a new less-conservative approach for solving the mixed H_2/H_∞ design problem is proposed and applied to the ACC benchmark problem. The approach is based on solving multi-objective optimization by genetic algorithms. For this purpose the Strengthen Pareto Evolutionary Algorithm 2 was utilized. The use of genetic optimization, allows the work with with fixed structure and order controllers.

The results of applying this multi-objective genetic algorithms (MOGA) are compared with the standard solutions, based on solving convex LMI problems, for the cases of state feedback and dynamic output feedback. The conservatism of standard LMI design problem is shown to be due to the Lyapunov paradigm, i.e. using a common P matrix when combining the H_2 and H_∞ problems to a mixed H_2/H_∞ problem.

First the MOGA approach is applied on the state-feedback design problem. It is compared with the standard LMI-based solution and experimentally shown to outperform it in terms of achieving lower values for the H_2 norm by the same set value for the H_∞ norm. Next it is compared with a less-conservative iterations-based LMI approach, shown previously to achieve less-conservative design compared to the central controller and other design techniques. Although MOGA was not able to achieve better results it found controllers with very close values for the H_2 and H_∞ norms.

As a next step the output feedback dynamic controller design problem is considered. Comparing the performance of MOGA with the standard LMI approach for the full controller order case, MOGA is experimentally shown to give better solutions (Pareto-surface dominating the one for the central controllers). Since MOGA allows working with a fixed order and structure controllers optimization runs with 3rd and 2nd order controllers are performed. Based on the results it is shown, that due to the conservatism of the standard LMI method, even a 3rd order controller designed with the MOGA approach achieves lower H_2 norm than the central controller for the same H_∞ norm. One should note, that there is no analytical approach, extending the iterative approach or other less-conservative approach, to the case of dynamic output feedback, yet.

Finally MOGA was applied for the synthesis of controller meeting the ACC benchmark design requirements. Although resulting in a not a convex problem, 3rd controller order was selected as a trade off between controller simplicity and performance. From the performed optimization runs a controller, meeting all design requirements was not found, but instead one achieving the desired settling time, maximal control value and $\gamma = 1.115$ was selected. The score for the controller is -3.32, which is better than 5 previously published designs working with 4th and higher order controllers.

In conclusion it could be summarized that the proposed MOGA method for solving mixed H_2/H_∞ design problems:

+ achieves less-conservative solutions than the central controller both for the case of state and output feedback;

+ can work with fixed structure and low order controllers;

+ works well for non-convex problems (i.e. low order controller design);

+ can easily be extended by other design requirements (e.g. poles region constraints, time domain requirements, etc.);

+ can be combined with other less-conservative techniques (e.g. each gene can code also a D-scaling matrix);

- is computationally more expensive than solving the convex LMI problem (when a convex problem formulation is available);

0 requires the designer to chose among the provided at the end set of Pareto-optimal controllers;

0 because its specific features SPEA2 attempts to span equally distributed points over the Pareto-surface and thus, sometimes can omit regions of interest for the designer.

A Appendix - Computing the Impulse Responses

The computation of the impulse responses is done by performing simulations in SIMULINK, which is computationally expensive, especially for controllers with high values for the coefficients. Another way to compute the impulse response is using the MATLAB function `impulse`. This however has the problem that for high value of the controller coefficients does not compute the real response when automatic sampling time is used. Although that manually setting the sampling time to sufficiently small values allows the function to compute the actual impulse response, this approach appears to be even more computationally expensive, because the small sampling time is not necessary during the whole simulation.

Besides those two factors (slow simulations of controllers with large coefficients and incorrect results with `impulse` functions) another reason, why big coefficients are undesired for practical implementations is the danger of saturation and numerical instability. To cope with those problems, for the design of output feedback controller meeting the design requirements, the absolute values of the controllers coefficients are limited to 100. This however doesn't fully solve the difference between simulating the impulse response by SIMULINK and with `impulse` function - see Figure A.1[3]. Thus, although computationally more expensive, the SIMULINK model is used for simulating the impulse responses in the whole project work.

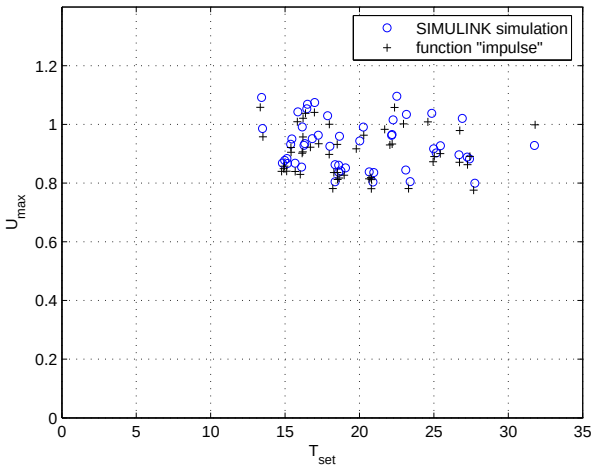

Figure A.1: Comparison of the results from computing the impulse response in SIMULINK and by "impulse" function

Performing MOGA run with the new range for the coefficients, yields controllers whose H_2 and H_∞ costs are shown in Figure A.2 and T_{set} and U_{max} shown in Figure A.3. In both figures, with small dots show the results for the 3rd order controllers with the larger

[3]For the experiments the plant with uncertainty in all parameters is used, but the results are valid also for the case of only varying k

range for the coefficients. One can see, that whereas there is not a big difference between the two Pareto-surfaces (γ is increased to 1.19), there is a significant improvement for the value for U_{max}, due to the smaller search space.

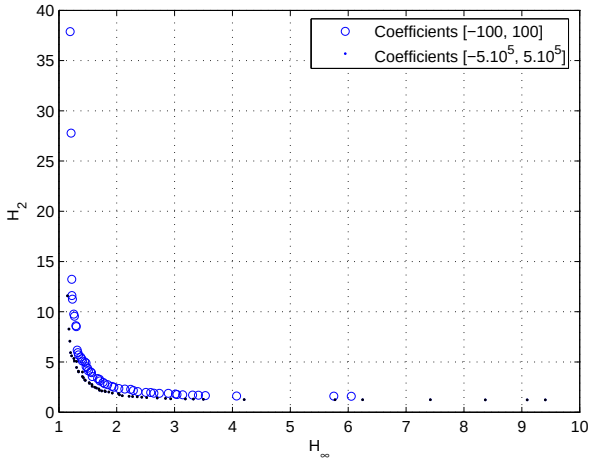

Figure A.2: Pareto-surface with 3rd order controllers with smaller range for the coefficients

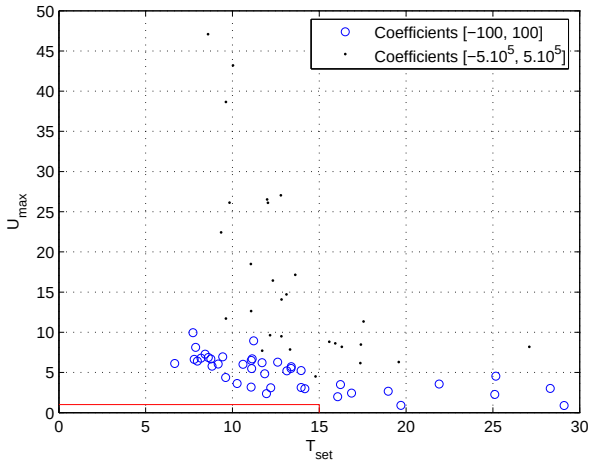

Figure A.3: Settling time and maximal control for the controllers from Figure A.2

References

[1] Stephen Boyd, L. Ghaoui, E. Feron and V. Balakrishnan. "Linear Matrix Inequalities in System and Control Theory", *Society for Industrial and Applied Mathematics*, Philadephia, 1994

[2] Gary Balas, R. Chiang, A. Packard, and M. Safonovm. "Robust Control Toolbox For Use with MATLAB, ver.3". *The MathWorks Inc.*, 2005

[3] Pierre Apkarian, G. Becker, P. Gahinet and H. Kajiwara. "LMI Techniques in Control Engineering from Theory to Practice". *Workshop Notes CDC 1996*, Kobe, Japan, 1996

[4] Herbert Werner. "Optimal and Robust Control - lecture notes". Technical University Hamburg-Harburg, Institute of Control Engineering, 2004

[5] Carlos de Fonseca. "Multiobjective Genetic Algorithms with Applications to Control Engineering Problems", PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, September 1995.

[6] Akira Oyama. "Wing Design Using Evolutionary Algorithms", PhD thesis, Department of Space Engineering, Tohoku University, March 2000

[7] Sanaz Mostaghim. "Multi-Objective Evolutionary Algorithms. Data Structures, Convergence, and Diversity, PhD thesis, Fakultt Elektrotechnik, Informatik und Mathematik der Universitt Paderborn, Paderborn, Germany, November 2004

[8] E. Zitzler, K. Deb, and L. Thiele. "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evolutionary Computation*, 8(2), 2000, pp. 173195.

[9] E. Zitzler, M. Laumanns and L. Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems*, Proceedings of the EUROGEN2001 Conference, 2001, pp.95-100.

[10] S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler. "PISA - A Platform and Programming Language Independent Interface for Search Algorithms", *Conference on Evolutionary Multi-Criterion Optimization*, 4/2003, pp 494-508.

[11] Bong Wie, Dennis Bernstein. "Benchmark Problems for Robust Controller Design", *Journal of Guidance Control and Dynamics*, vol.15, No.5, 1992

[12] Peter Thompson. "Classical/H_2 Solution for a Robust Controller Design Benchmark Problems", *Journal of Guidance Control and Dynamics*, vol.18, No.1, 1995

[13] B. Halder and T. Kailath. "LMI Based Design of Mixed H_2/H_∞ Controllers: The state Feedback Case", proceedings of *American Control Conference*, San Diego, California, 1999