Design of Reduced-Order Controllers via $H_\infty$ and Parametric Optimization: Comparison for an Active Suspension System.

D. U. Campos-Delgado*, R. Femat and E. Ruiz-Velázquez*
Dpto. Matemáticas Aplicadas y Sistemas Computacionales, IPICYT.
rfemat@ipicty.edu.mx

* Universidad Autónoma de San Luis Potosí, México.
ducl@galia.fc.uaslp.mx, tamaz@cactus.iico.uaslp.mx.
Abstract

The design of reduced-order controllers under specific performance and structure requirements is dealt in this contribution. Two controllers are designed and compared. The first one was designed using $H_\infty$ theory whereas the latter one is designed departing from a parametric-optimization via a two-stage algorithm. The time spent by the designer using our second approach is largely reduced. An active suspension system is selected as a case study. The performance of both controllers is tested experimentally in the active suspension set-up. The experimental results show that the parametric-optimization controller practically meets the desired performance specifications. Meanwhile, the $H_\infty$ controller cannot accomplish the imposed constraints just in the low-frequency range.
1 Introduction

In any control design scheme, there are certain performance specifications that should be addressed for the suitable operation of the feedback loop. Such restrictions change accordingly with the problem in hand and with the configuration of the control loop. All of these could be translated by the designer into several time and frequency response restrictions. In the literature, there are several techniques available to address them. However, if the order of the resulting controllers is crucial for the real time implementation, the designer should look for the minimal order controller that could satisfy the performance restrictions. There are several techniques to overcome this situation. On the one hand, the controller could be synthesized by using the high-order model followed by a controller order reduction scheme always looking to preserve the performance of the closed-loop system, or model reduction can be applied to the high-order model first and then the controller is synthesized based on the low-order model. In addition to these strategies, parametric-optimization, which is an heuristic approach, can be carried out to design the controller parameters and structure. However, the resulting nonlinear optimization is non-convex and present multi-modal characteristics.

In recent years, evolutionary schemes have been extensively used to solve nonlinear constrained optimization problems where multiple local minima can restrict global convergence. Evolutionary schemes are inspired by the natural selection criteria where the stronger organisms are likely to survive after generations. Thus, a parallelism can be drawn with an optimization problem where the evolution period is considered as the optimization time and the most fitted organism in the population will represent the optimal solution. Two evolutionary schemes, evolution algorithms and genetic algorithms, are most commonly used. These algorithms present two main characteristics: a multi-directional (random) search and an information exchange among best solutions. These properties can generate new search directions in order to avoid local minima. However, they present some differences as: the structure of the solutions vector and the methodology to obtain new solutions for testing. Applications of evolution/genetic algorithms to control and signal processing have been reported in literature: digital IIR filter design [10], adaptive recursive filtering, active noise control, systems model reduction [6], weighting function design for $H_\infty$ loop-shaping [15], etc. Moreover, if an evolution algorithm is used in conjunction with a gradient-based approach (see [12] and references therein) a powerful two-stage optimization method is derived. Other approaches for controller design based on optimization of multiple-models can also be followed [13].

A benchmark problem related to active suspension is dealt in this contribution. Active suspension is an interesting system which has several technological applications. For instance, suspension of ground vehicles aims to support the vehicle body. An appropriate active suspension design must resolve the inherent tradeoffs between ride comfort, road holding quality and suspension travel [7]. The design of controllers for such systems involves all steps of a standard control design: modelling and identification, robust (or optimal) control design and experimental testing.

The problem of designing reduced-order controller under specific control performance requirements is addressed in this paper. Two designs are presented: $H_\infty$ synthesis and parametric-optimization. The optimization proposed for the parametric-optimization presents nonlinear characteristics and a discontinuous cost-function. However, a two-stage optimization is capable of reaching a satisfactory solution. The two-stage optimization proposed is a combination of an evolution algorithm and gradient-based approach. The designed controllers were experimentally tested and compared for an active suspension system. Some advantages of the controller synthesized by parametric-optimization over the $H_\infty$-based can be observed, as the designing time and complexity of the resulting controller.

The paper has been organized as follows. Section 2 illustrates two methods for the synthesis of reduced-order controllers. Thus, the synthesis procedure for the $H_\infty$-based and parametric-
optimization controllers are presented. The system identification, problem formulation and controller synthesis are detailed for the active suspension system in Section 3. Section 4 shows the controller implementation, and the contribution is closed with some concluding remarks in Section 5.

1.1 Notations and Definitions

All the systems treated in the paper are linear, time-invariant and discrete. Thus, a discrete transfer function $G(z)$ with state-space realization $(A, B, C, D)$ is represented by

$$G(z) = \frac{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}{z - A} = C(zI - A)^{-1}B + D$$

where $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$. Let $H_\infty$ be the subspace of analytic and bounded transfer functions inside the unit disk. Let $RH_\infty$ denote the space of all proper and real rational stable transfer functions.

Consider a feedback system described by the block diagram in Figure 1 where the generalized plant $G$ and the controller $K$ are assumed to be real-rational and proper. Let $G$ be partitioned accordingly as

$$G = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

and

$$K = \begin{bmatrix} A_k \\ C_k \\ D_k \end{bmatrix}$$

Then the transfer function from $w$ to $z$ is given by $T_{zw} = \mathcal{F}(G, K)$ where $\mathcal{F}(\cdot, \cdot)$ is called a lower linear fractional transformation.

![LFT representation](image)

Figure 1: LFT representation.

2 Synthesis of Reduced Order Controllers

Two procedures were explored and they are detailed in this section. The first procedure consists in the design of a controller based on the high-order model and in the reduction of the controller order. The main reason to adopt this strategy is that it is more appealing to use the model with the full-dynamics to synthesize the controller. In this way, the controller captures the full spectrum of interactions of the plant. Next, the high-order controller is reduced to the minimal order that
preserves the desired performance. The second procedure comprises a parametric optimization. This procedure allows to avoid the step of the reduction of the controller order.

The control problem approached in the paper is the SISO output disturbance rejection with control penalty for discrete time systems (see Figure 2). Thus, the objective of the controller is to attenuate the effect of the output disturbance, \( p \), in the system output, \( y \), while keeping the control signal, \( u \), limited. These control objectives can be posed as frequency-domain restrictions on the output and input sensitivity functions: \( S_{yp} = 1/1 + KP \) and \( S_{up} = -K/1 + KP \). Consequently, \( S_{yp} \) and \( S_{up} \) will represent frequency domain templates for the desired closed-loop performance. Any other specific control problem can be approached with the same techniques that are going to be presented in this paper. Moreover, extensions to MIMO problems can be easily deduced.

![Figure 2: Block Diagram for Disturbance Rejection Problem.](image)

**2.1 \( H_\infty \) Design**

The design of reduced-order controllers under this framework is presented. The controller synthesis involves a minimization process of some \( \| \cdot \|_\infty \) of the closed-loop system. Thus, a controller that minimizes this index, but also keeps internal stability is pursued [16]. This norm is appealing to controller design where the closed-loop specifications are expressed in terms of frequency response performance. Now, the first step in the synthesis procedure is to identify the signals that are to be minimized and their corresponding performance weights. The control problem in hand is the output disturbance rejection with control penalty, so the closed-loop scheme of Figure 2 is followed. The output and control performance weights, \( W_y \) and \( W_u \), are design parameters during the synthesis procedure. Therefore, the controller looks to minimize the gain from the output disturbance \( w = p \) to the weighted signals \( z_1 = W_y y \) and \( z_2 = W_u u \), while maintaining closed-loop stability.

Next, the closed-loop system is put into the LFT framework, see Figure 1. The generalized plant \( G \) is then written for this problem as

\[
G = \begin{bmatrix}
W_y & W_yP \\
0 & W_u \\
-I & -P
\end{bmatrix}
\]

(4)

Thus, the controller can now be synthesized using the procedure presented on [16], which is based on the solution of two Riccati equations. Note that the resulting controller has the same order of the generalized plant \( G \). Consequently, if \( G \) is high-order, the controller \( K \) will be also high-order. However, model-reduction methods based on the \( \| \cdot \|_\infty \) norm can be applied: Balanced truncation, Weighted balance truncation, or Hankel norm approximation [16]. Finally, the performance with the reduced-order \( \hat{K} \) controller must be checked. The main steps are summarized below.
1. Identify the control problem in hand: input signals \( w \) and weighted outputs \( z \).

2. Obtain the generalized plant \( G \) in the LFT framework,

3. Perform a controller order reduction to obtain \( \hat{K} \) by the Balanced truncation, Weighted balance truncation or Hankel norm approximation,

4. Check the performance with the reduced controller \( \hat{K} \).

Note that if hard constraints are imposed into the closed-loop frequency response, an iterative procedure is necessary in order to find the correct weights that satisfy the closed-loop constraints.

### 2.2 Parametric Optimization

A discrete-time controller prototype is proposed with the following structure

\[
K(z) = k_0 \frac{(z + a_1)(z + a_2)(z^2 + 2a_3a_4z + a_4^2)(z^2 + 2a_5a_6z + a_6^2)}{(z + b_1)(z + b_2)(z^2 + 2b_3b_4z + b_4^2)(z^2 + 2b_5b_6z + b_6^2)}
\]  (5)

More terms could be introduced into the controller prototype, but at the expense of increasing the number of parameters in the optimization and consequently the complexity of the optimization itself. For simplicity, it was decided to limit the controller to two first and two second order terms in the numerator and denominator. In the first order terms, the parameters \((a_1, a_2)\) are related to real zeros and \((b_1, b_2)\) to real poles of the controller. In the second order terms, one parameter is related to the damping, \((a_3, a_5, b_3, b_5)\), and the other to the natural frequency, \((a_4, a_6, b_4, b_6)\).

The overall gain of the controller is fixed by \(k_0\). The controller is assumed to be stable, but can be non-minimum phase. Consequently, in order to enforce stability, the roots of the denominator must be inside of the unit circle. So, the interval of variation can be set for the denominator terms

\[
-1 < b_i < 1 \quad i = 1, 2, 3, 5 \quad (6)
\]

\[
0 \leq b_i < 1 \quad i = 4, 6 \quad (7)
\]

On the other hand, there are no restrictions on the zeros of the controller, the following intervals were chosen arbitrarily

\[
-3 < a_i < 3 \quad i = 1, 2 \quad (8)
\]

\[
-1 < a_i < 1 \quad i = 3, 5 \quad (9)
\]

\[
0 \leq a_i < 3 \quad i = 4, 6 \quad (10)
\]

The variation of the gain, \(k_0\), can be set to any interval, except if there is a saturation limitation. In that case, a maximum value that could keep the control signal inside the linear part of the saturation must be set. Consequently, a small value must be set first and it could be increased according with simulation. Alternatively, in the case of time-domain constraints, they might also be incorporated in the optimization [13]. However, this could make the optimization scheme very difficult to solve for a practical solution. Nevertheless, this is another idea that can be explored in future work.

The structure of the controller is not fixed in the algorithm. Hence the extra parameters \(c_i\) (structure parameters) with \(i = 1, \ldots, 8\) are included in the optimization to activate or omit each term in the transfer function of \(K(z)\):

\[
c_1 \rightarrow (z + a_1); \quad c_2 \rightarrow (z + a_2); \quad c_3 \rightarrow (z + 2a_3a_4 + a_4^2); \quad c_4 \rightarrow (z + 2a_5a_6 + a_6^2); \quad c_5 \rightarrow (z + b_1); \quad c_6 \rightarrow (z + b_2); \quad c_7 \rightarrow (z + 2b_3b_4 + b_4^2) \quad \text{and} \quad c_8 \rightarrow (z + 2b_5b_6 + b_6^2).
\]
Note that the optimization under the above criteria is not an easy task and most certainly presents a challenge in terms of frequency templates. The performance restrictions are assumed to be given in terms of the number of frequencies \( \omega_i \in [0, \pi] \). Note that first of all, the controller must always guarantee closed-loop stability. It is important to point out that, since \( K \) is stable, the stability of the transfer function \( T_{up} = -K/(1 + KP) \) is implied by the stability of \( T_{yp} = 1/(1 + KP) \). Therefore, the cost function \( J \) in eq. (11) was proposed

\[
J = \begin{cases} 
M \exp |\lambda_{max}(T_{yp})| & T_{yp} \notin \mathcal{RH}_\infty \\
\sum_i [\Delta^u_i + \Delta^y_i] & T_{yp} \in \mathcal{RH}_\infty \& \sum_i [\Delta^u_i + \Delta^y_i] \neq 0 \\
\sum_i [T_{up}(e^{j\omega_i})] - |S_{up}(e^{j\omega_i})| + |T_{yp}(e^{j\omega_i})| - |S_{yp}(e^{j\omega_i})| & T_{yp} \in \mathcal{RH}_\infty \& \sum_i [\Delta^u_i + \Delta^y_i] = 0 
\end{cases}
\]

(11)

where \( |\lambda_{max}(\cdot)| \) denotes the maximum absolute value for the poles of the given transfer function, \( M \) is a large positive number (on the order of \( 1 \times 10^4 \)), and

\[
\Delta^u_i = \begin{cases} 
|T_{up}(e^{j\omega_i})| - |S_{up}(e^{j\omega_i})| & \text{if } |T_{up}(e^{j\omega_i})| - |S_{up}(e^{j\omega_i})| > 0 \\
0 & \text{otherwise} 
\end{cases}
\]

(12)

\[
\Delta^y_i = \begin{cases} 
|T_{yp}(e^{j\omega_i})| - |S_{yp}(e^{j\omega_i})| & \text{if } |T_{yp}(e^{j\omega_i})| - |S_{yp}(e^{j\omega_i})| > 0 \\
0 & \text{otherwise} 
\end{cases}
\]

(13)

A brief explanation of the cost function \( J \) is:

- \( M \exp |\lambda_{max}(T_{yp})| \): this term is active if the closed-loop is unstable and looks to move the unstable poles back inside the unit circle. Since \( M \) is a large number, this evaluation will always return a much larger positive value than \( M \).

- \( \sum_i [\Delta^u_i + \Delta^y_i] \): this term is active if the closed-loop is stable and it is always positive in the case that the closed-loop violates the frequency restrictions (i.e., \( |T_{up}(e^{j\omega_i})| \geq |S_{up}(e^{j\omega_i})| \) or \( |T_{yp}(e^{j\omega_i})| \geq |S_{yp}(e^{j\omega_i})| \) for some \( \omega_i \)). Thus, its minimization looks to reduce the gap with the restrictions.

- \( \sum_i [T_{up}(e^{j\omega_i})] - |S_{up}(e^{j\omega_i})| + |T_{yp}(e^{j\omega_i})| - |S_{yp}(e^{j\omega_i})| \): finally, it is active if the performance is already satisfied and looks to maximize this gap. Therefore, the closed-loop response is below the frequency bound, and as a result this is term is always negative (i.e., \( |S_{up}(e^{j\omega_i})| \geq |T_{up}(e^{j\omega_i})| \) and \( |S_{yp}(e^{j\omega_i})| \geq |T_{yp}(e^{j\omega_i})| \forall \omega_i \)).

As a result, the cost function \( J \) in (11) was defined such that, as the conditions are being satisfied its value was decreasing constantly. The controller is now designed with the criteria \( \min_X J(X) \), where the controller is constructed from \( X \) according to the pole-zero structure of \( K(z) \) in (5). Note that the optimization under the above criteria is not an easy task and most certainly presents a multi-modal characteristic. Moreover, it combines two kinds of parameters: integer \( c_i \)'s and real \( a_i \)'s and \( b_i \)'s. In order to overcome this problem, the parameters \( c_i \)'s are let to vary just in the interval \([0, 1]\), and during the optimization a value for \( c_i \) below 0.5 will represent to omit the corresponding term in the controller and above 0.5 will include it. Consequently, all the parameters in the optimization are real parameters with a fixed interval of variation.

Note that the cost function (11) does not include a penalty in the terms \( c_i \)'s, that are related to the number of terms in the denominator and numerator (complexity) of the controller. Nevertheless,
it is intuitive that the optimal solution will always have the largest complexity. However, it might
not be reached in practice. In other words, it is possible that if the cost function is extremely
complex, that the optimal solution could not be achieved. Therefore, it was decided to include the
terms $c_i's$ in the parameters vector. In this way, during the optimization different structures of the
controller could be tested with less or higher orders.

If after the first optimization the resulting controller $K_0$ cannot satisfy the constraints, it is
proposed to perform a second iteration. Therefore, the initial controller is set $K_0$ and a second
transfer function $K_1$ is optimized now, so the complete controller is composed of two transfer
functions in series, $K = K_0K_1$. Note that during each iteration of the optimization, the structure
of each controller is not fixed. Hence, the controllers $K_0$ and $K_1$ do not necessarily have the same
orders. Also, another feasible approach is to define a fixed low-order controller $K_0$, and if the
constraints are not satisfied after the first iteration, then a second iteration is carried out for $K_1$
where the complete controller is given by $K = K_1K_0$.

### 2.2.1 Optimization Scheme

In general, nonlinear constrained optimization problems are difficult to solve with standard gradient-
simulated annealing, tabu search, etc., have been suggested to deal with this kind of problems
successfully. In the optimization presented in (11), a nonlinear and discontinuous cost function is
proposed for the controller synthesis. Hence, through the minimization problem posed:

$$\min_X J(X) \quad (14)$$

the controller parameters and structure are pursued, and since the optimization has to be restricted
to internally stabilizing controllers, the complexity of the cost function is raised. Therefore, it is not
feasible to solve the optimization by a gradient-based approach. However, an evolution algorithm
presents nice characteristics that made possible to reach a satisfactory solution, as it will be seen in
the next sections. Therefore, the evolution algorithm was applied first to perform a global search
in the parameters space and find a minimum solution. Next, a local search was conducted to
obtain the optimal solution. It is assumed that the minimum solution obtained by the evolution
algorithm is not close to the discontinuous points of the cost function. Consequently, the gradient-
based approach should not be affected in the local search. Thus, the best solution coming from the
evolution algorithm was used as a starting point for the gradient-based search. A brief description
of the evolution algorithm used in the paper is presented next. The gradient-based optimization
was carried out by using the Optimization Toolbox [4] of MATLAB (R).

### 2.2.2 Evolution Algorithm

Consider a function that has to be optimized with $m$ inputs and one output. The output of this
function is referred as its fitness. The idea is now to adjust the input parameters in order to find an
optimum in the fitness. One combination of $m$ input parameters is called an individual. A group
of $n$ individuals is a population. The idea is to start with a randomly selected initial population
(mutation), creating a group of children out of their parents. The fitness of the children is now
evaluated and compared with their parent’s fitness, and the best of both are selected to be the next
generation of parents. This procedure will go on until an optimum is found, or a given termination
criterion is fulfilled.

Following these ideas, the evolution algorithm EVAOCP (Evolution Algorithm for Optimization
of Continuous Parameters) is proposed for the optimization process
1. Initialize parameters of the evolution algorithm.

2. Select initial population (this can be a random selection, a guess or a result of a previous experiment).

3. Check if the conditions for the termination of the algorithm are satisfied: optimality, max. number of iterations or no-progress
   - YES : Set the best values obtained during the optimization process.
   - NO : Continue with the Evolution Algorithm.

4. Adjust step (mutation range) according with the progress achieved.

5. Create children from parents set.
   - Check mutation factor.
   - Determine new step.
   - Generate the children by adding a random perturbation of variance 'step' to the parents.
   - Check that the children satisfy the parameters bounds.

6. Evaluate the children fitness.

7. Compare the children and parents performance, and keep the best of both.

8. Compute the progress velocity: the number of children that have better performance than the parents. In order to expand the search if no progress is detected.

9. Select the best solutions to judge optimality.

10. Go back to 3.

3 System identification, Problem Formulation and Controllers Synthesis

An active suspension system is chosen as a case of study in this paper (see Figure 3). This system was chosen as a benchmark problem, and performance objectives were imposed into the frequency response of the input and output sensitivity transfer functions of the closed-loop system (i.e., $S_{up}$ and $S_{yp}$ in Figure 9). Moreover, data about the active suspension response was available for analysis. The principal parts of the system are: (i) an elastomer cone that encloses the main chamber filled with silicon oil, (ii) an inertia chamber enclosed with a flexible membrane, (iii) a piston that is fixed on a DC motor and (iv) an orifice that allows oil flow between chambers. There are two input signals and one measured output in the system. One input $u_p$ corresponds to the driving signal of a shaker that represents an input disturbance to the active suspension system (primary path). The control input $u_s$ drives the position of the piston via the DC motor (secondary path). The output $y$ is the measured voltage corresponding to the residual force.
3.1 Identification of the active suspension system

Two models were identified in order to obtain the system realization (1) of the active suspension system, corresponding to the primary and secondary path, respectively. The model for the primary path is used for simulation purposes and the model for the secondary path is used for control design purposes (see Figure 4). The data used for the model identification was provided as part of the characteristics of the benchmark problem. To identify the primary path \((u_p \rightarrow y)\) model the shaker is excited by a pseudo-random binary signal (PRBS) using a 10-bit shift register with a clock frequency of \(F_s/2\), where \(F_s=800\) Hz is the sampling frequency for data acquisition. The order of the model identified is \(\text{num}_p = 15\) and \(\text{den}_p = 16\) (\(\text{num}_p\)=primary numerator order and \(\text{den}_p\)=primary denominator order). This model holds the two principal vibration modes of the real system, which are around 31.5 Hz and 160 Hz. For the identification of the secondary path model \((u_s \rightarrow y)\), the input is excited by the same PRBS with a clock frequency of \(F_s\). In this case \(\text{num}_s = 13\) and \(\text{den}_s = 14\). Both models are stable and strictly proper. The identification procedure in this contribution was made via the subspace method to find the state-space realization \((A, B, C, D)\) [14]. The model identified by the subspace method was obtained under above sampling conditions and compared with the one provided by Laboratoire d’Automatique de Grenoble. However, the fitting is also precise for slightly smaller orders.

The identification procedure was carried out by using the Matlab’s System Identification Toolbox\(^{TM}\) [9]. Figure 5 shows the spectral density of the identified models (15) and (16) for the primary and secondary paths, respectively. Note that identified model (solid lines) retains all frequency modes of the active suspension response (dashed line). In this manner, the transfer function of the identified model for the primary path is given by...
Figure 4: Block Diagram of the Active Suspension System.

\[
G_p(z) = \frac{-3.801 \times 10^{-2}(z^2 - 2.146z + 1.153)(z^2 + 1.599z + 0.954)(z^2 + 0.739z + 0.949)}{(z^2 + 1.73z + 0.086)(z^2 - 1.908z + 0.971)(z^2 + 0.339z + 0.182)(z^2 + 1.085z + 0.926)} \\
\frac{(z^2 - 0.11z + 0.963)(z^2 - 0.425z + 10.38)(z + 3.026)(z + 1.335)(z - 2.349)}{(z^2 - 0.104z + 0.787)(z^2 - 0.539z + 0.914)(z^2 + 0.625z + 0.959)}
\]

(15)

while the corresponding identified model for the secondary path becomes

\[
G_s(z) = \frac{7.961 \times 10^{-5}(z^2 + 13.66z + 91.89)(z^2 - 0.098z + 0.923)(z^2 + 0.732z + 0.95)(z - 12.6)}{(z + 0.716)(z - 0.913)(z^2 + 0.636z + 0.967)(z^2 - 0.496z + 0.894)(z^2 + 1.094z + 0.936)} \\
\frac{(z^2 - 0.567z + 0.526)(z^2 + 0.622z + 0.434)(z^2 + 1.797z + 1.058)(z - 1)(z + 1.56)}{(z^2 + 0.272z + 0.613)(z^2 - 0.5787z + 0.4)(z^2 + 1.68z + 0.786)(z^2 - 1.904z + 0.965)}
\]

(16)

Now, it is intended to use linear theory to design the control system. Consequently, it is important to know if the response of the system to the disturbances is linear with respect to the control signal. In other words, if both input signals are active in the system, is the output just the sum of the individual responses? This information could give an indication of the performance that could be expected with a linear controller. In order to answer this question, a simple experimental test was performed [1]. The inputs \(u_s\) and \(u_p\) were induced simultaneously and the output data was recorded. Figure 6 shows the spectral density of both simultaneous and separated experiments. Note that the responses from simultaneous and separated inputs are similar. Hence one can assume that input-output response obeys the superposition principle. Thus, the input-output model is linear.

3.2 Control objective

For the active suspension system, the control objectives were defined in the benchmark problem. Thus, the control goal can be expressed as to compute a linear discrete-time controller which minimizes the residual force around the first and second vibration modes of the secondary path model, and to distribute the amplification of the disturbances over the higher frequencies. In addition, the controller gain should be equal to zero at the frequency of \(\frac{F_s}{2}\), and consequently the
Figure 5: Comparison Between the Spectral Density of the Time-Series (dashed) and Response of the Identified Model (solid) for the Primary (top) and Secondary (bottom) Paths.

The term $1 + z^{-1}$ must be incorporated into the controller structure. In addition, the control signal should not exceed -0.2 and 0.2 (saturation constraint). Closed-loop performance constraints were imposed in terms of the sensitivity functions $S_{up}$ and $S_{yp}$ (see Figure 9). The constraints on the output and input sensitivity functions were provided by Laboratoire d’Automatique de Grenoble.

Now, the controller complexity will be evaluated according to the following criterion:

$$C = (n_r + n_s)F_s$$

(17)

where $n_r$ and $n_s$ are the number of parameters (polynomial order+1) in the controller numerator and denominator respectively. From the spectral density of the experimental data (see Figure 5), one can observe that the main requirements in frequency response are approximately located at 31 and 160 Hz. From this fact, the sampling rate $F_s$ was fixed at 800 Hz, which is far enough from above critical frequencies.

3.3 Synthesis of the $H_\infty$-based controller

The steps outlined in Section 2.1 were followed to design an $H_\infty$-based controller for the benchmark problem. As it was discussed above, the control problem can be stated as an output disturbance rejection problem with control restrictions. The weighting functions $W_y$ and $W_u$ were selected in order to stress the frequency requirements in continuous-time version of the controller and then converted to discrete-time by 'bilinear' transformation. In this manner, one has that

$$W_y(z) = 0.18\frac{(z - 0.854)^3(z - 0.777)(z - 0.699)(z - 0.397)(z - 0.073)}{(z - 0.999)(z^2 - 1.788z + 0.823)(z^2 - 1.912z + 0.971)(z^2 - 0.463z + 0.777)}$$

(18)
Figure 6: Spectral Density of Output. Comparison of the Simultaneously Disturbed Open-loop Experiments and the Sum of the Individually Disturbed Experiments.

\begin{equation}
W_u(z) = 4.01 \frac{(z^2 - 0.267z + 0.125)}{(z + 0.157)(z + 0.0817)}
\end{equation}

The frequency response of these weighting functions is presented in Figure 7. The output weighting \( W_y \) presents two resonances at \( \approx 31 \text{ Hz} \) and \( 160 \text{ Hz} \). Therefore, it is stressed the importance of attenuating those frequencies. In regard to the weight in the control signal \( W_u \), two ideas are considered: (1) to maintain the control signal inside interval \([-0.2,0.2]\) and (2) to ensure that the resulting controller is stable.

The synthesis of controllers departing from an \( H_\infty \) design might be unstable \cite{2} even if the internal-stability of the closed-loop is guaranteed. This problem is present especially in the disturbance attenuation problem if the plant is non-minimum phase as in the case of the active suspension problem. Consequently, the \( H_\infty \)-based controller could be unstable, which is obviously undesirable in practical implementations. One heuristic way to avoid this, it is to increase the control penalty until the controller is finally stable. This problem was present during the \( H_\infty \) synthesis and it was observed that if the gain of \( W_u \) was high enough the resulting controller could be stable.

On the other hand, since the controller is required to include the dynamics of \( K_{im}(z) \), therefore this term is appended to the plant during the design. The secondary path model, which corresponds to the plant \( P \) in the design, is of \( 14^{th} \) order, the weights \( W_y \) and \( W_u \) are of \( 7^{th} \) and \( 2^{nd} \), order respectively, and finally the internal model is of \( 1^{st} \) order. Therefore, the generalized plant \( G \) in the LFT framework becomes

\begin{equation}
G = \begin{bmatrix}
W_y & W_yPK_{im} \\
0 & W_u \\
-I & -PK_{im}
\end{bmatrix}
\end{equation}

In this way, \( G \) is a transfer matrix of \( 24^{th} \) order. The controller was computed using the LMI Control Toolbox of MATLAB \(^{(R)} \) by the command \textit{dhinfnc}. As a result, a \( 24^{th} \) order controller \( \hat{K}(z) \) was obtained. The controller \( \hat{K}(z) \) was reduced using Balance Truncation Method \cite{16} to a \( 7^{th} \) order controller \( \tilde{K}(z) \), which was the minimum order under which the performance was still preserved. The frequency response of the full-order and reduced controllers are presented in Figure 8. Note that the frequency mismatches between the full-order and reduced controllers did not
affected the performance of the latter one. Alternatively, other model-reduction methods could be applied to compare the performance obtained, however this is out of the scope of the paper.

In this manner, the final controller is constructed \( K(z) = \hat{K}(z)K_{im} \), whose transfer function is given by

\[
K(z) = 4.906 \times 10^{-5} \frac{(z + 18.57)(z - 0.983)(z - 0.867)(z + 1)(z^2 - 1.903z + 0.967)}{z(z - 0.999)(z^2 - 1.792z + 0.847)(z^2 - 1.906z + 0.965)}
\]
\[
\frac{(z^2 - 0.996z + 8.475)}{(z^2 - 0.817z + 0.563)}
\]

In addition, the final closed-loop performance for the output and input sensitivity functions \( T_{yp} \) and \( T_{up} \) is shown in Figure 9. Note that the closed-loop restrictions are not completely satisfied. The restrictions for \( T_{up} \) are satisfied but not for \( T_{yp} \). The frequency gain for \( |T_{yp}| \) is raised above the restriction \( S_{yp} \) in the low-frequency. This means that during the synthesis procedure there could not be found a pair a weights \( W_y \) and \( W_u \) that could yield a controller that satisfied all the restrictions in a strict sense. The main restriction was the control saturation and the controller stability since a controller achieving all the requirements needed a control signal above the boundary of \(-0.2\) and \(0.2\) and an unstable controller. It is important to mention that this controller design technique was not easy to carry out. The design involved a time consuming and exhausting search for the appropriate weights, and even in this case the objectives could not be accurately fulfilled. In Table 1, the final order and performance index \( C \) are shown for the \( H_\infty \) controller.
Figure 8: Frequency responses of (dash-dot) Full-Order Controller $\tilde{K}(z)$ and (solid) Reduced Controller $\hat{K}(z)$ for $H_\infty$ Synthesis.

<table>
<thead>
<tr>
<th>Controller order</th>
<th>Parametric Optim. $5^{th}$</th>
<th>$H_\infty$-based $8^{th}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of Criterion $C$</td>
<td>9600</td>
<td>12800</td>
</tr>
</tbody>
</table>

Table 1: Characteristics of Final Controllers.

3.4 Controller from Parametric Optimization

The design procedure to obtain a controller based on a parametric optimization is presented now. The design steps and justification were introduced in Section 2.2. This design involves an optimization scheme for the controller parameters in order to satisfy some performance requirements. The optimization is computed by a two-stage algorithm where a global search is performed by an evolution algorithm, followed by a local-search computed with a gradient-based algorithm.

This design presents a nonlinear optimization which is certainly non-convex and presents a discontinuous behavior in the cost function. However, the two-stage optimization proposed was capable to solve efficiently this problem. Moreover, once the performance restrictions are specified and the interval of variation for the parameters are set, the algorithm looks automatically for the best locations of zeros and poles for the controller while keeping internal stability. Thus, the work for the designer is largely reduced since he/she is not involved with any extra iteration, like in the weights for the $H_\infty$ design. Initially, the gain of the controller was set to vary in the interval $[0.0001 < k_0 < 0.001]$. These small values were chosen since the controller output had to be restricted to the interval $[-0.2, 0.2]$, and varying the gain in that interval satisfied approximately that restriction. If after the first optimization the controller, say $K_0(z)$, cannot satisfy the closed-loop restrictions, then a second iteration is allowed for $K_1(z)$ where the controller is now constructed by $K(z) = K_1(z)K_0(z)$. Hence, $K_0(z)$ is fixed and the parameters of $K_1(z)$ are optimized now. Just one change is made in the optimization, since the gain of $K_1(z)$ is varied only in the interval $[0.25 < k_0 < 1.25]$. Furthermore, since the controller must include the dynamics $(z + 1)/z$, this term is appended to the controller during the optimization. In the first iteration, the resulting controller was of $5^{th}$ order and in the second iteration just a gain adjustment was obtained. However, after this two iterations the controller could not satisfy completely the frequency restrictions. However,
important improvements were observed compared with the $H_\infty$ design. The final controller from parametric optimization is given by

$$K(z) = 3.325 \times 10^{-3} \frac{(z + 1.085)(z + 1)(z + .285)(z^2 - 0.716z + 1.155)}{z(z - 0.9511)(z + 0.046)(z^2 + 0.4822z + 0.789)}$$  \(22\)

In Figure 9, the closed-loop performances for the output and input sensitivity functions $T_{yp}$ and $T_{up}$ are shown. Note that two advantages of this controller are seen compared to the $H_\infty$ design: (a) the order of the controller (22) is smaller than $H_\infty$-based controller (21), and (b) the closed-loop performance is slightly improved.

### 4 Controller Implementation

For the experimental testing, the controllers were implemented on the active suspension system by Laboratoire d’Automatique de Grenoble, which is shown in Figure 3. The sampling frequency was already chosen at 800 Hz during the controller design. The performance of both the $H_\infty$-based and parametric-optimization controllers is shown in Figures 10 and 11. Note that experimental results agree with the predicted performance obtained in Figure 9. This is expected in part because the system is linear with respect to both inputs as tested previously (see Figure 6). Moreover, the control signal is practically within the linear part of the saturation $[-0.2, 0.2]$ (see Figure 10). As a result, there is no degradation of performance due to induced nonlinearities. Thus, the parametric-optimization controller practically meets the desired performance specifications. Meanwhile, the $H_\infty$ controller cannot accomplish the imposed constraints just in...
Figure 10: Experimental Performance in Time-domain.

Figure 11: Experimental Performance in Frequency-domain (dash-dot) \( \mathcal{H}_\infty \) Design and (solid) Parametric Optimization.
the low-frequency range for the output sensitivity function. However, note that both controllers present small orders: $5^{th}$ (parametric-opt.) and $8^{th}$ ($H_\infty$).

5 Conclusions

This contribution dealt with the problem of designing reduced-order controller under specific control performance requirements. Two designs were presented: $H_\infty$ synthesis and parametric-optimization. The optimization proposed for the parametric-optimization presents nonlinear characteristics and a discontinuous cost-function. However, the two-stage optimization suggested was capable of reaching a satisfactory solution. The designed controllers were experimentally tested and compared for an active suspension system. Some advantages of the controller synthesized by parametric-optimization over the $H_\infty$-based can be observed. First and according to Table 1, the order of the controller based on parametric-optimization is less than the order of the $H_\infty$ controller. In addition, the controller based on parametric-optimization was much easier to compute since there is no iterative process of selection of extra parameters as in the case of the $H_\infty$ controller, which requires to adjust continuously the frequency response of the weight functions $W_y$ and $W_u$ in order to satisfy the requirements. Thus, the time spent by the designer using the parametric-optimization approach is largely reduced. Nevertheless, the control goals have been practically achieved for both designed controllers, i.e., the residual force around first and second vibration modes of the primary path is minimized and the amplification of the disturbances over high frequencies is distributed. Only the $H_\infty$-based controller cannot accomplish partially the low-frequency bound for the output sensitivity function.

Acknowledgment

The authors would like to thank the anonymous reviewers for helping to improve the quality of the paper.

References


