ALGORITHMIC DESIGN FOR A ROBUST CONTROL BENCHMARK PROBLEM

Aldo Xavier Caetano de Almeida, xavier@ita.br
Alberto Adade Filho, adade@ita.br
Aeronautics Institute of Technology, 12228-900 – São José dos Campos, SP, Brazil

Abstract. *This paper will investigate the results obtained through an algorithmic design approach for a benchmark control problem composed by a flexible transmission system. The problem is presented with a set of specifications that should be met to assure the desired performance. The design program is a new version of a program for algorithmic control design that is being developed by the authors. This program allows the user to choose a robust control design method – for example, H-infinity, H-infinity with Loop Shaping or Mu-synthesis - as well as the setting of pre and post-compensators and weighting functions which are subject to algorithmic search for tuning or optimization. The advantages and disadvantages of this type of design procedure are pointed out throughout the work, as well as the specifications met. The results are used both as a benchmark problem investigation and as a validation for the computational environment developed for this design program.*

Keywords*: Robust Control, Algorithmic Design, Control Benchmark*

1. INTRODUCTION

Control is an important field of study, with applications on various engineering areas. The ongoing development of automatic systems requires constant research efforts in this area, with ever-increasing performance needs and constraints; those constraints could be related to profit gains or high-security demands, to name a few.

For complex problems, the design stage is a crucial part for the success of a controller. Careful attention must be paid to the process to be controlled, and sometimes control engineers will find that prior experience is one of the best assets to use when facing initial design choices.

Particularly, some control techniques demand repetitive calculus methods, in order to search for a controller that will attain satisfactory performance. This may consume a lot of time, and an 'initial guess' could be the only difference between minutes or hours of design. For robust control, "finding appropriate weighting functions is a crucial step in robust controller design and usually needs a few trials. For complex systems, significant efforts may be required" (Gu et al., 2005).

With the above assertive in mind, employing search algorithms during this stage is a way to relinquish control engineers of this arduous task. Therefore, the authors have been developing and testing a control system design software that utilizes the algorithmic approach to robust control design. This paper focus a part of this work, presenting a classic robust control benchmark as a validation problem, and discuss the results obtained through the usage of this new toolbox.

2. PLANT DESCRIPTION AND REQUIREMENTS

The plant used as model for this work is a flexible transmission system originally built at the Grenoble Automatic Control Laboratory, and it is considered as a benchmark control problem. Figure 1 demonstrates a schematic for the physical system. The following description is presented at (Landau *et al.*, 1995), and is reproduced here for reader convenience. It should be noted that the plant was identified as a discrete-time model. The transfer function is Eq. (1) represents the plant:

$$H(q^{-1}) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})} \tag{1}$$

where q^{-1} is the backward shift operator, *d* is the integer number of sampling periods contained in the plant pure time delay and:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_A} q^{-n_A} \tag{2}$$

$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_{nB} q^{-nB} \tag{3}$$

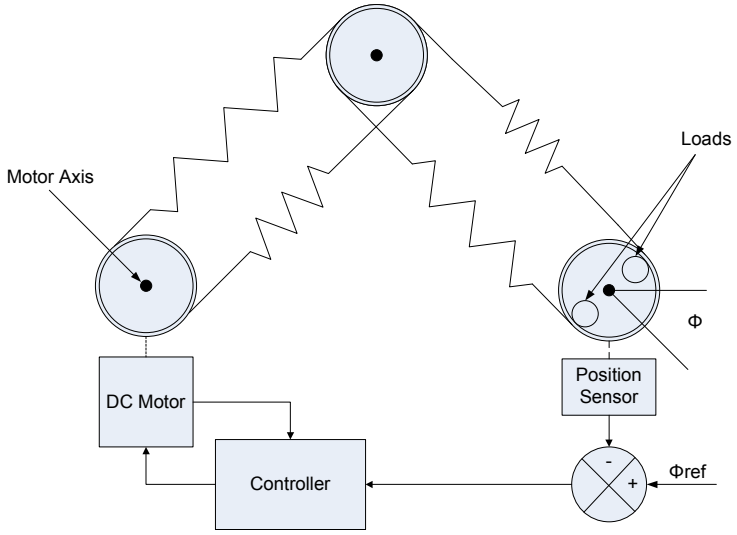

Figure 1. Schematic diagram of the flexible transmission.

These polynomials may change according to the loading level of the system, and in the original work they are presented for three different situations; first, the unloaded model is given by:

$$A(q^{-1}) = 1 - 1.41833q^{-1} + 1.58939q^{-2} - 1.31608q^{-3} + 0.88642q^{-4} \tag{4}$$

$$B(q^{-1}) = 0.28261q^{-1} + 0.50666q^{-2} \tag{5}$$

For the half-loaded model (1.8 *kg*), the polynomials are:

$$A(q^{-1}) = 1 - 1.99185q^{-1} + 2.20265q^{-2} - 1.84083q^{-3} + 0.89413q^{-4} \tag{6}$$

$$B(q^{-1}) = 0.10276q^{-1} + 0.18123q^{-2} \tag{7}$$

Finally, the full-loaded model (3.6 *kg*) is described by:

$$A(q^{-1}) = 1 - 2.09679q^{-1} + 2.31962q^{-2} - 1.93353q^{-3} + 0.87129q^{-4} \tag{8}$$

$$B(q^{-1}) = 0.06408q^{-1} + 0.10407q^{-2} \tag{9}$$

For all the presented models, the value of *d* equals 2.

Frequency response for the three systems is presented at Fig. 2, which clearly demonstrates the presence of two distinctive vibration modes and also the influence of the load over their position.

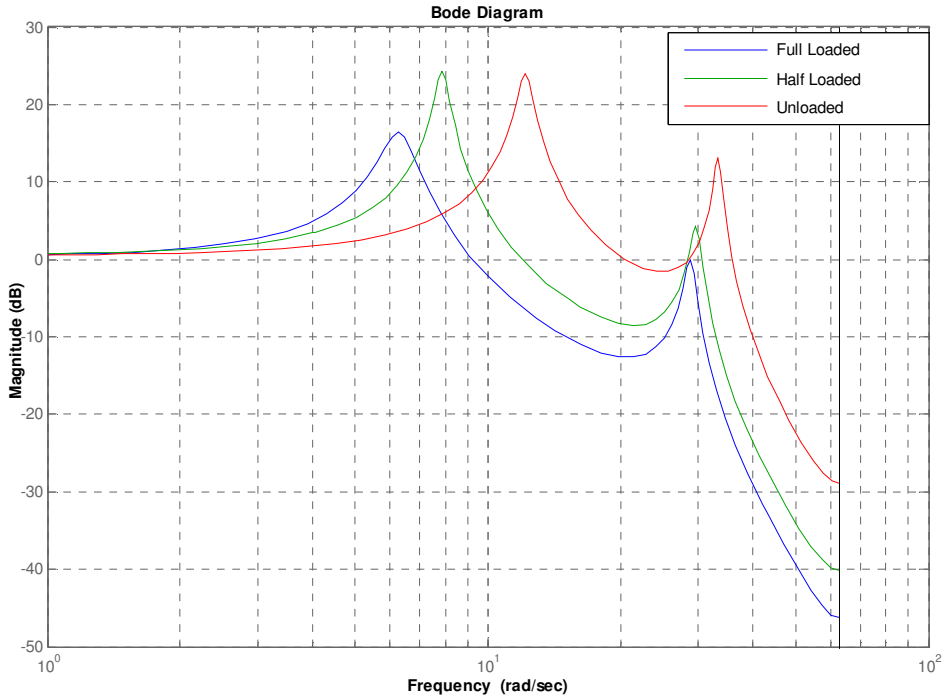

Figure 2. Bode Diagram for the flexible transmission with different loads.

For this paper, the choice was made to work in the continuous-time domain, and as such, all three plant models were converted using MATLAB® Control System Toolbox, which was also used to generate Fig. 2. Analysis of the Bode Magnitude and Step Response graphics indicates that the behavior of the three systems did not change significantly after they were converted to continuous-time models.

The work by Landau *et al.* (1995) presents the following eight specifications, which shall be used here as well:

1. A rise time (0 – 90% of the final value) of less than 1 s for a step change in the reference input;
2. Overshoot less than 10 % for a step change in the reference input;
3. Perfect rejection of constant disturbances (using integral action);
4. Rejection of a step output disturbance filtered by 1/*A* within 1.2 *s* (for 90 % rejection of the measured peak value);
5. Disturbance attenuation in the low frequency band from 0 to 0.2 Hz (Sensitivity function $\leq$ 1 in this frequency range)*;*
6. A maximum value of less than 6 *dB* of the output sensitivity function (modulus margin greater than 0.5);
7. A delay margin of at least 40 ms (80% of sampling period);
8. A maximum value of less than 10 dB of the input sensitivity function in the frequency range 8 – 10 Hz.

In our approach, the above specifications compose the functions for the *method of inequalities*; they create regions on the search space for the algorithmic project, and the optimal or feasible solution is located in regions where there are intersections among them.

3. BASIC CONCEPTS

3.1. The Method of Inequalities

The Method of Inequalities (MoI) is a multi-objective formulation for a design problem; it differs from optimization as it aims to achieve a satisfactory performance rather than an optimal solution. The process is interactive, as it requires intervention and supervision from the user (Whidborne and Liu, 1993).

The MoI was first proposed by (Zakian and Al-Naib, 1973), and is a computer-aided project in which the goal is to find a vector $p = \{ p_j , j = 1,\ldots,q\}$ of project parameters (such as parameters associated to a control law), satisfying the set of inequalities presented on Eq. (10):

$$\phi_i(\mathrm{p}) \leq \mathrm{C_i} \qquad i = 1,\ldots,m \tag{10}$$

The functions described in ϕ_i are indicators of system performance, and are considered as constraints for the problem. These indicators could be the rise time, settling time, overshoot percentage, etc., and the vector C_i contains real numbers with the accepted values for those parameters for a given problem.

The inequalities on Eq. (10) define a set of points on a space $\Re^q$, and each set has a frontier defined by Eq. (11):

$$\phi_i(p) = C_i \tag{11}$$

A point in this space is considered a solution for the problem if and only if it is situated in an intersection of all regions, thus implying that this point satisfies simultaneously all of the inequalities presented on Eq. (10). This is represented mathematically by Eq. (12):

$$v = \bigcap_{i=1}^{m} v_i \tag{12}$$

Zakian and Al-Naib (1973) also proposed a numerical search algorithm for the solution of this method, known as the Moving Boundaries Process, which is presented on the following section.

3.2. The Moving Boundaries Process

The *moving boundaries process* proceeds from an arbitrary initial point in the search space to an admissible point in an iterative way. Each iteration tries to improve the proximity to the solution space initially proposed by the problem, by gradually narrowing the boundaries. A formal explanation of the algorithm follows (Whidborne and Liu, 1993).

Let p^k be the value of p in the k-th move. Also, let S_i^k be the set formed by the inequality in Eq.(13):

$$\phi_i(p) \le \phi_i(p^k) \tag{13}$$

As mentioned in section 3.1, the boundary for the above case is defined by the equality between the terms of $\phi_i(p)$ and $\phi_i(p^k)$. The algorithm proceeds to a new trial point $\tilde{p}^k$. If for every i=1,2,..,n the boundary defined by $\phi_i(p) = \phi_i(p^k)$ is closer or no further away from the boundary S_i^k, then the point $\tilde{p}^k$ is accepted and becomes p^{k+1}.

In other words, the algorithm begins with an arbitrary point in the search space, whose boundaries are still far from the solution space. Each trial point generated during the iterations is tested to see if it gets closer to the solution space defined by the problem. In case of success, a new boundary is defined by this point, and the algorithm proceeds to a new trial. If the point does not move towards the solution, it is discarded, and a new trial point is generated based on the last successful point.

3.3. H-infinity controller design

H-infinity optimization was popularized as a control design method by the work of Zames (1981), and is being continuously developed for the last thirty years; it is considered an effective robust design method for linear, time invariant control systems (Gu et al., 2005). Skogestad and Postlethwaite (2005) point that the motivation for H-infinity optimization as a control design method are the shortcomings found on LQG (Linear Quadratic Gaussian) control theory. For further comparison between H-infinity design method and other techniques such as LQG and classical control design, the reader is referred to (Grimble, 1994).

Consider the general formulation for a control problem, represented by Fig. 3:

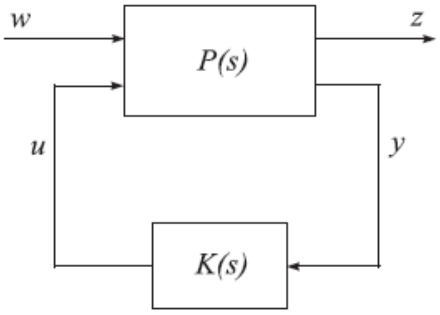

Figure 3. General control configuration.

The variables in the figure 3 are $P(s)$, the *generalized plant* (which includes the plant G and the weighting functions selected by the designer), the controller $K(s)$, the exogenous signals (such as reference and external disturbances) 'w', the error signals 'z', the measured variables 'y' and the control variables 'u'. Partitioning the generalized plant $P(s)$ as in Eq. (14):

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \tag{14}$$

a state-space realization of P(s) is given by:

$$P(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \tag{15}$$

The closed-loop transfer function from 'w' to 'z' is thus calculated by Eq. (16):

$$\mathrm{z} = \Im(\mathrm{P}, \mathrm{K})\mathrm{w} \tag{16}$$

$$\Im(\mathrm{P}, \mathrm{K}) = \mathrm{P}_{11} + \mathrm{P}_{12}\mathrm{K}(\mathrm{I} - \mathrm{P}_{22}\mathrm{K})^{-1}\mathrm{P}_{21} \tag{17}$$

Equation (17) represents a Linear Fractional Transformation between $P(s)$ and $K(s)$. The H-infinity control problem is to find all stabilizing controllers K which minimize Eq. (18):

$$\| \Im(\mathrm{P}, \mathrm{K}) \|_\infty = \max_\omega \bar{\sigma}(\Im(\mathrm{P}, \mathrm{K})(\mathrm{j}\omega)) \tag{18}$$

where $\bar{\sigma}(\cdot)$ denotes maximum singular value. As stated in (Skogestad and Postlethwaite, 2005), usually it is not necessary to find an optimal controller for the H-infinity problem, being computationally and theoretically simpler to design a suboptimal one. Therefore, let $\gamma \min$ be the minimum value of the H-infinity norm presented at Eq. 18 over all stabilizing controllers K; the H-infinity suboptimal control problem is, for a given $\gamma > \gamma \min$, find all stabilizing controllers K such that:

$$\| \Im(\mathrm{P}, \mathrm{K}) \|_\infty < \gamma \tag{19}$$

The most general algorithm utilized to solve such problem is the one proposed by Glover and Doyle (1988) and Doyle et al. (1989), since the problem satisfies a set of conditions (Skogestad and Postlethwaite, 2005; Gu et al., 2005; Maciejowski, 1989). The functions associated with H-infinity optimization found on the MATLAB® Robust Control Toolbox (and used in this work) utilize the Glover-Doyle algorithm, and as such, are sensible to such conditions; failure to run any of these functions may indicate that the control problem must be reformulated in order to attend the necessary assumptions.

4. CASE STUDY AND PROGRAM INTERFACE

The control design for the benchmark plant was performed utilizing a computational program developed by the authors. The current version of this program makes use of the aforementioned numerical search algorithm in order to find a possible solution for the problem. The idea is to compare this solution with the ones presented in (Landau *et al*., 1995), pointing out advantages and disadvantages found during the design process.

The program is entirely developed in MATLAB®, thus forming a toolbox compatible with the program. A user-friendly Graphical User Interface (GUI) was created to enhance the comprehension of new users and also to avoid the necessity of repeating command lines.

Aside from completely defining the plant, the user also has the option to define unstructured uncertainties, choose from five different types of robust control design methods (H-infinity, H-infinity with two degrees of freedom (DOF), Loop-Shaping Design Procedure (LSDP), LSDP 2-DOF and new to this version of the program, μ – Synthesis). Also, the user is able to specify performance parameters, the number of search iterations and in future versions of the program, which search algorithm the program should use.

Initially, for this work, the authors chose to work with an H-infinity, one DOF controller; the block diagram is presented on Fig. 4.

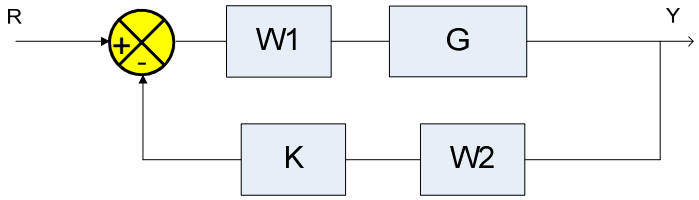

Figure 4. Block diagram for the control problem

In Fig. 4, *G* denotes the plant model for the flexible transmission system; *K*, a robust controller designed through H-infinity optimization; and W_1 and W_2, user-selected weighting functions, whose parameters are subject to change through the search algorithm in order to attain the desired performance.

The initial selection of the weighting functions is an important stage of the design, since it can influence the speed of convergence for the algorithm; since the Moving Boundaries Process is a local search algorithm, poor choice for W_1 and W_2 may prevent the finding of a solution, as a result of being trapped in a local minimum. Therefore, even though the program will greatly aid the work of a control engineer, it still needs attention during the design stage.

Also, during the early stages of the work there was a concern with the final order of the controller, since H-infinity designs tend to result in high order models; with that in mind, the first attempts were made with low order weighting functions, namely first order. The downside, however, is that the algorithm has less degrees of freedom for search. Consider a first order weighting function described by Eq. (20):

$$W_1 = \lambda * \left(\frac{s+\alpha}{s+\beta} \right) \tag{20}$$

where λ represents the gain and α and β the position of zero and pole, respectively. Those three variables are the only parameters that can be changed during the search process, and depending on the complexity of the program and number of constraints, this lack of flexibility tends to greatly increase the difficulty in attaining the desirable solution. Therefore, the user must be aware of this aspect when choosing the weighting functions. For the particular problem in focus (a benchmark problem in robust control), the choice was made to increase the order of W_1. For the case of dealing with problems with multiple inputs / outputs, the program offers the choice to work with diagonal weighting functions, and the user may chose if its elements are equal or different, thus assigning different weights to each input / output pair.

The final design for W_1 and W_2, obtained as a result of the design process just described, were:

$$W_1 = 30 * \frac{(s+4.2)*(s+5.4)*(s+9)}{(s+75.4)*(s+89.5)*(s+120)} \tag{21}$$

$$W_2 = 1 \tag{22}$$

If the choice is made to work with 2 – DOF design, the pre-filter parameters are also subject to the search algorithm, therefore increasing the flexibility of the program. There is, however, the downside of increased order for the final system.

The user should also input the desired constraints for the search problem in the form of desirable performance. This can be done as show in Fig. 5.

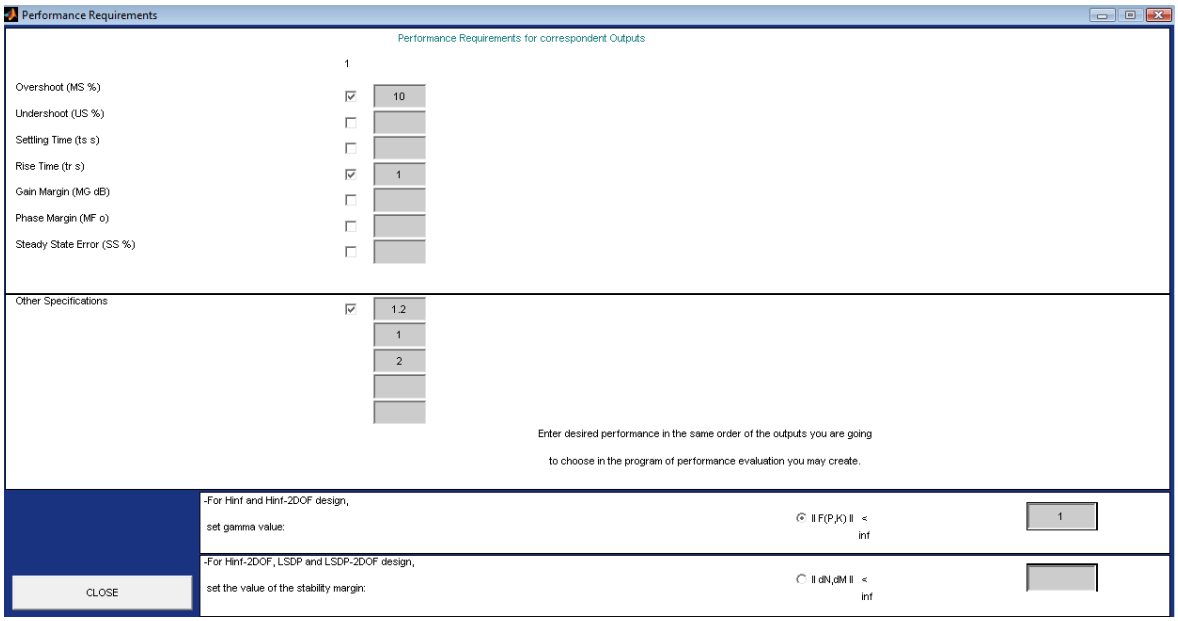

Figure 5. Performance requirements definition window.

Here, it should be stated that even though some of the performance criteria does not have a corresponding checkbox, the user has the option to define them using the "Other Specifications" field. By checking that option, the user is presented with a M-file where he can write the functions he desires to have as constraints, and the output of such functions will be compared to the values that were assigned before. It can be seen in Fig. 5 that the user can define five additional performance parameters. Specifications 4 to 8 in Section 2 were considered using this resource.

Also during the tests, the authors felt that the steady state error should be added as a constraint, hence there will be commentaries regarding this parameter; the other added parameter is the value of gamma, used during H-infinity design procedure.

The last parameter which should be specified before running the design program is the number of desired iterations for the search algorithm. Since the program is intended for the offline design of a fixed parameter controller, this should not be a critical choice, but the user has control over it nonetheless. If at the end of the selected number of iterations the program does not find an acceptable solution, the user may continue the search from the last found point; he only needs to select the "Run" option again, without the need to change any parameters. The program will continue from where it last stopped. This particular characteristic is of interest when running a design with a great number of performance criteria as constraints. The authors experienced an initial difficulty finding a solution point for those problems since the algorithm did not converge to a desirable solution. But, by dividing the number of constraints (four on the first run and five on the second) and running a second cycle of iterations beginning from where the program had stopped, the convergence was attained; the search began on a solution space where the first four constraints were already achieved. This result was a valuable lesson for further use of the program.

With all the specifications set, the program was set to run and therefore find a desirable solution. The first fifteen iterations showed convergence problems, and the steady state error in particular was too far from the specified 1% of tolerable error. Based on the arguments presented above, the decision was made to run the first set of iterations with four constraints, and later on continue the search including the other five constraints. The first run achieved the desired specifications after five iterations; Fig. 6 shows the parameter evolution and desired / achieved performance window:

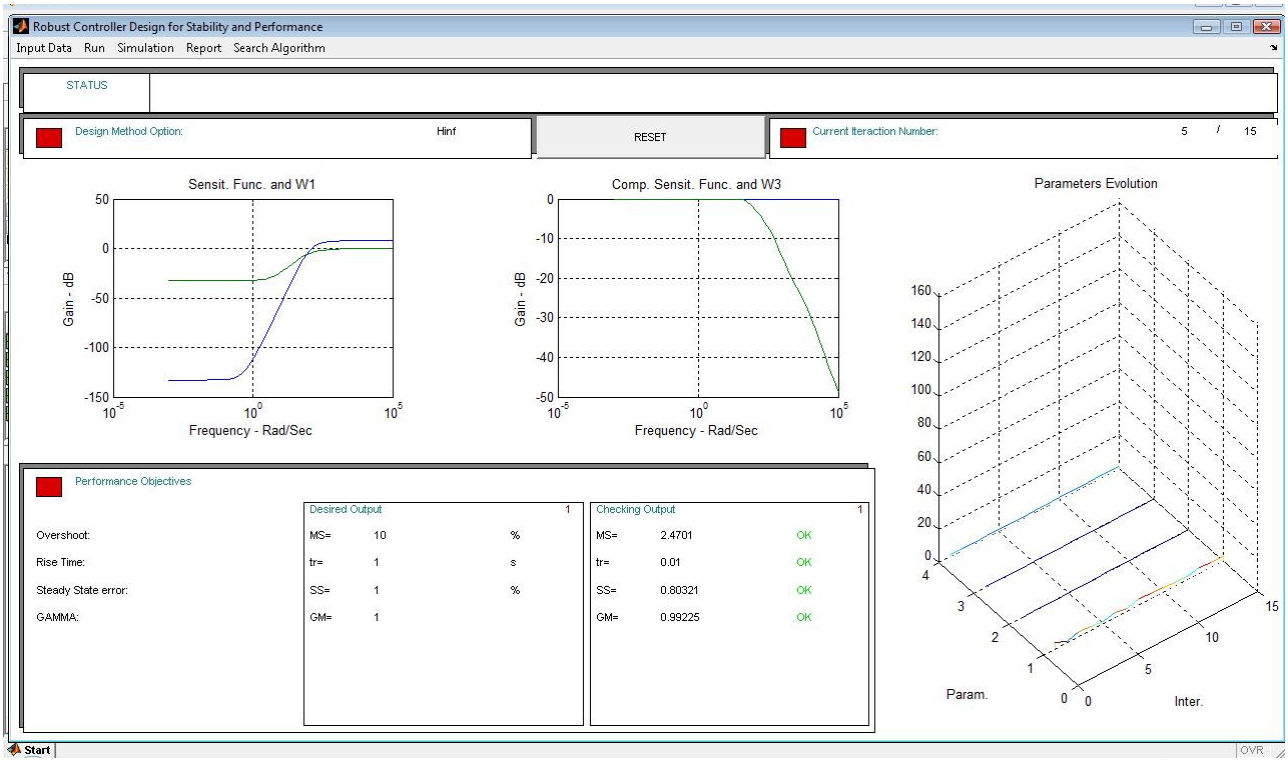

Figure 6. Parameter evolution and current performance.

After the first set of iterations, specifications 4 to 8 were added as constraints, and a new cycle of search was initiated. The delay margin was the only criterion that was not achieved, even after forty iterations. While investigating the evolution of this parameter, it became clear that it was not moving towards the desirable solution; therefore, it might be necessary to change the number of degrees of freedom available for the program, or even it could be the case that one of the added criteria is in conflict with the original specification for the delay margin.

As for the rejection time for a step output disturbance, the peak value for any of the cases was equal or less than 5 % of the unit step amplitude; therefore, the rejection of output disturbances is considered to be efficient enough. In order to certify this assertive, the system was simulated via SIMULINK® as illustrated by Fig.7. For the 'plant' block, the three systems – unloaded, half-loaded and full-loaded – were used one at a time, with the same fixed controller. The responses of the three systems to the disturbance signals are practically identical, and Fig.8 represents the system response for the full-loaded model and the step disturbance. The final controller is of order seven, and the closed-loop system is a eleven-state plant. This is also a satisfactory result, since H-infinity methods tend to result in high order controllers; also, when consulting the original work by Landau et al., the minimal order of the controllers presented in the paper is nine, with one of the controllers having as much as thirty five states.

Equation (23) represents the final controller in zero/pole/gain form.

$$C = \frac{33.438(s + 7.56 \cdot 10^4)(s^2 + 1.059s + 39.72)(s^2 + 125.3s + 4292)(s^2 + 1.697s + 833.6)}{(s + 1.711 \cdot 10^5)(s + 120)(s + 89.5)(s + 84.92)(s + 6.036x10^{-4})(s^2 + 47.75s + 933.7)} \tag{23}$$

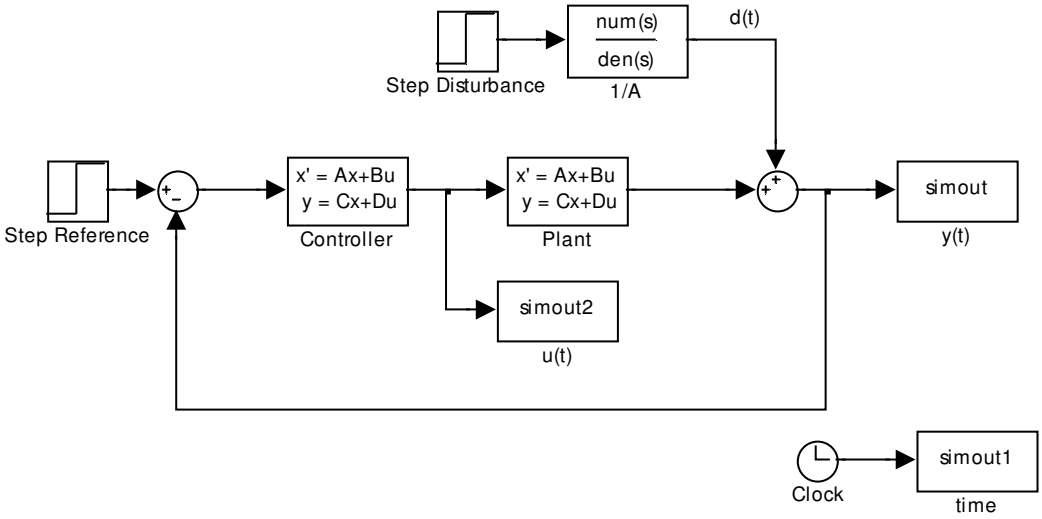

Figure 7. SIMULINK block model.

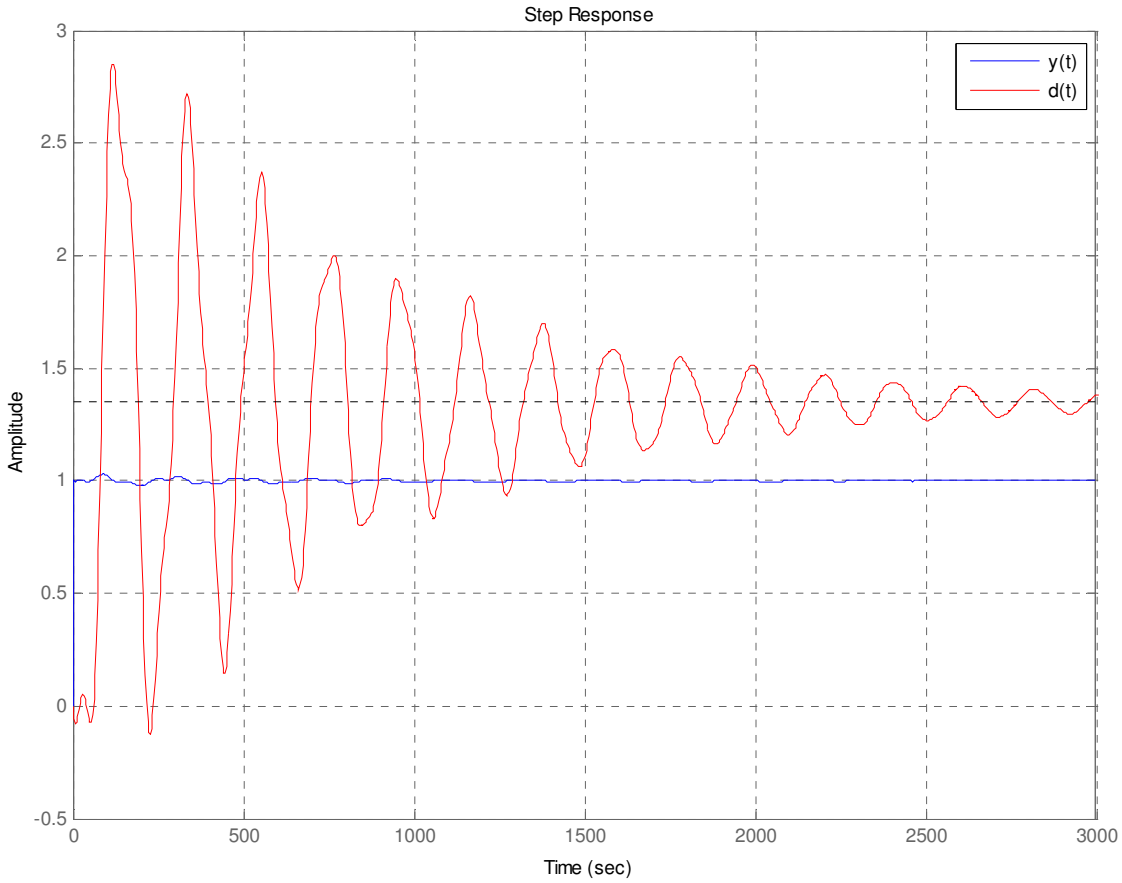

Figure 8. System response y(t) for a step reference and a step disturbance d(t) filtered by 1/A (see figure 7).

Table 1 summarizes the performance attained with the control system as designed.

Table 1. Performance criteria and experimental results for the proposed system with three different load characteristics.

Performance criteria	Desired value	No load	Half - load	Full load
Overshoot	≤ 10 %	0 %	2.52 %	0 %
Rise time	$\leq$ 1 s	0.917 s	0.763 s	0.827 s
Steady-state error	$\leq$ 1 %	0 %	0 %	0 %
Disturbance attenuation	≤ 1	0.4410	0.4327	0.446
Delay Margin	≥ 40 ms	40.04 ms	167 ms	598 ms
Output Sensitivity Function	≤ 6 dB	-31.40 dB	-31.47 dB	-31.31 dB
Input Sensitivity Function	≤ 10 dB	-58.22 dB	-58.446 dB	-58.50 dB

5. CONCLUSIONS

The controller computed by the program was able to achieve the desired performance criteria, whilst maintaining a relatively small order. This characteristic, though not mentioned as a constraint, is an important one, since high order controllers present a greater difficulty for physical implementation.

The algorithmic approach for robust control design is an important tool, saving time and effort during repetitive calculation stages; therefore, the control engineer has more time to focus on the analysis of the problem. It should be noted, however, that the main objective of the program is to be a powerful asset for control designers, and not a substitute for them; careful analysis and experience are still necessary to run the program in a way that will present satisfactory results. Though not yet finished, the program being developed is already capable of solving interesting control problems, and the authors expect to work on other benchmark cases with different characteristics.

6. ACKNOWLEDGEMENTS

The first author would like to thank CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior for scholarship support during his M.Sc course.

7. REFERENCES

Grimble, M.J., 1994, “Robust Industrial Control: optimal design approach for polynomial systems”, Prentice Hall, Southampton, Hampshire, United Kingdom, 497 p.

Gu, D.W., Petkov, P.H. and Konstantinov, M.M., 2005, “Robust Control Design with Matlab®”, Springer-Verlag,

Landau, I.D., Rey, D., Karimi, A., Voda, A. and Franco, A., 1995, “A Flexible Transmission System as a Benchmark for Robust Digital Control”, European Journal of Control.

Maciejowski, J.M., 1989, “Multivariable Feedback Design”, Addison-Wesley, Wokingham, United Kingdom. London, United Kingdom, 389 p.

Skogestad, S. and Postlethwaite, I., 1996, “Multivariable Feedback Control”, John Wileyand Sons Ltd, Baffins Lane, Chichester, United Kingdom, 574 p.

Whidborne, J.F. and Liu, G.P., 1993, “Critical Control Systems: Theory, Design and Applications”, Research Studies Press Ltd, Trenton-United Kingdom, 187 p.

Zakian, V. and Al-Naib, U., 1973, “Design of dynamical and control systems by the method of inequalities”, IEE Proceedings - Control & Science, 120:1421–1427, 1973.

Zakian, V., 1979, “New Formulation for the method of inequalities”, Proc. IEE, Part D, 126(6):579-584.

Zames, G., 1981, “Feedback and optimal sensitivity: Model reference transformations, multiplicative semi-norms and approximate inverses”, IEEE Transactions on Automatic Control, AC-26:301–320.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.