AUTOMATIC LOOP SHAPING IN QFT BY USING A COMPLEX FRACTIONAL ORDER TERMS CONTROLLER

Joaquín Cervera [*,1] **Alfonso Baños** [*,1]

[*] *Faculty of Computer Science, Computer Science and Systems Department, University of Murcia, Campus de Espinardo, 30100 Murcia, Spain; [jcervera,abanos]@um.es*

Abstract: This work focus on the problem of automatic loop shaping in QFT, where traditionally the search of a optimum design, a non convex and nonlinear optimization problem, is simplified by linearizing and/or convexifying the problem. In this work, the authors propose a suboptimal solution using a fixed structure in the compensator. However, in relation to previous work, the main idea consists in the study of the use of a fractional compensator, which give singular properties to automatically shape the open loop gain function by using a minimum set of parameters. A fractional controller, based on complex fractional order poles and zeros, is proposed and successfully applied.

1. INTRODUCTION

Quantitative Feedback Theory is a robust frequency domain control design methodology which has been successfully applied in practical problems from different domains (Horowitz, 1993). One of the key design steps is loop shaping of the open loop gain function to a set of restrictions (or *boundaries*) given by the design specifications and the (uncertain) model of the plant. Although this step has been traditionally performed by hand, the use of CACSD tools (e. g. the QFT Matlab Toolbox (Borghesani *et al.*, 1995), has made the manual loop shaping much more simple. However, the problem of automatic loop shaping is of enormous interest in practice, since the manual loop shaping can be hard for the non experienced engineer, and thus it has received a considerable attention, specially in the last two decades.

Optimal loop computation is a non linear and non convex optimization problem for which it is difficult to find a satisfactory solution, since there is no optimization algorithm which guarantees a globally optimum solution for such a problem.

A possible approach is to simplify the problem in some way, in order to obtain a different optimization problem for which there exists a closed solution, or an optimization algorithm which does guarantees a global optimum. A trade-off between necessarily conservative simplification of the problem and computational solvability has to be chosen. For instance, Thompson (Thomson, 1990) convexifies the problem by using rectangular templates and then solves the resulting nonlinear optimization problem. Gera and Horowitz (Gera and Horowitz, 1980) linearize the problem in their semi-automatic iterative algorithm by considering, in each iteration, a linear approximation of boundaries. Some authors have investigated the loop shaping problem in terms of particular structures, with a certain degree of freedom, which can be shaped to the particular problem to be solved. This is the case in (Fransson *et al.*, 2002), (Chait *et al.*, 1999), (Yaniv and Nagurka, 2004). Another possibility is to use evolutionary algorithms,

[1] This work has been partially supported by Ministerio de Ciencia y Tecnología under project DPI2004-07670-C02-02, which is greatly appreciated by the authors.

able to face nonlinear and non convex optimization problems. This is the approach adopted in (García and Guillén, 2000), (Chen *et al.*, 1998) and (Raimúndez *et al.*, 2001). Evolutionary algorithms' drawbacks are that they are computationally demanding and they do not guarantee, in general, an optimal solution.

However, by the use of information about the problem, instead of just a brute force method, computation effort can be reduced and reasonably close to the optimum solutions can be obtained. In this sense, a good structure for the compensator, in terms of using a reduced set of parameters, but with a rich frequency domain behavior, is of crucial importance. In previous work, the compensator has been fixed to a rational structure, with a finite (but no necessarily small) number of zeros and poles. In this work, the main contribution is to introduce a fractional compensator that, with a minimum number of parameters, gives a flexible structure in the frequency domain regarding automatic loop shaping. In fact, it can be approximated by a rational compensator, but with a considerably large number of parameters. This dramatic reduction in the number of parameters is of capital importance for the success of evolutionary algorithms in the resolution of the automatic loop shaping problem.

The key idea behind this structure choice is to use as much a priori information about the optimum as possible so that the evolutionary search does not have to try any possibility among a large set of parameters, but just to choose a small set of values which parameterize the set of possible controllers according to the incorporated information.

This work, following (Cervera and Baños, 2005), is a first step to introduce fractional control ideas in automatic loop shaping in QFT. It is considered the particular case of minimum phase open loop gain functions, for which the investigated compensators can give a good structure with a reduced set of parameters. The non-minimum phase case will be considered elsewhere.

This work is structured as follows: in section 2 some brief preliminaries about QFT and evolutionary algorithms are introduced; in section 3, a fractional compensator structure is presented and analyzed; in section 4, fractional controllers implementation is studied; in section 5, this structure is applied to a design example which has traditionally been used as benchmark problem in QFT; finally, in section 6 a QFT CACSD tool, able to deal with fractional structures, is presented.

2. SOME PRELIMINARIES

2.1 QUANTITATIVE FEEDBACK THEORY

The basic idea in QFT (Horowitz, 1993) is to define and take into account, along the control design process, the quantitative relation between the amount of uncertainty to deal with and the amount of control effort to use. Typically, the QFT control system configuration (see fig. 1) considers two degrees of freedom: a controller $C(s)$, in the closed loop, which cares for the satisfaction of robust specifications despite uncertainty; and a precompensator, $F(s)$, designed after $C(s)$, which allows to achieve the desired frequency response once uncertainty has been controlled.

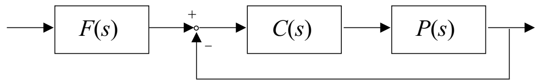

Fig. 1. Two degrees of freedom control system configuration.

For a given plant $P(s)$, its *template* $\mathcal{P}$ is defined as the set of plant possible frequency responses due to uncertainty. A *nominal* plant, $P_0 \in \mathcal{P}$, is chosen.

The design of the controller $C(s)$ is accomplished in the Nichols chart, in terms of the nominal open loop transfer function, $L_0(s) = P_0(s)C(s)$. A discrete set of design frequencies Ω is chosen. Given quantitative specifications on robust stability and robust performance on the closed loop system, *boundaries* $\mathcal{B}_\omega$, $\omega \in \Omega$, are computed. $\mathcal{B}_\omega$ defines the allowed regions for $L_0(j\omega)$ in the Nichols chart, so that $\mathcal{B}_\omega$ being not violated by $L_0(j\omega)$ implies specification satisfaction by $L(j\omega) = P(j\omega)C(j\omega)\forall P(s) \in \mathcal{P}$. The basic step in the design process, *loop shaping*, consists of the design of $L_0(j\omega)$ which satisfies boundaries and is reasonable close to optimum. QFT optimization criterion is the minimization of high frequency gain (Horowitz, 1973), i.e., K_{hf} in the expression

$$\lim_{\omega \to \infty} L(j\omega) = \frac{K_{hf}}{s^{n_{pe}}} \qquad (1)$$

where n_{pe} = excess of poles over zeros.

Loop shaping is traditionally carried out manually with the help of CACSD tools, like QFT Matlab toolbox (Borghesani *et al.*, 1995). This leads to a trial and error process, whose resulting product quality is strongly determined by the designer experience and intuition. There is no commercial tool for this purpose yet. An automatic loop shaping procedure obtaining is, so, a key issue which is still of a great interest.

2.2 EVOLUTIONARY ALGORITHMS

The term *evolutionary algorithms* (Spears *et al.*, 1993) groups a set of different problem solving algorithms which share the basic feature that they are based on natural evolutionary processes. To find a solution to a problem, they simulate the *survival of the fittest* process present in Nature. A *population* of *individuals*, each one representing a possible solution, evolves along a sequence of *generations*. In this process, population goes through *selection*, *mutation* and *reproduction* operations (search operations). Selection is done in terms of an optimization criterion which defines the fitness degree of each individual as solution to the problem. The reproduction process gives priority to the fittest individuals, and this way uses the available fitness information. Mutation arbitrary modifies existing individuals, thus permitting a general way to explore the search space.

Evolutionary algorithms, used for optimization purposes, can deal with complicated problems, including QFT non linear and non convex control design problem. But they cannot guarantee, in general, attainment of globally optimal solution.

Evolutionary algorithms used for optimization can be classified in three major groups: Genetic Algorithms (GA's), Evolutionary Programming (EP) and Evolution Strategies (ES'). The basic difference between GA's and EP/ES' is individual representation. In the first case, they are represented in terms of *genomes*, bit strings which codify the individual characteristics. Operations such us mutation or crossover take place at bit level, so there is a disconnection between evolutionary operations meaning and the change they produce in the individual. In the second case, individuals are directly represented in terms of the (real, integer, ...) variables characterizing them, so evolutionary operations acts on these variables. In this work EP/ES' have been used, by means of the *Genetic and evolutionary algorithm toolbox for use with Matlab* (GEATbx, (Pohlheim, 2004)).

3. AUTOMATIC LOOP SHAPING

The automatic loop shaping method proposed consists of using a fractional controller structure, based on fractional complex poles and zeros, to solve a QFT control design problem. The proposed structure is based on the traditionally used in linear control complex pole term,

$$T = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \qquad (2)$$

with a non integer exponent added in order to obtain higher flexibility in the design, so the controller is composed by terms

$$T_i = \left(\frac{\omega_{ni}^2}{s^2 + 2\delta_i\omega_{ni} s + \omega_{ni}^2}\right)^{e_i} \qquad (3)$$

with $e_i \in \mathbb{R}$, $e_i < 0$ corresponding to zeros.

Different combinations of this kind of terms are possible. Following the aim of parameter minimization, the choice has been the minimal number of poles and zeros which permit to obtain a close to optimum loop. As it has been shown in (Horowitz, 1973), (Horowitz, 1993) and (Lurie and Enright, 2000), based on ideas in (Bode, 1945), the optimal loop follows the right and bottom parts of the UHFB. This has led to the use of two poles and one zero in the following manner. One pole is used to get the loop close to the UHFB right side and to go along it with the constant phase given by the phase margin (see fig. 2). Second pole is used to get a new leftwards movement,

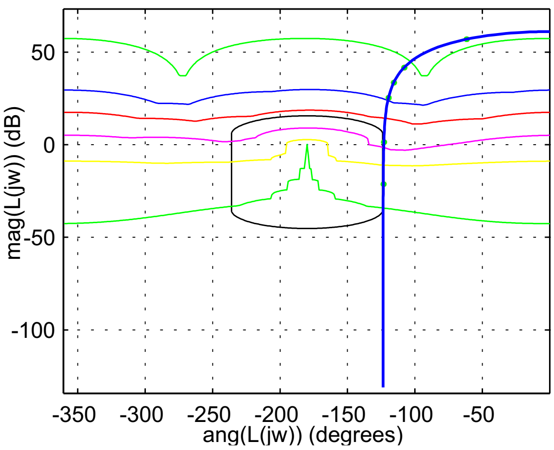

Fig. 2. First fractional order pole effect.

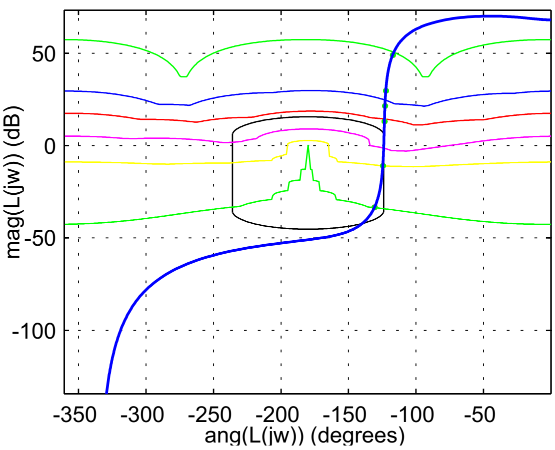

Fig. 3. Second fractional order pole effect.

after the UHFB right bottom corner, which permits approaching the UHFB bottom and a phase fast increase towards the final phase, given by the excess of poles over zeros (see fig. 3). This two poles do not give enough flexibility to tightly follow the UHFB. This is why the zero is added. Its additional degrees of freedom permit a precise adjust of the structure in fig. 3 to get results as shown in fig. 4.

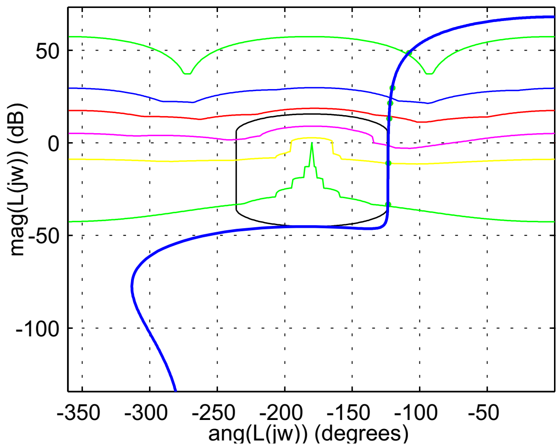

Fig. 4. Fractional order zero effect.

The resulting structure is defined by ten parameters, three per pole or zero plus a constant gain term K

$$L_0 = K\left(\frac{\omega_{n1}^2}{s^2+2\delta_1\omega_{n1}s+\omega_{n1}^2}\right)^{e_1} \left(\frac{\omega_{n2}^2}{s^2+2\delta_2\omega_{n2}s+\omega_{n2}^2}\right)^{e_2} \left(\frac{\omega_{n3}^2}{s^2+2\delta_3\omega_{n3}s+\omega_{n3}^2}\right)^{e_3} \quad (4)$$

Heuristic rules have been developed in order to obtain values for these parameters, which permits an optimization with a (to be chosen) reasonable number of parameters or even a direct automatic loop shape, without optimization.

The rule for d_i, $i \in \{1,2,3\}$ consists of a fixed value choice, which has proven to work fine in every case, for each one: $d_1 = 1$, $d_2 = 0.25$ and $d_3 = 1$.

The rule for e_1 is to choose it so that UHFB right side, given by robust stability specification, γ, and first pole high frequency phases are the same. This common phase can be computed as $\phi = \text{acos}\left(-\sqrt{1-\frac{1}{\gamma^2}}\right)$. The excess of poles over zeros, e_{zp}, is a fixed quantity. Since $e_{zp} = e_1+e_2+e_3$, there is only one degree of freedom in (e_2, e_3), expressed as re_{32} (e_3/e_2). re_{32} can be fixed, for instance $re_{32} = 0.5$ works quite fine.

ω_3 rule is stated in terms of ω_2 as $\omega_3 = 2\omega_2$. The rule for ω_2 has no closed form. It consists of a search algorithm which computes which ω_2 value makes the loop coincide with the bottom of the UHFB in its middle point. No specific rule has been defined for ω_1. It can manually be used to satisfy tracking boundaries.

About K, its rule depends on the choice of a ω_{cg} crossover frequency. It is computed so that the loop, in terms of the remaining already computed parameters, crosses 0 dB at the chosen frequency.

4. CONTROLLER IMPLEMENTATION

Fractional order transfer functions implementation in terms of rational terms can be performed in different ways. The AFT method developed in (Baños and Gómez, 1995), based on nonlinear optimization with a least square objective function, is a good approach, which converges very rapidly thanks to the use of gradient information. It has the drawback that a reasonably close to optimum first estimation has to be supplied, otherwise it easily converges to local minima. Evolutionary search techniques can also be applied. They are more robust against local minima, but its convergence is slower than the former's. A combination of both, consisting of an evolutionary pre-search which provides an initial estimation for AFT, has proven to give very good result. A trade-off between rational controller complexity an approximation quality has to be chosen. As an example, the fractional loop L_0 in (12) is approached by the order 6 rational transfer function

$$L_6(s) = \frac{14.9}{s}\frac{\left(\frac{s}{561}+1\right)\left(\frac{s}{687.6}+1\right)}{\left(\frac{s}{167}+1\right)\left(\frac{s}{701}+1\right)} \frac{\left(\frac{s^2}{170.6^2}\frac{2\times 7.5s}{170.6}+1\right)}{\left(\frac{s^2}{40.4^2}\frac{2\times 9.2s}{40.4}+1\right)\left(\frac{s^2}{844^2}\frac{2\times 0.01s}{844}+1\right)} \quad (5)$$

and also by the order 14 rational transfer function

$$L_{14}(s) = \frac{14.9}{s}\frac{\left(\frac{s}{993.9}+1\right)\left(\frac{s}{802.4}+1\right)\left(\frac{s}{56.3}+1\right)}{\left(\frac{s}{705.6}+1\right)\left(\frac{s}{614.4}+1\right)\left(\frac{s}{744.1}+1\right)} \frac{\left(\frac{s}{452.7}+1\right)\left(\frac{s}{564.1}+1\right)\left(\frac{s}{522.8}+1\right)}{\left(\frac{s}{654.4}+1\right)\left(\frac{s}{22.1}+1\right)\left(\frac{s}{547.7}+1\right)} \frac{\left(\frac{s}{377.4}+1\right)\left(\frac{s}{619.1}+1\right)\left(\frac{s}{1000}+1\right)}{\left(\frac{s}{689}+1\right)\left(\frac{s}{242.6}+1\right)\left(\frac{s}{203}+1\right)} \frac{\left(\frac{s}{205.4}+1\right)\left(\frac{s^2}{81.3^2}\frac{2\times 10s}{81.3}+1\right)}{\left(\frac{s}{724.6}+1\right)\left(\frac{s^2}{720^2}\frac{2\times 0.12s}{720}+1\right)\left(\frac{s^2}{27.4^2}\frac{2\times 10s}{27.4}+1\right)} \quad (6)$$

both shown in fig. 5 together with the orginal fractional transfer function.

5. DESIGN EXAMPLE

To illustrate the behavior of the proposed design procedure, the QFT Toolbox for Matlab (Borghesani *et al.*, 1995) Benchmark Example number 2 is be used. It has also been used, for instance, in (Chen *et al.*, 1998) and (Raimúndez *et al.*, 2001).

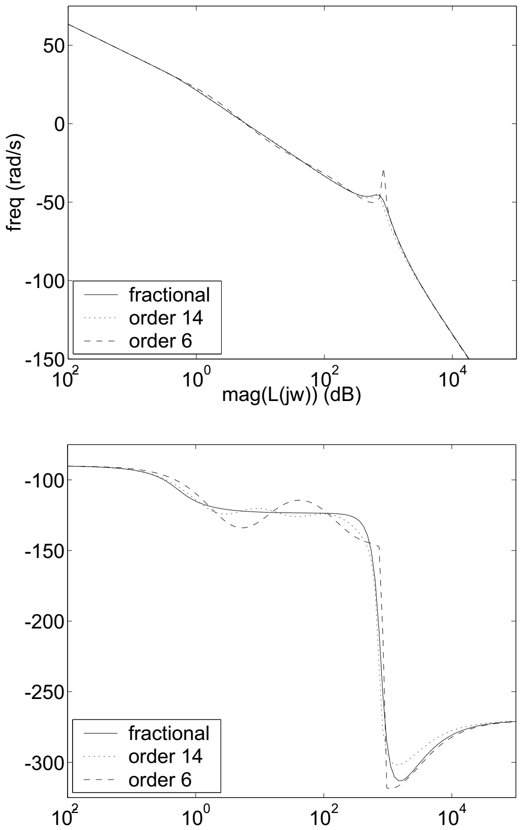

Fig. 5. Bode diagrams for $L_0(j\omega)$ in (12) and its rational approaches (5) and (6)

The plant is defined as

$$\mathcal{P} = \left\{ P(s) = \frac{ka}{s(s+a)}, k \in [1,10], a \in [1,10] \right\} \quad (7)$$

The specification is given by a robust stability specification,

$$\left| \frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)} \right| \leq 1.2, \ \forall P \in \mathcal{P}, \ \omega \geq 0 \quad (8)$$

and a robust tracking performance, given by

$$T_{min}(\omega) \leq \left| F(j\omega) \frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)} \right| \leq T_{max}(\omega) \quad (9)$$

where $F(j\omega)$ is the prefilter,

$$T_{min}(\omega) = \left| \frac{0.6584(j\omega + 30)}{(j\omega)^2 + 4(j\omega) + 19.752} \right| \quad (10)$$

and

$$T_{max}(\omega) = \left| \frac{120}{(j\omega)^3 + 17(j\omega)^2 + 82(j\omega) + 120} \right| \quad (11)$$

As in the QFT toolbox, the set of frequencies considered for checking tracking specification is $\{0.1, 0.5, 1, 2, 15, 100]\}$. For robust stability checking a unique universal high frequency boundary (UHFB) will we considered, for $\omega = 10000$ rad/s. According to plant open loop crossover frequency, around 1 rad/s, and to the boundaries observed, a tentative $w_{cg} = 10$ is established.

A polo at the origin is added to the structure in order to avoid cancellation. Using all the rules defined in section 3, the loop in fig. 6 is obtained, with $K_{hf} = 118.91$ dB. After trying some more

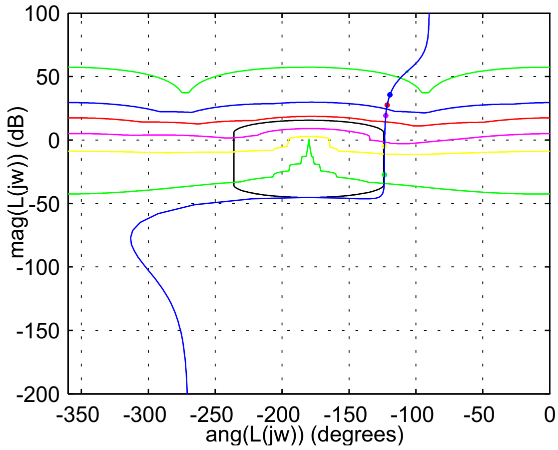

Fig. 6. Direct parameter assignment (no optimization) loop for the design example.

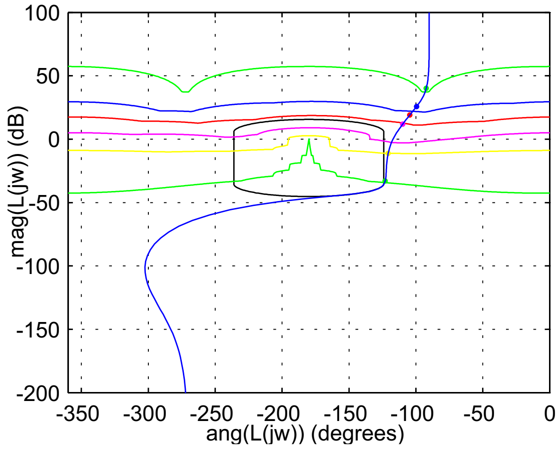

Fig. 7. Four free parameters optimization loop for the design example.

ω_{cg} values, $\omega_{cg} = 6rad/s$ is found to be closer to the optimum, with $K_{hf} = 105.6$ dB.

About optimization, experiments show including ω_2 algorithmic rule is not efficient, so it is not applied. Parameters d_1, d_2 and d_3 are left free. Additional free parameters can be considered, but the trade-off between quality of obtained results and computational effort has proven to be between 4 to 5 free parameters. A good candidate as a fifth free parameter is re_{32}.

In this example, for $\omega_{cg} = 6rad/s$, optimization high frequency gain in about 4 dB with respect to all rules method, with $K_{hf} = 101.763$ dB in a four free parameters optimization, with resulting loop

$$L_0 = 10.26 \left(\frac{1.77}{s^2 + 3.7s + 1.77} \right)^{0.19} \left(\frac{341000}{s^2 + 700.7s + 341000} \right)^{1.6} \left(\frac{1363900}{s^2 + 6063s + 1363900} \right)^{-0.78} \quad (12)$$

shown in fig. 7.

6. CACSD TOOL

The development of this research work has required programming some not available functionalities . Some of them were available from Matlab as standard functions, but only applicable to transfer functions involving integer exponents. For instance, standard Matlab *bode* function cannot be used with non integer exponents. Some other functionalities emerged specifically from the demands of the research process, for instance, the need to switch on/off each individual component in the visualization of a certain loop transfer function, or to switch on/off the automatic computation of certain parameters in terms of the others.

The progressive satisfaction of this software needs has led to the development of a software tool which lets the control designer to shape the open loop, given a fixed structure, in a quite comfortable and general way. This software is still a laboratory product and completely tailored to this research, but is being partly programmed thinking of making it evolve, in the near future, to a complete Matlab QFT fractional toolbox.

This tool consists of a set of interconnected windows showing or letting modify different data about the design process. All of them are interactively updated, so that when a parameter is changed in any window, this change is reflected in the others. This allows a interactive design in which one can have an updated information, from different points of view, of any taken action. For instance, it is possible to view the Nichols chart of the open loop (including boundaries and Nichols chart of individual loop components), Bode diagrams, noise charts, etc.

When the program is run the user is presented the *main window* (fig. 8), where the loop

structure or the plant to be controlled can be chosen. Some other minor actions can be performed from this window, like choosing the vector of frequencies to be considered during design, the

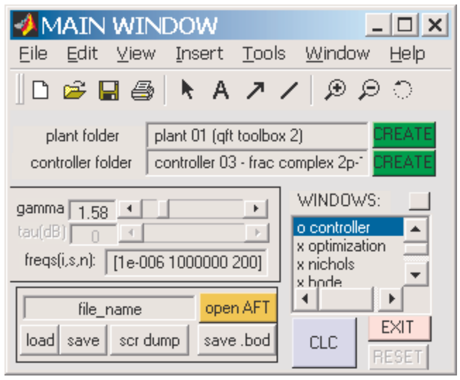

Fig. 8. Main and loop design windows.

Fig. 9. Main and loop design windows.

stability margin to be used (γ), etc. From this window all the other ones can be opened or closed. The windows in which the user can perform actions are the *loop design window* (second window in fig. 9) and the *optimization window*.

In the *loop design window*, for each parameter in the chosen loop structure, there are controls that let set its value directly, by writing its digits, or by auto scalable slider controls. According to the settings in the loop structure configuration file, some of the parameters, so-called *auto*, are (potentially) computed in terms of the values of the others parameters. Wether an auto parameter is or not computed this way can be chosen at any time by means of a radio button associated to it. Furthermore, it is possible to define the different components of the loop structure. For instance, for the controller structure defined in section 3 , components

$$\begin{cases} C_1(s) = \left(\dfrac{\omega_1^2}{s^2 + 2\delta_1\omega_1 s + \omega_1^2} \right)^{e_1} \\ C_2(s) = \left(\dfrac{\omega_2^2}{s^2 + 2\delta_2\omega_2 s + \omega_2^2} \right)^{e_2} \\ C_3(s) = \left(\dfrac{\omega_3^2}{s^2 + 2\delta_3\omega_3 s + \omega_3^2} \right)^{e_3} \end{cases} \quad (13)$$

are defined. Each component is represented in Nichols and Bode diagram by its own curve, and the effect of each component can be included or excluded from the complete open loop transfer function by its corresponding radio button C_i. This is especially useful for research purposes.

In the *optimization window*, optimization can be configured and launched. A log of computed optimizations is kept so that any obtained result can be easily recovered. Optimization is also connected to the other windows, in such a way the loop resulting from an optimization is represented and can be manipulated. While optimization is being carried out, considered loops can be displayed.

7. CONCLUSIONS

An automatic loop shaping procedure, based on evolutionary algorithms optimization on the parameters of a fixed, fractional order complex poles and zeros structure, has been proposed. The key idea behind this proposal is the introduction of a structure with few parameters but, at the same time, flexible enough, thanks to its fractional nature, to get results which are close to the optimum. It has been shown how this method produces quite close to the optimum solutions, and also that fractional obtained results are implementable by rational terms.

As additional contribution (necessary to obtain the first), a CACSD QFT has been (and is being) developed, tool which makes easier to go on the research of new structures, since has been programmed keeping generality in mind as a basic design principle. It is an author's aim to continue the development of this tool towards a complete QFT design Matlab toolbox.

REFERENCES

Baños, A. and F. Gómez (1995). Parametric identification of transfer functions from frequency response data. *IEE Computing and Control Engineering Journal* **137**, 137144.

Bode, H.W. (1945). *Network Analysis and Feedback Amplifier Design*. Van Nostrand. NY, USA.

Borghesani, C., Y. Chait and O. Yaniv (1995). *Quantitative Feedback Theory Toolbox*. The MathWorks, Inc.. Natick, MA, USA.

Cervera, J. and A. Baños (2005). Automatic loop shaping in qft by using 2nd generation crone structure. In: *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*. (submitted).

Chait, Y., Q. Chen and C.V. Hollot (1999). Automatic loop-shaping of qft controllers via linear programming. *ASME J. Dynamic Systems, Measurement, and Control* **121**, 351357.

Chen, W.H., D.J. Ballance and Y. Li (1998). Automatic loop-shaping in qft using genetic algorithms. Technical report. Centre for Systems and Control, University of Glasgow. Glasglow, UK.

Fransson, C.M., B. Lennartson, T. Wik, K. Holmström, M. Saunders and P.O. Gutman (2002). Global controller optimizacion using horowitz bounds. In: *Proceedings of the IFAC 15th Trienial World Congress*.

García, M. and J.C. Guillén (2000). Automatic loop shaping of qft controllers via genetic algorithm. In: *Robust Control Design 2000 (ROCOND 2000) Proc. volume from the 3rd IFAC Symposium*. Vol. 2. Elsevier Sci.. Kidlington, UK. pp. 603–608.

Gera, A. and I. Horowitz (1980). Optimization of the loop transfer function. *Int. J. of Control* **31**, 389–398.

Horowitz, I. (1973). Optimum loop transfer function in sigle-loop minimum-phase feedback systems. *Int. J. of Control* **18**, 97–113.

Horowitz, I. (1993). *Quantitative Feedback Design Theory - QFT (Vol.1)*. QFT Press. Boulder, Colorado, USA.

Lurie, B.J. and P.J. Enright (2000). *Classical Feedback Control with MATLAB*. Marcel Dekker. New York, NY.

Pohlheim, H. (2004). Geatbx: Genetic and evolutionary algorithm toolbox for use with matlab documentation. http://www.geatbx.com. Version 3.50.

Raimúndez, C., A. Baños and A. Barreiro (2001). Qft controller synthesis using evolutive strategies. In: *Proc. of the 5th International QFT Symposium on Quantitative Feedback Theory and Robust Frequency Domain Methods*. Pamplona, Spain. pp. 291–296.

Spears, W. M., K. A. De Jong, T. Bäck, D. B. Fogel and H. de Garis (1993). An overview of evolutionary computation. In: *Proceedings of the European Conference on Machine Learning*. Vol. 667. pp. 442–459.

Thomson, D.F. (1990). Optimal and Sub-Optimal Loop Shaping in Quantitative Feedback Theory. PhD thesis. School of Mechanical Eng., Purdue University. West Lafayette, IN, USA.

Yaniv, O. and M. Nagurka (2004). Automatic loop shaping of structured controllers satisfying qft performance. Technical report.