Automated Synthesis of Fixed Structure QFT Prefilters using Interval Constraint Satisfaction Techniques

By

P.S.V. Nataraj and M.M. Deshpande

IDP in Systems and Control Engineering,

Indian Institute of Technology Bombay, Mumbai, India.

March 3, 2010

Outline

- The problem
- QFT background
- Design approach
- Result
- Conclusion
- Future scope

Motivation

- Robust control system design is of great practical interest and its automation is a key concern in control system design.
- This is still an open problem.

Robust Control system

Robust control system is one which

- stabilizes the plant despite of plant parameters uncertainty, and
- achieves the performance specifications like rise time, peak overshoot etc.

Several robust control methodologies in the time as well as in the frequency domains have been proposed; for instance,

- LQG/LTR,
- Q-parametrization, H_2, H_∞, μ-synthesis, and
- Quantitative feedback theory (QFT).

The robust control methodology-QFT

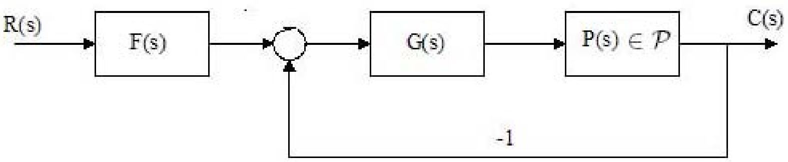
Horowitz's QFT approach to design robust feedback systems has been gaining popularity in the control literature.

In contrast to other robust control techniques, QFT provides:

- Design transparency.
- Enables the user to assess quantitatively the cost of feedback.
- Uses the phase information in the design process.

QFT...

Consider a two-degree-of-freedom feedback system configuration as shown in Figure



where $G(s)$ and $F(s)$ are the controller and prefilter respectively.

QFT...

The main steps of the QFT design procedure are:

1. Template Generation: takes into account parametric uncertainty, at each design frequency
2. Computation of QFT bounds: translate the stability and performance specifications using the plant templates to obtain the stability and performance bounds in the Nichols chart.
3. Design of Controller: such that
   - The bound constraints at each design frequency ω_i are satisfied.
   - The nominal closed loop system is stable.
4. Design of Prefilter: such that the robust tracking specifications are satisfied.

QFT...

Robust tracking performance: controller $G(s)$ and prefilter $F(s)$ are designed such that

$$|T_L(j\omega)| \leq \left| \frac{F(j\omega)L(j\omega)}{1 + L(j\omega)} \right| \leq |T_U(j\omega)|$$

where the transfer functions $T_L(s)$ and $T_U(s)$ are called the lower and upper tracking models (on the tracking specifications).

$L(s) = G(s)P(s) = l \exp^{j\Psi}$ is open loop transmission function.

Our work

- In this work we propose a new approach to automate the synthesis of a fixed structure QFT prefilter.
- First, the QFT prefilter synthesis problem is modeled as an interval constraint satisfaction problem (ICSP) and then solved using existing, efficient interval constraint satisfaction technique (ICST)-based algorithm.
- Traditionally, this synthesis is done *manually* by the designer, relying on design experience and skill.

The benefits of automatic design of prefilter are quite obvious. For instance, in the $n \times n$ MIMO design case of QFT, where n^2 number of prefilters are designed sequentially, any overdesign of the prefilter in a loop may lead to controller overdesign in the subsequent steps.

Automated Synthesis of Prefilters

- Selection of the robust control methodology.

Automated Synthesis of Prefilters

- Selection of the robust control methodology.
- Formulation of the synthesis problem:

Automated Synthesis of Prefilters

- Selection of the robust control methodology.
- Formulation of the synthesis problem:
  - Conversion of the prefilter synthesis problem into constraint satisfaction problem (CSP)

Automated Synthesis of Prefilters

- Selection of the robust control methodology.
- Formulation of the synthesis problem:
  - Conversion of the prefilter synthesis problem into constraint satisfaction problem (CSP)
  - Choice of the prefilter structure.

Automated Synthesis of Prefilters

- Selection of the robust control methodology.
- Formulation of the synthesis problem:
  - Conversion of the prefilter synthesis problem into constraint satisfaction problem (CSP)
  - Choice of the prefilter structure.
  - Specification of the prefilter parameter search space.

Automated Synthesis of Prefilters

- Selection of the robust control methodology.
- Formulation of the synthesis problem:
  - Conversion of the prefilter synthesis problem into constraint satisfaction problem (CSP)
  - Choice of the prefilter structure.
  - Specification of the prefilter parameter search space.
- Choice of the CSP solver for the above problem.

Prefilter Synthesis Problem

$$T(j\omega) = \frac{L(j\omega,\lambda)}{1+L(j\omega,\lambda)} = \frac{G(j\omega)P(j\omega,\lambda)}{1+G(j\omega)P(j\omega,\lambda)}$$

and

$$T_R(j\omega) = F(j\omega)T(j\omega)$$

$$|T_L(j\omega)| \leq |T_R(j\omega)| \leq |T_U(j\omega)|, \forall\omega, \forall P \in \mathcal{P}$$

Prefilter Synthesis Problem

At each design frequency ω, and for each $P \in \mathcal{P}$, the above results in two inequalities

$$|T_L(j\omega)| - |F(j\omega)||T(j\omega)| \leq 0,$$
$$|F(j\omega)||T(j\omega)| - |T_U(j\omega)| \leq 0$$

The variables of the inequalities are prefilter parameters. These inequalities form a constraint set $\mathcal{C}$.

Synthesis problem

Given the tracking specifications and a controller, the design of the prefilter can be carried out as follows:

Specify the prefilter as

$$F(s,x) = \frac{\prod_{i=1}^{n_z} (s/z_i + 1)}{\prod_{k=1}^{n_p} (s/p_k + 1) \prod_{k=1}^{n'_p} \left(s^2/\vartheta_{n_k}^2 + 2\xi_k s/\vartheta_{n_k} + 1\right)}$$

where the prefilter parameter vector x is

$$x = (z_1, .., z_{n_z}, p_1, .., p_{n_p}, \xi_1, .., \xi_{n'_p}, \vartheta_{n_1}, .., \vartheta_{n_{n'_p}})$$

Prefilter synthesis as ICSP

For example consider a filter structure

$$F(s) = \frac{1}{\left(\frac{s}{p_1} + 1\right)\left(\frac{s}{p_2} + 1\right)}$$

- Constraints

$$|T_L(\omega_i)|\sqrt{(1 + \frac{\omega_i^2}{p^2})(1 + \frac{\omega_i^2}{p_1^2})} - |T_k(\omega_i)| \leq 0$$

$$|T_U(\omega_i)|\sqrt{(1 + \frac{\omega_i^2}{p^2})(1 + \frac{\omega_i^2}{p_1^2})} - |T_k(\omega_i)| \geq 0$$

  where i is number of design frequencies and k is number of uncertain plants.

- Variables, $x = (\mathbf{p_1}, \mathbf{p_2})$
- Search domain, $\mathbf{x} = ([0, 100], [0, 100])$

Synthesis problem

- Solving above constraints for parameters of fixed structure prefilter in given search domain is prefilter synthesis problem.
- This becomes a problem of interval constraint satisfaction (ICSP) and finding prefilter parameters values which will satisfy above constraints will be its solution.

CSP

- A constraint is a relation between the variables or unknowns of a problem, each taking a value in a domain.
- A constraint satisfaction problem (CSP) is given by a set of constraints and search space defined as the Cartesian product of variable domains

$$c_i(x_1, x_2) \geq 0, \text{for}(i = 1, ..., n),$$

- The solution set of a CSP, c_i is the set of all elements from the search space, $x = x_1, x_2$ that satisfy all the constraints.
- The nonlinear and non-convex constraint satisfaction over real numbers is generally NP hard problem.
- Hence, constraint satisfaction over intervals is used.

ICSP

- In interval constraint, a constraint in which variables are associated with intervals denoting their domains of possible values.
- An interval constraint satisfaction problem (ICSP) has
  - A set of constraints $\mathcal{C} = (c_1, ...c_n)$
  - A set of variables $X = (x_1, ...x_n)$ and
  - Interval Domain of each variable $\mathbf{x} = (\mathbf{x_1}, ...\mathbf{x_n})$
- An (ICSP) can be stated as follows.

$$\mathcal{C}_i(x) \geq 0, for(i = 1, ..., n), x \in \mathbf{x}$$

ICSP solver

- Algorithm for solving an ICSP involves pruning and bisecting of domains.
- Number of bisection makes algorithm very slow.
- Number of Bisection can be reduced with the help of pruning techniques such as constraint consistency techniques -also called as interval constraint satisfaction techniques (ICST).
- The pruning step combines different techniques to narrow down the domains by removing locally inconsistent intervals containing no solution of some constraint.
- There are different types of consistencies but the main ones are box and hull consistency.

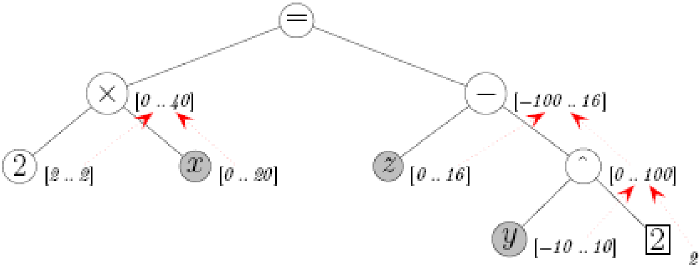
ICST

- In hull consistency, the structure of the initial constraints is broken. The new variables are introduced by decomposing the original constraints.
- For the constraints having occurrence of a variable once in a constraint, this is a very fast and effective approach.
- But domain tightening is hindered due to occurrence of a variable more than once in a constraint (dependency problem).
- The refinement in hull consistency results in HC4 algorithm.

HC4 Pruning

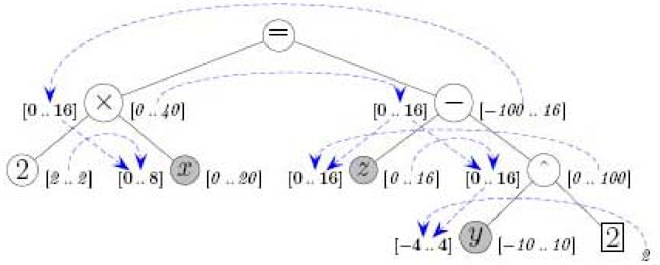
Forward traversing for

$$2x = z - y^2 \quad \text{with} \quad x = [0, 20], y = [-10, 10], z = [0, 16]$$



HC4 Pruning

Backward traversing



Proposed approach

Algorithm for ICSP

Input: Constraints C, Initial Search Box, **B** and accuracy ϵ

Output: Solution Box with all feasible prefilters or "NO Solution Exists "

1. Initialize the box list L with initial box B
2. Take a box from list L and prune it using HC4 pruning. If no box in list, Exit.
3. If box can not be pruned further and width of box is less or equal to ϵ, store the box as solution. Go to step 2
4. Bisect the box in maximum width direction and put the sub-boxes in list.
5. Go to step 2.

Features of the proposed Method

- It enables the designer to specify in advance the structure of the prefilter to be synthesized.
- It is computationally efficient method.
- If no feasible prefilter exists, then the method is guaranteed to computationally verify this fact.
- If a feasible prefilter does exist, then the method is guaranteed to find all prefilter lying within the search box.

Design Example

The plant $P(s)$ has a parametric uncertain model

$$P(s) = \frac{ka}{s(s+a)}, k = [1, 10], a = [1, 10]$$

The controller $G(s)$ is to be designed such that

- Robust stability

$$\left| \frac{P(j\omega)G(j\omega)}{1 + P(j\omega)G(j\omega)} \right| < \infty \quad \text{for all} \quad \omega$$

.

- Robust margins (via closed-loop magnitude)

$$\left| \frac{P(j\omega)G(j\omega)}{1 + P(j\omega)G(j\omega)} \right| \leq 1.2 \quad \text{for all} \quad \omega \geq 0.$$

Design Example contd.

Robust tracking (related to tracking step responses)

$$|T_L(j\omega)| \leq \left| F(j\omega)\frac{P(j\omega)G(j\omega)}{1+P(j\omega)G(j\omega)} \right| \leq |T_U(j\omega)|, \text{for all } \ \omega < 10.$$

where

$$|T_L(j\omega)| = \left| \frac{120}{(j\omega)^3 + 17(j\omega)^2 + 828(j\omega) + 120} \right|$$

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

Solution

The design frequency set is chosen as

$$\Omega = [0.1, 0.5, 1.0, 2.0, 15, 100]\,.$$

We shall first use the controller reported in MATLAB-QFT toolbox as

$$G_a(s) = \frac{9.01\left(\frac{s}{113.8}+1\right)\left(\frac{s}{1.1}+1\right)}{\left(\frac{s}{42.81}+1\right)\left(\frac{s^2}{10^6}+\frac{1486s}{10^6}+1\right)}$$

For this controller, we attempt to synthesize a prefilter of the form (with a complex pole pair)

$$F_a(s) = \frac{1}{\left(\frac{s^2}{\omega_n^2} + \frac{2\xi s}{\omega_n} + 1\right)}$$

The unknown variables are the parameters of the prefilter transfer function. The initial search box for the parameters $\mathbf{x} = (\xi, \omega_\mathbf{n})$ is constructed as

$$\mathbf{x} = \left([0.5, 2], [10^{-4}, 25]\right)$$

Prefilter synthesis as ICSP

- Constraints

$$|T_L(\omega_i)|\sqrt{(1-\frac{\omega_i^2}{\omega_n^2})^2+4\xi^2\frac{\omega_i^2}{\omega_n^2}}-|T_k(\omega_i)|\leq 0$$

$$|T_U(\omega_i)|\sqrt{(1-\frac{\omega_i^2}{\omega_n^2})^2+4\xi^2\frac{\omega_i^2}{\omega_n^2}}-|T_k(\omega_i)|\geq 0$$

- Variables, $\mathbf{x} = (\xi, \omega_\mathbf{n})$
- Search domain for the variables

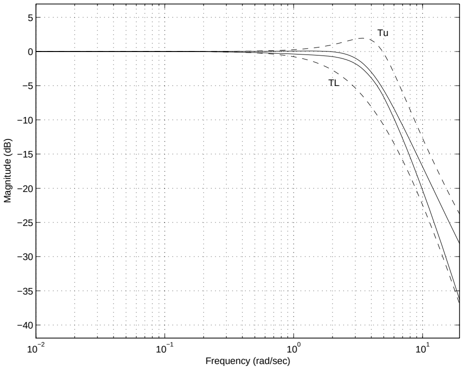
$$\mathbf{x} = \left([0.5, 2], [10^{-4}, 25]\right)$$

- The constraints are obtained at each design frequency and each plant in plant set.
- The ICSP is then solved using interval constraint solver, **RealPaver** with HC4 technique.
- The solver gives all possible values of ξ and ω_n in initial search domain that satisfy all the constraints.
- $\xi = 0.663$ and $\omega_n = 3.752$ is chosen

With those values we get the prefilter as

$$F_a(s) = \frac{1}{\left(\frac{s^2}{(3.752)^2} + \frac{2*0.663s}{3.752} + 1\right)}$$

The closed loop frequency responses of the entire plant set with the synthesized filter $F_a(s)$, denoted as solid lines.



To demonstrate further, we choose this time a different controller (designed using automated controller synthesis approach) as

$$G_b(s) = \frac{10.95\left(\frac{s}{2.1}+1\right)}{\left(\frac{s}{950}+1\right)\left(\frac{s}{982}+1\right)}$$

We also now specify a different prefilter structure (with two real poles)

$$F_b(s) = \frac{1}{\left(\frac{s}{p_1}+1\right)\left(\frac{s}{p_2}+1\right)}$$

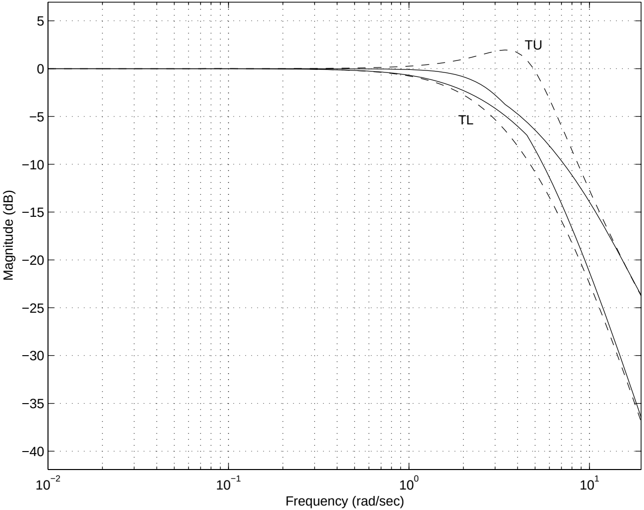
the initial search box $\mathbf{x} = (\mathbf{p_1}, \mathbf{p_2})$ as

$$\mathbf{x} = ([0, 100], [0, 100])$$

The algorithm gives all possible prefilter designs in about one second on a PC. From among these solutions, the following prefilter is chosen

$$F_b(s) = \frac{1}{\left(\frac{s}{3.197} + 1\right)\left(\frac{s}{8.61} + 1\right)}$$

The closed loop frequency responses of the entire plant set with the synthesized filter $F_b(s)$, denoted as solid lines.



Conclusions

1. An efficient approach is proposed for the automatic synthesis of fixed structure QFT prefilters using interval constraint satisfaction techniques like box and hull consistency.
2. The proposed ICST based design approach alleviate the difficulties faced by most QFT designers in key step: prefilter design.
3. The proposed approach is used successfully to design prefilters of two different structures for a benchmark problem.

References

- I. M. Horowitz. *Quantitative Feedback Design Theory* (QFT). QFT Publications, Boulder, Colorado, 1993.
- E. Hansen and G. Walster. *Global Optimization using Interval Analysis, Second Edition.* Marcel Dekker, New York, 2005.
- F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. *Revising hull and box consistency.* In Proc. of 16th International Conference on Logic Programming, pages 230244. MIT Press, Las Cruces, NM, USA, 1999.
- L. Granvilliers. *RealPaver: An interval solver using constraint satisfaction techniques.* ACM Transaction on Mathematical Software, 32(1):138-156, 2006.

Thank You !