Received: 2 May 2019 | Revised: 24 February 2021 | Accepted: 28 February 2021

DOI: 10.1002/rnc.5515

RESEARCH ARTICLE

WILEY

Automatic synthesis of feedforward elements in quantitative feedback theory

Jorge Elso[1] | J. Xabier Ostolaza[2]

[1]Department of Engineering, Public University of Navarre, Pamplona, Spain

[2]Systems Engineering and Control Department, University of the Basque Country UPV/EHU, San Sebastián, Spain

Correspondence
Jorge Elso, Department of Engineering, Public University of Navarre, Campus de Arrosadia, 31006 Pamplona, Spain.
Email: jorge.elso@unavarra.es

Funding information
Euskal Herriko Unibertsitatea, Grant/Award Number: GIU16/54; Ministerio de Economía y Competitividad, Grant/Award Number: DPI2015-64985-R

Abstract
Recent developments in quantitative feedback theory (QFT) lead to feedforward design problems with both magnitude and phase constraints. In these cases, manual feedforward tuning becomes much more challenging and time consuming than the traditional prefilter shaping taking place on the Bode plot. This article presents a general procedure for the automatic synthesis of such elements. Feedforward bounds in the complex plane are expressed as constraints of a linear programming problem in which the Bode real-complex relation is implicitly considered, ensuring a stable rational solution. The methodology is successfully tested in a well-known benchmark problem.

KEYWORDS
2DOF, feedforward synthesis, QFT

1 | INTRODUCTION

Quantitative feedback theory (QFT)[1-4] owes its name to the fact that feedback is used in the minimum amount required to fight the effects of plant uncertainty and unknown disturbances. Deeply rooted in classical control theory, QFT aims to provide low-order, low-bandwidth feedback controllers fulfilling a set of frequency-domain specifications. By doing so, QFT leads to designs with minimum cost of feedback, understood as the effect of sensor noise in the system's actuator and plant.

This philosophy defines the way in which QFT confronts two-degree-of-freedom (2DOF) problems, that is, problems in which both feedforward and feedback actions are involved. Unlike inversion-based techniques, in QFT the feedback element is designed first, and its purpose is to reduce the effect of uncertainty to a level in which a single feedforward element can handle the whole set of plants, making all of its members meet the specification.[1,5]

The paradox is that the closer the feedback controller gets to its theoretical optimum,[6] the harder it is to find a solution to the feedforward synthesis problem. However, given that most designers accept certain overdesign in exchange for low order controllers, the feedforward design usually becomes a trivial task. This is particularly the case in the classical QFT tracking problem, in which specifications are defined as upper and lower tolerances on the magnitude of the tracking frequency response. With this kind of specification, the feedforward controller—a prefilter preceding the feedback loop—is easily tuned on the Bode plot. All this could explain why only a few articles deal with automated methods for prefilter shaping.[7-9] This contrasts with the considerable effort that has been put into the automatic tuning of QFT feedback controllers.[10-28]

Abbreviations: 2DOF, two-degree-of-freedom; QFT, quantitative feedback theory.

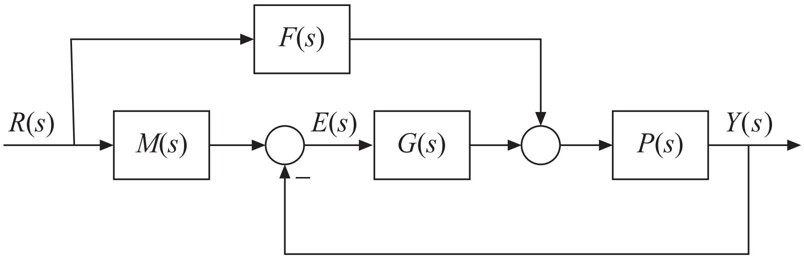


FIGURE 1 Model matching scheme

The situation has changed somewhat in recent years. There is growing interest in the QFT community for 2DOF problems involving restrictions both on the magnitude and the phase of the system responses,[29] such as the tracking error problem,[30-35] the model matching problem and the measurable disturbance rejection problem.[36,37] These specifications capture time-domain restrictions accurately, reduce the demand for feedback and enable the use of feedforward decoupling in multivariable systems.[35]

Despite their benefits, these approaches present a big drawback: the feedforward design must consider both magnitude and phase, and consequently takes place on the log-polar plot[31] rather than on the Bode plot. This represents a major challenge for the designer, who has to derive a transfer function whose responses at different frequencies have to match regions which are often very small. It is clear that an automatic feedforward shaping algorithm would be very useful in this scenario. The purpose of this article is to provide such algorithm.

This article is structured as follows: Section 2 reviews the main QFT problems with magnitude and phase constraints. Section 3 presents new algorithms to automatically synthesize feedforward elements for them. Section 4 applies the methodology to a popular benchmark problem. Finally, the conclusions obtained from this research work are explained in Section 5.

2 | QFT SOLUTIONS FOR 2DOF PROBLEMS INVOLVING MAGNITUDE AND PHASE CONSTRAINTS

For the sake of clarity, dependence on s or $i\omega$ is omitted at certain points of the discussion.

2.1 | Model matching

The model matching problem seeks to constrain the deviation of the closed-loop tracking response with respect to an ideal model $M(s)$. The problem assumes the 2DOF structure depicted in Figure 1. The uncertain plant P can be any element of $\mathcal{P} = \{P(s, q) : q \in Q\}$, where q is a vector of m real uncertain parameters and Q is the set in $\mathbb{R}^m$ defined by all its possible values. The objective is to find some $G(s)$ and $F(s)$ such that

$$\left|\frac{M(i\omega) - P(i\omega)F(i\omega)}{1 + P(i\omega)G(i\omega)}\right| \leq |B(i\omega)| \tag{1}$$

holds for all $\omega > 0$ and for all $P \in \mathcal{P}$, where $B(s)$ defines the allowed tolerance. To obtain the QFT bounds associated to this specification, notice that (1) defines, at each frequency and for given values of M, G, P and B, a set of values of F in the complex plane. It can be easily found that such set is a circle with center $o = M/P$ and radius $r = |B/P + BG|$.

All points inside such circle are valid solutions F to (1) for a given plant $P \in \mathcal{P}$. Thus, there exists a common solution F for two plants $P_u, P_v \in \mathcal{P}$ if their corresponding circles intersect, that is, if

$$\frac{1}{|B|}\left|\frac{M}{P_u} - \frac{M}{P_v}\right| \leq \left|\frac{1}{P_u} + G\right| + \left|\frac{1}{P_v} + G\right|. \tag{2}$$

Equation (2) defines an ellipse in the complex plane of G. All points outside that ellipse are valid controllers G for the two given plants. Thus, if (2) is applied to any possible pair of plants in $\mathcal{P}$, a family of ellipses is found. The envelope of such family, once multiplied by the nominal plant, constitutes the QFT bound at the considered frequency.[35] If $L_0(i\omega)$ meets such bound, there exists a nonempty intersection $\mathcal{F}(\omega)$ among the circles with center o and radius r defined by each one of the plants in $\mathcal{P}$.

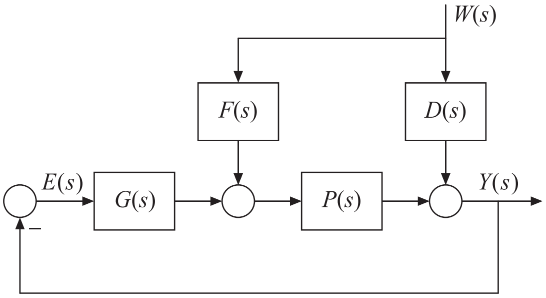


FIGURE 2 2DOF disturbance rejection scheme

These bounds are generated only for a discrete set of design frequencies Ω_G. In the loop-shaping stage, the controller $G(s)$ is tuned until the nominal open-loop response $L_0(i\omega) = G(i\omega)P_0(i\omega)$ meets the bounds, which means that (2) becomes true for all $\omega \in \Omega_G$.

The feedforward design stage consists in finding a stable rational function $F(s)$ such that $F(i\omega) \in \mathcal{F}(\omega)$ for all ω. Since this tuning involves both the magnitude and the phase of $F(s)$, it takes place on the Nichols plot.[31] Once such an $F(s)$ is found, (1) is guaranteed at the design frequencies for all $P \in \mathcal{P}$. But if the frequencies in Ω_G have been adequately chosen, we can expect the specification to be met for all $\omega > 0$.

When the model matching problem is framed in the canonical QFT structure with a prefilter preceding the loop, it is known as the tracking error problem.[34] Its bounds are also given by (2), and the circles of valid prefilters for each plant have center $o = M/T$ and radius $r = |B/T|$, where T denotes the system's complementary sensitivity function.

2.2 | Rejection of measurable disturbances

The procedure can be replicated with little change in the control scheme shown in Figure 2, in which a feedback controller $G(s)$ and a feedforward controller $F(s)$ fight the effects of a measurable disturbance on the system output.

In this case, constraining the error is the same as constraining the output, so the specification is given by

$$\left| \frac{D(i\omega) + P(i\omega) F(i\omega)}{1 + P(i\omega) G(i\omega)} \right| \leq |B(i\omega)| , \tag{3}$$

and the circle of valid feedforward elements F for each plant at a given frequency has now center $o = -D/P$ and radius $r = |B/P + BG|$.

Assuming that the uncertain parameters in q define both the plant dynamics $P(s)$ and the disturbance dynamics $D(s)$, there exists a region $\mathcal{F}$ of solutions to (3) at a given frequency and for all $q \in \mathcal{Q}$ if

$$\frac{1}{|B|} \left| \frac{D_u}{P_u} - \frac{D_v}{P_v} \right| \leq \left| \frac{1}{P_u} + G \right| + \left| \frac{1}{P_v} + G \right| , \tag{4}$$

where P_u, D_u, P_v, D_v are defined by any possible pair $q_u, q_v \in \mathcal{Q}$. Like in the tracking error case, this defines a family of ellipses whose envelope, once multiplied by the nominal plant, shapes the QFT bound to be met by $L_0(i\omega)$.[36]

Once again, the feedforward synthesis problem consists in finding a stable rational $F(s)$ such that $F(i\omega) \in \mathcal{F}(\omega)$ for all ω.

3 | AUTOMATED FEEDFORWARD SYNTHESIS

3.1 | Intersection algorithm

The first step toward an automatic tuning of feedforward elements for the tracking error problem and the measurable disturbance rejection problem is to find the intersection $\mathcal{F}$ of the circles defined by each $q \in \mathcal{Q}$. This intersection consists in a convex region enclosed by a set of connected circle arcs, which can be found using the algorithm presented in Table 1. Figure 3 shows an example in which the uncertain plant includes four elements.

TABLE 1 Algorithm to find the boundary of $\mathcal{F}$

Step	Operation
1	Remove all circles containing smaller ones.
2	Find the intersection points $\{p_{ij}\}$ among the remaining circles. Looking at the circles as clockwise paths, index i denotes the circle leaving the intersection region, and j defines the one entering it.
3	The subset of points $\{p_{ab}, p_{bc}, \dots\} \in \{p_{ij}\}$ such that $\lvert o_k - p_{ij} \rvert < r_k \ \forall k \neq i, k \neq j$, are the vertices of $\mathcal{F}$. Arrange them so that each j index is equal to the next point's i index.
4	Each pair of consecutive points (p_{ij}, p_{jk}) in the previous subset defines an arc $\widehat{p_{ij}p_{jk}}$ of the circle c_j with central angle (angle between two radii meeting at the center of a circle) θ_j.
5	The boundary of $\mathcal{F}$ is formed by all the arcs defined by consecutive elements in $\{p_{ab}, p_{bc}, \dots\}$. Thus, $\theta_b + \theta_c + \dots + \theta_a \leq 2\pi$.

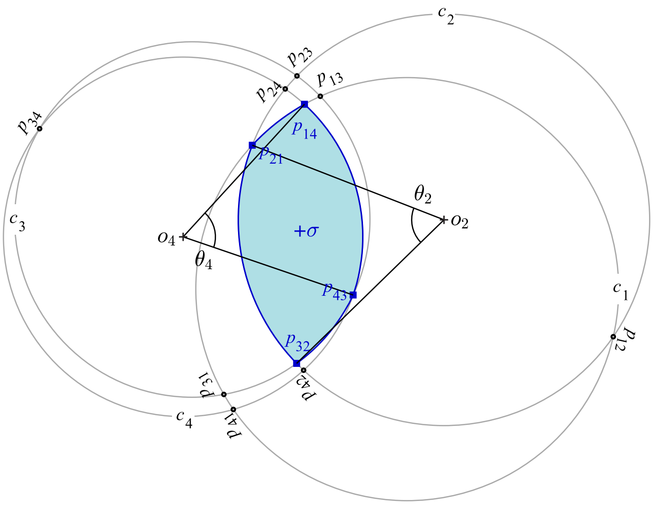


FIGURE 3 An example of region $\mathcal{F}$ given by the intersection of 4 circles at a given frequency ω, where σ denotes the position of the centroid and two of the central angles are shown as well [Colour figure can be viewed at wileyonlinelibrary.com]

3.2 | Automatic selection of design frequencies

Once the intersection region $\mathcal{F}$ is found at all frequencies, the most appropriate way to automatically derive the feedforward controller seems to consist in (i) finding the centroid of the intersection region at each design frequency, and (ii) using an existing algorithm to derive a stable rational $F(s)$ fitting them. However, such an $F(s)$ does not exist in general. The reason is that the frequency response of a stable rational $F(s)$, written as

$$F(i\omega) = \alpha(\omega) + i\beta(\omega) \tag{5}$$

must comply with Bode's real-imaginary parts relationship,[38] which links the imaginary part at certain frequency ω_k with the real part at all frequencies. More specifically, the relationship states that

$$\beta(\omega_k) = \frac{2\omega_k}{\pi} \int_0^\infty \frac{\alpha(\omega) - \alpha(\omega_k)}{\omega^2 - \omega_k^2} d\omega, \tag{6}$$

which is a weighted integral of $\alpha(\omega) - \alpha(\omega_k)$.

In general, the frequency response defined by the centroids does not meet such requirement, so it cannot be fit by a stable rational $F(s)$. As a consequence, the problem is to find a frequency response $F(i\omega)$ such that $F(i\omega) \in \mathcal{F}(\omega)$ for all ω, while obeying condition (6). If such a response is found, obtaining some $F(s)$ fitting it becomes relatively straightforward.

In the same way as for the loop-shaping stage, these conditions are translated to a discrete set of design frequencies Ω_F, in order to numerically compute Bode's integral (6). Whereas the frequencies used to design the feedback controller Ω_G are known, a first approximation to Ω_F would be $\Omega_F = \Omega_G$. But, depending on the curvature of the actual $\alpha(\omega)$, the discretization given by Ω_G may not provide an acceptable approximation to Bode's real-imaginary relationship.

To solve this problem, an algorithm to define a convenient set of design frequencies Ω_F for $F(s)$ is given next:

1. Ω_F is initialized as $\Omega_F = \Omega_G$.
2. A new logarithmically spaced frequency set Ω_{cont} is created covering the same range as Ω_G, but with a finer spacing among elements to capture the continuous behavior of the centroids. The algorithm to find intersections, described in Section 3.1, is applied to the whole set Ω_{cont}, and the centroids are computed as well. Behaviour of these centroids with respect to frequency acts as a good estimator of the difficulty of the feedforward design problem: when the allowed regions $\mathcal{F}$ are small—something that can be caused by great uncertainty, insufficient feedback or tight design specifications—the plot of the centroids tends to present sharper changes. In these cases, usually a high-order feedforward controller will be required. Conversely, when there is an excess of feedback for the given uncertainty and specifications, the centroids tend to evolve smoothly with frequency, and probably a low-order feedforward element will be enough.
3. The integral $\int_{\omega_i}^{\omega_{i+1}} \alpha(\omega)\, d\log(\omega)$ is computed for each interval $(\omega_i, \omega_{i+1}) \in \Omega_F$, using all the frequencies of Ω_{cont} inside it for the numerical integration. The result is compared with the value obtained disregarding the inner frequencies. If the difference is greater than a certain threshold, a new frequency is introduced at the geometric mean of the endpoints (ω_i, ω_{i+1}), and the comparison is repeated for each of the resulting pairs. Given that the Bode's integral (6) is just a weighted integral of the real part $\alpha(\omega)$, this procedure ensures that Ω_F has enough frequencies to compute it accurately.

As a result, once the set of design frequencies Ω_F is fixed, numerical integration methods can be used to approximate (6) by a linear relation between the imaginary part of $F(i\omega)$ at some frequency $\omega_k \in \Omega_F$ and the real parts at all frequencies in such set, that is,

$$\beta(\omega_k) \approx b_{k1}\alpha(\omega_1) + b_{k2}\alpha(\omega_2) + \ \dots\ + b_{kk}\alpha(\omega_k) + \ \dots\ b_{kn}\alpha(\omega_n). \tag{7}$$

Coefficients $b_{k1}, \dots, b_{kn}$ capture the numerical evaluation of the integral and depend only on the set of chosen frequencies. The Appendix shows how to obtain them using the trapezoidal rule.

3.3 | Feasibility study

Up to this point, and due to the fact that bounds of intersections $\mathcal{F}(\omega_k)$ are nonlinear, feedforward synthesis implies solving for the feasibility region of a nonlinear programming problem. But, if regions $\mathcal{F}(\omega_k)$ are approximated by convex polygons, the program becomes linear.

Vertices of complex polygons are obtained by splitting each arc $\widehat{p_{ij}p_{jk}}$ in the boundary into a number n_j of equally sized arcs with central angle $\frac{\theta_j}{n_j}$ smaller than certain parameter θ_{max} (see Figure 4). This way, the area missed by the polygonal approximation along each arc is

$$A_{e_j} = \frac{r_j^2}{2}\left[\theta_j - n_j \sin\left(\frac{\theta_j}{n_j}\right)\right], \tag{8}$$

and the overall relative error is simply the sum of the areas missed in each arc divided by the total area of the intersection:

$$e = \frac{\sum A_{e_j}}{A_{\mathcal{F}}}. \tag{9}$$

If the chosen θ_{max} leads to an error e above certain tolerance e_{max}, we must repeatedly increase in one unit the n_j of the arc in which such increment gives a lower e until $e < e_{max}$.

After the split, condition $F(i\omega_k) \in \mathcal{F}(\omega_k)$ is replaced by the set of n_k linear inequalities

$$\begin{bmatrix} a_1^k & b_1^k \\ a_2^k & b_2^k \\ \vdots & \vdots \\ a_{n_k}^k & b_{n_k}^k \end{bmatrix} \begin{bmatrix} \alpha(\omega_k) \\ \beta(\omega_k) \end{bmatrix} \leq \begin{bmatrix} c_1^k \\ c_2^k \\ \vdots \\ c_{n_k}^k \end{bmatrix}, \tag{10}$$

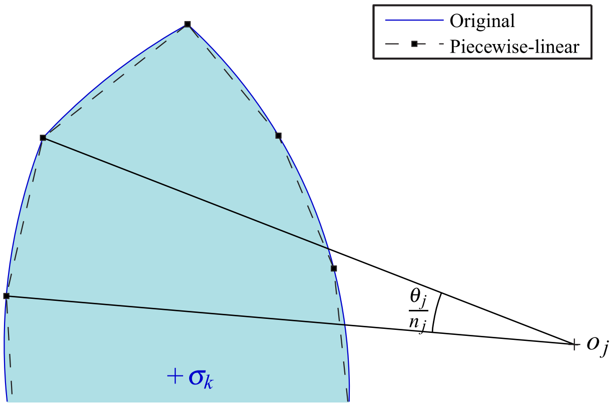


FIGURE 4 Approximation of $\mathcal{F}$ by a convex polygon [Colour figure can be viewed at wileyonlinelibrary.com]

where coefficients a_i^k, b_i^k and c_i^k are easily determined by the coordinates of each two consecutive vertices in the polygon (Figure 4).

Replacing (7) in (10), dependence on the imaginary parts is eliminated, leading to

$$\begin{bmatrix} b_1^k b_{k1} & b_1^k b_{k2} & \dots & a_1^k + b_1^k b_{kk} & \dots & b_1^k b_{kn} \\ b_2^k b_{k1} & b_2^k b_{k2} & \dots & a_2^k + b_2^k b_{kk} & \dots & b_2^k b_{kn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n_k}^k b_{k1} & b_{n_k}^k b_{k2} & \dots & a_{n_k}^k + b_{n_k}^k b_{kk} & \dots & b_{n_k}^k b_{kn} \end{bmatrix} \begin{bmatrix} \alpha(\omega_1) \\ \alpha(\omega_2) \\ \vdots \\ \alpha(\omega_k) \\ \vdots \\ \alpha(\omega_n) \end{bmatrix} \leq \begin{bmatrix} c_1^k \\ c_2^k \\ \vdots \\ c_{n_k}^k \end{bmatrix}, \tag{11}$$

or, in short notation,

$$\mathbf{A}^k \boldsymbol{\alpha} \leq \mathbf{c}^k. \tag{12}$$

Repeating the process for all $\omega_k \in \Omega_F$, we reach

$$\begin{bmatrix} \mathbf{A}^1 \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^n \end{bmatrix} \boldsymbol{\alpha} \leq \begin{bmatrix} \mathbf{c}^1 \\ \mathbf{c}^2 \\ \vdots \\ \mathbf{c}^n \end{bmatrix}, \tag{13}$$

which constitute the constraints of the linear programming problem to solve. Since any solution to the problem is acceptable, that is, we only care about feasibility, $\mathbf{0}^T\boldsymbol{\alpha}$ can be used as its cost function. If a compatible vector of real parts $\boldsymbol{\alpha}$ is found, the corresponding imaginary parts $\boldsymbol{\beta}$ are obtained by direct application of (7). On the other hand, we can say that if the linear programming problem is not feasible for a densely populated Ω_F and a convenient θ_{max}, there is no stable rational solution $F(s)$ to the feedforward synthesis problem. In those cases, we must go back to the loop-shaping stage and introduce more feedback on the set of frequencies at which the regions $\mathcal{F}$ are too small. If this is impossible to do due to feedback restrictions such as sensor noise amplification, then the specifications should be relaxed.

Additionally, linear constraints given by (13) can be used in a quadratic optimization problem to obtain the closest solution to the centroid of $\mathcal{F}(\omega_k)$, which could be considered the ideal response of the prefilter. However, this will only make a difference when $\mathcal{F}(\omega_k)$ is too big. In those cases, the designer should consider reducing the open-loop gain of the feedback controller, or alternatively, tightening the specification at ω_k.

The presented approach is similar to that of References 39 and 40, where convex optimization is used to synthesize a feedback controller meeting the QFT bounds. Apart from the fact that the present work deals with feedforward controllers, the main difference with respect to those approaches lies on the Bode relation used as a constraint. Here, the real-imaginary integral is preferred over the gain-phase integral, because it is not limited to minimum-phase systems, and also because it does not involve numerical integration of hyperbolic cotangents, which may not be very accurate at certain frequencies. Besides, in (13), only the real parts of the response must be found, which implies that the number of variables is reduced to a half with respect to the mentioned articles. Also, the Bode relation is implicitly considered, so the associated restrictions are removed from the problem.

The authors have also tested the implementation of cubic splines to approximate the frequency response of the function to shape, as in Reference 39. While in general it is true that the number of frequencies required to compute Bode's integral is lower, the use of splines does not prevent the need for a relatively large number of design frequencies in regions where sharp changes in the feedforward frequency response are required. For this reason, it is judged that the the benefits of splines are not enough to justify the increased complexity over the linear approximation.

3.4 | Transfer function fitting

Once a feasible solution is obtained, transfer functions fitting the points $\boldsymbol{\alpha} + i\boldsymbol{\beta}$ can be derived with the `fitfrd` algorithm, as given by Matlab or Octave packages. Although `fitfrd` by itself does not guarantee stable results, the fact that the points to fit meet Bode's relationship leads to a stable $F(s)$.

Solutions given by `fitfrd` evidently depend on the order n_F and the relative degree e_F of $F(s)$, but these parameters can be deduced in the following manner: the relative degree e_F is directly deduced considering the corresponding specification in the high frequency range, that is, when $L(i\omega) \to 0$. Depending on the problem considered, it is given by $e_F = e_M - e_P$ in the model matching problem (1) and by $e_F = e_D - e_P$ in the disturbance rejection problem (3). Regarding low frequency behavior, $F(s)$ presents as many derivatives as integrators in $P(s)$, $t_F = -t_P$, where t_P is the type number of $P(s)$.

Besides these conditions, high frequency phase angle asymptote—ϕ_∞—of the feasible $\boldsymbol{\alpha} + i\boldsymbol{\beta}$ fixes the number of zeros of $F(s)$ with positive real part, and consequently its minimum order must be at least

$$n_F \geq t_P + \frac{1}{2}\left(\frac{-\phi_\infty}{90^o} + e_F\right). \tag{14}$$

As a result, the number of independent parameters in $F(s)$ is $2 \cdot n_F + 1 - e_F - t_P$. But as the order n_F of $F(s)$ remains still open, optimum $F(s)$ will be the lowest order whose frequency response fulfills the original specification $F(i\omega) \in \mathcal{F}(\omega),\ \forall\omega \in \Omega_F$.

In this respect, if the solution of the problem relies only on `fitfrd`, there is a risk of overfitting $F(s)$, because the tendency is to increase the order n_F until all frequency responses satisfy the intersections.

Even though `fitfrd` has a very efficient algorithm, its objective is to obtain a transfer function optimally fitting a given frequency response, not to ensure that all points lie inside regions.

3.5 | Fine tuning of the transfer function

The transfer function obtained by `fitfrd` does not guarantee that its frequency response lies inside all the intersections $\mathcal{F}(\omega),\ \forall\omega \in \Omega_F$ but, in general, the obtained results are good enough from a practical point of view. However, it is possible to fine tune the feedforward term, $F(s)$, if, once validated the solution given by `fitfrd`, the results are judged not as good as they might be.

This final adjustment is obtained finding the minimum of a scalar function of several variables, starting at the initial estimate given by `fitfrd`, as is described in the next sections.

3.5.1 | Method to evaluate a point with respect to an intersection

The objective of the method is to define a scalar measure that evaluates the frequency response $F(i\omega_k)$ with respect to the intersection $\mathcal{F}(\omega_k)$. The main steps of the method are the following:

1. The relative vector $v_k = F(i\omega_k) - \sigma_k$, together with σ_k, defines the parametric line $(x + iy) = \sigma_k + \lambda v_k$ on the complex plane.

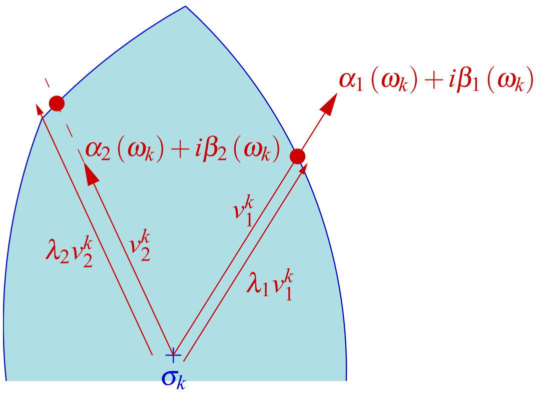

$\alpha_1(\omega_k) + i\beta_1(\omega_k)$
$\alpha_2(\omega_k) + i\beta_2(\omega_k)$
v_1^k
$\lambda_2 v_2^k$
v_2^k
$\lambda_1 v_1^k$
σ_k

FIGURE 5 Evaluation of two points with respect to the same intersection [Colour figure can be viewed at wileyonlinelibrary.com]

2. Compute the intersection ($\lambda > 0$) of the parametric line with the bound of the intersection area.
   - If $\lambda < 1$ the frequency response is outside the intersection, as $\alpha_1(\omega_k) + i\beta_1(\omega_k)$ in Figure 5.
   - If $\lambda > 1$ the frequency response is inside, as $\alpha_2(\omega_k) + i\beta_2(\omega_k)$ in Figure 5.
   - If $\lambda = 1$ the frequency response is exactly on the bound.
3. The output value is $\mu = 1/\lambda$, so $\mu \leq 1$ implies that the frequency response lies inside the intersection.

The measure $\mu_k = \mu(\mathcal{F}(\omega_k), F(i\omega_k))$ is independent of the size or shape of each intersection, and therefore is very appropriate for optimization purposes.

3.5.2 | Definition of the optimization problem

The objective function has been defined as a performance indicator of $F(i\omega)$ with respect to whole set $\mathcal{F}(\omega_k)$ as follows:

$$f_0 : \sum_{k=1}^{n} \mu_k^2 + (A-1) \sum_{\substack{k=1 \\ \mu_k > a}}^{n} \left(\frac{\mu_k - a}{1-a}\right)^4, \tag{15}$$

where $A = 100$ is the value of the function when the frequency response is on the bound and $a = 0.98$ is the threshold to activate the fourth order terms. This formulation penalizes especially frequency responses lying outside intersections, and with these values, one point lying outside its region at distance $\mu = 1.01$ penalizes more than 500 points at distance $\mu = 0.98$. Since when fine tuning is applied, most points $F(i\omega)$ already lie inside their corresponding $\mathcal{F}(\omega_k)$, and those which do not are close to the boundary, its net effect is simply to push slightly the responses inward.

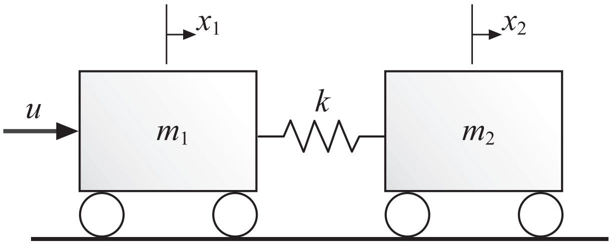
The optimization variables are the numerator and denominator coefficients of $F(i\omega)$, whose structure and initial values are taken from the results of `fitfrd`, which provides great reliability to the process. The algorithm used to find the minimum is the Nelder-Mead simplex algorithm that is provided by Matlab—`fminsearch`—or Octave—`fmins`—.

4 | DESIGN EXAMPLE: THE ACC'92 BENCHMARK PROBLEM

In this section, the feedforward design algorithm presented above is applied to the ACC'92 benchmark problem,[41] and in particular to the tracking challenge presented in Reference 42. The benchmark considers the two-mass-spring plant defined in Figure 6, where the force $u(t)$ is the control input and the position $x_2(t)$ of the mass m_2 is the measured output. The transfer function relating them is

$$P(s) = \frac{X_2(s)}{U(s)} = \frac{k}{m_1 s^2 \left[m_2 s^2 + \left(1 + \frac{m_2}{m_1}\right) k\right]}, \tag{16}$$

FIGURE 6 Two-mass-spring system



where $m_1 = 1$, $m_2 = 1$ and $k \in [0.5, 2]$. The objective is to define a feedback/feedforward controller for a unit-step output command tracking. Both the settling time and the overshoot must be minimized, and the control action must satisfy $|u| \leq 1$.

4.1 | Model matching solution

The present solution builds upon the work presented in Reference 43, where the problem was solved using the classical approach, but with the novelty of choosing high order Bessel filters as tolerances, in order to ensure a smooth control action. In this case, a model matching solution is pursued, in which the model to match is

$$M(s) = H_7(s, 1.59), \tag{17}$$

where $H_n(s, \omega_0)$ denotes a Bessel filter with order n and cutoff frequency ω_0. This filter was defined in Reference 43 as the one providing the fastest response which does not imply a violation of $|u| \leq 1$ by any member of the uncertain plant set.

The error tolerance $B(s)$ defining a region of acceptable responses around the ideal one is given by

$$B(s) = \frac{0.27s}{(4)(0.6)(0.7, 1.1)(0.4, 2)(0.4, 3)}, \tag{18}$$

where the short notation (p) for $(s/p + 1)$ and $(\zeta,\ \omega_n)$ for $\left(s^2/\omega_n^2 + 2\zeta s/\omega_n + 1\right)$ is used. The feedback controller bounds are generated using the method of Reference 36. These bounds are combined with stability bounds arising from the condition $|T(i\omega)| < 1.93$, which ensures a of robust phase margin of at least 30°. After proper loop-shaping—see Figure 7—, the obtained controller is

$$G(s) = \frac{0.7(1.4)(0.5, 0.42)(0.5, 1)(0.75, 7)(0.6, 24)}{(3.6)(20)^3(0.6, 1.2)(0.5, 32)^3}, \tag{19}$$

which presents feedback savings over the solution[43] at most frequencies.

4.1.1 | Feedforward design

For the feedforward design stage, the procedure described in Section 3 is applied. First, considering that the loop-shaping of the controller $G(s)$ has been done with a set Ω_G of 15 frequencies between 0.01 and 100 rad/s, a set Ω_{cont} of 1000 frequencies logarithmically spaced has been created, including the frequencies in Ω_G. Comparison among integrals has been done with a threshold of 1% with respect to the width of each interval. Figure 8 shows the outcome of the process for the present example: a total of 28 frequencies are added to the original set, resulting in a total of 43.

Second, matrices of the linear program have been constructed choosing a maximum angle of $\theta_{max} = 5°$ for the polygonal approximation of the intersection regions, and a feasible solution has been obtained, $\boldsymbol{\alpha} + i\boldsymbol{\beta}$.

Next, prior to calling to `fitfrd`, parameters of $F(s)$ have been defined: Relative degree, $e_F = e_M - e_P = 7 - 4 = 3$; Low frequency asymptote, s^2, or $t_F = -2$; High frequency phase angle asymptote of the feasible solution, $\phi_\infty = -630°$ (which implies the presence of two non-minimum phase zeros); Minimum order—following (14)—, $n_F \geq 7$. Choosing an order of 7 the obtained feedforward controller is

$$F_0(s) = \frac{1.1899 \cdot 10^{-3}(-0.9756, 0.0242)(-0.0948, 1.1779)}{(0.4863)(0.5123, 1.1263)(0.4124, 1.9722)(0.4259, 2.1434)}, \tag{20}$$

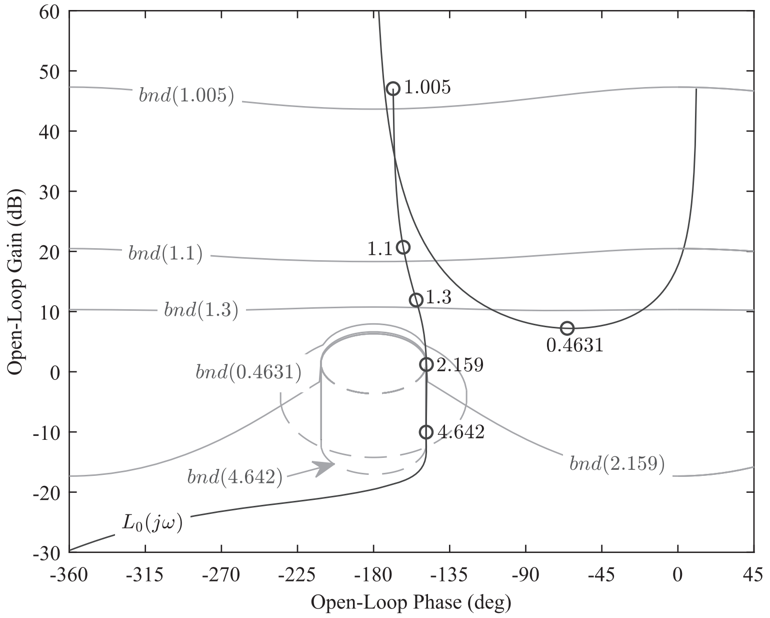


FIGURE 7 Loop-shaping of the controller $G(s)$

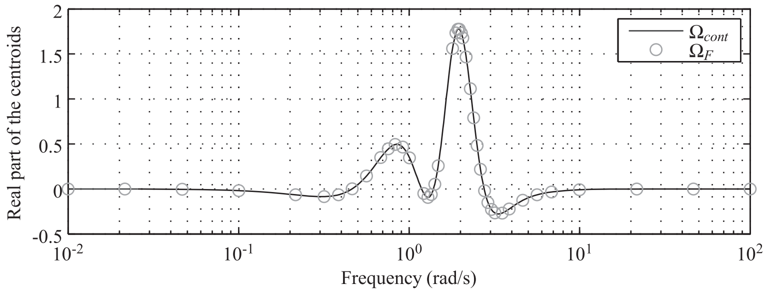


FIGURE 8 Set Ω_F of design frequencies for the feedforward

where the very low-frequency zeros arise as a consequence of the limited range of frequencies considered in the design. Therefore, they can be replaced by derivative elements to obtain

$$F(s) = \frac{2.0287s^2\,(-0.0948, 1.1779)}{(0.4863)\,(0.5123, 1.1263)\,(0.4124, 1.9722)\,(0.4259, 2.1434)}. \tag{21}$$

Even though $F(i\omega)$ fits remarkably to the feasible points—as Figure 9(A) shows—, there are three intersections that remain unfulfilled. In other words, the feedforward response meets the error specification at 93% (40/43) of the frequencies, as portrayed by Figure 9(B).

If parameters of the feedforward obtained by `fitfrd` (21) are taken as a starting for the nonlinear optimization, the final feedforward fulfills all intersections and the obtained transfer function is the following:

$$F_{final}(s) = \frac{1.5850s^2\,(-0.1340, 1.1943)}{(2.4567)\,(0.6707, 0.7251)\,(0.3449, 1.3798)\,(0.2335, 2.2836)}. \tag{22}$$

Figure 10(A) shows that the final feedforward has a frequency response that lies inside all intersections for the whole spectrum, as $\mu < 1$ for all frequencies. Equivalently Figure 10(B) presents tracking error frequency responses for 10 values of k covering the uncertainty range.

Finally, the time-domain results are shown in Figure 11. Notice that the responses of $M(s) + B(s)$ and $M(s) - B(s)$ define time-domain tolerances that are approximately, but not exactly respected by the system. The slightly loose translation of time-domain restrictions into frequency domain tolerances is shared by all QFT techniques. In fact, the use of specifications capturing magnitude and phase, as in this work, contribute to tighten the ties between both worlds. On the other hand, the fact that the control action is well inside its tolerances suggests that a faster step response could have been obtained. To do so, the process followed in Reference 43 to obtain a model to match should be revised.

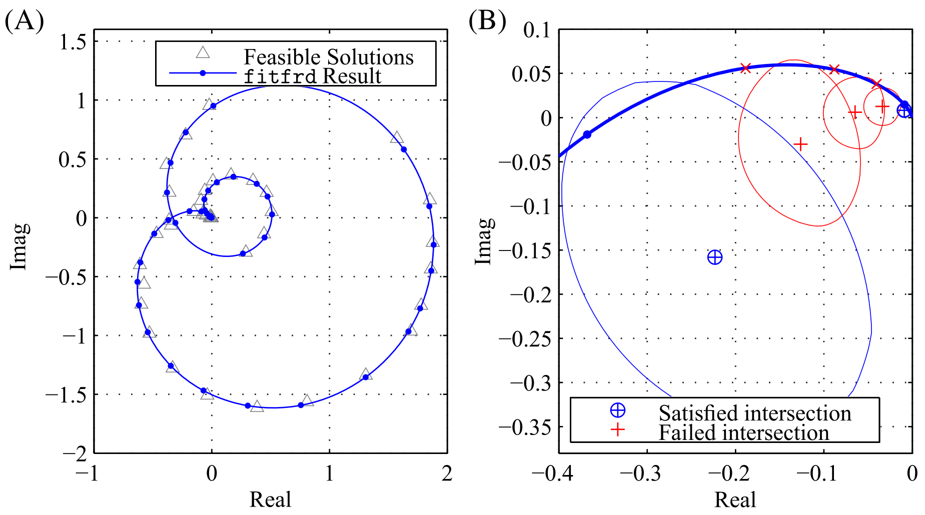


FIGURE 9 Validation of `fitfrd`: (A) $F(i\omega)$ against the feasible solution. (B) Detail of the failed three intersections [Colour figure can be viewed at wileyonlinelibrary.com]

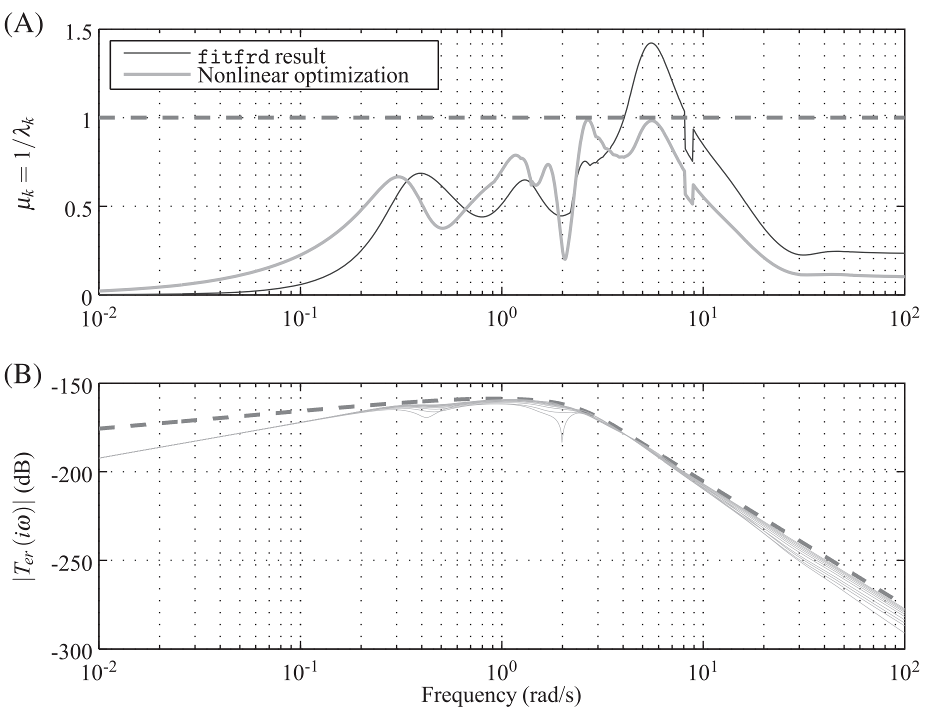


FIGURE 10 Effect of the fine tuning of $F(s)$: (A) Lowering of μ at all frequencies. (B) Final frequency response, Dashed black: tracking error tolerance. Solid gray: actual frequency responses

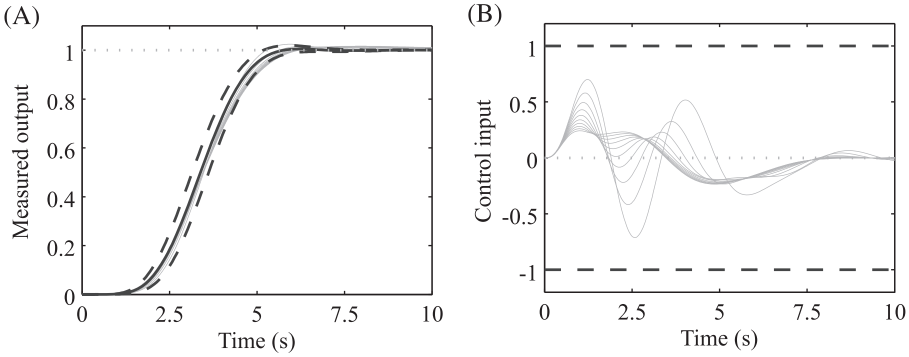


FIGURE 11 Time domain results for a unit step reference: (A) output, (B) control input. Solid black: ideal response. Dashed black: tolerances. Solid gray: responses for 10 values of k covering the uncertainty range

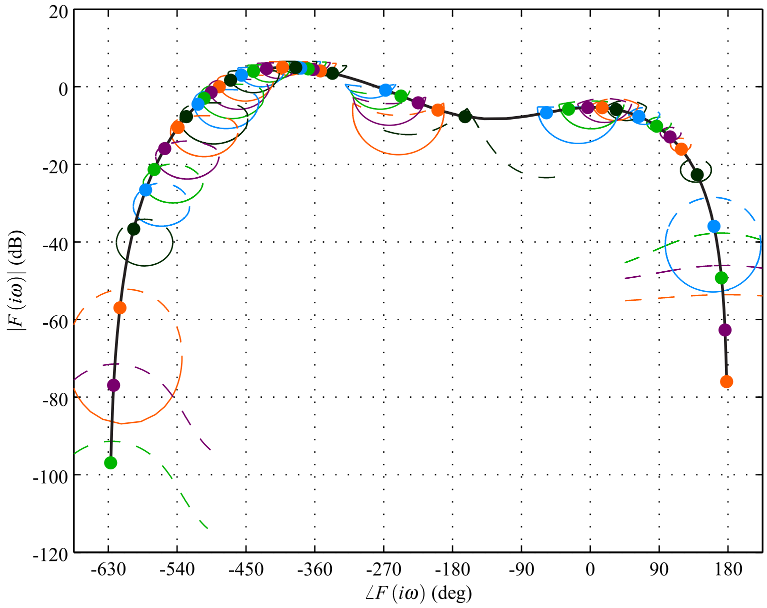


FIGURE 12 Feedforward shaping: regions to match and $F(i\omega)$ [Colour figure can be viewed at wileyonlinelibrary.com]

The whole design process has been implemented in Matlab© 2020a and the execution times on a i7-9700 processor are the following: Automatic selection of design frequencies, 0.01 s; Solution to the linear feasibility problem, 0.9 s; Transfer function fitting, 0.36 s; Nonlinear optimization, 8.4 s.

The most eloquent illustration of the benefits of the proposed technique is presented in Figure 12. The Figure shows the feedforward bounds for the problem, generated as described in References 31 and 36, as well as the frequency response of (21), meeting all of them. It is easy to see how hard this would be to achieve if $F(s)$ had to be manually shaped.

5 | CONCLUSION

This work has presented an automatic feedforward shaping algorithm for QFT problems involving restrictions both on the magnitude and the phase of system responses, where manual synthesis is harder than in traditional QFT.

It has been shown that the feedforward frequency response must lie inside the intersections of families of circles on the complex plane. Besides, such response must obey Bode's real-imaginary integral to ensure that the feedforward controller is rational and stable. The integral has been discretized for a given set of design frequencies and, as a result, imaginary parts of any realizable prefilter are written as linear functions of real ones, with coefficients explicitly defined. Under these conditions, feedforward synthesis is still a nonlinear programming problem. But if the intersections are approximated by convex polygons, the problem becomes linear.

Fitting to the solution of the feasibility problem does not guarantee that its frequency response lies inside all the intersections but, in general, the obtained results are good enough from a practical point of view. However, if once validated the solution given by `fitfrd`, the results are judged not as good as they might be, a second optimization step can provide a fine tuning of the transfer function, in order to improve the fulfillment of intersections, no matter their area or shape.

All technical aspects of this method have been given in detail, and practical remarks of the problem formulation have also been discussed.

The proposed method has been tested on the ACC'92 benchmark problem. Feedforward controllers that have been automatically obtained meet the design requirements, validating the developed algorithms.

ACKNOWLEDGEMENTS

This work has been carried out at the Dynamical Systems and Control Group of the Public University of Navarre (UPNA), and at the Intelligent Systems & Energy research group of the University of the Basque Country (UPV/EHU), and has been supported by the IT1256-19 research grant of the Basque Government and by the DPI2015-64985-R research grant of the Spanish Ministry of Economy and Competitiveness.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Jorge Elso https://orcid.org/0000-0003-3438-3524

REFERENCES

1. Horowitz I. Survey of quantitative feedback theory (QFT). *Int J Robust Nonlinear Control*. 2001;11(10):887-921. https://doi.org/10.1002/rnc.637.
2. García-Sanz M. *Robust Control Engineering: Practical QFT Solutions*. Boca Raton, FL: CRC Press; 2017.
3. Houpis CH, Rasmussen SJ, García-Sanz M. *Quantitative Feedback Theory. Fundamentals and Applications*. Boca Raton, FL: CRC Press; 2006.
4. Yaniv O. *Quantitative Feedback Design of Linear and Nonlinear Control Systems*. Hingham, MA: Kluwer Academic Publishers; 1999.
5. Horowitz I. Quantitative synthesis of uncertain multiple input-output feedback system. *Int J Control*. 1979;30(1):81-106. https://doi.org/10.1080/00207177908922759.
6. Gera A, Horowitz I. Optimization of the loop transfer function. *Int J Control*. 1980;31(2):389-398. https://doi.org/10.1080/00207178008961049.
7. Nataraj PSV, Tharewal S. Automatic design of QFT prefilter using interval analysis. Paper presented at: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design. Taipei, Taiwan; 2004:156-160; IEEE Control Systems Society.
8. Patil MD, Nataraj PSV. QFT prefilter design for multivariable systems using interval constraint satisfaction technique. *J Control Theory Appl*. 2013;11(4):529-537.
9. Nataraj PSV, Makwana D. Automated synthesis of fixed structure QFT prefilter using piecewise linear approximation based linear programming optimization techniques. *IFAC-PapersOnLine*. 2016;49(1):349-354. https://doi.org/10.1016/j.ifacol.2016.03.078.
10. Bailey FN, Hui CH. Loop gain-phase shaping for single-input-single-output robust controllers. *IEEE Control Syst*. 1991;11(1):93-101. https://doi.org/10.1109/37.103363.
11. Thompson DF, Nwokah ODI. Analytic loop shaping methods in quantitative feedback theory. *J Dyn Syst Measur Control Trans ASME*. 1994;116(2):169-177. https://doi.org/10.1115/1.2899208.
12. Chait Y, Chen Q, Hollot CV. Automatic loop-shaping of QFT controllers via linear programming. *J Dyn Syst Measur Control Trans ASME*. 1999;121(3):351-357. https://doi.org/10.1115/1.2802481.
13. Chen WH, Ballance DJ, Feng W, Li Y. Genetic algorithm enabled computer-automated design of QFT control systems. Paper presented at: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design. Kohala Coast, HI; 1999:492-497; IEEE Control Systems Society.
14. Croulard V, Godoy E, Boichot J. QFT controller optimization for automatic design. Paper presented at: Proceedings of the IEEE Conference on Decision and Control. Sydney, NSW, Australia; Vol. 5, 2000:4735-4740. https://doi.org/10.1109/CDC.2001.914676.
15. Yaniv O, Nagurka M. Automatic loop shaping of structured controllers satisfying QFT performance. *J Dyn Syst Measur Control Trans ASME*. 2005;127(3):472-477. https://doi.org/10.1115/1.1985441.
16. Qingwei W, Zhenghua L, Lianjie E. Automatic design of QFT robust controller based on genetic algorithms. Paper presented at: Proceedings of the World Congress on Intelligent Control and Automation (WCICA). Dalian, China; Vol. 1, 2006:2272-2276.
17. Nataraj PSV, Kubal N. Automatic loop shaping in QFT using hybrid optimization and constraint propagation techniques. *Int J Robust Nonlinear Control*. 2007;17(2-3):251-264. https://doi.org/10.1002/rnc.1085.
18. Nataraj PSV, Tharewal S. An interval analysis algorithm for automated controller synthesis in QFT designs. *J Dyn Syst Measur Control Trans ASME*. 2007;129(3):311-321. https://doi.org/10.1115/1.2397147.
19. Meng L, Xue D. Automatic loop shaping in fractional-order QFT controllers using particle swarm optimizations. Paper presented at: Proceedings of the 2009 IEEE International Conference on Control and Automation, ICCA 2009. Christchurch, New Zealand; 2009:2182-2187; IEEE Control Systems Society.
20. Molins C, García-Sanz M. Automatic loop-shaping of QFT robust controllers. Paper presented at: Proceedings of the IEEE National Aerospace and Electronics Conference. Dayton, Ohio; 2009:103-110; IEEE Dayton Section.
21. García-Sanz M, Molins C. Automatic loop shaping of QFT robust controllers with multi-objective specifications via nonlinear quadratic inequalities. Paper presented at: Proceedings of the IEEE 2010 National Aerospace and Electronics Conference, NAECON 2010. Dayton, Ohio; 2010:348-353; IEEE Dayton Section.
22. Comasòlivas R, Escobet T, Quevedo J. Automatic design of robust PID controllers based on QFT specifications. Paper presented at: IFAC Proceedings Volumes. Special Issue 2nd IFAC Conference on Advances in PID Control. Brescia, Italy; Vol. 45, 2012:715-720; The International Federation of Automatic Control.
23. Patil MD, Nataraj PSV. Automated synthesis of multivariable QFT controller using interval constraint satisfaction technique. *J Process Control*. 2012;22(4):751-765. https://doi.org/10.1016/j.jprocont.2012.02.006.

24. Cervera J, Baños A. QFT loop shaping with fractional order complex pole-based terms. *J Vib Control*. 2013;19(2):294-308. https://doi.org/10.1177/1077546311431271.
25. Jeyasenthil R, Purohit H, Nataraj PSV. Automatic loop shaping in MIMO QFT using interval consistency based optimization technique. Paper presented at: Proceedings of the IEEE International Symposium on Industrial Electronics. Istanbul, Turkey; 2014:75-80; IEEE Control Systems Society.
26. Makwana D, Nataraj PSV. Automated synthesis of fixed structure QFT controller using piecewise linear approximation based linear programming optimization techniques. Paper presented at: Proceedings of the 2015 International Conference on Industrial Instrumentation and Control, ICIC 2015. Pune, India; 2015:1597-1602.
27. Mercader P, Astrom KJ, Baños A, Hagglund T. Robust PID design based on QFT and convex-concave optimization. *IEEE Trans Control Syst Technol*. 2017;25(2):441-452. https://doi.org/10.1109/TCST.2016.2562581.
28. Purohit H, Goldsztejn A, Jermann C, et al. Simultaneous automated design of structured QFT controller and prefilter using nonlinear programming. *Int J Robust Nonlinear Control*. 2017;27(15):2529-2548. https://doi.org/10.1002/rnc.3695.
29. Moreno JC, Baños A, Berenguel M. The design of QFT robust compensators with magnitude and phase specifications. *Mathematical Problems in Engineering*. 2010;2010:105143. https://doi.org/10.1155/2010/105143.
30. Eitelberg E. Quantitative feedback design for tracking error tolerance. *Automatica*. 2000;36(2):319-326. https://doi.org/10.1016/S0005-1098(99)00149-1.
31. Boje E. Pre-filter design for tracking error specifications in QFT. *Int J Robust Nonlinear Control*. 2003;13(7):637-642. https://doi.org/10.1002/rnc.829.
32. Boje E. Multivariable quantitative feedback design for tracking error specifications. *Automatica*. 2002;38(1):131-138. https://doi.org/10.1016/S0005-1098(01)00177-7.
33. Mahdi Alavi SM, Khaki-Sedigh A, Labibi B, Hayes MJ. Improved multivariable quantitative feedback design for tracking error specifications. *IET Control Theory Appl*. 2007;1(4):1046-1053. https://doi.org/10.1049/iet-cta:20060378.
34. Elso J, Gil-Martinez M, García-Sanz M. Nonconservative QFT bounds for tracking error specifications. *Int J Robust Nonlinear Control*. 2012;22(18):2014-2025. https://doi.org/10.1002/rnc.1804.
35. Elso J, Gil-Martinez M, García-Sanz M. A quantitative feedback solution to the multivariable tracking error problem. *Int J Robust Nonlinear Control*. 2014;24(16):2331-2346. https://doi.org/10.1002/rnc.2991.
36. Elso J, Gil-Martinez M, García-Sanz M. Quantitative feedback-feedforward control for model matching and disturbance rejection. *IET Control Theory Appl*. 2013;7(6):894-900. https://doi.org/10.1049/iet-cta.2012.0596.
37. Elso J, Gil-Martinez M, García-Sanz M. Quantitative feedback control for multivariable model matching and disturbance rejection. *Int J Robust Nonlinear Control*. 2017;27(1):121-134. https://doi.org/10.1002/rnc.3563.
38. Seron MM, Braslavsky JH, Goodwin GC. *Fundamental Limitations in Filtering and Control*. London, UK: Springer-Verlag; 1997.
39. Unstead PA, Macleod IM. Synthesis of continuous-time minimum-phase frequency response specifications. *Int J Control*. 2001;74(6):571-585. https://doi.org/10.1080/00207170010017833.
40. Bryant GF, Halikias GD. Optimal loop-shaping for systems with large parameter uncertainty via linear programming. *Int J Control*. 1995;62(3):556-568. https://doi.org/10.1080/00207179508921556.
41. Wie B., Bernstein D. S.. A benchmark problem for robust control design. Paper presented at: Proceedings of the American Control Conference, IFAC; 1990:961-962; The International Federation of Automatic Control.
42. Wie B, Bernstein DS. Benchmark problems for robust control design. *J Guid Control Dyn*. 1992;15(5):1057-1059. https://doi.org/10.2514/3.20949.
43. Elso J, Miquelez I, Ostolaza JX. A QFT solution to the ACC benchmark tracking problem. Paper presented at: Proceedings of the 7th International Conference on Systems and Control, ICSC 2018. Valencia, Spain; 2018:283-287; IEEE Control Systems Society.

How to cite this article: Elso J, Ostolaza JX. Automatic synthesis of feedforward elements in quantitative feedback theory. *Int J Robust Nonlinear Control*. 2021;31:5525–5540. https://doi.org/10.1002/rnc.5515

APPENDIX

Let us begin by dividing the integral in (6) into

$$\int_0^{\infty} f(\omega)\, d\omega = \int_0^{\omega_1} f(\omega)\, d\omega + \int_{\omega_1}^{\omega_n} \omega f(\omega) \frac{d\omega}{\omega} + \int_{\omega_n}^{\infty} f(\omega)\, d\omega \tag{A1}$$

where

$$f(\omega) = \frac{\alpha(\omega) - \alpha(\omega_k)}{\omega^2 - \omega_k^2}. \tag{A2}$$

If ω_1 is a low enough frequency and $F(s)$ has no integrators—as in the case of feedforward elements—, we can assume that $\alpha(\omega) \approx \alpha(\omega_1)$ for all $\omega \in [0, \omega_1]$. Therefore, the first term in (A1) can be approximated by

$$\int_0^{\omega_1} f(\omega)\, d\omega \approx [\alpha(\omega_1) - \alpha(\omega_k)] \int_0^{\omega_1} \frac{d\omega}{\omega^2 - \omega_k^2}. \tag{A3}$$

Similarly, if ω_n is high enough and $F(s)$ is proper, we can assume that $\alpha(\omega) \approx \alpha(\omega_n)$ for all $\omega \in [\omega_n, \infty]$. As a consequence, the third term in (A1) can be approximated by

$$\int_{\omega_n}^{\infty} f(\omega)\, d\omega \approx [\alpha(\omega_n) - \alpha(\omega_k)] \int_{\omega_n}^{\infty} \frac{d\omega}{\omega^2 - \omega_k^2}. \tag{A4}$$

In both cases, we can use the result

$$\int_a^b \frac{1}{\omega^2 - \omega_k^2} d\omega = \frac{1}{2\omega_k}\left[\ln\left(\frac{b - \omega_k}{a - \omega_k}\right) + \ln\left(\frac{a + \omega_k}{b + \omega_k}\right)\right], \tag{A5}$$

which is defined as long as $\omega_k < a$ or $\omega_k > b$. Thus, we find that for all $\omega_k \in \Omega_F$

$$\int_0^{\omega_1} f(\omega)\, d\omega \approx \begin{cases} 0 & \omega_k = \omega_1 \\ \frac{\alpha(\omega_1) - \alpha(\omega_k)}{2\omega_k} \ln\left(\frac{\omega_k - \omega_1}{\omega_k + \omega_1}\right) & \omega_k > \omega_1 \end{cases}, \tag{A6}$$

and

$$\int_{\omega_n}^{\infty} f(\omega)\, d\omega \approx \begin{cases} \frac{\alpha(\omega_n) - \alpha(\omega_k)}{2\omega_k} \ln\left(\frac{\omega_n + \omega_k}{\omega_n - \omega_k}\right) & \omega_k < \omega_n \\ 0 & \omega_k = \omega_n \end{cases}. \tag{A7}$$

On the other hand, the second term in (A1) can be approximated by

$$\int_{\omega_1}^{\omega_n} \omega f(\omega) \frac{d\omega}{\omega} \approx \sum_{i=1}^{n-1} \frac{\omega_i f(\omega_i) + \omega_{i+1} f(\omega_{i+1})}{2} [\ln \omega_{i+1} - \ln \omega_i], \tag{A8}$$

which can be expanded and rearranged to yield

$$\int_{\omega_1}^{\omega_n} \omega f(\omega) \frac{d\omega}{\omega} \approx \frac{1}{2} \sum_{i=1}^{n} a_{ki} \alpha(\omega_i), \tag{A9}$$

where

$$a_{k1} = \omega_1 \frac{\ln \omega_2 - \ln \omega_1}{\omega_1^2 - \omega_k^2},$$

$$a_{kn} = \omega_n \frac{\ln \omega_n - \ln \omega_{n-1}}{\omega_n^2 - \omega_k^2},$$

$$a_{ki} = \omega_i \frac{\ln \omega_{i+1} - \ln \omega_{i-1}}{\omega_i^2 - \omega_k^2} \quad \text{for } i > 1, i < n, i \neq k,$$

and

$$a_{kk} = -\sum_{i \neq k} a_{ki}.$$

Taking (A6), (A7), and (A9) to (A1), and then to (6), we obtain the linear approximation

$$\beta\left(\omega_k\right) \approx \sum_{i=1}^{n} b_{ki}\alpha\left(\omega_i\right),$$

where the coefficients are defined by

$$\begin{aligned}
b_{k1} &= \frac{1}{\pi}\frac{\ln\omega_2 - \ln\omega_1}{\frac{\omega_1}{\omega_k} - \frac{\omega_k}{\omega_1}} + \frac{1}{\pi}\ln\left(\frac{\omega_k - \omega_1}{\omega_k + \omega_1}\right) \quad \text{for} \;\; k > 1,\\
b_{kn} &= \frac{1}{\pi}\frac{\ln\omega_n - \ln\omega_{n-1}}{\frac{\omega_n}{\omega_k} - \frac{\omega_k}{\omega_n}} + \frac{1}{\pi}\ln\left(\frac{\omega_n + \omega_k}{\omega_n - \omega_k}\right) \quad \text{for} \;\; k < n,\\
b_{ki} &= \frac{1}{\pi}\frac{\ln\omega_{i+1} - \ln\omega_{i-1}}{\frac{\omega_i}{\omega_k} - \frac{\omega_k}{\omega_i}} \quad \text{for} \;\; i > 1, i < n, i \neq k,
\end{aligned}$$

and

$$b_{kk} = -\sum_{i\neq k} b_{ki}.$$