https://doi.org/10.17979/spudc.9788497498081.0107

Selection of the Adequate Algorithm to Generate Plant Templates for Robust Control

José Manuel Díaz
Dpto. de Informática y Automática. ETSI Informática. UNED. 28040 Madrid. Spain, josema@dia.uned.es

Sebastián Dormido
Dpto. de Informática y Automática. ETSI Informática. UNED. 28040 Madrid. Spain, sdormido@dia.uned.es

Joaquín Aranda
Dpto. de Informática y Automática. ETSI Informática. UNED. 28040 Madrid. Spain, jaranda@dia.uned.es

Rocío Muñoz
Dpto. de Informática y Automática. ETSI Informática. UNED. 28040 Madrid. Spain, rmunoz@dia.uned.es

Abstract

The calculation of templates associated with plant uncertainty is one of the first steps in some control robust methodologies. This paper presents a comparative study of three classic algorithms proposed in the late 1980s and early 1990s for calculating plant templates. The results of this study can help the designer to choose the right template computation algorithm depending on the kind of plant.

Keywords: robust control, control education, frequency responses, QFT, templates, uncertain linear systems.

1 INTRODUCTION

A template Γ(ωi) (or value sets) is the representation in the Nichols diagram (or in the Nyquist diagram) of plant uncertainty at a given frequency ωi. Templates are used in different robust control techniques, like Quantitative Feedback Theory (QFT) created by I.M. Horowitz [5].

In QFT methodology [6]-[9], the templates calculation is one of the first steps. The template boundary points are the most important, because they are the only ones used to calculate the bounds associated with the design specifications. In the loop-shaping stage, with these bounds it is possible to know where the open-loop transfer function must be placed in the Nichols chart in order to fulfil the specifications. It is very important to obtain a good approximation of the templates in order to only add the necessary amount of conservatism in the controller designed.

Conceptually, the simplest procedure for calculating templates is the grid method. This is a brute-force method, which consists of selecting values inside each parameter uncertainty range, and evaluating the plant for these values at the desired frequency. While the procedure is simple, it has several disadvantages [1].

For around thirty years, the grid method was the only technique available for calculating templates. In the late 1980s and early 1990s, several algorithms were developed to obtain plant templates with special parametric dependencies, such as interval plants, and plants with affine parametric uncertainty. These kinds of plants frequently appear in control problems. In our opinion, the three most useful algorithms are: the algorithm proposed by Bailey & Hui [1], the algorithm proposed by Fu [4], and the method proposed by Barlett et al. [2]. This last method can easily be formulated as an algorithm. In this paper we refer to it as the "*Kharitonov segment algorithm*".

This paper presents the results of a comparative study between the Bailey & Hui algorithm (BHA), the Fu algorithm (FUA) and the Kharitonov segment algorithm (KSA). It has focused on useful features from a designer's point of view, such as the conservatism of the template boundary calculated, the algorithm yield, and the template calculation time. The results of this study can help the designer to choose the right template computation algorithm depending on the kind of plant.

The paper is organized as follows. Section 2 describes the problem statement. Section 3 briefly describes the main features of the algorithms considered in the comparative study. Section 4 and Section 5 describe the comparative study and the results obtained, respectively. Finally, Section 6 gives some concluding remarks.

2 PROBLEM STATEMENT

Let the uncertain parameters vector v be considered:

$$v = \left[v_1, v_2, ..., v_L \right] \qquad (1)$$

where elements v_j form part of the hyper-rectangle V:

$$V = \left\{ v_j \middle| v_j^- \le v_j \le v_j^+ \quad j = 1,...,L \right\} \qquad (2)$$

Let the family or set $\mathcal{P}_1=\{P_1\}$ of transfer functions be:

$$\mathcal{P}_1 = \left\{ P_1 \mid P_1(s;v) = \frac{b(s;v)}{a(s;v)} = \frac{\sum_{k=0}^{m} b_k(v) \cdot s^k}{\sum_{k=0}^{n} a_k(v) \cdot s^k} \right\} \qquad (3)$$

where the coefficients $b_k(v)$ and $a_k(v)$ are linear combinations of the uncertain parameters, i.e.,

$$b_k(v) = \beta_{k0} + \sum_{j=1}^{L} \beta_{kj} \cdot v_j \qquad (4)$$

$$a_k(v) = \alpha_{k0} + \sum_{j=1}^{L} \alpha_{kj} \cdot v_j \qquad (5)$$

where β_{kj} and α_{kj} j=0,...,L are real constants. Therefore, $\mathcal{P}_1$ is the family or set of transfer functions with affine parametric uncertainty.

A subset of $\mathcal{P}_1$ is the family of transfer functions $\mathcal{P}_2=\{P_2\}$, whose numerator uncertain parameters are independent of the denominator uncertain parameters:

$$\mathcal{P}_2 = \left\{ P_2 \mid P_2(s;q,r) = \frac{b(s;q)}{a(s;r)} = \frac{\sum_{k=0}^{m} b_k(q) \cdot s^k}{\sum_{k=0}^{n} a_k(r) \cdot s^k} \right\} \qquad (6)$$

The elements of the uncertain parameters vectors q and r form part of the hyper-rectangles Q and R, respectively, which are defined thus:

$$Q = \left\{ q_j \middle| q_j^- \le q_j \le q_j^+ \quad j = 1,...,L_n \right\} \qquad (7)$$

$$R = \left\{ r_j \middle| r_j^- \le r_j \le r_j^+ \quad j = 1,...,L_d \right\} \qquad (8)$$

Therefore, v can be defined as

$$v = \left[q_1,...,q_{L_n}, r_1,...,r_{L_d} \right] \qquad (9)$$

with

$$L = L_n + L_d \qquad (10)$$

Likewise, the coefficients $b_k(q)$ and $a_k(r)$ must have a similar structure to (4) and (5).

Another subset of $\mathcal{P}_1$ is the family of transfer functions $\mathcal{P}_3=\{P_3\}$:

$$\mathcal{P}_3 = \left\{ P_3 \mid P_3(s;v) = \frac{b(s;v)}{a(s;v)} = \frac{\sum_{k=0}^{m} b_k \cdot s^k}{\sum_{k=0}^{n} a_k \cdot s^k} \right\} \qquad (11)$$

where the numerator and denominator coefficients, $b_k \in [b_k^-, b_k^+]$ and $b_k \in [a_k^-, a_k^+]$, respectively, can be either one element of $v \in V$ or a constant value.

The intersection of the set $\mathcal{P}_2$ and $\mathcal{P}_3$ gives the family of transfer functions $\mathcal{P}_4=\{P_4\}$:

$$\mathcal{P}_4 = \left\{ P_4 \mid P_4(s;q,r) = \frac{b(s;q)}{a(s;r)} = \frac{\sum_{k=0}^{m} b_k \cdot s^k}{\sum_{k=0}^{n} a_k \cdot s^k} \right\} \qquad (12)$$

where a numerator coefficient b_k can be either one element of $q \in Q$ or a constant value. Analogously, a numerator coefficient a_k can be either one element of $r \in R$ or a constant value. The family of plants $\mathcal{P}_3$ and $\mathcal{P}_4$ are commonly named "*Interval plants*" in the literature.

The existence of uncertainty in the transfer function at a certain frequency ω_i caused its representation in the Nichols chart (or in the complex plane) to be a region and not just a point. This region is commonly named template (or value sets) $\Gamma(\omega_i)$,

$$\Gamma(\omega_i) = \left\{ P(j\omega_i; v) \middle| v \in V \right\} \qquad (13)$$

Given a transfer function belonging to $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, or $\mathcal{P}_4$, and a fixed frequency ω_i, the problem consists of determining the $\Gamma(\omega_i)$ boundary.

3 ALGORITHMS CONSIDERED

3.1 BAILEY & HUI ALGORITHM

The algorithm proposed by Bailey & Hui in 1989 [1] is based in geometric considerations. It can directly obtain the template boundary for a transfer function $P(s;v)$ in the form (4). The number of template boundary points N_c obtained with this algorithm can be determined using the expression

$$N_c = 2\cdot N_b \tag{14}$$

where N_b is an algorithm parameter associated with the number of upper (or lower) template boundary points.

3.2 FU ALGORITHM

The algorithm proposed by Fu in 1990 [4] is also based in geometric considerations. It works with the transfer function $P(s;v)$ in the form (3) with an uncertain parameter vector $v\in V$ of dimension L. It considers that $\Gamma(\omega_i)$ consists of $2^{L-1}\cdot L$ arcs or line segment, and calculates N_f points for each one of these arcs. Therefore, the total number of points calculated is

$$N_t = N_f\cdot 2^{L-1}\cdot L \tag{15}$$

Once the N_t points of $\Gamma(\omega_i)$ are obtained, the N_c points belonging to the boundary must be selected. This task can be done either manually or using some algorithm to recognise the template boundary.

3.3 KHARITONOV SEGMENT ALGORTHM

Barlett et al. in 1993 [2] proposed a method to calculate the templates of a transfer function $P(s;v)$ in the form (11). This method can be easily formulated as an algorithm. The algorithm considers that $\Gamma(\omega_i)$ consists of 32 arcs that are the Kharitonov segments derived from the transfer function. This is why we named this algorithm "*Kharitonov segment algorithm*". Moreover, it calculates N_k points for each one of these arcs. Therefore, the total number of points calculated is

$$N_t = 32\cdot N_k \tag{16}$$

Once the N_t points of $\Gamma(\omega_i)$ are obtained, it is necessary to use an additional algorithm to recognise the N_c points belonging to the boundary.

4 DESCRIPTION OF THE COMPARATIVE STUDY

In order to speed up and systematise the comparative study of the BHA, FUA and KSA, a matricial description of the transfer function coefficients was done. For the family of transfer functions $\mathcal{P}_1$, the numerator coefficients (4) can equivalently be expressed in the following form:

$$b(v) = B\cdot v + n_0 \tag{17}$$

where

$$B = \begin{bmatrix} \beta_{01} & \cdot\cdot & \beta_{0L} \\ : & & : \\ \beta_{m1} & \cdot\cdot & \beta_{mL} \end{bmatrix}_{(m+1)xL} \qquad n_0 = \begin{bmatrix} \beta_{00} \\ : \\ \beta_{m0} \end{bmatrix}_{(m+1)x1} \tag{18}$$

Analogously the denominator coefficients (5) can be expressed as

$$a(v) = A\cdot v + d_0 \tag{19}$$

where

$$A = \begin{bmatrix} \alpha_{01} & \cdot\cdot & \alpha_{0L} \\ : & & : \\ \alpha_{n1} & \cdot\cdot & \alpha_{nL} \end{bmatrix}_{(n+1)xL} \qquad d_0 = \begin{bmatrix} \alpha_{00} \\ : \\ \alpha_{n0} \end{bmatrix}_{(n+1)x1} \tag{20}$$

β_{kj} and α_{kj} are real constants.

For the family of transfer functions $\mathcal{P}_2$, the matrixes B and A have the following form:

$$B = \begin{bmatrix} \beta_{01} & \cdot\cdot & \beta_{0L_n} & 0 & \cdot\cdot & 0 \\ : & & : & : & & : \\ \beta_{m1} & \cdot\cdot & \beta_{mL_n} & 0 & \cdot\cdot & 0 \end{bmatrix}_{(m+1)xL} \tag{21}$$

$$A = \begin{bmatrix} 0 & \cdot\cdot & 0 & \alpha_{01} & \cdot\cdot & \alpha_{0L_d} \\ : & & : & : & & : \\ 0 & \cdot\cdot & 0 & \alpha_{m1} & \cdot\cdot & \alpha_{mL_d} \end{bmatrix}_{(n+1)xL} \tag{22}$$

For the family of transfer functions $\mathcal{P}_3$, the matrixes A, d_0, B and n_0 have the same structure as for $\mathcal{P}_1$. But the coefficients β_{kj} (or α_{kj}) can only have the values 0 or 1. Besides, in a row k of B (or A) there can only be one coefficient maximum β_{kj} (or α_{kj}) different to 0. In this case, the coefficient β_{k0} (or α_{k0}) of n_0 (or d_0) is equal to 0. Likewise, if all the coefficients β_{kj} (or α_{kj}) belonging to a row k of the matrix B (or A) are equal to 0, then the coefficient β_{k0} (or α_{k0}) of n_0 (or d_0) can be a real number.

Finally, for the family of transfer functions $\mathcal{P}_4$, the matrixes A, d_0, B and n_0 also have the same structure as for $\mathcal{P}_1$. Moreover, the coefficients β_{kj} (or α_{kj}) of B

(or A) and β_{k0} (or α_{k0}) of n_0 (or d_0) fulfil the same rules as the family $\mathcal{P}_3$.

In order to do a comparative study of the BHA, FUA and KSA, it is necessary to choose a working set of transfer functions, because the number of transfer functions belonging to one particular family is infinite. Our comparative study focused on the set of transfer functions belonging to the families $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, and $\mathcal{P}_4$, with the following structure:

$$P(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \qquad (23)$$

The maximum number of uncertain parameters considered is L=6, the maximum number of uncertain parameters in the numerator and denominator are, L_n=3 and L_d=3, respectively.

Twenty-five transfer functions with the indicated properties were generated by means of a random generation of v, B, n_0, A and d_0. Likewise, some transfer functions with higher n, m and L were considered in order to give more validity to our study. Only transfer functions that did not contain zeroes or poles on the $j\omega$-axis for all $v \in V$ were considered so that the three algorithms could simultaneously be used. It is important to bear in mind that BHA, FUA and KSA can only be used if $a(j\omega_i;\, v) \neq 0$, for any combination of the uncertain parameter values. The plant template $\Gamma(\omega_i)$ is not bounded if $a(j\omega_i;\, v) = 0$. Besides, if the hyper-rectangle formed from q or r contains the origin of the complex plane, then the BHA cannot be used.

This study was done with the assistance of Matlab and specific software called *Template Interactive Generator* (TIG) [3]. TIG is a free software tool with which it is possible to calculate the boundaries of templates associated to interval plant, or plants with affine parametric uncertainty. The main advantages of TIG are its ease of use and interactive nature.

Once a transfer function was generated in Matlab, it was loaded in TIG where the boundary of a template at a given frequency (for example Γ(1.5)) was calculated using BHA, FUA and KSA.

The comparison of BHA, FUA and KSA was done using the following magnitudes: number of template points N_t, number of points belonging to the template boundary N_c, algorithm yield R (in percentage) defined as

$$R = \frac{N_c}{N_t} \cdot 100 \qquad (24)$$

Furthermore, in the comparative study the following times were considered: time T_t for calculating the template and time T_c for calculating the template boundary using the ε-algorithm [8]. This is the algorithm implemented in TIG to recognize the points that form part of the boundary.

Three considerations must be made with respect to these times. First, in TIG the template computation algorithms must be manually tuned in order to obtain the right template boundary, i.e., users must choose the right value of the algorithm configuration parameter. Therefore, the values times T_t and T_c are only measured when each algorithm is already tuned. Second, TIG was run in a PC with a CPU of 2.8 GHz, a RAM of 512 Mb, and the Windows XP operating system. Third, the time T_c is exclusively associated with the ε-algorithm not with the FUA or KSA. However, it is interesting to know it in order to gain an idea about the time cost associated with the fact that the FUA and KSA do not directly give the template boundary.

5 RESULTS OBTAINED

5.1 FAMILY OF TRANSFER FUNCTIONS $\mathcal{P}_1$

One of the transfer functions $P_1 \in \mathcal{P}_1$, with the structure (23), considered in the study was

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -4 & -1 & 5 \\ 2 & -7 & 5 \\ 8 & 7 & 3 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} -3.9 \\ 6.4 \\ -3.0 \end{bmatrix} \qquad (25)$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 10 & 6 & -1 \\ 6 & -6 & -7 \\ 8 & -3 & 8 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} 1.4 \\ 9.3 \\ -0.1 \\ 1 \end{bmatrix} \qquad (26)$$

$$v_1 \in [-3.7, 9.1] \;\; v_2 \in [3.1, 6.6] \;\; v_3 \in [7.0, 7.2] \qquad (27)$$

Table 1 shows the magnitudes measured in the boundary calculation of Γ(1.5) associated with P_1 using the BHA, FUA and KSA. Likewise, Figure 1 shows the Γ(1.5) calculated with each algorithm, and a comparison of the template boundaries.

The comparative study for the family $\mathcal{P}_1$ of transfer functions indicates that the FUA is the only algorithm that is able to obtain the exact template boundary. Likewise, BHA generates a conservative template boundary. This conservatism increases when the number of common uncertain parameters in numerator and denominator increases. On the other hand, the KSA generates an erroneous template boundary.

Table 1: Magnitudes measured in the boundary calculation of Γ(1.5) associated with P_1

Algorithm	N_t	N_c	R**(%)**	T_t **(s)**	T_c **(s)**
BHA	122	122	100	0.156	0
FUA	324	181	55.8	$1.6 \cdot 10^{-2}$	$9.4 \cdot 10^{-2}$
KSA	672	191	28.4	$1.5 \cdot 10^{-2}$	0.125

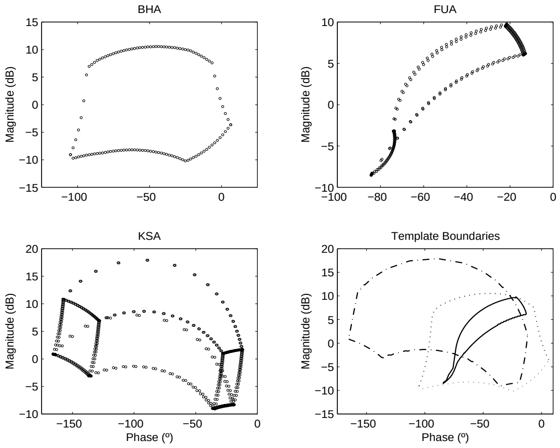

Figure 1: Γ(1.5) associated to $P_1 \in \mathcal{P}_1$ calculated by BHA, FUA and KSA, and boundary comparison

5.2 FAMILY OF TRANSFER FUNCTIONS $\mathcal{P}_2$

One of the transfer functions $P_2 \in \mathcal{P}_2$, with the structure (23), considered in the study was

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 8 & 9 & -1 & 0 & 0 & 0 \\ -1 & 5 & -7 & 0 & 0 & 0 \\ 4 & 3 & 6 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} 6.3 \\ 0.4 \\ 0.2 \end{bmatrix} \quad (28)$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -9 & -8 & 1 \\ 0 & 0 & 0 & 8 & 7 & 6 \\ 0 & 0 & 0 & 4 & 6 & -7 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} -6.4 \\ -0.9 \\ -7.7 \\ 1 \end{bmatrix} \quad (29)$$

$$\begin{array}{l} v_1 \in [-3.7, 9.1] \;\; v_2 \in [3.1, 6.6] \;\; v_3 \in [7.0, 7.2] \\ v_4 \in [0.9, 3.8] \;\; v_5 \in [-9.5, -8.9] \;\; v_6 \in [-0.7, 5.3] \end{array} \quad (30)$$

Table 2 shows the magnitudes measured in the boundary calculation of Γ(1.5) associated with P_2 using the BHA, FUA and KSA. Likewise, Figure 2 shows the Γ(1.5) calculated with each algorithm, and a comparison of the template boundaries.

The comparative study for the family $\mathcal{P}_2$ of transfer functions indicates that the BHA and FUA can be used to obtain the exact template boundary. However, it is better to use the BHA because it has a yield of 100 %, i.e., all the templates points calculated by the BHA belong to the template boundary. The FUA has a low yield. The computation time T_t for both the BHA and FUA is very similar. On the other hand, the KSA generates an erroneous template boundary.

Table 2: Magnitudes measured in the boundary calculation of Γ(1.5) associated with P_2

Algorithm	N_t	N_c	R**(%)**	T_t **(s)**	T_c **(s)**
BHA	100	100	100	0.125	0
FUA	960	56	5.8	0.125	$7.8 \cdot 10^{-2}$
KSA	896	179	19.9	$3.1 \cdot 10^{-2}$	0.22

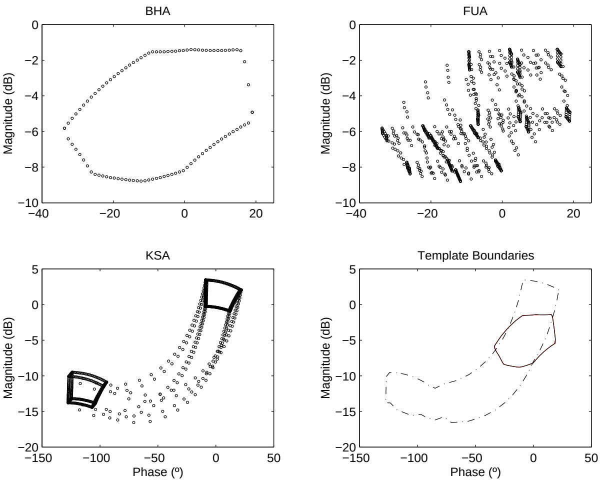

Figure 2: Γ(1.5) associated to $P_2 \in \mathcal{P}_2$ calculated by BHA, FUA and KSA, and boundary comparison

5.3 FAMILY OF TRANSFER FUNCTIONS $\mathcal{P}_3$

One of the transfer functions $P_3 \in \mathcal{P}_3$, with the structure (23), considered in the study was

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (31)$$

$$\begin{bmatrix}a_0\\a_1\\a_2\\a_3\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\end{bmatrix}\cdot\begin{bmatrix}v_1\\v_2\\v_3\end{bmatrix}+\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \qquad (32)$$

$$v_1\in[5.6,8.1]\ \ v_2\in[0.1,9.8]\ \ v_3\in[-5.9,-2.2] \qquad (33)$$

Table 3 shows the magnitudes measured in the boundary calculation of Γ(1.5) associated with P_3 using the BHA, FUA and KSA. Likewise, Figure 3 shows the Γ(1.5) calculated with each algorithm, and a comparison of the template boundaries.

The comparative study for the family $\mathcal{P}_3$ of transfer functions indicates that the FUA is the only algorithm that is able to obtain the exact template boundary. Likewise, the BHA and KSA generate the same conservative template boundary. This conservatism increases when the number of common uncertain parameters in numerator and denominator increase. The computation time T_t of the BHA is greater than the T_t of the KSA. However, in general, the BHA is preferable to the KSA because it has a yield of 100 %.

Table 3: Magnitudes measured in the boundary calculation of Γ(1.5) associated with P_3

Algorithm	$\boldsymbol{N_t}$	$\boldsymbol{N_c}$	***R*(%)**	$\boldsymbol{T_t}$ **(s)**	$\boldsymbol{T_c}$ **(s)**
BHA	150	150	100	0.203	0
FUA	204	137	67.2	$1.5\cdot10^{-2}$	$4.7\cdot10^{-2}$
KSA	352	69	19.6	$1.5\cdot10^{-3}$	$3.1\cdot10^{-2}$

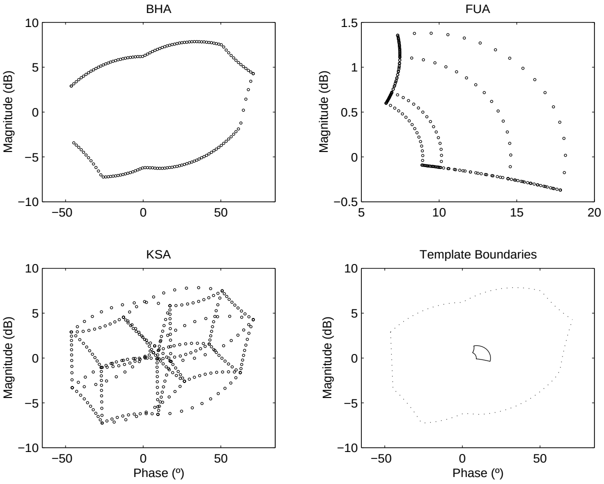

Figure 3: Γ(1.5) associated to $P_3\in\mathcal{P}_3$ calculated by BHA, FUA and KSA, and boundary comparison

5.4 FAMILY OF TRANSFER FUNCTIONS $\mathcal{P}_4$

One of the transfer functions $P_4\in\mathcal{P}_4$, with the structure (23), considered in the study was

$$\begin{bmatrix}b_0\\b_1\\b_2\end{bmatrix}=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\end{bmatrix}\cdot v+\begin{bmatrix}0\\0\\0\end{bmatrix} \qquad (34)$$

$$\begin{bmatrix}a_0\\a_1\\a_2\\a_3\end{bmatrix}=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix}\cdot v+\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \qquad (35)$$

$$v=\begin{bmatrix}v_1 & v_2 & v_3 & v_4 & v_5 & v_6\end{bmatrix}^T \qquad (36)$$

$$\begin{matrix}v_1\in[-6.8,-4.5]\ \ v_2\in[-0.9,0.4]\ \ v_3\in[-0.3,0.3]\\ v_4\in[3.1,6.1]\ \ v_5\in[-6.1,-0.1]\ \ v_6\in[-3.8,-0.1]\end{matrix} \qquad (37)$$

Table 4 shows the magnitudes measured in the boundary calculation of Γ(1.5) associated with P_4 using the BHA, FUA and KSA. Likewise, Figure 4 shows the Γ(1.5) calculated with each algorithm, and a comparison of the template boundaries.

Table 4: Magnitudes measured in the boundary calculation of Γ(1.5) associated with P_4

Algorithm	$\boldsymbol{N_t}$	$\boldsymbol{N_c}$	***R*(%)**	$\boldsymbol{T_t}$ **(s)**	$\boldsymbol{T_c}$ **(s)**
BHA	100	100	100	0.141	0
FUA	768	43	5.6	0.140	$3.1\cdot10^{-2}$
KSA	160	29	18.1	$9.9\cdot10^{-3}$	$7.8\cdot10^{-3}$

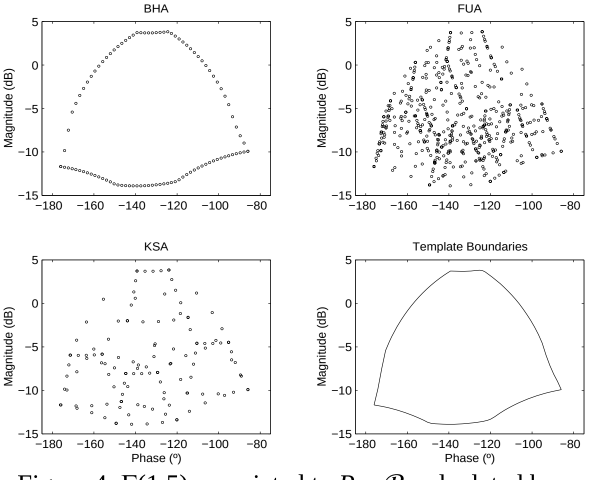

Figure 4: Γ(1.5) associated to $P_4\in\mathcal{P}_4$ calculated by BHA, FUA and KSA, and boundary comparison

The comparative study for the family $\mathcal{P}_4$ of transfer functions indicates that the three algorithms can be used to obtain the exact template boundary. However, it is better to use the BHA because it has a yield of 100 %, and its computation time T_t is very similar to the other two algorithms.

On the other hand, the KSA has a higher yield than the FUA. The KSA also needs a lower number of points N_t than the FUA to define the template boundary.

Moreover, the Bailey & Hui algorithm is the only one that directly generates the template boundaries. With the other two algorithms an additional algorithm is required to find the template boundaries.

6 CONCLUSIONS

According to the results obtained in the comparative study described in the previous section, it is possible to build Table 5. It shows the features of the template boundary that each algorithm is able to generate depending on the family of transfer functions considered.

Table 5: Features of the template boundary generated for each algorithm depending on the family of transfer functions

Family	***BHA***	***FUA****	***KSA****
$\mathcal{P}_1$	Conservative	Exact	Erroneous
$\mathcal{P}_2$	Exact	Exact	Erroneous
$\mathcal{P}_3$	Conservative	Exact	Conservative
$\mathcal{P}_4$	Exact	Exact	Exact

[a] An additional algorithm is required to recognise the points belonging to the template boundary.

This table can help the designer to choose the adequate template computation algorithm depending on the kind of plant. Thus if the transfer function belongs to $\mathcal{P}_1$ or $\mathcal{P}_3$, then the FUA must be used. If the transfer function belongs to $\mathcal{P}_2$ or $\mathcal{P}_4$, then the BHA must be used. Finally, if the BHA cannot be used because the transfer function has zeroes or poles on the $j\omega$-axis for any $v \in V$, and if the transfer function belongs to $\mathcal{P}_2$, then the FUA must be used. Likewise, if the transfer function belongs to $\mathcal{P}_4$, then the KSA must be used.

Acknowledgment

This research was supported by the Spanish CICYT under grants DPI2012-31303 and DPI2013-46665-C2-2-R.

References

[1] Bailey F. N., and Hui C. H., (1989) "A fast algorithm for computing parametric rational functions", IEEE transactions on automatic control, vol. 34, no. 11, pp. 1209-1212.

[2] Bartlett A. C., Tesi A., and Vicino A., (1993) "Frequency response of uncertain systems with interval plants", IEEE transactions on automatic control, vol. 38, no. 6, pp. 929-933.

[3] Díaz J. M., Dormido S., and Aranda J., (2008) "An interactive approach to the templates generation in QFT methodology", Asian Journal of Control, vol.10, no. 3, pp. 361-367

[4] Fu, M., (1990) "Computing the frequency response of linear systems with parametric perturbation", Systems & Control Letters, vol. 15, pp. 45-52.

[5] Horowitz, I. M., (1963) Synthesis of feedback systems, Academic Press, New York.

[6] Horowitz, I. M., (2001) "Survey of Quantitative Feedback Theory (QFT)", International Journal of Robust and Non-linear Control, vol. 11, no. 10, pp. 887-921.

[7] Houpis C. H., Rasmussen S. J., and García-Sanz M., (2006) Quantitative Feedback Theory: fundamentals and applications, 2nd Edition, CRC Taylor & Francis: Boca Ralen.

[8] Montoya F. J., (1998) "Diseño de sistemas de control no lineales mediante QFT: Análisis computacional y desarrollo de una herramienta CACSD". Ph.D. thesis. Universidad de Murcia (Spain), Departamento de Informática y Sistemas Informáticos.

[9] Yaniv, O. (1999) Quantitative feedback design of linear and nonlinear control systems, Kluwer Academic Publishers: Norwell, Massachusetts.