Abstract—This paper gives a general description of Template Interactive Generator (TIG), a free software tool which calculates the templates boundaries of plants with special parameter dependencies, such as interval plants; and plants with affine parametric uncertainty. TIG can also find the template boundaries associated with other kinds of plants whose templates have been previously computed in Matlab. Besides, several examples are included to illustrate the use of this software tool.

I. INTRODUCTION

A plant template $\Gamma(\omega_i)$ is a set of complex numbers representing the frequency response at a fixed frequency $\omega_i$ of an uncertain plant transfer. Plant templates are used in several approaches to analysis and design of robust control systems, for example, Quantitative Feedback Theory (QFT) [1]-[5].

Several approaches to solve the template generation problem exist. QFT designers usually use the grid method. This is a brute-force approach, 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 direct and simple, it has several disadvantages [6]. Fortunately, in the last fifteen years other methods or algorithms for calculating templates have been developed. Each one uses different strategies: geometric considerations [6]-[7], Kharitonov segments [8], factorisation in simple elements and convolution [9], symbolic calculation [10], and interval analysis [11].

The most used and known software tool to assist the designer in implementing the QFT methodology is the Matlab QFT toolbox [12]. However, this toolbox does not implement any algorithms to compute templates.

This paper gives a general description of the free software tool Template Interactive Generator (TIG) developed in Sysquake [13] which assists the QFT designer in calculating template of interval plants, and plants with affine parametric uncertainty in their coefficients. These kinds of plants are very usual in control problems. Besides, TIG can also find the template boundaries associated with other kinds of plants whose templates have been previously computed in Matlab.

The paper is organized as follows. Section II describes the basic features of TIG. Section III includes two examples to illustrate the use of TIG. Finally, Section IV gives some concluding remarks.

II. BASIC FEATURES OF THE INTERACTIVE TOOL

A. Purpose of the tool

Let the following uncertain parameters vector be

$$p = [p_1, p_2, \ldots, p_L]$$

where $p_j \in [p_{j\min}, p_{j\max}] j=1,2,\ldots,L$. With TIG it is possible to calculate the template boundaries of plants whose transfer functions have the following structure:

$$P(s; p) = \frac{b(s; p)}{a(s; p)} = \frac{\sum_{k=0}^{m} b_k(p) s^k}{\sum_{k=0}^{n} a_k(p) s^k}$$

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

$$b_k(p) = \beta_{k0} + \sum_{j=1}^{L} \beta_{kj} p_j$$

$$a_k(p) = \alpha_{k0} + \sum_{j=1}^{L} \alpha_{kj} p_j$$

In the previous expressions $\beta_{kj}$ and $\alpha_{kj}$ $j=0,\ldots,L$ are real constants. These kinds of plants are known as plants with affine parametric uncertainty. A particular case of (2) is obtained when the numerator uncertain parameters are independent of the denominator uncertain parameters:

$$P(s; q, r) = \frac{b(s; q)}{a(s; r)} = \frac{\sum_{k=0}^{m} b_k(q) s^k}{\sum_{k=0}^{n} a_k(r) s^k}$$
where \( q_k \in [q_k^{\text{min}}, q_k^{\text{max}}] \) \( h=1,2,...,L_h \) and \( r_i \in [r_i^{\text{min}}, r_i^{\text{max}}] \) \( i=1,2,...,L_i \), with \( L_h+L_i=L \). The coefficients \( b_h(q) \) and \( a_d(r) \) also present an affine structure similar to (3) and (4).

Another particular case of (2) is obtained when the plant transfer function coefficients are directly the uncertain parameters, such kinds of plants are known as interval plants:

\[
P(s; q, r) = \frac{b(s; q)}{a(s; r)} = \frac{\sum_{k=0}^{m} q_i \cdot s^k}{\sum_{k=0}^{n} r_k \cdot s^k} \tag{6}
\]

where \( q_i \in [q_i^{\text{min}}, q_i^{\text{max}}] \) and \( r_k \in [r_k^{\text{min}}, r_k^{\text{max}}] \).

For the plants described by (2), (5) and (6), TIG includes four algorithms to calculate the associated templates. These algorithms are the following: a) the Bailey & Hui algorithm [6], b) the Fu algorithm [7], c) the Kharitonov segment algorithm [8], and d) the grid algorithm.

The Bailey & Hui algorithm directly generates template boundary points for plant types (5) and (6). If there is any dependency between the numerator uncertain parameters and the denominator uncertain parameters, then this algorithm generates a conservative template boundary. The number of template boundary points \( N_t \) obtained with this algorithm can be determined using the expression:

\[
N_t = 2 \cdot N_b
\tag{7}
\]

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

The Fu algorithm considers \( 2^{L-1} \cdot L \) template edges (\( L \) is the number of uncertain plant parameters). A template edge can be external or internal. A template external edge forms part of the template boundary. Likewise, a template internal edge is contained inside the template boundary. Moreover, this algorithm generates \( N_t \) points for each edge. \( N_t \) is an algorithm parameter. Therefore, the number of generated template points is

\[
N_t = 2^{L-1} \cdot L \cdot N_f
\tag{8}
\]

The Kharitonov segment algorithm can be used for plant type (6). This algorithm considers 32 template edges (the Kharitonov segments), and it generates \( N_t \) points for each one. \( N_t \) is an algorithm parameter. Therefore, the total number of generated points is

\[
N_t = 32 \cdot N_k
\tag{9}
\]

The grid algorithm generates \( N_t \) points of a template, according to the following expression:

\[
N_t = \prod_{j=1}^{L} N_{p_j}
\tag{10}
\]

where \( N_{p_j} \) is the number of equispaced values for the uncertain parameters \( p_j \in [p_j^{\text{min}}, p_j^{\text{max}}] \), \( j=1,...,L \) which are considered for obtaining the template points. The \( N_{p_j} \) are algorithm parameters.

The Bailey & Hui algorithm, Fu algorithm and Kharitonov segment algorithm can only be used if \( a(j \omega; p) \neq 0 \), for any combination of the uncertain parameter values. The plant template \( \Gamma(a) \) is not bounded if \( a(j \omega; p) = 0 \). Besides, if the hyper-rectangle formed from \( q \) or \( r \) contains the origin of the complex plane, then the Bailey & Hui algorithm cannot be used.

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. TIG uses the \( \varepsilon \)-algorithm proposed by F.M. Montoya [14]. It basically works in the following way: it starts from a template boundary point, typically the template point with a greater real part. It traces a circle with radius \( \varepsilon \), and it looks for the template point with the minimum phase contained inside the circle. This point also belongs to the template boundary. This procedure is repeated till all of the template boundary points are located. The \( \varepsilon \)-algorithm implemented in TIG has three parameters which can be tuned by users: Weight, Factor X and Factor Y. Weight is a positive real number that multiplies the diagonal \( D \) of the rectangle which encloses all the template points. The radius \( \varepsilon \) is related to the parameter Weight in the following expression:

\[
\varepsilon = \text{Weight} \cdot D
\tag{11}
\]

Moreover, Factor X and Factor Y are real positive numbers which divide the coordinate \( x \) or the coordinate \( y \) of all template points.

B. Interactivity

The main features of TIG are its ease of use and strong interactivity. These are common features of all the software tools developed with Sysquake [18]. All that users have to do is to place the mouse pointer over the different items that the tool displays on the screen. Any action carried out on the screen is immediately reflected on all the graphs generated and displayed by the tool. This allows users to visually perceive the effects of their actions. For example, when users modify the configuration parameters of the template computation algorithms (\( N_b \) in Bailey & Hui algorithm, \( N_f \) in Fu algorithm, and \( N_t \) in Kharitonov segment algorithm) or the work frequency \( \omega_w \), the modifications to the work template \( \Gamma_w(\omega_w) \) and its boundary are immediately seen.

Other interesting features of TIG are:

--The simultaneous visualisation of the templates computed for each one of the four-template computation algorithms implemented in TIG.
--The automatic selection and removing of the interior points of \( \Gamma_w(\omega_0) \).
--The manual selection and removing of any point of \( \Gamma_w(\omega_0) \).
--Information about the template computation algorithm.

TIG can work with other software tools, like Matlab or QFTIT [15], to design robust controllers with QFT methodology.

C. TIG description

With TIG it is possible to work in two different modes: configuration and analysis. The analysis mode is described first because the tool always starts in this mode on a predefined example.

1) Analysis mode. The main window of the tool in the Analysis mode is divided into several zones (see Figure 1), and each zone contains different items and figures. There are six zones on the left side of the window (Work Mode, Status, Plant Uncertain Parameters, Analysis, Algorithm Information and Plant Transfer Function), and three zones on the right side of the window (Saved Templates, Frequencies and Work Template).

With the Work Mode zone it is possible to select the work mode. Users cannot force the transition from the configuration mode to the analysis mode. This transition is automatically done when they complete the last step of the configuration mode. However, users can go to the configuration mode when they wish.

The Saved Templates zone contains a Nichols chart where the saved templates \( \Gamma(\omega) \) \( \omega \in \Omega \) are represented in different colours. A saved template is a valid template which can be exported to Matlab or QFTIT. Likewise, the template points are represented with ‘o’ and their nominal value with ‘x’.

Initially, the templates shown in this zone are computed by the Bailey & Hui algorithm (with \( N_b=25 \)), when the transition from the configuration mode to the analysis mode takes place. Users can save other templates in this zone throughout the work session, clicking on the small red triangles located at both sides. With TIG it is only possible to save one template for each frequency \( \omega_0 \).

In the Frequencies zone, users can visualize and configure the work frequencies set \( \Omega \), and select the work frequency \( \omega \). The zone contains a horizontal axis which is graduated in radians per seconds using a logarithmic scale. On this axis, a vertical segment for each frequency \( \omega_0 \in \Omega \) is drawn in different colours. The colour code is the same used in the Saved Templates zone. Besides, users can add and remove frequencies \( \omega_0 \in \Omega \).

The Work Template zone contains a Nichols chart (or a complex plane) where the current work template \( \Gamma_w(\omega_0) \) is drawn. Different colours are used to distinguish the algorithm which is being used to compute the templates. The \( \Gamma_w(\omega_0) \) points are represented in this zone with the symbol ‘o’. The \( \Gamma_w(\omega_0) \) boundary points are connected using a solid line. Other items located in this zone are the text field \( w \) and the button Save. With the text field it is possible to visualise and configure the current value of \( \omega \). The button is used to save \( \Gamma_w(\omega_0) \) in the Saved Templates zone, i.e., if users click on it, then \( \Gamma(\omega_0)=\Gamma_w(\omega_0) \).

The Status zone contains three circular indicators: the upper one is associated with the \( \Gamma_w(\omega_0) \) calculation, the central one is associated with the \( \Gamma_w(\omega_0) \) boundary calculation, and the lower one is associated with the computational load state. Each indicator can take three different colours (red, yellow or green) to warn users about certain events.

In the Plant Uncertain Parameter zone, users can visualize and configure the maximum, nominal and minimum values of the plant uncertain parameters.

The Analysis zone allows users to do several actions: select the \( \Gamma_w(\omega_0) \) computation algorithm, configure the parameters of the selected algorithm \( (N_b, N_f \text{ or } N_l) \) and the parameters of the \( \varepsilon \)-algorithm \( (Weight, \text{Factor } X, \text{and Factor } Y) \), select and remove \( \Gamma_w(\omega_0) \) points, and correct the \( \Gamma_w(\omega_0) \) phase jumps.

Finally, the Algorithm Information and Plant Transfer Function zones have merely an informative function. The Algorithm Information zone shows the following data: \( N_f \) (number of \( \Gamma_w(\omega_0) \) points), \( N_c \) (number of \( \Gamma_w(\omega_0) \) boundary points), \( R \) (algorithm yield \( (N_f/N_c) \)), \( T_f \) (\( \Gamma_w(\omega_0) \) calculation time) and \( T_c \) (\( \Gamma_w(\omega_0) \) boundary calculation time). Furthermore, the Plant Transfer Function zone shows the symbolic expression of the plant transfer function.

2) Configuration mode. The configuration mode consists of four sequential steps:

--Step 1. Configuration of the plant uncertain parameters.
--Step 2. Configuration of the plant transfer function denominator.
--Step 3. Configuration of the plant transfer function numerator.
--Step 4. Configuration of the work frequencies set \( \Omega \).

In this work mode, the TIG window’s appearance changes (see Figure 2). Users must work principally in the Configuration Steps zone to configure the number of plant uncertain parameters, their values (minimum, maximum and nominal), and the work frequencies set \( \Omega \). The upper part of this zone contains four text lines associated with the four sequential steps of this mode. The current step is always highlighted.

In this work mode, other useful zones are: Configure Plant Uncertain Parameters, Table of Plant Uncertain Parameters and Plant Transfer Function. The first zone offers an alternative way of configuring the plant uncertain parameters. The other two zones have an informative function. Thus, the Table of Plant Uncertain Parameters zone shows a table which contains the maximum, nominal and minimum values of the plant uncertain parameters. The Plant Transfer Function zone shows, apart from the denominator and numerator expressions, the field \( N_{ID} \) with which it is possible to configure the number of plant integrators or derivatives.
Fig. 1. Example of the TIG window in the Analysis mode

Fig. 2. Example of the TIG window in the Configuration mode
III. ILLUSTRATIVE EXAMPLES

A. Plant with affine parametric uncertainty

Let the plant with affine parametric uncertainty originally proposed in [7] be

\[ P(s, p) = \frac{s^3 + b_2 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \]

\[ b_2 = 0.4 p_1 + 0.2 p_2 + 4 \]
\[ b_0 = p_1 - p_3 + 20 \]
\[ a_2 = 0.5 p_1 - 0.5 p_2 + 0.5 p_3 + 9.5 \]
\[ a_1 = 2.0 p_1 + p_2 + 27 \]
\[ a_0 = 2.0 p_1 + p_2 + 27 \]

(12)

It has three uncertain parameters:

\[ \{p_1, p_2, p_3\} \in [-3, 3] \]

(13)

Whose nominal values are:

\[ p_1^0 = 1.5, \quad p_2^0 = -1.5, \quad p_3^0 = 1.5 \]

(14)

In order to keep the description as brief as possible, it is going to be assumed that the work frequencies set consists of only one element: \( \Omega = \{1\} \) rad/s. This example is going to illustrate how it is possible to calculate \( \Gamma(1) \) and obtain its boundary using TIG.

First of all, users have to go to the configuration mode and sequentially introduce step by step all the information which TIG needs to be able to work: maximum, nominal and minimum values of the plant uncertain parameters, the numerator and denominator structures, and the work frequencies set \( \Omega \). Once the configuration has finished, TIG automatically computes \( \Gamma(1) \) using by default the Bailey & Hui algorithm with \( N_f = 25 \), and goes to the Analysis mode.

Let us assume a set of templates which have been computed in Matlab. To import these templates to TIG, it is necessary to use the function mat2tig.m (which is distributed together with TIG) to generate an external file. Then, this file must be loaded to TIG. The Saved Templates zone will show the templates imported from Matlab.

The Bailey & Hui algorithm generates conservative template boundaries. Therefore, another algorithm must be used. The best selection in this case [16] is the Fu algorithm. Thus, this algorithm must be selected as the work algorithm in the Analysis zone. Simultaneously, TIG calculates (using theFu algorithm with \( N_f = 5 \)) and draws \( \Gamma_w(1) \) in the Work Template zone. However, TIG has not drawn the \( \Gamma_w(1) \) boundary. This means that the \( \varepsilon \)-algorithm has not been able to find the boundary using the default values (\( Weight=1 \), Factor \( X = 1 \), and Factor \( Y = 1 \)). Therefore, these parameters must be modified using the appropriate text fields or sliders. For example, if \( Weight=0.37 \), a boundary of \( \Gamma_w(1) \) is obtained. However, it is not correct because the \( \Gamma_w(1) \) points placed at the lower right end do not belong to this boundary. Users can try to improve the boundary by decreasing \( Weight \) even more, but no improvement will be obtained. In fact, the boundary will disappear. Another option is to increase Factor \( X \). Users can check that a good boundary appears if Factor \( X \) is increased to 4 (see Figure 3).

Fig 3. \( \Gamma_w(1) \) calculated with the Fu algorithm, \( N_f = 5 \) and its boundary (solid line) found with the \( \varepsilon \)-algorithm, \( Weight=0.37 \), Factor \( X = 4 \) and Factor \( Y = 1 \)

Users can analyze whether it is necessary to increase \( N_f \) to improve the \( \Gamma_w(1) \) boundary or whether it is possible to decrease \( N_f \) without significantly changing the boundary shape. This study shows that \( N_f = 5 \) is a good value.

Finally, before saving \( \Gamma_w(1) \) in the Saved Templates zone, the non-boundary points must be removed. This action can be easily done in TIG, clicking on the Remove button located in the Analysis zone.

B. Boundary determination from a template calculated in Matlab

Let us assume a set of templates which have been computed in Matlab. To import these templates to TIG, it is necessary to use the function mat2tig.m (which is distributed together with TIG) to generate an external file. Then, this file must be loaded to TIG. The Saved Templates zone will show the templates imported from Matlab.

Let us assume that \( \omega_w = 1 \) rad/s is chosen as the work frequency. Simultaneously in the Work Template zone (see Figure 4) \( \Gamma_w(1) \) is drawn. The template boundary is found by the \( \varepsilon \)-algorithm. According to the information shown in the Algorithm Information, \( \Gamma_w(1) \) has \( N_f = 428 \) points, and its boundary has \( N_c = 72 \) (see Figure 5). Therefore, the yield is \( R = 16.82 \% \). Moreover, the boundary calculation time is \( T_c = 4.7 \cdot 10^{-2} \) s.

The boundary obtained with the \( \varepsilon \)-algorithm (considering the default values of Weight, Factor \( X \) and Factor \( Y \)) is correct. Therefore, it is not necessary to tune the configuration parameters. The non-boundary points can be easily removed in TIG, clicking on the Remove button located in the Analysis zone.
Some sides of the $\Gamma_w(1)$ boundary can remain well defined using a fewer number of points. With TIG it is possible to manually remove some template boundary points. Thus, a valid boundary is obtained with only $N_c=18$ points (see Figure 6). Note that using TIG the template points have been decreased from 428 to 18, i.e., a 95.8% reduction is obtained.

![Figure 6. $\Gamma_w(1)$ boundary with $N_c=18$ points.](image)

**IV. CONCLUSION**

This paper presents the main features of the free software tool TIG. It can calculate templates of interval plants, and plants with affine parametric uncertainty in their coefficients. TIG can also find the template boundaries associated with other kinds of plants whose templates have been previously computed in Matlab.

**REFERENCES**