International Journal of Advanced Science and Technology
Vol.99 (2017), pp.39-56
http://dx.doi.org/10.14257/ijast.2017.99.04

Inverse Interpolation Techniques for Temporary Specifications Translation to Frequency Domain

Javier Joglar Alcubilla

Avionics Department, Barajas Institute, Avda. América 119, 28042, Madrid, Spain
jjoglar@educa.madrid.org

Abstract

It is proposed a translation methodology applied to temporal transient response characteristics to obtain parameters of the typical second order transfer function, in the frequency domain. This method based on automatic successive approximations, applying an inverse interpolation, will minimize overdesign and manual iteration in the process of searching for a controller in frequency. For each type of requirement, specific techniques are recommended to obtain specification translation, using the "ascending and descending differences Newton's procedure" due to its practice advantages, compared to other interpolation procedures. It will be used as interpolation functions those describing step signal or pulse signal, with subcritical damping and approached with a second order transfer function.

***Keywords**: Temporary specifications translation, frequency domain, inverse interpolation, quantitative feedback theory, tracking, sensitivity.*

1. Introduction

In the technical design of controllers working in the frequency domain, as in QFT [9], the design specifications represent fulfillment requirements associated with the dynamics of the system to control in closed loop. These specifications or requirements are usually expressed in the time domain using temporary transient response characteristics [10], such as:

- The rise time "t_r" or "*tr*", the maximum overshoot "*Mp*", the maximum peak time "t_{Mp}", the settling time "t_s" or "*ts*" and settlement channel size or maximum permissible tolerance "*dev*", typical of the response of a closed loop control system to a step change in the input.
- The maximum peak value "f_{max}", peak times "t_p", settling time "t_s" or "*ts*" and settlement channel "*dev*", associated with the response of a control system in closed loop disturbance rejection.

The responses to these changes, unitary step or disturbance represented by unitary pulse, can be approximated well by the responses of underdamped second order systems of step or pulse, respectively, which are formally described based on the parameters, damping coefficient "ζ" and natural frequency "ω_n" [21].

If you start with time domain specifications and design stage of the control methodology used is performed in frequency, the initial requirements imposed must be transformed to the frequency domain. It is seeking transfer functions in the frequency domain that satisfy the envelope limits, upper and/or lower, generated by the temporary specification. Traditional methodologies, as described in [4], the technique in [17] or the model-based procedure used in [3], generally start with second order transfer functions, using different parameters of frequencies ω_n and coefficients ζ; adding poles and zeros to the function, you get to go closer as you can to the original temporal specification. But in any case, all procedures are based on trial and error manual iteration, which ends when the designer decides that the function obtained is adequate enough.

When you are talking about specifications translation from time domain to the frequency domain, actually you are not meaning to translate characteristics of the transient response (*tr*, *ts*, *MP*, ..) into specifications on the frequency response (gain margin and phase, bandwidth, M circles, ..). It is using the nomenclature, widely used by authors, as in [24] or [10], to reference the control stage that deals with the determination of any transfer function in the frequency domain, which provides an equivalent response to the given by temporary requirements imposed.

There are other ways to simplify the specifications transfer technique [21]:

- Use classical hypothesis approximation of settling times; but only valid for tracking step inputs and settlement channels with 2% or 5% of the steady output value.
- Graphically, using abacuses.

A more accurate method which minimizes manual iteration and related overdesign problems [8], consists of using successive approximations to obtain the characteristics of transient response required, associated to specific parameters ω_n and ζ. This is possible because if you observe the formal description of the responses step type or pulse approximated to a second order system, you can see that they are doubly transcendental equations in the desired time characteristic (settlement or rise), respect to ω_n and ζ . A particular method is presented in [15], for temporary specifications transfer tracking to obtain the lower frequency limit, described in the form of transfer functions with underdamped second order response. This method, specific for tracking requirements, uses inverse interpolation techniques described in this paper for step type responses.

The aim of this work is to find techniques that provide transfer functions of second order in the frequency domain, whose responses are equivalent to those provided by the specifications imposed in the time domain. Its application in the design stage of frequency controllers must ensure a reduced overdesign and minimal iteration in the controller search process, for purposes of compliance with these temporal specifications and frequency transferred too.

This paper is the continuation of the study begun to achieve the goal described above in Spanish in [16].

2. Temporary Specifications Translation to the Frequency Domain

The importance of having a precise, automatic and specific methodology to translate temporary specifications to frequency is obvious, especially in the final stage of system responses analysis in control techniques, where the controller design is done in the frequency domain [13]. With a precise temporary specification transfer to the frequency domain, the specifications compliance in frequency will also mean fulfilling of them in time with a higher probability. Thus, the necessary degree of manual iteration can be reduced simply carrying out an adequate translation of specifications.

The use of successive approximations to the time parameter searched, applying any inverse interpolation method [14], specific for each type of requirement, will provide a tool for reducing manual iteration in the design process.

Specific techniques for specifications translation developed will be based on the "Newton's method by ascending and descending differences of 5th order". This method was chosen over other traditional ones for interpolation, such as the Lagrange's polynomial, because of its practical advantages [11]:

- It requires less arithmetic operations.

- No need to restart the computer, if you want to add or remove a set point that is used to construct the final polynomial.

In this sense, the first thing to consider is the type of function and, secondly, the temporal response characteristic to interpolate. The function is directly related to the type of specification to be treated.

- There are the specifications based on responses to step inputs tracking:
  - *Upper limit tracking*, an input reference with maximum value "r_{max}" will be tracked by an output which must not exceed the amplitude "y_{max}" [18]. See Figure 1, case (a)
  - *Lower limit tracking*, an output signal will track a step input in an underdamped way, not exceeding a maximum specific values of overshoot, rise time, settling time and settlement channel [20]. See example in Figure 2.
- Also, there are specifications associated with responses typed underdamped pulse, induced by an input disturbance, such as:
  - *Generic sensitivity*, a disturbance "*In*" produces an output response pulse, not exceeding maximum specific amplitude "*Out*" and settling on after a specific maximum time *T* [5]. See in Figure 1, case (b).
  - *Specific sensitivity*, an input disturbance generates a pulse output between two specific values *"[± Out, ± Out_{max}]"*, but for a limited time *T* , settling into a channel with amplitude ±*Out* [7]. See example in Figure 4.

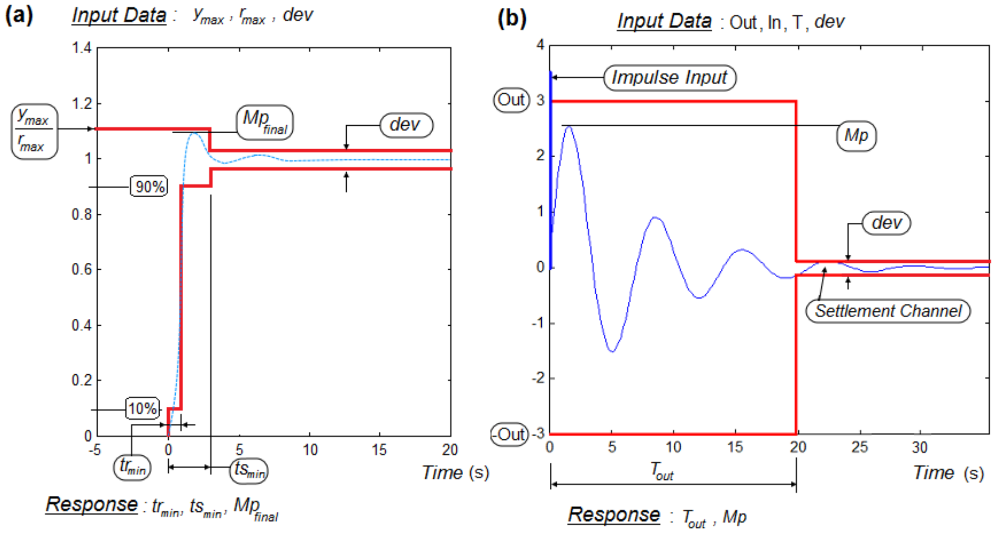

Figure 1. Specification Examples: (a) Upper Frequency Tracking Limit (Minimum Times and Maximum Overshoot), (b) Generic Sensitivity (Upper Frequency Limit)

The main objective of this work is to associate each time specification indicated with a transfer function, representative of a second order underdamped system with parameters ω_n and ζ, as an approximation in the frequency domain. Why use a second order approximation and not any other order? With this type of approach the process can be automatic. In [15] is also developed an approach to the envelope of lower and upper limits for tracking specifications based on higher order systems, but requires manual implementation procedure; compared to the automatic second order method, it is concluded that are not better the results with the higher-order system.

In the second order approximation, the way to relate the specification response (time domain) with the equivalent parameters ω_n and ζ (frequency domain) is through the function that formally describes the temporal response generated by the specification. That is, using the descriptive function of the unitary step signal or the unitary pulse signal. Thus, it is used as a time dependent function any of the following:

$$f_{step}(t)=1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}sen\left(\omega_n\sqrt{1-\zeta^2}t+a\cos\zeta\right) \tag{1}$$

$$f_{pulse}(t)=\frac{\omega_n}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t}sen\left(\omega_n\sqrt{1-\zeta^2}t\right) \tag{2}$$

Observe how in the functions (1) and (2) is implicit the temporal relationship to the frequency domain, through the parameters ω_n and ζ.Therefore, they can be used as interpolation functions of times (t_r and/or t_s) for different values of ω_n and ζ. In practice, the response characteristics to interpolate for each of the functions indicated will be,

- The rise time and settling time of the step function (1), when working with tracking specifications of upper or lower limit.
- The rise time and/or the settling time of the pulse function (2), when working with generic or specific sensitivity specifications.

On the other hand, it can be argued that there are powerful tools such as Matlab functions that solve transcendental equations as proposed above. For example, with “solve” and “fsolve” really is so. In particular, “solve” searches analytical solutions (exact), while “fsolve” solves nonlinear equations by Gauss-Newton’s method of least squares, using a mix of quadratic and cubic potential functions approximation. However, in the latter case, there is not an interpolation procedure, but an alternative method of nonlinear parametric adjustment [19]; i.e. “fsolve” can solve transcendental equations as proposed but, using an adjustment method, approximates to the function data, not to the function itself, as interpolation methods do; also, the approach is in one strict way, while the technique of differences that we use here, decides at each point if is more interested the progressive case (ascending) or the regressive one (descending), optimizing then the approach to each point and, obtaining a final function with minimal error respect to the initial desired. Moreover, it is important to note that in the differences method proposed is selected a specific range of interpolation for each equation to be treated, in each particular situation, while “solve” and “fsolve” use generic criteria applicable to any function. This makes these functions are not able to find the desired times in some situations, particularly the settlement one, on the step/pulse function considered; specifically, for t_s you need to search by intervals (between peak to valley or peak to peak) and, “solve/fsolve” work with the whole function directly. On the other hand, “solve” provides solutions in the complex plane (by requiring an exact solution), which in the case of step/pulse functions don’t interest; searching temporary response characteristics by interpolation are only valid real positive solutions, which “solve” does not give in many cases.

In conclusion, we will develop specifically independent technical for interpolation of temporal response characteristics on the step function and the pulse function, adapted and optimized to the type of specification that is intended to translate. In addition, a comparison of results will take place between interpolation techniques presented and resolution of the same problems with functions “solve” and “fsolve”.

3. Inverse Interpolation Technical on the Step Function

It will be used a technique of inverse interpolation on the step function (1), for the translation of tracking specifications to the frequency domain, for both the upper limit, as the lower. We will try to find the set of possible pairs (ω_n,ζ) associated with the underdamped second order functions that fulfill all temporary requirements for tracking required.

The technique described in this section is an outline of the inverse interpolation technique on the step function, which can be found developed with more detail in [14] and [15].

It is used automatic iteration process, within the range of subcritical damping and limited by the maximum overshoot imposed (i.e., $1>\zeta>\zeta_{min}$, with ζ_{min} corresponding to Mp), so that for each damping coefficient is sought:

1. On one hand, the rise time t_r specified, associated to the pair (ω_{nr},ζ). To do this, use the following relationship, demonstrated in [15]:

$$\frac{tr_1}{t_r}=\omega_{nr} \tag{3}$$

That is, given the initial temporary specification t_r and, calculated by interpolation tr_1 associated to the pair $(\omega_n=1,\zeta)$, is obtained from (3) the value ω_{nr}, related to t_r for the coefficient ζ.

2. On the other hand, it is sought the settlement time t_s required too, from which the response is adapted to a particular settlement channel and, which corresponds another pair (ω_{ns},ζ). To do this, use the next relationship obtained from [15].

$$\frac{ts_1}{t_s}=\omega_{ns} \tag{4}$$

So, with the temporary specification t_s, is calculated by interpolation ts_1 associated to the pair $(\omega_n=1,\zeta)$, and then the parameter ω_{ns} can get clearing in (4), related to t_s for the damping coefficient ζ.

3. Choose as possible valid pair (ω_n,ζ), the one which represents the most restrictive condition for temporary joint condition (t_r, t_s): for the lower frequency limit specification, ω_n will be the greater value between ω_{nr} and ω_{ns}; for the upper frequency limit specification, ω_n will be the lower value between both.
4. Determine times tr_{final} and ts_{final} associated with each possible pair (ω_n,ζ) found, again using on (1) inverse interpolation; if both tr_{final} and ts_{final} fulfill the initial requirements given for t_r and t_s, the pair (ω_n,ζ) associated will serve to generate a second order transfer function, possible solution in frequency.

Note that, in any case of interpolation, the natural frequency is a parameter given: initially $\omega_n=1$ or ω_n found later to verify possible validity of the pair (ω_n,ζ).

Obtaining the equations (3) and (4) is detailed in [15], and the sense of the steps outlined above, when attempting to translate temporary specifications associated with step-like responses.

3.1. Inverse Interpolation of the Rise Time $t_r(\omega_n,\zeta)$ on the Step Function

The rise time t_r of an underdamped step response is common to define between 10 and 90%, 5 to 95% or 0 to 100%, of its final value [21]. Here, we will apply the first interval (10 to 90%), because it is more complex, although for underdamped systems is more used the interval 0 to 100%. Therefore, using Alc as range percentage parameter of final steady value, t_r is defined associated to the pair *(ω_n, ζ)* as,

$$t_r(\omega_n,\zeta) = tr_{90}(\omega_n,\zeta, Alc = 90\%) - tr_{10}(\omega_n,\zeta, Alc = 10\%) \tag{5}$$

Thus, in this case, the statement is reduced to develop a technique for parameter interpolation type-like *tr_{Alc}(ω_n,ζ,Alc)*, on the second order step function *f(t)* with subcritical damping. Considering that for any time *t*, the function *f(t)* is described as in (1), then,

$$f(t) = f_{step}(t), \qquad \text{with} \quad 0<\zeta<1 \tag{6}$$

To tr_{Alc}, equation (6) takes the value,

$$f(tr_{Alc}) = \frac{Alc}{100} = 1 - \frac{e^{-\zeta\omega_n tr_{Alc}}}{\sqrt{1-\zeta^2}} sen\left(\omega_n\sqrt{1-\zeta^2}\,tr_{Alc} + a\cos\zeta\right) \tag{7}$$

If used for interpolation of *f(t)* the Newton's method by differences, accurately of 5th order, generic formulas to be applied are as follows [1]:

- In the case of "Descending Differences":

$$f(t_0+\theta h) = f(t_0) + \delta_d^{1°}\theta + \frac{\delta_d^{2°}}{2}\theta(\theta-1) + \frac{\delta_d^{3°}}{3!}\theta(\theta-1)(\theta-2) + \frac{\delta_d^{4°}}{4!}\theta(\theta-1)(\theta-2)(\theta-3) + \frac{\delta_d^{5°}}{5!}\theta(\theta-1)(\theta-2)(\theta-3)(\theta-4) \tag{8}$$

- In the case of "Ascending Differences":

$$f(t_0+\theta h) = f(t_0) + \delta_a^{1°}\theta + \frac{\delta_a^{2°}}{2}\theta(\theta+1) + \frac{\delta_a^{3°}}{3!}\theta(\theta+1)(\theta+2) + \frac{\delta_a^{4°}}{4!}\theta(\theta+1)(\theta+2)(\theta+3) + \frac{\delta_a^{5°}}{5!}\theta(\theta+1)(\theta+2)(\theta+3)(\theta+4) \tag{9}$$

Where *h* is the time interval chosen for differences δ^i of order *i* (1°,2°, ..). The general procedure consists of calculating the parameter θ per iteration. To do this, *θ=F(θ)* is cleared and assigned an initial value *F(θ_0)=θ_1*.

If θ_1 is appropriate, the process is terminated. Otherwise, continue assigning values *F(θ_1)=θ_2*, *F(θ_2)=θ_3*, .., until a result is obtained such that, $|\theta_i-\theta_{i-1}|<|\theta_{i-1}-\theta_{i-2}|$, with *i=1,2,..*

Then, clearing θ in equation (8) or (9); for example, in the descending differences (8) and, substituting, it has got,

$$\theta = \frac{f(t_0+\theta h) - f(t_0)}{\delta_d^{1°}} - \frac{\delta_d^{2°}}{2\delta_d^{1°}}\theta(\theta-1) + \frac{\delta_d^{3°}}{3!\delta_d^{1°}}\theta(\theta-1)(\theta-2) + \frac{\delta_d^{4°}}{4!\delta_d^{1°}}\theta(\theta-1)(\theta-2)(\theta-3) + \frac{\delta_d^{5°}}{5!\delta_d^{1°}}\theta(\theta-1)(\theta-2)(\theta-3)(\theta-4) \tag{10}$$

To start the iteration is chosen,

$$\theta_0 = \frac{f(t_0+\theta h) - f(t_0)}{\delta_d^{1°}} \tag{11}$$

For generic function (6) in question, the particular sequence of steps to be taken would be:

1. Define the time interval *h*. Since the maximum overshoot *Mp* is reached in time t_{Mp}, a valid practical value of interval is 1/10 of it. That is to say,

$$t_{Mp} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \Rightarrow \quad h = \frac{\pi}{10\omega_n\sqrt{1-\zeta^2}} \tag{12}$$

And as discrete times using the following,

$$t_i = \{t_1 = 0; t_2 = t_1 + h; ....; t_n = t_{n-1} + h\} \text{ , with } i=1,..,n \tag{13}$$

Later, we will see that the choice of interval h of 10% t_{Mp} is not random. For the moment, consider that to cover with intervals h the time between 0 (t_1) and t_{Mp} (t_{11}) is needed 11 time instants, ie, $n=11$.

2. Replace the values of earlier times (13) in the function *f(t)* (6), taking into account the known value of ω_n
3. Obtain the differences δ of 1st, 2nd, 3rd, 4th and 5th order, from the relationships expressed in a table of differences, such as the one built in Table 1. Note how to obtain a differences table of 5th order is necessary to sample 11 time instants. This is the reason to be for the chosen value h in (12). The user can increase the accuracy of interpolation using interpolation intervals h smaller, for example, 1/20 or 1/30 times the t_{Mp}
4. Find the t_0 parameter to define the initial parameter θ_0 (11), choosing the most appropriate differences procedure (ascending or descending), from the following criteria (for example in Table 1 above, with 11 time instants sampled),

If $f(t_i) > \frac{Alc}{100}$, with $i=1,2,..,6$, Then use,

- *Descending Differences* and, $f(t_0) = f(t_{i-1}) < f(t)$, with $t_i < t < t_{i+1}$ (14)

That is, it is the nearest "point" with less value than the desired one.

If $f(t_i) > \frac{Alc}{100}$, with $i=7,8,..,11$. Then use,

- *Ascending Differences* and, $f(t_0) = f(t_i) > f(t)$, with $t_{i-1} \le t < t_i$ (15)

That is, the nearest "point" with higher value than desired.

Being t the time to find, in this case defined as tr_{Alc}

Table 1. Differences δ for functions *f(t)* of 1st, 2nd, 3rd, 4th and 5th order

f(t)	*1st Order*	*2nd Order*	*3rd Order*	*4th Order*	*5th Order*
$f(t_1)$	$\delta_{11}= f(t_2)- f(t_1)$	$\delta_{21}=\delta_{12}-\delta_{11}$	$\delta_{31}=\delta_{22}-\delta_{21}$	$\delta_{41}=\delta_{32}-\delta_{31}$	$\delta_{51}=\delta_{42}-\delta_{41}$
$f(t_2)$	$\delta_{12}= f(t_3)- f(t_2)$	$\delta_{22}=\delta_{13}-\delta_{12}$	$\delta_{32}=\delta_{23}-\delta_{22}$	$\delta_{42}=\delta_{33}-\delta_{32}$	$\delta_{52}=\delta_{43}-\delta_{42}$
$f(t_3)$	$\delta_{13}= f(t_4)- f(t_3)$	$\delta_{23}=\delta_{14}-\delta_{13}$	$\delta_{33}=\delta_{24}-\delta_{23}$	$\delta_{43}=\delta_{34}-\delta_{33}$	$\delta_{53}=\delta_{44}-\delta_{43}$
$f(t_4)$	$\delta_{14}= f(t_5)- f(t_4)$	$\delta_{24}=\delta_{15}-\delta_{14}$	$\delta_{34}=\delta_{25}-\delta_{24}$	$\delta_{44}=\delta_{35}-\delta_{34}$	$\delta_{54}=\delta_{45}-\delta_{44}$
$f(t_5)$	$\delta_{15}= f(t_6)- f(t_5)$	$\delta_{25}=\delta_{16}-\delta_{15}$	$\delta_{35}=\delta_{26}-\delta_{25}$	$\delta_{45}=\delta_{36}-\delta_{35}$	$\delta_{55}=\delta_{46}-\delta_{45}$
$f(t_6)$	$\delta_{16}= f(t_7)- f(t_6)$	$\delta_{26}=\delta_{17}-\delta_{16}$	$\delta_{36}=\delta_{27}-\delta_{26}$	$\delta_{46}=\delta_{37}-\delta_{36}$	$\delta_{56}=\delta_{47}-\delta_{46}$
$f(t_7)$	$\delta_{17}= f(t_8)- f(t_7)$	$\delta_{27}=\delta_{18}-\delta_{17}$	$\delta_{37}=\delta_{28}-\delta_{27}$	$\delta_{47}=\delta_{38}-\delta_{37}$	
$f(t_8)$	$\delta_{18}= f(t_9)- f(t_8)$	$\delta_{28}=\delta_{19}-\delta_{18}$	$\delta_{38}=\delta_{29}-\delta_{28}$		
$f(t_9)$	$\delta_{19}= f(t_{10})- f(t_9)$	$\delta_{29}=\delta_{110}-\delta_{19}$			
$f(t_{10})$	$\delta_{110}= f(t_{11})- f(t_{10})$				
$f(t_{11})$					

Descending differences: Start at the last value *f(t11)*
Ascending differences: Start at the first value *f(t1)*

5. Calculate initial θ_0, using the expressions,

$$f(t_0+\theta h)=\frac{Alc}{100}\Rightarrow\ \theta_0=\frac{\frac{Alc}{100}-f(t_0)}{\delta_{1i}} \tag{16}$$

6. Calculate the values θ_i $(i=1,2,..)$ and the error in each iteration: it is used to control the correct or not loop output $(Error_{i-1}=|\theta_{i-1}-\theta_i|)$, besides the accuracy of the approximation.

If descending differences, use:

$$\begin{cases}\theta_1=\theta_0-\frac{\delta_d^{2^o}}{2\delta_d^{1^o}}\theta_0(\theta_0-1)-\frac{\delta_d^{3^o}}{3!\delta_d^{1^o}}\theta_0(\theta_0-1)(\theta_0-2)-\frac{\delta_d^{4^o}}{4!\delta_d^{1^o}}\theta_0(\theta_0-1)(\theta_0-2)(\theta_0-3)-\frac{\delta_d^{5^o}}{5!\delta_d^{1^o}}\theta_0(\theta_0-1)(\theta_0-2)(\theta_0-3)(\theta_0-4) & (17)\\ Error_0=|\theta_0-\theta_1| & (18)\end{cases}$$

$$\begin{cases}\theta_2=\theta_1-\frac{\delta_d^{2^o}}{2\delta_d^{1^o}}\theta_1(\theta_1-1)-\frac{\delta_d^{3^o}}{3!\delta_d^{1^o}}\theta_1(\theta_1-1)(\theta_1-2)-\frac{\delta_d^{4^o}}{4!\delta_d^{1^o}}\theta_1(\theta_1-1)(\theta_1-2)(\theta_1-3)-\frac{\delta_d^{5^o}}{5!\delta_d^{1^o}}\theta_1(\theta_1-1)(\theta_1-2)(\theta_1-3)(\theta_1-4) & (19)\\ Error_1=|\theta_1-\theta_2| & (20)\end{cases}$$

Apply the following process:

$$\begin{cases}\text{If } Error1<Eps\text{, then } \theta=\theta_2 \text{ and } t=t_0+\theta h\\ \text{If } Error1\geq Eps\text{, then continue iteration.}\end{cases}$$

Note 1: The accuracy obtained depends on the 'Eps' user-defined parameter, that tells you the maximum error allowed in the iteration of applied research.

If ascending differences, use:

$$\begin{cases}\theta_1=\theta_0-\frac{\delta_a^{2^o}}{2\delta_a^{1^o}}\theta_0(\theta_0+1)-\frac{\delta_a^{3^o}}{3!\delta_a^{1^o}}\theta_0(\theta_0+1)(\theta_0+2)-\frac{\delta_a^{4^o}}{4!\delta_a^{1^o}}\theta_0(\theta_0+1)(\theta_0+2)(\theta_0+3)-\frac{\delta_a^{5^o}}{5!\delta_a^{1^o}}\theta_0(\theta_0+1)(\theta_0+2)(\theta_0+3)(\theta_0+4) & (21)\\ Error_0=|\theta_0-\theta_1| & (22)\end{cases}$$

$$\begin{cases}\theta_2=\theta_1-\frac{\delta_a^{2^o}}{2\delta_a^{1^o}}\theta_1(\theta_1+1)-\frac{\delta_a^{3^o}}{3!\delta_a^{1^o}}\theta_1(\theta_1+1)(\theta_1+2)-\frac{\delta_a^{4^o}}{4!\delta_a^{1^o}}\theta_1(\theta_1+1)(\theta_1+2)(\theta_1+3)-\frac{\delta_a^{5^o}}{5!\delta_a^{1^o}}\theta_1(\theta_1+1)(\theta_1+2)(\theta_1+3)(\theta_1+4) & (23)\\ Error_1=|\theta_1-\theta_2| & (24)\end{cases}$$

Apply the following process:

$$\begin{cases}\text{If } Error1<Eps\text{, then } \theta=\theta_2 \text{ and } t=t_0+\theta h\\ \text{If } Error1\geq Eps\text{, then continue iteration.}\end{cases}$$

Note 2: For each θ_i $(i=1,2,..)$, then $\delta^{1^o}=\delta_{1i}$, $\delta^{2^o}=\delta_{2i}$, ...

3.2. Inverse Interpolation of the Time $t_s(\omega_n,\zeta,dev)$ on the Step Function

The search procedure for settling time t_s required is, in principle, the same as described for the previous rise time t_r, except for the following special considerations:

- With the parameter *dev*, it is defined the settlement channel with values in unitary percentage in the range given by,

$$1 \pm \frac{desv}{100} \tag{25}$$

- To obtain the settling time, it is going to observe the behavior of each signal semicycles at the zero-maximum_peak interval, for the first and, maximum_peak-minimum_peak, for the others:
  - If the first maximum peak is within the settlement channel, the settling time will be in the first semicycle, in the ascending interval (rise), before the peak considered.
  - If not, when a full semicycle (maximum_peak-minimum_peak) is within the settlement channel, time sought will be within the previous semicycle; otherwise, the settling time will be the time in reaching firstly the settlement channel plus the time sum of every semicycles $n\pi / \omega_n \sqrt{1-\zeta^2}$, until the signal is within the channel.

The inverse interpolation of t_s is performed between *0* and t_1 (first semicycle), t_1 and t_2 (second semicycle) and, so on until you find the appropriate value (Figure 3).

It is true that for the determination of establishment time you can use the envelope of the function described in (1), without losing too much precision and, therefore, no numerical method would be necessary to find these parameters. However, the interpolation method proposed has been developed to search for rise times, impossible to determine accurately through the envelope said and, its use takes advantage of determining with more accurate t_s or combinations type-like t_s-t_r, such as those that will be proposed in section 4 on the pulse function.

3.3. Example application of Inverse Interpolation on the Step Function

In the example of Figure 2, it has been used the application TD2WD.M, as indicated in Section 5, for obtaining the lower frequency limit in tracking. Resolved (6) applying different procedures,

1. The inverse interpolation technique described in Section 3.
2. The function 'solve' on independent equations with one unknown (ω_n and ζ known), combined with the procedure associated with equations (3) and (4).
3. The 'fsolve' function with one unknown equations, combined with the procedure associated with (3) and (4).
4. The 'solve' function on a system of four equations with unknowns, $tr_{10}(\omega_n,\zeta,Alc=10\%)$, $tr_{90}(\omega_n,\zeta,Alc=90\%)$, ω_n and $t_s(\omega_n,\zeta)$. The time t_s, in this case, is described from the equation of the envelope on the step function, i.e. with,

$$f(t_s) = 1 - \frac{e^{-\zeta\omega_n t_s}}{\sqrt{1-\zeta^2}} \tag{26}$$

5. The 'fsolve' function on a system of four equations with the unknowns indicated in paragraph 4 above.

A summary of the results achieved is described in comparison between procedures of Table 2, where it is given the deviation obtained with each, as to the rise time ($f(t_r)$-$f(t)$) and the settlement time ($f(t_s)$-$f(t)$).

a) Temporary specifications required were: 30% overshoot; maximum rise and settlement times of 5s and 20s, respectively, on a ±4% settlement channel.

Table 2. Results and Deviations Obtained with the 5 Procedures Proposed, for the Example in Section 3.3

Method	*f(t)=0.9*			*f(t)=0.1*			*f(t)=[0.96,1.04]*		
	t_r(s)	*$f(t_r)$*	*$f(t_r)$- f(t)*	*t_r(s)*	*$f(t_r)$*	*$f(t_r)$- f(t)*	*t_s(s)*	*$f(t_s)$*	*$f(t_s)$- f(t)*
1	*6.4993*	*0.90116*	*0.00116*	*1.4993*	*0.0953*	*-0.0047*	*17.0061*	*1.04*	*0*
2	*10.0268*	*0.30058*	*-0.5994*	*5.02678*	*0.1*	*0*	*10.2866*	*0.31177*	*-0.6482*
3	*3.6802*	*0.9*	*0*	*0.52787*	*0.1*	*0*	*20*	*1*	*0.04*
4	*No possible*	*-*	*-*	*No possible*	*-*	*-*	*No possible*	*-*	*-*
5	*5.8965*	*0.9*	*0*	*0.89645*	*0.1*	*0*	*20*	*1*	*0.04*

1, Inverse interpolation by Newton's method of 5th order
2, Function 'solve' on independent equations
3, Function 'fsolve' on independent equations
4, Function 'solve' on system of equations
5, Function 'fsolve' on system of equations

b) Inverse interpolation response by Newton's method. Results shown in Figure 2a:

- o Maximum time response characteristics:
  - ▪ Final *Mp* of *19.83%.*
  - ▪ Rise time calculated of 5s. Good deviations and perfect approach result to t_r demanded.
  - ▪ Settling time calculated of 17s. Exact approach, with no deviation.

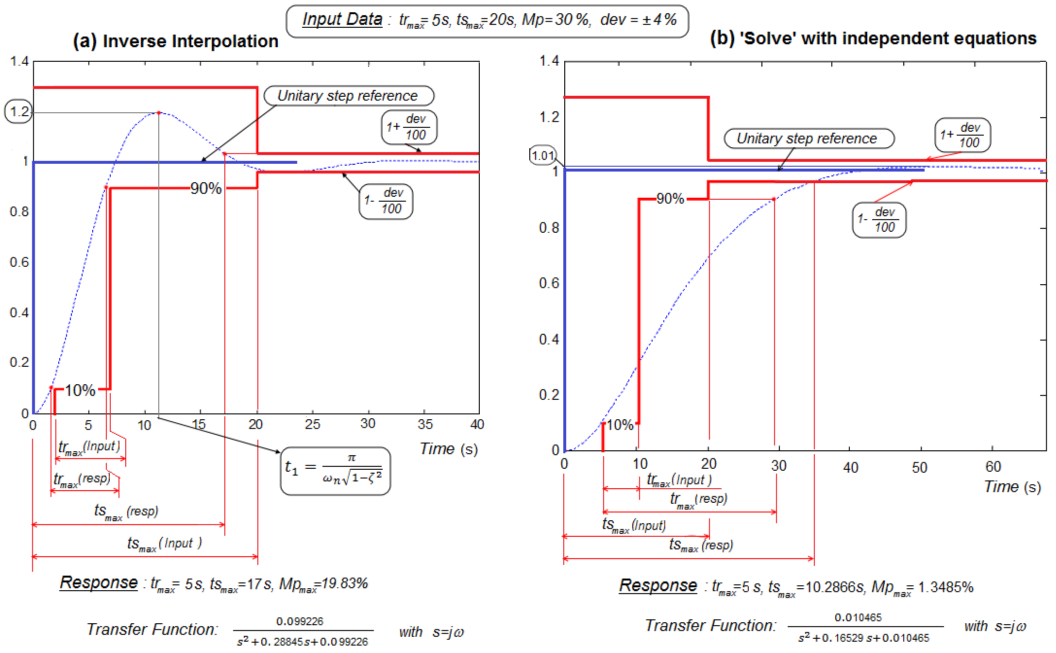

Figure 2. Example of Tracking Specification with Maximum Time Response Characteristics, (a) Approximated by Inverse Interpolation, (b) Approximated Applying 'Solve' with Independent Equations

c) The results obtained with the other four proposed comparative methods are:

- Response applying 'solve' with independent equations, shown in Figure 2b:
  - o Maximum time response characteristics:
    - ▪ Final *Mp* of *1.35%.*
    - ▪ Rise time calculated of 5s. But, because of the enormous deviation obtained for *f(t)= 0.9* (*90%*), the *t_r(resp)* is almost 25s.

    - Settling time calculated of 10.29s. With the large deviation obtained for *f(t)=0.96*, t_s*(resp)* is more than 35s.

- Response applying ‘fsolve’ with independent equations, shown in Figure 3a:
  - Maximum time response characteristics:
    - Final *Mp* of *0.0028%*.
    - Rise time calculated of 3.15s. Perfect approach result to t_r demanded.
    - Settling time calculated of 20s. With the bad deviation obtained for *f(t)=0.96*, t_s*(resp)* is about 5s for t_s*(calculated)* of 20s.
- Response using ‘solve’ with system of equations: For this example, there is no solution.
- Response applying ‘fsolve’ with equation system, shown in Figure 3b:
  - Maximum time response characteristics:
    - Final *Mp* of *0.11%*.
    - Rise time calculated of 5s. Very good result that matches the required t_r.
    - Settling time calculated of 20s. Bad results obtained for *f(t)=0.96*, so t_s*(resp)* is about 8s with t_s*(calculated)* of 20s.

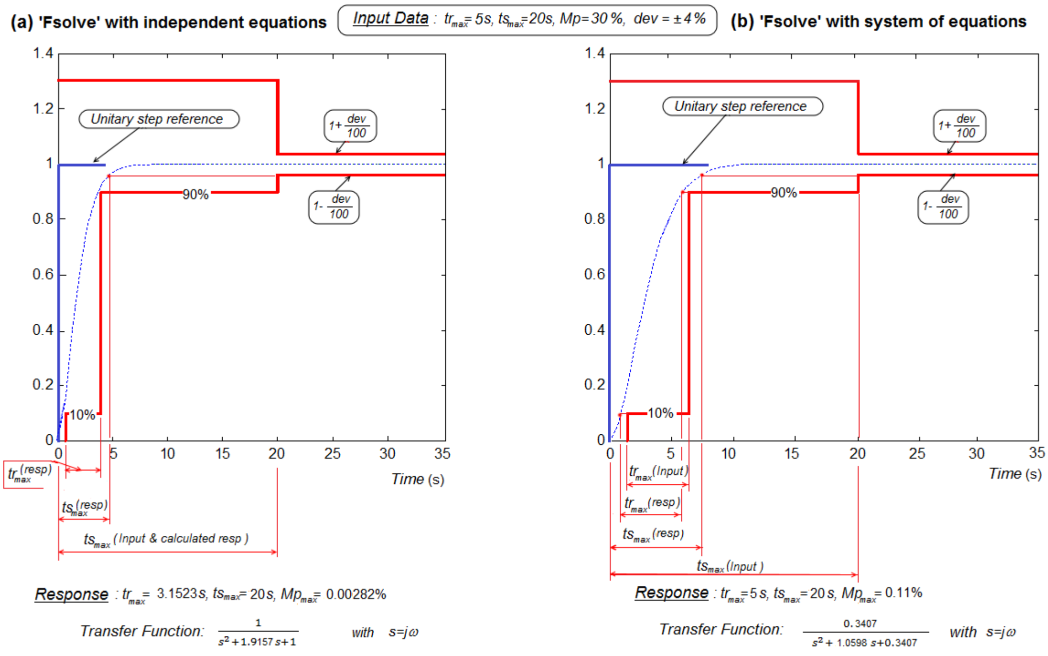

Figure 3. Example of Tracking Specification with Maximum Time Response Characteristics, (a) Approximated Applying 'Fsolve' with Independent Equations, (b) Approximated Applying 'Fsolve' with System of Equations

d) There is no any procedure comparable with the inverse interpolation proposed. Approaches for t_r are good using ‘fsolve’ application, but no so to obtain t_s. In general, results are not practical, except for inverse interpolation.

4. Inverse Interpolation Technical on the Pulse Function

For translation of sensitivity specifications in the frequency domain, in both generic and specific, as defined in section 2, is applying an inverse interpolation technique on the pulse function (2). We will try to find the set of possible pairs (ω_n,ζ) representing the second order functions, which fulfill all temporary requirements for regulation imposed.

As is the case in section 3, the technique described here is an outline of the inverse interpolation technique on the pulse function, developed in detail in [14].

It is used an automatic iteration process, within the range of subcritical damping; i.e. $1>\zeta>0$, so that for each damping coefficient and depending on the sensitivity type considered, is sought:

1. For generic sensitivity, you only need to work with settling time t_s, which must not exceed the specified T, associated with the pair (ω_n,ζ). To do this, use the relationship:

$$\frac{ts_1}{t_s}=\omega_n \tag{27}$$

That is, given $t_s=T$ as time specification and ts_1 associated with the pair $(\omega_n=1,\zeta)$, calculated by interpolation, it is obtained from (27) the parameter ω_n, related to t_s for the damping coefficient ζ. The validity of the pair (ω_n,ζ) is checked, applying again inverse interpolation on the function (2), to obtain ts_{final}, such that:

- If $ts_{final}(\omega_n,\zeta)\leq T$ and maximum amplitude imposed is not exceeded, the pair (ω_n,ζ) will be a possible second order solution in frequency.

2. For specific sensitivity, it is necessary to use inverse interpolation as on settling time t_s as the rise time t_r, although associated in both cases at the same amplitude on the pulse function (2). To do this, use the relationship described by:

$$\frac{ts_1-tr_1}{t_s}=\omega_n \tag{28}$$

That is, given the time specification $t_s=T$, it is calculated by interpolation $(ts_1\text{-}tr_1)$, time difference associated with the same pair $(\omega_n=1,\zeta)$, clearing in (28) the parameter ω_n, related with t_s for the coefficient ζ. Note that you have to find separate ts_1 and tr_1, so interpolation is used twice. The pair (ω_n,ζ) obtained is checked by applying again inverse interpolation on the function (2) for getting $(t_s\text{-}t_r)_{final}$, such that:

- If $(t_s\text{-}t_r)_{final}(\omega_n,\zeta)\leq T$ and maximum amplitude imposed is not exceeded, the pair (ω_n,ζ) can be a possible solution.

Obtaining the equations (27) and (28) is detailed in [14] and, the sense of the steps outlined above, when trying to translate temporary specifications associated with pulse type responses. On the other hand, the approach is equivalent to that defined in section 3, when working with step functions.

4.1 Inverse Interpolation of the Rise Time $t_r(\omega_n,\zeta,Alc)$ on the Pulse Function

It is about finding the rise time t_r on pulse signal $f(t)$, when it takes the rising value "*Alc*" and is defined by the parameters ω_n and ζ. The approach reduces to the technique development for parameter interpolation type-like $tr_{Alc}(\omega_n,\zeta,Alc)$, on the second order pulse function $f(t)$ with subcritical damping.

The function $f(t)$ is defined for each time t, as in (2), that is:

$$f(t)=f_{pulse}(t)\ \text{, with } 0<\zeta<1 \tag{29}$$

For tr_{Alc}, equation (29) takes the value,

$$f(tr_{Alc})=\frac{Alc}{100}=\frac{\omega_n}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n tr_{Alc}}sen\left(\omega_n\sqrt{1-\zeta^2}\,tr_{Alc}\right) \tag{30}$$

Interpolation procedure by Newton's differences of 5th order on *$f(tr_{Alc})$* is the same as described in section 3.1, with the following exceptions:

1. Time interval *h* of differences interpolation. Considering that the maximum peak value f_{max} is reached at the time instant t_{max} given by,

$$t_{\max} = \frac{\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)}{\omega_n\sqrt{1-\zeta^2}} \tag{31}$$

   A valid practical value of interval *h* is 1/10 of the time t_{max}, for the same reason to define the interpolation interval used in section 3.1. That is,

$$h = \frac{t_{\max}}{10} \tag{32}$$

   And use as discrete times those given in (13).
2. Substitute the values of previous times (13) in the interpolation function *f(t)* (29), given the known value ω_n.
   From here, the interpolation technique follows the same steps as in section 3.1, i.e.:
3. Get the values δ of 1st, 2nd, 3rd, 4th and 5th order, using Table 1 of differences.
4. Find the t_0 parameter, using the criteria described in section 3.1 from equations (14) and (15).
5. Calculate initial θ_0 using (16) and the value *h* in (32).
6. Calculate θ_i values ($i=1,2, ..$) and the error in each iteration ($Error_{i-1}=|\theta_{i-1}-\theta_i|$), by a method as indicated between equations (17) and (24).

4.2. Inverse Interpolation of the Time $t_s(\omega_n,\zeta,As)$ on the Pulse Function

The search procedure for settling time t_s required, for the pulse function, is similar to that described for the rise time t_r above, in section 4.1, except for the following special considerations (see Figure 4a):

- The "*As*" parameter defines the settlement channel of pulse response with values in unitary percentage within the range given by,

$$\pm\frac{As}{100} \tag{33}$$

- The settling time is obtained by observing the behavior of each signal semicycles in the range, peak (positive maximum) and valley (negative minimum) or valley-peak:
  - If the first peak is within the settlement channel, then settling time t_s is zero.
  - If not, when a full semicycle (peak-valley or valley-peak) is located within the settlement channel, other than the first, the time t_s will be associated with the previous semicycle; otherwise, the settling time will be this time, plus the sum of time of each semicycle $\frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$, until the peak or valley of the function is within the channel.
  - The temporal interpolation interval *h* is provided between a peak or valley and the next zero function, temporarily located on t_p and t_c, respectively. Where,

$$t_p = \frac{\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)+n\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{34}$$

$$t_c = \frac{n\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{35}$$

Both with $n=0, 1, ...$ Note that $t_p(n=0)=t_{max}$

The inverse interpolation of t_s within the range h (t_p to t_c) is between t_p and t_1, t_1 and t_2, and so on, until completion in t_c (t_1, t_2, .., are discrete times given in (13)) .

4.3. Example application of Inverse Interpolation on the Pulse Function

In the examples of Figure 4 and Figure 5, it has been used the application ESENSIB.M, accessible from section 5, to obtain the lower frequency limit for specific sensitivity. Resolved (29) using the following procedures:

1. The inverse interpolation technique described in Section 4.
2. The function 'solve' on independent equations with one unknown (ω_n and ζ known), combined with the procedure associated with equation (28).
3. The 'fsolve' function with one unknown equations, combined with the procedure associated with (28).
4. The 'solve' function on a system of three equations with unknowns, $tr_{Alc}(\omega_n,\zeta,Alc=Out)$, ω_n and $t_s(\omega_n,\zeta)$. The time t_s, in this case, is described from the equation of the envelope on the pulse function, i.e. with,

$$f(ts) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n ts} \tag{36}$$

5. The 'fsolve' function on a system of three equations with the unknowns indicated in the preceding paragraph.

Table 3: Results and Deviations Obtained with the 5 Proposed Procedures, for Example in Section 4.3

Method	*f(t)=0.25*			*f(t)=±0.25*		
	$t_r(s)$	$f(t_r)$	$f(t_r)- f(t)$	$t_s(s)$	$f(t_s)$	$f(t_s)- f(t)$
1	*0.38*	*0.2498*	*-0.0002*	*10.05*	*0.2501*	*0.0001*
2	*0.45*	*0.25*	*0*	*6.31*	*-0.3399*	*-0.0899*
3	$8.9*10^{-5}$	$5.4*10^{-5}$	*-0.2500*	*5.91*	*-0.3562*	*-0.1062*
4	*0.37*	*0.25*	*0*	*10.37*	*0.1613*	*-0.0887*
5	*0.54*	*0.2490*	*-0.0010*	*10.54*	*0.2090*	*-0.0411*

1, Inverse interpolation by Newton's method of 5th order
2, Function 'solve' on independent equations
3, Function 'fsolve' on independent equations
4, Function 'solve' on system of equations
5, Function 'fsolve' on system of equations

Note: In Table 3 is being used *f(t)=Out/In*

A summary of the results achieved is described in comparison between procedures of Table 3, where it is given the deviation obtained with each, as to the rise time ($f(t_r)$-$f(t)$) and the settlement time ($f(t_s)$-$f(t)$).

a) Temporary specifications required are: amplitude of the pulse type response ranging between 1.5 and 4.35, for no more than 10s. Therefore, the settlement channel is in the range ± 1.5.

b) Response by inverse interpolation, Newton's method. Results are shown in Figure 4a:
   - o Maximum time response characteristics:
     - ▪ $f_{max}=4.153$ which does not exceed the value imposed of 4.35
     - ▪ Rise time calculated response 0.38s. For $f(t)=1.5$ (Out or $Out/In=0.25$), it is obtained the tr_{out} indicated in the first semicycle, between 0 and t_{c1}.
     - ▪ Settling Time calculated response, 10.05s. For $f(t)=\pm 1.5$ (settlement channel) the obtained ts_{out} produces $T_{out}=ts_{out}-tr_{out}=9.67s \leq 10s$.The first semicycle full enter into settlement channel is number 4.

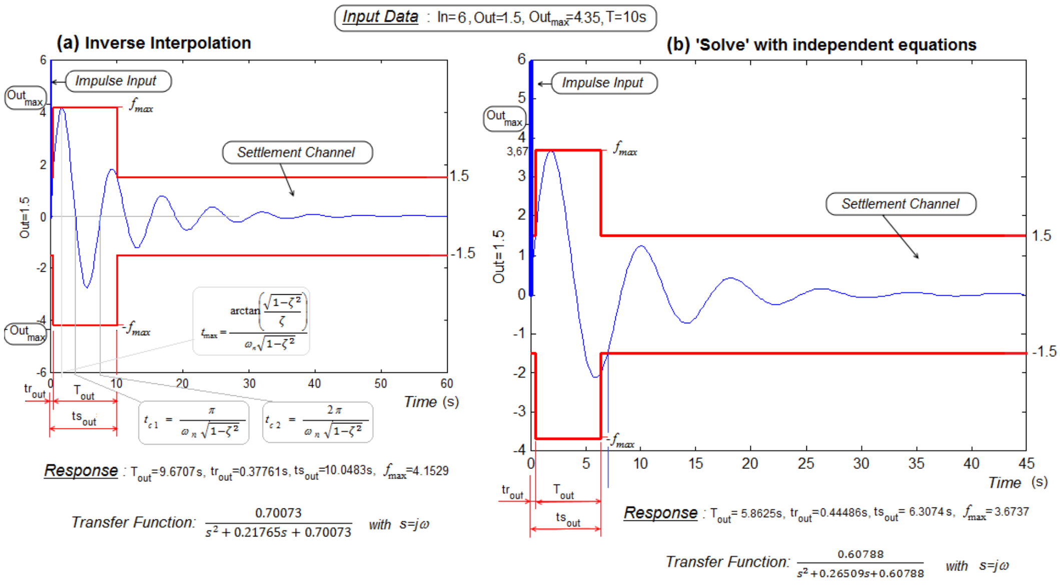

Figure 4. Example of Sensitivity Specification with Characteristics of Maximum Response, (a) Approximated by Inverse Interpolation, (b) Applying 'Solve' with Independent Equations

c) The results obtained with the other four proposed comparative methods are:

- • Response applying 'solve' with independent equations, shown in Figure 4b:
  - o Maximum time response characteristics:
    - ▪ $f_{max}=3.674$ which does not exceed the value imposed of 4.35
    - ▪ Rise time calculated response, 0.45s, for $f\,(t)=1.5$. No deviation.
    - ▪ Settling time calculated response, 6.31s. For $f(t)=\pm 1.5$, it is obtained $T_{out}=ts_{out}-tr_{out}=5.86s \leq 10s$. High deviation, so the actual T_{out} does not match.
- • Response applying 'fsolve' with independent equations, shown in Figure 5a:
  - o Maximum time response characteristics:
    - ▪ $f_{max}=3.674$ which does not exceed the value imposed of 4.35
    - ▪ Rise time calculated response, $9*10^{-5}s$, for $f(t)=1.5$. Very high deviation, so the tr_{out} does not match the real.
    - ▪ Settling Time calculated response, 5.91s. For $f(t)=\pm 1.5$, it is obtained $T_{out}=ts_{out}-tr_{out}=6.91s \leq 10s$. High deviation, so T_{out} differs from the actual.

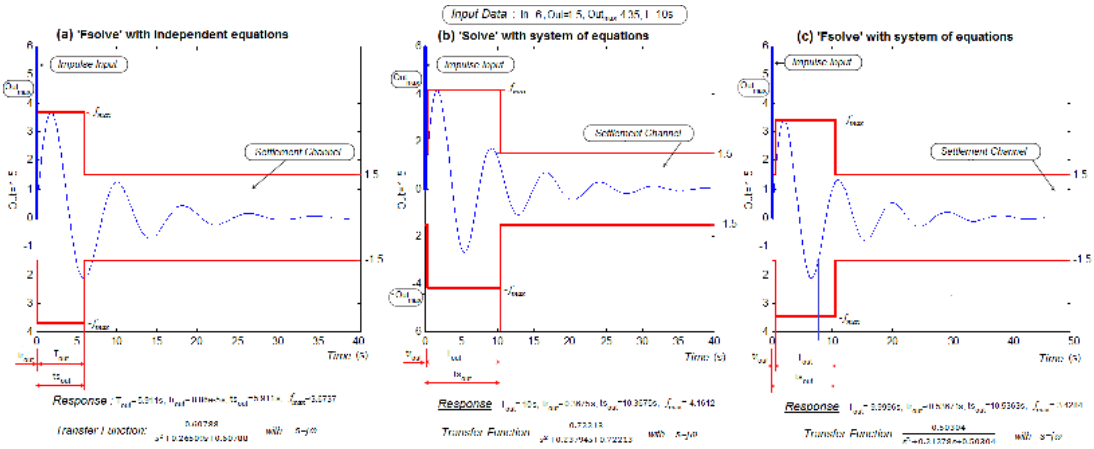

Figure 5. Example of Sensitivity Specification with Characteristics of Maximum Temporal Response, (a) Approximated Applying 'Fsolve' with Independent Equations, (b) Approximated Applying 'Solve' with System of Equations, (c) Approximated Applying 'Fsolve' with System of Equations

- Response using 'solve' with system of equations, shown in Figure 5b:
  - o Maximum time response characteristics:
    - ▪ $f_{max}=4.161$ which does not exceed the value imposed of 4.35
    - ▪ Rise time calculated response, 0.3675s, for $f(t)=1.5$. No deviation.
    - ▪ Settling Time response 10.3675s. For $f(t)=\pm 1.5$, it is obtained $T_{out}=ts_{out}-tr_{out}=10s \leq 10s$. High deviation, so the actual T_{out} does not match.
- Response applying 'fsolve' with equation system, shown in Figure 5c:
  - o Maximum time response characteristics:
    - ▪ $f_{max}=3.428$ which does not exceed the value imposed of 4.35
    - ▪ Rise time calculated response, 0.5367s, for $f(t)=1.5$. Correct deviation.
    - ▪ Settling Time calculated response 10.5363s. For $f(t)=\pm 1.5$, it is obtained $T_{out}=ts_{out}-tr_{out}=9.9996s \leq 10s$. Middle deviation, so T_{out} differs from the actual.

d) Noting that data obtained, maximum precision and final results are given by inverse interpolation, against the implementation of 'solve' and 'fsolve': compare deviations shown in Table 3 and graphically in Figure 4 and Figure 5.

5. Application Development

The author has developed in Matlab [2] applications to verify all examples given in this paper and also serve to translate the main temporary specifications to frequency domain already named.

The "Specifications Translation" folder contains the various techniques described for upper/lower limit tracking and generic/specific sensitivity, which can be downloaded from the address [22],

6. Conclusions

It has been discussed and presented sufficient arguments to justify the importance of having a methodology for temporary specifications translation to frequency domain with precise, automatic and specific character, for each type of requirement. To get it, a solution is to use successive approximations to the temporal response characteristic sought (rise time and/or settling time), applying an inverse interpolation method.

Specific techniques for specifications translation presented will be based on the "Newton's method by ascending and descending differences of 5th order", advantageous in practice against other traditional use for interpolation too, as can be Lagrange's polynomial.

Justified the need to use successive approximations by interpolation for specifications translation, next is to consider the type of function and characteristics of temporal response to interpolate. The function is directly related to the type of specification to be treated. Thus, for responses based on specifications for tracking step inputs, such as tracking lower limits and/or higher, the function is the step type. For sensitivity specifications, both general and specific, are based on underdamped pulse type responses. In any case, the function will be that descriptive of the step signal or pulse signal, with subcritical damping ($0<\zeta<1$) and approximated by a second order system.

In this paper, it has been developed independent techniques, for interpolation of time response characteristics on the step function and the pulse function, adapted to the type of specification that is intended to translate.

The goodness of each is demonstrated by the proposed practical examples and comparative results raised, respect to the implementation of functions for direct solving equations, 'solve' and 'fsolve' in Matlab.

On the other hand, applications developed for each type of specification are offered with free access by the author, from the URL indicated in the previous section.

References

[1] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal and C. A. Sagastizábal, "Numerical optimization: Theoretical and practical aspects", Universitext Berlin: Springer-Verlag,. DOI:10.1007/978-3-540-35447-5. ISBN 3-540-35445-X. MR 2265882, **(2006)**, pp. xiv+490.

[2] C. Borghesani, Y. Chait and O. Yaniv, "The QFT control desing Toolbox - for use with MATLAB", Terasoft Inc., **(2003)**

[3] J. D'Azzo and C. Houpis, "Linear control systems analysis and design: conventional and modern", New York, USA: McGraw-Hill, 4ªEd, Cap18, **(1995)**

[4] M. A. Franchek and P. A. Herman, "Direct connection between time-domain performance and frequency-domain characteristics", International Journal of Robust and Nonlinear Control, 8(12), **(1998)**, pp. 1021-1042.

[5] M. A. Franchek, P.A. Herman and O. D. I. Nwokah, "Robust nondiagonal controllers design for uncertain multivariable regulating systems", J. Dyn Syst-T ASME, 119, **(1997)**, pp. 80-85.

[6] M. A. Franchek and O. D. I. Nwokah, "Robust multivariable control of distillation columns using nondiagonal controller matrix", IMECE, ASME Journal of Dynamic Systems and Control, 57-1, **(1995)**, pp. 257-274.

[7] GARTEUR Action Group FM(AG08), "Robust flight control design challenge. Problem formulation and manual: the research civil aircraft model (RCAM)", GARTEUR/TP-088-3, **(1997)**

[8] M. Gil-Martínez, "Síntesis de controladores robustos mediante el análisis de la compatibilidad de especificaciones e incertidumbre", Ph.D. thesis, Universidad de la Rioja, Spain, **(2008)**.

[9] I. M. Horowitz and M. Sidi, "Synthesis of feedback systems with large plant ignorance for prescribed time domain tolerances", International Journal of Control, 16, **(1972)**, pp. 287-309.

[10] C. H. Houpis and S. J. Rasmussen, "Quantitative Feedback Theory: fundamentals and applications", Marcel Dekker Inc, New York, **(1999)**.

[11] S. R. K. Iyengar, "Numerical methods and computation. Interpolation and approximation", Lecture Nº 28, Dep. of Mathematics, Indian Institute of Technology, Delhi, **(2008)**

[12] S. Jayasuriya and M. A. Franchek, "Frequency domain design for maximal rejection of persistent bounded disturbances", J. Dyn Syst-T ASME, 113(2), **(1991)**, pp. 195-205**.**

[13] J. Joglar and J. Aranda, "QFT multivariable control for the longitudinal dynamics of an air vehicle", The 9th International Conference on Electrical and Control Technologies, ECT2014, Lithuania, **(2014)**, pp. 30-35**.**

[14] J. Joglar-Alcubilla, "Contribución al estudio de diseño de sistemas de control mediante QFT: aplicaciones al diseño de sistemas de control de vuelo y navegación", Ph.D. thesis, ETSI Informática, UNED. Spain, **(2015)**, March 10.

[15] J. Joglar-Alcubilla, "Generic tracking specifications translation from time domain to frequency domain", ArXiv:1601.00346 [cs.SY], **(2016)**, Jan16.

[16] J. Joglar Alcubilla, "Interpolación inversa de características de respuesta transitoria temporal a parámetros de la función de transferencia típica de segundo orden", Revista Iberoamericana de Automática e Informática Industrial, 14(1), **(2017)**, pp. 44-55, DOI: 10.1007/978-3-540-35447-5

[17] K. R. Krishnan and A. Cruickshanks, "Frequency-domain design of feedback systems for specified insensitivity of time-domain response to parameter variation", International Journal of Control, 25(4), **(1977)**, pp. 609-620.

[18] T. Kobylarz , I. M. Horowitz and C. H. Houpis, "Flight controller design for uncertain and nonlinear plants with pilot compensation", International Journal of Robust and Nonlinear Control, 2(3), **(1992)**, pp. 183-209.

[19] G. Mateos-Aparicio Morales, "Métodos de interpolación y ajuste. Elementos de matemática actuarial", UCM, ISBN 9788493811204, Madrid. www.ucm.es/info/sevipres/P4/01/ANEXOS01.php, **(2010)**

[20] P. S. V. Nataraj, "Computation of QFT bounds for robust tracking specifications", Automática 38, **(2002)**, pp 327-334.

[21] K. Ogata, "Ingeniería de control moderna", Prentice Hall, **(1993)**

[22] Translation, http://es.mathworks.com/matlabcentral/fileexchange/59068-specifications-translation, **(2016)**

[23] S. F. Wu, M. J. Grimble and S.G. Breslin, "Introduction to quantitative feedback theory for lateral robust flight control systems design", Control Engineering Practice 6, **(1998)**, pp. 805-828.

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

Author

Javier Joglar-Alcubilla works as a professor in the avionics department of the Barajas Institute in Madrid, Spain. He received his B.Sc in Aeronautics Engineering from Polytechnic University at Madrid, Spain, in 1990 and his M.Sc degree in Electronics and Automation Physics from UNED, Spain, in 1994. He received his Ph.D. degree in Informatics Engineering from ETSII at UNED, Spain, in 2015. He is the author of nine technical books.