Muñoz, J., Monje, C. A., Nagua, L. F. & Balaguer, C. (2020). A graphical tuning method for fractional order controllers based on iso-slope phase curves. *ISA Transactions, 105*, 296–307.

DOI: 10.1016/j.isatra.2020.05.045

A graphical tuning method for fractional order controllers based on iso-slope phase curves

Abstract

Fractional order controllers are widely used in the robust control field. As a generalization of the ubiquitous PID controllers, fractional order controllers are able to reach design specifications their integer counterparts cannot, and as a result they outperform them at particular situations. Their main drawback is that generalization of the design tools is not always evident, and therefore tuning this kind of controller is always a new and different challenge. Existing methods often use numerical computation to find the controller parameters that fit the specifications. This paper describes a graphical solution for fractional order controllers, which avoids the solution by nonlinear equations and helps designer to solve the control problem in a very intuitive way. This approach is tested in the servomotors of a real bio-inspired soft neck and results are compared with those obtained from other control strategies. The experiments show that the controller tuned by this method works as expected from a robust controller and that this approach is very competitive compared to other state of the art methods, while offering a more simplified and direct tuning process.

Keywords: fractional order controller, tuning methods, robust control, soft robotics

2010 MSC: 26A33, 44A10

1. Introduction

Fractional calculus is an old issue. It was first established together with integer calculus, and afterwards rediscovered in the last decades in order to get a better mathematical description of the environment. Then, it has been

found useful in many fields, from Economics [1] to Physics [2], and therefore, Engineering. Modeling [3],[4],[5] and system control [6], specially robust control, as in [7] and [8], are the main applications of fractional calculus in Engineering.

The key advantage of using fractional calculus in robust control applications is the wide range of operation that a fractional order system may present. For instance, the phase of a fractional derivative may range from 0 deg to 90 deg while the exponent ranges from 0 to 1, whereas the phase of the first order integer derivative has a fixed value of 90 deg. This feature enables the design of controllers that shape the system in a way that Hendrick Bode described as ideal [9], achieving robustness around the specified design frequency (see [6],[10],[11],[12]). For a comparative of fractional order controllers see [13].

Oustaloup introduced fractional-order algorithms in control engineering with the CRONE method [14]. Later, the form of Fractional Proportional Integral Derivative controller ($PI^{\lambda}D^{\mu}$) was formulated by Podlubny in [15] and further developed in works such as [10],[12],[16]. This form is usually preferred and has received most attention in the last decades, probably due to its simple control law and strong similarities with the ubiquitous PID controller, letting classic design tools to be adapted from integer to fractional exponents. In our present research, the particular cases of PD^{μ} and PI^{λ} have been considered, while the complete $PI^{\lambda}D^{\mu}$ controller will be considered in future research.

Although some works are based on root locus analysis in order to tune a fractional controller (see [17], [18]), most current tuning methods are based on frequency analysis, probably due to their straightforward generalization from integer calculus. According to [19], existing frequency-based tuning methods for fractional controllers can be classified in three groups:

- Analytical: Based on the solution of the constraint equations
- Numerical: Based on the optimization of selected fitness functions
- Tuning rules: Formulas to roughly meet control specifications

The analytical approach, as in [10] and later works such as [6], [11], [12],

involves solving a set of nonlinear equations for controller tuning. Nonlinear equations need a numerical solution, but the problem formulation is still analytical. Numerical approaches used in fractional order controller tuning are based on different optimization methods, such as Particle Swarm Optimization (PSO) [20], Artificial Bee Colony (ABC) [21], Firefly Algorithm (FA), Cuckoo Search (CS), or Differential Evolution (DE), as in [22], [23], [24], [25], [26], respectively. Recent developments have explored the use of iterative methods for parameter tuning, as in [27], but flat phase specification is still solved through optimization. See [28] for a comparative study of fractional order controllers tuned by using optimization algorithms. Finally, tuning rules can also be used on fractional controllers **(see [29])**, but they only show rough results, and their application is restricted to a limited number of plant types. See [19] for a description of current fractional tuning methods.

A common drawback of classic tuning methods found in the literature is their computational complexity, which is mainly solved through numerical or optimization methods, like in [30], where the CS algorithm was used for tuning the controller gains of a 6-degrees of freedom underactuated system. Although the precision of these methods may be very accurate, the insight received from their application is generally very poor, as the only information about the solution is the solution itself. Narrowing down the possible solutions, as described in [31], is useful, however the numerical solution of nonlinear equations may remain tricky.

In our work, a new tuning method for fractional order controllers has been described. Unlike other tuning methods shown in the literature, this approach is graphical in nature and allows fractional controller tuning in a very intuitive and simple way, as a designer can observe the effects of parameter variations graphically. Furthermore, the computational effort is minimized regarding the state of the art methods. This approach is described step-by-step and compared to the results from the implementation of Monje's method [10] and with a PID designed by using the method described in [32].

Finally, robustness of the system is tested in a real robotic platform for each

proposed tuning method. The test bench is a soft robotic neck described in [33], consisting of a three wire-driven parallel mechanism of an elastic link inspired by human neck. Due to actuator geometry and link elasticity, the system gain will be different at every position; therefore, the robustness of each motor, treated as a separate control system, will impact on the overall performance, making the system a great platform for robust control testing.

The next sections of this paper are arranged as follows: A description of the experimental platform used as case study is presented in Section 2. Control specifications and tuning methods for comparison are discussed in Section 3. The proposed tuning method is described in Section 4. Section 5 compares and discusses the test bench results obtained by different control strategies, and finally, Section 6 outlines our main conclusions.

2. Case study

The case study consists on the servomotor position control of the soft robotic neck prototype shown in Fig. 1 [33]. This design is based on a central soft link which acts as a neck and a three Cable-driven Parallel Mechanism (CDPM) used to bend the central link, enabling the upper platform to reach any inclination and orientation in the range of interest. The main components of the neck are: 1. Base; 2. Moving platform; 3. Soft link; 4. Tendons; and 5. Motors. Those components were 3D printed and then assembled to build the neck prototype, including the elastic link in the center. This link weighs 67 g and has been designed to support a weight of 600 g, which means a payload ratio of 895.5%.

For the case study we have focused on the motors driving the CDPM tendons. Platform target positions are defined by their inclination and orientation. Lengths for each cable are obtained by solving the inverse kinematics as described in [33]. Due to the transmission mechanism, each cable length corresponds to a specific motor angular position; therefore, motor position control is required to achieve and hold the platform target positions.

The platform has three actuators located in the base, each one consisting of

four elements:

- Driver: Technosoft iPOS4808 MX-CAN; 400 W, 12-50 Volt, 8 Amp (intelligent motor driver).
- Motor: Maxon RE 35; graphite brushes, 42 Volt, 90 Watt.
- Gear: Maxon planetary gearhead GP32A (3.7 : 1).
- **Encoder: Cui AMT20; 12-bit absolute position (4096 positions).**

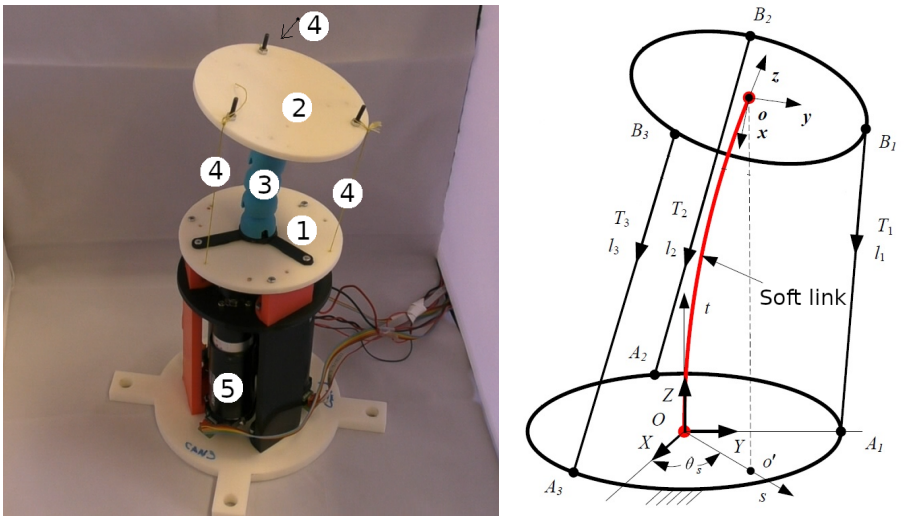

Figure 1: Robotic soft neck prototype (left) and kinematics model (right). *1. Base; 2. Moving platform; 3. Soft link; 4. Tendons; 5. Motors.* See [33] for details on kinematics.

The proposed control scheme for each motor will be a position control with a velocity command; therefore, the system model input is the velocity target and the model output is the measured position.

As the encoder parameters are available, a velocity model cascaded with an integrator will be used. A frequency response curve fitting (see [34]) has been used to identify each motor velocity response. The resulting system model is as

follows:

$$P(s) = \frac{2048.6337 + 54.893316s}{2048.7922 + 67.066887s + s^2}, \tag{1}$$

where s is the Laplace transform complex frequency variable.

Frequency and time responses are shown in Fig. 2 for the velocity model.

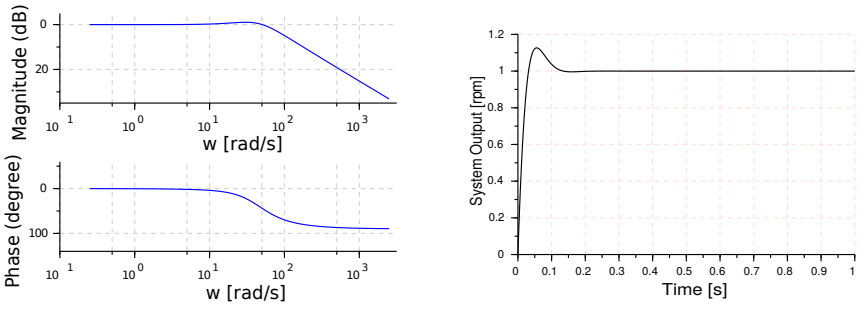

Figure 2: Bode diagram (left) and unit step response (right) for the individual motor velocity model.

This model can be simplified by the same fitting approach with a lower order model. As one of the benefits of robust control is to deal with uncertainty, our controller is expected to show a good performance even using a simpler system model.

The simplified velocity transfer function obtained for each actuator after the fitting procedure is as follows:

$$P(s) = \frac{54.89}{54.89 + s}. \tag{2}$$

Simplified velocity model frequency and time responses are shown in Fig. 3

According to their low Vinnicombe metrics [24] (d=0.157), the complete and simplified models show similar dynamics (see [35] for more details). This simplified model will be used for controller tuning, and then robustness will be tested in the real platform.

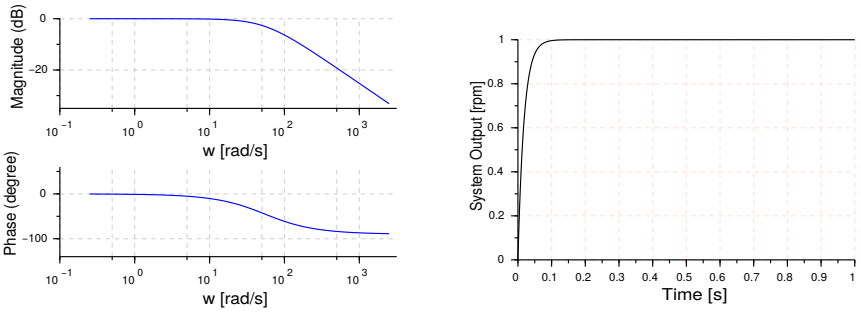

Figure 3: Bode diagram (left) and unit step response (right) for the simplified individual motor system model.

Final open loop system model cascaded with the integrator, with velocity input and position output, results in the following expression:

$$P(s) = \frac{54.89}{54.89 + s} \cdot \frac{k_{enc}}{s}, \tag{3}$$

where $k_{enc} = 6$ is a known gain introduced by the encoder.

3. Current robust control tuning approaches

We are now describing two tuning methods for a later comparison to our new approach. Firstly, the method proposed in [32] has been used for a PID controller tuning, and shows that integer controllers have certain limitations. Then, Monje's method [10] has been implemented, fulfilling the specifications with a fractional order PI controller (fPI).

In related works such as [10] and [32], usual control specifications are crossover frequency (ω_{cg}) and phase margin (ϕ_m). The first is related to system responsiveness (rise time) and the second to system stability (overshoot). Equations (4) and (5) formulate these specifications, respectively:

$$|C(j\omega_{cg})G(j\omega_{cg})|_{dB} = 0 \text{ dB}, \tag{4}$$

$$\arg\left(C(j\omega_{cg})G(j\omega_{cg})\right) = -\pi + \phi_m, \tag{5}$$

where $C(j\omega_{cg})$ is the controller sinusoidal response at ω_{cg}, $G(j\omega_{cg})$ is the plant sinusoidal response at ω_{cg}, and ϕ_m is the desired open loop phase margin for the controlled system.

Another specification is usually considered in robust controllers. First established by Hendrik Bode, and later in works such as [6], [7], [10], [11], flat phase specification in open loop is a usual robustness constraint in controller design. Its formulation is shown in the following equation:

$$\left(\frac{d(\arg\left(C(j\omega_{cg})G(j\omega_{cg})\right))}{d\omega}\right)_{\omega=\omega_{cg}} = 0. \tag{6}$$

The flat phase specification in Eq. (6) ensures that plant gain changes will not affect the system phase around a frequency of interest. As different gains have similar phases, the system response will show a constant overshoot despite gain variations. This property is known as iso-damping.

Once the gain crossover frequency ω_{cg} and phase margin ϕ_m are selected, a set of three nonlinear equations (Eqs. (4), (5) and (6)) with three unknowns (controller's parameters) must be solved in order to fit the controller's parameters. With three constraints, the design is not possible using controllers such as PI or PD, and using a PID controller as in [32] has important limitations, as it is shown in the results section.

In order to ensure a fair comparison of the performance results, all the tuning methods presented in this paper are based on the fulfillment of these three design specifications, and all controllers are PID-based and have the same number of parameters: three.

For the integer order PID and Monje's controller, this set of three nonlinear equations and three unknowns is solved numerically using Matlab, which makes the tuning process not intuitive and very dependent on the initial conditions and local minima. For the new iso-m method presented in this paper, this set of equations and three unknowns is solved graphically, as explained in Section 4, reducing the computational effort and providing performance curves where the effects of controller's parameters variations on the design specifications can

be directly observed.

Since the system to be controlled is the bio-inspired soft neck introduced in Section 2, the specification has been chosen to get a performance close to a real human neck. In clinical studies such as [36], normal neck motion speed may be about 150 deg/s, and taking into account the artificial neck mechanics, a crossover frequency of $\omega_{cg} = 12\,\text{rad/s}$ will grant a similar speed with a good reference tracking.

Choosing a phase margin of $\phi_m = 60\,\text{deg}$ some overshoot is granted, helping to notice the system robustness. Further discussion on this specification is included in the results section.

3.1. Robust PID

Using a PID controller to achieve Eq. (6) is very limited because of its integer nature, which shapes the phase graph with asymptotes at -90 deg and 90 deg and slope of 1 rad/decade (57.3 deg/decade). Therefore, as controller and plant phase slopes must add zero at the desired frequency, as will be discussed in section 3, results are restricted to a number of conditions for the plant and specifications. For instance, small plant phase slopes will force the controllers to have phases close to -90, 0 or 90 deg. Then, possible specifications are reduced to those cases where the equation solution is possible. Therefore, using a PID controller leaves some performances out of scope, not because of the plant, but because of the controller.

For instance, for our case study, with $\omega_{cg} = 12\,\text{rad/s}$, $\phi_m = 60\,\text{deg}$ and the system model in Eq. (3), fulfilling all three specifications at once is not possible. Numerical solution is only possible with a relaxed constraint (6), by allowing the slope to be close to but not equal to zero. The controller parameters obtained with this method are $k_p = 1.95$, $k_i = 7.46$, $k_d = 0$. The controller transfer function is shown in Eq. (7).

$$C(s) = 1.95 + \frac{7.46}{s}. \tag{7}$$

Open loop frequency response (Bode diagram) and closed loop time response of the PID controlled system are shown in Fig. 4. The Bode diagram shows how crossover frequency and phase margin specifications are fulfilled, and how the phase slope is (almost) zero at the crossover frequency vicinity. The time response graph shows the overshoot for different system gain multiples (MxG) and fractions (G/D). It can be observed that these gain changes deviate the system from a zero slope phase curve at the new crossover frequencies for the PID case, and then iso-damping property is not completely achieved, showing a slight overshoot difference between the responses.

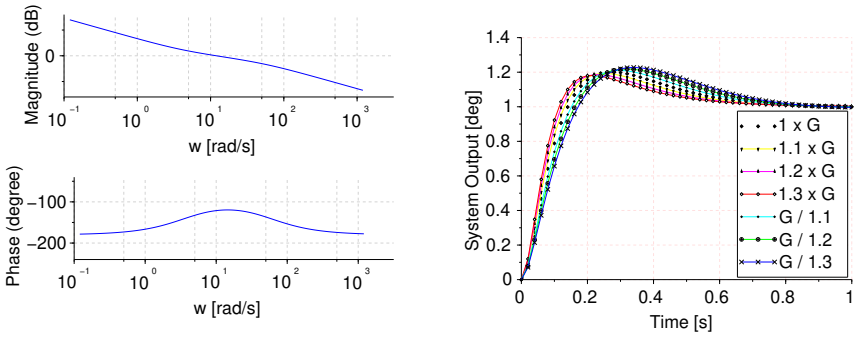

Figure 4: Bode diagram (left) and step response (right) for the system with PID controller. $\phi_m = 60$ deg, $\omega_{cg} = 12$ rad/s.

For the specifications given in this paper, the resulting PID has a derivative gain equal to zero (kd=0); therefore, the PID controller is actually a PI controller in this particular case. However, for the sake of generality, the term PID will be preserved throughout the paper, since the controller could act as a complete PID for other design specifications, having all its gains different from zero.

3.2. Robust fractional order PI by Monje's method

Fractional order control is a better approach to satisfy Eq. (6). If derivative and integral exponents are allowed to be non-integer, additional degrees of freedom can be used, and therefore more specifications may be met. Then, a fractional order PD (fPD), as in Eq. (8), or a fractional order PI (fPI), as in

Eq. (9), can be used to grant all three control specifications:

$$C(s) = k_p + k_d s^{\mu}, \tag{8}$$

$$C(s) = k_p + \frac{k_i}{s^{\lambda}}. \tag{9}$$

By solving for the parameters of an fPD (k_p, k_d, μ) or an fPI (k_p, k_i, λ) in Eqs. (4), (5) and (6), a controller meeting the three specifications is found.

Using Monje's method with $\omega_{cg} = 12\,\text{rad/s}$, $\phi_m = 60\,\text{deg}$ and the system model in Eq. (3), the following fPI controller parameters are obtained: $k_p = 1.7499168$, $k_i = 4.791173$, $\lambda = 0.8$, which results in the following controller:

$$C(s) = 1.7499168 + 4.791173 s^{-0.8} \tag{10}$$

Figure 5 shows the Bode diagram of the open loop system with this controller and its feedback step response. The specifications are clearly fulfilled. It can be checked that phase slope is flat at the crossover frequency and time response overshoot remains constant for different system gain multiples (n x G) and fractions (G / n).

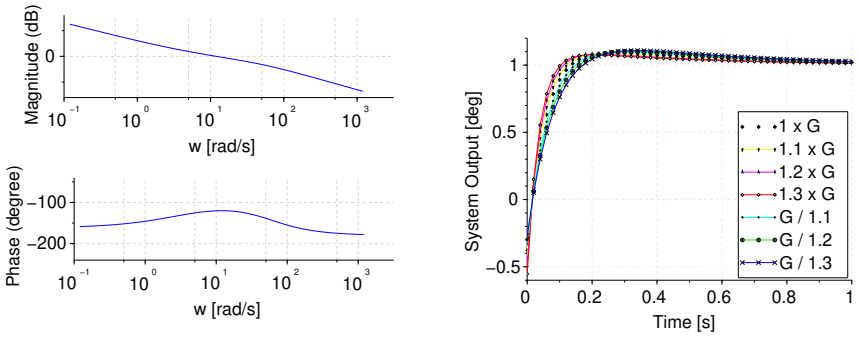

Figure 5: Bode diagram (left) and step response (right) of the system with fPI based on Monje's method. $\phi_m = 60\,\text{deg}$, $\omega_{cg} = 12\,\text{rad/s}$.

4. Graphical tuning method based on iso-slope phase curves

In this section, a graphical tuning method based on the system phase slope cancellation is proposed. This method shows very competitive results compared to the state of the art methods, and its simplicity makes it very suitable to implement with limited computation resources.

The method is based on the phase additive property. The frequency response phase of two systems connected in series is equivalent to the addition of their frequency response phases, and of course, the same applies for slopes. Therefore, known the phase slope of a system at some frequency, another system with opposite slope at that frequency could cancel the slopes to zero. For example, first order systems have a varying negative phase slope in the range of $0\,\mathrm{deg}/\log_{10}(\omega)$ to $57.3\,\mathrm{deg}/\log_{10}(\omega)$. A controller with the correct positive slope at the right frequency will cancel the system slope, resulting in a flat slope around that frequency.

The controllers in Eqs. (8) and (9) may present any phase slope in the range $(0, \infty)$ depending on the fractional exponent for values in the range $(0, 2)$. Therefore, phase cancellation is always possible for any plant with these controllers. As both controllers can be formulated by one equation just letting the fractional exponent to be both, positive or negative, in the following, only the fPD case will be considered (positive exponent), and the fPI case can be formulated just considering a negative exponent. For simplicity, the controller equation is reformulated as:

$$C(s) = k_p + k_a s^{\alpha}, \tag{11}$$

where α is the derivative exponent, $\alpha \in \mathbb{R}$ in the interval $(-2, 2)$.

The controller phase is only affected by the exponent value (α) of s in (11), and takes values between $0\,\mathrm{deg}$ and $90\alpha\,\mathrm{deg}$ as can be seen in a phase plot. Phase slope (m) starts at zero, grows with the frequency to a maximum value (m_{max}) that depends on the α exponent, and then decreases to zero again.

It will be proven that there is always at least one exponent having the same

(but opposite) slope as the system for the specified frequency, which allows the phase cancellation and grants a flat slope around that frequency.

Expressing the controller in Eq. (11) as a function of the frequency implies $s^\alpha = (jw)^\alpha = j^\alpha w^\alpha$. Then, the controller formulation is now:

$$C(\omega,\alpha) = k_p + k_a\omega^\alpha \cdot j^\alpha = k\left(1 + \tau_a\omega^\alpha \cdot j^\alpha\right) = k\left(1 + \tau_a\omega^\alpha \cdot e^{(j\pi/2)\alpha}\right), \tag{12}$$

$$\bar{C} = 1 + \tau_a\omega^\alpha \cdot (\cos(\alpha\pi/2) + j\sin(\alpha\pi/2)) = \frac{C(\omega,\alpha)}{k},$$

where the proportional gain $k_p = k$ and the derivative gain $k_a = k\tau_a$. Note here that k_a can act as a derivative or integral gain, depending on the sign of α (positive sign implies a derivative action and negative sign implies an integral action).

According to Eq. (12), it is clear that the phase depends on α exponent, τ_a, and ω, but not on k, as a scalar multiplication. For clarity, in the following, $\bar{C}$ will be considered for the phase calculation instead of C. This will only affect the resulting crossover frequency (ω_{cg}), that will be later adjusted using k.

The phase shift of $\bar{C}$ as a function of frequency is given by:

$$\phi_c(\omega) = arctan\left(\frac{\tau_a\omega^\alpha\sin(\alpha\pi/2)}{1 + \tau_a\omega^\alpha\cos(\alpha\pi/2)}\right) = \arctan\left(\frac{\sin(\alpha\pi/2)}{\frac{1}{\tau_a\omega^\alpha} + \cos(\alpha\pi/2)}\right). \tag{13}$$

There are three singular cases in Eq. (13). First, for $\omega = 0$, the phase shift is $\phi(0) = 0\,\text{rad}$. Second, when $\omega = \tau_a^{-\alpha}$, the phase reaches its maximum slope, as shown later in this section through Eq. (18). Third, when $w \to \infty$, after the maximum slope, the term $\frac{1}{\tau_a\omega^\alpha}$ tends to zero, and the phase tends to $\alpha\pi/2\,\text{rad}$. The following listing shows a summary.

- $\omega = 0$; $(\frac{1}{\tau_a\omega^\alpha} = \infty) \to \phi = 0\,\text{rad}$
- $\omega = \tau_a^{-\alpha}$; $(\frac{1}{\tau_a\omega^\alpha} = 1) \to \phi = \alpha\pi/4\,\text{rad}$
- $\omega = \infty$; $(\frac{1}{\tau_a\omega^\alpha} = 0) \to \phi = \alpha\pi/2\,\text{rad}$

Bode phase plots use logarithmic scale for ω variable in abscissas axis, resulting in an axis showing $\log_{10}(w)$ marks. Slope calculations are simplified through changes of variables such as $x = \log_{10}(\omega)$ and later, $\tau_x = (\tau_a 10^{x\alpha})^{-1}$. Variable changes resulting in the new controller phase formula are shown at the following equation.

$$\phi_c(x) = \arctan\left(\frac{\sin(\alpha\pi/2)}{\frac{1}{\tau_a 10^{x\alpha}} + \cos(\alpha\pi/2)}\right) = \arctan\left(\frac{\sin(\alpha\pi/2)}{\tau_x + \cos(\alpha\pi/2)}\right). \quad (14)$$

The slope at any frequency point is the derivative of the phase equations with respect to x, as follows:

$$m = \frac{d\phi_c(\omega)}{d\log_{10}(w)} = \frac{d\phi(x)}{dx} = \frac{d}{dx}\arctan\left(\frac{\sin(\alpha\pi/2)}{\tau_x + \cos(\alpha\pi/2)}\right). \quad (15)$$

After some calculations, the following slope equation yields:

$$m = \frac{\log(10)\alpha\sin(\alpha\pi/2)}{\left(\left(\frac{\sin^2(\alpha\pi/2)}{\tau_x+\cos(\alpha\pi/2)}\right)^2 + 1\right)(\tau_x\cos(\alpha\pi/2))^2} = \frac{\log(10)\alpha\sin(\alpha\pi/2)}{\tau_x + \frac{1}{\tau_x} + 2\cos(\alpha\pi/2)}. \quad (16)$$

Finally, $\tau_x + \frac{1}{\tau_x}$ is replaced by σ_x, resulting the slope formula:

$$m = \frac{\log(10)\alpha\sin(\alpha\pi/2)}{\sigma_x + 2\cos(\alpha\pi/2)} \quad , \quad \sigma_x = (\tau_a 10^{x\alpha})^{-1} + \tau_a 10^{x\alpha}. \quad (17)$$

At this point, the slope (m) is a function of three variables (x, α, τ_a), two of them (α, τ_a) being frequency independent parameters. Then, the slope depends only on x trough σ_x along the phase plot. As the slope is divided by σ_x, the maximum slope (m_{max}) occurs when σ_x reaches its minimum value. Now it will be proved that the frequency for this maximum slope is $\omega = \tau_a^{-\alpha}$.

Making σ_x derivative equal to zero results in the following equation:

$$\frac{d\sigma_x}{dx} = \frac{d}{dx}\left(\frac{1}{\tau_a 10^{x\alpha}} + \tau_a 10^{x\alpha}\right) = \log(10)\tau_a\alpha 10^{x\alpha} - \frac{\log(10)\alpha}{\tau_a 10^{x\alpha}} = 0, \quad (18)$$

and solving for x results in:

$$\tau_a 10^{x\alpha} = 1 \rightarrow x = -\log_{10}(\tau_a^{\alpha}). \tag{19}$$

Checking the function vicinity of this unique singular point ($x = -\log_{10}(\tau_a^{\alpha})$) will tell if the extreme is maximum or minimum:

$$\begin{aligned} \sigma_x(-\log_{10}(\tau_a^{\alpha}) \cdot \frac{1}{2}) &= \frac{1}{2} + 2 > 1, \\ \sigma_x(-\log_{10}(\tau_a^{\alpha}) \cdot 2) &= 2 + \frac{1}{2} > 1. \end{aligned} \tag{20}$$

Therefore, as the singular point is unique and its neighbours are bigger at both sides, that point is a minimum for σ_x, and the slope reaches its maximum at this frequency. Using (19) in (17), the maximum slope formula is found:

$$m_{max} = \frac{\log(10)\alpha \sin(\alpha\pi/2)}{2 + 2\cos(\alpha\pi/2)} = \frac{\log(10)}{2} \cdot \alpha \tan(\alpha\pi/4). \tag{21}$$

Also, using Eq. (19) in Eq. (13) gives a phase of $\phi(\tau_a^{-\alpha}) = \alpha\pi/4$. This means that when the slope is maximum, the phase is at its middle run, and both occur at point $x = -\log_{10}(\tau_a^{\alpha})$.

Although Eq. (21) cannot be solved for x, it is possible to plot a graph like in Fig. 6 relating m_{max} to α, and thus α can be found from any target slope. This curve was made for a list of α values in the range $(0, 2)$ using Eq. (21). Another possibility is to fit the non solvable function with another solvable function and then isolate α. Figure 6 also shows two examples based on the fit function $m = a\tan(b\alpha)^2$, which allows to calculate the exponent as $\alpha = b^{-1}\arctan(\sqrt{m/a})$.

Based on the previous result, it is possible to set any controller phase slope by choosing the correct exponent for the s term. This will cancel any system phase slope to achieve the flat phase robust constraint. Once the required slope is known, this α exponent can be obtained graphically by using the curve in Fig. 6.

The drawback of our method at this point is that the phase margin cannot be specified, and it will depend on the system frequency response at the chosen

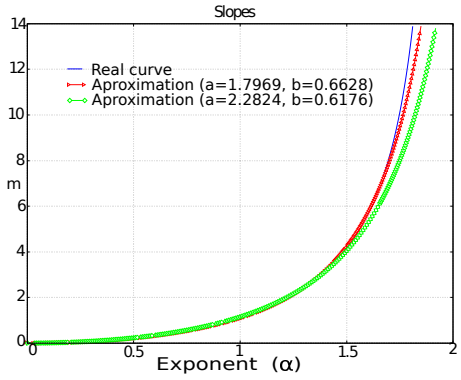

Figure 6: Maximum slope plot showing α in front of m based on Eq. (21) and two fits based on equation $m = a\tan(b\alpha)^2$: (a=1.7969, b=0.6628), more accurate for $\alpha \geq 1$ and (a=2.2824, b=0.6176), more accurate for $\alpha \leq 1$.

crossover frequency. It may be suitable when robustness specification is preferred over stability margin, but this may be granted by an integer order PD. Fortunately, this drawback can be solved using the variable previously defined as σ_x.

Cancellation has been done using the maximum slope, but there are many other slopes available in the controller phase curve. As the slope shifts from zero to a maximum, and then to zero again, a controller with a greater m_{max} than the system slope will have at least two frequencies at which the phase slope is exactly the one needed for cancellation. In fact, there must be one in the range $m = 0$ to $m = m_{max}$, before the maximum slope, and another in the range $m = m_{max}$ to $m = 0$, after the maximum slope.

Taking the tangent of Eq. (14) yields:

$$\tan(\phi_c(x)) = \frac{\sin(\alpha\pi/2)}{\frac{1}{\tau_a 10^{x\alpha}} + \cos(\alpha\pi/2)} = \frac{\sin(\alpha\pi/2)}{\tau_x + \cos(\alpha\pi/2)}, \tag{22}$$

where $\tau_x = (\tau_a 10^{x\alpha})^{-1}$. Isolating τ_x in Eq. (22), results in:

$$\tau_x = \frac{\sin(\alpha\pi/2)}{\tan(\phi_c(x))} - \cos(\alpha\pi/2), \tag{23}$$

and using τ_x to solve σ_x:

$$\sigma_x = \frac{\sin(\alpha\pi/2)}{\tan(\phi_c(x))} - \cos(\alpha\pi/2) + \frac{1}{\frac{\sin(\alpha\pi/2)}{\tan(\phi_c(x))} - \cos(\alpha\pi/2)}. \tag{24}$$

Now, replacing σ_x in the slope Eq. (17) and simplifying as shown in Eq. (25), m as a function of $\phi_c(x)$ and α is obtained:

$$m = \log(10)\alpha \frac{1 - \frac{\tan(\phi_c(x))}{\tan(\alpha\pi/2)}}{\tan(\phi_c(x)) + \frac{1}{\tan(\phi_c(x))}} = \frac{\log(10)}{\sin(\phi_c(x))}\alpha\left(1 - \frac{\tan(\phi_c(x))}{\tan(\alpha\pi/2)}\right). \tag{25}$$

Equations (24) and (25) are not dependent anymore on x directly, but on the phase angle, which depends on x. As the slope calculation variable is ϕ_c (whatever it depends on), it is possible to set the desired controller phase (ϕ_c) and solve for α. As a remainder, the phase in Eq. (25) is only the controller phase. In order to find a solution for the phase margin specification in Eq. (5), the system phase must be known.

Again, it is not possible to isolate the variable of interest, therefore one of the alternatives stated before can be used to find a solution. Figure 7 shows the resulting iso-slope (iso-m) curves relating the controller phase ϕ_c [deg], the fractional order α and the slope m [deg/decade], both for positive and negative controller phase contributions (according to control specifications).

The curves in Fig. 7 were obtained through Eq. (25)**. This equation can also be used to make a lookup table that can be easily programmed in a Programmable Logic Controller (PLC). Any precision can be reached by interpolation or increasing the table granularity in order for the function approximation to be as precise as required. Controller phase ϕ_c and phase slope m_c are table inputs, and α will be the table output. The authors published a previous paper [37] in which the implementation of this type of PLC function is addressed. Another option is to use a fit function as shown in Fig. 6 in order to be able to isolate α.**

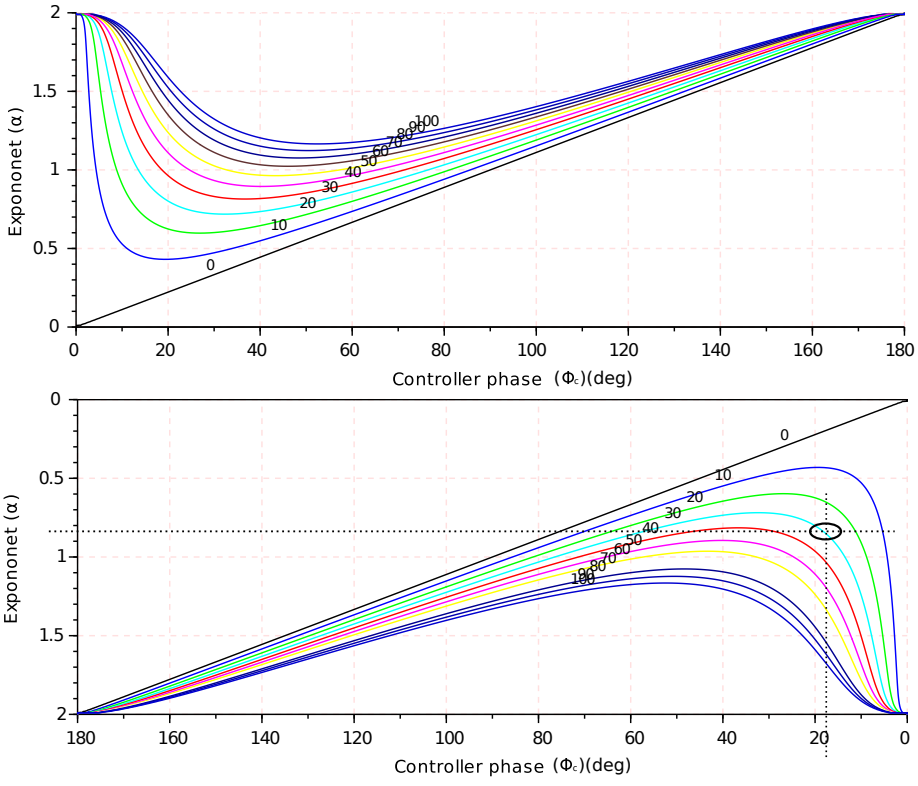

Figure 7: $Iso-m$ method. Fractional order α exponent in ordinates and controller phase ϕ_c [deg] in abscissas. Iso-m curves in [deg/decade].

For the controller to correctly cancel the system phase slope, their frequencies have to match: the phase cancellation must occur at the same crossover frequency. Knowing the α exponent and given the crossover frequency, τ_x can be calculated from Eq. (23) as follows:

$$\frac{1}{\tau_x} = \tau_a 10^{x\alpha} = \tau_a \omega^{\alpha} \rightarrow \tau_a = \frac{1}{\tau_x \omega^{\alpha}}. \tag{26}$$

Using this equation, τ_a can be found so that the slope matches ω_{cg}, and this enables to adjust the frequency at which the cancellation is planed. In the special case when $\tau_x = 1$ (m_{max}), Eq. (26) simplifies to $\tau_a = 1/\omega^{\alpha}$, and the exponent can be obtained from Fig. 6 or Eq. (21).

Finally, the crossover frequency must be adjusted by setting the controller

gain. Using Eq. (4), k is found as follows:

$$k = \frac{1}{|\bar{C}(j\omega_{cg})G(j\omega_{cg})|}. \tag{27}$$

Once the three parameters are found, our controller is tuned, fulfilling the three initial specifications from Eqs. (4), (5) and (6).

For fPI case generalization, we should consider that the controller phase starts from 0 rad in the fPD case, but from $-\alpha\pi/2$ rad in the fPI case; then the controller phase in Fig. 7 will be subtractive instead of additive in the region of negative exponents, as shown at the controller phase axis for the fPI curve.

4.1. Step by step iso-m controller design

The steps for fractional order fPD controller design according to the proposed iso-m method are summarized below. Two values must be selected by the designer:

- Desired crossover frequency ω_{cg}
- Desired phase margin ϕ_m

After selecting those values, the summary of the method is as follows:

Step 1: Obtain system phase and phase slope at the selected frequency

Based on a system model or its frequency response information, phase and phase slope can be obtained. When these values are found in Bode plots, the phase and phase slope units are m_s [deg/log(ω)] and ϕ_s [deg].

Step 2: Get α exponent from slope m_c and phase margin ϕ_m

Knowing the system phase ϕ_s and the desired phase margin ϕ_m, the required controller phase is calculated from equation $\phi_c = (-\phi_s + \phi_m - 180)$ deg. Then, the required controller phase slope is obtained as $m_c = -m_s$ deg/log(ω). Finally, using both values (controller phase and phase slope), α exponent is obtained **either graphically from Fig. 7 or numerically using a lookup table or an approximation function as stated before.**

Setp 3: Compute τ_a from α and ω

From Eq. (23), τ_x is calculated, and then, using Eq. (26) gives the solution for τ_a, so the calculated controller has the correct slope at the specified frequency. At this point, the partial solution is done. The controller $\bar{C}(s)$ is complete as all the parameters in Eq. (12) are known.

Step 4: Compute k according to ω_{cg} specification

With the correct phase, the controller still has to meet the specification in Eq. (4). Using this equation, k can be computed as in Eq. (27) to force the open loop system gain to be 0 dB at the crossover frequency ω_{cg}.

Step 5: Controller solution

Once all parameters are known, the product of k by controller $\bar{C}$ from step 3 is the final solution:

$$C(s) = k(1 + \tau_a s^{\alpha}) = k_p + k_a s^{\alpha}, \tag{28}$$

where α derives from Fig. 7, τ_a derives from Eq. (26), k derives from Eq. (27) and $k_a = k\tau_a$.

4.2. Example of iso-m tuning method.

In this example, the case study system will be tuned, using the step by step method shown at the previous section. Control specifications described in section 3 are:

- Desired crossover frequency $\omega_{cg} = 12\,\text{rad/s}$
- Desired phase margin $\phi_m = 60\,\text{deg}$

Besides both specifications, this method will ensure inherently the robustness property based on the resulting flat phase of the open loop system around the crossover frequency (Eq. (6)), which enhances the system robustness to gain variations.

The steps shown at the previous section will be followed in this example.

Step 1: The Eq. (3) system frequency response is shown in Fig. 8. Using this Bode diagram, the system phase and phase slope found at ω_{cg} are $\phi_s = -102.4\,\text{deg}$ $m_s = -30\,\text{deg/log}(\omega)$ respectively. Actually, the transfer function is not needed if the system frequency response is available.

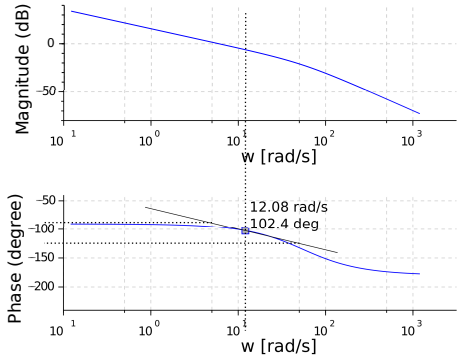

Figure 8: System phase and phase slope at the selected crossover frequency $\omega_{cg} = 12\,\text{rad/s}$ in the system Bode diagram.

Step 2: Opposite to system phase slope, the controller phase slope is $m = 30\,\text{deg/log}(\omega)$, and the required controller phase to achieve the phase margin specification at ω_{cg} is $\phi_c = (-(-102.4) + 60 - 180)\,\text{deg}$, $\phi_c = -17.6\,\text{deg}$. Based on that, the exponent $\alpha = -0.81$ results from Fig. 7.

Step 3: According to those values, τ_a is calculated using Eq. (26), resulting $\tau_a = 2.72$.

Step 4: Finally, k can be computed using Eq. (27), resulting $k = 1.76$, and forcing the open loop system gain to be 0 dB at $w_{cg} = 12\,\text{rad/s}$.

Step 5: Therefore, controller parameters are : $k_p = 1.76$, $k_a = 4.7872$, $\alpha = -0.81$, with the following transfer function:

$$C(s) = 1.76(1 + 2.72s^{-0.81}) = 1.76 + 4.7872s^{-0.81}. \tag{29}$$

Figure 9 shows the frequency (open loop) and time (feedback) responses of the system with this controller. Specifications are fulfilled as shown in the Bode diagram. Closed loop step responses for different system gain multiples (n x G)

and fractions (n / D) are also drawn in this figure, showing constant overshoot (iso-damping property).

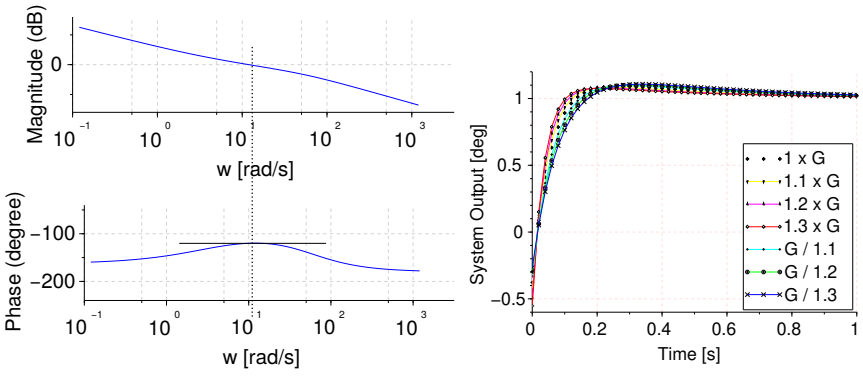

Figure 9: Open loop Bode diagram and step feedback response of the system with the controller designed using the iso-m tuning method. Case: $\omega_{cg} = 12\,\text{rad/s}$, $\phi_m = 60\,\text{deg}$, gain range $(1.3G,\ \frac{1}{1.3}G)$, where G is the system default gain ($G = 6$).

5. Experimental results

The testing platform used for experimentation is the soft robotic neck described in Section 2, a soft link actuated by a cable-driven mechanism operated by three DC motors.

Setting the neck target in 90 deg of orientation and 15 deg of inclination and solving inverse kinematics, results in a neck motor configuration of $-40\,\text{deg}$, $80\,\text{deg}$, $-40\,\text{deg}$ for motors M1, M2 and M3, respectively. Setting the motor targets accordingly, the system moves from the original position (rest) to the desired position. Sensor (encoder) data and control signals were measured for all motors during the experiments. Although all of them were actuated for the test, since the results are very similar for all three motors, only the performance of the first motor is discussed.

5.1. Robustness results

In order to compare their performances, step responses under different load conditions are presented in Fig. 10, which shows one graph for each tuning

method. The control signals for these step responses are shown in Fig. 11.

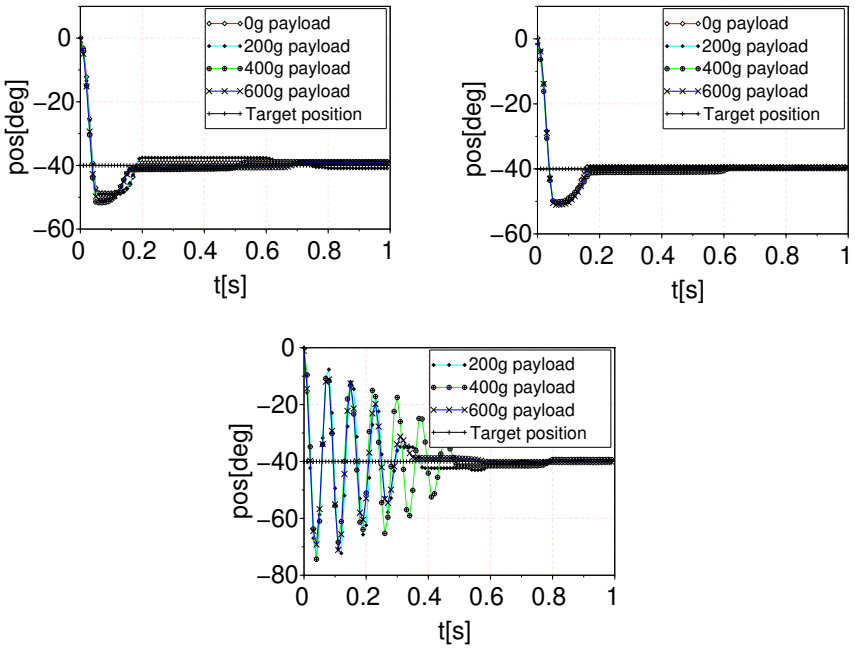

Figure 10: Comparison of the step responses of all the controllers for different masses at the tip. Step input of -40 deg. Iso-m(top left), Monje's (top right) and PID (bottom) methods.

The four different payload configurations have similar performances and overshoot is almost invariant for $iso-m$ and Monje's methods despite mass differences, starting from 0 g and reaching 600 g; therefore, robustness is achieved with both tuning methods for the payload configurations proposed reaching also a zero steady state error.

At this point, it is important to highlight that while Monje's method is based on numerical solution of nonlinear equations, iso-m is a graphical method, which minimizes the complexity and computation efforts while keeping results very competitive.

As shown in Section 3.1, the integer PID controller is stable for the ideal simplified motor model and for the set of payloads considered. However, its behavior, even if it is robust and overshoot keeps almost constant for different

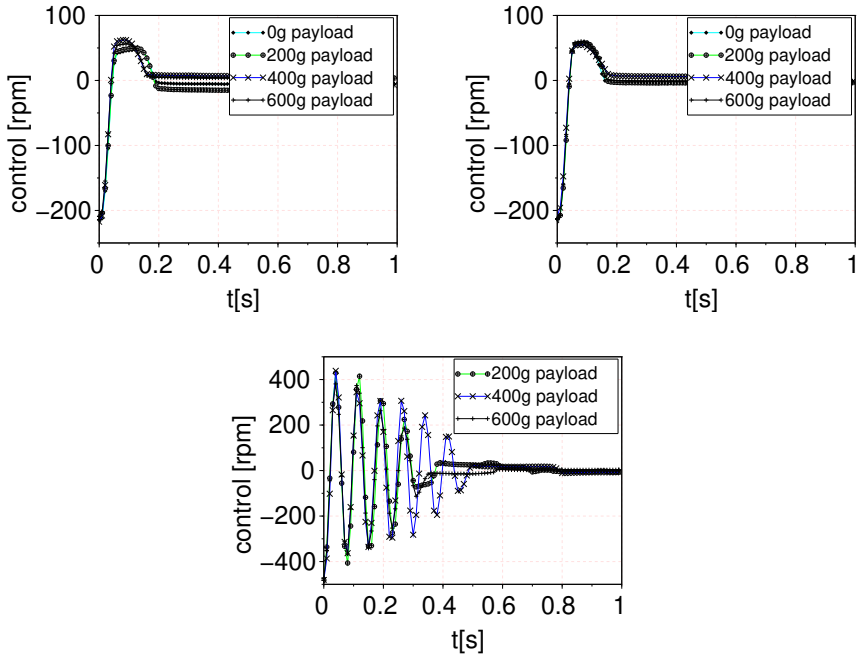

Figure 11: Control signals of all the controllers for different masses at the tip. Step input of −40 deg. Iso-m(top left), Monje's (top right) and PID (bottom) methods.

payloads, is very oscillatory, as expected from the presence of an integer order integral controller.

With the fractional order controller, the integral action is alleviated thanks to its fractional order, allowing the system to reach a zero steady state error but with a much less oscillatory response.

During experiments, this oscillatory behavior of the integer PID controller is also shown. However, it has to be taken into account that, after finishing the experiment and when going back to the initial position, the oscillation amplitude can reach a value that triggers the driver protections and switches the system into an error state. Even if the system would reach a stable behavior, oscillations cause an undesirable performance in the real platform.

5.2. Performance results

For a deeper comparison of the performance results among the three tuning methods proposed, Table 1 shows detailed performance data gathered from Fig. 10. The worst values are shown within all payload configurations.

Method	Maximum Overshoot (%)	Maximum Rise time (s)	Maximum Settling time (s)	Overshoot Variation (ΔO %)
Monje's	25	0.05	0.15	1
$iso-m$	30	0.05	0.19	5
PID	90	0.04	0.6	1

Table 1: Worst performance data, including overshoot variation (ΔO) obtained as the difference between highest and lowest overshoots.

In order to show the system performance for a specific configuration, the step responses and control laws for 400 g payload are shown in Fig. 12.

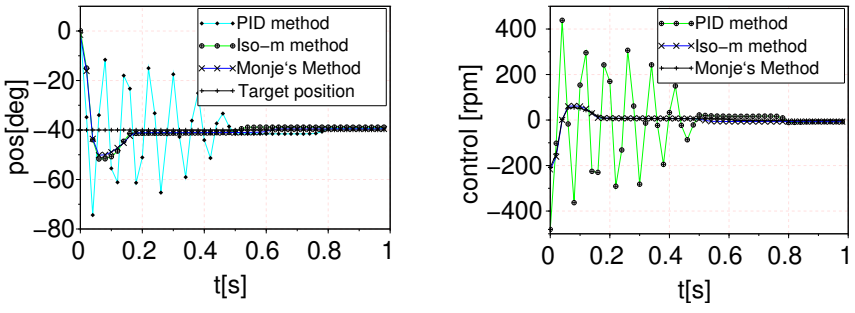

Figure 12: Step responses (left) and control signals (right) of the system with the three controllers proposed and for a 400 g payload.

It is noticeable that performances for both $iso-m$ and Monje's methods are very similar, while PID results are not so good, with very high settling times (0.6 s for the worst case), mainly due to the oscillations produced by the controller pure integrator. The steady state error is zero for all control strategies. Table 2 shows detailed performance data for each approach based

Method	Overshoot [%]	Rise time [s]	Settling time [s]
Monje's	25	0.05	0.15
$iso-m$	30	0.05	0.19
PID	90	0.04	0.35

Table 2: Performance data for 400 g payload configuration.

on data gathered from Fig. 12.

It is important to remark that both Monje's and iso-m methods provide very similar results (as expected, since the same control specifications are aimed for the sake of comparison). In fact, the difference between both controllers' parameters is minor, as shown in Eq. (10) **and Eq.** (29)**, respectively. The slight differences observed during the performances of both controllers are not due to a superiority of Monje's method with respect to iso-m method, but to implementation issues. It has to be taken into account that once the controller parameters are obtained, the controllers are approximated to a transfer function using Grünwald–Letnikoff discrete approximation as described in [19], and the final performance slightly changes according to the resulting transfer function.**

In view of the experimental results, it is important to remark the following with respect to measurement noise, parameter variation and external disturbances:

Regarding measurement noise, as can be seen from the experimental results in Fig. 10, Fig. 11 and Fig. 12 of step responses and control signals, the controllers are performing correctly and measurement noise (encoder signals) is not affecting the system.

Regarding parameter variation, in our system the gain of the model is mainly affected by the load of the device (masses at the top of the neck). Besides, a change in the tendons location can also result in a different gain value. The tuning methods proposed in this work specifically deal with system gain changes

as the main parameter of interest, as explained throughout the paper. That is why an experiment with different payloads has been run for the neck: to check the real robustness of the system to these physical gain changes the system is suffering. The results show that the performance of the fractional order controller is very robust to payload (gain) changes, showing an iso-damping property.

Regarding external disturbances, some experiments have been run where the neck is hitted during its movement. Even if a good tracking performance has been observed, a test is still to be designed that allows us to measure the behavior in detail and that guarantees that the disturbances are repeatable and distributed along the complete neck. These results will be included and discussed in future works.

As a summary, robust behavior is granted from the application of both fractional controllers, obtaining much better results with the $iso{-}m$ and Monje's methods in robustness and performance than with the PID controller, and thus validating the proposed new approach. A video of the system performance with iso-m controller and PID controller can be seen at `https://youtu.be/cnQG5WYkg2Y`, which shows the soft neck controlled with both fractional and integer controllers.

6. Conclusions

A new tuning method for fractional order controllers (fPD, fPI) has been proposed in this paper. The system used for testing the resulting controllers, a bio-inspired neck made of soft material, performs according to the specifications and shows high robustness to mass changes, even when the tip payloads are very big in comparison with the neck weight (895.5% heavier).

Differently from other tuning methods found in the literature, our approach has a graphical nature and allows controller tuning in a very intuitive and simple way, and lets the designer observe the effects of parameter variations graphically. This contribution is very important from our point of view, since it allows an

easy and fast re-tuning of the controller just using the curves provided without the need to run the algorithm over and over again.

For applications that require controller's parameters to be changed during operation in order to enhance the response towards a particular direction, this method provides a fast and robust solution, since the performance curves can be used to directly select the controller's parameters values that achieve the new control requirements.

Besides, the iso-m method aims to offer both good performance and low computational cost. Computational complexity for a lookup table is similar to a memory access, which is the case of the iso-m method. Although the exact results are never achieved, any precision can be reached by interpolation or increasing the table granularity.

Computational complexity for nonlinear equation solvers (Monje's method) is different, as it depends on multiple factors like iteration number, precision, and initial conditions (difficult to measure, but definitely much higher). For instance, Monje's method is solved numerically and the tuning process is very dependent on the initial conditions and local minima.

Therefore, implementing Monje's method using a PLC would not be possible from a computational point of view. This problem is totally avoided using iso-m method. As can be seen through the algorithm-based description of the method in Section 4.1, the iso-m approach implementation in a PLC could be more easily addressed, which would benefit the industrial application of the method.

To finish, nonlinear equation solvers can only be used in offline control schemes, due to the heavy computational effort, which makes impossible to use them in adaptive control strategies; in contrast, the iso-m method can be implemented in adaptive control strategies.

Future works will focus on extending the method to tune fractional order PID controllers.

Acknowledgments

Research leading to these results has received funding from HUMASOFT project, with reference DPI2016-75330-P, funded by the Spanish Ministry of Economy and Competitiveness, and from RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by "Programas de Actividades I+D en la Comunidad de Madrid" and cofunded by Structural Funds of the EU.

References

[1] Z. Wang, X. Huang, H. Shen, Control of an uncertain fractional order economic system via adaptive sliding mode, Neurocomputing 83 (2012) 83 – 88. doi:10.1016/j.neucom.2011.11.018.

[2] R. K. Saxena, G. Pagnini, Exact solutions of triple-order time-fractional differential equations for anomalous relaxation and diffusion I: The accelerating case, Physica A: Statistical Mechanics and its Applications 390 (4) (2011) 602 – 613. doi:10.1016/j.physa.2010.10.012.

[3] L. Dorcak, Numerical models for the simulation of the fractional-order control systems, arXiv Mathematics e-prints (2002) math/0204108.

[4] R. Cipin, C. Ondrusek, R. Huzlík, Fractional-order model of DC motor, in: T. Březina, R. Jabloński (Eds.), Mechatronics 2013, Springer International Publishing, Cham, 2014, pp. 363–370.

[5] W. Zheng, Y. Luo, Y. Chen, Y. Pi, Fractional-order modeling of permanent magnet synchronous motor speed servo system, Journal of vibration and control 22 (9) (2016) 2255–2280.

[6] I. Petras, Fractional order feedback control of a DC motor, Journal of Electrical Engineering 60 (3) (2009) 117–128.

[7] V. Feliu, B. M. Vinagre, C. A. Monje, Fractional-order control of a flexible manipulator, in: J. Sabatier, O. P. Agrawal, J. A. T. Machado (Eds.), Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering, Vol. 1, Springer Netherlands, Dordrecht, 2007, pp. 449–462.

[8] M. J. Grimble, M. A. Johnson, C. Monje, Y. Chen, B. Vinagre, D. Xue, V. Feliu, Fractional-order Proportional Derivative Controller Tuning for Motion Systems, Springer London, London, 2010, pp. 107–120. doi:10.1007/978-1-84996-335-0_6.

[9] H. W. Bode, Network analysis and feedback amplifier design, Bell Telephone Laboratory series, Van Nostrand, New York, NY, 1945.
URL http://cds.cern.ch/record/105206

[10] C. A. Monje, B. M. Vinagre, V. Feliu, Y. Chen, Tuning and auto-tuning of fractional order controllers for industry applications, Control Engineering Practice 16 (7) (2008) 798–812.

[11] T. Qingshun, W. Chunfu, Y. Yuanhui, L. Guodong, Z. Fengyu, Design and implementation of fractional order controller for service robots, International Journal of Control and Automation 8 (5) (2015) 209–220.

[12] K. Ranjbaran, M. Tabatabaei, Fractional order [pi], [pd] and [pi][pd] controller design using Bode's integrals, International Journal of Dynamics and Control 6 (1) (2018) 200–212. doi:10.1007/s40435-016-0301-7.

[13] D. Xue, Y. Chen, A comparative introduction of four fractional order controllers, in: Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No.02EX527), Vol. 4, 2002, pp. 3228–3235 vol.4. doi:10.1109/WCICA.2002.1020131.

[14] A. Oustaloup, La dérivation non entière théorie, synthèse et applications (1995) 507Bibliographie à la fin des chapitres.
URL http://infoscience.epfl.ch/record/25108

[15] I. Podlubny, Fractional-order systems and PI$^{\lambda}$D$^{\mu}$-controllers, IEEE Transactions on Automatic Control 44 (1) (1999) 208–214. doi:10.1109/9.739144.

[16] R. S. Barbosa, J. A. T. Machado, I. M. Ferreira, Tuning of PID controllers based on Bode's ideal transfer function, Nonlinear Dynamics 38 (1) (2004) 305–321. doi:10.1007/s11071-004-3763-7.

[17] F. Merrikh-bayat, M. Afshar, Extending the root-locus method to fractional-order systems, Journal of Applied Mathematics 2008 (2008) 12. doi:10.1155/2008/528934.

[18] I. Petras, Stability of Fractional-Order Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 55–101. doi:10.1007/978-3-642-18101-6_4.

[19] D. Valerio, J. S. da Costa, An Introduction to Fractional Control, Control, Robotics and Sensors, Institution of Engineering and Technology, 2012. URL http://digital-library.theiet.org/content/books/ce/pbce091e

[20] J. Kennedy, R. Eberhart, Particle swarm optimization, in: In Proceedings of IEEE International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

[21] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm, Journal of global optimization 39 (3) (2007) 459–471.

[22] K. Khandani, A. A. Jalali, Robust fractional order control of a DC motor based on particle swarm optimization, in: MEMS, NANO and Smart Systems, Vol. 403 of Advanced Materials Research, Trans Tech Publications, 2012, pp. 5030–5037. doi:10.4028/www.scientific.net/AMR.403-408.5030.

[23] A. Rajasekhar, P. Kunathi, A. Abraham, M. Pant, Fractional order speed control of DC motor using Levy Mutated Artificial Bee Colony Algorithm, in: 2011 World Congress on Information and Communication Technologies, 2011, pp. 7–13. doi:10.1109/WICT.2011.6141192.

[24] V. H. Haji, C. A. Monje, Fractional order Fuzzy-PID control of a combined cycle power plant using Particle Swarm Optimization algorithm with an improved dynamic parameters selection, Applied Soft Computing 58 (2017) 256–264. doi:10.1016/j.asoc.2017.04.033.

[25] M. Itik, E. Sahin, M. S. Ayas, Fractional order control of conducting polymer artificial muscles, Expert Systems with Applications 42 (21) (2015) 8212 – 8220. doi:https://doi.org/10.1016/j.eswa.2015.06.033.

[26] F. Martín, C. A. Monje, L. Moreno, C. Balaguer, DE-based tuning of $PI^{\lambda}D^{\mu}$ controllers, ISA Transactions 59 (2015) 398 – 407. doi:10.1016/j.isatra.2015.10.002.

[27] C. I. Muresan, E. H. Dulf, R. Both, Vector-based tuning and experimental validation of fractional-order PI/PD controllers, Nonlinear Dynamics 84 (1) (2016) 179–188. doi:10.1007/s11071-015-2328-2.

[28] S. Al-Ratrout, A. Saleem, H. Soliman, Optimization methods in fractional order control of electric drives: A comparative study, in: 2015 10th International Symposium on Mechatronics and its Applications (ISMA), 2015, pp. 1–6. doi:10.1109/ISMA.2015.7373460.

[29] F. Padula, A. Visioli, Tuning rules for optimal PID and fractional-order PID controllers, Journal of Process Control 21 (1) (2011) 69 – 81. doi:10.1016/j.jprocont.2010.10.006.

[30] R. Miranda-Colorado, L. T. Aguilar, J. E. Herrero-Brito, Reduction of power consumption on quadrotor vehicles via trajectory design and a controller-gains tuning stage, Aerospace Science and Technology 78 (2018) 280 – 296. doi:https://doi.org/10.1016/j.ast.2018.04.027.

[31] V. Badri, M. S. Tavazoei, On tuning fractional order [proportional–derivative] controllers for a class of fractional order systems, Automatica 49 (7) (2013) 2297 – 2301. doi:10.1016/j.automatica.2013.04.026.

[32] Y. Chen, K. L. Moore, Relay feedback tuning of robust PID controllers with iso-damping property, Trans. Sys. Man Cyber. Part B 35 (1) (2005) 23–31. doi:10.1109/TSMCB.2004.837950.

[33] L. Nagua, J. Muñoz, C. A. Monje, C. Balaguer, A first approach to a proposal of a soft robotic link acting as a neck, in: Actas de las Jornadas de Automática, Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura, 2018, pp. 522–529.

[34] E. C. Levy, Complex-curve fitting, Automatic Control, IRE Transactions on AC-4 (1959) 37–43. doi:10.1109/TAC.1959.6429401.

[35] G. Vinnicombe, Frequency domain uncertainty and the graph topology, IEEE Transactions on Automatic Control 38 (9) (1993) 1371–1383. doi:10.1109/9.237648.

[36] D. Z. De Beyl, P. Salvia, Neck movement speed in cervical dystonia, Movement Disorders 24 (15) (2009) 2267–2271. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/mds.22830, doi:10.1002/mds.22830.

[37] C. A. Monje, B. M. Vinagre, G. E. Santamaría, I. Tejado, Auto-tuning of fractional order $PI^{\lambda}D^{\mu}$ controllers using a PLC, in: 2009 IEEE Conference on Emerging Technologies Factory Automation, 2009, pp. 1–7. doi:10.1109/ETFA.2009.5347104.