uc3m | Universidad Carlos III de Madrid

e-Archivo

This is a postprint version of the following published document:

Muñoz, J., et al. A robust control method for the elbow of the humanoid robot TEO based on a fractional order controller, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) [Towards a robotic society], October 1-5, 2018, Madrid, Spain. IEEE Press, 2019, Pp. 6378-6383*
DOI: https://doi.org/10.1109/IROS.2018.8593732

© 2018, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

A novel robust method for the elbow of the humanoid robot TEO based on a fractional order PD controller

Jorge Muñoz[1], Concepción A. Monje[1]*, Fernando Martín[1] and Carlos Balaguer[1]

***Abstract*— This paper presents a novel method for the control of the elbow joint of the humanoid robot TEO, based on a fractional order PD controller. Due to the graphical nature of the proposed method, a few basic operations are enough to tune the controller, offering very competitive results compared to classic methods. The experiments show a robust performance of the system to mass changes at the tip of the humanoid right arm.**

I. INTRODUCTION

Fractional Calculus (FC) has been used with success in quite different fields, from economics [1] to physics [2], and of course, engineering, with applications in modeling and system control [3][4].

The fractional order controllers receiving the most attention in the last decades are the Fractional Proportional Integral Derivative ($PI^{\lambda}D^{\mu}$) controllers, formulated for the first time by Podlubny [5] and studied in works such as [6], [7], or [8].

A remarkable number of articles, specially when the subject involves motion control, focus on the derivative control, leaving the integrator out. This control scheme has the advantage of using the position sensor as an integrator, which simplifies the controller, while the steady state error is still cancelled, making the integral part unnecessary, even undesired. The first to use this kind of fractional controller was Dorcak in [9], and later in [10] or [11], with application to the control of the joints of a robot. Similar approaches have been proposed in [12], [13] and [14], the last approach applied to a legged robot.

There are different tuning approaches, based on different techniques. Many of them are based on the numeric solution of nonlinear equation systems [6][8][10][11]. Other approaches based on optimization methods can be found in the literature, such as Particle Swarm Optimization algorithm (PSO) [7][13][15], Artificial Bee Colony algorithm (ABC) [16][17], Firefly Algorithm (FA) [18], or Differential Evolution method (DE) [19]. A comparative study for optimization algorithms applied to fractional controllers can be found in [20].

The purpose of this work is to propose a strategy for the control of the elbow joint of the humanoid robot TEO (Fig. 1), a robot built by the Robotics Lab team of Carlos III University of Madrid. For this purpose, a novel method for the tuning of a Fractional order PD (FPD) controller is presented in this paper. Differently from other tuning methods in the literature based on numeric solutions of nonlinear equation systems or optimization, our proposal has a graphical nature, and allows the tuning of the controller in a very intuitive way through very simple and straightforward steps that do not require computational efforts.

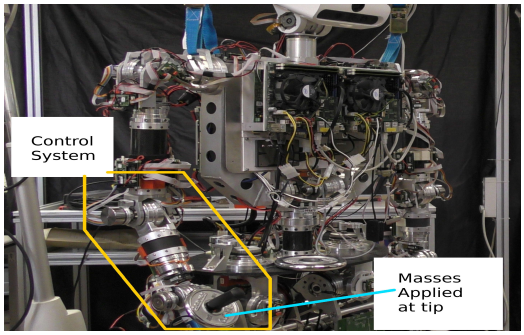

Fig. 1. Humanoid robot TEO. Control system (yellow) and the masses applied (blue).

In order to validate our proposal, it will be compared with the results from the application of Monje's method [6] and ABC optimization algorithm [16]. The robustness of the system to mass changes at the tip of the arm will be pursued.

The paper is organized as follows. The system will be described in Section II, and the control problem will be stated in Section III. Section IV presents the design of the FPD controller based on methods already known from the literature. Section V introduces a new proposal for the tuning of the FPD, based on a graphical analysis of the control problem. Section VI discusses the experimental results obtained from the application of the different control strategies to the real robot. Finally, Section VII outlines the main conclusions of the paper.

II. SYSTEM DESCRIPTION

TEO is a full-size humanoid robot (see Fig. 1) developed by the Robotics Lab team of Carlos III University of Madrid [21]. It is an improved version of its predecessor, RH1, and in a similar way, TEO has six joints on each limb, making a total of 24 joints at the four limbs. Adding other two joints for trunk pitch and yaw, and other two for head pitch and yaw, they sum a total of 28 joints.

In this paper, the system considered will be the forearm (see Fig. 1), and the other joints are kept static, therefore

*The research leading to these results has received funding from the HUMASOFT project, with reference DPI2016-75330-P, funded by the Spanish Ministry of Economy and Competitiveness.

[1] Jorge Muñoz, Concepción A. Monje, Fernando Martín and Carlos Balaguer are with Robotics Lab of the Carlos III University of Madrid, Avda de la Universidad 30, 28911 Leganés, Madrid, Spain. jmyanezb@ing.uc3m.es

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

our system can be modeled as a solid of diameter $8\,[\mathrm{cm}]$ and length $20\,[\mathrm{cm}]$, rotating around the joint axis.

Every joint is composed by a driver-motor-gear system:

- Driver: Technosoft iPOS4808 MX-CAN; 400 W, 12-50 Volt, 8 Amp (intelligent motor driver).
- Motor: Maxon EC 45; flat 42.8 mm, brushless, 70 Watt.
- Gear: Harmonic Drive™ (of different ratios).

Since the driver iPOS4808 has a local low level control loop, the joints have a trapezoidal velocity profile, as shown in Fig. 2. This feature makes the motion smooth, but also adds nonlinearity to the system. This nonlinearity makes the system identification challenging, and also the control will be harder when compared to a linear system.

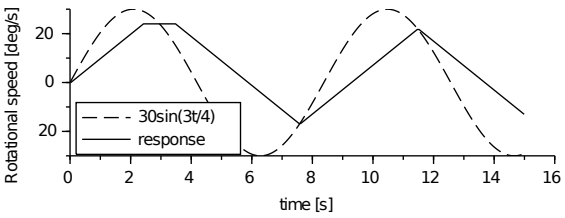

Fig. 2. Velocity system response to a sinusoidal input, showing constant acceleration and saturation at $24\,[\mathrm{deg/s}]$. Although the frequency is the same for input and output, their waveforms are different.

In order to design the fractional order controller, a plant model is needed. As the block diagram of the motor is known (see Fig. 3), the first step will be to simplify this block diagram to a linear one, comparing later the outputs and checking if the linear model is accurate enough.

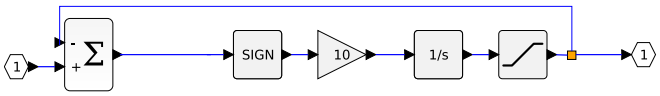

Fig. 3. Block diagram for the system. Left to right: Input, Difference, Sign, Gain, Integrator, Saturation, Output.

Choosing the linearization equilibrium point at zero velocity allows us to remove the saturation block around this point. Sign function keeps the sign of the input while converts the output to a gain of one, so the aim of this block is to keep acceleration constant. If this block is removed, the resulting system will have a very similar unitary step response, as can be seen in Fig. 4 (Right). Then, removing nonlinear Sign and Saturation blocks from Fig. 3, a unitary feedback system with a gain and integrator results. The closed loop system is a first order system with the following transfer function: $G(s) = 1/(0.1s + 1)$.

Now, in order to compare the models, a unitary step is fed into both first order blocks using the simulation scilab tool (xcos), resulting the plot shown in Fig. 4 (Right). Due to the similarity of the curves, and considering that the frequency is kept in the output for both models (Fig. 2), it is possible to use the simplified model for the controller design. Though there is a difference between both systems, it will be negligible for this task. In addition, the kind of controller proposed is known to deal fine with plant uncertainties, so the difference will be even lesser, as will be shown in the experimental results section.

Therefore, the motor will be modeled as a first order system, having velocity target input and angular joint velocity output, with a gain of 1, because the motor is commanded directly from the intelligent motor driver. If the encoder is considered, the system output will be the position, while the input will still be velocity. In this case, the control variable will be the position, and an integrator must be added to the transfer function. The final system will be the one in (1). See the Bode diagram for the system in Fig. 4 (Left). Note that phase margin $\phi_m = 84.3\,[\mathrm{deg}]$ and crossover frequency $\omega_c = 0.995\,[\mathrm{rad/s}]$.

$$G(s) = \frac{1}{(0.1s + 1)s} \tag{1}$$

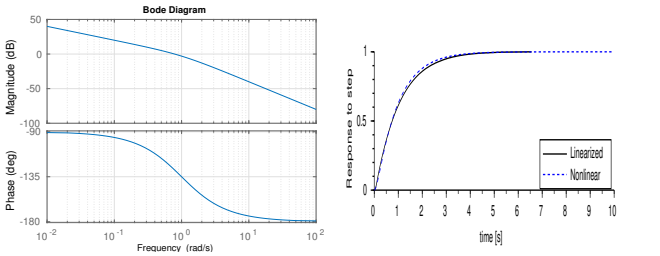

Fig. 4. Bode diagram for the system model with transfer function in (1) (left). Response to a unitary step in $[\mathrm{deg/s}]$ showing the difference between linearized and nonlinear models (rigth).

III. CONTROL PROBLEM

The elbow joint is commanded in velocity, while the aim for the robot is to reach and stand any desired position; therefore, position control with velocity command is the best option, as stated before. Having an integrator in the plant, the best option is to avoid another integrator in the controller, what leads to a proportional derivative approach.

In addition, different payloads will be attached to the tip (at the hand of the robot) and the responses will be studied in order to analyze the robustness of the system to mass changes. Besides, in order to include gravity effects, the input reference will be a negative step moving the arm down. This way we can study the performance of the system in the presence of model uncertainties.

According to all this, the controller proposed will be an FPD controller as:

$$C(s) = k_p + \tau_d s^{\mu} \tag{2}$$

where k_p is the proportional gain, τ_d is the derivative gain, and μ is the fractional order of the derivative. The control specifications to be met can be summarized as:

- Robustness to mass changes on the tip at a nominal crossover frequency ω_c
- Robustness to model uncertainties

The way to meet the first specification is by getting the phase of the open loop system to be flat at ω_c so as the system

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

will be robust to gain changes, as demonstrated in [22]. In this system, the gain of the model is mainly affected by the payloads of the device, as proved in [4]. Therefore, meeting this specification will make the system robust to changes in the load.

For our case study, the selected crossover frequency will be $\omega_c = 1\,[\mathrm{rad/s}]$. This frequency is very close to the elbow joint nominal frequency, and the phase margin of the system at that frequency is around $\Phi_m = 84.3\,[\mathrm{deg}]$, which is high enough to avoid oscillations in the temporal response. Since the FPD has a derivative component, it will favor the increment of the phase margin, improving the performance of the system in the time domain.

IV. FPD TUNING: CLASSIC METHODS

This section will describe the results obtained when different classic methods are used, particularly Monje's method [6] and ABC optimization algorithm [17].

A. Monje's Method

Having gain changes robustness as specification, the first method revisited will be the one described in [6]. The following conditions will be used for the solution:

1) Phase margin Φ_m at crossover frequency ω_c:
   - $|C(j\omega_c)G(j\omega_c)|_{dB} = 0\,[\mathrm{dB}]$
   - $\arg\left(C(j\omega_c)G(j\omega_c)\right) = -\pi + \Phi_m$
2) Plant gain changes robustness:
   - $\left(\frac{d(\arg F(s)}{d\omega}\right)_{\omega=\omega_c} = 0$, F(s) the open loop system.

Having three constraints allows solving the system for three variables: k_p, k_d, μ of the FPD in equation (2). A crossover frequency must be defined, which will be the mentioned $\omega_c = 1\,[\mathrm{rad/s}]$. The fsolve function, based on [23] from Matlab, has been used to find the solution for the above constraints, resulting the following controller parameters: $k_p = 0.463$, $k_d = 0.621$, $\mu = 0.495$. The Bode diagram of the open loop system with this controller and the step response are shown in Fig. 5, where it is clearly seen that the specifications are fulfilled. It can be checked that the phase slope is flat at the selected frequency. $\Phi_m = 110\,[\mathrm{deg}]$, $\omega_c = 1\,[\mathrm{rad/s}]$

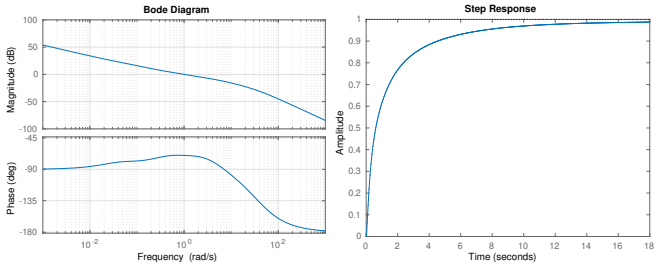

Fig. 5. Bode diagram and step response for FPD based on Monje's method.

B. Artificial Bee Colony (ABC) Algorithm

As discussed above, here we use the ABC algorithm in [16], based on the search for the best solution by discarding the ones that will not fit the specifications. The parameters used were: onlookers = employed (50), iterations (100), acceleration coefficient upper bound (1). After applying the algorithm, the resulting controller parameters are: k_p = 0.009, k_d = 0.996, $\mu = 0.99$. The results are shown in Fig. 6, with $\Phi_m = 175\,[\mathrm{deg}]$, $\omega_c = 1\,[\mathrm{rad/s}]$.

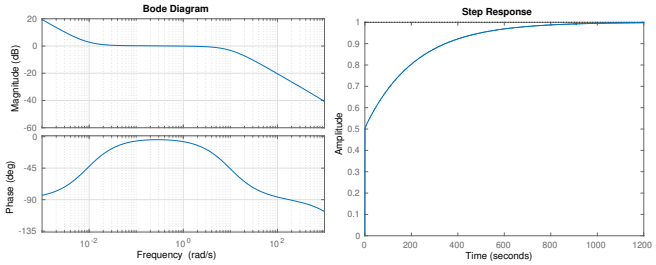

Fig. 6. Bode diagram and step response for FPD controller based on ABC algorithm.

V. FPD TUNING: ISO_ω METHOD

A new method for the tuning of the FPD controller is presented here. As will be seen, the results are very competitive compared to the former methods, and even better in some cases. Besides, the graphical nature of the method makes it applicable in a very direct and easy way, without the need of solving nonlinear equations, as happens in the previous methods. The design of the controller becomes very intuitive, and the designer can tune the controller with a better understanding of the control problem.

This method is based on the phase slope cancellation at a chosen frequency in the Bode diagram. The first and second order systems, such as the system proposed, have a negative slope at some frequency range. Then, as the slope is known at any frequency of the Bode diagram, it is possible to foresee the slope of the phase of the controller as the opposite of that of the system at that frequency point. To achieve a resulting zero slope, it will be enough to cascade the controller and the system, since the resulting phase curve will be the sum of the two previous ones. With this phase cancellation, we are fulfilling the iso-damping property described in section III, to achieve robustness to mass changes on the tip.

On the other hand, the phase slope of the FPD controller proposed in (2) depends only on the fractional exponent of the complex variable s (the derivative order), and takes values between $0\,[\mathrm{deg/log_{10}}(\omega)]$ and $\infty\,[\mathrm{deg/log_{10}}(\omega)]$ when the exponent varies between 0 and 2. This can be seen in Fig. 7, that shows the slopes (in $[\mathrm{deg/log_{10}}(\omega)]$) for every μ in the range $(0, 1.6)$ (more ranges can be plotted). In this way, it is possible to get a zero phase slope in the open loop Bode diagram, given a frequency ω, by finding the controller parameters that result in a slope contrary to the system slope, and the exponent will be $\mu = f(m)$, where m is a known slope.

From the controller expression (2), it is easy to see that, if $s = jw$, then $s^\mu = (jw)^\mu = j^\mu w^\mu$. It is clear that the exponent affects the phase, and also modifies the module. The parameter k_p has no effect on the phase, so it will be dismissed for the calculation of the slope. Then, the equation of the controller as a function of the frequency is

Preprint submitted to 2018 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 1, 2018.

the following:

$$\frac{C(\omega)}{k_p} = 1 + k_d\omega^\mu \cdot j^\mu = 1 + k_d\omega^\mu \cdot e^{(j\pi/2)^\mu} =$$

$$= 1 + k_d\omega^\mu \cdot (\cos(\mu\pi/2) + j\sin(\mu\pi/2)) = \bar{C}(\omega) \quad (3)$$

Therefore, as $\tan(\Phi(\omega)) = \Im(\bar{C}(\omega))/\Re(\bar{C}(\omega))$, the phase for this type of controller will be:

$$\Phi(\omega) = \arctan\left(\frac{\sin(\mu\pi/2)}{\frac{1}{k_d\omega^\mu} + \cos(\mu\pi/2)}\right) \quad (4)$$

Three options can be discussed: $\omega^\mu = 0$, $\omega^\mu = 1/k_d$, and $\omega^\mu = \infty$. The first is obvious: when $\omega^\mu = 0$, the phase is $\Phi = 0$. When $\omega^\mu = 1/k_d$, this happens to be the frequency point at which the phase has the highest rate of change, and coincides with the half of the total phase (see trigonometric identities for half angle); so, $\Phi(\omega) = \mu\pi/4$. Once the highest slope point has been left, and for higher values of the frequency, the term $\frac{1}{k_d\omega^\mu}$ tends to zero, and the phase tends to $\Phi(\omega) = \mu\pi/2$.

Throughout the curve, the rate of change of the phase, or its slope, in the Bode diagram will be:

$$m = \frac{d}{d\log_{10}(\omega)}\arctan\left(\frac{\sin(\mu\pi/2)}{\frac{1}{k_d 10^{\log_{10}(\omega)^\mu}} + \cos(\mu\pi/2)}\right) \quad (5)$$

And simplifying through a variable change of the form $\log_{10}(\omega) = x$, results:

$$m = \frac{d}{dx}\arctan\left(\frac{\sin(\mu\pi/2)}{\frac{1}{k_d 10^{x\mu}} + \cos(\mu\pi/2)}\right) \quad (6)$$

As the actual aim is to find the controller parameters, the variable to look for in (6) is μ. It can be observed that in the previous formula, m_{max} depends on three variables: ω, k_d and μ. That makes a total of four variables. Then, to simplify the problem, the interest will be focused on the point $\omega^\mu = 1/k_d$ (greatest slope), that is, at the point where $\Phi(\omega) = \mu\pi/4$. This makes m in (6) depend only on μ, while keeping the feature (slope) sought.

Simplifying (6) when $\omega^\mu = 1/k_d$ leads to the following equation:

$$m_{max} = \tan(\mu\pi/4)\mu\log(10)/2 \quad (7)$$

Plotting the values of m in front of μ, we can find the corresponding μ for a required slope. For the elaboration of this curve, all values of m are calculated, as the parameter μ is changed, always when the phase equals the half of the total phase ($\Phi(\omega) = \mu\pi/4$). The resulting curve is shown in the Fig. 7.

Starting from the previous result, any desired slope can be cancelled by assigning the appropriate fractional derivative order μ of the regulator. Now, it is easy to see that the fractional order is a trivial parameter, since it can be obtained from Fig. 7 directly.

In order for this negative slope to match the frequency of interest in the system, more parameters are needed. Both

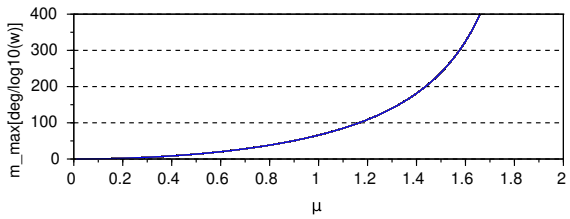

Fig. 7. Phase slope plot in the Bode diagram at the frequency at which $\phi(\omega) = \mu \cdot 45\,[\text{deg}]$, as μ changes.

parameters k_d and μ have an impact on the crossover frequency, and therefore, on the place where the slope is cancelled in the Bode diagram.

We can plot a graph with all the values of ω as a function of μ and k_d. Of course, ω takes all values between $(0, \infty)$, so a value must be chosen to solve the equation. Again, the maximum slope point ($\Phi(\omega) = \mu\pi/4$) will be considered, so that the results keep consistent. Therefore, $\omega = \omega_{max}$ for that plot.

The construction of this surface requires the calculation of as many solutions as points on the surface to be represented. Though it can be a demanding computation, the graphs need to be computed just once, and later, all the necessary values can be obtained by graphical interpolation. The surface of solutions is shown in Fig. 8.

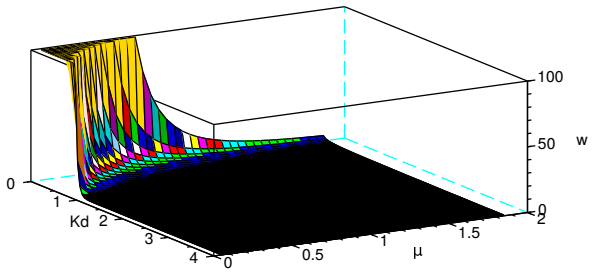

Fig. 8. Surface of controller parameters, showing ω as a funtion of μ and k_d.

Once the surface is computed, an ISO_ω graph can be obtained as shown in Fig. 9, which allows, given a value of ω within the range of values chosen during the graph computing, and a value of μ that cancels the slope, to select a value of k_d that sets the controller frequency in order to perform the cancellation at the correct place (ω_c).

Since k_p does not affect the phase, it is used later to set the magnitude of the open loop system to $0\,[\text{dB}]$ at ω_c.

Therefore, to sum all up, a tuning method for fractional FPD controller is proposed, whose application is summarized in the following steps:

-Step 1: Obtaining the slope at ω_c

The frequency response of the system is needed for this step, which can be obtained through the analysis of its model, or the frequency analysis of the system. Directly from the graph, or by calculations, the slope of the phase curve is obtained for the chosen frequency. Since the curve is logarithmic, the slope obtained will be $m\,[\text{deg}/\log_{10}(\omega)]$.

-Step 2: Find the exponent μ that cancels the slope

The opposite slope is chosen for the controller phase slope:

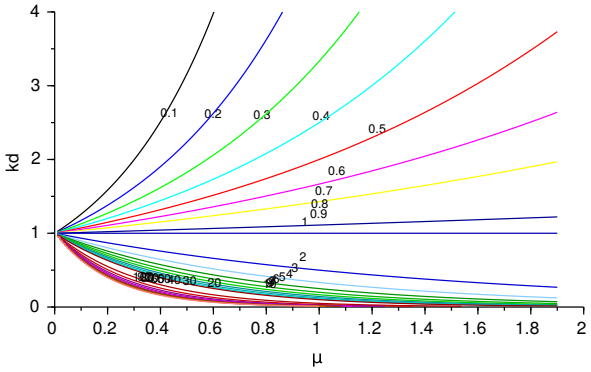

Fig. 9. Parameter tuning plot, showing iso_ω curves as a funtion of μ and k_d.

$m_c = -m\,[\text{deg}/\log_{10}(\omega)]$. The value of the exponent is found using Fig. 7.

-Step 3: Search for k_d

Following the ISO_ω curves numbered in Fig. 9, and having the value of μ, there is only one value for k_d that complies with the specification of slope cancellation.

Once μ and k_d are known, the equation of the controller can be obtained as:

$$\frac{C(s)}{k_p} = 1 + k_d s^\mu \tag{8}$$

-Step 4: Set magnitude to $0\,[\text{dB}]$ at ω_c

Once the phase slope is adjusted, the whole transfer function of the controller is multiplied by a constant k_p, which will cause the magnitude diagram to move without changing the phase. The calculation of this constant will be done so that the magnitude is $0\,[\text{dB}]$ for the desired frequency:

$$k_p = \frac{1}{|\bar{C}(j\omega_c)G(j\omega_c)|} \tag{9}$$

-Step 5: Final parameters

Once all the constants are known, the controller results

$$C(s) = k_p + \tau_d s^\mu \tag{10}$$

where μ comes from Fig. 7, k_d comes from Fig. 9, $k_p = 1/|\bar{C}(j\omega_c)G(j\omega_c)|$, and $\tau_d = k_p k_d$.

Let's apply this method for our case study.

At $\omega_c = 1\,[\text{rad/s}]$, a slope of $-12.94\,[\text{deg}/\log_{10}(\omega)]$ is found from the system Bode diagram (Fig. 4). Then, according to Fig. 7, it is found that $\mu = 0.47$.

Having these values of μ and ω , and using Fig. 9, a value of $k_d = 1$ is found.

To finish, $k_p = 0.541$ is computed, which makes the magnitude to be $0\,[\text{dB}]$ at $\omega_c = 1\,[\text{rad/s}]$. Therefore the resulting controller is:

$$C(s) = 0.541(1 + s^{0.47}) = 0.541 + 0.541 s^{0.47}$$

The resulting Bode diagram is shown in Fig. 10. A null slope is observed at the specified frequency, as well as a crossover frequency according to the requirements. The step response of the controlled system is also shown.

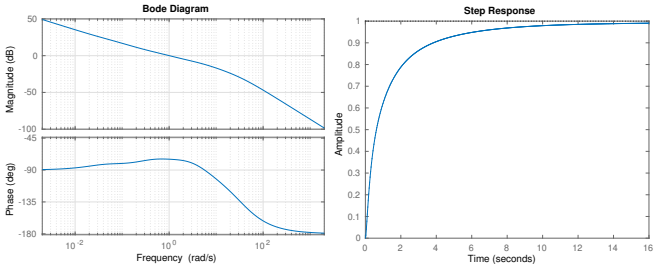

Fig. 10. Bode diagram and step response for FPD controller based on the ISO_ω method.

VI. EXPERIMENTAL RESULTS

Once the different FPD controllers have been tuned, they will be implemented and tested in the real platform, and the results will be discussed in this section.

For the controller implementation, an integer order approximation for s^μ, based on an iterative algorithm described in [24], will be used for each exponent. Then, the transfer function for s^μ will be discretized, using the Tustin method with a sampling period $t_s = 0.01$.

The test setup consists of the robot standing with all joints blocked, except for the fourth joint of right arm, that will receive a $-30\,[\text{deg}]$ step input. The arm will move through a descending trajectory, starting at $60\,[\text{deg}]$, that will be the worst case of uncertainty, as the gravity will act in opposition to the control effort.

Different masses will be attached to the tip to test the robustness of each controller.

A. Comparison

The compared results for all tests are shown in Fig. 11 with no load at the tip. Note that both methods ISO_ω and Monje's have very similar results, the ISO_ω control system presenting a slightly faster response. The ABC method results are not that good, presenting a very high settling time ($800\,[\text{s}]$), mainly due to the resulting very high phase margin. The steady state error is zero for Monje's and ISO_ω control systems. The error for ABC-based control system reaches zero too, but the system is highly slow.

B. Robustness

The robustness is shown for the different controllers in Fig. 12. Observe that the response is invariant for Monje's and ISO_ω despite the different loads at the tip, starting at $500\,[\text{g}]$ and reaching $2000\,[\text{g}]$. Once again the results are worse for the ABC-based control system, since the phase is not that around ω_c.

VII. CONCLUSIONS

In this paper, a novel tuning method for a fractional order PD controller (FPD) has been proposed with applications to the control of the elbow joint of the humanoid robot TEO. This method is based on the graphical solution of the controller parameters, avoiding the resolution of nonlinear equations sets and allowing to solve the control problem in a very intuitive and direct way. The results obtained are

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

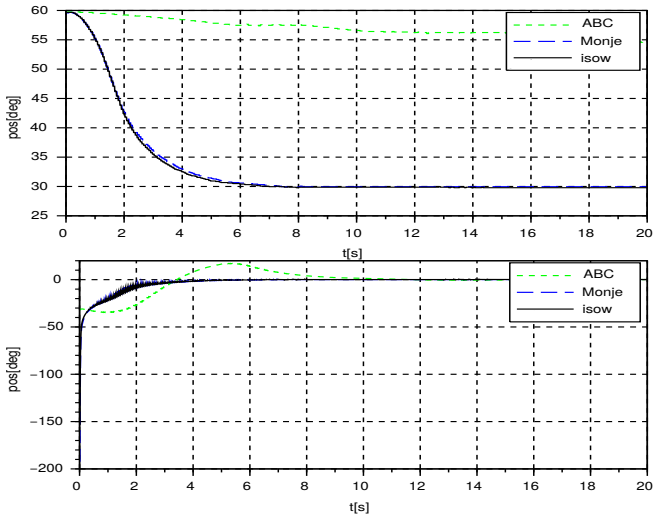

Fig. 11. Comparison of the responses of all the controllers. Step input of -30 [deg].

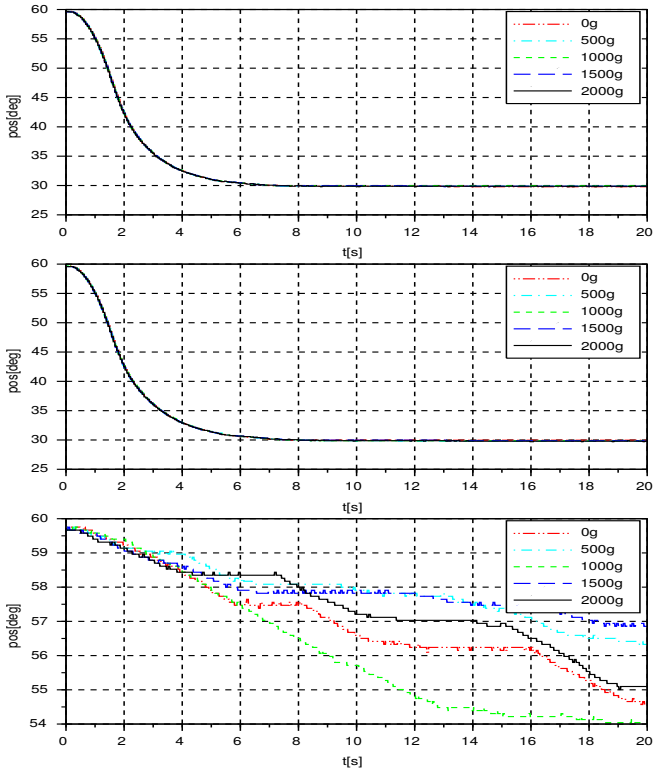

Fig. 12. Comparison of the responses of all the controllers for different masses at the tip. Step input of -30 [deg]. ISO_{ω}(top), Monje's (middle) and ABC (bottom).

very competitive in comparison to other tuning methods well known in the literature. The system performs according to the specifications and shows high robustness to mass changes at the tip (robot's hand).

Future research steps will focus on the application of the method for the tuning of fractional order PID controllers and their application to other joints of the robot.

References

[1] Z. Wang, X. Huang, and H. Shen, "Control of an uncertain fractional order economic system via adaptive sliding mode," *Neurocomputing*, vol. 83, pp. 83 – 88, 2012.

[2] R. K. Saxena and 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*, vol. 390, no. 4, pp. 602 – 613, 2011.

[3] R. Cipin, C. Ondrusek, and R. Huzlík, "Fractional-order model of dc motor," in *Mechatronics 2013*, T. Březina and R. Jabloński, Eds. Cham: Springer International Publishing, 2014, pp. 363–370.

[4] V. Feliu, B. M. Vinagre, and C. A. Monje, *Fractional-order Control of a Flexible Manipulator*. Dordrecht: Springer Netherlands, 2007, pp. 449–462.

[5] I. Podlubny, "Fractional-order systems and pi/sup /spl lambda//d/sup /spl mu//-controllers," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 208–214, Jan 1999.

[6] C. Monje, B. Vinagre, V. Feliu, and Y. Chen, "Tuning and auto-tuning of fractional order controllers for industry applications," *Control Engineering Practice*, vol. 16, no. 7, pp. 798 – 812, 2008.

[7] A. Rastogi and P. Tiwari, "Optimal tuning of fractional order pid controller for dc motor speed control using particle swarm optimization," 2013.

[8] K. Ranjbaran and M. Tabatabaei, "Fractional order [pi], [pd] and [pi][pd] controller design using bode's integrals," *International Journal of Dynamics and Control*, vol. 6, no. 1, pp. 200–212, Mar 2018. [Online]. Available: https://doi.org/10.1007/s40435-016-0301-7

[9] L. Dorcak, "Numerical models for the simulation of the fractional-order control systems," *ArXiv Mathematics e-prints*, Apr. 2002.

[10] I. Petras, "Fractional order feedback control of a dc motor," *Journal of Electrical Engineering*, vol. 60, no. 3, p. 117128, 2009.

[11] T. Qingshun, W. Chunfu, Y. Yuanhui, L. Guodong, and Z. Fengyu, "Design and implementation of fractional order controller for service robots," *International Journal of Control and Automation*, vol. 8, no. 5, pp. 209–220, 2015.

[12] *Fractional-order Proportional Derivative Controller Tuning for Motion Systems*. London: Springer London, 2010, pp. 107–120.

[13] K. Khandani and A. A. Jalali, "Robust fractional order control of a dc motor based on particle swarm optimization," in *MEMS, NANO and Smart Systems*, ser. Advanced Materials Research, vol. 403. Trans Tech Publications, 2 2012, pp. 5030–5037.

[14] M. F. Silva and J. A. T. Machado, "Fractional order pd joint control of legged robots," 2006.

[15] V. H. Haji and 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*, vol. 58, pp. 256 – 264, 2017.

[16] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[17] A. Rajasekhar, P. Kunathi, A. Abraham, and M. Pant, "Fractinal order speed control of dc motor using levy mutated artificial bee colony algorithm," in *2011 World Congress on Information and Communication Technologies*, Dec 2011, pp. 7–13.

[18] V. Haji Haji and C. A. Monje, "Fractional-order pid control of a chopper-fed dc motor drive using a novel firefly algorithm with dynamic control mechanism," *Soft Computing*, Jun 2017. [Online]. Available: https://doi.org/10.1007/s00500-017-2677-5

[19] F. Martn, C. A. Monje, L. Moreno, and C. Balaguer, "De-based tuning of pid controllers," *ISA Transactions*, vol. 59, pp. 398 – 407, 2015.

[20] S. Al-Ratrout, A. Saleem, and 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)*, Dec 2015, pp. 1–6.

[21] P. Pierro, S. Martinez, A. Jardon, C. Monje, and C. Balaguer, "Teo: Full-size humanoid robot design powered by a fuel cell system," *An International Journal on Cybernetics and Systems*, vol. 43, no. 3, pp. 163 – 180, 2012.

[22] Y. Chen and K. L. Moore, "Relay feedback tuning of robust pid controllers with iso-damping property," *Trans. Sys. Man Cyber. Part B*, vol. 35, no. 1, pp. 23–31, Feb. 2005. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2004.837950

[23] E. Powell, M.J.D.and P. Rabinowitz, *Numerical Methods for Nonlinear Algebraic Equations*. London: Gordon and Breach, 1970, ch. A FORTRAN subroutine for solving systems of nonlinear algebraic equations.

[24] *10. Nonlinear Least Squares*, pp. 218–238. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611971200.ch10