2021 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 28-29, Santa Maria da Feira, Portugal

2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 978-1-6654-3198-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICARSC52212.2021.9429809

Improving a position controller for a robotic joint

A. Paulo Moreira
Faculty of Engineering
University of Porto
INESC Technology and Science
Porto, Portugal
amoreira@fe.up.pt

José Lima
Research Centre in Digitalization and Intelligent Robotics (CeDRI)
Instituto Politécnico de Bragança
INESC Technology and Science
Porto, Portugal
jllima@ipb.pt

Paulo Costa
Faculty of Engineering
University of Porto
INESC Technology and Science
Porto, Portugal
paco@fe.up.pt

***Abstract*—There are several industrial processes that are controlled by a PID or similar controller. In robotics it is also usual the need of position control of joints. Tune a controller is the process to obtain the gains that optimise the behaviour of the system while maintaining its stability and robustness. This paper presents an approach of tuning a speed controller using the Internal Model Control (IMC) method and a position controller using the second order Bessel prototype while testing in different controllers methodology, such as PID, Cascade and feedforward combination with dead zone compensation. In order to compare the controllers, results for an Hermite reference position will allow to validate the proposed solution.**

***Index Terms*—PID, Cascade controller, Feedforward, Gains, Tuning, IMC.**

I. Introduction

In robotics the performance of position control of joints is a key factor. Theoretically it seems a simple task but in practical applications we must deal with several nonlinearities like the motor dead zone and the saturation of the control signal. At the same time the dynamical model for the motor plus load is not perfect and it can change with time usually due to temperature changes and viscous changes [1]. A cascade controller with a Proportional plus Derivative (PD) controller for the position and a Proportional plus Integral (PI) controller for the speed is a good solution easy to implement and to tune and with better results than the normal PD controller without cascade. In theory we have zero error to constant reference positions but in real situations the position error is non zero because of the dead zone of the motor. Cascade control minimizes this error. Nevertheless, the integral action in the speed controller can create oscillations in a constant reference position, related also to the dead zone of the motor. For non-constant reference signals, the PD+PI cascade controller has a constant error for a ramp signal and infinite error for quadratic or higher order reference signals. The only way to have zero steady state error, even with a wrong model, is to introduce the dynamics of the reference signal in the direct path of the controller. This solution increases the order of the dynamical model resulting in a controller much more difficult to tune. By the other hand this additional dynamic only guaranties zero error in steady state and its control is purely reactive without any anticipation. A Feed Forward (FF) block in the position controller creates an anticipation signal that can decrease the error to near zero, not only in the steady state but also in the transitory phase, and at the same time does not increases the dynamics and complexity of the controller. The FF signal is based on the motor plus load dynamic model. The question is how robust is this controller to changes in this model. In this paper is performed an exhaustive analysis of these questions using a real example. Finally the reference signal used is a cubic Hermite spline. With that function it is possible to go from one position to another with a certain initial and final derivative. This signal is much better than a step change in the reference position, something that it is always impossible to follow with near zero error. Adjusting the time requested to change from one position to the other position it is possible to avoid the limits in the control signal and to have a very good result in terms of following error and not only in a constant reference position. The remaining of the paper is organised as follows: after the introduction, section II presents the related work where other approaches for tuning position controllers are addressed. Section III details the system architecture that was implemented to test and compare the different control methods. Next section will detail each one of the controllers used on this work. Further section V will present and discuss the results obtained by the proposed controllers as well as the real implementation of the proposed controller. Finally, last section will conclude the paper and will point some future work directions.

II. Related work

The control and its related topics such as tuning, gains, controller types, among others have been addressed for several years ago [2], [3]. Regarding the controller types, [4] presents an exhaustive list with more than 30 controller types splited by 4 categories. The first one has been commercially available for over 70 years for many batch processing operations. The second category, are referred to as classical because they have been used in industry for over 40 years. The third category have been widely used in industry and is described in many process control textbooks [5]. Finally, there is the process control research in this area that has been largely concerned with three methods: knowledge-based systems, neural networks,

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

fuzzy logic, and various combinations of these techniques. As presented, there are several works that address tune and comparison among different control schemes such as feedback, feedback plus feed-forward, cascade and cascade plus feed-forward using PID controllers [6] but it lacks the motor Dead Zone compensation that is the proposed controller in the presented paper.

III. System Architecture

The simulation environment, based on SimTwo [7], was used to create the environment as the test-bed for this comparison between different controller methods, since it presents a realistic dynamic and lower sensory errors comparing with real scenario. As presented in figure 1, the SimTwo imports the world scene in a *.xml* file and uses it to build the dynamics of the system. Internally, there is an embedded script that is configured to run cyclically at 20 Hz (presented by the yellow box) where the several controllers will be implemented. It receives the angle of the motor θ and sends to the motor the voltage (u). The controller will use as angular position reference (θ_{ref}) an Hermite spline [8].

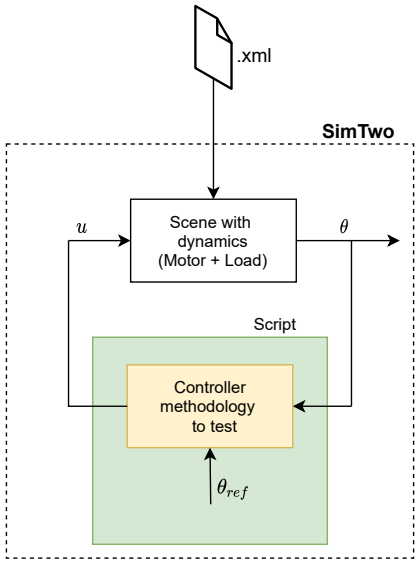

Fig. 1. Developed Simulation environment architecture for controller comparison purpose.

The model of the motor was based on JGY-371 and its parameters were measured and estimated according to the method presented at [9]. The table I presents the values obtained by the estimation of parameters where the main electrical parameters can be described as: the r_i is the internal electrical resistance of the armature, the L_i is the equivalent inductance of the armature, the K_i is the torque constant, the V_{MAX} is the maximum supported voltage and the I_{MAX} is the maximum allowed current. On the other hand, the mechanical parameters are also addressed where *rotor j* is the inertia of the rotor axis, *gear ratio* is the gear ratio, the *Friction bv* is the viscous friction and finally the *Friction fc* is the Coulomb friction. Although the inertia perceived at each joint changes dramatically with the robot configuration (joints angles), in this paper it is not considered since this controller is proposed to address one joint only.

TABLE I
Motor parameters

Parameter	Value	Units
r_i	7.1	Ω
L_i	3.4	mH
K_i	0.509	$N.m/A$
V_{MAX}	15	V
I_{MAX}	10	A
rotor j	0.0037	$kg.m^2$
gear ratio	30	-
Friction bv	0.0004655	$N.s/rad$
Friction fc	0.8	N

The developed scene to test the different control methods is composed by a fixed motor (fixed to a tower) and a load (represented by a red bar) that is coupled to the motor axis, as presented in Figure 2.

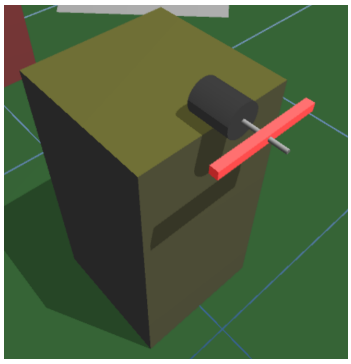

Fig. 2. SimTwo scene used for the tests of the implemented controllers.

The load is a light red cobblestone with characteristics stressed on table II.

TABLE II
Load characteristics

Dimension	Value	Units
Length	0.25	m
Height	0.02	m
Depth	0.02	m
Mass	0.25	kg
Inertia	0.001310	$kg.m^2$

IV. Position Controller

The main goal of this paper is the comparison of five position controller methods detailed on the further subsections as: PID controller without feedforward, PID Controller with Inverse Dynamic Feedforward, Cascade PID controller without Inverse Dynamic Feedforward, Cascade PID controller with Inverse Dynamic Feedforward and, at the end, a Cascade PID controller with Dead Zone Compensation (DZC) and Inverse Dynamic Feedforward. The same input reference of Hermite

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

polynomials [8] was applied. Next section V will address the results of each control method and a comparison will be stressed. The speed controller is based on a PI controller and its gains are adjusted using the Internal Model Control (IMC) method [10], [11]. After, the speed control block is approximated as a first order system with a unitary gain and a time constant τ_{CL}.

On the other side, the position controller is based on a PD controller which gains are calculated by the second order Bessel prototype.

A. PID Controller without Feedforward

The well known PID (Proportional Integral Derivative Controller) is nowadays used to control a huge number of industrial processes. Nevertheless, it still lacks on some applications such as positioning controller. Figure 3 presents the main block of this controller whereas equation (1) is used to model the PID controller where the used notation $u(t)$ is the output of the controller, $e(t)$ is the error input, K_c is the proportional gain, T_i if the integral time and T_d is the derivative time gains of the controller. A PD controller will be used since the integral component does not bring any advantage to this application. The $u(t)$ is the input of Motor and Load system which outputs a speed (ω). After an integration block ($1/s$) the position is expressed as θ.

$$u(t) = K_c \cdot \left(e(t) + \frac{1}{T_i} \cdot \int e(t)dt + T_d \cdot \frac{de(t)}{dt} \right) \qquad (1)$$

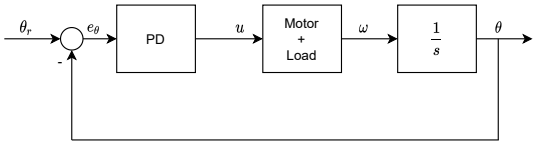

Fig. 3. PD position controller block diagram.

B. PID Controller with Inverse Dynamic Feedforward

In the PID Controller with Inverse Dynamic Feedforward method, a Feedforward block (FF) can be added, as presented in figure 4. This controller responds to its control signal in a pre-defined way without responding to how the load reacts [12]. The FF transfer function is the inverse of the process transfer function (Motor+load+integration). It is the opposite of a system that only has feedback, which adjusts the input to take account of how it affects the load.

C. Cascade controller

Cascade control uses two controllers with the output of the first controller providing the set point for the second controller where the feedback loop for one controller nestling inside the other. This controller can give an improved response to disturbances [13]. It is presented by figure 5.

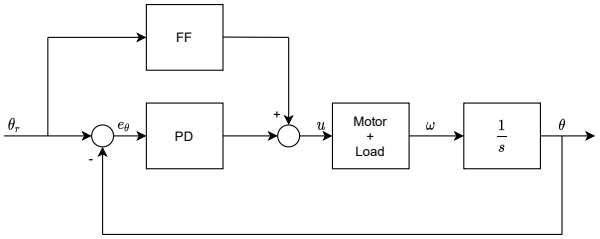

Fig. 4. PD position controller with Inverse Dynamic Feedforward block diagram.

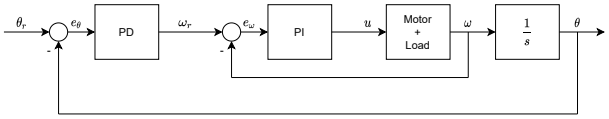

Fig. 5. Cascade controller block diagram.

D. Cascade PID controller with Inverse Dynamic Feedforward

The Feedforward block can be added to the previous system resulting in the system presented in figure 6. The Inverse dynamic feedforward controller would be able to control the system if the model is completely accurate. Simplifications of the model will need a compensation that can be carried by the PD position controller. Although the linear models disregard nonlinearities, they can be used to their compensation by taking into account deviations. The advantage is that arbitrary effects can be compensated, even effects which are not included in the complex nonlinear model [14].

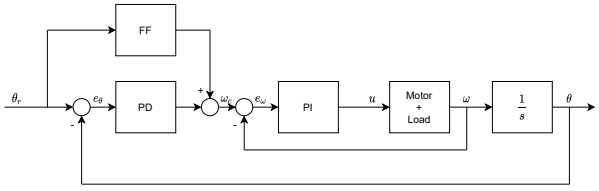

Fig. 6. Cascade controller with Inverse Dynamic Feedforward block diagram.

E. Cascade PID controller with Dead Zone Compensation (DZ) and Inverse Dynamic Feedforward

The last controller method that was proposed is based on the previous Cascade controller with Inverse Dynamic Feedforward but the speed controller will own a compensation for the dead zone of the motor. This non-linearity can be compensated by an inverse model [15]. The integral part of the controller should be carefully handled with an antiwindup procedure and initialization to avoid bumps in the control signal. By other words, the starting direction of the motion, from a stopped state, is known so, with a proper integral initialization (Ierror), the voltage will be triggered at the dead zone limit ($offset\_dz$), as presented by next code where *Per* is the position error, *speed_ref* is the speed reference, *MIN_ER* and *MIN_VEL* are the threshold of minimum position error

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

and the minimum speed respectively, where we can consider that the motor is stopped. *Signal_offset* is 1 or -1 according to the direction of the future motion.

```
if (abs(Per) < MIN_ER && abs(speed_ref) < MIN_VEL)
{
  speed_ref = 0;
  Ierror = Ti*(signal_offset*offset_dz)/Kc;
  Volts_to_motor = signal_offset*offset_dz;
}
```

The Cascade PID controller with Dead Zone Compensation (DZC) and Inverse Dynamic Feedforward block' diagram is presented by figure 7.

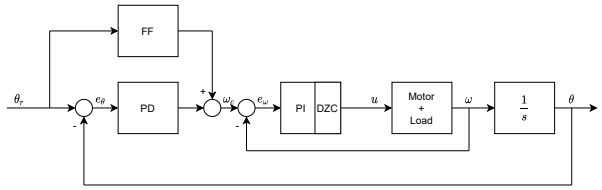

Fig. 7. Cascade PID controller with Dead Zone Compensation (DZC) and Inverse Dynamic Feedforward.

V. Results

To test the proposed controller strategies, the same input reference of Hermite polynomials [8] was applied for each controller. The transitory time of this function was adjusted to have a large motion but without reach the voltage limits of the motor's drive. The gains are tuned according to the Internal Model Control (IMC) method [10] and then the controller is estimated as a first order system with a unitary gain and a time constant τ_{CL}. At the beginning, it was optimised the τ_{CL}, as presented in subsection V-A. Then a comparison between all controllers will be made and presented in subsection V-B followed by a discussion presented in subsection V-C. Subsection V-D will stress the proposed system with a perturbation. Finally, the proposed cascade controller was implemented and tested in a real application, as detailed on subsection V-E.

A. Performance and robustness analysis of the Cascade PI controller

This subsection presents an analysis of the cascade controller as a function of time constant (τ_{CL}). For the adjustment of the position controller it was used a Bessel prototype with a settling time equal to 0.8 seconds. It is used to find the τ_{CL} that optimizes a global performance index calculated with the sum of all error results for each situation. As presented on table III, it will be used the τ_{CL} = 0.25 taking into account that with this value we have the best balance between absolute position errors and robustness to model errors.

B. Controllers results

For evaluation, it was measured the maximum absolute error, the mean absolute error, the mean squared error and the root mean squared error for the proposed controllers. Figures 8, 9, 10, 11 and 12 present the response of the controllers addressed: PID controller without feedforward, PID Controller with Inverse Dynamic feedforward, Cascade PID controller without Inverse Dynamic feedforward, Cascade PID controller with Inverse Dynamic feedforward and at the end Cascade PID controller with Dead Zone Compensation and Inverse Dynamic feedforward, respectively. The blue line is the input reference and the orange one is the position of the system whereas the grey line is the control output signal (voltage) applied to the motor.

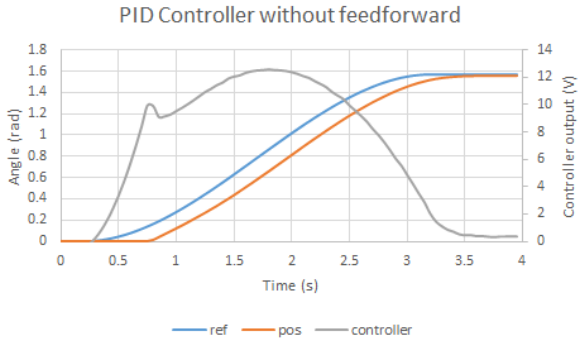

Fig. 8. Positioning Results for PID Controller without feedforward.

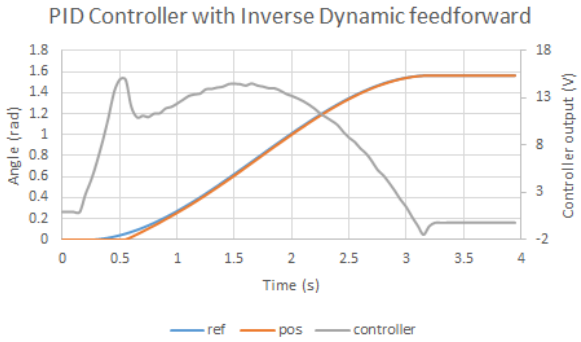

Fig. 9. Positioning results for PID Controller with Inverse Dynamic feedforward.

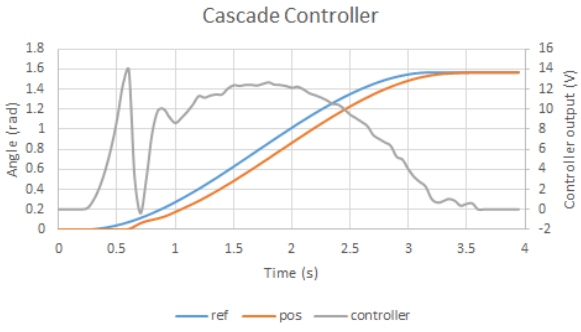

Fig. 10. Positioning results for Cascade controller without feedforward.

Next subsection will make a brief discussion of comparison results between controllers as well as the disturbance analysis.

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE ANALYSIS OF CLOSED LOOP GAIN

τ_{CL} **Gains for position controller**	**Model and speed controller parameters**	**Total absolute error (all period) [rad]**	**Total absolute error (transitory) [rad]**	**Total absolute error (perturbation 5 Volts) [rad]**	**Global performance index [rad]**
τ_{CL} = 0.05 Kc = 1.71 Td = -0.288 sec	Nominal Model Kc=36.4 Ti=0.122 sec	1.07	0.274	0.258	5.256
	K=K*1.25 τ=τ*0.75 Kc=21.9 Ti=0.0915 sec	0.818	0.121	0.277	
	K=K*0.75 τ=τ*1.25 Kc=60.7 Ti=0.153 sec	1.78	0.415	0.243	
τ_{CL} = 0.1 Kc = 3.42 Td = 0.00387 sec	Nominal Model Kc=18.2 Ti=0.122 sec	0.988	0.132	0.221	3.5792
	K=K*1.25 τ=τ*0.75 Kc=10.9 Ti=0.0915 sec	0.458	0.0482	0.256	
	K=K*0.75 τ=τ*1.25 Kc=30.3 Ti=0.153 sec	1.001	0.271	0.204	
τ_{CL} = 0.15 Kc = 5.13 Td = 0.101 sec	Nominal Model Kc=12.1 Ti=0.122 sec	0.573	0.0793	0.211	3.2433
	K=K*1.25 τ=τ*0.75 Kc=7.28 Ti=0.0915 sec	0.856	0.095	0.256	
	K=K*0.75 τ=τ*1.25 Kc=20.2 Ti=0.153 sec	0.757	0.222	0.194	
τ_{CL} = 0.2 Kc = 6.84 Td = 0.15 sec	Nominal Model Kc=9.1 Ti=0.122 sec	0.403	0.057	0.204	3.166
	K=K*1.25 τ=τ*0.75 Kc=5.46 Ti=0.0915	1.043	0.12	0.256	
	K=K*0.75 τ=τ*1.25 Kc=15.2 Ti=0.153	0.694	0.201	0.188	
τ_{CL} = 0.25 Kc = 8.56 Td = 0.179 sec	Nominal Model Kc=7.28 Ti=0.122sec	0.289	0.0409	0.2	2.9979
	K=K*1.25 τ=τ*0.75 Kc=4.37 Ti=0.0915	1.118	0.133	0.256	
	K=K*0.75 τ=τ*1.25 Kc=12.1 Ti=0.153	0.594	0.183	0.184	
τ_{CL} = 0.3 Kc = 10.3 Td = 0.199 sec	Nominal Model Kc=6.07 Ti=0.122 sec	0.231	0.0319	0.199	3.0969
	K=K*1.25 τ=τ*0.75 Kc=3.64 Ti=0.0915	1.184	0.144	0.256	
	K=K*0.75 τ=τ*1.25 Kc=10.1 Ti=0.153	0.693	0.178	0.18	
τ_{CL} = 0.4 Kc = 13.7 Td = 0.223 sec	Nominal Model Kc=4.55 Ti=0.122 sec	0.129	0.0206	0.197	3.1826
	K=K*1.25 τ=τ*0.75 Kc=2.73 Ti=0.0915	1.3	0.156	0.256	
	K=K*0.75 τ=τ*1.25 Kc=7.59 Ti=0.153	0.778	0.169	0.177	

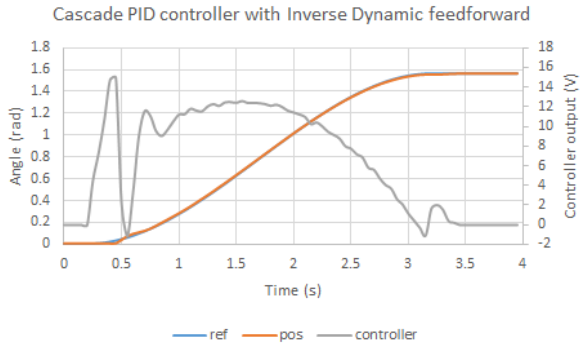

Fig. 11. Positioning results for Cascade controller with Inverse Dynamic Feedforward.

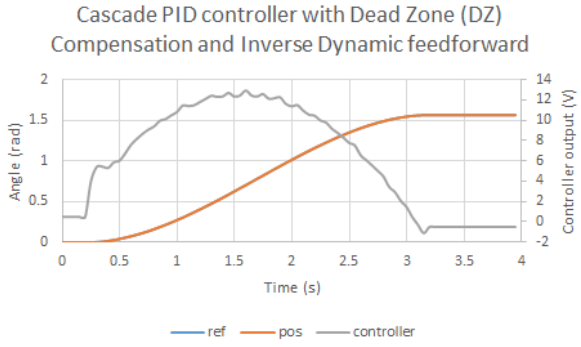

Fig. 12. Positioning results for Cascade controller with Dead Zone (DZ) compensation and Inverse Dynamic feedforward

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
TRANSITORY ERROR

Controller method	Maximum absolute error [rad]	Mean absolute error [rad]	Root mean squared error [rad]
PID controller	0.2008	0.1408	0.1523
PID Controller with ID FF	0.0529	0.0152	0.0182
Cascade controller	0.1523	0.1026	0.1116
Cascade controller with ID - FF	0.0265	0.0056	0.0072
Cascade controller with DZ and ID - FF	0.0025	0.0009	0.0011

TABLE V
STEADY-STATE ERROR

Controller method	Maximum absolute error [rad]	Mean absolute error [rad]	Root mean squared error [rad]
PID controller	0.0070	0.0059	0.0059
PID Controller with ID FF	0.0036	0.0036	0.0036
Cascade controller	0.0030	0.0017	0.0017
Cascade controller with ID - FF	0.0030	0.0018	0.0018
Cascade controller with DZ and ID - FF	0.0014	0.0014	0.0014

C. Results Discussion

The tables IV and V show the errors for the transitory and for the steady-state periods of the previous tests.

Regarding the figure 13 that presents the total error (transitory plus steady-state) for each controller, it is possible to verify that the cascade controller with DZC and Inverse Dynamics feedforward obtains the lowest Maximum absolute error, the lowest Mean absolute error and the lowest Root mean squared error.

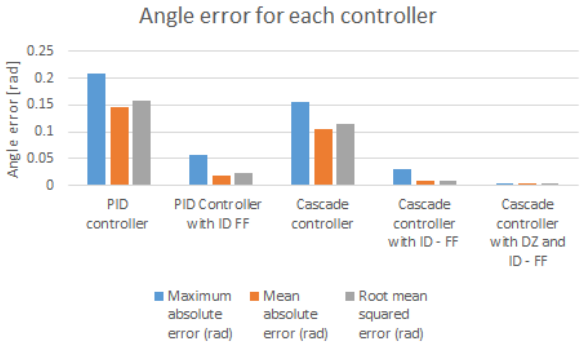

Fig. 13. Comparison of total error for each controller.

D. Disturbance Analysis

In order to verify how the proposed controller behaves, a perturbation on the plant was introduced. A signal of 5 Volts are summed in the motor voltage at 30% of the variation of the reference. Figure 14 presents the controller compensation, ensuring the position of the motor.

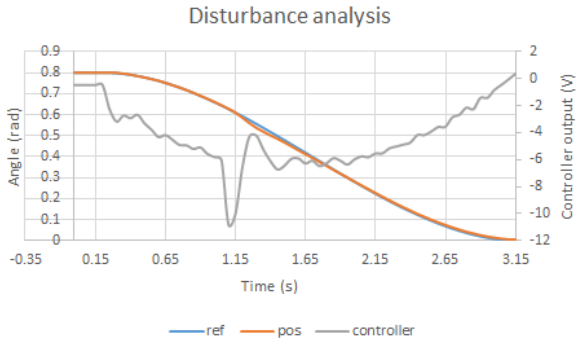

Fig. 14. Disturbance Analysis.

E. Real implementation

In order to validate the proposed controller, a real application was stressed with the cascade position controller. The developed application named PNEUMA is an add-on device that automates the self-inflating bag (Ambu), thus presenting itself as a last resort solution to ventilate patients having difficulty breathing. This field ventilator includes and enhances the main benefits of the self-inflating bag, recognised by healthcare professionals. The device is used regularly in the clinical context, but is now being adapted to deal with the COVID-19 pandemic [17]. It is a very interesting application to test the controller since the Ambu changes the motor load based on elasticity and friction that depends on the lung state and the desired ventilation. This is why it is so relevant the robustness analysis.

The main architecture for this device is presented on figure 15, where the main controller is an Arduino Uno board that controls a DC motor by the TB9051FTG Single Brushed DC Motor Driver Carrier. The motor owns an encoder that will feedback the microcontroller with the position and speed of the motor shaft. An I/O interface was developed to allow technicians to change the parameters of the respiratory system, such as frequency, amplitude among others.

The figure 16 presents the inside hardware and mechanical parts that compose the PNEUMA ventilator. It is possible to see the gear of the motor with a red line, the processing unit at yellow and the interface at blue.

As result for this real application, the upper graphic at figure 17 presents the speed reference (line) and the measured speed (red dots). This speed reference is created by the PD position controller that is presented at the bottom figure (black line is the Hermite reference curve whereas the red line represents the measured position). The control period used was 25 ms. Notice that the motor is coupled with a low cost magnetic

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.

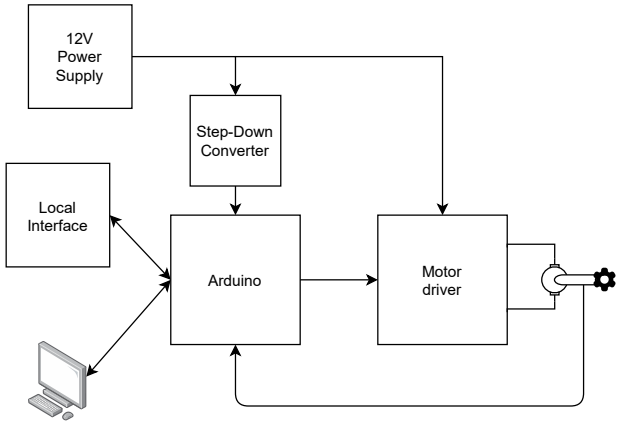

Fig. 15. Main architecture of the PNEUMA ventilator.

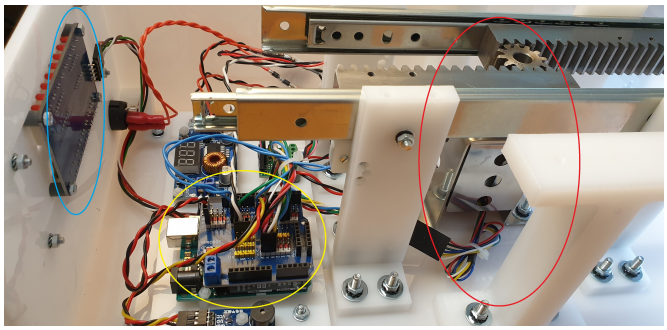

Fig. 16. Hardware of PNEUMA ventilator.

encoder thus low number of pulses per revolution that comes out with noise on measures.

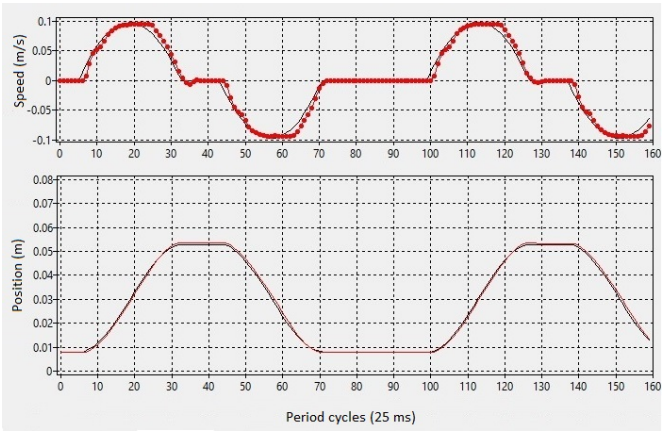

Fig. 17. References and measurements for speed (top picture) in m/s and position (bottom picture) in m.

A video demonstrating the developed ventilator is presented on site [18] that shows the PNEUMA validation on a respiratory simulator.

VI. Conclusion and Future Work

This paper presented different controllers than can be used to position a robotic joint. In this work it was used the Internal Model Control (IMC) method to tune and adjust the speed controller gains that is based on a PI controller. The position controller gains are calculated by the second order Bessel prototype. All controllers were stressed with an input reference of Hermite polynomials that allowed to compare the results. The Cascade PID controller with Dead Zone Compensation (DZC) and Inverse Dynamic Feedforward was the one with lowest error following the position reference. A real application of the proposed position controller was presented on this paper as result validating the solution.

References

[1] Mark W. Spong, Seth Hutchinson and W. Vidyasagar, Robot modelling and control, John Wiley and Sons, Inc. 2006.
[2] Karl J. Astrom, T. Hagglund, PID Controllers Theory, Design and Tuning, (1995) Second Edition, ISA Verlag, ISBN: 978-1-55617-516-9
[3] Michael W. Foley, Navin R. Ramharack, and Brian R. Copeland, Comparison of PI Controller Tuning Methods, Ind. Eng. Chem. Res. 2005, 44, 17, 6741-–6750, (2005) DOI: 10.1021/ie040258o
[4] Štefan Kozák, State-of-the-art in control engineering, Journal of Electrical Systems and Information Technology, Volume 1, Issue 1, 2014, 1–9, ISSN 2314-7172, https://doi.org/10.1016/j.jesit.2014.03.002
[5] Š. Kozák, Development of control engineering methods and their applications in industry, 5th International Scientific-Technical Conference Process Control 2002, Kouty nad Desnou, Czech Republic, June 9–12 (2002)
[6] R. Kumar, S.K. Singla, V. Chopra, Comparison among some well known control schemes with different tuning methods, Journal of Applied Research and Technology, 13 (2015) 409–415, DOI:10.1016/j.jart.2015.07.007
[7] Costa, P. et al., SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software, Theory and Applications of Mathematics & Computer Science 1 (2011): 17–33.
[8] Hermite, C. (1864). "Sur un nouveau développement en série de fonctions" [On a new development in function series]. C. R. Acad. Sci. Paris. 58: 93–100. Collected in Œuvres II, 293–303.
[9] Pinto V.H., Gonçalves J., Costa P. (2021) Model of a DC Motor with Worm Gearbox. In: Gonçalves J.A., Braz-César M., Coelho J.P. (eds) CONTROLO 2020. CONTROLO 2020. Lecture Notes in Electrical Engineering, vol 695. Springer, Cham. https://doi.org/10.1007/978-3-030-58653-9_61
[10] Paul S. Fruehauf, I-Lung Chien, Mark D. Lauritsen, Simplified IMC-PID tuning rules, ISA Transactions, Volume 33, Issue 1, 1994, Pages 43-59, ISSN 0019-0578, https://doi.org/10.1016/0019-0578(94)90035-3.
[11] Yongho Lee, Sunwon Park, Moonyong Lee, Coleman Brosilow, PID controller tuning for desired closed-loop responses for SI/SO systems, AIChE Journal Volume 44 Issue 1, Pages 106 - 115, 2004.
[12] Meckl, P.H. and Seering, W.P., "Feedforward Control Techniques Achieve Fast Settling Time in Robots", Automatic Control Conference Proceedings. 1986, pp 58–64.
[13] William Bolton, Chapter 4 - Control Systems, Editor(s): William Bolton, Instrumentation and Control Systems (Second Edition), Newnes, 2015, Pages 81–98, ISBN 9780081006139, https://doi.org/10.1016/B978-0-08-100613-9.00004-3.
[14] Grotjahn M, Heimann B. Model-based Feedforward Control in Industrial Robotics. The International Journal of Robotics Research. 2002;21(1):45–60. doi:10.1177/027836402320556476
[15] N. J. Ahmad, H. K. Ebraheem, M. J. Alnaser and J. M. Alostath, Adaptive control of a DC motor with uncertain deadzone nonlinearity at the input, 2011 Chinese Control and Decision Conference (CCDC), Mianyang, China, 2011, pp. 4295–4299, doi:10.1109/CCDC.2011.5968982.
[16] Daniel E. Rivera, Manfred Morari and Sigurd Skogestad, Internal Model Control. 4 PID Controller Design, Ind. Eng. Chem. Process Des. Dev. 252–265, 1986.
[17] Official Website of PNEUMA development, https://pneuma.inesctec.pt/en/pneuma/
[18] Online video of PNEUMA at respiratory simulator, https://www.youtube.com/watch?v=MLEuX2VdAOM

Authorized licensed use limited to: b-on: Instituto Politecnico de Braganca. Downloaded on February 22,2023 at 23:46:04 UTC from IEEE Xplore. Restrictions apply.