A Study of the Influence of Stochastic Fractional-Order Delay Dynamics in a Networked Control System

Jairo Viola * **Piotr Oziablo** ** **YangQuan Chen** *

* *University of California, Merced, CA 95340 USA*
e-mails: (jviola, ychen53) @ucmerced.edu.
** *Bialystok University of Technology, Bialystok, Poland*
(e-mail:p.oziablo@student.pb.edu.pl)

Abstract: Stochastic communication delays are present in networked control systems and have a significant influence over its closed-loop performance and stability. These delays are time-variant and have infinite variance, indicating fractional-order behavior that can be modeled using α-stable distribution. This paper presents a quantitative analysis of the effects of fractional-order α-stable distribution random communication delays in a networked control system for a temperature control application. The networked closed-loop system assessment is performed using four different controllers. Two integer order PI controllers tuned using classic tuning methods, and two fractional-order PI controllers tuned using standard tuning rules for fractional-order system. One hundred numerical simulation experiments are performed, modeling the random communication delays between the plant and controller using an α-Stable distribution with different α values. Obtained results show that fractional-order controllers can deal better with time-variant fractional-order random communication delays that follows an α-Stable distribution compared with the integer-order controllers.

Keywords: Networked Control Systems, Fractional-Order delay dynamics, α-Stable distribution.

1. INTRODUCTION

Networked control systems (NCS) have increased its relevance in the industry due to their capability of performing closed-loop control over the internet networks, reducing wiring and infrastructure costs, allowing the design of more flexible control strategies for multi-agent systems in the context of Industrial Internet of Things (IoT). There are several applications of NCS for different types of control systems. For example, Youness and Lobusov (2019) shows the application of an NCS for controlling a car suspension system considering each damper as a network element. In Wu et al. (2017), a fuzzy logic-based control law is designed for an NCS system for the position control of an inverted pendulum. Also, Li et al. (2019) presents an NCS system with fault-tolerant capabilities for remote airplane control. However, NCS faces many control challenges as external attacks, bandwidth usage, and communication delays. The last one becomes critical for closed-loop control because it can lead the system into unstable behavior that avoids reaching the desired closed-loop specifications. Stochastic processes can describe the communication delays with a spiky nature (which is specific, e.g., for TCP/IP network), which leads into an infinite variance with a heavy tail probability distribution that suggests a fractional-order behavior. Indeed, according to Chen (2010); Mukhopadhyay et al. (2009), the random nature of the communication delays for NCS cannot be fitted as Normal Gaussian distribution, but by using α-Stable distribution, which gives a more accurate estimation of the NCS random communication delay.

This paper presents a simulation study of the influence of fractional-order random communication delays on the closed-loop performance of NCS. The analyzed system corresponds to a remote-controlled temperature process developed in the Mechatronics, Embed System, and Automation Laboratory (MESALab) of the University of California, Merced. The random communication delays for this system have a spiky nature and follow an α-Stable distribution. The process is modeled as a First Order plus Dead Time (FOPDT) system, and four different controllers are tested in the presence of random communication delay. Two controllers correspond to a fractional order PI (FOPI) controller tuned using the fractional-order M-constrained Integral Gain Optimization method (FMIGO) Monje et al. (2010) and an optimal constrained PI controller design based on the system jitter margin (JM) proposed by Bhambhani et al. (2008). On the other hand, the other two controllers are integer order PI (IOPI) controllers tuned using the Modified Ziegler-Nichols method Xue et al. (2007) and the Approximate M-constrained Integral Gain Optimization (AMIGO) method proposed by Åström and Hägglund (2006). The stochastic communication delays are simulated using α-stable distribution to analyze the closed-loop system behavior of the system considering variations on the delay fractional-order α. One hundred numerical simulations are performed to analyze the effects of fractional order random noise in the time response and control action. The Root Mean Square Error (RMSE) and the Root Mean Square value (RMS) are employed as performance indices for the time response and control action, respectively. The main contribution of this paper is to perform a quantitative study of the effect of fractional-order random communication delays over the closed-loop performance of the system and check how this effect can be alleviated using fractional-order controllers. This paper is structured as follows. Section 2 presents a brief background of the system delay dynamic behavior and the α-Stable distribution. Section 3 describes the NCS system to be analyzed and the fractional-order delay modeling. Section 4 performs a simulation study using the proposed controllers to analyze the closed-loop performance of the system in the presence of random delay in the communication channel. Finally, conclusions and future works are presented.

2. NCS DELAY DYNAMICS

2.1 α-Stable distribution

According to Chen (2010) and Nikias and Shao (1995), a random variable X can be considered α-Stable if its characteristic function is given by (1), which is modeled by four parameters α, β, γ, δ. The exponent α determines the thickness of the probability density function (PDF) tail, which relates to the spiky behavior of the random delay. It means that for larger values of α, the spiky behavior is reduced and is closer to a normal Gaussian distribution. The skewness factor β indicates if the distribution is skewed to the right or left tail if its value is positive or negative, respectively. The factor γ is related to the dispersion of the distribution. Finally, factor δ is the scale parameter and represents the mean or median of the entire distribution.

$$\varphi(v) = \exp\{j\delta\varphi - \gamma|v|^{\alpha}[1 + j\beta\text{sign}(v)\omega(v,\alpha)]\} \qquad (1)$$

where:

$$\omega(v,\alpha) = \begin{cases} \tan\frac{\alpha\pi}{2} & for \;\; \alpha \neq 1 \\ \frac{2}{\pi}\log|v| & for \;\; \alpha = 1 \end{cases}$$

$$\text{sign}(v) = \begin{cases} 1 & for \;\; v > 0 \\ 0 & for \;\; v = 0 \\ -1 & for \;\; v < 0 \end{cases}$$

and

$$0 < \alpha \leq 2, -1 \leq \beta \leq 1, \gamma > 0, -\infty < \delta < \infty.$$

2.2 α-Stable applied to network delay modeling

According to Chen (2010), α-Stable distributions follows two relevant properties. The first one is the stability property, which states that the sum of weighted independent α-Stable random variables is still stable under the same order α. The second one is based on the generalized central limit theorem, which states that the sum of some independent and identically distributed (i.i.d) random variables tends to a stable distribution. Besides, the generalized central limit theorem defines the randomness as a result of cumulative effects, which are distributed with heavy-tailed probability density. Based on these properties, the random behavior of communication delays in NCS can be modeled using α-Stable distribution. As an example, a random communication delay of an NCS and its variance are presented in Fig. 1 and Fig. 2. As can be observed, the burst in the delay (red circle) shows the spiky behavior, which indicates a heavy tail feature of the communication delay probability distribution. On the other hand, Fig. 2 shows that the delay variance increases over time instead of converging into some constant value, which is also an indicator of fractional order behavior. This infinite variance behavior is present for any α-Stable distribution with $\alpha < 2$, which can be called fractional low order moment statistic. Notice that if $\alpha = 2$, the delay dynamics corresponds to a normal Gaussian distribution. Therefore, considering the reasons presented above, the α-Stable distribution is employed in this work to model the communication delay dynamics of the NCS.

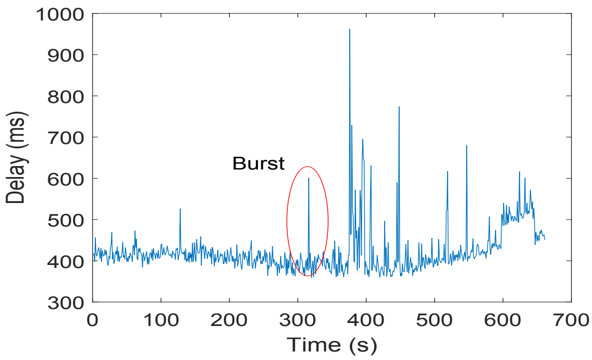

Fig. 1. Random communication delay in NCS

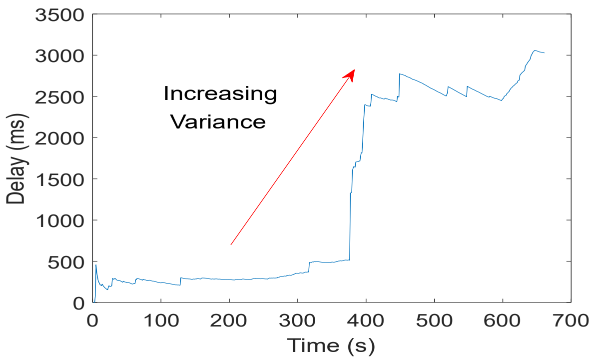

Fig. 2. Random communication delay infinite variance

3. STUDY CASE: NCS FOR TEMPERATURE DISTRIBUTION PROCESS CONTROL

The remote temperature control system show in Fig. 3 was developed in the Mechatronics, Embedded Systems and Automation Laboratory (MESALab) of the University of California Merced to analyze the stochastic delay dynamics of NCS. The system is composed of a Peltier cell (M1) that works as a heating or cooling element, a thermal infrared camera (M2) acting as a temperature sensor that allows feedback over multiple points of a surface to perform

Fig. 3. Remote temperature control system

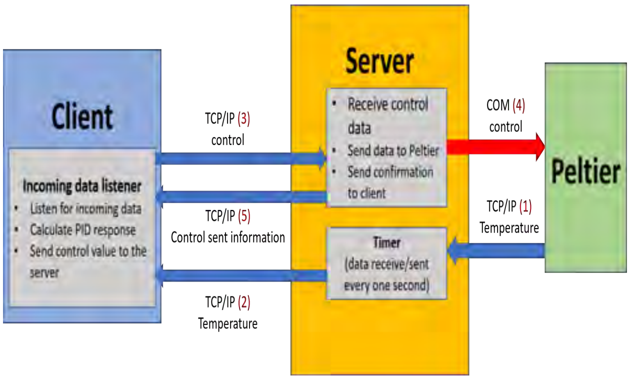

Fig. 4. NCS temperature control system architecture

uniformity temperature control. Also, a LattePanda board (M3) runs the server application for the NCS control system, which incorporates an Arduino board to manage the power applied to the Peltier cell via Pulse Width Modulation (PWM). The NCS architecture is presented in Fig. 4. The system is composed of the temperature platform shown in Fig. 3 and the NCS application, which is divided into two parts - client and server. The user's browser executes the client part of the application. It contains HTML and JavaScript, which is used to generate a graphical user interface (e.g., the input boxes where the user can provide control parameters or charts which can be used to display the results of the control process). Additionally, JavaScript code is used to calculate and send to the server current control values for the Peltier module, which are calculated based on the temperature values received by the client from the server application. The server role (besides generation of the client code because it also works as a Web application server) is to receive the temperature information from the thermal infrared camera and send the data to the client. The data is send to client with specific interval that could be e.g. 0.5[s], 1[s], 2[s], etc. The server also receives the control action information from the client and applies to the Peltier module using serial communication (the data is sent to the Arduino board that generates the PWM signal used to control Peltier temperature). A more detailed description of the Peltier system and the NCS architecture can be found on the papers by (Viola et al., 2019; Oziablo et al., 2019).

3.1 Open-loop dynamic behavior

The Peltier temperature system without random communication delays has an open-loop behavior that can be modeled using a First Order Plus Dead Time transfer function (FOPDT) given by (2), where k is the process gain, T is the time constant of the system, and L corresponds to the system delay. After performing an identification process based on real data of the system, the FOPDT system parameters are given as $k = 0.38$, $T = 29.22$, and $L = 0.1$.

$$G(s) = \frac{k}{(Ts+1)}e^{-Ls}. \tag{2}$$

3.2 Controllers design

The closed-loop control performance of the temperature system in the presence of random communication delay is evaluated using an integer order PI controller (IOPI) given by (3) and a fractional order PI controller (FOPI) (4), where k_p and k_i are the proportional and integral terms, and γ is the order of the integral term.

$$C_{IOPI}(s) = k_p + \frac{k_i}{s} \tag{3}$$

$$C_{FOPI}(s) = k_p + \frac{k_i}{s^\gamma}. \tag{4}$$

The IOPI controller is tuned using the Modified Ziegler Nichols (ZNM) and AMIGO methods. For ZNM method Xue et al. (2007) proposes the tuning rules (5) to find k_p and k_i terms for PI controllers, where K_{cr} corresponds to the critical frequency of the system given by its open-loop gain margin, $T_c = \frac{2\pi}{w_c}$ with w_c is the open-loop gain crossover frequency, and r_b and ϕ_b are tuning parameters.

$$\begin{aligned} k_p &= K_{cr} r_b \cos(\phi_b) \\ k_i &= kp\left(-\frac{T_c}{2\pi\tan(\phi_b)}\right)^{-1}. \end{aligned} \tag{5}$$

The AMIGO method proposed by Åström and Hägglund (2006), provides a simple optimal tuning rules (6) for PI controllers depending only on the FOPDT system parameters, gain K, time constant T, and delay L.

$$\begin{aligned} k_p &= \frac{0.15}{K} + \left(0.35 - \frac{LT}{(L+T)^2}\right)\frac{T}{KL} \\ k_i &= k_p\left(0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2}\right)^{-1}. \end{aligned} \tag{6}$$

On the other hand, the FOPI controller is tuned using the FMIGO and Jitter Margin methods. The FMIGO method proposed by Monje et al. (2010) is an extension of the AMIGO method that can be applied for tuning fractional-order controllers based on the relative dead time of the FOPDT system τ defined as $\tau = L/(T + L)$. Notice that if τ is close to one, the system is delay dominated, but if τ goes to zero, the system is lag dominated. The FMIGO tuning rules are giving by (7) and (8), where the fractional-order γ depends on the relative dead time of the system τ.

$$k_p = \frac{1}{k}\left(\frac{0.2978}{\tau + 0.000307}\right),$$
$$k_i = kp\left(\frac{\tau^2 - 3.402\tau + 2.405}{0.8578}\right), \tag{7}$$

$$\gamma = \begin{cases} 1.1, \; if & \tau \geq 0.6 \\ 1.0, \; if & 0.4 \leq \tau < 0.6 \\ 0.9, \; if & 0.1 \leq \tau < 0.4 \\ 0.7, \; if & \tau < 0.1 \end{cases} . \tag{8}$$

The jitter margin method (JM) presented in Bhambhani et al. (2008), consists of an optimal set of tuning rules for FOPI controllers for FOPDT systems, which are based on the optimization of the system jitter margin due to the random delay modeled as unstructured uncertainty. The tuning rules for the jitter margin method are given by (9).

$$k_p = \frac{0.2T}{L} + 0.16$$
$$k_i = \frac{0.25}{TL} + \frac{0.19833}{L} + 0.09 \tag{9}$$
$$\gamma = \tau - 0.04L + 1.2399$$

The resulting terms of the IOPI and FOPI controllers for the FOPDT system given by (2) are shown in Table 1. Notice that in the case of ZNM IOPI controller $\phi_b = 5$ and $r_b = 0.2$. The IOPI and FOPI controllers includes the anti-windup back-calculation compensation method.

Table 1. IOPI and FOPI controllers design

Controller	Tuning method	k_p	k_i	γ
IOPI	ZNM	100.49	138.10	1
	AMIGO	111.45	86.83	1
FOPI	FMIGO	88.02	8.40	0.7
	JM	64.39	2.15	1.2393

3.3 NCS system delay modeling

Some real tests of the closed-loop NCS system were performed to model the α-Stable random communication delay, gathering the information about the communication delays of the system. For this test, the NCS system was controlled using the FOPI controller, tuned by the FMIGO method, and the NCS control was executed for a total time of 10000 seconds. Fig. 5 shows the random communication delay as well as the variance of the delay. As can be observed, the delay has spiky behavior and infinite variance. Therefore, the random communication delay can be approached using α-Stable distribution. Using the statistics toolbox of Matlab, the random communication delay PDF is fitted into a α-Stable and normal Gaussian distributions. The parameters of each distribution are presented in Table 2. Likewise, the PDF and CDF histograms for the α-Stable and Gaussian distributions are shown in Fig. 6. As can be observed, the random communication delay PDF and CDF fits better into the α-Stable rather than the Gaussian distribution. So, the communication delay can be modeled better with the α-Stable distribution.

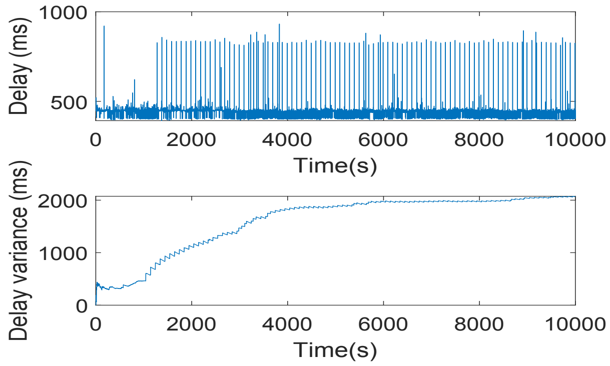

Fig. 5. Real NCS system random communication delay between Merced and Romania measured on July 8, 2019, 5:59;33 PM PST

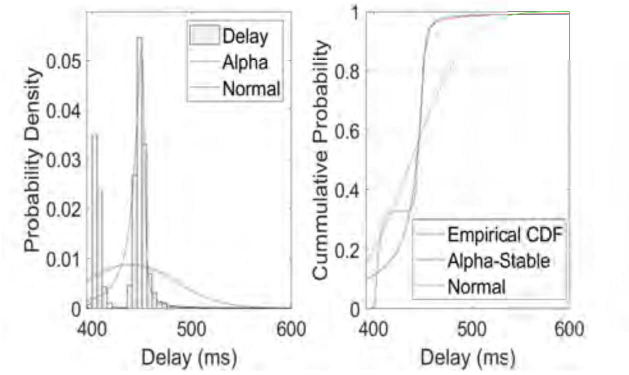

Fig. 6. PDF and CDF of the random communication delay using α-Stable and normal Gaussian distribution

Table 2. α-Stable and normal Gaussian distribution for random communication delay on NCS

Distribution	Parameter	Value
α-Stable	α	0.6880
	β	-0.6195
	γ	0.02
	δ	0.44
Normal	μ	0.4241
	σ	0.29

4. SIMULATION BENCHMARK FOR RANDOM DELAY ANALYSIS

A simulation benchmark was performed to analyze the effect of the α-Stable distribution random communication delay in the closed-loop behavior of the NCS for temperature control. One hundred numeric simulations of the closed-loop behavior of the NCS system were runt using Matlab-Simulink, considering the IOPI and FOPI controllers presented in Table 1 under the random communication delay with α-Stable distribution presented in Table 2. In each simulation, the α characteristic exponent of the random communication delay varies between $0.2 \leq \alpha \leq 1.9$, increasing at steps of 0.1. For each α exponent, the RMSE and RMS performance indices are calculated for the time response and the control action respectively to perform a quantitative performance analysis of the NCS system.

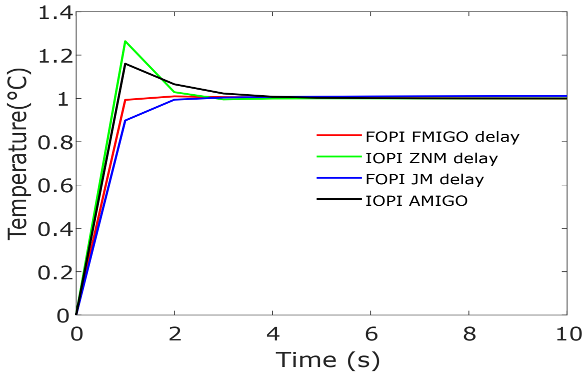

Fig. 7. Time response of the temperature system without random communication delay

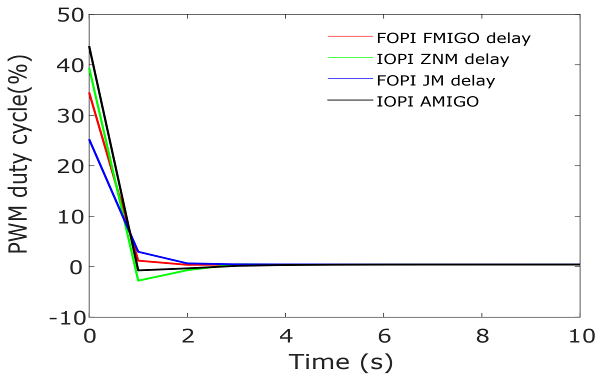

Fig. 8. Control action of the temperature system without random communication delay

4.1 *Temperature distribution control system with no random communication delay*

The time response and control action of the IOPI and FOPI controllers presented in Table 1 without random communication delay is shown in Fig. 7 and Fig. 8. As can be observed, all the proposed controllers reach the desired setpoint with some overshoot in the case of the IOPI ZNM and IOPI AMIGO controllers. Regarding the control action, it can be appreciated that FOPI controllers employ a small control effort regarding the IOPI controllers.

4.2 *NCS with random communication delay*

The time response and control action of the system for $\alpha = 1.9$ is presented in Fig. 9 and Fig. 10 in order to show the effects of random communication delay over the closed-loop system. It can be noticed that the presence of random communication delay has a significant influence on the time response of the system compared with Fig. 7 and Fig. 8. In the case of the IOPI controllers, the time response has a significant oscillating behavior. Besides, the FOPI controllers have a more robust response to the random communication delay, which is reflected by a stable time response with fewer oscillations and the reaching of a steady-state value after a certain amount of time. Regarding the control action, it can be noticed that FOPI controllers have a smooth performance compared with the IOPI controllers.

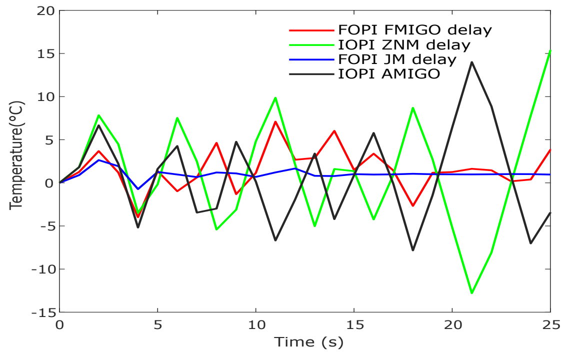

Fig. 9. Time response of the temperature system with random communication delay of $\alpha = 1.9$

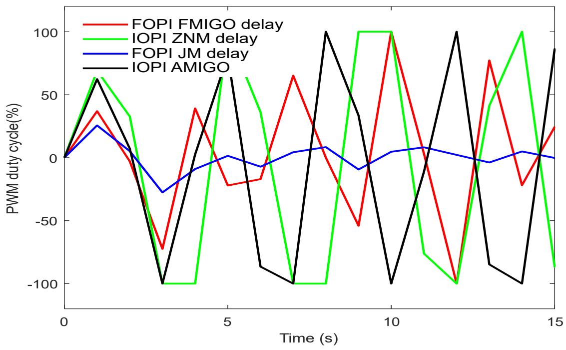

Fig. 10. Control action of the temperature system with random communication delay of $\alpha = 1.9$

4.3 *Quantitative performance analysis*

The RMSE and RMS values are calculated for the 100 numeric simulation runs of the system with the proposed IOPI and FOPI controllers, which result is shown in Fig. 11 and Fig. 12 for $0.2 \leq \alpha \leq 1.9$. As can be observed, the FOPI controllers have a lower RMSE and RMS values compared to the IOPI controllers. In particular, the FOPI tuned by the JM method exhibits the lowest values, indicating that the controller has a good performance in the presence of random communication delay. On the other hand, the lowest RMS values of the control action of the FOPI controllers require less energy to execute and fulfill the desired closed-loop specifications compared with the IOPI controllers. Besides, the black dashed lines represent the mean variation of each controller during the 100 simulations, and the colored dashed lines correspond to the upper and lower variation boundaries of each controller given as the mean plus two standard deviations. As can be observed, in the case of IOPI controllers, the variation is not higher compared with the FMIGO controller. However, the RMSE and RMS values indicate that there is a significant error over the steady-state response of the system as well as higher energy consumption. In the case of the FOPI controller tuned by the JM method, the dispersion is lower as well as the indices values, indicating a good time response and low energy consumption for a different run of the system with random communication delays.

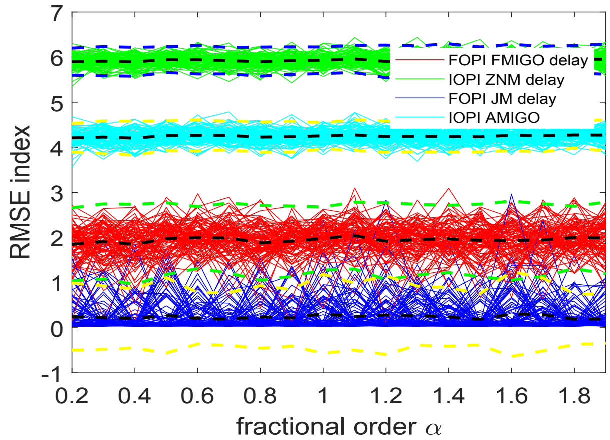

Fig. 11. RMSE value of the system time response for 100 simulation events using random communication delay with $0.2 \leq \alpha \leq 1.9$

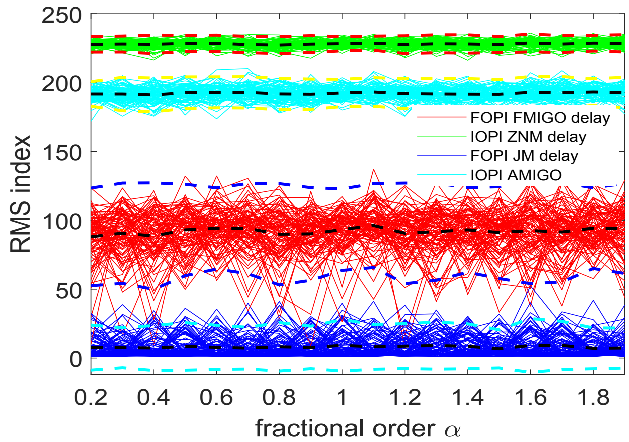

Fig. 12. RMS value of the system control action for 100 simulation events using random communication delay with $0.2 \leq \alpha \leq 1.9$

From the above test, we can say that fractional order controllers:

- Have a robust behavior to alleviate the effects of random communication delay with α-Stable dynamics
- Reduce the controller energy consumption
- Are an alternative to implementing distributed controllers on NCS for Industry 4.0

5. CONCLUSIONS

This paper presented a quantitative analysis of the random communication delays in the closed-loop performance of a networked control system for temperature applications with random communication delays following α-Stable distribution with spiky nature and infinite variance. Four controllers were analyzed in the presence of random noise, two integer order PI, and two fractional-order PI controllers for different fractional orders α of the distribution. Obtained results showed that fractional-order controllers deal better with random communication delays, showing lower RMSE and RMS values for the time response and control action. Thus, we can say that fractional-order controllers are more robust in the presence of random communication delay noise with lower energy consumption. As future works, the experimental validation of these results, as well as the proposal of new tuning rules and methods for fractional-order controllers that can deal with α-Stable network delays are proposed.

ACKNOWLEDGEMENTS

The authors of this paper would like to thank Derek Hollenbeck of MESA Lab for his discussions.

REFERENCES

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA-The Instrumentation, Systems, and Automation Society.

Bhambhani, V., Chen, Y.Q., and Xue, D. (2008). Optimal fractional order proportional integral controller for varying time-delay systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 17(1 PART 1), 4910–4915.

Chen, Y.Q. (2010). Fractional calculus , delay dynamics and networked control systems. *Proceedings of the 3rd International Symposium on Resilient Control Systems.*

Li, T., Tang, X., Ge, J., and Fei, S. (2019). Event-based fault-tolerant control for networked control systems applied to aircraft engine system. *Information Sciences*. doi:https://doi.org/10.1016/j.ins.2019.10.039.

Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., and Feliu-Batlle, V. (2010). *Fractional-order Systems and Controls: Fundamentals and Applications*. Springer.

Mukhopadhyay, S., Yiding, H., and Chen, Y.Q. (2009). Fractional order networked control systems and random delay dynamics : A hardware-in-the-loop simulation study. *Proceedings of the 2009 American Control Conference*, 1418–1423.

Nikias, C.L. and Shao, M. (1995). *Signal Processing with Alpha-stable Distributions and Applications*. Wiley-Interscience, New York, NY, USA.

Oziablo, P., Viola, J., and Chen, Y. (2019). Networked Control System with Fractional Order Delay Dynamics. In *Proceedings of the 7th International Conference on Control, Mechatronics and Automation*, 1–6. Nov. 2019, TU Delft, Netherlands.

Viola, J., Radici, A., and Chen, Y. (2019). Low-Cost Real-Time Vision Platform for Spatial Temperature Control Research Education Developments. In *Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE2019*, 1–7. Aug. 2019, Anaheim, CA, USA.

Wu, C., Liu, J., Jing, X., Li, H., and Wu, L. (2017). Adaptive Fuzzy Control for Nonlinear Networked Control Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(8), 2420–2430. doi:10.1109/TSMC.2017.2678760.

Xue, D., Chen, Y., and Atherton, D.P. (2007). *Linear Feedback Control*. Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898718621.

Youness, S.F. and Lobusov, E.C. (2019). Networked Control for Active Suspension System. *Procedia Computer Science*, 150, 123–130. doi:https://doi.org/10.1016/j.procs.2019.02.025.