Interactive Whiteboards in Extended Reality for Advanced Control System Design

Aleksei Tepljakov
Department of Computer Systems, Tallinn University of Technology, Tallinn, Estonia
Email (corresponding author): aleksei.tepljakov at taltech.ee

***Abstract*—From chalkboards to whiteboards to interactive digital solutions, significant progress has been achieved in the past decades from the perspective of technological advances. Applications of interactive whiteboards are manifold and span across classrooms, business meetings and industrial practices alike. In this paper, we explore the convergence of several technological advances towards as single focal point—user experience with control system design in an extended reality (XR) environment. The main research question is related to understanding the aspects of using an interactive whiteboard in an XR environment with a specific manipulator that is represented by a specialized physical pen that appears to the user in extended reality while being tracked in the real environment. Furthermore, we also provide the description of the proposed testing system involving several software environments and describe the type of control design experiment that is performed. Some experimental results are presented and discussed.**

***Index Terms*—Fractional PID control; Control design; Software implementation; Extended Reality; Human-Machine Interaction**

I. Introduction

The rapid pace of technological development clearly necessitates introducing changes into the knowledge transfer model used in educational institutions [1], [2], [3]. At the same time, old teaching methods, including the classic chalkboard or whiteboard instruction style are still relevant, though now comprise only a part of the instructor's toolbox. From literature, it becomes apparent that adding interactivity to the classic whiteboard experience can enhance learning outcomes [4]. When it comes to instruction related to system modeling topics and automatic control design, the interactive demonstration of various processes appears to be highly relevant [5], [6], [7].

On the other hand, many additional possibilities that significantly complement the interactive instructions are afforded by the application of extended reality (XR) [8]. For example, using the concept of digital twins, one can design a virtual laboratory experience for control system design [9] which is especially helpful if the physical lab becomes inaccessible due to a global health emergency [10].

To make the digital whiteboard experience more natural means that a suitable physical manipulator representing a marker or pen is required. Recent studies clearly show that using hand and finger tracking for writing on an artificial whiteboard is not ideal [11] and that users prefer a manipulator with a precision grip for interactions with the whiteboard rather than the typical XR controller such that comes bundled with Oculus Rift or HTC Vive [12]. In this study, we evaluate the Logitech VR Ink Pilot Edition pen.

Next, taking into consideration a typical control systems curriculum in the universities, sufficient coverage of fractional calculus based modeling and control [13] as a generalization of classical modeling and control concepts is still missing from most study programs even though fractional-order proportional integral derivative (FOPID) controllers have been shown to offer benefits in industrial control loops [14] compared to ordinary PID controllers. Therefore, to improve on this aspect, some effort was exhibited in the past to create MATLAB-based interactive tools to provide a more clear view of the benefits of this generalization in control applications including the BODEFOPID tool presented in [15]. To implement fractional-order modeling and control in this effort, the FOMCON toolbox for MATLAB is used [16].

Thus, combining the above into a coherent contribution is the main purpose of the work described in the present paper. Looking forward, further convergence of technologies could result in a more involved user experience where interactive whiteboards, digital twins, Internet of Things, and machine learning are concerned [17]. The envisioned goal of the research of which this effort is but a part of is to achieve a truly immersive experience for the user—irrespective of the actual application, whether consumer or industrial—by implementing intelligent immersive virtual environments through advances in behavioral modeling, machine learning, and data science [18].

The main contribution of this paper lies at the intersection of many different technologies which makes this work rather interdisciplinary. We present the methodology for the design of the XR-based interactive whiteboard and discuss the relevant mathematical background related to the tuning of FOPID controllers. In addition, in this paper the JavaScript based control system toolbox [19] is introduced for the first time. Furthermore, we attempt to give a preliminary answer to the following research question:

Problem 1. Is it feasible to use a physical pen input device for creating an interactive whiteboard experience in extended reality for control system applications including (a) inserting mathematical models via written input; and (b) drawing a desired time domain response of a control loop?

The paper has the following structure. The mathematical tools used in the work are discussed in Sec. II. The proposed methodology and implementation is presented in Sec. III. An example interactive session in XR is presented in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. Mathematical Tools

Part of the novelty of this effort is that we apply fractional-order modeling and control ideas in the context of research and development of an interactive whiteboard. Therefore, the main mathematical tool used in this effort is fractional-order calculus. We reflect on the main ideas of this specific effort in what follows, but in the interest of space we do not provide an introduction to fractional calculus itself. For that, the reader can turn to excellent introductory works on this subject [13], [20].

As the purpose of the interactive whiteboard experience in this case is tuning of a FOPID controller, we consider the parallel form thereof that has the following transfer function

$$C(s) = K_p + K_i s_i^{-\lambda} + K_d s^{\mu}, \tag{1}$$

where K_p, K_i, and K_d are real numbers representing the proportional, integral, and differential gains, respectively, and $0 \leqslant \lambda \leqslant 2$ and $0 \leqslant \mu \leqslant 2$ are orders of the integral and differential components, respectively. The plant to be controlled is described by an arbitrary fractional-order transfer function model in the form

$$G_p(s) = \left(\sum_{j=0}^{M} b_j s^{\beta_j} \Bigg/ \sum_{i=0}^{N} a_i s^{\alpha_i} \right) \mathrm{e}^{-Ls}, \tag{2}$$

where generally $M \leqslant N$, and $\alpha_0 = \beta_0 = 0$ in which case $K = b_0/a_0$ is the static gain; L is the transport delay. For many industrial control tasks, the plant can be well approximated by a FO-FOPDT model—which is a simplified form of (2)—with the transfer function

$$G_f(s) = \frac{K\mathrm{e}^{-Ls}}{Ts^{\alpha} + 1}, \tag{3}$$

where K is the static gain, T is the pseudo time constant, L is the input-output delay, and α is the order of the fractional operator [21]. Further, for simulating the systems in the time domain, we consider the typical negative unity feedback control loop represented by a closed-loop model

$$G_c(s) = \frac{C(s)G_p(s)}{1 + C(s)G_p(s)}. \tag{4}$$

In this work, for the purpose of tuning the FOPID controller in (1), the Nelder-Mead Simplex Method is employed since it has proven to be very useful in industrial control loop tuning tasks [22]. The specific task is to find a least squares solution for the objective defined by

$$J(\theta) = \sum_{t=0}^{N} \left(r(t) - y(t) \right)^2, \tag{5}$$

where $y(t)$ is the step response of the closed loop in (4), $r(t)$ is the desired reference signal provided by the user, t is the sample index, N is the number of simulated samples, and θ is the FOPID controller parameter vector such that

$$\theta = \begin{bmatrix} K_p & K_i & \lambda & K_d & \mu \end{bmatrix}. \tag{6}$$

III. Methodology and Implementation

The complete technical solution is depicted in a diagram showing the major components involved in the implementation in Fig. 1. In what follows, we comment on the specifics of implementing the major components.

A. The XR Environment in Unreal Engine 4

Regarding the hardware stack choice for XR implementation, since VR Ink is used as the pen, one must also use the SteamVR stack, including the "lighthouse" infrared emitters for tracking as VR Ink requires them to operate. On the other hand, this also gives the possibility to use HTC Vive Tracker devices to implement certain features. HTC Vive Pro is used as the head-mounted display (HMD).

The extended reality environment is implemented in Unreal Engine [23] (UE), specifically version 4.23. The reason for this is that this is the last version of UE which supports the old SteamVR plugin architecture—VR Ink has its own source development kit (SDK) for UE based on SteamVR, and it only presently works without serious modifications up to UE version 4.23. Further development necessitates a complete rewrite of the plugin for VR Ink. The so called VR Ink Pawn is used to implement the XR part. When the user wears the HMD, the VR Ink is turned on, and the VR level in UE is started, the user is presented with the interactive whiteboard implemented as a *Browser Window* component in UE through a dedicated plugin. A *Widget Interaction* component is attached to the 3D model of the VR Ink and allows to interact with the web application shown—the whiteboard—in the browser window. The location of the whiteboard in the virtual environment can be adjusted by moving an HTC Vive tracker placed into the physical environment. The implementation of the web application itself is described in the next section.

Towards answering research question 1, we follow the path of studying empirical evidence through experimenting with the specific implementation of the virtual environment including the configuration of the VR Ink Pilot Edition device. To improve the user experience with the pen, several items have been implemented:

- When the pen tip comes sufficiently close to the white-board, a small sphere appears on the tip, signaling this way that the user can proceed with either writing an equation, or drawing a trajectory.
- Force feedback is sent to the pen whenever there is virtual friction between the whiteboard and the pen. This is a very important feedback mechanism.

The proposed answer to the research question is presented in the concluding section.

The key point of the implementation of XR is that nothing related to control design is computed in UE: the latter only serves as an graphical user interface to the web application which, in turn, communicates with MATLAB for performing all necessary computations.

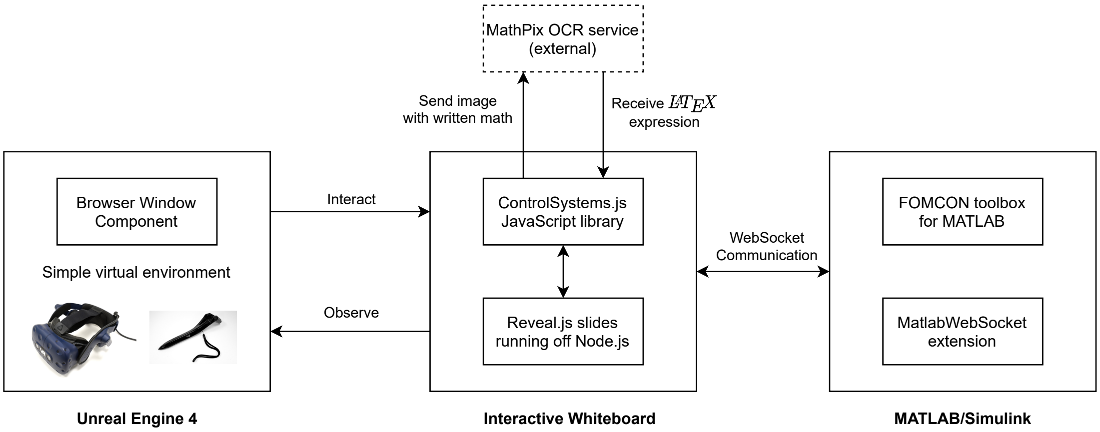

Fig. 1. The complete configuration of the implemented interactive whiteboard solution with three major components: the XR implementation, the web-based whiteboard, and the MATLAB/Simulink environment.

B. The Web-based Implementation of the Interactive Whiteboard

The interactive whiteboard implementation currently uses the Reveal.js slides framework [24] as basis and uses the ControlSystems.js framework [19] for several features, namely:

- Written input for equations;
- Drawing the desired time domain response;
- Plotting all necessary graphs showing time domain and frequency domain characteristics;
- Module for communicating with an online OCR service [25] for written input conversion into LaTeX code;
- Implementation of the LaTeX parser for processing the response from the external OCR service.

No features related to actual control system functions are used in this application, because the computational burden is offloaded to MATLAB.

It should be noted that all components can be run locally with the exception of the MathPix OCR service [25] for written math recognition which needs an Internet connection. Most other operations rely on the locally running MATLAB Websocket server described in the next section.

In total, the application comprises three slides with the following functions:

1) User writes the transfer function describing the plant (2) into the input box, and when ready presses the "parse" button. An image is captured from the written input and sent to MathPix OCR for processing (this requires an account with MathPix, and in case of extensive use of the API, a paid subscription). The LaTeX code is returned and parsed locally. The equation then appears in the top box. The user can proceed to the optimization phase.
2) The user is presented with a closed loop time domain response of a trivial control system where in (4) we set $C(s) = 1$. This response is obtained from MATLAB. Now, the user must draw a freehand trajectory describing the desired time domain step response. Some knowledge of possible trajectories is assumed in this case. The interactive plot allows to set only one point per vertical thus representing a true time domain response. Once the user is satisfied with the trajectory, he or she presses the "optimize button". The desired trajectory is then submitted to MATLAB and an optimization procedure is initiated in the latter. On each iteration step, MATLAB sends back the current response, so that the user could see how well the obtained response represents the chosen one. When optimization is completed, irrespective of whether it was successful or not, the next slide is shown to the user.
3) The final slide shows the parameters of the obtained FOPID controller as well as time domain and frequency domain characteristics and several parameters reflecting the quality of the control loop, such as the percentage overshoot, settling time, gain and phase margins. The user can decide whether the results are satisfactory or not and go back to either the first slide to change the process model, or to the second slide to draw another desired trajectory.

The slides—as seen from the perspective of the user in the XR environment—are also shown in the example session in Fig. 2.

C. MATLAB/Simulink Compute Server on Websocket

The implementation of the Websocket server itself is due to [26]. Other functions are implemented to support the control system design idea.

First, a connection must be established between the client (the whiteboard application) and MATLAB. A simple protocol is implemented to handle further transactions between application. The message exchange format is based on JSON encoded strings representing arrays, character strings, and dictionaries. Therefore, complex data can be transferred easily between different applications.

MATLAB is responsible for computing all required time domain and frequency domain characteristics of the open and closed loops. Let us discuss some specific operations that MATLAB performs after receiving corresponding requests via the Websocket.

First, when the user supplied transfer function is supplied to MATLAB and trivial closed loop simulation is requested, one needs to obtain a suitable simulation time range, on which all consequent optimization attempts will be based. Towards this, the following procedure is carried out:

1) An open loop simulation is performed on the user supplied plant modified as follows:

$$\tilde{G}_p(s) = \frac{1}{K} G_p(s), \tag{7}$$

where K is the static gain of the plant. MATLAB's auto-ranging facilities are used on the Oustaloup approximation [22] of (7) and a time vector $\mathbf{t}$ is obtained.

2) The closed loop in (4) with the plant $G_p(s)$ and $C(s) = 1$ is simulated with the time range obtained in the previous step. The pair of vectors $(\mathbf{t}, \mathbf{y})$ is returned to the calling application containing the auto-ranged time vector $\mathbf{t}$ and time domain response vector $\mathbf{y}$.

The optimization procedure is set up in a way that allows to send time domain simulation results from each iteration back to the client (i.e., the whiteboard application) so that the user could observe the evolving time domain response. The cost function is implemented according to (5). Only the integrator and differentiator order λ and μ in (6) are bounded to the range $\lambda, \mu \in [0, 2)$. The gains of the FOPID controller are left unbounded. Once optimization is finished, the specific time domain and frequency domain parameters are also computed and sent back to the client along with the optimal θ values.

IV. Example Session

Let us now consider an example session to illustrate the use of the obtained solution.

The experiment is performed using a personal computer having an Intel i9-9900K CPU, 64GB of RAM, and an NVidia RTX 3090 graphics adapter, though the latter is not a requirement, and a GTX 1080 would work just as well for this application. HTC Vive Pro HMD is used with VR Ink Pilot Edition pen.

The plant to be controlled is described by a transfer function

$$G(s) = \frac{1}{s^{0.5} + 3}. \tag{8}$$

The user writes manually equation (8) and it is recognized successfully. User then proceeds to the optimization slide.

The example session consists of two optimization attempts with various desired responses. All the steps for the first optimization attempt are shown in Fig. 2.

1) In the first attempt, the user desires to obtain a system with about 10% overshoot and with a settling time of 20 seconds. The resulting controller has the following gains and exponents:

$$K_p = -0.19681, K_i = 0.7677, \lambda = 1.193,$$
$$K_d = 0.02638, \mu = 1.037,$$

the resulting settling time is 21 seconds, percentage overshoot 10.21%, phase margin $\varphi_m = 63°$ at $\omega_c = 0.291$rad/s.

2) In the second attempt, the user wishes for a smooth transient, so draws the corresponding response having the settling time still at around 20 seconds. After optimization, the following controller is achieved:

$$K_p = 0.1067, K_i = 0.5081, \lambda = 1.013,$$
$$K_d = 0.009151, \mu = 0.9922,$$

the settling time is computed as 20.7 seconds, there is a negligible overshoot of about 0.2%, phase margin is now $\varphi_m = 85.8°$ at $\omega_c = 0.158$ rad/s.

The user can now decide which controller works best for the intended application.

V. Conclusions

In this work, we have presented a method for implementing an interactive whiteboard in extended reality geared towards solving advanced control design tasks assuming that the classical control curriculum is extended to include the study of fractional calculus in modeling and control. Specifically, we evaluated the use of a tracked physical pen—the VR Ink Pilot Edition from Logitech—in this scenario to answer the posited research question whether its use is feasible in this context. The proposed answer is yes, but with a few limitations. In this work, we implemented a whiteboard that has no physical counterpart in the real world. While its use is possible, the necessity to provide force feedback simulating the friction of the pen against the virtual whiteboard's surface results in a quick battery drain of the pen. Thus, to improve on this aspect, it is better to use the pad that comes together with VR Ink to provide a physical surface having a virtual counterpart in XR. The location of the pad in XR can be easily tracked using, e.g., and HTC Vive Tracker. Future work should include the improvement of the implementation, revision of the controller tuning strategy to include, e.g., frequency domain characteristics directly, and implementing a digital twin in XR that would visually demonstrate the control loop tuning performance—this also makes the choice to use an XR implementation instead of just using flat screen fully justified.

Acknowledgment

This article is partially based upon work from COST Action CA15225, a network supported by COST (European Cooperation in Science and Technology). The work of Aleksei Tepljakov is supported by the Estonian Research Council grant PRG658 and this specific effort was also supported by a teaching grant (with funding source SS19013) awarded to Aleksei by the School of Information Technologies, TalTech, in 2019. We would like to express our gratitude to Logitech for sending a VR Ink Pilot Edition device for evaluation in re:creation XR laboratory, Centre for Intelligent Systems, TalTech. Finally, we would like to thank HTC, who, through the VRFirst (https://vrfirst.com/) program, provided us with an HTC Vive Pro HMD.

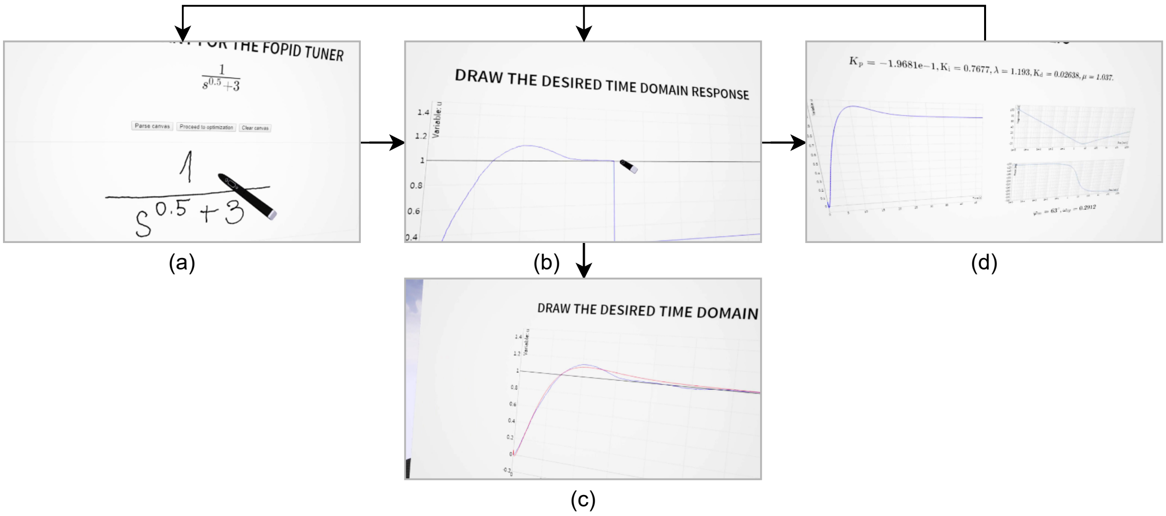

Fig. 2. The steps represented as screen grabs from VR in the example session with the interactive whiteboard: (a) The user writes the desired transfer function, presses a button, it is automatically recognized, the user can proceed to the next step. (b) The user draws a freehand trajectory of the desired setpoint and presses a button, the optimization process begins. (c) The user observes the progress of optimization in real time—the optimized response is shown as a red line. Once optimization completes, the user is taken to the next step. (d) The user is shown a page where the results of optimization are displayed, both time domain and frequency domain characteristics of the obtained control loop are shown. From this point, the user can either return to (a) or (b) and continue to work with controller design either for the same plant, or for another one.

REFERENCES

[1] M. Achacoso, "Evaluating technology and instruction: Literature review and recommendations," *Texas: The University of Texas*, vol. 3, 2003.

[2] M. Moran, M. Hawkes, and O. E. Gayar, "Tablet personal computer integration in higher education: Applying the unified theory of acceptance and use technology model to understand supporting factors," *Journal of Educational Computing Research*, vol. 42, no. 1, pp. 79–101, 2010.

[3] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, "Predicting student performance from LMS data: A comparison of 17 blended courses using moodle LMS," *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 17–29, 2017.

[4] W. D. Beeland, Jr., "Student engagement, visual learning, and technology: Can interactive whiteboards help?" *Action Research Exchange*, vol. 1, no. 1, 2002.

[5] W. H. Bares and J. C. Lester, "Realtime generation of customized 3D animated explanations for knowledge-based learning environments," in *AAAI/IAAI*, 1997.

[6] R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido, "Development of a Web-based Control Laboratory for Automation Technicians: The Three-Tank System," *IEEE Transactions on Education*, vol. 51, no. 1, pp. 35–44, 2008.

[7] M. Roussou and M. Slater, "Comparison of the effect of interactive versus passive virtual reality learning activities in evoking and sustaining conceptual change," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2017.

[8] A. Köse, "Towards dynamic modeling in immersive environments with assessment of user experiences," Ph.D. dissertation, Tallinn University of Technology, 2020.

[9] S. Jersov and A. Tepljakov, "Digital twins in extended reality for control system applications," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2020.

[10] S. Ray and S. Srivastava, "Virtualization of science education: a lesson from the COVID-19 pandemic," *Journal of Proteins and Proteomics*, vol. 11, no. 2, pp. 77–80, 2020.

[11] K. Pietroszek and C. C. Lin, "UniVResity: Face-to-face class participation for remote students using virtual reality," in *25th ACM Symposium on Virtual Reality Software and Technology*. ACM, 2019.

[12] A. U. Batmaz, A. K. Mutasim, and W. Stuerzlinger, "Precision vs. power grip: A comparison of pen grip styles for selection in virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020.

[13] C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls: Fundamentals and Applications*, ser. Advances in Industrial Control. Springer Verlag, 2010.

[14] A. Tepljakov, B. B. Alagoz, C. Yeroglu, E. A. Gonzalez, S. H. Hosseinnia, E. Petlenkov, A. Ates, and M. Cech, "Towards industrialization of fopid controllers: A survey on milestones of fractional-order control and pathways for future developments," *IEEE Access*, vol. 9, pp. 21 016–21 042, 2021. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2021.3055117

[15] A. Tepljakov, E. Petlenkov, E. A. Gonzalez, and I. Petras, "Design of a MATLAB-based teaching tool in introductory fractional-order systems and controls," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017.

[16] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON toolbox for modeling, design and implementation of fractional-order control systems," in *Applications in Control*, I. Petráš, Ed. De Gruyter, 2019, pp. 211–236.

[17] M. J. Kaur, V. P. Mishra, and P. Maheshwari, "The convergence of digital twin, IoT, and machine learning: Transforming data into action," in *Internet of Things*. Springer International Publishing, 2019, pp. 3–17.

[18] A. Tepljakov, "Towards Intelligent Immersive Virtual Environments," Keynote talk on EuroVR 2019 conference, 2019, slides for Keynote are available, retrieved May 2021. [Online]. Available: https://recreation.ee/at/eurovr2019/

[19] A. Tepljakov. (2019) Control systems toolbox for JavaScript. [Retrieved 01.12.2019]. [Online]. Available: https://github.com/extall/control-systems-js

[20] Y. Q. Chen, I. Petráš, and D. Xue, "Fractional order control - a tutorial," in *Proc. ACC '09. American Control Conference*, 2009, pp. 1397–1411.

[21] B. B. Alagoz, A. Tepljakov, A. Ates, E. Petlenkov, and C. Yeroglu, "Time-domain identification of one noninteger order plus time delay models from step response measurements," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 10, no. 01, p. 1941011, 2019.

[22] A. Tepljakov, *Fractional-order modeling and control of dynamic systems*. Springer-Verlag GmbH, 2017.

[23] Epic Games, "Unreal Engine," Available online: https://www.unrealengine.com/, 2020, retrieved 06.03.2020. [Online]. Available: https://www.unrealengine.com/

[24] H. E. Hattab. (2019) Reveal.js: The HTML Presentation Framework. [Retrieved 16.01.2019]. [Online]. Available: https://revealjs.com/

[25] Mathpix company, "Mathpix Snip: Scanner app for math and science," 2021, electronic service. [Online]. Available: https://mathpix.com/

[26] J. Béjanin, "MatlabWebSocket," GitHub repository, 2021. [Online]. Available: https://github.com/jebej/MatlabWebSocket