

Evaluation of GPTs for Control Engineering Education: Towards Artificial General Intelligence

P. B. de Moura Oliveira* and Damir Vrančić**

* INESC-TEC, UTAD-Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal

** J. Stefan Institute, Ljubljana, Slovenia

(e-mail: oliveira@utad.pt ; damir.vrancic@ijs.si)

Abstract: Recently introduced Generalized Pre-trained Transformers (GPT) and conversational chatbots such as ChatGPT are causing deep society transformations. The incorporation of these Artificial Intelligence technologies can be beneficial in multiple science and development areas including Control Engineering. The evaluation of GPTs within Control Engineering Education and PID control is addressed in this work. Different types of interactions with GPTs are evaluated and the use of a personalized GPT for PID tuning explored.

Keywords: GPT, ChatGPT, Control Engineering Education, PID Control.

1. INTRODUCTION

At present, there is an observable escalation in the influence exerted by Artificial Intelligence (AI) methodologies across various facets of human existence. The Singularity advent, as the moment as the technology progress will enable AI to surpass human intelligence, may be closer than forecasted by Kurzweil (2005). One of the most actual research topics is Artificial General Intelligence (AGI) (see Fig. 1 motivation figure), which may be an important step towards the Singularity. A company that currently is a major player in developing AGI techniques is OpenAI (2023). The work undertaken by OpenAI on Generalized Pre-trained Transformers (GPT) led to a disruptive breakthrough with the public release of ChatGPT (2023) on November 15, 2022. As this paper is being written, one year after ChatGPT-3 was released, its impact and repercussions are undeniable, reflected also by a substantial number of related scientific publications (e.g. Mogavi *et al.*, 2024; Javaida *et al.*, 2023, Neumann *et al.*, 2023, Moura Oliveira, 2023). Consequently, in most scientific domains, it can be asserted that a distinct 'before and after' era has been delineated by the advent of ChatGPT. This is also true for Control Engineering, motivating GPTs exploration in this area is reported in this paper.

The first generative pretrained model (GPT-1) development presented a breakthrough (in 2018) in Natural Language Processing (NLP) and understanding. Large Language Models (LLM) are pretrained using Machine Learning techniques such as Deep Learning, Unsupervised Learning, Transfer Learning and Reinforcement Learning. Reinforcement Learning has been employed in later stages for fine-tuning and optimizing the model's responses using Proximal Policy Optimization (PPO) methods (OpenAI, 2023). The use of larger data-sets to train the models, with more parameters and more sophisticated algorithms, resulted in the progressive development and release of GPT-2 (2019), GPT-3 (2020) and more recently GPT-4 (2023). The different versions improved text generation and

understanding. GPT-3 presented problem solving involving some mathematics and programming coding capabilities. An important GPT-3 feature is the ability to provide answers based on Zero-Shot or Few-Shot Learning, meaning that it needs zero examples or few examples to perform a specified task. While GPT-5 is not released, the version used in this work is GPT-4 with DALL.E 3 (2023), which has Web browsing and data analysis capabilities. Note that GPT 3.5 is freely available by OpenAI. However, GPT-4 requires the ChaGPT Plus payed subscription (20USD a month) and currently its use is limited to 40 messages every 3 hours.



Fig. 1. A conceptual art piece integrating Artificial General Intelligence (AGI), ChatGPT and Industry 4.0 themes generated by Dall-E 3 (2023).

Recently OpenAI (2023) enabled every ChatGPT Plus subscriber to make its own personalized or custom GPT. These are now referenced as GPTs. So, this new option can also be easily deployed for control engineering.

The main research questions this article tries to provide answers to are the following:

1. How can GPT-4 be used for teaching/learning/training control engineering? We live in an immediate world where time is of the essence, as technology changes at a rapid pace. Thus, any help we can obtain to learn (and work) more efficiently is welcome. As will be shown, ChatGPT can act as a kind of co-pilot in various useful activities. This is important both in Academia and in Industry.
2. What type of prompts are more appropriate for control engineering? Exploration examples will be provided with prompts involving the following types: *i)* Textual based; *ii)* Image based *iii)* Code based, and *iv)* File based.

The remainder of the paper is organized as follows: section 2 presents general control engineering and industrial automation ChatGPT-4 use cases. Section 3 presents a GPT called PID Tuning Advisor, elaborated by this paper authors. Section 4 presents different evaluation examples regarding possible use cases of GPTs for PID control and finally section 5 concludes the paper and outlines further work.

2. GPT-4 EVALUATION FOR GENERAL CONTROL ENGINEERING

This section considers some general use-cases when GPT-4 acts as users tutor. The first example, depicted in Figure 2, uses a prompt featuring a simple Ladder Diagram (LD) figure with a timer. GPT is asked to explain the diagram.

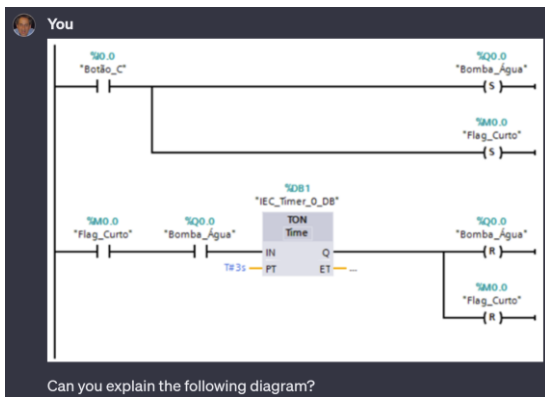


Fig. 2. Simple Ladder Diagram with a TON Delay timer.

The answer provided by GPT-4 is presented in Figure 3. As can be observed, based on the Portuguese to English translation of some labels, GPT-4 appears to provide a good explanation concerning the different diagram elements. However, there are some confusing elements such as the blue highlighted part in Figure 3. Even after adding the complementary information to consider the LD as complete, it failed to explain that the Water Pump, once activated, is deactivated 3 seconds later.

The second example uses a prompt including a GRAFCET model extracted from (eGRAFCET, 2023), depicted in Figure 4, in which the GPT is asked to explain the diagram. A GPT-4 answer is provided in Figure 5. As it can be seen from Figure 5, there are some image labels correctly interpreted by GPT-4

relating the diagram with the cars example. The response provided is sensible, despite the user not providing information that specifies the role played by the PLC outputs controlling the motors. So, the information provided by GPT-4 could help user/student in model understanding.

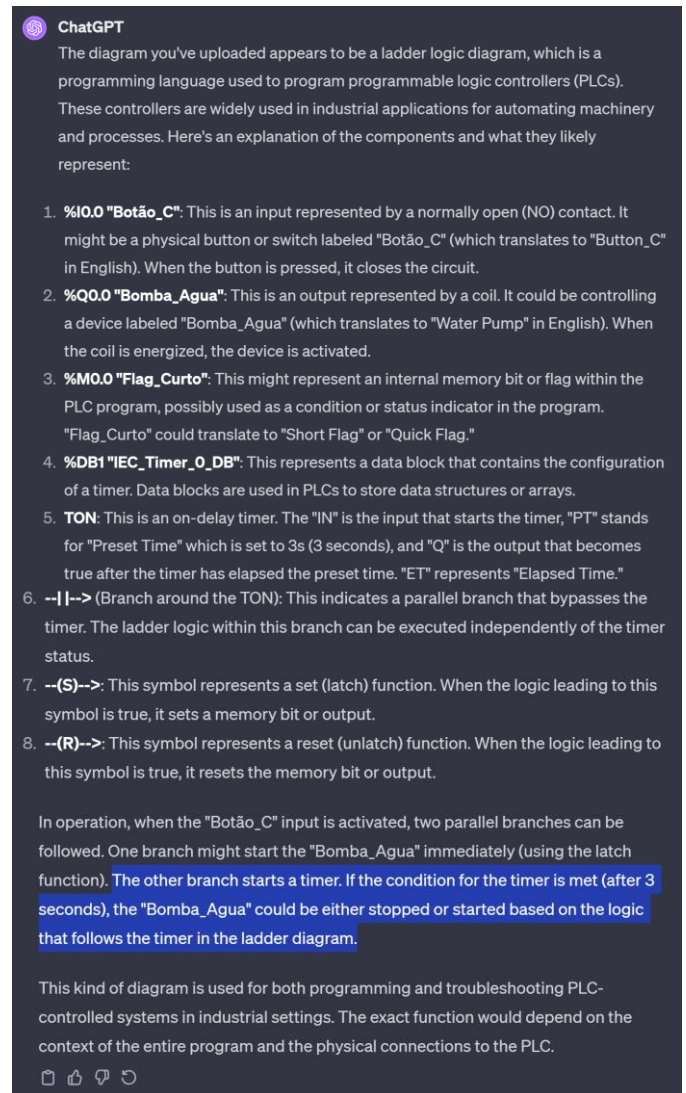


Fig. 3. PrintScreen of an answer provided to GPT-4 regarding the explanation of LD presented in Figure 2.

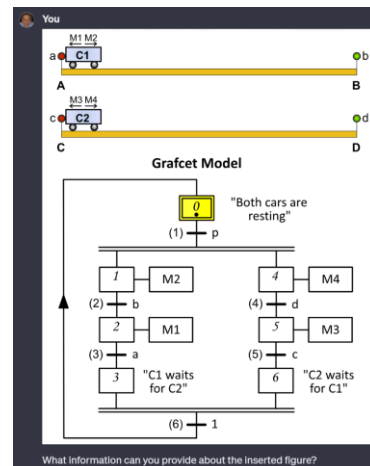


Fig. 4. eGRAFCET example regarding the control of two cars.

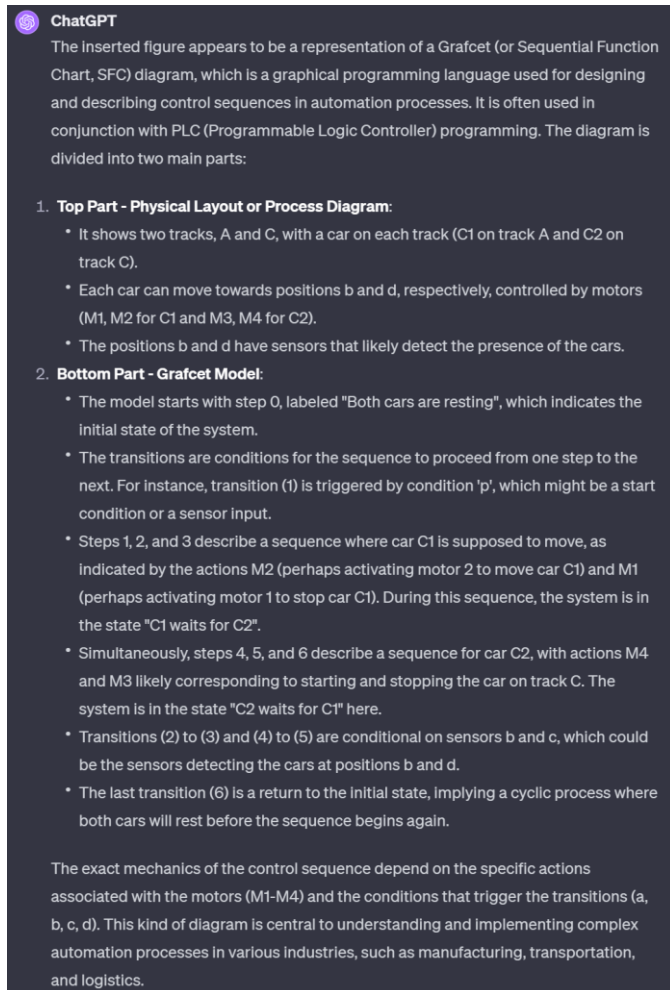


Fig. 5. GPT-4 answer regarding the example provide in Figure 4.

The example provided in Figures 6 and 7 tries to evaluate if GPT-4 can provide answers about a system based on an open-loop step response. GPT is also asked to provide the first overshoot estimate. The GPT response, presented in Figure 8, provide not only a good estimate for the first overshoot but also a detailed explanation of how this value was evaluated.

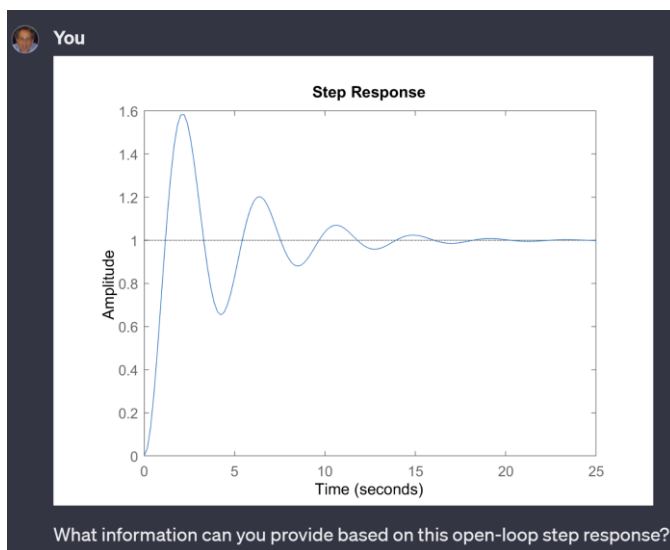


Fig. 6. Second-order system step-response example (part I).

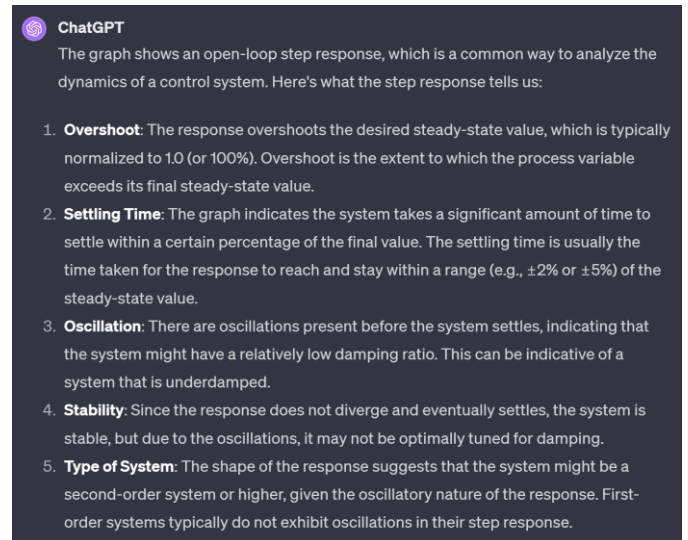


Fig. 7. Second-order system step-response example: part II.

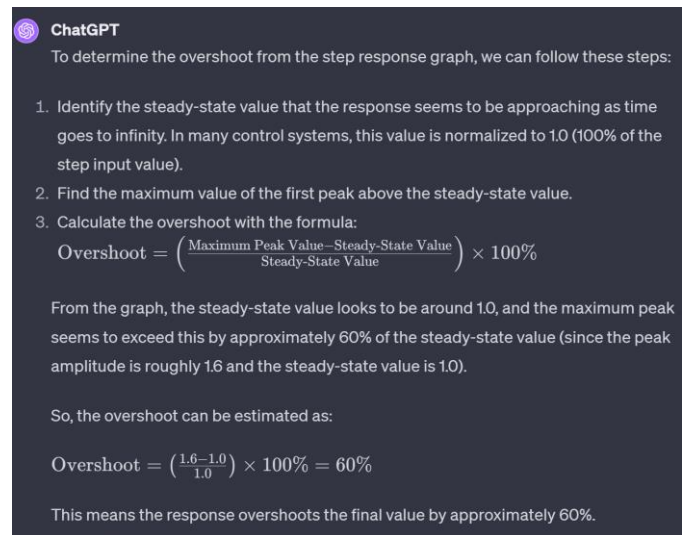


Fig. 8. GPT-4 answer regarding a first overshoot estimation based on the step response provided.

3. GPT PID TUNING ADVISOR

Here a custom specific GPT called "PID Tuning Advisor" (see Fig. 9), designed with some simple specifications related to the PID topic is evaluated using different prompting approaches.

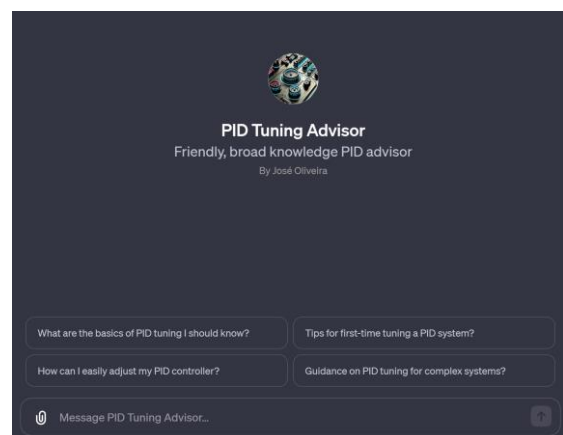


Fig. 9. GPT PID Tuning Advisor.

This customGPT was made with the GPT Builder (OpenAI, 2023). This tool guides the user to tailor its GPT, using a step-by-step approach. This requires defining the GPT general objectives, conversation starters examples, conversational style, etc. The GPT builder configuration menu, provides the user with the options to personalize its GPT to accept additional knowledge files, to perform web browsing, to generate image using Dall.E and use the Code Interpreter. All the mentioned options were used for creating the PID Tuning Advisor. The GPT builder also enables the user to use third party APIs.

In the next example, GPT has been asked to provide a Matlab code to simulate a PID controller in the digital domain, specifying the type of algorithm to be used. The answer is provided in Figures 10, 11 and 12.

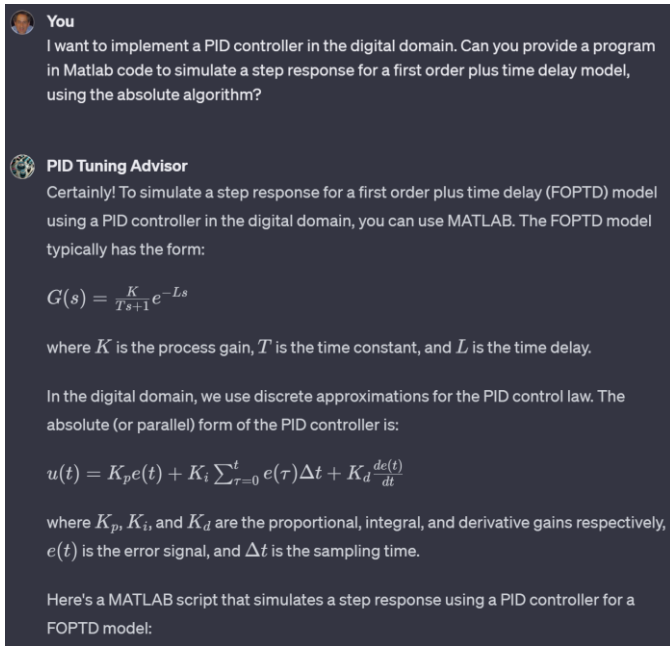


Fig. 10. Question and part of the GPT-4 answer regarding the simulation of a digital PID loop.

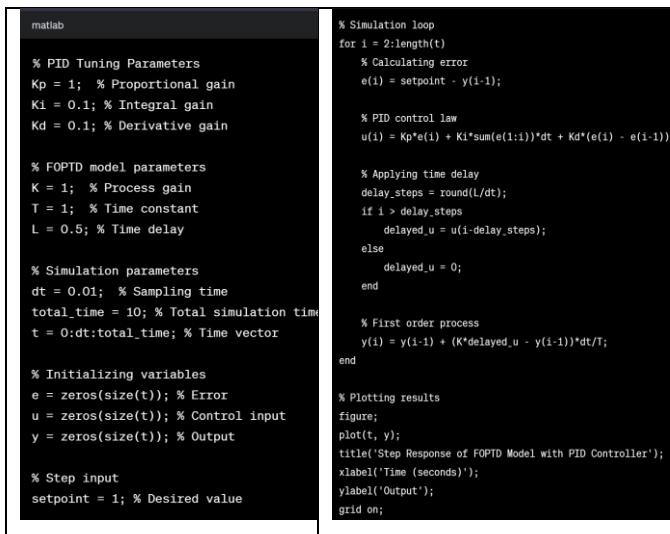


Fig. 11. Matlab Code generated by Chat-GPT.

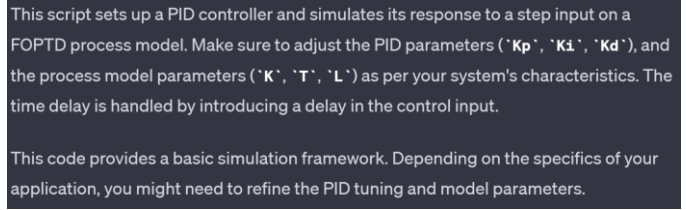


Fig. 12. Remaining of the answer presented in Figure 11.

Testing the provided code (Figure 11) in Matlab yielded the step response shown in the left plot of Figure 13, demonstrating that it functioned correctly without any modifications. GPT-4 is asked if it can run the provided script, and it managed to provide a replica of the Matlab plot (see right side of Figure 13).

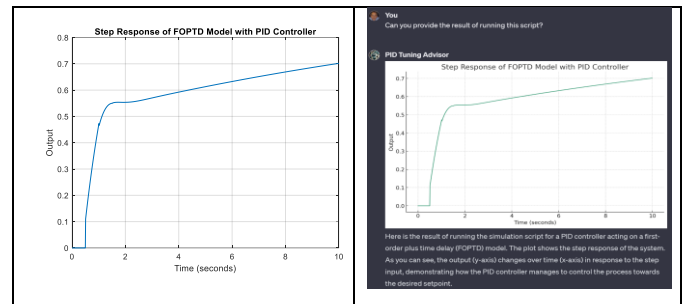


Fig. 13. Matlab Code generated by Chat-GPT tested in Matlab (left plot) and replicated by GPT-4 (right plot).

Asking GPT to analyse data inserted using a CSV file with comma-separated values is the objective of the example depicted in Figure 14. Besides the label for each file column, no additional information is provided to GPT. The open-loop test was obtained using a TLab Arduino kit for a Control Engineering experiment (for more details see Moura Oliveira and Hedregreen, 2019).

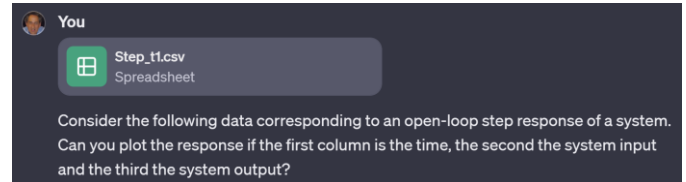


Fig. 14. Prompt using an open-loop step response data file.

The result of the prompt submitted in Figure 14 is presented in Figure 15. As it can be observed, after an analysis error (frequent at this ChatGPT development stage), the GPT-4 is able to correctly extract the data series from the supplied file and elaborate the corresponding plot.

The following prompts objective is to evaluate if the GPT can perform some data analysis. In Figure 16, GPT is asked to estimate the first order plus time delay (FOPTD) model from the open-loop step response. The answer was correct both in terms of the explanation provided regarding the method used, as well as in terms of the model parameters evaluation.

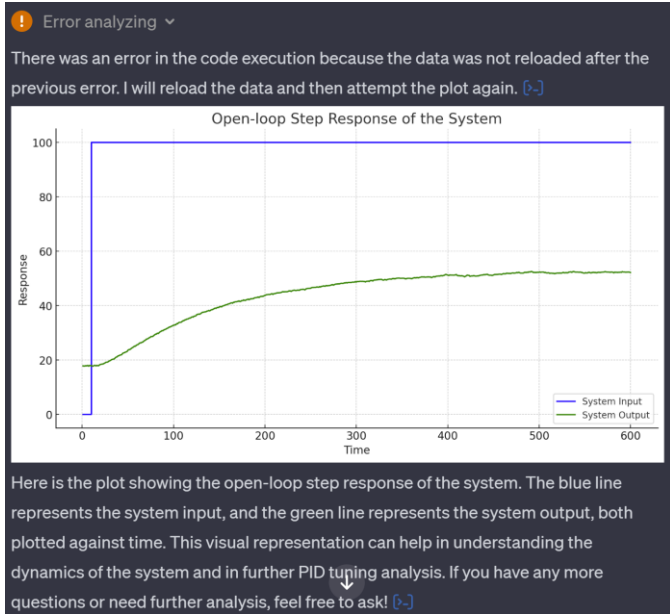


Fig. 15. GPT-4 response, plotting the open-loop response extracted from a data file.

You
I want to estimate a first order plus time delay model parameters from this open-loop test. Can you provide those values, and explain the method used?

PID Tuning Advisor
Estimating the parameters of a first-order plus time delay (FOPTD) model from open-loop test data involves identifying three key parameters:

- Process Gain (K):** The ratio of the change in output to the change in input.
- Time Constant (τ):** The time it takes for the system to reach approximately 63.2% of its final value after a step change.
- Dead Time (θ):** The delay between the input change and when the output starts to respond.

The standard FOPTD model is represented as:

$$G(s) = \frac{K}{\tau s + 1} e^{-\theta s}$$

To estimate these parameters from the provided data, we typically use graphical methods. Here's a simplified approach:

- Process Gain (K):** Determine the steady-state values of input and output. Calculate $K = \frac{\Delta \text{Output}}{\Delta \text{Input}}$.
- Dead Time (θ):** Identify the point where the input changes and find the corresponding time. Then find the time when the output starts to respond significantly. The difference is the dead time.
- Time Constant (τ):** After the dead time, find the time at which the output reaches 63.2% of its total change (from the point where it starts responding). The difference between this time and the dead time is the time constant.

Let's apply these steps to your data to estimate the FOPTD model parameters.

Based on the open-loop step response data provided, the estimated parameters for the First-Order Plus Time Delay (FOPTD) model are:

- Process Gain (K):** Approximately 0.348
- Dead Time (θ):** Approximately 10.142 seconds
- Time Constant (τ):** Approximately 143.368 seconds

Fig. 16. Results of a FOPTD model estimation.

The following results were obtained after a series of interactions with the GPT-4 to evaluate its data analysis and

simulation capabilities. Due to space limitations it is not possible to present all the successive prompts and plots herein. After GPT being asked to determine the initial process output steady-state value it was asked to simulate the step response with the estimated FOPTD model and overlap the plot with the real process data. The outcome result is illustrated in Figure 17. After asking GPT to evaluate the integral of the square error between the estimated model and the data it was asked to run an optimizer to minimize this error to improve the model fitting. It successfully presented an optimized model corresponding to $K= 0.351$, $\tau=130.95$ seconds and $\theta= 16.293$ seconds (*dc* gain, time constant and time delay respectively).

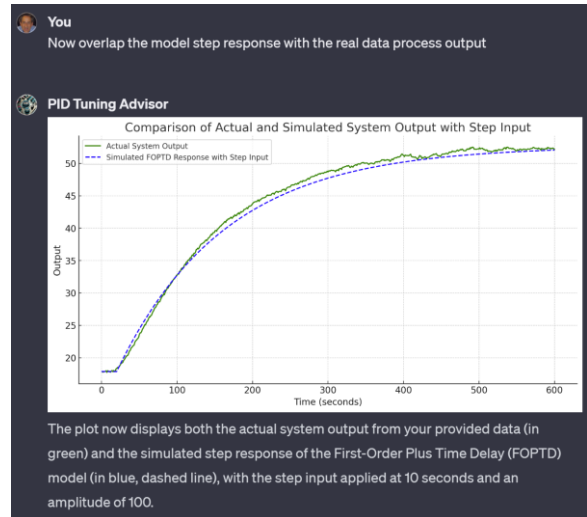


Fig. 17. Open-loop plots obtained with GPT-4.

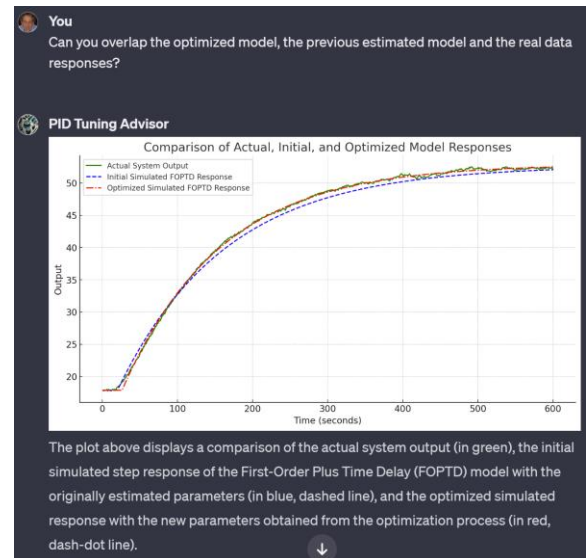


Fig. 18. Comparison between an optimized FOPTD model with the previous FOPTD model and real data.

The overlapped responses of FOPTD models and real data are presented in Figure 18. The results presented converged to the values obtained in a similar experience reported in (Moura Oliveira and Hedregreen, 2019) in which a Particle Swarm Optimization algorithm was used as the optimizer ($K= 0.35$, $\tau=131$ seconds and $\theta= 16.3$ seconds).

The next prompt (see Figure 19) asked GPT to use the classical Cohen-Coon tuning rules to evaluate a PID controller gains,

using the optimized FOPTD model parameters. Here the GPT got the wrong expressions for the integrative time constant, T_i , and derivative time constant, T_d , as the multiplying factor is θ and not τ . Note that this type of error is difficult to detect. After some interactions the GPT-4 corrected both the tuning formulas and evaluated the PID gains: $K_p=31.28$, $T_i=38.12$ seconds and $T_d=5.79$ seconds.

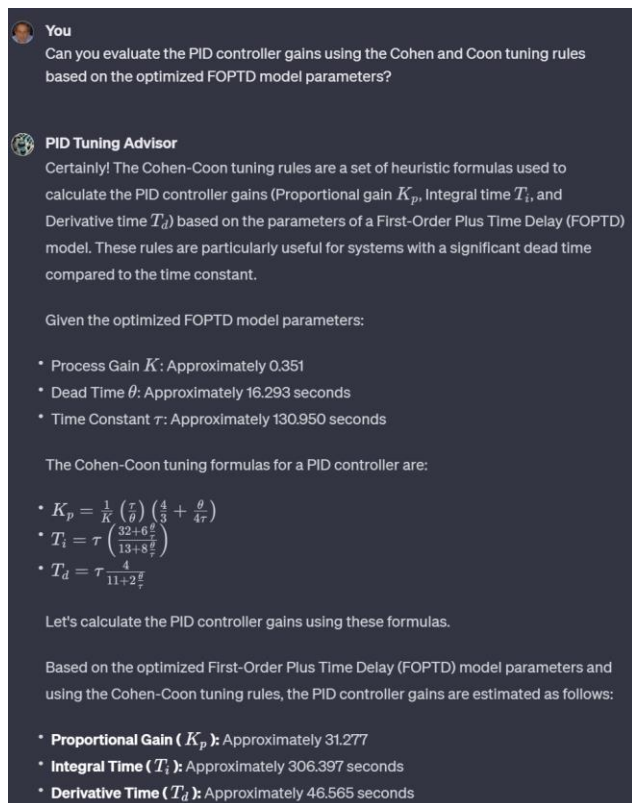


Fig. 19: GPT Cohen and Coon (wrong) tuning rules evaluation.

5. CONCLUSION

GPT-4 was explored in the context of control engineering by evaluating the results provided to a series of different types of prompts. When OpenAI enabled building a custom GPTs for specific purposes, it opened a wide range of new possibilities. A GPT entitled PID tuner advisor was made using the standard step-by-step procedure available in ChatGPT. This GPT was used to provide answers to some general model estimation and controller design issues.

Considering that GPTs does not understand the real meaning of the generated sentences, in the way humans do, “just” predicts the next word, the overall quality of the responses provided is very high. The obtained results showed that quality questions usually generate quality answers. It is clear from the results obtained, that if one knows how to make a good question, it will obtain a good answer. However, we cannot yet trust all the answers provided. GPT hallucinations may occur, meaning that it can provide wrong answers. The quality of the answers provided can easily elude an user. So, an important question remains: how can an unexperienced user be certain about the correctness of the provided answers? The user certainly must double check the information received using other credible information sources. This issue is associated with another problem which is the learner promotion of critical thinking

skills. In the other hand, for experienced users, GPTs can be used to facilitate performing different tasks, such as involving programming languages and image interpretation. It is important to state that the GPT-4 payed subscription is a disadvantage, as may users (e.g. students) can not afford it. Also, the current maximum 40 messages every 3 hours use may be restrictive in some cases. Future development can pass by including external actions for increasing the personalized GPT capabilities. Future work is necessary to consolidate and improve the PID Tuner Advisor as well as test it in Control Engineering Education courses, collecting and analysing students’ feedback.

ACKNOWLEDGEMENTS

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.
DOI10.54499/LA/P/0063/2020
<https://doi.org/10.54499/LA/P/0063/2020>

REFERENCES

- ChatGPT (2023), <https://chat.openai.com/chat>, retrieved in 12-12-2023.
- Dall-E 3 (2023), <https://openai.com/dall-e-3>, retrieved in 12-12-2023.
- Javaida M., Haleema A. and Singhb R. P. (2023), A study on ChatGPT for Industry 4.0: Background, potentials, challenges, and eventualities, *Journal of Economy and Technology 1*, pp. 127–143.
- Koubaa A., Qureshi B., Ammar A., Khan Z., Boulila W. and Ghouti L. (2023), Humans are still better than ChatGPT: Case of the IEEEExtreme competition, *Heliyon 9*, e21624, <https://doi.org/10.1016/j.heliyon.2023.e21624>
- Kurzweil R., (2005), *The Singularity Is Near: When Humans Transcend Biology*, Viking, ISBN: 978-0-670-03384-31.
- Mogavi R. H., Deng C., Kim J. J., Zhou P., Kwon Y. D., Metwally A. H. S., Tlili A., Bassanelli B., Bucchiarone A., Gujar S., Nacke L. E. and Hui P., (2024), ChatGPT in education: A blessing or a curse? A qualitative study exploring early adopters’ utilization and perceptions *Computers in Human Behavior: Artificial Humans 2*, 100027, pp. 20.
- Moura Oliveira P. and Hedengren J. D., (2019), An APMonitor Temperature Lab PID Control Experiment for Undergraduate Students, 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 790-797.
- Moura Oliveira P., (2023), ChatGPT: Assessing Impacts and Perspectives in Engineering Education using a Genetic Algorithms Case Study, 2023 6th Experiment@ International Conference (expat23) June 5-7, University of Évora, Portugal, IEEE, pp. 181-186.
- Neumann M., Rauschenberger M. and, Schon E.M. (2023), “We Need to Talk About ChatGPT”: The Future of AI and Higher Education, 2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG), IEEE, pp. 29-32.
- OpenAI (2023), <https://openai.com/>, retrieved in 12-12-2023.