**Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024**

Multi-objective Parameter Tuning and Performance Comparison of PI and FOPI Controllers for a Children's Anesthesia Administration System [⋆]

Paul Arpi [*] **Gilberto Reynoso-Meza** [*,**]

[*] *Department of Industrial and Systems Engineering (PPGEPS), Pontificia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição 1155, Curitiba, Brazil, (e-mail: paul.coellar@pucpr.edu.br)*
[**] *Control Systems Optimization Laboratory (LOSC), Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição 1155, Curitiba, Brazil,(e-mail: g.reynosomeza@pucpr.br)*

Abstract: This work compares the performance of PI and FOPI controllers for a children's anesthesia administration system applied to a FOPDT model based on data of 46 patients. Parameter tuning is performed through a multi-objective differential evolution optimization algorithm that includes a spherical pruning mechanism to obtain a good distribution along the Pareto front. The design objectives were the median and interquartile range of the integral of the absolute value of error (IAE). The Pareto front of the solutions of FOPI dominated the ones of the PI controller, although the PI Pareto front had a less disperse distribution. The best option of each Pareto set was determined by testing each option with the 46 patients and looking for the case that complies with the most or all problem constraints. The results of the tests showed that PI had a stable behavior with all patients, while FOPI showed an undesirable behavior with 5 out of 46 patients that did not comply with the constraints of the problem. Despite FOPI had a time response 37.33% faster, PI had better control characteristics and complied with all the design constraints in all patients, hence a control scheme such as FOPI is not justified.

Keywords: Multi-objective, PI, FOPI, Controller tuning, Propofol, Anesthesia.

1. INTRODUCTION

Control theory can be found in a wide variety of fields of engineering and science, having automatic control as an essential tool (Ogata, 2010). A popular technique of automatic control is the use of a feedback signal to generate a closed loop and tell the controller the actual state of the plant for it to perform the control action (Díaz-Rodríguez et al., 2019). This approach is used with PID controllers, which use a proportional, integral and derivative action to perform corrections. The proportional action reduces the error between the setpoint and the value measured in the feedback loop from the output of the process. The integral action reads the feedback signal over time and reduces the offset from the measured feedback and the derivative action reads the rate of change in the feedback signal and corrects the value if there are unusual variations (Borase et al., 2021). This structure is preferred due to its ease of development and implementation. It is extensively found in applications of power electronics, pneumatics, motion control, process control and others (Åström and Hägglund, 2001). In order to improve the characteristics of PI controllers, Fractional Order controllers were proposed by generalizing the order of the integral control action through the addition of a fractional integral of order λ in the controller equation(Seo and Choi, 2019). Its effectiveness in comparison to the integer order PI was tested by Pullaguram et al. (2018) within the implementation of an autonomous microgrid VSC system, Dabiri et al. (2018) used it in the design of controllers for linear dynamic systems, Edet and Katebi (2018) applied it in process control systems and Xu et al. (2022) used it in the optimal control of a pumped storage unit. As any PID based control structure, its efficiency depends in the correct tuning of its parameters to the problem at hand. Theorists of this area have shown greater interest in automatic tuning, and here is where the criteria and background knowledge of the control engineer becomes vital Su et al. (2020). Although it provides a good improvement in comparison to classical approaches, it requires a well tuned base to develop, and benchmarks for tuning such as Ziegler-Nichols have shown poor results in many cases (Kumar et al., 2018; Åström and Hägglund, 2001). As the control architecture increases in complexity, more variables are required to be tuned correctly. This can be achieved in various ways that range from automatic tuning, to empiric methods based on the knowledge of the plant and the experience in the field, but all of them result time consuming both in implementation and analysis. This is where multi-objective optimization becomes important as it allows the designer to have a set of viable options that consider performance metrics for

[⋆] This work was partially funded by the National Council of Scientific and Technological Development of Brazil (CNPq) through the grants PQ-2/310195/2022-5 & Universal/408164/2021-2.

their election, giving more relevant information for decision making and the election of the controller variables. It is important to remember that the set of possible variables at the end of the optimization process have different trade-offs and advantages that must be analyzed before choosing one option (Marques and Reynoso-Meza, 2020).

This work seeks to determine if the use of a FOPI controller improves significantly the administration of anesthesia and the control characteristics presented by a PI controller and would justify its added complexity on its deploy, taking into account two performance metrics and analyzing their effectiveness. The structure of this work is the following: Section 2 provides the theoretical information related with PI and FOPI controllers, Section 3 describes the multi-objective algorithm used for the optimization of control variables, Section 4 describes the model and variables involved, Section 5 explains the optimization process and testing, Section 6 shows the results of the tests with both controllers and Sections 6 and 7 discusses the findings and conclusions of this paper.

2. THEORETICAL BACKGROUND

This section deals with the necessary theoretical concepts in order to fully understand the proposed comparison, mainly focused in the definition of the controllers.

2.1 FOPI Controller

A fractional-order proportional integral takes the regular proportional integral control action and adds a fractional operator in the integral block of the controller. It has been found that it improves the stability, transient time, and overall precision (Zaid et al., 2023). Examples both from medical and industrial areas were presented by (Wang et al., 2023) and (Reddy and Devabhaktuni, 2022). A comprehensive expression of the transfer function of this controller, in the Laplace form, can be found in Equation 1.

$$G_c = K_p + \frac{1}{T_i s^{\lambda}} \tag{1}$$

Where K_p and K_i are the gains for the proportional and integral action, and λ is the fractional order operator that can take values between 0 and 2.

3. MULTI-OBJECTIVE OPTIMIZATION ALGORITHM

A more complex control architecture has more variables to tune correctly. This represents a challenge taking into account that there is not one perfect combination of values that will work for any case. Moreover, if the performance needs to be assessed, it is very common to use two or more metrics in order to have a good representation of the effectiveness of the control action, not only measuring the usual time domain responses. Multi-objective optimization (MOO) seeks for a group of viable options to implement, considering two or more performance metrics that may be in conflict. It is important to understand that MOO presents a set of options with different trade-offs between the design objectives, giving the designer the last word on deciding the best option, or options, for the analyzed case.

3.1 Pareto Front

It is a set of optimal solutions to a problem with two or more design objectives that may be in conflict. Each element has different characteristics regarding the performance of each design objective, presented in the objective vector $J(\theta)$. This individual trade-off has to be analyzed to select the optimal option to the problem (Meza et al., 2017). Figure 1 shows an example of Pareto Front.

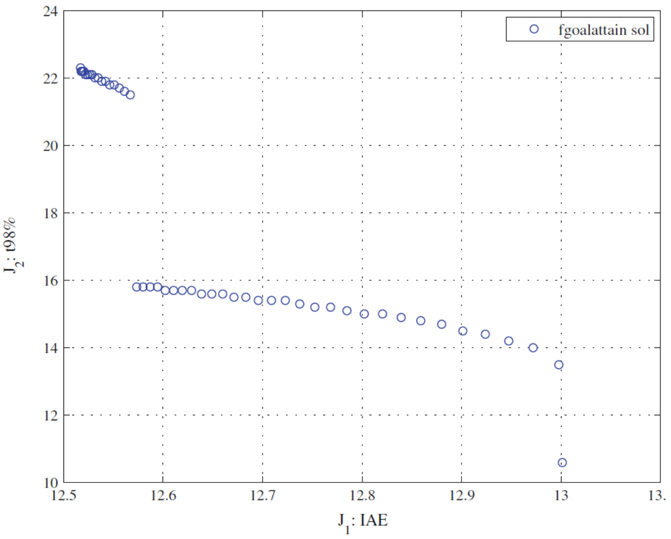

Fig. 1. Pareto Front approximation. (Meza et al., 2017).

3.2 Pareto Dominance

The concept of dominance defines a better set of optimal options for the analyzed design objectives, which form the decision vector θ.

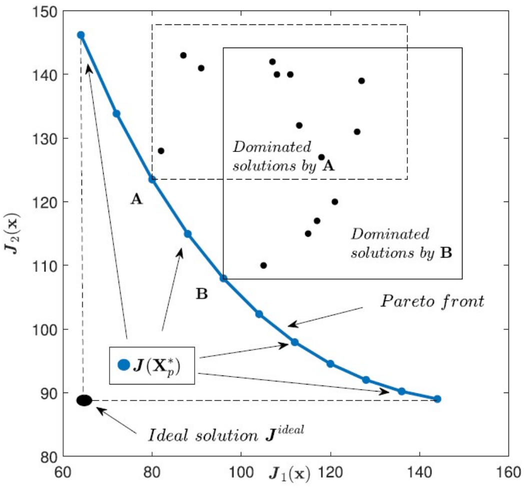

Fig. 2. Pareto dominance between two Pareto sets. (Meza et al., 2017).

When comparing Pareto sets, it is possible to determine which has better performance when a decision vector θ^1 dominates another vector θ^2, denoted as $\theta^1 \preceq \theta^2$, if the objective vector $J\left(\theta^1\right)$ is better than $J\left(\theta^2\right)$ in at least one objective, and is not worse than $J\left(\theta^2\right)$ in all objectives. An example can be seen in Figure 2, where the blue Pareto front has a better response of $J_1(x)$ and $J_2(x)$ than the other solution options. MOO has shown good results in works like the one developed by Demirtas et al. (2019), where an Elman Neural Network and a FOPI controller were used to control the speed of induction motor. Another example is the use of a multi-dimensional particle swarm optimization algorithm for the parallel tuning of six fractional order controllers for a PMSG based wind energy conversion system, performed by Sahraoui et al. (2023). An application of particle swarm optimization was developed by Aguilar et al. (2020) for the PI controllers of a wind turbine under electrical fault conditions to improve the control behaviour. Within the concepts of MOO, it is important to determine the design concept of the controller before establishing other procedures. The characteristic values of controllers will be optimized taking into account the general description of the PID controller shown in Equation 2, which can be generalized and applied for any combination of controllers:

$$\begin{aligned} C\left(s\right) = K_c\left(b + \frac{1}{T_i s^{\lambda}} + c\frac{T_d \cdot s^{\mu}}{\frac{T_d}{N}s^{\mu}+1}\right)R\left(s\right) \\ -K_c\left(1 + \frac{1}{T_i s^{\lambda}} + \frac{T_d \cdot s^{\mu}}{\frac{T_d}{N}s^{\mu}+1}\right)Y\left(s\right) \end{aligned} \qquad (2)$$

where K_c is the proportional gain, T_i the integral time, T_d the derivative time, N the derivative filter, b, c the set-points; λ and μ are the fractional order. In order to have a successful optimization process, it is vital to separate it into three stages: multi-objective problem definition, multi-objective optimization process and multi-criteria decision making.

Multi-objective Problem Definition (MOP) It defines the characteristics of the problem to be solved, the mathematical model and the optimization objectives for measuring the desired performance to be achieved. MOP characterizes to be formulated from the point of view of the designer, instead of designing from the point of view of the optimizer.

Multi-objective Optimization Process Evolutionary algorithms could be used to approximate a Pareto set of viable solutions. These algorithms base their behaviour in evolutionary and nature-inspired techniques due to their ability to evolve an entire population towards the Pareto front.

Multi-criteria Decision Making (MCDM) This step uses the Pareto front obtained previously to show all the viable solutions for the MOP. It is more common to use visualization tools in order to understand the trade-offs present in the Pareto front. Two-dimensional problems can be accurately assessed using graphical analysis of the Pareto front, but it becomes more complex as the problem has more dimensions to be evaluated.

4. MODEL OF THE ANESTHESIA ADMINISTRATION SYSTEM

This work seeks to control an anesthesia administration system designed specially for children, where propofol is the main drug used to induce a state of deep hypnosis. The medical and scientific support for this model is provided by the work of van Heusden et al. (2013). It proposes a closed-loop control system that regulates the infusion of propofol to the patient, using the depth of hypnosis (DOH) as the measured signal for feedback, obtained from the NeuroSENSE DOH monitor. The following constraints need to be considered for the controller:

(1) A state of deep anaesthesia within 40-60% must be achieved within the first 5 minutes of operation;
(2) Hypnosis states below 40% should be avoided for more than 5 minutes;
(3) Settling time must be minimized.

4.1 Patient database

The information of infusion rates and recordings were obtained using the WAV_{CNS} index in a closed-loop control of propofol anesthesia (van Heusden et al., 2013). A total of 99 patients were documented in an initial phase between 6-16 years of age. After analyzing the database, 16 patients were discarded due to insufficient data quality or corrupted data and 36 patients were discarded due to a strong reaction to stimulation during the induction of anesthesia. From this new total of 50, 4 models could not be validated due to a lack of robustness. The final number of patient models to be used was 46 and an extract of their information can be seen in Table 1. E_0 is the average effect measured during the first 50s of propofol infusion, T_d is the time delay between the infusion and the observed clinical effect, γ is the cooperativity coefficient, E_{50} is the effect site concentration corresponding to a 50% of effect and k is the time needed for the patient to reach a stable DOH.

Table 1. Extract of model parameters of 15 patients

Patient	E_0	T_d $[s]$	k $[min^{-1}]$	E_{50} $[\mu g/kg/min]$	γ
1	93.11	35	0.152	217	1.77
2	92.46	82	0.135	316	1.91
3	92.46	21	0.254	385	1.94
4	91.47	48	0.188	515	1.57
5	91.60	41	0.108	315	1.58
6	88.45	40	0.214	365	1.80
7	92.91	68	0.194	282	1.63
8	88.89	94	0.212	473	1.53
9	94.58	16	0.132	263	1.71
10	92.89	115	0.177	415	1.56
11	91.68	29	0.133	267	1.83
12	90.30	4	0.058	228	1.64
13	91.38	41	0.131	229	2.01
14	92.76	58	0.251	400	1.81
15	91.78	117	0.288	282	1.81

4.2 Model of the problem

The model of this problem was obtained through supervised experiments both in open and closed loops and these produced a first-order plus dead-time (FOPDT) model (Equation 3) and a Hill function (Equation 4) to capture the non-linearity inherent of the process, where the parameters E_0, E_{50} and γ were identified from data and E_{LTI} is the effect predicted by the FOPDT model (van Heusden et al., 2013).

$$P_i(s) = \frac{1}{\frac{s}{k}+1} e^{T_d} \tag{3}$$

$$E(t) = E_0 - E_0 \frac{\mathrm{E}_{LTI}^{\gamma}}{\mathrm{E}_{50}^{\gamma} + \mathrm{E}_{LTI}^{\gamma}} \tag{4}$$

5. METHODS AND TOOLS

This section describes all the elements involved in the application, optimization and test of the controllers under the established constraints.

5.1 Definition of the Multi-objective Problem

After identifying the process and the mathematical model described in Section 4, the control loop presented in Figure 3 will be used.

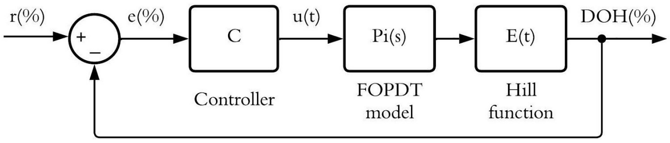

Fig. 3. PI control loop of the problem.

The measured output will be the DOH. Due to the similar structure between PI and FOPI, the same control loop will be applied for the optimization process. For the controller equation presented in Equation 2, the following design concepts and decision variables will be used:

(1) PI: $\theta_{PI} = [K_c, T_i], b = 1, T_d = 0, \lambda = 1$.
(2) FOPI: $\theta_{FOPI} = [K_c, T_i, \lambda], b = 1, T_d = 0$.

The MOP statement can be defined as the minimization of the following multi-objective vector:

$$\min_{\theta} J(\theta) = [J_1(\theta), J_2(\theta)] \tag{5}$$

The design objectives are based on the Integral of the Absolute Value of Error (IAE), understood as:

$$IAE(\theta) = \int_{t=t_0}^{t_f} |r(t) - y(t)|\, dt = \int_{t=t_0}^{t_f} |e(t)|\, dt \tag{6}$$

The objectives are:

Median of the Integral of the Absolute Value of Error (IAE)

$$J_1(\theta) = median(IAE(\theta)) \tag{7}$$

Interquartile range of the Integral of the Absolute Value of Error (IAE)

$$J_2(\theta) = IQR(IAE(\theta)) \tag{8}$$

5.2 Multi-objective Optimization Algorithm

Algorithm 1 is a modified version of the Differential Evolution (DE) algorithm that includes convergence capabilities and spherical pruning to promote diversity properties (Meza et al., 2017).

Algorithm 1: MOEA with pruning mechanism

```
[h] Generate initial population $P|_0$ with $N_p$ individuals;
Evaluate $P|_0$;
Apply dominance criterion on $P|_0$ to get archive $\hat{A}|_0$;
Apply pruning mechanism to prune $\hat{A}|_0$ to get $A|_0$;
Set generation counter $G = 0$;
while maximum number of generations reached do
    Update generation counter $G = G + 1$;
    Get subpopulation $S|_G$ with solutions in $P|_{G-1}$ and $A|_{G-1}$;
    Generate offspring $O|_G$ with $S|_G$;
    Evaluate offspring $O|_G$;
    Update population $P|_G$ with offspring $O|_G$;
    Apply dominance criterion on $O|_G$;
    Apply pruning mechanism to prune $\hat{A}|_G$ to get $A|_G$;
    Update environment variables (if using a self-adaptive mechanism);
end
return Pareto set approximation $\Theta_p^* = A|_G$;
```

The spherical pruning modification is added to this algorithm to attain a good distribution along the Pareto Front through the analysis of the proposed solutions in the current Pareto Front approximation by using normalized spherical coordinates from a reference solution. The following definitions will be taken into account.

Definition 1 Normalized spherical coordinates Given a solution θ^i and $J(\theta^i)$, where r is the radius and β the angle, let

$$S\left(J(\theta^i), J^{ref}\right) = [r, \beta] \tag{9}$$

Definition 2 Sight range The sight range from the reference solution J^{ref} to the Pareto Front approximation J_P^* is bounded by:

$$\begin{aligned} \beta^U &= \left[max\beta_1\left(J\left(\theta^i\right)\right), ..., max\beta_{m-1}\left(J\left(\theta^i\right)\right)\right] \forall J\left(\theta^i\right) \\ \beta^L &= \left[min\beta_1\left(J\left(\theta^i\right)\right), ..., min\beta_{m-1}\left(J\left(\theta^i\right)\right)\right] \forall J\left(\theta^i\right) \end{aligned} \tag{10}$$

Definition 3 Spherical grid Given a set of solutions in the objective space, the spherical grid on the m-dimensional space in arc increments is defined as:

$$\Lambda^{J_P^*} = \left[\frac{\beta_1^U - \beta_1^L}{\beta_1^{\epsilon}}, ..., \frac{\beta_{m-1}^U - \beta_{m-1}^L}{\beta_{m-1}^{\epsilon}}\right] \tag{11}$$

Definition 4 Spherical sector The normalized spherical sector of a solution θ^i is defined as:

$$\Lambda^{\epsilon}\left(\theta^i\right) = \left[\left\lceil \frac{\beta_1\left(J\left(\theta^i\right)\right)}{\Lambda_1^{J_P^*}} \right\rceil, ..., \left\lceil \frac{\beta_{m-1}\left(J\left(\theta^i\right)\right)}{\Lambda_{m-1}^{J_P^*}} \right\rceil\right] \tag{12}$$

Definition 5 Spherical pruning Given two solutions θ^i and θ^j from a set, θ^i has preference in the spherical sector over θ^j if:

$$\begin{aligned} &\left[\Lambda^{\epsilon}\left(\theta^i\right) = \Lambda^{\epsilon}\left(\theta^j\right)\right] \\ \wedge &\left[\left\|J\left(\theta^i\right) - J^{ref}\right\|_p < \left\|J\left(\theta^j\right) - J^{ref}\right\|_p\right] \end{aligned} \tag{13}$$

Algorithm 2 shows the implementation of spherical pruning.

5.3 Multi-criteria Decision Making definition

A graphical analysis will be performed as it is a two-dimensional problem where the Pareto front of the PI controller solution will be compared to its FOPI counterpart.

Algorithm 2: Spherical pruning mechanism

```
Read archive $\hat{A}|_G$;
Read and update extreme values for $J^{ref}|_G$;
for each member in $\hat{A}|_G$ do
    calculate its normalized spherical coordinates (Definition 1);
end
Build the spherical grid (Definition 2 and 3);
for each member in $\hat{A}|_G$ do
    calculate its spherical sector (Definition 4);
end
for i=1:Solutions In Archive do
    Compare with the remainder solutions in $\hat{A}|_G$;
    if no other solution has the same spherical sector then
        it goes to archive $A|_G$;
    end
    if other solutions are in the same spherical sector then
        it goes to archive $A|_G$ if it has the lowest norm (Definition 5);
    end
end
return Archive $A|_G$;
```

It is expected to have a dominant Pareto front, or at least, a group of solutions that dominate the other options.

5.4 Description of the Equipment

The platform used for the simulation was an Asus ROG Strix G513QR personal computer, with a Ryzen 9, 8 core, 5000 series processor, 32GB DDR4 of RAM and GeForce RTX3070 8GB video card, running on Windows 11. Processing and simulation were performed in Matlab R2023a. In Simulink, the fixed integration step for the simulation was $1e-3$ with ode8 (Dormand-Prince) solver.

6. RESULTS

The multi-objective optimization process was performed both for the PI and the FOPI controllers using the same equipment and initial conditions. The optimization time for the PI controller was around 13 hours and for FOPI was around 11 hours, due to the cost function that has a skip criteria that prevents the simulation to stay in an error, reducing the optimization time. Each optimization process produced a Pareto front, both are presented in Figure 4. It can be seen that the Pareto front of the solutions of FOPI dominate the ones of the PI controller, although the PI Pareto front has a less disperse front. After evaluating the fronts, the best option of each Pareto set was determined by testing each option with the 46 patients and looking for the case that complies with the anesthesia administration system constraints, or that has the highest number of cases that comply with the aforementioned constraints explained in Section 4. For the PI, the chosen constants were: $k_p = 10.0629$ and $k_i = 2.1669$. For FOPI, the chosen constants were: $k_p = 28.685$, $k_i = 0.9773$ and $\lambda = 1.6175$. The results of these tests with the 46 patients can be seen in Figure 5. The PI controller showed a stable behaviour with all patients staying within 40% and 60% DOH and with a good steady state value. FOPI controller showed a stable behaviour with most patients, 41 cases stayed within 40% and 60% DOH with a good steady state value, while 5 cases (patients 15, 16, 17, 19 and 26) fell into instability over the threshold and did not comply with the established design constraints. It was found that, in these five cases, a combination of high values of k and *delay* parameters led into instability for FOPI. The time improvement in control of FOPI when compared to PI had an average of 2.0767 minutes, considering that the average time that the PI controller needed to get into the desired DOH was 5.5621 minutes.

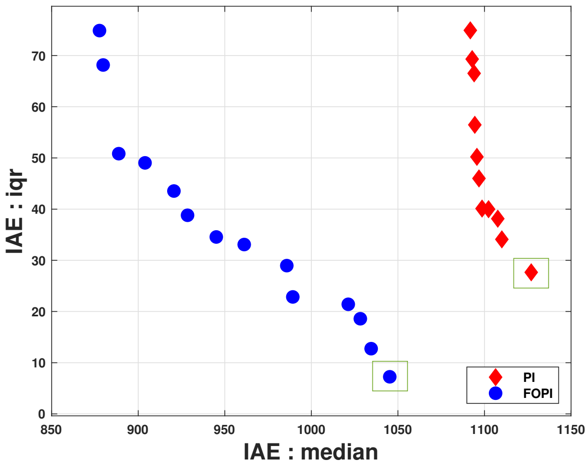

Fig. 4. Comparison of Pareto fronts after the multi-objective optimization.

7. CONCLUSION

Regarding the Pareto fronts, the FOPI Pareto front presents a better set of control options than PI when analyzing the concept of dominance. After testing the best option of control variables with all patients, the FOPI controller presented a faster control response but was not able to control 5 patients that had high values of k and *delay*, while the PI controller had a more stable response in all cases and was able to control all patients. For this particular problem, it can be concluded that, although FOPI had a time response 37.33% faster, the PI controller has better overall control characteristics and complied with all the design constraints in all patients, hence, it is more desirable for this problem and a more complicated control scheme (FOPI) is not justified.

REFERENCES

Aguilar, M.E.B., Coury, D.V., Reginatto, R., and Monaro, R.M. (2020). Multi-objective pso applied to pi control of dfig wind turbine under electrical fault conditions. *Electric Power Systems Research*, 180, 106081.

Borase, R.P., Maghade, D., Sondkar, S., and Pawar, S. (2021). A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, 9, 818–827.

Dabiri, A., Moghaddam, B., and Machado, J.T. (2018). Optimal variable-order fractional pid controllers for dynamical systems. *Journal of Computational and Applied Mathematics*, 339, 40–48. doi: https://doi.org/10.1016/j.cam.2018.02.029.

Demirtas, M., Ilten, E., and Calgan, H. (2019). Pareto-based multi-objective optimization for fractional order

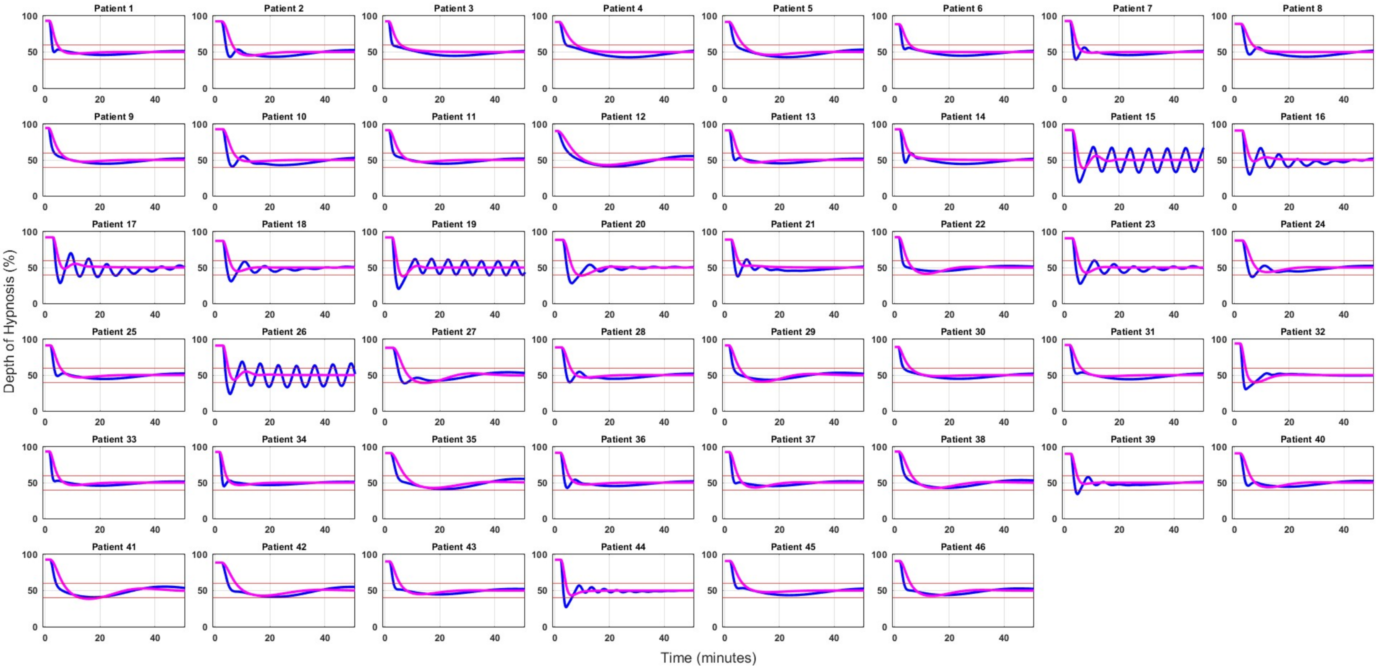

Fig. 5. Results of the test with the PI controller (pink) and the FOPI controller (blue) for 46 patients.

pi λ speed control of induction motor by using elman neural network. *Arabian Journal for Science and Engineering*, 44(3), 2165–2175.

Díaz-Rodríguez, I.D., Han, S., and Bhattacharyya, S.P. (2019). *Analytical design of PID controllers*. Springer.

Edet, E. and Katebi, R. (2018). On fractional-order pid controllers. *IFAC-PapersOnLine*, 51(4), 739–744. doi:https://doi.org/10.1016/j.ifacol.2018.06.208. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.

Kumar, J., Kumar, V., and Rana, K. (2018). A fractional order fuzzy pd+i controller for three-link electrically driven rigid robotic manipulator system. *Journal of Intelligent and Fuzzy Systems*, 35, 1–13. doi:10.3233/JIFS-169812.

Marques, T. and Reynoso-Meza, G. (2020). Applications of multi-objective optimisation for pid-like controller tuning: a 2015-2019 review and analysis. *IFAC-PapersOnLine*, 53(2), 7933–7940. doi: https://doi.org/10.1016/j.ifacol.2020.12.2140.

Meza, G.R., Ferragud, X.B., Saez, J.S., and Durá, H. (2017). Controller tuning with evolutionary multiobjective optimization. *Switzerland: Springer*.

Ogata, K. (2010). *Modern control engineering fifth edition*. Prentice Hall.

Pullaguram, D., Mishra, S., Senroy, N., and Mukherjee, M. (2018). Design and tuning of robust fractional order controller for autonomous microgrid vsc system. *IEEE Transactions on Industry Applications*, 54(1), 91–101. doi:10.1109/TIA.2017.2758755.

Reddy, V.S. and Devabhaktuni, S. (2022). Enhanced low-speed characteristics with constant switching torque controller-based dtc technique of five-phase induction motor drive with fopi control. *IEEE Transactions*.

Sahraoui, K., Lalalou, R., Boutasseta, N., Attoui, I., Fergani, N., and Frikh, M.L. (2023). Multi-objective optimization strategy for parallel tuning of multiple fractional order controllers in a pmsg based wecs. *Wind Engineering*. doi:10.1177/0309524X231203377.

Seo, S.W. and Choi, H.H. (2019). Digital implementation of fractional order pid-type controller for boost dc–dc converter. *IEEE Access*, 7, 142652–142662. doi: 10.1109/ACCESS.2019.2945065.

Su, Y., Yu, Q., and Zeng, L. (2020). Parameter self-tuning pid control for greenhouse climate control problem. *IEEE Access*, 8, 186157–186171. doi: 10.1109/ACCESS.2020.3030416.

van Heusden, K., Ansermino, J.M., Soltesz, K., Khosravi, S., West, N., and Dumont, G.A. (2013). Quantification of the variability in response to propofol administration in children. *IEEE Transactions on Biomedical Engineering*, 60(9), 2521–2529. doi: 10.1109/TBME.2013.2259592.

Wang, B., Yu, T., Zhou, T., Wang, L., Li, J., and Xie, N. (2023). Fractional order pi λ d μ for tracking control of a novel rehabilitation robot based on iimo-bp neural network algorithm. *Journal of Mechanics in Medicine and Biology*, 23(01), 2350010.

Xu, Y., Zhao, K., Guo, P., Jiang, W., Wu, X., and Sun, W. (2022). Design of type-2 fuzzy fractional-order proportional-integral-derivative controller and multi-objective parameter optimization under load reduction condition of the pumped storage unit. *Journal of Energy Storage*, 50, 104227. doi: https://doi.org/10.1016/j.est.2022.104227.

Zaid, S., Bakeer, A., Albalawi, H., Alatwi, A., AbdelMeguid, H., and Kassem, A. (2023). Optimal fractional-order controller for the voltage stability of a dc microgrid feeding an electric vehicle charging station. *Fractal and Fractional*, 7. doi: 10.3390/fractalfract7090677.

Åström, K. and Hägglund, T. (2001). The future of pid control. *Control Engineering Practice*, 9(11), 1163–1175. doi:https://doi.org/10.1016/S0967-0661(01)00062-4.