Design of Digital PID Controllers using Particle Swarm Optimization: A Video Based Teaching Experiment

P. B. de Moura Oliveira

UTAD-Universidade de Trás-os-Montes e Alto Douro, Escola de Ciências e Tecnologia, 5000-801 Vila Real, Portugal
INESC-TEC Technology and Science, Campus da FEUP, 4200 - 465 Porto, Portugal
(e-mail: oliveira@utad.pt)

Abstract: The use of videos is a valuable and powerful tool which may significantly contribute to change and improve teaching and learning methods. Lecturers can made their own videos addressing specific topics suitable to fulfill their student's needs. These videos can address control engineering syllabus as well as complementary topics. This paper proposes using video as a tool to introduce the particle swarm optimization algorithm to students within a digital PID control simulation experiment. The experience preliminary results and feedback received from students are quite positive.

Keywords: PID Control, Videos, Swarm Optimization, Internet based learning.

1. INTRODUCTION

Current students demand new teaching and learning methodologies. Videos can be used for different purposes, such as the following examples: i) to record classes and make them available in an internet repository or university learning management system (Crook and Schofield, 2017); This approach is currently used both by eLearning and classical courses; ii) to provide short introductions addressing topics covered in classes (Rossiter, 2013). Videos are the flipped (or inverted) classroom (FC) approach key essence (Rossiter, 2014; Oliveira and Boaventura, 2017). In the FC, short videos can be used by students as a preparation element for the next class, releasing class-time which can be best used to promote students engagement and motivation with other learning activities (e.g. group problem solving, quizzes answering, computer simulations, group critical debate, etc.). However, videos can also be used by students to complement their learning process whenever they feel like it and at their own pace; iii) control simulations or practical rig demos; iv) technical training support (Starr *et al.*, 2015). Indeed, it has been found that videos can help in increasing student's motivation to the learning process (Bravo *et al.*, 2011). Videos can also be used to provide introduction to complementary topics not covered in industrial control and automation courses. This is the case of some Artificial Intelligence (AI) and Machine Learning topics, which are skills highly requested in the Internet of Things and Industry 4.0.

Proportional, Integrative and Derivative (PID) controllers are a fundamental control engineering education topic, quite relevant due to its extensive practical use in industrial systems. This topic is transversal to different engineering applications (Electrical, Mechanical, Chemistry, Biomedical, etc.). An important skill to be acquire by students is how to design PID controllers. Since Ziegler and Nichols (1942) breakthrough techniques many alternative and complementary PID tuning and design methods have been proposed (e.g. Åström and Hägglund, 2004; Vrančić 2001; O'Dwyer A., 2006)). With the development of computer based methods, the incorporation of optimization approaches constitute a strong alternative to design PID controllers (Mercader *et al.*, 2017). Optimization methods which are inspired in nature and biological (NABI) phenomena have been successfully applied to design PID controllers. Examples of the most well-established methods are genetic algorithms (Holland, 1975), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995) , differential evolution (Storn and Price, 1995), etc. Indeed, NABI can be used as an alternative to classical design methods. Some of their advantages are the following: they just need a cost function to guide the search procedure; they do depend on the evaluation of derivatives or gradients; they are independent of the type of system to be controlled and may not require any knowledge regarding their specific dynamics. Thus considering the success popularity attained by NABI techniques in solving a wide range engineering problems it is natural to teach this methods in control engineering courses.

In this paper, following previous author experiences regarding the use of NABI in control engineering education (Oliveira, 2005; Oliveira and Boaventura, 2016), a video based approach is proposed within a student's simulation assignment, bridging PSO and control engineering topics. The PSO algorithm is adopted as an optimization tool to design digital PID controllers within a teaching/learning experience. This study was performed in a Digital Control course of the 4th year of UTAD Electrical Engineering and Computers (5 years course). As it will be further detailed, the experience is based on two main stages:

1. Understanding of basic PSO algorithm through a computer program implementation, to minimize a benchmark continuous quadratic function. Skills regarding some of the main PSO issues perception are promoted, in order to make the transition to PID controller design easier. An introductory video was made available to students (English version available in (Oliveira, 2018),

presenting elementary notions about the PSO dynamics as well as the some tests assuming different settings for major PSO heuristic parameters. Students are expected to replicate these tests in their own simulations.

2. To extend and adapt the implemented PSO algorithm in stage 1 to design digital PID controllers. Here the main objective lies in the design of an objective function to simulate the digital control loop. This function returns values for pre-defined performance criteria, such as: the integral of absolute error (IAE), integral of absolute error weighted by time (ITAE) or integral of square error (ISE).

The remainder of the paper is organized as follows: section 2 presents some key issues regarding digital PID controllers. Section 3 presents the simulation experiment description. Section 4 presents aspects regarding the type of results expected as well as students feedback and finally section 5 concludes the paper and outlines further work.

2. DIGITAL PID CONTROLLERS ISSUES

A single-input single-output control loop is considered in the digital domain such as the one represented in Fig.1:

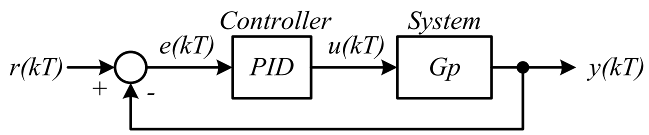

Fig. 1. Digital control feedback loop.

with r, e, u, y, T, k representing respectively the: reference input, system error, controller output, controlled output, sampling interval and sample index. The main learning issues related with digital implementation of PID controllers in the reported simulation experiment promotes student's perception of:

1. The PID control law digitalization: starting with an ideal parallel format, the digital approximation of the continuous integrative and derivative PID components are the main issues. The use of different approximations for the integrative of the error are tested, namely: first regressive error, first forward error, trapezoidal and Tustin. A first order plus time (FOPTD) delay model is adopted in this experiment and simulation of the digital versus the continuous implementations are tested. After this stage students can test other PID control laws (e.g. with derivative action filtering, set-point weighting, etc.)

2. Absolute and incremental PID control implementations: the results of simulating the PID control loop with both algorithms are compared.

3. The outcome of testing different settings for the PID controller gains obtained using some tuning rules;

4. The effect of derivative kicking and the influence of applying the derivative action to the system output.

5. The actuator saturation effect in the control system performance, namely by observing the integrative windup and the relevance of halting the respective control action in the saturation period to avoid it.

Students are asked to implement the digital control loop by developing a script (in Matlab or Python) based on a difference equations approach. The following equations for the positional and incremental PID controller based on the rectangular backward error difference, can serve as a starting point:

$$u(kT) = K_p \left[e(kT) + \frac{1}{T_i}\sum_{i=1}^{k} Te(kT) + T_d \frac{e(kT) - e((k-1)T)}{T} \right] \quad (1)$$

$$\Delta u(kT) = K_p \left[\Delta e(kT) + \frac{Te(kT)}{T_i} + T_d \frac{\Delta_2 e(kT)}{T} \right] \quad (2)$$

with: K_p, T_i and T_d, representing respectively the proportional constant, the integrative and derivative time constants; $\Delta e(kT)=e(kT)-e((k-1)T)$ and $\Delta_2 e(kT)=e(kT)-2e((k-1)T)+e((k-2)T)$ and the digital FOPTD model using a zero-order hold:

$$y(kT) = e^{\frac{-T}{\tau}} y((k-1)T) + K(1- e^{\frac{-T}{\tau}})u((k-1-\Gamma)T) \quad (3)$$

with: K, τ and Γ representing respectively the dc gain, dominant time constant and time delay samples.

3. PARTICLE SWARM OPTIMIZATION BASIC ISSUES

Basic issues regarding the PSO algorithm are addressed in the experiment supporting video. Each swarm particle is characterized by two variables, x and v, representing respectively its position and velocity in the search space. The search space is d-dimensional, but for simplicity of exposition particle position and velocity equations are introduced without considering the dimension index. The new particle velocity, is evaluated from the current velocity, corresponding to iteration t, using the following equation:

$$v_i(t+1) = v_i(t)+c_1 \varphi_1 (b_i(t) - x_i(t))+ c_2 \varphi_2 (g(t) - x_i(t)) \quad (4)$$

with: b representing particle i best position obtained until the current iteration; g representing the global best position, which in this case considers the entire swarm; c_1 and c_2 are known as the cognitive and social constants; φ_1 and φ_2 are random numbers generated in the interval [0,1]. After each particle velocity is evaluated the new particle position can be updated using:

$$x_i(t+1) = x_i(t)+ v_i(t+1) \quad (5)$$

As in any search technique, it is important to guaranty a compromise among a swarm exploratory behavior in initial search stage and a specialization behavior toward the end. This compromise among exploration and exploitation, can be obtained by incorporating a inertia weight, ω, in (4), as follows:

$$v_i(t+1) = \omega\, v_i(t)+c_1 \varphi_1 (b_i(t) - x_i(t))+ c_2 \varphi_2 (g(t) - x_i(t)) \quad (6)$$

The inertia weight is often decreased from a higher value to a lower value along the search.

4. SIMULATION EXPERIMENT DESCRIPTION

The experiment is organized in two parts:

1. PSO algorithm implementation and testing using a simple benchmark function minimization problem. This stage enables the PSO key principles to be apprehended by students and then easily adapted to design PID controllers. A video (Oliveira, 2018) was produced by the paper author and made available to students, providing a brief

introduction to the PSO algorithm and experiment test demos that students are asked to replicate. As it will be further described, this stage main learning objective is that students successfully implement a simple PSO algorithm. Once the PSO is implemented, students should test the effect of adjusting some of its heuristic parameters, namely: population size, number of iterations per run, inertia weight, maximum velocity clamping and bounding particles position in the search space.

2. PID digital controller design adapting the PSO implemented in Part 1. This requires the PID controller implementation in the digital domain and respective control loop simulation.

Part II addresses digital controls topics which are currently well-known, and whose basics have been introduced in section 2. Thus, the remaining of this section will be focus to explain Part I.

The simple function used to demonstrate PSO concepts is a quadratic expression represented by:

$$f(x_1,x_2) = (x_1 - 50)^2 + (x_2 - 50)^2 \qquad (7)$$

and the decision variables are allowed to vary within the interval [0,100] and the optimum value of 0 is obtained in [50,50]. The corresponding search space can be visualized in Fig. 2.

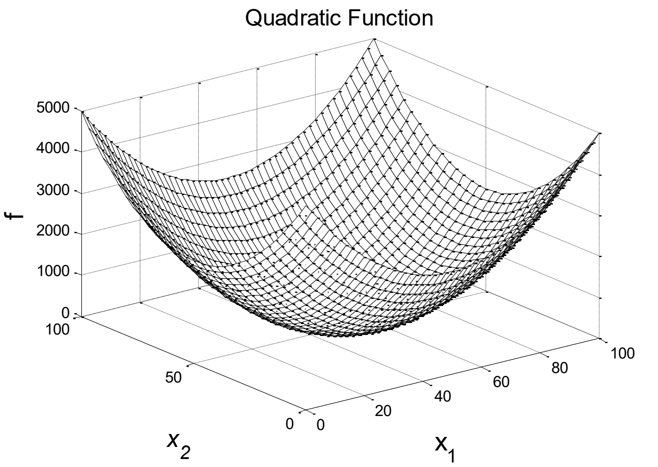

Fig. 2. Considered search space for function (7).

Particles initial velocities were set to 0 in all the video (Oliveira, 2018) simulations. A key aspect to be addressed in the experiment is the inertia weight influence in the PSO search. While students can test other settings, two different conditions are proposed to be tested and illustrated in the video:

- Linearly decaying ω from ω_{max}=0.9 to ω_{max}=0.4 along the search.
- Fixed value, ω=0.4, along the search.

The initial number of iterations considered per run is 70 iterations. Two swarm sizes are considered in the PSO demos: n=4 particles and n=50 particles. Regarding the small sized swarm (n=4), three different cases are considered regarding the particles initialization and starting positions:

- Random initialization considering the entire search space.
- Random initialization considering a corner of the search space (e.g. range [90,100]).
- Fixed initialization, with a particle assigned to each corner of the search space.

The velocity value was clamped to a maximum absolute value of V_{max}=3.33 per iteration. However, it is pedagogical that students start the PSO simulations without limiting the maximum velocity value. The results of running the PSO, considering a swarm with 4 particles, randomly initializing the swarm in the entire search space, and decaying the inertia weight is presented in Fig. 3. In this figure initial solutions are represented inside a square sign and final solutions with a white circle. The evolution of the best values for both parameters is illustrated in Fig. 4. The results show that all 4 particles converged to the global minimum. Around iteration 36 the decision variables best value reached a steady-state value.

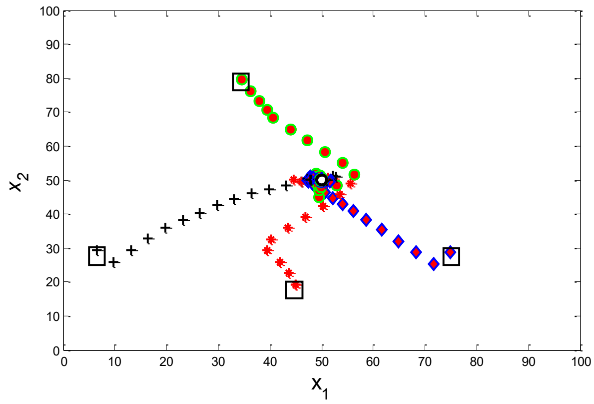

Fig. 3. Test 1: swarm with n=4, randomly initialization considering the entire search space and decaying ω in the interval [0.9, 0.4].

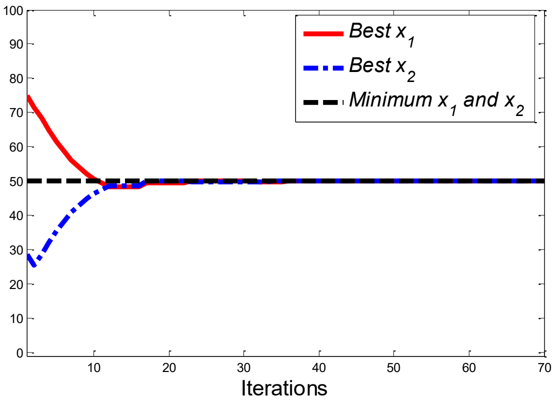

Fig. 4. Evolution of the best values for x_1 and x_2 for test 1 (Fig. 2).

The results of a PSO run with four particles starting from positions initialized in the [90,100] for both dimensions, are presented in Fig. 5 and Fig. 6. These figures illustrate that the swarm could leave the initialization region towards the optimum value region. However, this run failed to reach the optimum value in one dimension. This indicates that more iterations would be necessary to reach the optimum value. If a constant value of ω=0.4 is used keeping the same remaining test conditions the swarm is prone to stay in the initialization corner. It is important to remark that in these tests the initial particles velocity was set to zero and the maximum value it were allowed to change in each iteration is a low value.

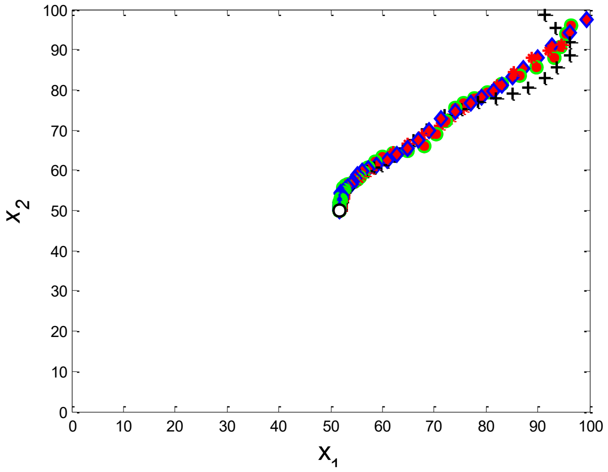

Fig. 5. Test 2: swarm with n=4, randomly initialization in interval [90,100] and decaying ω in the interval [0.9, 0.4].

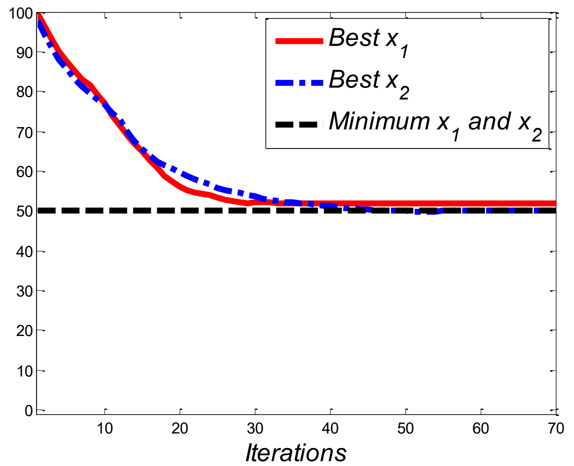

Fig. 6. Evolution of the best values for x_1 and x_2 for test 2 (Fig. 5).

The results of a test run with the four particles starting from initial positions defined in the search space four corners are presented in Fig. 7 and Fig 8. The results obtained with a swarm size of n=50 and a fixed inertia weight of 0.4 are presented in Fig 9 and Fig 10. Even for this simple two dimensional function the speed of convergence tends to be reduced as the swarm size is increased.

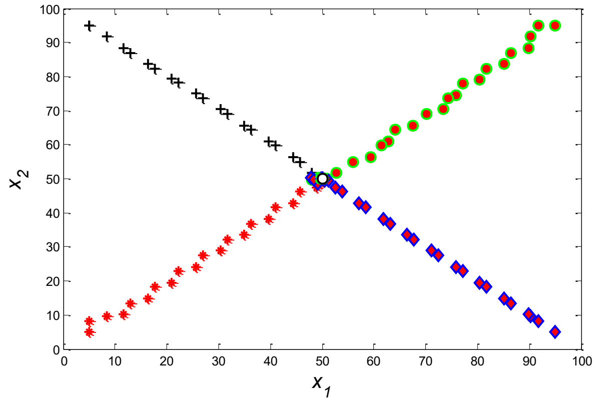

Fig. 7. Test 3: swarm with n=4, initialized in the four corners of the search space and decaying ω in the interval [0.9, 0.4].

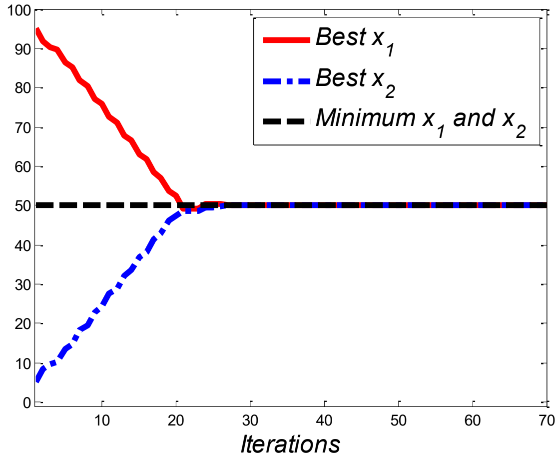

Fig. 8. Evolution of the best values for x_1 and x_2 for test 3 (Fig. 6).

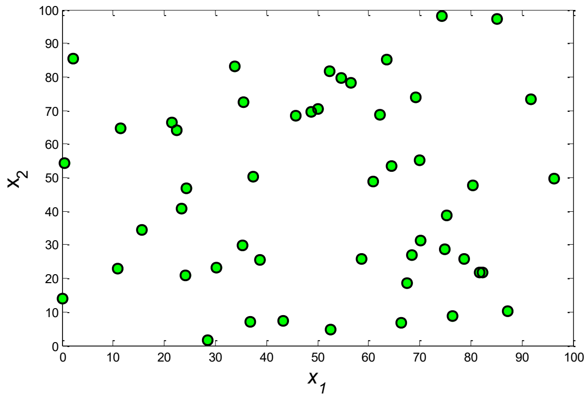

Fig. 9. Test 4: swarm with n=50, initialized randomly in the entire search space. Inertia weight fixed, ω=0.4.

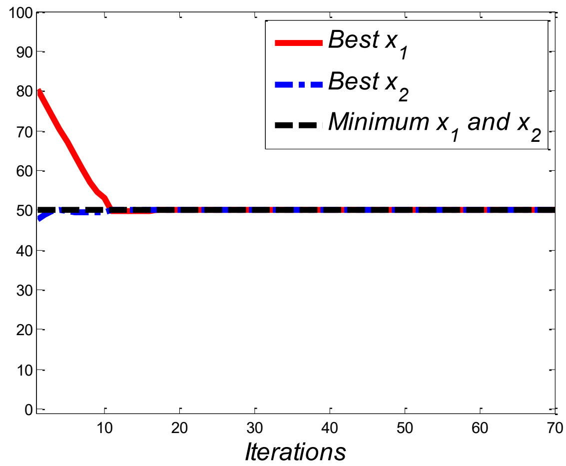

Fig. 10. Evolution of the best values for x_1 and x_2 for test 4 (Fig. 8).

Results using the same test conditions of tests 3 and 4 with different inertia weight are included in (Oliveira, 2018). Once students completed the PSO learning stage by apprehending the main concepts they can move forward to adapt the PSO to design PID controllers. A first step can be considering optimizing (7) with three dimensions.

4. DESIGN OF DIGITAL PID CONTROLLERS WITH PSO

In this stage students must incorporate into the PSO script an objective function to simulate the digital control system response to a step input. While, this objective function can be more easily implemented using Matlab functions and/or Simulink models, authors are convinced it is pedagogical to code the feedback loop using difference equations representing both the PI controller and plant model. The following specifications are proposed to students as a starting point, by considering:

- a FOPTD model with $K=1$ and $L=T$=1s, using a sampling time, T=0.1s.
- The absolute PID controller digital form represented by the approximated model (1) applying the derivative action to the system output.
- A search space defined by interval [0.01 5] for the three controller gains.
- Set-point tracking performance optimization by minimizing an error based criterion such as: IAE, ITAE or ISE when an unit step is applied to the reference input.

Regarding the PSO the following issues are proposed to be addressed as a starting point, by considering:

- particles randomly initialized in the search space with zero value for their velocity. This means that the initial swarm is allowed to have particles representing unstable controller settings. Students can in a later stage test informed population initialization techniques (e.g. by using PID tuning rules).
- Linearly decayed inertia weight and fixed value inertia weight along the search.
- Forcing particles to stay within the parameter search limits. Particles generated outside the limits are clamped to the nearest parameter interval limit.
- Starting by not limiting the velocity value, and then testing limiting the velocity to a maximum absolute value per iteration (v_{max}).

Considering as illustrative examples the design based on the IAE and ITAE minimization, the type of results that can be obtained and analyzed by students are presented in Fig. 11-13. It is clear from Fig. 11 and Fig 12 that with the fixed value inertia weight, the gains parameter variation is smaller along the search compared to the linear decayed case, thus confirming a faster PSO convergence rate. The gain sets obtained for the IAE designs are: [K_p=0.687, T_i=1.39, T_d=0.01] both for the inertia decayed and fixed inertia cases, resulting in IAE=2.16. For the ITAE designs are: [K_p=0.57, T_i=1.21, T_d=1.89] [K_p= 0.57, T_i=1.21, T_d=2.12] for the inertia decayed and fixed inertia cases, respectively, both with ITAE=2.89.

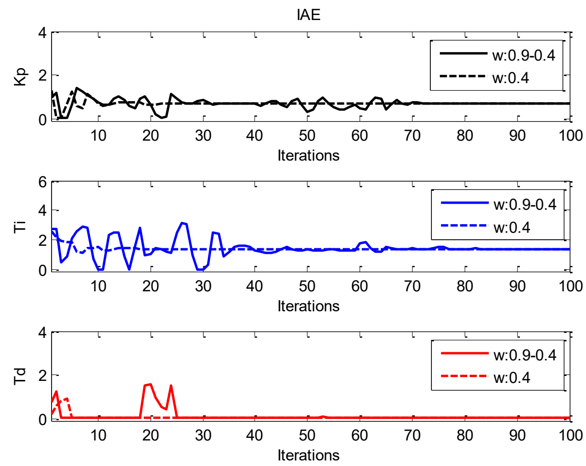

Fig. 11. Variation of best PID gains versus iteration number using IAE criterion.

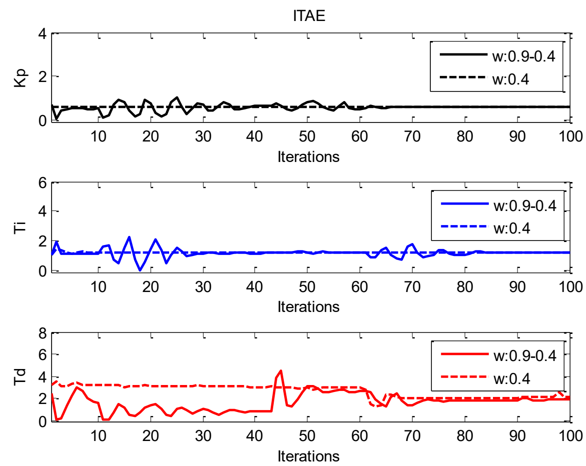

Fig. 12. Variation of best PID gains versus iteration number using ITAE criterion.

The results of these gains tracking responses are shown in Fig. 13 for both the IAE and ITAE designs.

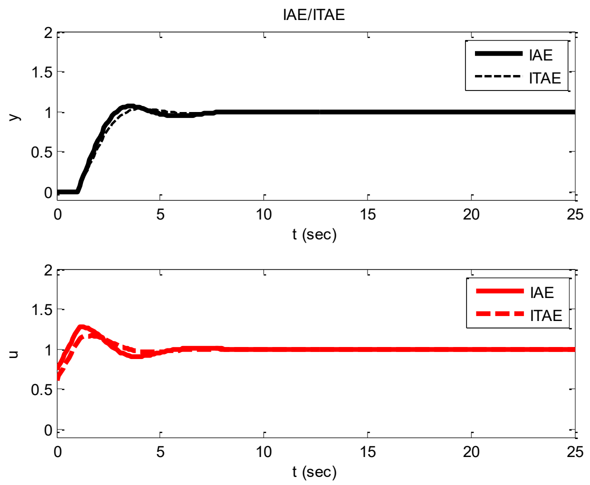

Fig. 13. Set-point tracking response for final best gains IAE versus ITAE design.

5. DISCUSSION AND CONCLUSION

A PSO based experiment to design digital PID controllers has been reported. The experiment was conducted in the first semester of 2017-2018, within a digital control course to the 4th year electrical engineering and computers degree (5 years course). The PSO algorithm was introduced to students by providing a video explaining the algorithm bare bone dynamics, as well as presenting some test results. These tests regard the minimization of a simple function, and students are expected to implement the PSO algorithm and be able to replicate similar results.

The results obtained in practical classes indicated that students could successfully implement a PSO algorithm and replicate the proposed tests. Perception and practical sensibility regarding the principal PSO adjustable heuristic parameters was gained. This PSO learning stage, allowed a fast transition from the benchmark function optimization to the PID digital controller optimization. Different aspects regarding the digital PID control implementation were implemented and tested allowing students to acquire skills in two domains: artificial intelligence and control engineering. The feedback received from students and the author perception regarding student's enthusiasm in classes was quite positive. Students were quite surprised with the effectiveness obtained with the PSO algorithm. Moreover, it was clearly demonstrated by this experience that in the same way students progressed from a simple two decision variables function optimization problem to designing digital PID controllers, they can solve more complex control engineering and other domains problems.

REFERENCES

Åström K. J. and Hägglund T., (2004), Revisiting the Ziegler–Nichols step response method for PID control, Journal of Process Control, Vol. 14, pp. 635–650.

Bravo E., Amante B., Simo P. and Enache M., (2011). Video as a new teaching tool to increase student motivation. IEEE Global Engineering Education Conference (EDUCON), pp. 636-642.

Crook C. and Schofield L. (2017). The video lecture, The Internet and Higher Education, Elsevier, Vol. 34, pp. 56–64.

Halupa C. M. and Caldwell B. W., (2015), A Comparison of a Traditional Lecture-based and Online Supplemental Video and Lecture-Based Approach in an Engineering Statics Class, International Journal of Higher Education, Vol. 4, No. 1, pp. 232-240.

Kennedy, J. and Eberhart, R. C., (1995): Particle swarm optimization. In Proc. of the IEEE Int. conf. on neural networks IV, Piscataway, pp. 1942–1948.

Mercader P, Åström K. J., Banos A and Hägglund T., (2017), Robust PID Design Based on QFT and Convex–Concave Optimization, IEEE Transactions on control Systems Technology, Vol.. 25, No. 2, pp. 441-452.

Moura Oliveira P. B., (2005), Modern Heuristics Review for PID Control Systems Optimization: a Teaching Experiment, IEEE Int. Conf. Control and Automation, pp. 828-833.

Moura Oliveira P. B. and Boaventura J. C., (2016). Blending Artificial Intelligence into PID Controller Design: A Biomedical Engineering Experience, 11th IFAC Symposium on Advances in Control Education, 2016, Bratislava, Slovakia, IFAC-PapersOnLine 49-6, pp. 366–371.

Moura Oliveira P. B. and Boaventura J. C., (2017). Classroom Partial Flip for Feedback Control Systems: A Biomedical Engineering Experience, IEEE 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta, pp. 957-961.

Moura Oliveira P. B., (2018). PSO Introductory Video https://www.dropbox.com/s/y9x1wc6ai4cr78j/PSO_1_EN.mp4?dl=0, Retrieved in 22-1-2018.

O'Dwyer A., (2006), Handbook of PI and PID Controller Tuning Rules (2nd Edition), Imperial College Press, ISBN 1-86094-622-4.

Rossiter, J. A. (2013). Using online lectures to support student learning of control engineering, 10th IFAC Symposium Advances in Control Education The International Federation of Automatic Control, Sheffield, UK.

Rossiter, J. A. (2014). Lecture Flipping for control engineers, Proceedings of the 19th World IFAC Congress, South Africa. August, pp. 10592-10597.

Starr K., Bauer M and Horch A., (2015). An Industry Sponsored Video Course for Control Engineering Practitioners, IFAC-PapersOnLine 48-29, Internet Based Control Education - 3rd IBCE, Brescia, Italy, pp. 59–064.

Vrančić, D., Strmčnik, S. and Juričić, Đ., (2001). "A magnitude optimum multiple integration method for filtered PID controller", In: Automatica. Vol. 37, pp. 1473-1479

Ziegler J. G. and Nichols N. B. (1942), Optimum Settings for Automatic Controllers, Transaction of the ASME, pp. 759-768.

Holland J. H. (1975): Adaptation in Natural and Artificial Systems. The Univ. of Michigan Press, Ann Arbor.

Storn, R. and Price, K.V., (1995): Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI. March.