

## Article

# Optimal Identification and Metaheuristic PID Control of a Two-Tank System

Dan Stefanou  and Janetta Culita 

Faculty of Automatic Control and Computers, "Politehnica" University of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania; dan.stefanoiu@upb.ro

\* Correspondence: janetta.culita@upb.ro

**Abstract:** In the modern optimization context, this paper introduces an optimal PID-based control strategy for a two-tank installation, namely ASTANK2. The process model was identified by using raw and spline smoothed measured data, respectively. Two PID controller configurations, a standard (regular) one (PID-R) and a non-standard one (PID-N), were considered for each type of model, resulting in four regulators. The optimal tuning parameters of each regulator were obtained by a searching approach relying on a combination of two metaheuristics. Firstly, an improved version of the Hill Climbing algorithm was employed to comprehensively explore the searching space, aiming to find fairly accurate tuning parameters. Secondly, an improved version of the Firefly Algorithm was proposed to intensively refine the search around the previously found optimal parameters. A comparative analysis between the four controllers was achieved in terms of performance and robustness. The simulation results showed that all optimal controllers yielded good performance in the presence of exogenous stochastic noise (bounded error tracking, setpoint tracking, reduced overshoot, short settling time). Robustness analysis is extensive and illustrates that the PID-R controllers are more robust to model uncertainties, whilst PID-N controllers are more robust to tracking staircase type references.

**Keywords:** multivariable nonlinear system; PID control; Hill Climbing; Firefly Algorithm



**Citation:** Stefanou, D.; Culita, J. Optimal Identification and Metaheuristic PID Control of a Two-Tank System. *Electronics* **2021**, *10*, 1101. <https://doi.org/10.3390/electronics10091101>

Academic Editors: Dumitru Popescu, Haoping Wang, Severus C. Olteanu and Ciprian Lupu

Received: 14 April 2021

Accepted: 28 April 2021

Published: 7 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Although there is a variety of complex control strategies (such as model predictive, sliding mode, adaptive control, neural and fuzzy control [1]), in control engineering practice, the Proportional–Integral–Derivative (PID) control structures are the most widely used (over 95% from industrial applications [2]), owing to their simple structure, good stability [3], high reliability, reasonable closed loop performance and robustness [4–6].

The key problem in designing a PID controller ensuring these features is to tune the parameters accurately and efficiently. This is a challenging task when dealing with industrial processes that usually are modelled as multi-variable systems, with nonlinear dynamics and a large number of tunable variables [5]. Adjusting controller parameters in Multi-Input–Multi-Output (MIMO) process can become a difficult problem, especially in case of strong coupling of the output channels, for which Single-Input–Single-Output (SISO) decoupled loops might affect the performance [4].

Over the years, the PID tuning methods evolved from empirical (Ziegler–Nichols [7,8], Cohen–Coon) and analytical (internal model, pole placement, gain-phase margin [9]) to adaptive and optimization-based [5]. Unlike the empirical and analytical methods that provide weak or only satisfying performance, the optimization techniques applied to PID tuning yield good performance (disturbance rejection, set-point tracking) and robustness [4,10,11]. Therefore, in recent decades, the problem of designing an optimal PID controller that drives the systems into a desired behavior has increasingly captured the interest of the automatic control community.

To find the optimal PID controller, a multi-objective optimization problem with constraints has to be solved [4,12–14]. Although there is no clear methodology to define the cost function to optimize [12] (usually, to minimize), most of them rely on the tracking error variation (both on transient and steady-state zones in signal dynamics). For example, root mean square error (RMSE) and integral RMSE are basic minimization criteria, from which different variants have been derived, such as integral absolute error, integral square error, integral time square error [1], integral time weighted square error and integral time absolute error [3,6]. By default, minimizing these criteria guarantees the trade-off between of a set of desired but conflicting control performance indexes (e.g., overshoot, settling time, rise time, decay ratio, steady-state error) [3,15,16]. Moreover, in real applications, since the process outputs are affected by environmental and/or measure disturbances (exogenous noises) that are stochastic in nature, the performance indexes cannot be computed by means of closed form expressions. In this case, building the multi-objective cost function is not an easy task. A compromise between opposite desired performances has to be found. Usually, a weighted sum criterion, with some experimentally set weights, is defined for this aim [12].

Two main classes of optimization techniques are mostly referred to in the literature for optimal PID controller design. The first one is represented by traditional (exact) optimization methods, being applied in context of linear/nonlinear control theory. The second one is concerned with metaheuristics.

Interestingly, some methods usually employing state space representations can be adapted to find optimal PID controllers. For example, one can mention the methods to synthesize Linear Quadratic Regulators or H-infinity controllers (with the help of the Linear Matrix Inequality optimization procedure) [11,17]. This approach has been successfully employed to design multivariable and multiloop PID controllers implemented either in continuous time [10,11] or in discrete time [18]. Nevertheless, this approach involves an elaborated mathematical foundation and complex algorithms [12]. Additionally, the proposed solutions seem to be insufficiently tested for noisy output signals.

The second class deals with non-traditional optimization techniques, also referred to as metaheuristics, mainly based on Artificial Intelligence [14], which led to evolutionary computing. In the last two decades, metaheuristics has gained increasing attention in various research domains, including control engineering, due to its capacity to find optimal solutions (often close to global optima) in multidimensional search spaces [12], with reasonable computational burden and a good trade-off between multiple performance objectives [5,14]. Many metaheuristics use the swarm intelligence revealed by the natural (individual and social) behavior of some living entities (insects, animals) [9]. A wide range of metaheuristics has been used in optimal PID design, as reported in the literature [5]. For example, the following metaheuristics were preferred: the Genetic Algorithms [19] in [6,8,16,17,20]; Particle Swarm Optimization [19] in [8,13,21]; Ant Colony Optimization [19] in [6,9]; Artificial Bee Colony [19] in [3,13,21,22]; Differential Evolution [23] in [8]; Firefly Algorithm [19] in [4,15,24]; Cuckoo Search [25] in [2,24]; Bat Algorithm Optimization [19] in [13,22]; Symbiotic Organisms Search [26] in [9]; and Grey-Wolf Optimization [27] in [3]. In some of these references, two or more metaheuristics were combined.

Intensive research has been reported in the literature for tuning the parameters of PID or Fractional Order PID [1] controllers using metaheuristics in the genuine or improved versions. Additionally, numerous comparative studies highlight the superiority of a metaheuristic over the others [6,8,21,22,24] or over empirical tuning methods [2,9,16,21].

Most contributions deal with continuous time SISO PID controllers [6,9,12,13,15,21,22,24]. To the best of our knowledge, few contributions are concerned with designing optimal PID controllers for the real-world MIMO processes [4,9,20]. In this case, the plant constructive characteristics, the operating conditions and, moreover, the noises corrupting the useful signals impose significant restrictions. Moreover, real-world processes often fail to verify some sufficient hypotheses usually employed to prove robustness of designed controllers. This involves the necessity to test robustness for each controller. Generally, the control

strategy either is of the multi-loop PID type (decentralized) [4] or is based on decoupled SISO loops [3].

In this paper, two multivariable PID-based control strategies for a laboratory installation (ASTANK2 [28]) were designed and implemented starting from the identification models proposed in [29]. This resulted in two control strategies for each type of identification model. Based on the modern optimization context, to find the optimal PID parameters, a combination of two metaheuristics was proposed. The first is the Advanced Hill Climbing Algorithm (AHCA) [29], aiming to widely explore the search space and to find a sufficiently accurate optimal point. The second is an improved version of the Firefly Algorithm (FA) [19,30], employed to efficiently refine the exploitation around the previously found optimum point and to improve accuracy.

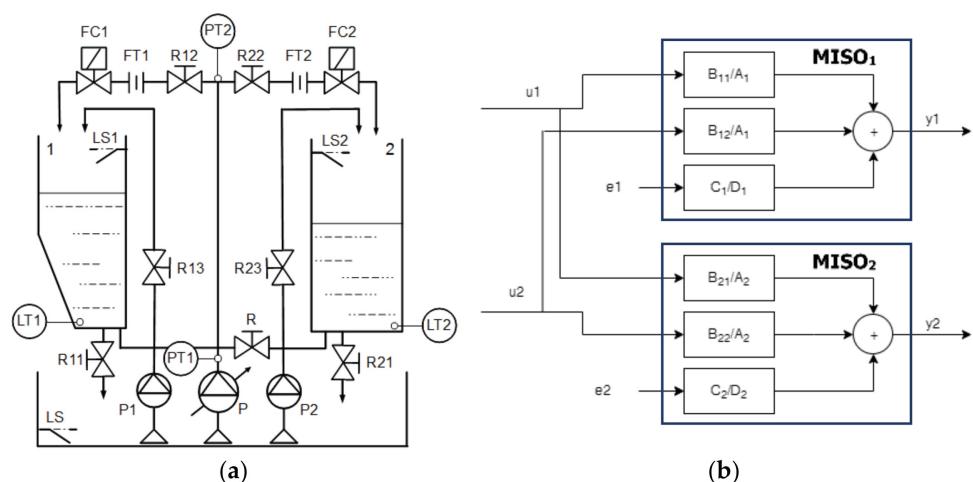
The article is structured as follows: Within the next section, the context of research development is described at length. Additionally, the optimization problem is formulated, and the methods to find optimal solutions, based on AHCA and improved FA metaheuristics, are proposed. Section 3 is entirely devoted to simulation results analysis. The investigations are directed towards three main directions: metaheuristics behavior, PID controllers performance for step references (such as bounded tracking error of step references, commands smoothness, short transient response, reduced overshoot, disturbance rejection) and robustness. The performance and robustness are comparatively analyzed. Four types of robustness are considered: to sudden change of references, to stochastic and constant disturbances, as well as to model uncertainties. To this aim, automatic procedures to define and detect fairly accurate robustness margins are introduced. The supplementary explanations are given in three appendices. Some concluding remarks, together with future developments are included into the last section of article.

## 2. Materials and Methods

### 2.1. Main Framework

The present study aims to design and implement a PID-based control strategy of a two-tank system, starting from the identification models determined in [29]. Actually, this research was already mentioned as possible future development in the end of that article.

The two-tank system, namely ASTANK2, is a laboratory installation, as illustrated in the functional diagram in Figure 1a.



**Figure 1.** (a) ASTANK2 block-diagram. (b) ASTANK2 global identification scheme.

The plant is composed of two main tanks (#1 and #2) located above an auxiliary storage reservoir, a main inverted-driven pump P (acting in the [0, 10] V range), two auxiliary on-off pumps (P1, P2) and a flexible pipeline network handled by seven manual taps (R11, R12, R13, R21, R22, R23, R). By acting the taps, different configurations of the installation can be set, such as serial/cascaded tank connection or independent tanks. The

water is continuously recycled from the storage reservoir by means of the main pump P through the main vertical pipe that forks at the upper side and feeds the upper tanks. The supply flows of both tanks can be controlled by means of the two electro-valves (e-v) FC1 and FC2 (the voltage of which can also vary in the [0, 10] V range). The water can be evacuated from the main tanks either by a communication horizontal pipeline (in cases of serial configuration) or directly in the storage tank through the emptying pipes (operated by the manual taps R11, R21). The water level data are acquired by means of transducers LT1 and LT2. Additionally, the feeding water flows can be monitored thanks to FT1 and FT2 transducers. A detailed description of ASTANK2 can be found in [28].

As explained in [29], the two-tank system can be represented as a nonlinear MIMO system with  $2 \times 2$  I/O channels. The inputs are the voltages  $u_1$  and  $u_2$  of the two e-v. The outputs are the water levels in the upper tanks,  $h_1$  and  $h_2$ , respectively. Note that the main pump voltage  $U$  is kept constant to protect the pump against sudden variations or shocks.

Despite the nonlinear dynamics, in [29], a linear MIMO Box-Jenkins (BJ) identification model (idmo) of ASTANK2 were determined and implemented for simulations. In order to identify the MIMO system, two Multi-Input Single-Output (MISO) systems corresponding to each output channel  $y_1$  and  $y_2$ , namely  $MISO_1$  and  $MISO_2$ , respectively, were identified, starting from the experimental data. A comprehensive block diagram of the global model identification is illustrated in Figure 1b.

Each MISO component model includes three additive filters. Each input ( $u_1, u_2$ ) stimulates two useful linear filters of B/A type. A noise linear filter of the C/D type modifies each output after processing two uncorrelated Gaussian white noises ( $e_1, e_2$ ). For each output channel  $y_j$  ( $j \in \{1, 2\}$ ), the MISO model relies on polynomials  $A_j, B_{j1}, B_{j2}, C_j$  and  $D_j$ , of degrees  $na_j, nb_{j1}, nb_{j2}, nc_j$  and  $nd_j$ , respectively, with intrinsic delays  $nk_{j1}, nk_{j2}$ , defined as:

$$\left\{ \begin{array}{l} A_j(q^{-1}) = 1 + a_{1,j}q^{-1} + \dots + a_{na_j,j}q^{-na_j} \\ B_{j1}(q^{-1}) = q^{1-nk_{j1}}(b_{1,j1}q^{-1} + b_{2,j1}q^{-2} \dots + b_{nb_{j1},j1}q^{-nb_{j1}}) \\ B_{j2}(q^{-1}) = q^{1-nk_{j2}}(b_{1,j2}q^{-1} + b_{2,j2}q^{-2} \dots + b_{nb_{j2},j2}q^{-nb_{j2}}) \\ C_j(q^{-1}) = 1 + c_{1,j}q^{-1} + \dots + c_{nc_j,j}q^{-nc_j} \\ D_j(q^{-1}) = 1 + d_{1,j}q^{-1} + \dots + d_{nd_j,j}q^{-nd_j} \end{array} \right. \quad (1)$$

Two types of experimental data were employed to identify the BJ models: raw (directly acquired from the sensors) and smoothed. The smoothed data were built from the raw data by using a least squares optimal stochastic cubic spline model [29]. In both cases, the optimal structure of the BJ models was obtained by means of a metaheuristic that maximizes the Signal-to-Noise Ratio (SNR). Table 1 illustrates the structural indices of each optimal MISO subsystem for both raw and smoothed data (whilst  $nk_{j1} = nk_{j2} = 1$ ), as well as the variances of both Gaussian white noises.

**Table 1.** Structural indices of optimal MISO BJ models and estimated noise variances.

Channel	Output Data Type	Optimal ( $na, nb_1, nb_2, nc, nd$ )	Noise Variance ( $\lambda^2$ )
1	Raw	(30,22,16,65,0)	0.0067
1	Spline Smoothed	(3,21,47,89,0)	0.006
2	Raw	(27,7,26,95,0)	0.006
2	Spline Smoothed	(2,2,46,95,0)	0.0063

The simulation results showed that when using smoothed data, the useful models have significantly fewer poles than when using raw data (see the first structural index,  $na$ ). Such models are expected to yield better performance in closed loop than the models identified from raw data.

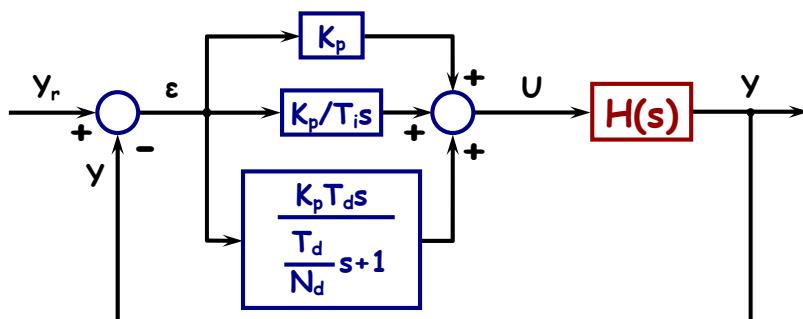
In addition to the idmo of ASTANK2, a static input/output (I/O) map was built experimentally. This map includes quadruples  $(u_{1,s}, u_{2,s}, y_{1,s}, y_{2,s})$  with the following meaning: if the tanks #1 and #2 need to be filled with water at constant levels  $y_{1,s}$  and  $y_{2,s}$ , respectively, then, in the absence of perturbations, the corresponding constant voltages of e-v FC1 and FC2 are  $u_{1,s}$  and  $u_{2,s}$ , respectively. This map is extremely useful in setting nominal functioning points of installation, around of which the dynamics of nonlinear installation are almost linear. Hence, if some reference water levels  $y_{1,\text{ref}}$  and  $y_{2,\text{ref}}$  are prescribed, then, from the static map, one can obtain the necessary voltages  $u_{1,\text{ref}}$  and  $u_{2,\text{ref}}$  to be applied on each e-v. If  $y_{1,\text{ref}}$  and/or  $y_{2,\text{ref}}$  do not correspond to the static points in the map, then the voltages  $u_{1,\text{ref}}$  and  $u_{2,\text{ref}}$  are obtained by linear interpolation. The static map includes 49 combinations of static water levels (7 for each tank), between 7 cm and 11 cm. This range was preferred to avoid water turbulences in tanks, which may provide wrong acquired data. Moreover, the linear models above were identified for the same level range.

The static map also revealed some constructive limitations of the ASTANK2 plant, to be considered as constraints in the design of any controller. The most important of them are as follows: (a) the e-v voltages are saturated at 2 V and 10 V (below 2 V, the e-v does not react any more); (b) the water levels in any tank can vary between 0 cm and 15 cm only; (c) to avoid hard saturation of voltages on e-v, the absolute difference between level references,  $|y_{1,\text{ref}} - y_{2,\text{ref}}|$ , is bounded by 2 cm; (d) to protect e-v against premature failure, any voltage cannot jump from the current value to the next one with more than  $\pm 3$  V.

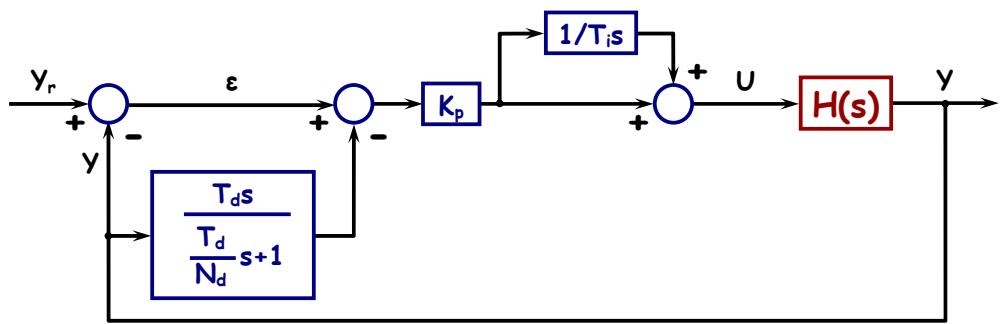
## 2.2. Selected PID Controllers and Closed Loop Configurations

The proposed control strategy relies on PID type controllers. Since the ASTANK2 plant is a multivariable system, two PID-based control configurations were designed and investigated (among many possible others) in order to meet certain performances, such as bounded tracking error of step references, smooth commands, short transient response, reduced overshoot, disturbance rejection, robustness against reference shocks, external perturbations and model uncertainties. An extended analysis of performance indicators is made in Section 3, devoted to simulation results. Note, however, that the optimal PID controllers were synthesized by means of a cost function, referred to as *fitness*, defined in the next subsection. The fitness does not consider all possible performance indicators, especially because many of them are defined for systems with smooth I/O signals. Alternatively, ASTANK2 installation works with noisy signals, as shown later.

The two selected PID are described by the schemes in Figures 2 and 3. They are depicted for SISO continuous time systems to allow for easy understanding. In both configurations,  $H$  is the transfer function of system to regulate,  $u$  is the input (the command),  $y$  is the output,  $y_r$  is the reference signal,  $\varepsilon \equiv y_r - y$  is the tracking error, whereas  $K_p > 0$  (the proportionality constant),  $T_i > 0$  (the integral constant),  $T_d \geq 0$  (the derivative constant) and  $N_d > 0$  (the derivative divisor) are parameters of the PID controller.



**Figure 2.** Regular, standard Proportional–Integral–Derivative(PID) structure (PID-R).



**Figure 3.** Non-standard PID structure (PID-N).

The derivative component is filtered by a first order system to yield physical realizability. The derivative divisor  $N_d$  usually is an integer at least equal to 10. Thus, the time constant of filter ( $T_d/N_d$ ) is small enough (i.e., its bandwidth is large enough) to minimally distort the derivative component of the command. Here and hereafter,  $N_d$  was set to 10.

The first PID structure, as seen in Figure 2, is a regular (standard) one, being abbreviated by PID-R. The characteristic transfer function of the PID-R controller, from tracking error  $\epsilon$  to command  $u$ , is:

$$H_{\text{PID-R}}(s) = K_p \left[ 1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N_d} s + 1} \right] = K_p \frac{T_i T_d (N_d + 1)s^2 + (T_d + N_d T_i)s + N_d}{T_d T_i s^2 + N_d T_i s}. \quad (2)$$

The second structure, illustrated in Figure 3, is non-standard and was considered for its main feature; the derivative component of the command is furthermore filtered by the other two components (P, I), which results in quite smooth commands (with as few spikes as possible). The acronym for this structure is PID-N, and the corresponding transfer function is:

$$H_{\text{PID-N}}(s) = K_p \frac{T_i s + 1}{T_i s} \left[ 1 - \frac{T_d (N_d + 1)s + N_d}{T_d s + N_d} \right]. \quad (3)$$

Unlike PID-R, the transfer function of PID-N is of the MISO type, with two inputs: the reference signal  $y_r$  and the system output  $y$ . This allows for working with two regulation blocks of the SISO type: one on feedback,  $H_R$  (the "pure" regulator), and another one in front of the system,  $H_S$  (the "sensitive" controller), providing the command  $u$ :

$$H_{\text{PID-N}}(s) = H_S(s) \left[ 1 - H_R(s) \right], \text{ with : } \begin{cases} H_S(s) = K_p \frac{T_i s + 1}{T_i s} \\ H_R(s) = \frac{T_d (N_d + 1)s + N_d}{T_d s + N_d} \end{cases} \quad (4)$$

Since the idmo of the ASTANK2 plant is discrete, it is suitable to discretize the PID regulators. A fairly accurate discretizing method is the bilinear Tustin method, in which  $s \simeq \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}}$ , for some sampling period  $T_s > 0$ . After discretization, the transfer functions (2) and (4) become:

$$H_{\text{PID-R}}(s) \rightleftharpoons H_{\text{PID-R}}(z) = \frac{Q(z)}{P(z)} \text{ and } H_{\text{PID-N}}(s) \rightleftharpoons H_{\text{PID-N}}(z) = \frac{Q_S(z)}{P_S(z)} \left[ 1 - \frac{Q_R(z)}{P_R(z)} \right], \quad (5)$$

where  $Q, P, Q_S, P_S, Q_R, P_R$  are polynomials of the second degree at most. Their coefficients are expressed as follows, depending on the controller type:

- P controllers ( $T_i = \infty, T_d = 0$ ):

$$\begin{cases} Q_S(z) = Q(z) = K_p \\ P_S(z) = P(z) = 1 \end{cases}, \quad \begin{cases} Q_R(z) = 1 \\ P_R(z) = 1 \end{cases} \quad (6)$$

2. PI controllers ( $T_d = 0$ ):

$$\begin{cases} Q(z) = Q_S(z) = K_p[(T_s + 2T_i)z + (T_s - 2T_i)] \\ P(z) = P_S(z) = 2T_i(z - 1) \end{cases}; \begin{cases} Q_R(z) = 1 \\ P_R(z) = 1 \end{cases}. \quad (7)$$

3. PD controllers ( $T_i = \infty$ ):

$$\begin{cases} Q(z) = K_p Q_R(z) = K_p \{[N_d T_s + 2(N_d + 1)T_d]z + [N_d T_s - 2(N_d + 1)T_d]\} \\ P(z) = P_R(z) = (N_d T_s + 2T_d)z + (N_d T_s - 2T_d) \end{cases}; \begin{cases} Q_S(z) = K_p \\ P_S(z) = 1 \end{cases}. \quad (8)$$

4. PID controllers (general case):  $\begin{cases} Q(z) = q_2 z^2 + q_1 z + q_0 \\ P(z) = p_2 z^2 + p_1 z + p_0 \end{cases}$ , with:

$$\begin{cases} q_2 = K_p \left[ 2 \frac{(N_d + 1)T_i T_d}{N_d T_s} + \frac{T_s}{2} + \frac{T_d}{N_d} + T_i \right] \\ q_1 = K_p \left[ T_s - 4 \frac{(N_d + 1)T_i T_d}{N_d T_s} \right] \\ q_0 = K_p \left[ 2 \frac{(N_d + 1)T_i T_d}{N_d T_s} + \frac{T_s}{2} - \frac{T_d}{N_d} - T_i \right] \end{cases}; \begin{cases} p_2 = 2 \frac{T_i T_d}{N_d T_s} + T_i \\ p_1 = -4 \frac{T_i T_d}{N_d T_s} \\ p_0 = 2 \frac{T_i T_d}{N_d T_s} - T_i \end{cases}; \quad (9)$$

$$\begin{cases} Q_S(z) = K_p [(T_s + 2T_i)z + (T_s - 2T_i)] \\ P_S(z) = 2T_i(z - 1) \\ Q_R(z) = [N_d T_s + 2(N_d + 1)T_d]z + [N_d T_s - 2(N_d + 1)T_d] \\ P_R(z) = (N_d T_s + 2T_d)z + (N_d T_s - 2T_d) \end{cases}. \quad (10)$$

For ASTANK2 installations, the selected closed loop configurations are displayed in Figure 4, for the two PID structures. There are 4 characteristic polynomials for the configuration in Figure 4a:  $Q_1, P_1, Q_2$  and  $P_2$ . Additionally, in Figure 4b, 8 polynomials have to be set:  $Q_{S,1}, P_{S,1}, Q_{R,1}, P_{R,1}, Q_{S,2}, P_{S,2}, Q_{R,2}$  and  $P_{R,2}$ . Their coefficients are determined by the 6 parameters of continuous time PID controllers, namely  $K_{p,1}, T_{i,1}, T_{d,1}$  (for the first channel), and  $K_{p,2}, T_{i,2}, T_{d,2}$  (for the second channel). To compute them, Equations (6)–(10) can be employed. Each PID controller is fed with the output and/or the tracking error of one channel, but provides a command that applies to both MISO blocks.

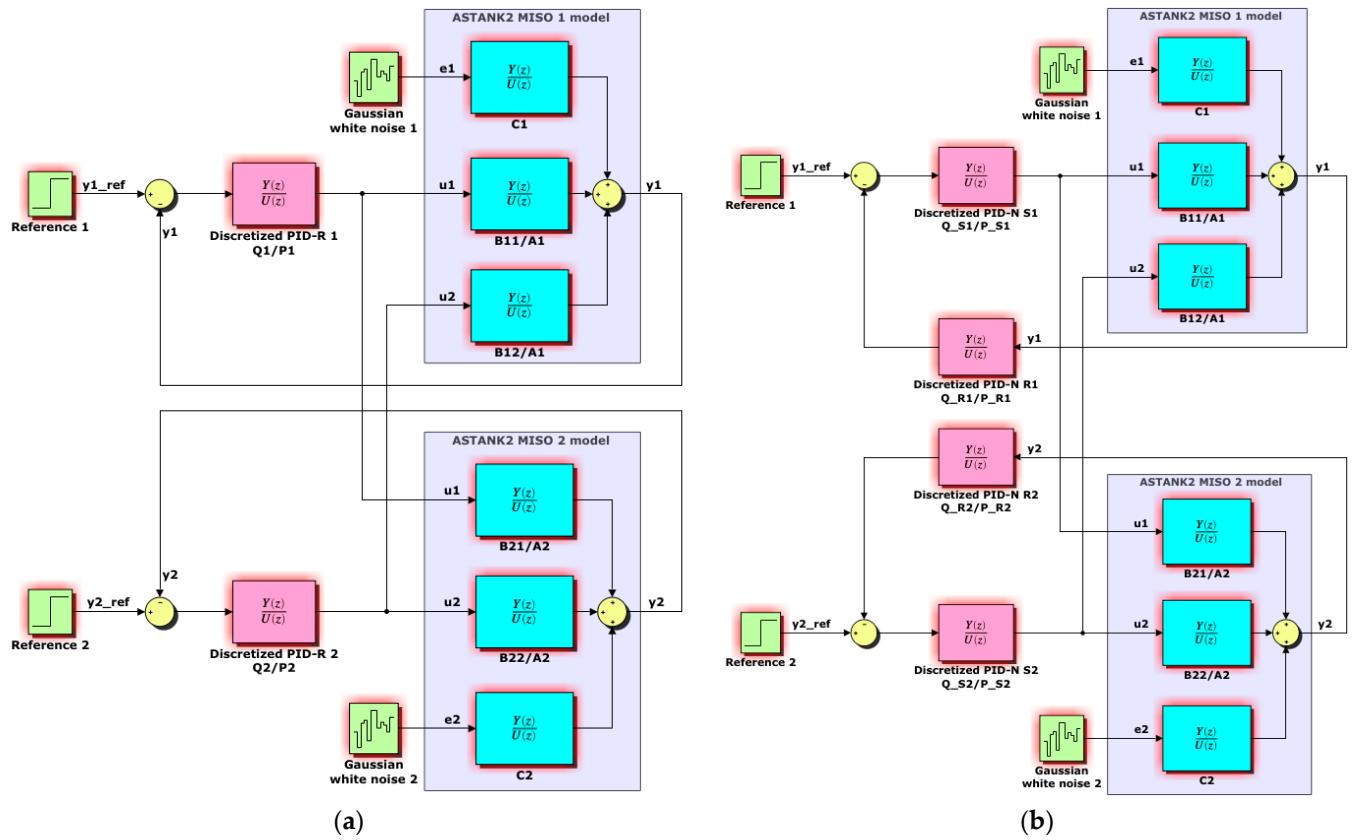
### 2.3. Optimization Problem Statement and Cost Function Definition

One seeks to find the optimal parameters of the two PID controllers by optimizing a cost function. There is an extensive discussion in the literature concerning the choice of such a cost function. Most approaches rely on the RMSE applied on the tracking error and the norm of command. Both have to be minimized. Usually, they are combined in a cost function with some weights in their original or modified versions. Basically, the cost function defined in this article, referred to as *fitness*, also employs the RMSE, but with some refinements adapted to ASTANK2 dynamics.

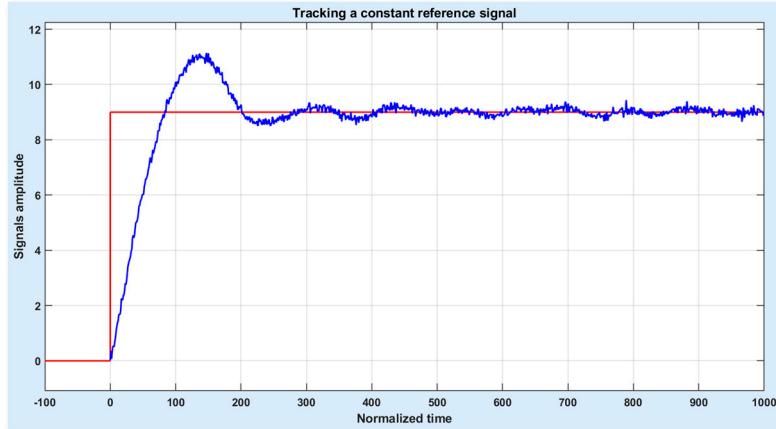
Take, for example, the output signal  $y$  of Figure 5, tracking some reference  $y_{\text{ref}}$ . The global RMSE is then:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N |y[n] - y_{\text{ref}}[n]|^2}, \quad (11)$$

where  $N \in \mathbb{N}^*$  is the number of acquired output samples. Evidently, RMSE is a cost function to minimize. Using definition (11) in the context of the ASTANK2 plant revealed several disadvantages. The most important comes from the fact the tracking error can have similar values on transient and steady-state zones, which can lead to output signals that oscillate too much around references, while RMSE takes relatively small values. Thus, separating the two zones is necessary. Since the output signal is corrupted by stochastic noises, no analytical formula can be employed to compute the transient time,  $N_t$ . Therefore,  $N_t$  has to be estimated by using statistical tools.



**Figure 4.** Closed loop PID structures for ASTANK2 installation with (a) PID-R; and (b) with PID-N.



**Figure 5.** Example of noisy output signal tracking the prescribed reference.

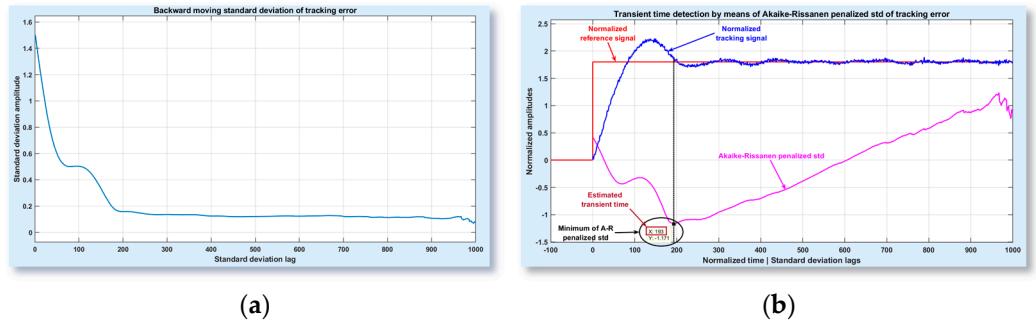
Let  $\varepsilon \equiv y - y_{\text{ref}}$  be the tracking error. A key parameter in estimating  $N_t$  is the backward moving standard deviation (std) of tracking error:

$$\sigma_{\varepsilon}^b[k] = \sqrt{\frac{1}{k+1} \sum_{n=N-k}^N |\varepsilon[n] - \mu_{\varepsilon}^b[k]|^2}, \quad \forall k \in \overline{1, N-1}, \quad (12)$$

where  $\mu_{\varepsilon}^b$  is the backward moving average:

$$\mu_\varepsilon^b[k] = \frac{1}{k+1} \sum_{n=N-k}^N \varepsilon[n], \quad \forall k \in \overline{1, N-1}. \quad (13)$$

For the signal in Figure 5,  $\sigma_{\varepsilon}^b$  is displayed in Figure 6a. At the right side end, the std of final tracking error is computed, whereas the larger values obtained at the left side end correspond to the first values of tracking error.



**Figure 6.** (a) Backward std variation. (b) Detection of transient time by means of A-R std.

Actually, the variation above was horizontally flipped, as, according to the computing formula, the first values of std are the ones that lie at the right side end. Normally, the transient time is given at the left entry of the flat variation in the graphic. However, finding this entry is rather difficult, because, as one can easily notice, any of the lags between 180 and 300 can play this role. Therefore, a penalty can be applied to the std, aiming to better isolate the transient time. Among several penalties known in the literature, the Akaike-Rissanen (A-R) criterion seems to be suitable to the variation above. The A-R penalized std is then:

$$\sigma_{\varepsilon}^{b,A-C}[k] = \log(\sigma_{\varepsilon}^b[k]) + \frac{k}{2N} \log(N), \quad \forall k \in \overline{1, N-1}. \quad (14)$$

In Figure 6b, the penalized A-R std is overlapped on normalized signal and reference variations (in range [0, 2]). One can notice how easy it now is to find the transient time (the minimum point of the penalized std) and how well this value corresponds to the subjective selection performed by visual inspection of signal variation.

In the framework of the ASTANK2 plant, the two output signals  $y_1$  and  $y_2$  are tracking the references  $y_{1,\text{ref}}$  and  $y_{2,\text{ref}}$ , with errors  $\varepsilon_1^y$  and  $\varepsilon_2^y$ , respectively. To define the fitness, the two zones of each signal dynamics are processed separately. Let  $N_{t,1}$  and  $N_{t,2}$  be the two transient times detected on output signals by using the procedure above. For the transient zone, the fitness can be defined as follows:

$$\varphi_{t,j}^y \equiv \frac{100}{1 + \sigma \sqrt{\frac{1}{N_{t,j}} \sum_{n=1}^{N_{t,j}} |\varepsilon_j^y[n]|^2}} [\%], \quad \forall j \in \{1, 2\}, \quad (15)$$

where  $\sigma > 0$  is a factor that allows for scaling of the fitness. For example, the function  $100/(1+x)$  (on which the definition (15) relies) takes the value of 58.82% in  $x = 0.7$ ; by using the scaling factor  $\sigma = 0.5$ , i.e.,  $\sigma x$  instead of  $x$ , the value can be increased to 74.07%. Assuming both output signals have the same length,  $N$ , for the steady-state zone, the fitness is:

$$\varphi_{s,j}^y \equiv \frac{100}{1 + \sigma \sqrt{\frac{1}{N-N_{t,j}} \sum_{n=N_{t,j}+1}^N |\varepsilon_j^y[n]|^2}} [\%], \quad \forall j \in \{1, 2\}. \quad (16)$$

Now, it is natural to consider that each fitness above has a weight proportional to the duration of its zone, which leads to the fitness of each output channel:

$$\phi_j^y \equiv \frac{N_{t,j} \varphi_{t,j}^y + (N - N_{t,j}) \varphi_{s,j}^y}{N}, \forall j \in \{1, 2\}. \quad (17)$$

In the case of commands  $u_1$  and  $u_2$ , the references  $u_{1,\text{ref}}$  and  $u_{2,\text{ref}}$  are computed from the static map, depending on references  $y_{1,\text{ref}}$  and  $y_{2,\text{ref}}$ , as explained in the end of Section 2.2. Two zones in command variation are considered as well, with transient times  $M_{t,1}$  and  $M_{t,2}$ . Unlike for outputs, the two commands should be analyzed as a whole, since they are simultaneously applied to both MISO models that produce the outputs; therefore, on the transient zones, a unique fitness is defined by means of tracking errors  $\varepsilon_1^u$  and  $\varepsilon_2^u$ :

$$\varphi_{t,12}^u \equiv \frac{100}{1 + \sigma \sqrt{\frac{1}{M_{t,1} + M_{t,2}} \left( \sum_{n=1}^{M_{t,1}} |\varepsilon_1^u[n]|^2 + \sum_{n=1}^{M_{t,2}} |\varepsilon_2^u[n]|^2 \right)}} [\%]. \quad (18)$$

Similarly, the fitness on the steady-state zone is:

$$\varphi_{s,12}^u \equiv \frac{100}{1 + \sigma \sqrt{\frac{1}{2N - M_{t,1} - M_{t,2}} \left( \sum_{n=M_{t,1}+1}^N |\varepsilon_1^u[n]|^2 + \sum_{n=M_{t,2}+1}^N |\varepsilon_2^u[n]|^2 \right)}} [\%]. \quad (19)$$

Finally, the fitness of both input channels is a weighted average of (18) and (19):

$$\phi_{12}^u \equiv \frac{(M_{t,1} + M_{t,2}) \varphi_{t,12}^u + (2N - M_{t,1} - M_{t,2}) \varphi_{s,12}^u}{2N}. \quad (20)$$

The fitness definitions (17) and (20) can be combined into a weighted average, to obtain the overall fitness of PID-ASTANK2 scheme:

$$\Phi \equiv \alpha (\phi_1^y + \phi_2^y) + \beta \phi_{1,2}^u [\%], \quad (21)$$

where  $2\alpha + \beta = 1$ . For example,  $\alpha = 0.35$  and  $\beta = 0.3$ . The fitness of output signals has the same weight, as no channel is more important than the other. Obviously, the fitness is a cost function to maximize.

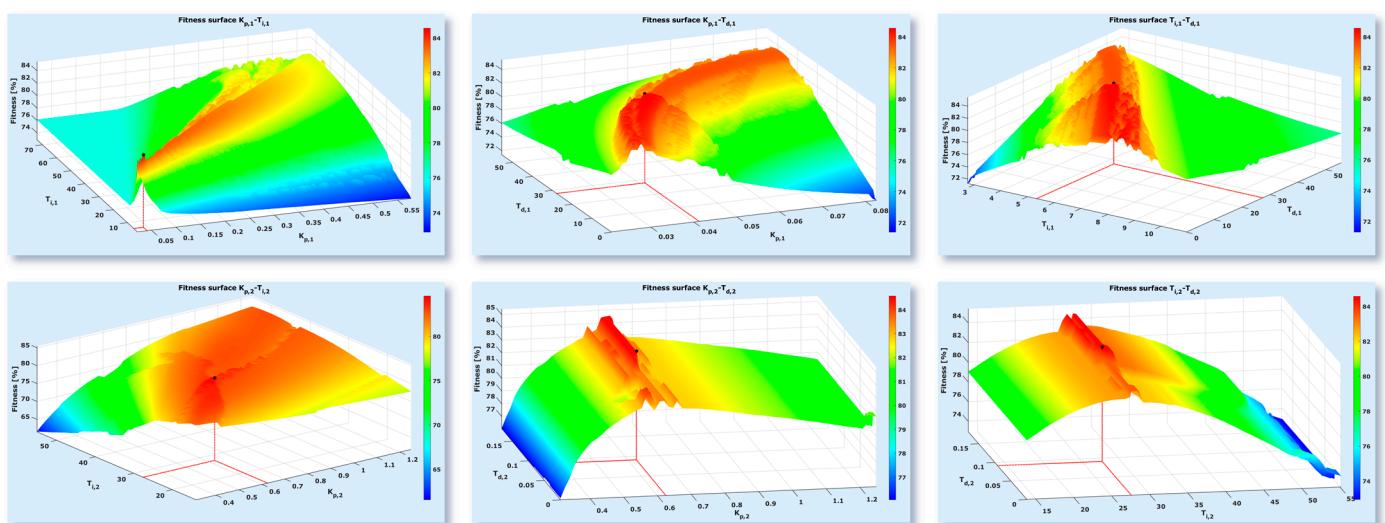
Definition (21) has an important drawback. Theoretically, a certain value of fitness,  $\Phi_0$ , can be obtained by an infinity of triplets  $\{\phi_1^y, \phi_2^y, \phi_{1,2}^u\}$ . Thus, one of the fitness functions from the triplet can be small, whilst, in compensation, the other two are large. Usually, this happens when  $\phi_{1,2}^u$  is small (the commands having big oscillations) and  $\phi_1^y$  and  $\phi_2^y$  are large (tracking of references being very fast, with small overshoot or even without overshoot). This means the e-v are worked hard in order to fasten the reference tracking. Or, such a situation is to be avoided. It is more suitable to have smoother commands, even though the references are tracked slower and/or with some overshoot. To deal with this ambiguity, the characteristics of a closed loop system can be taken into consideration. For an ASTANK2 installation, simulations have shown that good fitness values are above 75% (for definition (21) and the scaling factor  $\sigma = 0.5$ ). In this case, the difference between  $\phi_1^y$  and  $\phi_2^y$  is less than 10% ( $|\phi_1^y - \phi_2^y| < 0.1$ ). Moreover, commands fitness exhibit two behaviors: (a) if  $\phi_{12}^u < \frac{\phi_1^y + \phi_2^y}{2}$ , then the overall fitness can take big values, although the commands are oscillating too much (increasing the risk of e-v premature wearing out or failure); (b) if, on the contrary,  $\phi_{12}^u \geq \frac{\phi_1^y + \phi_2^y}{2}$  (or, even better, if  $\phi_{12}^u \geq \max\{\phi_1^y, \phi_2^y\}$ ), then the commands are quite smooth, and the overall fitness can take large values. Obviously,

the last case is suitable for finding optimal controllers. In order to drive the search towards smooth commands and large overall fitness values, definition (21) can be refined:

$$\Phi \equiv \begin{cases} \beta \frac{\Phi_{12}^u + \max\{\Phi_1^y, \Phi_2^y\}}{2} + 2\alpha \min\{\Phi_1^y, \Phi_2^y\} & , \min\{\Phi_1^y, \Phi_2^y\} < 75 \\ \beta \frac{\Phi_1^y + \Phi_2^y}{2} + 2\alpha \Phi_{12}^u & , \min\{\Phi_1^y, \Phi_2^y\} \geq 75 \text{ & } \Phi_{12}^u < \frac{\Phi_1^y + \Phi_2^y}{2}, \\ \alpha(\Phi_1^y + \Phi_2^y) + \beta \Phi_{12}^u & , \min\{\Phi_1^y, \Phi_2^y\} \geq 75 \text{ & } \Phi_{12}^u \geq \frac{\Phi_1^y + \Phi_2^y}{2} \end{cases} \quad (22)$$

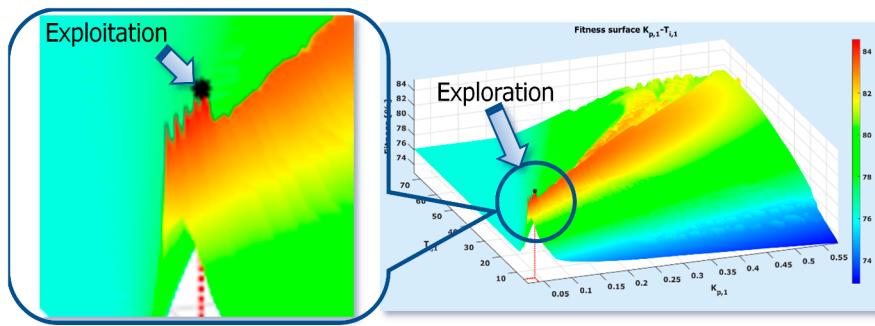
where  $\alpha \in (0.25, 0.5)$ . Practically, in definition (22), the former definition (21) is only employed to find optimal regulators with smooth commands—see the third case. For the other two cases, the term that produce fitness ambiguity is now multiplied by large weight (of at least 0.5), which results in smaller overall fitness values.

The fitness depends on the 6 parameters of continuous time PID controllers. Obviously, it is not possible to visually observe the characteristics of fitness in this case. Nevertheless, 6 fitness surfaces can be drawn, by varying couples of parameters, while keeping constant the other four. Such surfaces are depicted in Figure 7.



**Figure 7.** Example of fitness surfaces.

The stochastic nature of the I/O measured data induces irregular variations, with many local extremes of fitness. A mutual characteristic of all surfaces above is the alternance between smooth slopes and fractal heights or ridges, where the maximum point lies. Solving the maximization problem becomes a difficult task that cannot be accomplished by using exact methods [31]. Instead, a suitable alternative to this problem is adopting evolutionary computing techniques, referred to as metaheuristics [19]. In this article, two metaheuristics are employed: one to widely explore the search space and another one to exploit some narrow zones where the maximum seems to exist. Figure 8 suggests the search strategy. For exploration, the AHCA from [29] was employed. Its main task is to find an optimum with fair accuracy, while exploring large zones of search space. Since in case of the 6 optimal PID parameters the upper bound is unknown and the fitness exhibits smooth slopes where the gradient is not oscillating, AHCA fits very well. Once the climbers have found some optima, a vicinity around them can be isolated. Now the search can be performed within a bounded zone. To exploit this zone, the FA was selected (and improved), due to its simplicity and effectiveness. Whilst the climbers are strongly dependent on the local gradient of fitness, the fireflies can easily jump from one peak to another, regardless of the gradient (which, anyway, may oscillate too much—see the magnification on the left side of Figure 8).



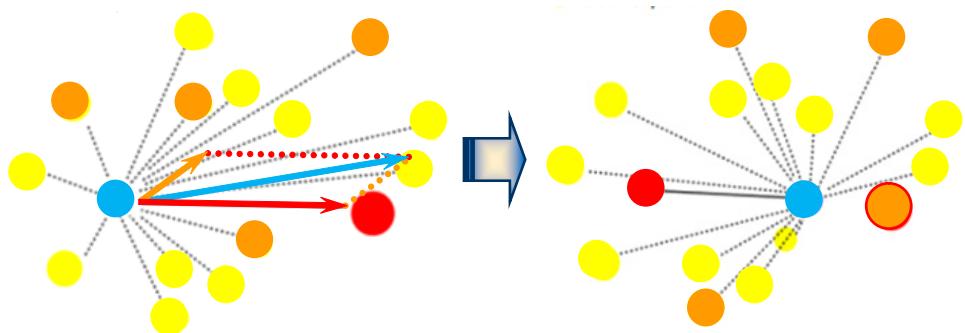
**Figure 8.** Exploration–exploitation strategy in fitness maximization.

It has to be outlined that both the exploration and the exploitation phases are performed by varying all 6 parameters (and not only couples of them, as Figure 7 might suggest). The improved FA (IFA) is described next.

#### 2.4. Improved Firefly Algorithm

The FA [19,30] was inspired by the principle of mutual attraction between insects and the phenomenon of natural attenuation of light with distance. The fireflies are small flying beetles, usually luminescent at twilight, that produce a cold light to attract mates or prey. They are grouped into swarms and, apparently, fly at random. In its flight, a firefly is tempted to fly towards other shinier fireflies in the swarm, being attracted by the light signals the latter are emitting. In this way, the shinier fireflies become targets for the current one. The strongest target is the shiniest firefly, which attracts the entire swarm. Thus, the swarm is more and more grouped around the firefly emitting the most intense light. Nevertheless, the shiniest firefly flies as well and their light can fade. In this case, another firefly becomes the shiniest. Even though the swarm was grouped around a firefly, it can move away towards another shinier firefly. In this way, there is a good balance between the convergence and the diversity of firefly swarms.

Figure 9 illustrates a swarm of fireflies and its evolution from current to next positions. The current firefly is colored in blue, and the shiniest firefly is in red, whereas the other fireflies are in orange (the shinier ones) or in yellow (the less shining ones).



**Figure 9.** Principle of the Firefly Algorithm.

The perceived light attractiveness depends on the brilliance of the casting beetle. The brighter a firefly, the more fireflies it attracts. The current firefly is motivated to fly towards the shiniest one along the arrow in red (see the left side of Figure 9). Nevertheless, while tracking the target, our firefly can perceive lights from other fireflies, especially from the shinier ones, which constitute attraction points as well. Hence, the other fireflies in the swarm, especially the ones in orange, could disturb the flight along the direction colored in orange as well. By composing the two directions (in red and in orange), the final direction, in blue, is taken by the current firefly. Meanwhile, the swarm moves to new positions and changes light intensities, as suggested in the right side of Figure 9. It is possible that

the shiniest firefly decreases its light intensity, whilst another firefly becomes the shiniest. All fireflies can change not only in position, but in light intensity as well. The swarm permanently tries to group around the shiniest firefly, but this becomes possible only if no other firefly becomes shinier. Otherwise, the swarm can oscillate between several positions, until a best position (of maximum light intensity) is found, eventually.

This behavior can be simulated by means of a numerical algorithm based on the following recursive equation expressing how the position of an artificial firefly changes:

$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^k + \sum_{q \in \mathbf{S}_p} \beta_{p,q}^k (\mathbf{x}_q^k - \mathbf{x}_p^k) + \eta^k \xi^k(\mathbf{S}), \quad \forall p \in \overline{1, P}, \quad \forall k \in \mathbb{N}. \quad (23)$$

The notations in Equation (23) are explained next. Assume the swarm includes  $P \in \mathbb{N}^*$  fireflies. Each firefly  $p \in \overline{1, P}$  takes a position  $\mathbf{x}_p^k$  at instant  $k \in \mathbb{N}$ , into the search space  $\mathbf{S} \subseteq \mathbb{R}^{nx}$ , where  $nx \in \mathbb{N}^*$  is the length of position vector. (For example, in the case of PID-ASTANK2 configurations,  $nx = 6$ .) The search space can be specified as a cartesian product of compact intervals:

$$\mathbf{S} = [x_{\min,1}, x_{\max,1}] \times [x_{\min,2}, x_{\max,2}] \times \cdots \times [x_{\min,nx}, x_{\max,nx}] = \prod_{i=1}^{nx} [x_{\min,i}, x_{\max,i}]. \quad (24)$$

To compute the next position, the firefly will take  $(\mathbf{x}_p^{k+1})$ , and the current position is affected by two disturbances: one produced by the shinier fireflies than the current one (including the shiniest one), referred to as *diversity disturbance*, and another one generated by the entire swarm—the *convergence disturbance*. The sum term in Equation (23) stands for the diversity disturbance, where all the shinier fireflies than the current one are grouped into the set  $\mathbf{S}_p$ . This disturbance actually contributes to swarm diversity and is expressed as a linear combination of attraction directions from the current firefly to its shinier mates,  $\{\mathbf{x}_q^k - \mathbf{x}_p^k\}_{q \in \mathbf{S}_p}$ . Thus, the current firefly is not exclusively attracted by the shiniest firefly, but by the other shinier fireflies as well. This phenomenon keeps the diversity of fireflies' swarms to some extent. The corresponding coefficients,  $\{\beta_{p,q}^k\}_{q \in \mathbf{S}_p}$ , are light intensities perceived by the current firefly from the shinier ones. According to Biology, each perceived intensity can be computed as follows:

$$\beta_{p,q}^k = I_0 \exp \left[ - \frac{\|\mathbf{x}_p^k - \mathbf{x}_q^k\|}{\mathbf{D}(\mathbf{S})} \right], \quad \forall p \in \overline{1, P}, \quad \forall q \in \mathbf{S}_p, \quad \forall k \in \mathbb{N}, \quad (25)$$

where  $I_0$  is a biological constant standing for the maximum light intensity a firefly can produce (usually set to unit) and  $\mathbf{D}(\mathbf{S})$  is the diameter of search space, defined by:

$$\mathbf{D}(\mathbf{S}) = \sup_{\mathbf{x}, \mathbf{y} \in \mathbf{S}} \{\|\mathbf{x} - \mathbf{y}\|\} = \sqrt{\sum_{i=1}^{nx} (x_{\min,i} - x_{\max,i})^2}. \quad (26)$$

Definition (25) shows that the perceived light intensity is higher if the fireflies are closer to each other. However, the current firefly also can be strongly attracted by a distant firefly radiating an intense light. This case is not taken into consideration in definition (25). Therefore, a first improvement consists of including the radiated light intensity into the perceived intensity formula. Let  $\Phi_p^k$  be the light intensity radiated by firefly  $p \in \overline{1, P}$  at instant  $k \in \mathbb{N}$ . Then, the intensity of light the other fireflies perceive should increase as  $\Phi_p^k$  increases and the distance from the current firefly decreases. A second improvement concerns the diameter. Normally, the perceived intensity should adapt itself to the space

the entire swarm of fireflies is taking at current instant, denoted by  $\mathbf{S}^k$ . Obviously,  $\mathbf{S}^k \subseteq \mathbf{S}$  and then:

$$\mathbf{S}^k = \prod_{i=1}^{nx} \left[ \min_{p \in \overline{1, P}} \{x_{p,i}^k\}, \max_{p \in \overline{1, P}} \{x_{p,i}^k\} \right] = \prod_{i=1}^{nx} [x_{\min,i}^k, x_{\max,i}^k]; \mathbf{D}(\mathbf{S}^k) = \sqrt{\sum_{i=1}^{nx} (x_{\max,i}^k - x_{\min,i}^k)^2}. \quad (27)$$

With the two improvements above, definition (25) can be replaced by:

$$\beta_{p,q}^k = I_0 \exp \left[ -\frac{\|\mathbf{x}_p^k - \mathbf{x}_q^k\|}{\mathbf{D}(\mathbf{S}^k)} \frac{\Phi_p^k}{\Phi_q^k} \right], \forall p \in \overline{1, P}, \forall q \in \mathbf{S}_p, \forall k \in \mathbb{N}, \quad (28)$$

where, evidently,  $\Phi_q^k \geq \Phi_p^k, \forall q \in \mathbf{S}_p$ .

The last term of Equation (23) expresses the second disturbance, generated by the entire swarm at each instant  $k \in \mathbb{N}$ . The disturbance intensity is given by the scalar factor  $\eta^k$ , uniformly selected at random in the interval  $[-1, +1]$ . Its sign encodes whether the disturbance is direct (+1) or opposite (-1). The other factor,  $\xi^k(\mathbf{S}) \in \mathbb{R}^{nx}$ , stands for the absolute disturbance vector generated by the swarm. According to biologists, each component of this vector,  $\xi_i^k(\mathbf{S}) (i \in \overline{1, nx})$ , is generated at random as well, but not uniformly. This time, the random numbers generator uses Lévy's (probability) distribution:

$$p(u) = u^{-\lambda}, \forall u \in \mathbb{R}_+^*, \quad (29)$$

where  $\lambda \in [0, 3]$  is preset. Usually,  $\lambda = 1.5$ . To generate numbers with prescribed probability distribution, Baker's algorithms can be implemented and run [19,32]. The Lévy's distribution is assigned to each component of search space, i.e., to each interval  $[x_{\min,i}, x_{\max,i}] (i \in \overline{1, nx})$ . Thus,  $\xi_i^k(\mathbf{S}) \in [x_{\min,i}, x_{\max,i}] (i \in \overline{1, nx})$ , which explains why the vector  $\xi^k$  depends on the search space  $\mathbf{S}$ . However, during the run of IFA, this definition revealed two caveats: (a) Lévy's distribution over  $[x_{\min,i}, x_{\max,i}]$  yields generating the most values in the vicinity of  $x_{\min,i}$ , although there is no reason the fireflies should be directed to the lower bound of the search space; and (b) as the swarm concentrates around a shiniest firefly, the disturbing direction becomes unrealistic, because the vector  $\xi^k$  does not belong anymore to the space the fireflies are moving into, i.e.,  $\mathbf{S}^k$ . Therefore, two more improvements of basic FA were made: (a) Lévy's distribution is centered around the *light center* of the swarm; and (b) instead of using  $\mathbf{S}$  to generate the components of vector  $\xi^k$ , the current search space  $\mathbf{S}^k$  is employed. The *light center* can be determined by using the radiated light of each firefly:

$$\mathbf{x}_c^k = \frac{\sum_{p=1}^P \Phi_p^k \mathbf{x}_p^k}{\sum_{p=1}^P \Phi_p^k}, \forall k \in \mathbb{N}. \quad (30)$$

Now Lévy's distribution is defined as follows, for each component  $i \in \overline{1, nx}$ :

$$p(u) = |u - x_{c,i}^k|^{-\lambda}, \forall u \in [x_{\min,i}^k, x_{\max,i}^k], \quad (31)$$

Hence, in definition (23), the factor  $\xi^k(\mathbf{S})$  is replaced by  $\xi^k(\mathbf{S}^k)$ . The product  $\eta^k \xi^k(\mathbf{S}^k)$  is a disturbance that ensures the swarm convergence to some extent.

After being computed, the next position  $\mathbf{x}_p^{k+1}$  may need to be made viable. This means bringing back all outsider components of  $\mathbf{x}_p^{k+1}$  inside the corresponding min–max interval,

according to the initial search space definition (24). Assume  $x_{p,i}^{k+1} \notin [x_{\min,i}, x_{\max,i}]$  for some  $i \in \overline{1, nx}$ . Then, the integer below can be computed:

$$n_{p,i}^{k+1} = \left\lfloor \frac{x_{p,i}^{k+1} - x_{\min,i}}{x_{\max,i} - x_{\min,i}} \right\rfloor, \text{ if } x_{p,i}^{k+1} > x_{\max,i} \text{ or } n_{p,i}^{k+1} = \left\lceil \frac{x_{\max,i} - x_{p,i}^{k+1}}{x_{\max,i} - x_{\min,i}} \right\rceil, \text{ if } x_{p,i}^{k+1} < x_{\min,i}. \quad (32)$$

Consequently,  $x_{p,i}^{k+1}$  can be replaced by the new component below:

$$x_{p,i}^{k+1} \leftarrow \begin{cases} x_{p,i}^{k+1} - n_{p,i}^{k+1}(x_{\max,i} - x_{\min,i}) & , x_{p,i}^{k+1} > x_{\max,i} \\ x_{p,i}^{k+1} + n_{p,i}^{k+1}(x_{\max,i} - x_{\min,i}) & , x_{p,i}^{k+1} < x_{\min,i} \end{cases}. \quad (33)$$

It is easy to prove that the new component belongs now to the corresponding min-max interval.

The notation for radiated light intensity is not selected at random. In fact,  $\Phi$  is the cost function to maximize among the fireflies' swarm during the search (and will stand for fitness, in case of PID controllers optimization).

The numerical procedure of IFA (the Algorithm 1) is summarized next.

---

**Algorithm 1.** Main steps of IFA procedure

---

- 1 **Input data:**
    - cost function to maximize,  $\Phi$  (associated with light intensity eradicated by fireflies);
    - search space boundaries (see Equation (24));
    - configuring parameters: swarm size  $P$ , specific light intensity  $I_0$ , Lévy's constant  $\lambda$ , maximum number of iterations  $K$ , surviving (aging) factor  $S$ .
  - 2 **Initialization.** Spread the fireflies approximately uniform over the search space  $S$  and evaluate their performance (by using  $\Phi$ ). Save the shiniest firefly.
  - 3 **Main loop.** While the number of iterations is less than  $K$  and the shiniest firefly survived over less than  $S$  iterations, do the following:
    - 3.1. Evaluate the current search space  $S^k$  and its diameter using Equation (27).
    - 3.2. Evaluate the mass left of swarm, by using Equation (30).
    - 3.3. For each firefly in the swarm:
      - 3.3.1. For each shinier firefly:
        - 3.3.1.1. Evaluate the perceived light intensity by using Equation (28).
        - 3.3.1.2. Compute the corresponding disturbance, according to the second term of Equation (23).
      - 3.3.2. Update the current position by adding all disturbances of shinier fireflies.
      - 3.3.3. Set the swarm disturbance intensity at random, uniformly, in  $[-1, +1]$ .
      - 3.3.4. Use Baker's algorithm to generate the absolute swarm disturbing direction with Lévy's distribution (31).
      - 3.3.5. Update the current position by adding the convergence disturbance, according to the last term of Equation (23).
      - 3.3.6. Make viable the new position by using Equations (32) and (33).
    - 3.4. Update the swarm performance and determine the new shiniest firefly.
    - 3.5. If the current shiniest firefly still is shinier than the current one, increment the surviving counter.
    - 3.6. Otherwise, replace the current shiniest firefly by the new one and reset the surviving factor.
    - 3.7. Increment the number of iterations.
  - 4 **Output data:**
    - optimal value of  $\Phi$  and position of the shiniest firefly;
    - performance parameters of IFA: total number of iterations, surviving counter.
- 

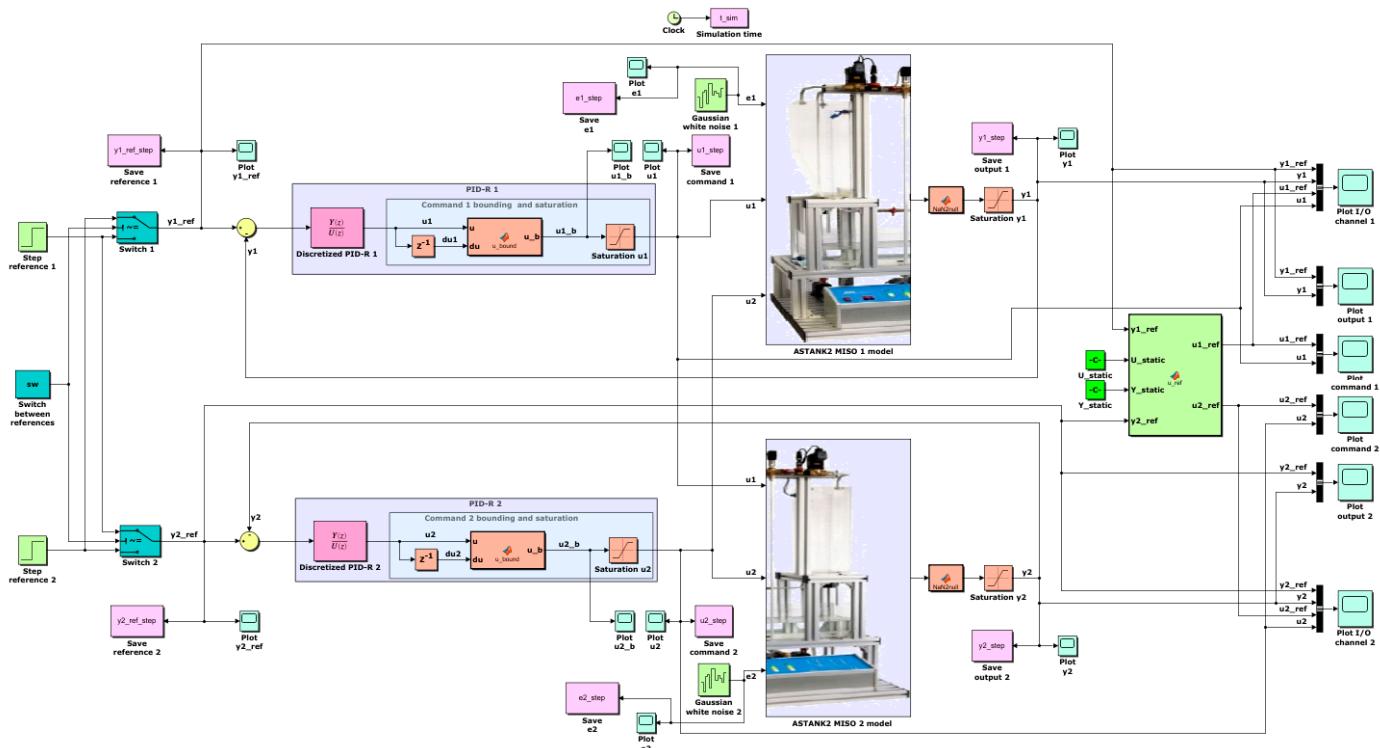
As in cases of many metaheuristics, there are no theoretical results proving the IFA is convergent. Therefore, to terminate the numerical procedure, the stop tests of the main loop in Algorithm 1 are necessary. Additionally, the position of the current shiniest firefly changes only because of the convergence disturbance, and its light can fade.

### 3. Analysis of Simulation Results

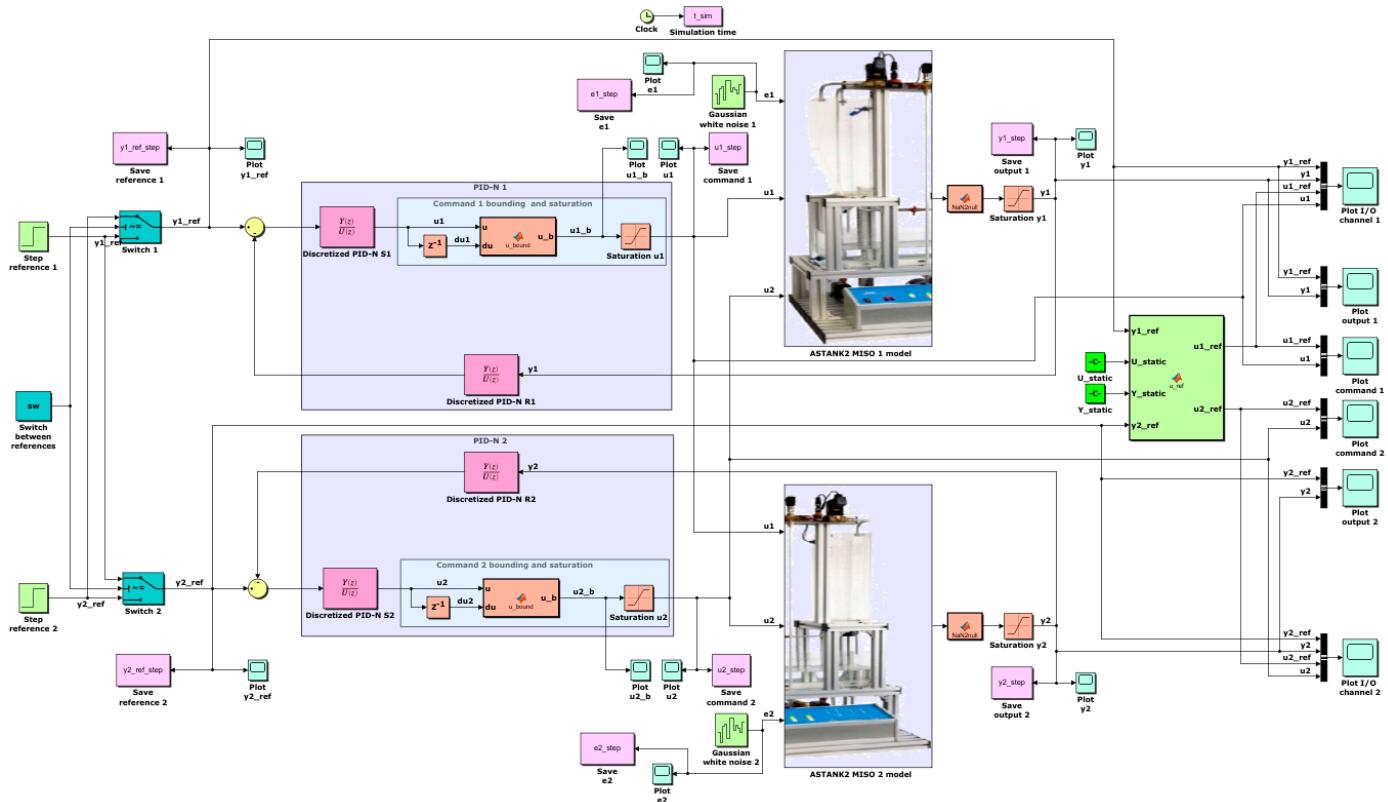
#### 3.1. Simulations Setting

The simulations were performed within the MATLAB-SIMULINK™ environment. The two closed loop configurations of Figure 4 were implemented as SIMULINK models, as depicted in Figures 10 and 11. Unlike the schemes in Figure 4, the SIMULINK models include the characteristic limitations of ASTANK2 plant (enumerated in the end of Section 2.1). The output references are of two types: a step for optimal PID design, and a staircase for robustness analysis. In Figures 10 and 11, the step references are set as follows:  $y_{1,\text{ref}} = 8.5 \text{ V}$ ,  $y_{2,\text{ref}} = 9.5 \text{ V}$ . They can be switched, as displayed in the left side of both schemes (see the block **sw**). Additionally, the sampling period was set to unit ( $T_s = 1 \text{ s}$ ) in all simulations. Since ASTANK2 has a slow dynamic, the simulation duration was set to  $T = 1000T_s = 1000 \text{ s}$  for step references. Longer durations are employed in cases of other references types.

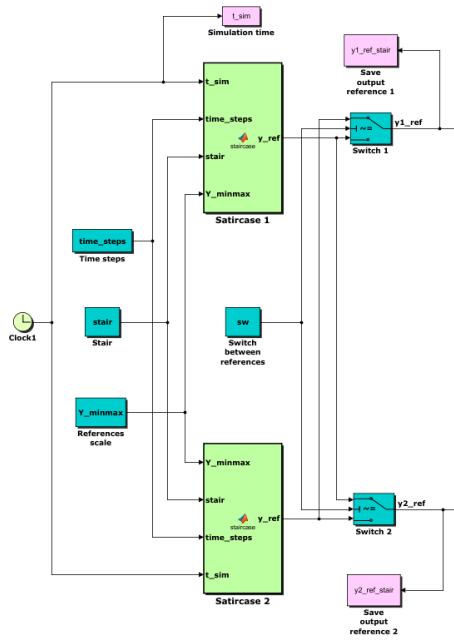
In the case of staircase references, the input of each scheme is changed to the one in Figure 12. The staircases can be designed by the user within an M-script with two variables: one giving the height of each step (**stair**), and another one setting the time instants of each step (**time\_steps**). The staircases have to comply with ASTANK2 limitations of outputs (no more than 2 cm difference between steps on different channels).



**Figure 10.** Closed loop PID-R-ASTANK2 SIMULINK model.



**Figure 11.** Closed loop PID-N-ASTANK2 SIMULINK model.



**Figure 12.** Staircase type inputs.

Additionally, the staircases can be scaled in a suitable interval  $[Y_{\min}, Y_{\max}]$  (specified by means of block  $Y_{\minmax}$ ). By default, the scaling interval is [7, 11] cm.

To the right side of schemes in Figures 10 and 11, the characteristic static map of the ASTANK2 plant with linear interpolation is implemented (see the block  $u_{\text{ref}}$ ). Thus, the command references are available during the simulation, depending on output references  $y_{1,\text{ref}}$  and  $y_{2,\text{ref}}$ .

The Gaussian white noises are generated by means of corresponding blocks on each channel. Their variances are set according to the last column of Table 1, where the estimations resulted after linear identification of the ASTANK 2 plant.

Many signals are displayed in both schemes. Nevertheless, when simulation is employed to evaluate the regulators fitness, signals plotting is suppressed in order to speed up the search.

For the two metaheuristics (AHCA and IFA), the configuring parameters are listed in Table 2, where, beside  $P$ ,  $K$ ,  $S$ ,  $I_0$ ,  $\lambda$ , the following notations are used:  $M$  is the maximum number of mutations per climber,  $\delta$  is the climber's altitude relative accuracy,  $P_e$  is the number of elite climbers (see [29] for details of AHCA),  $\rho_{\min}$  is the relative lower bound of the search space,  $\rho_{\max}$  is the relative upper bound of the search space.

**Table 2.** Configuring parameters of metaheuristics.

AHCA—Exploration						IFA—Exploitation						
$P$	$K$	$S$	$M$	$\delta$	$P_e$	$P$	$K$	$S$	$I_0$	$\lambda$	$\rho_{\min}$	$\rho_{\max}$
40	$30P$	30	30	1%	3	25	100	30	1	1.5	0.5	2

The last two notations are useful to confine the search space for IFA. Assume AHCA provided a sufficiently accurate optimal solution,  $x_o^{\text{AHCA}}$ . Then the boundaries of search space for fireflies are:  $x_{\min} = \rho_{\min} x_o^{\text{AHCA}}$  and  $x_{\max} = \rho_{\max} x_o^{\text{AHCA}}$ .

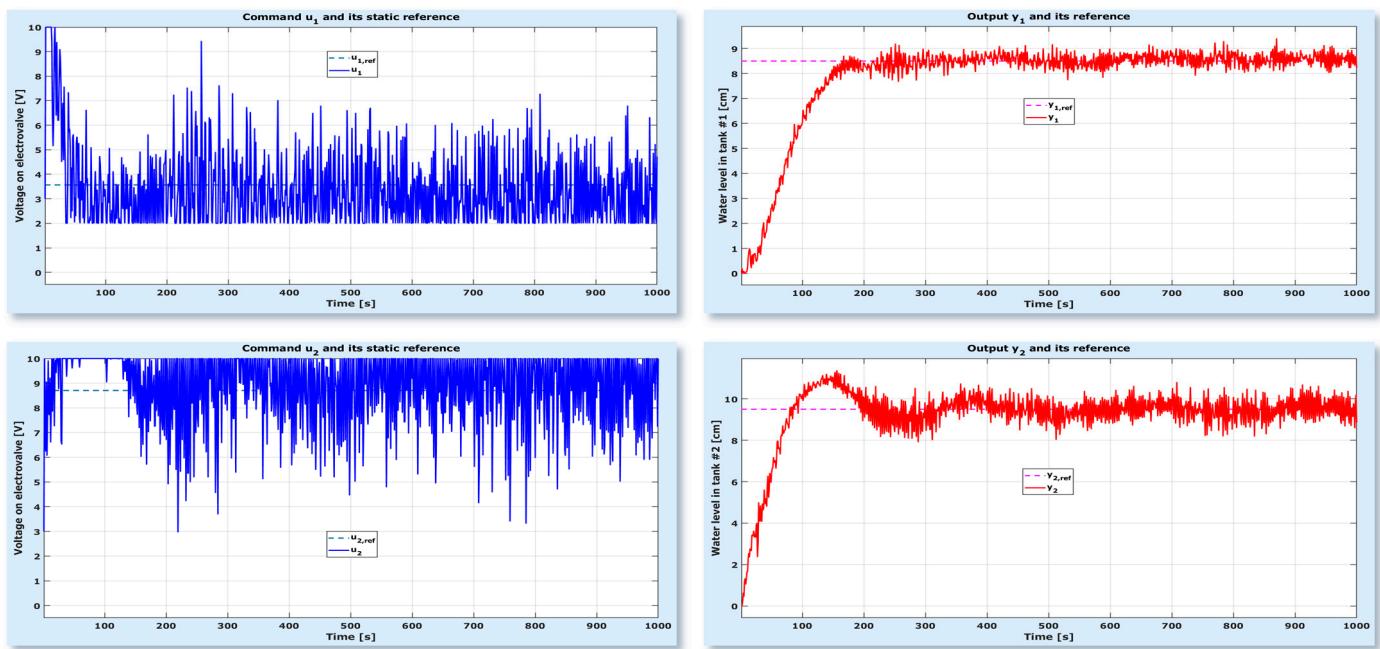
### 3.2. Metaheuristic Optimization Results

Before running AHCA and IFA, the PID-R regulators were tuned by means of the well-known Ziegler–Nichols empirical method [7]. The resulting parameters of PID-R controllers are listed in Table 3 (where  $K_u$  is the critical gain that pushes the system to stability limit, with oscillations of period  $T_u$ ).

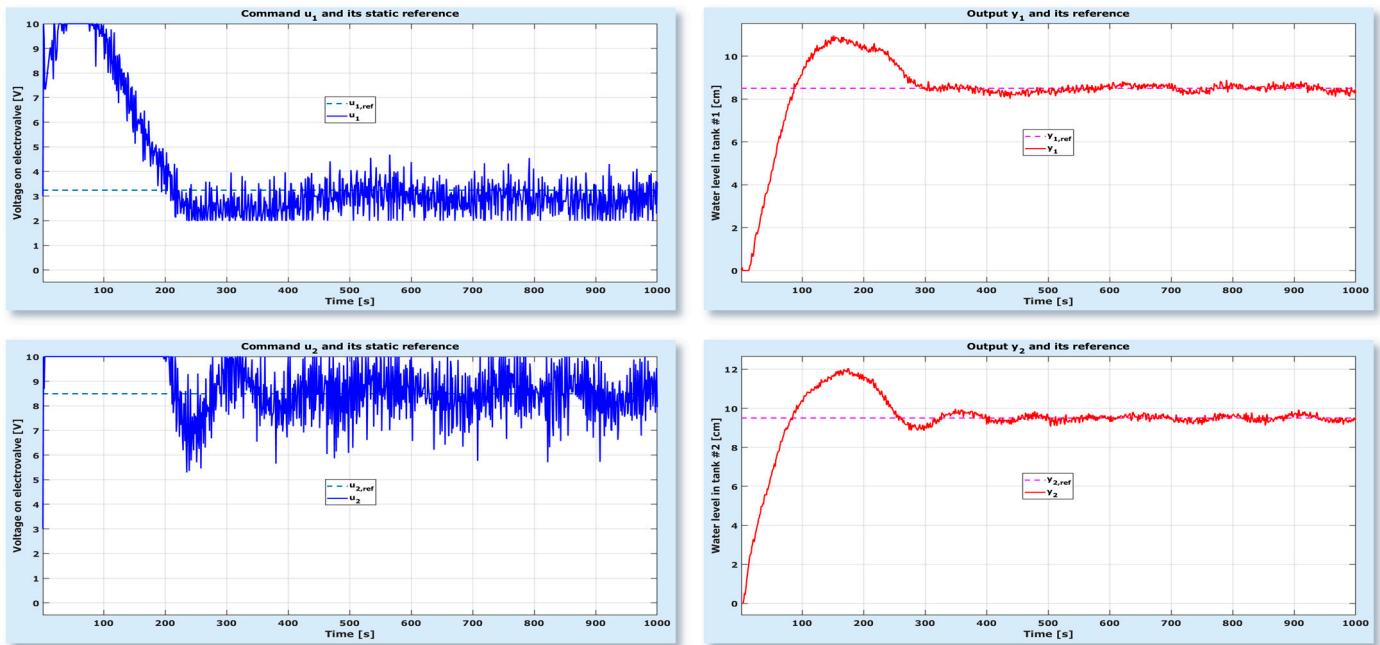
**Table 3.** Parameters of PID-R controllers tuned by the Ziegler–Nichols method.

Identification Model	Channel	$K_u$	$T_u$ [s]	$K_p$	$T_i$ [s]	$T_d$ [s]	$\Phi$ [%]
<raw>	1	1.3851	355.31	0.83106	177.66	44.41	<b>70.12</b>
	2	1.2519	40.83	0.75114	20.41	5.10	
<spline>	1	1.233625	57.09	0.740175	28.54	7.14	<b>73.12</b>
	2	2.479675	28.15	1.487805	14.07	3.52	

After simulation with step references, the I/O signals of PID-R controllers are displayed in Figure 13 (<raw> idmo) Figure 14 (<spline> idmo). Channels 1 and 2 are represented on top and at bottom of figures, respectively. Additionally, variation of commands and outputs are depicted to the left side and right side of figures, respectively. The commands are represented in blue (left side) and the outputs in red (right side). All references are tracked in both cases, but the I/O signals are smoother for <spline> idmo than for <raw> idmo. This is proven by the corresponding fitness values in Table 3 as well (73.12% versus 70.12%). It is well known that Ziegler–Nichols method usually yields oscillating commands, but the variations for <raw> idmo are unacceptable (Figure 13). In the case of <spline> idmo (Figure 14), while the first command can be marginally accepted, the second command is still too variable. The results after running the two metaheuristics are now analyzed. In Table 4, some running performances of AHCA and IFA are given, such as the final number of iterations,  $k$ ; the surviving counter of best climber/firefly,  $ns$ ; the total number of mutations applied to climbers,  $m$ ; the total number of fitness evaluations  $n_\Phi$ ; and the optimal fitness,  $\Phi_o$ .



**Figure 13.** I/O signals for Ziegler–Nichols tuned PID-R controller, in case of <raw> idmo,  $\Phi^{ZN} = 70.12\%$ .



**Figure 14.** I/O signals for Ziegler–Nichols tuned PID-R controller, in case of <spline> idmo,  $\Phi^{ZN} = 73.12\%$ .

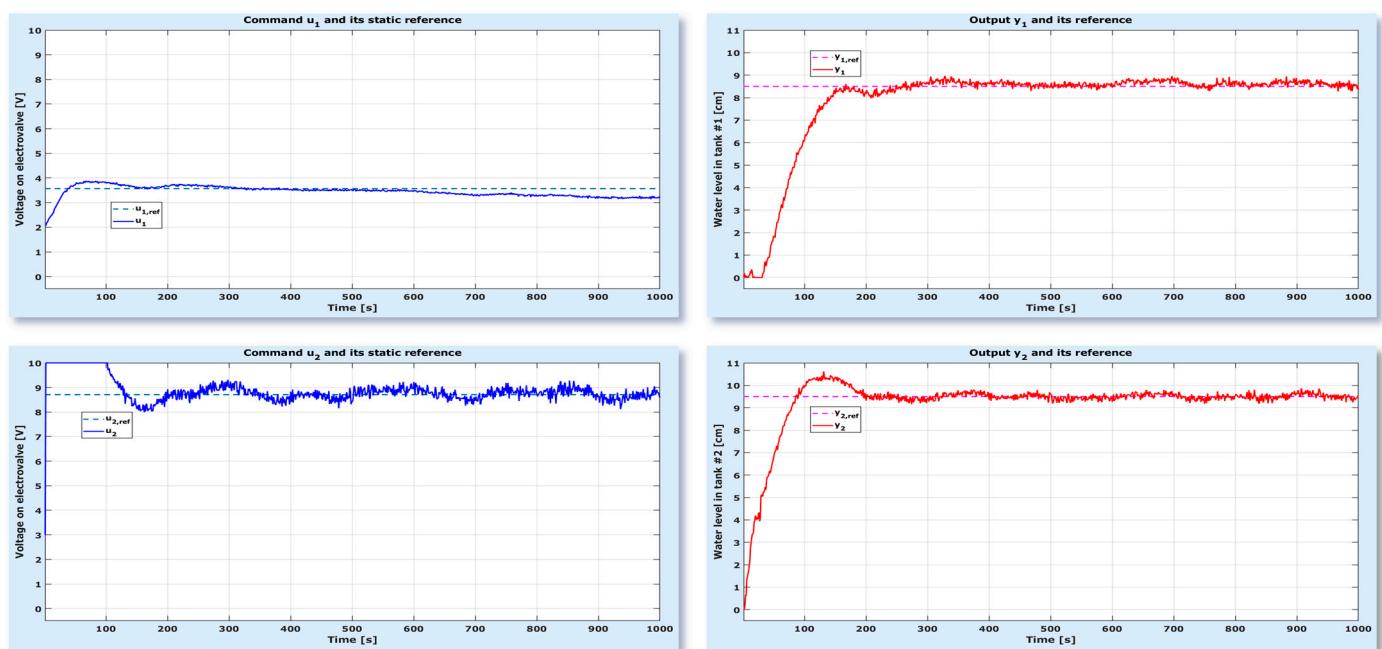
**Table 4.** Running performances of metaheuristics.

AHCA—Exploration							IFA—Exploitation						
PID	Identification Model	$k$	$ns$	$m$	$n_{\Phi}$	$\Phi_0$ [%]	PID	Identification Model	$k$	$ns$	$n_{\Phi}$	$\Phi_0$ [%]	
R	<raw>	64	30	4565	5421	80.03	R	<raw>	96	30	28,595	83.90	
	<spline>	61	30	5143	6127	83.08		<spline>	100	29	29,799	85.01	
N	<raw>	62	30	4192	4982	81.68	N	<raw>	67	30	19,866	83.84	
	<spline>	53	30	4484	5322	83.50		<spline>	66	30	19,565	85.07	

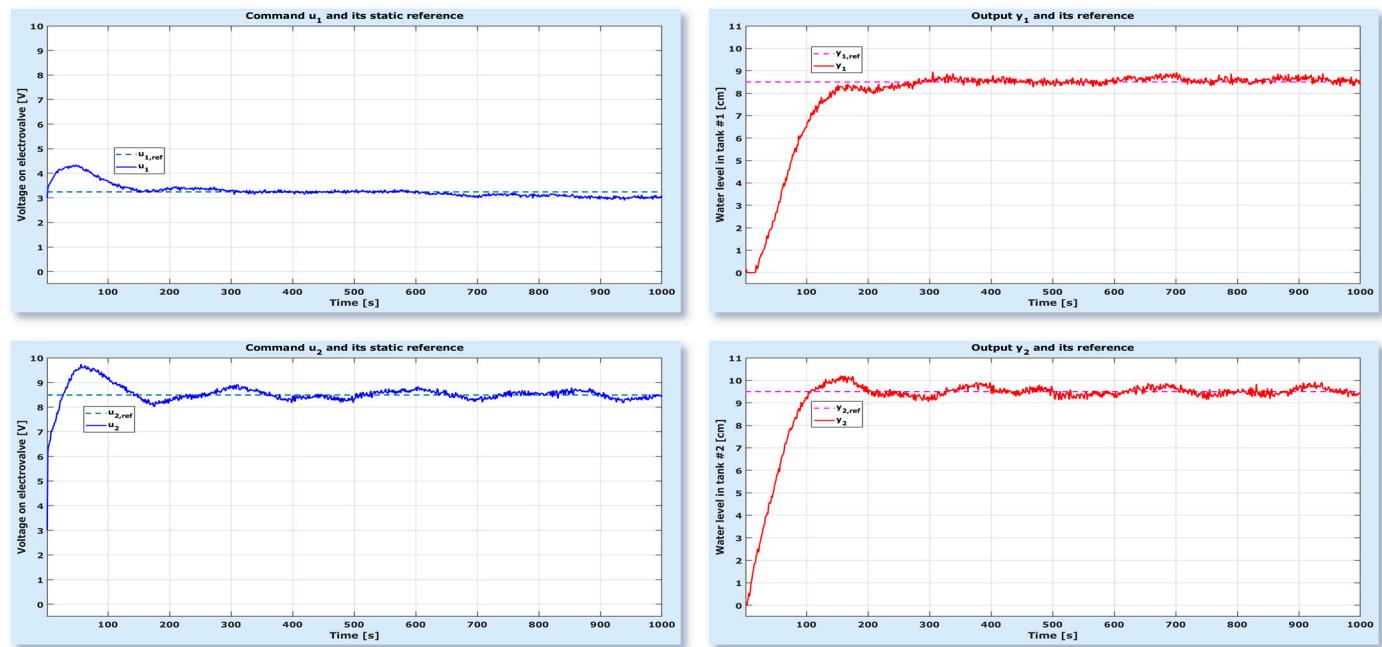
The algorithms were run on a regular computer at current technological levels. Usually, the exploration by AHCA takes about 20 min, whereas the exploitation by IFA is about 90 min long. (The optimal PID-N controllers were found to be a few minutes faster than the optimal PID-R controllers.) The difference between the two durations is due to the fact that the climbers are only employed to find the most promising zone where the optimum could exist. Inside that zone, as the fitness rather exhibits fractal variations, the climbers cannot use their compasses based on local gradient evaluation and, therefore, AHCA usually oscillates. Fireflies do not use the gradient to fly, but IFA is not as efficient as AHCA. Consequently, the configuring parameters were set to allow longer search durations and, thus, to increase the optimal point accuracy. The number of fitness evaluations,  $n_{\Phi}$ , is much higher for IFA than for AHCA, which means the search for optimum is more intense in the case of fireflies than in the case of climbers. One can see that both algorithms complete the search after the best optimal climber or firefly has survived at maximum (see parameters  $S$  and counters  $ns$  in Tables 2 and 4, respectively) and not after maximally iterating the positions updating (see parameters  $K$  and counters  $k$ , respectively). There is one exception; the optimal PID-R for <spline> idmo was found with a maximum number of iterations (100), but the surviving factor was almost maximum too (29 instead of 30). For AHCA, an interesting parameter is  $m$ , with values starting from 4192. Mutations are applied to blocked climbers on local peaks, for which the fitness cannot increase by using the gradient. The purpose of mutation is to throw the climber on another slope of fitness hyper-surface in the hope that, from there, it can continue to climb. Good exploration of search space is thus ensured. This short analysis proves that, very likely, the best local optimum was found by combining the two metaheuristics.

Looking again at Table 4, one can see that the optimal fitness of climbers is smaller than the optimal fitness of fireflies. Nevertheless, the difference is not so big. Furthermore, this difference can barely be observed when comparing the corresponding I/O signals in Figures 15–18. (The signals are arranged exactly like in the case of Figures 13 and 14.) Additionally, the optimal PID controllers for <spline> idmo are slightly fitter than the optimal ones for <raw> idmo, as expected. However, one can notice that both PID-N and PID-R controllers exhibit almost the same fitness for the same idmo, which does not meet the initial expectations. Normally, the commands the PID-N controller provides should have been smoother than the commands of the PID-R controller. Or, the command smoothness is almost the same for both PID controllers. One can notice, though, a slight improvement for PID-N commands in the case of <spline> idmo.

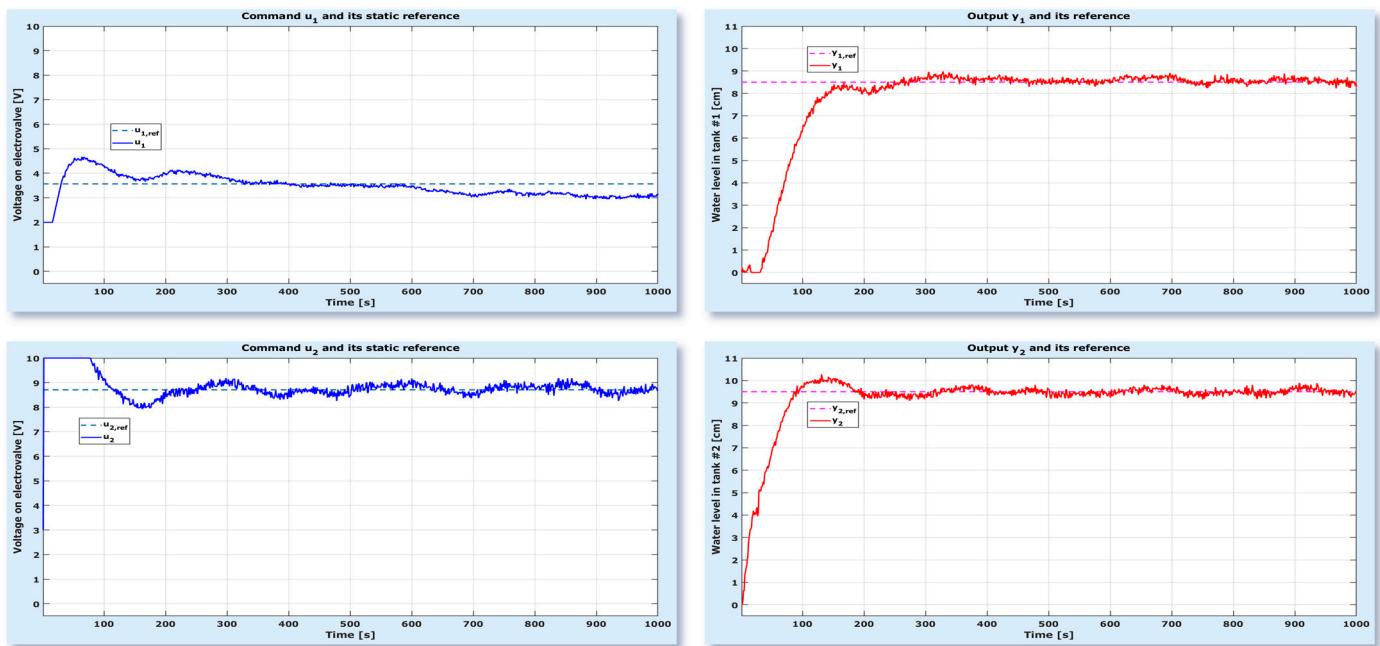
The parameters of optimal PID controllers found by IFA are listed in Table 5. The optimal PID-R parameters are very different from the ones in Table 3, especially in the case of derivative time constant ( $T_d$ ). Not only the parameters are different, but the I/O signals are as well, as revealed by Figures 15–18. Interestingly, the best PID controller found (PID-N for <spline> idmo) works with a PI rule on the first channel (see the null  $T_{d,1}$  in Table 5), whilst the derivative time constant on the second channel is at least 70 times larger than the other derivative time constants on that channel. Simulations proved that the command on channel 2 mastered very well both output channels, which, in turn, were less sensitive to the variations of command on channel 1.



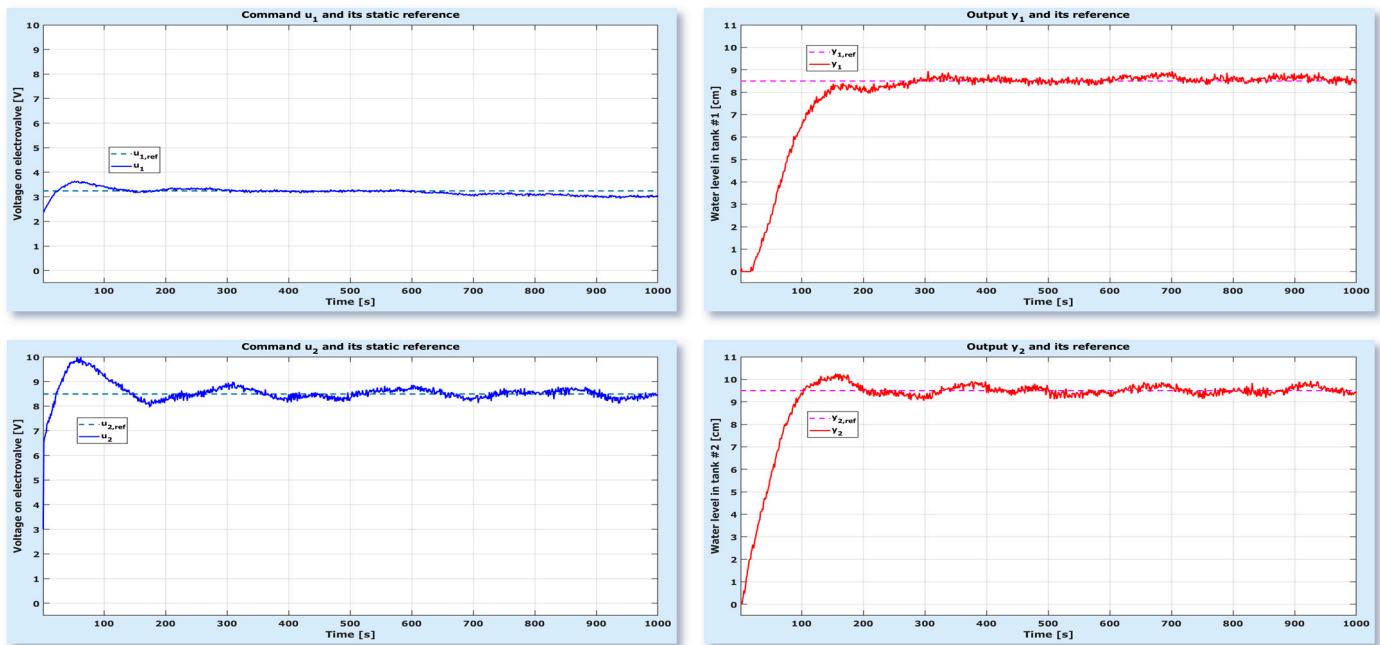
**Figure 15.** I/O signals for optimal PID-R controllers; in the case of <raw> idmo,  $\Phi_o^{\text{IFA}} = 83.90\%$ .



**Figure 16.** I/O signals for optimal PID-R controllers; in the case of <spline> idmo,  $\Phi_o^{\text{IFA}} = 85.01\%$ .



**Figure 17.** I/O signals for optimal PID-N controllers; in the case of <raw> idmo,  $\Phi_o^{\text{IFA}} = 83.84\%$ .



**Figure 18.** I/O signals for optimal PID-N controllers; in the case of <spline> idmo,  $\Phi_o^{\text{IFA}} = 85.07\%$ .

**Table 5.** Parameters of optimal PID controllers found by means of IFA.

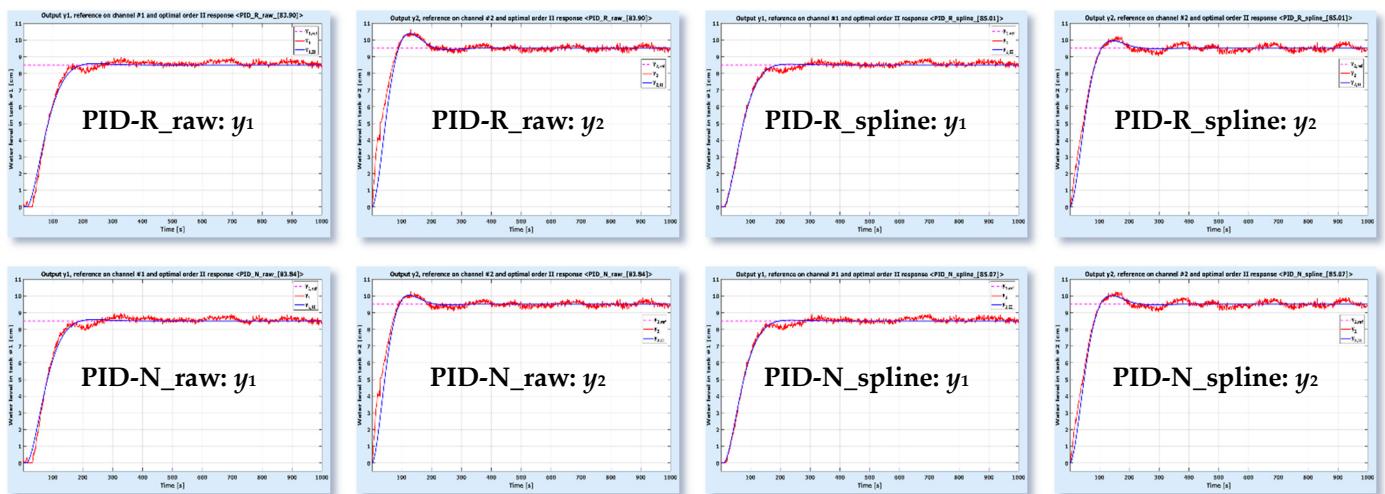
PID	Identification Model	$K_{p,1}$	$T_{i,1}$ [s]	$T_{d,1}$ [s]	$K_{p,2}$	$T_{i,2}$ [s]	$T_{d,2}$ [s]
R	<raw>	0.2413502	45.6150814	0.0001611	1.4740651	42.7184059	0.0005492
	<spline>	0.397183	76.6958348	0.00002	0.6197605	28.9826891	0.0012607
N	<raw>	0.0406009	3.1435647	45.1171123	1.1140813	37.4705655	0.0013905
	<spline>	0.2778442	55.2662226	0	0.6518653	28.9044758	0.1017565

Moreover, as seen in Figures 15 and 17, corresponding to <raw> idmo, the command  $u_2$  exhibits saturation of e-v voltage at 10 V on the first 100 s of transient zone. Though this may have speeded up the references tracking on output channels, saturating the voltages at their upper limit is to be avoided in order to protect the e-v. Fortunately, saturation does not occur in the case of <spline> idmo (Figures 16 and 18), which constitute an important advantage of this model.

Another interesting characteristic concerns the outputs. In all variations above, the output signals are like step responses of a second order system, corrupted by some noises.

This allows some supplementary performance parameters (beside the fitness) to be estimated, provided that the noise can be filtered. Such a filter can be designed by identification of each noisy output with the ideal step response of order II.

In Appendix A, the identification method is briefly described. Let  $y_{j,\text{II}}^{(\zeta, \omega, \tau)}$  be the identified step response of order II corresponding to output  $y_j$  ( $j \in \{1, 2\}$ ). Then, after performing identification on outputs in Figures 15–18, the matched step responses of order II are depicted in Figure 19.



**Figure 19.** Outputs for the optimal PID controllers and their identified step responses of order II.

Starting from  $y_{j,\text{II}}^{(\zeta, \omega, \tau)}$  ( $j \in \{1, 2\}$ ), several specific performances can be evaluated. In Appendix B, the most important performance parameters are defined. Table 6 encompasses the values of performance parameters corresponding to signals in Figure 19.

**Table 6.** Optimal PID controllers performance parameters related to second order step response.

PID	Identification Model	Channel	$\zeta$	$\omega$ [rad/s]	$\tau$ [s]	$T$	RO [%]	$\tau_p$ [%]	$\tau_d$ [s]	$\tau_r$ [s]	$\tau_s$ [s]	$N_t$
R	<raw>	1	0.8243590	0.0243468	11.03	49.82	1.030554	238.98	75.8	195.32	175.57	165
		2	0.6028431	0.0309242	0	53.64	9.31346	127.33	45.98	89.89	172.81	195
	<spline>	1	0.8506410	0.0267793	9.03	43.90	0.620133	232.16	68.60	192.84	168.76	165
		2	0.6946152	0.0304711	0	47.25	4.814988	143.32	48.78	106.69	157.10	108
N	<raw>	1	0.8287348	0.0250022	11.03	48.26	0.954089	235.55	74.23	193.10	183.62	165
		2	0.6724633	0.0338145	0	43.98	5.759211	125.53	43.49	92.23	144.98	192
	<spline>	1	0.8506410	0.0262748	9.03	44.74	0.620133	236.45	69.75	196.37	171.83	165
		2	0.6787924	0.0305118	0	48.28	5.480398	140.21	48.35	103.41	159.55	194

The damping factor  $\zeta$  in the table shows that all outputs are under damped, although the relative overshoot RO is noticeably bigger on the second channel compared to the first channel (where this parameter is very small). Nevertheless, the maximum RO is less than 10% of reference for all controllers, which constitutes a good performance. The natural pulsation  $\omega$  is slightly bigger on the second channel. This property is confirmed

by the parameter  $T_u$  in Table 3 (smaller for the second channel) and the small, damped oscillations of  $y_2$  in Figures 15–19. On channel 1, an intrinsic delay  $\tau$  between 9 and 11 s was detected, which corresponds to variations of  $y_1$  in Figures 15–19. The controllers working with <spline> idmo seem to be slightly faster than the ones synthesized for <raw> idmo, not only in terms of intrinsic delay, but also when looking at the other time parameters: shorter time delay  $\tau_d$ , rise time  $\tau_r$  and settling time  $\tau_s$ . However, PID-N controllers are not necessarily better than PID-R controllers, according to the parameters in Table 6. In the last column of the table, the estimated transient time  $N_t$  is shown for each controller. This parameter is compared with the settling time,  $\tau_s$ , to validate the estimation method described in Section 2.3. One can notice that the approximation is fair. Nevertheless, recall that  $\tau_s$  is computed for a clearly specified tolerance tube around the reference in the steady-state zone (set to  $\pm 5\%$ ), whereas this specification is not employed in estimation of  $N_t$ .

### 3.3. Robustness Analysis

Controllers can or cannot be robust in several respects. Four types of robustness are investigated within this subsection: (a) against sudden change of references; (b) against constant perturbations on outputs; (c) against stochastic perturbations on outputs; and (d) against idmo uncertainties (stochastic variations of model parameters). The reason such an analysis is developed hereafter is twofold. On one hand, very often, installations like ASTANK2 do not comply with the theoretical hypotheses founding the very sound robustness theory devised in the literature. The hypotheses regarding the nature and power of exogenous noises usually are violated. Since such hypotheses only are sufficient (but not necessary) conditions of robustness, the fact they are not verified does not mean the controller cannot be robust to some extent. So, robustness has to be tested in order to estimate robustness margins without employing theoretical results in this aim. On the other hand, robustness analysis of such a real-world plant (especially the robustness against model uncertainties) are very useful in practice. Knowing the robustness margins, even with satisfactory accuracy, can help in predicting the mean time before failure or the right instant to act in correcting defects (not too soon, not too late). This may result in a better maintenance of installation, with optimally small costs.

#### 3.3.1. Robustness against Sudden Change of References

The Simulink model of Figures 10 and 11 with inputs of Figure 12 were employed to test this type of robustness. More specifically, the step references were replaced by staircases. The performance of each PID controller was evaluated by means of fitness. Unlike fitness in (22), in this case, the fitness was computed according to the procedure described in Appendix C. Several staircase references were tested. Overall, if the references steps comply with ASTANK2 limitations, in all tested scenarios, the outputs succeed to track them, which means this type of robustness is accomplished (although with different fitness values, for different references).

To illustrate the tracking performance, the matrices of variations in Figures 20 and 21 were built. In Figure 21, the results correspond to switched staircase references of Figure 20. Additionally, each row of matrix exhibits I/O signals of a controller, as specified on the graphical windows. (Many of the following figures have the same structure.)

The best fitness values are obtained by the PID-N controller with <raw> idmo (77.84% and 76.96%), for which the commands better follow their static references and the outputs better track the specified references. The second best is the PID-R controller with <spline> idmo (75.18% and 74.99%). Additionally, one can easily see that channel 1 is less robust than channel 2, where the references are sensibly better tracked. This phenomenon probably is due to the fact the first tank of installation, with a slanted face, induces stronger dynamic non linearities than the second tank (a regular parallelepiped). Hence, the first linear MISO idmo is not as accurate as the second MISO idmo when approximating the nonlinearities of the ASTANK2 plant.

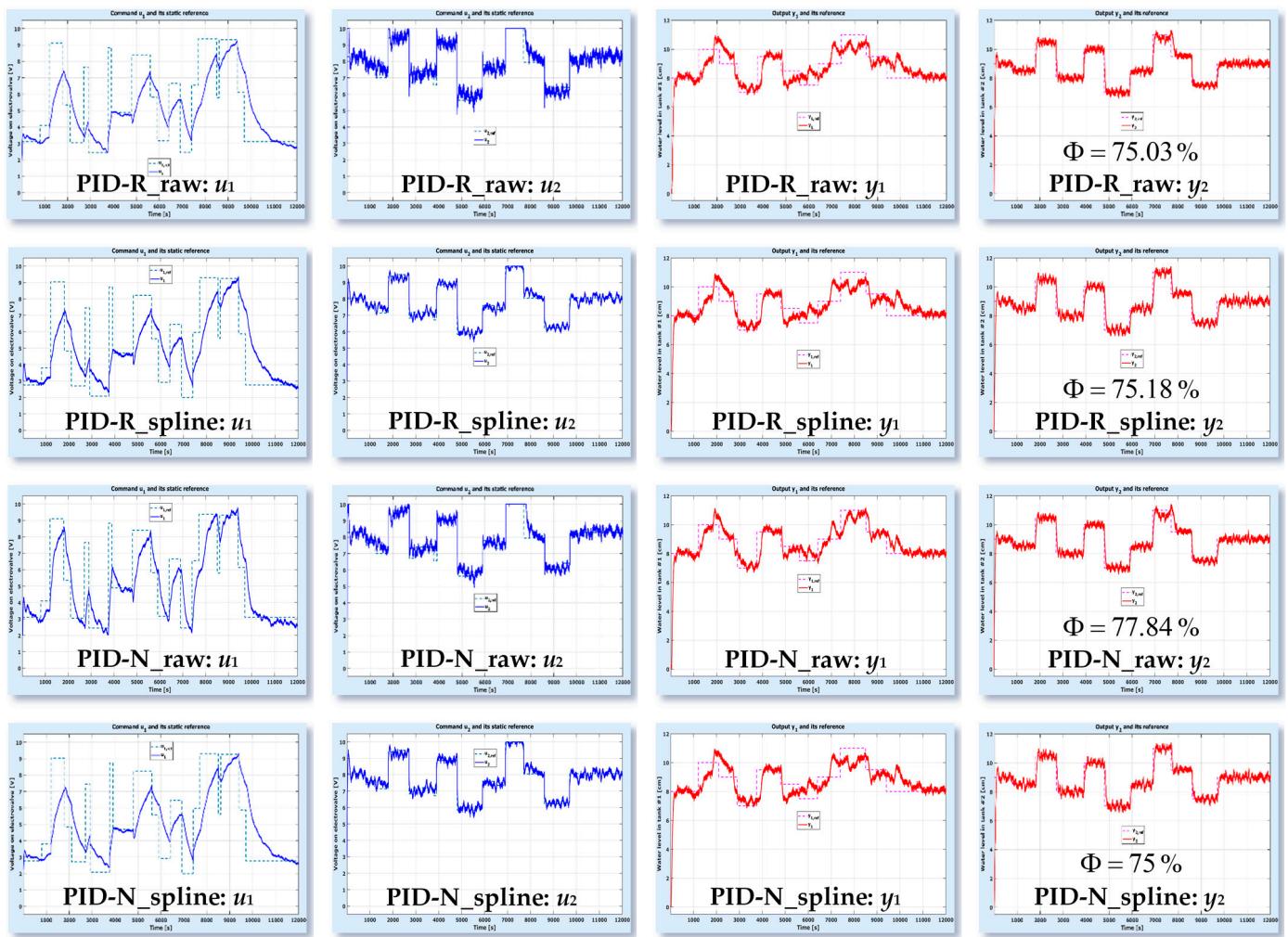


Figure 20. PID-ASTANK2 performance for a configuration of staircase references.

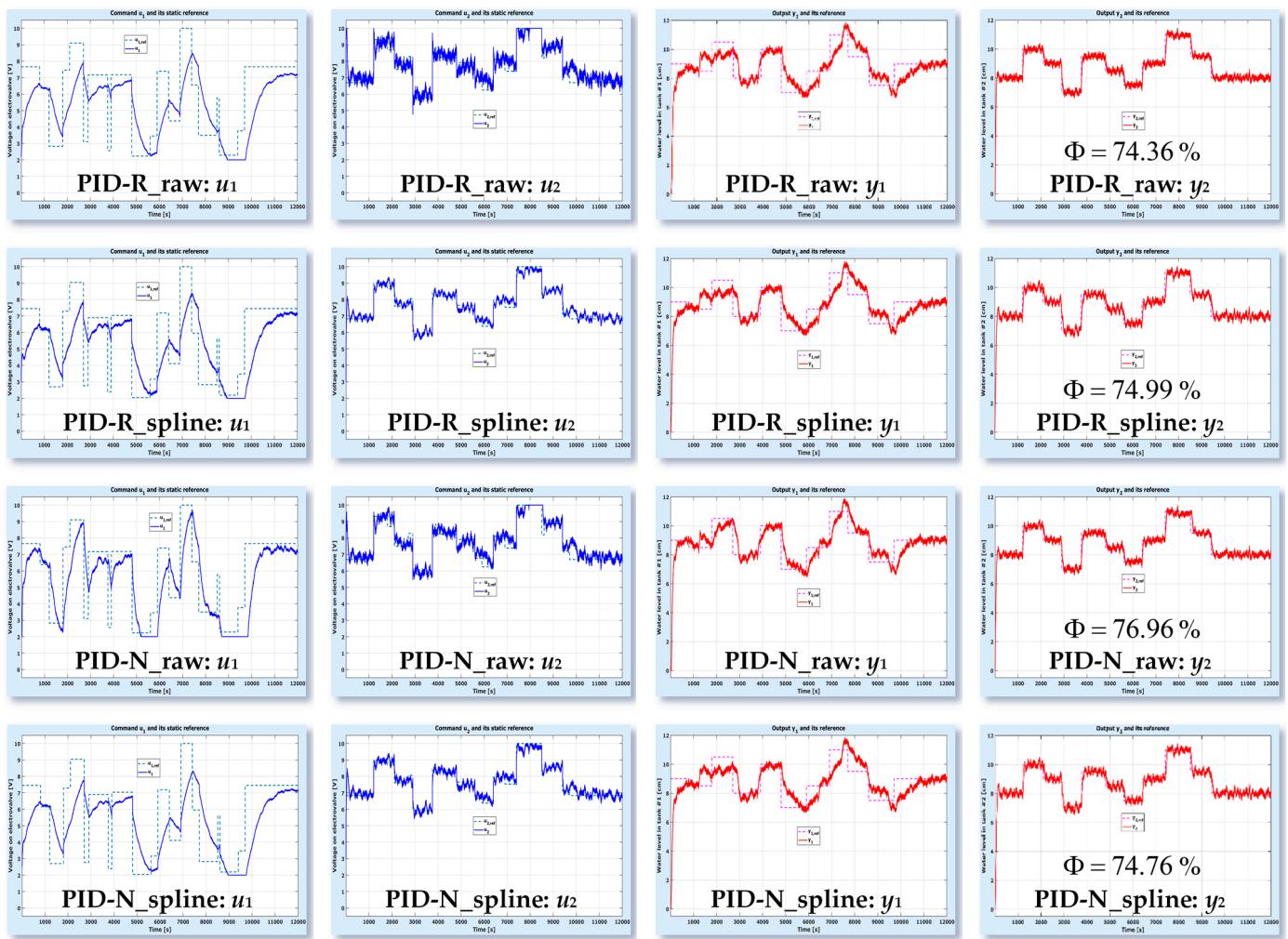
### 3.3.2. Robustness against Constant Perturbations on Outputs

Two constant perturbations were artificially added to each output, while keeping constant the reference: the first one above the reference and the second one, later, beneath the reference. Additionally, the perturbations were not synchronized between channels. The simulation results are displayed by the matrix of variations in Figure 22. All four controllers succeeded in rejecting the constant perturbations, but no one had sensibly better fitness. The commands were slightly smoother in the case of <spline> idmo.

### 3.3.3. Robustness against Stochastic Perturbations on Outputs

The previous simulations were performed by using two sources of Gaussian white noise with variances estimated after ASTANK2 identification (see the last column of Table 1).

To investigate controllers robustness against stochastic perturbations, the variances of noises generators were varied in range  $[0, \lambda_{\max}^2]$ , where  $\lambda_{\max}^2$  was experimentally set to 1 (after few simulations). The problem is how to define and estimate the robustness margin. The fitness can be used for this aim. Obviously, the fitness decreases when noises increase in power, as the variation in Figure 23a illustrates. Here, the horizontal index points to the noise variance. In this example,  $N = 31$  noise variance levels are employed within the interval  $[0, 1]$ . Thus, if the index is set to  $n \in \overline{1, N}$ , then the two noises have the same variance  $\lambda_{1,n}^2 = \lambda_{2,n}^2$ .



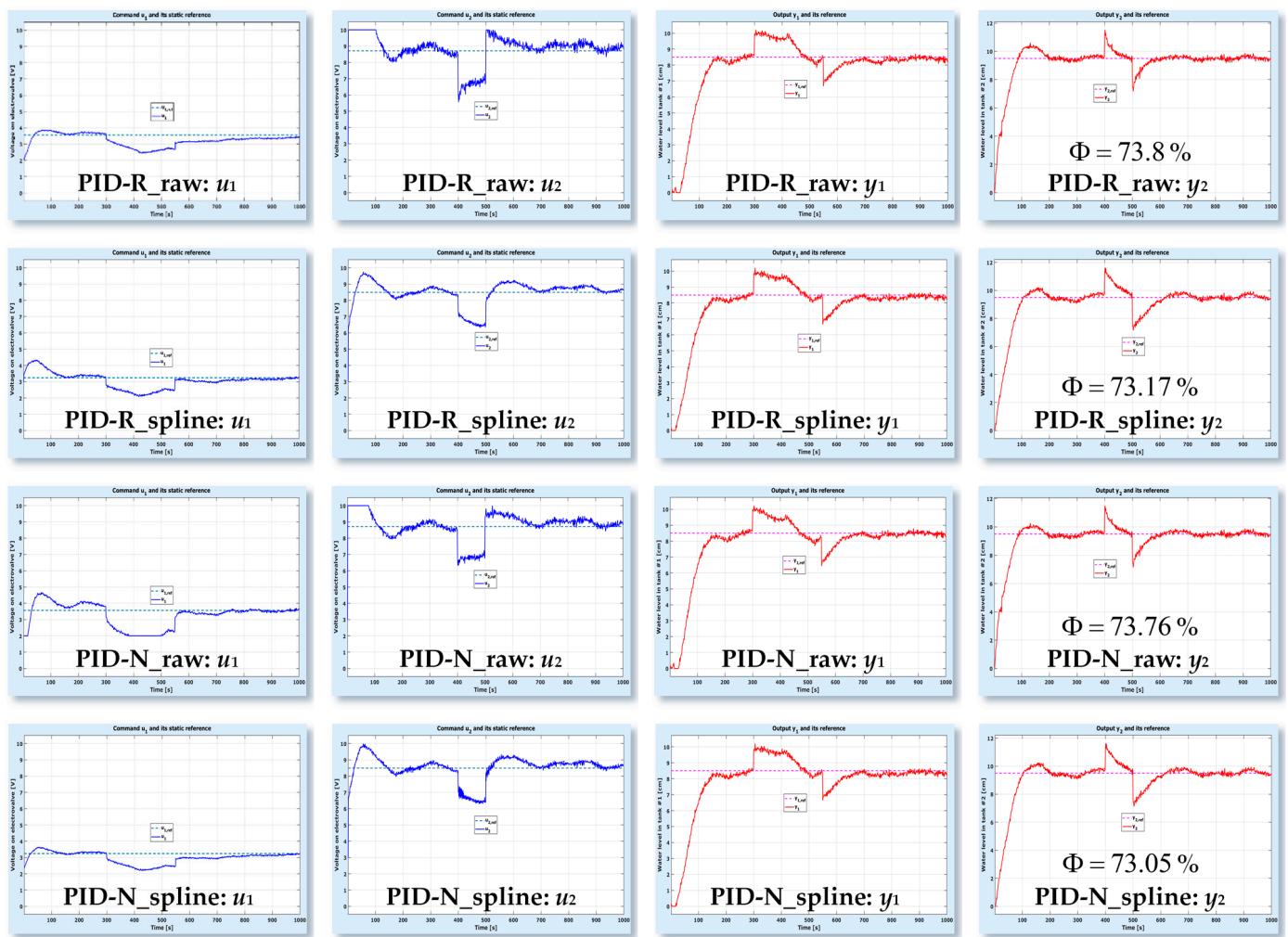
**Figure 21.** PID-ASTANK2 performance for switched references of the previous figure.

The problem with fitness is that, apparently, the robustness margin can only be set by subjective means while visually inspecting the I/O signals of the plant. Clearly, the marginal behavior of a closed loop system is expressed by outputs that oscillate around references with approximately constant amplitudes. However, there is a fine line between various oscillating behaviors. Hence, setting the right robustness margin becomes a difficult task. To remove the subjectiveness, a supplementary parameter can be employed. Consider the tracking errors on the steady-state zone for all I/O signals (commands and outputs). Evidently, when the noise variance increases, the std of each tracking error increases as well.

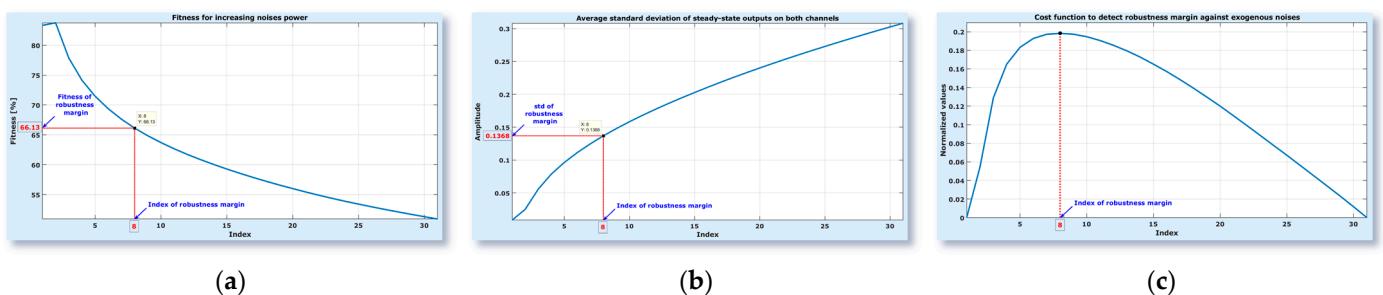
Therefore, a global std on the steady-state zone can be computed as follows (with natural notations related to I/O signals):

$$\sigma^{SSZ} = \alpha \left( \sigma_{\varepsilon_{y,1}}^{SSZ} + \sigma_{\varepsilon_{y,2}}^{SSZ} \right) + \frac{\beta}{2} \left( \sigma_{\varepsilon_{u,1}}^{SSZ} + \sigma_{\varepsilon_{u,2}}^{SSZ} \right), \quad (34)$$

where the weights  $\alpha$  and  $\beta$  are the ones employed to compute fitness (see definition (21)), e.g.  $\alpha = 0.35$  and  $\beta = 0.3$ . Such a global std varies as shown in Figure 23b. The variation is increasing with the noise index, as expected.



**Figure 22.** PID-ASTANK2 performance for constant perturbations applied on outputs.



**Figure 23.** Principle of robustness margin estimation for increasing noises variance.

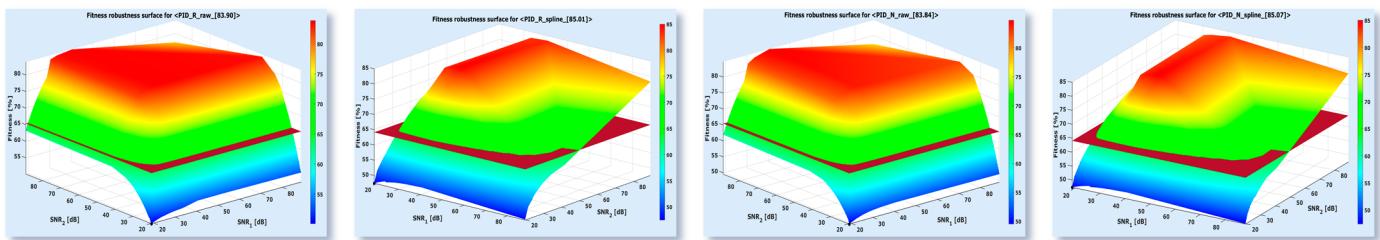
Since the fitness decreases and the variance increases, a natural way to quantify the robustness margin is to normalize both variations and to define a cost function by taking their product. More specifically, one can define  $J \equiv \Phi_{\text{norm}} \sigma_{\text{norm}}^{\text{SSZ}}$ , where:

$$\Phi_{\text{norm}} \equiv \frac{\Phi - \min_{n \in \overline{1, N}} \{\Phi[n]\}}{\max_{n \in \overline{1, N}} \{\Phi[n]\} - \min_{n \in \overline{1, N}} \{\Phi[n]\}} \quad \text{and} \quad \sigma_{\text{norm}}^{\text{SSZ}} \equiv \frac{\sigma^{\text{SSZ}}[n] - \min_{n \in \overline{1, N}} \{\sigma^{\text{SSZ}}[n]\}}{\max_{n \in \overline{1, N}} \{\sigma^{\text{SSZ}}[n]\} - \min_{n \in \overline{1, N}} \{\sigma^{\text{SSZ}}[n]\}}. \quad (35)$$

Figure 23c shows the variation of cost function  $J$ . The maximum point,  $n_{\max}$ , can be set to the index of robustness margin. The robustness margin is then defined as

$\{n_{\max}, \Phi[n_{\max}]\}$ . If the exogenous noises lead to fitness values at least equal to  $\Phi[n_{\max}]$ , then one considers the controller is robust. Otherwise, the controller is not performing well. Nevertheless, it is possible that the closed loop system remains at stability limit even outside the robustness zone.

After simulations with step references and variable noise variances, the results in Figure 24 were obtained. In fact, in the figure, the noise variance was replaced by the estimated SNR, on each channel, computed on the steady-state zone of outputs. The fitness surfaces in Figure 24 are split in two regions by a horizontal plane determined after maximization of cost function  $J$ . One considers that all points on the surface above the plane are due to the robust behavior of controller. The height of each separation plan is listed in Table 7, together with the corresponding marginal SNR. The lower the separation plan height, the larger the robustness zone of the controller. One can notice that the PID-N controller for <raw> idmo is slightly more robust than the other three controllers. Additionally, every controller can be robust if one noise is strong while the other one is weak.



**Figure 24.** Fitness robustness surfaces depending on exogenous noise variances.

**Table 7.** Robustness margins of PID controllers depending on output SNR.

PID	Identification Model	Marginal Fitness [%]	Marginal SNR [dB]	Optimal Fitness [%]	SNR <sub>1</sub> [dB]	SNR <sub>2</sub> [dB]
R	<raw>	65.37	25.99	83.90	40.37	41.34
	<spline>	64.12	25.95	85.01	40.80	41.23
N	<raw>	66.45	25.98	83.84	40.34	41.46
	<spline>	64.01	25.96	85.07	40.80	41.25

Another remark concerns the robustness surface: large fitness values cover a larger area for <raw> idmo than for <spline> idmo. This means that, in general, the stochastic noises are better rejected by PID-R and PID-N controllers determined for <raw> idmo than by the other two controllers.

The I/O signals at the robustness margin are shown by the matrix of variations in Figure 25. Such behavior is considered as robust. Many other situations with similar I/O variations exist, but the robustness is decided by fitness margins in Table 7.

### 3.3.4. Robustness against Identification Model Uncertainties

The uncertainties are induced by changing the coefficients of idmo polynomials. Assume  $P$  is any of the model polynomials, expressed as:

$$P(q^{-1}) = p_0 + p_1 q^{-1} + \cdots + p_{np} q^{-np}. \quad (36)$$

Then, the coefficient  $p_i$  ( $i \in \overline{1, np}$ ) is replaced by a (uniform) random value in the interval  $[1 - \eta, 1 + \eta]p_i$ , if  $p_i \neq 0$  or in the interval  $[-\eta, +\eta]$ , if  $p_i = 0$ . The parameter  $\eta \in [0, 1]$  is referred to as *uncertainty power*.

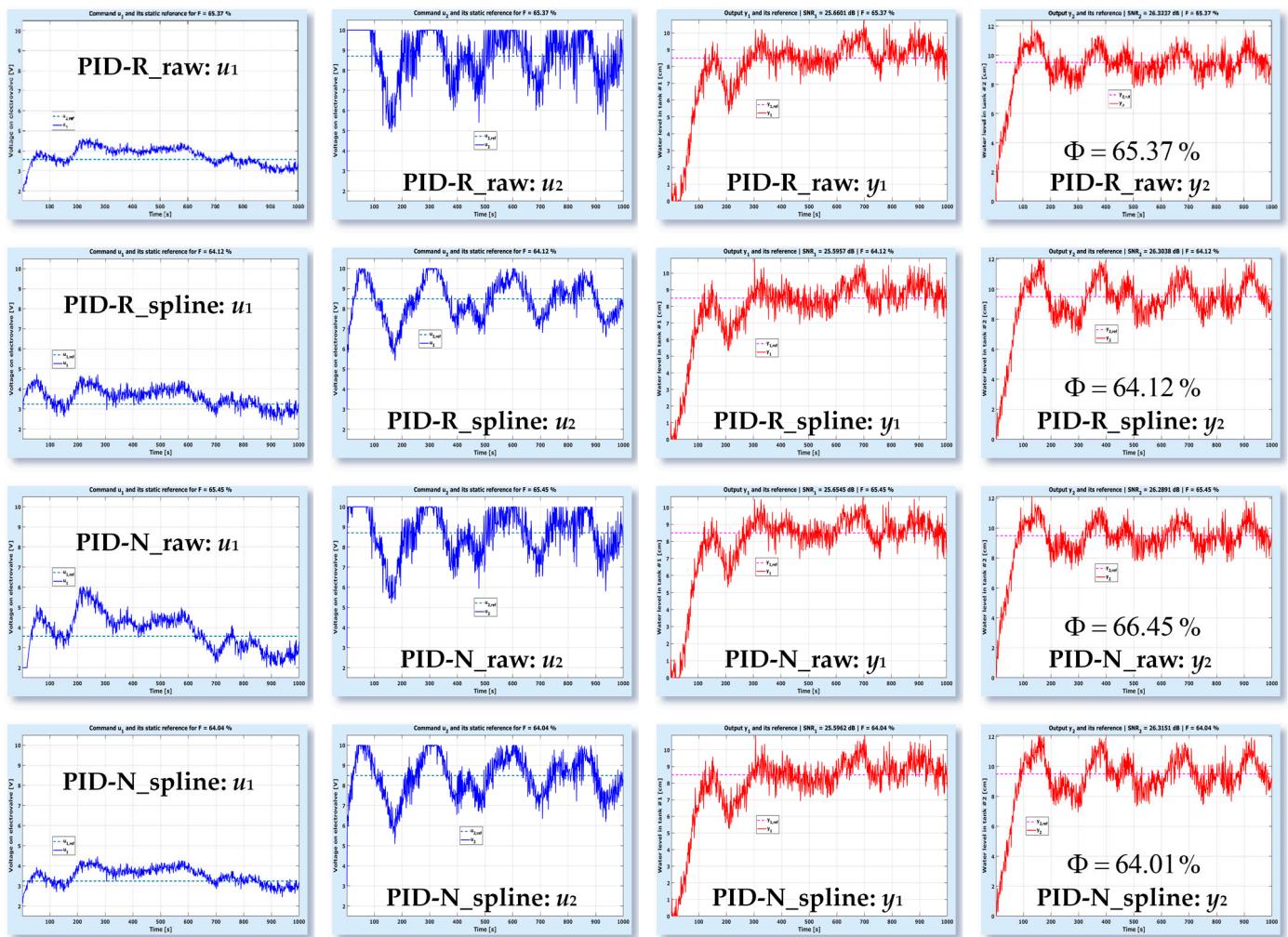


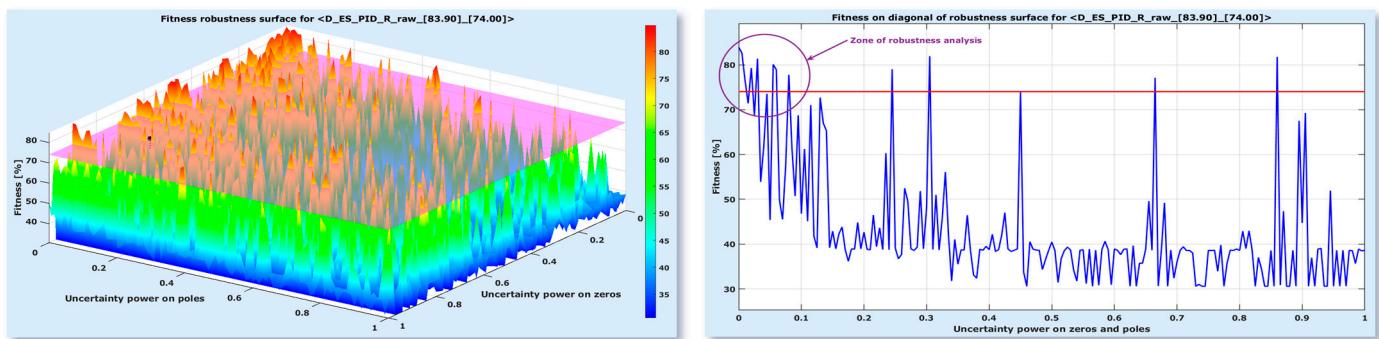
Figure 25. PID-ASTANK2 performance at the margin of robustness against exogenous noises.

The problem is to define and determine the robustness margin of a PID regulator, while working with uncertain model polynomials and variable uncertainty powers. Obviously, the PID performance is expressed in terms of fitness. Nevertheless, to solve the problem of uncertainty margin detection, the fitness needs to be put into correspondence with the reference tracking property. One can consider that the uncertainty margin is reached as soon as the closed loop system is unable to track at least one output reference signal. In this case, the polynomials are affected by uncertainties of some power, whilst the closed loop system exhibits a marginal fitness. Thus, the uncertainty margin can be defined by the smallest fitness value for which the closed loop system still manages to track all given references. This lower limit also reveals the marginal (maximal) values of uncertainty power. For example, the robustness margin of 75% (in terms of fitness) can be obtained by affecting the plant model with uncertainties of estimated marginal (maximal) power of 10% ( $\eta \in [0, 1]$ ). Increasing the uncertainty power leads to the incapacity of the closed loop system to track references.

In the following, an example is given. The output  $y_1$  of ASTANK2 installation is acquired after applying uncertainties on the numerator and denominator of the useful transfer function in MISO<sub>1</sub> idmo (see Figure 1b again). The uncertainties were generated at random, by using the uniform pseudo-random generator of MATLAB (`rand` function).

The uncertainties can affect the 3 polynomials of useful transfer function, namely:  $A_1$ ,  $B_{1,1}$  and  $B_{1,2}$ . As Equation (1) shows, the first coefficient of polynomials is normalized: 1 for  $A_1$  and 0 for  $B_{1,1}$ ,  $B_{1,2}$ . Uncertainties cannot affect this coefficient. By inducing uncertainties on the remaining coefficients, a fitness surface can be generated, depending

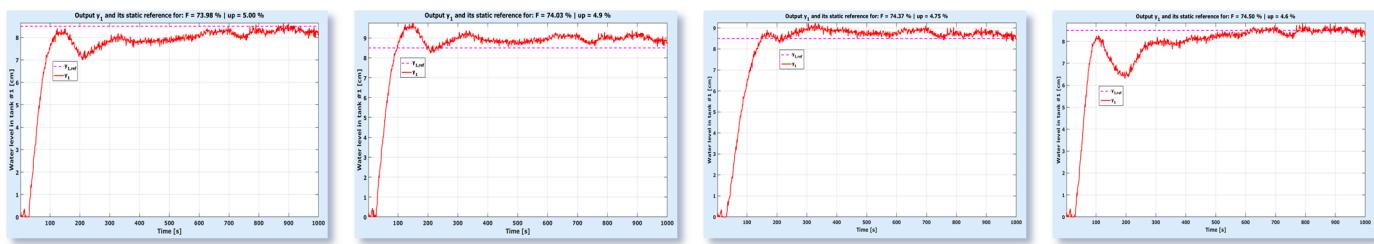
on two uncertainty powers: one affecting the denominator  $A_1$  and another one modifying the numerators  $B_{1,1}$ ,  $B_{1,2}$ . The ASTANK2 idmo usually have polynomials with many coefficients (up to 50, in some cases). Trying to test all combinations of uncertainty applied on model coefficients is unrealistic. Therefore, the Monte-Carlo method can be employed to estimate the uncertainty margin. Hence, 100 uncertainty cases are uniformly generated at random on each axis. The resulted fitness surface looks like that in the left side of Figure 26. The estimated fitness margin is represented by the horizontal plane intersecting the fitness surface. All points above the plane correspond to the robust behavior of the closed loop system. The other fitness values, beneath the plane, rather reveal the incapacity of the closed loop system to track at least one reference. As the figure clearly shows, there are cases in which the robustness is preserved even though one uncertainty power is high and the other one is low. Consequently, it is quite difficult to determine the marginal powers if they are different. It is more suitable to determine such a marginal power along the main diagonal of the separation plane, where the two powers are equal. In the right side of Figure 26, the fitness variation along the diagonal (with equal uncertainty powers) is displayed. The horizontal line refers to the separation plane.



**Figure 26.** Principle of robustness margin estimation for increasing uncertainty power.

One can see that the most fitness values are beneath this line, as expected. Increasing the uncertainty power naturally leads to the loss of robustness. Therefore, to estimate the robustness margin, one should focus on the zone with small uncertainty powers. In the figure above, one can easily notice that the marginal uncertainty power is less than 10%.

The problem with fitness is the same as in the previous subsection; apparently, the robustness margin can only be set by subjective means, while visually inspecting the I/O signals of the plant. For example, the first two output signals in Figure 27 reveal that the system is unable to track the reference, but is very close to this goal. The fitness equals 73.98% with 5% uncertainty power and 74.03% with 4.9% uncertainty power, respectively. By decreasing the uncertainty power, the last two output signals in Figure 27 were obtained. At 74.37% fitness with 4.75% uncertainty power, the reference started to be tracked in the end of variation (see the third signal of figure). Furthermore, decreasing the uncertainty power to 4.6% leads to a fitness of 74.5% and to an output that succeeds in reaching the reference sooner than previously (see the last signal). In this case, the robustness margin can be set to 74.4% fitness with 4.7% uncertainty power. The estimation of uncertainty margin is fair but subjectively set.



**Figure 27.** Output signals behavior around the robustness margin.

The robustness margins can thus be estimated by varying the separation plane and analyzing the cases leading to fitness values close to the plane height. If the uncertain idmo that leads to fitness values above and beneath the plane height still allow tracking of the references, then the plane should be lowered. On the contrary, if the idmo with fitness values around the plane height yield loss of robustness, then the plane should be raised. The separation plane is placed at the right height when the idmo with fitness below that height cannot allow reference tracking, whereas for the other idmo, with fitness above the height, the references are eventually tracked. Because of the noises corrupting the data, such a balanced behavior can be discovered experimentally, as a theoretical approach on this topic is difficult (if not impossible) in this case.

For ASTANK2 installation, three scenarios can be investigated: (a) affect MISO<sub>1</sub> model by uncertainties and the keep the genuine MISO<sub>2</sub> model; (b) switch MISO models (keep the genuine MISO<sub>1</sub> and work with uncertain MISO<sub>2</sub>); (c) induce uncertainties on both MISO models. After several simulations, it has been noticed that output  $y_1$  is more sensitive to idmo uncertainties than output  $y_2$ . This means the controller loses robustness because of  $y_1$  in the first. Therefore, in the following figures, only  $y_1$  is displayed.

Figures 28–30 display the simulation results corresponding to the three scenarios mentioned above. Each figure includes a matrix of variations, as usual, but with different representations from the previous figures.

In all figures, the fitness surface obtained by Monte-Carlo method takes the first column of matrix. The second column displays the diagonal of fitness surface. The last two columns are devoted to output  $y_1$  in two cases: non robust (column 3) and robust (column 4). The corresponding fitness values of closed loop system for columns 3 and 4 are below and above the estimated marginal fitness of robustness zone, respectively.

The simulation scenarios providing the results in Figures 28–30 offer further parameters allowing analyzing robustness. Hence, the robustness zone can be confined by: the marginal fitness  $\Phi_{min}$ ; the marginal uncertainty power  $\eta^{max}$ ; the relative density  $\rho$  of robust behavior; the relative robustness margin  $\mu$ . The first two parameters ( $\Phi_{min}$  and  $\eta^{max}$ ) are determined experimentally. The height of separation plane is set to  $\Phi_{min}$ . The fitness values  $\Phi_{down}$  and  $\Phi^{up}$  corresponding to variations of output  $y_1$  in Figures 28–30 are close to each other.

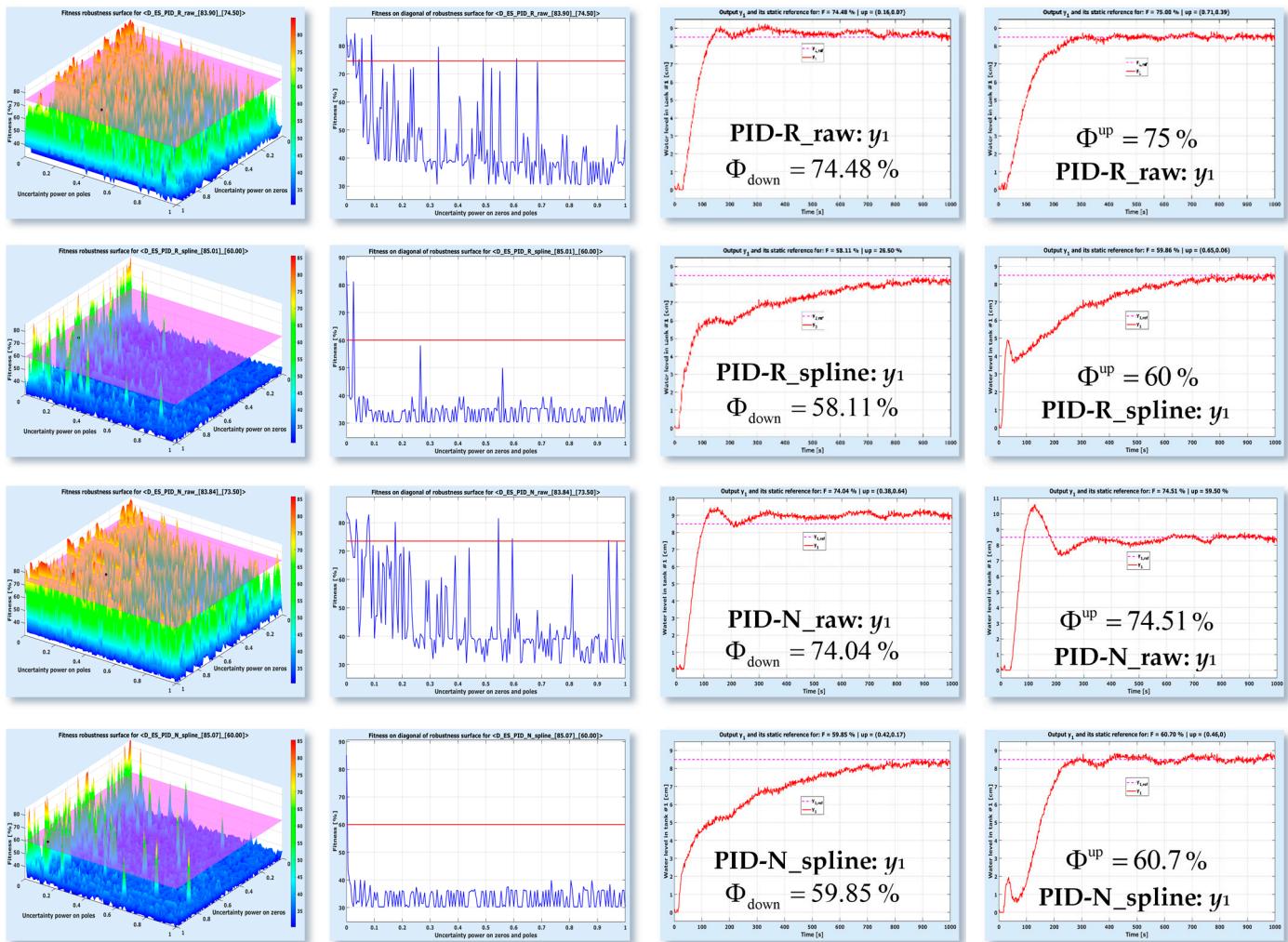
Moreover,  $\Phi_{down} < \Phi_{min} < \Phi^{up}$ , which proves that  $\Phi_{min}$  is fairly accurate. The marginal uncertainty power  $\eta^{max}$  was determined by using the fitness diagonal drawn in column 2 of graphical matrices. The principle illustrated in Figure 26 was used in this aim.

The other two parameters ( $\rho$  and  $\mu$ ), although conventionally defined, allow performing a top of controllers with respect to robustness against uncertainties. Since the Monte-Carlo method is employed to draw fitness surfaces, one can compute the ratio between the number of robust behaviors and the number of all behaviors (10.000). If this ratio is multiplied by 100, the relative density  $\rho$  of robust behavior is estimated. Now, one controller is more robust than another if  $\Phi_{min}$  is as small as possible, whilst  $\eta^{max}$  and  $\rho$  are as big as possible. Then, the relative robustness margin  $\mu$  can be defined as:

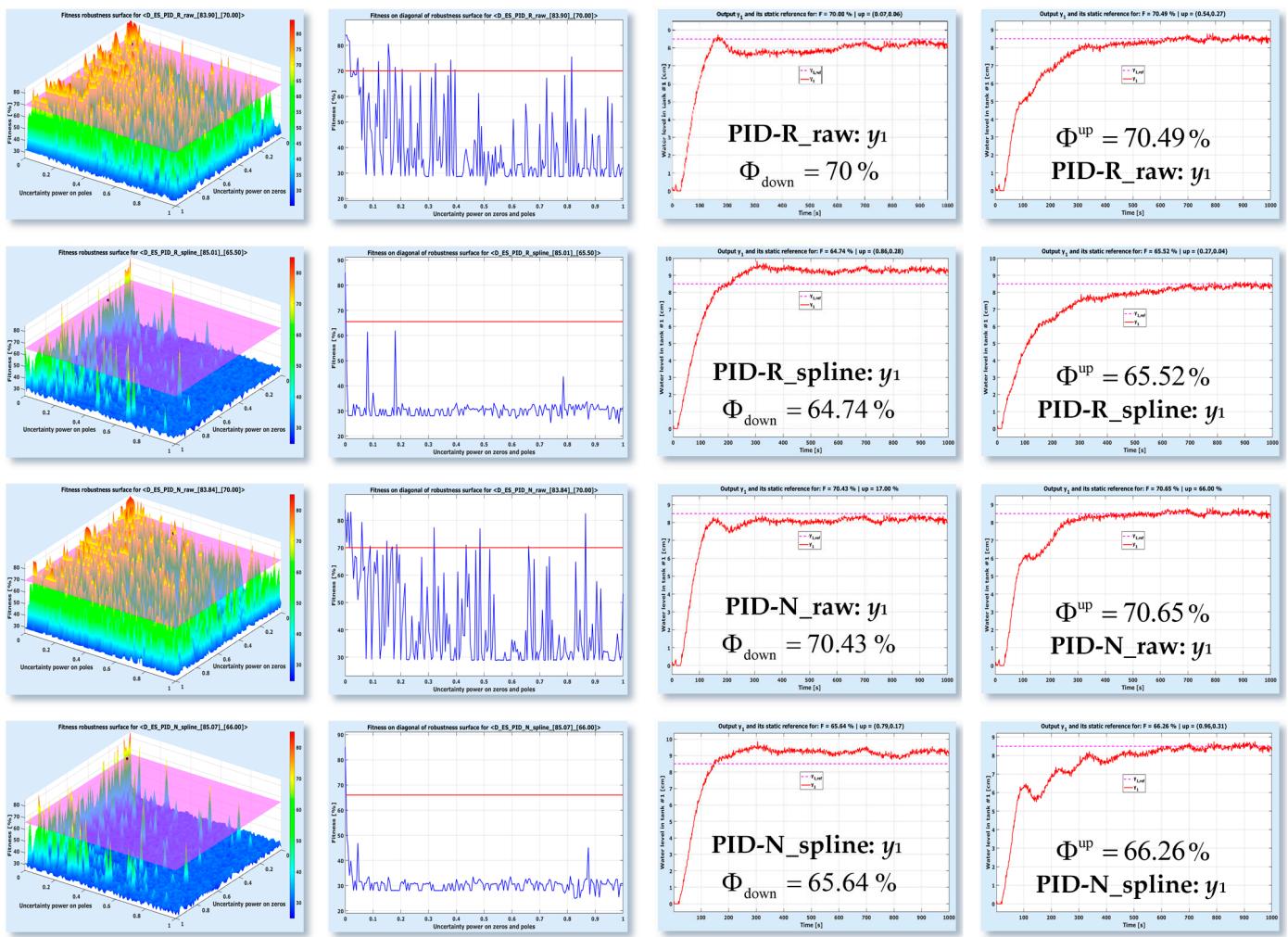
$$\mu = \frac{100}{1 + \sigma \frac{\Phi_{min}}{\rho \eta^{max}}} [\%], \quad (37)$$

where  $\sigma > 0$  is a constant scaling factor set to 0.04, such that  $\mu$  can vary in a suitable range (recall the explanation between Equations (15) and (16)).

The robustness parameters are listed in Table 8, for all four controllers, in all scenarios. Without the last column, Table 8 is rather confusing and says almost nothing. The  $\mu$  parameter can enlighten the closed loop behavior against model uncertainty.



**Figure 28.** Estimating the robustness margin against the uncertainties of the MISO<sub>1</sub> model.



**Figure 29.** Estimating the robustness margin against the uncertainties of the MISO<sub>2</sub> model.

The first clear insight is that the PID controllers for <raw> idmo are noticeably more robust to uncertainties than the PID controllers for <spline> idmo. This can be explained by the fact the <spline> idmo has a reduced number of poles, which are more exposed to the loss of stability in a way that the PID controllers cannot compensate. Secondly, among controllers for <raw> idmo, PID-R is slightly more robust than PID-N when coping with uncertainties that affect one MISO model at a time, whereas when both models are corrupted, PID-N seems to be the best (62.65% versus 45.97%). Thirdly, for <spline> idmo, the PID-R controller is better than the PID-N controller, in terms of robustness, in all scenarios. To conclude this analysis, one can notice that, regardless of the idmo type, robustness of MISO<sub>2</sub> is superior to robustness of MISO<sub>1</sub>, as expected (since, in all simulations, output  $y_1$  loses its tracking ability before output  $y_2$ ). Moreover, another expectation is met: the relative robustness margin of the entire ASTANK2 idmo (in the last scenario) is the smallest one (compared to the other two scenarios).

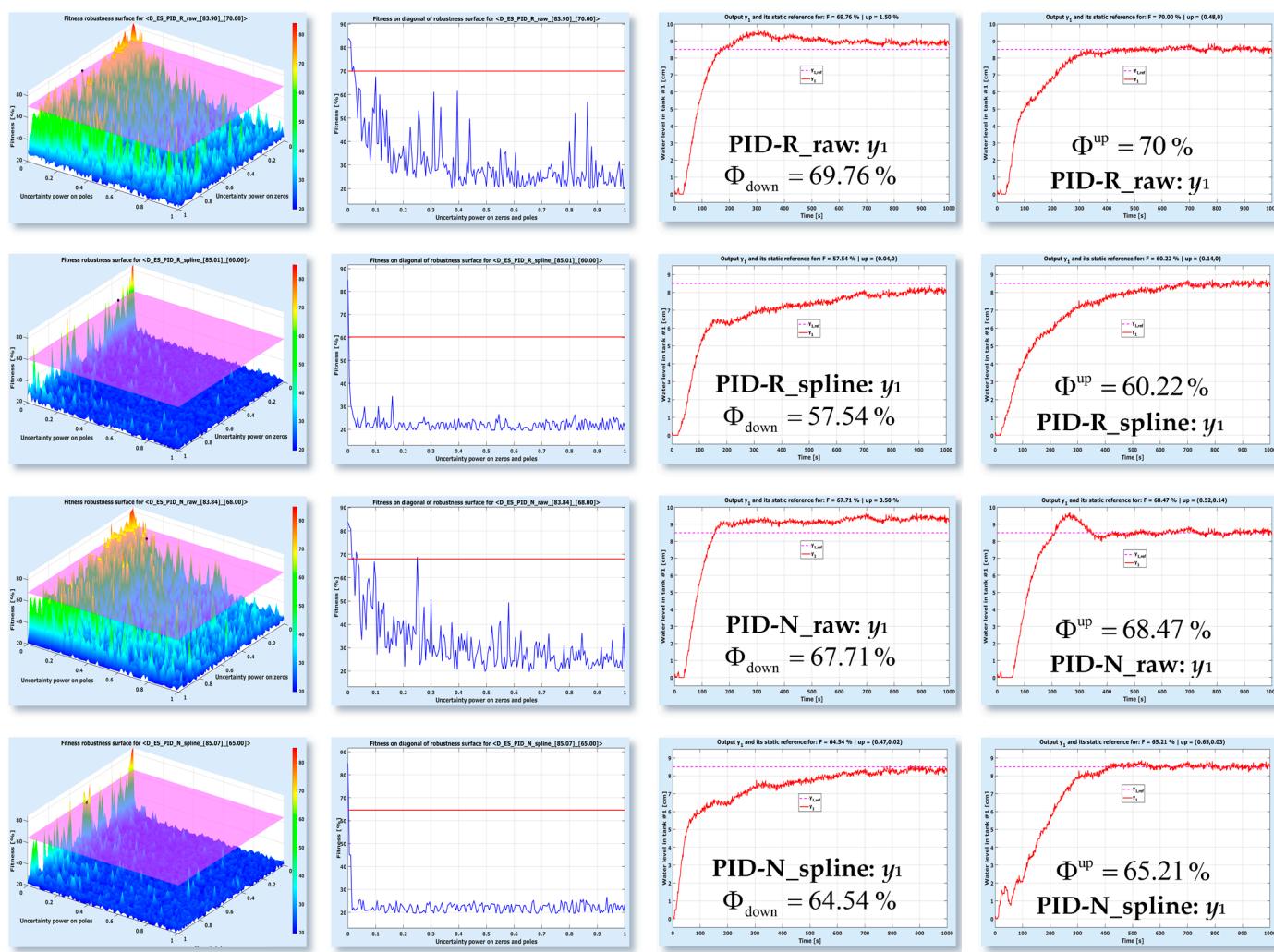


Figure 30. Estimating the robustness margin against uncertainties of the overall ASTANK2 model.

Table 8. Parameters of robustness against model uncertainties.

PID	Identification Model	Uncertainties Affected	$\Phi_{min}$ [%]	$\eta^{max}$ [%]	$\Phi_{down}$ [%]	$\Phi^{up}$ [%]	$\rho$ [%]	$\mu$ [%]
R	<raw>	MISO <sub>1</sub>	74.5	9	74.48	75	2.81	89.47
		MISO <sub>2</sub>	70.5	15.5	70	70.49	5.84	96.98
		MISO <sub>1</sub> and MISO <sub>2</sub>	70	2.25	69.76	70	1.06	45.97
	<spline>	MISO <sub>1</sub>	60	2.5	58.11	60	0.7	42.03
		MISO <sub>2</sub>	65.5	1	64.74	65.52	0.45	14.68
		MISO <sub>1</sub> and MISO <sub>2</sub>	60	1	57.54	60.22	0.17	6.49
N	<raw>	MISO <sub>1</sub>	74.5	8	74.04	74.51	2.54	87.21
		MISO <sub>2</sub>	70.5	9	70.43	70.65	4.51	93.5
		MISO <sub>1</sub> and MISO <sub>2</sub>	68	3.5	67.71	68.47	1.3	62.65
	<spline>	MISO <sub>1</sub>	60	1	59.85	60.7	1.08	31
		MISO <sub>2</sub>	66	1	65.64	66.26	0.12	4.22
		MISO <sub>1</sub> and MISO <sub>2</sub>	65	1	64.54	65.21	0.16	5.69

#### 4. Concluding Remarks and Future Developments

The main goal of this article was to perform an extensive analysis of metaheuristic PID control applied to a nonlinear installation such as ASTANK2, after identification with linear multi-MISO models. Two PID configurations were investigated, PID-R and PID-N, for each

identification model type, <raw> and <spline>. Two metaheuristics were employed in the design of four optimal controllers: AHCA to extensively explore the search space, and IFA to intensively exploit a restricted zone around some promising optimal point found by climbers. Four improvements of classical FA were introduced in order to fasten the search and increase the final results accuracy. Defining a fair cost function (referred to as *fitness* in the article) is not an easy task. Many other authors coped with the same difficult problem, as reported in their publications. Unfortunately, to the best of our knowledge, no general cost function can be defined to be employed in any automatic control optimization problem. Therefore, adapting the cost function to the installation characteristics (and limitations) seemingly is a wise way to proceed.

The simulation results revealed the performance as well as the limitations of designed controllers. Overall, one can say that any of the four PID controllers can be employed in a closed loop with an ASTANK2 installation, as they exhibit almost the same performance. Basically, PID-R controllers are recommended to reject model uncertainties, whilst PID-N controllers are better in tracking staircase type references.

Various future research directions can be approached. A multivariable (non-decentralized) PID control structure likely yields superior performance. Another development is possible in the direction of metaheuristic multivariable RST control. Of course, the metaheuristic control based on state space representation of ASTANK2 installation is another interesting direction to investigate.

**Author Contributions:** Both authors equally contributed to conceptualization; methodology; software; validation; formal analysis; investigation; using resources; original draft preparation writing, review and editing; visualization; supervision. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** To the best of their knowledge, the authors declare no conflicts of interest.

## Abbreviations

A–R	Akaike–Rissanen (criterion)
AHCA	Advanced Hill Climbing Algorithm
BJ	Box-Jenkins (model)
FA	Firefly Algorithm
I/O	input/output (signals, behavior etc.)
IFA	Improved Fireflies Algorithm
MIMO	Multi-Input–Multi-Output (model)
MISO	Multi-Input–Single-Output (model)
PID	Proportional–Integral–Derivative
PID-N	non-standard PID
PID-R	regular PID
RMSE	root mean square error
SISO	Single-Input–Single-Output (model)
SNR	Signal-to-Noise Ratio
e–v	electro-valve(s)
idmo	identification model(s)
std	standard deviation

## Appendix A. Identification of Second Order Step Response from Noisy Signals

Consider the usual transfer function of a second order system (in continuous time):

$$H(s) = \frac{\omega^2 e^{-s\tau}}{s^2 + 2\zeta\omega s + \omega^2}, \quad (A1)$$

where  $\omega > 0$  is the natural pulsation ([rad/s]),  $\zeta \geq 0$  is the (relative) damping factor and  $\tau \geq 0$  is the intrinsic delay. When stimulating the system with a unit step, four types of responses are possible, depending on the damping factor:

1. Harmonic oscillation, when  $\zeta = 0$

$$y(t) = 1 - \cos[\omega(t - \tau)], \forall t \geq \tau. \quad (\text{A2})$$

2. Under damped signal (with overshoot), when  $\zeta \in (0, 1)$ :

$$y(t) = 1 - \frac{e^{-\zeta \omega (t-\tau)}}{\sqrt{1-\zeta^2}} \left[ \sqrt{1-\zeta^2} \cos[\omega_d(t-\tau)] + \zeta \sin[\omega_d(t-\tau)] \right], \forall t \geq \tau, \quad (\text{A3})$$

where  $\omega_d = \omega \sqrt{1-\zeta^2}$  is the damped pulsation of system.

3. Critically damped signal, when  $\zeta = 1$ :

$$y(t) = 1 - [1 + \omega(t - \tau)] e^{-\omega(t-\tau)}, \forall t \geq \tau. \quad (\text{A4})$$

4. Over damped signal (without overshoot), when  $\zeta > 1$ :

$$y(t) = 1 - \frac{1}{2\sqrt{\zeta^2 - 1}} \left[ \frac{e^{-\zeta^+ \omega (t-\tau)}}{\zeta^+} - \frac{e^{-\zeta^- \omega (t-\tau)}}{\zeta^-} \right], \forall t \geq \tau, \quad (\text{A5})$$

where  $\zeta^\pm = \zeta \pm \sqrt{\zeta^2 - 1}$ .

During the delay, the response is set to null (although, by using the equations above in a mathematical sense, the response is non null).

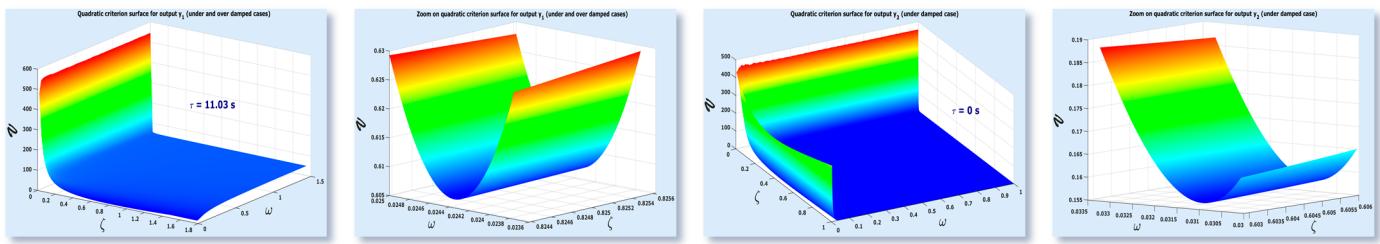
In case of noisy signals looking like a step response of order II, the identification problem can be formulated by means of a quadratic criterion. More specifically, if  $y$  is such a signal (in discrete time) and  $y_{\text{II}}^{(\zeta, \omega, \tau)}$  is any of the step responses (A2)–(A5), then the following quadratic cost function has to be minimized:

$$\mathbf{V}(\zeta, \omega, \tau) = \sum_{n=0}^{N-1} \left[ y(nT_s) - y_{\text{II}}^{(\zeta, \omega, \tau)}(nT_s) \right]^2, \quad (\text{A6})$$

where  $\zeta \geq 0$  (the damping factor),  $\omega > 0$  (the natural pulsation) and  $\tau \geq 0$  (the intrinsic delay) completely determine the response  $y_{\text{II}}^{(\zeta, \omega, \tau)}$ . Additionally,  $N \in \mathbb{N}^*$  is the total number of samples and  $T_s > 0$  is the sampling period (in seconds).

As one can easily notice, the response  $y_{\text{II}}^{(\zeta, \omega, \tau)}$  is nonlinear with respect to the three parameters. Consequently, solving the quadratic optimization problem is not an easy task. Moreover, the criterion can exhibit large flat zones where the global minimum may lie. In Figure A1, criterion surfaces were drawn for each output of ASTANK2 plant ( $y_1$ —the left side couple,  $y_2$ —the right side couple).

The images 1 and 3 reveal how large the flat zone can be. The surface includes all four types of second order responses. For output  $y_1$ , the intrinsic delay is set to about 11 s, whereas, for the output  $y_2$ ,  $\tau = 0$  s. A zoom on the plateau is displayed in images 2 and 4. Despite what the images 1 and 3 suggest, the plateau is not so flat. It seems that the global optimum lies somewhere in this apparently flat zone.



**Figure A1.** Quadratic criterion surfaces in identification of second order step response.

To solve the optimization problem, a numerical procedure is needed. Given the optimization criterion expression, an appropriate numerical recipe can be selected from the variable advancing step, gradient based class. The variable step allows fast jumps towards the global optimum in the flat zone. The Newton–Raphson method [31] is a good candidate. However, this method involves computing and inverting the Hessian matrix of quadratic criterion, which is both difficult and time consuming. The Gauss–Newton method is better, since the Hessian matrix is approximated by only using the gradient of quadratic criterion (see the details of this method in [31]). To implement the Gauss–Newton method, the gradient of quadratic criterion is needed. Fortunately, this gradient can be evaluated in closed form, as shown next. The gradient expression of is

$$\nabla V(\zeta, \omega, \tau) = \begin{bmatrix} \frac{\partial}{\partial \zeta} & \frac{\partial}{\partial \omega} & \frac{\partial}{\partial \tau} \end{bmatrix}^T V(\zeta, \omega, \tau) = -2 \sum_{n=N_0}^{N-1} [y(nT_s) - y_{II}^{(\zeta, \omega, \tau)}(nT_s)] \nabla y_{II}^{(\zeta, \omega, \tau)}(nT_s). \quad (\text{A7})$$

It suffices to derive the expression of second order response. The responses (A2)–(A5) are processed next. Note that all gradients below are null for  $t < \tau$ . Hence:

1. Harmonic oscillation ( $\zeta = 0$ ):

$$\nabla y_{II}^{(0, \omega, \tau)}(t) = [0 \quad (t - \tau) \sin[\omega(t - \tau)] \quad -\omega \sin[\omega(t - \tau)]]^T, \quad \forall t \geq \tau. \quad (\text{A8})$$

2. Under damped signal ( $\zeta \in (0, 1)$ ):

$$\begin{cases} \frac{\partial}{\partial \zeta} y_{II}^{(\zeta, \omega, \tau)}(t) = \frac{e^{-\zeta \omega (t-\tau)}}{1-\zeta^2} \left[ \omega(t-\tau) \cos[\omega_d(t-\tau)] - \frac{1}{\sqrt{1-\zeta^2}} \sin[\omega_d(t-\tau)] \right] \\ \frac{\partial}{\partial \omega} y_{II}^{(\zeta, \omega, \tau)}(t) = \frac{e^{-\zeta \omega (t-\tau)}}{\sqrt{1-\zeta^2}} \sin[\omega_d(t-\tau)] \\ \frac{\partial}{\partial \tau} y_{II}^{(\zeta, \omega, \tau)}(t) = -\frac{\omega e^{-\zeta \omega (t-\tau)}}{\sqrt{1-\zeta^2}} \sin[\omega_d(t-\tau)] = -\omega \frac{\partial}{\partial \omega} y_{II}^{(\zeta, \omega, \tau)}(t) \end{cases}, \quad \forall t \geq \tau. \quad (\text{A9})$$

3. Critically damped signal, ( $\zeta = 1$ ):

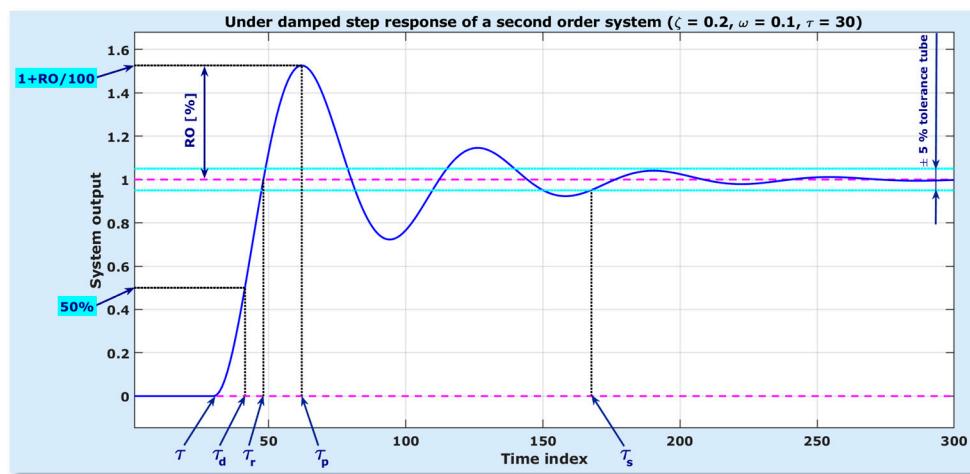
$$\nabla y_{II}^{(1, \omega, \tau)}(t) = [0 \quad \omega(t-\tau)^2 e^{-\omega(t-\tau)} \quad -\omega^2(t-\tau) e^{-\omega(t-\tau)}]^T, \quad \forall t \geq \tau. \quad (\text{A10})$$

4. Over damped signal ( $\zeta > 1$ ):

$$\begin{cases} \frac{\partial}{\partial \zeta} y_{II}^{(\zeta, \omega, \tau)}(t) = \frac{e^{-\zeta^+ \omega (t-\tau)} - e^{-\zeta^- \omega (t-\tau)}}{2\sqrt{(\zeta^2-1)^3}} + \omega(t-\tau) \frac{e^{-\zeta^+ \omega (t-\tau)} + e^{-\zeta^- \omega (t-\tau)}}{\zeta^2-1} \\ \frac{\partial}{\partial \omega} y_{II}^{(\zeta, \omega, \tau)}(t) = (t-\tau) \frac{e^{-\zeta^+ \omega (t-\tau)} - e^{-\zeta^- \omega (t-\tau)}}{2\sqrt{\zeta^2-1}} \\ \frac{\partial}{\partial \tau} y_{II}^{(\zeta, \omega, \tau)}(t) = -\omega \frac{e^{-\zeta^+ \omega (t-\tau)} - e^{-\zeta^- \omega (t-\tau)}}{2\sqrt{\zeta^2-1}} \end{cases}, \quad \forall t \geq \tau. \quad (\text{A11})$$

## Appendix B. Performance Parameters of Second Order Step Response

The quality of a step response provided by the order II system (A1) is analyzed according to following parameters, also indicated in Figure A2:



**Figure A2.** Main performance parameters of second order step response.

a. Time constant of second order system:

$$T = \frac{1}{\zeta \omega} [\text{s}]. \quad (\text{A12})$$

b. Relative overshoot (only in under damped case,  $\zeta \in (0, 1)$ ):

$$\text{RO} = 100 \exp\left(-\frac{\zeta \pi}{\sqrt{1 - \zeta^2}}\right) [\%]. \quad (\text{A13})$$

It shows how far from step reference the maximum of system response can rise. Therefore, RO is computed relatively to the reference height.

c. Peak time (only in under damped case,  $\zeta \in (0, 1)$ ):

$$\tau_p = \frac{\pi}{\omega \sqrt{1 - \zeta^2}} + \tau [\text{s}]. \quad (\text{A14})$$

It points to the time instant of when the response reaches its maximum. Note that the time delay  $\tau$  adds to the maximum point value.

d. Delay time,  $\tau_d$ . It stands for the necessary duration of response to rise from 0% to 50% of reference signal. In case of under damped response ( $\zeta \in (0, 1)$ ),  $\tau_d$  is:

$$\tau_d = \frac{1 + 0.7 \zeta}{\omega} + \tau [\text{s}]. \quad (\text{A15})$$

For the critically damped or over damped responses ( $\zeta \geq 1$ ), closed form expressions are impossible to derive. Therefore, in such cases, the equation that gives  $\tau_d$  can be solved by numerical algorithms. An easy way to approximate  $\tau_d$  is as follows: express the condition  $y(t) = 0.5$ ; expand the exponentials in Taylor series up to the 7-th term (as it is well known that the exponential is accurately approximated by such an expansion); compute the coefficients of polynomial that results and derive its roots; among them, there is a unique, real valued, positive root, which stands for  $\tau_d$ .

e. Rise time,  $\tau_r$ . This is the necessary duration of response to rise from a lower bound to a higher bound expressed as percentages from the reference. In case of under damped

response ( $\zeta \in (0, 1)$ ), the bounds are 0% and 100%, which leads to the following closed form expression:

$$\tau_r = \frac{\pi - \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)}{\omega\sqrt{1-\zeta^2}} + \tau [\text{s}]. \quad (\text{A16})$$

For the critically damped response ( $\zeta = 1$ ), the bounds are 5% and 95%, respectively. For over damped responses ( $\zeta > 1$ ), the bounds are 10% and 90%, respectively. Like in case of rise time, it is not possible to derive closed form expressions of  $\tau_r$ . Nevertheless, numerical procedures are available. The 7-th term expansion approach, previously described, can be considered as well.

- f. Settling time,  $\tau_s$ . This parameter is equal to the duration of transient zone, also known as *transient time*. Normally, the steady-state zone is supposed to begin when the response enters and remains within a tolerance tube around the reference. Usual tubes are form  $\pm 2\%$  to  $\pm 10\%$ ; in case of 10% tube width (i.e., by taking a  $\pm 5\%$  tolerance tube),  $\tau_s$  can be computed differently, depending on damping factor. If the response is under damped ( $\zeta \in (0, 1)$ ), then:

$$\tau_s = \tau - \frac{\log(0.5\sqrt{1-\zeta^2})}{\zeta\omega} = \tau - T \log(0.5\sqrt{1-\zeta^2}) [\text{s}]. \quad (\text{A17})$$

Otherwise, an empirical expression is currently employed (for  $\pm 5\%$  tolerance tube):

$$\tau_s = 3T + \tau [\text{s}]. \quad (\text{A18})$$

### Appendix C. Computing Fitness for Staircase References

The computing procedure of fitness is easier to understand following the example of Figure A3. Unlike for the outputs, for the inputs, the references need to consider the couple of output references at each time instant. Therefore, as the figure displays, seven zones have to be considered to extract input references from the static table. Each change of an output reference creates a delimiter for a new zone. In the example of Figure A3, each new zone preserves one output reference and changes the other one.

However, both output references can change at the same time as well, without affecting the input references extraction procedure. The main zones 1, 3, 5 and 7 have longer durations than the secondary zones 2, 4 and 5.

Like in the case of single step references, there are three fitness components to compute two for the outputs ( $\phi_1^y, \phi_2^y$ ) and one for both inputs  $\phi_{12}^u$ .

For each output, the fitness is computed by processing all the steps of corresponding staircase. Let  $N_j^s$  be the number of steps exhibited by the staircase of channel  $j \in \{1, 2\}$  (e.g.,  $N_1^s = N_2^s = 4$ , as in Figure A3). Then:

$$\phi_j^y \equiv \frac{1}{N} \sum_{i=1}^{N_j^s} n_i \phi_{j,i}^y, \quad (\text{A19})$$

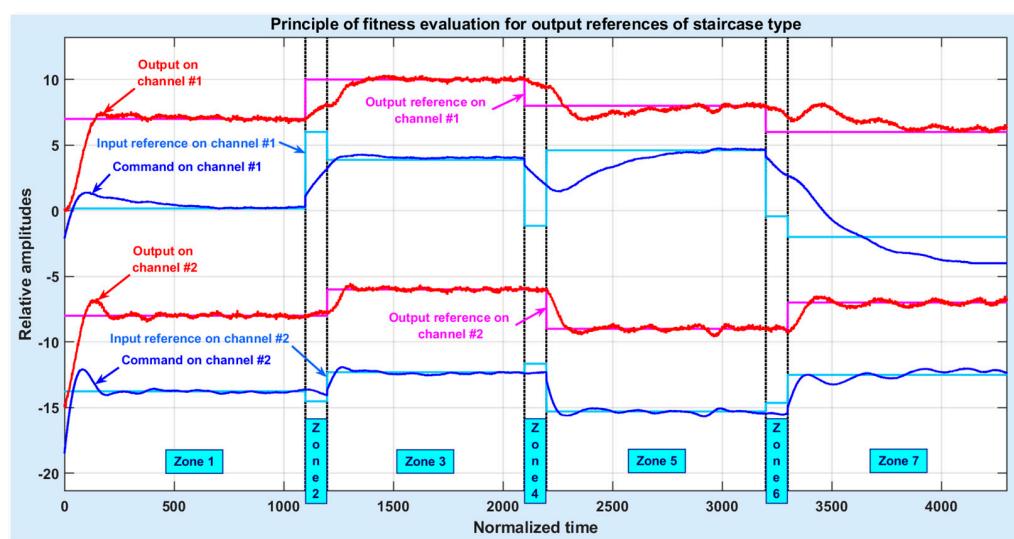
where  $n_i \in \mathbb{N}^*$  is the duration of current step  $i \in \overline{1, N_j^s}$ . Obviously,  $\sum_{i=1}^{N_j^s} n_i = N$  (the duration of all signals). In the weighted average of (A19), the fitness of current step,  $\phi_{j,i}^y$ , is computed like in case of single step output reference (see Equation (17)). For the two input signals, the fitness is a weighted average as well

$$\phi_{12}^u \equiv \frac{1}{N} \sum_{i=1}^{N_{12}^s} n_i \phi_{12,i}^u, \quad (\text{A20})$$

where  $N_{12}^z$  is the number of zones to extract the input references for (e.g., 7 in the previous figure);  $n_i \in \mathbb{N}^*$  is the duration of current zone  $i \in \overline{1, N_{12}^z}$  (with  $\sum_{i=1}^{N_{12}^z} n_i = N$ );  $\phi_{12,i}^u$  is the fitness of current zone  $i \in \overline{1, N_{12}^z}$ , computed like in case of step output references (see Equation (A20)).

Overall, the fitness is computed by considering that the controllers are optimal. Therefore, in the refined definition of step references fitness (22), only the third case is employed

$$\Phi \equiv \alpha(\phi_1^y + \phi_2^y) + \beta \phi_{12}^u. \quad (\text{A21})$$



**Figure A3.** Principle of fitness evaluation for output references of staircase type.

## References

- Bhookya, J.; Jatoh, R.K. Fractional Order PID Controller Design for Multivariable Systems using TLBO. *Chem. Prod. Process. Model.* **2020**, *15*. [[CrossRef](#)]
- Jitwang, T.; Puangdownreong, D. Application of Cuckoo Search to Robust PIDA Controller Design for Liquid-Level System. *Int. J. Innov. Comput. Inf. Control.* **2020**, *16*, 189–205. [[CrossRef](#)]
- Soued, S.; Ebrahim, M.A.; Ramadan, H.S.; Becherif, M. Optimal Blade Pitch Control for Enhancing the Dynamic Performance of Wind Power Plants via Metaheuristic Optimizers. *IET Electr. Power Appl.* **2017**, *11*, 1432–1440. [[CrossRef](#)]
- Coelho, L.D.; Mariani, V.C. Firefly Algorithm Based on Chaotic Tinkerbell Map Applied to Multivariable PID Controller Tuning. *Comput. Math. Appl.* **2012**, *64*, 2371–2382. [[CrossRef](#)]
- Rodriguez-Molina, A.; Mezura-Montes, E.; Villarreal-Cervantes, M.G.; Aldape-Perez, M. Multi-Objective Meta-Heuristic Optimization in Intelligent Control: A Survey on the Controller Tuning Problem. *Appl. Soft Comput.* **2020**, *93*. [[CrossRef](#)]
- Vijayakumar, K.; Manigandan, T. Nonlinear PID Controller Parameter Optimization using Enhanced Genetic Algorithm for Nonlinear Control System. *Contr. Eng. Appl. Inform.* **2016**, *18*, 3–10.
- Ziegler, J.G.; Nichols, N.B. Optimum Settings for Automatic Controllers. *Trans. ASME* **1942**, *64*, 759–768. [[CrossRef](#)]
- Moharam, A.; El-Hosseini, M.A.; Ali, H.A. Design of Optimal PID Controller Using Hybrid Differential Evolution and Particle Swarm Optimization with an Aging Leader and Challengers. *Appl. Soft Comput.* **2016**, *38*, 727–737. [[CrossRef](#)]
- Janprom, K.; Permpoonsinsup, W.; Wangnippaparto, S. Intelligent Tuning of PID Using Metaheuristic Optimization for Temperature and Relative Humidity Control of Comfortable Rooms. *J. Control. Sci. Eng.* **2020**. [[CrossRef](#)]
- Hernandez-Osorio, M.A.; Ochoa-Velasco, C.E.; Garcia-Alvarado, M.A.; Escobedo-Morales, A.; Ruiz-Lopez, I.I. Sequential synthesis of PID controllers based on LQR method. *Rev. Mex. Ing. Quim.* **2020**, *19*, 913–928. [[CrossRef](#)]
- Pradhan, J.K.; Ghosh, A.; Bhende, C.N. Two-Degree-of-Freedom Multi-Input Multi-Output Proportional-Integral-Derivative Control Design: Application to Quadruple-Tank System. *J. Syst. Control Eng.* **2019**, *233*, 303–319. [[CrossRef](#)]
- Serradilla, F.; Canas, N.; Naranjo, J.E. Optimization of the Energy Consumption of Electric Motors through Metaheuristics and PID Controllers. *Electronics* **2020**, *9*, 1842. [[CrossRef](#)]

13. Azar, A.T.; Ali, N.; Makarem, S.; Diab, M.K.; Ammar, H.H. Design and Implementation of a Ball and Beam PID Control System Based on Metaheuristic Techniques. In Proceedings of the 5th International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 26–28 October 2019.
14. Wang, P.C.; Zhou, Y.Q.; Luo, Q.F.; Han, C.; Niu, Y.B.; Lei, M.Y. Complex-Valued Encoding Metaheuristic Optimization Algorithm: A Comprehensive Survey. *Neurocomputing* **2020**, *407*, 313–342. [[CrossRef](#)]
15. Khanduja, N.; Bhushan, B.; Mishra, S. Control of CSTR Using Firefly and Hybrid Firefly-Biogeography Based Optimization (BBFFO) Algorithm. *J. Inf. Optim. Sci.* **2020**, *41*, 1443–1452. [[CrossRef](#)]
16. Angel, L.; Viola, J.; Vega, M. Metaheuristic Tuning and Practical Implementation of a PID Controller Employing Genetic Algorithms. In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019.
17. Neeraj, P.J.; Seema, P.N. Control of Twin Rotor MIMO System using Cross PID Control Technique. In Proceedings of the IEEE International Conference on Communication and Signal Processing, Melmaruvathur, India, 3–5 April 2018; pp. 744–750.
18. Guo, H.R.; Feng, Z.Y.; She, J.H. Discrete-Time Multivariable PID Controller Design with Application to an Overhead Crane. *Int. J. Syst. Sci.* **2020**, *51*, 2733–2745. [[CrossRef](#)]
19. Stefanou, D.; Borne, P.; Popescu, D.; Filip, F.G.; El Kamel, A. Optimization in Engineering Sciences—Metaheuristics, Stochastic Methods and Decision Support. John Wiley & Sons: London, UK, 2014; ISBN 978-1-84821-498-9.
20. Souza, D.A.; de Mesquita, V.A.; Reis, L.L.N.; Silva, W.A.; Batista, J.G. Optimal LQI and PID Synthesis for Speed Control of Switched Reluctance Motor Using Metaheuristic Techniques. *Int. J. Control. Autom. Syst.* **2021**, *19*, 221–229. [[CrossRef](#)]
21. Goud, H.; Swarnkar, P. Investigations on Metaheuristic Algorithm for Designing Adaptive PID Controller for Continuous Stirred Tank Reactor. *MAPAN J. Metrol. Soc. India* **2019**, *34*, 113–119. [[CrossRef](#)]
22. Vanchinathan, K.; Valluvan, K.R. A Metaheuristic Optimization Approach for Tuning of Fractional-Order PID Controller for Speed Control of Sensorless BLDC Motor. *J. Circuits Syst. Comput.* **2018**, *27*. [[CrossRef](#)]
23. Storn, R.; Price, K. Differential Evolution a Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
24. Nedic, N.; Stojanovic, V.; Djordjevic, V. Optimal Control of Hydraulically Driven Parallel Robot Platform Based on Firefly Algorithm. *Nonlinear Dyn.* **2015**, *82*, 1457–1473. [[CrossRef](#)]
25. Yang, X.S.; Deb, S. Cuckoo Search Via Levy Flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 9–12 December 2009; pp. 210–214.
26. Cheng, M.Y.; Prayogo, D. Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [[CrossRef](#)]
27. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
28. Culita, J.; Stefanou, D.; Dumitrescu, A. ASTANK2: Analytical Modeling and Simulation. In Proceedings of the 20-th International Conference on Control Systems and Computer Science, CSCS-20, Bucharest, Romania, 27–29 May 2015; Volume 1, pp. 141–148. [[CrossRef](#)]
29. Stefanou, D.; Culita, J. Joint Stochastic Spline and Autoregressive Identification Aiming Order Reduction Based on Noisy Sensor Data. *Sensors* **2020**, *20*, 5038. [[CrossRef](#)]
30. Yang, X.S. YAN2009b Firefly Algorithm for Multimodal Optimization. In *Stochastic Algorithms: Foundation and Applications*; (Ed.) SAGA 2008; Series: Lectures Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
31. Borne, P.; Popescu, D.; Filip, F.G.; Stefanou, D. *Optimization in Engineering Sciences—Exact Methods*; John Wiley & Sons: London, UK, 2013; ISBN 978-1-84821-432-3.
32. Baker, J.E. Adaptive Selection Methods for Genetic Algorithms. In Proceedings of the first International Conference on Genetic Algorithms and Applications; Grefenstette, J.J., Ed.; Erlbaum Printing House: New York, NY, USA, 1985.