

Article

# Fractional-Order PID Controller (FOPID)-Based Iterative Learning Control for a Nonlinear Boiler System

Ahmed Zubair Jan <sup>1,\*</sup>, Krzysztof Kedzia <sup>1</sup> and Muhammad Jamshed Abbass <sup>2</sup>

<sup>1</sup> Faculty of Mechanical Engineering, Wrocław University of Science and Technology, 50-370 Wrocław, Poland

<sup>2</sup> Faculty of Automation, Electronic and Electrical Engineering, Wrocław University of Science and Technology, 50-370 Wrocław, Poland

\* Correspondence: ahmed.jan@pwr.edu.pl

**Abstract:** For the boiler-turbine unit in power systems, a coordinated control structure plays a crucial role in maintaining the balance in supply and demand of energy, reducing pollutant emissions and optimizing energy efficiency. The matching requirements of the turbine and boiler, the wide range of load changes, and the cooperative operation of many devices in the power system pose many challenges to designing a coordinated control system for the boiler turbine system, thus making the control design a difficult task. In this paper, iterative learning control (ILC) is used to maintain the throttle pressure and megawatt output power of a boiler turbine at the desired set points by controlling the hybrid pattern design structure. Simulation results show that the proposed approach can maintain the desired set points, and the desired response can also be obtained faster by using the proposed approach compared to the ones available in the literature.

**Keywords:** boiler turbine unit; coordinated control; throttle pressure control; iterative learning control; optimization of energy efficiency

## 1. Introduction

The global energy crisis is one of the most pressing issues. A large amount of research has been presented to optimize energy usage and find new sources of energy production. Renewable energy is one of the replacements and has an impact on green energy. The world has invested a lot in renewable energy units like solar and wind. These units have filled the demand for energy to some extent, but renewable energy plants have some issues with stability, standalone functionality, and immediate response to load changes. Solar plants, for example, are without a doubt the most environmentally friendly renewable energy sources, but they cannot operate independently. They require power backup from a battery bank or the power grid [1]. Because of this, the coal-fired thermal power plant has a distinctive position due to its efficient operation, stability with changes in load, and efficient or optimizing control [2]. Various types of coal turbine units are utilized according to the user requirements and location of the plants. Some plants produce steam, and this high-pressure steam runs the turbine to produce electricity; the rest of the low-pressure steam can also be utilized in the process industry and space heating as per requirement. Coal power generation has a much lower cost than other conventional plants. Boiler plants have many input/output devices and need an efficient control system for smooth operation. The latest distribution type controller is used for controlling purposed [3]. The distributive control system consists of sub-control systems that control the plants' sub-parts independently and send feedback and data to the main HMI (Human Machine Interface) in the DCS (distributed control system). According to the output load requirement of the concerned sub-control system, these sub-control systems run their subsystems independently and give feedback to the main pannel [4–9]. In any case, the chances of a full system collapse are very slim, and rectification of the fault is much more comfortable than a centralized control system. The DCS architecture has independent control loops, so the time cycle for



**Citation:** Jan, A.Z.; Kedzia, K.; Abbass, M.J. Fractional-Order PID Controller (FOPID)-Based Iterative Learning Control for a Nonlinear Boiler System. *Energies* **2023**, *16*, 1045. <https://doi.org/10.3390/en16031045>

Academic Editor: Xiaofeng Lu

Received: 14 November 2022

Revised: 7 December 2022

Accepted: 21 December 2022

Published: 17 January 2023



**Copyright:** © 2023 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/>).

executing the instructions or changing the load is much shorter. The balance between load and plant generation must be considered in boiler generation. Synchronization between the boiler and turbine and coordination between all input and output devices and the sub-control system are also important parts of boiler generation. The above goals are not easily achieved in boiler plants due to the lack of coordinated and synchronized boiler turbine operation. To achieve a goal, all sub-control system units must work in concert. As a result, in order to provide electricity to all users efficiently, smoothly, steadily, and with fewer losses, a centrally coordinated control system is required. In this work, a simple yet efficient control scheme is proposed to ensure stability, robustness, and disturbance rejection due to abrupt load changes shown in Figure 1. Keeping in view the design requirements, an FOPID controller combined with an ILC is proposed in this work to ensure the efficient control of the boiler turbine system. Designing a control system that can respond quickly while keeping up with changing load demands is a big challenge. This is because the control system needs to be able to keep track of all the different parts of the system, such as the demand-side requirements, and make sure they are met as quickly as possible.

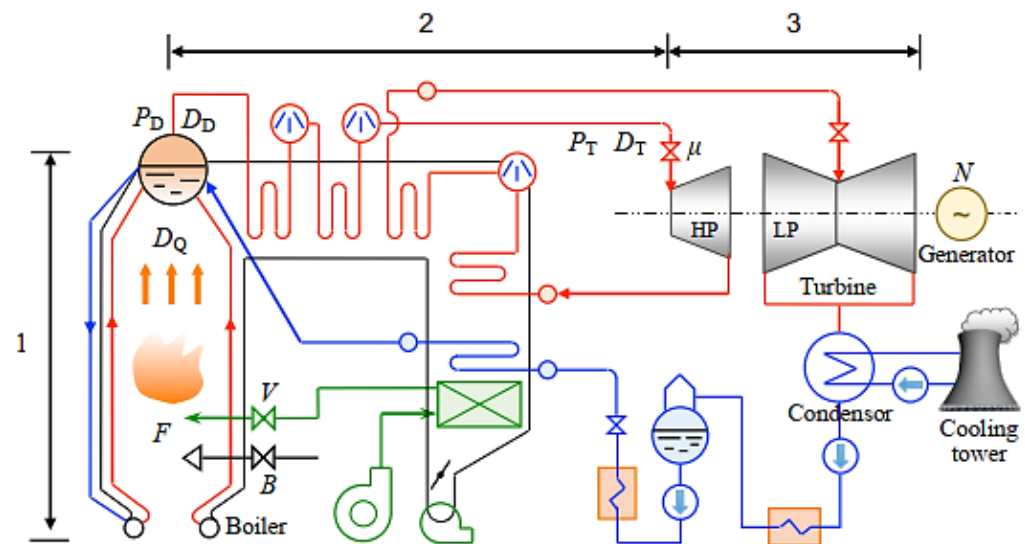


Figure 1. Boiler-turbine unit (schematic diagram) [9]; 2011, IEEE.

A robust intelligent control structure must adapt itself and generate a control law that can govern the generation units to satisfy the load demand in a considerably short amount of time. In this work, a FOPID-based ILC is designed for reference tracking with minimum overshoot and steady-state error [5]. Efforts have been made in the literature to design such control laws for different generating units, including solar, wind, diesel generators, and boiler turbine systems. All in all, the boiler-turbine system is a slow process, as heating the water, converting it into steam, and then using that steam to generate power take a considerable amount of time. Simultaneously, fast-changing loads necessitate the power generated by the boiler turbine following itself. Thus, a robust, intelligent, and learning controller is required to adapt its control law in accordance with the power demand and ensure that desired set points, i.e., power, are successfully tracked in a shorter amount of time. This research aims to design an intelligent learning controller for the boiler turbine unit to meet the power demand of the fast-changing load.

## 2. Literature Review

In [6] the decentralized controller technique is applied to the drum boiler system. The drum boiler system has 03 inputs and 03 outputs, and the inputs are the flow of water, the fuel flow in the firebox, and the user load. In these three inputs, only water and fuel flow are the control inputs, while load is the disturbance input. The system's outputs are

water level in the drum, steam pressure, and the ratio of air to fuel; however, the third input and output are not utilized while the output load is considered as constant, which shows the limitations of the work. This system controller is used to control the overshoot and settling time, and finally, in the results, it was shown that this nonlinear model has zero overshoot and the shortest settling time. In [6] the adaptable nonlinear controller technique is applied to a multi-input, multi-output nonlinear boiler system. The results show that this design has good tracking ability with zero steady-state error. In [9] the mismatch between the command and the actual value is addressed. The PID controller applied is the value of  $\alpha$  and  $\beta$  which are integers, but when we apply FOPID, the values of  $\alpha$  and  $\beta$  may be utilized in fraction order, giving more precision in the results. It is also very well known that the standalone  $H_\infty$  controller technique is not suitable for real-time applications, especially for the boiler turbine unit. In [10] it is studied with respect to efficient combustion in the chamber we want combustion to be efficient at any cost, and an optimized combustion for the thermal plant is proposed in [11]. It has certain limitations as it gave only one algorithm, i.e. ES controller, which behaves accurately only when the controller varying speed is slower than the plant's slowest operating point speed and faster regarding the model changes rate.

In [12] it was presented regarding the working principle of incinerators was presented in which waste is used as fuel to produce steam with very high pressure to produce power. These systems or plants decentralized PID controllers are not suitable because they do not deal effectively with input side disturbance, which eventually shuts down the whole plant and produces hazardous unincinerated material. Various other advanced control techniques are utilized for boiler control such as  $H_\infty$ , predictive control, fuzzy control, etc., but in the said case or plant, we are not well aware of the types of waste material content, so the neural adaptive controller is more efficient and stable. In [12] it is presented that in real-time systems, the load of the user can be changed at any time, so the controller must be intelligent enough to absorb changes with certain limitations, which we have already defined for the controller. For example, if we take the one perimeter of the boiler for understanding the level of water in the drum because this water is responsible for producing steam pressure at a certain required value, then one of the jobs of controller is to maintain the level of water in boiler drum so that pressure of steam easily be controlled, so in case of any change of load, the system has the capabilities to accommodate the changes. In [13–16] a feedforward link is used to improve the control scheme, but research clearly showed that the robustness of the controller is not considered in his scheme and the  $1 \times$  control variable is not taken into account in the feedback loop.

In [17] we study the basic operation of the PID controller for multi-input, multi-output (MIMO) boiler systems; as we know our boiler plant also has multi-input and multi-outputs. In our investigation, initially I applied the PID controller, but as we already discussed, the PID controller has values of  $\alpha$  and  $\beta$  are integers, but when we apply FOPID, the values of  $\alpha$  and  $\beta$  can be utilized in fraction orders in this way, and we obtained better results due to the better accuracy factor of the valve controlled by FOPID. In [18], it is presented that any intelligent technique, that is, the fuzzy control system, the neural control system and the ILC, will give the most efficient and optimal values of air & coal in the combustion chamber and also ensure efficient fuel use. In real-time processes, intelligent techniques use the self-learning method to update and optimize all inputs so that any disturbance or change in load never affects the system or the controller. The work of Arimoto [19,20] and Cryer is very important in bringing up this concept of ILC. The primary purpose of ILC is to improve the transient response of the system, especially for those who continue to repeat the action over and over again within a fixed time interval. Different types of ILC are developed, each with its own merits and demerits in a different scenario. The use of error derivatives to update the control input was used in the robot manipulator. The most influential work on ILC is carried out in the field of robotics. As in robots, mostly working in industries, they have assigned a single task that had to be performed with great accuracy. ILC provides the best possible solution for these types of system. Different types of controller's techniques

were proposed in the past for controlling boiler unit like robust control  $H_\infty$ , fuzzy logic, adaptive techniques, and sliding mode nonlinear control, these all mentioned techniques have good enough tracking and error reduction abilities in a specific scenario and mostly applicable only in just simulations environment, but when we look at industrial control systems, then they have certain limitations in real plant environment like online tuning and ease of implementations during real-time running plant. In it was already proved that the practical use of the PID controller for the boiler turbine unit in combination with any intelligent technique is good enough for real-time systems, and all other techniques like  $H_\infty$  along with LQR have specific applications and limitations. In our model, we proposed a more precise controller model, so we utilized the FOPID controller instead of the PID due to limitations of the PID controller in the case of controlling the angle values of the steam valve along with the ILC technique for faster set point tracking to compensate for changes in load with optimum matching requirements between a boiler and a turbine.

### 3. Boiler System and Control Design

The energy conversion and transfer process of a coal-fired boiler-turbine unit, can be divided into three processes. The combustion and heat transfer process in the furnace can be described as:

$$\delta D_Q(s) = \frac{k_1}{(T_1s + 1)(T_2s + 1)} \delta B(s) \quad (1)$$

Pipe Transfer Process; When we treat the boiler and the main steam pipes as a concentrate thermal storage container, the pipe transfer process can be described as

$$\Delta D_Q(t) - \Delta D_T(t) = C \frac{(d\Delta P_D(t))}{dt} \quad (2)$$

$$P_D(t) - P_T(t) = k_T D_T^2(t) \quad (3)$$

$$D_T(t) = k(T)\mu(t)P_T(t) \quad (4)$$

Turbine Working Process; For a turbine with a re-heater, the dynamic transfer function of its working process can be given by:

$$N(s) = \frac{(\alpha T_s)_3s + 1}{(T_3s + 1)} k_2 * D_T(s) \quad (5)$$

The coefficients in Equations (1)–(5) are described in Table 1.

The dynamic diagram of the three processes is shown in Figure 2. This is the so-called Chere model [12,14], a well-known simplified nonlinear model for a boiler-turbine unit. The model has been tested by Chere's in 5 different capacity units and has been widely recognized and applied in control system analysis and design for over 20 years.

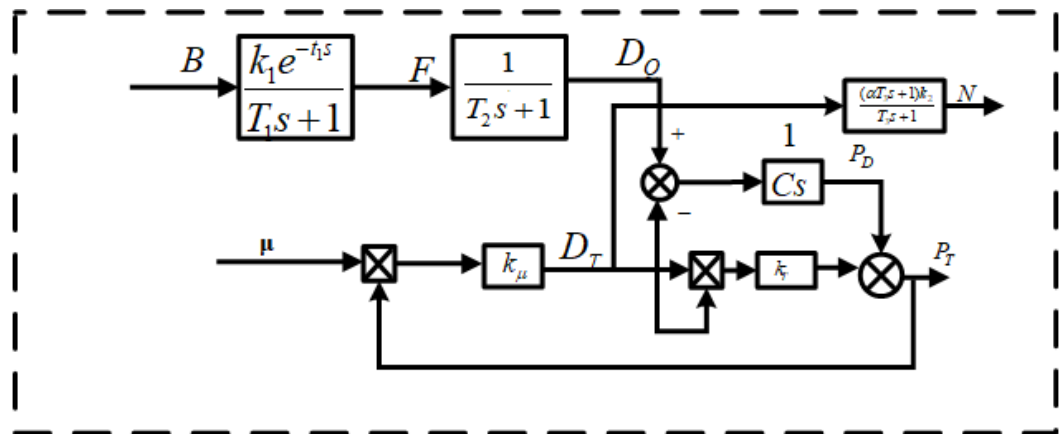


Figure 2. Overall dynamics of the system.

In Figure 2, the non-linear system has been linearized as [16]. The parameter values used in the boiler, fuel, and turbine processes are given in Table 1.

Table 1. Values of system parameters.

Parameter	Value	Parameter	Value
$T_o$	135	$T_1$	150
$T_3$	6	$T_b$	30
$k_1$	1	$k_\mu$	1
$\alpha$	0.25	$\mu_o$	6.89
$P_{T_o}$	105	-	-

The above system of equations for a boiler turbine unit can be converted to the state-space model. The generalized state-space representation of the boiler turbine system can be given as follows:

$$\dot{x} = Ax + Bu \tag{6}$$

#### 4. Control Design

This section deals with the controller design of the proposed method, which is the FOPID controller design based on ILC for the boiler system. This section covers some background knowledge of the FOPID controller and Iterative Learning Control (ILC), assumptions, and basic update laws used in ILC algorithms. The importance of the proposed technique with previous methods is also discussed. Subsequently, the simulation results of the proposed technique for different reference signals are presented, and a comparative study with simple ILC and FOPID is included to get a clear idea of the advantages of the proposed method over conventional methods. In 1999, GorPodlubny [13,17] proposed the FOPID Controller. It contains a fractional-order integrator and a fractional-order differentiator. The common mathematical formula of FOPID is:

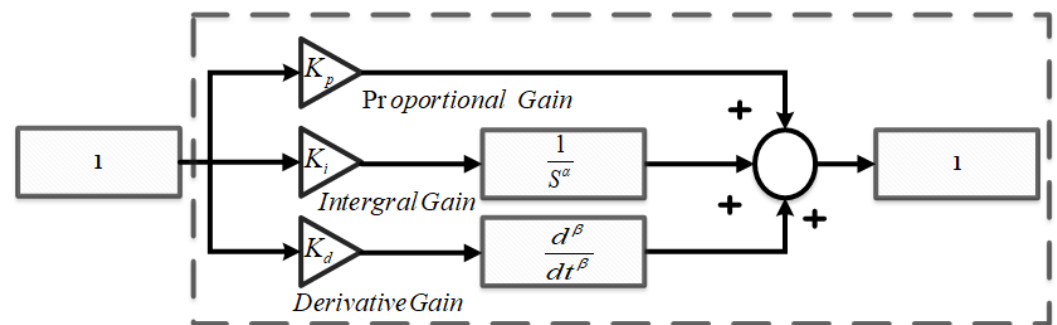
$$G_{FOPID}(s) = k_p + \frac{k_i}{s^\alpha} + k_d s^\beta \tag{7}$$

where  $k_p$ = proportional gain,  $k_i$  = integral gain and  $k_d$ = derivative gain while  $\alpha$  and  $\beta$  are the fractional powers of integrator and differentiator, respectively. Clearly, from the above Equations (6) and (7). FOPID has five unknown parameters, three of PID controller, and two additional parameters  $\alpha$  and  $\beta$  lead to various controllers. There are many classical and modern techniques to tune these parameters. The stability margins of the system are improved when modeled in fractional order and preserve robustness compared to classical PID [14,18]. A fractional-order controller representation is presented in Figure 3. FOPID controllers are popular due to the following provisions;

- It is used to eliminate steady-state error;

- Provides gain cross-over frequencies and phase margin, ensure performance even when the system's gains are changing;
- It has the disturbance rejection capability. The art of learning has been started since the evolution of humanity.

Being an engineer, if we are in charge of a process that runs several times and the same error appears, we might think of some learning mechanism that can reduce this error through some algorithm.



**Figure 3.** Block diagram of FOPID.

There are several applications where the process is repeated, following the same action of the plane, such as the chemical batch process and in robot manipulators. The feedback system's main issue is that if the user repeats the process hundreds of times, it will produce the same response and error every time. But ILC uses the information from the past attempt and tries to reduce in the next attempt. The control input from the previous attempt is saved in memory, and with some function on error, it is added to the previous control signal to get an accurate tracking, also which can be called a perfect tracking. Any effort can be used to create a new control input for the present trial. One can use P, PD, PI, or simply proportional, integral, and derivative control laws to update the new control input [14,16]. For the design of ILC, few assumptions are made to improve learning from the previous learning mechanism. These assumptions are listed as follows. The starting point of every iteration will always remain the same, that is, if the system starts from  $t = 0$ , having a magnitude of zero as the initial condition, then all trials must begin with the same initial condition all the time. The error should be converged after every iteration, which is the error of the second trial must be less than the error of the first trial and so on and the time for each iteration should be the same, which is if first trials were of duration 10 s, then all other trails will also have the same time. These assumptions must be followed to perfectly follow the system. In Figure 4 below shows the basic structure of ILC in which  $u(i, k)$  is the current control input,  $y(i, k)$  is the current output, and  $u(i, k + 1)$  is the new control input generated by ILC. The  $(i, k)$  represents the  $i^{th}$  time interval and  $k^{th}$  batch or iteration.  $u(i, k)$  is given to the system and the required output response of the system is obtained. All values are stored in a memory and a new control input is generated in such a way that the desired signal  $y^d(i)$  is perfectly tracked [16]. There are different updating laws for designing ILCs. Different scholars have used different ways to design ILC. Here, PID type ILC used, Integral gives extra benefits in error convergence because it removes the steady-state error from the response. As PID is widely used in feedback systems and its practical results in the process industries, gives a motivation to designing a PID type ILC. The mathematical form of the PID type ILC is:

$$u(i, k + 1) = u(i, k) + k_p e(i, k) + k_d \dot{e}(i, k) + k \int e(i, k) dt \quad (8)$$

Here a complete PID update equation is created to generate a new control input for the system [17]. There are different ways in which ILC can be applied to a system. One thing should always be in mind that ILC is only applicable to a stable system. So, if the

system is unstable, some feedback mechanism should cooperate with a system to make it stable, then further ILC can be used to improve the transient response as much as possible by keeping the system's saturation limits. ILC is not used mainly independently because almost all real-time systems are unstable. ILC gives open-loop control for the system, so if the system is unstable, an error will never minimize, and finally the desired response can never be attained. Some basic configurations are used in ILC shows in Equation (8). They are divided into two main categories, embedded and cascaded. In embedded, ILC is used by making some changes in the system's actual loop, while in cascaded, ILC is incorporated separately without disturbing any existing system configuration. Out of the basic ILC structure in literature, learn from previous cycle. Experiments on learning through the previous cycle do not cover up the disturbances present in the current cycle, so some feedback mechanism should be used to cope with the uncertainties. This can be done by closing the control loop in the time domain. Below is the block diagram of such a system in Figure 4.

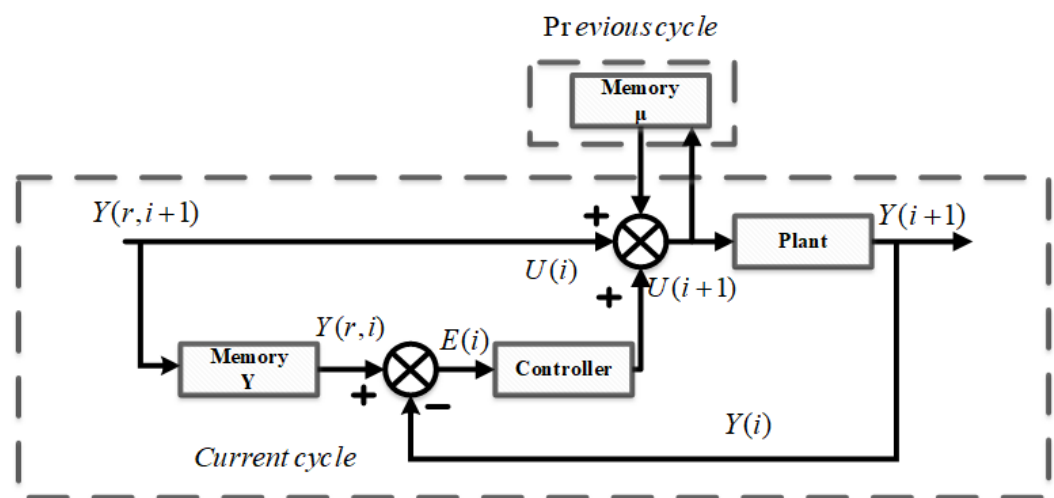


Figure 4. ILC learning through the previous cycle.

In all those types, the system's present control loop has to be restructured, and a feed-forward ILC is added to the loop. So, if ILC is to be used with an existing control process to make it more efficient, then the whole programming of the loop is to be re-done, or some time the microcontroller used in the loop has to be replaced. In a real-time process, the microcontroller's replacement or rewriting executional code is not acceptable due to many factors such as cost, security, and other constraints. For example, in many control processes, we have to control the desired set points; that is, the controller is used to tuning for the setpoints. In these scenarios, cascade ILC is the best option to achieve the desired results. The block diagram of such a system is shown in Figure 5. Here, ILC is used as an external controller without disturbing the existing control process. ILC uses the desired signal and the error stored from the previous cycle to make a new reference trajectory for the existing control process. No change in the existing loop has to be done, and just a few commands for the desired signal have to be rewritten. In cascaded ILC, the next trial will be better than the previous trial if the convergence condition holds.

The ILC in cascade form has many advantages compared with traditional ILC, which is open-loop in nature and generates the control signal that tries to track the reference trajectory as possible. In cascaded ILC, ILC iterates the reference signal that is best suited to attain the desired response. The advantage of cascaded ILC is that the use of a feedback controller reduces the learning process of the ILC because the existing loop controller allows the system to remain as close to the reference trajectory. Such an arrangement speeds up the entire learning process. In other words, the existing controller loop is used for fast initial compensation of error while ILC iterates the reference signal to further minimize the error. Using ILC for the outer loop, the condition for tuning the existing loop through complex

schemes to attain the desired result is not required. Complex processes are challenging to model in an experimental environment. Therefore, cascaded ILC provides an alternative to achieve perfect tracking using simple control strategies. In this proposed research, FOPID is used as an existing controller, while ILC is used as a new input controller combined with the existing system.

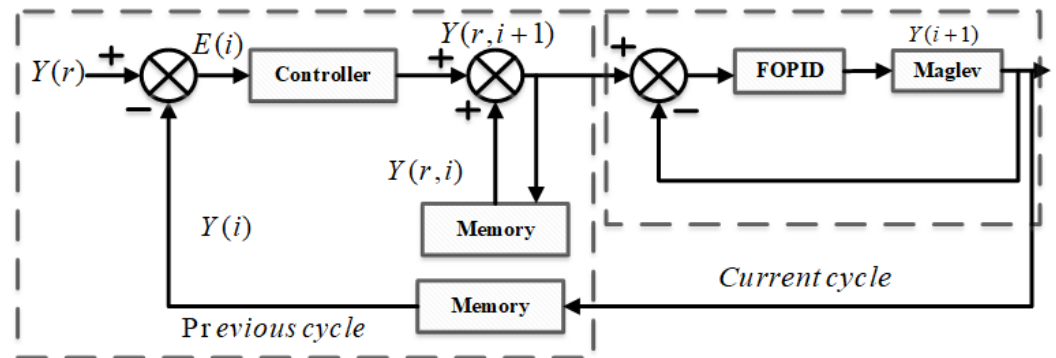


Figure 5. Block diagram of cascaded ILC.

### 5. Results and Discussion

FOPID has several advantages over PID controllers. As discussed earlier that two additional parameters  $\alpha$  and  $\beta$ , the dominant pole calculation is done in this work to formulate the objective function, then the "nonlinear interior point optimization method" is used for the minimization of the cost function to find the value of the parameters  $k_p, k_i$  and  $k_d$  of controller [14]. The design specifications for this system are:

$$\begin{aligned} \text{Damping ratio} &= \zeta = 0.9 \\ \text{Settling time} &= t_s = \frac{4}{\zeta\omega_n} \leq 1s \end{aligned}$$

According to the above specifications of the system, the dominant poles are obtained from the characteristic equation ( $s^2 + 2\sigma\omega_n s + \omega_n^2$ ) are:

$$s_{1,2} = -3.996 \pm 1.934i \tag{9}$$

The objective function is minimized to obtain the FOPID controller parameters:

$$f = |R| + |I| \tag{10}$$

where "R" is the real part of the characteristic equation  $1 + G_p(s_1)G_c(s_1) = 0$  when we put the value of  $s_1$  and "I" is the imaginary part of the equation.  $G_c(s_1)$  is FOPID controller transfer function, which is equal to:

$$k_p + \frac{k_i}{s^\alpha} + k_d s^\beta \tag{11}$$

and  $G_p(s_1)$  is the system transfer. The closed-loop illustration of the system with the FOPID controller is shown in Figure 6.



Figure 6. Block diagram of the feedback system.



The characteristic equation of the closed-loop system with the FOPID controller is the following:

$$1 + G_p(s)G_c(s)^{\alpha,\beta} = 0 \quad (12)$$

$$1 + \left( \frac{-1.226 \times 10^4}{s^3 + 231.9s^2 - 1687s - 3.912 \times 10^5} \right) (k_p + \frac{k_i}{s^\alpha} + k_d s^\beta) = 0 \quad (13)$$

By putting the value of  $s$  from a given equation, we have the following equation.

$$1 + \frac{(-1.226 \times 10^4)}{(-3.996 + 1.934i)^3 + 231.9(-3.996 + 1.934i)^2 - 1687(-3.996 + 1.934i)(-3.912 \times 10^5)} (k_p + \frac{k_i}{(-3.996 + 1.934i)^\alpha} + k_d(-3.996 + 1.934i)^\beta) = 0 \quad (14)$$

Now real and imaginary parts can be extracted:

$$R = (1 + 0.031)k_p + \left( \frac{0.0316}{4.394^\alpha} \times \cos(25.82\alpha) \right) - \frac{2.8435 \times 10^{-7}}{4.394^\alpha} \times \sin(25.82\alpha) k_i + (0.0316 \times 4.394^\beta \times \cos(25.82\beta) - 2.8434 \times 10^{-7} \times \sin(25.82\beta))k_d \quad (15)$$

$$I = -2.8437 \times 10^{-7}k_p + \left( -\frac{0.0316}{4.394^\alpha} \times \sin(25.82\alpha) - \frac{2.8435 \times 10^{-7}}{4.394^\alpha} \times \cos(25.82\alpha) \right) k_i + (0.0316 \times 4.394^\beta \times \sin(25.82\beta) - 2.8434 \times 10^{-7} \times \cos(25.82\beta))k_d \quad (16)$$

The objective function  $f = |R| + |I|$  is considered to find the parameters  $k_p, k_i, k_d, \alpha$  and  $\beta$  of the FOPID controller. The algorithm for the “nonlinear interior-point method” is given as; Minimize  $h_i(x) = 0$  where,  $i = 1, 2, 3, 4, \dots, p$ ,  $g_j(x) \leq 0$ , and  $j = 1, 2, 3, \dots, m$   $f(x)$  is objective function,  $h_i(x)$  represent equality constraints and  $g_j(x)$  represents inequality constraints, and the internal penalty function is as follows:

$$p(x, \lambda_k) = f(x) + \lambda_k \sum_{j=1}^m -\frac{1}{(g_j(x))} f \quad (17)$$

In the above equation,  $\lambda_k$  is the penalty coefficient and  $p(x, \lambda_k)$  is a penalty function. Table 2 represents the initial estimate for the FOPID controller parameters, the limits of the controller parameters, and the final values of the controller parameters obtained after optimization of a cost function using the FMINCON toolbox in MATLAB.

**Table 2.** FOPID parameter (ranges).

Parameter	Initial Guess	Lower Value	Upper Value	Final Value
$k_p$	-50.95	-51.15	-50.35	-50.45
$k_i$	-225.55	-225.95	-224.9	-225.8
$k_d$	-6.9	-7.1	-6.4	-7.086
$\alpha$	0.80	0.2	0.975	0.900
$\beta$	0.60	0.1	0.65	0.5497

To test the performance of the proposed controller, in this section a practical 500 MW nonlinear boiler-turbine model is used in [1], as shown in Figure 6, is used in this section. The system considered in this work is taken from [1]. The aim is to maintain the desired level of  $PT_r$  for a specific time, then reduce its level and maintain that, that is, PT will remain 17 MPa for 100 s and then suddenly reduce to 14 MPa and maintain there, N is 500 MW initially after 100 s, and then it is reduced to 380 MW. It should be noted that the changes in the desired trajectory of PT and N selected in this work are 3 times faster than in [1]. The reason for selecting the fastest trajectory shows that the proposed hybrid approach

is more practical than that proposed in [1]. To maintain  $PT_r$  at the desired level, a hybrid control scheme, ILC based on FOPID, has been proposed and applied to the system. At first, the non-linear boiler turbine system is linearized and a FOPID controller has been designed to accomplish the desired task, as discussed in the previous section. The FOPID controller designed for the system is applied as feedback to the boiler system, and the simulation results can be seen in Figure 7.

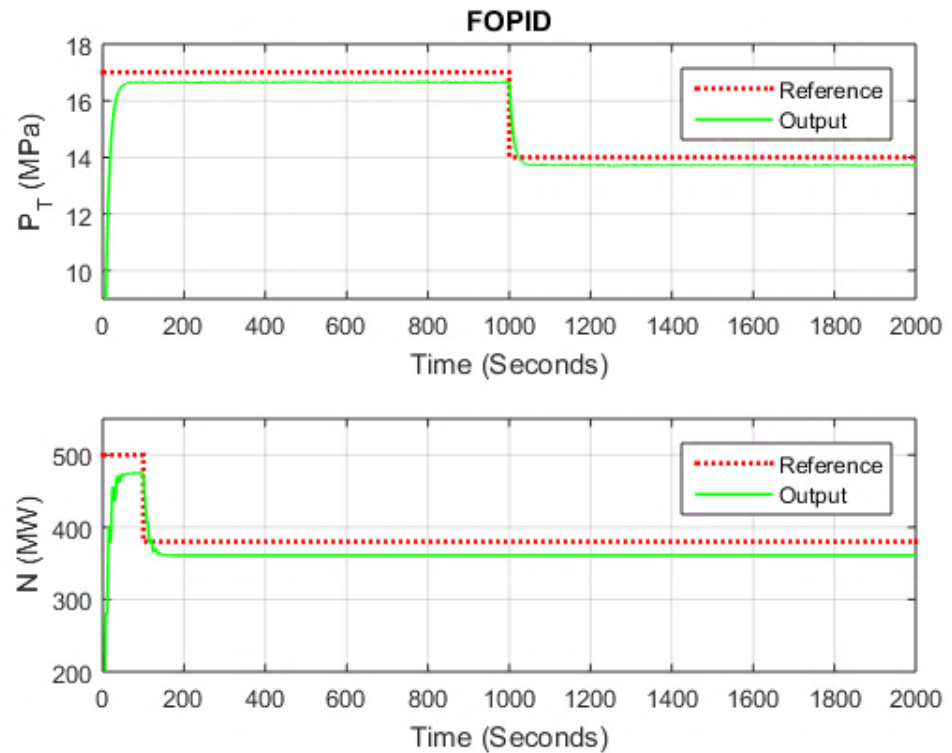


Figure 7. Closed-loop system with Fractional Order PID (FOPID) controller.

It can be seen from Figure 7 that the FOPID can ensure the desired trajectory with a substantial steady-state error in the tracking of  $PT_a$  and  $N$ . Also, some high-frequency noise in  $N$  requiring the controller to generate the high-frequency control effort is not desirable. As once connected to the hardware for real-time experimentation, the control effort's fast changes will cause heat dissipation in the micro controller and may eventually end up damaging it after some time. To avoid these problems, high steady-state error and high frequency noise in  $N$ , a hybrid control scheme with the control configuration in Figure 7  $P, PD$  and PID-type ILC gains in the hybrid structure with FOPID are designed using Zeigler-Nichols tuning rules [20].

Table 3 presents the formulation to design the PID controller using the Zeigler-Nichols method. Notice that the PID controller tuned by the first method of Ziegler-Nichols rules gives:

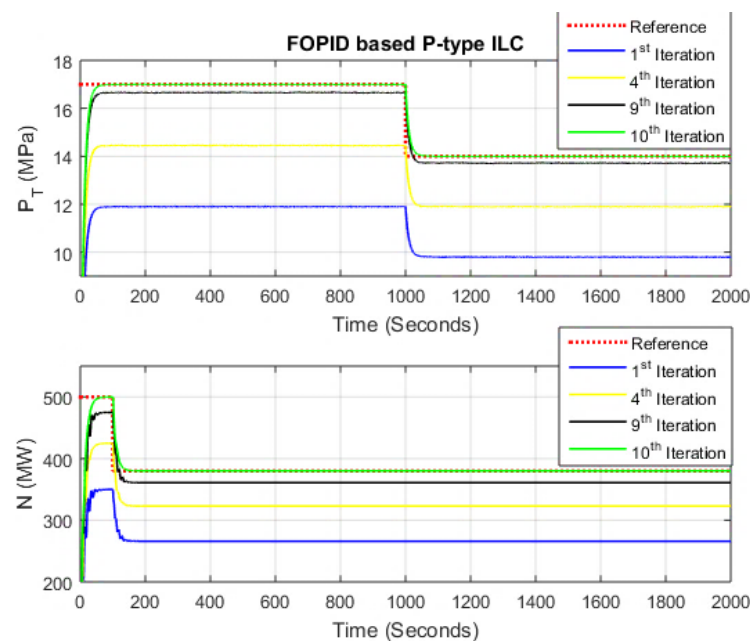
$$G_c(s) = 1.2 \frac{T}{L} \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (18)$$

In the proposed work,  $T, L, T_i$  and  $T_d$  were designed using controller performance parameters, that is, rise time, settle time, time delay and steady-state error.

**Table 3.** Ziegler-Nichols table for PID tuning [20].

Controller	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	inf	0
PI	$(0.9)\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$(1.2)\frac{T}{L}$	$2L$	$0.55L$

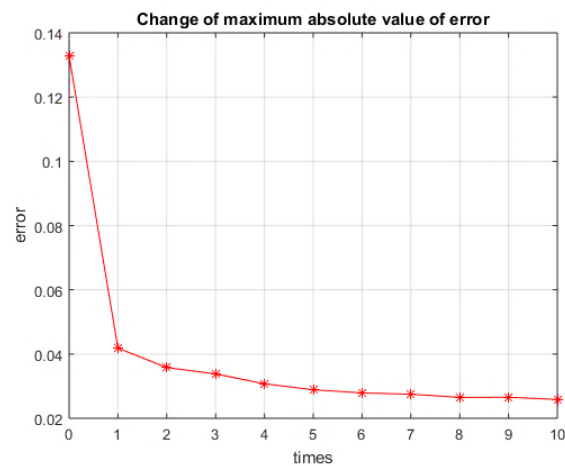
These values of  $T$ ,  $L$ ,  $T_i$  and  $T_d$  were then used to find  $K_p$ ,  $K_i$  and  $K_d$  (gains of PID-type ILC) as shown in the table above. Gains of the PID-type ILC controller used in the feedforward path with the FOPID-based closed-loop structure. The ILC controller designed for the closed plant system with the FOPID controller is first designed using a simple approach, i.e., P-type ILC. The simulation results for FOPID-based P-type ILC iteration-wise are shown in Figure 8.

**Figure 8.** Closed-loop response of the system with FOPID based P-type ILC.

It can be seen from Figure 8 that reference tracking is improved with ILC in feedforward of the FOPID based closed-loop system, i.e., the steady-state error in each subsequent iteration is smaller than that of the previous iteration. However, there is still an undesired high-frequency noise in  $N$ , which needs to be removed. The error versus iteration plot of the FOPID-based P-type ILC can be seen in Figure 9. It can be seen from Figure 9 that the error in steady-state error with FOPID-based P-type ILC and is reduced to 0.025, that is, 2.5% in just 10 iterations. The following equation calculates the maximum absolute value of error in Figure 9.

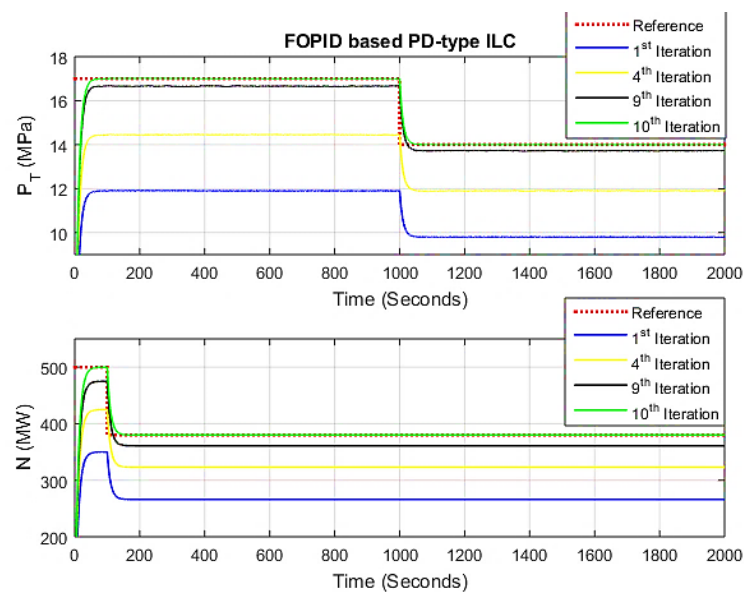
$$|e|_{max} = \max |PT_{ref} - PT|, |N_{ref} - N| \quad (19)$$

Since the output error is in vector form and there are two outputs, the tracking can be accurately represented by taking the maximum value of error at each iteration for both outputs and then taking the maximum of the two. However, the high-frequency fluctuation in  $N$  is not reduced; see Figure 8.



**Figure 9.** Change of closed-loop steady-state error in case of Hybrid-ILC each iteration.

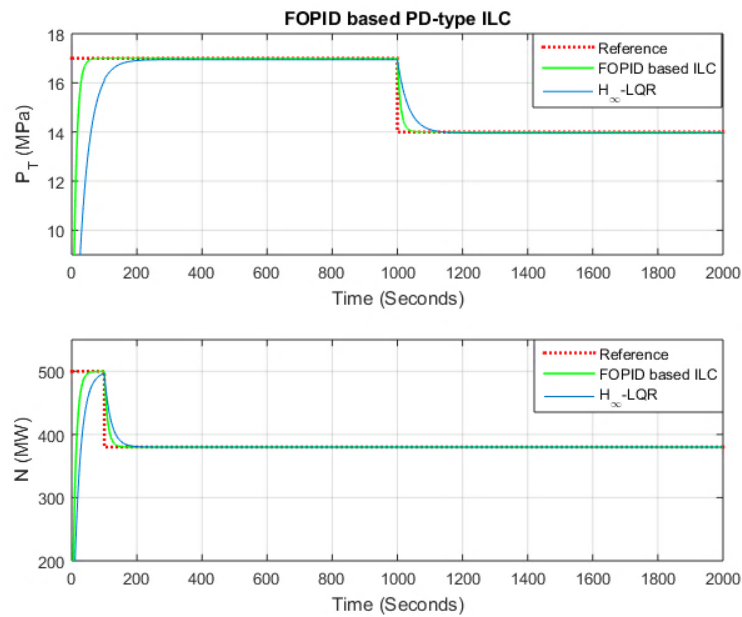
For removing this fluctuation, PD-type ILC has been designed. The simulation results of the closed-loop system with FOPID bas PD type ILC are shown in Figure 10. It can be seen from Figure 10 that not only the reference track has been consistently improving in each successive iteration, i.e., the steady-state error between reference and output in each successive iteration is smaller than that in the previous iteration, but the smoothness in  $N$  can also be observed. This shows that PD-type ILC can remove high-frequency noise, causing heat dissipation in the microcontroller to have been removed. This shows the effectiveness of the FOPID-based PD-type ILC.



**Figure 10.** Closed-loop response of the system with FOPID based PD-type ILC.

#### Comparison of Results

Comparison of the proposed method with the control design approach based on  $H_\infty$ -LQR presented in [1].  $H_\infty$ -LQR was implemented in MATLAB and the closed-loop results were compared with the hybrid ILC proposed in this work. The comparison can be seen in Figure 11. It is clear from the figure that the proposed approach ensures tracking the desired set point faster than the one presented in [1] while maintaining the desired steady-state performance.



**Figure 11.** Comparison of FOPID based PD-type ILC with  $H_{\infty}$ -LQR.

## 6. Conclusions

A novel hybrid control design approach for tracking throttle pressure and megawatt output has been proposed. The throttle pressure (PT) and megawatt output (N) were maintained at the desired levels for a specific time and then reduced. A hybrid control system, a FOPID-based ILC controller, is proposed for fast variations in PT and N. A simple FOPID controller ensures tracking with a quite sizeable steady-state error. The steady-state error is removed once a P-type ILC is designed and applied as a feedforward controller. A high-frequency fluctuation in N that requires the controller to generate the high-frequency control effort, on the other hand, is undesirable. Once connected to the hardware for real-time experimentation, the control effort's fast changes will cause heat dissipation in the microcontroller and eventually damage it after some time. To avoid this fluctuation, derivative action is applied to the P-type ILC to feed forward the closed-loop structure leading from the P-type ILC to the PD-type ILC. Simulation results show that the FOPID-based PD-type ILC has a 3.5% tracking error with no high-frequency output fluctuation. This shows the effectiveness of the proposed approach. It should also be noted that it took 9 min and 43 s on a dual-core PC with a 2 MHz processor and 2 GB of RAM to simulate the proposed work in MATLAB. But it is a one-time process for the specified system, i.e., once an updated control effort is generated, it will work all the time unless, of course, there is the same major change in the system parameters for which the controller has been designed.

**Author Contributions:** All authors contribute same to this article. All authors have read and agreed to the published version of the manuscript.

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

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Reference

1. Wei, L.; Fang, F.  $H_\infty$ -LQR-Based Coordinated Control for Large Units, Coal-fired Boiler-turbine Generation. *IEEE Trans. Ind. Electron.* **2016**, *64*, 5212. [[CrossRef](#)]
2. Hasan, F.A.; Rashad, L.J. Fractional-order PID controller for permanent magnet DC motor based on PSO algorithm. *Int. J. Power Electron. Drive Syst.* **2019**, *10*, 1724. [[CrossRef](#)]
3. Pirasteh-Moghadam, M.; Saryazdi, M.G.; Loghman, E.; Kamali E.A.; Bakhtiari-Nejad, F. Development of neural fractional order PID controller with emulator. *ISA Trans.* **2020**, *106*, 293–302. [[CrossRef](#)]
4. Mok, R.; Ahmad, M.A. Fast and optimal tuning of fractional order PID controller for AVR system based on memorizable-smoothed functional algorithm. *Eng. Sci. Technol. Int. J.* **2022**, *35*, 101264. [[CrossRef](#)]
5. Marjanovic, A.; Krstic, M.; Djurovic, Z.; Kovacevic, B. Control of thermal power plant combustion distribution using extremum seeking. *IEEE Trans. Control Syst. Technol.* **2016**, *25*, 1670–1682. [[CrossRef](#)]
6. Selvam, A.M.D.P.; Hussin, F.A.; Ibrahim, R.; Bingi, K.M.N. *Optimal Fractional-Order Predictive PI Controllers: For Process Control Applications with Additional Filtering*; Springer Nature: Cham, Switzerland, 2022.
7. Balko, P.; Rosinová, D. Robust Decentralized Control of Nonlinear Drum Boiler. *IFAC Pap. Online* **2015**, *48*, 432–437. [[CrossRef](#)]
8. Zhenlong, W.; Li, D.; Wang, L. Control of the superheated steam temperature: A comparison study between PID and fractional order PID controller. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016.
9. Ponce, I.U.; Bentsman, J.; Orlov, Y.; Aguilar, L.T. Generic Nonsmooth  $H_\infty$  Output Synthesis: Application to a Coal-Fired Boiler/Turbine Unit With Actuator Dead Zone. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 2117–2128. [[CrossRef](#)]
10. Takaghaj, S.M.; Macnab, C.J.B.; Westwick, D.; Boiko, I. Neural-adaptive control of waste-to-energy steam generators. *IEEE Trans. Control Syst. Technol.* **2013**, *22*, 1920–1926. [[CrossRef](#)]
11. El-Guindy, A.; Han, D.; Althoff, M. Formal analysis of drum-boiler units to maximize the load-following capabilities of power plants. *IEEE Trans. Power Syst.* **2016**, *31*, 4691–4702. [[CrossRef](#)]
12. Birk, W.; Dudarenko, N. Reconfiguration of the air control system of a bark boiler. *IEEE Trans. Control Syst. Technol.* **2015**, *24*, 565–577. [[CrossRef](#)]
13. Sakuragi, Y.; Sato, T.; Araki, N.; Konishi, Y. Self-tuning PI control for a boiler control system. In Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 10–13 September 2013; pp. 1–5.
14. Jing, L.; Jian-Hua, L.; Zhi, L. The intelligent boiler control system of self-learning and self-optimizing. In Proceedings of the International Conference on Automatic Control and Artificial Intelligence (ACAI 2012), Xiamen, China, 3–5 March 2012.
15. Arimoto, S.; Kawamura, S.; Miyazaki, F. Bettering operation of robots by learning. *J. Field Robot.* **1984**, *1*, 123–140. [[CrossRef](#)]
16. Cryer, B.; Nawrocki, P.; Lund, R. *A Road Simulation System for Heavy Duty Vehicles*; SAE Technical Paper; SAE: Warrendale, PA, USA, 1976; pp. 148–7191.
17. Uchiyama, M. Formation of high-speed motion pattern of a mechanical arm by trial. *Trans. Soc. Instrum. Control. Eng.* **1978**, *14*, 706–712. [[CrossRef](#)]
18. Xu, J.-X. A survey on iterative learning control for nonlinear systems. *Int. J. Control.* **2011**, *84*, 1275–1294. [[CrossRef](#)]
19. Wang, Y.; Gao, F.; Doyle, F.J., III. Survey on iterative learning control, repetitive control, and run-to-run control. *J. Process. Control.* **2009**, *19*, 1589–1600. [[CrossRef](#)]
20. Dong, S. Iterative learning control with incomplete information: A survey. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 885–901.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.