

## Article

# Design and Implementation of an Accelerated Error Convergence Criterion for Norm Optimal Iterative Learning Controller

Saleem Riaz <sup>1</sup>, Hui Lin <sup>1</sup> and Muhammad Pervez Akhter <sup>2,\*</sup>

<sup>1</sup> School of Automation, Northwestern Polytechnical University, Xi'an 710072, China; saleemriaznwpu@mail.nwpu.edu.cn (S.R.); linhui@nwpu.edu.cn (H.L.)

<sup>2</sup> School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072, China

\* Correspondence: pervezbcs@gmail.com

Received: 20 September 2020; Accepted: 19 October 2020; Published: 23 October 2020



**Abstract:** Designing an optimal iterative learning control is a huge challenge for linear and nonlinear dynamic systems. For such complex systems, standard Norm optimal iterative learning control (NOILC) is an important consideration. This paper presents a novel NOILC error convergence technique for a discrete-time method. The primary effort of the controller is to converge the error efficiently and quickly in an optimally successful way. A new iterative learning algorithm based on feedback based on reliability against input disruption was proposed in this paper. The illustration of the simulations authenticates the process suggested. The numerical example simulated on MATLAB@2019 and the mollified results affirm the validation of the designed algorithm.

**Keywords:** iterative learning control (ILC); NOILC; feedback control; PMSM; optimal control; error convergence; piezoelectric motor

## 1. Introduction

Many of the industrial process operating systems follow the same operation mode, where they once again completed a specific task within a particular time interval. For example, in industrial production processes, the continuous batch production works are usually involved. That is, the system completes a batch production under a given procedure within the required time interval and then repeats it again and again. It is evident for a system with continuous operations divided into batches. If the operation time of each batch has the same length for different quantities of action, then operation statistics and experience may be used to adjust the entire operating strategy for the next batch. This core and main idea lead us to “learning” that further tends to iterative learning control (ILC), which is an essential branch of intelligent control [1,2]. The human brain system inspires the learning algorithm. For example, a coach of a team commanded an ideal repetitive physical exercise task to the group. Then, the whole team tries to converge to that particular commanded exercise motion. Once the job has been performed after short period of time through sufficient trials or repetitions, the automatic program generated in the central nervous system (CNS), this further produces a sequence of input signals that stimulates the internal muscles and tendons according to the ideal motion pattern and approaches the required form of motion.

ILC is a universal control strategy framing the studies of the learning process of a human being, in which the main idea is to learn the time in which the consistent factors of the operation of the system based on different data full batches track the performance to become gradually better [3]. This control strategy is less dependent on system information and thus, generally, data-based control procedures, which effectively handle deals with traditional control challenges such as high-precision,

steady connection, modeling difficulties, and efficient tracking performance. The basic approach is for a robot that performs trajectory tracking tasks in a finite time interval, the error correction control information input of the current or previous trial used to make the repetitive task do better in the next operation. These repetitions continue until the trajectory of the output over the entire time interval tracks the desired path [4]. If there is an error that exists from trial to trial, this is a very challenging task to overcome. To cope with this problem, Arimoto, the inventor of iterative learning control, proposed that both the current and the previous trial data that further used to generate proper and precise control action [5,6].

The ILC is suitable for controlled objects with repetitive motion tasks, and iteratively corrects the improvement of a precise control target. Iterative learning control does not depend on the system's accurate mathematical description model. Within a finite time interval, a nonlinear high-coupling dynamic system control with high uncertainty can easily track the given expectation precisely [7]. This kind of trajectory tracking is used effectively in the field of motion control. ILC control adapted to systems that can be continuously repeated in operation during a finite time. Often used to track a specific target, the tracking target tends to remain the same. Is it very similar to repetitive control? The ILC control method is an intelligent control method, which can continuously learn evolution by increasing the number of iterations to adapt to the optimal control method. ILC does not need to care about the precise control model of the system, but it can achieve very high precision control performance with straightforward control.

## 2. ILC History and Background

Uchiyama in 1978 first proposed the idea of iterative learning control (ILC) but the paper was written in Japanese, so as a whole the impact was not so impressive. Furthermore, in 1984, Arimoto [1] introduced the method in English. This refers to a control method that continuously repeats a control attempt of the same trajectory and corrects the control law to obtain a very good control effect. It is worth noting that around 1984, five teams reported the fact that the repetition by the system can be utilized to improve performance. These teams follow:

- S. Arimoto, S. Kawamura and F. Miyazaki [1];
- G. Casalino and G. Bartolini [8];
- J. J. Craig [9];
- R. H. Middleton, G. C. Goodwin and R. W. Longman [10];
- E. G. Harakopoulos [11].

The characteristic of iterative learning control is “learning in repetition”, and through repeated iterative correction, the purpose of improving the control effect is achieved. The basic key principle of iterative learning is the error is taken from the previous output and correcting it for the next trial. The control input is continuously corrected during the control process. As the iteration progresses, the next control input is gradually improved to make the control effective. Iterative learning control requires only a small amount of previous information and is an effective control strategy for dealing with systems with unknown models and repeated motion properties. ILC control is adapted to systems that can be continuously repeated during operation. It is often used to track a specific target, and the tracking target tends to remain the same. Is it very similar to repetitive control? The ILC control method is an intelligent control method, which can continuously learn evolution by increasing the number of iterations to adapt to the optimal control method. ILC does not need to care about the precise control model of the system, but it can achieve very high precision control performance with very simple control.

The basic approach is for a robot that performs trajectory tracking tasks in a finite time interval, the error correction control information input of the previous or previous operations is used to make the repetitive task do better in the next operation. This is repeated continuously until the trajectory of the output over the entire time interval tracks the desired trajectory. The ILC is suitable for the

controlled objects with repetitive motion tasks, and iteratively corrects the improvement of a definite control target. Iterative learning control does not depend on the system's accurate mathematical description model. Within a finite time interval, a nonlinear high-coupling dynamic system control with high uncertainty can easily track the given expectation precisely. This kind of trajectory tracking is effectively used in the field of a system with motion control. ILC ensued several valuable results in both theory and applications in these three decades of evolution. It was concluded from these [12–17] survey papers that it is the key factor for the dynamics of the system such as the same tracking reference, the same operation length, and the same initial state, which can reduce the proposed update law as well as improve the tracking performance.

To date, for the ideal trajectory tracking, the present maximum ILC work requires that the length of the trail is permanent, and it is constant in each iteration [18–20]. However, in many ILC engineering applications, fixed requirements of the uniform trail length may not be satisfied. The span of the system output, states and control input will be due to complexity; each iteration will randomly change factors or random events. Recent studies about variable trail length are discussed and try to make the controller robust [21–27]. In a dynamic system with iteratively varying path lengths, other ILC algorithms based on iterative moving average operators can also be found [22,25,26]. Recently, in [28], ILC tracks the length of different trajectories of discrete-time systems to ensure that the  $p$ -type ILC scheme modifies the deterministic convergence method. Current ILC studies based on different trajectory lengths usually assume that the trajectory lengths of states, control inputs and outputs are the same in a particular iteration. As for the problem of changing the trajectory length, in most ILC algorithms, it is generally considered that the country length, control input and output are the same at particular iterations. In many practical applications, people expect that the controlled system can achieve the control goal without extra effort. For example, for a car or robotic speed control, when the speed control is to be close to the desired value of the speed, the system can operate freely without any extra control effort. It indicates that the control input in the specific time limit can be removed during the repetitive process. This means a repeating system with a randomly variable input trial length.

Accelerated error convergence is always a challenging task in a dynamical and maneuverable system where external disturbances are dominant such as the precise trajectory tracking of the Omni-wheel mobile robot for a highly maneuverable model [29]. The author experimentally proved and described the PID (proportional integral derivative) type control to compensate for the disturbances. Newton–Euler's method and its stability for the hexacopter nonlinear system proposed to eliminate the instabilities [30]. In addition, unconventional control methods are adopted to improve the tracking accuracy and performance of the soft robot manipulator. Open-loop control based on learning, relying entirely on mechanical feedback, demonstrated a pneumatic soft manipulator in [31]. The author of [32] achieved a substantial decrease in the tracking error of fabric-based soft arms by using model predictive control. The control method of [33] is an extended neural network-based nonlinear model predictive controller used to describe dynamics. In [34], a reinforcement learning method was applied together to optimize the stiffness and precise position tracking of the manipulator for auxiliary applications.

This paper is organized to propose novel error convergence topology through the aforementioned ILC methodology, and a typical norm optimal iterative learning error convergence control algorithm is presented. A novel optimal criterion is precise and robust for the input disturbance accommodation. The core advantage of the proposed algorithm is the fast error rate convergence that is a vital concern for every controller design. The new algorithm is very effective and robust for the input disturbances. Simulation results illustrate the optimal control algorithm validation.

### 3. Norm Optimal ILC

The main objective of the iterative learning control algorithm is to determine an ideal input  $u_{i+1}$  for the system under trial [35–38]. The plant output should track the desired trajectory output when this input is applied. This input should produce output  $y$  that follows the specific required trajectory  $y_d$  as precisely as possible. The term “as precisely as possible” states that the error should be as small

as possible. Furthermore, the difference between the measured output and the desired output should be as small as possible. The primary and vital role of the ILC algorithm is to minimize the error to ultimately approach zero. Thus, it shows that the ILC control law works in two dimensions, both the time index and iteration index. To overcome this issue, a cost function is proposed for the system under trial:

$$e_{i+1} = y_d - y_{i+1}, y_{i+1} = Gu_{i+1} \quad (1)$$

where the optimal criterion cost function is defined as follows:

$$J_{i+1}(u_{i+1}) = \|e_{i+1}\|_Q^2 + \|u_{i+1} - u_i\|_R^2 \quad (2)$$

where  $J_{i+1}(u_{i+1})$  and  $\|u_{i+1} - u_i\|$  are the cost function for the current trial input and the difference of norm for the present and preceding input trial, respectively;  $\|e_{i+1}\|$  shows the error norm of the present trial.  $Q$  and  $R$  are the  $l_2$  spaces of both  $m$  and  $q$  vectors, respectively, which are isometrically equivalent on  $[1, M-1]$  and  $[1, M]$ .

For simplicity, the most familiar sums and performance index of the norm are understood as follows:

$$J_{i+1} = \sum_{t=1}^M [y_d - y_{i+1}]^T Q [y_d - y_{i+1}] + \sum_{i=0}^{M-1} [(u_d - u_{i+1})^T R (u_d - u_{i+1})] \quad (3)$$

$Q$  and  $R$  are the weight matrices that must be positive symmetric definite in the aforementioned Equations (1) and (3) for the whole time  $t$ , assuming that  $Q$  and  $R$  are the diagonal matrices and for the positive real numbers  $mM$  and  $qM$ .

Further computation of the cost function is described as follows:

$$J_{i+1}(u_{i+1}) = \sum_{t=1}^N [y_d(t) - y_{i+1}(t)]^T Q(t) [y_d(t) - y_{i+1}(t)] + \sum_{t=0}^{N-1} [U_{i+1}^{ilc}(t) - U_i^{ilc}(t)]^T R(t) [U_{i+1}^{ilc}(t) - U_i^{ilc}(t)] \quad (4)$$

In the formula,  $Q(t)$  and  $R(t)$  are, respectively, the error weight and control law variation weight matrix at time  $t$ , and at any time  $t$ ,  $Q(t) > 0$  and  $R(t) > 0$ . Constructing an  $n$ -dimensional real symmetric positive definite matrix  $Q, R$ , whose values on the diagonal are respectively  $Q(t)$  and  $R(t)$ :

$$Q = \text{diag}\{Q(1), Q(2), \dots, Q(N)\} \\ R = \text{diag}\{R(1), R(2), \dots, R(N)\} \quad (5)$$

The purpose of this performance index is to find an optimal  $U_{i+1}^{ilc}$ , which can minimize the tracking error and ensure that the deviation from  $U_{i+1}^{ilc}$  is not too large. How to balance these two goals can be decided by designing  $Q$  and  $R$ .  $U_{i+1}^{ilc}$  can be obtained by taking the partial derivative of the performance index;

$$\frac{1}{2} \frac{\partial J_{i+1}}{\partial U_{i+1}^{ilc}} = -G^T Q e_{i+1} + R(U_{i+1}^{ilc} - U_i^{ilc}) = 0 \quad (6)$$

Since  $R$  is a positive definite real symmetric matrix, it is invertible, and the solution Formula (6) provides the optimal control law update equation:

$$U_{i+1}^{ilc} = U_i^{ilc} + R^{-1} G^T Q e_{i+1} \quad (7)$$

$G^* = R^{-1} G^T Q$  is defined by the following theorem.

**Theorem 1.** If there is  $GG^* \dots \sigma^2 I$ , then  $e_i \rightarrow 0$  when the number of iterations  $i$  goes to infinity, that is, the designed controller can realize the full tracking of the expected trajectory.

It is proven that according to the tracking error  $e_{i+1} = y_d(t) - GU_{i+1}^{ilc}$  during the  $i + 1$  iteration, the updating equation of tracking error on the iteration axis can be calculated as

$$e_{i+1} = (I + GG^*)^{-1}e_i \quad (8)$$

Since there is  $GG^* \dots \sigma^2 I$ , it is obvious that  $e_{i+1} \dots \frac{1}{1+\sigma^2} e_i$  and the error will converge to zero. At the same time, the control law designed according to this optimal index can change the value of 2 by adjusting the gain matrix  $Q$  and  $R$ , and then adjusting the convergence rate of the tracking error.

A literature study shows that there is a lot of cost function computation and solving ILC problems. Some arguments prove the appropriation of the cost function (Equation (2)) and its effectiveness:

- (i) The choice to  $\|u_{i+1} - u_i\|^2$  demonstrates that the deviation in the input from trial to trial is minimal. This means that the convergence is automatically smooth which further leads to produce the stable control input signal to manipulate the actuator behavior.
- (ii) The choice of the factor  $\|e_{i+1}\|^2$  shows the reduction in the tracking error from trial to trial.
- (iii) The cost function is dependent on both aforementioned terms, and uses  $Q$  and  $R$  weight matrices. The ratio  $\beta = \frac{m}{q}$  has to be significant to keep the deviation very small. This ration is chosen as little as possible to make the error the lowest. The terms small and large depend on system constraints and the system to be used.
- (iv) Choosing  $u_{k+1} - u_k = 0$  makes the optimal value of the cost function bound and becomes  $J(u_i) = \|u_i - u_i\|_R^2 + \|e_i\|_Q^2 = \|e_i\|_Q^2$ , which further results in the optimal amount  $J(u_{i+1}) - \|e_i\|_Q^2 \leq 0$ . Hence, the optimal value has a lower bound:  $J(u_{i+1}) = \|u_{i+1} - u_i\|_R^2 + \|e_{i+1}\|_Q^2 \geq \|e_i\|_Q^2$

Combining the above two equations resulted as

$$\|e_{i+1}\|^2 \leq J_{i+1}(u_{i+1}) \leq \|e_i\|^2 \quad (9)$$

The input update law can be obtained easily by differentiating the cost function concerning  $u_{i+1}$ :

$$\frac{\partial J}{\partial u} = 0 \Rightarrow u_{i+1} - u_i = R^{-1}G^T Q e_{i+1} = u_i + G^* e_{i+1} \quad (10)$$

$$u_{i+1} = u_i + R^{-1}G^T Q e_{i+1} \quad (11)$$

where  $u_{i+1}$  and  $u_i$  represents the current and previous trial input, respectively.  $e_{i+1}$  shows the error of the ongoing trial and  $G^* = R^{-1}G^T Q$  is the required weight of the cost function.

#### 4. ILC Algorithm Design and Practical Application

The design process for the aforementioned problem norm optimal iterative learning controller comprises as following steps:

Step 1:

ILC casual implementation is possible, usually when there is a full state knowledge of the system. The ideal control is changed by writing for the costate:

$$\Psi_{i+1}(t) = [-\hat{K}(t)\{I + BR^{-1}(t)B^T\hat{K}(t)\}^{-1} \cdot \{x_{i+1}(t) - x_i(t)\}] + \zeta_{i+1}(t) \quad (12)$$

Typically, a standard strategy as taken in [28,39], adopted to calculate the feedback gain  $\hat{K}(t)$ , is taken from the renowned equation of the discrete Riccati solution that can be defined as follows [40].

Step 2:

$$\begin{aligned} \hat{K}(t) = & A^T \hat{K}(t+1)A + \bar{C}^T Q(t+1)\bar{C} - \{A^T \hat{K}(t+1)B \dots \\ & \dots \times (B^T \hat{K}(t+1)B + R(t+1))^{-1} \cdot B^T \hat{K}(t+1)A\} \quad t \in [0, N-1] \end{aligned} \quad (13)$$

For this equation, the terminal condition is  $K(N) = 0$ . This is also independent of the system states, inputs, and outputs. Hence, the feedback gain can be solved offline before the trial sequence is initiated.

Step 3:

The feedforward action is called the predictive term  $\zeta_{k+1}(t)$  which is produced by

$$\zeta_{i+1}(t) = [I + \hat{K}(t)BR^{-1}(t)B^T]^{-1} \cdot \{A^T \zeta_{i+1}(t+1) + \bar{C}^T Q(t+1)e_i(t+1)\} \quad (14)$$

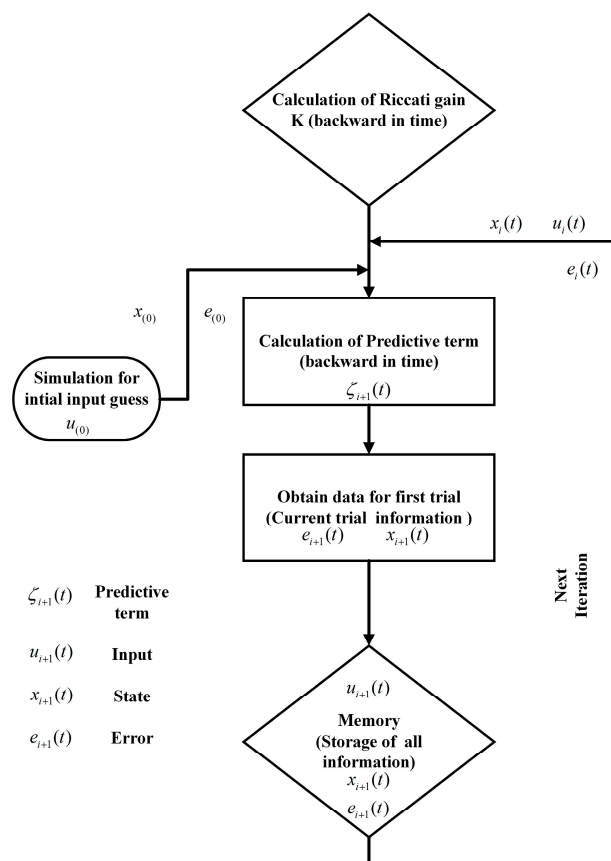
With the terminal condition  $\zeta_{i+1}(N) = 0$ . The predictive term consequently determined by the following error on the previous (i.e.,  $i$ th trial).

Step 4:

The input update law is for the aforementioned steps described as

$$u_{i+1}(t) = u_i(t) - \{B^T \hat{K}(t)B + R(t)\}^{-1} B^T \hat{K}(t) \cdot A \{\ddot{x}_{i+1}(t) - \dot{x}_i(t)\} + [R^{-1}(t) \cdot B^T \zeta_{i+1}(t)] \quad (15)$$

Normally, a typical iterative learning control algorithm comprises the  $(I + 1)$ th trial full state-input joint with a feedforward term from the last trial error data. This representation of the ILC algorithm is non-casual since Equations (8) and (9) can be solved offline. For each trial, it is a switch time simulation that utilizes available previous trial information. The block diagram of the proposed controller design algorithm is shown in Figure 1.



**Figure 1.** Algorithm of proposed norm optimal iterative learning controller.

As it can be observed, the original prediction error values are fed into the workflow at the same time as the first stage information gain matrix by the  $K(t)$  calculation. The workflow is also being passed on. The second step uses these two input sources and initializes all the information which was

saved in the workspace till now. During the process of the second step, the values for the predictive term array are provided to the third section, in sequence. Finally, all workspace values are referenced in the third stage and operated on a system offline to control the plant.

The availability of the feedback full state knowledge assures the implementation of norm optimal iterative learning control. Both of the parameters  $Q$  and  $R$  must be selected intuitively, keeping in mind the error norm. The convergence property depends on the choice of the cost function  $J_{i+1}$  which signifies the performance of both the error sequence and input rate change. To demonstrate the effect of  $Q$  and  $R$ , the following assumptions are made:

- i- Keeping  $Q$  fixed;
- ii-  $R = (\rho.R_0)$  where  $R_0 > 0, (0, T)$ ;
- iii- The variable parameter  $\rho$  ( $\rho > 0$ ).

This shows that the variable  $\rho$  has full control of the proposed algorithm. The error converges subsequently by changing the parameter  $\rho$ .

## 5. Practical Application of NOILC Scheme for Piezo Motor Position Tracking $\epsilon$

Different mathematical models have been presented for practical applications of piezoelectric actuators [41–43]. However, the researcher modified and proposed a novel Preisach model to simulate the hysteresis effect in a piezoelectric stack actuator and eventually experimentally proved the feasibility of the model.

For the standard norm optimal iterative learning control (NOILC) criteria for the faster rate of error convergence and to prove the efficacy of the proposed iterative learning algorithm to control the existence of unknown dead-zone nonlinearity, simulations have been conducted with a linear model of a piezoelectric, as shown in Figure 2. There are many auspicious applications of motors in the industry. Piezoelectric motors have the characteristics of low speed and high torque, which are insignificant differences with the features of typical electromagnetic motors such as high speed and low torque. In addition, features like a compact structure, light weight, and the quiet operation of the piezoelectric motor makes them resilient to external magnetic as well as radioactive fields. Piezoelectric motors are widely used for high-precision control applications because they can easily reach the accuracy of the level of micrometers or even the level of nanometers. Such a kind of control brings additional difficulties to the establishment of a precise mathematical model of the piezoelectric motor: slighter nonlinearity and unknown factors significantly disturb the control and features' characteristics of the motors. Specifically, compared with the allowable level of the input signal, the piezoelectric motor has a large dead zone, and the size of the dead zone changes with position displacement, the control scheme for position tracking, where  $\theta_d$  indicates the desired value and  $\theta$  shows the measured output. This also denotes that the feedforward term is  $u_{ff}$  and the feedback term  $u_{fb}$  while the error is  $e$ . ILC (Iterative Learning Control) means the control law applied from first to last iteration as shown in Figure 2.

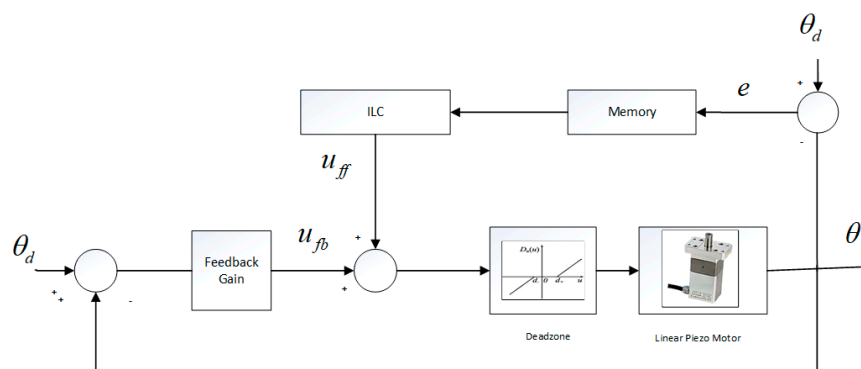


Figure 2. Block diagram of control scheme for piezo-motor position control.



$$\begin{aligned}
 \dot{x}_1(t) &= x_2(t) \\
 \dot{x}_2(t) &= -\frac{k_{fv}}{M}x_2(t) + \frac{k_f}{M}u(t) \\
 y(t) &= x_1(t)
 \end{aligned} \tag{16}$$

where  $x_1, x_2$  are the position and the velocity of the motion, respectively. The damping velocity factor is  $k_{fv} = 144(N)$  constant of the force  $k_f = 6(N/V)$  and moving mass  $M = 1(kg)$ :

$$\begin{aligned}
 x_1^d(i+1) &= x_1^d(i) + 3 \times 10^{-3} \cdot x_2^d(i) + 6.662 \times 10^{-6} \cdot u^d \\
 x_2^d(i+1) &= 65.21 \times 10^{-2} \cdot x_2^d(i) + 3 \times 10^{-2} \cdot u^d \\
 y(i+1) &= x_1(i+1)
 \end{aligned} \tag{17}$$

where the full state knowledge is assumed and shows the measured values of angular position and velocity. The simulation results are shown in Figure 3, where we have taken  $\rho = 0.001$ . The reference trajectory is taken as  $y_d(i) = 20\{1 + 3 \sin(0.35iT)\}mm$ ,  $i \in (0, 1, 2, \dots, 100)$ , where the position displacement and velocity were tracked. The reference preserves the system far beyond the linearization point and guarantees that the nonlinearity disturbs the system dynamics. Since the algorithm contains a feedback term, no additional pre-compensation is needed. It is worth noting that despite using a linear model, even though using improper physical constants in this model, the algorithm attains a rapid rate of error convergence, as shown in Figure 4. It is evident that the proposed algorithm is robust. In almost all cases,  $e_i$ , the  $L_2$  norm ore, can be controlled by the parameter  $\rho$ . Only plants with non-minimum phases show a slower convergence rate.

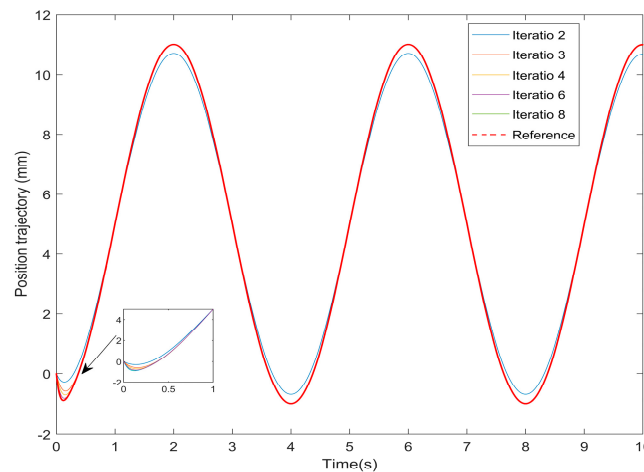


Figure 3. Desired reference trajectory tracking.

As mentioned above, the hysteresis effect makes the system complex in order to track the precise position. The proposed controller has satisfactory results compared to the typical PID [44], inverse system control [45] method and robust control [46] as well as sliding mode control [47]. A further experimental approach for the proposed controller can be adopted using any of the aforementioned models to verify this robust NOILC technique.

In general, a new iterative learning control algorithm was discovered to be fruitful to achieve a rapid rate of error convergence.



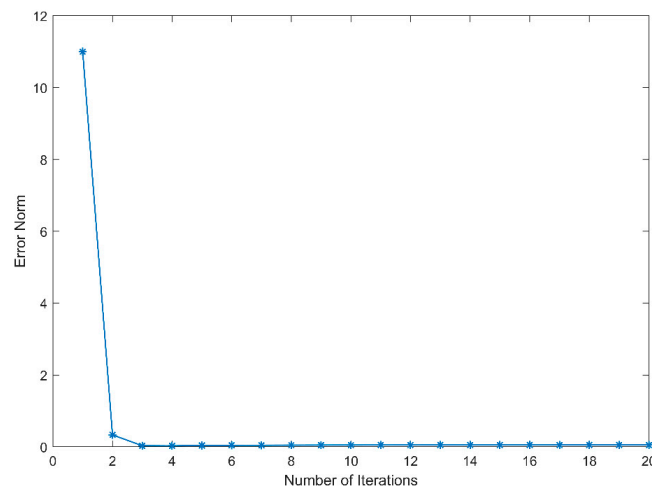


Figure 4. Error convergence rate.

#### Higher Order Linear Discrete Time Plant Simulation Example

In this section, a numerical example is used for the validation of the proposed control algorithm, considering a linearized discrete-time system as follows:

$$x_i(t+1) = \begin{bmatrix} 0.60 & 0.30 & -0.07 \\ 0.04 & -0.50 & 0.03 \\ 0.90 & 0.90 & 0.80 \end{bmatrix} x_i(t) + \begin{bmatrix} 0 \\ 0 \\ 0.2 \end{bmatrix} \quad (18)$$

$$y_i(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x_i(t) \quad (19)$$

The desired trajectory is  $y_d = 0.4(t) \sin(2\pi t/N)$  for the interval  $t = [0, T]$  where  $t \in (0, 1, 2, 3, \dots, 10)$  and  $N = 10$ . Initially, the input of the system is  $u_0 = 0.6$ . The parameters  $Q$  and  $R$  are both set to be 1; only the variable  $\rho$  changed. It is supposed that the full state knowledge of the system is taken into consideration. Input for the different trials is shown in Figure 5.

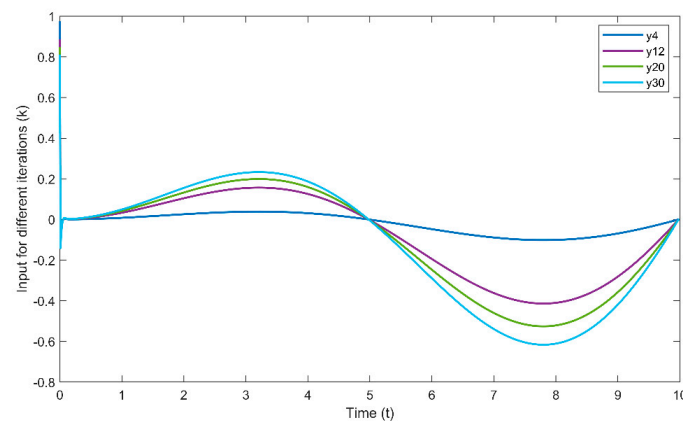


Figure 5. Input for different trials.

## 6. Simulation Results and Discussions

It can merely be observed that the performance of convergence is better for the given system. The output  $y$  of the plant after 20 or more iterations follows the desired trajectory  $y_d$ . The output for different trials is shown in Figure 6. The parameter  $\rho$  set to be 0.8384.

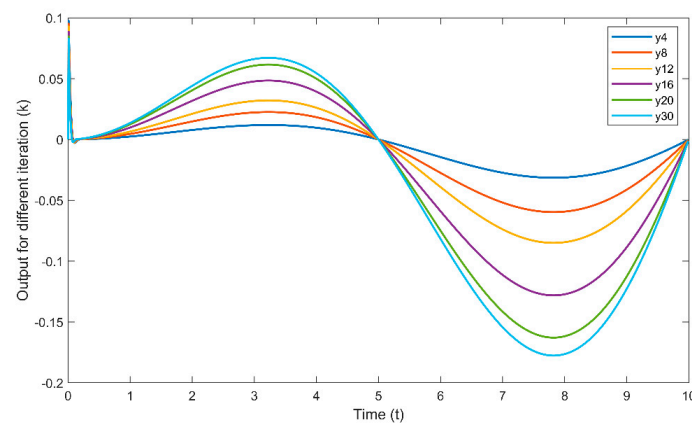


Figure 6. Output at different trials.

The error convergence results for both values  $\rho$  is shown in Figure 7. The convergence rate is faster for the smaller  $\rho$  as compared to a bigger value. This shows that the design of the controller is robust and considers the little value as much as possible for better and precise convergence. It can observe that as the number of trials increases, the rate of convergence increases accordingly. The prediction change of the reference further modifies the control action in an optimal method. Eventually, the output follows the desired trajectory after the 10th iteration more precisely because of the smaller value of  $\rho$ , as shown in Figure 8. This shows that the solution is more robust against different input trials. Because the control action is highly convergent and the error converges at a faster rate, the reference trajectory follows the output too quickly and the time settling is very short, as shown in Figure 7.

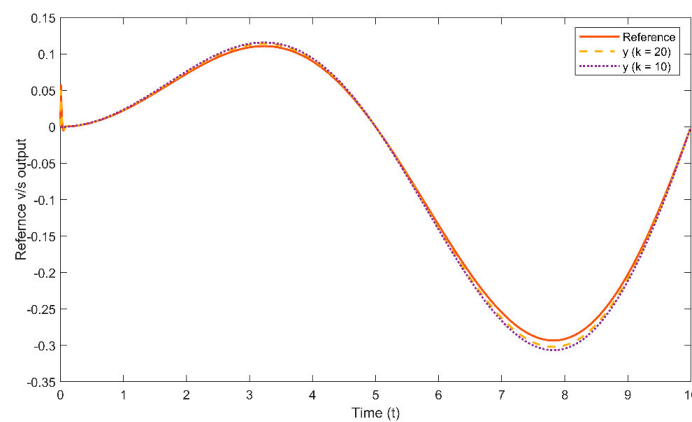


Figure 7. Desired reference trajectory tracking.

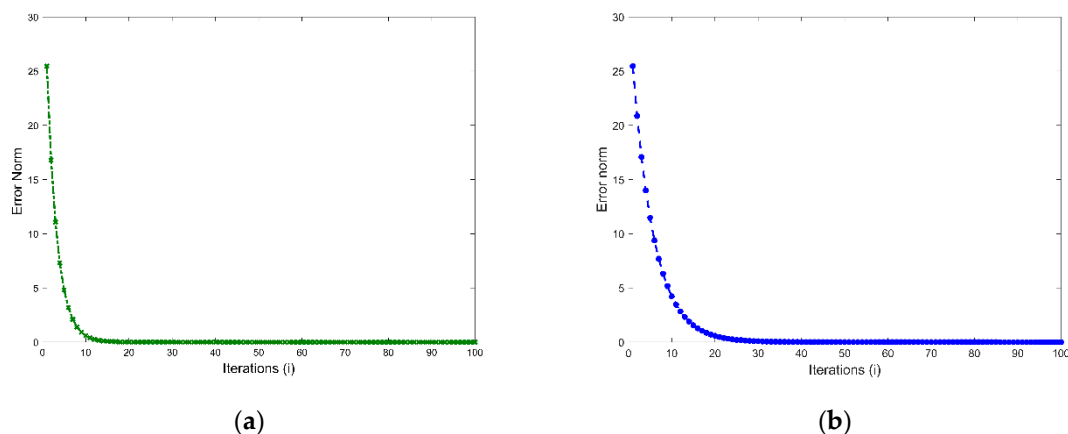


Figure 8. Error norm for  $\rho = 0.8384$  in (a) and error norm for  $\rho = 10$  in (b).

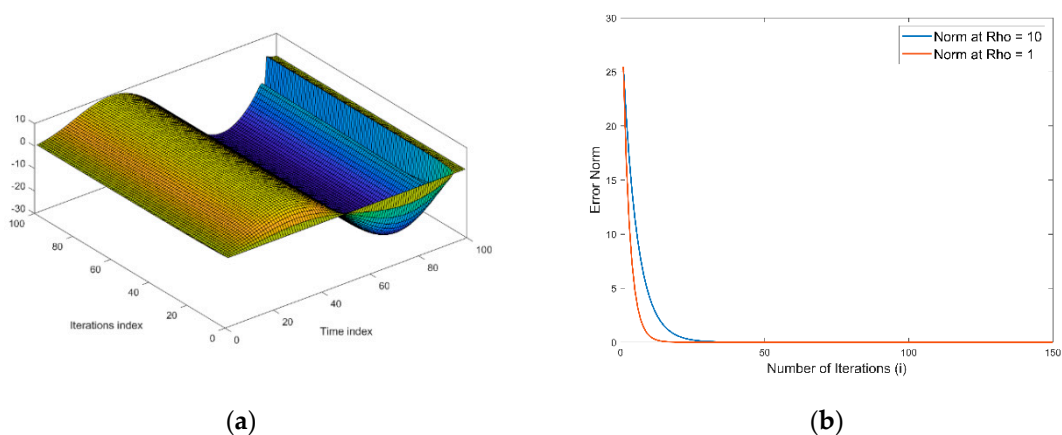
The rate of error convergence for the proposed norm optimal ILC for the whole trials shown in the two following Figure 8a,b. This shows that the convergence is guaranteed even for the higher order system. The error convergence rate and the error norm shown in Figure 8a indicate that a smaller value  $\rho$  tends to converge an error faster, and if it is large, the error converges at a slower rate as shown in Figure 8b.

Alternatively, the limitations of the algorithm can be considered as follows. Although this algorithm can effectively deal with the noise output because it is downstream of the control action, it still needs to be modified as fast as possible in the future. The robustness of the algorithm for nonlinear dynamical systems has not been proven theoretically. The algorithm can only be implemented when the plant model is known in advance and the algorithm is only valid for reversible systems.

### 6.1. Controller Robustness against Input Disturbance

Taking the same plant model and disturbance added in the system, the existence of white noise signifies the disturbance behavior. In the Simulink model used for the current simulation, the location of the noise entry point is given. The influence of disturbance on the error evolution is shown in Figure 9a,b:

$$x_i(t+1) = \begin{bmatrix} 0.60 & 0.30 & -0.07 \\ 0.04 & -0.50 & 0.03 \\ 0.90 & 0.90 & 0.80 \end{bmatrix} x_i(t) + [0]u_i(t) + d \quad (20)$$



**Figure 9.** Input for the whole interval in (a) and the error convergence for the whole iteration (b).

The result shows that under the influence of input white noise, the error still decreases exponentially. A convergence of errors is fast for the convergence rate of the first trial, and so on. However, even the first ten trials are enough to make the tracking error constraint reach the ideal value limit. Through the observation of the error surface graph, this assumption has been strengthened. After 10 trials, it is evident that the error range is small in size, as shown in Figure 9b. Therefore, the algorithm successfully controlled the controlled object despite the presence of noise in the input.

### 6.2. Controller Validation for Dynamical Nonlinear System

For the effectiveness of the proposed controller design, we suppose the nonlinear Permanent Magnet Synchronous Motor (PMSM) dynamical model that can be described as follows:

$$\begin{aligned} x_1 &= x_2 \\ x_2 &= \frac{1}{M}(F_m - \varepsilon x_2 - f_{cog} - f_{rel} - f_{fric} + f_{dis}) \\ y &= x_1 \end{aligned} \quad (21)$$

where  $M$  is the weight of the load,  $\varepsilon$  is the damping coefficient,  $x_1, x_2$  are the moving position and velocity, respectively,  $f_{dis}$  is the disturbance acting on the load,  $f_{fric}$  is the friction,  $f_{cog}$  denotes the thrust ripple of the alveolar, and  $f_{rel}$  is the thrust ripples reluctance. The electromagnetic force  $F_m$  is responsible for eliminating these thrust ripples.

The nonlinear permanent magnet synchronous motor model in Figure 10 is decoupled and transformed into a new linear system through feedback linearization structure. This satisfies the linearization conditions and the exact sampled time linearized model [7] described as follows:

$$x_i^d = Ax_i^d + Bu_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_i^d + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i \quad (22)$$

$$y^d = Cx_i^d = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i \quad (23)$$

where  $x_i^d = \begin{bmatrix} x_1^d \\ x_2^d \end{bmatrix}$ ,  $u_i = (1/M)(K_F i_{gi} - \xi x_2 - \xi x_i)$  and the sampling time  $t = 0.01$  s, the desired trajectory is  $y_d = 5 \sin(\pi t)$  for the interval  $[0, 1.5]$ . As shown in Figure 11a, the output of the nonlinear dynamical system tracks perfectly the desired trajectory in a smooth way. Eventually, the error converges to zero. The value of the control parameter can be changed to achieve the target sequence and it can be seen that the algorithm is highly effective and valid for the dynamical systems as well.

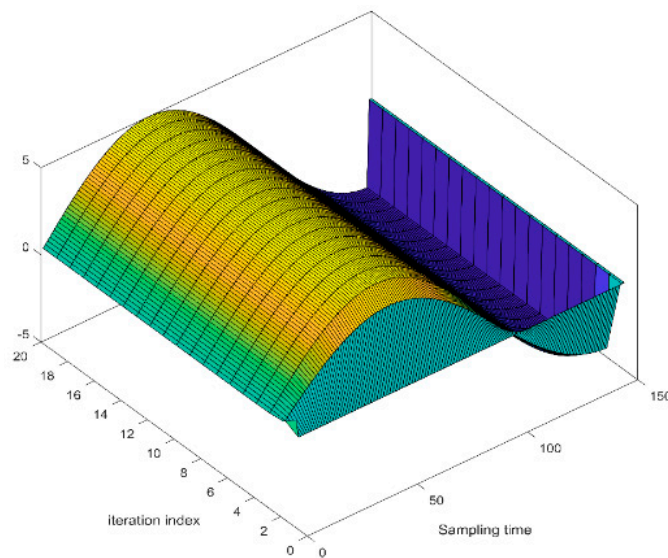


Figure 10. Input of the system for complete trials.

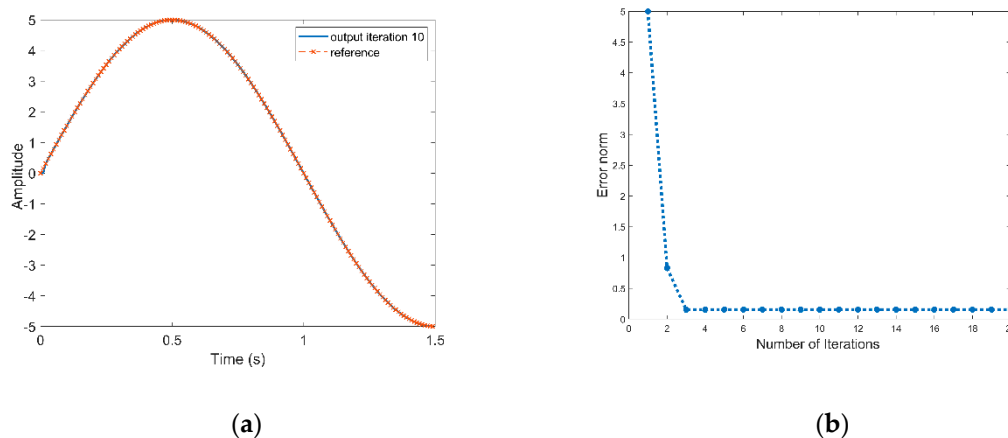


Figure 11. Output and desired trajectory for the final iteration in (a) and the error convergence in (b).

The rate of error convergence is shown above in Figure 11b. It can be observed that the error convergence for the first five trials is very fast but slowly goes to zero from 5 to 10 trials. The convergence also depends on the system as well. If the system is highly dynamical and has noise, the error convergence is also slower but ultimately goes to zero. Finally, the proposed algorithm is effective and the error exponentially converges quickly. Thus, it can be implemented in many practical applications such as aircraft maneuvering, altitude control, XY plotter system, robot manipulations, satellite control, position control, power electronic control and in many automation industrial applications.

## 7. Conclusions

An optimization method-based error convergence criterion in iterative learning control was discussed and the controller design methodology proposed in this paper. The fallouts showed the usual implementation of an iterative learning control algorithm for the linear discrete-time plant. This algorithm also works properly for a nonlinear discrete time system as well. The feedback and feedforward action (current and previous trial) arrangement is the leading topology in this proposed algorithm. The main advantage of this research shows the ability of this optimal criterion against disturbances' rejection and the convergence of error. The proposed method is robust compared to typical basic linear quadratic regulators (LQR). The rate of error convergence is the core attention in every control algorithm. The proposed method is valid for linear and nonlinear control systems and able to converge error at faster rate ultimately, which tends to reach its convergence limit. Robustness theory and practical application for this typical optimal control algorithm for the nonlinear plant have future concern such as:

- Dynamical and maneuverable system like omni-wheal mobile robots;
- Applications of soft robotics position control and disturbance rejection;
- Maneuvering and disturbance rejections of quad-core and hexa-copter applications;
- Piezo-ceramic actuators and nano positioning systems.

The XY plotter system, robot manipulations, Satellite position control as well as human robot collaboration are the future extensions of this proposed research work.

**Author Contributions:** Conceptualization, H.L. and S.R.; methodology, S.R.; validation, M.P.A., and H.L.; formal analysis, M.P.A.; investigation, S.R.; resources, H.L.; writing—original draft preparation, S.R.; writing—review and editing, M.P.A.; visualization, S.R.; supervision, H.L. All authors have read and agreed to the published version of the manuscript.

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

**Acknowledgments:** Authors would like to thank all those people who helped to complete this study.

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

## References

1. Arimoto, S.; Kawamura, S.; Miyazaki, F. Bettering operation of Robots by learning. *J. Robot. Syst.* **1984**, *1*, 123–140. [\[CrossRef\]](#)
2. Nandi, C.; Debnath, R.; Debroy, P. Intelligent Control Systems for Carbon Monoxide Detection in IoT Environments. In *Guide to Ambient Intelligence in the IoT Environment. Computer Communications and Networks*; Mahmood, Z., Ed.; Springer: Cham, Switzerland, 2019; pp. 153–176. [\[CrossRef\]](#)
3. Lu, J.; Cao, Z.; Zhang, R.; Gao, F. Nonlinear Monotonically Convergent Iterative Learning Control for Batch Processes. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5826–5836. [\[CrossRef\]](#)
4. Zhao, X.; Wang, Y. Energy-optimal time allocation in point-to-point ilc with specified output tracking. *IEEE Access* **2019**, *7*, 122595–122604. [\[CrossRef\]](#)
5. Chen, Y.; Wen, C. High-Order Iterative Learning Control: Convergence, Robustness and Applications. In *Iterative Learning Control: Convergence, Robustness and Applications*; Springer: London, UK, 1999; Volume 248.
6. Bien, Z.; Xu, J.-X. *Iterative Learning Control Analysis, Design, Integration and Applications*; Springer Science & Business Media: Boston, MA, USA, 1998; pp. 313–334. [\[CrossRef\]](#)

7. Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors* **2019**, *19*, 24. [[CrossRef](#)] [[PubMed](#)]
8. Casalino, G.; Bartolini, G. A learning procedure for the control of movements of robotic manipulators. *Iasted Symp. Robot. Autom.* **1984**, 108–111.
9. Craig, J.J. Adaptive Control of Manipulators Through Repeated Trials. *Proc. Am. Control Conf.* **1984**, *3*, 1566–1573.
10. Middleton, R.H.; Goodwin, G.C.; Longman, R.W. A Method for Improving the Dynamic Accuracy of a Robot Performing a Repetitive Task. *Int. J. Rob. Res.* **1989**, *8*, 67–74. [[CrossRef](#)]
11. Harokopos, E.G. *Learning and Optimal Control of Industrial Robots in Repetitive Motions*; Society of Manufacturing Engineers: Dearborn, MI, USA, 1986.
12. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114. [[CrossRef](#)]
13. Ahn, H.S.; Chen, Y.Q.; Moore, K.L. Iterative learning control: Brief survey and categorization. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1099–1121. [[CrossRef](#)]
14. Wang, Y.; Gao, F.; Doyle, F.J. Survey on iterative learning control, repetitive control, and run-to-run control. *J. Process Control* **2009**, *19*, 1589–1600. [[CrossRef](#)]
15. Shen, D.; Wang, Y. Survey on stochastic iterative learning control. *J. Process Control* **2014**, *24*, 64–77. [[CrossRef](#)]
16. Moore, K.L.; Jian-Xin, X.U. Special issue on iterative learning control. *Int. J. Control* **2000**, *73*, 819–823. [[CrossRef](#)]
17. Freeman, C.; Tan, Y. Iterative learning control and repetitive control. *Int. J. Control* **2011**, *84*, 1193–1195. [[CrossRef](#)]
18. Xu, Q.Y.; Li, X.D. HONN-Based Adaptive ILC for Pure-Feedback Nonaffine Discrete-Time Systems with Unknown Control Directions. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 212–224. [[CrossRef](#)] [[PubMed](#)]
19. Si, T.; Long, L. A small-gain approach for adaptive output-feedback NN control of switched pure-feedback nonlinear systems. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 784–801. [[CrossRef](#)]
20. Meng, D.; Moore, K.L. Robust iterative learning control for nonrepetitive uncertain systems. *IEEE Trans. Autom. Contr.* **2017**, *62*, 907–913. [[CrossRef](#)]
21. Seel, T.; Schauer, T.; Raisch, J. Monotonic convergence of iterative learning control systems with variable pass length. *Int. J. Control* **2017**, *90*, 409–422. [[CrossRef](#)]
22. Li, X.; Shen, D. Two novel iterative learning control schemes for systems with randomly varying trial lengths. *Syst. Control Lett.* **2017**, *107*, 9–16. [[CrossRef](#)]
23. Wei, Y.S.; Li, X.D. Varying trail lengths-based iterative learning control for linear discrete-time systems with vector relative degree. *Int. J. Syst. Sci.* **2017**, *48*, 2146–2156. [[CrossRef](#)]
24. Shen, D.; Xu, J.-X. Adaptive Learning Control for Nonlinear Systems with Randomly Varying Iteration Lengths. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 1119–1132. [[CrossRef](#)]
25. Liu, S.; Wang, J.R. Fractional order iterative learning control with randomly varying trial lengths. *J. Frankl. Inst.* **2017**, *354*, 967–992. [[CrossRef](#)]
26. Liu, S.; Debbouche, A.; Wang, J.R. On the iterative learning control for stochastic impulsive differential equations with randomly varying trial lengths. *J. Comput. Appl. Math.* **2017**, *312*, 47–57. [[CrossRef](#)]
27. Meng, D.; Zhang, J. Deterministic Convergence for Learning Control Systems Over Iteration-Dependent Tracking Intervals. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3885–3892. [[CrossRef](#)]
28. Molinari, B.P. The time-invariant linear-quadratic optimal control problem. *Automatica* **1977**, *13*, 347–357. [[CrossRef](#)]
29. Božek, P.; Al Akkad, M.A.; Blištan, P.; Ibrahim, N.I. Navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator. *Int. J. Adv. Robot. Syst.* **2017**, *14*. [[CrossRef](#)]
30. Kilin, A.; Bozek, P.; Karavaev, Y.; Klekovkin, A.; Shestakov, V. Experimental investigations of a highly maneuverable mobile omniwheel robot. *Int. J. Adv. Robot. Syst.* **2017**, *14*. [[CrossRef](#)]
31. Thuruthel, T.G.; Falotico, E.; Manti, M.; Laschi, C. Stable Open Loop Control of Soft Robotic Manipulators. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1292–1298. [[CrossRef](#)]
32. Gillespie, M.T.; Best, C.M.; Killpack, M.D. Simultaneous position and stiffness control for an inflatable soft robot. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1095–1101.



33. Gillespie, M.T.; Best, C.M.; Townsend, E.C.; Wingate, D.; Killpack, M.D. Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In Proceedings of the 2018 IEEE International Conference on Soft Robotics (RoboSoft), Livorno, Italy, 24–28 April 2018; pp. 39–45.
34. Ansari, Y.; Manti, M.; Falotico, E.; Cianchetti, M.; Laschi, C. Multiobjective Optimization for Stiffness and Position Control in a Soft Robot Arm Module. *IEEE Robot. Autom. Lett.* **2018**, *3*, 108–115. [\[CrossRef\]](#)
35. Arimoto, S.; Kawamura, S.; Miyazaki, F. Bettering Operation of Dynamic Systems By Learning: A New Control Theory for Servomechanism or Mechatronics Systems. *Proc. IEEE Conf. Decis. Control* **1984**, 1064–1069. [\[CrossRef\]](#)
36. Buchheit, K. Optimal iterative learning control of an extrusion plant. In Proceedings of the IEE International Conference on Control '94, Coventry, UK, 21–24 March 1994; Volume 1994, pp. 652–657.
37. Meng, D.; Jia, Y.; Du, J. Stability of varying two-dimensional Roesser systems and its application to iterative learning control convergence analysis. *IET Control Theory Appl.* **2015**, *9*, 1221–1228. [\[CrossRef\]](#)
38. Ouerfelli, H.; Attia, S.B.; Salhi, S. Switching-iterative learning control method for discrete-time switching system. *Int. J. Dyn. Control* **2018**, *6*, 1755–1766. [\[CrossRef\]](#)
39. Nusawardhana, A.; Zak, S.H.; Crossley, W.A. Nonlinear synergetic optimal controllers. *J. Guid. Control. Dyn.* **2007**, *30*, 1134–1148. [\[CrossRef\]](#)
40. Lin, M.M.; Chiang, C.Y. An accelerated technique for solving one type of discrete-time algebraic Riccati equations. *J. Comput. Appl. Math.* **2018**, *338*, 91–110. [\[CrossRef\]](#)
41. Liu, L.; Tan, K.K.; Chen, S.-L.; Huang, S.; Lee, T.H. SVD-based Preisach hysteresis identification and composite control of piezo actuators. *ISA Trans.* **2012**, *51*, 430–438. [\[CrossRef\]](#)
42. Hassani, V.; Tjahjowidodo, T.; Do, T.N. A survey on hysteresis modeling, identification and control. *Mech. Syst. Signal Process.* **2014**, *49*, 209–233. [\[CrossRef\]](#)
43. Hui, C.; Yonghong, T.; Xingpeng, Z.; Ruili, D.; Yahong, Z. Identification of Dynamic Hysteresis Based on Duhem Model. In Proceedings of the 2011 Fourth IEEE International Conference on Intelligent Computation Technology and Automation, Shenzhen, China, 28–29 March 2011; pp. 810–814.
44. Sung, B.-J.; Lee, E.-W.; Kim, I.-S. Displacement Control of Piezoelectric Actuator using the PID Controller and System Identification Method. In Proceedings of the 2008 Joint International Conference on Power System Technology and IEEE Power India Conference, New Delhi, India, 12–15 October 2008; pp. 1–7.
45. Chang-Li, L.; Shou-Zhu, H.; Hai-Lin, G.; Xue-Jun, W.; Wen-Jun, Z. Feed-forward control of stack piezoelectric actuator. *Opt. Precis. Eng.* **2016**, *24*, 2248–2254. [\[CrossRef\]](#)
46. Chao, P.C.-P.; Liao, P.-Y.; Tsai, M.-Y.; Lin, C.-T. Robust control design for precision positioning of a generic piezoelectric system with consideration of microscopic hysteresis effects. *Microsyst. Technol.* **2011**, *17*, 1009–1023. [\[CrossRef\]](#)
47. Liu, D.; Guo, W.; Wang, W. Second-order sliding mode tracking control for the piezoelectric actuator with hysteretic nonlinearity. *J. Mech. Sci. Technol.* **2013**, *27*, 199–205. [\[CrossRef\]](#)

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



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).