Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **68**, *4*, pp. 423–429, Bucarest, 2023

LEARNING CONTROL FOR TRAJECTORY TRACKING OF NONLINEAR INERTIA WHEEL INVERTED PENDULUM

IMEN SAIDI[1,2], NAHLA TOUATI[1]

Keywords: Nonlinear underactuated system; Iterative learning control algorithm; Stability; Tracking; Maximum absolute of errors.

This paper proposed the iterative learning control for nonlinear underactuated systems. To improve the convergence speed of iterative learning control for such systems and reduce the fluctuation of the system error, an exponential variable gain D-type closed-loop iterative learning control algorithm was chosen. MATLAB simulation analysis was then performed on an underactuated, nonlinear, and unstable system, namely the inertia wheel inverted pendulum. The simulation results show that the algorithm is effective. Good tracking performance was achieved. The system converges to stable limit cycles after a few iterations, ensuring smooth errors and satisfactory convergence speed.

1. INTRODUCTION

Many industrial processes are intrinsically characterized by repetitive behavior. The iterative learning control (ILC) has been used to control such systems. It has grown in popularity considerably over the past two decades [1]. ARIMOTO proposed it in 1984 [2]. The ILC control is an efficient control concept that iteratively improves the behaviors of systems that perform repeated tasks such as assembly line tasks, chemical batch processes, reliability testing rigs, etc. Just as humans learn skills by trial and error, the ILC system learns the dynamics of the system by repeated trials [3]. The fundamental principle of the ILC algorithm is to properly refine the input sequence from one trial to the next so that as more and more trials are executed, the output will approach the desired trajectory, and the tracking error will converge to zero iteratively [4,5]. In the literature, many systems with repetitive behavior, such as manipulator robots, were controlled by ILC [4]. It has been reported that the error can be reduced to 1/1000 by 12 iterations with only simple changes in the control signals.

The ILCs can be classified into three major categories: the previous cycle learning or offline learning, the current cycle learning or online learning, and the previous and current learning control or online-offline learning.

The online ILC only relies on current iteration errors [5]. In the offline ILCs, information from the previous iteration(s) was used to calculate the control input for the current iteration.

The existing ILC methods use the PID controller, characterized by their simplicity, robustness, fast response, stability, adaptability, and widespread usage. It is relatively easy to implement and understand, can handle disturbances and parameter changes, responds quickly to system changes, achieves stable control, can be tuned for specific systems, and has a long history of successful application in various industries [6–8].

Among the iterative learning controllers, which are based on a combination of the PID controller, we cite D-Type closed-loop ILC, PD-Type closed-loop ILC, and the ILC with exponential variable gain [4,9].

All these techniques were proposed for continuous nonlinear systems, MIMO linear systems, and sampled linear systems [10]. However, considering the ILC control becomes more challenging for complex systems such: as underactuated, non-minimum phase, or nonlinear systems. In particular, the under-actuation has led researchers in the field to analyze, on a case-by-case basis, examples of under-actuated mechanical systems of small dimensions (*i.e.*, having little degrees of freedom). These systems, although small, exhibit strongly nonlinear dynamics, which complicates their control even more. Some efforts to classify under-actuated mechanical systems have been carried out in [11], where the classification is based on certain characteristics of the model of the studied systems. As for non-minimum phase systems, they often exhibit harmful behavior since they can even react in the opposite direction to the set points in transient conditions.

This paper focuses on controlling an underactuated, nonlinear, and non-minimum phase system, namely the inertia wheel inverted pendulum. The ILC approach was proposed to ensure satisfactory tracking features. The proposed ILC controller was based on optimizing the reference trajectories for the unactuated coordinates. The control objectives were to stabilize the internal dynamics of the closed-loop system and converge the actuated coordinates to stable limit cycles. The ILC was proposed for the inertia wheel inverted pendulum. It was validated through numerical simulations. The obtained results prove the effectiveness of this approach and its ability to ensure good tracking performance.

The remainder of this paper is organized as follows. In section 2, the iterative learning control structure is presented. Section 3 is devoted to the description of the inertia wheel inverted pendulum. The proposed control strategy is illustrated via simulation in section 4. Finally, concluding remarks are drawn in section 5.

2. ITERATIVE LEARNING CONTROL

The iterative learning control approach is a very effective control strategy in dealing with repeated tracking control that has grown in popularity significantly over the past decades. It aims to improve tracking performance for systems that work in a repetitive mode. The ILC's basic idea is that the data generated from the previous trial are used to construct the control input for the current trial. The control input in each trial is adjusted by using the tracking error obtained from the previous trial [3].

When designing a controller, the control objective is to ensure for a given system good performance, acceptable steady-state error, and transient response. Controlling the transient response is difficult considering the model's nonlinearities, uncertainties that may occur, and under-actuation [12].

[1] Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, Automatic Research Laboratory, E-mail : imen.saidi@enit.utm.tn
[2] Université de Tunis, Ecole Nationale Supérieure d'Ingénieurs de Tunis, E-mail : nehla.kermani@esprit.tn

DOI: 10.59277/RRST-EE.2023.68.4.17

2.1. MAIN OF ITERATIVE LEARNING CONTROL

Consider a nonlinear dynamic model described as follows:

$$\begin{cases} x(t) = f(x(t), u(t), t), \\ y(t) = g(x(t), u(t), t), \end{cases} \quad (1)$$

where $x(t) \in \Re^n$, $y(t) \in \Re^m$ and $u(t) \in \Re^r$ are respectively the system state, output, and input vectors, and $f(.)$, $g(.)$ are nonlinear vector functions.

For the expected control $u_d(t)$, if the initial states $x_k(0)$ and expected output $y_d(t)$ are given, for the given period $t \in [0, T]$, according to the learning algorithm by repeated operation, we can realize $u_k(t) \to u_d(t)$ and $y_k(t) \to y_d(t)$ in a k times.

It can be represented below

$$\begin{cases} x_k(t) = f(x_k(t), u_k(t), t) \\ y_k(t) = g(x_k(t), u_k(t), t) \end{cases} . \quad (2)$$

The iterative learning control can be derived into open-loop learning control and closed-loop learning control [13]. For the open loop control, the $k+1$ times control is equal to the correction of the k times control combine with the k times output error:

$$u_{k+1}(t) = L(u_k(t), e_k(t)) . \quad (3)$$

The tracking error is defined as

$$e_k(t) = y_d(t) - y_k(t), \quad (4)$$

where $y_d(t)$ is the desired trajectory to be tracked by the system and $t \in [0, T]$ and $y_k(t)$ is the system output at the iteration *k*. While the closed-loop learning strategy is to take the error in *k+1* times as the correction of learning

$$u_{k+1}(t) = L\big(u_k(t), e_{k+1}(t)\big), \quad (5)$$

where *L* is a linear or nonlinear operation.

ILC algorithms were created to improve transient responses for repetitive systems. In traditional control, a given repetitive system will produce the same error cycle after cycle since the error and the control input will be the same. ILC algorithms attempt to improve the tracking performance by adjusting the input for the next cycle, based on the error from the previous cycle. This technique has the advantage that it can be used without much knowledge of the system dynamics to be controlled [14,15].

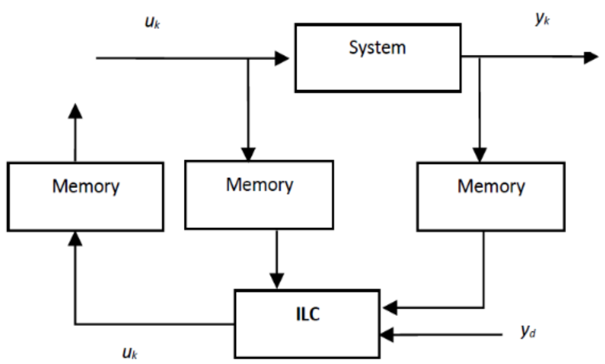

Fig. 1 – Control block diagram of ILC.

The classical formulation of the ILC design problem is to find an update mechanism for the input trajectory of a new cycle based on the information from previous cycles so that the output trajectory converges asymptotically to the reference trajectory [16]. A typical ILC idea is depicted in block diagram form in Fig. 1. It shows the next trial's control input to be calculated from the previous trial's control input and transient output error [17,18]. Here wandre $y_k$and $u_k$are respectively the output and the control input of the system in the previous iteration k, and $-u_{k+1}$is the control input at the current iteration $k+1$.

The control input u_{k+1} is evaluated using the error e_k which represents the difference between y_d and y_k and the control signal u_k collected in the previous cycle *k*.

According to the block diagram illustrated in Fig. 1, the ILC law updated the input with the following equation:

$$u_{k+1}(t) = u_k(t) + Ke_k(t+1). \quad (6)$$

The control input of the following cycle $u_{k+1}(t)$, is deduced from the previous cycle control action $u_k(t)$ and the previous error $e_k(t+1)$ multiplied by the learning gain *K*.

The objective of the iterative learning control is to find a control sequence with the ability to reduce tracking error for the whole trajectory based on past tracking.

$$\lim_{k\to\infty} e_k(t) = \lim_{k\to\infty} \dot{e}_k(t) = 0. \quad (7)$$

2.2. DIFFERENT ITERATIVE LEARNING CONTROL ALGORITHMS

There are many ILC algorithms, such as traditional P-type ILC, which updates the control input signal as a function of the previously stored control input and the stored output errors. In contrast, a D-type ILC updates the control input as a function of the previously stored control input and the stored derivative of the output errors. In the literature, several ILC algorithms were discussed and described as follows:

2.2.1. D-type closed-LOOP ILC

The D-type ILC is a classic ILC that uses the derivative of the error rather than the error itself. This means that the ILC seeks to perfectly follow the curvature of the trajectory and not the trajectory itself. As soon as the trajectory is traced, the derivative of the error drops to zero, and the algorithm waits for a disturbance that would deviate the system from the desired trajectory. The D-type ILC can potentially reduce the tracking errors caused by the disturbances [19].

The following equation describes the control law output of the D-type iterative learning of the system:

$$u_{k+1}(t) = u_k(t) + K_d\big(\dot{y}_d(t) - \dot{y}_{k+1}(t)\big), \quad (8)$$

where K_d is the constant gain matrix.

2.2.2. PD-type closed-LOOP ILC

One advantage of the P-type ILC is the sole measurement requirement. However, P-type ILC alone is poorly suited to integrating plants and is particularly sensitive to non-repeating disturbances; therefore, it is often coupled with a feedback controller. Therefore, a combination of P-type and D-type ILC, or PD-type ILC, is a promising approach to improving the tracking performance and reducing the influence of the disturbance [20,21]. The PD-type ILC combines the 'P' and 'D' approaches to achieve a fast convergence speed and better tracking performance. It is possible to vary the gains to give more priority to a part of the algorithm. In addition, it is important to do this since the magnitude of the error and its derivative may be completely out of proportion. The control input equation for such an algorithm is as follows:

$$u_{k+1}(t) = u_k(t) + K_p\big(y_d(t) - y_{k+1}(t)\big) + \\ +K_d\big(\dot{y}_d(t) - \dot{y}_{k+1}(t)\big), \tag{9}$$

where K_p is the constant gain matrix.

2.1.3. EXPONENTIAL VARIABLE GAIN D-TYPE CLOSED-LOOP ILC

The exponential variable gain D-type learning law is defined by eq. (10). The current control input is derived from the derivative signal of the past control input and the current output error signal by learning law. At the end of each repeated operation, the new control is calculated according to the learning law and stored to refresh the old control quantity [22]

$$u_{k+1}(t) = u_k(t) + \lambda(t)K_d\big(\dot{y}_d(t) - \dot{y}_{k+1}(t)\big), \tag{10}$$

where $\lambda(t) = e^{nt} (n > 0)$.

To evaluate the initial value of *n*, we begin by exploring a range from 0.1 to 10 and then adjusting based on the simulation results and the system's performance. A higher value is recommended if the response is slow and a lower value if there is instability. The iterative adjustment of *n* relies on analyzing results and understanding dynamic characteristics, considering stability, response speed, and damping.

3. THE INERTIA WHEEL INVERTED PENDULUM

The inertia wheel inverted pendulum is a nonlinear, underactuated mechanical system with fewer actuators than degrees of freedom. Underactuated mechanical systems bring challenges to the control task since conventional controls for fully actuated mechanical systems cannot be implemented for such systems [23,24]. The control inputs act only on the actuated dynamics. Besides, these systems present non-minimum phase behaviour which complicates their control even more. Suitable control techniques should then be considered to control and stabilize the actuated coordinates and the remaining part of the system called internal dynamics [25–27].

The inertia wheel inverted pendulum, depicted in Fig. 2, was widely considered as an academic system to implement new control approaches. It includes an inverted pendulum equipped with a rotating wheel [29].

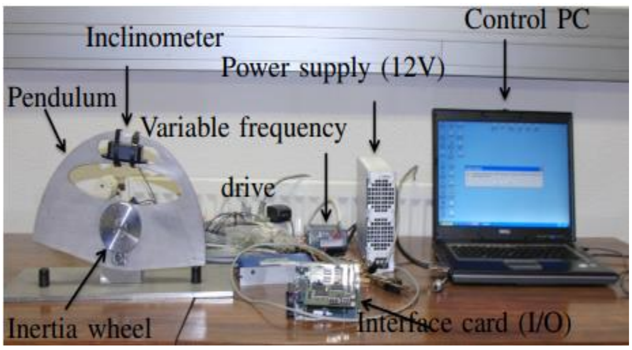

Fig. 2 – The inertia wheel inverted pendulum [29].

Figure 3 shows the geometry of the inverted pendulum in a Galilean frame. θ_1 is the angle between the axis of the pendulum and the vertical which corresponds to the so-called passive connection. As for θ_2, it designates the angle between the inertia wheel and the pendulum which corresponds to the active connection. G_1 presents the center of gravity of the pendulum and G_2 is that of the inertia wheel. We denote by G the center of gravity of the global system.

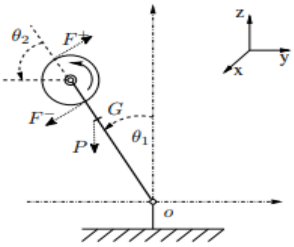

Fig. 3 – Schematic view of the system.

The Euler-Lagrange formalism is based on the Lagrange equation [11,26,29]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial q_i}\right) - \frac{\partial L}{\partial q_i} = Q_i; \quad i = 1,2, \tag{11}$$

where $L = T - V$ is the Lagrangian of the system where T is the total kinetic energy of the system and V is the total potential energy of the system, $q_i = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ are the vectors of the generalized coordinates, $Q_q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$: the vector of generalized forces associated with θ_i.

The total kinetic energy T is decomposed into pendulum energy and inertia wheel energy [11,26,29]:

$$T = \frac{1}{2}(m_1 l_1^2 + i_1)\dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 + \frac{1}{2} i_2(\dot{\theta}_1 + \dot{\theta}_2)^2. \tag{12}$$

The total potential energy of the system is decomposed as follows [11, 26, 29] :

$$V = m_1 l_1 g \cos\theta_1 + m_2 l_2 g \cos\theta_1. \tag{13}$$

Using the equations for the kinetic and potential energies eq. (12) and (13), the Lagrangian can be defined as follows

$$L = T - V = \frac{1}{2} I \dot{\theta}_1^2 + \frac{1}{2} i_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 - \overline{ml} g \cos\theta_1, \tag{14}$$

with $I = m_1 l_1^2 + m_2 l_2^2 + i_1$ and $\overline{ml} = m_1 l_1 + m_2 l_2$, where: m_1 and $m_2$are the pendulum and the inertia wheel masses, l_1 and $l_2$are respectively the distances between the origin to the gravity center of the pendulum and the wheel; $i_1$and $i_2$are respectively the inertia moments of the beam and the wheel; –$\dot{\theta}_1$ and $\dot{\theta}_2$ are respectively the velocity of the pendulum body and the velocity of inertia wheel.

By replacing eq. (14) in eq. (11), the nonlinear model of the inertia wheel inverted pendulum is described by the following equation

$$\begin{cases} \ddot{\theta}_1 = \frac{1}{I}\left[\tau_1 - \tau_2 + \overline{ml}\, g \sin\theta_1\right], \\ \ddot{\theta}_2 = \frac{1}{I i_2}\left[-i_2 \tau_1 + (i_2 + I)\tau_2 - i_2 \overline{ml}\, g \sin\theta_1\right], \end{cases} \tag{15}$$

where –$\ddot{\theta}_1$and $\ddot{\theta}_2$ are respectively the acceleration of the pendulum body and the acceleration of the inertia wheel; -τ_2 is the torque induced by the motor while τ_1 is the disturbing torque which is assumed to be zero [11,29]. Table 1 presents the summary of the geometric and dynamic parameters of the system.

Table 1.
The parameters of the inertia wheel inverted pendulum

Description	Parameters values
Body mass	$m_1 = 3.3081$ kg
Wheel mass	$m_2 = 0.33081$ kg
Body center of mass position	$l_1 = 0.06$ m
Wheel center of mass position	$l_2 = 0.044$ m
Body inertia	$i_1 = 0.03146$ kgm^2
Wheel inertia	$i_2 = 0.00041$ kgm^2
Gravity acceleration	$g = 9.81$ms^{-2}

In the next step, we will implement the PD-type iterative learning control on the inertia wheel inverted pendulum system. This PD controller provides the advantages of stability and precision for our study system [31,32].

4. SIMULATION RESULTS

An inverted inertia wheel pendulum has the dynamic characteristics of high nonlinearity of an underactuated system and strong coupling. Iterative learning control seems effective for underactuated systems because a simple learning algorithm characterizes it and is independent of the detailed model of the controlled system.

A fundamental algorithm for the iterative learning control algorithm involves using proportional-derivative (PD) control. In this iterative process, the control input is iteratively adjusted based on the proportional and derivative components. This allows the system to learn from past errors and enhance its performance in tracking a specified trajectory over successive iterations. The iterative learning control algorithm is given as follows:

- **Step 1: Initialize**

  Set iteration count k = 1
  Initialize control input u[k-1]
  Set reference trajectory r[k]
- **Step 2: Repeat until convergence**
  1. Measure system output y[k]
  2. Compute tracking error e[k] = r[k] - y[k]
  3. PD Control :
     a. Compute proportional term : Kp_term = Kp * e[k]
     b. Compute derivative term:
        Kd_term = Kd * (e[k] - e[k-1])
     c. Compute updated control input :
        u[k] = u[k-1] + Kp_term + Kd_term
  4. Apply control input u[k] to the system
  5. Increment iteration count: k = k + 1
  6. Check convergence criteria:
- **Step 3: Store current control input for the next iteration**

  Set u[k-1] = u[k-1]
- **End Repeat**

Here k is the iteration index; u[k] is the control input at iteration k; y[k] is the system output at iteration k; r[k] is the reference trajectory at iteration k; e[k] is the tracking error at iteration k; K_p and K_d are respectively the proportional and derivative gains.

To underline the interest and the performance of the iterative learning control, this command is applied to the inertia wheel inverted pendulum and is described by the eq. (15).

The system's desired trajectories to be tracked are described by:

$$\begin{cases} \theta_{1ref}(t) = \sin(3t)\,, t \in [0,T] \\ \theta_{2ref}(t) = \cos(3t)\,, t \in [0,T] \end{cases} . \tag{16}$$

The desired trajectories of the velocities $\dot{\theta}_{1ref}$, $\dot{\theta}_{2ref}$ are obtained by deriving the outputs described by eq. (16). They can be calculated as in eq. (17).

$$\begin{cases} \dot{\theta}_{1ref}(t) = 3\cos(3t)\,, t \in [0,T] \\ \dot{\theta}_{2ref}(t) = -3\sin(3t)\,, t \in [0,T] \end{cases} . \tag{17}$$

The tracking errors of pendulum angular position Error 1 and inertia wheel position Error 2 are described by:

$$\begin{cases} \text{Error1} = \theta_{1ref} - \theta_1 \\ \text{Error2} = \theta_{2ref} - \theta_2 \end{cases} \tag{18}$$

The tracking errors of pendulum angular velocity Error 3 and inertia wheel velocity Error 4 are then defined as:

$$\begin{cases} \text{Error3} = \dot{\theta}_{1ref} - \dot{\theta}_1, \\ \text{Error4} = \dot{\theta}_{2ref} - \dot{\theta}_2. \end{cases} \tag{19}$$

The simulation parameters are selected as $T = 3$ s, $n = 0.8$, $K_d = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$ and $K_p = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}$. The maximum number of iterations is 20, and the simulation lasts 3 seconds. The system simulation results are shown in Fig. 4 to Fig. 15.

Scenario 1: D-type closed-LOOP ILC

Figure 4 shows the evolution of the outputs θ_1 and θ_2 their references to the last tracking process of the final iteration. We notice that the outputs in the blue line of the system suitably follow their references represented in the red line.

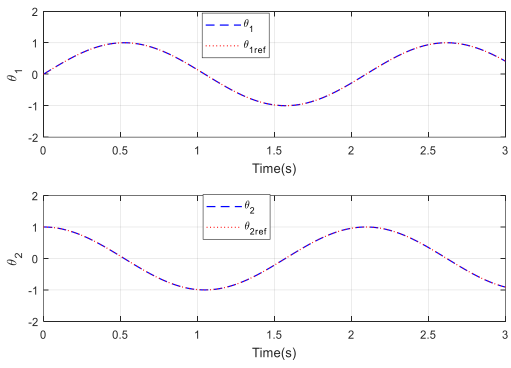

Fig. 4 – Pendulum angular position θ_1 and inertia wheel position θ_2 of the last iteration using D-type closed-loop ILC algorithm.

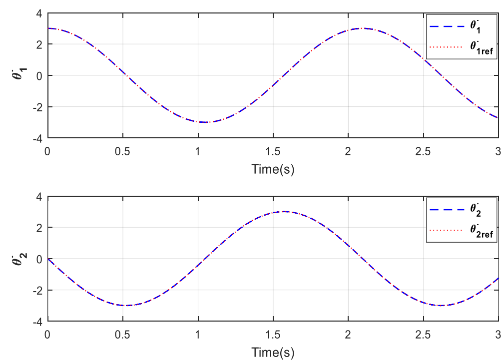

Fig. 5 – Pendulum angular velocity $\dot{\theta}_1$ and inertia wheel velocity $\dot{\theta}_2$ of the last iteration using D-type closed-loop ILC algorithm.

In Fig. 5, the two angular velocity signals $\dot{\theta}_1$ and $\dot{\theta}_2$, both have very good tracking effects. ILC method with D-type closed-loop can stably and completely track the expected trajectories after 20 iterations.

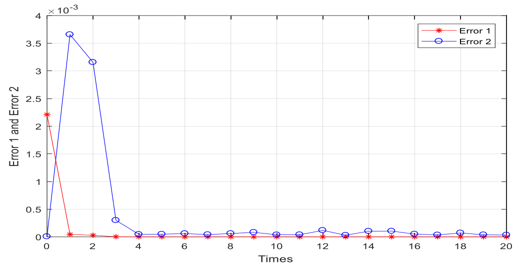

Fig. 6 – Maximum absolute values of Error 1 and Error 2 in 20 iterations using D-type closed-loop ILC algorithm.

Figure 6 shows a curve chart that describes the maximum absolute values of Error 1 for the pendulum angular position, which is represented by blue data, and Error 2 for the inertia wheel position, which is represented by red data. These errors are shown in 20 iterations. With the increase in the number of iterations, the maximum angular errors for θ_1 and θ_2 decrease rapidly and converge faster to zero, showing very good tracking effects.

The maximum errors Error 3 and Error 4 have very low values which show that the D-type closed-loop ILC command, applied in this scenario is very effective and that the inverted pendulum system is very precise because the steady state errors of velocities for outputs are very close to zero.

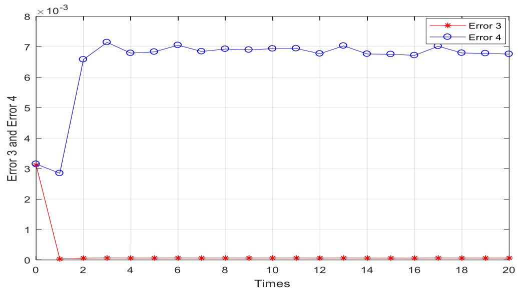

Fig. 7 – Maximum absolute values of Error 3 and Error 4 in 20 iterations using D-type closed-loop ILC algorithm.

Simulation results demonstrate the effectiveness of the ILC algorithm of closed-loop D-type in terms of stability, rapidity response of the system, and accuracy for inertia wheel inverted pendulum control.

Scenario 2: PD-type closed-LOOP ILC

Figure 8 represents the evolutions of θ_1 and θ_2 in solid line while their references are in dashed line. This figure shows clearly that the static steady-state errors converge to zero. The output trajectories converge asymptotically to the desired reference trajectories after 20 iterations with good tracking performance.

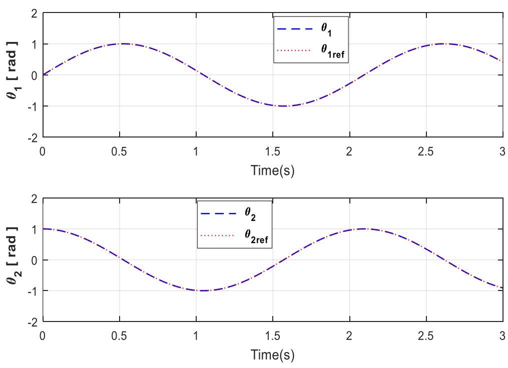

Fig. 8 – Pendulum angular position θ_1 and inertia wheel position $\theta_2$of the last iteration using PD-type closed-loop ILC algorithm.

Figure 9 shows that outputs $\dot{\theta}_1$ and $\dot{\theta}_2$ match perfectly with their reference inputs. Figure 10 shows the variations of Error 1 and Error 2 corresponding to the errors between theta sub 1, theta sub 2, and their references. We notice that the errors decrease with the number of iterations and tend towards zero. The system corrects itself, and after only one iteration, the real output θ_1 tends towards the desired output. After 5 iterations, the real output $\theta_2$converges to its reference too.

Figure 11 illustrates the maximum absolute values of Error 3, for $\dot{\theta}_1$, and Error 4, for $\dot{\theta}_2$, with iterative times. It is quite clear that the ILC law in this scenario manages to bring the trajectories back around their references in a few learning cycles, hence the effectiveness of the ILC control for nonlinear under-actuated systems.

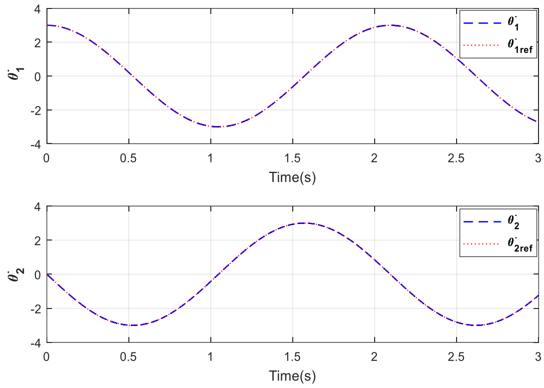

Fig. 9 – Pendulum angular velocity $\dot{\theta}_1$ and inertia wheel velocity $\dot{\theta}_2$ of the last iteration using PD-type closed-loop ILC algorithm.

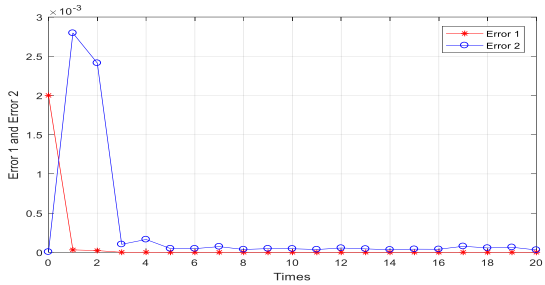

Fig. 10 – Maximum absolute values of Error 1 and Error 2 in 20 iterations using PD-type closed-loop ILC algorithm.

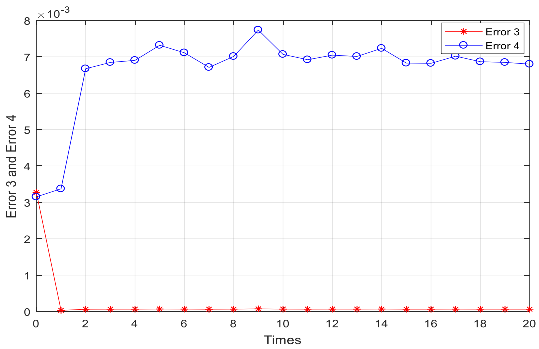

Fig. 11 – Maximum absolute values of Error 3 and Error 4 in 20 iterations using PD-Type closed-loop ILC algorithm.

The maximum error amplitudes for Error 1, Error 2, Error 3, and Error 4 are reduced, which shows the superiority of the closed-loop PD-type algorithm over the closed-loop D-type algorithm.

Scenario 3: Exponential variable gain D-type closed-LOOP ILC

From Fig. 12, we notice that the angular positions of the pendulum and the inertia wheel perfectly follow the desired trajectories with zero static errors. In addition, the system can carry out a complete follow-up of the reference trajectories. Similarly, the velocities $(\dot{\theta}_1, \dot{\theta}_2)$ of the inertia wheel inverted system follow one another at their references according to Fig. 13.

According to Fig. 14, the output pendulum angular position θ_1 and position of the inertia wheel θ_2 are precise. From iteration 1, the output θ_1 follows its setpoint. While from iteration 5, output θ_2 and her input reference follow one another.

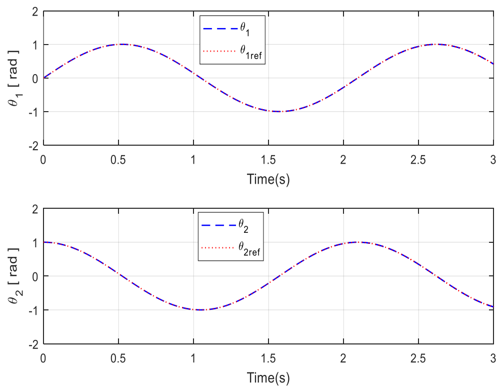

Fig. 12 – Pendulum angular position θ_1 and inertia wheel position θ_2 of the last iteration using exponential variable gain D-type closed-loop ILC.

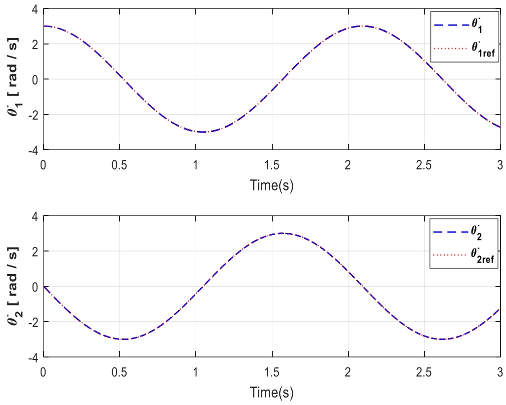

Fig. 13 – Pendulum angular velocity $\dot{\theta}_1$ and inertia wheel velocity $\dot{\theta}_2$ of the last iteration using exponential variable gain D-type closed-loop ILC.

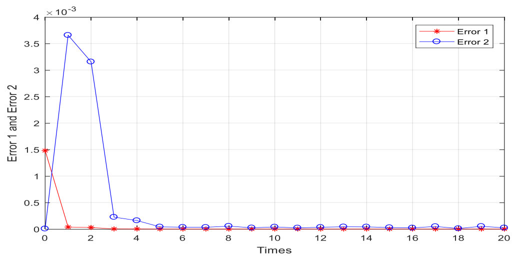

Fig. 14 – Maximum absolute values of Error 1 and Error 2 in 20 iterations using exponential variable gain D-Type closed-loop ILC.

Figure 15 shows $\dot{\theta}_1$ follows its input reference from iteration 1 while $\dot{\theta}_2$ follows its reference with a very low static error speed.

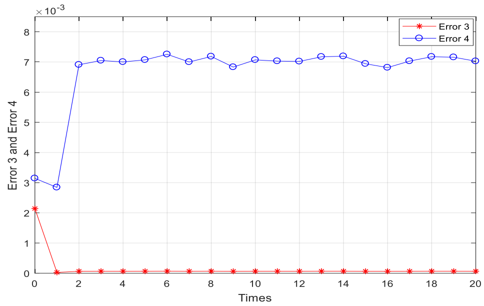

Fig. 15 – Maximum absolute values of Error 3 and Error 4 in 20 iterations using exponential variable gain D-type closed-loop ILC.

During the tracking processes of the proposed three ILC algorithms, the comparisons of the maximum displacement errors and the maximum speed errors of the inertia wheel inverted pendulum system are shown in Table 2. The maximum errors of angular displacements and angular velocities of the third algorithm in scenario 3 are Error1 =14.8 10^{-3} rad, Error2 = 3.64 10^{-3} rad, Error3 = 2.1 10^{-3} rad/s and Error4 = 7.1310^{-3} rad/s. These maximum errors are very low values compared to the maximum errors obtained in scenarios 1 and 2. Compared with the results of the former two algorithms controlling the P-type closed-loop ILC algorithm and PD-type closed-loop ILC algorithm, the best advantage is demonstrated. The Exponential variable gain D-type closed-loop ILC algorithm shows better error results; hence, this algorithm has higher iteration accuracy and is recommended preferentially.

Table 2

Comparisons of the maximum displacement and maximum velocity errors of inertia wheel inverted pendulum

Name of controllers	Max error position	Max error velocity
D-type closed-loop ILC	Error 1=2.21 10^{-3} Error 2= 3.66 10^{-3}	Error 3=3.11 10^{-3} Error 4= 7.14 10^{-3}
PD-type closed-loop ILC	Error 1=2 10^{-3} Error 2= 2.79 10^{-3}	Error 3=3.15 10^{-3} Error 4=7.73 10^{-3}
Exponential variable gain D-type closed-loop ILC	Error 1=14.8 10^{-3} Error 2= 3.64 10^{-3}	Error 3=2.1 10^{-3} Error 4= 7.1310^{-3}

4. CONCLUSIONS

This paper studies an iterative learning control for a class of nonlinear underactuated systems. The PD-Type ILC was chosen and proposed for the inertia wheel inverted pendulum. We have proved that the proposed algorithm can guarantee that the output scan tracks the desired trajectories completely on a finite and small-time interval. The simulation example of an inertia wheel inverted pendulum shows the effectiveness of the proposed iterative learning control algorithm.

The proposed control approach can be extended to deal with robustness towards disturbances and for online learning strategies. In future work, the proposed method can also be implemented on real platforms, such as industrial shock absorbers, which present characteristics very close to those considered in our contribution.

The simulation results show that the algorithm is effective. the improved iterative learning law makes the iterative error smoother and improves the convergence speed. Applying the iterative learning control in trajectory tracking of an inertia wheel inverted pendulum has achieved a good control effect.

Received on 29 June 2023

REFERENCES

1. H.S. Ahn, Y. Chen, K L. Moore, *Iterative learning control: brief survey and categorization*, IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), **37**, *6*, pp. 1099–1121 (2007).
2. S. Arimoto, S. Kawamura, F. Miyazaki, *Bettering operation of robotics by leaning*, Journal of Robotic Systems, **1**, *2*, pp. 123–140 (1984).
3. D. Meng, K.L. Moore, *Robust iterative learning control for non-repetitive uncertain systems*, IEEE Transactions on Automatic Control, **62**, *2*, pp. 907–913 (2017).
4. Y. Li, A. An, J. Wang, H. Zhang, *Iterative learning control of exponential variable gain based on initial state learning for upper limb rehabilitation robot*, Chinese Automation Congress, **11**, *22*, pp. 4947–4952 (2019).
5. F. Bouakrif, *Iterative learning control for strictly unknown nonlinear systems subject to external disturbances*, International Journal of Control, Automation, and Systems, **9**, *4*, pp. 642–648 (2011).
6. N. Ali, W. Alam, M. Pervaiz, J. Iqbal, *Nonlinear adaptive backstepping control of permanent magnet* synchronous motor, Rev. Roum. Sci. Techn.– Électrotechn. et Énerg., **66**, *1*, pp. 9–14 (2021).
7. R. Arun, R. Muniraj, M.S.W. Iruthayarajan, P*erformance analysis of proportional integral derivative controller with delayed external reset and proportional integral derivative controller for time delay process*, Rev. Roum. Sci. Techn.– Électrotechn. et Énerg., **66**, *4*, pp. 267–273 (2021).
8. R. Muniraj, M.W. Iruthayarajan, R. Arun, T. Sivakumar, *Parameter optimization of multi-objective robust proportional integral derivative controller with filter using multi-objective evolutionary algorithms*, Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **64**, 3, pp. 259–265 (2019)
9. Y. Q. Chen, K.L. Moore, J. Yu, T. Zhang, *Iterative learning control and repetitive control in hard disk drive industry – a tutorial*, International Journal of Adaptive Control and Signal Processing, **22**, *4*, pp. 325–343 (2008).
10. F. Bouakrif, *Reference model iterative learning control for linear systems*, 18th Mediterranean Conference on Control and Automation, Marrakech, 2010.
11. I. Saidi, N. Touati, *Nonlinear predictive control for trajectory tracking of underactuated mechanical systems*, Przegląd Elektrotechniczny, **97**, *6*, pp. 30–33 (2021).
12. I. Bejaoui, I. Saidi, D. Soudani, *New internal model controller design for discrete over-actuated multivariable system*, 4th International Conference on Control Engineering & Information Technology, Tunisia, 2016.
13. Y. Hu, B. Zuo, J. Li, *Fast algorithm of open-closed loop pd-type iterative learning control for dc motor*, IEEE 6th World Congress on Intelligent Control and Automation, China, 2006.
14. R. Longman, *Iterative learning control and repetitive control for engineering practice*, Automation and Systems, **73**, 10, pp. 930–954 (2012).
15. D. Ke, Q. Ai, W. Meng, C. Zhang, Q. Liu, *Fuzzy PD-type iterative learning control of a single pneumatic muscle actuator, intelligent robotics and applications*, **7**, *16*, pp. 812–822 (2017).
16. Y. Sun, Z A. Li, *Open-loop D-type iterative learning control for a class of nonlinear systems with arbitrary initial value*, Journal of System Simulation, **20**, *24*, pp. 6767–6770 (2008).
17. Q. Jia, R. Ma , C. Zhang, R. Varatharajoo, *Spacecraft attitude fault-tolerant stabilization against loss of actuator effectiveness: a novel iterative learning sliding mode approach*, Advances in Space Research, **72**, *2*, pp. 529–540 (2023).
18. S. Lu, S. Jingzhuo, *Adaptive PI control of ultrasonic motor using iterative learning methods*, ISA Transactions, In press (2023).
19. Y. Qiongxia, H. Zhongsheng, X. Jian-Xin, *D-Type ILC based dynamic modeling and norm optimal ILC for high-speed trains*, IEEE Transactions on Control Systems Technology, **26**, *2*, pp. 652 – 663 (2017).
20. M.X. Sun, B.J. Huang, X.Z. Zhang, *PD-type iterative learning control for a class of nonlinear systems*, Acta Automatica Sinica, **4**, *5*, pp. 711–714 (1998).
21. A. Norouzi, C.R. Koch, *Integration of PD-type iterative learning control with adaptive sliding mode control*, 21st IFAC World Congress, **53**, *2*, pp. 6213–6218 (2020).
22. Y. Li, A. An, J. Wang, H. Zhang, *Iterative learning control of exponential variable gain based on initial state learning for upper limb rehabilitation robot*, Chinese Automation Congress, 2019.
23. R.O. Saber, *Global stabilization of a flat underactuated system: the inertia wheel pendulum*, Proceedings of the 40th IEEE Conference on Decision and Control, pp. 306–308, 2001.
24. S. Andary, A. Chemori, S. Krut, *Control of the underactuated inertia wheel inverted pendulum for stable limit cycle generation*, Advanced Robotics, **3**, *15*, 1999–2014 (2009).
25. N.H. Manai, I. Saidi, D. Soudani, *Predictive control of an underactuated System*, Proceedings of the 2nd International Conference on Advanced Systems and Electric Technologie, Hammamet, Tunisia, pp. 90–95, 2019.
26. I. Saidi, N. Touati, *Sliding mode control to stabilization of nonlinear underactuated mechanical systems*, Przegląd Elektrotechniczny, **97**, *7*, pp. 106–109 (2021).
27. O. Khan, M. Pervaiz, E. Ahmad, J. Iqbal, *On the derivation of novel model and sophisticated control of flexible joint manipulator*, Rev. Roum. Sci. Techn. – Electrotechn. et Energ., **62**, *1*, pp. 103–108 (2017).
28. O.G. Muler, *Modeling and simulation of an aircraft electrical power*, Rev. Roum. Sci. Techn. – Electrotechn. et Energ., **68**, 2, pp. 224–231 (2023).
29. A. Hfaiedh, A. Chemori, A. Abdelkrim, *Stabilization of the Inertia wheel inverted pendulum by advanced IDA-PBC based controllers: comparative study and real-time experiments*, 17th International Multi-Conference on Systems, Signals & Devices, Tunisia, 2020.
30. M. EL Mesbahi, A. Tgarguifa, H. Hachimi, *Robots morphologies and communication strategies trade-off in a dynamic multi-robot collaborative environment,* Rev. Roum. Sci. Techn.–Électrotechn. et Énerg., **67**, *2*, pp. 193–198 (2022).
31. R. Arun, R. Muniraj, M S. W. Iruthayarajan, *Performance analysis of proportional integral derivative controller with delayed external reset and proportional integral derivative controller for time delay process*, Rev. Roum. Sci. Techn.– Électrotechn. et Énerg., **66**, *4*, pp. 267–273 (2021).