

Research Article

Iterative Learning Control Experimental Results in Twin-Rotor Device

R. Mascaró Palliser,¹ Ramon Costa-Castelló,^{1,2} and G. A. Ramos³

¹*Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB), Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI), Universitat Politècnica de Catalunya, Barcelona, Spain*

²*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, Barcelona, Spain*

³*Department of Electric and Electronic Engineering, Universidad Nacional de Colombia, Bogotá, Colombia*

Correspondence should be addressed to G. A. Ramos; garamosf@unal.edu.co

Received 28 April 2017; Accepted 25 September 2017; Published 26 October 2017

Academic Editor: Yuri Vladimirovich Mikhlin

Copyright © 2017 R. Mascaró Palliser et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the results of applying the Iterative Learning Control algorithms to a Twin-Rotor Multiple-Input Multiple-Output System (TRMS) in order to achieve high performance in repetitive tracking of trajectories. The plant, which is similar to a prototype of helicopter, is characterized by its highly nonlinear and cross-coupled dynamics. In the first phase, the system is modelled using the Lagrangian approach and combining theoretical and experimental results. Thereafter, a hierarchical control architecture which combines a baseline feedback controller with an Iterative Learning Control algorithm is developed. Finally, the responses of the real device and a complete analysis of the learning behaviour are exposed.

1. Introduction

As a result of the development of modern society and its technology, control problems have become quite common in practically all engineering areas. In industry, control systems can be found in numerous applications such as automation, robotics, machine tool control, transportation systems, chemical processing, and medical engineering. The need of efficient control systems has led to the appearance and development of new branches among classical control, for example, adaptive control, optimal control, neural network, or intelligent control.

Iterative learning control (ILC), which has been categorized as an intelligent control methodology [1–6], is an approach for improving the performance of systems that operate repetitively. This area in control theory is based on the idea that the performance of a system that executes the same task multiple times can be improved by learning from previous executions [5]. For such systems, a controller designed using the classical tools is supposed to yield the same tracking error on each iteration because it only uses information provided by the current error. The objective of

ILC is to improve performance by storing error information from previous iterations and incorporating it into the control signals for subsequent iterations. By doing so, high performance can be achieved with low transient tracking error despite model uncertainty and repeating disturbances [5].

ILC has been applied to nonlinear system employing neural networks [7] and fuzzy strategies [8], using an inverse identification procedure [5] and with iteration-varying parameters [9]. A case for nonaffine MIMO systems is presented in [10] with a combination of linear and nonlinear ILC. As it can be seen, all these approaches use rather complex strategies to accomplish the repetitive task and the nonlinear plant model at the same time.

In this work, a Twin-Rotor plant has been used as the study case, a standard didactic device for control experiments. This laboratory setup, which is similar to a model of a helicopter, has attracted a great attention during the last decades due to its highly nonlinear and cross-coupled dynamics. Being seen as a challenging engineering problem, the Twin Rotor has been widely used as an experimentation platform in research projects [11–16] as well as in MS and PhD theses [17] to validate different types of control algorithms.

Several approaches to the tracking problem in the Twin-Rotor system can be found in recent literature. Among them, in [18], a fuzzy-PID strategy is proposed and, in [19], a Model Predictive Control is used, while, in [20, 21], the controller is based on a sliding mode strategy and, in [22], a dynamic surface controller is proposed. Recently, an ILC strategy combined with an online optimization algorithm was implemented in [23] for the tracking trajectory of a quadcopter. All these techniques show an important level of complexity in order to achieve the desired performance.

In this paper, ILC has been successfully applied to a Twin Rotor with the aim of reducing the tracking error. In a first stage, two cascade PID loops are used as the baseline controller. Although many tuning procedures can be applied [24], this classical controller is designed to provide stability at the nominal operational point and also is constructed with enough stability margins to guarantee the stable operation of the plant in a large interval around this point. Under this concept, the PI controller cannot offer good tracking of the repetitive task. Therefore an ILC strategy is then added to deal with the tracking performance. In this manner, the inherent robustness of the ILC combined with the selected baseline controller makes this approach a particularly simple and effective way of obtaining good tracking response to repetitive tasks in the Twin-Rotor system. The advantage of the proposed strategy relies on its simple structure compared with other proposals applied to nonlinear plants and also for the specific Twin-Rotor plant with repetitive tasks. Experimental results show the excellent tracking properties of the proposed strategy. Analysis of ILC tuning parameters and their influence on tracking performance is discussed.

The present paper is organized as follows. Section 2 describes the Twin Rotor and presents its dynamic model developed using the Lagrangian approach. Sections 3 and 4 present the control architecture designed in this work, which combines a baseline feedback controller with an ILC algorithm. The baseline controller, as explained in Section 3, consists of two cascade control loops: the inner one is governed by two PI type controllers that adjust the rotational speed of the main and the tail rotors, while the outer one is governed by two PID type controllers which are intended to stabilize the plant around a wide range of operating points. The ILC algorithm, which is described in Section 4, has been implemented in order to achieve high performance in periodic reference tracking. The responses of the real device and the experimental results are illustrated in Section 5. Finally, conclusions are drawn in Section 6.

2. The Twin Rotor

The Twin-Rotor MIMO System (TRMS), shown in Figure 1, is a laboratory setup developed by Feedback Instruments Ltd. [25, 26] for control experiments. It is a mechanical system formed by a beam that can rotate freely in both the vertical and the horizontal planes by means of two joints placed on its centre. The beam is driven by two perpendicular propellers located at its ends and actuated by DC motors. The body of the Twin Rotor is completed by a counterbalance beam perpendicularly fixed on the main beam's centre. The larger

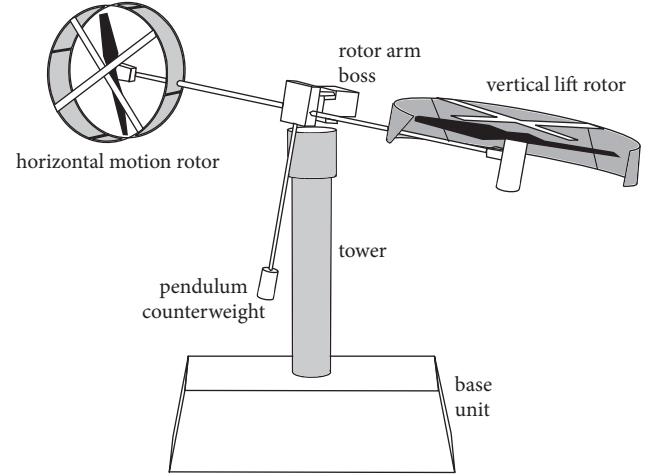


FIGURE 1: Twin-Rotor MIMO System [25].

propeller with vertical axis is called the main rotor, and the half of the beam to which it is attached is called the main beam. The second propeller and its corresponding half of the beam are called the tail rotor and the tail beam, respectively. Although the Twin Rotor has a similar structure to that of a helicopter, the pitch angles of its rotating blades are fixed. Thus, the aerodynamic forces are not controlled by changing the orientation of the blades, as it is usual in conventional helicopters, but by changing the rotational speed of the rotors. The system inputs are then the tension values provided to the DC motors.

To describe the dynamics of the Twin Rotor, the following state vector has been chosen:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} q_v \\ q_h \\ \theta_t \\ \theta_m \end{bmatrix}, \quad (1)$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{q}_v \\ \dot{q}_h \\ \omega_t \\ \omega_m \end{bmatrix},$$

where q_v denotes the pitch angular position, q_h the yaw angular position, θ_t the angular position of the tail rotor, and θ_m the angular position of the main rotor. Together with these state variables, a set of coordinate systems has been defined in order to describe the dynamical laws of the device, as shown in Figure 2.

Since the Twin Rotor is fundamentally a mechanical system, its equations of motion can be obtained directly through an energetic formulation based on the Euler-Lagrange equations [12, 13, 16]. For that, it is necessary to calculate the kinetic energy $T(\mathbf{x}, \dot{\mathbf{x}})$ and the potential energy $U(\mathbf{x})$ of its moving parts.

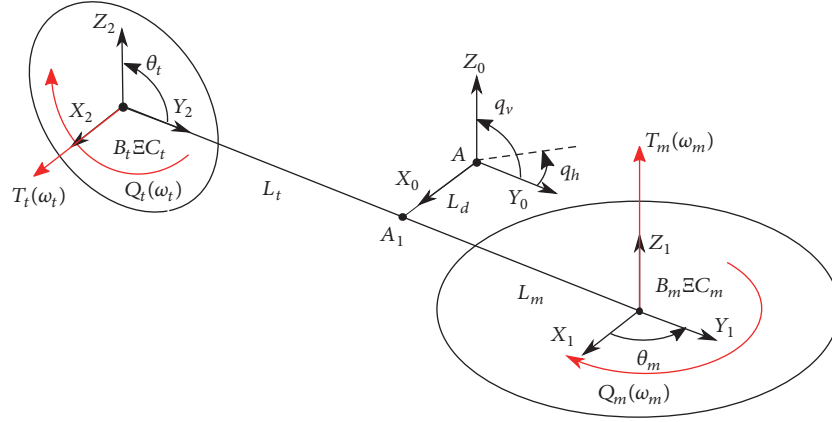


FIGURE 2: State variables and coordinate systems.

From the kinetic and the potential energies, Lagrangian $L(\mathbf{x}, \dot{\mathbf{x}}, t) = T(\mathbf{x}, \dot{\mathbf{x}}, t) - U(\mathbf{x})$ is directly obtained and finally the equations of motion can be derived:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) - \frac{\partial L}{\partial \mathbf{x}} = \mathbf{F}, \quad (2)$$

where \mathbf{F} denotes the external forces.

It can be proved that, expanding these expressions, the equations of movement take the final form:

$$\mathbf{M}(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{c}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{g}(\mathbf{x}) = \mathbf{F}, \quad (3)$$

being $\mathbf{M}(\mathbf{x}) \in \mathbb{R}^{4 \times 4}$ the inertia tensor, $\mathbf{c}(\mathbf{x}, \dot{\mathbf{x}}) \in \mathbb{R}^4$ the term related to the Coriolis forces, and $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^4$ the term associated with the potential energies (gravity).

This equation can be decomposed in its components:

$$\begin{bmatrix} \mathbf{M}_q(\mathbf{q}) & \mathbf{M}_{q,\theta}(\mathbf{q}) \\ \mathbf{M}_{q,\theta}(\mathbf{q})^T & \mathbf{M}_\theta \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_q(\mathbf{q}, \dot{\mathbf{q}}, \omega) \\ \mathbf{c}_\theta(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{g}_q(\mathbf{q}) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_q \\ \mathbf{F}_\omega \end{bmatrix}. \quad (4)$$

Note that the $\mathbf{M}_{q,\theta}(\mathbf{q})$ terms take the following form:

$$\begin{bmatrix} J_{tr} & 0 \\ 0 & J_{mr} \cos q_v \end{bmatrix}, \quad (5)$$

where J_{tr} and J_{mr} are the tail and main rotor masses. In this type of systems, these two terms are considerably smaller than the masses of the other elements. Consequently, the simplification $\mathbf{M}_{q,\theta}(\mathbf{q}) \approx \mathbf{0}$ is usually assumed, which leads to a resultant decoupled system formed of two subsystems. On the one hand, we have a simplified model which describes the dynamics of the mechanical system:

$$\mathbf{M}_q(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{c}_q(\mathbf{q}, \dot{\mathbf{q}}, \omega) + \mathbf{g}_q(\mathbf{q}) = \mathbf{F}_q \quad (6)$$

and, on the other hand, we have the model which describes the rotational dynamics of the propellers:

$$\mathbf{M}_\theta \ddot{\theta} + \mathbf{c}_\theta(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{F}_\omega. \quad (7)$$

In the first case (see (6)), the external forces \mathbf{F}_q can be divided into two terms. The first one is related to friction in the joints (a linear model is assumed) and the second one is referred to the resultant propulsion forces exerted by the propellers:

$$\mathbf{F}_q = -\mathbf{B}_q \dot{\mathbf{q}} + \gamma_q(\mathbf{q}, \omega). \quad (8)$$

In this work, \mathbf{B}_q and γ_q have been experimentally determined.

In the second case (see (7)), the external forces \mathbf{F}_ω can be also decomposed in two parts. The first one is due to friction between the air and the blades of the propellers, while the second one is the torque exerted by the corresponding DC motors:

$$\mathbf{F}_\omega = -\mathbf{B}_\omega \omega + \phi_\omega(\mathbf{v}), \quad (9)$$

where \mathbf{v} denotes the voltage applied to the propellers' motors. As before, \mathbf{B}_ω and ϕ_ω have been experimentally determined.

3. Baseline Controller

The control system that is proposed in this paper is based on a strategy known as Current-Iteration Iterative Learning Control, which is described in [5] as a method for incorporating a baseline feedback controller with ILC algorithms.

Figure 3 shows the complete control scheme that has been developed. Note that all of its branches correspond to two-dimensional signals. Subscripts j are used to indicate the iteration number (these indicators are needed because learning is achieved by repeating the same task multiple times). Thus, in an iteration j , $\mathbf{q}^{\text{des}} = (q_v^{\text{des}}, q_h^{\text{des}})$ is the reference trajectory, $\mathbf{u}_j = (u_{v,j}, u_{h,j})$ are the signals provided by the ILC algorithm, $\mathbf{e}_j = (e_{v,j}, e_{h,j})$ are the pitch and yaw position errors, $\Gamma_j^{\text{des}} = (\Gamma_{v,j}^{\text{des}}, \Gamma_{h,j}^{\text{des}})$ are the desired torques in the TRMS's articulations, $\omega_j^{\text{des}} = (\omega_{m,j}^{\text{des}}, \omega_{t,j}^{\text{des}})$ are the desired rotational speed of the rotors, $\mathbf{V}_j = (V_{m,j}, V_{t,j})$ are the voltage levels applied to the DC motors, and $\mathbf{q}_j = (q_{v,j}, q_{h,j})$ and $\omega_j = (\omega_{m,j}, \omega_{t,j})$ are, respectively, the measured angular positions of the TRMS body and the measured rotors' speed.

As shown in Figure 3, the control system can be divided into two main parts: the baseline feedback controllers and

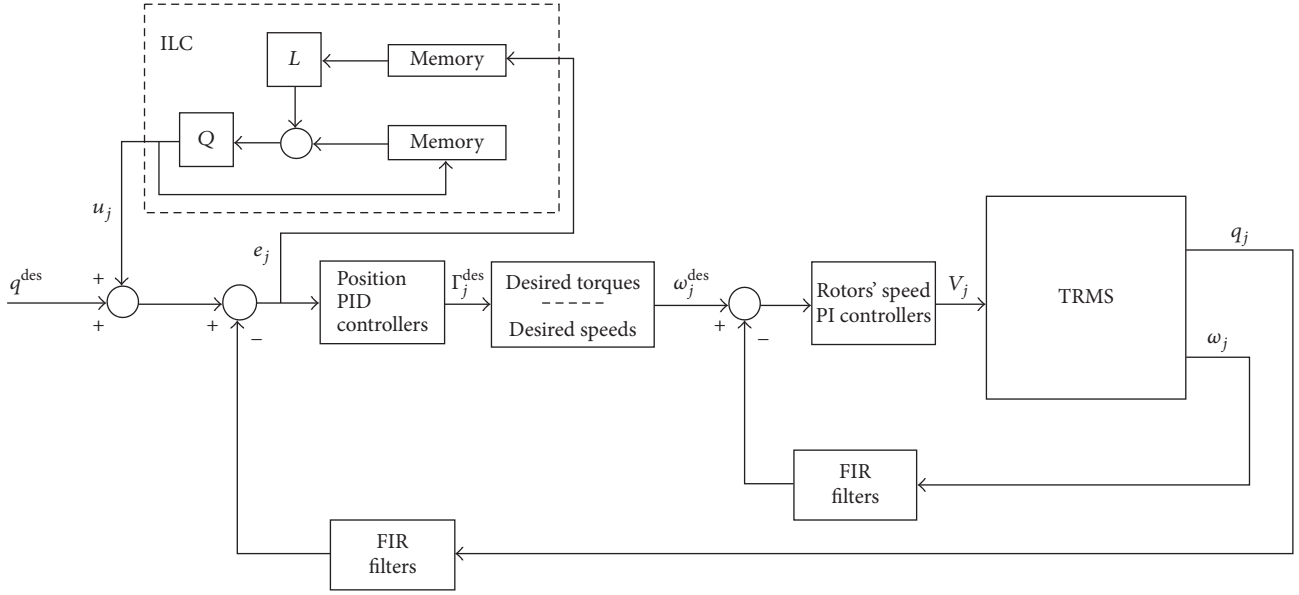


FIGURE 3: General control scheme.

the learning algorithm. This section focuses on the baseline control architecture, which its main goal is to assure a stable behaviour of the plant in a wide range of operating points. Issues related to the ILC algorithm will be detailed in Section 4.

In this work, the baseline control scheme is composed by two cascade feedback loops. Similar hierarchical control structures have been proposed in some reference papers [12, 14, 17]. As a starting point, it is important to distinguish between the external and the inner loops (see Figure 3).

The external control loop, namely, the position control loop, is governed by two linear PID type controllers that have been designed to stabilize the plant around the (0,0) operating point. The stability in other point has been tested through a robust stability test (assuming the plant variation in different equilibrium point as uncertainty). These controllers provide the reference torques that have to be applied on the pitch and yaw angles to reach the desired position.

The inner loop, namely, the rotors' speed control loop, is governed by two linear PI type controllers that adjust the rotational speed of the rotors to reach the desired position. It is preceded by a computational block that calculates the desired rotors' speed from the reference torques provided by the position controllers, that is, function $\omega_i^{des}(\Gamma_i^{des})$, which is shown in Table 3.

3.1. Rotors' Speed PI Controllers Design. As it has been already introduced, the goal of the rotor's speed controllers is to adjust the rotational speed of the propellers to a desired velocity in a fast and proper way.

To apply classical controller design methods, the DC motors dynamics have been modelled as first-order systems plus nonlinear static characteristics which establish

a relationship between the control inputs V_i and the rotating speed ω_i of the propellers:

$$W_i(s) = \frac{K_i}{1 + \tau_i \cdot s} \cdot W_{rp,i}(V_i(s)), \quad (10)$$

where K_i is the gain, τ_i is the time constant of the system, $W_i(s)$ and $W_{rp,i}(V_i(s))$ are the Laplace transforms of ω_i and $\omega_{rp,i}(V_i)$, respectively. These parameters have been experimentally determined together with the $\omega_{rp,i}(V_i)$ static characteristic for each DC motor.

To control the DC motors by means of reference velocity signals, the inverse static characteristics $V_i(\omega_{rp,i})$ can be used as shown in Figure 4 so that the behaviour of the resulting system is that of a linear first-order system:

$$\begin{aligned} W_i(s) &= \frac{K_i}{1 + \tau_i \cdot s} \cdot W_{rp,i}(V_i(W_i^d(s))) \\ &\approx \frac{K_i}{1 + \tau_i \cdot s} W_i^d(s). \end{aligned} \quad (11)$$

Therefore, linear controllers can be designed using classical control strategies. For the purposes of this work, two linear PI controllers have been implemented to guarantee a fast and nonoscillatory response of the DC motors. The gains of these controllers have been tuned to stabilize the response in less than 2 seconds. Table 1 shows the PI controller values.

It is important to point out that the rotors' speed is measured by encoders. Encoders quantification generates similar effect to that of a noise into the control system. What is more, the noise is amplified as the rotational speed increases (see Figures 5 and 6). This phenomenon has had to be taken into account in the design of the control architecture, because the ILC algorithms are highly sensitive to noise. So,

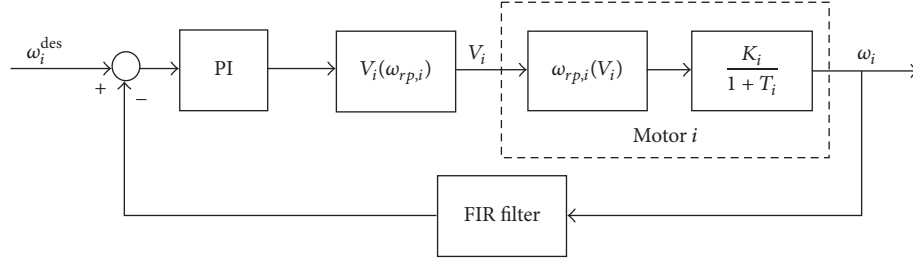


FIGURE 4: Rotors' speed control loop.

TABLE 1: Speed controller values.

Motor	Main	Tail
K_p	2.427	1.600
K_i	2.445	3.682
$V_i(\omega_{rp,i})$	$7.8762 \cdot 10^{-13} \omega_m^7 -$ $3.7020 \cdot 10^{-12} \omega_m^6 -$ $5.0381 \cdot 10^{-9} \omega_m^5 +$ $9.3735 \cdot 10^{-9} \omega_m^4 +$ $1.6150 \cdot 10^{-5} \omega_m^3 -$ $7.1247 \cdot 10^{-6} \omega_m^2 +$ $0.0084 \omega_m - 0.0293$	$5.4367 \cdot 10^{-13} \omega_t^7 +$ $3.8501 \cdot 10^{-12} \omega_t^6 -$ $4.2122 \cdot 10^{-9} \omega_t^5 - 1.6967 \cdot$ $10^{-8} \omega_t^4 + 1.3784 \cdot$ $10^{-5} \omega_t^3 + 7.4374 \cdot 10^{-5} \omega_t^2 +$ $0.0241 \omega_t + 0.0580$

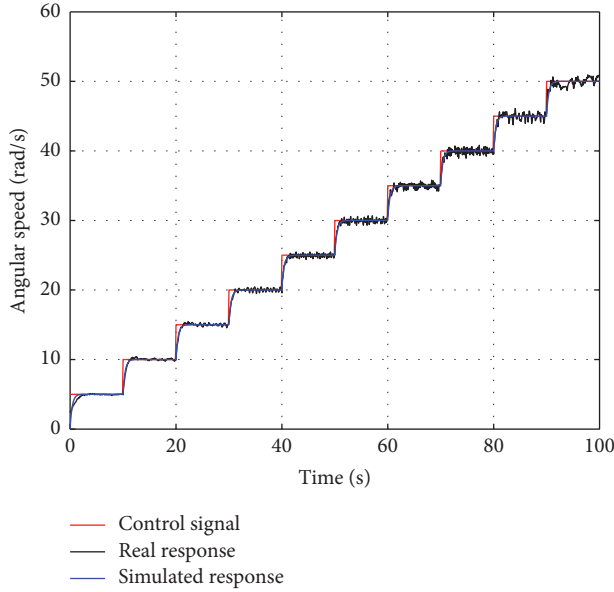


FIGURE 5: Simulated and real response of the main rotor's control loop.

to minimize the noise introduced by the encoders, two FIR filters with a bandwidth of 250 Hz have been placed in the feedback loops as shown in Figure 4.

Figures 5 and 6 show the performance of the main and tail rotors' speed control loops for step signals after all its parameters have been tuned.

3.2. PID Position Controllers Design. As described in Section 2, the Twin Rotor has nonlinear dynamic equations. To

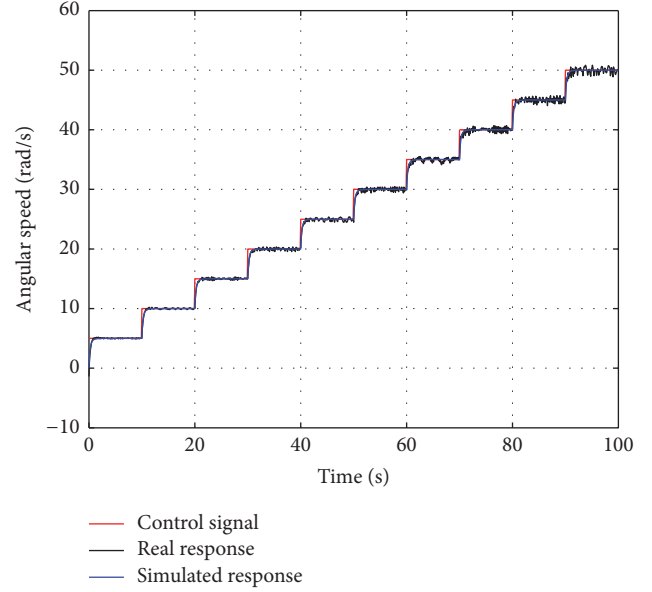


FIGURE 6: Simulated and real response of the tail rotor's control loop.

design linear PID controllers on the system, these equations have been linearised in the state-space formulation around the (0, 0) point. Designing linear controllers using this strategy is justified for controlling nonlinear systems that work around an operation point. In this case, the main goal is not to achieve very good performance with these controllers but stabilize the system in a wide enough workspace.

From (6) and (8), the nonlinear model results are as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} [\boldsymbol{\gamma} - \mathbf{B}\dot{\mathbf{q}} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] \end{bmatrix}, \quad (12)$$

$$\mathbf{y} = \mathbf{q},$$

where

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{q,11} & M_{q,12}(q_v) \\ M_{q,12}(q_v) & M_{q,22}(q_v) \end{bmatrix},$$

$$\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{q,1}(q_v) \dot{q}_h^2 \\ c_{q,12}(q_v) \dot{q}_v \dot{q}_h + c_{q,2}(q_v) \dot{q}_v^2 \end{bmatrix},$$

TABLE 2: Speed controller values.

Term	Value	Units
$M_{q,11}$	0.0349	$\text{kg}\cdot\text{m}^2$
$M_{q,12}(q_v)$	$-2.8218 \cdot 10^{-4} \sin(q_v) + 5.0500 \cdot 10^{-4} \cos(q_v)$	$\text{kg}\cdot\text{m}^2$
$M_{q,22}(q_v)$	$0.0233 \cos^2 q_v + 0.0089$	$\text{kg}\cdot\text{m}^2$
$c_{q,1}(q_v)$	$0.0233 \cos(q_v) \sin(q_v)$	$\text{kg}\cdot\text{m}^2$
$c_{q,12}(q_v)$	$-0.0466 \cos(q_v) \sin(q_v)$	$\text{kg}\cdot\text{m}^2$
$c_{q,2}(q_v)$	$-2.8218 \cdot 10^{-4} \sin(q_v) - 5.0500 \cdot 10^{-4} \cos(q_v)$	$\text{kg}\cdot\text{m}^2$
$g_{q,1}(q_v)$	$0.1982 \sin(q_v) + 0.1547 \cos(q_v)$	$\text{N}\cdot\text{m}$
$g_{q,2}(q_h)$	$0.0075(q_h - 1.455)$	$\text{N}\cdot\text{m}$
$B_{q,11}$	0.0016	
$B_{q,22}$	0.009	

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g_{q,1}(q_v) \\ g_{q,2}(q_h) \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} B_{q,11} & 0 \\ 0 & B_{q,22} \end{bmatrix}, \quad (13)$$

with $\mathbf{x} = [q_v \ q_h \ \dot{q}_v \ \dot{q}_h]^T$ and $\mathbf{q} = [q_v \ q_h]^T$. Numerical values of these matrices are shown in Table 2. In this manner, the linearised system around the equilibrium point $\mathbf{q} = (0, 0)$ is obtained:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -5.6825 & 0.0034 & -0.0459 & 0.0041 \\ 0.0892 & -0.2333 & 0.0007 & -0.2799 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 28.6760 & -0.4503 \\ -0.4503 & 31.1042 \end{bmatrix} \mathbf{\Gamma}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}. \quad (14)$$

To design the position control loop, control actions in the pitch and yaw angles have been considered separately. Thus, two decoupled PID controllers have been implemented: the first one adjusts the desired torque in the pitch angle according to the pitch position error (design based on transfer function from first input to first output of the linearised plant model), whereas the second one sets the desired torque in the yaw angle depending on the yaw position error (design based on transfer function from second input to second output of the linearised plant model). See PID values on Table 3.

Once the desired torque is obtained, the propulsion force is calculated as follows:

$$\begin{bmatrix} T_m^{\text{des}} \\ T_t^{\text{des}} \end{bmatrix} = \begin{bmatrix} L_m & k_t \\ -L_d \sin q_v - k_m \cos q_v & L_t \cos q_v \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_m^{\text{des}} \\ \Gamma_t^{\text{des}} \end{bmatrix}, \quad (15)$$

where $L_m = 0.24 \text{ m}$, $L_t = 0.25 \text{ m}$, and $L_d = 0.025 \text{ m}$ are the distances shown in Figure 2, $k_m = 0.0235$ and $k_t = 0.02$ are the constants that relate the reactive torques of propellers, Q_i , and the propulsion forces T_i , which are obtained experimentally. Finally, to obtain the desired speeds of the motors, the functions that relate the desired propulsion forces and rotor speeds have been derived experimentally and are presented in Table 3.

As with the rotors' speed, the pitch and yaw angles are measured by encoders which add noise into the system. To avoid problems with the control signals, introducing two FIR filters with a bandwidth of 250 Hz in the position feedback loops is needed (see Figure 3).

Figure 7 shows the step response of the baseline control architecture after it has been implemented on the real device. Additional experiments have demonstrated that the PID controllers are capable of stabilizing the TRMS body in a wide set of positions around the $(0, 0)$ point. However, to deal with nonlinearities and avoid abrupt oscillations during the transient period, they have been tuned to make the system's response quite slow (note that the plant's desired position is reached after approximately 30 seconds). Consequently, only with these controllers, good reference tracking cannot be achieved. Table 3 shows the PID controller values together with the propulsion forces-speed function.

4. ILC Algorithm

Once a stable behaviour is guaranteed, the ILC algorithm can be implemented to achieve higher performance in reference tracking. This section explains the basic concepts of the ILC laws and describes the structure of the learning algorithm that has been implemented together with the baseline control architecture.

As an introduction to the formulation of the ILC problem, consider a discrete-time, linear time-invariant SISO system:

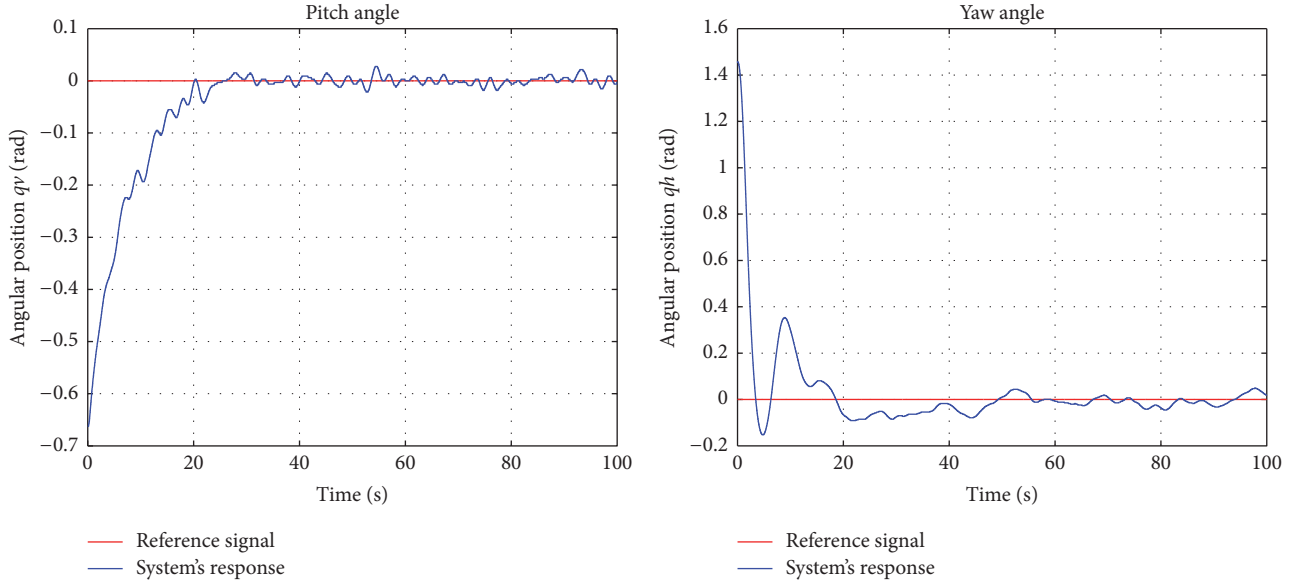
$$y_j(k) = P(q)u_j(k) + d(k), \quad (16)$$

where k is the time index, j is the iteration index, q is the forward time-shift operator $qx(k) = x(k+1)$, y_j is the output, u_j is the control signal, and d represents a disturbance that repeats each iteration. The plant $P(q)$ has a relative degree of m and is assumed to be asymptotically stable (or stabilized by a feedback controller, as in this case). In addition, initial conditions are assumed to be the same on each trial.

The performance will be measured through the error signal defined by $e_j(k) = y_d(k) - y_j(k)$, being $y_d(k)$ the desired system output. Thus, the objective of the ILC is to infer a control signal $u^*(k)$, $k \in [0, N-1]$, by evaluating the

TABLE 3: Position controller values.

Motor	Main	Tail
K_p	0.0250	0.0250
K_i	0.0350	0.0030
K_d	0.0250	0.0250
$\omega_i^{\text{des}}(T_i^{\text{des}})$	$-8.7395T_m + 69.0628\sqrt{T_m}, T_m \geq 0$ $-36.7207(-T_m) - 60.9066\sqrt{-T_m}, T_m < 0$	$-5.9652T_t + 96.9668\sqrt{T_t}, T_t \geq 0$ $68,2969(-T_t) - 126,0478\sqrt{-T_t}, T_t < 0$

FIGURE 7: Step response ($\mathbf{q}^{\text{des}} = (0, 0)$) of the position control loop on the TRMS.

error e_j in the interval $k \in [m, N + m - 1]$ and updating the next iteration control signal $u_{j+1}(k)$ accordingly. Note that discrete time is the natural domain for ILC because it explicitly requires the storage of past-iteration data, which is typically sampled.

A widely used ILC learning algorithm [5] is

$$u_{j+1}(k) = Q(q) [u_j(k) + L(q) e_j(k+1)], \quad (17)$$

where the LTI dynamics $Q(q)$ and $L(q)$ are defined as the Q -filter and learning function, respectively.

At this point, it should be noted that the ILC objective is to generate an open-loop signal that inverts the plant's dynamics and is capable of compensating periodic disturbances and getting good reference tracking. Ideally, ILC learns only from repetitive disturbances, but it does not provide any feedback mechanism to react to noise or unanticipated, nonperiodic disturbances [5]. Consequently, a feedback controller combined with an ILC algorithm is used in this work to achieve robustness and higher performance. The complete control structure in Figure 3, as explained in the previous section, shows the simultaneous implementation of both feedback and ILC strategies.

To implement an ILC algorithm in the TRMS control scheme, one of the most popular design techniques has been used: the PD-type learning function, which consists of a proportional and derivative gain on the error. This is probably

the most widely used type of learning function so far. In the particular case of nonlinear systems, like the Twin Rotor, this learning function is well suited because it does not require extensive analysis nor plant modelling to be applied [5]. The integrator is rarely used for learning functions because ILC has a natural integrator action from one trial to the next.

The discrete-time, PD-type learning function applied to a SISO system can be written as follows:

$$u_{j+1}(k) = Q(q) \cdot [u_j(k) + k_p e_j(k+1) + k_d (e_j(k+1) - e_j(k))], \quad (18)$$

where k_p is the proportional gain and k_d is the derivative gain.

In the case of the TRMS, as it is a MIMO system with two input and two output signals, the learning function takes the following form:

$$\begin{aligned} \begin{bmatrix} u_{q_v, j+1}(k) \\ u_{q_h, j+1}(k) \end{bmatrix} &= \begin{bmatrix} Q_{q_v}(q) & 0 \\ 0 & Q_{q_h}(q) \end{bmatrix} \begin{bmatrix} u_{q_v, j}(k) \\ u_{q_h, j}(k) \end{bmatrix} \\ &+ \begin{bmatrix} k_{p, q_v} & 0 \\ 0 & k_{p, q_h} \end{bmatrix} \begin{bmatrix} e_{q_v, j}(k+1) \\ e_{q_h, j}(k+1) \end{bmatrix} \\ &+ \begin{bmatrix} k_{d, q_v} & 0 \\ 0 & k_{d, q_h} \end{bmatrix} \begin{bmatrix} e_{q_v, j}(k+1) - e_{q_v, j}(k) \\ e_{q_h, j}(k+1) - e_{q_h, j}(k) \end{bmatrix}. \end{aligned} \quad (19)$$

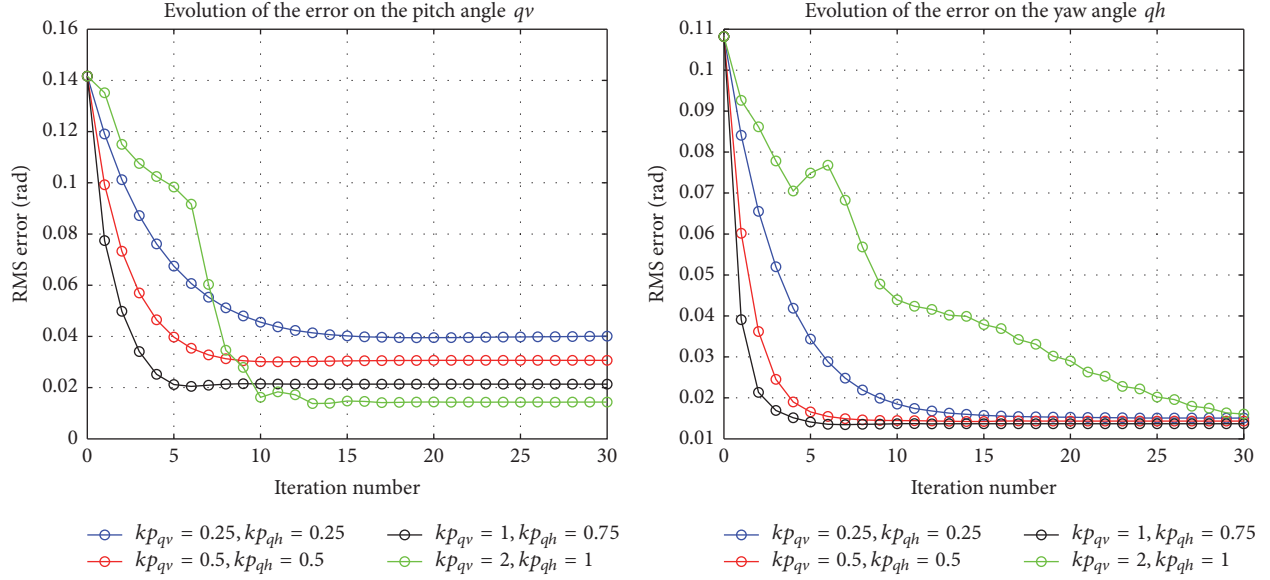


FIGURE 8: Transient behaviour and asymptotic performance for P gains tuning, with $k_{d,qv} = k_{d,qh} = 0$ and Q -filters bandwidths set to 0,1 Hz.

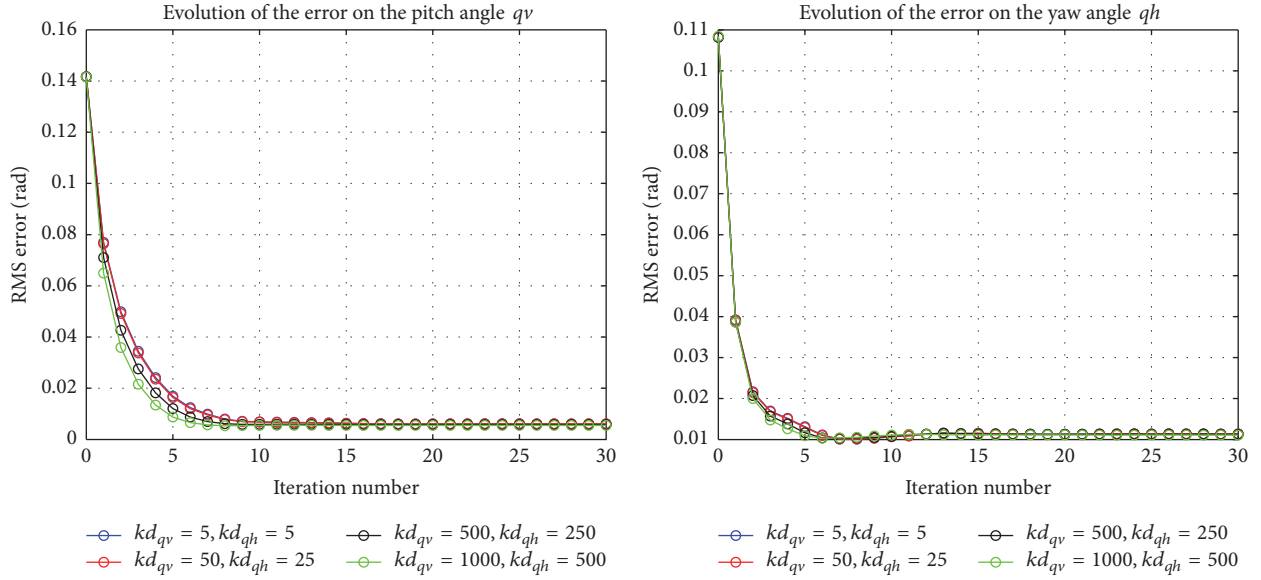


FIGURE 9: Transient behaviour and asymptotic performance for D gains tuning, with $k_{p,qv} = 1$, $k_{p,qh} = 0,75$ and Q_{qv} , Q_{qh} filters bandwidths set to 0,2 Hz and 0,15 Hz, respectively.

Although it is not strictly necessary in this type of learning functions, including a low-pass Q -filter is usually favourable to achieving monotonic convergence because it can be used to disable learning at high frequencies [5]. The Q -filter also has the benefits of added robustness and high-frequency noise filtering.

The gains of the PD-type learning function have been tuned with the aim of achieving both good learning transients and low error. For each set of gains k_p and k_d , the learning has been reset and run in simulation for sufficient iterations to determine the transient behaviour and asymptotic error. Initially, the learning gains and filter bandwidth have been

set low. After a stable baseline transient behaviour and error performance have been obtained, the gains and the bandwidth have been increased. As shown in Figures 8, 9, and 10, experiments have demonstrated that the learning gains influence the rate of convergence, whereas the Q -filter influences the converged error performance.

More specifically, plots in Figures 8 and 9 show that increasing the proportional and derivative learning gains makes the error converge in fewer iterations. However, if the gains are set too high, the learning transients do not converge monotonically and can lead to problems of stability.

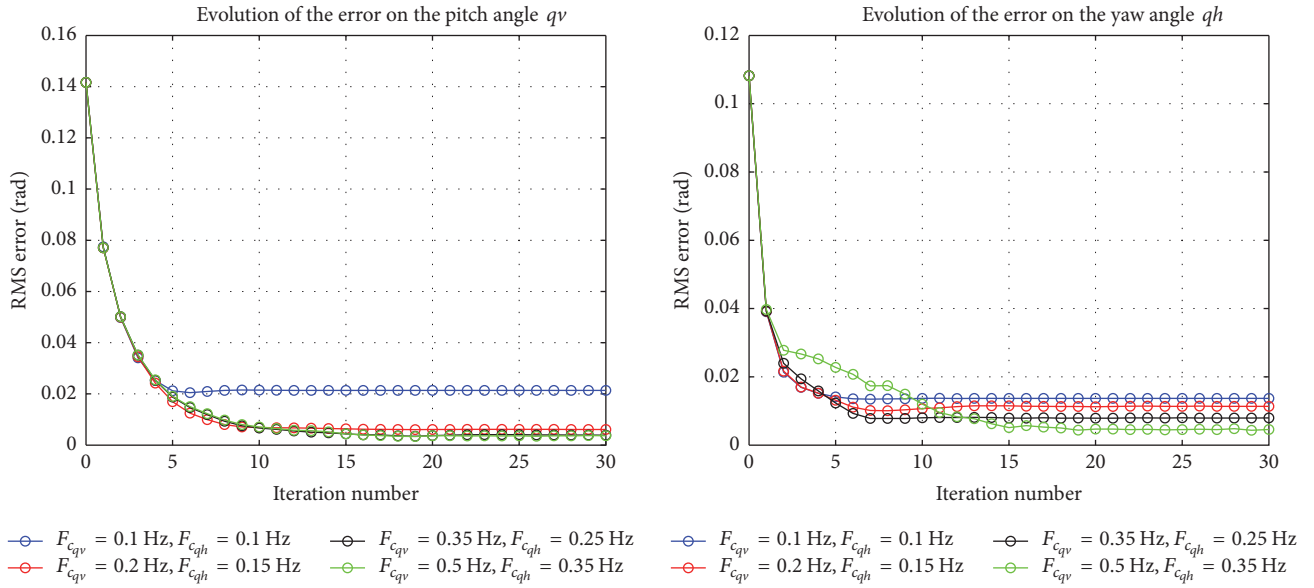


FIGURE 10: Transient behaviour and asymptotic performance for Q-filters bandwidths tuning (where F_c is the cutoff frequency), with the following PD learning gains: $k_{p,qv} = 1$, $k_{p,qh} = 0.75$, and $k_{d,qv} = k_{d,qh} = 0$.

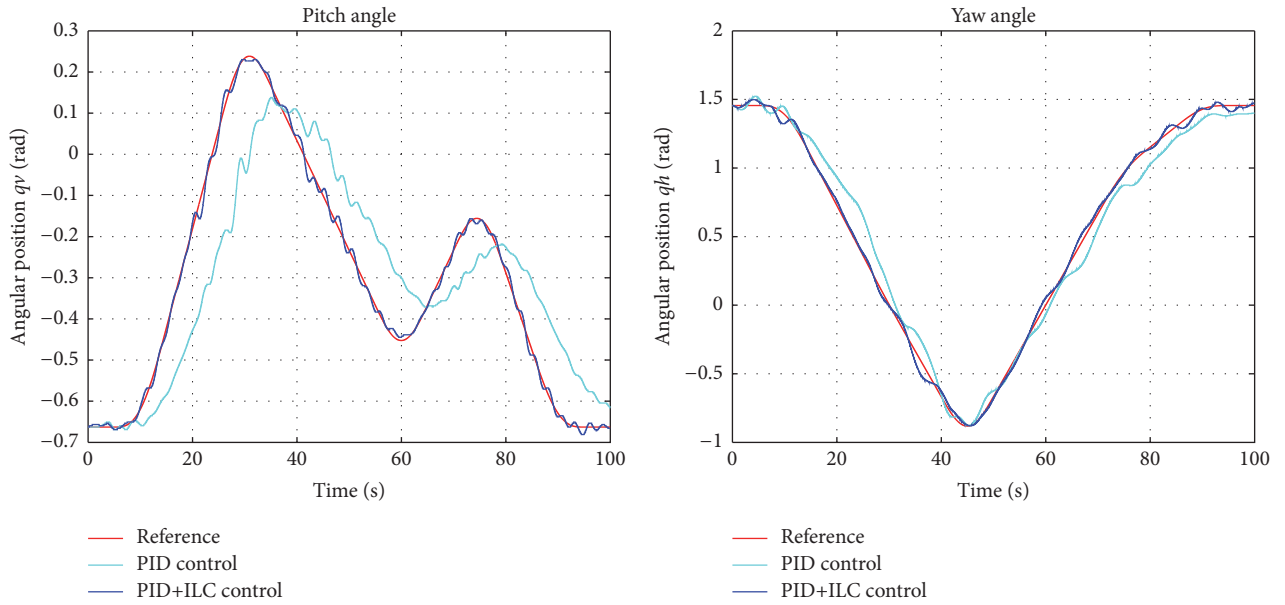


FIGURE 11: Tracking results with and without ILC on the TRMS.

In addition, plots in Figure 10 show that the Q-filter has little effect on the convergence rate of the error but instead primarily determines the magnitude of the converged error and stability of the system. Although increased bandwidth improves the error, too high bandwidth will result in transients that do not converge monotonically.

In the field of ILC, a practical stability condition is to achieve good learning transient behaviour, which is defined as monotonic convergence [5]. Furthermore, when robustness is considered, model uncertainty and noise lead to limitations on the performance of LTI learning algorithms. So, as the real device is affected by noise and high nonlinearities,

the learning gains and the Q-filter bandwidths have been selected in a conservative way to guarantee the stability of the complete system.

5. Experimental Results

In this section, the results from the experimental application of the ILC algorithm to the TRMS are exposed.

To test the ILC law, a fifth-order polynomial trajectory has been set as the tracking reference. Figure 11 shows a comparison between the followed trajectory without ILC (first trial) and after 20 learning iterations. Note that the

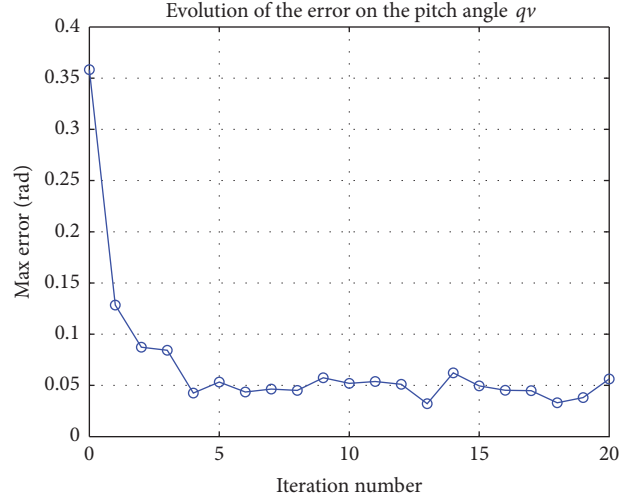
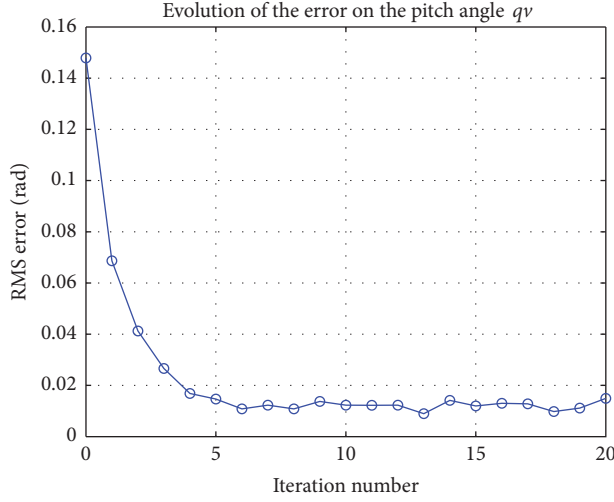


FIGURE 12: Transient behaviour and asymptotic performance on pitch angle of the TRMS.

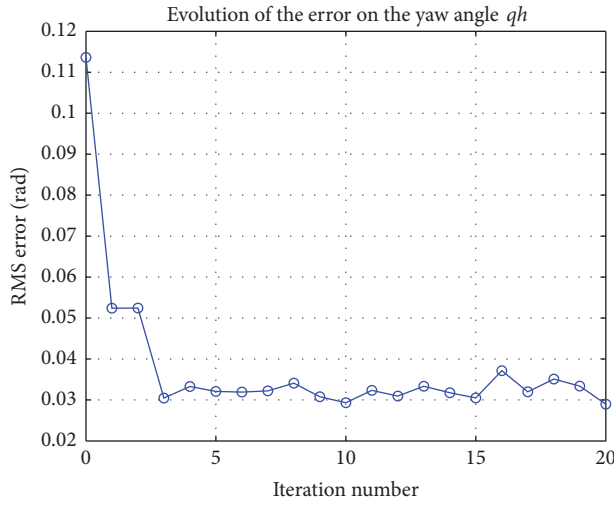


FIGURE 13: Transient behaviour and asymptotic performance on the yaw angle of the TRMS.

feedback controllers show a lagged response, whereas the ILC algorithm increases considerably the system's performance by achieving nearly perfect tracking.

The error on each axis is calculated as the difference between the reference and obtained angle at given instant k and iteration j , resulting in

$$\begin{aligned} e_{v,j}(k) &= q_{v,\text{des}}(k) - q_v, j(k), \\ e_{h,j}(k) &= q_{h,\text{des}}(k) - q_h, j(k). \end{aligned} \quad (20)$$

After that, the root mean square (RMS) and maximum error are calculated for each axis $i = \{v, h\}$ and each iteration j as follows:

$$\begin{aligned} E_{i,j}^{\text{RMS}} &= \sqrt{\frac{1}{N} \sum_{k=1}^N e_{i,j}^2(k)}, \\ E_{i,j}^{\text{max}} &= \max_k (e_{i,j}(k)). \end{aligned} \quad (21)$$

Thus, Figures 12 and 13 show the transient learning behaviour by capturing the RMS and the maximum tracking errors on each iteration in the pitch and yaw angles, respectively. Note that initial errors decrease rapidly during the first iterations, so that the learning process converges in approximately six trials. In addition, monotonic convergence is achieved, which is fundamental to assure the stability of the ILC law.

From these results, the ability of the ILC algorithm to reduce tracking errors has been analysed. For that, the following expression,

$$\% \Delta_p^{\text{error}} = \frac{E_{i,0}^p - (1/15) \sum_{j=6}^{20} E_{i,j}^p}{E_{i,0}^p} \cdot 100, \quad (22)$$

$$p = \{\text{RMS}, \text{max}\},$$

has been used to calculate the percentage difference between the initial error and the mean of the errors after learning has converged.

Note that the $E_{i,j}$ variables in expression (22) denote the RMS error or the maximum absolute error, depending on the case, in the corresponding axis with $i = \{v, h\}$ and in the j iteration. It is important to distinguish them from e_j variables used in this paper to denote the tracking error vector on each trial.

After applying (22) to experimental results, it follows that, with the ILC algorithm, RMS errors are reduced by 91,83% in the pitch angle and a 71,62% in the yaw angle. Similarly, maximum absolute errors are reduced by 86,78% in the pitch angle and 68,19% in the yaw angle.

6. Conclusions

This paper has presented the implementation of an ILC algorithm to achieve high performance in trajectory tracking with a Twin Rotor, a mechanical system characterized by its nonlinear and cross-coupled dynamics. As any control problem, the initial phase has been focused on obtaining an accurate model of the system's behaviour. For that, a complete simulator of the TRMS's dynamics has been developed using the Lagrangian formulation and adjusting its parameters experimentally. This model has been used to design a feedback control system with the aim of stabilizing the TRMS's body in different operation points. Finally, a PD-type learning function based on the principles of the ILC has been tuned and implemented together with the feedback controllers. The efficiency of the complete control system has been tested on both the simulation model and the real device and has demonstrated the capability of ILC algorithms to improve the performance of systems that operate in a repetitive way.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

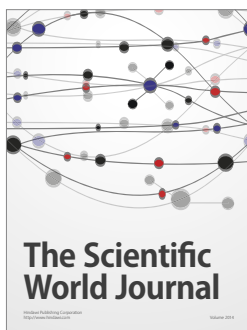
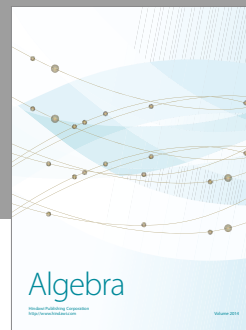
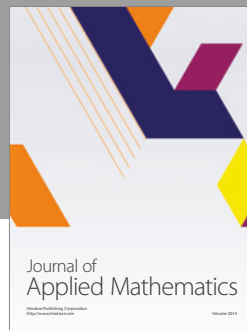
Acknowledgments

This work has been supported by Project DPI2015-69286-C3-2-R of Ministerio de Educación de España and 2014 SGR 267 of the AGAUR agency of the Generalitat de Catalunya.

References

- [1] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.
- [2] D. H. Owens and S. Daley, "Iterative learning control—monotonicity and optimization," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 3, pp. 279–293, 2008.
- [3] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, Lecture Notes in Control and Information Sciences, Springer, New York, NY, USA, 1st edition, 2003.
- [4] H. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.
- [5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: a learning-based method for high-performance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [6] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.
- [7] W. Zuo and L. Cai, "A new iterative learning controller using variable structure fourier neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 2, pp. 458–468, 2010.
- [8] H. A. Hashim and M. A. Abido, "Fuzzy controller design using evolutionary techniques for twin rotor MIMO system: a comparative study," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 704301, 11 pages, 2015.
- [9] C. Yin, J.-X. Xu, and Z. Hou, "A high-order internal model based iterative learning control scheme for nonlinear systems with time-iteration-varying parameters," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 55, no. 11, pp. 2665–2670, 2010.
- [10] J.-X. Xu and Y. Tan, "New iterative learning control approaches for nonlinear non-affine MIMO dynamic systems," in *Proceedings of American Control Conference*, vol. 2, pp. 896–901, Arlington, VA, USA, June 2001.
- [11] P. Mullhaupt, B. Srinivasan, J. Lévine, and D. Bonvin, "A toy more difficult to control than the real thing," in *Proceedings of the 4th European Control Conference (ECC '97)*, pp. 1652–1657, July 1997.
- [12] M. López-Martínez, M. G. Ortega, C. Vivas, and F. R. Rubio, "Control no lineal robusto de una maqueta de helicóptero con rotores de velocidad variable," *Revista Iberoamericana de Automática e Informática Industrial IRIAI*, vol. 4, no. 3, pp. 46–60, 2007.
- [13] A. Rahideh and M. H. Shaheed, "Mathematical dynamic modelling of a twin-rotor multiple input-multiple output system," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 1, pp. 89–101, 2007.
- [14] P. Wen and T.-W. Lu, "Decoupling control of a twin rotor MIMO system using robust deadbeat control technique," *IET Control Theory & Applications*, vol. 2, no. 11, pp. 999–1007, 2008.
- [15] M. Belkheiri, A. Rabhi, F. Boudjema, A. Elhajjaji, and J. Bosche, "Model parameter identification and nonlinear control of a twin rotor MIMO system—TRMS," in *Proceedings of the 15th IFAC Symposium on System Identification (SYSID '09)*, pp. 1487–1492, July 2009.
- [16] A. Tastemirov, A. Lecchini-Visintini, and R. M. Morales-Viviescas, "Complete dynamic model of the twin rotor MIMO System (TRMS) with experimental validation," in *Proceedings of the 39th European Rotorcraft Forum*, vol. 66, pp. 89–98, Moscow, Russia, 2013.
- [17] M. López-Martínez, *Control no lineal de sistemas con dos rotores independientes [Ph.D. thesis]*, Universidad de Sevilla, 2005.
- [18] A. Rahideh and M. H. Shaheed, "Hybrid fuzzy-PID-based control of a twin rotor MIMO system," in *Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics (IECON '06)*, pp. 48–53, November 2006.
- [19] D. Dutescu, M. Radac, and R. Precup, "Model predictive control of a nonlinear laboratory twin rotor aero-dynamical system," in *Proceedings of the 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, pp. 000037–000042, Herl'any, Slovakia, January 2017.

- [20] P. U. Shaikh and S. Karvekar, "Implementation of sliding mode controller for twin rotor multi input multi output system without state observer," in *Proceedings of the 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES '16)*, 6, 1 pages, July 2016.
- [21] S. Mondal and C. Mahanta, "Adaptive second-order sliding mode controller for a twin rotor multi-input-multi-output system," *IET Control Theory & Applications*, vol. 6, no. 14, pp. 2157–2167, 2012.
- [22] J. Sharma and B. Pratap, "Nonlinear robust dynamic surface controller design for twin rotor control system," in *Proceedings of the 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES '16)*, pp. 1–6, July 2016.
- [23] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33, no. 1-2, pp. 103–127, 2012.
- [24] R. Moliner and R. Tanda, "Tool for robust tuning of PI/PID controllers with two degree of freedom," *RIAI: Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 13, no. 1, pp. 22–31, 2016.
- [25] "Feedback Instruments Limited," Twin Rotor MIMO System Getting Started, 33-007-1M5.
- [26] "Feedback Instruments Limited," Twin Rotor MIMO System Control Experiments, 33-949S.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

