

Received October 2, 2021, accepted October 25, 2021, date of publication October 27, 2021, date of current version November 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123668

An Innovative MIMO Iterative Learning Control Approach for the Position Control of a Hydraulic Press

IGNACIO TROJAOLA¹, IKER ELORZA¹, ELOY IRIGOYEN², (Senior Member, IEEE), ARON PUJANA-ARRESE¹, AND GORKA SORROSAL¹

¹Ikerlan Technology Research Centre, Arrasate-Mondragón, 20500 Gipuzkoa, Spain

²Systems Engineering and Automation Department, University of the Basque Country (UPV/EHU), 48940 Bizkaia, Spain

Corresponding author: Ignacio Trojaola (itrojaola@ikerlan.es)

This work was supported in part by the Department of Development and Infrastructures of the Government of the Basque Country via Industrial Doctorate Program BIKAINTEK under Grant 20-AF-W2-2018-00015.

ABSTRACT To improve the performance of hydraulic press position control and eliminate the need to manually define control signals, this paper proposes a multi-input-multi-output (MIMO) Iterative Learning Control (ILC) algorithm. The MIMO ILC algorithm design is based on the inversion of the known low frequency dynamics of the hydraulic press, whereas the unknown and uncertain high frequency dynamics are discarded due to their low influence in the learning transient. Moreover, for the MIMO ILC convergence condition, a graphical method is proposed, in which the ILC learning filter eigenvalues are analyzed. This method allows studying the stability and convergence rate of the algorithm intuitively. Theoretical analysis and results prove that with the MIMO ILC algorithm the position control is automated and that high precision in the position tracking is gained. A comparison with other model inverse ILC approaches is carried out and it is shown that the proposed MIMO ILC algorithm outperforms the existing algorithms, reducing the number of iterations required to converge while guaranteeing system stability. Furthermore, experimental results in a hydraulic test rig are presented and compared to those obtained with a conventional PI controller.

INDEX TERMS Iterative learning control, position control, MIMO, electro-hydraulics.

I. INTRODUCTION

Hydraulic presses have traditionally been used for high-force applications due to their easy operation and adaptability to suit a wide range of forming conditions. One of the advantages of hydraulic presses is that the force can be applied throughout the stroke, unlike mechanical presses where the force is maximum at the bottom.

During the hydraulic press operation, shown in Fig. 1, position and force control needs to be done to guarantee the correct forming of the workpiece. The position control ensures the correct trajectory tracking during the *Free fall* and *Drawing* phases, whereas the force control maintains a specific force level during the *Making Force* phase.

Nowadays, in the working operation of a hydraulic press the proportional valve control signals are defined manually,

The associate editor coordinating the review of this manuscript and approving it for publication was Wonhee Kim¹.

with a predefined closing ramp, so the position is correctly tracked during the *Free fall* and *Drawing* phases. This is a tedious and time-consuming process that typically includes several days and can involve high costs. Furthermore, the success of such a control depends on the valve pre-defined signals design, which usually depends on the operator's ability and experience.

We seek to automate the rather tedious manual position control, with no need to manually define any signal. Nowadays, the desired falling velocity during the *Free Fall* phase is obtained by manually setting a constant spool position to the auxiliary chamber proportional valve, see Fig. 2. Furthermore, during the *Free Fall* and *Drawing* phases transition, a closing ramp must be defined to this valve which is modified every time the position reference changes.

We propose to control the auxiliary chamber pressure with the variable axial piston pump, while controlling the slide position with the auxiliary chamber proportional valve.

in a cost function. In such a cost function we would have to include a condition for fast convergence, robustness towards uncertainties, long-term stability and satisfy the input constraints. This approach requires a too complex optimization problem.

To reduce the model dependent design, model-free MIMO ILC approaches were proposed in [12], [13], to reduce the modeling requirement. However, these approaches directly affect the convergence rate of the ILC algorithm, and the learning process requires more iterations to converge.

The advantages of using model-based approaches were shown in [14], where an overview of which MIMO ILC design approach to use, illustrated on an industrial printer, is given. It concludes that the centralized, model-based, approach achieves the better performance in terms on convergence, however it requires more designing and computation effort, in comparison to the decentralized, model-free, approach. In the position control problem of a hydraulic press, we can manage the modeling designing effort better than a slow convergence. Investing in an accurate plant model design can optimize the convergence of the algorithm, which results in commissioning cost savings. Else, the slower the convergence, the more iterations will be required to obtain an appropriate workpiece forming that fulfills all the design requisites.

So far, no studies have been done regarding the implementation of a MIMO ILC algorithm in hydraulic systems. Previous works have just focused on the application of SISO ILC algorithms either for the position control or for the force control of hydraulic circuits.

An ILC algorithm for the position control of an electro-hydraulic system is developed in [15]. The ILC algorithm improves the PID controller performance, however no analysis of the robustness and the convergence rate of the algorithm is provided.

In [16] the tuning of the learning gain and the delay time is analyzed. Increasing the learning gain value the convergence rate is improved, however the system response oscillates harshly. Increasing the delay time the tracking error is decreased, but as iterations go on the system becomes unstable.

Most of the ILC studies in hydraulic systems do not consider the convergence rate of the algorithm, the focus has been on the applicability of ILC in hydraulic systems, no matter the number of iterations required to converge.

One should aim to obtain the fastest convergence rate possible to reduce the number of iterations needed. The fewer iterations required, the shorter the commissioning time of the hydraulic press control will be, which translates into a reduction of the production costs.

The convergence rate of the ILC is improved in [17], to track the displacement curve of the hydraulic press slide. A fuzzy ILC is developed, to adapt the ILC gains based on the error and change in the error. The requirement of defining fuzzy strategies manually opposes the objective of this study to automate the control process.

A PD-type ILC is implemented for the trajectory tracking in [18]. The number of iterations required to converge is considerably reduced compared to a conventional proportional type ILC, as it is also shown in [19]. However, no analysis of how to design the PD-type ILC gains is provided. The only procedure followed is that the higher the gain the faster the convergence will be. However, this design method can lead to stability problems as if the gains are set too high the learning transients will not converge.

One can conclude that there exists a gap in the literature regarding the design of MIMO ILC algorithms in hydraulic systems. No analysis of the convergence rate improvement of the MIMO ILC algorithm to optimize the press working operation has been done. Furthermore, this improvement should be enhanced with a stability analysis of the algorithm, as stability is an essential property for industrial practice.

The main contributions of this paper are summarized as follows. A MIMO ILC is proposed for the position control of a hydraulic press, to automate the press operation and eliminate the position control reliance on pre-defined control inputs. With the MIMO ILC controller, we can control simultaneously the pump outflow and the valve spool position, to control the cylinder pressure and piston position, respectively. The MIMO ILC algorithm design is based on the inversion of the low frequency hydraulic press dynamics, discarding the high frequency dynamics. This design method obtains a better convergence rate compared to other model inverse ILC algorithms in the literature. Furthermore, a graphical method is proposed which provides the user the possibility to analyze the convergence rate and the stability of the MIMO ILC algorithm intuitively.

In Section II, the MIMO ILC algorithm is introduced for the position control. The results of the MIMO ILC algorithm are shown in Section III. A comparison of the proposed ILC design with other ILC methods is carried out in Section IV. The proposed MIMO ILC is implemented in a hydraulic test rig in Section V. Lastly, the conclusions are provided in Section VI.

II. MIMO ITERATIVE LEARNING CONTROL

ILC is ideal for a system that performs the same process repeatedly and where uncertainties abound, as it is capable of learning the necessary control action via the repeated process. ILC was first proposed for improving the reference tracking of a system that follows a repeated trajectory by Uchiyama in [20]. It was further extended by Arimoto *et al.* [21], for a mechanical robot operation. The learning control scheme proposed by Arimoto was:

$$U_{j+1}(s) = Q(s)(U_j(s) + L(s)E_j(s)) \quad (1)$$

which is a past-error feed-forward law. $U_j(s)$ is the Laplace transform of the entire input vector to the system $G(s)$ at the j -th learning iteration. The iteration error $E_j(s)$ is given by the difference between the position reference and the iteration system output, $E_j(s) = R(s) - Y_j(s)$. $L(s)$ is the learning

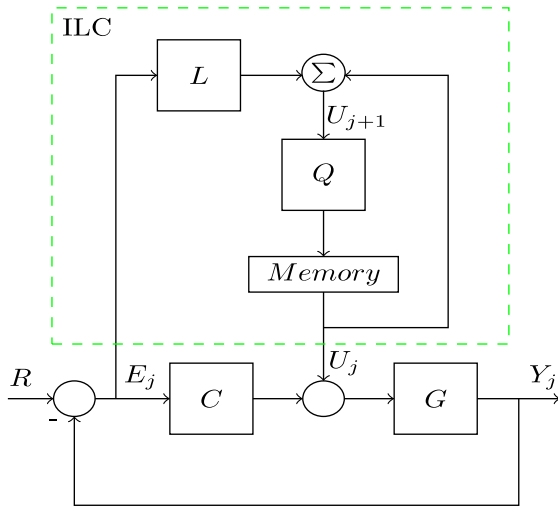


FIGURE 3. ILC parallel arrangement with feedback controller.

filter and $Q(s)$ is a gain that enlarges the stability region of the algorithm.

A. MIMO ILC ALGORITHM ANALYSIS

In the position control loop, the slide hydraulic circuit can be regarded as a general time invariant MIMO linear continuous state-variable equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_s \mathbf{x}(t) + \mathbf{B}_s \mathbf{u}(t); \quad \mathbf{y}(t) = \mathbf{C}_s \mathbf{x}(t) \quad (2)$$

where \mathbf{x} is the four-dimensional state vector, consisting of the piston position and velocity, and the pressure inside the two cylinder chambers. \mathbf{u} is the two-dimensional learning control input vector, consisting of the pump swash angle and the valve spool position. \mathbf{y} is the two-dimensional output vector, consisting of the piston position and the cylinder auxiliary chamber pressure. The Laplace transform of the system equation is: $\mathbf{G}(s) = \mathbf{C}_s (s\mathbf{I} - \mathbf{A}_s)^{-1} \mathbf{B}_s$. Note that $\mathbf{G}(s)$ is a two-by-two transfer matrix.

In [22] the ILC error propagation equation for a SISO system is shown. Transforming it into a MIMO problem results in:

$$\mathbf{E}_{j+1}(s) = (\mathbf{I} - \mathbf{S}(s)\mathbf{G}(s)\mathbf{L}(s))\mathbf{E}_j(s) \quad (3)$$

where $\mathbf{S}(s)$ is the sensitivity transfer matrix $\mathbf{S}(s) = (\mathbf{I} + \mathbf{G}(s)\mathbf{C}(s))^{-1}$, and $\mathbf{L}(s)$ the learning transfer matrix. $\mathbf{C}(s)$ is the controller transfer matrix containing the pump PI controller, $C_P(s)$, and the valve PI controller, $C_V(s)$:

$$\mathbf{C}(s) = \begin{bmatrix} C_P(s) & 0 \\ 0 & C_V(s) \end{bmatrix}. \quad (4)$$

From equation (3), if the term $\mathbf{\Lambda}(s) = \mathbf{I} - \mathbf{S}(s)\mathbf{G}(s)\mathbf{L}(s)$ is less than one, then the error vector at the next iteration will be smaller than the error at the current iteration. Otherwise, the error would increase causing instability in the system. This brings a sufficient condition for the stability and the convergence of the error vector for the design of $\mathbf{L}(s)$, which is given for the frequency domain by [23]:

$$|\lambda_i(\mathbf{\Lambda}(s))| \leq 1 \quad \text{for } s = j\omega, \forall \omega \in [-\infty, \infty]. \quad (5)$$

The tracking error will converge to zero as iterations go on if $\mathbf{\Lambda}(s)$ has all its eigenvalues, λ_i , less than one.

From equation (5), the learning transfer matrix $\mathbf{L}(s)$ must be designed so the eigenvalues values of $\mathbf{\Lambda}(s)$ are less than one. The two-by-two $\mathbf{L}(s)$ learning transfer matrix will have the following structure:

$$\mathbf{L}(s) = \begin{bmatrix} L_{11}(s) & L_{12}(s) \\ L_{21}(s) & L_{22}(s) \end{bmatrix}. \quad (6)$$

For the $\mathbf{L}(s)$ design the plant dynamics inversion is a benchmark method to achieve convergence in one iteration. However, the uncertainty present in plant models can lead to sub-optimal tracking, stability issues and convergence rate penalization. Therefore, one must accept that without an exact plant model perfect tracking of the reference is unattainable.

Several studies have addressed the incorporation of model dynamics in the $\mathbf{L}(s)$ design, to increase the convergence rate of the ILC algorithm. The convergence condition and the robustness of an inverse model-based ILC are analyzed in [24]. A quadratic criterion is introduced to analyze the convergence on a model-based ILC in [25]. The optimization problem for the ILC controller is also developed in [26], [27]. As it has been explained in Section I, it would be too laborious to solve the optimization problem of a cost function that considers the algorithm convergence rate, stability and input constraints.

For systems with unstable zeros, pseudo-inverse and stable inversion methods have been proposed in [28] and [29], respectively. A thorough analysis of which inversion approach to use depending on the system characteristics is carried out in [30]. The hydraulic slide circuit in Fig. 2 is a minimum-phase system, therefore, direct inversion can be applied.

In [31] direct inversion was applied in an electronic printer, where the learning filter was designed as $\mathbf{L}(s) = (\mathbf{G}(s)\mathbf{S}(s))^{-1}$. Instead of including a low-pass filter in the $\mathbf{L}(s)$ design, the $\mathbf{Q}(s)$ filter was used to guarantee robustness to modeling errors. However, $\mathbf{Q}(s)$ should be designed properly, as a $\mathbf{Q}(s)$ with a module different from one penalizes the algorithm performance [32].

Direct inversion was also applied in [33], for a linear SISO motor of a wafer system. In [33], it is proposed to include the controller in the model inverse ILC design based on the series and parallel ILC designs. The series, $P_s(s)$, and parallel, $P_p(s)$, ILC architectures have the following model transfer functions:

$$P_s(s) = \frac{\mathbf{G}(s)\mathbf{C}(s)}{1 + \mathbf{G}(s)\mathbf{C}(s)}, \quad P_p(s) = \frac{\mathbf{G}(s)}{1 + \mathbf{G}(s)\mathbf{C}(s)}. \quad (7)$$

$P_s(s)$ and $P_p(s)$ are the process sensitivity functions, which are used as a parametric model of the system to design the learning filter. In [33], unlike in [31], the low-pass filter was applied to the inverse of the process sensitivity functions to deal with the high frequency differences.

This is the most common procedure that one can find in the literature when applying direct inversion [33]–[36].

Although fast convergence tracking is achieved, the algorithm performance is penalized as the inclusion of the low-pass filter to counteract model mismatches affects both $G(s)$ and $C(s)$.

We claim that it is not necessary that the inclusion of the low-pass filter in the learning filter design affects $C(s)$. One can design the learning filter so that the model differences only affect the inverse of $G(s)$. For MIMO systems with a parallel structure consisting of an ILC algorithm and feedback controllers, we propose the following model inverse $L(s)$ design:

$$\begin{aligned} L(s) &= \mathbf{G}^{-1}(s)\mathbf{S}^{-1}(s) = \mathbf{G}^{-1}(s)(I + \mathbf{G}(s)\mathbf{C}(s)) \\ &= \mathbf{G}^{-1}(s) + \mathbf{C}(s). \end{aligned} \quad (8)$$

In (8), only $\mathbf{G}(s)$ is inverted and $\mathbf{C}(s)$ is completely known, see (4). The low-pass filters to deal with high frequency differences will only apply to $\mathbf{G}^{-1}(s)$, not affecting $\mathbf{C}(s)$. With this $L(s)$ design, the convergence rate is improved with respect to the traditional model inverse approach, as the model differences are reduced.

Furthermore, in the $\mathbf{G}(s)$ inversion we only include the plant low frequencies that we are interested in, discarding the high frequencies. The low frequencies provide us with information about the pressure change due to the velocity, which is necessary for the position and force control. The hydraulic press model uncertainties abound at high frequencies, such as pump and valve dynamics and friction forces, these uncertain high frequencies will be attenuated with zero-phase filtering.

B. HYDRAULIC MODELING

In order to carry out the model inverse $L(s)$ design, the hydraulic press modeling is derived in this section.

The axial-piston pump outflow is proportional to the swash plate angle:

$$q_A = q_N \frac{\omega}{\omega_N} \alpha \quad (9)$$

where q_A is the volumetric flow out of the pump (m^3/s), q_N is the nominal flow (m^3/s), ω the shaft rotational speed (rad/s), ω_N the nominal shaft rotational speed (rad/s) and α is the swash plate angle. The terms q_N and ω_N are used to normalize α within [0-1] interval.

Equation (10) shows the relationship between the pressure into the cylinder main chamber and the volumetric flow out of the pump [37]:

$$q_A = A_A \dot{x} + (V_A + A_A x) \beta \dot{P}_A \quad (10)$$

where A_A is the cylinder main chamber area (m^2), V_A is the main cylinder chamber dead volume (m^3), β is the hydraulic compressibility ($1/\text{bar}$), x is the piston position (m) and P_A is the main chamber pressure (bar).

The relationship between the pressure in the cylinder auxiliary chamber and the volumetric flow out of the chamber is as follows [37]:

$$q_B = -A_B \dot{x} + (V_B + A_B(l - x)) \beta \dot{P}_B \quad (11)$$

where q_B is the volumetric flow out of auxiliary chamber (m^3/s), A_B is the auxiliary chamber area (m^2), V_B is the auxiliary chamber dead volume (m^3), l is the cylinder piston length (m) and P_B is the auxiliary chamber pressure (bar).

The relationship between the pressure in the auxiliary chamber and the flow through the valve is shown [38]:

$$q_B = K_v(y_v) q_{ref} \sqrt{P_B / \Delta P_{ref}} \quad (12)$$

where $K_v(y_v)$ is the hydraulic conductance ($\text{m}^3/(\text{bar}\cdot\text{s})$), which is function of the valve spool position y_v . q_{ref} and ΔP_{ref} are included to normalize $K_v(y_v)$ with respect to the nominal flow and the nominal pressure of the valve, respectively.

The force balance equation of the cylinder is:

$$F = m\ddot{x} = P_A A_A - P_B A_B \quad (13)$$

where m is the cylinder moving mass (kg) and F is the output force of the rod (N).

During the *Free Fall* and *Drawing* phases, the oil flow rate into the cylinder main chamber will be equal to the pump outflow. Likewise, the oil flow rate out of the auxiliary chamber will equal the oil flow through the proportional valve.

Rearranging (9) with (10), (11) with (12), and using (13) to relate both cylinder chamber pressures, we can obtain the slide hydraulic circuit system equations, from which the linearized state-space system is derived in equation (14):

$$\begin{aligned} \begin{bmatrix} \delta\ddot{x} \\ \delta\dot{x} \\ \delta\dot{P}_A \\ \delta\dot{P}_B \end{bmatrix} &= \begin{bmatrix} 0 & 0 & A_{13} & A_{14} \\ 1 & 0 & 0 & 0 \\ A_{31} & 0 & 0 & 0 \\ A_{41} & 0 & 0 & A_{44} \end{bmatrix} \begin{bmatrix} \delta\dot{x} \\ \delta x \\ \delta P_A \\ \delta P_B \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_{31} & 0 \\ 0 & B_{42} \end{bmatrix} \begin{bmatrix} \delta\alpha \\ \delta y_v \end{bmatrix} \\ \begin{bmatrix} \delta x \\ \delta P_B \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [\delta\dot{x} \ \delta x \ \delta P_A \ \delta P_B]^T, \end{aligned} \quad (14)$$

The cylinder piston position and velocity, and the pressure in both chambers are chosen as state variables. The swash plate angle and the valve spool position are the system inputs; the piston position and the main chamber pressure are the system outputs.

$\delta P_x = P_x - \bar{P}_x$, $\delta x = x - \bar{x}$, $\delta y_v = y_v - \bar{y}_v$ and $\delta\alpha = \alpha - \bar{\alpha}$, are the small-signal deviation from an operating point obtained in steady-state conditions, \bar{P}_x , \bar{x} , \bar{y}_v and $\bar{\alpha}$ respectively. The definition of every term A_{xx} is shown in Appendix A.

C. HYDRAULIC PRESS MIMO ILC DESIGN

By Laplace transforming equation (14), the system transfer matrix $\mathbf{G}(s)$ is obtained, from which the system inverse will be calculated to carry out the $L(s) = \mathbf{G}^{-1}(s) + \mathbf{C}(s)$ design.

$\mathbf{G}(s)$ is a two-by-two transfer matrix with four transfer functions with a common denominator, from which the system poles can be obtained. To analyze how the system parameters affect the poles of the system, the $\mathbf{G}(s)$ denominator based on the system parameters is obtained. The denominator

is a fourth-order polynomial, each coefficient is shown in Table 1, with $V_1 = V_A + A_A \bar{x}$ and $V_2 = V_B + A_B(l - \bar{x})$.

TABLE 1. $G(s)$ system denominator coefficients.

Order	Coefficient
4	$4V_1^2 V_2^2 \beta^2 m \omega_N \sqrt{\bar{P}_B \Delta P_{ref}}$
3	$-V_1^2 V_2 \beta m \omega_N K_v(\bar{y}_v) q_{ref}$
2	$4A_A V_1 \beta \omega_n \sqrt{\bar{P}_B \Delta P_{ref}} (A_A V_2^2 + A_B^2 V_1 V_B)$
1	$4A_A^2 V_2^2 \beta q_N \omega \sqrt{\bar{P}_B \Delta P_{ref}} - A_A^2 V_1 V_2 \omega_N K_v(\bar{y}_v) q_{ref}$ $+ 2A_B^2 V_1^2 \beta \omega_N K_v(\bar{y}_v) \bar{P}_B \Delta P_{ref}$
0	0

We can put the polynomial in Table 1 into numbers, with the values provided by the data-sheets, shown in Appendix B, and obtain the poles of our system:

TABLE 2. $G(s)$ system poles.

Pole	Damping	Frequency (rad/s)
0.00	-1.00	0.00
-2.09e-01	1.00	2.09e-01
-4.26e-03 + 5.86e+01i	7.28e-05	5.86e+01
-4.26e-03 - 5.86e+01i	7.28e-05	5.86e+01

From Table 2 we can see that we have a pole in zero, a low frequency pole and a high frequency complex pole. The high frequency poles have really small damping, as the hydraulic slide energy does not dissipate anywhere as no friction has been included in the system model, e.g. friction between the slide and the rails or between the cylinder piston and the chambers. Although some authors include the friction forces in the force balance equation of the cylinder, see [39], we will discard them as they will only affect our plant at high frequencies. The low frequency pole describes the cylinder pressure change due to the velocity.

In the $G(s)$ inverse design, we will not include the high frequencies, as these frequencies are where the uncertainties of our plant abound. The effects of the oil compressibility appear at high frequencies, so we could discard every term containing β in the coefficients shown in Table 1. However, the terms containing β and P_B provide us with information of the rate at which the pressure is increasing, necessary for the pressure control.

As shown in Appendix B, β is a small number. In Table 1, in the fourth-order coefficient β appears squared, so the value of the coefficient will be insignificant in comparison to the others. In the third-order coefficient, β does not multiply any pressure variable, therefore we can discard it as well, it will not provide us with any valuable information at low frequencies. After these considerations the resulting polynomial, with the low frequency poles is as follows:

The numerators, $b_{g_{xx}}(s)$, corresponding to each transfer function of $G(s) = [g_{11}(s) \ g_{12}(s); \ g_{21}(s) \ g_{22}(s)]$ are shown:

$$b_{g_{11}}(s) = -A_A V_1 V_2 q_N \omega (q_{ref} K_v(\bar{y}_v) - 4V_2 \beta \sqrt{\bar{P}_B \Delta P_{ref}} s)$$

$$b_{g_{12}}(s) = -2A_B \bar{P}_B V_1^2 V_2 \beta \omega_N q_{ref} \sqrt{\Delta P_{ref}} s$$

$$b_{g_{21}}(s) = A_A A_B q_N \omega \sqrt{\bar{P}_B \Delta P_{ref}} \left(2V_2 s + K_v(\bar{y}_v) q_{ref} \sqrt{\bar{P}_B} \right)$$

$$b_{g_{22}}(s) = 2\bar{P}_B V_2 q_{ref} \sqrt{\Delta P_{ref}} (\omega_N A_A^2 V_1 s + q_N \omega A_A^2 + \beta m \omega_N V_1^2 s^3) \quad (15)$$

TABLE 3. Low-frequency $G(s)$ system denominator coefficients.

Order	Coefficient
2	$4A_A V_1 \beta \omega_n \sqrt{\bar{P}_B \Delta P_{ref}} (A_A V_2^2 + A_B^2 V_1 V_B)$
1	$4A_A^2 V_2^2 \beta q_N \omega \sqrt{\bar{P}_B \Delta P_{ref}} - A_A^2 V_1 V_2 \omega_N K_v(\bar{y}_v) q_{ref}$ $+ 2A_B^2 V_1^2 \beta \omega_N K_v(\bar{y}_v) \bar{P}_B \Delta P_{ref}$
0	0

In the same way as for the poles, the zeros at high frequencies are undesired. In $b_{g_{22}}(s)$, a high frequency zero appears in the terms multiplied by β and no pressure variable appears. This term is discarded resulting in:

$$\hat{b}_{g_{22}}(s) = 2\bar{P}_B V_2 q_{ref} \sqrt{\Delta P_{ref}} (\omega_N A_A^2 V_1 s + q_N \omega A_A^2) \quad (16)$$

The Bode diagram of the system shown in (14), $G(s)$, and the system without the high frequency poles and zeros, $\hat{G}(s)$, is shown in Fig. 4. At low frequencies, both systems have the same response and as frequency increases, both systems' responses deviate.

The new learning transfer matrix design will be $L(s) = \hat{G}^{-1}(s) + C(s)$, with the simplified system.

Not incorporating the high frequencies in $L(s)$ will affect our learning rate, however, it will have little impact as the low frequencies have the greatest influence on the convergence rate. Furthermore, we will obtain an equivalent lower-order system from which the inverse calculation will be simpler.

We include a low-pass filter, to attenuate the high frequencies of the system that are not included in the plant design. Furthermore, with this filter, we will attenuate the non-modeled physical dynamics of the proportional valve and the pump plate, as they are unknown.

Incorporating a fourth-order low-pass filter to the plant inverse modeling we obtain the following $L(s)$ design:

$$L(s) = \underbrace{\hat{G}^{-1}(s)}_{\hat{G}_f^{-1}(s)} \frac{\omega_c^4}{(s + \omega_c)^4} + C(s). \quad (17)$$

where ω_c is the filter cutoff frequency in rad/s.

ω_c is set to 2 rad/s to ensure the robustness of the learning law to the high frequency dynamics that we have not modeled. However, producing a cutoff we introduce a considerable phase lag in our system, see the response of $\hat{G}_f^{-1}(s)$ in Fig. 5.

To avoid the additional phase lag, zero-phase low-pass filters (ZPF) are widely used in ILC design [22], [23], as they allow to filter the error signal forward and then, apply backward filtering to the filtered signal. Backward filtering generates phase lead to compensate for the phase loss of the forward filtering to achieve a zero-phase effect.

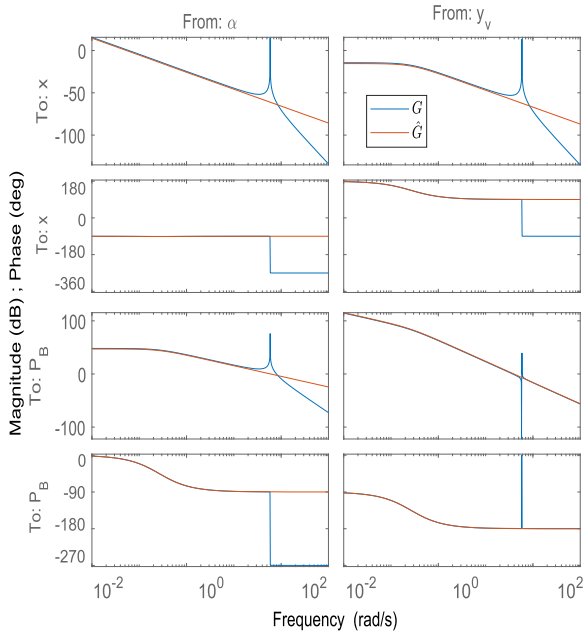


FIGURE 4. Bode diagram of $G(s)$ and $\hat{G}(s)$.

The fourth-order filter is divided into two second-order low-pass filters, by which the iteration is filtered twice, once in the positive direction of time and once in the negative direction of time. The resulting plant inverse transfer matrix is:

$$L(s) = \underbrace{\hat{G}^{-1}(s) \frac{\omega_c^2}{(s + \omega_c)^2} \frac{\omega_c^2}{(s - \omega_c)^2}}_{\hat{G}_{zpf}^{-1}(s)} + C(s). \quad (18)$$

To perform the filtering the entire iteration error vector must be available for processing, therefore the ZPF cannot be done in real time, but one can perform the filtering between iterations. In Fig. 5, the Bode diagram of the resulting plant inverse with ZPF is shown, $\hat{G}_{zpf}^{-1}(s)$, where the unknown high frequencies are attenuated without losing phase.

In order to show the advantages of not affecting $C(s)$ with the low-pass filter, we compare the plant inverse design proposed in (18), $L_{MIC}(s)$, and the traditional direct model inverse design used in literature [33]–[36], $L_{MI}(s)$, with the inverse design without system simplifications nor low-pass filter introduction $L_D(s)$. The learning filters are defined as follows:

$$\begin{aligned} L_{MIC}(s) &= \hat{G}^{-1}(s) \frac{\omega_c^2}{(s + \omega_c)^2} \frac{\omega_c^2}{(s - \omega_c)^2} + C(s) \\ L_{MI}(s) &= \left(\hat{G}^{-1}(s) + C(s) \right) \frac{\omega_c^2}{(s + \omega_c)^2} \frac{\omega_c^2}{(s - \omega_c)^2} \\ L_D(s) &= G^{-1}(s) + C(s) \end{aligned} \quad (19)$$

The Bode diagrams of the three different model inverse approaches are shown in Fig. 6. It can be seen that $L_D(s)$ contains the high frequency dynamics of the slide hydraulic

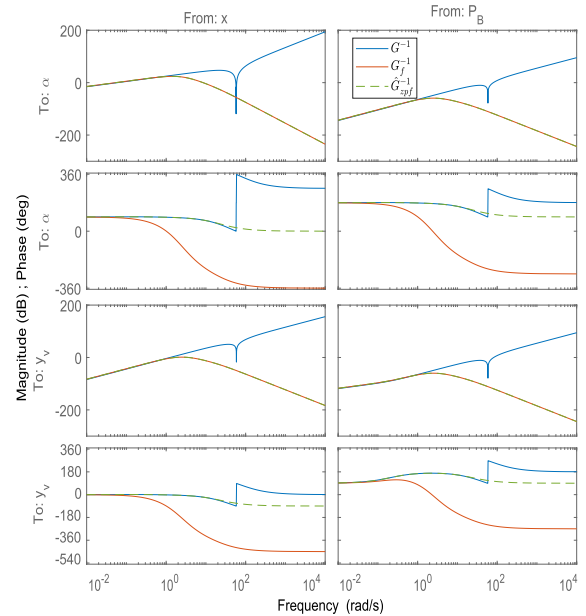


FIGURE 5. Bode diagram of the plant inverse $G^{-1}(s)$, with low-pass filters $\hat{G}_f^{-1}(s)$ and with zero-phase filtering $\hat{G}_{zpf}^{-1}(s)$.

system, as no simplifications have been done and no low-pass filter has been applied.

We can see, in Fig. 6, that the Bode diagrams (b) and (c) are not affected by where the low-pass filter is placed, as $C(s)$ only consists of elements in the diagonal. However, in (a) and (d) both systems $L_{MIC}(s)$ and $L_{MI}(s)$ have different frequency responses from 0.2 rad/s on. In (d) it is more visible how $L_{MIC}(s)$ is closer to the exact model than $L_{MI}(s)$. Indeed, from frequency 0.2 rad/s to 100 rad/s, $L_{MIC}(s)$ behaves in the same way as $L_D(s)$.

This is a consequence of applying a really low cutoff frequency in the low-pass filter introduced, as we want to get rid of most of the high frequencies and just kept those very low frequencies that we are interested in. In case we applied a higher cutoff frequency, the differences between $L_{MIC}(s)$ and $L_{MI}(s)$ would reduce at low frequencies, there would only be differences at the high frequency regions.

D. ZERO-PHASE FILTER WITH ANTI-WINDUP

The ZPF is applied to obtain the control inputs for the pump swash plate and the proportional valve position. Both actuators have a physical limitation, the input obtained from the ZPF must not exceed the range $[-1, 0]$ and $[0, 1]$ for the valve and pump, respectively. To limit the signal obtained from the ZPF, an anti-windup has been included to the filtering process.

The general recursive formulation for discrete-time filters in z -transform holds as follows:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_J z^{-J}} \quad (20)$$

The forward and backward filters have been designed as second-order filters, therefore, we set $M = J = 2$.

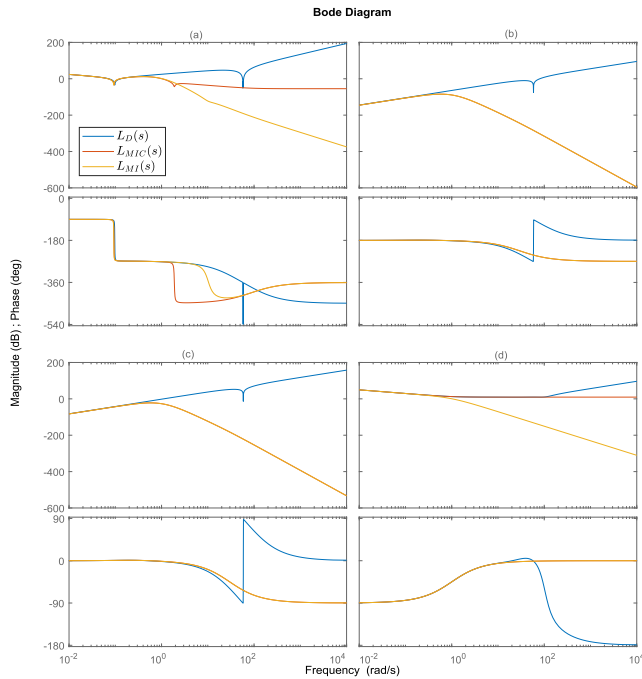


FIGURE 6. Bode diagram of $L_{MIC}(s)$, $L_{MI}(s)$ and $L(s)$.

The resulting linear differential equation is:

$$y(n) = \frac{b_0}{a_0}x(n) + \frac{b_1}{a_0}x(n-1) + \frac{b_2}{a_0}x(n-2) - \frac{a_1}{a_0}y(n-1) - \frac{a_2}{a_0}y(n-2) \quad (21)$$

where n is a nonnegative integer.

The static gain of the filters is given as:

$$K_s = \frac{\sum_{i=1}^M b_i}{\sum_{i=1}^J a_i} \quad (22)$$

The filter output is given by (21). However, the obtained output value of the filter must not exceed the specified ranges, else the valve and pump control inputs will saturate.

If the output of the filter exceeds the ranges of interest, the anti-windup is applied to the recursive filter for the last J outputs and M inputs. In this way, we avoid the integration of the error for the recursive outputs and inputs, increasing the performance of the filter.

The resulting filter algorithm including the anti-windup is shown in Algorithm 1.

E. MIMO ILC STABILITY ANALYSIS

The convergence condition of the ILC problem has been widely considered in the literature to guarantee algorithm stability. The Hurwitz stability is used in [40], the Schur stability is applied in [41] and vertex Markov parameters are used in [42].

These methods require a large number of computations and cannot be applied directly to ILC. A more intuitive method is to analyze the stability graphically, from a frequency point of view, as it is done for SISO systems in [22], [24], [42], [43].

Algorithm 1 Recursive Filtering With Anti-Windup

input: b (numerator coefficients), a (denominator coefficients), u (input), y_{max} (upper saturation), y_{min} (lower saturation)

```

1: for  $k = 1 \rightarrow \text{length}(u)$  do
2:    $y(k) = \frac{b_0}{a_0}x(k) + \frac{b_1}{a_0}x(k-1) + \frac{b_2}{a_0}x(k-2) - \frac{a_1}{a_0}y(k-1) - \frac{a_2}{a_0}y(k-2)$   $\triangleright$  Filter the inputs signal sample by sample using (21)
3:   if  $y(k) \leq y_{min}$  then
4:     for  $i = 1 \rightarrow \text{length}(a)$  do
5:        $y(i) = y_{min}$   $\triangleright$  Saturate previous outputs
6:     end for
7:     for  $i = 1 \rightarrow \text{length}(b)$  do
8:        $u(i) = y_{min}(i) * \frac{1}{K_s}$   $\triangleright$  Saturate previous inputs by multiplying the minimum value by the inverse of the static gain
9:     end for
10:    end if
11:    if  $y(k) \geq y_{max}$  then
12:      for  $i = 1 \rightarrow \text{length}(a)$  do
13:         $y(i) = y_{max}$   $\triangleright$  Saturate previous outputs
14:      end for
15:      for  $i = 1 \rightarrow \text{length}(b)$  do
16:         $u(i) = y_{max}(i) * \frac{1}{K_s}$   $\triangleright$  Saturate previous inputs by multiplying the maximum value by the inverse of the static gain
17:      end for
18:    end if
19:  end for
20: return:  $y$   $\triangleright$  The entire filtered signal is returned

```

In these studies, the frequency response for all the frequencies up to the Nyquist frequency is plotted to analyze the convergence condition:

$$|1 - G(j\omega)S(j\omega)L(j\omega)| \leq 1/Q(j\omega) \quad \forall \omega \in [-\infty, \infty]. \quad (23)$$

Equation (23) can be interpreted on a Nyquist plot; if the frequency response remains inside the unit circle, the ILC algorithm will converge to zero error. Else, if the response gets out of the unit circle, the ILC algorithm will be unstable.

For the convergence condition of a MIMO ILC problem, the eigenvalues should be analyzed at all frequencies, instead of the frequency response as in the SISO case. We introduce again the MIMO convergence condition with the new $L(s)$ design:

$$|\lambda_i(\mathbf{I} - \mathbf{S}(s)\mathbf{G}(s)(\hat{\mathbf{G}}_{zpf}^{-1}(s) + \mathbf{C}(s)))| \leq |\mathbf{Q}^{-1}(s)|. \quad (24)$$

for $s = j\omega$, $\forall \omega \in [-\infty, \infty]$.

So far, no studies have been done regarding the graphical analysis of the convergence condition of a MIMO ILC algorithm. We propose to apply the same procedure as in the SISO ILC, however, instead of looking to a Nyquist plot of the frequency response, we analyze the eigenvalues of (24) at all frequencies.

From equation (24), if every eigenvalue of the term $\Lambda_2(s) = I - S(s)G(s)(\hat{G}_{zpf}^{-1}(s) + C(s))$ is less than one, then the convergence condition of the MIMO ILC will be guaranteed.

$\Lambda_2(s)$ is a two-by-two transfer matrix from which two eigenvalues can be obtained at each frequency point. Therefore, in order to guarantee the stability for all frequencies we evaluate the frequency response of the term $\Lambda_2(s)$ up to the Nyquist frequency.

Ideally, if an accurate model inverse is included in the $L(s)$ design, $\Lambda_2(s)$ would be a zero matrix with every eigenvalue at zero for all frequencies. However, our $L(s)$ design is far from being ideal as some simplifications have been considered in the inverse model and low-pass filters have been added to attenuate the uncertain high frequencies.

Hence, one should aim to obtain every eigenvalue as close as possible to the origin. The closer the eigenvalues to the origin, the faster those error frequencies will be corrected by the ILC algorithm. Figure 7 shows each eigenvalue obtained from $\Lambda_2(s)$ at each frequency. We are not interested in the phase of the eigenvalues, but in their module, else the error would not asymptotically converge. It would in general oscillate around a value and eventually decay to it.

At low frequencies, the eigenvalues are close to zero, as it has been pointed out in Fig. 7 for a frequency of $\omega = 2$ rad/s. At high frequencies the eigenvalues value increases, therefore, it will take more iterations to correct the high frequencies of the error. Although the eigenvalues increase, none of them get out of the unit circle, which guarantees the stability of the algorithm at high frequencies.

If the eigenvalues got out of the unit circle at a specific frequency, we could make use of $Q(s)$ to enlarge the stability circle so the eigenvalues at all frequencies remain inside the new stability circle. By setting a $|Q(s)| < |I|$ the stability circle increases accordingly, see in (24) that the smaller the value of $Q(s)$ the bigger the value of the right-hand side. However, one should be cautious in the $Q(s)$ design, as a $Q(s)$ value different from one affects the convergence performance of the ILC algorithm, as it was shown in [44].

Some authors design $Q(s)$ as a low-pass filter, to ensure the condition for stability is met [44], [45]. Using this $Q(s)$ design approach, one can determine which frequencies are emphasized in the learning process. However, perfect tracking will not be achieved.

III. SIMULATION RESULTS

The designed MIMO control ILC algorithm has been implemented in a nonlinear model of a hydraulic press in Matlab/Simulink. The press model has been developed in a novel library made by Ikerlan [46]. With this library, components are easily parametrized with data-sheet information in contrast to Simscape, which requires acquiring constructive parameters obtainable only from laboratory experiments. This novel approach allows reproducing the physical behavior of industrial components with high precision without losing Real-Time capabilities.

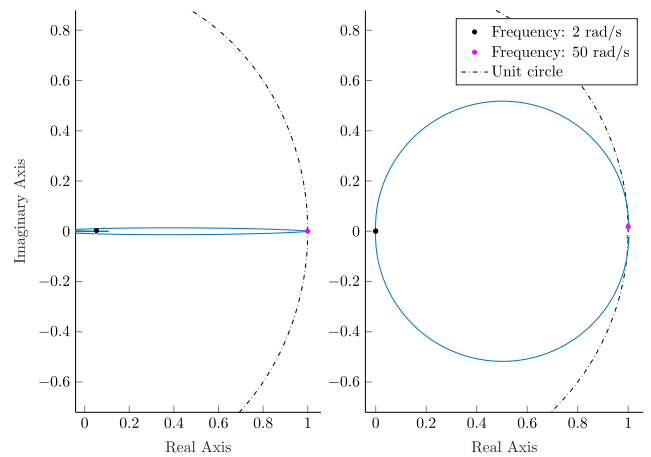


FIGURE 7. $\Lambda_2(s)$ eigenvalues plot at each frequency.

The ILC algorithm is compared to the performance of a PI controller, which is the most prevalent controller applied in hydraulic circuits. Two different PI controllers have been designed, for the pump swash plate angle and the auxiliary chamber proportional valve spool position. The proportional valve PI controller gains have been set to $K_{Pv} = 3$ and $K_{Iv} = 3$. The pump PI controller gains have been set to $K_{Pp} = 3.86e-05$ and $K_{Ip} = 0.82$. Both PI controllers have been discretized applying Tustin approximation as the hydraulic press models run in discrete time with a fixed step-size of 0.002s.

A. POSITION CONTROL RESULTS

As explained in Sec. I, the control objective is to track a predefined slide trajectory by controlling the pump swash plate angle and the proportional valve spool position.

Figure 8 shows both *Free Fall* and *Drawing* phases, where the position control is carried out. The ILC signal is introduced at $t \approx 0.5s$, before the transition between the two phases.

At the first iteration, with the PI controller, there exists a manifest rebounding which results in the deceleration of the slide velocity. After the rebounding, the proportional valve spool position oscillates around -0.2, see Fig. 9, and it is not until $t \approx 3.5s$ that the position reference is reached.

As iterations go on, the ILC corrects the proportional valve spool position during both phases transition, introducing an overshoot before oscillating around -0.2 input signal value. This overshoot avoids the slide rebounding, resulting in a perfect tracking of the *Free Fall* and *Drawing* transition. After this transition, the input signal reaches the value -0.2 and then, oscillates smoothly around this value to maintain the constant falling velocity of the slide.

This effect can be seen in the slide velocity progress through iterations in Fig. 10. At the first iteration, there exists a notorious overshoot in the velocity signal which is attenuated as iterations go on. With the introduction of the ILC signal, once the slide reaches -100 mm/s speed it is maintained at this level.

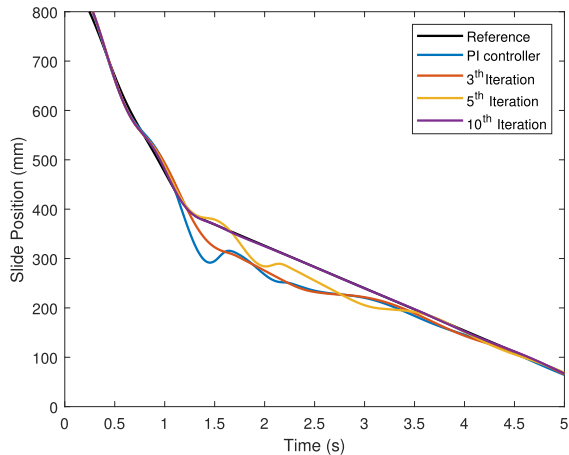


FIGURE 8. Free Fall and Drawing phases slide position control.

The pump swash plate angle normalized between [0-1], and the auxiliary chamber pressure are shown in Fig. 11 and Fig. 12, respectively. During the *Free Fall* phase the auxiliary chamber pressure is given by the slide weight and no pressure control is done. This can be seen in Fig. 11 where the pump plate is closed until $t \approx 1.0$ s.

Once the *Drawing* phase starts, from $t \approx 1.0$ s onwards, the pump swash plate is opened completely to reach the pressure reference of 180 bar. At the first iteration, with the PI controller, the pressure signal oscillates harshly around the reference as the swash plate angle total input has large oscillations. These swash plate oscillations are reduced as iterations go on, and the total input settles around 0.7 value. With this improvement the pressure reference signal is reached faster and the oscillations are considerably reduced.

It should be pointed out that perfect pressure tracking is not needed, it is sufficient to keep the auxiliary chamber pressure close to the reference so, when the slide strikes the workpiece during the *Drawing* phase, the extra pressure needed can be supplied by the auxiliary chamber pressure.

As a performance index for the proposed position control ILC algorithm, the RMSE between the desired trajectory and the slide trajectory over iterations is shown in Fig. 13. The RMSE is reduced by a factor of 7 in the position tracking, and a fast convergence is obtained as at the seventh iteration the error is considerably reduced, with respect to the first iteration with a PI controller.

IV. ILC PERFORMANCE COMPARISON

In this study, we propose an ILC algorithm design based on the simplified model inverse and the feedback controller (MIC-ILC). With this design, we optimize the convergence rate of the ILC algorithm to reduce the required number of iterations.

To analyze the performance of the MIC-ILC, we compare it with the proportional-type ILC (P-ILC) [47] and the traditional direct model-inverse ILC (MI-ILC) algorithm approach with direct inversion and the low-pass filter influencing the controller [34]. The comparison will be done for

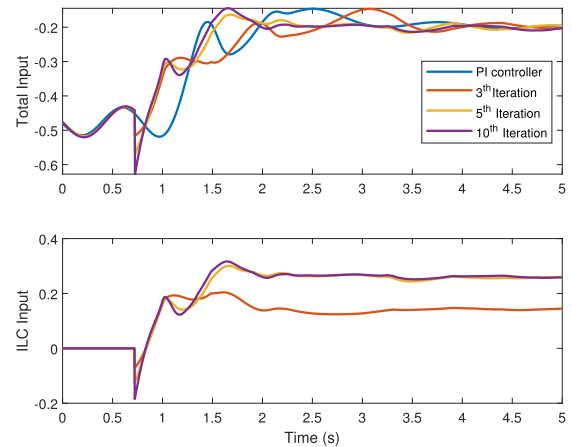


FIGURE 9. Total input to the valve and ILC input to the valve over iterations.

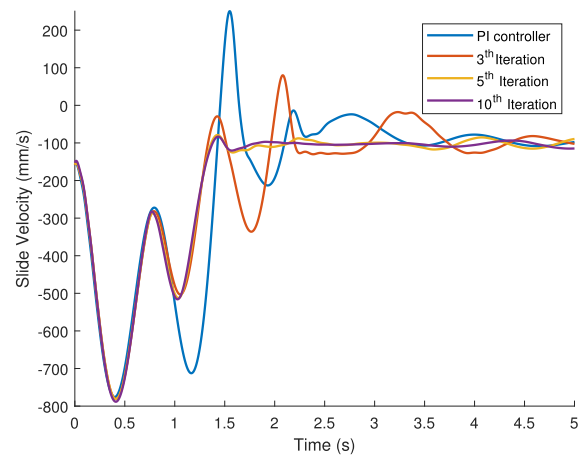


FIGURE 10. Slide velocity during Free Fall and Drawing phases.

the MIMO position control problem, and the RMSE between the position reference and the slide position will be used as the performance index.

A. P-ILC DESIGN

The P-ILC is one of the most used ILC algorithms in the literature due to its simplicity, as it calculates an input signal proportional to the error. In this section we will extend the P-ILC algorithm proposed by Arimoto in [47] to the MIMO case. The MIMO P-ILC learning control law is as follows:

$$U_{j+1}(s) = U_j(s) + K_P E_j(s) \quad (25)$$

where K_P is the proportional learning matrix, which is used to maximize the convergence rate of the algorithm. The following K_P design is proposed:

$$K_P = \begin{bmatrix} 0.001 & 0.2 \\ 0.0001 & 2.5 \end{bmatrix} \quad (26)$$

The eigenvalues of the P-ILC algorithm at each frequency are shown in Fig. 14, which we compare with the response obtained with MIC-ILC in Section II-E. With the P-ILC, at zero frequency, the eigenvalues lie at (+1,0), whereas with

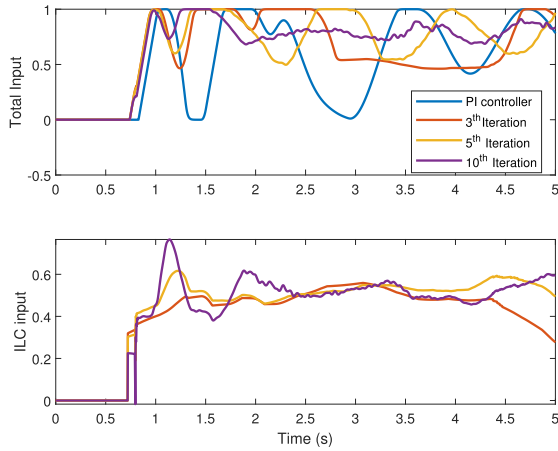


FIGURE 11. Total input to the pump swash plate and ILC input to the pump swash plate over iterations.

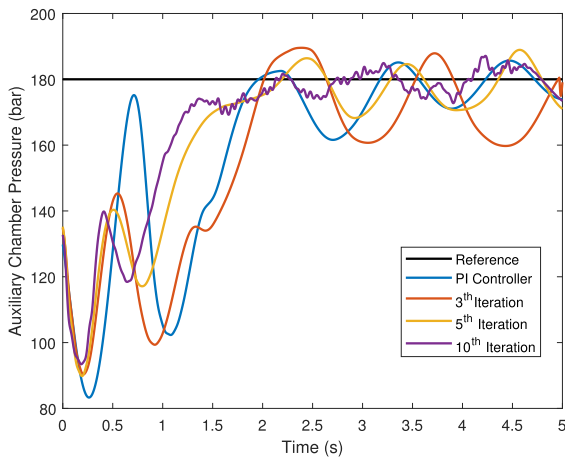


FIGURE 12. Auxiliary chamber pressure through iterations.

MIC-ILC, at zero frequency, the eigenvalues lie at the origin. This is due to the cancellation of the model dynamics at low frequencies, that optimizes the convergence rate of the ILC algorithm.

The low frequencies will be corrected faster with MIC-ILC algorithm than with the P-ILC, however, we can still obtain fast convergence with the P-ILC if K_P is increased so that the eigenvalues are as close as possible to the origin at low frequencies. However, the very low frequencies will always be close to $(+1,0)$, penalizing its performance. As it is pointed out in Fig. 14, in both designs at 2 rad/s frequency the eigenvalues are close to the origin, remarkably closer with the MIC-ILC.

The RMSE between the position reference and the slide position with the P-ILC and the MIC-ILC algorithm is shown in Fig. 15. A similar convergence rate is obtained with both algorithms, although the P-ILC converges to a bigger steady-state error.

As it was shown in [19], the P-ILC can become unstable in the presence of measurement noise and state disturbance, and one would expect to obtain a noisy signal from the hydraulic press position sensor. Therefore, we add a varying white noise

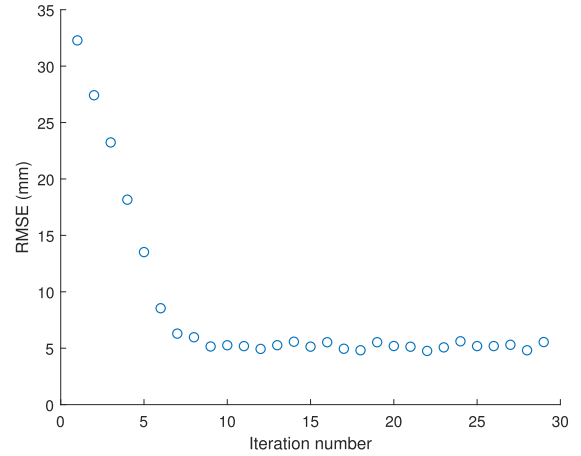


FIGURE 13. RMSE between the position reference and slide position over iterations.

to the position sensor to analyze the P-ILC performance under noisy conditions. The noisy slide position error during the *Free Fall* and *Drawing* phases is shown in Fig. 16.

The RMSE over iterations is shown in Fig. 17, the P-ILC algorithm converges fast at the first iterations, however, as iterations go on it gets unstable and the error increases. We could decrease the gains in (26), so the amplification of the error is not that severe. However, decreasing the P-ILC gains the convergence rate is also decreased, see Fig. 17. The MIC-ILC algorithm remains stable due to the low-pass filters included in its design.

B. TRADITIONAL MODEL INVERSE DESIGN

In this section the traditional model inverse ILC (MI-ILC) will be designed, in which the low-pass filter is applied to the entire sensitivity function. The approach proposed by [34] is followed and the learning filter is designed as follows:

$$L_{MI}(s) = \hat{G}^{-1}(s) \underbrace{(I + G(s)C(s))}_{S^{-1}(s)}. \quad (27)$$

In order to attenuate the model high frequency uncertainties and the noise added to the position sensor, a fourth-order low-pass filter is added to $L_{MI}(s)$ and zero-phase filtering is carried out to avoid the phase loss. Following [33] design, the filter is added to the entire $L_{MI}(s)$ transfer matrix:

$$L_{MI}(s) = \hat{G}^{-1}(s) \underbrace{(I + G(s)C(s))}_{S^{-1}(s)} \frac{\omega_c^2}{(s + \omega_c)^2} \frac{\omega_c^2}{(s - \omega_c)^2}. \quad (28)$$

with $\omega_c = 2$ rad/s.

Note that the placement of the low-pass filter changes with respect to the MIC-ILC algorithm, recall equation (18), where the low-pass filter is only applied to $G^{-1}(s)$. This variation directly affects the convergence, as the model differences are bigger and the eigenvalues lie further from the origin, see Fig. 18.

In both MI-ILC and MIC-ILC designs, at zero frequency the eigenvalues lie at the origin as a result of the model dynamics cancellation. Therefore, the MI-ILC will perform

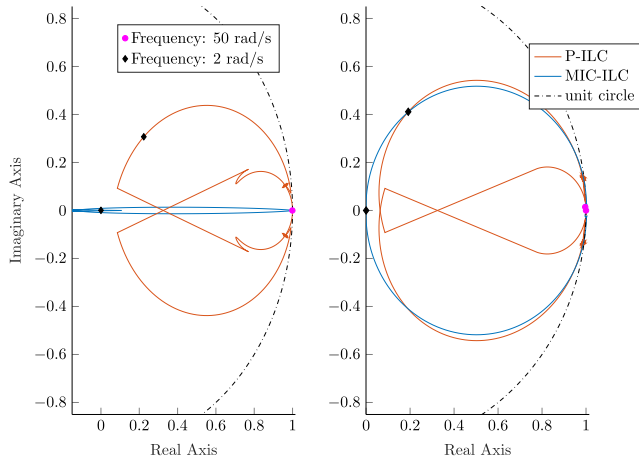


FIGURE 14. Eigenvalues plot at each frequency for the P-ILC and MIC-ILC algorithms.

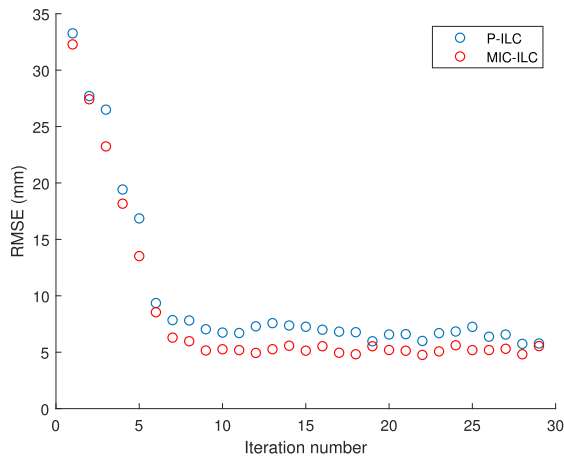


FIGURE 15. RMSE comparison with P-ILC and the MIC-ILC algorithm.

better than the P-ILC as the very low frequencies lie close to the origin. However, due to the application of the low-pass filter to the entire $L_{MI}(s)$, the eigenvalues grow apart from the origin faster in the MI-ILC than in the MIC-ILC algorithm. This can be seen in particular at 2 rad/s frequency point.

In Fig. 19, a faster convergence rate is obtained with the MIC-ILC in comparison to the MI-ILC. The former converges in 10 iterations and the latter in 20 iterations. Although the MI-ILC obtains fast convergence, from iteration 20 onwards the convergence rate gets reduced and the error converges gradually. This is a consequence of the eigenvalues distance from the origin as the frequency increases. With the MIC-ILC, the eigenvalues are closer to the origin so the low frequency components of the error are corrected faster.

Apart from the RMSE as a performance indicator, the Power Spectral Density (PSD) also provides useful insights for the comparison between ILC algorithms. In Fig. 20, the PSD of the error signal over iterations is shown. It can be seen that the energy is concentrated on the frequency band $[0 \sim +30]$ Hz.

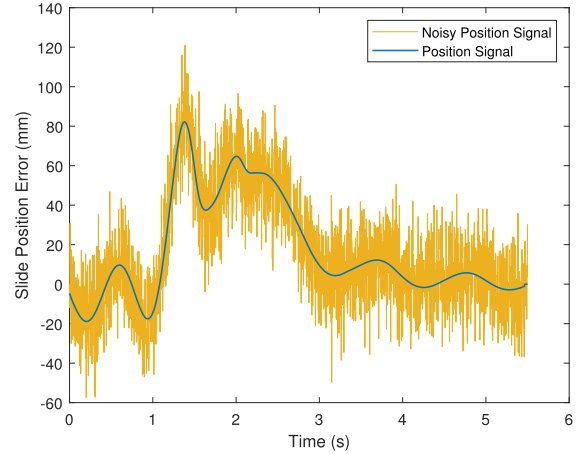


FIGURE 16. Position error with and without noise during the Free Fall and Drawing phases.

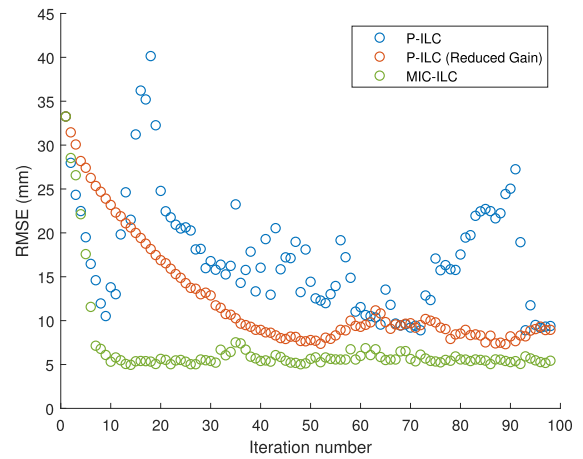


FIGURE 17. RMSE comparison with P-ILC and the MIC-ILC algorithm under noisy conditions.

The simulation has been carried out without noise, so the performance of the three algorithms can be analyzed in an ideal scenario. The main enhancement is at low frequencies, where the MIC-ILC outperforms the other ILC algorithms. At the first 10 iterations, the MIC-ILC reduces faster the low frequency components of the error. Especially the frequency band from 0 Hz to 5 Hz is decreased considerably, as those frequencies are where the eigenvalues lie close to the origin.

At higher frequencies, the three algorithms perform similarly as, at those frequencies, we only seek stability regardless of the convergence rate. It can be seen that at high frequencies the error reduction is slower, however, the MIC-ILC still achieves a faster correction rate. In line with the results obtained with the RMSE, at iteration 20, the three algorithms learning process has already converged and the difference between iterations is minimal, the frequency component of the error remains similar.

V. EXPERIMENTAL VALIDATION IN HYDRAULIC TEST RIG

The proposed MIC-ILC position control algorithm has been implemented in the hydraulic test rig shown in Fig 21.

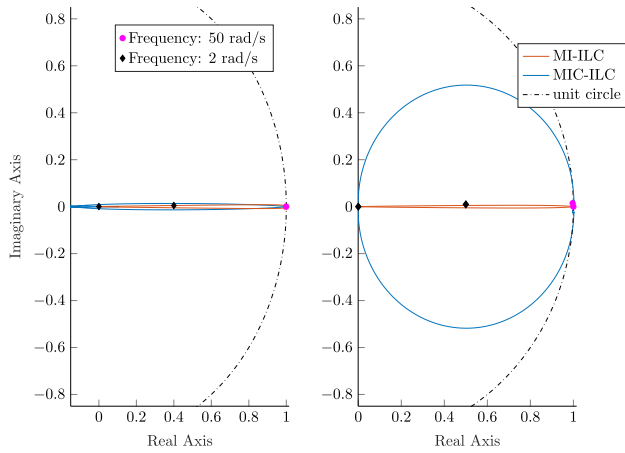


FIGURE 18. Eigenvalues plot at each frequency for the MI-ILC and MIC-ILC algorithms.

It consists of two identical hydraulic circuits and each circuit comprises of: a double-acting cylinder, a 4-way proportional valve, a hydraulic pump, a pressure relief valve, an accumulator, a pressure sensor at each cylinder chamber and a flow sensor.

The hydraulic pumps are fixed displacement pumps, which flow displacement varies depending on the rotational speed of the pump. With a fixed displacement pump, instead of controlling the swash angle, as in the pump shown in Section II-B, the pump’s angular velocity is controlled. The relationship between the fixed displacement axial piston pump outflow and the shaft speed is as follows:

$$q = \frac{q_N}{\omega_N} \omega_s \quad (29)$$

where ω_s is the shaft rotational speed (rad/s).

Every system actuator and sensor are connected via Ethernet communication to a Beckhoff 6930-0050. The Beckhoff Industrial PC (IPC) acts as a control cabinet and enables real-time communications and high-performance to control the hydraulic test rig. The Beckhoff IPC receives and processes the sensor signals, and delivers back the corresponding control inputs to the actuators. The position MIC-ILC algorithm is implemented in TwinCAT 3, which is a platform developed by Beckhoff, where PLC programming is carried out.

The position reference the press must follow during the *Free Fall* and *Drawing* phases is shown in Fig. 22. From the figure, the difference between the two phases falling velocity can be seen, this transition takes place at $t \approx 2s$. For the pressure reference tracking, a 60 bar pressure level is specified, so it can be maintained during the *Free Fall* and *Drawing* phases.

As a first approach we design two PI controllers, to control the pump and the valve. The pump PI controller gains values are: $K_{Pp} = 4$ and $K_{Ip} = 3.8$. The valve PI controller gains values are: $K_{Pv} = 1.5$ and $K_{Iv} = 0.25$. The resulting position and force control is shown in Fig. 23 where the coupling

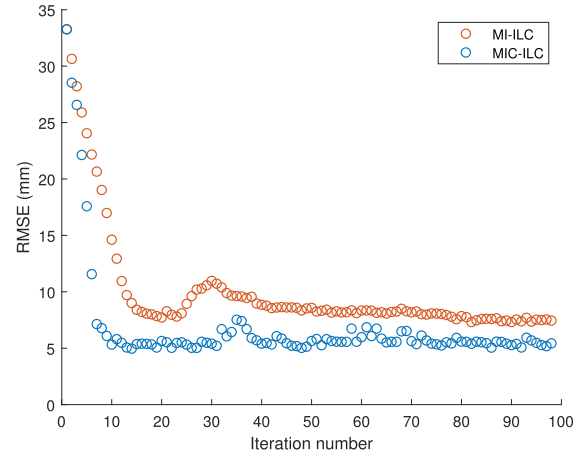


FIGURE 19. RMSE comparison with MI-ILC and the MIC-ILC algorithm under noisy conditions.

between the two control loops can be seen. The pressure signal oscillates around the pressure reference of 60 bar. When the pressure is higher than the reference is due to an excessive pump velocity, which yields faster falling velocity of the slide. On the contrary, when the pressure is lower than the reference, is due to insufficient pump velocity, which translates into a slide velocity slowing down.

To improve the PI controller performance the MIC-ILC algorithm is implemented. The MIC-ILC input signals are introduced from the start of the *Free Fall* phase to the end of the *Drawing* phase.

We design the learning gain based on (18). The two PI controllers’ gains are depicted above, and the system is modeled following (14) state space design with the test rig parameters depicted in Appendix C. The resulting eigenvalues of the frequency response matrix are shown in Fig. 24.

A similar response to that shown in Fig. 7 for the simulations is obtained. At low frequencies, the eigenvalues are close to zero, as it has been pointed out for a frequency of $\omega = 2$ rad/s. At high frequencies the eigenvalues value increases, diverging from the origin. However, the responses get out of the unit circle, therefore we make use of the Q matrix to enlarge the stability circle. A $Q = 0.93I$ is set, which will penalize the convergence of the algorithm but will keep it stable.

The result is shown in Fig. 25. The position tracking has already converged at iteration 10, where the oscillations existing with the PI controllers are reduced. In Fig. 26, how the MIC-ILC has adjusted the valve spool position to the slide velocity change can be seen. At iteration 10, from $t \approx 0s$ to $t \approx 2.2s$, the valve spool position is reduced progressively from -0.2 to -0.1, to achieve fast velocity during the *Free Fall* phase. Once in the *Drawing* phase, the valve spool position is maintained at a -0.1 value, so a constant falling velocity is achieved.

The oscillations in the pressure reference tracking are also reduced, as shown in Fig. 27. At the beginning of the step the MIC-ILC introduces a positive velocity, see Fig. 28, to reach

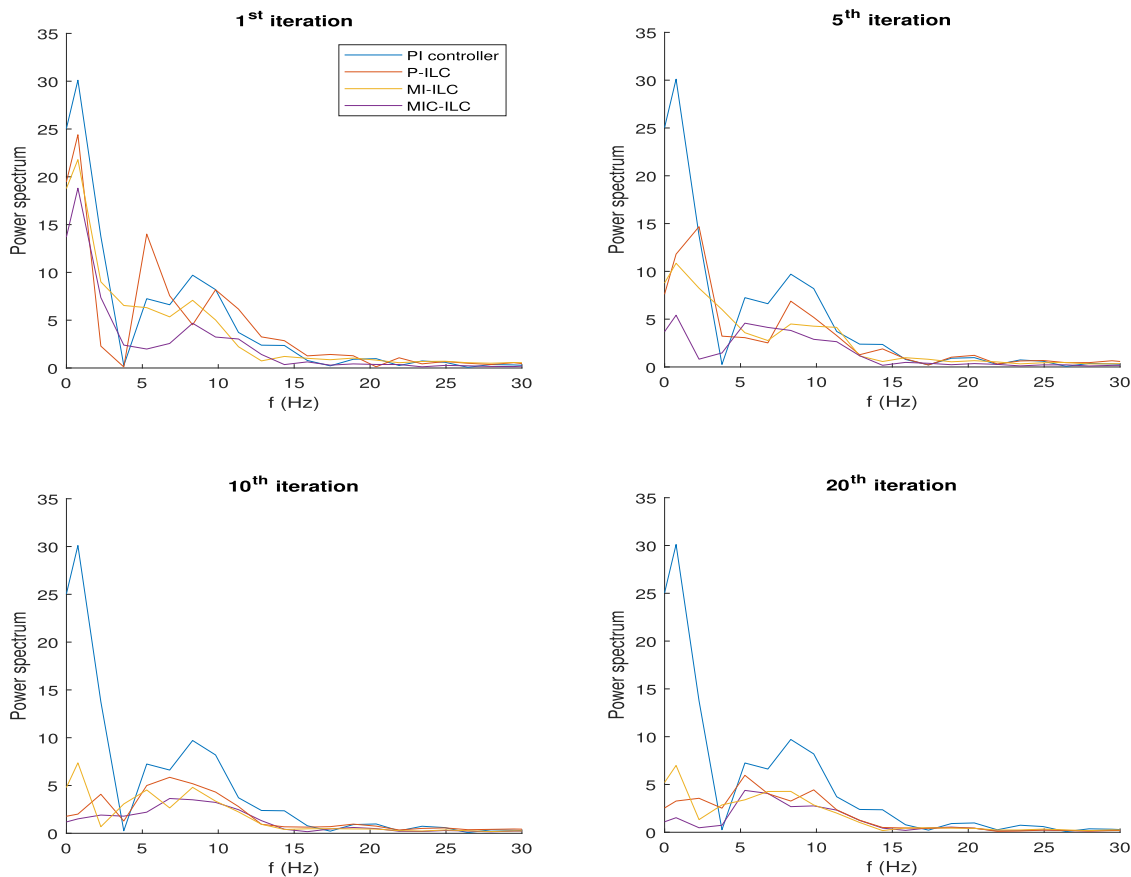


FIGURE 20. Power spectrum of the pressure error signal, for the four different controllers.

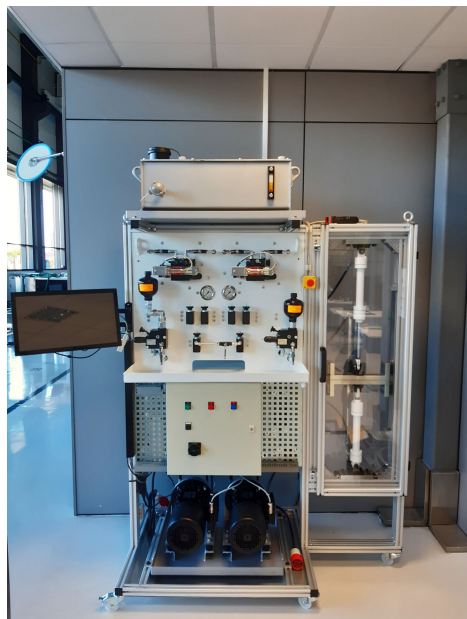


FIGURE 21. The available hydraulic test rig at Ikerlan Technology Research Center.

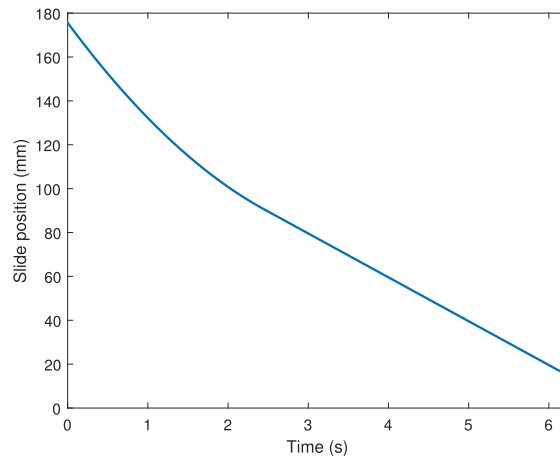


FIGURE 22. Press trajectory, Free Fall and Drawing phases.

the reference pressure as fast as possible, and then, a negative signal, to reduce the overshoot. From $t \approx 2s$ on, the MIC-ILC

introduces a constant velocity signal to maintain the pressure level at the reference.

As a performance index for the proposed MIMO ILC algorithm, the RMSE between the position reference and the slide position signal is shown in Fig. 29. The RMSE is reduced by a factor of 4 in the position tracking, and a fast convergence is obtained as at the tenth iteration the error is

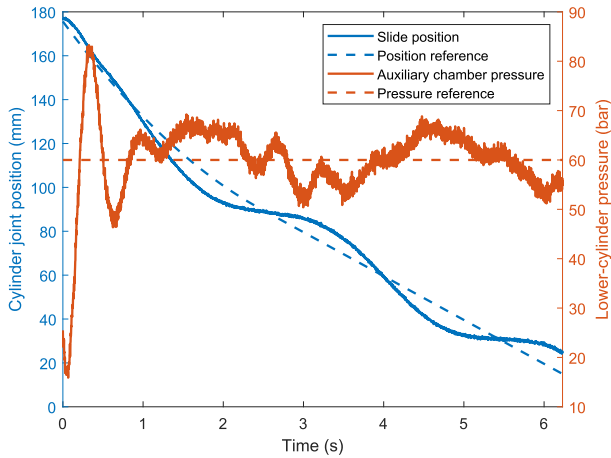


FIGURE 23. Position and pressure signal during *Free Fall* and *Drawing* phases with PI controller.

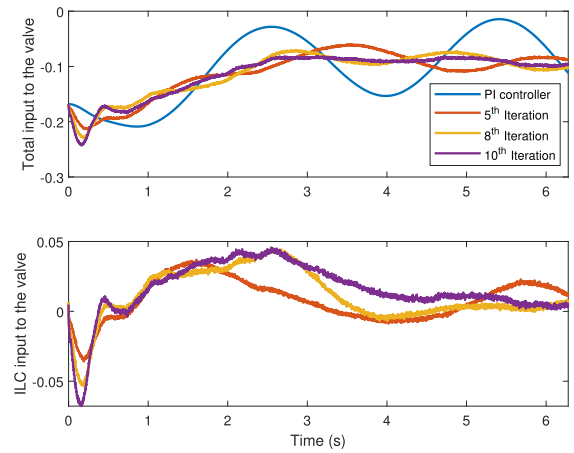


FIGURE 26. Valve input signal during *Free Fall* and *Drawing* phases in hydraulic test rig.

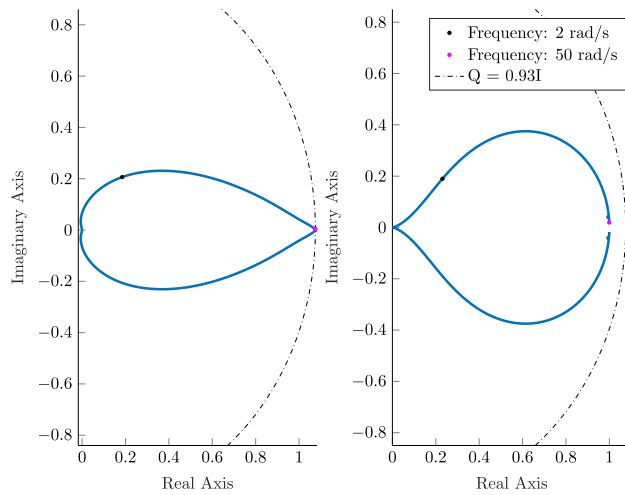


FIGURE 24. Eigenvalues plot at each frequency for the hydraulic test rig model inverse design.

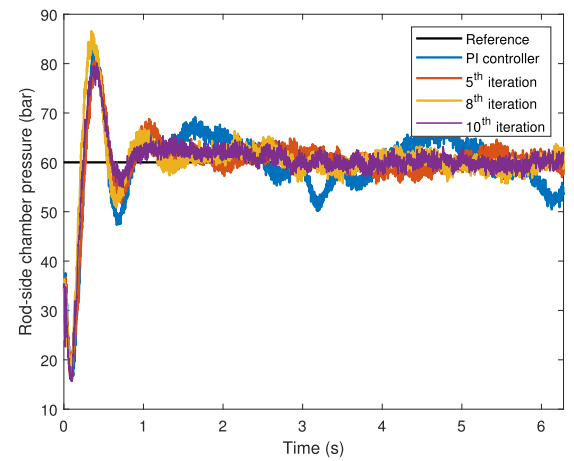


FIGURE 27. Pressure tracking for a 60 bar reference during *Free Fall* and *Drawing* phases in hydraulic test rig.

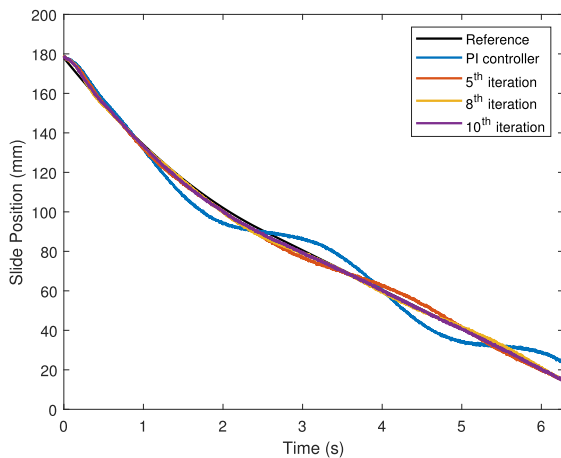


FIGURE 25. Position signal during *Free Fall* and *Drawing* phases in hydraulic test rig.

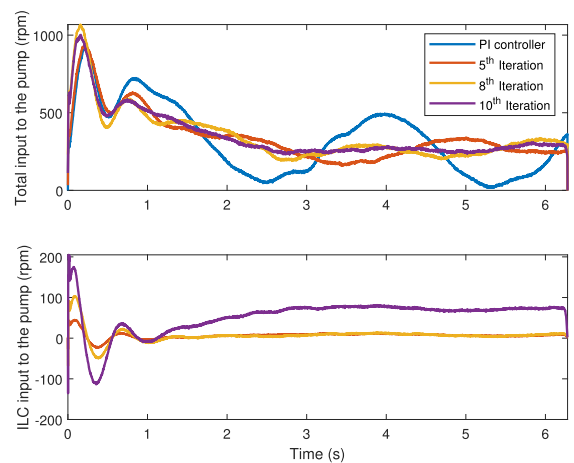


FIGURE 28. Pump velocity during *Free Fall* and *Drawing* phases in hydraulic test rig.

considerably reduced, with respect to the first iteration with the PI controller.

The RMSE between the 60 bar pressure reference and auxiliary chamber pressure signal is shown in Fig. 30. The error is reduced by a factor of 2.6 regarding the first iteration with the PI controller.

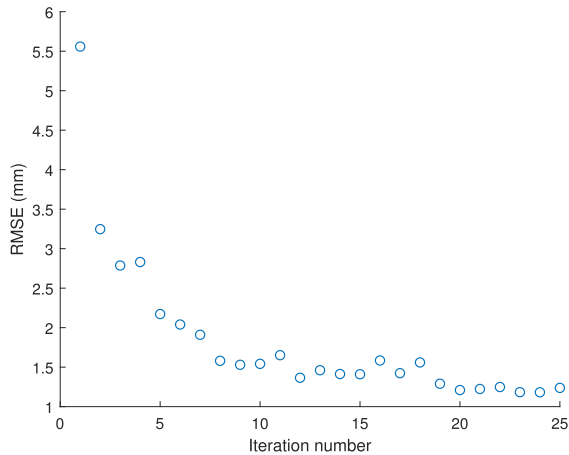


FIGURE 29. RMSE between the position reference and press position over iterations.

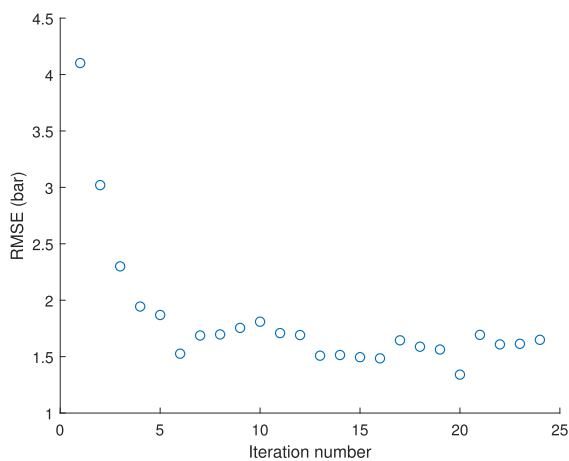


FIGURE 30. RMSE between the pressure reference and the auxiliary chamber pressure over iterations.

VI. CONCLUSION

A new hydraulic press multiple-input-multiple-output (MIMO) position control based on Iterative Learning Control (ILC) is proposed. The MIMO ILC position control eliminates the need to manually define control signals and reduces the number of iterations required to converge with respect to other ILC algorithms.

The proposed ILC algorithm design is based on the simplification of the model inverse and the existing plant feedback controller (MIC-ILC) to increase the error convergence rate. A graphical evaluation is provided to analyze the stability and convergence condition. With this method, the stability of the algorithm is fulfilled if the learning filter eigenvalues lie inside the unit circle. The closer the eigenvalues lie to the origin, the faster the MIMO ILC algorithm will converge.

The MIMO ILC position control has been implemented in a nonlinear hydraulic press model in Matlab/Simulink for which satisfactory results have been obtained. The proposed MIC-ILC algorithm has been compared to the most common ILC algorithms in the literature. By means of the proposed graphical stability design, as the MIC-ILC eigenvalues lie

closer to the origin than the other algorithms, faster convergence rate has been achieved.

This is one of the main contributions of the MIC-ILC, the ability of correcting the low frequency components faster than the existing ILC algorithms in the literature. The enhancement of the MIC-ILC has been verified in the frequency domain, with the Power Spectral Density, analyzing the error signal at each iteration for each ILC algorithm. The evolution of the error signal in the frequency domain also shows that the MIC-ILC outperforms other ILC methods in terms of stability and convergence rate.

The MIC-ILC has been implemented in a hydraulic test rig available at Ikerlan Research Center. The implementation results correspond to the results obtained through simulations. The MIC-ILC not only achieves good tracking in the position and pressure signals but also gets fast convergence.

The proposed algorithm has shown to perform correctly in hydraulic press tests, in which the experiments are mainly carried out at low frequencies. It remains to analyze the performance of the algorithm at higher frequency response experiments, such as in fatigue testing machines or servo drives.

Although ILC yields satisfactory results and the convergence rate that we obtain is high with respect to other works, it requires starting the learning from scratch every time the reference is modified. This could be a drawback when implementing it in a real press operation, as every time the design of a workpiece is modified the ILC algorithm would need to perform the entire learning task to obtain the optimal input.

Techniques such as Reinforcement Learning (RL) could be of use, to obtain a nonlinear policy from a specific cost function. Some works have used PILCO probabilistic model [48], for learning a throttle valve control in a combustion engine with successful results [49]. It is to be analyzed whether via RL techniques the performance of a PI controller could be further improved.

APPENDIX.

A. LINEARIZED SYSTEM PARAMETERS

The complete expressions of the terms introduced in the linearized state-space model, introduced in (14), are as follows:

$$\begin{aligned}
 A_{13} &= \frac{A_A}{m} \\
 A_{14} &= -\frac{A_B}{m} \\
 A_{31} &= -\frac{A_1}{(V_A + A_A \bar{x})\beta} \\
 A_{41} &= \frac{A_2}{(V_B + A_B(l - \bar{x}))\beta} \\
 A_{44} &= \frac{K_v(\bar{y}_v)q_{ref}}{(V_B + A_B(l - \bar{x}))\beta} \frac{1}{2\sqrt{\bar{P}_B/\Delta P_{ref}}} \\
 B_{31} &= \frac{\frac{q_N}{\omega_N}\omega}{(V_A + A_A \bar{x})\beta} \\
 B_{42} &= \frac{\sqrt{\bar{P}_B/\Delta P_{ref}}}{(V_B + A_B(l - \bar{x}))\beta} \dot{K}_v(\bar{y}_v)q_{ref} \quad (30)
 \end{aligned}$$

B. NOMINAL VALUES FOR HYDRAULIC CIRCUIT PARAMETERS

The system parameters appearing in the terms introduced in the linearized state-space model, shown in (14), are as follows:

- Cylinder moving mass: $m = 26500$ kg.
- Cylinder main chamber area: $A_A = 0.16$ m².
- Cylinder auxiliary chamber area: $A_B = 0.02$ m².
- Main chamber dead volume: $V_1 = 0.35$ m³.
- Auxiliary chamber dead volume: $V_2 = 0.14$ m³.
- Cylinder stroke length: $l = 1.2$ m.
- Hydraulic compressibility: $\beta = 1.23 \cdot 10^{-4}$ 1/bar.
- Pump nominal flow: $q_N = 0.0088$ m³/s.
- Shaft rotational speed: $\omega = 150$ rad/s.
- Nominal shaft rotational speed: $\omega_N = 138.23$ rad/s.
- Auxiliary chamber operating point: $\bar{P}_B = 180$ bar.
- Hydraulic conductance operating point: $K_v(\bar{y}_v)$ m³/s.

C. HYDRAULIC TEST RIG PARAMETERS

The hydraulic rig system parameters are shown:

- Cylinder main chamber area: $A_A = 0.000804$ m².
- Cylinder auxiliary chamber area: $A_B = 0.000424$ m².
- Cylinder stroke length: $l = 0.2$ m.
- Valve nominal pressure: $\Delta P_{ref} = 35$ bar.
- Valve nominal flow rate $q_{ref} = 1.667 \cdot 10^{-4}$ m³/s.
- Hydraulic compressibility: $\beta = 1.23 \cdot 10^{-4}$ 1/bar.
- Nominal shaft rotational speed: $\omega_N = 1000$ rad/s.

REFERENCES

- [1] P. Pratumswan and A. Junchangpoed, "Force and position control in the electro-hydraulic system by using a MIMO fuzzy controller," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2013, pp. 1462–1467.
- [2] M. Cheng, J. Zhang, B. Xu, R. Ding, and J. Wei, "Decoupling compensation for damping improvement of the electrohydraulic control system with multiple actuators," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1383–1392, Jun. 2018.
- [3] Y. Lim, R. Venugopal, and A. G. Ulsoy, "Multi-input multi-output (MIMO) modeling and control for stamping," *J. Dyn. Syst., Meas., Control*, vol. 132, no. 4, pp. 1–12, Jul. 2010.
- [4] A. Jansson, P. Krus, and J.-O. Palmberg, "Decoupling of response and pressure level in a hydraulic actuator," in *Proc. 4th Bath Int. Fluid Power Workshop*, 1991, pp. 260–275.
- [5] M. Elfving and J. O. Palmberg, "Distributed control of pressure controlled fluid power actuators-decoupled chamber pressure controlled cylinder," in *Proc. Fluid Power Syst., 9th Bath Int. Fluid Power Workshop*, 1996.
- [6] S. Schröders, C. C. Maier, W. Ebner, A. Fidlín, and C. Hametner, "Two-degree-of-freedom MIMO control for hydraulic servo-systems with switching properties," *Control Eng. Pract.*, vol. 95, Feb. 2020, Art. no. 104246.
- [7] X. Wang, D. Cong, Z. Yang, S. Xu, and J. Han, "Iterative learning control with complex conjugate gradient optimization algorithm for multiaxial road durability test rig," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 233, no. 7, pp. 2349–2360, Apr. 2019.
- [8] X. Wang, D. Cong, Z. Yang, S. Xu, and J. Han, "Modified quasi-Newton optimization algorithm-based iterative learning control for multi-axial road durability test rig," *IEEE Access*, vol. 7, pp. 31286–31296, 2019.
- [9] T. Muller and C. Endisch, "An estimation based iterative learning control approach for cross-coupled vehicle test systems," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2017, pp. 1375–1381.
- [10] S. Chen and C. T. Freeman, "Decentralised collaborative and formation iterative learning control for multi-agent systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 2729–2734.
- [11] D. Yoon, X. Ge, and C. E. Okwudire, "Optimal inversion-based iterative learning control for overactuated systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1948–1955, Sep. 2020.
- [12] M.-B. Radac, R.-E. Precup, and E. M. Petriu, "Model-free primitive-based iterative learning control approach to trajectory tracking of MIMO systems with experimental validation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2925–2938, Nov. 2015.
- [13] C. E. Boudjedir, M. Bouri, and D. Boukhetala, "Model-free iterative learning control with nonrepetitive trajectories for second-order MIMO nonlinear systems—Application to a delta robot," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7433–7443, Aug. 2021.
- [14] L. Blanken and T. Oomen, "Multivariable iterative learning control design procedures: From decentralized to centralized, illustrated on an industrial printer," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 4, pp. 1534–1541, Jul. 2020.
- [15] D. N. C. Nam, N. M. Tri, H. G. Park, and K. K. Ahn, "Position control of electro hydraulic actuator (EHA) using an iterative learning control," *J. Drive Control*, vol. 11, no. 4, pp. 1–7, 2014.
- [16] C.-K. Chen and W.-C. Zeng, "The iterative learning control for the position tracking of the hydraulic cylinder," *JSME Int. J. C Mech. Syst., Mach. Elements Manuf.*, vol. 26, no. 2, pp. 720–726, 2003.
- [17] P. Q. Ling, "Fuzzy iterative learning control of electro-hydraulic servo system for SRM direct-drive volume control hydraulic press," *J. Central South Univ. Technol.*, vol. 17, pp. 316–322, Apr. 2010.
- [18] C.-K. Chen and J. Hwang, "PD-type iterative learning control for the trajectory tracking of a pneumatic X-Y table with disturbances," *JSME Int. J. C Mech. Syst., Mach. Elements Manuf.*, vol. 49, no. 2, pp. 520–526, 2006.
- [19] P. R. Ouyang and P.-I. Pipatpaibul, "Iterative learning control: A comparison study," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, 2012, pp. 939–945.
- [20] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial," *Trans. Soc. Instrum. Control Eng.*, vol. 19, pp. 709–712, Jan. 1978.
- [21] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.
- [22] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Control*, vol. 73, no. 10, pp. 930–954, 2010.
- [23] D. Wang, Y. Ye, and B. Zhang, *Practical Iterative Learning Control With Frequency Domain Design and Sampled Data Implementation*. Springer, 2014.
- [24] T. J. Harte, J. Hatonen, and D. H. Owens, "Discrete-time inverse model-based iterative learning control: Stability, monotonicity and robustness," *Int. J. Control*, vol. 78, no. 8, pp. 577–586, May 2005.
- [25] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.
- [26] D. H. Owens, C. T. Freeman, and B. Chu, "Multivariable norm optimal iterative learning control with auxiliary optimisation," *Int. J. Control*, vol. 86, no. 6, pp. 1026–1045, 2013.
- [27] D. H. Owens and J. Häätönen, "Iterative learning control—An optimization paradigm," *Annu. Rev. Control*, vol. 29, no. 1, pp. 57–70, 2005.
- [28] J. Ghosh and B. Paden, "Pseudo-inverse based iterative learning control for nonlinear plants with disturbances," in *Proc. 38th IEEE Conf. Decis. Control*, Dec. 2000, pp. 5206–5212.
- [29] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. Autom. Control*, vol. 41, no. 7, pp. 930–942, Jul. 1996.
- [30] J. V. Zundert and T. Oomen, "On inversion-based approaches for feedforward and ILC," *Mechatronics*, vol. 50, pp. 282–291, Apr. 2018.
- [31] N. Strijbosch, L. Blanken, and T. Oomen, "Frequency domain design of iterative learning control and repetitive control for complex motion systems," in *Proc. IEEE Int. Workshop Sens., Actuation, Motion Control, Optim.*, 2018, pp. 1–2.
- [32] D. A. Bristow, K. L. Barton, and A. G. Alleyne, *Iterative Learning Control*. 2010.
- [33] B. T. Fine, S. Mishra, and M. Tomizuka, "Model inverse based iterative learning control using finite impulse response approximations," in *Proc. Amer. Control Conf.*, 2009, pp. 931–936.
- [34] J. S. Yeon, J. H. Park, S.-W. Son, and S.-H. Lee, "Model-based iterative learning control for industrial robot manipulators," in *Proc. IEEE Int. Conf. Autom. Logistics*, Aug. 2009, pp. 24–28.

- [35] K. S. Lee, S. H. Bang, and K. S. Chang, "Feedback-assisted iterative learning control based on an inverse process model," *J. Process Control*, vol. 4, no. 2, pp. 77–89, May 1994.
- [36] M. Norrlöf and S. Gunnarsson, *A Model Based Iterative Learning Control Method Applied to an Industrial Robot*. Linköping, Sweden: Linköping Univ. Press, 1999.
- [37] B. Eryilmaz and B. H. Wilson, "Unified modeling and analysis of a proportional valve," *J. Franklin Inst.*, vol. 343, no. 1, pp. 48–68, Jan. 2006.
- [38] H. Merritt, *Hydraulic Control Systems*. Hoboken, NJ, USA: Wiley, 1967.
- [39] V. Z. Šitum, "Force and position control of a hydraulic press," *Krmiljenje Hidravlične Stiskalnice*, vol. 17, no. 4, pp. 314–320, 2011.
- [40] C.-L. Jiang, "Sufficient condition for the asymptotic stability of interval matrices," *Int. J. Control*, vol. 46, no. 5, pp. 1803–1810, 1987.
- [41] P. Batra, "On necessary conditions for real robust Schur-stability," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 259–261, Feb. 2003.
- [42] K. Kinoshita, T. Sogo, and N. Adachi, "Iterative learning control using adjoint systems and stable inversion," *Asian J. Control*, vol. 4, no. 1, pp. 60–67, 2002.
- [43] P. H. Gøyttil, M. R. Hansen, and G. Hovland, "Iterative learning applied to hydraulic pressure control," *Model., Identificat. Control*, vol. 39, no. 1, pp. 1–14, 2018.
- [44] D. A. Bristow and M. Tharayil, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.
- [45] K. Vinther, V. Chandan, and A. G. Alleyne, "Learning/repetitive control for building systems with nearly periodic disturbances," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 1198–1203.
- [46] J. Rodriguez-Guerra, C. Calleja, I. Elorza, A. M. Macarulla, A. Pujana, and I. Azurmendi, "A methodology for real-time HiL validation of hydraulic-press controllers based on novel modeling techniques," *IEEE Access*, vol. 7, pp. 110541–110553, 2019.
- [47] S. Arimoto, "Learning control theory for robotic motion," *Int. J. Adapt. Control Signal Process.*, vol. 4, no. 6, pp. 543–564, Dec. 1990.
- [48] M. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 408–423, Feb. 2015.
- [49] B. Bischoff, D. Nguyen-Tuong, T. Koller, H. Markert, and A. Knoll, "Learning throttle valve control using policy search," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Artificial Intelligence), vol. 8188. Springer, 2013, pp. 49–64.



ELOY IRIGOYEN (Senior Member, IEEE) received the M.Eng. degree in electrical engineering from the University of the Basque Country, in 1992, and the Ph.D. degree in industrial engineering from the Public University of Navarre, in 2003. He has been a Lecturer of system engineering and automation with the University of the Basque Country (UPV/EHU), since 2001. He has also been an Assistant Lecturer with the Public University of Navarre and the University of Deusto. He is leading and managing the Intelligent Control Research Group, UPV/EHU. His main research interests include intelligent control and biomedical engineering.



ARON PUJANA-ARRESE received the bachelor's degree in physic sciences from the Faculty of Science, UPV/EHU, in 2001, and the master's degree in advanced studies due to research work focused on modeling and controlling an actuator device based on pneumatic muscles from the Department of Computer Science and Automation, UNED University, in 2008, under the co-direction of the prestigious Dr. Sebastián Dormido Bencomo.

In 2001, he joined the Department of Control and Automation, IKERLAN, where he is currently the Head of the Department. Since his early beginning as a Researcher, his work is geared toward the fields of modeling complex systems, and virtual prototyping and designing advanced control techniques. He has authored or coauthored more than 20 papers and conferences related to the subjects of modeling and controlling mechatronics systems. In the last years, his focus has changed to the field of renewable energies and the design of advanced controllers for hydraulic actuators.



IGNACIO TROJAOLA received the degree in renewable energies engineering from the University of the Basque Country, Spain, and the M.Sc. degree in control and automation from Aalborg University, Denmark. He is currently a Researcher with the Department of Control and Monitoring, IKERLAN. He is also involved in the control of hydraulic systems research line at IKERLAN, where he is researching for his Ph.D.



IKER ELORZA received the degree in mechanical engineering from the University of the Basque Country and the Ph.D. degree from the University of Bristol, in 2011. He conducted his Ph.D. research at the University of Bristol. He is currently a Researcher with the Department of Control and Monitoring, IKERLAN. His research interest includes the modeling and control of industrial machinery.



GORKA SORROSAL graduated in automatic and industrial electronic engineering from the University of Deusto, in 2011. He received the M.Sc. degree in control, automatic and robotic engineering from the University of the Basque Country, in 2013, and the Ph.D. degree from the University of Deusto, in 2018, with the thesis "Intelligent modeling and optimization strategies for the BTO process (Bioethanol-To-Olefins)."

From 2009 to 2016, he worked at DeustoTech Energy (Deusto Foundation) in projects related to computer vision, namely RELIFO, IkusGurune, and VisionTech4Life, and processes optimization, such as BIOTRANS and EOLICO. He started to work at IKERLAN, in 2016, where he is currently a member of the Control and Monitoring Team. His current research interests include the modeling, control, and optimization of complex systems, and computer vision.

...