International Congress
Motor Vehicles & Motors 2024
Kragujevac, Serbia
October 10th - 11th, 2024

Motor Vehicles & Motors

MVM2024-NNN

Milan Matijevic[1]
Vojislav Filipovic[2]
Dragan Kostic[3]

ITERATIVE LEARNING CONTROL (ILC) IN MANUFACTURING SYSTEMS: DESIGN OF ILC ALGORITHMS AND OVERVIEW OF MODEL INVERSION TECHNIQUES FOR ILC SYNTHESIS

ABSTRACT: This paper presents a comprehensive overview of Iterative Learning Control (ILC) and its application in manufacturing systems, focusing on the design of ILC algorithms and the use of model inversion techniques for ILC synthesis. General principles of the standard approach to ILC algorithm design are discussed, and various model inversion techniques are compared. The analysis is supported by an illustrative example involving a general mechatronic subsystem, demonstrating the advantages of ILC over conventional control methods. The results highlight the potential of ILC to significantly enhance the performance of manufacturing systems.

KEYWORDS: Manufacturing systems, Iterative Learning Control (ILC), ILC design, Model inversion, Mechatronic systems

INTRODUCTION

The productivity and consistent quality of products in modern manufacturing systems are greatly influenced by the design of mechatronic systems and the employed motion control techniques. These advanced systems enable precise and rapid movements, which are essential for maintaining high standards in manufacturing. By combining data-based (learning-type) control and conventional (non-learning-type) control methods, these systems ensure high-precision tracking and control of machines' mechanical parts and robotic manipulators, thus enhancing overall productivity and uniformity of products [11], [15], [26].

Non-learning-type control methods include various traditional and well-established techniques (Internal Model Control (IMC), Internal Model Principle (IMP), IMPACT structure, robust control, LQ/LQG control, estimation-based control and so on). These non-learning-type methods are crucial for achieving desired control performance in various applications, ensuring stability, robustness, and precision in system operations.

Iterative Learning Control (ILC) is a data-driven control method that has become essential with the technological advancements in repetitive robotic manipulators in manufacturing, such as car-body paint shops filled with robots performing repetitive tasks. Over the past 30 years, ILC has grown in significance within control theory and applications, particularly for tasks requiring high accuracy beyond the capabilities of traditional feedback control [15].

ILC functions similarly to human learning through repetition. A system performs a task multiple times, identifies errors, and adjusts to reduce these errors in subsequent attempts, gradually improving performance. This method is

[1] *Milan Matijevic, University of Kragujevac, Faculty of Engineering, Sestre Janjic 6, Kragujevac, Serbia, matijevic@kg.ac.rs*
[2] *Vojislav Filipovic, University of Kragujevac, Faculty of Mechanical and Civil Engineering in Kraljevo, Serbia*
[3] *Dragan Kostic, Enabling Technology Mechatronics, ASMPT Center of Competency, Netherlands*

especially effective in systems with repeated tasks, as initial errors are used to enhance future performance. ILC can achieve highly precise tracking of periodic trajectories with a simple update rule [17].

ILC's application spans various fields, initially focusing on improving manufacturing systems and industrial robots' performance. Its use has expanded to include rotary systems, batch processes, manufacturing processes, chemical processes, biomedical applications, actuators, semiconductors, power electronics, unmanned aerial vehicles (UAVs), and high-speed trains. For instance, in automotive testing, ILC is used in laboratory rigs to simulate road conditions for testing car performance, highlighting its importance in achieving precise control through continual adjustment and improvement [16], [11], [14].

A comprehensive overview of the synthesis methods for Iterative Learning Control (ILC) algorithms and their applications can be found in various review papers and doctoral theses. These sources include works by Poot [16], [2], [8], [11], [14], [3], [1], [25], [19], [20], and [9].

ILC is applied in modern mechatronic systems in numerous fields such as lithography, data storage, industrial printing, advanced manufacturing, robotic systems, pick-and-place machines (e.g., wire-bonders and die-bonders), microscopy, wind turbine centres, and magnetic field mapping using Simultaneous Localization and Mapping (SLAM).

This paper describes the standard methodology for synthesizing Iterative Learning Control (ILC) algorithms, also known as model inversion-based ILC or frequency domain ILC synthesis [14], [3], [11]. The fundamental assumption is that systems perform the same repetitive tasks under similar initial conditions and disturbances over a finite time interval (like robotic painting or welding on a moving factory line). The core of ILC algorithm synthesis is the design of a learning filter, which is based on model inversion of sensitivity or complementary sensitivity function of mechatronic system. Designing a learning ILC filter can be challenging because ideally inverting the dynamic model is not always possible due to potential unstable and/or non-minimum phase (NMP) characteristics of the digital system. Systems with unstable zeros are known as non-minimum phase (NMP) systems, and their inversion would result in an unstable system. In general, inverting models that contain unstable or lightly damped zeros/poles is not a trivial task [14],[3], [23], [11]. The inversion of the dynamic model is a typical procedure in the synthesis of feedforward controllers [10].

Several methods are used to approximate the stable inverse of NMP systems [12], including:

- Nonminimum phase zeros ignore (NPZ-ignore), [21],
- Zero-phase-error tracking controller (ZPETC), [22], [6],
- Zero-magnitude-error tracking controller (ZMETC), [6], [18],
- Non-causal series expansion, [5], [4],
- Exact unstable inverse,
- Zeroth order series, and so on. However, using a zeroth-order series expansion is essentially equivalent to ignoring the nonminimum-phase zeros (while accounting for the appropriate DC gain). Approximating the inverse of a system in this manner provides a more straightforward approach, but it may not be as accurate [7], [18].

In this paper, we will present and examine the impact of known model inversion methodologies on the quality of ILC system synthesis. The learning filter should ensure quick convergence of ILC signals and high accuracy in tracking the reference trajectory of the controlled system, ultimately leading to high overall performance in the manufacturing system. ILC is a powerful data-driven control method that effectively handles complex control issues such as high nonlinearity, strong coupling, modelling difficulties, and precise tracking. However, its efficiency also depends on numerical synthesis aspects and the applied approximation technique for inverting the digital dynamic model. Their convergence rates depend on the accuracy of the system's dynamics model, which can be error-prone and time-consuming to obtain [23].

Our analysis will be conducted on a typical mechatronic subsystem consisting of a flexible coupling between the motor and the load [24].

ITERATIVE LEARNING CONTROL (ILC) – MODEL INVERSION APPROACH

The integration of the Iterative Learning Control (ILC) algorithm into a closed-loop feedback system is illustrated in Fig.1 (index *j* in Fig. 1 denotes *j*-th repeated task or iteration). After each repetition *j* the ILC signal f_{j+1} for the next repetition is updated by learning from past executions [23]. An ILC system is composed of two dimensions: time (or frequency) domain and iteration domain. The time or frequency domain shows the behaviour of the plant dynamics (*G*) and closed-loop dynamics (see boxed sections in Fig. 1) while the iteration domain demonstrates the learning behaviour of ILC ($f_{j+1}, e_j, \ldots$. i.e. signal dynamics with respect to iteration *j*). An important assumption for the application of the ILC system is that the repetitive tasks in each iteration *j* always begin from the same initial condition. Also, the disturbance effects are the same in each iteration. A key assumption in ILC is that the task of the system is invariant under the repetitions ($r_j = r_{j+1}$). Consequently, the learned signal will be optimal for the specific task (r_j) only.

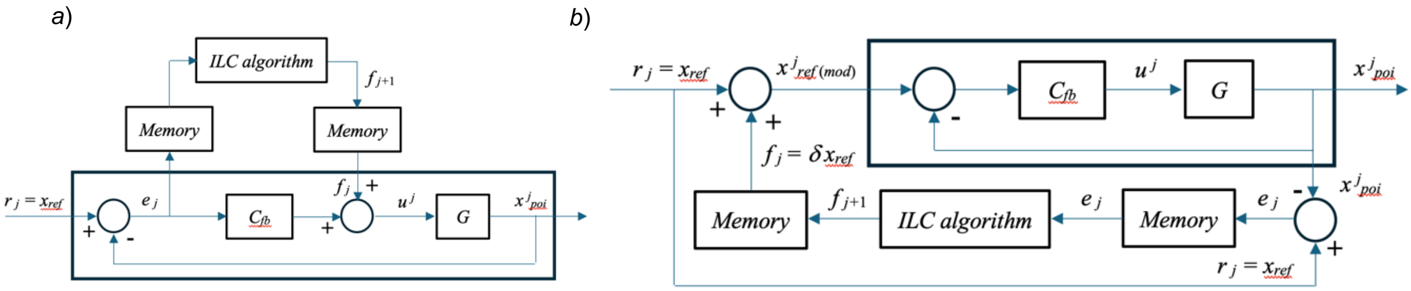

Figure 1 ILC integration to an existing feedback control system: a) Parallel ILC connection, b) Serial ILC connection

The fundamental problem in ILC is to design an update algorithm (for calculation of the ILC signal f_{j+1} in iteration $(j+1)$) such that the tracking error signal (e_{j+1}) of the closed-loop system is minimised in some appropriate sense. A general updating equation is given by

$$f_{j+1}(z) = Q(z)\big(f_j(z) + L(z)E_j(z)\big) \tag{1}$$

where $L(z)$ is the learning filter, and $Q(z)$ is the robustness filter. Filtering using $L(z)$ and $Q(z)$ is performed off-line using non-causal filtering methods; time-shifting is used after filtering with $L(z)$; filtering with $Q(z)$ is based on magnitude characteristics of $|Q(z)|$ only (zero-phase). Typically, $Q(z)$ is a low-pass filter.

In Fig.1, C_{fb} and G denote the feedback controller and the plant, respectively. If

$$S(z) = \left(1 + C_{fb}(z)G(z)\right)^{-1} \tag{2}$$

the sensitivity function,

$$PS(z) = S(z)G(z) \tag{3}$$

is the process sensitivity function,

$$T_{cs}(z) = \left(1 + C_{fb}(z)G(z)\right)^{-1} C_{fb}(z)G(z) \tag{4}$$

is the complementary sensitivity function, the output signals of the systems from Fig. 1(a), and Fig. 1(b) are

$$x^j_{poi}(z) = PS(z)f_j(z) + T_{cs}(z)r(z) \tag{5a}$$
$$x^j_{poi}(z) = T_{cs}(z)f_j(z) + T_{cs}(z)r(z) \tag{5b}$$

Based on Fig. 1 and relations (1) and (5), we obtain

$$E_{j+1}(z) = Q(z)[1 - PS(z)L(z)]\, E_j(z) + [1 - Q(z)]r(z)\text{ , for parallel ILC algorithm} \tag{6a}$$
$$E_{j+1}(z) = Q(z)[1 - T_{cs}(z)L(z)]\, E_j(z) + [1 - Q(z)]r(z)\text{, for serial ILC algorithm} \tag{6b}$$

According to (6a) or (6b), the tracking error converges monotonically if and only if

$$|Q(z)[1 - PS(z)L(z)]| < 1\text{, or } |Q(z)[1 - T_{cs}(z)L(z)]| < 1 \tag{7}$$

at all frequencies f_h with $z = e^{i2\pi f_h T}$ $(\omega T = 2\pi f_h T,\ \ \omega T \in [-\pi,\ \pi])$. Namely, the robustness filter $Q(z)$ should ensure the convergence condition:

$$|Q(i\omega T)[1 - PS(i\omega T)L(i\omega T)]|_{dB} < 0\ \ [dB], \forall \omega T \in [-\pi,\ \pi], \text{ for parallel ILC algorithm} \tag{7a}$$

$$|Q(i\omega T)[1 - T_{cs}(i\omega T)L(i\omega T)]|_{dB} < 0\ \ [dB], \forall \omega T \in [-\pi,\ \pi], \text{ for serial ILC algorithm} \tag{7b}$$

Based on the tracking error propagation equation (6), design the learning filter $L(z)$ should be

$$L(z) \approx \left(S(z)G(z)\right)^{-1} = \left(PS(z)\right)^{-1}, \text{ for parallel ILC algorithm} \tag{8a}$$

$$L(z) \approx \left(T_{cs}(z)\right)^{-1}, \text{ for serial ILC algorithm} \tag{8b}$$

The design of the learning filter $L(z)$ is based on an adequate model inversion. The design of the robustness filter $Q(z)$ is based on convergence condition (7) to ensure the convergence of the ILC algorithm (1). For nonminimum-phase or strictly proper subsystem models, model inversion (8) is not always straightforward. Namely, when the process sensitivity function $PS(z)$ or the complementary sensitivity function $T_{cs}(z)$ (refer to relation (8)) is nonminimum phase, the resulting $L(z)$ becomes unstable. If process sensitivity function $PS(z)$ or the complementary sensitivity function $T_{cs}(z)$ is strictly proper (the order of the denominator polynomial exceeds that of the numerator), $L(z)$ is improper, resulting in non-causal filtering. When implementing the ILC algorithm, non-causal filtering is feasible because it is done offline, and the input signal sequences are fully known. However, calculating the digital learning filter (8) is usually only feasible approximately.

After determining the learning filter $L(z)$, the robustness filter $Q(z)$ is typically set to 1 or a low-pass filter whose frequency bandwidth is reduced until the stability condition of the ILC algorithm (7) is satisfied.

DESIGN EXAMPLE: SUBSYSTEM WITH FLEXIBLE MOTOR – LOAD COUPLING

In mechatronic systems, the motion control subsystem very often involves the coupling of an AC electric motor and the load. Designing controllers and control structures in the field of motion control very often assumes an ideal, rigid transmission train, adopting the following nominal plant model

$$G^0(s) = \frac{\theta_l(s)}{M_{em}(s)} = \frac{1}{(J_m + J_l)s^2} \tag{9}$$

where J_m, J_l are motor and load inertia, M_{em} is control signal or electromagnetic torque, and θ_l is load shaft position. However, in high-performance electric drives, the flexible coupling of the motor axle and the load cannot be ignored. Fig. 2 shows the schematic diagram of a two-mass motor-load system with flexible coupling [24]. In mechatronics subsystem from Fig.2, the motor inertia and load inertia are connected through a shaft or transmission system with a finite stiffness coefficient, c_s. This flexible coupling increases the number of state variables in the mechanical subsystem. The friction coefficient, b_v, is generally low, leading to weakly damped mechanical oscillations. The torsional torque M_o is equal to the load torque M_l only in the steady state.

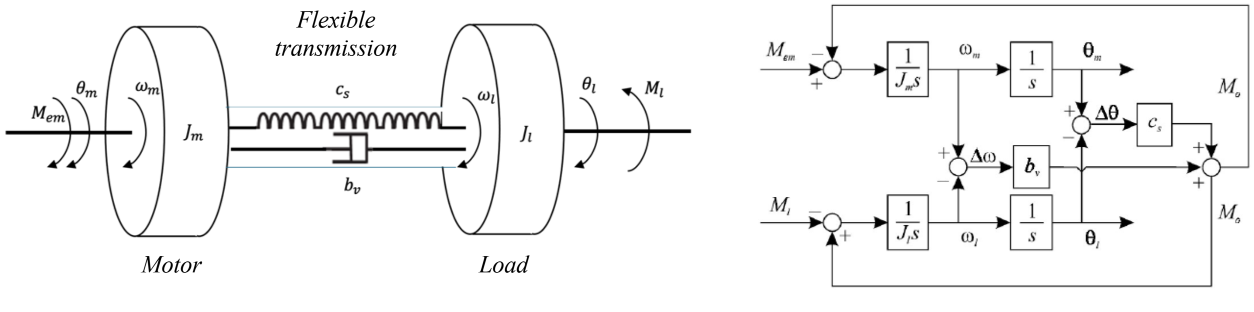

Figure 2 Flexible coupling of motor axle and load [24], [13]

Dynamical model of the mechanical subsystem differs from the simple nominal model (9). If the shaft sensor is mounted *on the motor*, the plant model is

$$G_m(s) = \frac{\theta_m(s)}{M_{em}(s)} = \frac{1}{(J_m + J_l)s^2} \frac{1 + \frac{2\zeta_z}{\omega_z}s + \frac{1}{\omega_z^2}s^2}{1 + \frac{2\zeta_p}{\omega_p}s + \frac{1}{\omega_p^2}s^2} \tag{10}$$

and, if the shaft sensor is mounted *on the load*, the plant model is

$$G_l(s) = \frac{\theta_l(s)}{M_{em}(s)} = \frac{1}{(J_m + J_l)s^2} \frac{1 + \frac{2\zeta_z}{\omega_z}s}{1 + \frac{2\zeta_p}{\omega_p}s + \frac{1}{\omega_p^2}s^2} \tag{11}$$

where *resonance* and antiresonance undamped natural frequencies ω_p and ω_z, and relative damping coefficients ζ_p and ζ_z are given by

$$\begin{aligned} \omega_p &= \sqrt{\frac{c_s(J_m + J_l)}{J_m J_l}}, & \omega_z &= \sqrt{\frac{c_s}{J_l}}, \\ \varsigma_p &= \sqrt{\frac{b_v^2(J_m + J_l)}{4c_s J_m J_l}}, & \varsigma_z &= \sqrt{\frac{b_v^2}{4c_s J_l}} \end{aligned} \tag{12}$$

The subsystem shown in Fig. 2 is commonly encountered in industrial practice. If the controller design is based on models (9) or (10), there is a significant risk of torsional oscillations in the system [24], [13]. Using model (11) is not always realistic because it's difficult to place sensors at the point of interest and because of uncertainties in modeling parameters (12). This type of system also appears in electric vehicles and their transmissions, where using the simple model (9) isn't realistic because it ignores possible nonlinearities and weakly damped of the flexible transmission.

In this paper, we will test the effectiveness of applying ILC algorithms on the subsystem shown in Fig. 2 and Fig.3. We will assume that during the controller design phase, it is possible to mount angular velocity and position sensors at the point of interest on the loaded shaft. During the operational phase, the sensor on the loaded shaft will not be used. The system will use the ILC feedforward signal $f_{j\,(=p)}$ (in Fig. 3, *p* is the number of iterations for which the signal f_j converges) and feedback controller based on the motor shaft signals (θ_m, ω_m). The design of the ILC feedforward signal is based on the load shaft signals (θ_l, ω_l) and the defined reference trajectory $\theta_{ref}(t)$.

Our aim is to control the load shaft position in the absence of a dedicated load side position sensor for motion control systems with flexible coupling. In our simulation analysis, we use a model of flexible coupling between the motor axle and the load as the actual plant model [24]. The important data are as follows (see Fig. 2): J_m=0.000620 kgm^2, J_l=0.000220 kgm^2, c_s=350 Nm/rad, b_v=0.004 Nms/rad. The sample rate is T = 0.5 ms. The desired close-loop system transfer function is specified by undamped natural frequency ω_n=400 rad/s and relative damping coefficient ζ=1.

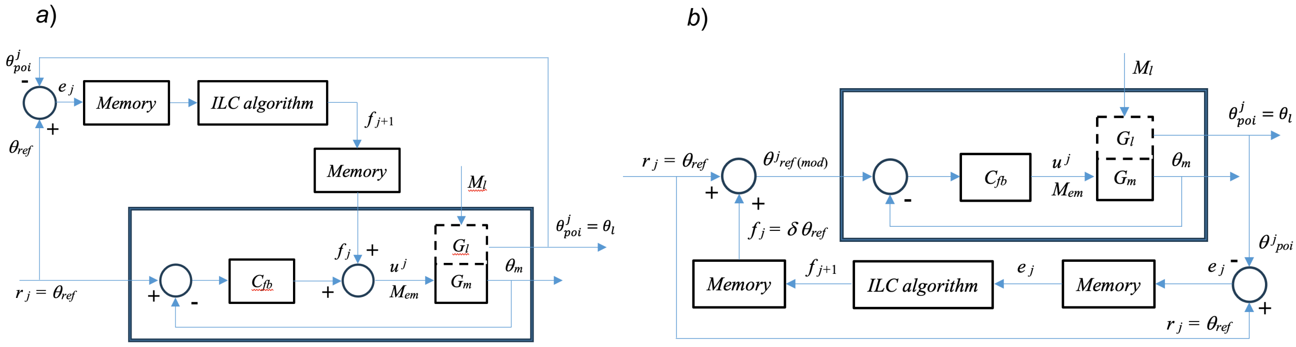

Figure 3 ILC integration to servo control system: a) Parallel ILC connection, b) Serial ILC connection

In accordance with the definition of the sensitivity function

$$
\begin{aligned}
S &= \frac{e}{r} = \frac{u}{d} = \frac{1}{1+HC} \quad (Sensitivity\ function) \\
T_{cs} &= \frac{y}{r} = \frac{y}{\eta} = \frac{HC}{1+HC} \quad (Complementary\ sensitivity\ function) \\
PS &= \frac{y}{d} = \frac{H}{1+HC} = HS \quad (Process\ sensitivity\ function) \\
CS &= \frac{u}{r} = \frac{C}{1+HC} \quad (Control\ sensitivity\ funkction)
\end{aligned}
\tag{13}
$$

related to the general structure of feedback control systems (Fig. 4)

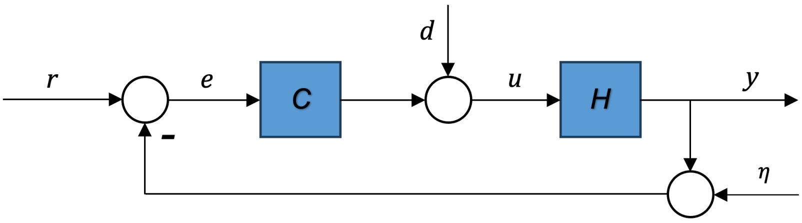

Figure 4 General structure of feedback control systems

we define the sensitivity functions for the system structure in Fig. 3.

$$
\begin{aligned}
S(z) &= \frac{1 + C_{fb}(z)\big(G_m(z) - G_l(z)\big)}{1 + C_{fb}(z)G_m(z)} \\
T_{cs}(z) &= \frac{C_{fb}(z)G_l(z)}{1 + C_{fb}(z)G_m(z)} \\
PS(z) = S(z)G_l(z) &= \frac{1 + C_{fb}(z)\big(G_m(z) - G_l(z)\big)}{1 + C_{fb}(z)G_m(z)} G_l(z)
\end{aligned}
\tag{14}
$$

Equations (14), as well as (2), (3), and (4), are important for the design of the ILC algorithm. The design of the digital feedback controller $C_{fb}(z)$ can be carried out in the frequency domain by shaping the sensitivity functions and according to the criterion that the closed-loop system should achieve a bandwidth of 300 Hz. The feedback controller is obtained

$$
C_{fb}(z) = \frac{2233.3(z - 0.5075)^2(z - 0.7096)(z^2 - 1.474z + 0.9867)}{(z + 0.209)(z^2 - 1.111z + 0.4977)(z^2 - 1.605z + 0.9883)} \tag{15}
$$

It should be noted that designing controller (15) is not a trivial task. The control plant (see Fig. 2) has a bandwidth of approximately 6 Hz and exhibits resonant modes at 233 Hz (see Fig. 5). The controller must be designed to ensure that the closed-loop system achieves a bandwidth of 300 Hz and can respond accurately to step disturbances and step reference signals without steady-state error.

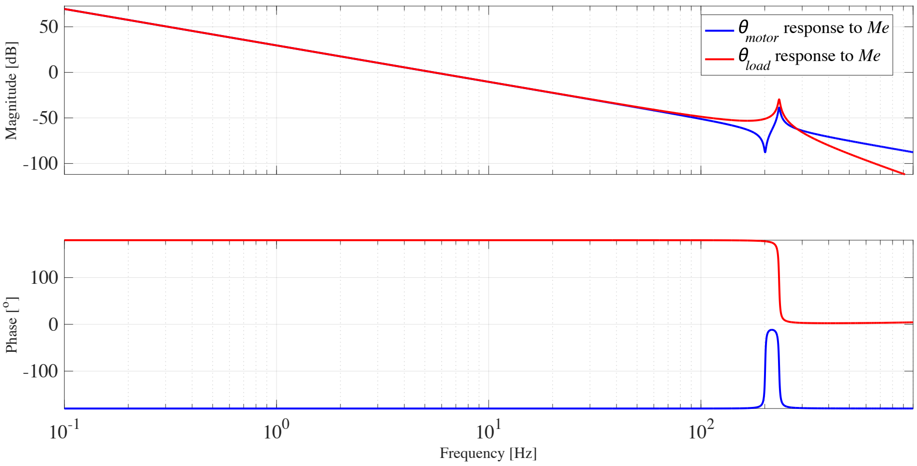

Figure 5 Bode diagram of the controlling plant (flexible coupling of the motor axle and load) shown in Figure 2.

The effects of the operation of a closed-loop system designed in this manner, without the implementation of the ILC algorithm, are shown in Figure 6. The effect of the disturbance - load torque $M_l(t) = h(t - 0.25)$ - was also simulated. The reference trajectory is an S-curve (not a step signal!) with a rise time of 0.05 [s].

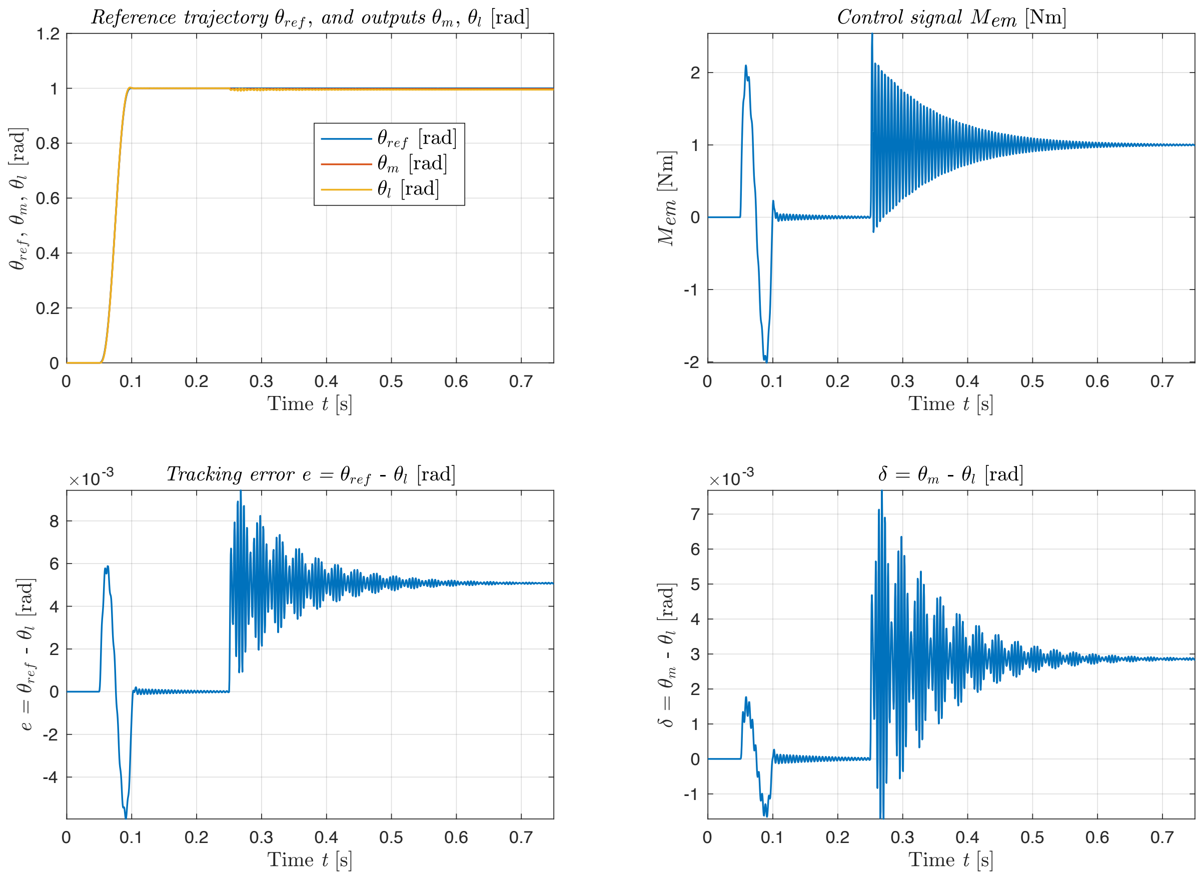

Figure 6 Simulation verification of the feedback controller design for the system shown in Figure 3 (the ILC algorithm has not yet been implemented)

ILC DESING BASED ON ZPETC (Zero Phase Error Tracking Control) METHODOLOGY FOR DYNAMIC MODEL INVERSION

Design of ILC algorithm is based on the control system structure in Fig. 1, learning updating equation (1), convergence condition (7), and relation (8) for calculating the learning filter $L(z)$. For calculation of learning filter $L(z)$ ZPETC methodology can be used [22], [6]. This method is able to achieve "zero-phase error" at all frequencies.

ZPETC (Zero Phase Error Tracking Control) methodology: inversion of the model in the form of a discrete time transfer function

$$\frac{z^{-d}b(z^{-1})}{a(z^{-1})}=\frac{z^{-d}b_s(z^{-1})b_u(z^{-1})}{a(z^{-1})} \tag{16}$$

where d is integer value representing transport time-delay; $a(z^{-1})$ is a polynomial with zeros within and eventually on the unit disc (stable and marginally stable poles); $a(z^{-1})\neq 0$; $b_s(z^{-1})$ is a polynomial with zeros within the unit disc (stable zeros); $b_u(z^{-1})$ is a polynomial with zeros on and outside the unit disc (marginally stable and unstable zeros); is based on design

$$L(z,z^{-1})=\frac{z^d a(z^{-1})b_u(z)}{\beta b_s(z^{-1})} \tag{17}$$

such that $L(z,z^{-1})PS(z^{-1})=\frac{b_u(z)b_u(z^{-1})}{\beta}$ or $L(z,z^{-1})T_{cs}(z^{-1})=\frac{b_u(z)b_u(z^{-1})}{\beta}$; where $\frac{b_u(z)b_u(z^{-1})}{\beta}$ has zero phase, and β is chosen such that $\frac{b_u(z)b_u(z^{-1})}{\beta}\approx 1$ (for z=1). Namely, $\beta=(b_u(1))^2$. Note that both $L(z,z^{-1})PS(z^{-1})$ and $L(z,z^{-1})T_{cs}(z^{-1})$ are finite-impulse-response (FIR) filters

In our illustrative example (see Fig. 2 and Fig. 3), the relevant sensitivity functions (3) and (4), or (14), contain nonminimal phase zeros, which means that the calculation of the learning filter L (8) can only be approximative, as the exact inversion of (14) is not feasible. It concerns the fact that the continuous transfer function (11)

$$G_l(s)=\frac{29326(s+8.75e04)}{s^2(s^2+24.63s+2.155e06)} \tag{18}$$

after discretization (with a sampling time of 0.5 [ms]), becomes non-minimal phase (NMP) discrete transfer function

$$G_l(z)=\frac{7.1402e-06(z+8.956)(z+0.9539)(z+0.09688)}{(z-1)^2(z^2-1.476z+0.9878)} \tag{19}$$

As a consequence of discretization with a small sampling time, non-minimal phase (NMP) characteristics appear in the transfer function $G_l(z)$ (which contains unstable zero at $z=-8.956$), and consequently, in the complementary sensitivity function $T_{cs}(z)$ (14), as well as in the process sensitivity function $PS(z)$ (14). According to (8), both serial and parallel ILC need to address the same modeling inaccuracies when inverting $T_{cs}(z)$ and $PS(z)$. After applying the ZPETC algorithm to calculate (8), we evaluate the efects of the modeling error in the inverse transfer function by checking the robustness criteria (7), which assesses the convergence of the ILC algorithm.

In Figure 7, the robustness curves must be below 0 [dB] for the ILC algorithm to converge. A lower curve means the learning filter L is more accurate. Since we used the ZPETC method on the sensitivity functions $T_{cs}(z)$ and $PS(z)$, which both have the same non-minimum phase part, the robustness curves are the same. The calculation error becomes from the non-minimum phase part of a dynamic model. The inverse models $\left(PS(z)\right)^{-1}$ and $\left(T_{cs}(z)\right)^{-1}$ contain the same calculation error.

It is possible to design and use a robustness filter Q based on relation (7) to correct the results shown in Figure 7, if needed. However, the modeling error in the inverse sensitivity functions might be so large that designing a robustness filter to ensure ILC algorithm convergence may not be feasible. Using a robustness filter also affects the system's dynamic properties.

Using the ILC algorithm, specifically the ILC signal developed through the iterative learning process (Fig. 9), the tracking error norm decreases by 450 million times (Fig. 9) after 10 iterations of learning. Comparing the results from Fig. 8 and Fig. 6 clearly highlights the advantages of using ILC control.

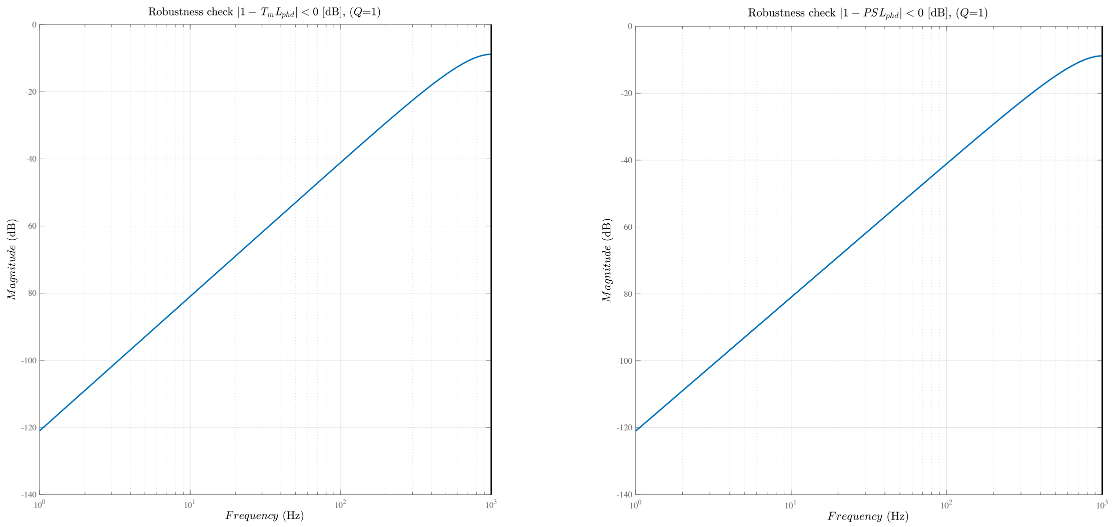

Figure 7 Checking if the approximately calculated inverse sensitivity functions $\left(\boldsymbol{T_{cs}(z)}\right)^{-1}$ and $\left(\boldsymbol{PS(z)}\right)^{-1}$ using the ZPETC method ensure the convergence of the ILC algorithm

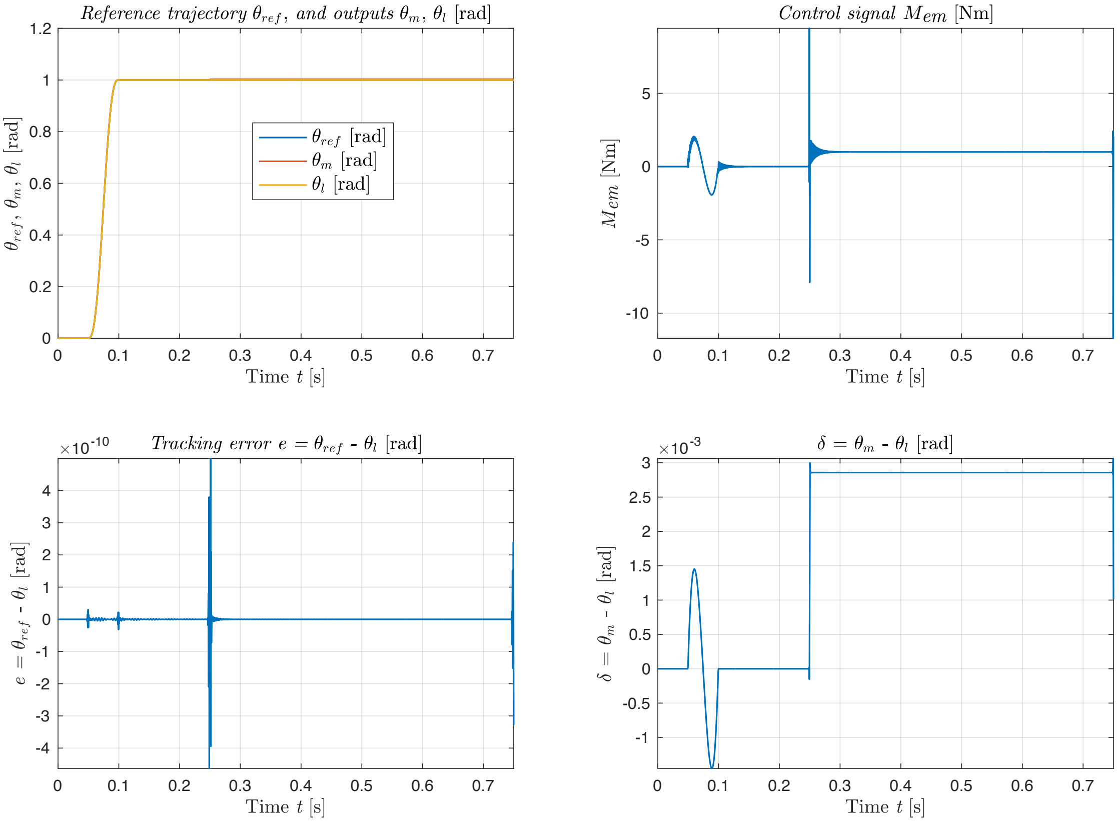

Figure 8 Simulation verification of the parallel ILC algorithm (Fig. 3) using the ZPETC methodology for model inversion (the result refers to the ILC signal after 10 iterations of learning)

It is important to note that using the ILC control algorithm also affects the suppression of oscillations in systems with flexible coupling. This is not always the case, and the structures shown in Fig. 3 do not yield the same results, but this consideration is beyond the scope of this paper. The presented results effectively illustrate the effects and potential of using ILC algorithms. Suppressing torsional oscillations in systems with flexible couplings is of great practical importance.

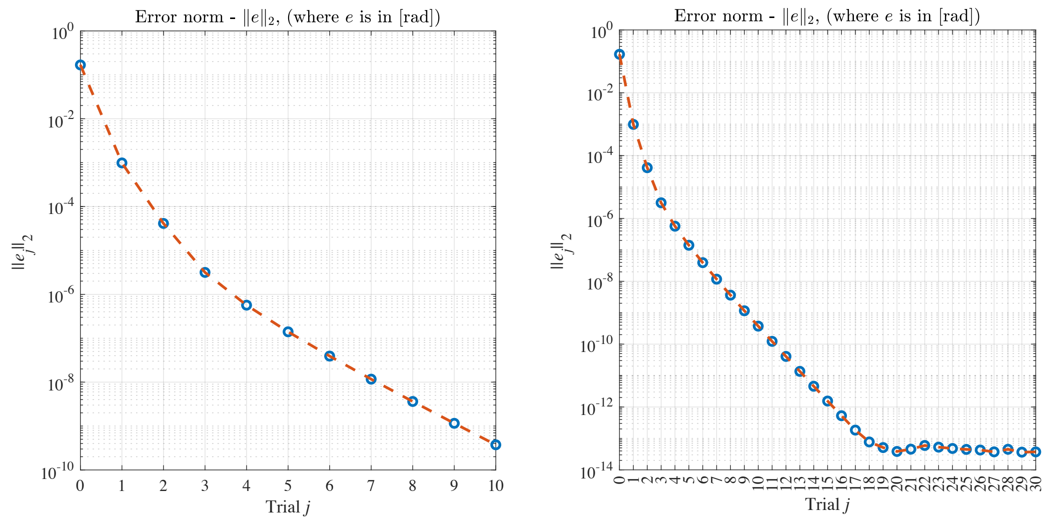

Figure 9 Illustration of iterative learning of the ILC signal and its effect when applied to systems with parallel connection of the ILC signal (Figure 3)

ILC DESING BASED ON ZMETC (Zero Magnitude Error Tracking Control) METHODOLOGY FOR DYNAMIC MODEL INVERSION

Unlike the ZPETC method, which turns the model's NMP zeros into stable zeros in the approximate inverse, the ZMETC technique changes the model's NMP zeros into stable poles in the approximate inverse. This method is able to achieve "zero-magnitude error" at all frequencies. The inverse model performs as feedforward controller that compensates the magnitude loss and phase delay of proces sensitivity function $PS(z^{-1})$ or complementary sensitivity function $T_{cs}(z^{-1})$.

ZMETC (Zero Magnitude Error Tracking Control) methodology: inversion of the model in the form of a discrete time transfer function (13) is filter

$$L(z,z^{-1}) = \frac{z^d a(z^{-1})}{b_s(z^{-1})b_u(z)} \tag{20}$$

such that $L(z,z^{-1})PS(z^{-1}) = \frac{b_u(z^{-1})}{b_u(z)}$ or $L(z,z^{-1})T_{cs}(z^{-1}) = \frac{b_u(z^{-1})}{b_u(z)}$; where the transfer function $\frac{b_u(z^{-1})}{b_u(z)}$ creates a pole-zero pairing that relates to a discrete-time representation of a Padé approximation and represents a delay model. No compensation for changes in DC gain is required for the ZMETC method because the DC gain of $\frac{b_u(z^{-1})}{b_u(z)}$ remains at unity by design. Unlike the NPZ-Ignore and ZPETC methods, the ZMETC technique results in an overall transfer function that is an infinite-impulse-response (IIR) filter [5].

According to [5], for systems with RHP NMP zeros near the unit circle and the real axis, ZMETC is a very good choice.

The simulation results shown in Fig. 10 and Fig. 11 are inferior compared to those obtained using the ZPETC method (see Fig.8 and Fig.9). The modeling error of $\left(T_{cs}(z)\right)^{-1}$ and $\left(PS(z)\right)^{-1}$ is unacceptable above 180 Hz (Fig. 10). Therefore, a robustness filter Q is introduced, chosen as a first-order Butterworth filter with a cutoff frequency of 180 Hz. This robustness filter ensures the convergence of the ILC algorithm but affects the system's dynamic properties (as shown in Fig. 11). The slope of the robustness curve in Fig. 10, at 20 dB/decade, suggests using a first-order low-pass filter for Q. Since the convergence criterion curve has a cutoff frequency of 180 Hz, the maximum frequency bandwidth for the robustness filter is 180 Hz. To achieve the best dynamic properties, the robustness filter Q should be of the lowest possible order and the highest possible cutoff frequency. The system's performance depends on the dynamic characteristics of the controlled plant and the limitations in designing the control structure for that plant. The ILC algorithm improves system performance (Fig. 10). However, the oscillations in the system are not significantly suppressed, and the improvement in servo system accuracy is limited. Nevertheless, the tracking error norm is reduced by 5 times using ILC control and ZMETC methodology for inversion dynamical models.

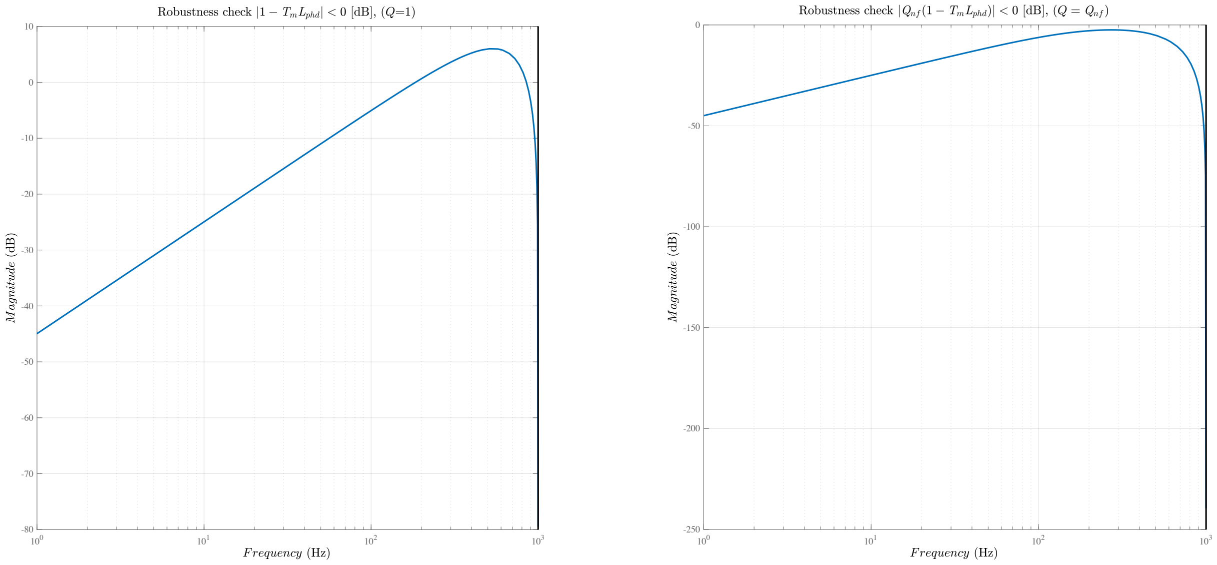

Figure 10 Checking if the approximately calculated inverse sensitivity functions $\left(T_{cs}(z)\right)^{-1}$ and $\left(PS(z)\right)^{-1}$ using the ZMETC method ensure the convergence of the ILC algorithm

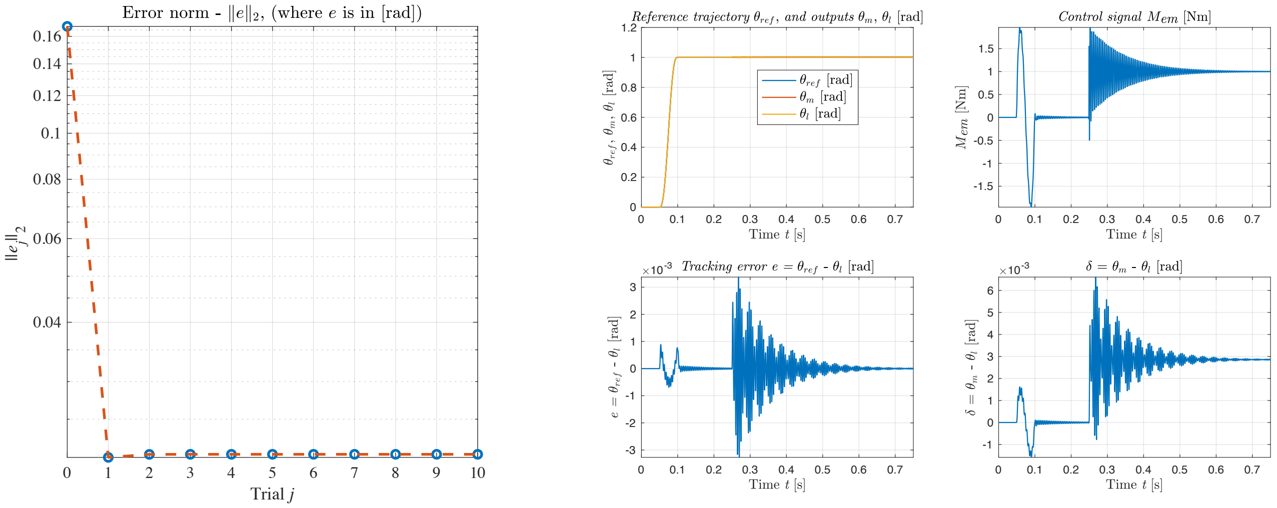

Figure 11 Simulation verification of the parallel ILC algorithm (Fig. 3) using the Z***M***ETC methodology for model inversion (the result refers to the ILC signal after 10 iterations of learning)

ILC DESING BASED ON NPZ-Ignore (Nonminimum-Phase Zeros Ignore) METHODOLOGY FOR DYNAMIC MODEL INVERSION

The NPZ-Ignore technique for model inversion ignores nonminimum-phase dynamics, i.e., NMP zeros in the system model, and makes the proper adjustments for how this might affect the DC gain of the overall system [4].

NPZ-Ignore (Nonminimum-Phase Zeros Ignore) methodology: inversion of the model in the form of a discrete time transfer function (13) is filter

$$L(z, z^{-1}) = \frac{z^d a(z^{-1})}{\beta b_s(z^{-1})} \qquad (21)$$

such that $L(z, z^{-1})PS(z^{-1}) = \frac{b_u(z^{-1})}{\beta}$ or $L(z, z^{-1})T_{cs}(z^{-1}) = \frac{b_u(z^{-1})}{\beta}$; where $\frac{b_u(z^{-1})}{\beta}$ is a finite-impulse-response (FIR) filter which ignores NMP dynamics. Parameter β is chosen such that $\frac{b_u(z^{-1})}{\beta} \approx 1$ ($\beta = b_u(1)$). It is a tuning parameter and typically used to compensate for DC gain by setting $\beta = b_u(1)$.

Recalling that ideally $L(z,z^{-1})PS(z^{-1}) = 1$ or $L(z,z^{-1})T_{cs}(z^{-1}) = 1$ at all frequencies, but if there are nonminimum phase zeros, there will be an error in both magnitude and phase. Namely, we can note that that ZPETC has zero phase error, ZMETC has zero magnitude error, and NPZ-Ignore has both a magnitude and phase error.

Applying the NPZ-Ignore methodology for model inversion in our example gives better results than the ZMETC methodology but is worse than the ZPETC methodology. Figure 12 clearly shows that the robustness filter *Q* should be a first-order low-pass filter with a cutoff frequency of 375 Hz. Suppressing torsional oscillations in a flexible system is more effective than when using the ZMETC methodology in designing ILC control. In general, the ability to implement a robustness filter *Q* with a higher bandwidth and lower order will contribute to better suppression of disturbances' effects and oscillations within the system, and reduce the tracking error of the reference trajectory.

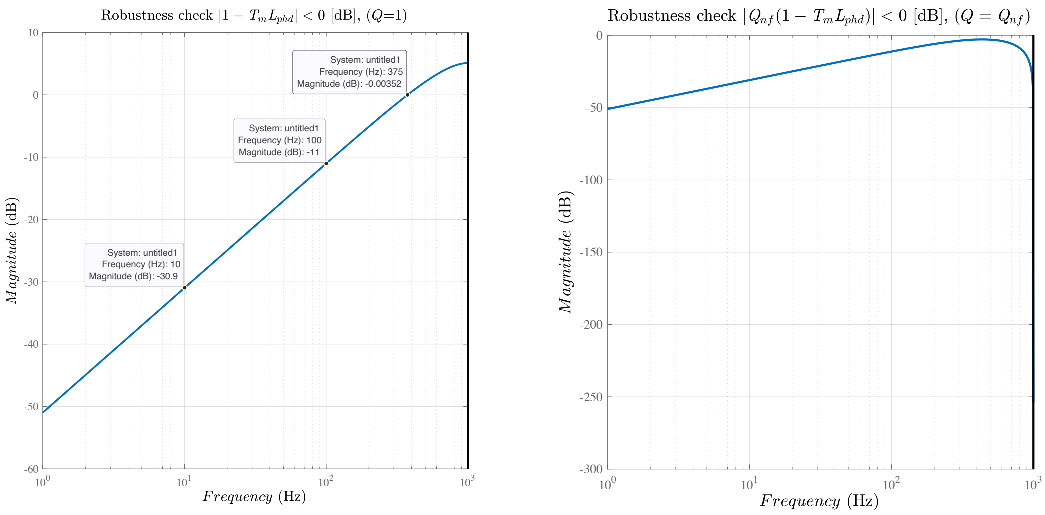

Figure 12 Checking if the approximately calculated inverse sensitivity functions $\left(T_{cs}(z)\right)^{-1}$ and $\left(PS(z)\right)^{-1}$ using the NPZ-Ignore method ensure the convergence of the ILC algorithm

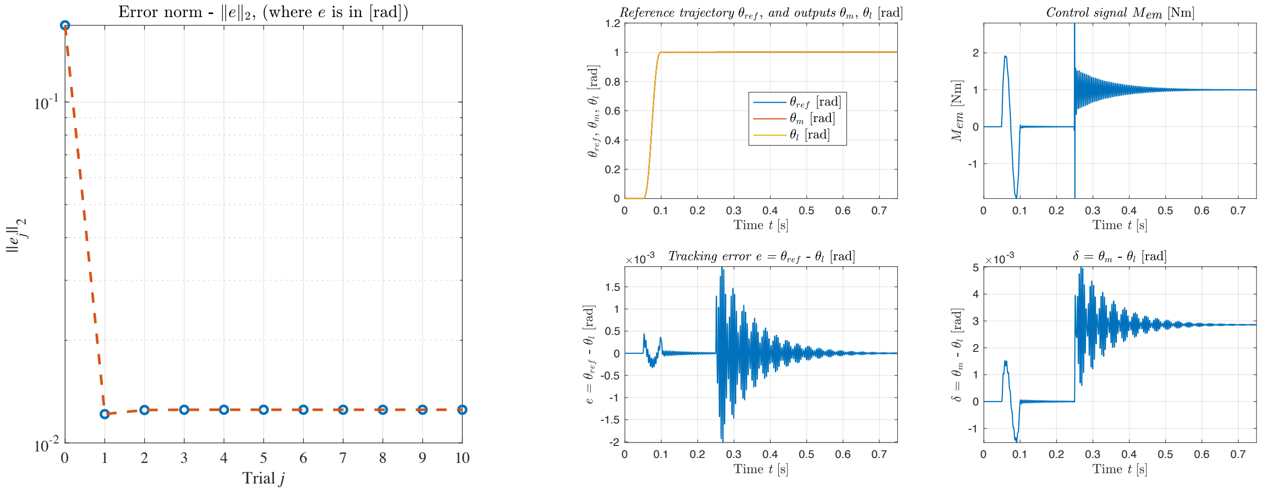

Figure 13 Simulation verification of the parallel ILC algorithm (Fig. 3) using the NPZ-Ignore methodology for model inversion (the result refers to the ILC signal after 10 iterations of learning)

ILC DESING BASED ON OPTIMAL THEORY CALCULATION APPROACHES AND DATA-DRIVEN DYNAMIC MODEL INVERSION

Based on relations (6), (7), and (8), we can form a cost function

$$J\big(L(z)\big) = \int_0^{2\pi} |1 - PS(z)L(z)|^2 \, d\theta,\ z = e^{i\theta}, \text{for parallel ILC algorithm} \tag{22a}$$

$$J\big(L(z)\big) = \int_0^{2\pi} |1 - T_{cs}(z)L(z)|^2 \, d\theta,\ z = e^{i\theta}, \text{for serial ILC algorithm} \tag{22b}$$

where $\theta = \omega T$ is digital frequency ($\omega T = 2\pi\, f_h\, T,\ \ \omega T \in [-\pi,\ \pi]$). Namely, at all frequencies f_h [Hz], $J\left(L(e^{2\pi\, f_h\, T i})\right)$ should be zero. Minimization of the cost function (22) should provide the solution for the learning filter $L(z)$. On the other hand, the quality of the approximate solution for model inversion (8) could be evaluated through the performance index (22).

Loop shaping methods in the frequency domain for determining the learning filter $L(z)$ are also possible based on equation (8)

$$\left|PS(e^{\theta i})L(e^{\theta i})\right|_{dB} = 0\ \ [dB], \forall\theta \in [-\pi,\ \pi], \text{for parallel ILC algorithm} \tag{23a}$$

$$\left|T_{cs}(e^{\theta i})L(e^{\theta i})\right|_{dB} = 0\ \ [dB], \forall\theta \in [-\pi,\ \pi], \text{for serial ILC algorithm} \tag{23b}$$

Designing the learning filter $L(z)$ based on equation (8) is equivalent to designing a feedforward controller described by the inverse of the system dynamics (in our case – process sensitivity function or complementary sensitivity function). If it is not possible to calculate the learning filter $L(z)$ exactly according to equation (8), the calculation criterion for finding $L(z)$ can be the minimization of the H_∞ norm

$$J(L) = \left\|1 - PS(e^{\theta i})L(e^{\theta i})\right\|_\infty,\ \theta \in [-\pi,\ \pi], \text{for parallel ILC algorithm} \tag{24a}$$

$$J(L) = \left\|1 - T_{cs}(e^{\theta i})L(e^{\theta i})\right\|_\infty,\ \theta \in [-\pi,\ \pi], \text{for serial ILC algorithm} \tag{24b}$$

or minimization of the H_2 norm

$$J(L) = \left\|1 - PS(e^{\theta i})L(e^{\theta i})\right\|_2,\ \theta \in [-\pi,\ \pi], \text{for parallel ILC algorithm} \tag{25a}$$

$$J(L) = \left\|1 - T_{cs}(e^{\theta i})L(e^{\theta i})\right\|_2,\ \theta \in [-\pi,\ \pi], \text{for serial ILC algorithm} \tag{25b}$$

$$L(z, z^{-1}) = \arg\min_L J(L) \tag{26}$$

In each of these cases, it is necessary to check the convergence criterion and introduce the robustness filter $Q(z)$ if needed.

Thus, there are also optimal inversion-based methods that minimize an objective function ($J(L)$), as well as model inversion-based methods. According to [24] the following optimal inversion-based methods exist: the norm-optimal feedforward approach, the H_∞ preview approach, and the H_2 preview approach.

The model-based inversion approach has two main issues: 1) the inverse input can be inaccurate if there is significant modelling uncertainty, and 2) the inverse input may be unacceptable if it exceeds input energy or bandwidth limits. The optimal-inversion technique has been developed to address these two problems by focusing on data-driven solutions related to the issue of dynamic model inversion.

CONCLUSIONS

Iterative Learning Control (ILC) is a powerful control technique used to enhance dynamic accuracy in high-performance mechatronic systems in manufacturing, where processes repeat under the same conditions for a finite time duration. A key part of designing ILC algorithms involves inverting dynamic models. However, when the dynamic model has non-minimum phase (NMP) zeros, exact model inversion isn't possible. NMP zeros in a discrete-time model can arise from sampling a continuous-time system, from sensors and actuators being physically separated, or from unaccounted transport delays. These factors make NMP zeros common in discrete-time systems. The challenge of stable approximate inversion of NMP dynamic models can be addressed using model-based and optimal-based techniques.

In this paper, we explore stable approximate model-inverse control techniques for more effective ILC control design. We explain a basic concept of designing ILC algorithm and select an illustrative example commonly found in manufacturing practice. Flexible motor-load couplings are frequently seen in industrial machinery, electric vehicles, and various mechatronic systems. We discuss the specifics of designing ILC control for this system, providing examples of learning filter calculations using the ZPETC, ZMETC, and NPS-Ignore model-inversion techniques. The paper also defines optimal approaches for designing the learning filter *L*. The choice between NPZ-Ignore, ZPETC, or ZMETC techniques for maximizing performance depends largely on the system being applied to, as their effectiveness varies based on the location of NMP zeros.

The paper includes a detailed explanation and simulation verification of the ILC control synthesis methods presented. The simulation conditions considered the impact of disturbances on the system, the presence of actuator saturation, and the system's tendency to develop oscillations. It was demonstrated that the ILC algorithm can significantly improve system performance in terms of dynamic accuracy, as well as in suppressing disturbances' effects and oscillations within the system. However, modeling errors in the inverse dynamics can significantly limit system performance.

REFERENCES

[1] Ahn H.S., Chen Y., Moore K.L.: “Iterative learning control: Brief survey and categorization”, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 37, No. 6, 2007, 1099-1121.

[2] Bolder J.J.: “Flexibility and robustness in iterative learning control: with applications to industrial printers”, PhD Thesis, TU/e, NL., 2015.

[3] Bristow D.A., Tharayil M., Alleyne A.G.: “A survey of iterative learning control: A learning-based method for high-performance tracking control”, IEEE Contr. Syst. Mag. Vol. 26, 2006, 96-114.

[4] Butterworth J.A., Pao L.Y., Abramovitch D.Y.: “Architectures for tracking control in atomic force microscopes”, In: Proc IFAC world congress, 2008, 8236–50.

[5] Butterworth J., Pao L., Abramovitch D.: “Analysis and comparison of three discrete time feedforward model-inverse control techniques for nonminimum-phase systems”, Mechatronics, Vol. 22, No. 5, 2012, 577–87.

[6] Gross E., Tomizuka M., Messner W.: “Cancellation of discrete time unstable zeros by feedforward control”, J. Dyn. Syst. Meas. Control, Vol. 116, No. 1, 1994, 33–8.

[7] Haack B., Tomizuka M.: “The effect of adding zeroes to feedforward controllers”, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 113, 1991, 6-10.

[8] Hätönen J.: “Issues of algebra and optimality in iterative learning control”, PhD Thesis, Faculty of Technology, University of Oulu, Finland, 2004.

[9] Landers R.G., Barton K., Devasia S., Kurfess T., Pagilla P., Tomizuka M.: “A Review of Manufacturing Process Control”, J. Manuf. Sci. Eng., Vol. 142, No. 11, 2020.

[10] Liu L., Tian S., Xue D., Zhang T., Chen Y.: “Industrial feedforward control technology: a review”. Journal of Intelligent Manufacturing, Vol. 30, 2019, 2819–2833.

[11] Koçan O.: “Feedback via iterative learning control for repetitive systems”, PhD Thesis, ISAE, France, 2020.

[12] Kuo TT and Tsao TC. (2015) A comparison of inversion based iterative learning control algorithms. In: *American Control Conference (ACC).* Chicago, IL USA, pp. 3564-3569.

[13] Matijević M., Sredojević, Stojanović M.V.: “Robust RST controller design by convex optimization”, Electronics, Vol. 15, No. 1, 2011, invited paper, 25-29.

[14] Norrlöf M.: “Iterative learning control – analysis, design, and experiments”, PhD Thesis, Linkoping University, Sweden, 2000.

[15] Owens H.D.: “Iterative learning control – An optimization paradigm”, Springer-Verlag London, 2016.

[16] Poot M.: “Gaussian processes for learning in motion control: Applied to semiconductor back-end machines”, PhD Thesis, Eindhoven University of Technology, Eindhoven, The Neatherlands, 2024.

[17] Poot M., van Haren M., Kostic D., Portegies J., Oomen T.: “Position-dependent motion feedforward via Gaussian processes: Applied to snap and force ripple in semiconductor equipment”, IEEE Transactions on Control Systems Technology, 2024, 1-16

[18] Rigney B., Pao L., Lawrence D.: "Nonminimum Phase Dynamic Inversion for Settle Time Applications," *in* IEEE Transactions on Control Systems Technology*,* Vol. 17, No. 5, 2009, 989-1005.

[19] Shen D., Wang Y.: (2014) “Survey on stochastic iterative learning control”, Journal of Process Control*,* Vol. 24, No. 12, 2014, 64–77.

[20] Shen D.: “Iterative learning control with incomplete information: a survey”, IEEE/CAA Journal of Automatica Sinica Vol. 5, No. 5, 2018, 885-901.

[21] Torfs D, De Schutter J, Swevers J.: “Extended bandwidth zero phase error tracking control of nonminimal phase systems”, J Dyn Syst Meas Control., Vol. 114, No. 3, 1992, 347–51.

[22] Tomizuka M. (1987) “Zero phase error tracking algorithm for digital control”, J Dyn Syst Meas Control, Vol. 109, No. 1, 1987, 65–68.

[23] van Zundert J., Oomen T.: “On inversion-based approaches for feedforward and ILC”, Mechatronics, Vol. 50, 2018, 282–291.

[24] Vukosavic S., Stojic M., “Suppression of torsional oscillations in a high-performance speed servo drive”, IEEE Trans. on Industrial Electronics, Vol. 45, No. 1, 1998, 108-117.

[25] Wang Y., Gao F., Doyle F.J.: “Survey on iterative learning control, repetitive control, and run-to-run control”, J Proc Cont., Vol. 19, 2009, 1589–1600.

[26] Wen J.T., Potsaid B.: “An experimental study of a high performance motion control system”, Proceedings of the 2004 american control conference, Boston,Massachusetts. 2004. p. 5158–63.