Iterative Control for Periodic Tasks with Robustness Considerations, Applied to a Nanopositioning Stage ⋆

Robin de Rozario * **Andrew J. Fleming** ** **Tom Oomen** *

* *Eindhoven University of Technology, Department of Mechanical engineering, P.O. Box 513, 5600 MB Eindhoven (e-mail: r.d.rozario@tue.nl, t.a.e.oomen@tue.nl)*
** *Precision Mechatronics Lab, School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, N.S.W. 2308, Australia (andrew.fleming@newcastle.edu.au)*

Abstract: Nanopositioning stages are an example of motion systems that are required to accurately perform high frequency repetitive scanning motions. The tracking performance can be significantly increased by iteratively updating a feedforward input by using a nonparametric inverse plant model. However, in this paper it is shown that current approaches lack systematic robustness considerations and are suffering from limited design freedom to enforce satisfying convergence behavior. Therefore, inspired by the existing Iterative Learning Control approach, robustness is added to the existing methods to enable the desired convergence behavior. This results in the Robust Iterative Inversion-based Control method, whose potential for superior convergence is experimentally verified on a Nanopositioning system.

Keywords: Periodic motion, Iterative methods, Robustness, Nanopositioning, Nonparametric

1. INTRODUCTION

The demand for increased accuracy and speed in precision applications has led to the adoption of dedicated feedforward control inputs (Devasia et al., 2007; Pipeleers et al., 2009). For example, in Atomic Force Microscopy, a sample is moved relative to a probe with nanometer resolution and scan frequencies reaching hundreds of Hertz. Since the resonance frequencies of the stages are not significantly higher than desired scan rate, achieving an acceptable bandwidth is a non-trivial task, as argued in Fleming and Leang (2014). Fortunately, since the setpoint is a periodically repeating trajectory, the desired scanning speeds can be achieved by iteratively updating a feedforward input signal as is shown in Tien et al. (2005). For such an iterative control method to be successful, it should generally posses the following three features;

(1) the converged performance should meet the desired level of accuracy;
(2) the convergence speed of the iterative solution should be within an acceptable number of iterations;
(3) and the algorithm should be robust against perturbations to the controlled plant.

Several iterative methods have been proposed that aim to increase the tracking performance of motion systems that perform periodic tasks, while satisfying these criteria. In Tien et al. (2005) the Iterative Inversion-based Control (IIC) method was proposed that employs an inverse plant model to iteratively update the feedforward input. This approach is reported to lead to a significant increase in tracking performance in case the plant model is sufficiently accurate. However, it was argued that the convergence rate can be prohibitively slow in case the model mismatch is large. Consequently, an extension to this method (EIIC) is made in Zou et al. (2007), which relaxes the convergence criteria to some extent but the restrictions imposed by the required accuracy of the inverse plant model remain. The Model-less IIC (MIIC) as approach, as presented in Kim (2008) aims to remove the convergence criteria completely by estimating the inverse plant model in the iterative process. Promising results are reported in Bechhoefer (2008) for the case in which the effects of nonlinearities such as hysteresis are limited.

Although these important developments in IIC have led to significantly increased tracking performance, robustness aspects are not considered systematically. Moreover, the connection to Repetitive Control (RC) and Iterative Learning Control (ILC) is not yet established and with that, similar design guidelines have been left largely unformulated. Therefore, the aim of this paper is to fill this gap and experimentally verify the proposed approach.

In this paper, RC, ILC and IIC are formulated in a general lifted signal description. This unified formulation shows that the IIC is very similar to ILC and consequently the ILC design guidelines can be modified to systematically treat robustness and convergence rate aspects

⋆ This work is supported by the Innovational Research Incentives Scheme under the VENI grant Precision Motion: Beyond the Nanometer (no. 13073) awarded by NWO (The Netherlands Organisation for Scientific Research) and STW (Dutch Science Foundation).

in IIC methods. Thus, this paper contains the following contributions.

C1. In section 2 a systematic comparison of the RC, ILC and IIC methods in a unifying description is provided.
C2. In section 3 the IIC method is extended, based on existing ILC results and design guidelines are provided.
C3. In section 4 the proposed method is experimentally validated on a nanopositioning system.

2. A UNIFYING ANALYSIS

In this section, the lifted signal description as first presented in Bamieh et al. (1991) and the problem of periodic disturbance rejection are introduced, which enable a unifying time-domain analysis of the RC, ILC and IIC methods.

2.1 Rejection of periodic disturbances in a lifted framework

A linear time invariant (LTI), single-input-single-output (SISO), stable discrete time system, $G(z)$ is considered, where $z \in \mathbb{C}$ is a complex indeterminate. The evolution of the output $y(k) \in \mathbb{R}$, subjected to the input $u(k) \in \mathbb{R}$ and initial state $x(t_0) = x_0$, is described by the state space equations,

$$x(k+1) = Ax(k) + Bu(k), \tag{1a}$$
$$y(k) = Cx(k) + Du(k), \tag{1b}$$

where (A, B, C, D) is a realization of G and $x(k) \in \mathbb{R}^n$ is the state vector. Now define N as the number of samples of a single trial and denote the trial number by i. A lifted signal $\bar{s}$ is now defined as the discrete time sequence of the signal $s(k)$ during a single trial, whose elements are stored in a column, i.e.,

$$\bar{s}_i \triangleq \begin{bmatrix} s(Ni) \\ s(Ni+1) \\ \vdots \\ s(Ni+N-1) \end{bmatrix}. \tag{2}$$

By evaluating the state space equations as given by (1a) and (1b), it can be found that the lifted system G_l can be represented as,

$$G_l \triangleq \begin{cases} x(Ni+N) = Fx(Ni) + G\bar{u}_i, & \text{(3a)} \\ \bar{y}_i = Hx(Ni) + J\bar{u}_i, & \text{(3b)} \end{cases}$$

$$\left[\begin{array}{c|c} F & G \\ \hline H & J \end{array}\right] = \left[\begin{array}{c|cccc} A^N & A^{N-1}B & \dots & AB & B \\ \hline C & h(0) & 0 & \dots & 0 \\ CA & h(1) & h(0) & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ CA^{N-1} & h(N-1) & \dots & h(1) & h(0) \end{array}\right], \tag{4}$$

where $h(k)$ are the Markov parameters,

$$h(k) = \begin{cases} D & k = 0 \\ CA^{k-1}B & k \geq 1 \end{cases}. \tag{5}$$

This system representation is key in the analysis of periodic signals that are periodic with N samples. Note that for these signals holds that $\bar{s}_i = \bar{s}\ \forall j$, since $s(k) = s(k+N)\ \forall k$. The problem of Iterative Periodic Disturbance Rejection (IPDR) is now formulated as follows.

Problem 1. (IPDR, time domain). Consider a periodic disturbance $\bar{r}$ and define the lifted tracking error as,

$$\bar{e}_i \triangleq \bar{r} - \bar{y}_i. \tag{6}$$

Then, (iteratively) find $\bar{u}_i$ such that $\lim_{i\to\infty} \bar{e}_i = 0$.

2.2 Repetitive Control

The RC method aims to solve the IPDR problem in case the state x simply results from the past inputs during the previous trial as is reflected by equation (3a). This is typically the case for systems that perform continuous periodic tasks such as hard disk drives and power electronics. Hence, the RC method aims to provide $\bar{u}_i$ in case $\bar{e}_i$ is given by (3a), (3b) and (6). See for example Hara et al. (1988) and Roover et al. (2000).

2.3 Iterative Learning Control

The ILC approach considers the case for which the state x resets after each trial, i.e. $x(Ni) = \hat{x}$, which can be assumed to be zero without loss of generality, see for example Bristow et al. (2006). This is typically the case for batch-to-batch processes such as pick-and-place and printing tasks. In this case, equations (3a) and (3b), combined with (6) reduce to,

$$\bar{e}_i = \bar{r} - J\bar{u}_i, \tag{7}$$

where J is also known as the discrete impulse response matrix. The ILC method aims to provide $\bar{u}_i$ by iteratively updating the input as,

$$\bar{u}_{i+1} = Q\bar{u}_i + L\bar{e}_i. \tag{8}$$

Here, L is the learning filter which is often taken to be such that it approximates the inverse of J, while considering causality and minimum-phase aspects. Robustness is introduced by sensibly shaping Q in which case it is no longer equal to the unity matrix, see for example Tousain et al. (2001), van de Wijdeven et al. (2009).

2.4 Iterative Inversion-based Control and related methods

The IIC and related approaches aim to solve the IPDR problem by means of a frequency domain approach. In these methods, it is assumed that the input signal u is periodic with a periodicity of N samples, i.e., $u(k) = u(k+N)\ \forall k$, and this signal has been driving the system for an infinite time from an arbitrary initial condition. Consequently, the periodic output y is in steady state and is given in the frequency domain as,

$$Y(\omega) = G(e^{j\omega})U(\omega), \quad \omega \in \Omega, \tag{9}$$
$$\Omega = \left\{ \omega \in \mathbb{R} \;\middle|\; \omega = \tfrac{2\pi}{N}k,\ k = 0, \dots, N-1 \right\}. \tag{10}$$

Here, Y and U are the Fourier coefficients of the output y and the input sequence u, respectively and Ω is the discrete frequency grid. The IPDR problem can now be formulated in the frequency domain as follows.

Problem 2. (IPDR, frequency domain). Consider the Fourier coefficients $R(\omega)$ of the periodic disturbance reference signal $\bar{r}$ and define the tracking error as,

$$E(\omega) \triangleq R(\omega) - Y(\omega), \quad \omega \in \Omega, \tag{11}$$

Then, (iteratively) find $U_i(\omega)$ such that $\lim_{i\to\infty} E_i(\omega) = 0$ $\forall \omega \in \Omega$.

Note that the frequency domain formulation is more specific since it assumes that the system is in steady state.

The time domain formulation is now readily obtained by recognizing that also the state x has settled to a periodic sequence, such that $x(Ni+N) = x(Ni)$. Substituting this in equation (3a) and combining the result with (3b) and (6) leads to,

$$\bar{e}_i = \bar{r} - J_p \bar{u}_i, \tag{12}$$

where the discrete periodic response matrix J_p is given by,

$$J_p = H(I - F)^{-1}G + J. \tag{13}$$

Now note that the lifted error (12) is given by a similar expression in the ILC case (7). As such, the existing ILC update law (8) and the design guidelines to shape L and Q can be used to extend the IIC approach.

In this section it is argued that the RC, ILC and IIC methods are based on different premises which are expressed in the way the state x is assumed to evolve along the trials. In RC, the state evolves continuously along the trials, while in ILC the state resets at the start of each trial and in IIC the state is periodic with N samples. It is shown that the equations that govern in the tracking error are similar in nature for the ILC and IIC method and therefore the IIC method can be extended by using the ILC approach as is presented in the next section.

3. INVERSION-BASED ITERATIVE CONTROL WITH ROBUSTNESS

In this section, the IIC update law, originally presented in the frequency domain in Tien et al. (2005), is transformed to the time domain. This allows a comparison to the ILC method, as represented by equation (8), which shows that existing IIC method lacks robustness in the ILC sense. An extension towards the inclusion of robustness is then proposed.

3.1 The IIC update law

The IIC method aims to solve this Problem 2 by employing a model of the inverse frequency response function (FRF) $G_m^{-1}(\omega)$ of G and updating the input signal as follows,

$$U_{i+1}(\omega) = U_i(\omega) + \rho(\omega)G_m^{-1}(\omega)E_i(\omega) \quad j \geq 1, \tag{14}$$

$\forall \omega \in \Omega$. Here, $\rho(\omega)$ is the real valued iteration coefficient, which should be chosen such that,

$$|1 - \rho(\omega)G(e^{j\omega})G_m^{-1}(\omega)| < 1, \quad \forall \omega \in \Omega. \tag{15}$$

It can be shown that this iterative process solves the IPDR problem in case (15) holds for $\rho(\omega) \neq 0$. This iterative process is equivalent to a Newton approach to minimizing $\mathcal{J} = \sum_{k=0}^{N-1} |E(\omega)|^2$, with $U(\omega)$ the unknown parameters. The convergence condition (15) can then be interpreted as a restriction on the step size to ensure convergence. A sensible initial input is for example given by $U_1(\omega) = G_m^{-1}(\omega)R(\omega)$.

3.2 Comparing IIC to ILC

The IIC approach to the IPDR problem can now be compared to the ILC solution by employing the Discrete Fourier Transformation (DFT) and its inverse in the lifted signal format. To this end, a lifted frequency domain signal $\bar{S}$ is defined as the column vector that contains the DFT coefficients S of a periodic signal s on its rows, i.e.,

$$\bar{S} = \begin{bmatrix} S(\frac{2\pi j}{N}0) \\ S(\frac{2\pi j}{N}1) \\ \vdots \\ S(\frac{2\pi j}{N}(N-1)) \end{bmatrix}. \tag{16}$$

The transformation between a periodic lifted signal $\bar{s}$ and its its frequency domain equivalent $\bar{S}$ can now be written as $\bar{S} = \mathcal{F}\bar{s}$, with $\mathcal{F} \in \mathbb{C}^{N\times N}$ the Discrete Fourier Transform matrix which is defined as,

$$\mathcal{F} \triangleq \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \mu & \mu^2 & \dots & \mu^{N-1} \\ 1 & \mu^2 & \mu^4 & \dots & \mu^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mu^{N-1} & \mu^{2(N-1)} & \dots & \mu^{(N-1)(N-1)} \end{bmatrix}, \tag{17}$$

i.e. $\mathcal{F}_{jk} = \frac{1}{\sqrt{N}}\mu^{jk}$ for $j, k = 0, \dots, N-1$, where $\mu = e^{-\frac{2\pi j}{N}}$ is a primitive N-th root of unity. Since the Vandermonde matrix $\mathcal{F}$ is a unitary matrix, i.e., $\mathcal{F} = \mathcal{F}^\dagger$ with $(.)^\dagger$ the conjugate transpose, it holds that $\bar{s} = \mathcal{F}^\dagger \bar{S}$. Consequently the IIC update law (14) in the time domain reads,

$$\bar{u}_{i+1} = \bar{u}_i + \mathcal{F}^\dagger \Theta \mathbb{G}_m^{-1} \mathcal{F} \bar{e}_i, \quad j \geq 1 \tag{18}$$

where, $\mathbb{G}_m = \text{diag}(G_m(\omega))$ and $\Theta = \text{diag}(\rho(\omega))$ are the diagonal matrices containing the FRF coefficients of G_m and the real valued iteration coefficients respectively. It is assumed here that $G_m(e^{j\omega}) \neq 0 \; \forall \omega \in \Omega$ which can readily be achieved for discrete time systems and particular if G_m results from an experimental, non-parametric estimation.

Comparing the IIC update law (18) with the ILC update law (8) shows that both laws are identical if,

$$Q = I, \quad L = \mathcal{F}^\dagger \Theta \mathbb{G}_m^{-1} \mathcal{F}. \tag{19}$$

This leads to the conclusion that no robustness is introduced in the IIC method through the Q-filter since $Q = I$.

3.3 Robustness in IIC

Inspired by infinite time ILC (Bristow et al., 2006, page 107), robustness can now be introduced in the IIC method by introducing a robustness coefficient $\phi(\omega) \in \mathbb{R}$ in the frequency domain update law as follows,

$$U_{i+1}(\omega) = \phi(\omega)\left(U_i(\omega) + \rho(\omega)G_m^{-1}(\omega)\right)E_i(\omega). \tag{20}$$

Or equivalently in the time domain as (8), with the following filters,

$$Q = \mathcal{F}^\dagger \Phi \mathcal{F}, \quad L = \mathcal{F}^\dagger \Phi \Theta \mathbb{G}_m \mathcal{F}, \tag{21}$$

where $\Phi = \text{diag}(\phi(\omega))$. Next, a brief frequency domain analysis of the robust IIC method is provided which leads to a number of design guidelines to ensure that the three requirements as stated in the introduction are satisfied.

3.4 Robust IIC Design Guidelines

A strategy to choose $\phi(\omega)$ and $\rho(\omega)$ is now presented $\forall \omega \in \Omega$, based on the following properties.

Disturbance rejection First, it is assumed that the error is converged, i.e., $\bar{E}_{i+1} = \bar{E}_i$ for $i \to \infty$. Ideally, the converged error should be equal to zero. It is readily shown that the converged error $E_\infty(\omega)$ is given by,

$$E_\infty(\omega) = \frac{(1-\phi(\omega))}{1-\phi(\omega)(1-\rho(\omega)\Delta(\omega))}R(\omega), \tag{22}$$

where $\Delta(\omega) \triangleq G(e^{j\omega})G_m^{-1}(\omega)$ is the inverse multiplicative mismatch. This shows that for $\phi(\omega) = 1$ perfect rejection is obtained and consequently, one should aim to have $\phi(\omega) = 1$ and only deviate from 1 in case convergence is not achieved at the desired rate, or robustness is not guaranteed, as is argued next.

Convergence & Convergence rate It is key that the robustness gain $\phi(\omega)$ and the iteration gain $\rho(\omega)$ are chosen such that the algorithm converges. It can be shown that this is the case if,

$$\gamma(\omega) < 1 \quad \forall \omega \in \Omega, \tag{23}$$

where $\gamma(\omega)$ is the rate of convergence, which is given by,

$$\gamma(\omega) \triangleq \frac{|E(\omega)_{i+1} - E_\infty(\omega)|}{|E(\omega)_i - E_\infty(\omega)|} = |\phi(\omega)\,(1 - \rho(\omega)\Delta(\omega))\,|. \tag{24}$$

Ideally, $\gamma(\omega)$ is as small as possible and here lies one of the key benefits of the robustness coefficient $\phi(\omega)$. Namely, in the IIC case, i.e., for $\phi(\omega) = 1$, convergence can be achieved by choosing $\rho(\omega)$ with the correct sign. However, if the $\Delta(\omega)$ is large, $\rho(\omega)$ has to be very small, as is shown in Tien et al. (2005) which leads to a prohibitively slow rate of convergence. This situation can be alleviated by taking $\phi(\omega) < 1$ leading to an improvement in convergence speed but possibly at a minor loss of disturbance rejection, as is expressed by (22).

Robustness Robustness against plant uncertainties can now be understood as achieving convergence for $\Delta(\omega)$ varying in a certain set of complex numbers, i.e. $\Delta(\omega) \in \mathbf{\Delta} \subseteq \mathbb{C}$. In case this leads to a sign change in $\rho(\omega)\Delta(\omega)$, then robustness can only be achieved by taking $\phi(\omega) < 1$ as will be illustrated by means of an example.

Example 1. Consider the case with $\Delta = \frac{1}{2}$, and $\rho = 2$, then $\gamma(\omega) = |\phi(1-1)| = 0$ and perfect tracking is achieved after a single iteration. However, if $\Delta = -\frac{1}{2}$, then $\gamma = |\phi(1+1)| < 1$ can only be achieved by taking $\phi < \frac{1}{2}$.

This trivial example shows that $\phi(\omega)$ is also key in achieving robust convergence. Based on the insights as presented above, the following design rules can now be formulated to achieve robust convergence at a satisfactory convergence rate for each $\omega \in \Omega$.

(1) Set $\phi(\omega) = 1$ and choose $\rho(\omega) \in \mathbb{R}$ (with the correct sign) such that convergence is achieved.
(2) In case $\rho(\omega)$ is very small ($\ll 1$), the convergence speed is likely to be unsatisfactory and robustness is not guaranteed for a large set of perturbations. To alleviate this reduce $\phi(\omega)$.

Remark 1. The above tuning procedure is based on a sensible trial-and-error approach. Note that a more rigorous procedure can be formulated similarly in case an uncertainty model of G_m^{-1} is available.

In this section, robustness is added to the IIC method akin to the ILC approach. A brief frequency domain analysis resulted in a frequency-by-frequency design procedure which will be applied to a nanopositioning stage in the next section.

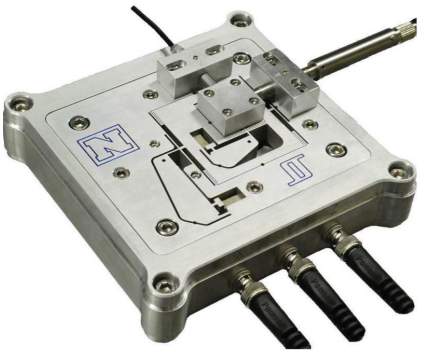

Fig. 1. Three-axis nanopositioner.

4. APPLICATION: NANOPOSITIONING STAGE

It was argued that robustness and improved convergence speed can be achieved by the addition of a robustness coefficient to the IIC method. In this section, these claims are verified on a experimental nanopositioning system as shown in Figure 1.

In this experiment, the x-axis will be controlled while the y- and z-axis are uncontrolled. The desired motion of the x-axis is taken to be non-smoothed triangle wave as shown in the upper plot of Figure 3. A single trial consists of 10 periods with period time 0.02 leading to a total trial time is 0.2 s. The sampling frequency is set to 20 kHz leading to $N = 4000$ samples per trial. The amplitude of the triangle wave is 5 μm which spans exactly one-third of the total range of motion. In this way, the system behaves approximately linear.

To increase the low frequency disturbance rejection and to attenuate output noise, a low bandwidth controller C_{fb} is connected in feedback with the system H, as shown in Figure 2. The controller consists of a weak integrator and two second order low-pass filters and is given by,

$$C_{fb} = \frac{6.1078 \cdot 10^{-6}(z+1)^4(z-0.9844)}{(z-0.9099)^2(z-0.975)(z-1)(z-0.7007)}. \tag{25}$$

4.1 The inverse system model

The inverse model of the system, G_m^{-1}, is constructed by empirically identifying a frequency response function (FRF) of the plant $\hat{H}$ and computing the process sensitivity G_m by using the FRF of the feedback controller C_{fb} as follows,

$$G_m(\omega) = \frac{\hat{H}(\omega)}{1 + \hat{H}(\omega)C_{fb}(e^{j\omega})}. \tag{26}$$

Here, $\hat{H}$ is obtained by exciting the system with a (pseudo) random binary sequence of 0.2 s and dividing the averaged cross power spectrum density of this input and the output, by the averaged input autopower spectrum. The resulting FRF of the process sensitivity is shown in Figure 4. The inverse of this model is simply the element-wise inverse of G_m.

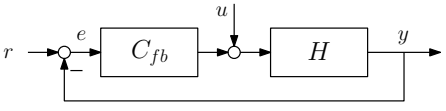

Fig. 2. Closed-loop scheme.

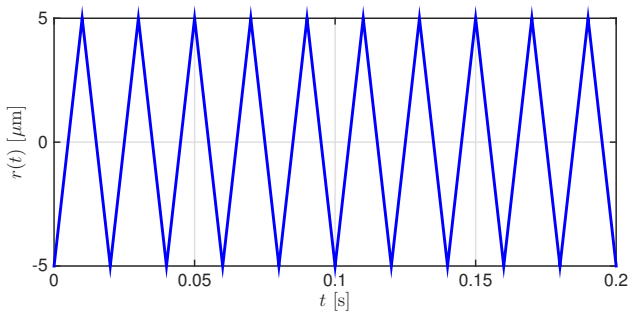

Fig. 3. Upper: reference trajectory $r(t)$ in the time domain. Lower: powerspectrum density of the reference.

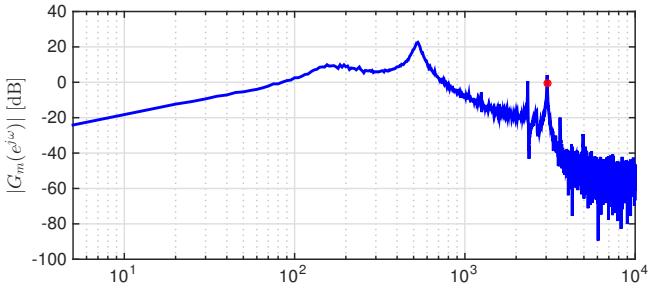

Fig. 4. Frequency response function of the process sensitivity G_m. Frequency axis in Hz.

4.2 Implementation

The Robust IIC method is implemented as follows. First the system is brought in continuous operation by means of the feedback controller. At the end of each trial a trigger indicates that data-acquisition can start. In this way, the right sequences can be obtained from the operating system. At the start of the algorithm, the initial feedforward is identically zero and is subsequently iteratively updated each time after two trials. The first trial is used for the system to settle in the steady state, which is the key assumption in the IIC methods. The second trial is measured and used to update the feedforward input according to equation (20). The implementation is facilitated by the fast Fourier transform routine `fft.m` in MATLAB.

4.3 Experimental tuning procedure

Now consider the design guidelines as presented in Section 3.4. In line with the first step, ϕ is set equal to 1 for each frequency. Then, ρ is set to 0.6 for all frequencies to allow convergence over a large frequency range. After the first experiment, it turns out that convergence is achieved up to 3700 Hz, except for a small set of frequencies, including the third resonance frequency $\omega^* = 3050$ Hz, as in indicated in Figure 4. At these frequencies, and for frequencies higher than 3700 Hz, ρ is set to 0. The second experiment is conducted with these settings and leads to a significant improvement in the tracking performance as shown in Figure 5. The results in this figure were obtained by performing the same experiment 30 times and averaging the results per iteration. The upper plot shows the reference $r(t)$ and the initial error without feedforward control $e_0(t)$. This shows that without feedforward, there is a relatively large tracking error with a maximum of

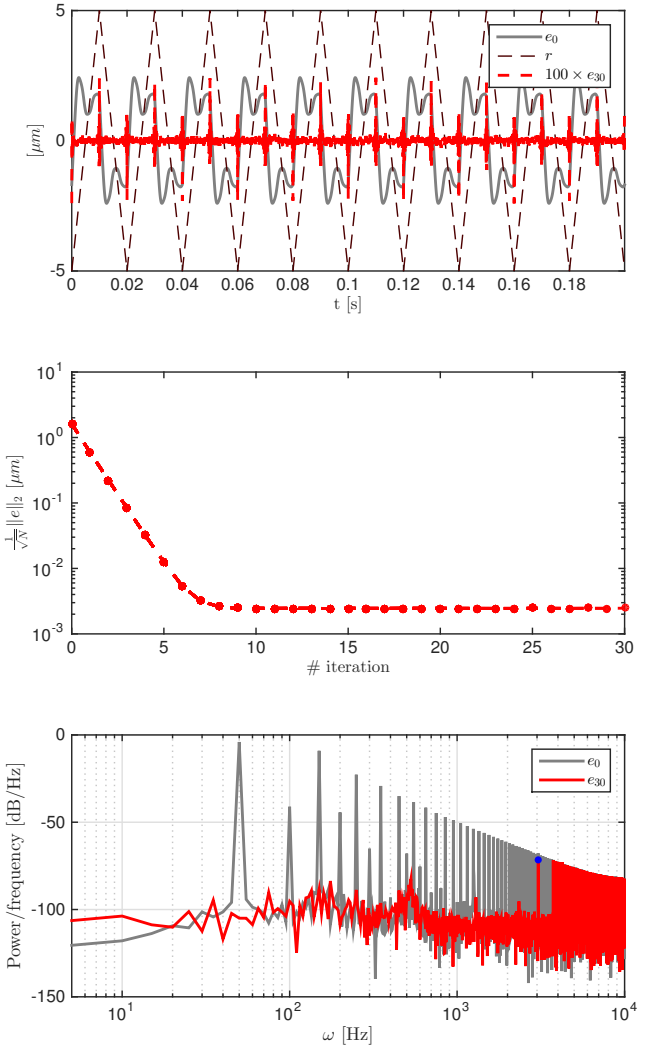

Fig. 5. Upper: Reference $r(t)$, the initial error without feedforward control $e_0(t)$ and the converged error $e_{30}(t)$ scaled by a factor 100 for visualization. Middle: The root mean square (RMS) of the scaled error sequence as function of the trial number. Lower: Power spectral density (PSD) of the initial and converged error sequences. The blue dot indicates the error at 3050 Hz.

about 50% of the reference. In addition, the error after 30 iterations is shown, $e_{30}(t)$, which is scaled by a factor 100 for visualization. This shows that the maximum of the converged error is less than 0.5% of the reference amplitude, which is a significant improvement with respect to the initial error. In the middle plot of Figure 5 the root mean square (RMS) of the error sequence is shown as a function of the iteration number. This shows that over the course of 9 iterations, the RMS-value decreases nearly 3 orders in magnitude.

The bottom plot of Figure 5 shows the Power spectral densities (PSD) of the initial and the converged error sequences. The PSD of the initial error reveals that the initial error is largest at frequencies where the reference is nonzero, while the converged error is significantly reduced up to 3700 Hz. However, at $\omega^* = 3050$ Hz, there is still one large peak present in the PSD since $\rho(\omega^*) = 0$, since

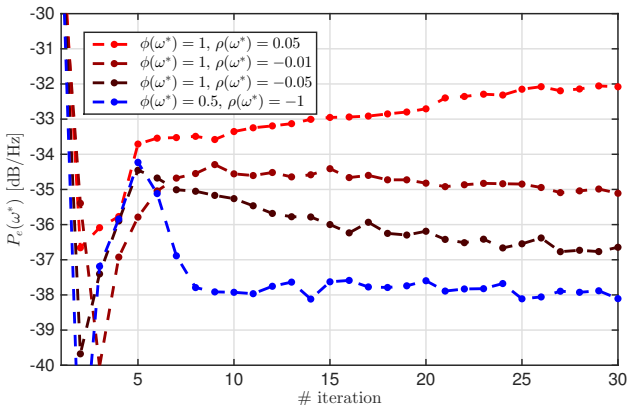

Fig. 6. Power spectrum density of the tracking error at $\omega^* = 3050$ Hz during the tuning procedure.

convergence was not achieved in the first experiment and consequently the error is not actively reduced at this frequency. This frequency is now specifically targeted by means of the design guidelines and the result is shown in Figure 6. The first step is conducted as follows.

(1) First it is chosen to take $\rho(\omega^*)$ small and positive. By taking $\rho(\omega^*) = 0.05$ the PSD of the error diverges.
(2) Based on this result, it is decided to take $\rho(\omega^*)$ even smaller and negative. By taking $\rho(\omega^*) = -0.01$ the PSD of the error seems to converge slightly.
(3) Based on this result, it is decided keep ρ negative and to increase the magnitude of $\rho(\omega^*)$ as much as possible such that the PSD of the error still converges. It turns out that $\rho(\omega^*) = -0.05$ yields the best possible result for $\phi = 1$ and the PSD of the error seems to be converging at a relatively slow rate.

The the robustness coefficient ϕ now enables an increase in convergence speed by taking $\phi(\omega^*) < 1$. In addition, note that ω^* occurs at a resonance peak, as is indicated by the red dot in Figure 4, which may shift for different load types, or may even vary for different stages due to manufacturing tolerances. To simulate this effect of varying Δ, an artificial mismatch in introduced by taking $\rho(\omega^*) = -1$, since convergence is known only to be achievable for $-0.05 < \rho < 0$. Now, by taking $\phi = 0.5$ convergence is nevertheless achieved and at a much higher speed as shown in Figure 6, thereby verifying the previously made claims.

In this section the Robust IIC method is successfully applied to a nanopositioner by following the design guidelines. It is shown that addition of the robustness coefficient indeed leads to improved convergence speed and robustness to plant variations.

5. CONCLUSION

An extended IIC method is developed and experimentally validated. IIC is placed in the general framework of lifted system descriptions, revealing a strong similarity to ILC algorithms. As a result, the robustness analysis of ILC algorithms directly applies to IIC methods. Inspired by ILC, robustness is added to the IIC method by means of a robustness coefficient. A frequency domain analysis of the Robust IIC method shows that in addition to robustness to plant variations, superior convergence speed can be achieved. This analysis leads to a set of design guidelines to choose the iteration- and robustness coefficients on a frequency-wise basis. Experimental application of the Robust IIC method to a nanopositioning system shows that the Robust IIC method indeed enables increased convergence speed, even when the model mismatch is artificially increased.

Future research will consider the effect of trial varying disturbances and will extend the design guidelines in the case where an uncertainty model of the plant is available.

ACKNOWLEDGEMENTS

Yik Ren Teo is thanked for his valuable assistance with the nanopositioning stage setup.

REFERENCES

Bamieh, B., Pearson, J.B., Francis, B.A., and Tannenbaum, A. (1991). A lifting technique for linear periodic systems with applications to sampled-data control. *Systems & Control Letters*, 17, 79–88.

Bechhoefer, J. (2008). Feedforward control of a piezoelectric flexure stage for AFM. *2008 American Control Conference*, 2703–2709.

Bristow, D., Tharayil, M., and Alleyne, A. (2006). A survey of iterative learning control. *Control Systems, IEEE*, 26(3), 96–114.

Devasia, S., Member, S., and Eleftheriou, E. (2007). A Survey of Control Issues in Nanopositioning. 15(5), 802–823.

Fleming, A.J. and Leang, K.K. (2014). Design, Modeling and Control of Nanopositioning Systems.

Hara, S., Yamamoto, Y., Omata, T., and Nakano, M. (1988). Repetitive Control System: A New Type Servo System for Periodic Exogenous Signals. *IEEE Transactions on Automatic Control*, 33(7), 659–668.

Kim, K.S. (2008). Model-less inversion-based iterative control for output tracking: Piezo actuator example. *2008 American Control Conference*, 2710–2715.

Pipeleers, G., Demeulenaere, B., and Swevers, J. (2009). *Optimal Linear Controller Design for Periodic Inputs*. Springer.

Roover, D., Bosgra, O.H., and Steinbuch, M. (2000). Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control*, 73(10), 914–929.

Tien, S., Zou, Q., and Devasia, S. (2005). Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed AFM operation. *IEEE Transactions on Control Systems Technology*, 13(6), 921–931.

Tousain, R., van der Meché, E., and Bosgra, O. (2001). Design strategies for iterative learning control based on optimal control. *Selected Topics in Signals, Systems and Control*, 12(September).

van de Wijdeven, J., Donkers, T., and Bosgra, O. (2009). Iterative Learning Control for uncertain systems: Robust monotonic convergence analysis. *Automatica*, 45(10), 2383–2391.

Zou, Q., Kim, K.S., and Su, C. (2007). Iteration-based Scan-Trajectory Design and Control with Output-Oscillation Minimization: Atomic Force Microscope Example. *Proceedings of the 2007 American Control Conference*, 4227–4233.