FREQUENCY DOMAIN ITERATIVE LEARNING CONTROL FOR DIRECT-DRIVE ROBOTS

Björn Bukkems,[1] Dragan Kostić,[2] Bram de Jager,[3] and Maarten Steinbuch[4]

Technische Universiteit Eindhoven, Department of Mechanical Engineering
Dynamics and Control Technology Group, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Fax: +31 40 246 1418, Phone: [1]+31 40 247 2841, [2]+31 40 247 5730, [3]+31 40 247 2784, [4]+31 40 247 5444
[1]B.H.M.Bukkems@tue.nl, [2]D.Kostic@tue.nl, [3]A.G.de.Jager@wfw.wtb.tue.nl, [4]M.Steinbuch@tue.nl

Keywords: Robotics, Dynamics, Model-based control, Iterative Learning Control, Direct-drive robots

Abstract

This paper presents an Iterative Learning Control algorithm for direct-drive robots. The learning algorithm assumes linear dynamics, which is created using a nonlinear model-based compensator. The convergence criterion of the learning controller is derived in the frequency domain. Rules for designing the filters, used in the update law, are explained. The effectiveness of the algorithm is demonstrated in experiments on a spatial direct-drive robot. The root-mean-square values of the tracking errors in a demanding writing task are over 10 times smaller after just eight iterations of the learning algorithm, compared with the errors before learning.

1 Introduction

Increasing demands on the performance of modern robot manipulators have led to the development of various motion control approaches. They can be divided into three main categories [3]. The first one is decentralized control, relying on independent feedback loops that implement PD and PID controllers [12]. Nonlinear robot dynamics is neglected and coupling effects are treated as disturbances. Although still a dominant practical solution, it does not maximize robot performance. The second category is model-based control. To achieve better tracking accuracy, dynamic couplings between robot axes are taken into account, which is realized by integrating the dynamics into the control design.

Whereas the previous two categories essentially resort on feedback control design, the third category is focussed on the design of a feedforward control input. A special approach within this category is the so-called Iterative Learning Control (ILC) approach. Becoming more attractive in robotics, it is used for accurate tracking of repetitive motions, which are often met in practice. An ILC algorithm calculates a feedforward control input iteratively, based on the error and the feedforward control input of a previous trial. The tracking performance is improving through the repetition of trials, until the reproducible part of the tracking error becomes small and non-reproducible errors become dominant.

After its introduction in robotics [1], the number of ILC techniques proposed for robotic problems has become almost as large as the number of practitioners. Commonly, they require very little knowledge on the robot dynamics, assuming either just an approximate nonlinear dynamics with roughly known parameters [2, 5, 6], or even linear dynamics [9]. Regarding the domain in which the learning controller is designed, a distinction can be made between time-domain [1, 2, 5, 6], and frequency-domain ILC techniques [9]. Usually, the time-domain techniques offer rather conservative rules for tuning the learning controller. This is considered to be a basic limitation, since the maximum effect on the tracking performance cannot be anticipated. On the other hand, frequency-domain techniques can take into account additional knowledge on particular plant dynamics and thus they enable building up a real tuning formalism, but a drawback is that they work only with linear plants.

In robotics, assuming linearity is justified only if high-geared transmission elements are used in the robot joints. Due to high reduction ratios, nonlinear dynamic couplings between robot axes are reduced and the linear behavior along each axis prevails [15]. However, this is not the case for manipulators with direct-drive actuators, as nonlinear dynamics are inherent to them. To use an ILC design for such manipulators, one should combine ILC and model-based control approaches.

In this paper, we consider a frequency-domain ILC technique for improving the performance of direct-drive robots. The design of the learning filter is based on the dynamics that remains after decoupling the robot axes with a model-based compensator for the nonlinear dynamics. To the best of our knowledge, this is the first attempt to practically investigate the functional combination of a nonlinear model-based compensator and frequency-domain ILC. A notable difference between other frequency-domain techniques used in robotics (e.g., [9]) and our approach, is that we provide a formalism for tuning the filters based on the measured dynamics. Such formalism enables one to anticipate the benefits of the ILC algorithm on the tracking performance. To demonstrate the merits of the suggested technique, we perform experiments on a direct-drive robotic arm with three rotational joints, implemented as a waist, shoulder and elbow. As the considered kinematic structure is often met in industry, results of this study are representative for industrial cases. Experiments using a demanding writing task [11] show the potential of the proposed ILC technique.

The paper is organized as follows. In the next section we introduce ILC for a single-input single-output (SISO) linear system. Use of the nonlinear model-based compensator to establish a linear decoupled dynamics in the robot joints is explained in section 3. The experimental setup and the learning filter design are discussed in section 4. Section 5 demonstrates the effectiveness of the proposed technique with experiments, and conclusions will come at the end.

2 Iterative Learning Control

Consider the block diagram in Figure 1. The SISO linear plant P is described by its transfer function:

$$P(s) = \frac{\Theta(s)}{U(s)}, \tag{1}$$

where $U(s)$ and $\Theta(s)$ are the Laplace transforms of the applied control input $u(t)$ and the plant output $\theta(t)$, respectively. The feedback controller C is used to stabilize the system by generating the feedback control input $u_{fb}(t)$, corresponding to the tracking error:

$$e(t) = \theta_r(t) - \theta(t), \tag{2}$$

where θ_r is the desired output of P. The input $u_{ilc}(t)$ represents the feedforward control signal, which is updated after each trial. In the Laplace domain[1], the update rule of the learning algorithm is defined as follows:

$$U_{ilc}^{k+1}(s) = Q(s)(U_{ilc}^{k}(s) + L(s)E^k(s)), \tag{3}$$

where k is the number of the trial, L is the learning filter and Q is a robustness filter. According to Figure 1, the errors in two successive trials satisfy the following relations:

$$E^k(s) = -P(s)S(s)U_{ilc}^{k}(s), \tag{4}$$

$$E^{k+1}(s) = -P(s)S(s)U_{ilc}^{k+1}(s), \tag{5}$$

where S denotes the sensitivity function:

$$S(s) = \frac{1}{1+P(s)C(s)}. \tag{6}$$

The product $P(s)S(s)$ is called the process sensitivity. Substitution of (3) and (4) into (5) results in:

$$E^{k+1}(s) = E^k(s)Q(s)(1-L(s)P(s)S(s)). \tag{7}$$

From this equation it can be seen that the tracking error decreases if the following convergence criterion is satisfied:

$$\|Q(s)(1-L(s)P(s)S(s))\|_\infty < 1, \tag{8}$$

with $\|.\|_\infty$ denoting the infinity norm. To maximize the convergence speed, the learning filter L should be identical to the inverse of the process sensitivity:

$$L(s) = \frac{1}{P(s)S(s)}. \tag{9}$$

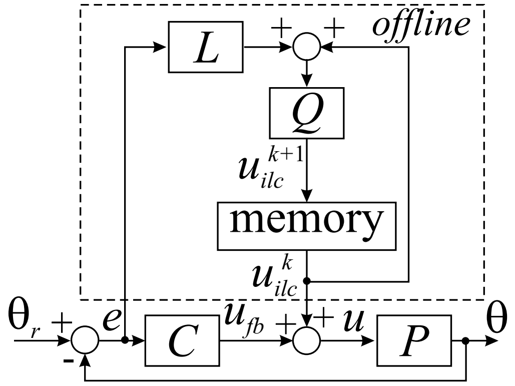

Figure 1: Iterative Learning Control block diagram

To compute the L-filter, a parametric model of the process sensitivity has to be made. If the plant P is non-minimum phase, the L-filter in (9) will be unstable when the process sensitivity is inverted directly. To avoid this, the Zero Phase Error Tracking Algorithm for Digital Control [14] can be used, which makes use of a discrete-time model of the process sensitivity. It inverts its poles and stable zeros and cancels the phase shift induced by the unstable zeros. To ensure that the convergence criterion (8) is satisfied for the real plant dynamics, the robustness filter Q is introduced. It is needed, as the L-filter is designed using a parametric model of the process sensitivity which, in practice, cannot cover the complete dynamics of the real system but only some low-frequency range. The Q-filter should guarantee the validity of (8) for the frequency components at which the model is inaccurate, as at these frequencies the convergence criterion can be violated. Since in practice modelling errors occur in the high-frequency range, Q is mostly chosen to be a low-pass filter. In the simplest case, below its cut-off frequency f_Q, the Q-filter has a pass-band equal to 1, while above f_Q its amplitude is decreasing. The Q-filter should influence the frequency content of the feedforward control input only by its magnitude characteristics, but should not introduce any phase distortion. To explain this, let us discuss the learning mechanism defined within the parenthesis in the update rule (3). The learning mechanism produces an input signal from the tracking error and the feedforward input of the previous trial. This input signal is effective in reducing the tracking error only in the frequency range at which the L-filter is a correct representation of the inverse of the measured process sensitivity. Frequency components of the input signal that are outside this range are not effective in decreasing the error, so they must be suppressed by the Q-filter. However, in time-domain, the input signal should exactly match the measured tracking error and the feedforward input from the previous trial. That is why the Q-filter is not allowed to introduce phase distortion. Consequently, filtering the input signal with the Q-filter is a noncausal filtering operation that will be performed offline during the experiments, as indicated in Figure 1.

The disadvantage of the introduction of the Q-filter is that a zero tracking error cannot be guaranteed above f_Q. This can be

[1] Because we assume identical initial conditions in each trial, the analysis in the Laplace domain is allowed, even though each trial has finite time-length [10].

verified using the following formula, derived from (3) and an expression for the tracking error obtained from Figure 1:

$$\lim_{k\to\infty} E^k(s) = \frac{1-Q(s)}{(S(s))^{-1}(1-Q(s))+Q(s)P(s)L(s)}\Theta_r(s). \quad (10)$$

At low frequencies the tracking error can become zero, as Q is identity below f_Q, while at high frequencies $|Q| << 1$ and the error will be:

$$\lim_{k\to\infty} E^k(s) \approx S(s)\Theta_r(s), \quad \forall f >> f_Q. \quad (11)$$

However, this is not a drawback in practical robotics problems, since the harmonic content of the reference motions is typically present at low frequencies.

3 Robot dynamics

Let us now investigate the possibility to apply the ILC algorithm of the previous section in a direct-drive robot control problem. The dynamics of a robot with n actuated joints can be represented using the Euler-Lagrange formalism:

$$\mathbf{D}(\theta(t))\ddot{\theta}(t)+\mathbf{c}(\theta(t),\dot{\theta}(t))+\mathbf{g}(\theta(t))+\mathbf{f}(t)=\tau(t), \quad (12)$$

where θ, $\dot{\theta}$ and $\ddot{\theta}$ are the $n\times 1$ vectors of joint motions, speeds, and accelerations, respectively, $\mathbf{D}$ is the $n\times n$ inertia matrix, $\mathbf{c}$, $\mathbf{g}$ and $\mathbf{f}$ are 3×1 vectors of nonlinear Coriolis/centripetal, gravitational and friction forces, respectively, and τ is the $n\times 1$ vector of control inputs (joint forces/torques). Since the ILC algorithm can be applied to linear plants only, the nonlinear and highly coupled dynamics in (12) have to be compensated for by a model-based compensator for the system dynamics:

$$\begin{aligned}\tau_{\mathbf{c}}(t) = \bar{\mathbf{D}}(\theta_r(t))\mathbf{u}(t)+\bar{\mathbf{c}}(\theta_r(t),\dot{\theta}_r(t))+ \\ +\bar{\mathbf{g}}(\theta_r(t))+\bar{\mathbf{f}}(t). \quad (13)\end{aligned}$$

Here $\mathbf{u}$ represents the $n\times 1$ vector of new control inputs that should ensure stable robot motions and accurate tracking of the reference θ_r. The upper bars indicate that a model is used to compensate the real dynamic effects. Ideally, when the controller (13) is applied to the dynamics (12), a linear plant remains:

$$\ddot{\theta}(t) = \mathbf{u}(t). \quad (14)$$

This holds only if the dynamic model perfectly matches the real robot dynamics. In practice this rarely happens, especially in the presence of flexible effects that are not included in the model. As discussed in [8], the dynamics in the robot joints obtained after applying the nonlinear model-based compensator is more complex than the ideal one in (14). For the i-th degree of freedom (d.o.f.), it is defined as the transfer function:

$$P_i(s) = \frac{\Theta_i(s)}{U_i(s)}, \quad (15)$$

where P_i is not just a double integrator, as suggested by (14), but it has poles and zeros due to the appearance of resonances and anti-resonances. The frequencies and damping of these resonances vary as the robot configuration changes. Capturing all possible variations in the plant dynamics with a single model could be a difficult task. To avoid this, we choose a strategy of adopting a nominal model of the plant, to represent its average dynamics. Differences from the nominal model are interpreted as perturbations in the dynamics.

In order to calculate the nominal plant model P_i^0, needed in the ILC design, the frequency response functions $G_i^l(j\omega)$ from $u_i(t)$ to $q_i(t)$ are measured for N different robot configurations (l= $1,\ldots,N$), as explained in [8]. These configurations should span the full range of the robot joints. Once a set of measurements has been collected, at least two possibilities can be used to calculate the nominal frequency response $G_i^0(j\omega)$:

$$G_i^0(j\omega) = arg \min_{G_i(j\omega)} \max_{l} \left|\frac{G_i^l(j\omega)-G_i(j\omega)}{G_i(j\omega)}\right|, \quad (16)$$

or

$$G_i^0(j\omega) = \frac{1}{N}\sum_{l=1}^{N} G_i^l(j\omega). \quad (17)$$

The first possibility minimizes the distance between the nominal and the N measured frequency responses at each frequency ω. The second possibility calculates the average of the N measured frequency responses. We choose the latter possibility, because it provides us smoother amplitude and phase plots. Smoother plots will facilitate fitting the parametric model P_i^0 onto the nominal frequency response data G_i^0. Perturbations from this nominal dynamics must be taken into account during the ILC design, which will be particularly illustrated by the case study in the next section.

After applying the nonlinear model-based compensator, the control situation illustrated by Figure 1 can be established. Consequently, the ILC design of section 2 becomes possible.

4 Experimental Setup

The robotic arm, shown in Figure 2, is the subject of our case study. The photo and kinematic parameterisation according to the well-known Denavits-Hartenberg's (DH) notation [4], reveal three revolute degrees of freedom (d.o.f.), which makes such a kinematic structure referred to as RRR. Each d.o.f. is actuated by a brushless DC direct-drive motor and has an infinite range of motions, thanks to the use of slip-rings for the transfer of power and sensor signals [15], [16]. The actuators are Dynaserv DM-series servos with nominal torques of 60, 30 and 15 [Nm], respectively. The servos are driven by power amplifiers with built in current controllers. Joint motions are measured using incremental optical encoders, with a resolution of 10^{-5} [rad]. A PC-based platform is used for implementation of control algorithms using Matlab/Simulink software.

Detailed models of the robot kinematics and dynamics are available in [7]. From Figure 2, one can determine the DH parameters, whose numerical values are presented in Table 1. The DH parameters are: twist angles α_i, link lengths a_i, joint displacements θ_i, and link offsets d_i.

After applying the nonlinear model-based compensator (13) and stabilizing the robot using a PD feedback control law:

$$\begin{array}{c}\mathbf{u}_{fb} = \mathbf{K_p}\mathbf{e} + \mathbf{K_d}\dot{\mathbf{e}}, \\ \mathbf{K_p} = \text{diag}\left[K_{p,1}, K_{p,2}, K_{p,3}\right], \mathbf{K_d} = \text{diag}\left[K_{d,1}, K_{d,2}, K_{d,3}\right]\end{array}, \quad (18)$$

the remaining dynamics is identified.

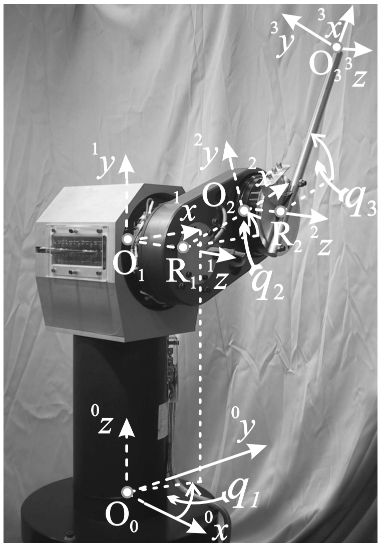

Figure 2: The RRR robotic arm and the kinematic parameterisation according to the DH notation

dof	α_i [rad]	a_i [m]	θ_i	d_i [m]
1	$\frac{\pi}{2}$	0	θ_1	O_0O_1=0.560
2	0	R_1O_2=0.200	θ_2	O_1R_1=0.169
3	0	R_2O_3=0.415	θ_3	O_2R_2=0.090

Table 1: DH parameters of the experimental RRR robotic arm

As a case study, we explain the ILC design for the first robot joint only, but the experimental results, given in the next section, cover the case with learning controllers for all three joints. Figure 3 shows frequency responses for the first joint corresponding to N=16 static postures, spanning the complete range of joints 2 and 3: $\left[\theta_{r,2}^1\,\theta_{r,3}^1\right] = [0°\,0°]$, $\left[\theta_{r,2}^2\,\theta_{r,3}^2\right] = [0°\,90°], \ldots, \left[\theta_{r,2}^{16}\,\theta_{r,3}^{16}\right] = [270°\,270°]$. From this figure it is obvious that the theoretically expected dynamics of a double integrator (see (14)) holds at low frequencies only (below 20 [Hz]), while the real dynamics is more involved at higher frequencies. The increasing phase lag, which can be observed in the phase plot in Figure 3, is caused by the time delay in the control system [8]. The nominal frequency response G_1^0, calculated according to (17), is shown in Figure 3 as well. It is also depicted in Figure 4, together with its parametric fit P_1^0. The fitting is done in the least-squares sense, using an output-error model structure [13]. As can be seen from this figure, the fit is accurate only until 70 [Hz] because we do not expect it would be necessary to learn above this frequency. Fitting the high frequent dynamics is therefore unnecessary.

The PD feedback controller C_1, with $K_{p,1} = 1000$ and $K_{d,1} = 2\sqrt{1000}$, is adopted to stabilize the robot. Given C_1 and P_1^0, the parametric model of the process sensitivity can be calculated. Based on this model the L-filter can be calculated as explained in section 2. Figure 5 shows the L-filter, the process sensitivities calculated for all 16 frequency responses $G_1^1, \ldots, G_1^{16}$, and their products. A perfect L-filter would result in 0 [dB] magnitude and 0° phase shift of these products. From Figure 5, it can be seen that the obtained L-filter matches the inverse of the process sensitivities until approximately 20 [Hz]. The convergence criterion (8) can be preserved above 20 [Hz] by means of the low pass robustness filter Q. Therefore, the cut-off frequency of the Q-filter is chosen to be 18 [Hz]. For joints 2 and 3, the cut-off frequencies of the Q-filters are 25 [Hz] and 27 [Hz], respectively.

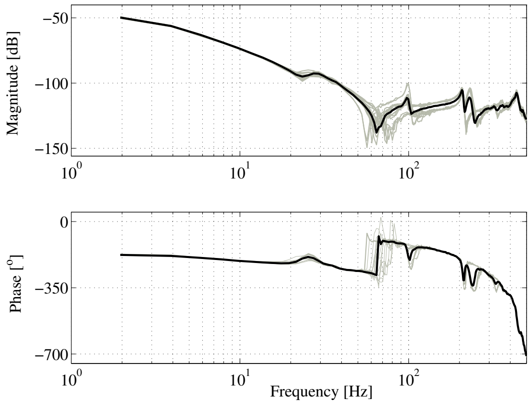

Figure 3: Experimentally obtained frequency responses $G_1^1, \ldots, G_1^{16}$ (gray) and the nominal frequency response G_1^0 (black).

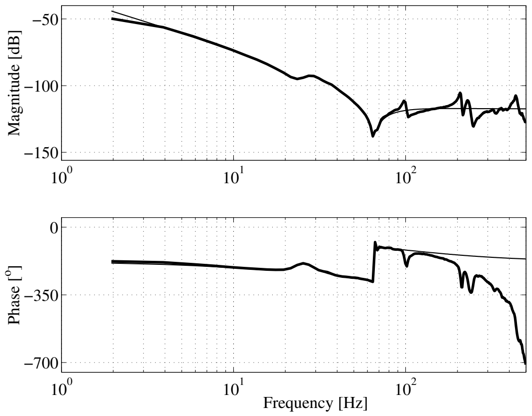

Figure 4: The nominal frequency response G_1^0 (thick line) and its parametric fit P_1^0 (thin line)

5 Experimental results

To demonstrate the effectiveness of the proposed control design, experiments have been conducted. In these experiments, the robotic arm had to perform a writing task, which is often recognized as very demanding for the dynamics of a mechanical system due to its fast motions [11]. In this writing task, shown in three dimensional space on the left hand side of Figure 6, the robot-tip had to track the path starting from "A" in the direction of the arrows. The corresponding joint motions are depicted on the right hand side of Figure 6.

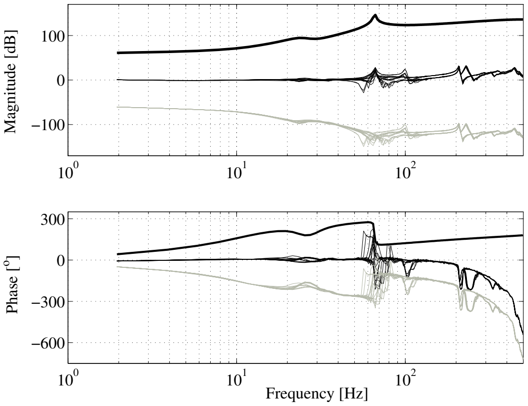

Figure 5: Frequency response of the *L*-filter for the first joint (black, thick), the process sensitivities calculated for 16 distinct robot postures (gray) and their products (black, thin)

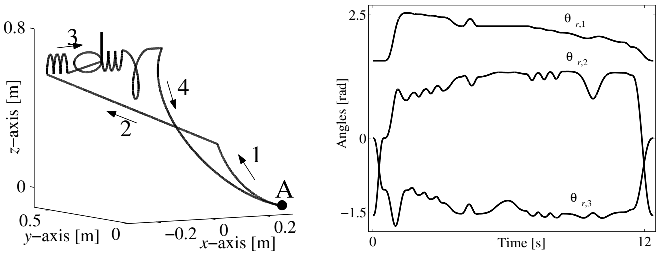

Figure 6: Reference path of the robot tip (left) and corresponding joint motions (right)

During the experiments, the robot motions are stabilized by the PD feedback controllers that have also been used in the derivation of the *L*-filters. After each trial, the new feedforward control input is calculated according to (3). This means filtering of the measured tracking errors with the corresponding *L*-filters, adding the previous feedforward input to this filtered signal, and, finally, filtering of the obtained signals with the corresponding *Q*-filters. The result is the feedforward control input for the next trial.

The performed experiment consisted of 10 trials. In Figure 7 the tracking errors during the first, second and tenth trial in all three joints are depicted. From this figure it can be seen that the tracking errors have already reduced considerably in the second trial, i.e., after just one iteration.

As depicted in Figure 8, the root-mean-square values of the errors in all joints keep decreasing until they reach lower bounds, that cannot be further reduced. For the first two joints, seven trials are needed, while for the third joint it takes eight trials. The root-mean-square values of the errors after the learning process are over ten times smaller than before learning. Figure 9 shows the auto power spectra of the tracking errors in all joints, based on data measured during the first, second and tenth trial. The spectra show significant reduction of frequency components until approximately 10 [Hz] in the first two joints and 15 [Hz] in the third joint. Such reduction is very important as the frequency content of the reference motions are within these ranges.

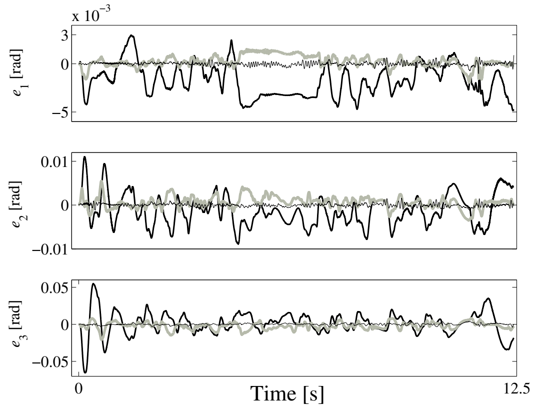

Figure 7: Tracking errors in the first (upper), second (middle) and third joint (bottom) during trial 1 (black, thick), trial 2 (gray) and trial 10 (black, thin)

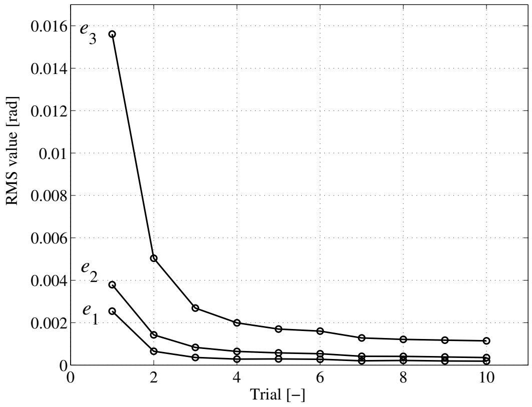

Figure 8: Root-mean-square values of the errors in all joints

6 Conclusions

This paper presents an Iterative Learning Control algorithm for direct-drive robots. As it is a frequency-domain approach, it is applicable to linear systems only. Therefore, a nonlinear model-based compensator is used to obtain linear, decoupled dynamics, after which this dynamics is identified. A formalism for tuning the filters based on the measured dynamics is given, taking into account learning capabilities and convergence issues. The effectiveness of the algorithm is experimentally demonstrated on a spatial direct-drive robotic arm with three rotational joints. A considerable reduction in the tracking error is obtained in all three joints.

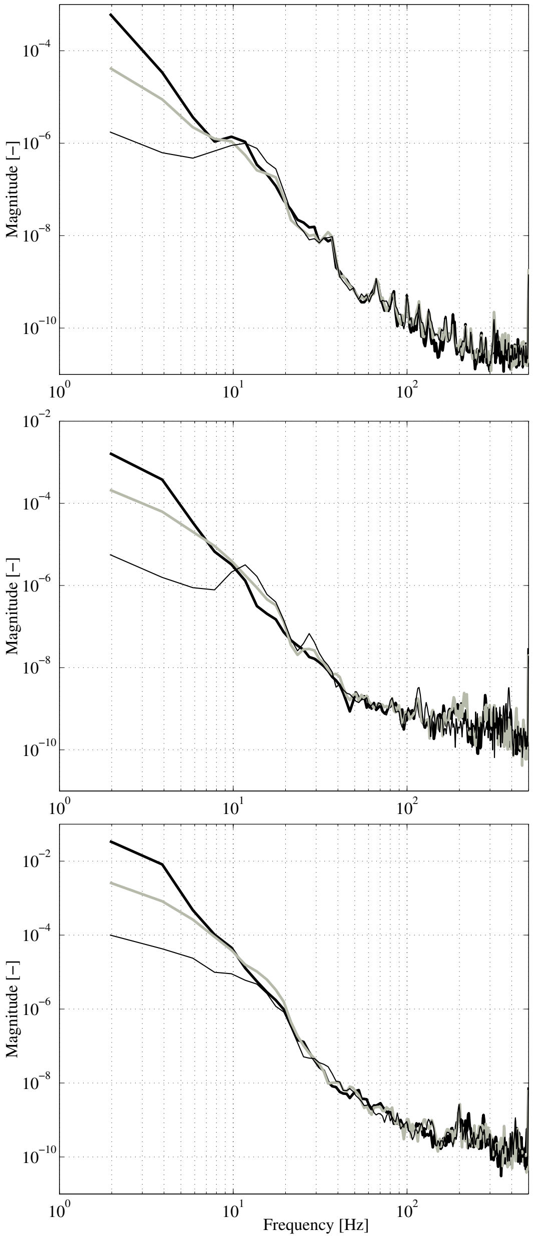

Figure 9: The auto power spectra of the tracking errors in the first (top), second (middle) and third (bottom) joint for trial 1 (black, thick), trial 2 (gray) and trial 3 (black, thin)

References

[1] S. Arimoto, S. Kawamura, F. Miyazaki. “Bettering operation of robots by learning”, *J. Rob. Sys.*, **Volume 1(2)**, pp. 123–140, (1984).

[2] S. Arimoto. “Control theory of non-linear mechanical systems: A passivity-based circuit-theoretic approach”, Oxford Univ. Press, Oxford, (1996).

[3] P. Bondi, G. Casalino, L. Gambardella. “On the iterative learning control theory of robotic manipulators”, *IEEE Trans. Rob. Autom.*, **Volume RA-4**, pp. 14–22, (1988).

[4] K.S. Fu, R.C. Gonzales, C.S.G. Lee. “Robotics: Control, sensing, vision and intelligence.”, McGraw-Hill, London, (1996).

[5] K. Hamamoto, T. Sugie. “Iterative Learning Control for Robot Manipulators Using The Finite Dimensional Input Subspace”, *IEEE Trans. Rob. Autom.*, **Volume 18, No. 4**, pp. 632–635, (2002).

[6] S. Kawamura, F. Miyazaki, S. Arimoto. “Realization of Robot Motion Based on a Learning Method”, *IEEE Trans. Sys., Man, Cyb.*, **Volume 18, No. 1**, pp. 126–134, (1988).

[7] D. Kostić, B. de Jager, M. Steinbuch. “Closed-form Kinematic and Dynamic Models of an Industrial-like RRR Robot”, *IEEE Int. Conf. Rob. Autom.*, pp. 1309–1314, (2002).

[8] D. Kostić, B. de Jager, M. Steinbuch. “Experimentally Supported Control Design for a Direct Drive Robot”, *IEEE Int. Conf. Control Appl.*, pp. 186–191, (2002).

[9] A. de Luca, G. Paesano, G. Ulivi. “A Frequency Domain Approach to Learning Control: Implementation for a Robot Manipulator”, *IEEE Trans. Ind. Elec.*, **Volume 39, No. 1**, pp. 1–10, (1992).

[10] K.L. Moore. “Iterative Learning control for deterministic systems”, Springer Verlag, London, 1993.

[11] V. Potkonjak, S. Tzafestas, D. Kostić, G. Djordevic, M. Rasic. “Illustrating Man-Machine Motion Analogy in Robotics - The Handwriting Problem”, *IEEE Trans. Rob. Autom. Mag.*, **Volume 10, No. 1**, pp. 35–46, (2003).

[12] L. Sciavicco, B. Siciliano. “Modeling and Control of Robot Manipulators”, McGraw-Hill, London, (1996).

[13] C.K. Sanathanan, J. Koerner. “Transfer Function Synthesis as a Ratio of Two Complex Polynomials”, *IEEE Trans. Autom. Control*, **Volume 8, No. 1**, pp. 56–58, (1963).

[14] M. Tomizuka. “Zero Phase Error Tracking Algorithm for Digital Control”, *J. Dyn. Sys., Meas., and Control*, **Volume 109**, pp. 65–68, (1987).

[15] B. van Beek, B. de Jager. “RRR-Robot design: Basic outlines, servo sizing, and control”, *Proc. IEEE Int. Conf. Control Appl.*, pp. 36–46, (1987).

[16] B. van Beek, B. de Jager. “An experimental facility for nonlinear robot control”, *Proc. IEEE Int. Conf. Control Appl.*, pp. 668–673, (1999).