Learning-Based Identification and Iterative Learning Control of Direct-Drive Robots

Björn Bukkems, Dragan Kostić, Member, IEEE, Bram de Jager, and Maarten Steinbuch, Senior Member, IEEE

Abstract—A combination of model-based and iterative learning control (ILC) is proposed as a method to achieve high-quality motion control of direct-drive robots in repetitive motion tasks. We include both model-based and learning components in the total control law, as their individual properties influence the performance of motion control. The model-based part of the controller compensates much of the nonlinear and coupled robot dynamics. A new procedure for estimating the parameters of the rigid body model, implemented in this part of the controller, is used. This procedure is based on a batch-adaptive control algorithm that estimates the model parameters online. Information about the dynamics not covered by the rigid body model, due to flexibilities, is acquired experimentally, by identification. The models of the flexibilities are used in the design of the iterative learning controllers for the individual joints. Use of the models facilitates quantitative prediction of performance improvement via ILC. The effectiveness of the combination of the model-based and the iterative learning controllers is demonstrated in experiments on a spatial serial direct-drive robot with revolute joints.

Index Terms—Dynamics, identification, iterative learning control (ILC), model-based control, robotics.

I. INTRODUCTION

Increasing demands on the performance of modern robotic manipulators have led to the development of various motion control approaches. These approaches can be divided into three main categories [1]. The first one is decentralized control, relying on independent feedback loops that implement conventional proportional-integral-derivative (PID) control laws [1], [2]. Nonlinear couplings in the robot dynamics are treated as disturbances. Because of simplified implementation and tuning requirements, decentralized control is still a dominant solution in practice. However, this solution is also recognized as being unable to provide high performance motion control, which in this paper is defined as high accuracy in realizing the reference robot trajectory, without violation of stability criteria, without excitation of parasitic robot dynamics (e.g., flexibilities) and without amplification of noise (disturbances).

The second category is model-based control [1], [2], which is realized by integrating the nonlinear robot dynamics into the control design. This control approach is computationally more intensive than decentralized control. However, it is supposed to provide better tracking accuracy. By augmenting the nonlinear model-based controller with feedback laws that include models of the robot flexibilities and disturbances, higher robustness against these undesirable effects can be achieved. Model-based control can be used for the design of both feedback and feedforward control laws.

The third category derives control input signals through a learning process. Learning means refinement of the input signals, based on past performance in executing the reference motion. The refined control input should provide performance improvement. A special approach within this category is iterative learning control (ILC). This approach is very attractive in robotics, because of its performance improving capabilities along repetitive trajectories, often met in practice. A learning algorithm calculates an ILC input based on the tracking error and the ILC input of the previous trial. The tracking performance is improving through the repetition of trials, until the reproducible parts of the tracking errors are reduced, and nonreproducible errors become dominant.

This paper investigates the combination of nonlinear model-based robot motion control and ILC. The particular objective is to realize high performance motion control of direct-drive robotic manipulators. Because of direct-drive actuation, dynamics of these manipulators are highly nonlinear and coupled, which impedes motion control of high quality. The additional requirement is to realize high motion performance while implementing feedback controllers that are weakly tuned. Weak tuning means less stiffness in the robot joints, which in turn implies that the structure of the manipulator will exert less force/torque during an unexpected contact with some object (e.g., when hitting an obstacle). The model-based controller, with a conventional feedback control law, is used for stabilization and compensation of much of the nonlinear and coupled robot dynamics. Imperfections of the nonlinear model-based controller at low frequencies are tackled via ILC. Both model-based and learning control components are important for the performance of motion control, and, therefore, they will be equally emphasized in this paper. In particular, we will present experimental procedures to identify rigid-body and flexible dynamics of a robot. The identified rigid-body dynamics are used in the model-based controller. The flexible dynamics are considered in the ILC design. The theories behind identification and control designs will be presented in sufficient detail. These theories will be practically demonstrated on a spatial serial direct-drive robotic manipulator with three revolute joints. Experimental results will confirm performance improve-
ment when the model-based and the learning controllers are applied in combination.

The paper is organized as follows. In the next section, we summarize the relevant literature. We also indicate what is innovative in our concept of joining model-based control and ILC. In Section III, we formulate a model of both the rigid body dynamics and the flexible robot dynamics. Identification of the robot dynamics is considered in Section IV. The design of the iterative learning controller is explained in Section V. In Section VI, we present the robotic setup used in the experiments, and the experimental results are given in Section VII. The conclusion and final remarks are given in Section VIII.

II. MODEL- AND LEARNING-BASED ROBOT MOTION CONTROL

Opportunities in model-based robot control are boosted by the computational power of modern digital processors, by advances in the theory on robot modeling and identification [1], [3]–[6], and by incorporation of control theory in general. A model may be used to compensate for nonlinear dynamic couplings between robot axes, enabling linear robust feedback control designs that ensure accurate and disturbance-free robot operation in an outer loop [7]. A model improves the control performance to the extent it matches the real robot dynamics. A model is of sufficient quality, if its structure describes all relevant aspects of the physics (i.e., inertial, Coriolis/centripetal, gravity, and friction effects), and if its parameters fit the real values.

So far, a number of theoretical and experimental studies on estimating the model parameters have been reported. Each exploits the well-known property that a model of the rigid body robot dynamics can be represented linearly in a set of identifiable parameters, called the base parameter set (BPS) [3]. The elements of the BPS are nonlinear combinations of robot inertial parameters, such as masses and moments of inertia of the robot links, as well as the Cartesian coordinates of the link centers of masses. Since friction can cause control problems (e.g., static errors, limit-cycles) [8]–[10], the model should also account for friction effects. The friction force can be counteracted using model-based [8], [9] or nonmodel based techniques [10]. If model-based techniques are used, then the relevant values of the friction parameters are needed in addition to the BPS.

In general, we can distinguish between least-squares-like [4], [5], maximum-likelihood [4], [6], and adaptive techniques [11], [12] for estimation of the BPS elements and friction parameters. If the inertial and friction parameters are jointly estimated, then one should consider a friction model that admits a linear representation in its parameters.

In this paper, we use a particular batch adaptive control technique for the problem of estimating both the BPS and the friction model parameters. The control technique was originally proposed in [13], as a way to accurately realize a repetitive reference trajectory, when the correct parameters of the robot dynamic model with friction are unknown. A newer version of this technique has appeared in [14], improving the robustness against disturbance and parasitic conditions, as well as speeding-up the convergence characteristics. Being computationally cheap, the batch adaptive control technique admits a relatively simple online implementation. When it realizes a reference trajectory which excites the robot dynamics persistently [12], the technique can deliver accurate estimates of the BPS elements and the friction parameters. Design rules for a persistently exciting trajectory, later on referred to as the excitation trajectory, are well known in robotics [4], [5]. Among others, the design suggested in [4] results in a cyclic excitation trajectory of which the frequency resolution and the frequency content can be directly assigned. Control of the frequency content is an important precaution against excitation of dynamics that is not covered by the model (parasitic dynamics) of which the BPS is estimated. The design suggested in [4] is promoted in this paper, too. A preliminary presentation of the parameter estimation using batch adaptive control can be found in [15].

The batch adaptive algorithm has several advantages with respect to other identification procedures, e.g., least-squares [4], maximum-likelihood [4], [6], and normal adaptive techniques [11]. First of all, the algorithm admits a relatively simple online implementation, which contributes to the efficiency of identification. Consequently, variable robot loads can be easily taken into account. While the maximum-likelihood technique of [6] searches for estimates using a nonconvex optimization strategy, the batch adaptive algorithm takes advantage of the model representation that is linear in the parameters, and which facilitates the convexity of the estimation. Consequently, the batch adaptive algorithm is more efficient than the maximum-likelihood technique. When compared with the standard least-squares estimation, the batch adaptive algorithm features higher robustness against measurement noise and against the initial guess of the estimates, together with fast convergence of the estimates to the steady-state values [14].

It is recognized that a model-based approach is capable of improving the performance of motion control [1], [7]. It is evident that the control quality is sensitive to modeling errors, e.g., parameter inaccuracy or neglected (unmodeled) dynamics. Stabilization and high performance motion control can be realized by augmenting the model-based controller with feedback loops, that are shaped such to achieve high-gain feedback [7] in the frequency range of the reference trajectory, and rejection of disturbances and parasitics beyond that range. Because of the high feedback gain, the robot joints are usually very stiff, while the feedback controllers are typically of high order. If the reference trajectory is repetitive, which is common for industrial robots, then motion control of high quality can be achieved by means of ILC. With ILC, there is no need for high-gain feedback control. The combination of model-based control and ILC will result in high performance control for a given repetitive trajectory.

After being introduced in robotics [16], ILC has evolved into a control discipline utilizing a number of algorithms [1], [13], [14], [17]–[19]. Coarsely speaking, variants of ILC proposed for robotic problems have become almost as numerous as the number of practitioners applying ILC in robotics. Commonly, the ILC design requires very little knowledge about the robot dynamics, either looking for just an approximate dynamic model which parameters are only roughly known [1], [13], [14], [17], or even considering only linearized dynamics [18], [19]. Regarding the domain in which the learning controller is designed, a distinction can be made between time-domain [1], [13], [14],
[16], and frequency-domain ILC techniques [18], [19]. Commonly, the time-domain techniques only make use of rigid robot dynamics. Their rules for tuning the learning controllers are usually rather conservative, since they do not allow one to predict the maximum effect of ILC on the tracking performance [20]. On the other hand, frequency-domain techniques can take into account additional knowledge about the robot dynamics, e.g., about dynamics not covered by the model-based compensation. Hence, frequency-domain techniques enable a formalism for designing the learning controller. A limitation is that the controller design is based on the linearized robot dynamics, so nonlinear effects cannot be directly integrated.

In this paper, we consider a frequency-domain ILC technique. The design of the learning controller is based on the linearized dynamics that remain after decoupling the robot with a model-based controller. Although already suggested as a possible solution for robots whose dynamics feature high nonlinearities, like direct-drive robots [18], it seems that the combination of nonlinear model-based and ILC has not been practically investigated yet. To the best of our knowledge, this paper is the first study of ILC applied to direct-drive robots, where the frequency-domain design of the learning controllers is enabled by nonlinear model based compensation. Preliminary results of this study were presented in [21]. A notable difference between other frequency-domain techniques used in robotics and our approach in this paper, is that we provide a formalism for tuning the learning filters based on the dynamics that are experimentally measured on the robot.

A direct-drive robot with three revolute joints was used to experimentally demonstrate the considered techniques for parameter estimation and ILC. The robot joints are implemented as a waist, shoulder, and elbow, which is a kinematic structure, often met in industry. Therefore, the results to be presented in this paper should be representative for industrial case studies.

III. ROBOT DYNAMICS

We represent the rigid-body dynamics of a general serial direct-drive robotic manipulator with \( n \) actuated joints using the Euler–Lagrange formalism [22]

\[
D(\theta(t))\ddot{\theta}(t) + c(\dot{\theta}(t), \ddot{\theta}(t)) + g(\theta(t)) + f(\theta(t)) = \tau(t)
\]

(1)

where \( \theta, \dot{\theta}, \ddot{\theta} \) are the \( n \) vectors of joint motions, speeds, and accelerations, respectively, \( D \) is the \( n \times n \) inertia matrix, \( c, g, f \) are the \( n \) vectors of Coriolis/centripetal, gravity, and friction effects, respectively, and \( \tau \) is the \( n \) vector of control inputs (joint forces/torques). We assume that the analytical expressions of \( D, c, g, f \) are readily available, while the parameters involved in these expressions have to be determined.

The objective of the motion control problem is steering the joint motions \( \dot{\theta}(t) \) along the reference trajectory \( \dot{\theta}_r(t) \). This problem can be solved by using the model (1) in the following nonlinear model-based compensator:

\[
\tau(t) = D(\theta_r(t))\dot{u}(t) + c(\theta_r(t), \ddot{\theta}_r(t)) + g(\theta_r(t)) + f(\theta_r(t))
\]

(2)

where \( u \) represents the \( n \) vector of new control inputs that should ensure stable robot motions and accurate tracking of the reference. Assuming \( \theta_r(t) \) is known, \( u \) can be postulated as

\[
u(t) = \ddot{\theta}_r(t) + u_{fb}(t) + u_{ILC}(t)
\]

(3)

where \( \ddot{\theta}_r(t) \) is the reference acceleration, \( u_{fb}(t) \) is the control input calculated by a suitable feedback controller, and \( u_{ILC}(t) \) is the output of the iterative learning controller. The implementation of the motion controller (2), (3) is illustrated in Fig. 1. To design the feedback and learning controllers, we need the robot dynamics that are not covered by the nonlinear model-based compensator (2). To design an iterative learning controller, a parametric representation of these dynamics is required.

Ideally, when the controller (2) is applied to the dynamics (1), \( n \) linear plants remain

\[
\ddot{\theta}(t) = u(t)
\]

(4)

This holds only if \( \theta(t) = \theta_r(t), \dot{\theta}(t) = \dot{\theta}_r(t), \) and \( \ddot{\theta}(t) = \ddot{\theta}_r(t) \). In practice, this rarely happens, especially in the presence of flexible effects that are not included in (1). As discussed in [7] and [23], the dynamics in the robot joints are more complex than the ideal one in (4). In practice, the model-based compensation strongly reduces the robot nonlinearities, but it hardly eliminates the nonlinear effects completely. Although present, the remaining nonlinearities are usually less significant for motion performance than the parasitic robot dynamics, i.e., the dynamics not covered by the rigid-body model. Therefore, the remaining nonlinear effects are not considered hereafter, while the linear parasitic dynamics in joint \( i \) can be represented by the transfer function

\[
P_i(s) = \frac{\Theta_i(s)}{U_i(s)}
\]

(5)

with \( s \) representing the Laplace variable. In (5), \( P_i \) is not just a double integrator, as suggested by (4), 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. Instead, we choose a strategy of adopting a nominal model to represent an average of the linearized dynamics. Differences from the nominal model are interpreted as perturbations in the dynamics (model uncertainty) and have to be counteracted by ILC. This strategy is discussed in Section IV-B.

![Fig. 1. Motion control using model-based and ILC.](image-url)
IV. LEARNING-BASED PARAMETER IDENTIFICATION

In this section, we first suggest identification of the parameters of the robot rigid-body dynamic model with friction using a batch adaptive control algorithm [13], [14]. As mentioned in Section II, the algorithm is applicable to the rigid-body dynamic model with friction when this linear is in the unknown parameters. The unknown friction coefficients are concatenated with the elements of the BPS, forming the complete parameter vector to be estimated. The next subsection will briefly describe the batch adaptive control algorithm. For a more detailed description, the reader is referred to [14]. Second, the parameterized modeling of the flexible dynamics, obtained after applying the motion controller (2), (3), and needed for the design of the iterative learning controllers, is discussed in Section IV-B.

A. Parameter Estimation via Batch Adaptive Control

A representation of (1) linear in the parameters is

$$R\left(\theta(t), \dot{\theta}(t), \ddot{\theta}(t)\right) p = \tau(t) \quad (6)$$

where $R \in \mathbb{R}^{n \times m}$ is the regression matrix, $p \in \mathbb{R}^m$ is the vector of BPS elements and friction parameters, with $m$ the number of parameters to be estimated. The robot can be stabilized using a conventional PD feedback controller

$$\tau(t) = K_p e(t) + K_d \dot{e}(t) + v(t) \quad (7)$$

where $K_p$ and $K_d$ are $n \times n$ diagonal matrices of positive proportional and derivative gains, respectively, $v$ the $n$ vector with new input signals from the batch adaptive controller, and $e$ the tracking error

$$e(t) = \theta_r(t) - \theta(t). \quad (8)$$

If (7) is applied to (6), then the closed-loop system takes the form

$$R\left(\theta(t), \dot{\theta}(t), \ddot{\theta}(t)\right) p - K_p e(t) - K_d \dot{e}(t) = v(t) \quad (9)$$

When all motions, speeds, and accelerations in (9) are chosen to be reference ones, the input signal $v_r$ is obtained

$$R\left(\theta_r(t), \dot{\theta}_r(t), \ddot{\theta}_r(t)\right) p = v_r(t) \quad (10)$$

and the residual error dynamics, with input $\delta v(t) = v_r(t) - v(t)$, is given by

$$R\left(\theta_r(t), \dot{\theta}_r(t), \ddot{\theta}_r(t)\right) p - R\left(\theta(t), \dot{\theta}(t), \ddot{\theta}(t)\right) p + K_p e(t) + K_d \dot{e}(t) = \delta v(t). \quad (11)$$

For appropriate choices of $K_p$ and $K_d$ [13], the residual error dynamics can be made passive for the output

$$\delta y = \dot{e} + \beta s(e) \quad (12)$$

where $\beta$ is a positive constant scalar, and $s(e)$ is an $n$ vector of gradients of a potential function. For a suitable choice of this potential function, the reader is referred to [13] and [14].

Before showing a batch adaptive update law, suitable for the estimation of the BPS elements and parameters of the friction model, several definitions and assumptions are needed

$$R_r := R\left(\theta_r(t), \dot{\theta}_r(t), \ddot{\theta}_r(t)\right) \quad (13)$$

$$L_r := \int_0^{t_f} R_r^T R_r dt. \quad (14)$$

Here, $t_f$ denotes the duration of one trial of the repetitive trajectory. The matrix $L_r$ is assumed to be nonsingular, which holds if $p$ in (6) contains a minimum number of BPS elements and friction parameters, and if the reference trajectory excites the robot dynamics persistently.

The batch adaptive update law is given as follows [14]:

$$p^{k+1} = p^k + K_r (R_r \delta y^k) \phi \quad (15)$$

where $k$ denotes the number of the trial ($k = 0, 1, 2, \ldots$), $K_r$ is an $m \times m$ positive definite matrix called the adaptation gain, and $\langle R_r, \delta y^k \rangle_\Phi$ represents the inner product between $R_r$ and $\delta y^k$, weighted by a positive definite matrix $\Phi \in \mathbb{R}^{n \times n}$. For two matrices of equal number of rows, this inner product is defined as

$$\langle U, V \rangle_\Phi = \begin{bmatrix} \langle u_1, v_1 \rangle_\Phi & \langle u_1, v_2 \rangle_\Phi & \cdots \\ \langle u_2, v_1 \rangle_\Phi & \langle u_2, v_2 \rangle_\Phi & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (16)$$

where $u_i$ and $v_j$ are the $i$th column of $U$ and the $j$th column of $V$, respectively. The weighting matrix $\Phi$ is omitted if it is identity. The inner product of two columns is defined as follows:

$$\langle u_i, v_j \rangle = \int_0^{t_f} u_i^T (t) \Phi v_j(t) dt. \quad (17)$$

If the adaptation gain $K_r$ satisfies

$$K_r = L_r^{-1} (R_r, R_r) L_r^{-1} \quad (18)$$

then the convergence of the tracking error (8)

$$\| e^k(t) \| \rightarrow 0 \quad (k \rightarrow \infty) \quad (19)$$

can be proven [14], with $\| \|$ denoting a signal norm. The convergence of the tracking error implies convergence of the elements in $p^k$ to steady-state values ($p^k = p^{k+1}$). Theoretically, there is no restriction on the initial guess of $p^0$, required in (15). Practically, when control inputs $\tau$ are bounded, $p^0$ cannot be chosen arbitrarily, but should be constrained to a certain feasible region corresponding to limits on the control inputs.

In the algorithm presented, the tuning parameters are contained in the matrices $K_p$, $K_d$, and $\Phi$. The coefficient $\beta$ should be tuned as well, and there is also a possibility to use various $s(e)$ by choosing different potential functions, as long as these functions satisfy the properties specified in [13] and [14]. The
PD gains should ensure passivity of the robot dynamics when realizing the reference trajectory. The matrix \( \Phi \), the scalar \( \beta \), and the vector function \( s(e) \) influence the rate of convergence to the steady-state values of \( p \). Inequality conditions that must be fulfilled when tuning \( \Phi \) and \( \beta \) can be found in [14].

1) Design of Excitation Trajectories: With the algorithm presented in the previous subsection, all elements of the vector \( p^k \) converge after a sufficient repetition of trials. We make use of this property to obtain an estimate of \( p \) that provides a close match between the model (1) and the real robot dynamics. Obtaining such an estimate is the goal of robot identification, and it can be obtained if the robot realizes an appropriate excitation trajectory.

To design such a trajectory, one should optimize some property of the information matrix \( \Pi \)

\[
\Pi = \begin{bmatrix}
R \left( \theta_r(t_1), \dot{\theta}_r(t_1), \ddot{\theta}_r(t_1) \right) \\
R \left( \theta_r(t_2), \dot{\theta}_r(t_2), \ddot{\theta}_r(t_2) \right) \\
\vdots \\
R \left( \theta_r(t_f), \dot{\theta}_r(t_f), \ddot{\theta}_r(t_f) \right)
\end{bmatrix}
\]  

This matrix is formed by vertically stacking the matrices \( R \) that correspond to different points of the excitation trajectory. Some choices of the performance criteria to be optimized are the condition number of the information matrix, its extreme singular value, its Frobenius condition number, or the determinant of the weighted product between the information matrix and its transpose [5]. In our opinion, “a good trajectory” is one that provides us with estimates that are relevant for at least those motions that the robot is supposed to realize in practice.

An excitation trajectory that meets the given requirements can be determined by postulating it in the form of a finite Fourier series [4]

\[
\theta_{r,i}(t) = \theta_{0,i} + \sum_{j=1}^{N} \frac{1}{j\omega} \left[ a_{i,j} \sin(j\omega t) - b_{i,j} \cos(j\omega t) \right],
\]

\((i = 1, 2, \ldots, n)\)  

with the prescribed fundamental frequency \( \omega \), and the parameters \( \theta_{0,i}, a_{i,j}, \) and \( b_{i,j} \) to be computed via optimization. Constraints on mechanical limits in the joint robots and permissible levels of joint speeds and accelerations, must be taken into account. The fundamental frequency defines the resolution of the frequency content of the excitation trajectory. Together with the number of harmonics \( N \), which affects persistency of excitation, the fundamental frequency determines the frequency content of excitation. This content should be rich enough to cover all possible motions of the robot, but also well below the eigen-frequencies of the first flexible mode of the robot structure.

No matter which property of \( \Pi \) is optimized via \( \theta_{0,i}, a_{i,j}, \) and \( b_{i,j} \), the resulting optimization problem is essentially non-convex, which means that the global optimum for the excitation trajectory cannot be obtained easily. Different initial conditions may lead to different excitation trajectories, all corresponding to local minima of the optimization problem. Our experience [23], though, shows that even suboptimal excitation trajectories provide satisfactory estimates.

B. Parameterized Modeling of Flexible Dynamics

A nominal parametric model \( P^0_i \), needed for the ILC design, can be determined based on the frequency response functions (FRFs) directly measured on the robot. A procedure to measure these functions is explained in [7]. In short, FRFs for each joint are measured separately in a single-input–single-output (SISO) framework and under closed-loop operating conditions. Each FRF is computed assuming \( u_i(t) \) as excitation and \( \theta_i(t) \) as response. The model \( P^0_i \) is based on FRFs that correspond to different robot configurations. For instance, assume \( N \) FRFs \( G_i(j\omega), (l = 1, \ldots, N) \), have been determined. Then, there are at least two possibilities to calculate the nominal FRF \( G_i^0(j\omega) \)

\[
G_i^0(j\omega) = \text{arg} \min_{G_i(j\omega)} \max_l \left| \frac{G_i(j\omega) - G_i^l(j\omega)}{G_i(j\omega)} \right| 
\]

(22)

or

\[
G_i^0(j\omega) = \frac{1}{N} \sum_{l=1}^{N} G_i^l(j\omega).
\]

Possibility (22) minimizes the distance between the nominal and the \( N \) measured FRFs at each frequency \( \omega \), while (23) calculates their average. We choose (23), because it provides smoother magnitude and phase characteristics. Smoother data will facilitate fitting the parametric model \( P^0_i \) onto the data \( G_i^0 \). Perturbations from \( P_i^0 \) will be taken into account during the ILC design. The fitting itself can be done in the least-squares sense, using an output-error model structure [24]. The model order is up to the designer, whose objective is to achieve a sufficient match between \( G_i^0(j\omega) \) and the Bode plot of \( P_i^0 \). The obtained model will be used in the ILC design, which will be explained in the next section. As this design is performed offline, the order of \( P_i^0 \) is typically not critical for the design. In practice, arbitrarily high model orders might not be admissible due to numerical problems. Consequently, the designer chooses the order such that, on one hand, a reasonable match between the measured data and the model is achieved, and, on the other hand, no numerical problems arise.

V. ILC

This section formulates the design of the iterative learning controllers in the frequency domain. Consider the block diagram in Fig. 2. The SISO linear plant \( P_i \) is described by its transfer function (5). The feedback controller \( C_i \) generates the feedback control input \( u_{p,i}(t) \), according to (3). Apart from stabilization, the feedback controller is used to generate position errors that are necessary to initialize the ILC algorithm. The input \( u_{i,i}(t) \)
represents the output of the learning controller, and it is updated after each trial \( k \).

In the Laplace domain,\(^1\) the update rule of the learning algorithm is defined as follows \([25]\):

\[
U^{k+1}_{lde,i}(s) = Q_i(s) \left( U^k_{lde,i}(s) + L_i(s) E^k_i(s) \right) \tag{24}
\]

where \( k \) is the number of the trial, \( L_i \) is the learning filter, and \( Q_i \) is the robustness filter. According to Fig. 2, the errors in two successive trials can be calculated from the closed-loop transfer functions from the control input, \( U_{lde,i} \), to the tracking error, \( E_i \)

\[
E^k_i(s) = - P_i(s) S_i(s) U^k_{lde,i}(s) + S_i(s) \Theta_{r,i}(s)
\]

\[
E^{k+1}_i(s) = - P_i(s) S_i(s) U^{k+1}_{lde,i}(s) + S_i(s) \Theta_{r,i}(s)
\]

(25)

(26)

where \( S_i \) denotes the sensitivity function:

\[
S_i(s) = \frac{1}{1 + P_i(s) C_i(s)} \tag{27}
\]

and the product \( P_i(s) S_i(s) \) is the process sensitivity. Substitution of (24) and (25) into (26), results in

\[
E^{k+1}_i(s) = E^k_i(s) Q_i(s) \left( 1 - L_i(s) P_i(s) S_i(s) \right) + S_i(s) \left( \Theta_{r,i}(s) - Q_i(s) \Theta_{r,i}(s) \right) - P_i(s) S_i(s) \\
\times \left( \Theta_{r,i}(s) - Q_i(s) \Theta_{r,i}(s) \right)
\]

\[
= E^k_i(s) Q_i(s) \left( 1 - L_i(s) P_i(s) S_i(s) \right) + S_i(s) \\
\times \left( 1 - Q_i(s) \right) \left( \Theta_{r,i}(s) - P_i(s) \Theta_{r,i}(s) \right). \tag{28}
\]

If the second term on the right-hand side of (28) is negligible, then the condition for the amplitude of the error in trial \( k+1 \) to be smaller than the amplitude of the error in trial \( k \) is given by

\[
\| Q_i(s) \left( 1 - L_i(s) P_i(s) S_i(s) \right) \|_{\infty} < 1 \tag{29}
\]

with \( \| \cdot \|_{\infty} \) denoting the infinity norm. The inequality (29) is known as the condition for error convergence toward zero. The contribution of the second term on the right-hand side of (28) can be minimized by suitable design of the \( Q \)-filter, as explained in the following. To maximize the speed of error convergence, the learning filter \( L_i \) should be identical to the inverse of the process sensitivity

\[
L_i(s) = \frac{1}{P_i(s) S_i(s)}. \tag{30}
\]

To compute the \( L \)-filter, a parametric model of the process sensitivity has to be made. By virtue of (27) and (30), the order of the \( L \)-filter depends on the orders of the plant model and the controller. As discussed in Section IV-B, a high-order process sensitivity model is acceptable, as long as it does not cause numerical problems. In the next section, we will present an example of the plant model and its corresponding process sensitivity model. If the plant \( P_i \) is nonminimum phase, the \( L \)-filter will be unstable if the process sensitivity is inverted directly. To avoid this, the zero phase error tracking algorithm for digital control \([26]\) can be used. This algorithm provides a stable approximation of the inverted process sensitivity.

The role of the robustness filter \( Q_i \) is to ensure error convergence in the presence of modeling errors. Namely, the \( L \)-filter is based on a parametric model which, in practice, cannot cover all dynamics of the real system, but only some low-frequency part. The \( Q \)-filter should guarantee the validity of (29) for the frequency components at which the model is inaccurate. Since in practice modeling errors arise in the high-frequency range, \( Q_i \) is typically chosen to be a low-pass filter. Below its cutoff frequency \( f_Q \), the \( Q \)-filter has a passband equal to 1, while above \( f_Q \) its magnitude is decreasing. Thus, within the passband of \( Q_i \), the second term on the right-hand side of (28) diminishes. Since this term depends on the reference joint trajectory, of which the harmonic content is typically within the passband of the \( Q \)-filter, it also has negligible contribution to the error outside the passband of the \( Q \)-filter.

The \( Q \)-filter should influence the frequency content of \( u^{k+1}_{lde} \) only by its magnitude characteristics and should not introduce any phase distortion. To explain this further, let us discuss the error (28). The objective of learning is to achieve the error in trial \( k+1 \) to be smaller than the error in the previous trial, i.e., \( |e^{k+1}_i| < |e^k_i| \). The convergence criterion (29) ensures decrease of the part of \( e^{k+1}_i \) which directly depends on \( e^k_i \). As already mentioned, the second term on the right-hand side of (28) is canceled if \( Q_i \) is a real scalar equal to 1. Therefore, the \( Q \)-filter must not introduce any phase shift, especially within its passband, since this would violate the condition \( Q_i \equiv 1 \). Consequently, the \( Q \)-filtering is a noncausal operation that is performed offline, as indicated in Fig. 2.

VI. EXPERIMENTAL SETUP

The robotic arm, shown in Fig. 3, is the subject of our case study. It is an experimental facility for the research in motion control \([27]\). The photo and kinematic parameterizations according to the well-known Denavit–Hartenberg (DH) notation \([28]\), reveal three revolute degrees of freedom (DOF), which makes such a kinematic structure referred to as RRR. Each DOF is actuated by a gearless brushless dc direct-drive motor and has an infinite range of motions, thanks to the use of sliprings for the transfer of power and sensor signals. The actuators are

---

\(^1\)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 \([25]\).
Dynaserv motors 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]. Both amplifiers and encoders are connected to a PC-based control system. This systems consists of a MultiQ I/O board from Quanser Consulting (8 × 13-b analog-to-digital converter (ADC), 8 × 12-b DAC, 8 digital I/O, 6 encoder inputs, and 3 hardware timers), combined with a real-time controller for Matlab/Simulink (Wincon). This facilitates the design of controllers in Simulink and their real-time implementation. The control system features a time delayed joint angular response to the given control input. For a sampling time of $T_s = [1 \text{ ms}]$, there is a delay of $\delta = 2T_s$ [23].

Detailed closed-form models of the robot kinematics and dynamics are available in [23]. From Fig. 3, one can determine the DH parameters, whose numerical values are presented in Table I. The DH parameters are: twist angles $\alpha_i$, link lengths $a_i$, joint displacements $\theta_i$, and link offsets $d_i$.

### VII. EXPERIMENTAL RESULTS

This section contains the results of the experiments conducted to show the effectiveness of the identification procedure, presented in Section IV, as well as the performance improving capabilities of the ILC algorithm discussed in Section V.

#### A. Identification Results

1) Parameter Estimation Results: The rigid body dynamics of the RRR-robot, as presented in [23], feature 15 BPS elements. Due to the symmetry in distribution of masses in the second and the third robot link, two BPS elements are identically equal to zero [23]. Consequently, only 13 BPS elements had to be identified using the strategy presented in Section IV. The adopted friction model covers Coulomb and viscous effects

$$ f(\dot{\theta}(t)) = F_c \text{sign}(\dot{\theta}(t)) + B \dot{\theta}(t) \quad (31) $$

where $f$ is a 3 vector of friction torques, while $F_c$ and $B$ are $3 \times 3$ positive definite diagonal matrices of Coulomb and viscous friction parameters, respectively. This friction model admits a linear parametrization, which facilitates joint identification of the BPS elements and friction parameters.

The excitation trajectories, needed in the identification experiments, were calculated using (21). For all three joints, the fundamental frequency was chosen to be 0.1 [Hz], which is usually taken as sufficient [4]. The number of harmonics $N$ was 100, so the frequency contents were constrained up to 10 [Hz], which is sufficient to cover operations the RRR-robot performs and to not excite flexible dynamics. The tuning parameters $\beta_{0,i}$, $\alpha_{i,j}$, and $\beta_{i,j}$ were computed by optimizing the condition number of the information matrix $\mathbf{P}$, and the constraints were

$$ \begin{align*}
|\beta_1| & \leq 2\pi/\text{[rad/s]}, \\
|\beta_2| & \leq 3\pi/\text{[rad/s]}, \\
|\beta_3| & \leq 3\pi/\text{[rad/s]}, \\
|\dot{\theta}_i| & \leq 2\pi/\text{[rad/s]}, \\
|\dot{\theta}_j| & \leq 3\pi/\text{[rad/s]}^2, \\
\end{align*} \quad (i = 1, 2, 3). \quad (32) $$

After optimization, the motions of the excitation trajectories, depicted in Fig. 4, were obtained.

In order to demonstrate the full potential of the estimation algorithm from Section IV-A, three different identification experiments were performed. In the first experiment, no load has been attached to the third robot joint, in the second experiment, a steel rod of 1.22 [kg] (Fig. 3) has been attached, whereas in the third one, an extra load of 0.45 [kg] has been attached onto the tip of the steel rod. Before all three experiments, the initial value of the vector of parameters to be estimated, $\mathbf{p}^0$, was set to zero. Consequently, the extra input $\mathbf{v}$ in (7) was also zero during the first trial, see (15).

For online implementation of the batch adaptive algorithm, the feedback controller gains $\mathbf{K}_p$ and $\mathbf{K}_d$ in (7) were weakly tuned for each load separately, such as to stabilize the system and to avoid excitation of parasitic dynamics (flexibilities). The scalar $\beta$ in (12) and the weighting matrix $\mathbf{P}$ in (15), were tuned such as to achieve fast convergence of the estimates to steady-state values.

Fig. 5 shows the results of the identification experiments with different loads. The parameters $p_{11}, \ldots, p_{13}$ represent the BPS estimates, whereas the parameters $p_{14}, \ldots, p_{19}$ represent the estimates of the friction parameters. By inspection of the given plots, it can be seen that the parameters are updated at the end of each trial, i.e., after every 10 [s]. Almost all parameters have reached their steady-state values after not more than eight trials. It can be also noticed that the steady-state estimates of $p_{21}, p_{31}, p_{41}, p_{51}, p_{11}, p_{12}$, and $p_{33}$, are not the same when estimated with different loads. This can be expected, as these BPS elements depend on the inertial properties of the third robot link [23]. Obviously, the variations in the mass attached to the third joint, induces differences in the estimates.

### TABLE I

<table>
<thead>
<tr>
<th>dof</th>
<th>$\alpha_i$ [rad]</th>
<th>$a_i$ [m]</th>
<th>$\theta_i$</th>
<th>$d_i$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\pi/2$</td>
<td>0</td>
<td>$\theta_1$</td>
<td>0.560</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>$P_1O_2=0.200$</td>
<td>$\theta_2$</td>
<td>0.169</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>$P_2O_3=0.415$</td>
<td>$\theta_3$</td>
<td>0.090</td>
</tr>
</tbody>
</table>

![Fig. 4](image-url) - Motions of the excitation trajectories: $\theta_1$—thick solid, $\theta_2$—thin solid, and $\theta_3$—dashed.
To evaluate if the estimates provide a close match between the model (1) and the real robot dynamics for all three loads, validation experiments have been conducted. In these experiments, the RRR-robot performed a writing task, suggested in [29], and depicted in Fig. 6. This task is recognized as demanding for the robot dynamics, due to fast motions with discontinuous accelerations. The task consists of writing the presented sequence of letters in the vertical plane, as shown in Fig. 6(a). The corresponding joint motions are depicted in Fig. 6(b).

During the validation experiments, the robotic arm has been stabilized using a conventional PD feedback controller

$$\tau(t) = K_p e(t) + K_d \dot{e}(t)$$

where $K_p$ and $K_d$ are $n \times n$ diagonal matrices of positive proportional and derivative gains, respectively. To verify agreement of torques rendered by the model (1) and measured torques under PD control (33), the joint motions, speeds, and accelerations were reconstructed online using a Kalman observer described in [23]. In Fig. 7, outputs of the PD controllers of the second joint have been depicted for the three different validation experiments, together with the torques reconstructed using (1). From this figure, it can be seen that in all cases, the torques rendered by the model closely match the outputs of the PD controllers, which indicates that relevant estimates have been obtained.

Small deviations between the output of the PD feedback controller and the reconstructed torques can be seen when the torques peak, corresponding to the changes in the sign of the joint speed. These indicate that the adopted friction model does not compensate properly for the real friction. In Fig. 7(c), it can be seen that the output of the feedback controller is clipped at its maximum permissible value (30 [Nm]). The saturation occurs as the extra load attached to the robot tip requires more actuation power than permissible. Under normal operating conditions, i.e., when no extra load has been attached to the robot tip [Figs. 7(a) and 7(b)], the saturation effect does not occur.

2) Parameterized Models of Flexible Dynamics: For clarity, it is pointed out that the results discussed in this subsection have been obtained using the RRR-robotic arm with the third link connected to the third joint, but without the extra mass attached to the robot tip. After applying the nonlinear model-based compensator (2) and stabilizing the robot using a PD feedback control law

$$u_{th} = K_p e + K_d \dot{e}$$

with $K_p$ and $K_d$ again representing diagonal matrices of positive proportional and derivative gains, the remaining dynamics have been identified in all three joints. We explain the identification results of these remaining flexible dynamics in the first joint. However, the experimental results given in this subsection...
Fig. 7. Results of the validation experiments. (a) No load attached to the third joint. (b) Steel rod attached to the third joint. (c) Steel rod and an extra mass attached to the third joint.

Fig. 8. Experimentally obtained FRFs $G_{1}^{0}$, $G_{1}^{10}$ (gray) and the nominal FRF $G_{1}^{0}$ (black).

The FRFs have been measured under closed-loop conditions, since the plant itself is not asymptotically stable. White noise is added to the control input $u_{\text{in}}$, and the transfer function from the white noise signal to the total control input signal, the sensitivity function, is measured. Since the stabilizing feedback controller (34) is known, the remaining plant dynamics can be extracted from this sensitivity function. The results of this identification procedure are reliable outside the bandwidth of the closed-loop system, since in that range the coherence of the measurement is good [7]. Fig. 8 shows that the theoretically expected dynamics of a double integrator (4) holds at low frequencies only (below 20 [Hz]), while the real dynamics are more involved at higher frequencies. Also, up to 4 Hz, the slope of the magnitudes is less steep than $-2$. This frequency range is within the bandwidth of the closed-loop system established via (34). Because of the authority of the PD controller, we cannot achieve sufficient coherence of the measurements within the closed-loop bandwidth of 4 [Hz]. Consequently, the part of the identified FRF up to 4 [Hz] is not reliable. Note the resonance around 28 [Hz], caused by vibrations at the robot base, and more profound resonances at higher frequencies. The phase has a lag growing with increasing frequency, which is caused by the time delay in the control system [23]. The nominal FRF $G_{1}^{0}$, calculated according to (23), is shown in Fig. 8 as well. It is also depicted in Fig. 9, together with its parametric fit $F_{2}^{1}$. As obvious from this figure, there is a discrepancy between the unreliable data below 4 [Hz] and the fit. The order of the model is 8, which allows an accurate fit between 4 and 70 [Hz]. Fitting the high frequent dynamics is omitted, as we do not expect it would be possible to learn above 70 [Hz].

B. ILC Results

In this subsection, the ILC design for the first joint is demonstrated, but the experimental results cover the case with learning controllers for all three joints.

A PD feedback controller $C_1$, with $K_{p,1} = 1000$ and $K_{d,1} = 2\sqrt{1000}$, is used to stabilize the first robot joint. Given $C_1$ and
In order to preserve the convergence criterion (29) in the frequency ranges where the $L$-filter mismatches the inverse of the process sensitivities, the robustness filter $Q$ is used, as explained in Section V. This filter is chosen to be a fourth-order low-pass digital Butterworth filter with a passband magnitude equal to one. The cutoff frequency $f_Q$ of the $Q$-filter can never exceed the frequency up to which the $L$-filter is a good representation of the inverse of the process sensitivity model. In determining the optimal cutoff frequency, the convergence criterion (29) has to be evaluated for all process sensitivities calculated from the $N$ measured FRFs, rather than for the nominal one only, to be sure that the convergence criterion is satisfied for the complete set of robot postures. The cutoff frequency of the first $Q$-filter is chosen to be 18 [Hz]. The cutoff frequencies of the $Q$-filters for the second and third joint have been derived similarly and are 25 [Hz] and 27 [Hz], respectively. These frequencies can be compared with the crossovers of the three SISO closed-loop systems. The crossover frequency is defined as the first 0 [dB] crossing of the open-loop gain $P_i(s)C_i(s)$ ($i = 1, 2, 3$). Since the crossover frequencies of the three SISO systems are all approximately equal to 9 [Hz], we can conclude that learning takes place well above the crossovers.

To demonstrate the effectiveness of the proposed control design, experiments have been conducted. In these experiments, the robotic arm had to perform the writing task, which has also been used in the validation experiments discussed in Section VII-A. The reference trajectory has been modified slightly, in order to obtain a repetitive trajectory. In this writing task, shown in three dimensional space in Fig. 11(a), the robot-tip had to track the path starting from “A” in the direction of the arrows. The corresponding joint motions are depicted in Fig. 11(b).

During the experiments, the robot motions have been stabilized by the PD feedback controllers that have also been used in the derivation of the $L$-filters. After each trial, the new ILC input is calculated according to (24). This means filtering of the measured tracking errors with the corresponding $L$-filters, adding the previous ILC input to this filtered signal, and, finally, filtering of the obtained signals with the corresponding $Q$-filters. The result is the ILC input for the next trial. No numerical problems have been experienced during the filtering operations, despite the fairly high orders of the filters involved.

The performed experiment consisted of 18 trials. Since the errors had converged after 10 trials, and no further accumulation of error powers was observed during the other trials, we discuss
the results up to trial 10. In Fig. 12, 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. Final tracking errors in the tenth trial are approximately 10 times smaller than the errors in the first trial.

As depicted in Fig. 13, 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 one trial more. The root-mean-square values of the errors after the learning process are over ten times smaller than before learning.

For the frequency analysis of the tracking errors in joint space, auto power spectra of these errors have been calculated. Fig. 14 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 a significant reduction of frequency components until approximately 10 [Hz] in the first two joints and until 12 [Hz] in the third joint. This is very important as the frequency content of the reference motions is within these ranges. It can also be seen that the frequency content of the errors in the high frequency range in the tenth trial does not differ from the frequency content in the first trial, which was expected, since no learning takes place in
this range. From Fig. 14(a) and (b), it can be seen that at the end of the learning process, the errors in the first and in the second joint have slightly amplified power between 13 [Hz] and 20 [Hz]. The amount of amplification is not an issue for the motion performance. However, it is peculiar why amplification occurs in the frequency range within the bandwidth of the corresponding robustness filters (18 [Hz] for joint 1 and 25 [Hz] for joint 2). This phenomenon might be caused by nonreproducible effects, such as variable friction in the robot joints and variable sampling times of the PC-based control system.

It is well-known that nonreproducible errors can be slightly amplified by ILC within the passband of the Q-filter [30]. Both effects are evident in the experiments on the RRR-robot. Another reason for the amplification phenomenon could be the influence of residual nonlinear behaviors contained in the measured data. The model-based compensation strongly reduces robot nonlinearities but hardly eliminates these. The residual nonlinearities influence the outcome of ILC design, which does not directly take into account nonlinear effects. The design itself assumes linear plant dynamics only. Therefore, an interesting topic for future research could be analysis of the influence of nonlinear behaviors on the considered ILC algorithm. The amplification of the harmonic components might have been avoided by reducing the passband magnitude of the robustness filter. However, this would slow down the convergence rate of learning, i.e., more trials would be needed for the errors to reach their lower bounds. There is no amplification of the error harmonic content between 13 [Hz] and 20 [Hz] in the third joint, as seen in Fig. 14(c).

VIII. CONCLUSION

A procedure to achieve high quality performance in direct-drive robot motion control is presented. It starts with the design of a nonlinear model-based controller that compensates for nonlinear robot dynamics, leaving mostly decoupled and linear dynamics in the joints. The rigid body dynamic model with friction, used for this compensation, is assumed to be known. A procedure for identifying the parameters of this model has been presented. The identification is based on a batch adaptive control algorithm that admits online implementation, which is quite appealing for use in practice.

The robot dynamics that remain after the model-based compensation, is identified using FRF measurements. Using these measurements, parametric models of the remaining dynamics are made. These models are used in constructing the filters of the iterative learning controllers. A procedure for filter tuning is given, taking into account learning capabilities and convergence issues.

The identification procedure, as well as the design of the learning controller, are experimentally demonstrated on a serial spatial direct-drive robotic manipulator. Experiments show a considerable improvement of motion control performance in all robot joints.

REFERENCES


Björn Bukkems received the M.Sc. degree (cum laude) in mechanical engineering from the Technische Universiteit Eindhoven, Eindhoven, The Netherlands, in 2003. Since 2003, he has been a Research Assistant in the Dynamics and Control Systems Technology Group, Department of Mechanical Engineering, Technische Universiteit Eindhoven. His research interests include the multidisciplinary design and analysis of dynamic embedded systems. Special focus is on using hybrid formalisms for modeling industrial motion systems, and on supervisory controller synthesis for this kind of systems.

Dragan Kostić (S’98–M’05) received the Graduate Engineer and M.Sc. degrees in automatic control and robotics from the Faculty of Electronic Engineering, University of Niš, Serbia, in 1994 and 2000, respectively, and the Ph.D. degree from the Dynamics and Control Technology Group, Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, in 2004. He is a Research Fellow at the Embedded Systems Institute, Eindhoven, The Netherlands. From 1996 to 1999, he was a Research and Teaching Assistant with the Group for Automatic Control, Faculty of Electronic Engineering, University of Niš, where he was investigating problems of biomechanical analogies in robotics. His current research interests include dynamic modeling, system identification, and robust model-based control of robots, as well as data-based control of motion systems.

Bram de Jager received the M.Sc. degree in mechanical engineering from Delft University of Technology, Delft, The Netherlands, and the Ph.D. degree from the Technische Universiteit Eindhoven, Eindhoven, The Netherlands. He was with Delft University of Technology and Stork Boilers BV, Hengelo, The Netherlands. Currently, he is with the Technische Universiteit Eindhoven. His research interests include robust control of (nonlinear) mechanical systems, integrated control and structural design, control of fluidic systems, control structure design, and application of symbolic computation in nonlinear control.

Maarten Steinbuch (S’83–M’89–SM’02) received the M.Sc. degree (cum laude) in mechanical engineering and the Ph.D. degree in modeling and control of wind energy conversion systems from Delft University of Technology, Delft, The Netherlands, in 1984 and 1989, respectively. He is currently a Full Professor in the Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands. From 1984 to 1987, he was a Research Assistant at Delft University of Technology and KEMA (Power Industry Research Institute), Arnhem, The Netherlands. From 1987 to 1998, he was with Philips Research Laboratories, Eindhoven, as a Member of the Scientific Staff, working on modeling and control of mechatronic applications. From 1998 to 1999, he was a Manager of the Dynamics and Control Group, Philips Center for Manufacturing Technology. His research interests are modeling and control of motion systems. He was an Associate Editor of IFAC Control Engineering Practice from 1994 to 1996 and is currently Editor-at-Large of the European Journal of Control. Dr. Steinbuch was an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY from 1993 to 1997 and of the IEEE Control Systems Magazine from 1999 to 2002. In 2003, he received the Best-Teacher 2002/2003 award from the Department of Mechanical Engineering, Technische Universiteit Eindhoven.