Proceedings of the 2007 American Control Conference
Marriott Marquis Hotel at Times Square
New York City, USA, July 11-13, 2007

Fixed Structure Feedforward Controller Tuning Exploiting Iterative Trials, Applied to a High-Precision Electromechanical Servo System

Stan van der Meulen, Rob Tousain, and Okko Bosgra

Abstract— **In this paper, the feedforward controller design problem for high-precision electromechanical servo systems that execute finite time tasks is addressed. The presented procedure combines the selection of the fixed structure of the feedforward controller and the optimization of the controller parameters on the basis of measurement data from iterative trials. A linear parameterization of the feedforward controller in a two-degree-of-freedom control architecture is chosen, which for a linear time-invariant (LTI) plant results in a feedforward controller that is applicable to a class of motion profiles as well as in a convex optimization problem with the objective function being a quadratic function of the tracking error. Optimization by iterative trials results in the controller parameter values that are optimal with respect to the actual plant, which leads to a high tracking performance. The use of iterative trials in general outperforms techniques that are based on a detailed *a priori* plant model only, whereas the fixed structure of the feedforward controller, *i.e.*, the approximative inverse plant model, guarantees a high tracking performance for a class of motion profiles, unlike for example iterative learning control (ILC). Experimental results on a high-precision wafer stage illustrate the procedure.**

I. INTRODUCTION

The general trend in the field of industrial high-precision electromechanical servo systems is that the performance requirements are ever increasing. Examples of such systems are pick-and-place robots, laser welding robots, and motion stages. The performance requirements for these systems typically relate to the throughput and the quality of the products, which translates to aggressive motion profiles and high tracking accuracies, respectively. Typical tasks that are executed by such systems are given by finite time tasks, more specifically point-to-point motions. Often, a series of point-to-point motions is executed, in which the motion profile is not necessarily the same for each task.

To track an aggressive motion profile with high accuracy, the machines are typically equipped with a feedback controller and a feedforward controller. The design of the feedforward controller is crucial to achieve the performance requirements in high-precision electromechanical servo systems, since a transient error is inherently present in case only a feedback controller is implemented. Several approaches to feedforward control are discussed next.

A straightforward approach is given by model-based feedforward control. The inverse plant model is used as a feedforward controller in a two-degree-of-freedom control architecture. Various examples can be found in [1], [2], [3], [4], and [5]. In this approach, the model is only an approximation of the plant, which hampers or possibly prevents the achievement of the performance requirements. To improve the quality of the feedforward controller, it is possible to adapt the controller parameters either directly or indirectly, where use is made of measurement data. Various examples of adaptive feedforward control can be found in [6], [7], and [8]. However, adaptive feedforward control is less suited for the application to finite time tasks, due to the adaptation at each sample instant. In addition, it is generally required that the persistent excitation condition is satisfied, which imposes undesired requirements on the motion profile.

The concept of iterative learning control (ILC) applies to systems that execute the same motion profile over and over again. Essentially, this technique determines the feedforward signal that forces the output to track this motion profile by iterative trials. With learning by trials, the need for detailed knowledge of the system is avoided, since use is made of measurement data. Excellent overviews can be found in [9], [10], [11], and [12], for instance. However, the strength of ILC, *i.e.*, the possibility to eliminate all deterministic components in the tracking error that are constant in the trial domain, is at the same time its weakness, *i.e.*, the motion profile is necessarily constant in the trial domain.

Actually, ILC is a specific direct tuning method. In a direct tuning method, a controller parameter optimization problem is formulated and the basic idea is to use numerical optimization and to use measurement data from iterative trials to optimize the controller parameters without intermediate identification steps. In ILC, the controller parameters are represented by the individual samples of the feedforward signal and the tracking error from the previous trial is used to update the feedforward signal. One possibility is to update the feedforward signal according to Newton's method [13], where the objective function is a quadratic function of the tracking error.

A direct tuning method with more freedom is given by iterative feedback tuning (IFT) [14]. This approach optimizes the controller parameters that appear in arbitrary one-degree-of-freedom or two-degree-of-freedom control architectures according to Newton's method, where the objective function typically is a quadratic function of the tracking error and the control effort. The key feature of this approach is that it only uses measurement data from iterative trials, *i.e.*, no

S. van der Meulen and O. Bosgra are with the Department of Mechanical Engineering, Control Systems Technology Group, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands S.H.v.d.Meulen@tue.nl, O.H.Bosgra@tue.nl

R. Tousain is with the Department of Mechatronics, Drives and Control Group, Philips Applied Technologies, High Tech Campus 7, 5656 AE Eindhoven, The Netherlands Rob.Tousain@philips.com

1-4244-0989-6/07/$25.00 ©2007 IEEE.

model knowledge is required. However, this method does not provide any directions how to select the structure of the feedforward controller.

In this paper, the gap between the selection of the structure of the feedforward controller and the optimization of the corresponding controller parameters by iterative trials is bridged, see also [15]. The contribution of this approach is that it results in a feedforward controller that 1) is applicable to a class of motion profiles, in contrast to ILC, 2) has low complexity to facilitate industrial implementation, in contrast to model-based feedforward control and IFT in general, and 3) incorporates the controller parameter values that are optimal with respect to the actual plant, which is generally not achieved by, *e.g.*, model-based feedforward control and manual tuning. This enables the achievement of the severe performance requirements in high-precision electromechanical servo systems.

The remainder of this paper is organized as follows. The feedforward controller design procedure, which consists of the design of the feedforward controller itself in combination with the design of the direct tuning method, is considered in Section II. In Section III, this design procedure is applied to a high-precision wafer stage, where experimental results are shown. Finally, conclusions are drawn in Section IV.

II. Feedforward Controller Design Procedure

The goal of the feedforward controller is to attenuate the tracking error that appears during the execution of a finite time task by the realization and the application of a feedforward signal. To obtain a feedforward controller that is applicable to a class of motion profiles, a two-degree-of-freedom control architecture is considered, see Figure 1. Here, P denotes the plant, which is considered to be discrete time, single input single output (SISO), and linear time-invariant (LTI). Furthermore, K_{fb} represents the feedback controller and K_{ff} represents the feedforward controller. The position setpoint is denoted by r, the tracking error by e, the feedback signal by u_{fb}, the feedforward signal by u_{ff}, the plant input by u, the disturbances by w, and the plant output by y. The feedback controller K_{fb} is designed to stabilize the plant P and to suppress the disturbances w.

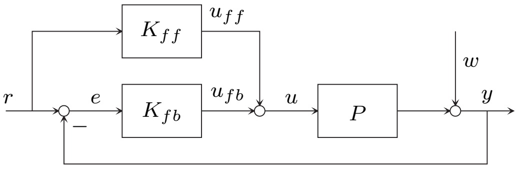

Fig. 1. System with a two-degree-of-freedom control architecture.

The transfer function between e and r is given by:

$$\frac{e}{r} = \frac{1 - P(z)K_{ff}(z)}{1 + P(z)K_{fb}(z)}, \tag{1}$$

where w is omitted for convenience. Obviously, the transfer function (1) is zero if the feedforward controller is equal to the inverse of the plant. Essentially, the feedforward controller design procedure consists of the design of the feedforward controller itself in combination with the design of the direct tuning method that is used to optimize the controller parameters in the feedforward controller by iterative trials. The design of the feedforward controller itself involves the steps “Fixed Structure Feedforward Controller Parameterization” and “Initial Controller Parameters”. The design of the direct tuning method involves the steps “Objective Function” and “Optimization Algorithm”.

Once this design procedure is completed, the operation of the direct tuning method is as follows. First, a finite time task is executed by the system. This is called a trial and the trial number is denoted by l. Then, a signal-based objective function is evaluated. If the objective function value is satisfactory, the system is allowed to operate again. Otherwise, the optimization algorithm utilizes measurement data to adjust the controller parameters, after which the system is allowed to operate again.

A. Fixed Structure Feedforward Controller Parameterization

The parameterization of the feedforward controller results in a fixed structure of the feedforward controller, which incorporates one or more controller parameters. Attention is restricted to a so-called linear parameterization of the feedforward controller. This restriction is motivated in Section II-C. A familiar linear parameterization of the feedforward controller for high-precision electromechanical servo systems is found in [16], [17], and [18]. There, the feedforward signal is given by:

$$u_{ff} = kfs \cdot s + kfj \cdot j + kfa \cdot a + kfv \cdot v, \tag{2}$$

which is a linear function of the constant, real controller parameters kfs, kfj, kfa, and kfv. In (2), s, j, a, and v denote the snap setpoint, the jerk setpoint, the acceleration setpoint, and the velocity setpoint, respectively. Actually, s, j, a, and v correspond to the fourth-, third-, second-, and first-order derivative of the position setpoint. This linear parameterization of the feedforward controller provides a good description of the low-frequency behaviour of the inverse of the plant for high-precision electromechanical servo systems [17]. The system that is obtained with the application of snap feedforward, jerk feedforward, acceleration feedforward, and velocity feedforward is depicted in Figure 2.

Due to the utilization of iterative trials, it is convenient to package the information in each trial together. All signals in Figure 2 are discrete time signals, which suggests the application of the lifted signal description [19], [20]. In the lifted signal description, a discrete time signal $x(k)$ in trial l is defined by:

$$x_l = \left[\begin{array}{ccc} x(0) & \cdots & x(N-1) \end{array} \right]^T, \tag{3}$$

for $k = 0, \ldots, N-1$. Here, k denotes the sample instant and N denotes the number of samples in trial l. Furthermore, the following abbreviation is introduced to facilitate the notation:

$$\theta_l = \left[\begin{array}{cccc} kfs_l & kfj_l & kfa_l & kfv_l \end{array} \right]^T. \tag{4}$$

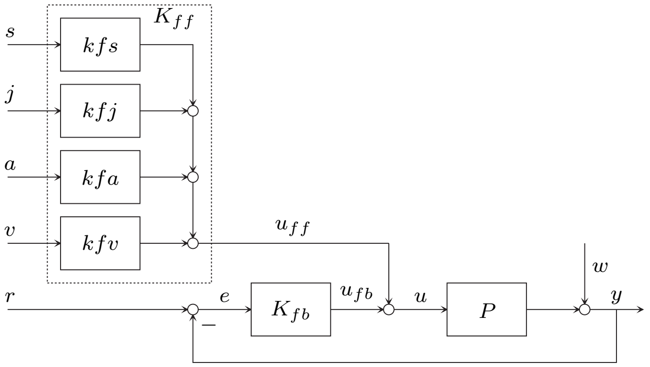

Fig. 2. System with snap feedforward, jerk feedforward, acceleration feedforward, and velocity feedforward.

B. Initial Controller Parameters

The initial controller parameter values are typically given by zeros, unless explicit knowledge with respect to the optimal controller parameter values is available.

C. Objective Function

The direct tuning method relies on the optimization of a certain objective function. It is required that the objective function is representative of the machine performance. To obtain a high tracking accuracy, the signal-based objective function V is chosen equal to:

$$V(\theta_l) = e_l^T(\theta_l)e_l(\theta_l), \tag{5}$$

which is the square of the 2-norm of the tracking error. The objective function (5) in combination with the structure of the feedforward controller (2) results in a convex optimization problem, which implies that the global optimal solution is achievable, see [21]. The essential observation for this conclusion is that the tracking error is a linear (affine) function of the controller parameters. As a result, the objective function is a quadratic function of the controller parameters, which implies that the Hessian of the objective function is independent of the controller parameters. Under the assumption that the plant is LTI, the Hessian of the objective function is positive definite and the optimization problem is convex.

D. Optimization Algorithm

It is assumed that no constraints are present on the controller parameters, *i.e.*, the optimization problem is unconstrained, although this is not essential. A well-known optimization algorithm in the context of unconstrained optimization is given by Newton's method. The definition of Newton's method is given by:

$$\theta_{l+1} = \theta_l - \alpha_l \left(\nabla^2 V(\theta_l)\right)^{-1} \nabla V(\theta_l), \tag{6}$$

see [21]. Here, α is the step length, ∇V is the gradient of the objective function, and $\nabla^2 V$ is the Hessian of the objective function. Many ways exist to compute α and to approximate ∇V and $\nabla^2 V$. Approximations are inevitable, since the actual system is unknown. Apart from the differences, the quantities that result are used in the same algorithm.

Algorithm 1 (Direct tuning method):

1) Set the trial number l equal to $l = 0$.
2) Set the initial controller parameter values θ_0.
3) Execute a finite time task r_l and measure the tracking error e_l.
4) Evaluate the objective function (5). Proceed with Step 5 if the objective function value is not satisfactory. Otherwise, proceed with Step 6.
5) Execute the optimization algorithm (6).
6) Set the trial number l equal to $l = l + 1$. Proceed with Step 3.

A constant value for α is employed, because the application of line search optimization results in a more complex algorithm, see [21]. Two approaches to approximate ∇V and $\nabla^2 V$ are considered. The gradient of the objective function (5) with respect to the controller parameters is given by:

$$\nabla V(\theta_l) = 2\nabla e_l^T(\theta_l)e_l(\theta_l), \tag{7}$$

whereas the Hessian of the objective function (5) with respect to the controller parameters is given by:

$$\nabla^2 V(\theta_l) = 2\nabla e_l^T(\theta_l)\nabla e_l(\theta_l). \tag{8}$$

The approaches to approximate (7) and (8) are named the model-based approach and the data-based approach. In both approaches, $e_l(\theta_l)$ is obtained from measurement data, which requires the execution of one finite time task. In the model-based approach, $\nabla e_l(\theta_l)$ is obtained from model knowledge (cheap, less accurate), see Section II-D.1. In the data-based approach, $\nabla e_l(\theta_l)$ is obtained from measurement data (expensive, more accurate), which requires the execution of another finite time task, see Section II-D.2. Furthermore, the optimization of the so-called delay correction is easily incorporated in both approaches, which is discussed in Section II-D.3 for the model-based approach.

1) Model-based Approach: Consider the discrete time system in Figure 2, to which the lifted system description is applied [19], [20]. In the lifted system description, a state-space representation of a certain transfer function is considered:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \qquad (9)\\ y(k) &= Cx(k) + Du(k), \qquad (10)\end{aligned}$$

where $u(k)$ denotes the input and $y(k)$ denotes the output, which are both arbitrary. Then, the lifted system description is defined by:

$$y_l = \underbrace{\begin{bmatrix} D & 0 & \cdots & \cdots & 0 \\ CB & D & \ddots & & \vdots \\ \vdots & CB & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ CA^{N-2}B & CA^{N-3}B & \cdots & CB & D \end{bmatrix}}_{\mathcal{T}} u_l, \tag{11}$$

for $k = 0, \ldots, N-1$. Here, the $N \times N$ Toeplitz matrix $\mathcal{T}$ contains N impulse response coefficients, *i.e.*, Markov parameters. In Figure 2, the map between r_l and e_l is defined by the sensitivity Toeplitz matrix $\mathcal{S}$, whereas the map between u_{ff_l} and y_l is defined by the process sensitivity Toeplitz matrix $\mathcal{PS}$. In addition, the following abbreviation is introduced to facilitate the notation:

$$\xi_l = \left[\begin{array}{cccc} s_l & j_l & a_l & v_l \end{array} \right]. \tag{12}$$

With these definitions, the tracking error in the trial domain is approximated by:

$$e_l(\theta_l) = \mathcal{S}(r_l - w_l) - \mathcal{PS}\xi_l\theta_l, \tag{13}$$

where $\mathcal{S}$ and $\mathcal{PS}$ are based on model knowledge. Using (13), it is possible to derive the gradient of the error signal with respect to the controller parameters:

$$\nabla e_l(\theta_l) = -\mathcal{PS}\xi_l. \tag{14}$$

Subsequently, (14) is substituted into (7) and (8). Hence, the gradient of the objective function is approximated by using both model knowledge (∇e_l) and measurement data (e_l), whereas the Hessian of the objective function is approximated by using model knowledge (∇e_l) only. This implies that the model-based approach requires the execution of one finite time task per trial.

The model-based approach for the approximation of the gradient of the objective function and the Hessian of the objective function has similarities with the utilization of basis functions in lifted ILC, see [22]. Here, the basis functions are given by the snap setpoint, the jerk setpoint, the acceleration setpoint, and the velocity setpoint.

Convergence to a (local) minimum of the objective function is guaranteed if a certain stability condition is satisfied. This stability condition applies to the linear discrete time system that is obtained after substitution of (7) and (8) into (6), using (14). It is required that this linear discrete time system is stable, which implies that $0 < \alpha_l < 2$, see [23].

2) Data-based Approach: Consider the discrete time system in Figure 1, where the discrete time feedback controller $K_{fb}(z)$ is fixed. The discrete time feedforward controller $K_{ff}(z, \theta_l)$ is defined by:

$$K_{ff}(z, \theta_l) = kfs_l\beta_s + kfj_l\beta_j + kfa_l\beta_a + kfv_l\beta_v, \tag{15}$$

where:

$$\beta_s = \frac{z^4 - 4z^3 + 6z^2 - 4z + 1}{T_s^4 z^4} \tag{16}$$

$$\beta_j = \frac{z^4 - 3z^3 + 3z^2 - z}{T_s^3 z^4} \tag{17}$$

$$\beta_a = \frac{z^4 - 2z^3 + z^2}{T_s^2 z^4} \tag{18}$$

$$\beta_v = \frac{z^4 - z^3}{T_s z^4}, \tag{19}$$

where T_s denotes the sample time.

The transfer function (1) is rephrased as follows:

$$e_l(\theta_l) = \frac{1 - P(z)K_{ff}(z, \theta_l)}{1 + P(z)K_{fb}(z)} r_l, \tag{20}$$

where w_l is omitted for convenience. In the situation under consideration, the following expressions hold for the derivatives of $K_{fb}(z)$ and $K_{ff}(z, \theta_l)$ with respect to the controller parameters:

$$\frac{\partial K_{fb}(z)}{\partial \theta} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \end{array} \right] \tag{21}$$

$$\frac{\partial K_{ff}(z, \theta_l)}{\partial \theta} = \left[\begin{array}{cccc} \beta_s & \beta_j & \beta_a & \beta_v \end{array} \right]. \tag{22}$$

Using these derivatives, it is possible to derive the gradient of the error signal with respect to the controller parameters:

$$\nabla e_l(\theta_l) = -\frac{\partial K_{ff}(z, \theta_l)}{\partial \theta} \frac{1}{K_{fb}(z) + K_{ff}(z, \theta_l)} \frac{P(z)(K_{fb}(z) + K_{ff}(z, \theta_l))}{1 + P(z)K_{fb}(z)} r_l. \tag{23}$$

Next, it is observed from Figure 1 that the following expression holds for the plant output:

$$y_l(\theta_l) = \frac{P(z)(K_{fb}(z) + K_{ff}(z, \theta_l))}{1 + P(z)K_{fb}(z)} r_l, \tag{24}$$

where w_l is omitted again. Substitution of (24) into (23) leads to the following expression:

$$\nabla e_l(\theta_l) = -\frac{\partial K_{ff}(z, \theta_l)}{\partial \theta} \frac{1}{K_{fb}(z) + K_{ff}(z, \theta_l)} y_l(\theta_l). \tag{25}$$

Since the actual plant output $y_l(\theta_l)$ is contaminated with the disturbance w_l, (25) is only an approximation of the gradient of the error signal with respect to the controller parameters. Subsequently, (25) is substituted into (7) and (8). Hence, the gradient of the objective function is approximated by using measurement data from both the first finite time task (e_l) and the second finite time task (∇e_l), whereas the Hessian of the objective function is approximated by using measurement data from the second finite time task (∇e_l) only. This implies that the data-based approach requires the execution of two finite time tasks per trial. Of course, instead of a renewed approximation of the gradient of the error signal with respect to the controller parameters during each trial, it is also possible to perform this approximation only once.

The data-based approach for the approximation of the gradient of the objective function and the Hessian of the objective function has similarities with the approach that is used in IFT, see [24] and [14]. Here, the so-called dedicated experiment is not required, since the feedback controller $K_{fb}(z)$ is fixed.

Convergence to a (local) minimum of the objective function is guaranteed if certain conditions are satisfied. These conditions apply among other things to the disturbance w_l, *e.g.*, w_l has zero mean and sequences of w_l are mutually independent. Then, (7) is an unbiased approximation of the gradient of the objective function, which is essential for the convergence proof. All conditions and the formal convergence proof can be found in [14].

3) Delay Correction Optimization: The purpose of the delay correction τ is to advance the feedforward signal in such a way that the delay in the system between the feedforward signal and the position setpoint is compensated for. Several sources contribute to this delay, *e.g.*, the actuator system, the sensor system, and the hold circuit. An initial estimate of the delay correction τ_0 is obtained from the phase plot of the Bode diagram of the frequency response function (FRF) measurements. Using τ_0, it is possible to optimize the delay correction τ for each of the setpoints that comprise the feedforward signal. This is illustrated for the acceleration setpoint.

Consider the illustration in Figure 3. Two acceleration setpoints a_l^m and a_l^n are considered, which are identical to the acceleration setpoint a_l. However, a_l^m leads r_l in time by m samples, whereas a_l^n leads r_l in time by $n = m+1$ samples. Here, m is chosen in such a way that τ_0 is in between m samples and n samples. Next, the controller parameters kfa_l^m and kfa_l^n are introduced, which correspond to a_l^m and a_l^n, respectively. Then, the actual controller parameter kfa_l and the actual delay correction τ_l are given by:

$$kfa_l = kfa_l^m + kfa_l^n \tag{26}$$

$$\tau_l = \left(m + \frac{kfa_l^n}{kfa_l^m + kfa_l^n}\right) T_s. \tag{27}$$

Here, it is assumed that the timing of a_l and r_l is identical. Hence, the definition of the acceleration setpoints a_l^m and a_l^n with the controller parameters kfa_l^m and kfa_l^n allows for the simultaneous optimization of the controller parameter kfa_l and the delay correction τ_l. Actually, it is possible to generate any delay correction value in between m samples and n samples. Generalization of the algorithm to incorporate the optimization of the delay correction is obvious.

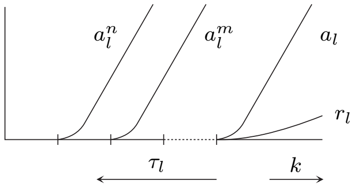

Fig. 3. Optimization of delay correction for acceleration setpoint.

III. Application to a Wafer Stage

The feedforward controller design procedure from Section II is applied to a high-precision wafer stage that is part of a wafer scanner. A wafer scanner is used in the mass production process of integrated circuits (ICs), see [25], where it is responsible for the photolithographic process in which the IC pattern is printed onto a silicon disk, *i.e.*, a wafer. In a wafer scanner, the wafer stage is the high-precision electromechanical servo system that positions the wafer with respect to the imaging optics. As a result, the wafer stage determines the throughput and the quality of the products to a large extent and it is subject to severe performance requirements. Typical velocities and accelerations are 0.5 m/s and 10 m/s^2, respectively, whereas the tracking accuracy is in terms of nanometers and microradians, which demands for a sound feedforward controller design.

The wafer stage is actuated and controlled in six degrees of freedom: three translations (x, y, and z) and three rotations (R_x, R_y, and R_z, where the subscripts refer to the rotation axis). Use is made of six Lorentz actuators and a laser interferometer measurement system. Here, the wafer stage dynamics in the y-direction is considered, which is the main scan direction. The frequency response function (FRF) measurements and the corresponding second-order discrete time transfer function model are depicted in Figure 4. The feedback controller is of the proportional integral derivative (PID) type with high-frequency roll-off, which is designed in the continuous time domain, see [26]. Afterwards, it is discretized on the basis of a first-order hold discretization scheme. A typical finite time task that is executed in the y-direction is given by a point-to-point motion, see Figure 5, which is generated by a third-order setpoint generator, see [16].

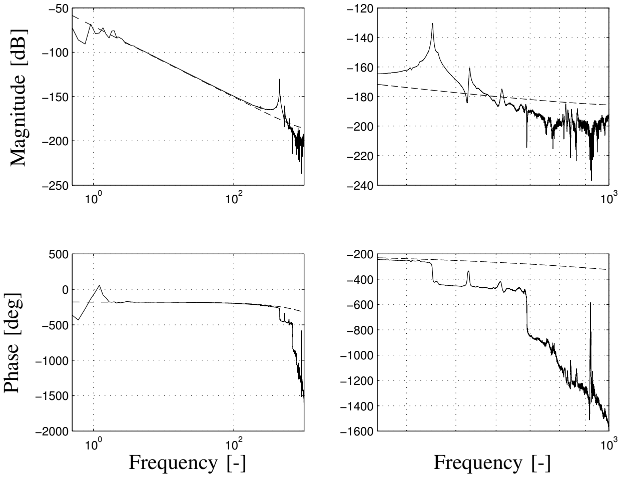

Fig. 4. Bode diagram of the wafer stage dynamics in the y-direction, where the figures on the right are close-ups of the figures on the left (solid: frequency response function measurements; dashed: second-order discrete time transfer function model).

A. Fixed Structure Feedforward Controller Parameterization

For the wafer stage, the structure of the feedforward controller consists of acceleration feedforward and snap feedforward. The motivation for this specific structure of the feedforward controller is found in [17]. There, it is shown that acceleration feedforward exactly compensates for the rigid body mode, whereas snap feedforward exactly compensates for the low-frequency contributions of all residual plant modes. As a result, acceleration feedforward and snap feedforward allow for the exact description of the low-frequency behaviour of the inverse of the plant. This implies that acceleration feedforward and snap feedforward are particularly effective in case of position setpoints that contain mostly low-frequency energy, such as the position setpoint from Figure 5. As a result, the abbreviations reduce

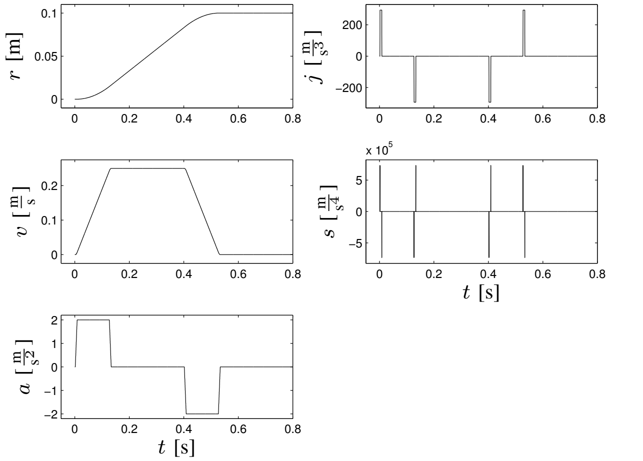

Fig. 5. The position r, the velocity v, the acceleration a, the jerk j, and the snap s of the point-to-point motion.

to:

$$\xi_l = \left[\begin{array}{ccc} s_l & a_l^m & a_l^n \end{array} \right] \tag{28}$$

$$\theta_l = \left[\begin{array}{ccc} kfs_l & kfa_l^m & kfa_l^n \end{array} \right]^T. \tag{29}$$

B. Initial Controller Parameters

An initial estimate of the controller parameter kfa_0 is obtained from the fit of a second-order differentiator through the inverse of the original frequency response function measurements. An initial estimate of the controller parameter kfs_0 is obtained from the fit of a fourth-order differentiator through the frequency response function measurements that result after subtraction of the aforementioned fit from the inverse of the original frequency response function measurements. See also [17].

C. Experimental Results

Utilization of Algorithm 1 in combination with either the model-based approach or the data-based approach allows for the optimization of the controller parameters. In addition, the delay correction τ is optimized for the acceleration setpoint. Only the results for the model-based approach are considered, since similar results are obtained for the data-based approach.

Each finite time task is executed at the same position in the operating area of the wafer stage, *i.e.*, the horizontal xy-plane. More specifically, the finite time task from Figure 5 is executed in the y-direction, across the centre position, where $y(0) = -0.05$ m and $y(N-1) = 0.05$ m. In the model-based approach, the process sensitivity Toeplitz matrix $\mathcal{PS}$ is based on the second-order discrete time transfer function model and the discretized feedback controller. The step length is equal to $\alpha_l = 0.8$ and the number of trials is equal to six. The tracking errors are depicted in Figure 6, the controller parameters are depicted in Figure 7, and the objective function is depicted in Figure 8.

From Figures 6 and 8, it is concluded that the tracking error decreases and the machine performance increases as a function of the trial number. From Figure 7, it is concluded that convergence is achieved for all controller parameters. It is observed that the controller parameter kfa and the delay correction τ converge monotonically, where the exponential behaviour is due to the choice of the step length. Obviously, the controller parameter kfs does not converge monotonically. This is due to the fact that the process sensitivity Toeplitz matrix $\mathcal{PS}$ is based on the second-order discrete time transfer function model from Figure 4, where the resonant dynamics is not taken into account. When the resonant dynamics is taken into account, the controller parameter kfs converges monotonically.

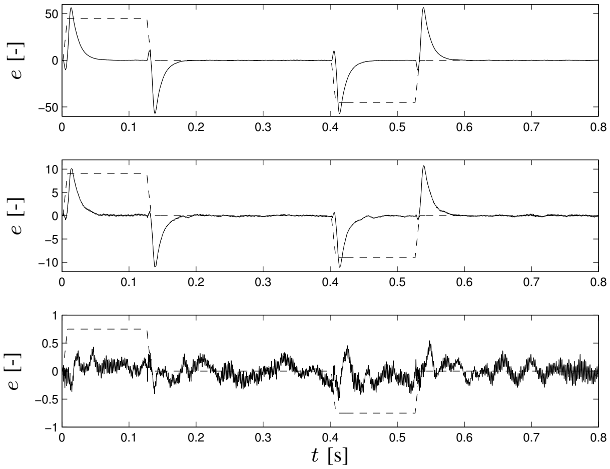

Fig. 6. Experimental tracking errors obtained with model-based approach in trial 0 (top), 1 (middle), and 5 (bottom) (solid: tracking error; dashed: scaled acceleration setpoint). Notice the scales!

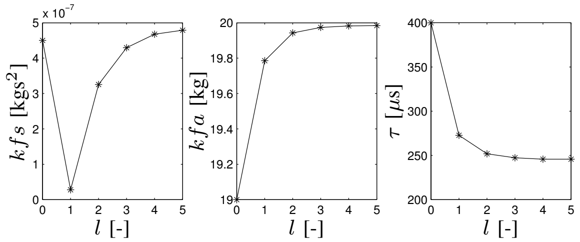

Fig. 7. Experimental controller parameters obtained with model-based approach.

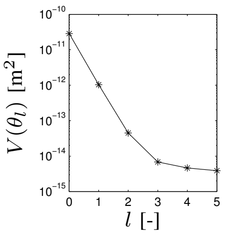

Fig. 8. Experimental objective function obtained with model-based approach.

IV. Conclusions

A feedforward controller design procedure for high-precision electromechanical servo systems that execute finite time tasks is presented. The procedure consists of the selection of the fixed structure of the feedforward controller in a two-degree-of-freedom control architecture in combination with the optimization of the controller parameters by iterative trials. It is successfully applied to a high-precision wafer stage, where high tracking accuracies are achieved.

This paper has shown that the use of iterative trials in general outperforms techniques that are based on a detailed *a priori* plant model only, *i.e.*, model-based feedforward control, whereas the fixed structure of the feedforward controller guarantees a high tracking performance for a class of motion profiles, unlike for example ILC. It is shown that the chosen linear parameterization of the feedforward controller is applicable to the common class of fourth-order position setpoints and has low complexity, which facilitates industrial implementation. Furthermore, a convex controller parameter optimization problem is obtained, since the objective function is a quadratic function of the tracking error. It is shown that optimization of this objective function by iterative trials provides a high accuracy, which is generally not obtained on the basis of a detailed *a priori* plant model only. Convergence of the algorithm is guaranteed, which is experimentally demonstrated. The experiments show a high tracking performance as well.

The present paper has motivated the choice of a fixed structure of the feedforward controller under the assumption that it enables the application to a class of motion profiles. Further research is required to validate this assumption and to characterize this class of motion profiles. In this context, the relation between the choice of the fixed structure of the feedforward controller and the class of motion profiles deserves further attention.

References

[1] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *J. Dyn. Syst. Meas. Contr.*, vol. 109, no. 1, pp. 65–68, Mar. 1987.

[2] L. R. Hunt, G. Meyer, and R. Su, "Noncausal inverses for linear systems," *IEEE Trans. Automat. Contr.*, vol. 41, no. 4, pp. 608–611, Apr. 1996.

[3] S. Devasia, "Should model-based inverse inputs be used as feedforward under plant uncertainty?" *IEEE Trans. Automat. Contr.*, vol. 47, no. 11, pp. 1865–1871, Nov. 2002.

[4] D. E. Torfs, R. Vuerinckx, J. Swevers, and J. Schoukens, "Comparison of two feedforward design methods aiming at accurate trajectory tracking of the end point of a flexible robot arm," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, no. 1, pp. 2–14, Jan. 1998.

[5] M. Boerlage, M. Steinbuch, P. Lambrechts, and M. van de Wal, "Model-based feedforward for motion systems," in *Proc. 2003 IEEE Conf. Contr. Appl.*, vol. 2, Istanbul, Turkey, June 2003, pp. 1158–1163.

[6] T. Hägglund and K. J. Åström, "Industrial adaptive controllers based on frequency response techniques," *Automatica*, vol. 27, no. 4, pp. 599–609, July 1991.

[7] T.-C. Tsao and M. Tomizuka, "Robust adaptive and repetitive digital tracking control and application to a hydraulic servo for noncircular machining," *J. Dyn. Syst. Meas. Contr.*, vol. 116, no. 1, pp. 24–32, Mar. 1994.

[8] S. Zhao and K. K. Tan, "Adaptive feedforward compensation of force ripples in linear motors," *Contr. Eng. Pract.*, vol. 13, no. 9, pp. 1081–1092, Sept. 2005.

[9] K. L. Moore, "Iterative learning control – an expository overview," *Appl. Comput. Contr. Signal Processing Circuits*, vol. 1, no. 1, pp. 151–214, 1998.

[10] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Contr.*, vol. 73, no. 10, pp. 930–954, July 2000.

[11] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control – a learning-based method for high-performance tracking control," *IEEE Contr. Syst. Mag.*, vol. 26, no. 3, pp. 96–114, June 2006.

[12] Z. Bien and J.-X. Xu, *Iterative Learning Control – Analysis, Design, Integration, and Applications*. Boston: Kluwer Academic Publishers, 1998.

[13] S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011–2016, Dec. 2001.

[14] H. Hjalmarsson, "Iterative feedback tuning – an overview," *Int. J. Adaptive Contr. Signal Processing*, vol. 16, no. 5, pp. 373–395, June 2002.

[15] S. H. van der Meulen, R. L. Tousain, and O. H. Bosgra, "Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer," Submitted for publication.

[16] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Contr. Eng. Pract.*, vol. 13, no. 2, pp. 145–157, Feb. 2005.

[17] M. Boerlage, R. Tousain, and M. Steinbuch, "Jerk derivative feedforward control for motion systems," in *Proc. 2004 Amer. Contr. Conf.*, vol. 5, Boston, Massachusetts, June/July 2004, pp. 4843–4848.

[18] B.-H. Chang and Y. Hori, "Trajectory design considering derivative of jerk for head-positioning of disk drive system with mechanical vibration," *IEEE/ASME Trans. Mechatron.*, vol. 11, no. 3, pp. 273–279, June 2006.

[19] M. Phan and R. W. Longman, "A mathematical theory of learning control for linear discrete multivariable systems," in *Proc. AIAA/AAS Astrodyn. Conf.*, Minneapolis, Minnesota, Aug. 1988, pp. 740–746.

[20] R. Tousain, E. van der Meché, and O. Bosgra, "Design strategy for iterative learning control based on optimal control," in *Proc. 40th IEEE Conf. Dec. Contr.*, vol. 5, Orlando, Florida, Dec. 2001, pp. 4463–4468.

[21] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. London: McGraw-Hill, 1996.

[22] J. A. Frueh and M. Q. Phan, "Linear quadratic optimal learning control (LQL)," *Int. J. Contr.*, vol. 73, no. 10, pp. 832–839, July 2000.

[23] T. Kailath, *Linear Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1980.

[24] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Contr. Syst. Mag.*, vol. 18, no. 4, pp. 26–41, Aug. 1998.

[25] G. Stix, "Trends in semiconductor manufacturing: Toward "point one"," *Scientific American*, vol. 272, no. 2, pp. 72–77, Feb. 1995.

[26] M. van de Wal, G. van Baars, F. Sperling, and O. Bosgra, "Multivariable $\mathcal{H}_\infty/\mu$ feedback control design for high-precision wafer stage motion," *Contr. Eng. Pract.*, vol. 10, no. 7, pp. 739–755, July 2002.