Comparison of Two Feedforward Design Methods Aiming at Accurate Trajectory Tracking of the End Point of a Flexible Robot Arm
Dirk E. Torfs, Rudi Vuerinckx, Jan Swevers, and Johan Schoukens

Abstract—This paper discusses the design and properties of two trajectory tracking controllers for linear time-invariant systems, and compares their implementation and experimental results on a flexible one-link robot equipped with a velocity-controlled actuator. High positioning accuracy and low tracking errors within a specified bandwidth are their performance specifications. Both controllers use the same state feedback controller, but have a different feedforward design approach. Both feedforward methods design stable prefilters which approximate the unstable inverse system model. The first method designs a stable prefilter using the extended bandwidth zero phase error tracking control (EBZPETC) method [11], which is an extension of Tomizuka’s well-known zero phase error tracking control (ZPETC) method [10]. The second feedforward method adds delay to the inverse model and then uses common filter design techniques to approximate this delayed frequency response.

Index Terms—Control systems, identification, IIR digital filters, motion control, tracking.

I. INTRODUCTION

In robotics as well as in precision machining, high positioning accuracy and low tracking errors (position and velocity contouring errors) are mechanical and performance specifications and the driving elements for any servo design. The complex shapes involved in modern product design and the ever increasing pressure for higher productivity call for a drastic improvement of the dynamic behavior of the motion control systems used in production machinery. The complex workpiece surfaces encountered in present-day products and generated by computer-aided-design systems are to be transformed into tool paths for machine tools or into end effector trajectories for industrial robots. The more discontinuous these tool paths are and the higher the speed requirements, the higher the acceleration requirements are on the motion control systems and the more difficult the required tracking accuracies are to be met.

Mechatronic systems using traditional controllers often exhibit oscillatory behavior due to structural flexibility, and unacceptable large tracking errors due to inaccurate feedforward control. This approach completely ignores the dynamic effects resulting from fast varying reference commands and is thus not realistic for high-speed contouring.

Oscillation-free positioning and accurate tracking require an advanced controller, whose design is based on an accurate high-order model of the system which includes the main flexible modes. This paper discusses two discrete-time state feedback controllers with feedforward for linear time-invariant systems, and discusses their implementation on a flexible robot link. Both controllers use the same state feedback controller. Their difference lies in the feedforward design. The system model has been identified experimentally using frequency domain system identification techniques and broadband multisine excitation. It is a linear time-invariant discrete-time nonminimum phase model. The state feedback controller is designed using pole placement. The state-space model is chosen such that all state variables are measurable signals and hence no state observer is necessary. This simplifies the implementation of the controller significantly, which is important if implementation on industrial controllers has to be considered.

Both feedforward methods involve the use of a prefilter which corresponds to the inverse system model. This causes problems in the case of nonminimal phase systems, such as the considered flexible robot link, because nonminimal phase zeros become unstable poles in the inverse model. The first feedforward method is the extended bandwidth zero phase error tracking control (EBZPETC) method [11]. This method is based on the zero phase error tracking control (ZPETC) substitution scheme of Tomizuka [10], but adds additional feedforward terms to compensate for its gain error. This method is taken as being sufficiently representative of the class of ZPETC related methods. All ZPETC related methods approximate the inverse system model without introducing phase errors, but they do introduce gain errors which increase with frequency (being equal zero at zero frequency).

The second feedforward method is an iterative process: delays are added to the inverse system model until the resulting delayed frequency response function of the inverse model can be approximated with a stable prefilter. The approximation is...
II. MODELING A FLEXIBLE ROBOT ARM DRIVEN BY A VELOCITY-CONTROLLED ACTUATOR

This section presents input–output models for a flexible robot equipped with a VCA. The models are given for an infinitely stiff velocity loop, i.e., an infinite proportional gain. Previous research has shown that this is a reasonable assumption and that the use of a VCA is superior compared to using a torque controlled actuator [12], [14].

Fig. 1 shows a simple mass-spring-damper system. This system can be used as an aid for the modeling of the flexible robot arm if only one flexible mode is considered [12], [18], although it is not equivalent with the flexible robot arm since it is of minimal phase. This mass-spring-damper system helps to visualize how the real system can be split up in two subsystems and helps to determine the order of the different subsystems and the number of poles and zeros at the origin of the s-plane (i.e., the number of integrators and differentiators in the subsystems).

This mass-spring-damper system can be modeled as a series combination of two transfer functions: The first transfer function relates the angular velocity input \( \dot{\theta}_p \) to the flexible deformation, \( \varepsilon \) (\( \equiv \theta_p - \theta_m \)), \((H_{\dot{\theta} \rightarrow \varepsilon}(s))\). The second transfer function relates the flexible deformation to the integral velocity input, i.e., the position of the motor, \( \theta_m \) \((H_{\varepsilon \rightarrow \theta_m}(s))\).

\[
H_{\dot{\theta} \rightarrow \varepsilon}(s) = \frac{s}{s^2 + \frac{c}{I} s + \frac{k}{I}} \quad (1)
\]

\[
H_{\varepsilon \rightarrow \theta_m}(s) = \frac{s^2 + \frac{c}{I} s + \frac{k}{I}}{s^2} \quad (2)
\]

\(k, c, \) and \(I\) are, respectively, the stiffness, damping, and payload inertia. Transfer function (1) contains a resonance frequency \( \omega_r = \sqrt{k/I} \) and a damping ratio \( \zeta_r = c/2\omega_r I \). The transfer function relating the angular velocity input \( \dot{\theta}_p \) to the angular position of the payload mass \( \theta_p \) follows straightforwardly from (1) and (2).

The choice of the input–output models and the selection of state-space representations is made in view of the design and implementation of a simple state feedback controller which achieves accurate positioning (regulation) and accurate tracking. Good regulation requires separate feedback of collocated and noncollocated sensor signals [5]. Accurate tracking requires high model accuracy, i.e., accurate identification of the system parameters. Simplicity, which is pursued in view of industrial implementation, is achieved through the selection of measurable state variables. Hence, a closed-loop state estimator is not necessary.

Observation spill-over [6] is avoided by filtering the measured encoder and strain gauge signals with a digital second-order Butterworth low-pass filter with a cutoff frequency of 10 Hz. This filter removes the response of unmodeled resonance frequencies from the measured signals such that they can not influence the controlled system. This filter only extends the order of the system with two, due to the use of the same filters for both the strain gauge signals and the encoder signal and the linearity of the system representation.

The modeling of the flexible robot arm is based on the transformation of continuous-time models (1) and (2) to the following general discrete-time model forms:

\[
H_{\dot{\theta} \rightarrow \varepsilon}(z^{-1}) = \frac{b_{11} z^{-1}(1 - z^{-1})}{1 + a_{11} z^{-1} + a_{12} z^{-2}} \quad (3)
\]

\[
H_{\varepsilon \rightarrow \theta_m}(z^{-1}) = \frac{b_{20} + b_{21} z^{-1} + b_{22} z^{-2}}{1 - 2 z^{-1} + z^{-2}} \quad (4)
\]

This transformation yields information only about the number of numerator and denominator parameters for both models, i.e., the number of parameters that have to be estimated. The next step in the modeling procedure is the estimation of the unknown parameters of (3) and (4) based on (time or frequency domain) input–output measurements (see Section V-B). The total discrete-time state-space model of the flexible robot arm combines the identified discrete-time transfer functions (3) and (4) and the second-order digital antispillover filter

\[
H_{d}(z^{-1}) = \frac{b_{30} + b_{31} z^{-1} + b_{32} z^{-2}}{1 + a_{31} z^{-1} + a_{32} z^{-2}} \quad (5)
\]

The input of the state-space model is modified to the differential velocity command \( \Delta v[k] \) in order to improve the disturbance rejection capability [13]

\[
\Delta v[k] = v[k] - v[k-1] = \frac{z-1}{z} v[k].
\]

This reduces transfer function (3) to

\[
H_{\dot{\theta} \rightarrow \varepsilon}(z^{-1}) = \frac{b_{11} z^{-1}}{1 + a_{11} z^{-1} + a_{12} z^{-2}} \quad (6)
\]

Fig. 2 shows the division of the total model into the mentioned submodels. The Fig. 2(c) combines the relations shown in Fig. 2(a) and (b). It is the series combination of transfer functions (5), (6), and (4). The total state-space model is the...
series combination of the state-space representations of these transfer functions, i.e., the series combination of:

1) the state-space representation of the digital filter, $H_{\Delta v \rightarrow e}(z^{-1})$; this submodel relates the differential input velocity signal, $\Delta v[k]$, to its filtered value, i.e., $\Delta v_f[k]$;

2) the state-space representation of $H_{\Delta v \rightarrow e}(z^{-1})$, relating the filtered differential input velocity signal $\Delta v_f[k]$, to the filtered strain gauge signal, i.e., $\varepsilon_f[k]$;

3) the state-space representation of $H_{\varepsilon \rightarrow \theta_m}(z^{-1})$, relating the filtered strain gauge signal $\varepsilon_f[k]$, to the filtered encoder signal, $\theta_{mf}[k]$.

The six state variables of the model are the filter states, and the filtered strain gauge and filtered motor position at the actual and previous time instant, i.e., the state vector equals

$$X(k) = [x_1(k) x_2(k) \varepsilon_f(k) \varepsilon_f(k-1) \theta_f(k) \theta_f(k-1)]^T.$$  \hspace{1cm} (7)

State variables $x_1$ and $x_2$ are the filter states. They can be easily calculated based on an open-loop state estimator. Superscript $T$ denotes the transpose of a vector.

**Remark:** This modeling approach is based on the division of the continuous-time system description in a series combination of two submodels, and on the $z$-transform of these submodels. This approach is only applicable to systems which can be modeled as a series combination of submodels relating input and output system variables, such as flexible robot arms with one (dominant) flexible mode. This procedure introduces errors if the $z$-transforms are proceeded by the model parameter estimation, because the $z$-transform of the cascade of (1) and (2) is different from the cascade of the $z$-transforms of the separate transfer functions since there is no sampling in between. The applied approach is however different since the $z$-transforms are applied to the general forms of models (1) and (2) and not to specific forms in which the parameters are obtained through identification. The errors mentioned above are avoided by estimating the parameters of the discrete-time models (3) and (4), assuming that the general forms of these models is correct, i.e., that the $z$-transforms do not introduce incorrect model set limitations.

**III. FEEDBACK AND FEEDFORWARD CONTROLLER DESIGN**

Fig. 3 shows the control structure for the flexible robot. It consists of feedforward and feedback. Feedback control design for the flexible one-link robot is based on the total state-space model described in the previous section. Feedforward control design is based on identified input–output models of the system, which contain the nonminimum phase behavior. The input of all models is the differential velocity $\Delta v[k]$. Hence, the calculated control signal has to be integrated before it is applied to the real physical system

$$u[k] = u[k-1] + \Delta v[k] = \frac{z}{z-1} \Delta v[k].$$

The control input consists of state feedback and feedforward

$$\Delta v_f[k] = \Delta v_{fb}[k] + \Delta v_{ff}[k].$$

State feedback $\Delta v_{fb}[k]$ deals with regulation behavior and suppresses internal and external disturbances. The feedback signal equals

$$\Delta v_{fb}[k] = K(X_d[k] - X[k]),$$

i.e., the difference between the desired states $X_d[k]$ and the actual states $X[k]$ at each time instant multiplied by the state feedback gain vector $K$. The state feedback controller has been designed using pole placement.

The desired state $X_d[k]$ is the state vector if there is an exact correspondence between the output $y[k]$ and the desired trajectory $y_d[k]$ or

$$X_d[k+1] = AX_d[k] + B \Delta v_{ff}[k]$$  \hspace{1cm} (8)

$$y_d[k] = CX_d[k] + D \Delta v_{ff}[k].$$  \hspace{1cm} (9)

Feedforward $\Delta v_{ff}[k]$ realizes the desired path when the physical system behaves exactly like the model and when there are no disturbances. Hence, calculation of feedforward involves a prefiltering of the desired trajectory with the inverse dynamic model.

Nonminimum phase systems, such as the flexible one-link robot, cause problems with the prefiltering because unstable zeros become unstable poles in the inverse model [10], [11], [3]. This paper describes and compares two different solutions to resolve the unstable prefiltering in detail. It does not intend to review all existing feedforward control design methods.

The first solution aims at substituting the unstable poles of the inverse model without introducing a phase error. However,
a gain error which depends on the location of the unstable zero is introduced.

The second solution involves an iterative procedure in order to find a stable approximation of the inverse model, using common filter design techniques, by adding a delay to the inverse system transfer function.

### A. Extended Bandwidth Zero Phase Error Tracking Control

The ZPETC of Tomizuka [10] divides the numerator $B(z)$ of the system transfer function in a part that contains the acceptable plant zeros (zeros inside the unit disk), $B^u(z)$, and a part that contains the unacceptable plant zeros (zeros outside the unit disk), $B^a(z)$: $B(z) = B^u(z)B^a(z)$. Subsequently, $B^u(z)$ is changed according to the following substitution scheme:

Replace $\frac{1}{B^u(z^{-1})}$ by $B^u(z) \cdot B^a(z)$.  \hspace{1cm} (10)

This substitution introduces no phase error but a gain error which is substantial for zeros close to $z = 1$. A large gain error may result in large tracking errors. Torfs et al. [11] present a substitution scheme which extends the frequency range in which the gain errors are small by adding additional feedforward terms [called extended bandwidth-ZPETC (EBZPETC)].

The EBZPETC algorithm calculates the feedforward signal in two steps. The first step generates a feedforward signal using the ZPETC algorithm. The second step tends to compensate the remaining error by adding additional feedforward signals, which repeatedly reduce the tracking error proportional to $\varepsilon^2(k), \varepsilon^3(k), \varepsilon^4(k), \ldots$, where $\varepsilon$ is the ZPETC tracking error operator

\[ \varepsilon(z) = 1 - \frac{B^u(z^{-1})}{B^u(1)} \cdot \frac{B^u(z)}{B^a(1)}. \]

The EBZPETC algorithm can also be formulated as a one step substitution scheme (see Fig. 4)

Replace $\frac{1}{B^u(z^{-1})}$ by $B^u(z) \cdot \sum_{i=0}^{m-1} (1-G(z))^i$ \hspace{1cm} (11)

in which $(m-1)$ is the number of additional feedforward terms. $G(z)$ is the overall open-loop transfer function including the feedforward controller but without feedback controller. It relates the desired output to the actual output, and is equal to

\[ G(z) = \frac{B^u(z^{-1})B^a(z)}{B^u(1)} \cdot \frac{1}{B^a(1)}. \]

**Remarks:**

1) \[ \lim_{m \to \infty} \left[ \sum_{i=0}^{m-1} (1-G(z))^i \right] = \frac{1}{1 - (1-G(z))} = \frac{1}{G(z)} \] for $|1-G(z)| < 1$, e.g., for $0 < G(z) < 2$, according to the convergence region of the Taylor series. The substitution scheme then becomes

Replace $\frac{1}{B^u(z^{-1})}$ by $B^u(z) \cdot \frac{1}{B^u(1)} \cdot \frac{1}{G(z)}$ \hspace{1cm} (12)

This proves that, at least in the frequency range where $0 < G(z) < 2$, $1/B^u(z^{-1})$ can be approximated with arbitrary accuracy by adding feedforward terms [11], [15].

2) Adding $(m-1)$ feedforward terms results in a theoretical error of $\varepsilon^m(k), \ldots$, but in practice other effects (disturbances and modeling errors) will cause the error to be larger.

3) Tracking of desired output signals containing frequencies for which $G(z) > 2$ deteriorates the performance for both ZPETC and EBZPETC. This can be avoided by eliminating these frequencies from the desired output signal. Or, in the time domain, the desired output signal has to be sufficiently smooth, with bounded higher order differences.

4) Torfs and De Schutter [15] have developed the optimal prefilter design with frequency domain specification (OPDFDS). This prefilter is related to the EBZPETC
method. It introduces weighting factors for the additional feedforward terms. This new prefilter allows to specify the tracking performance in the frequency domain. Hence, any trajectory can be tracked with arbitrary accuracy.

B. Feedforward Controller Design by Adding Delay

Another method for constructing the feedforward controller is to use classical filter design techniques that approximate a given complex-valued frequency response \( T(\omega) \) by an IIR filter of the form

\[
H(z^{-1}) = \sum_{i=0}^{m-1} b_i z^{-i} - \sum_{i=0}^{n-1} a_i z^{-i}
\]

with \( m \) and \( n \) the order and with \( b_i \) and \( a_i \) the real filter coefficients of, respectively, the numerator and denominator. The filter design method used in this paper minimizes the \( L_2 \) norm of the complex error between the target \( T(\omega) \) and the designed frequency response \( H(z^{-1}) \) \[16\]

\[
\min_{a_i,b_i} \left( \sum_{\omega \in \Omega} W(\omega)^2 |T(\omega) - H(e^{-j\omega})|^2 \right). \tag{14}
\]

\( \Omega \) is a dense frequency grid, \( \Omega \subset [0,2\pi] \), that specifies the frequency band in which the controller must operate and \( W(\omega) \) is a strictly positive, real weighting function that allows the designer to give more importance to the error in a specific frequency band, at the expense of the error at other frequencies. In the case of a feedforward controller, this target frequency response \( T(\omega) \) is the inverse of the measured frequency response of the system, such that the overall response of the feedforward controller and the system becomes a unity frequency response. Because of the nonminimum phase nature of the system, the resulting feedforward controller will be unstable if no additional steps are taken. In order to ensure stability, a delay \( \tau \) is introduced. It has been noted that when a delay \( \tau \) is added to the target response \( T(\omega) \), this not only greatly influences the quality of the approximation, it also can make the difference between a stable and an unstable filter. Therefore, (14) is changed into

\[
\min_{a_i,b_i} \left( \sum_{\omega \in \Omega} W(\omega)^2 |e^{-j\omega \tau} T(\omega) - H(e^{-j\omega})|^2 \right) \tag{15}
\]

where the constraints on \( \tau \) are twofold: of all \( \tau \) that result in a stable filter, the one should be used for which the approximation is best. Determining such a delay is an iterative process, which is summarized by the following steps (more details can be found in \[17\]).

1) A filter is designed according to (15) with no delay added to the target \( (\tau = 0) \).
2) All unstable poles of that filter are mirrored inside the unit circle. This operation does not change the amplitude response of the filter, only its phase response.

3) The phase difference between the original and stabilized version of the filter is examined. Since this phase difference corresponds to the difference between a stable and an unstable filter with the same amplitude response, it indicates how the phase of the target must be adapted in order to get a stable filter. However, the only allowed change to the target is adding a fixed delay, while the phase difference between the stable and unstable filter is not a fixed delay. To overcome this, the average of the difference between the group delay of the stable and unstable filters is added as delay to the target.
4) A new filter is calculated with the delayed target.
5) Steps 2)–4) are repeated until a stable filter is found.

Most physical systems have a lowpass characteristic, i.e., they do not react on excitation signals beyond a certain frequency, say \( \omega_s \). Because a feedforward controller needs to approximate the inverse of this lowpass response, it should have a very high gain beyond this \( \omega_s \). Even if the frequencies included in \( \Omega \) are all well below \( \omega_s \), because \( \omega_s \) is usually lying outside the band of interest anyway, the filter resulting from (15) will still have this high gain. Such behavior is unacceptable, since this high gain will amplify any high-frequency noise beyond an admissible level. Therefore it is necessary to control the gain of the feedforward controller outside the band of interest \( \Omega \). This is possible by extending \( \Omega \) up to the Nyquist frequency. The corresponding \( T(\omega) \) are set to zero and a small weight \( W(\omega) \) is associated to them.

When the feedforward controller only needs to operate in a small band compared to the Nyquist frequency, its design will become very difficult because the zeros and poles operating in the band of interest will all be clustered together in a small region of the \( z \)-plane (near \( z = 1 \)) and a lot of zeros need to be spent beyond this active region for controlling the gain. To circumvent this problem, the feedforward controller can be designed at a lower sampling frequency. The output signal at this lower sampling frequency needs then to be interpolated to the original sampling frequency, which is mainly a low-pass filtering operation (see \[1\] for details). In this way, the active frequency band of the feedforward controller will span most of the reduced Nyquist band and the gain at higher frequencies will be small due to the interpolation process.

C. Comparison of Both Methods

Compared to the EBZPETC method, the added delay method allows a better control of the errors in the frequency domain. The errors in some frequency bands can be given a greater importance than in others and if the overall error is judged too large, then the order of the approximating filter simply has to be increased until the required error is attained. Furthermore, the added delay method allows to control the gain beyond the active frequency band, which is not possible with the EBZPETC method. As a result, the feedforward controller designed with this latter method will usually have an unacceptable high gain at the high frequencies.

Most physical systems have a low-pass characteristic. Therefore the inverse of their model, i.e., the feedforward transfer function, exhibits a very high gain near the Nyquist
frequency, even if the high frequencies are not included in the filter design. This behavior is unacceptable, since the noise present at these frequencies will be amplified a lot and will eventually become larger than the signal itself. With the added delay method it is possible to control the gain of the inverse filter at these “don’t care” frequency bands. This is not the case for the EBZPETC method.

Compared to the added delay method, the EBZPETC method gives a much better approximation of the inverse system model at low frequencies. Each additional feedforward term extends the low-frequency band. The added delay method aims at minimizing an objective function and as a result low frequency errors are traded off against high-frequency errors. Previous experiments have shown that high accuracy at low frequencies is very important since most common position profiles are mainly composed of low frequency components. Moreover, the method [15] (see Remark 4 of the previous section) regains the full control of the errors in the frequency domain, but as it is based on minimizing an objective function it also trades off different frequency errors.

Another interesting feature of the EBZPETC method is its simplicity with respect to on-line computations and implementation. While the EBZPETC method has a closed-form formulation, the added delay method requires a double iterative procedure; the selection of the approximation order (number of poles and zeros) and the computation of a stable approximation. Note also 1) that the preview horizon required by the added delay method is significantly larger than the preview horizon required by the EBZPETC method and 2) that the preview of the desired output signal increases with the complexity of the approximation for both methods.

IV. TEST SETUP

Both feedforward calculation methods have been evaluated experimentally on a flexible one-link robot.

The one-link robot, shown in Fig. 5, consists of a flexible aluminum prismatic beam (100 × 10 mm, length = l₀ 921 mm) with a payload (3 kg), connected to a flexible torsional aluminum beam (ϕ 20 mm, length 490 mm), which itself is connected to a direct drive brushless dc-motor.

The velocity operation mode is realized by an analog high-gain velocity loop (lag-compensator). The motor velocity input command is a voltage between -6 and 6 V, which corresponds to a desired angular velocity of -7.5 and 7.5 rad/s.

A built in encoder measures the angular position. It has a resolution of 1 024 000 pulses per revolution. The motor position is expressed in radians. Strain gauges near the axis of rotation measure the beam deflections. The strain gauge signal is expressed in \( \text{m/m} \). The payload position is expressed in mm arc length. It corresponds to the payload angular position of the mass-spring-damper system (see Fig. 1). This position is not measured directly, but is calculated from an experimentally determined static relation between the strain gauges and the motor encoder signals

\[
y[k] = l_0 \theta_n[k] + a_1 \varepsilon[k]
\]

\( l_0 = 921 \text{ mm}, \quad a_1 = -9.3617 \times 10^{-2} \text{ mm/m/m} \).

IV. EXPERIMENTAL RESULTS

A. Objectives

The two described feedforward methods are evaluated experimentally on the flexible one-link robot which is described above. The modeling of the robot is based on an errors-in-variables frequency domain identification method. The bandwidth of the state feedback controller is designed to be 5 Hz. The evaluation trajectories are 1) a broad-band periodic (i.e., a multisine) position profile and 2) a smooth position step based on a ninth-order polynomial. The feedforward controllers, which are added to the feedback controller, are designed to track these trajectories accurately up to 5 Hz.

B. Identification of the Robot Dynamics

The identification of the transfer functions \( H_{u \rightarrow e}(z^{-1}) \) and \( H_{e \rightarrow y}(z^{-1}) \) [(3) and (4)] is performed with an errors-in-variables frequency domain identification method, called ELiS [7]. Starting from the measured Fourier coefficients of the input and the response signals (motor position and strain gauge signal), the \( z \)-domain transfer functions are estimated [4], [7], [2]. This identification approach allows to account for disturbing noise on the input and output Fourier coefficients.

The measurements are made in the time domain and transformed to the frequency domain using the discrete Fourier transform (DFT), calculated with the fast Fourier transform (FFT). In order to avoid leakage, a multisine \( x(t) \), which is a broad band periodic excitation signal, is used

\[
x(t) = \sum_{k=1}^{F} A_k \sin(2\pi f_k t + \phi_k)
\]
with $F$ the number of frequency components included in the signal and $A_k$, the user definable amplitude of the component at frequency $l_k, f_0, l_k \in \mathbb{N}$. If leakage is to be avoided the base frequency $f_0$ must be equal to $F_s/N$, with $F_s$ the sampling frequency and $N$ the number of samples used in the FFT calculation. The phases $\phi_k$ are chosen [7], [8] such that the peak value of the signal is as small as possible such that it is possible to make measurements with a maximum signal to noise ratio. Broad-band excitation signals with minimum peak value allow to reduce the measurement time significantly.

Several periods are measured. Then, the Fourier coefficients, the variance of the noise on the input and output Fourier coefficients, and the cross-correlation between the input and output noise, which can be very high on mechanical systems, are calculated [9].

For the one-link robot, a large frequency range was needed. For this reason an excitation signal with a semilogarithmic distribution of the frequencies was made in the frequency band from 0.06 to 10 Hz, with constant amplitude. The state feedback controller design is based on the identified models $H_{C+e}(z^{-1})$ and $H_{C+\theta_m}(z^{-1})$ ([3] and [4]).

In order to design the feedforward controller, an accurate input–output model of the closed-loop system is necessary. Therefore, a new identification of the system with the designed feedback controller active, is performed, using the same ELiS techniques as previously described. The model used for the total closed-loop system is

$$H_{CL}(z^{-1}) = \frac{\sum_{i=0}^{6} b_i z^{-i}}{(1-z^{-1}) \left( \sum_{i=0}^{6} q_i z^{-i} \right)}.$$ (18)
Table I
Identified Parameters of the Closed-Loop System

<table>
<thead>
<tr>
<th>$z^{-i}$</th>
<th>$b_i$</th>
<th>$a_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.03255790</td>
<td>0.00452324</td>
</tr>
<tr>
<td>$z^{-1}$</td>
<td>0.19567601</td>
<td>0.02623347</td>
</tr>
<tr>
<td>$z^{-2}$</td>
<td>0.48965172</td>
<td>0.06348097</td>
</tr>
<tr>
<td>$z^{-3}$</td>
<td>0.65300271</td>
<td>0.08203056</td>
</tr>
<tr>
<td>$z^{-4}$</td>
<td>0.48949069</td>
<td>0.05969283</td>
</tr>
<tr>
<td>$z^{-5}$</td>
<td>0.19554521</td>
<td>0.02318987</td>
</tr>
<tr>
<td>$z^{-6}$</td>
<td>0.03252419</td>
<td>0.00375685</td>
</tr>
</tbody>
</table>

The excitation signal used for the identification is a multisine with energy up to 5 Hz and a resolution of approximately 0.025 Hz. Table I shows the identified parameters of the model (18). Fig. 6 shows the frequency response of model and measurements.

C. Feedback and Feedforward Controller Design

1) Design of the Feedback Controller: A state feedback controller has been designed using pole placement. The Butterworth filter dynamics are preserved in the closed-loop control system and all other system poles are chosen at 5 Hz with a damping ratio of one, except for the filter poles; they are retained as closed-loop poles. The chosen sampling frequency is 200 Hz.

2) Design of Both Feedforward Controllers: Both feedforward controllers approximate the inverse of the frequency response [(18) and Fig. 6] identified in the previous section.

The input of the feedforward controller is the desired payload position, and its output is the corresponding differentiated velocity signal.

a) Design of the EBZPETC feedforward controller: The feedforward controller based on the EBZPETC method is designed according to its description in Section III-A.

The total transfer function has one unstable zero: $z_1 = 1.1745$. To guarantee accurate tracking, two extra feedforward terms are added to reduce the induced gain errors [$r_1 = 3$ in (11)].

b) Design of the added delay feedforward controller: Since the robot should be able to track signals accurately up to 5 Hz, the feedforward controller must closely follow the target up to 5 Hz and its amplitude response must remain small beyond it. This results in a filter design with a very narrow pass-band (from 0 to 5 Hz), and a large stop-band (from 5 to 100 Hz), because the sampling frequency is 200 Hz. To avoid this problem, a subsampling with a factor eight is used, reducing the sampling frequency of the feedforward controller to 25 Hz. Because the low-pass filters used in the interpolation process are not ideal filters, this interpolation will change the overall frequency response of the feedforward controller, partially undoing its beneficial action. In order to account for this, the target frequency response of the feedforward controller is augmented with the inverse frequency response of the interpolation filters in its active frequency band. To add not too much complexity to the feedforward filter design, the low-pass filters used are Chebyshev type II filters, which are free of pass-band ripple. For reasons of simple and efficient implementation, the interpolation process is done in three stages, each stage interpolating by a factor of two.

Table II summarizes the used parameters for each interpolation filter, with $f_s$ the sample frequency at which that particular interpolation filter is operating, $f_{stopband}$ the cutoff frequency of the stopband and $A_{stopband}$ the attenuation in dB for the stopband. The $(1 - z^{-1})$ factor showing up in the model for the feedforward controller is calculated separately at the full 200 Hz sampling frequency, after the interpolation. Finally, the parameters of the feedforward filter design are as follows.
Fig. 9. Target position of the payload: multisine tracking test.

<table>
<thead>
<tr>
<th>$f_s$ (Hz)</th>
<th>order</th>
<th>$f_{\text{stepband}}$ (Hz)</th>
<th>$A_{\text{stepband}}$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>7</td>
<td>20</td>
<td>120</td>
</tr>
<tr>
<td>100</td>
<td>7</td>
<td>30</td>
<td>120</td>
</tr>
<tr>
<td>200</td>
<td>7</td>
<td>70</td>
<td>140</td>
</tr>
</tbody>
</table>

- The orders of the numerator and denominator are, respectively, 20 and 10.
- The inverse of the absolute value of the target is used as weighting function, guaranteeing small errors where the target is small.
- Zeros are added to the target frequency response in the band between 10 and 12 Hz, with a weight of $10^{-6}$.
- In order to obtain a stable filter, a delay of 26 samples at the 25 Hz sample frequency (208 samples at 200 Hz) is added to the target.

Fig. 8 shows the frequency response of the total feedforward controller designed in this way.

**D. Tracking Results**

Both feedforward design methods are validated and compared on the described flexible robot arm (Fig. 5). The actual payload position is calculated from the encoder and strain gauge measurements [16], and compared with the desired position.

1) **Tracking Results for a Multisine**: The first test trajectory is based on a multisine [17] with energy in the band from 0.5 to 5 Hz and with 2048 samples in one period. In order to keep the acceleration of the motor small, which is necessary to remain in the linear region of the motor, the amplitude spectrum is chosen inversely proportional to square of the frequency. In order to avoid transitional effects, the position profile is preceded with a number of zeros and the first multisine period is multiplied with a window in order to gradually increase the amplitude. Fig. 9 shows the desired position profile, i.e., the added zeros, the two multisine periods of which the first is modified.

The results of both feedforward controllers are shown in Figs. 10 and 11, respectively. The upper plot is the tracking error in mm in the time domain. The lower plot shows the frequency domain representation of the periodic part of the tracking error (i.e., the FFT of the tracking error during the second multisine period), such that no leakage occurs. The $y$-axis of these frequency plots have no absolute value, and they are only meant to compare both methods.

The results of both feedforward controllers are shown in Figs. 10 and 11, respectively. These frequency plots are included to show the different frequency behavior of both methods only. The absolute value of there $Y$ axis is not that important.

The upper plot is the tracking error in the time domain. The lower plot shows the frequency domain representation of the part of the tracking error that corresponds to the second multisine period. This part of the tracking error is periodic, such that transformation to the frequency domain using the DFT is free of leakage errors.

**Comparison of Tracking Results**

- Figs. 10 and 11 show that both the EBZPETC method and added delay method are able to accurately approximate the inverse model within the tracking bandwidth ($<5$ Hz). The added delay method also has control over the amplitude of the approximation outside the tracking bandwidth.
Both methods perform very well taking into account the high frequency content of the desired position profile. The maximum tracking error for the EBZPETC method and the added delay method is, respectively, 10.3 and 2.3 mm. Hence there is a significant difference between both methods.

The frequency domain analysis of the tracking errors shows that the EBZPETC method achieves more accurate tracking at low frequencies, while the added delay method keeps the error low over the entire band. This advantage of the EBZPETC method is important if the desired trajectory is a standard position profile, since most position profiles are mainly composed of low-frequency components.

The complexity and the required preview horizon are significantly smaller. The EBZPETC method requires a preview horizon of three samples and the added delay method requires a preview horizon of 161 samples.

2) Tracking Results for a Smooth Position Step: The second test trajectory is a smooth position step which is based on a ninth-order polynomial (see Fig. 12). The trajectory starts and ends with zero velocity, acceleration, jerk, and derivative of jerk. Together with the total displacement (5786.8 mm, i.e., one revolution of the motor) and the positioning time (2.675 s) this completely determines the polynomial. The resulting maximum payload velocity and acceleration are 6.2 m/s and 10.4 m/s², respectively. This ninth-order polynomial is sufficiently smooth in order to avoid large peaks in the feedforward command.

Comparison of Tracking Results

Fig. 13 shows the time domain representation of the tracking errors for both feedforward controllers. The frequency domain
representation of these tracking errors is not shown because it contains leakage errors due to their unperiodic nature.

The maximum tracking error for the EBZPETC feedforward controller is less than 1 mm (solid line). The added delay feedforward controller yields a steady-state error of 18 mm (dashed line), which is of the same magnitude as for the multisine considering that the step excitation is about ten times larger that the multisine excitation. The smooth step tracking error for the EBZPETC method is considerably smaller than the tracking error obtained during the multisine test, which confirms (see Fig. 10) that this method is more accurate at low frequencies.

Note that the bad results for the added delay method in the case of a smooth step are mainly due to the fact that this feedforward controller was designed to perform equally well over the entire frequency band up to 5 Hz, while the step excitation mainly contains low-frequency components. Using a different weighting, an other feedforward controller could be designed that gives more importance to these lower frequencies, and thus would perform better when excited with a smooth step.

VI. CONCLUSION

This paper compares the design, implementation and experimental results of two trajectory tracking controllers for a flexible one-link robot equipped with a VCA.

Both controllers use the same state feedback controller, but have a different feedforward design approach. They approximate the unstable inverse system model by a stable prefilter.

A detailed comparison is made between methods both from a theoretical and experimental point of view. Both prefilters perform very well taking into account the frequency content.
Fig. 12. Target position of the payload: smooth step tracking test.

Fig. 13. Tracking error for the EBZPETC feedforward controller (solid line) and for the added delay feedforward controller (dashed line).

of the desired position profile. The experimental results are very conclusive.

The tracking results highlight that while the EBZPETC is very accurate at the low frequencies, the added delayed method can make the tracking errors small in a large range of frequencies, making the overall tracking error small if the signal contains such frequencies.

ACKNOWLEDGMENT

This text presents research results of the Belgian program on Interuniversity Poles of attraction by the Belgian State, Prime Minister’s Office, Science Policy Programming. The scientific responsibility is assumed by its authors.

REFERENCES

Dirk E. Torfs received the bachelor’s degree in 1989 and the Ph.D. degree in 1995, both in mechanical engineering from the Katholieke Universiteit Leuven, Belgium.

From 1989 to 1994, he was a Research Assistant at the Katholieke Universiteit Leuven in the field of automation and robotics with emphasis on fast and accurate tracking control of motion controlled systems in the presence of flexibilities. During this period, he was involved in several industrial projects in a national and international context and projects for the European Space Agency. He is author and coauthor of several papers in proceedings and journals. In 1995, he joined Trasys Space as a Senior Project Engineer. Since 1996, he has headed the robotics and telescience department. His main activities include system engineering, product assurance analysis, and project management including control of subcontractors.

Rudi Vuerinckx was born in Belgium in 1966. He received the bachelor’s degree in engineering in 1989 from the Vrije Universiteit Brussel (VUB). Since then he had been working at the VUB as a Researcher, mainly in the field of digital filter design.

Jan Swevers was born in Leuven, Belgium, in 1963. He received the B.E. degree in electrical engineering in 1986 and the Ph.D. degree in mechanical engineering in 1992, both from the Katholieke Universiteit Leuven, Belgium. Since 1992, he has been working as a Senior Research Assistant of the Fund for Scientific Research, Flanders, at the Katholieke Universiteit Leuven, Department of Mechanical Engineering, Division P.M.A. (Production, Machine Design, and Automation), and he became Assistant Professor in 1995. His research interests include various aspects of system identification and control.

Johan Schoukens was born in Belgium in 1957. He received the bachelor’s degree in engineering in 1980, the degree of doctor in applied sciences in 1985, and the qualification to teach at university level (geaggregeerde voor het hoger onderwijs) in 1990, all from the Vrije Universiteit Brussel (VUB). He is presently a Research Director of the Fund for Scientific Research (FWO) and part-time Lecturer at the VUB. The prime factors of his research are in the field of system identification for linear and nonlinear systems.