MODELING, IDENTIFICATION AND CONTROL, 1993, VOL. 14, NO. 3, 161–174
doi:10.4173/mic.1993.3.4

Frequency domain synthesis of trajectory learning controllers for robot manipulators

TOM KAVLI†

Trajectory learning control is a method for generating near to optimal feedforward control for systems that are controlled along a reference trajectory in repeated cycles. Iterative refinements of a stored feedforward control sequence corresponding to one cycle of the control trajectory is computed based upon the recorded trajectory error from the previous cycle. Several learning operators have been proposed in earlier work, and convergence proofs are developed for certain classes of systems, but no satisfactory method for design and analysis of learning operators under the presence of uncertainties in the system model have been presented. This article presents frequency domain methods for analysing the convergence properties and performance of the learning controller when the amplitude and phase of the system transfer function is assumed to be within specified windows. Experimental results with an industrial robot manipulator confirm the theoretical results.

1. Introduction

Trajectory learning control is a method for generating near to optimal feed-forward control for systems that are controlled along a reference trajectory in repeated cycles. A recorded trajectory of the feedforward control corresponding to one cycle of the reference trajectory is incrementally adjusted for each repetition of the control cycle until it converges to a final value. The incremental adjustments are computed by a learning operator that normally is implemented as a digital filter operating on the measured control error of the last cycle. This article presents practical methods for analysis and design of the learning filters based upon analysis in the frequency domain. Criteria for analysing the convergence properties and performance of the iterative learning algorithm when the amplitude and phase of the system transfer function is assumed to lie within specified windows are included. The method has been used for the design of a trajectory learning controller for a Trallfa TR4000 industrial robot manipulator, and experimental results confirm the theoretical results.

The two most popular control schemes for advanced robot manipulator control—*model-based feedforward control* and *computed torque control*—both require accurate inverse dynamic models for the manipulator arm and actuators to achieve good performance (Khosla 1988, An, Atkeson and Hollerbach 1986, Egeland 1986). To compensate for nonmodeled dynamics and model errors, the trajectory learning control scheme has been proposed (Craig 1988). For hydraulic manipulators, it is very difficult to obtain good dynamic models due to unknown and not easily described nonlinearities in the servo valve/actuator system (Kavli 1990, Williams and Mason 1990, Watton 1989). The model-based control techniques are thus less applicable for this kind of system, and trajectory learning may show its full potential.

Received 12 March 1993.

† SINTEF SI, Box 124 Blindern, N-0314 Oslo 3, Norway.

Reprinted, with permission, from *Journal of Robotic Systems* 9(5), 663–680 (1992).

The trajectory learning algorithm as initially proposed by Arimoto *et al.* (1984) had the form

$$u^{k+1}(t)=u^{k}(t)+\Gamma\dot{e}^{k}(t)$$

where $u^k(t)$ is the feedforward control and $\dot{e}^k(t)$ is the time derivative of the trajectory error after the kth iteration. Γ is a gain constant. This learning algorithm was proven to converge to zero error when applied to linear time-varying first-order systems. This learning algorithm was later generalised to include various linear combinations of the trajectory error and its first and second time derivatives, $e^k(t)$, $\dot{e}^k(t)$, and $\ddot{e}^k(t)$:

$$u^{k+1}(t)=u^{k}(t)+\alpha e^{k}(t)+\beta\dot{e}^{k}(t)+\gamma\ddot{e}^{k}(t) \tag{1}$$

where α, β, and γ are learning gain constants (Arimoto, Kawamura, Miyazaki and Tamaki 1985). Kawamura, Miyazaki and Arimoto (1988) proved the convergence of this algorithm with $\gamma=0$ for linear time-varying second-order systems. Hauser (1987) proved the convergence of (1) used for a class of nonlinear time-variant systems with linear input action and provided a measure for how close the model should be to the actual system to guarantee convergence. Heinzinger, Fenwick, Paden and Miyazaki (1989) extended this proof to more general learning operators that also include the integral of the trajectory error in the update.

Bondi, Casalino and Gambardella (1988) proposed to set the learning gains equal to the corresponding closed-loop feedback gains. Oh, Bien and Suh (1988) used parameter estimation techniques to identify the time-varying plant dynamics, and the values of learning gain constants are computed so that the learning operator corresponds to the inverse of the plant dynamics. Lunde and Balchen (1990) compared the performance of variations of (1) in a series of simulation experiments. Bien and Huh (1989) further generalised the learning algorithm (1) by updating the control trajectory not only based upon the last recorded trajectory error but also including earlier error trajectories in the learning algorithm, and by that achieved faster convergence and increased noise immunity.

Craig (1984, 1988) chose to implement the learning operator in the more general form

$$u^{k+1}(z)=u^{k}(z)+L(z)e^{k}(z) \tag{2}$$

where $L(z)$ is a causal or anticausal FIR filter giving more freedom in the design of the learning operator, and also argues for the necessity of low pass filtering the error signal to eliminate the effect of unmodeled higher-order dynamics.

Atkeson and McIntrye (1986) argued that the complete nonlinear inverse model of the manipulator dynamics should be used as the learning operator and showed instability problems when a simplified model is used. In Hideg and Judd (1988), a frequency domain analysis of learning operators of the form (2) can be found.

In all the referred work, the robustness of the learning controller is analysed under the assumption that the order of the plant dynamics is known. Unmodeled higher-order dynamics or unmodeled transfer delays in the system may introduce phase errors in the iterative adjustments of the feedforward signal, and oscillations may build up at frequencies with more than 90° phase error. Especially for hydraulic manipulators, where the elasticity of the hydraulic fluid and the nonlinearity of the actuator system introduce significant higher-order dynamics, serious problems with high-frequency oscillations have been observed using learning operators of any of the forms given above. A general analysis of the performance and convergence properties of the class of learning operators given by (2) is presented in the next section.

Moore, Dahleh and Bhattacharyya (1989) proposed a further generalisation of (2) by adding a second filter that operates on the previously learned feedforward signal, resulting in a learning operator of the form

$$u^{k+1}(z) = B(z)u^k(z) + L(z)e^k(z) \tag{3}$$

The new filter $B(z)$ introduces new design flexibility that may be used to improve the convergence properties of the learning operator, and simulation results show the abilities of the system for controlling systems with unknown nonlinear dynamics.

This article presents in the third section frequency domain methods for designing the two filters $B(z)$ and $L(z)$ of (3) and analysing the convergence properties under consideration of anticipated model errors. Based upon these methods, learning filters are then designed for a hydraulic robot manipulator, followed by experimental results confirming the theoretical results.

2. Iterative improvements of the control trajectory by learning

Consider a linear time-invariant sampled servo system as shown in the lower part of Fig. 1. The plant having a transfer function, $G(s)$, is controlled to follow a desired trajectory in time, $d[n]$, by means of the discrete linear feedback controller $F(z)$ and $C(z)$. $d[n]$ denotes the reference at time $t=nT$, $n=1,\ldots,N$, and T is the sampling period. Since the reference trajectory is assumed to be repeated in cycles, $d[n]$ is periodic with $d[n]=d[n+N]$. The plant is also subject to a repetitive disturbance force, $v(t)=v(t+NT)$. In the case of a robot manipulator, the external force would typically be dynamic coupling forces between the axes and external forces generated by the robot task process. $u^k[n]$ is a learned feedforward control signal generated by repeated lookup from a time series table in the computer memory. The content of the feedforward table is adjusted for each cycle by adding a correction to the old stored values. The correction is computed by applying the learning operator (filter), $L(z)$, on the observed trajectory error from the previous cycle, $e^k[n]=d[n]-q^k[n]$, where $q^k[n]$ is the measured response of the system and the superscript k indicates the training cycle. The updating of the feedforward table is done offline based upon logged values so that the learning operator $L(z)$ does not need to be causal.

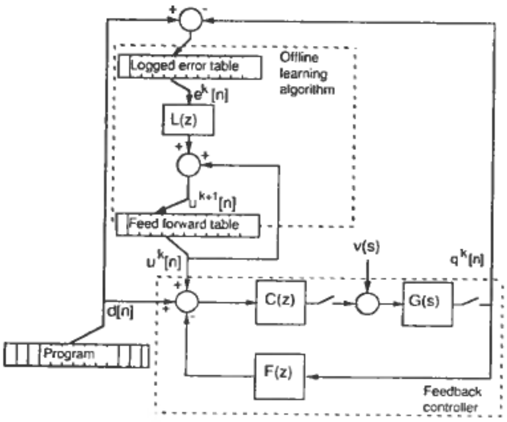

Figure 1. Block diagram of the trajectory learning system.

The trajectory error after k training cycles, $e^k[n]$, may be expressed by its z-transform as

$$e^k(z)=d(z)-\frac{C(z)G(z)[d(z)+u^k(z)]+G(z)v(z)}{1+F(z)C(z)G(z)} \tag{4}$$

$$=d(z)-P(z)[d(z)+u^k(z)]+Q(z)v(z)$$

where

$$P(z)=\frac{C(z)G(z)}{1+F(z)C(z)G(z)}$$

and

$$Q(z)=\frac{G(z)}{1+F(z)C(z)G(z)}$$

$G(z)$ is the discrete equivalence for $G(s)$, and $v(z)$ is the z-transformed sampled disturbance. If we apply a learning operator that updates the feedforward according to

$$u^{k+1}(z)=u^k(z)+L(z)e^k(z) \tag{5}$$

we get an expression of the trajectory error after $k+1$ training cycles as a function of the error at the previous cycle

$$e^{k+1}(z)=[1-L(z)P(z)]e^k(z)$$

If the feedforward signal is initialized to zero, and the initial trajectory error is $e^0(z)$, the trajectory error after k training cycles can easily be found to be

$$e^k(z)=[1-L(z)P(z)]^k e^0(z) \tag{6}$$

For this iteration to converge for all frequency components of e^0, we see that we must have

$$|1-L(\omega)P(\omega)|<1, \quad \forall \quad \omega\in[-\pi, \pi] \tag{7}$$

A rigorous proof of this convergence criterion based on the Parseval's relation may be found in Craig (1988). We see that the error in (6) would converge to zero after one single cycle if we were able to choose a learning operator equal to the inverse closed-loop transfer function of the system, i.e.,

$$L(z)=P^{-1}(z)$$

There are several reasons why this is an impractical approach. First, the reason why we want to use the trajectory learning scheme is that we do not have a good model for the inverse transfer function; and second, the transfer function of all practical manipulator arms falls rapidly toward zero as the frequency increases, implying that the learning operator should have very high gain for high frequencies. Since we must expect the recorded trajectory data to contain random high-frequency noise, this is a very undesirable property. Finally, it may well be that we do not want to force the manipulator to follow all the high-frequency signals in the reference since this might put undesirable stresses on the manipulator arm and actuators.

To determine the degree of freedom we have in the synthesis of $L(z)$, we must look at the region of convergence for the iterative learning as given by (7) and illustrated in Fig. 2. The convergence criterion requires that the complex value $[1-L(\omega)P(\omega)]$ is in the interior of a unit circle in the complex plane. It can be seen that as long as $L(\omega)$ is close to

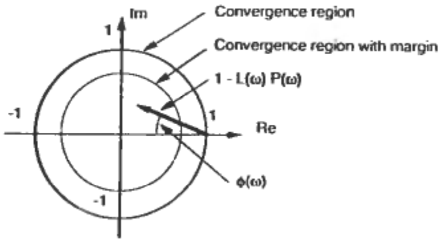

Figure 2. Convergence region for the learning operator.

$P^{-1}(\omega)$ in magnitude and phase, i.e., $L(\omega)P(\omega)\approx 1$, the system operates in the centre of the convergence region. It is also evident that the larger the deviation in phase, larger $\phi(\omega)$, the more accurate the amplitude will have to be and vice versa. More generally, the system operates within the convergence region if (8) and (9) are satisfied for all frequencies

$$|L(\omega)P(\omega)| < 2\cos[\phi(\omega)] \qquad \text{for } \phi(\omega)\in[-\pi/2, \pi/2] \tag{8}$$

$$|L(\omega)P(\omega)| = 0 \qquad \text{for all other } \phi(\omega) \tag{9}$$

where

$$\phi(\omega) = \angle[L(\omega)P(\omega)]$$

In practical situations, one would require a convergence margin to guarantee fast convergence and safe operation. One way of setting such a margin is to say that

$$|1-L(\omega)P(\omega)| < \alpha < 1, \quad \forall\omega\in[-\pi, \pi]$$

where α is a chosen constant. The equations corresponding to (8) and (9) will now be a function of α. If we choose the particular case $\alpha = 1/\sqrt{2} \approx 0{\cdot}707$ (see Fig. 2), we get the following equations:

$$\cos[\phi(\omega)] - \{\cos^2[\phi(\omega)] - 0{\cdot}5\}^{1/2} < |L(\omega)P(\omega)|$$
$$< \cos[\phi(\omega)] + \{\cos^2[\phi(\omega)] - 0{\cdot}5\}^{1/2} \qquad \text{for } \phi(\omega)\in[-\pi/4, \pi/4] \tag{10}$$

$$|L(\omega)P(\omega)| = 0 \quad \text{for all other } \phi(\omega) \tag{11}$$

For robot applications, we would normally want to compensate for all frequencies up to about the bandwidth of the robot position servo and try not to compensate for higher frequencies. To fulfil the requirements of (10) and (11), the learning operator should thus have zero gain for the higher frequencies that we do not want to compensate, and track the inverse dynamic transfer function for all other frequencies.

3. Improving robustness by introducing a second filter

Due to nonlinearities, higher-order dynamics, unknown time delays, etc., it is very unlikely that an identified inverse model will be able to compensate the plants phase and amplitude to the accuracy required by (10) for more than a limited range of frequencies. The learning operator will therefore have to drop in amplitude to zero at a certain frequency when the uncertainty in phase and/or amplitude exceeds the specified limits. Due to the limitations in the transition rate between pass bands and stop bands of digital filters, there will always be a transition zone where requirement (10) is violated. Also, there will be side lobes in the stop band with nonzero amplitudes so that requirement (11) is violated. To eliminate oscillatory problems caused by these effects,

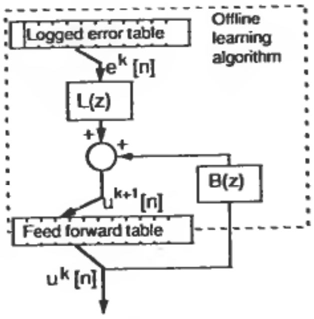

Figure 3. Modified learning operator. The added filter, $B(z)$, gives increased robustness for modeling errors.

we have chosen to low pass filter the previous learned feedforward signal before adding the new corrections generated by the learning operator. This is shown in Fig. 3 as the additional filter $B(z)$. The updated feedforward signal now becomes

$$u^{k+1}(z)=B(z)u^k(z)+L(z)e^k(z)$$

and the trajectory error update becomes

$$e^{k+1}(z)=[B(z)-L(z)P(z)]e^k(z)+[1-B(z)][1-P(z)]d(z)+[1-B(z)]Q(z)v(z)$$

For this iteration to converge for all frequency components, we must have that

$$|B(\omega)-L(\omega)P(\omega)|<1, \quad \forall\omega\in[-\pi, \pi] \tag{12}$$

for which case it converges to

$$e^\infty(\omega)=[1+L(\omega)P(\omega)-B(\omega)]^{-1}$$
$$\{[1-B(\omega)][1-P(\omega)]\,d(\omega)-[1-B(\omega)]Q(\omega)v(\omega)\} \tag{13}$$

As seen from (12), we are given extra freedom in the design of $L(z)$ since we are also free to choose $B(z)$. $B(z)$ is typically chosen as a low pass filter with zero phase and a cut-off frequency ω_B. The effect of $B(z)$ is to move the base of the vector shown in Fig. 2 along the abscissa toward the origin as the frequency increases. We see that for high frequencies when the vector base is close to the origin the phase error is of no importance as long as $|L(\omega)P(\omega)|$ is small. In the worst case when the phase error $\phi(\omega)=\pm\pi$, we have that $|B(\omega)-L(\omega)P(\omega)|=|B(\omega)|+|L(\omega)P(\omega)|$, and (12) is only satisfied if $|B(\omega)|<1-|L(\omega)P(\omega)|$. It is thus still important that $|L(\omega)P(\omega)|$ is much less than one for frequencies where the phase error may approach $\pm\pi$.

With $B(z)$ chosen as a low pass filter with cut-off frequency ω_B, $B(\omega)\approx 1$ for $\omega\ll\omega_B$ and we are back to eqns. (5) and (6). For $\omega\gg\omega_B$ we have that $B(\omega)\approx 0$ and we see from (13) that

$$e^\infty(\omega)\approx[1-P(\omega)]\,d(\omega)+Q(\omega)v(\omega)$$

which corresponds to the error without feedforward, i.e., the trajectory learning control will not improve the accuracy for high-frequency errors. For intermediate frequencies, eqn. (13) must be used in its full form to evaluate the steady-state error.

A robot manipulator is a highly nonlinear and coupled multivariable dynamic system, and the analysis carried out above cannot directly be applied to manipulator control. Most industrial manipulaters are today controlled by single-joint linear PID-type regulators. For robot manipulators designed for high speed and accuracy, it is becoming more common to use regulators with nonlinear feedback and model-based decoupling of the axes. A linear or nonlinear feedback controller will have a linearizing effect on the closed-loop transfer function. It is thus reasonable to assume that the closed-loop transfer function may be approximated by a linear model in the neighborhood of the reference trajectory. But, depending upon the degree of nonlinear compensation incorporated in the feedback controller, the closed-loop dynamics will change within certain limits as a function of the system state (arm position). The learning operators must hence be designed for stable operation for all possible variations of the closed-loop transfer function, also taking into consideration the effects of the nonlinearities. The learning operator may also be made state dependent so that it tracks the state-dependent transfer function. Although the previous analysis is strictly valid only for time-invariant systems, the extra complexity introduced by the time variable transfer function is not expected to have a significant influence on the results.

4. Synthesis of a learning controller for a hydraulic manipulator

A trajectory learning controller has been designed for and tested on the ABB Trallfa robot TR4000. This is a hydraulic robot manipulator specially designed for spray painting, and trajectory accuracy combined with high speed is of great importance for high quality and high throughput performance. This article describes only the synthesis of the controller and test results for axis one (the azimuth axis at the manipulator base) although similar results were obtained for the other axes.

4.1. *System identification*

For the synthesis of the learning operator, it was necessary to have an approximate model for the dynamics of the closed-loop servo system. This was obtained by measuring the phase and amplitude of the joint movements when different constant-frequency sine signals were applied on the input (the amplitude was 200 resolver units for all signals). The same measurements were taken with the arm (joints 2–6) locked in a centre position, fully retracted, and fully extended. Three different fourth-order autoregressive models were identified by fitting the gain and phase to the respective data points. The coefficients of the AR models are given in Table 1. Sampling frequency is 100 Hz.

The gain and phase of the three AR(4) models are plotted in Fig. 4. Resonances can be observed at approximately 4, 6, and 8 Hz for the different arm positions. These are due to mechanical and fluid elasticity. Since the hydraulic servo valve and actuator system have nonlinear torque and damping characteristics, the identified transfer function will depend upon the amplitude of the sine signal, and the identified linear models should only be regarded as coarse approximations. The inaccuracies are especially large for frequencies above the resonance frequencies. When analysing the

Table 1. Coefficients for estimated fourth-order AR models given by the transfer function $P'(z)=z^{-1}/(a_0+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4})$.

Arm position	a_0	a_1	a_2	a_3	a_4
Centred	333·3	−1173·4	1596·2	−991·8	236·7
Retracted	111·8	−357·8	455·0	−271·6	63·6
Extended	651·5	−2301·3	3093·2	−1871·5	429·2

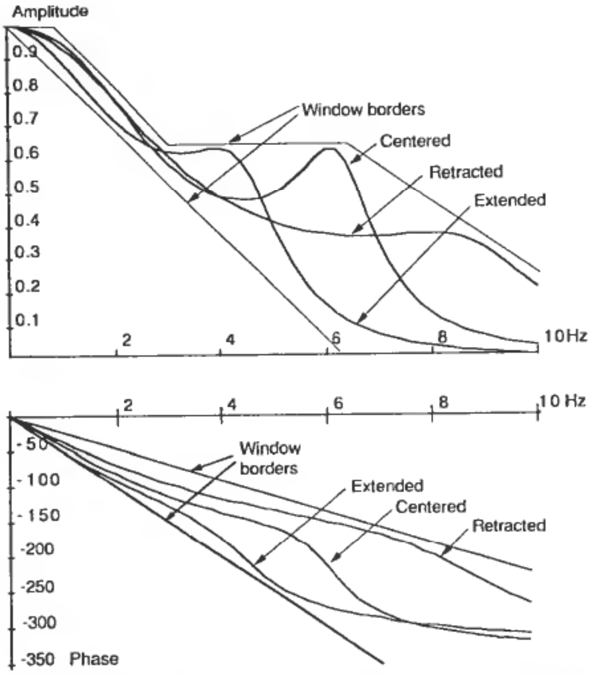

Figure 4. Linear gain and phase plots for the identified fourth-order models for axis one of the Trallfa robot manipulator. The plotted curves correspond to three arm positions. In the convergence analysis are the curves assumed to be within the indicated windows for all arm positions.

convergence properties of the learning operator, the closed-loop transfer function was assumed to lie between the indicated lower and upper window borders in phase and amplitude for all possible arm positions.

4.2. *Learning filter synthesis*

The synthesis of the filter $L(z)$ was based upon the identified fourth-order AR model for the arm in the centred position given in Table 1. By directly inverting $P'(z)$, we get a fourth-order anticausal FIR filter with gain approximately one at zero frequency and increasing to very high values at high frequencies. To suppress the gain at high frequencies, the inverse model was combined with a zero-phase 5 Hz low pass FIR filter. This low pass filter was chosen to be the same as the filter used for $B(z)$ in Fig. 3, and is a 27 coefficient Hanning window filter with coefficients given in Table 2 and gain plotted in Fig. 5. The gain of the resulting learning filter $L(z)$ is plotted in Fig. 6.

To check the convergence properties, eqn. (12) was evaluated for the complete frequency range and with $P(z)$ at the extremes of the windows shown in Fig. 4. The maximum and minimum amplitudes were combined with the maximum and minimum phases, giving a total of four cases, all plotted in Fig. 7. In all cases, the function was well within the convergence region, and extruding the 0·7 convergence margin for only a small frequency range for the case with maximum amplitude and most negative phase. In fact, the convergence criterion should be checked for all possible combinations of amplitudes and phases within the respective windows, but since we know, referring to Fig. 2, that the length of the vector is proportional to the amplitude, and that both extremes are within the convergence circle, all intermediate values must also be within

Table 2. Filter coefficients for the filter $B(z)$.

$b_0 = b_{26} =$	$-0·00026$
$b_1 = b_{25} =$	$-0·00080$
$b_2 = b_{24} =$	$-0·00101$
$b_3 = b_{23} =$	0
$b_4 = b_{22} =$	0·00322
$b_5 = b_{21} =$	0·00948
$b_6 = b_{20} =$	0·01919
$b_7 = b_{19} =$	0·03219
$b_8 = b_{18} =$	0·04764
$b_9 = b_{17} =$	0·06412
$b_{10} = b_{16} =$	0·07982
$b_{11} = b_{15} =$	0·09280
$b_{12} = b_{14} =$	0·10138
$b_{13} =$	0·10438

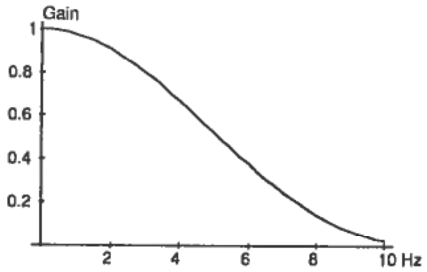

Figure 5. Linear gain plot for the filter $B(z)$.

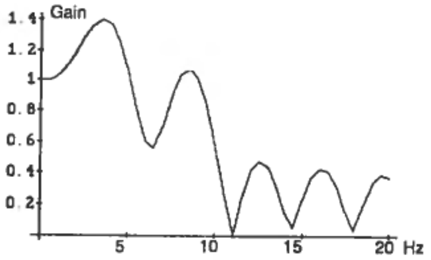

Figure 6. Linear gain plot for the learning filter $L(z)$.

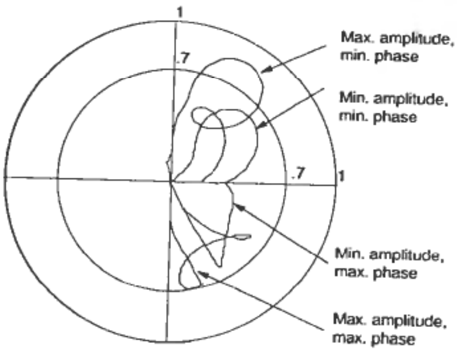

Figure 7. Plots of the convergence criterion $B(\omega)-L(\omega)P(\omega)$ for frequencies from 0–10 Hz and for the four extreme combinations of phase and amplitude as defined by the windows in Fig. 4. The curves should all be within the unit circle to guarantee convergence.

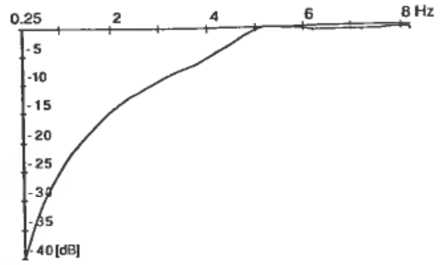

Figure 8. Theoretical reduction of the trajectory error after convergence as a function of frequency with the arm in the extended position.

the circle. We also see directly from Figs 4–6 that eqn. (12) is satisfied for all frequencies above 10 Hz independent of the phase angle due to the low gain of all transfer functions at high frequencies. For frequencies below 10 Hz, we see from Fig. 4 that the phase difference between the upper and lower window limits is always less than 300°, and since the learning filter, $L(z)$, equals minus the phase of the centred model that is inside the window the combined phase of $L(z)B(z)$ will always be in the range $-300° < \angle[L(z)B(z)] < 300°$. This means that the end point of the vector in Fig. 2 will draw an arch inside the convergence circle when varying the phase from the lower to the upper window limit at a given frequency, and convergence can be guaranteed as long as the extreme values are within the convergence region.

Figure 8 shows the theoretical compensation of the trajectory errors after convergence as a function of frequency given by eqn. (13). The extended arm model was used for this computation. As can be seen, the error should be reduced to less than 1% (−40 dB) of the uncompensated error for frequencies below 0·25 Hz. For frequencies above 4 Hz, the remaining error will be more than 50% (−6 dB) of the uncompensated error.

5. Experimental results with the trajectory learning controller

5.1. *Experimental set-up*

The robot controller was connected via a serial link to an IBM AT where the learning algorithms were run off-line. The reference trajectories were specified in joint coordinates as a time series with 10 ms sampling period. The response of the robot was logged from resolvers at the manipulator joints at the same sampling rate and transferred to the IBM. The learning filters were applied on the 100 Hz data without any preprocessing, and the new computed feedforward was added to the original reference trajectory as an offset. The modified reference trajectory was then converted into a continuous-path program with 50 ms sampling period and transferred to the robot for the next cycle. The continuous-path points were linearly interpolated in the robot controller to get a 100 Hz reference signal.

5.2. *Experimental results*

The performance and convergence of the learning feedforward controller was tested on two different test trajectories. The first trajectory, shown in Fig. 9, is a fifth-order polynomial given by

$$d(t) = -1500 + 204800t^3 - 245760t^4 + 78643{\cdot}2t^5 \quad t \in [0, 1{\cdot}25]$$

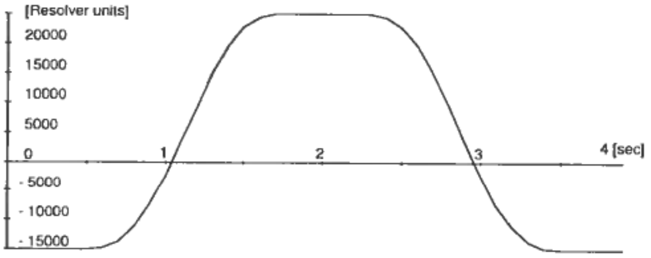

Figure 9. Polynomial test trajectory.

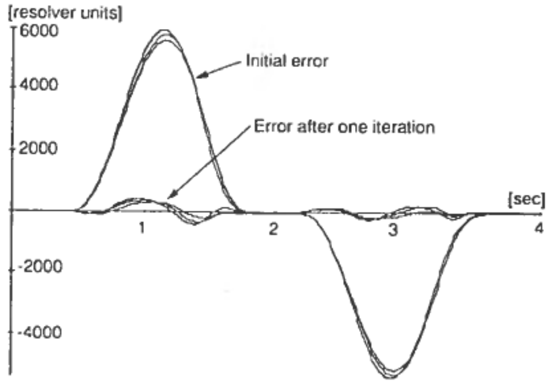

Figure 10. Trajectory error before and after first training cycle, for the arm in retracted, centre, and extended positions. The larger error is for the extended arm and the smaller for the retracted arm.

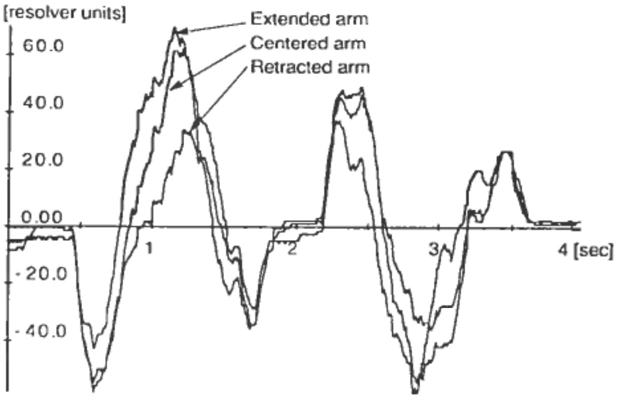

Figure 11. Trajectory error after eight training cycles for the arm in retracted, centre, and extended positions. Compare with Fig. 10.

where *t* denotes time in seconds. After half a second delay, the reference returned to the starting position by mirroring the trajectory. The learning of this trajectory was tested with the other joints of the robot arm in the centre position and the two extremes: fully retracted and fully extended. The initial trajectory error without feedforward and the error after one training cycle is shown in Fig. 10. After three cycles, the errors had converged to their final values. To test the convergence of the learning algorithm, the trajectory was trained for eight cycles with the arm in centre and extreme positions. The resulting trajectory errors plotted in Fig. 11 are similar to the error after three cycles.

There were only small and random variations between the cycles after the third iteration, and there is no sign of oscillations building up. The convergence rate is almost equal for all positions of the arm. The remaining error is about 1% of the initial error, which is in accordance with the theoretical values for signals of less than 0·8 Hz, the dominant frequency of the error. Experiments were also carried out with the 5 Hz low pass filter $B(z)$ replaced by a 6 Hz filter. The bandwidth of the $L(z)$ filter was also increased correspondingly. In this case, the plot of $B(\omega)-L(\omega)P(\omega)$ extruded the unit circle for a frequency range from approximately 3–9 Hz as shown in Fig. 12. Experiments also confirmed that oscillations of approximately 7 Hz were building up after three to five iterations.

Finally, the results from a second test trajectory that involves movement of all manipulator joints are presented. The desired path is specified in Cartesian coordinates as shown in Fig. 13. The corner points are rounded so that accelerations are limited. The path is in a horizontal plane giving rise to changing inertias and cross-coupling effects including Coriolis forces. Figure 14 shows the reference trajectory of axis one

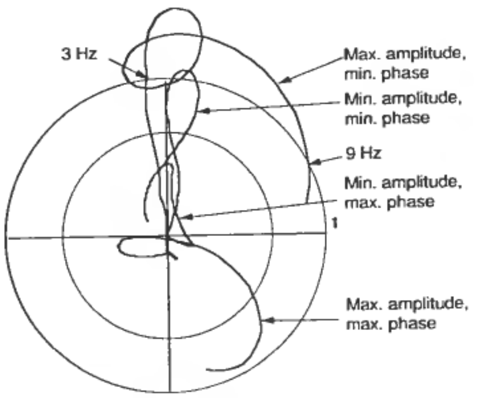

Figure 12. Plots of the convergence criterion $B(\omega)-L(\omega)P(\omega)$ for frequencies from 0–10 Hz using a 6 Hz low pass filter for $B(z)$ and for high-frequency suppression of $L(z)$. The curve extruding the unit circle between 3 and 9 Hz indicates a possible instability in the iterative learning operator. Experiments confirmed that oscillations were building up at 7 Hz.

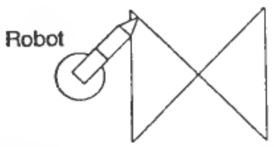

Figure 13. Top view of the test trajectory specified in Cartesian coordinates.

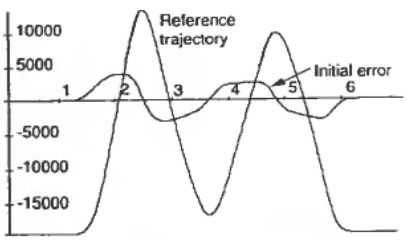

Figure 14. Reference trajectory for axis one and the initial trajectory error.

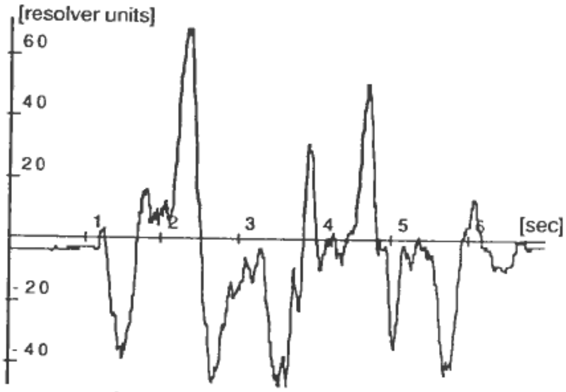

Figure 15. Trajectory error after fourth training cycle. Compare with the trajectory error in Fig. 14.

and the error without learning. The trajectory error was reduced to a similar level as for the first test after four cycles as can be seen in Fig. 15. The changing inertia and coupling forces do not seem to cause any problems.

6. Conclusions

A new design method for trajectory learning feedforward controllers intended for robot manipulator control is presented. The design of the learning operator is carried out in the frequency domain and is based upon approximate linear models of the manipulator dynamics. Practical methods for analysing the convergence and performance under consideration of the uncertainty in the linear models are included. A learning controller is designed for a hydraulic industrial robot, and experimental results confirm the theoretical analysis.

Acknowledgments

The author expresses thanks to Martin Brooks, Børn Lillekjendlie, and Rolf Bjerknes for valuable contributions in discussions and careful reading of the manuscript, and to Trond Høvde for implementing and testing the ideas. This work was supported in part by ABB Trallfa Robotics A/S and Nordisk Industrifond.

References

An, C. H., Atkeson, C. G. and Hollerbach, J. M. (1986). Experimental determination of the effect of feedforward control on trajectory tracking errors, *Proc. 1986 IEEE International Conf. on Robotics and Automation*, San Francisco, CA, April 1986, pp. 55–60.

An, C. H., Atkeson, C. G. and Hollerbach, J. M. (1988). *Model Based Control of a Robot Manipulator* (MIT Press, Cambridge, MA).

Arimoto, S., Kawamura, S. and Miyazaki, F. (1984). Bettering operation of robots by learning, *J. of Robotic Systems*, **1**, 123–140.

Arimoto, S., Kawamura, S., Miyazaki, F. and Tamaki, S. (1985). Learning control theory for dynamical systems, *Proc. of the 24th Conf. on Decision and Control*, Ft. Lauderdale, FL, December 1985, p. p. 1375–1380.

Atkeson, C. G. and McIntyre, J. (1986). Robot trajectory learning through practice, *Proc. 1986 IEEE International Conf. on Robotics and Automation*, San Francisco, CA, pp. 1737–1742.

Bondi, P., Casalino, G. and Gambardella, L. (1988) On the iterative learning control theory for robotic manipulators, *IEEE J. of Robotics and Automation*, **4**, 14–22.

Bien, Z. and Huh, K. M., (1989). Higher-order iterative learning control algorithm, control theory and applications, *IEEE Proc.*, **136**, 105–112.

CRAIG, J. J. (1984). Adaptive control of manipulators through repeated trials, *IEEE American Control Conf.*, San Diego, CA, pp. 1566–1573.

CRAIG, J. J. (1988). *Adaptive Control of Mechanical Systems* (Addison-Wesley, Reading, MA).

EGELAND, O. (1986). On the robustness of the computed torque technique in manipulator control, *Proc. 1986 IEEE International Conf. on Robotics and Automation*, San Francisco, CA, pp. 1203–1208.

HAUSER, J. E. (1987). Learning control for a class of nonlinear systems, *Proc. of the 26th Conf. on Decision and Control*, Los Angeles, CA, pp. 859–860.

HEINZINGER, G., FENWICK, D., PADEN, B. and MIYAZAKI, F. (1989). Robust learning control, *Proc. of the 28th Conf. on Decision and Control*, Tampa, FL, pp. 435–440.

HIDEG, L. M. and JUDD, R. P. (1988). Frequency domain analysis of learr-ing systems, *Proc. of the 27th Conf. on Decision and Control*, Austin, TX, pp. 586–591.

KAWAMURA, S., MIYAZAKI, F. and ARIMOTO, S. (1988). Realisation of robot motion based on a learning method, *IEEE Trans. on Systems, Man, and Cybenerrics*, **18**, 126–134.

KAVLI, T. (1990). Nonuniformly partitioned piecewise linear representation of continuous learned mappings. *Proc. 1990 IEEE International Workshop on Intelligent Motion Control*, Istanbul, Turkey, pp. 115–122.

KHOSLA, P. K. (1988). Some experimental results on model-based control schemes, *Proc. 1988 IEEE International Conf. on Robotics and Automation*, Philadelphia, PA, pp. 1380–1385.

LUNDE, E. and BALCHEN, J. G. (1990). Practical trajectory learning algorithm for robot manipulators, *Proc. IEEE International Conf. on Robotics and Automation*, Cincinnati, OH.

MOORE, K. L., DAHLEH, M. and BHATTACHARYYA, S. P. (1989). Iterative learning for trajectory control, *Proc. of the 28th Conf. on Decision and Control*, Tampa, FL, pp. 860–865.

OH, S. -R., BIEN, Z. and SUH, I. H. (1988). An iterative learning control method with application for robot manipulator, *IEEE J. of Robotics and Automation*, **4**, 508–514.

OPPENHEIM, A. V. and SCHAFER, R. W. (1975). *Digital Signal Processing* (Prentice Hall, Englewood Cliffs, NJ).

WATTON, J. (1989). *Fluid Power Systems* (Prentice Hall Int. (UK) Ltd).

WILLIAMS, S. J. and MASON, B. (1990). Impact of control system design on a flexible manufacturing system for colliery arches, *IEEE Proc.*, **137**, 157–164.