Iterative Learning Control

A theoretical and practical overview

B.P. Lemmen BSc.
CST 2010.055

Eindhoven, August 19, 2010
## Contents

1 Experimental setup ................................................. 7
  1.1 Printer .......................................................... 7
  1.1.1 Controller design ........................................... 8
  1.2 Two inertia ..................................................... 9
  1.2.1 Controller design ........................................... 10

2 Frequency domain ILC .............................................. 13
  2.1 Simulations ..................................................... 15
  2.2 Experiments .................................................... 17

3 Parameterized ILC .................................................. 21
  3.1 Simulations ..................................................... 22
  3.2 Experiments .................................................... 25

4 Hankel ILC .......................................................... 29
  4.1 Simulations ..................................................... 31
  4.2 Experiments .................................................... 32

5 Conclusions ........................................................ 35
Introduction

The control of a plant can be split in two parts. Stability and performance. Both of these are considered when designing a PID feedback controller. For instable systems the stabilization of course has priority. When tuning for performance a limiting factor is the robustness of the controlled system. Robustness margins have to be considered to assure a stable controlled system under various conditions. Performance with this kind of controller is also limited to the fact that an error has to be present for the controller to produce an input for the plant. So to increase performance feed-forward can be introduced. With a feed-forward controller an input signal is produced based on the knowledge of the reference signal the plant should follow. Iterative learning control can be used for the computation of a feed-forward signal in order to reduce the systematic error that occurs. The systematic part of an error is the part that, when following the same reference over and over again, occurs every trial.

In this report three different iterative learning control (ILC) setups are discussed. All of them will be tuned off line using the error that was measured during the trial. Both the theory and practical implementation will be discussed in the from of a simulation and a physical experiment. The three different ILC types are:

- Frequency based ILC. With this type of ILC frequency content of the closed loop process will be used to compute a learning filter. This learning filter will, on the basis of the error in the last trial, compute a better feed-forward signal. A second filter will also be used for stability and robustness of the algorithm.

- Parameterized ILC. In this type of ILC the feed-forward signal is based on a linear parametrization. So for each derivative of the reference signal a parameter is computed such that a optimal feed-forward signal is reached. With the parameters resulting from this ILC it should be possible to also follow different reference trajectories.

- Hankel ILC. Hankel ILC is different from the first two types because it only uses a part of the trial. First an actuation part. This is where the feed-forward signal is input to the control loop. After this actuation window the error is measured during a sensing window. So this ILC will try to influence the future error with its feed-forward signal. As will be made clear in this report it can be used to reduce the residual error after a motion.

A more extensive theoretical insight into ILC than given in this report can be found in [1], [2] and [3]. Note that all of these ILC’s are based on the error measured during the execution of the trial. After the trial this error is used for the updating of the feed-forward signal. So during the execution of the trial the ILC’s can be seen as a feed-forward controller. While in between trials it is a feedback controller. In the first chapter the experimental setups used for both simulation and experiments are introduced and analyzed. After this the three different ILC algorithms are discussed.
Chapter 1

Experimental setup

Before any ILC controller will be discussed first the experimental setups used for all the different controllers will be discussed in this chapter. Two different setups will be used. The first one is a printer which will be used for the frequency based and parameterized ILC. Second is a two inertia setup for the Hankel ILC. For both these experimental setups a model will be derived in this chapter. Also the PID controller used for the control of these setups is discussed.

1.1 Printer

For both the simulations and the practical experiments of the frequency based ILC and parameterized ILC a printer is used as shown in figure 1.1. In this case it is possible to control the position of the print-head. The movement of this print-head is driven by an electric motor which is supplied with a voltage between $-2.5 \text{ [V]}$ and $2.5 \text{ [V]}$ by an external amplifier.

As a first step a frequency response is measured for this printer. The resulting bode diagram is shown in figure 1.2. Using this data a fourth order model, with two integrators, is computed. Equation 1.1 is the resulting transfer function.

![Figure 1.1: Printer used for experiments](image)


\[ G(s) = \frac{1.455 \cdot 10^{-10}s^2 - 1.788 \cdot 10^{-7}s + 6.267 \cdot 10^{10}}{s^4 + 104.8s^3 + 1.549 \cdot 10^6s^2} \]  \hspace{1cm} (1.1)

### 1.1.1 Controller design

For the control of the position of the printer a PID controller is designed. The controller consists of a gain and a lead filter around the bandwidth of 5 Hz, so a zero at 5/3 Hz and a pole at 5 \cdot 3 Hz. Because the eigenfrequency is highly damped no Notch filter is needed. This is sufficient to create a stable closed loop system. In figure 1.3 the open loop bode diagram and Nyquist diagram are shown.

Figure 1.3: Open loop and Nyquist diagram for controlled printer system

Figure 1.4 shows the sensitivity plot for the closed loop printer system.
1.2 Two inertia

For the Hankel ILC a different plant will be used. This is because of the amount of friction present in the printer. As a result of this an input has little effect on a future output. Therefore the same two inertia setup is used as in [6]. This setup consists of an electrical motor of which the input voltage can be controlled. The motor is directly connected to a mass and a second mass is connected by a flexible shaft. For this experiment the encoder on the second mass is used as the output for the system. The measured bode diagram and the model are shown in figure 1.5. This bode diagram clearly shows the eigenfrequency at a frequency of approximately 59 Hertz.

Figure 1.5: Bode diagrams for model and FRF measurement

The model that was fitted on the FRF measurement is in equation 1.2.
\[ G(s) = e^{-0.0025s} \frac{5.081 \cdot 10^8}{s^4 + 1.793s^3 + 1.361 \cdot 10^5 s^2} \] \hspace{1cm} (1.2)

Note that in this model also a output delay (0.0025 seconds) is included. This property can be derived from the phase delay dropping for higher frequencies.

### 1.2.1 Controller design

As with the printer a PID controller is designed for the control of this system. Unlike the printer a notch filter will be used for the suppression of the eigenfrequency. The zeros of this Notch filter are at 58.7 Hz with a damping of 0.003. The poles are at 100 Hz with a damping of 0.5, this is done so the phase lag of the poles does not influence the phase lead necessary around the bandwidth. For this phase lead a lead filter is used this time around a bandwidth of 4.3 Hz. In figure 1.6 the open loop bode diagram and Nyquist diagram are shown.

![Open loop Bode and Nyquist diagrams](image.png)

**Figure 1.6:** Open loop and Nyquist diagram for controlled two inertia system

Figure 1.7 shows the sensitivity plot for the closed loop two inertia system.

![Sensitivity plot](image.png)

**Figure 1.7:** Sensitivity for controlled two inertia system
Using these results it is possible to compute an ILC controller. In the next chapters a frequency domain, parameterized and Hankel ILC controller will be determined respectively. Keep in mind that changing the PID controller for one of these plants will also influence the ILC performance.
Chapter 2

Frequency domain ILC

In this chapter an ILC controller will be designed on the basis of frequency information. The implementation of this control scheme is shown in figure 3.1 In this figure \( k \) represents the trial number. So each trial the feed-forward signal \( f_k \) is updated with the error \( \epsilon_k \) that was measured during the trial. For this type of ILC two filters have to be designed. First the learning filter \( L \) after this a second filter \( Q \). This \( Q \) filter will assure convergence for the ILC controller. With these two filters the updating of the feed-forward signal will be computed using equation 2.1, where \( k \) is the trial number. As in equation 2.2 the filter \( L \) can best be chosen as the inverse of the process sensitivity [7]. This process sensitivity is based on the model determined in the first chapter. For the controller to be convergent the requirement in equation 2.3 has to be fulfilled.

\[
f_{k+1} = Q(f_k + L\epsilon_k) \tag{2.1}
\]

\[
L = S_P^{-1} = (1 + PC)/P \tag{2.2}
\]

\[
|Q(1 - S_P L)| < 1, \forall \omega \tag{2.3}
\]

The computation of the learning filter \( L \) is performed using a ”Zero-Phase Error Tracking Controller” [4] algorithm. This controller is computed in such a way that it is the best possible inverse of the process sensitivity. When the system only has stable zeros the inverse can be exact. But when there are also unstable zeros these can not be inverted directly. The inverse of these zeros is then approximated as best as possible but in such a way that the resulting \( L \)

![Figure 2.1: Printer used for experiments](image)
filter is stable. In figure 2.2 both the process sensitivity and the filter L are shown. Note that for high frequencies the phase is not a direct inverse while the magnitude is the direct inverse for all frequencies.

![Bode Diagram](image)

**Figure 2.2: Process sensitivity and L filter**

Next is the tuning of the Q filter. If the L filter would be the exact inverse of the process sensitivity the Q filter would not be needed. This because in that case the term $S_P L$ in equation 2.3 would be one for all frequencies. In figure 2.3 $|1 - S_P L|$ is plotted. It can be seen that the maximum is well above 0 dB. The Q filter now is chosen as a second order low-pass filter, with the poles at 50 Hz, as in figure 2.4. In figure 2.5 it can be seen that now the requirement $|Q(1 - S_P L)| < 1$ is satisfied.

![Bode Diagram](image)

**Figure 2.3: Convergence requirement without Q filter**
2.1 Simulations

Now that the L and Q filter have been determined they can be used in a simulation. For this simulation the model for the printer as computed in the first chapter will be used. For the simulations and experiments a reference signal is needed for the printer-head to follow. The reference signal that will be used throughout this report is shown in figure 2.6.
Now the simulation can be performed. In one trial the reference signal will be input once and the error saved. After this the feed-forward signal will be updated using equation 2.1. The initial feed-forward signal is set to zero. In figure 2.7 the results are shown for different trials. Figure 2.8 shows the 2-norm of the error for each trial. It can be seen that the in four trials the 2-norm of the error has converged.
The final feed-forward signal resulting from these simulations is shown in figure 2.9.

2.2 Experiments

After the simulations now the actual printer is used and the same experiments will be performed. It is expected that, just as in the simulations, the ILC controller is able to reduce the error after several trials. But the steady state error will probably be larger. Furthermore because now the model differs more from the actual system the convergence rate will be lower. An experiment is started with a total of 30 trials. For trials 1, 5 and 30 the error is shown in figure 2.10. The 2-norm of the error for each trial is shown in figure 2.11 and in figure 2.12 the final feed-forward signal is shown. This feed-forward signal is clearly different from the final feed-forward signal obtained in the simulations.
Figure 2.10: Error for trials 1, 5 and 30

Figure 2.11: Maximum error for each trial
In this chapter a ILC was designed on the basis of frequency information. Both in simulation and experiment this ILC was able to reduce the error of the controlled system. In the simulations no encoder counts were introduced in the model. This is the reason why the controller is able to get the error smaller than one encoder count. In the experiments this is not possible. But still a good performing feed-forward signal resulted after several iterations. This type of ILC can be used on any kind of plant with controller as long as a good model is present. A disadvantage is that whenever the reference signal changes the ILC has to be start all over again.

Figure 2.12: Final feed-forward signal
Chapter 3

Parameterized ILC

In this chapter a second ILC controller will be used to determine a feed-forward signal. This ILC controller will compute parameters for a linear parametrization of the feed-forward controller. The feed-forward signal is computed using an equation like equation 3.1.

\[ u_{ff} = kfs \cdot s + kfj \cdot j + kfa \cdot a + kfv \cdot v \] (3.1)

Here the controller parameters \( kfs, kfj, kfa \) and \( kfv \) are parameters for the snap, jerk, acceleration and velocity feed-forward respectively. As will be shown in the simulation and experimentation part of this chapter a variation on this parametrization is possible as well. Of course only for reference signals with a high enough order the derivative can be computed and used for the parametrization. For this parametrization the system with feed-forward is as shown in figure 3.1. Of course a different feed-forward parametrization can be chosen as well. For easy notation purposes the notation for the feed-forward parameters is chosen as in equation 3.2.

\[ \theta_k = [kfs_k \ kfj_k \ kfa_k \ kf_k]^T \] (3.2)

Where \( k \) is the trial number. For the tuning of the parameters an objective function \( V \) is defined. \( V \) is a function of the controller parameters \( \theta_k \). Now the optimization problem is defined by the minimization of this objective function \( V \) as defined in equation 3.3.

\[ \min_{\theta_k} V(\theta_k) \] (3.3)

Figure 3.1: Printer used for experiments
As described in [5] for high tracking accuracy the objective function should be chosen as 3.4. Note that this 3.4 is the 2-norm of the error.

\[ V(\theta_k) = e_k^T (\theta_k) e_k(\theta_k) \]  \hspace{1cm} (3.4)

Where \( e_k \) is the tracking error in the k-th trial. The choice of the feed-forward controller (eq. 3.1) in combination with the objective function (eq. 3.4) will result in convex optimization problem with a global optimum. In order to find this optimum Newton’s method is applied. Equation 3.5 gives the update of the parameters by Newton’s method.

\[ \theta_{k+1} = \theta_k - \alpha_k (\nabla^2 V(\theta_k))^{-1} \nabla V(\theta_k) \]  \hspace{1cm} (3.5)

Here \( \alpha \) is the step length, \( \nabla V \) is the gradient of the objective function and \( \nabla^2 V \) is the Hessian of the objective function. These three parameters will have to be determined for the ILC controller to function. The approximation for these parameters is performed using model information. Equations for the gradient and Hessian of \( V \) are given by equations 3.6 and 3.7.

\[ \nabla V(\theta_k) = 2 \nabla e_k^T (\theta_k) e_k(\theta_k) \]  \hspace{1cm} (3.6)

\[ \nabla^2 V(\theta_k) = 2 \nabla e_k^T (\theta_k) \nabla e_k(\theta_k) \]  \hspace{1cm} (3.7)

With the choice of \( \alpha = 1 \) only an approximation is needed for \( \nabla e_k(\theta_k) \). For an easy notation \( \xi_k \) is defined as a matrix containing the feed-forward signals as in equation 3.8. The approximation is given by equation 3.9.

\[ \xi_k = [s_k \ j_k \ a_k \ v_k] \]  \hspace{1cm} (3.8)

\[ \nabla e_k(\theta_k) = -T_{u_{ff,k} \rightarrow y_k} \xi_k \]  \hspace{1cm} (3.9)

Here \( T_{u_{ff,k} \rightarrow y_k} \) is a lower triangular Toeplitz containing the Markov parameters from the process sensitivity. In the process sensitivity a model is used to estimate the plant. Note that for a constant reference signal and plant dynamics this matrix will also be constant. So only when the reference or plant dynamics change will a new \( T_{u_{ff,k} \rightarrow y_k} \) have to be computed. Now substitution of equations 3.6, 3.7 and 3.9 into equation 3.3 leads to the equations 3.10 and 3.11.

\[ \theta(k+1) = \theta_k + \alpha_k L e_k(\theta_k) \]  \hspace{1cm} (3.10)

\[ L = (\xi_k^T T_{u_{ff,k} \rightarrow y_k}^T T_{u_{ff,k} \rightarrow y_k} \xi_k)^{-1} \xi_k^T T_{u_{ff,k} \rightarrow y_k} \]  \hspace{1cm} (3.11)

Note that the inverse that has to be computed is a square matrix with a size equal to the number of parameters, so the inverse will in general be fast to compute. As will be shown in the chapter about Hankel ILC it is possible to reduce the order of \( T_{u_{ff,k} \rightarrow y_k} \) With this learning algorithm simulations and experiments can be performed.

### 3.1 Simulations

For the simulations a parametrization is chosen for the acceleration and velocity feed-forward signal. So \( \theta_k \) and \( \xi_k \) are now as defined in equations 3.12 and 3.13 Using the learning filter and update algorithm from the previous section simulations are performed. The reference signal used is the same as with the frequency domain ILC. In figure 3.2 the corresponding velocity and acceleration reference signals are plotted.

\[ \theta_k = [kfa_k \ kfv_k]^T \]  \hspace{1cm} (3.12)

\[ \xi_k = [a_k \ v_k] \]  \hspace{1cm} (3.13)
In figure 3.3 the error is shown for trial numbers 1, 5 and 11.

Looking at figure 3.4 it can be seen that the error converges after 11 iterations. In figure 3.5 the evolution of the two feed-forward parameters is plotted.
The final feed-forward parameters are $kfv = -1.7062 \cdot 10^{-7}$ and $kfa = 2.4778 \cdot 10^{-6}$. With these values the final feed-forward signal is plotted in figure 3.6. It should be noted that this feed-forward signal for the biggest part consists of acceleration feed-forward and only a small part is velocity feed-forward.
3.2 Experiments

As with the frequency domain ILC now the experimental setup is used to using the same algorithm as with the simulations. In the model used in the simulations friction is not considered. While in the experimental setup this is a large influence. For this reason a third feed-forward parameter $k_{fc}$ is defined to compensate for the Coulomb friction of the printer. The reference signal for the Coulomb friction is shown in figure 3.7.

Because during the experiments the printer can run out of bounds, sometimes only the first four seconds of the error is used for the updating of the feed-forward parameters. As the feed-forward parameters converge to their optimum the printer is less likely to run out of bounds and also the error on the last two seconds of the trial is used. Errors for trials 1, 5 and 30 are shown in
figure 3.8. For the error of trial 5 it can be seen that at the end the error is 8000 counts. This is because the is the point where the printer runs out of bounds and is stopped. So in that case only the first four seconds of the error is usable for the updating of the feed-forward signal. The 2-norm of the error for each trial is plotted in figure 3.9.

In figure 3.10 the evolution of the three feed-forward parameters is plotted. What can be seen is that while the acceleration \((kfa)\) parameter converges with the 2-norm of the error, the \(kfv\) and \(kfc\) parameters do not converge. This could suggest that the system is over parameterized.
and now a multiple of parameter values will result in the same 2-norm of the error. In this case dismissing the parameter on the velocity ($kfv$) could lead to the same result.

Using the final feed-forward parameters a feed-forward signal can be computed. In figure 3.11 this signal is compared with the final feed-forward signal resulting from the frequency domain ILC. It can be seen that the resulting signal is very similar to the signal from the frequency ILC. A benefit of this type of ILC is that with the computed parameters also other reference signals can be tracked without having to redo the optimization. Compared to the frequency based ILC the performance is worse, so the ILC is more flexible but has less performance.
Chapter 4

Hankel ILC

As a final ILC controller a Hankel ILC will be designed in this chapter. Hankel ILC differs from the previously described ILC controllers because now not the entire trial will be used for the feed-forward. With the Hankel ILC that will be designed in this chapter only on a part of the trial the feed-forward signal will be updated. Also the time window for actuation and the time window for measurement are separated so the feed-forward signal is updated based on the error that is measured after the signal has ended. As discussed in the first chapter the two inertia plant will be used for this Hankel ILC. Also a point to point motion will be used as a reference. This reference is shown in figure 4.1. In this figure also the actuation (m) and sensing (n) windows are shown. Actuation by the ILC will take place on the deceleration part of the reference. So the ILC will try to stop the system during the actuation window. The actuation window is from 0.45 [s] until 0.5 [s] and the sensing window is from 0.501 [s] until 1 [s]. Note that the sensing window follows the actuation window directly, which is a requirement for this Hankel ILC.

![Figure 4.1: Reference signal for 2 mass inertia setup](image)

The goal for this controller is to determine a feed-forward signal within the actuation window in such a way that the error in the sensing window is minimal. With the actuation and sensing windows defined the input output relation can be computed. This is done by taking a specific part of the Toeplitz matrix. The part of the Toeplitz matrix that has to be used is defined by the actuation and sensing windows, as defined by equation 4.1.
As a next step a singular value decomposition of $J_h$ is performed. This because it provides a basis to split $J_h$ into two full rank matrices. By using the controllability and observability matrix this could be achieved as well but in this case the numerics of the problem are better using the results from the singular value decomposition. The singular value decomposition is given in equation 4.2. Here $\Sigma_1$ contains the $p$ singular values, where $p$ is equal to the order of the process sensitivity. Note that this singular value decomposition is for $n \leq m$.

$$J_h = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$ (4.2)

Using the results from the singular value decomposition the two full rank matrices $J_o$ and $J_c$ are defined as in equation 4.3.

$$J_o = U_1 \Sigma_1$$

$$J_c = V_1^T$$

Now two learning filters $L_c$ and $L_o$ are defined as in equation 4.4. These filters are defined in such a way that the overall learning loop is stable and convergent [6]. When defining the division of the plant into $J_o = U_1 \Sigma_1^{1/2}$ and $J_c = \Sigma_1^{1/2} V_1^T$, $L_c$ would change to $V_1 \Sigma_1^{-1/2}$ this will work in simulation but fails to work in the experiment. The loop for this system is now shown in figure 4.2. In this figure $\epsilon_k$ is the error during the sensing window and $f_k$ is the feed-forward signal only during the actuation window.

$$L_c = V_1$$

$$L_o = (J_o^T J_o + L_c R L_c)^{-1} J_o^T$$

With this choice of $L_c$ the convergent criteria reduces to $\rho(I_p - J_o L_o) < 1$ because $J_o L_c = I$. The equation for $L_o$ now also contains a $R$ matrix which penalizes the input. By defining $R = r L_c (L_c^T L_c)^{-2} L_c^T$ with $r > 0$ and also defining a learning gain $g$ the learning filter $L_o$ is now defined as in equation 4.5. With this choice of $R$ now not the inputs are penalized but the $u_k$ in the reduced space instead.

$$L_o = g (J_o^T J_o + r I_p)^{-1} J_o^T, \quad r > 0, \quad 0 < g < 2$$ (4.5)

Now the updating algorithm for this ILC is given in equation 4.6.

$$u_{k+1} = u_k + L_o \epsilon_k$$

$$f_k = L_c u_k$$ (4.6)

The boundaries on $r$ and $g$ are defined such that stability and convergence is assured.
4.1 Simulations

As discussed in the first chapter a two inertia setup will be used for the simulations and later on the experiments. For the evaluation of the performance of this controller only the error in the sensing window will be assessed. In both the simulations and experiments values of $r = 0.1$ and $g = 0.5$ are used. The error for trials 1, 3 and 30 is shown in figure 4.3.

In figure 4.4 the 2-norm of the error is shown. It can be seen that the 2-norm is monotonic convergent. The final feed-forward signal is shown in figure 4.5.
With the simulations complete now the same learning filter can be used in the experimental setup.

4.2 Experiments

For the experiments the same reference signal, actuation and sensing window will be used. In figure 4.6 the errors are shown for trial 1, 2 and 30.
Figures 4.7 and 4.8 show the 2-norm of the error and the final feed-forward signal. What can be noted is that the 2-norm of the error converges after a few iterations. After this the 2-norm varies. This is partly because the system starts and ends at different positions each trial. Because of position dependent friction the error differs.
In this chapter Hankel ILC was used to reduce the residual error after a motion. Only a part of the total trial length was used for actuation and sensing. Furthermore actuation and sensing were set in separated time windows. But Hankel ILC could be applied to various different problems. In the case of the printer for instance it could have been used on the part of the trajectory where the print-head is turned around. The Hankel ILC could be used to make sure that in that time window the print-head is decelerated, accelerated again and reached its constant velocity at the right time.
Chapter 5

Conclusions

In this report three different ILC types were described in both theory and practical implementation for the computation of a feed-forward signal on a controlled system. The frequency based and parameterized ILC’s where used in a setup where a print-head was controlled. The Hankel ILC was used in a two inertia setup. All types of ILC where able to reduce the error in both simulations and experiments.

When comparing the results from the frequency based ILC and parameterized ILC the resulting feed-forward signals are very similar. But the frequency based ILC is able to reduce the error more when compared to the parameterized ILC. The parameters of the parameterized ILC can very easy for a different reference trajectory, as long as the system behaves linear. This is not possible for the frequency based ILC and the optimization process would have to be restarted for a new reference signal. So depending on the sort of task that needs to be performed a ILC type can be chosen.

For the Hankel ILC a different framework is used. Here only a portion of the total trial length is used for the actuation and an adjacent part used for the measurement of the error. In this case the ILC was used to reduce the error after a point to point motion. But this type of ILC can be used to perform various tasks. As with the frequency based ILC though it will have perform new computation for a different task. Time windowing as in Hankel ILC could be used as well for the parameterized ILC but not for the frequency ILC because it is based on the frequency domain rather than the time domain.

One thing that links all of these ILC types is the fact that they all use a model of the plant for the computation of the feed-forward signal. This would mean that a better model could result in better performance. Performance could be a smaller error but also faster convergence.
Bibliography


