APPLICATION OF ITERATIVE LEARNING CONTROL TO A LINEAR MOTOR.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

AUTOMATIC CONTROL LABORATORY

David Delcor Poch

Professor: Roland Longchamp
Assistants: Mark Butcher and Alireza Karimi

Table of contents:

1. Introduction:

Iterative Learning Control deals with the set of repetitive processes and the notion that performance can be increased using the information from previous iterations to improve it. As opposed to traditional controllers that yield the same error each trial ILC aims to learn from previous iterations to reduce the error from one iteration to the next. It can be shown that the error converges despite plant modelling uncertainty and repeating disturbances.

A common industrial application of linear motors is in production lines where they carry out repetitive tasks additionally as precision might be required it makes them ideal candidates for ILC.

A logical choice seems to be to combine ILC with optimisation techniques. However, most of the existing optimal algorithms are computationally complex requiring large calculation times between trials. Thus it is important to find optimal algorithms that keep the rapid convergence properties but at the same time are simple to implement and do not require extensive calculation. In [1] Owens and Feng introduced a parameterisation of an optimal ILC algorithm that has monotonic convergence and achieves zero tracking error if a positivity condition is satisfied.

In [3] Owens, Hatonen and Feng proposed a more general higher order version of the previous algorithm. The main contribution of this new version is the inclusion of "basis functions", that allow convergence to zero error even if the plant is not positive.

Inverse model type is an intuitive approach for ILC since in the ideal case of perfect knowledge of the system, using its inverse as a learning operator would lead to perfect tracking in one iteration. When the system is not perfectly modelled, as is always the case in practise, rapid convergence can still be achieved. Convergence conditions tacking into account model uncertainty have been established in [5]. Combining inverse model-based and optimality, in [5] Harte, Hatonen and Owens presented an inverse type parameter optimal ILC, where similar convergence conditions are derived.

2. Aim of the project:

The aim of the project is to apply these newly developed Iterative Learning Control algorithms, namely Parameter optimal and Inverse model-based to a linear motor system manufactured by ETEL and compare their ease of application and performance properties i.e. error convergence rate, robustness to model uncertainty and robustness to stochastic disturbances.

Some more simple techniques such as phase lead have already been successfully applied in the linear motor. The performance of this technique will be compared to the proposed algorithms.

This project is structured in three parts. First some of the underlying theory is presented. Then a model of the motor is obtained to be used in simulation, to specifically analyse the algorithms sensitivity to model uncertainty and stochastic disturbances influence. The next natural step is the implementation of the algorithms on the motor.

Finally the algorithms are compared in terms of the properties defined, some conclusions are made and advice for possible future applications is given.

3. ILC Theory

3.1. Generic ILC algorithm and problem statement:

The control objective is to find a signal u_k that allows G to track as accurately as possible y_d. The tracking operation is repetitive. Figure 1 shows a basic schema of the ILC control.

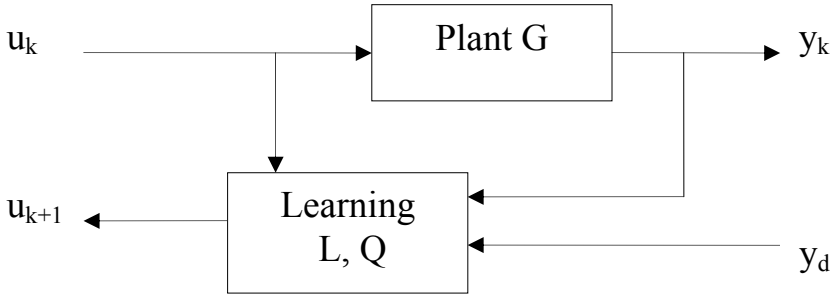

Figure 1: *Iterative learning control schematic*

The following assumptions are made in ILC:

a) Repetition length is fixed.
b) There exists a reasonable control sequence $u(t)$ in terms of bounds of amplitude and frequency that tracks $y_d(t)$.
c) The initial condition of the plant is the same for all iterations.
d) The system is iteration invariant.

A generic ILC law has the form

$$u_{k+1}(t) = Q(q)\left[u_k(t) + L(q)e_k(t)\right] \tag{1}$$

where the LTI operators Q(q) and L(q) are defined as the Q-filter and learning function, respectively and q is the forward time shift operator. Note that since there is information stored from previous iterations Q and L can be non-causal.

3.2. Parameter optimal ILC*(POILC):

Analytical results for POILC are found via the lifted system representation. This consists of a matrix model relating vectors of inputs to vector of outputs for each trial.

$$\begin{aligned} u_k &= (u_k(0), u_k(1), \cdots, u_k(N-m)) \\ y_k &= (y_k(m), y_k(m+1), \cdots, y_k(N)) \\ r &= (r(m), r(m+1), \cdots, r(N)) \end{aligned} \tag{2}$$

where N is the length of the movement.

An LTI plant can then be written as

$$y_k = G_e u_k + d \tag{3}$$

where k denotes the repetition number and d is the initial condition response.

The tracking error at trial number k is defined as $e_k = r - G_e u_k - d$ and without lost of generality we assume that $d = 0$.

In the lifted representation, the components of y_k and d are shifted by m time steps to accommodate the delay in the plant, ensuring that the diagonal entries of G_e are nonzero. Matrix G_e contains the Markov parameters of the plant.

Thus, for a plant with an m-step delay, the lifted system representation is:

$$\begin{bmatrix} y_k(m) \\ y_k(m+1) \\ \vdots \\ y_k(m+N-1) \end{bmatrix} = \begin{bmatrix} p_m & 0 & \cdots & 0 \\ p_{m+1} & p_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{m+N-1} & p_{m+N-2} & \cdots & p_m \end{bmatrix} \times \begin{bmatrix} u_k(0) \\ u_k(1) \\ \vdots \\ u_k(N-1) \end{bmatrix} + \begin{bmatrix} d_k(m) \\ d_k(m+1) \\ \vdots \\ d_k(m+N-1) \end{bmatrix} \tag{4}$$

Finally the POILC update law follows:

$$u_{k+1}(t) = u_k(t) + \beta_{k+1} e_k(t+1) \tag{5}$$

Where β_{k+1} varies each iteration and is selected to be the solution of the quadratic optimization problem:

$$\beta_{j+1} = \arg\min_{u_{j+1}} \left\{ J_{j+1}(\beta_{j+1}) : e_{j+1} = r - y_{j+1}, y_{j+1} = G_e u_{j+1} \right\} \tag{6}$$

With its performance index:

$$J(\beta_{j+1}) = \left\| e_{j+1} \right\|^2 + w\beta_{j+1}{}^2 \tag{7}$$

* The complete development of POILC theory can be found in [1].

The parameter w introduces caution in the algorithm by weighting the parameter β_{j+1} to prevent big changes in the input from one iteration to the next happening.
The stationary condition $\frac{dJ}{d\beta_{k+1}} = 0$ is a necessary and sufficient condition that gives the optimal solution:

$$\beta_{k+1} = \frac{\langle e_k, Ge_k \rangle}{w + \|Ge_k\|^2} \tag{8}$$

3.2.1. Convergence properties:

POILC has several important properties stated below*. They guarantee monotonic convergence and perfect tracking when a positivity condition is respected.

Theorem 1:

(a) The performance index satisfies the interlacing monotonicity condition $\|e_k\|^2 \geq J(\beta_{k+1}) \geq \|e_{k+1}\|^2$ with equality holding if, and only if $\beta_{j+1} = 0$.

(b) The parameter sequence satisfies $\sum_{k=0}^{\infty} \beta_{k+1}^2 < \infty$ and thus

(c) $\lim_{k\to\infty} \beta_{k+1} = 0$

Theorem 2:

Assuming the results of Theorem 1, and additionally that the symmetric part of G is positive definite i.e.

$$G_e + G_e^T > 0 \tag{9}$$

then the convergence of the error to zero is guaranteed.

$$\lim_{k\to\infty} \|e_k\| = 0 \tag{10}$$

3.2.2. Adaptive weights Parameter ILC:

In [1] a way of choosing an adaptive weighting parameter w in the performance index is presented.

The general idea is that the error becomes small as $k \to \infty$ and β_{k+1} goes to zero, which in the end might mean a possible reduction in the convergence rate. Then a new performance index is defined then as:

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 + \left(w_1 + w_2 \|e_j\|^2\right)(\beta_{k+1})^2 \tag{11}$$

* For proofs see[1]

3.3.High order parameter optimal ILC and Basis Functions:

3.3.1. High order and Basis functions fundamentals:

The main idea of the high order algorithm is to add flexibility to the solution. Not just increasing the number of previous errors but also adding previous inputs. In this way, the span of vectors from where the solution is found lies in

$$\left[u_k, u_{k-1}, \cdots, u_{k-M_1}, e_k, e_{k-1}, \cdots, e_{k-M_1}\right] \tag{12}$$

Despite the apparent power of this new law when the error decreases the elements of the mentioned span become highly collinear or infinitesimally small. Therefore the possibility of ill conditioning is high. The aim of the basis functions is to expand this span of vectors.

A new update law is proposed in [3] as:

$$\begin{aligned} u_{k+1}(k) &= u_k(t) + \delta_{k+1}(t) \\ \delta_{k+1}(k) &= \sum_{i=1}^{M_1} \alpha_{k+1}(i) u_{k-i+1}(t) + \sum_{i=1}^{M_2} \beta_{k+1}(i) e_{k-i+1}(t+1) + \sum_{i=1}^{M_3} \gamma_{k+1}(t) f_i(t) \end{aligned} \tag{13}$$

and the new cost function is

$$J_{k+1} := \left\|e_{k+1}\right\|^2 + \alpha_{k+1}^T W_1 \alpha_{k+1} + \beta_{k+1}^T W_1 \beta_{k+1} + \gamma_{k+1}^T W_1 \gamma_{k+1} \tag{14}$$

for the optimal problem

$$\left[\alpha_{k+1}^T, \beta_{k+1}^T, \gamma_{k+1}^T\right] = \arg\min\left\{J_{k+1}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1}); y_{k+1} = G_e u_{k+1}\right\} \tag{15}$$

Analogous to the procedure for the first order the optimisal solution of (14) is characterised by the equations:

$$\frac{\partial J_{k+1}}{\partial \alpha_{k+1}} = 0, \frac{\partial J_{k+1}}{\partial \beta_{k+1}} = 0, \frac{\partial J_{k+1}}{\partial \gamma_{k+1}} = 0 \tag{16}$$

leading to the equation

$$A_k \left[\alpha_{k+1}^T \ \beta_{k+1}^T \ \gamma_{k+1}^T\right]^T = B_k \tag{17}$$

whose solution is:

$$\left[\alpha_{k+1}^T \ \beta_{k+1}^T \ \gamma_{k+1}^T\right]^T = A_k^{-1} B_k \tag{18}$$

where taking the notation

$$\begin{aligned} U_k &:= \left[u_k \; u_{k-1} \cdots u_{k-M_1} \right] \\ E_k &:= \left[e_k \; e_{k-1} \cdots e_{k-M_2} \right] \\ F &:= \left[f_1 f_2 \cdots f_{M_3} \right] \end{aligned} \tag{19}$$

is equivalent to

$$A_k = \begin{bmatrix} U_k^T & U_k^T G_e^T G_e E_k & U_k^T G_e^T G_e F \\ E_k^T G_e^T G_e U_k & E_k^T G_e^T G_e E_k + W_2 & E_k^T G_e^T G_e F \\ F^T G_e^T G_e U_k & F^T G_e^T G_e E_k & F^T G_e^T G_e F + W_3 \end{bmatrix} \tag{20}$$

and

$$B_k = \begin{bmatrix} U_k^T G_e^T e_k \\ E_k^T G_e^T e_k \\ F^T G_e^T e_k \end{bmatrix} \tag{21}$$

3.3.2. Convergence properties:

In [3] the authors present the analysis of the convergence properties of the algorithm. There are 4 main results. The first two are equivalent to those presented in the first order problem. The other two analyse the effect of using previous inputs and basis functions, which in the end means a relaxation on the positivity condition.

Theorem 3:

a) If $\left(\alpha_{k+1}^*, \beta_{k+1}^*, \gamma_{k+1}^*\right)$ *is the optimal solution in (18). Then the performance index satisfies the interlacing property*

$$\left\| e_{k+1} \right\| \le J_{k+1}\left(\alpha_{k+1}^*, \beta_{k+1}^*, \gamma_{k+1}^*\right) \le \left\| e_k \right\|$$

b) The optimal parameters satisfy the condition

$$\lim_{n\to\infty} \left[\sum_{i=0}^{n} {\alpha_{k+1}^*}^T W_1 \alpha_{k+1}^* + {\beta_{k+1}^*}^T W_2 \beta_{k+1}^* + {\gamma_{k+1}^*}^T W_{3k+1}^* \right]$$

c) That implies $\lim_{k\to\infty} \alpha_{k+1}^* = 0, \lim_{k\to\infty} \beta_{k+1}^* = 0, \lim_{k\to\infty} \gamma_{k+1}^* = 0$

Theorem 4:

Assuming the results of Theorem 3, and additionally that the symmetric part of G_e *is positive definite i.e.*

$$G_e + G_e^T > 0 \tag{22}$$

then the convergence of the error to zero is guaranteed.

$$\lim_{k\to\infty} \left\| e_k \right\| = 0 \tag{23}$$

The cluster points of the algorithm:

It can be shown that as $\{e_k\}$ is bounded and G_e is non-singular, $\{u_k\}$ is bounded according to the equation $e_k = G_e\left(u^* - u_k\right)$. Consequently the sequences A_k and B_k in (17) are bounded. Subsequently, equation implies that $\lim_{k\to\infty} B_k = 0$ or equally that

$$\begin{aligned} &\lim_{k\to\infty} U_k^T G_e^T e_k = 0 \\ &\lim_{k\to\infty} E_k^T G_e^T e_k = 0 \\ &\lim_{k\to\infty} F^T G_e^T e_k = 0 \end{aligned} \tag{24}$$

Suppose that the original plant is not positive and let e_∞ be a cluster point of the algorithm, whose corresponding input is u_∞ and connected by the equation $e_\infty = r - G_e u_\infty$. From (24) and applying continuity u_∞ and e_∞ satisfy the following equations

$$\begin{aligned} &\text{(a) } u_\infty^T G_e^T e_\infty = 0 \\ &\text{(b) } e_\infty^T G_e^T e_\infty = 0 \\ &\text{(c) } F^T G_e^T e_\infty = 0 \\ &\text{(d) } \|e_\infty\| \le \|e_0\| \end{aligned} \tag{25}$$

It is interesting to see that the possible cluster points are generated by $2 + M_3$ equations, while $M_1 + M_2 + M_3$ equations define the optimal parameters. This result shows that the flexibility added in α_{k+1} and β_{k+1} has ultimately no effect on the limit error cluster, but may have an influence on the convergence rate.

The cluster points are defined by the equations in (25) where (b) and (c) depend on the nature of the plant G whereas (a) depends only on the reference signal. Note that condition (a) is equivalent to $\left(r - e_\infty\right)^T e_\infty = 0$. Condition (d) is simply a geometric condition.

Theorem 5:
Assuming that there exists a non-zero e_∞ in the limit set defined in (25). Then there exists an initial condition so that the algorithm converges to this non-zero e_∞.

Theorem 6:
Suppose that $G_e^T + G_e$ has both strictly positive and non-positive eigenvalues and let S_1 be the span of eigenvectors of $G_e^T + G_e$ that correspond to the non-positive eigenvalues of $G_e^T + G_e$. Assume that $\{f_i\}$ is selected so that $S_2 := Span\{G_e f_i\}_{j\in\{1,2,\ldots,M_3\}} \supset S_1$. Then $\lim_{k\to\infty} e_k = 0$, in norm.

Taking into account that $F^T G_e^T e_k = 0$ it is obvious that e_∞ is orthogonal to S_2. Because $S_1 \subset S_2$, e_∞ also belongs to the orthogonal complement of S_1. Bringing back to mind what S_1 is, i.e. the span of eigenvectors of $G_e^T + G_e$ that correspond to the non-positive eigenvalues of $G_e^T + G_e$; the relation $e_\infty^T \left(G_e^T + G_e \right) e_\infty = 0$ leads to $e_\infty = 0$.

To put it briefly, if the basis functions are selected in a way that considers the eigenvectors corresponding to the negative eigenvalues the algorithm will result in perfect tracking.

In practise however the number of samples N is high and a precise application of Theorem 6 would require a large number of basis functions and the dedication to find a basis for the space generated by the eigenvectors. Additionally, under these circumstances the algorithm would become computationally more complex.

Frequently, real application situations tolerate an error if it is smaller than a certain amount. In this case a selection of a subset of basis functions might reduce the final error to an acceptable level.

3.4. Inverse model based ILC:

Keeping the matrix representation of the system, the inverse model-based ILC has the update law:

$$u_{k+1} = u_k + \beta G_e^{-1} e_k \tag{26}$$

where coefficient β is introduced to influence performance
The error evolution analysis based on (27) suggests that if $\beta = 1$ the error converges in one iteration:

$$e_1 = r - y_1 = r - G_e u_1 = r - G_e \left(u_0 - G_e^{-1} e_0 \right) \tag{27}$$

and in a more general case ($\beta \neq 1$)

$$e_1 = \left(1 - \beta \right)^k e_0 \tag{28}$$

and thus convergence is assured in the range $0 < \beta < 2$.

However, these results require an exact model of the plant that is never available in practice. A nominal model subject to uncertainty may be available, though.
The nominal model can be linked to the real plant via multiplicative uncertainty

$$G_e = U G_0 \tag{29}$$

Where U is the multiplicative uncertainty of the plant.

The ILC law is in this case is

$$u_{k+1} = u_k + \beta G_e^{-1} e_k \tag{30}$$

whose error evolution, from which convergence is analysed, is

$$e_{k+1} = \left(I - \beta G_e G_o^{-1}\right) e_k \tag{31}$$

Theorem 7: *A necessary and sufficient condition for convergence of the inverse ILC algorithm is that the spectral radius of* $\left(I - \beta G_e G_o^{-1}\right)$ *is strictly less than one. Alternatively it is necessary and sufficient that* $G_e G_o^{-1}$ *has only eigenvalues in the right half complex plane and* $\beta > 0$ *has a sufficiently small magnitude.*

3.4.1. Matrix condition for robust monotone convergence:

This section provides an insight into the uncertainty admissible in inverse-model based-ILC to ensure monotonic convergence to zero tracking error.

Theorem 8: *Suppose that the symmetric matrix* $U + U^T$ *is positive definite. Then there exists a learning gain* $\beta^* > 0$ *such that* $0 < \beta < \beta^*$ *ensures that* $\|e_{k+1}\|^2 < \|e_k\|^2$. *Furthermore the value of any such gain can be obtained by satisfying the matrix inequality*

$$\left(\frac{1}{\beta_{k+1}} I - U\right)^T \left(\frac{1}{\beta_{k+1}} I - U\right) < \frac{1}{\beta_{k+1}^2} \tag{32}$$

Theorem 9: *If* $0 < \beta < \beta^*$ *and* $U + U^T$ *is a positive definite matrix, then there exists a real scalar* $\alpha \in (0,1)$ *(dependent on* β *but not on* e_0*) that* $\|e_{k+1}\| \le \alpha \|e_k\|$ *for all* $k \ge 0$.

3.4.2. Frequency domain convergence test:

Whereas in the previous section the convergence was analysed by means of matrix inequality in this section the analysis is shown in the more natural z-domain.

Theorem 10*: Suppose that* $U(z)$ *is a stable causal system and* $G(z) = U(z) G_0(z)$, *then*

(1) $U + U^T > 0$ *if* $\mathrm{Re}\left[U(z)\right] > 0$ *at all points on the unit circle* $|z| = 1$.

(2) a sufficient condition for monotonic convergence is that

$$\sup_{|z|=1} \left| \frac{1}{\beta} - U(z) \right| < \frac{1}{\beta} \tag{33}$$

Convergence condition (33) can be interpreted in terms of a Nyquist plot on the unit circle. For monotonic convergence it is enough that the Nyquist plot of $U(z)$ lies in a circle of radius $1/\beta$ and centre $(1/\beta, 0)$ on the complex plane.

A simple consequence is that as $\beta \to 0^+$ the circle fills the whole right half complex plane and convergence can be achieved for any strictly positive uncertainty and a sufficiently small gain $\beta > 0$.

Then the uncertainty tolerance is $\pm 90°$ phase shift for all frequencies but the gain tolerance is phase dependent.

3.4.3. Inverse of a Non-minimum phase system:

A common problem is that if the plant $G(z)$ has non-minimum phase zeros, its inverse becomes unstable, as the zeros become poles of the inverse model.

The procedure to overcome these is to exploit the offline characteristics of ILC allowing non-causal filtering to be used.
$G(z)$ is factorised in two parts.

$$G(z) = G_+(z)\,G_-(z) \tag{34}$$

where $G_+(z)$ contains the minimum phase zeros and $G_-(z)$ the non minimum phase zeros.

First the signal is filtered through $G_+(z)$ which might be non-causal

$$w(t) = G_+^{-1}(q)\,y(t) \tag{35}$$

and the result filtered as follows

$$u(t) = G_-^{-1}(q)\,w(t) \tag{36}$$

This can be performed by filtering the reversed sequence $p(t) = w(N-t)$ through the stable causal filter $v(t) = H_-^{-1}(q)p(t)$ and then reversing again $u(t) = v(N-t)$.

3.5. Inverse type Parameter Optimal ILC:

In the previous section the inverse model-based was introduced considering a benchmark gain β_{k+1} that can be obtained taking into account the parameter optimal algorithm [5]. Again the update law is

$$u_{k+1} = u_k + \beta_{k+1} G_o^{-1} e_k \tag{37}$$

This new update law leads to the following error evolution:

$$e_{k+1} = \left(I - \beta_{k+1} G_e G_o^{-1}\right) e_k \tag{38}$$

And β_{k+1} is chosen again to minimize the same cost function as in (6)

$$\beta_{j+1} = \arg\min_{u_{j+1}} \left\{ J_{j+1}\left(\beta_{j+1}\right) = \left\| e_{k+1} \right\|^2 + w_{k+1}\beta_{k+1}^2 \right\} \tag{39}$$

Assuming perfect plant knowledge the optimisation of (39) gives the optimal value

$$\beta_{k+1} = \frac{\|e_k\|^2}{w + \|e_k\|^2} \tag{40}$$

3.5.1. Robustness of inverse POILC:

The convergence conditions of the inverse model-based parameter optimal are derived directly from the inverse model-based. The gain β_{k+1} is derived from (40) tacking no account of the presence of uncertainty.

3.5.2. Matrix conditions for monotone convergence of inverse type POILC:

Matrix conditions for monotone convergence of the inverse parameter optimal are the same as in Theorem 8. Moreover, if (32) is satisfied for k=0 it is satisfied for all indices and thus the algorithm converges to a zero error.

3.5.3. Frequency domain convergence conditions of inverse type POILC:

Frequency domain test is exactly the same as for the fix gain inverse ILC.

$$\sup_{|z|=1} \left| \frac{1}{\beta_1} - U(z) \right| < \frac{1}{\beta_1} \tag{41}$$

$$\beta_1 = \frac{\|e_0\|^2}{w + \|e_0\|^2} \tag{42}$$

If (41) is satisfied for k=0 it is satisfied for all $k \geq 0$, and the method becomes more robust. Remember that a decrease in β relaxes the frequency condition.

3.6. Phase lead compensator:

A simple approach to ILC is the phase lead compensator of the form:

$$L(q) = \eta q^{\gamma - 1} \tag{43}$$

In [8] a frequency domain convergence test for phase lead, analogous to Theorem 7, is presented. The condition is:

$$\sup_{w \in [0, w_n]} \left| 1 - \eta e^{i\gamma wT} G\left(e^{iwT}\right) \right| < 1 \tag{44}$$

Again the condition has a graphical translation. The Nyquist plot of $1 - \eta e^{i\gamma wT} H\left(e^{iwT}\right)$ has to remain within the unit circle centred at (0,0).

A standard way to tune the phase lead controller is to select a gain η, taking into account that higher values lead to faster convergence but lower values give less sensitivity to stochastic disturbances, and then find the highest frequency at which condition (44) is violated for different values of lag γ. Because frequencies over this would make the error diverge, a low pass zero-phase filter is introduced. A possible method for zero-phase filtering is to first filter the data forward in time and then backward in time. This non-causal filtering is possible because ILC is not a real time algorithm.

Condition (44) is rewritten including the filter as:

$$\sup_{w\in[0,w_n]}\left|Q\left(e^{iwT}\right)\right|\left|1-\eta e^{i\gamma wT}G\left(e^{iwT}\right)\right|<1 \tag{45}$$

4. Identification of the plant:

A correct identification of the motor is important to be able to design the ILC algorithms and to test them in simulation in order to analyse their robustness to plant variability and disturbances.

The linear motor manufactured by ETEL is a double axis linear motor, see Figure 2. It allows very accurate positioning since the resolution of the encoders is 0.24nm. The ILC techniques are only applied on the upper axis and the lower remains fixed in a central position. The identified model is a closed-loop system with an RST controller where the T polynomial is a gain equal to the static gain of S.

4.1. Considerations regarding the identification:

Figure 2: *Picture of the double axis motor manufactured by ETEL*

For the identification a first approach preceded the ultimate below. It consisted of identifying a continuous frequency domain transfer function using Levy's method and related techniques such as output-error. The data lead to an ill-conditioned matrix in the regression algorithm. The procedure to solve this was to use logarithmically spaced points of the frequency response. However, the transfer functions found were unstable for orders higher than 4, thus the desired level of modelling accuracy was not achievable.

A different approach was thus adopted. Superposition properties were assumed and the time domain signals were added. Finally, a discrete time domain transfer function was determined by prediction error methods. The procedure is explained in more detail in Section 4.2.

An alternative to the method proposed would be to identify the system at different positions and adjust an averaged model. This procedure would have allowed a characterization of uncertainty.

4.2. The identification procedure:

The model was parametrically identified using the Matlab© system identification toolbox commands. A systematic procedure was followed.

The sampling time is the sampling time of the controller, 2Khz. The input data is a sum of sinusoids inserted in 6 frequency bands to excite the system at the maximum amplitude possible. Each experiment was repeated 5 times and the data was periodic, of period 4, to get better accuracy.

The maximum amplitude for each frequency band is actually quite small compared to the amplitude of the application movement. Thus, even though the model might be very accurate around the experimental point the uncertainty increases with distance. On the other hand ILC techniques do not suffer from the computational problems of the classical controllers when the order is high so it is possible to select higher orders.

4.2.1. Delay:

The delay was identified using the impulse response calculated from the data, since large impulse signals cannot be applied to the motor directly. Confidence intervals (3SD) were given for the zero value and real values of the impulse response, resulting in an estimated delay of 1 sample, see Figure 3. The command nk=delayest(data) that compares different ARX models (selected order=10) and different delays fits in with the previous result.

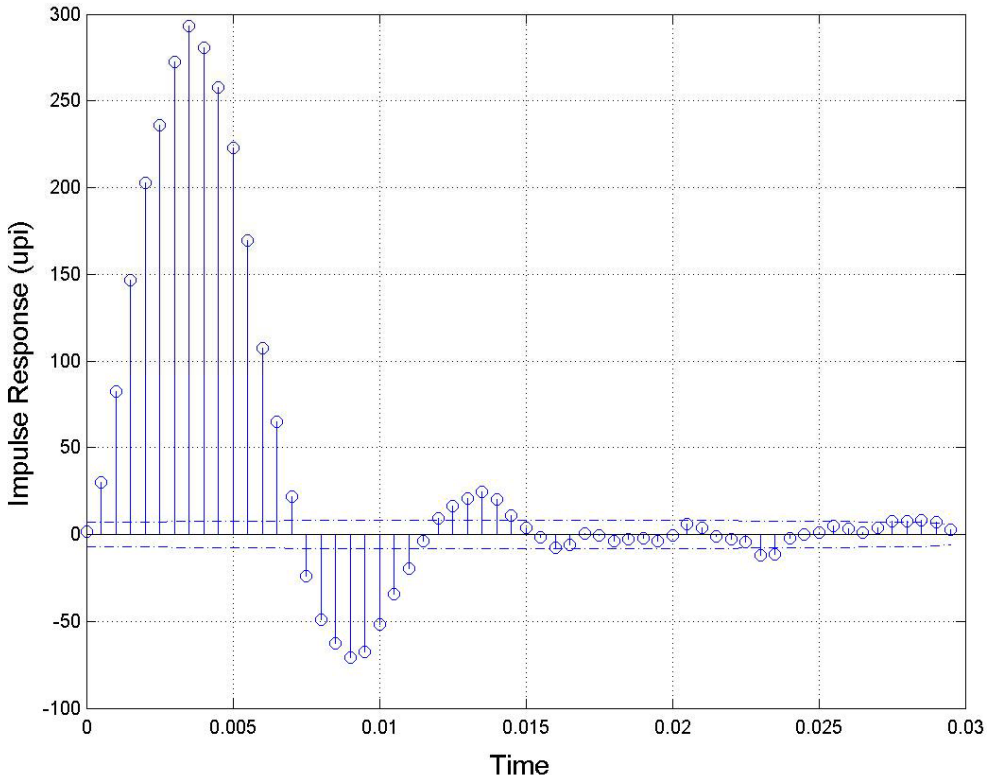

Figure 3: *Impulse response obtained from the data. Horizontal lines enclose the 3 standard deviation confidence region for the zero value.*

4.2.2. Order selection: Linear Regression (ARX)

The next step was a linear regression (ARX) to determine the order of the plant. The automatic decrease of the loss function due to plant flexibility increase was compensated using AIC (Akaike's information theoretic criterion) and MDL (Rissanen's minimum description length criterion). The orders found by these methods were respectively 39 and 37. However, Figure 4 shows a very small improvement for orders higher than 10 (20 parameters) in the loss function. Thus, smaller orders were taken.

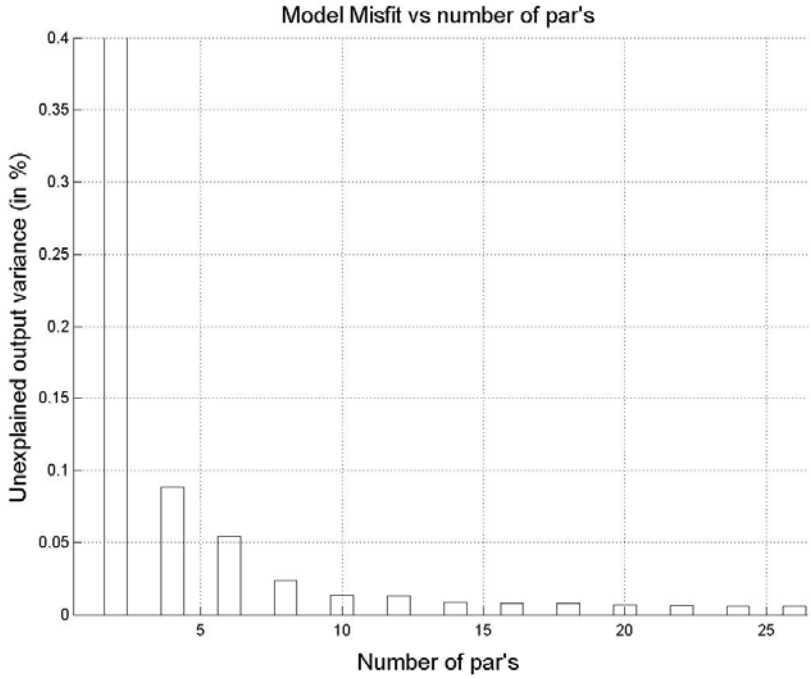

Figure 4: *Model Misfit vs number of parameters.*

4.2.3. Model selection: Non-Linear Regression (ARMAX)

After the linear regression a non-linear one was performed using the ARMAX algorithm. The model fit is 94%. A plot of the zeros and poles is shown in Figure 5. The relative degree is 1 and the model has 3 non-minimum phase zeros.

$$G(z)=\frac{0.0066731\ (z+1.546)\ (z-0.6273)\ (z+0.1115)\ (z^2-1.171z+0.5918)\ (z^2+0.2918z+0.73)\ (z^2+1.334z+1.759)}{(z-0.5967)\ (z+0.3159)\ (z^2-1.573z+0.6436)\ (z^2-1.657z+0.7986)\ (z^2+0.4082z+0.4864)\ (z^2+0.4889z+0.9561)} \quad (46)$$

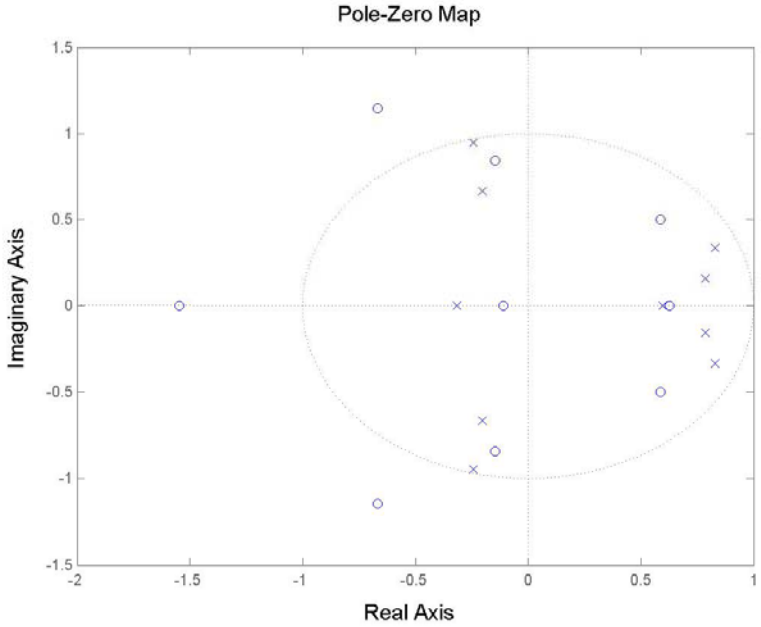

Figure 5: *zeros and poles of the identified model on the z-plane*

4.2.4. Model Validation:

Finally the model was validated in 3 ways. First of all its output was compared to a new prediction data set. Then the error was tested by analyzing its autocorrelation and its cross-correlation with the input. Finally the bode plots of the data and the model where superimposed in Figure 6 and Figure 7.

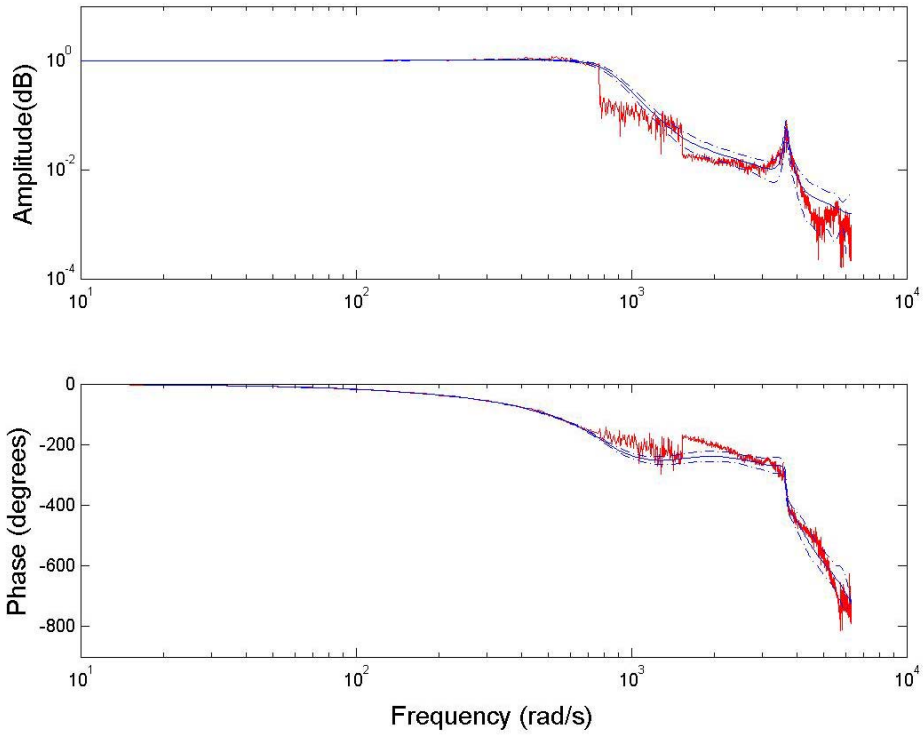

Figure 6: *Superposition of the model fit and the experimental frequency response bode plots.*

Discontinuities in the frequency response data appear because the calculation operates in different ranges and are visually magnified by the logarithmic representation. Moreover, since the model is generated adding all the signals these discontinuities do not affect the model identification.

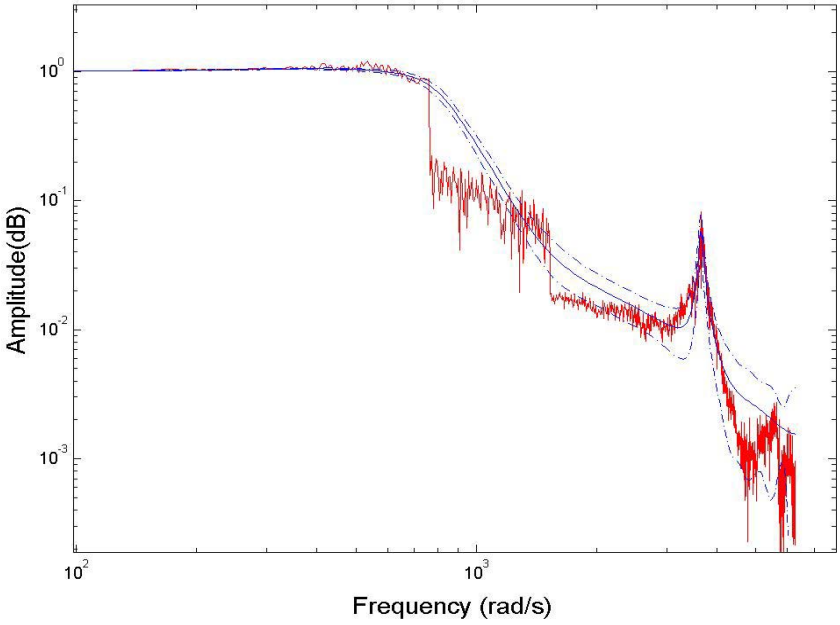

Figure 7: *Detail of the superposition of the model fit and the experimental frequency response bode plots.*

Figure 8 shows the auto-correlation function of the error and the correlation with the input. As the residuals and the input are uncorrelated the input-output model is well defined. The model of the error is not so good because the autocorrelation of the residuals is outside the confidence area.

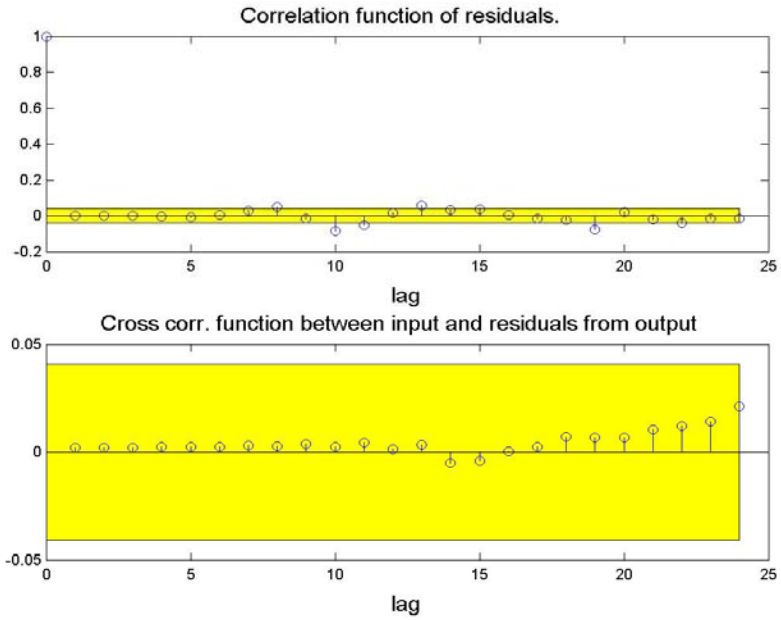

Figure 8: *Auto-correlation of residuals and cross correlation between input and output. Yellow areas represent 99% confidence regions.*

5. Simulation Results:

This section is divided in three parts. First the simulation results in the ideal case. Then the algorithms are simulated under stochastic disturbances. Finally, a variation of the model is generated and the algorithms are simulated using the initial model.

The desired output position for the simulation is a series of three low-pass filtered steps of amplitude 25 mm in the positive direction followed by a similar series of steps to reach the initial position. The desired output is shown in Figure 9. The signal lasts 2 seconds and there's a pause at the end to ensure a correct resetting.

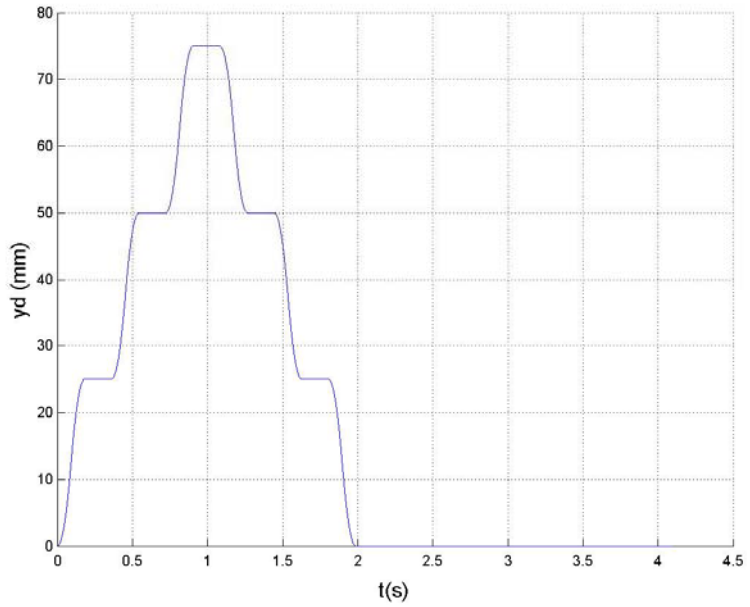

Figure 9: *Desired output signal.*

Without the ILC command the system has a reasonable tracking but still some error can be appreciated in Figure 10.

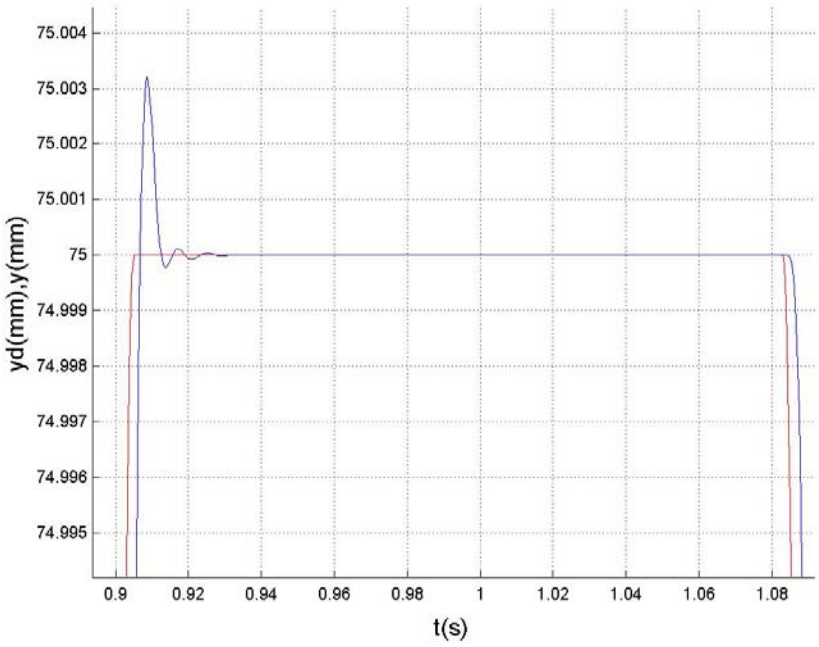

Figure 10: *Detail of desired output and simulated output without ILC.*

5.1. First order POILC:

The eigenvalues of the symmetric part of G_e lie between -0.5683 and 3.096. Therefore, the positivity condition (9) is not fulfilled and the final error will not be zero.

Internally the algorithm uses the units of the controller [upi] (user position increments). The conversion from mm to upi is: $1[mm] = 4096000[upi]$. Therefore the numerical representation of the error is high, and the effect on convergence of the weighting parameter defined in (7) can be seen for high values. Remark that the best convergence of the selected weighting parameters is obtained for w=1e7. Below this value convergence is the same.

The convergence to zero of the gain β is very fast in the first order ILC algorithm as it can be seen in Figure 12. Note that even though the tracking error does not converge to zero, the convergence is monotonic as it was stated by a) in Theorem 1.

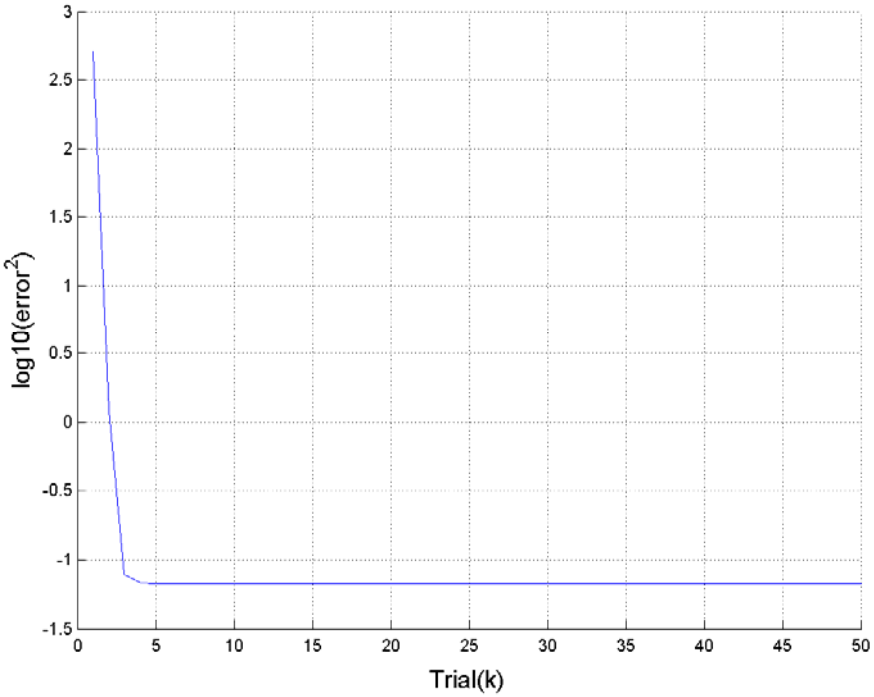

Figure 11: *Error evolution of the first order parameter optimal ILC*

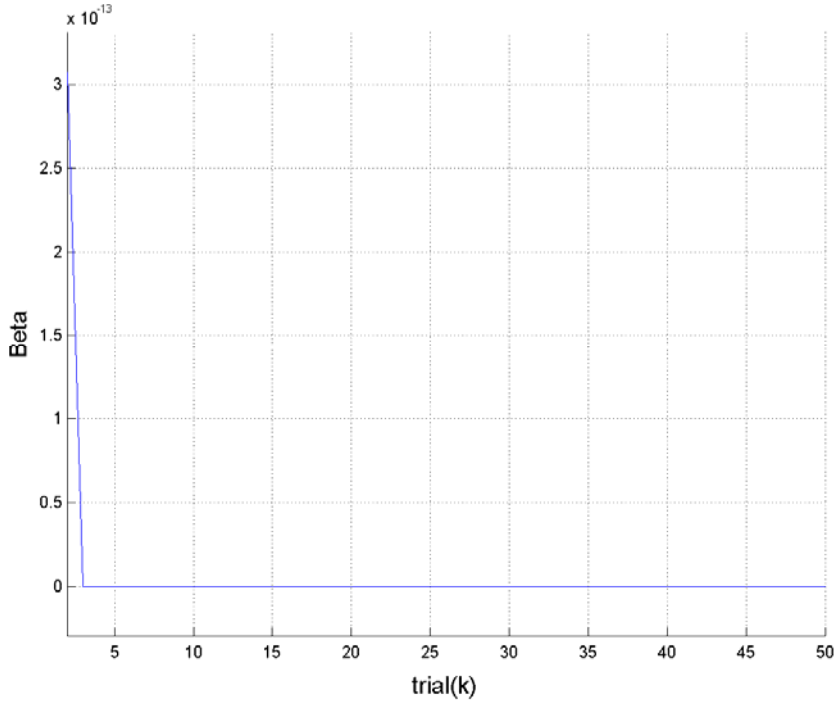

Figure 12: *Beta coefficient evolution of the first order parameter optimal ILC.*

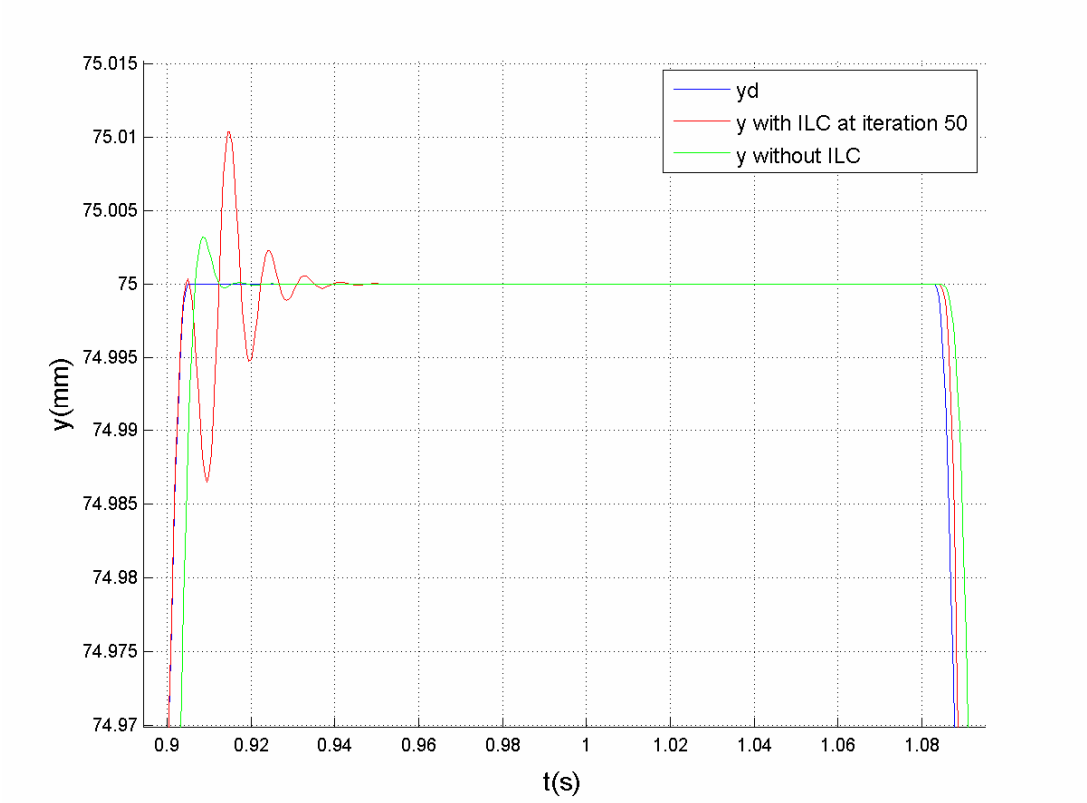

Figure 13: *Simulation response of the system at iteration 50 vs response of the system model without ILC controller.*

Figure 13 might seem to contradict the error evolution in Figure 11 where the overshoot at iteration 50 is much higher than without ILC. But looking at Figure 14 carefully one can see that the error is largely reduced in the intermediate stages. The different behaviour illustrates the importance of convergence to zero error.

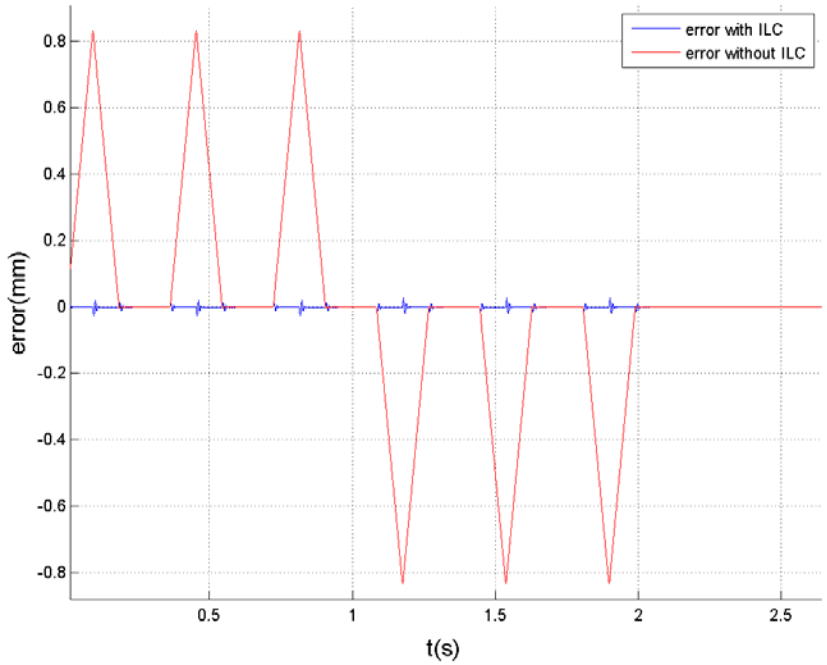

Figure 14: *Time error after iteration 50 and without ILC*

5.2. High order POILC:

In order to measure the influence of the inclusion of previous input and error terms on the POILC algorithm, the system was simulated using the update laws:

$$u_{k+1}(k) = u_k(t) + \sum_{i=1}^{M1} \alpha_{k+1}(i) u_{k-i+1}(t) + \beta_{k+1}(i) e_k(t+1) \tag{47}$$

and

$$u_{k+1}(k) = u_k(t) + \alpha_{k+1} u_k(t) + \sum_{i=1}^{M_2} \beta_{k+1}(i) e_{k-i+1}(t+1) \tag{48}$$

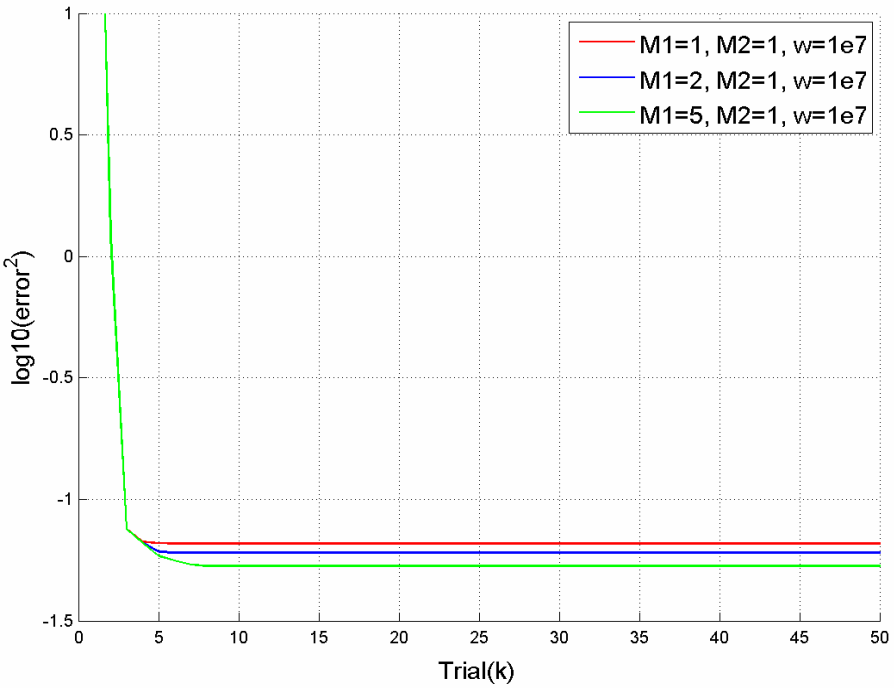

Figure 15: *Error evolution for a different number of previous inputs.* $M_1 = 1, 2, 5$;
$W1 = wI_{1x1}, wI_{2x2}, wI_{5x5}$; $M_2 = 1; W_2 = 1,\ w = 10^3$

Observe that it is consistent that the effect of previous terms in the update law take effect after some iterations because they are set to zero as initial condition.

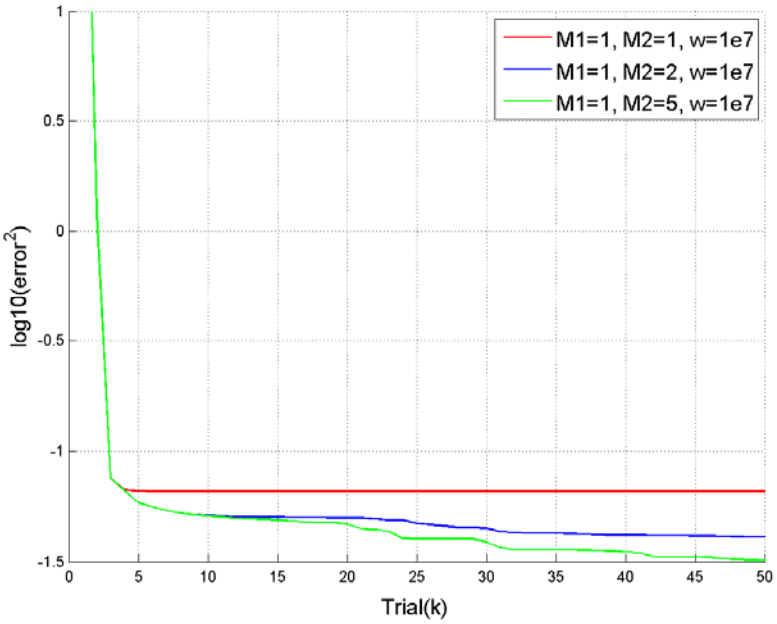

Figure 16: *Error evolution for a different number of previous errors.* $M_2 = 1,2,5$;
$W_2 = wI_{1x1}, wI_{2x2}, wI_{5x5}$; $M_1 = 1; W_1 = 1;\ w = 10^3$

The cluster point of the final error discussion has shown that the inclusion of previous terms does not actually have an effect on the final error, which may appear contradictory with Figure 15 and Figure 16. But this theoretical result has to be interpreted in the limit.

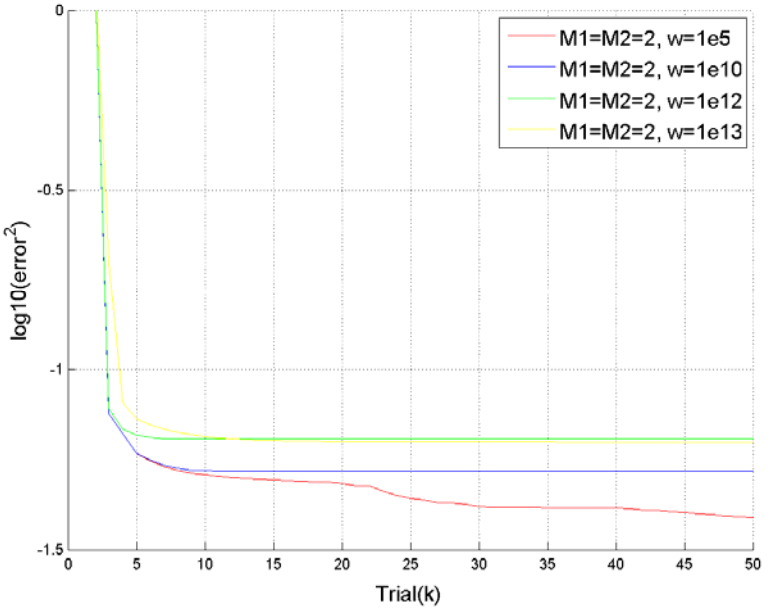

Figure 17: *Error evolution of a high order POILC as a function of weighting factor w.*
$M_1 = M_2 = 2;\ w = 10^5, 10^{10}, 10^{12}, 10^{13}$

With the purpose of testing the influence of the weighting parameter in the case of high order POILC the second order update law is chosen as:

$$u_{k+1}(k) = u_k(t) + \sum_{i=1}^{2} \alpha_{k+1}(i) u_{k-i+1}(t) + \sum_{i=1}^{2} \beta_{k+1}(i) e_{k-i+1}(t+1) \tag{49}$$

and the results are shown in Figure 17, which shows that a larger weighting parameter adds cautiousness and the convergence is reduced.

Influence of basis functions:

In general the duration of the iteration is long so the inclusion of the whole set of basis functions is simply unfeasible. A subset selection of basis functions that keeps the effect on performance can in general be made.

To measure the influence the effect of the basis functions, three subsets of basis functions are chosen based on those proposed in [3].

1) f_1 is selected to be the step response in order to minimize the bias in the tracking error . Then $G_e f$ is the step response of the system and essentially a constant after a transient.

2) Alternatively f_2 chosen to remove the drift from the error by choosing a basis function $f_2 = c \cdot t$. Then $G_e f$ is the ramp response of G_e.

Approaches 1) and 2) do not have any effect on the simulation results compared to the ILC without basis functions, likely since the system has static gain equal to one and thus there's no drift in the error to remove.

3) A subset of the eigenvectors corresponding to negative eigenvalues of $G_e + G_e^T$ can be chosen. Note that because of the fast sampling rate the number of points N is very large requiring large time for the eigenvectors calculation. Even though it is proven that the system converges to zero when all the eigenvectors are chosen, if they were all taken the time between trials would be enormous.

Different subsets of eigenvectors have been tried as basis functions, up to 30 vectors. However there was no improvement in the simulations. Note that just the computation of the eigenvectors takes up several hours in a normal computer.

4) Finally, a subset of sinusoids $f_i = \sin(w_i t + \theta_i)$ and $f_i = \cos(w_i t + \theta_i)$ with the frequencies corresponding to the main Fourier components of the signal y_d and the angle θ to the delay of the system at these frequencies.

Again the simulation results do not show improvements in the convergence.

5.3. Inverse model-based ILC:

Following the procedure defined in (34) the system is spited in two parts

$$G_{+}(z)=\frac{0.0066731\left(z-0.6273\right)\left(z+0.1115\right)\left(z^{2}+0.2918z+0.73\right)}{\left(z-0.5967\right)\left(z^{2}-1.573z+0.6436\right)\left(z^{2}-1.657z+0.7986\right)\left(z^{2}+0.4082z+0.4864\right)\left(z^{2}+0.4889z+0.9561\right)} \tag{50}$$

$$G_{-}(z)=\left(z+1.546\right)\left(z^{2}+1.334z+1.759\right)$$

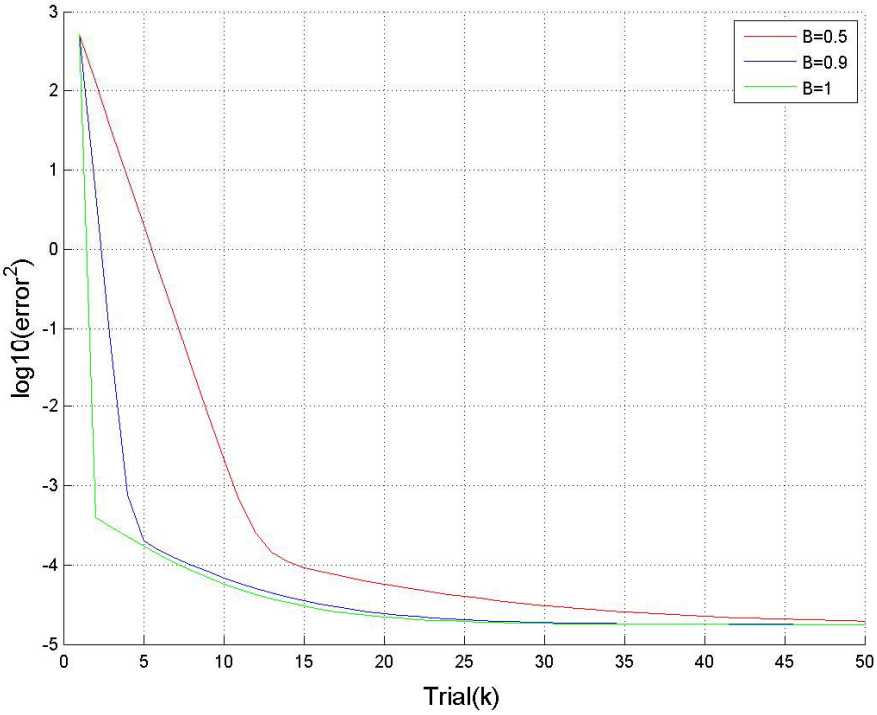

Figure 18: *Error evolution of the system depending on the benchmark parameter B.*

The parameter β affects the convergence rate. The final error remains the same, though. Error evolution is shown in Figure 18. At first glance the result is much better than the obtained with parameter optimal, but we must remember that these simulations do not consider model uncertainty. The reason why despite high accuracy there is not perfect convergence are the stabilization of the non-minimum phase zeros and the initial conditions of the filtering.

5.4. Phase lead compensator:

The parameter η of the phase lead algorithm is chosen $\eta = 0.85$. For this value the frequencies at which the condition (44) is no longer satisfied are shown in Table 1.

Figure 19 is the graphical interpretation of this condition and Figure 20 illustrates the error evolution for the phase lead compensator.

γ	Cut-off frequency (Hz)
1	56.7906
2	66.6989
3	73.7594
4	85.0718
5	104.1538
6	122.5996
7	305.1217
8	268.3153
9	239.7602
10	216.9949

Table 1: *Cut-off frequencies for different* γ *values and* $\eta = 0.85$

With the results of Table 1 the phase lead has the form:

$$L(q) = 0.85q^7 \tag{51}$$

Where γ is selected to maximise the frequency at which condition (44) is satisfied and a low-pass filer is designed as in (45) with a cut-off frequency equal to 300 Hz.

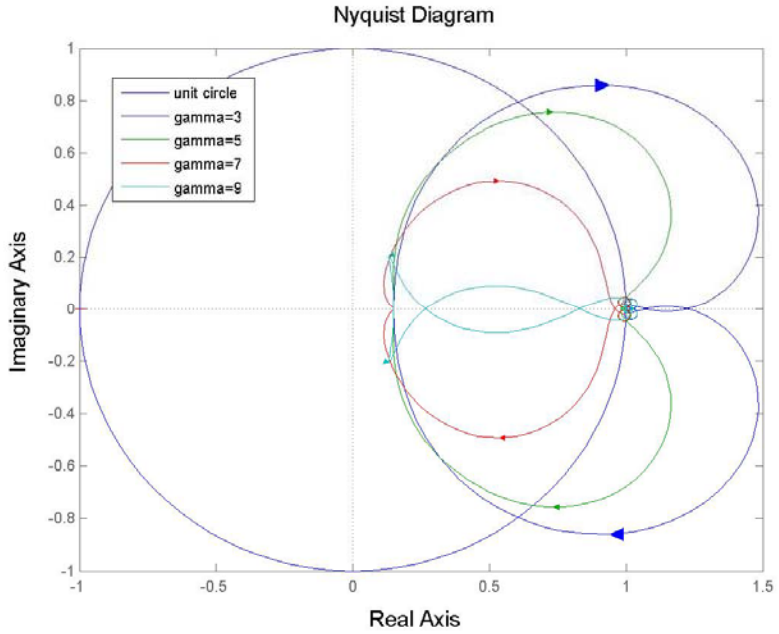

Figure 19: *Nyquist plot of* $1-\eta q^{\gamma}G(q)$ *for different* γ *values. Blue* $\gamma=3$, *Green* $\gamma=5$, *Red* $\gamma=7$, *Cian* $\gamma=9$.

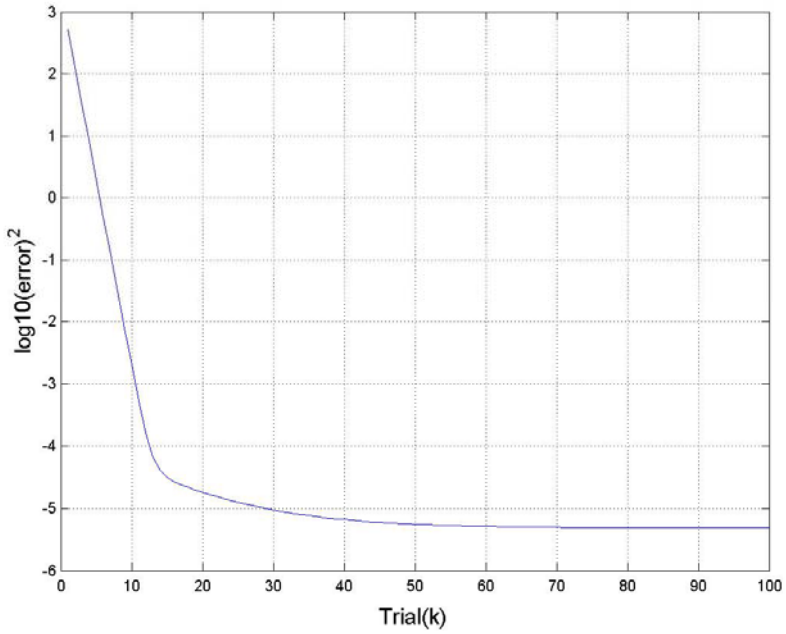

Figure 20: *Error evolution per iteration* $\eta=0.85$, $\gamma=7$ *and* $w_{cutoff}=300Hz$

5.5. Effect of stochastic disturbances

Deterministic and stochastic disturbances have different effects when present in systems controlled with ILC techniques. ILC is in general capable of learning from repeating disturbances. However, when dealing with non-repeating ones the behaviour can be severely degraded. A schema of the stochastic disturbances is shown in Figure 21.

Figure 21*: Schema of the stochastic disturbances.*

Techniques to reduce the effect of non-repeating (stochastic) disturbances will be discussed in further sections. They include:

- Use of a low pass filter Q(q).
- Use of a smaller gain.
- Use of an iteration varying learning operator.

The disturbances for the simulations are taken as normally distributed random sequence with $E\{n_k(t)\} = 0$ and $\sigma_n^2 = 0.003$. In order to estimate the expected value and variance of the error at a specific time each simulation was repeated 50 times. Remark that the error is measured before the noise, i.e. form $z_k(t)$.

Tables 2, 3, 4 and 5 contain the mean and variance error at t=1s and iteration 50 as well as the mean error norm at iteration 50. Time t=1 is chosen because it corresponds to the extreme position of the movement. See Figure 9.

In Figures 22, 23, 24 and 25 the expected value of the norm square error is represented with a solid blue line. Red lines correspond to the standard deviations of the same magnitude.

5.5.1. Stochastic disturbances on POILC

Parameter optimal ILC can be interpreted as a decreasing learning gain algorithm, as shown in c) of Theorem 1, which suggests that it will have a good performance in the presence of stochastic disturbances.

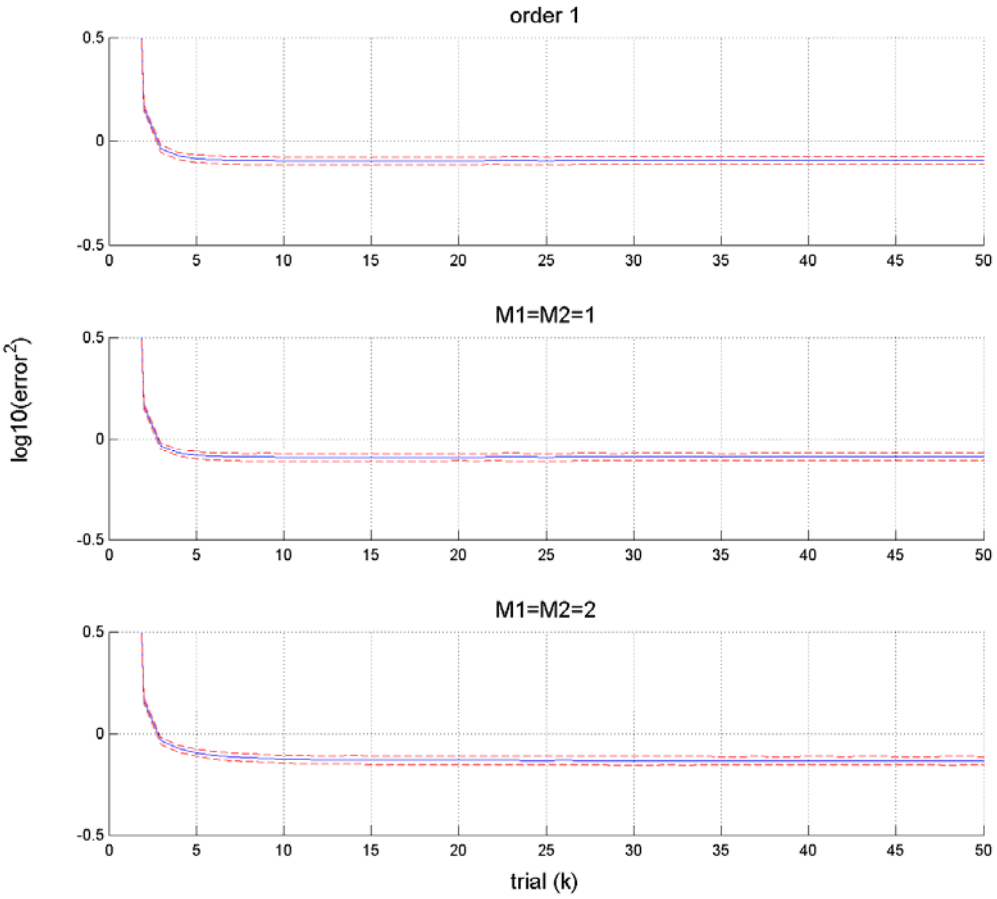

Figure 22: *Error evolution of the first order POILC in presence of non-repeating disturbances.*

The inclusion of high order terms improves very slightly convergence; see Table 2. The variability of the norm square error at each iteration remains almost the same.

Algorithm	Mean error at t=1s and iteration 50	Variance error at t=1s and iteration 50	Expected value of mean square error of iteration 50
POILC first order	-2.8616e-04	7.5780e-05	0.9163
POILC M2=M1=2	-4.2224e-04	6.2561e-05	0.8571
POILC M2=M2=3	-5.4990e-04	6.1007e-05	0.8463

Table 2: *Comparison of the stochastic disturbances effect for different POILC algorithms.*

The fact that variance of the error is reduced, see Table 2, using high order algorithms is consistent with Norrolf and Gunnarsson [9].

5.5.2. Stochastic disturbances on inverse model-based

5.5.2.1. Use of a low pass filter:

A standard technique in order to reduce the effect of stochastic disturbances, usually found at higher frequencies is to apply a low pass filter. Figure 23 shows the simulation results of the inverse ILC with three low-pass filters of Butterworth type and order five. The cut-off frequencies are 50, 100 and 200Hz.

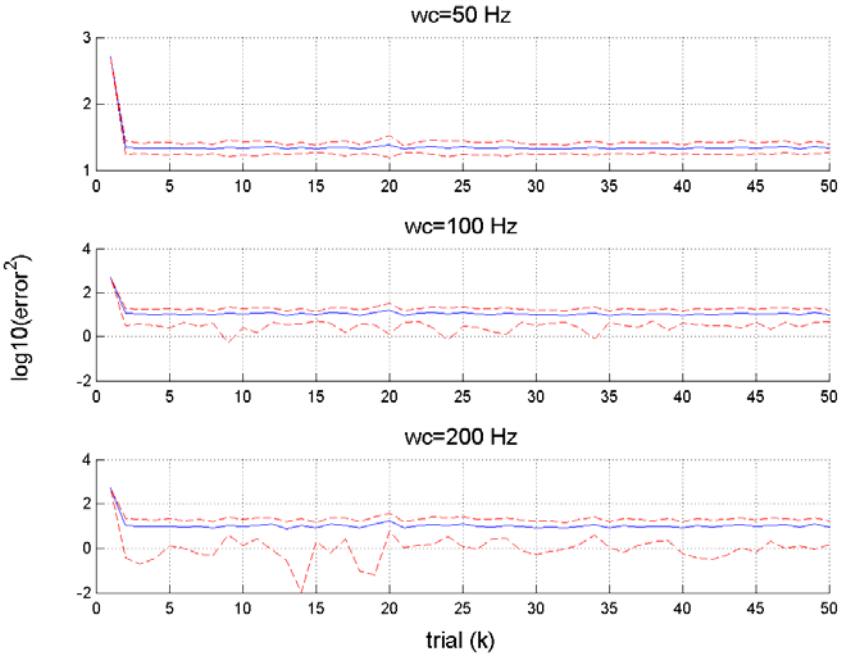

Figure 23: *Error evolution for different cut-off frequencies*

The intuitive idea that low-pass filters cut the learning also matches with the expected value of square error in Table 3. This value decreases with increasing the cut-off frequency as allows learning at frequencies important to y_d. In the same table, the variance error at t=1 and iteration 50 also fit in with the idea that the filter cuts out high frequency disturbances responsible for the variance. It is important to mention that this technique is possible as there is small frequency overlap between the command and the disturbances.

Algorithm	Mean error at t=1s and iteration 50	Variance error at t=1s and iteration 50	Expected value of square error of iteration 50
Model inverse with low-pass filter Cut-off $wc = 50Hz$	-1.4067e-05	2.8410e-05	4.6244
Model inverse with low-pass filter Cut-off $wc = 100Hz$	-0.0017	6.3248e-05	2.9379
Model inverse with low-pass filter Cut-off $wc = 200Hz$	0.0032	1.2793e-04	2.4764

Table 3: *Comparison of the stochastic disturbances effect for different low-pass filters.*

5.5.2.2. Use of a smaller gain:

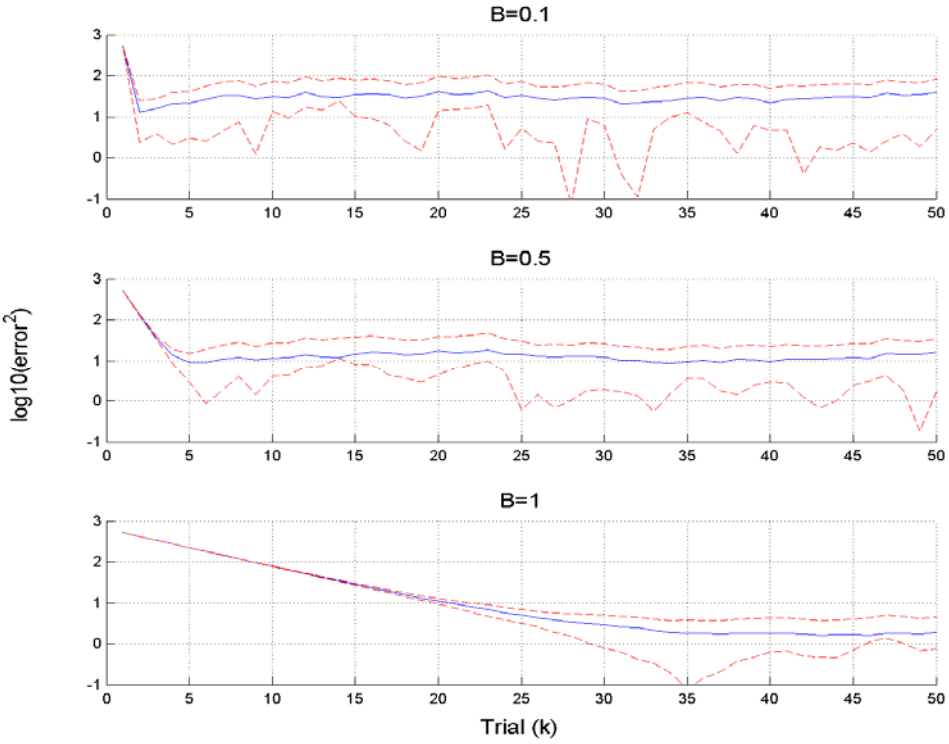

Figure 24: *Comparison of the error evolution depending on the gain*

As suggested in [5] the use of a smaller gain reduces the variance and the norm square error but entails slower convergence rate. Figure 24 shows this effect.

Algorithm	Mean error at t=1s and iteration 50	Variance error at t=1s and iteration 50	Expected value of square error of iteration 50
Model inverse Gain B=1	-7.6338e-04	7.2104e-04	3.4448
Model inverse Gain B=0.5	4.3923e-04	2.3666e-04	2.7829
Model inverse Gain B=0.1	4.7115e-04	2.4050e-05	1.0967

Table 4: *Comparison of the stochastic disturbances effect for different gains.*

5.5.3. Stochastic disturbances on phase lead:

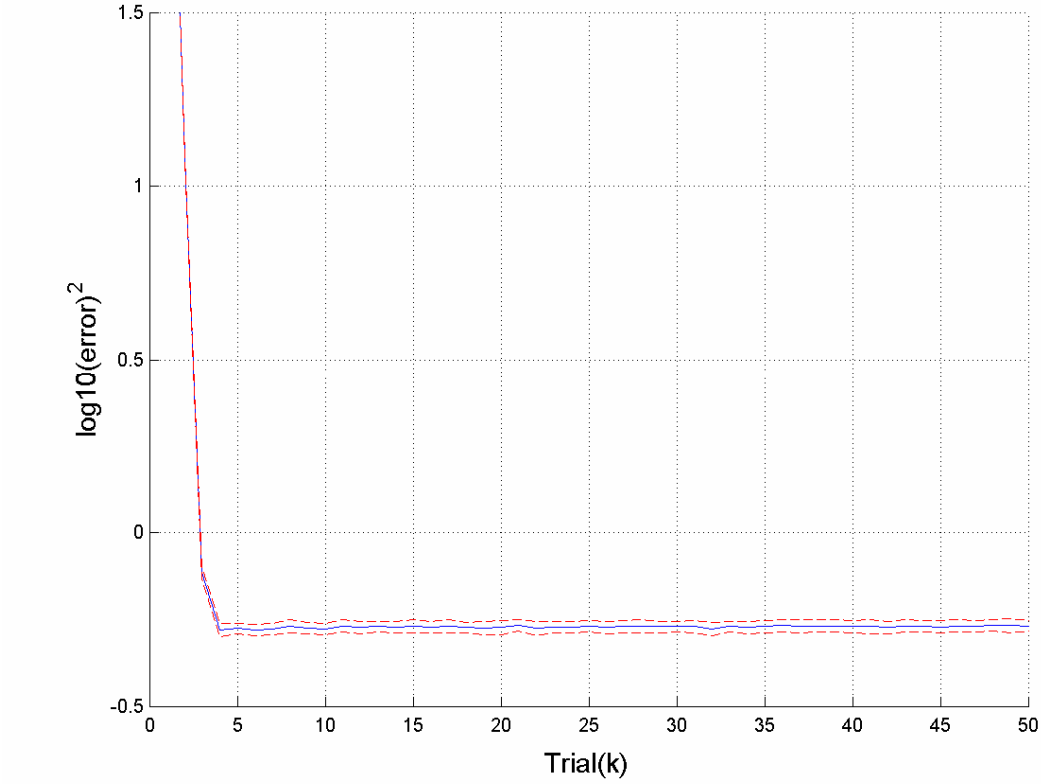

Figure 25: *Error evolution of phase lead ILC in presence of stochastic disturbances.*

Phase lead simulations are shown in Figure 25. Not just the variability is small, but also the convergence is fast and the final error small.

Algorithm	Mean error at t=1s and iteration 50	Variance error at t=1s and iteration 50	Expected value of mean square error of iteration 50
Phase lead $\gamma = 7$ Cut-off $wc = 300Hz$	0.0015	8.2621e-05	0.8334

Table 5: *Stochastic disturbances effect for the phase lead algorithm.*

5.6. Effect of model uncertainty:

In order to test the model uncertainty a new model was obtained varying the coefficients of the model inside the region defined by the standard deviations of the parameters in the identification. In other words, the coefficients of the new model are randomly chosen with a maximum variation equal to their standard deviation.

The model variation is:

$$H_1(z) = \frac{0.0078846\,(z+1.545)\,(z-0.6258)\,(z+0.1122)\,(z^2 - 1.17z + 0.5887)\,(z^2 + 0.2956z + 0.728)\,(z^2 + 1.331z + 1.765)}{(z-0.5565)\,(z+0.3158)\,(z^2 - 1.842z + 0.9062)\,(z^2 - 1.439z + 0.6082)\,(z^2 + 0.4078z + 0.4889)\,(z^2 + 0.4842z + 0.947)} \quad (52)$$

Figure 26 shows the superposition of the bode diagram of the variation and the original model.

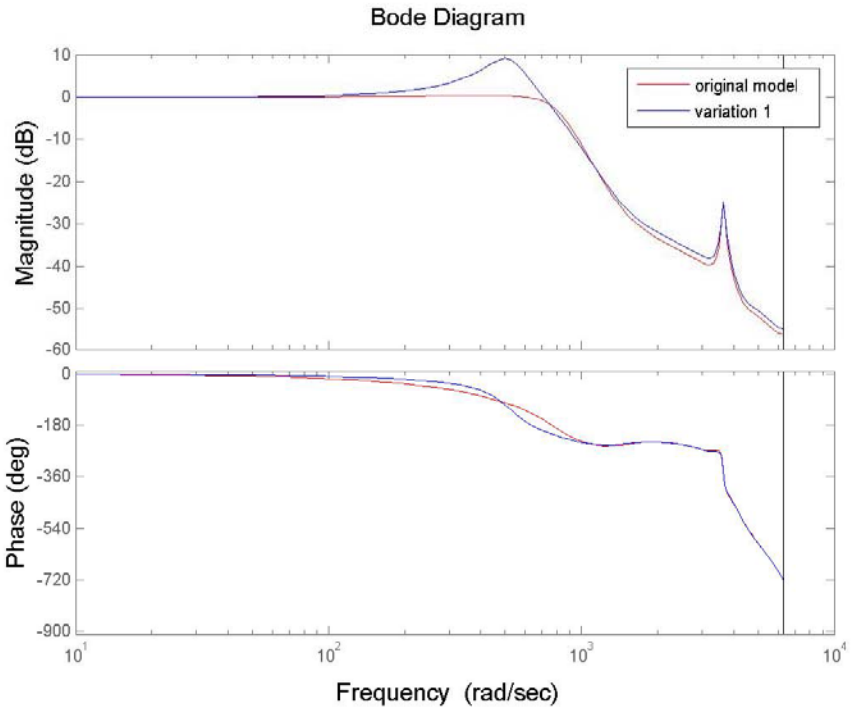

Figure 26: *Bode diagram comparison of the original model and variation_1*

5.6.1. Effect of model uncertainty on POILC:

The parameter optimal ILC is redefined using the variation model and then simulated employing the original model. Specifically 3 high order ILC algorithms are simulated. The error evolution, see Figure 27, is similar both in convergence rate and final error to the result obtained in the simulation with the original model.

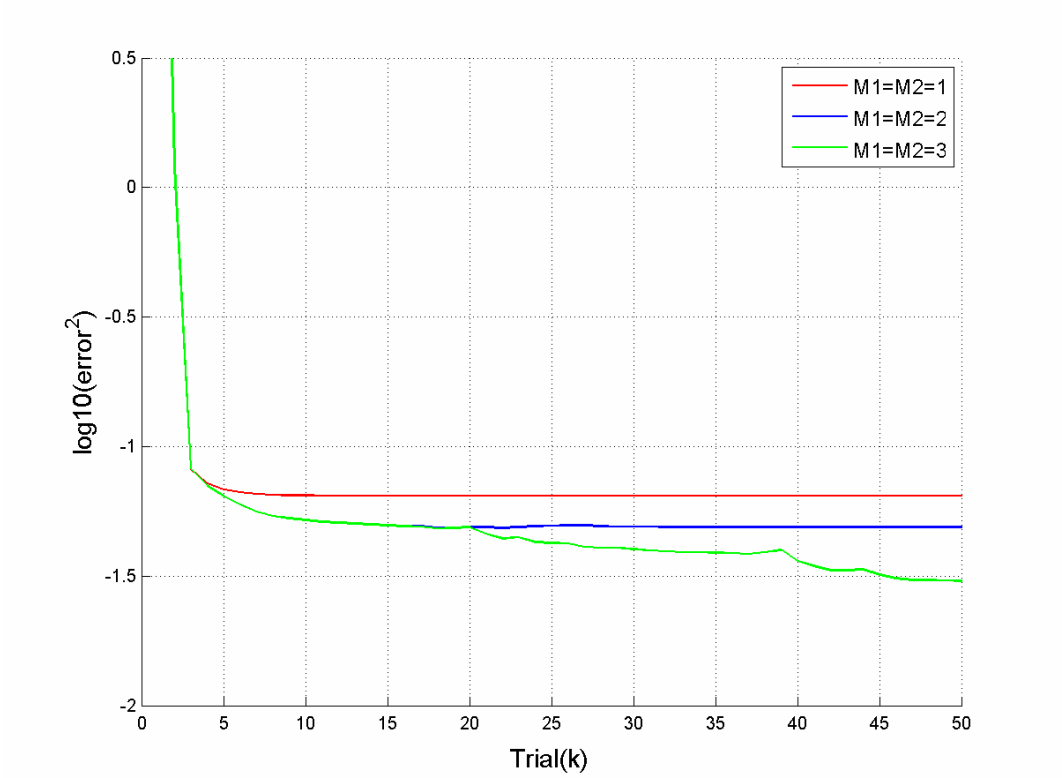

Figure 27: Error evolution of the model variation simulated with the original model.

Simulations show monotonic convergence which is encouraging, though there is no final proof for monotonic convergence with model uncertainty.

5.6.2. Effect of model uncertainty on inverse model-based ILC:

The variation model keeps the non-minimum phase zeros. Again the system is separated in two parts, $H_1^+(z)$ containing the minimum phase zeros and $H_1^-(z)$ containing the minimum phase zeros. The stable filtering presented in (35) and (36) is applied.

$$H_1^+(z) = \frac{0.0078846\,(z-0.6258)\,(z+0.1122)\,(z^2 - 1.17z + 0.5887)\,(z^2 + 0.2956z + 0.728)}{(z-0.5565)\,(z+0.3158)\,(z^2 - 1.842z + 0.9062)\,(z^2 - 1.439z + 0.6082)\,(z^2 + 0.4078z + 0.4889)\,(z^2 + 0.4842z + 0.947)} \quad (53)$$

$$H_1^-(z) = (z+1.545)\,(z\text{^}2 + 1.331z + 1.765)$$

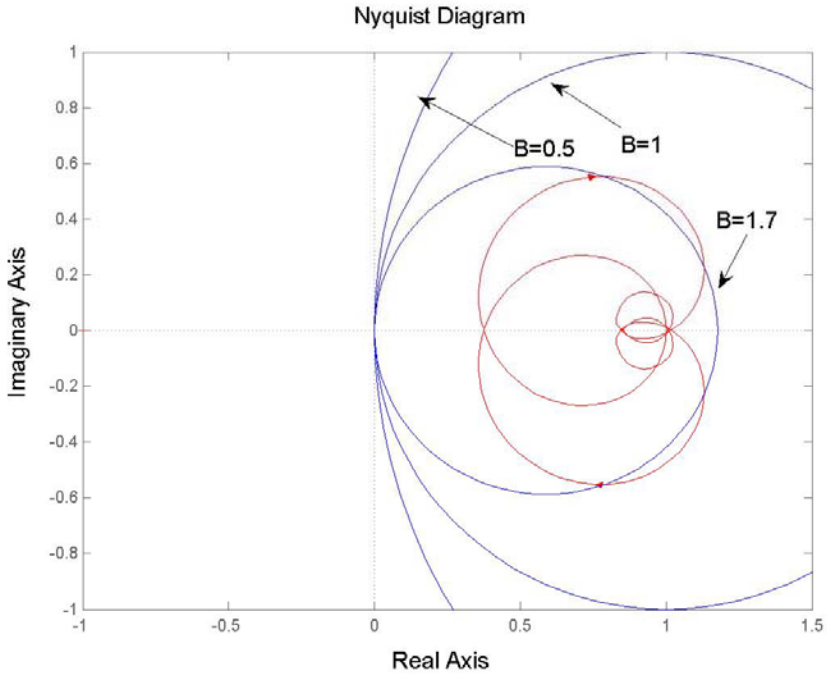

Figure 28: Nyquist plot of the uncertainty $U(z)$

Model uncertainty $U(z)$, understood as the difference between the original model and the variation, is plotted in Figure 28. For $\beta = 1.7$ the uncertainty is outside its corresponding circle and it will diverge. This theoretical result is supported by the simulation result in Figure 29.

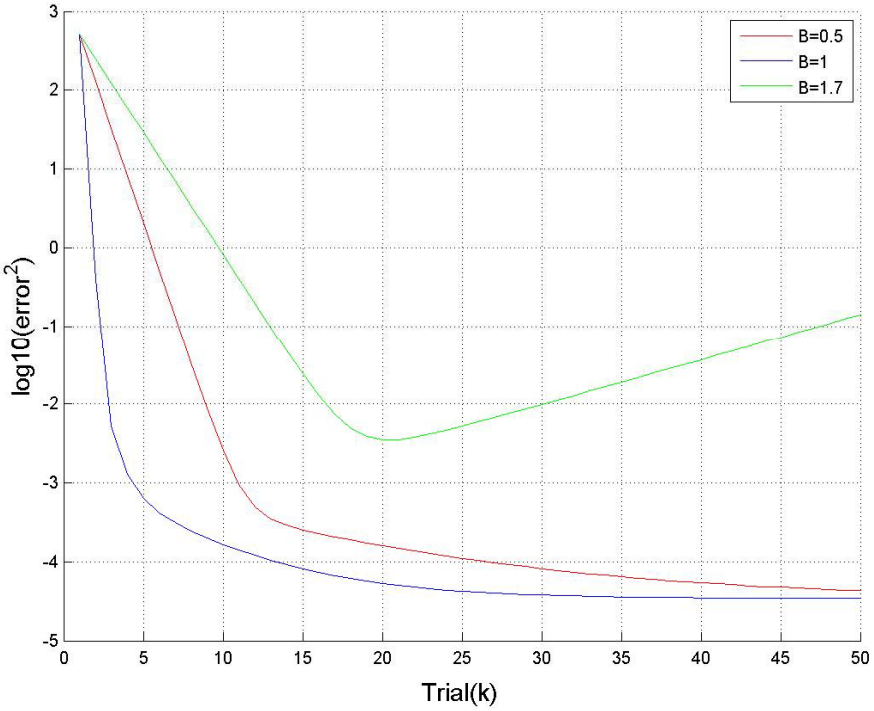

Figure 29: *Error evolution for different values of gain*

5.6.3. Model-based inverse ILC with POILC:

Figure 30 shows the error evolution in the iteration domain of this algorithm. Results are very similar to those obtained with the fixed gain $\beta = 1$.

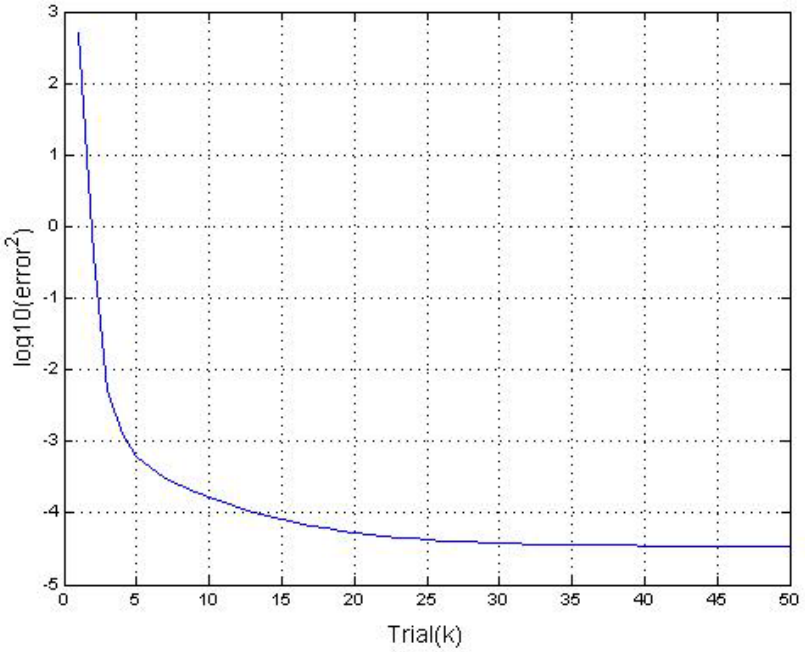

Figure 30: *Error evolution for different values of gain*

In Figure 31 the gain is plotted. As predicted β decreases as iterations go on making the algorithm more robust.

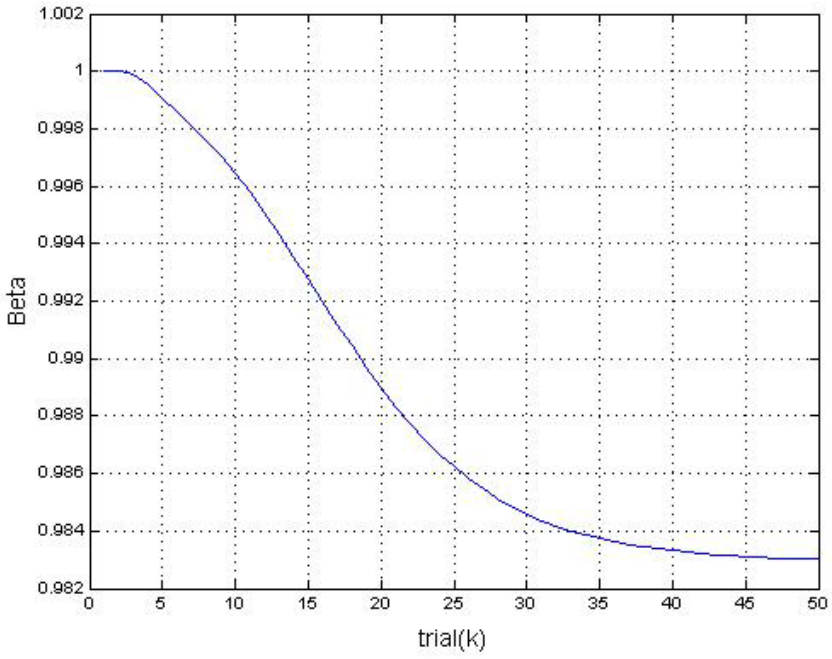

Figure 31: *Beta evolution of the inverse model based POILC.*

5.6.4. Phase lead:

The parameters of the phase lead are redefined with the model variation and then simulated using the original model. Keeping the gain $\eta = 0.85$, the new frequencies are:

γ	Cut-off frequency (Hz)
1	56.4944
2	59.7982
3	62.5680
4	64.9386
5	67.0056
6	68.8380
7	68.8380
8	67.0056
9	62.5680
10	56.4944

Table 6: *Cut-off frequencies for different γ values and $\eta = 0.85$ and model variation.*

The new phase lead coincides in the parameters γ and η. The new cut-off frequency is much smaller, around 68 Hz. It is obvious that if the only thing that changes is the cut-off frequency, which becomes lower, the system will still converge but to a higher error. The error evolution in the iteration domain of the simulation is shown in Figure 32.

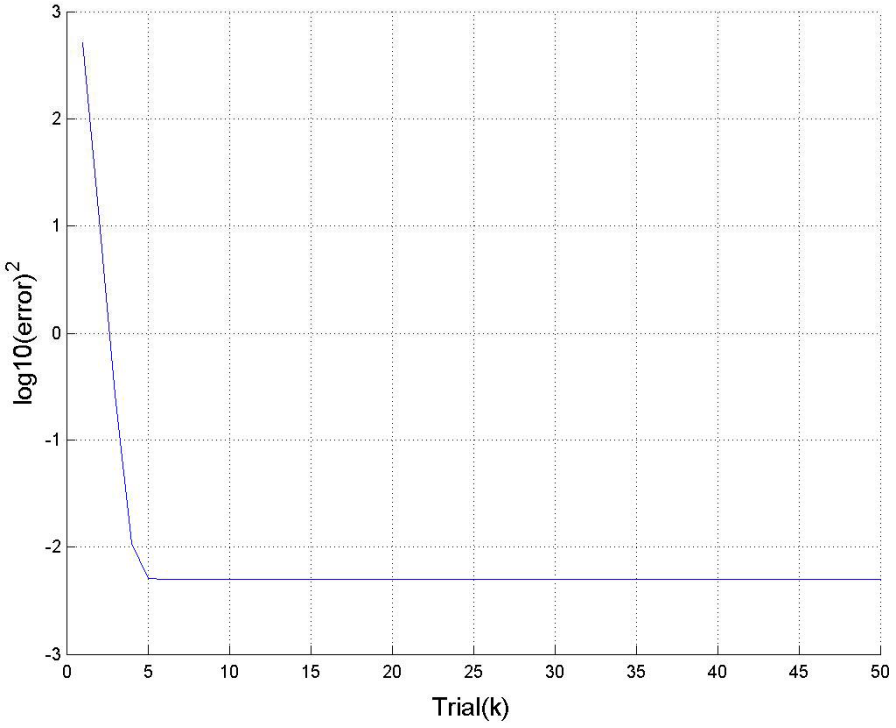

Figure 32: *Error evolution of the phase lead algorithm when*

6. Application Results:

6.1. Parameter optimal ILC:

The experimental results are similar to those obtained in the simulation. Even though the error reduction is significant it is not homogeneous and for the specific positioning application not accurate enough.

The inclusion of high order terms to the algorithm has a slight effect on convergence performance, but increases considerably the calculation time.

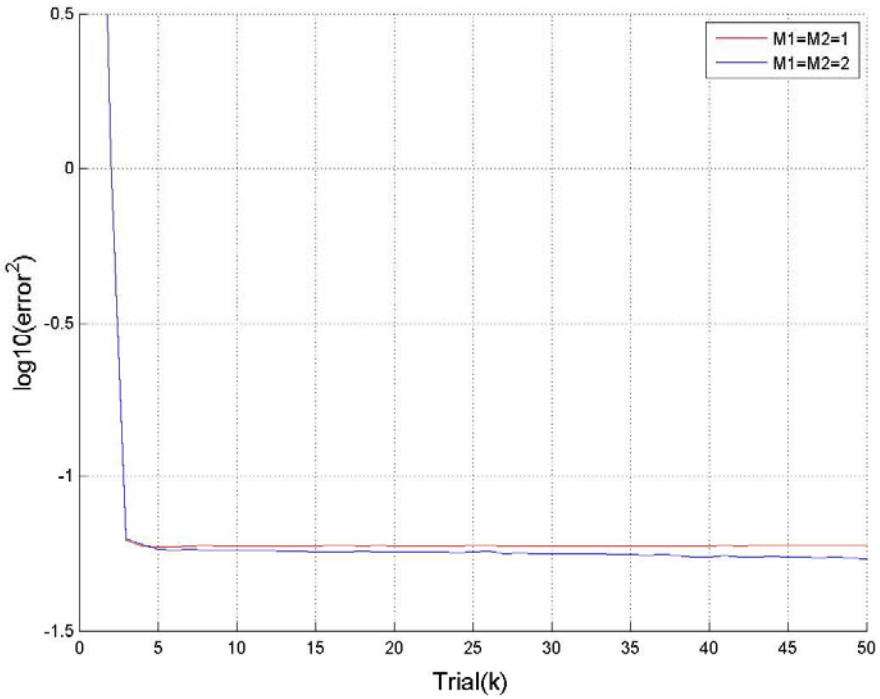

Figure 33: *Error evolution of parameter optimal ILC with a zero-phase filter.*

Figure 33 shows the error evolution in the iteration domain of two high order parameter optimal algorithms. The number of parameters are $M_1 = M_2 = 1$ and $M_1 = M_2 = 2$. The weighting parameter is w=1e7. W_1 and W_2 are identity matrices.

Results in Figure 34, from the first of the high order algorithms, are very similar to the equivalent in simulation Figure 13. In both cases, even though the error is reduced considerably the real positioning application gets worse performance.

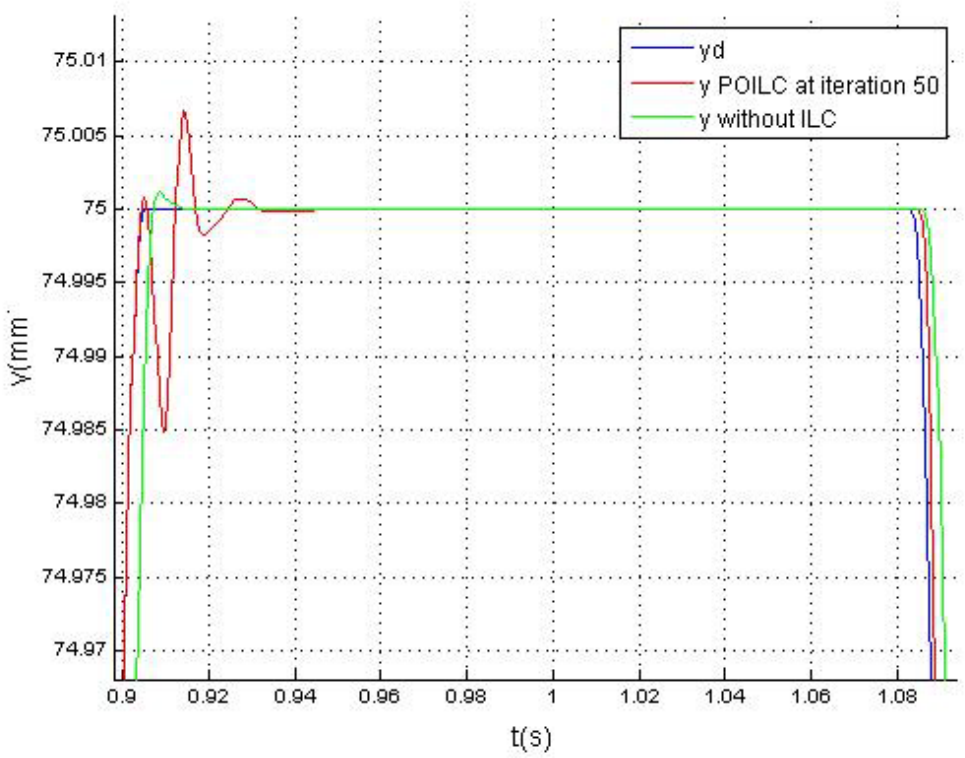

Figure 34: Error at iteration 50 for the parameter optimal ILC command

Figure 35 shows the error in the time domain of the system with the first algorithm considered ($M_1 = M_2 = 1$).

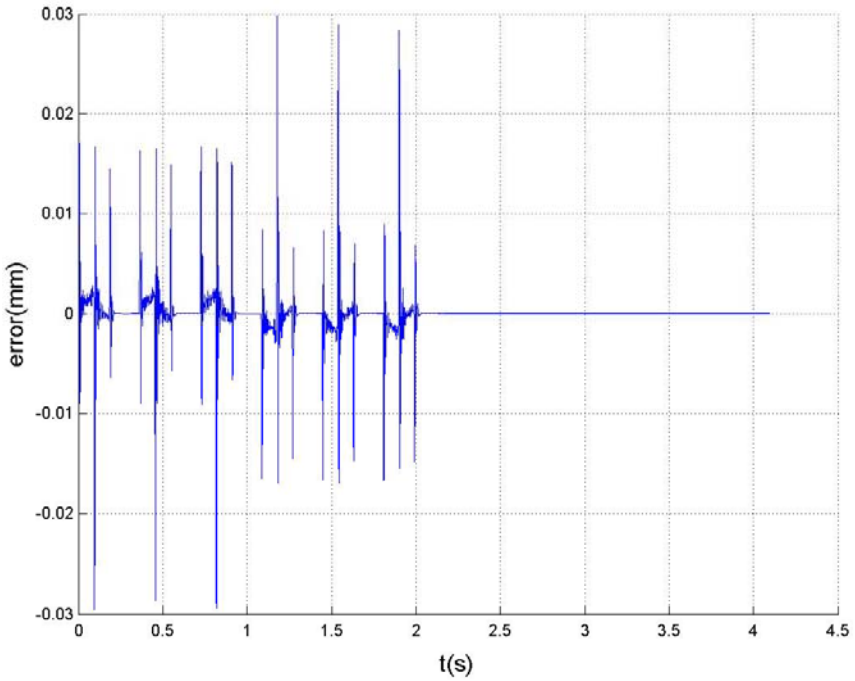

Figure 35: *System response at iteration 1 and iteration 50*

6.2. Inverse Model-based:

The application of the inverse model ILC algorithm entails facing power protection errors since the input contains high frequency components. As it is not obvious were these components come from, a simple solution is to apply a low-pass filter to cut them off at 50 Hz. Besides, to minimise the possible effect of non-repetitive disturbances a smaller gain has been chosen as $\beta = 0.5$.

However this procedure alters the learning and convergence is even lower than with the parameter optimal.

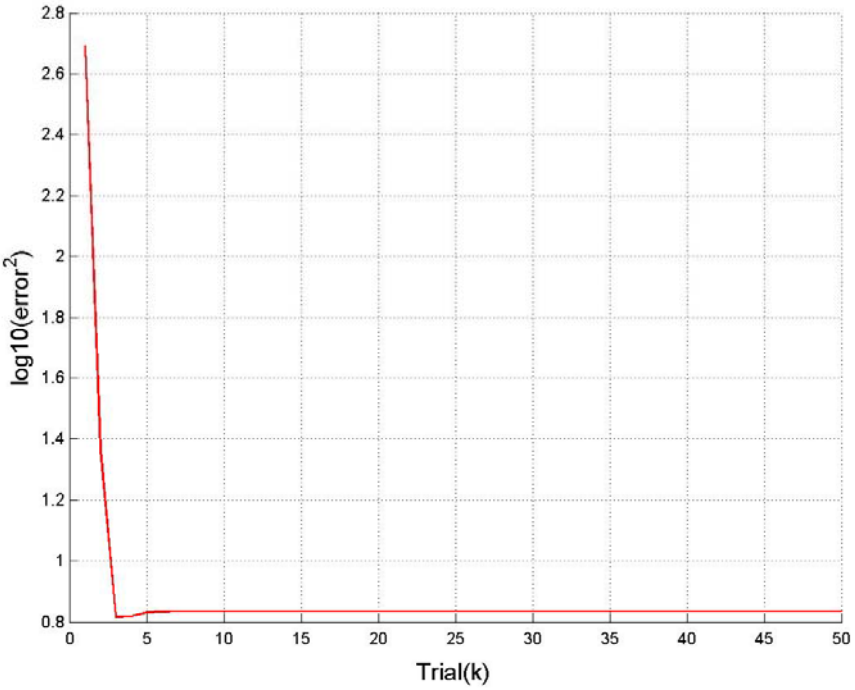

Figure 36: *Error evolution of model inverse ILC.*

Figure 38 shows the response of the inverse ILC and the system without ILC. The overshoot is much bigger than without the ILC technique making it not possible to apply it on a real application purpose. The level of convergence is similar to the one obtained in the stochastic-disturbances simulation. See Figure 23 and Figure 24. The difference between the disturbances in the simulation and the real experiment are unknown, though.

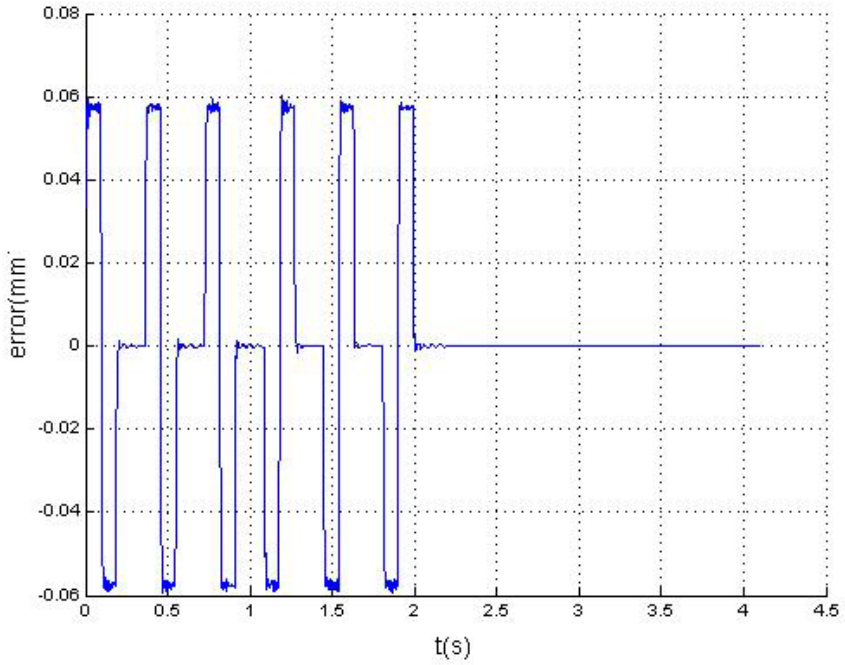

Figure 37: *Error at iteration 50 model inverse ILC command.*

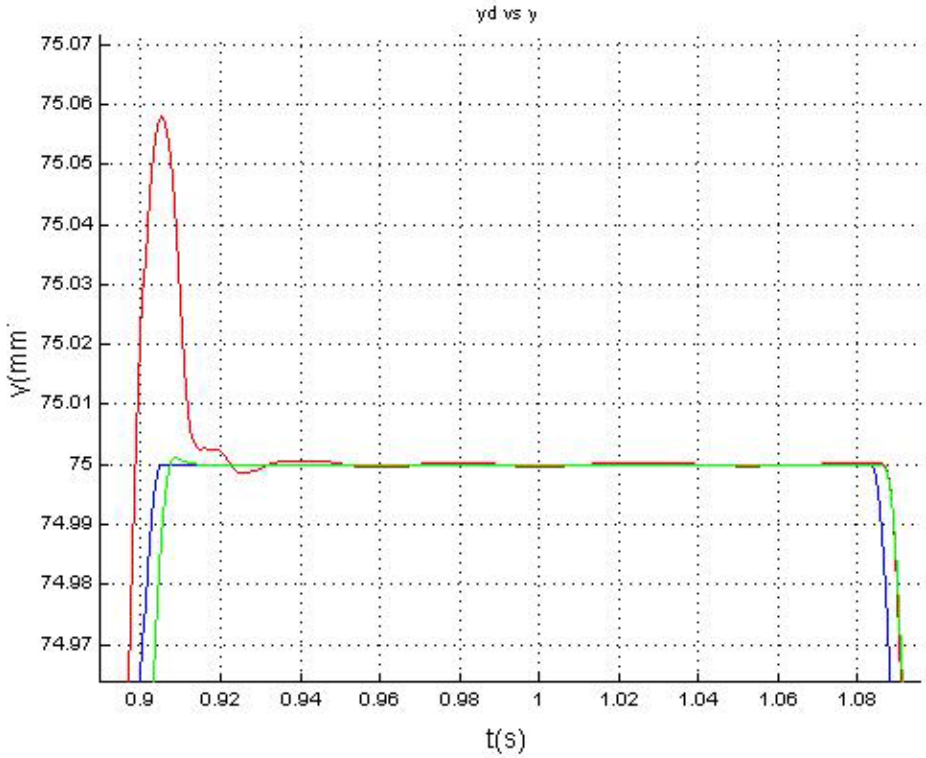

Figure 38: *Model inverse system response at iteration 50.*

6.3. Phase Lead:

A phase lead controller was applied to the system. The 50 iterations were performed giving a fast convergence and very small error. In Figure 39 the error evolution in the iteration domain is shown giving a much better convergence than either in the parameter optimal or the inverse model ILC.

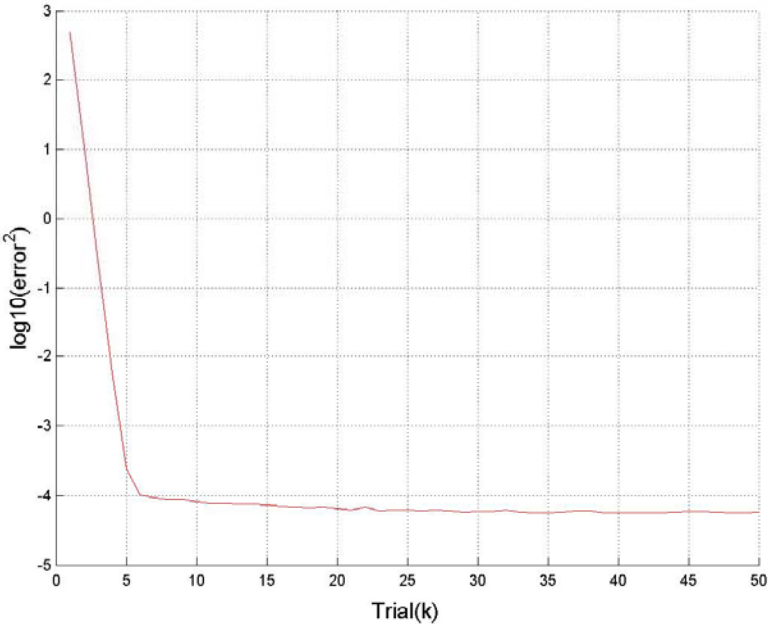

Figure 39: *Error evolution of phase lead ILC.*

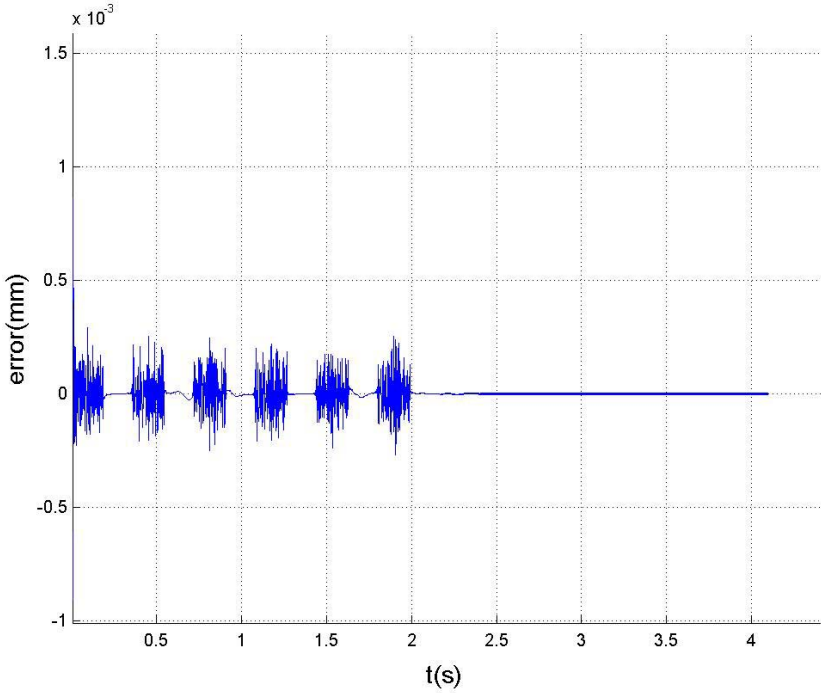

Figure 40: *Error at iteration 50 for phase lead ILC command.*

Figure 41 shows the output at iteration 50. Error is considerably smaller than the initial, i.e. without ILC.

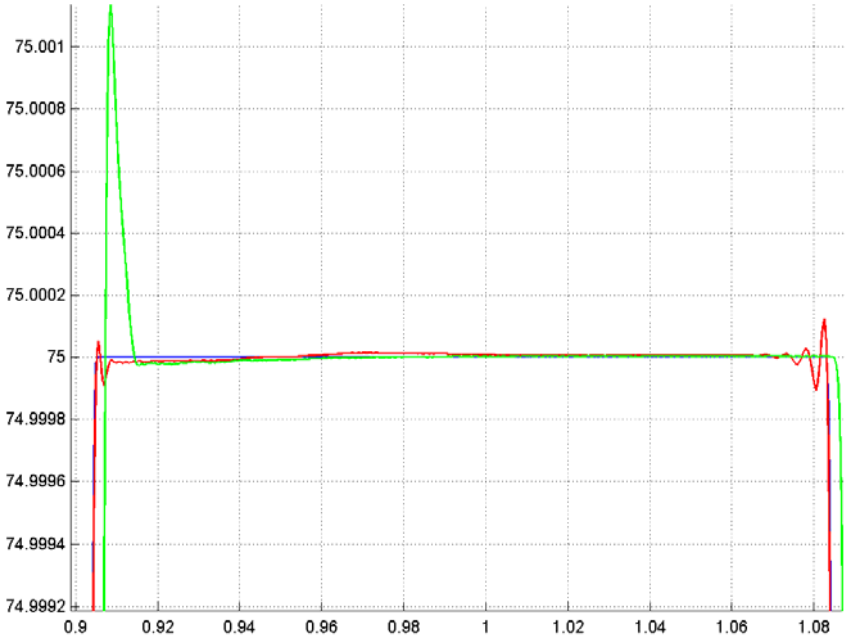

Figure 41: *Phase lead system response at iteration 50.*

7. Conclusions:

Simulations have shown that parameter optimal ILC algorithms respond well to non-repeating disturbances and model uncertainty. On the other hand there is no framework that relates convergence and model uncertainty for this particular case. Moreover, parameter optimal ILC requires a considerable amount of memory for the system representation storage. The order of the algorithm increases calculation time, too.

Although error reduction has been successfully achieved for the linear motor with the parameter optimal ILC algorithm, there was no convergence to zero error and accurate homogeneous positioning was not obtained. The results with this algorithm are not satisfactory since the step type positioning for an industrial application is concerned with the performance on the flat stages rather than on the intermediate trajectory. Attention must be paid to possible bad performance on these regions hidden by globally decreasing error.

Although basis functions guarantee convergence to zero error, for systems working at high sampling frequencies where even short trajectories might have thousands of points, the inclusion of a whole set of basis functions is just unfeasible. Subset selection is not so straight forward and very application dependent. Furthermore, in feedback controlled systems like this one, common subset selection such as step and ramp do not have any effect on the error.

Whereas the Inverse model-based ILC worked in simulation, the actual application on the motor presented difficulties related to power protection. Simulations suggest that inverse model is vulnerable to non-repeating disturbances, but if a trade-off between convergence rate and final error is accepted, a small gain reduces final error.

An iteration varying gain was applied in simulation using a parameter optimal approach giving similar results to the fixed gain version of the inverse algorithm.

The identified discrete model presented non-minimum phase zeros. Thus non causal filtering was satisfactorily applied.

Phase lead has been proven as the best approach in practice. It is simple to implement and robust to non repeating disturbances. But more importantly the level of error convergence is much higher than in the other cases.

Future work might include identification at different positions and model uncertainty determination as well as theoretical results that link parameter optimal ILC to model uncertainty. It would be interesting to investigate the effect of parameter optimal over-lifting the system. That is to say, to forward shift the data as in the phase lead case.

8. References:

[1] D.H.OWENS and K. FENG “Parameter optimization in iterative learning control”, Int. J. CONTROL, 2003, VOL. 76, NO. 11, 1059-1069.

[2] D.H. OWENS, J. HÄTÖNEN “Iterative Learning Control – An optimization Paradigm”, Annual Reviews in Control 29 (2005) 57 - 70

[3] J. HÄTÖNEN, D.H.OWENS and K. FENG “Basis functions and parameter optimisation in high-order iterative learning control”. Automatica, 42, 287-294, 2006.

[4] DOUGLAS A. BRISTOW, MARINA THARAYIL AND ANDREW G. ALLEYNE, “A Survey of iterative learning control. A learning-based method for high-performance tracking control”. IEEE Control systems magazine, June 2006.

[5] T.J. HARTE, J. HÄTONEN and D.H.OWENS, “Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness”. International Journal of Control Vol. 78, No. 8, 20 May 2005, 577-586.

[6] M. BUTCHER, A. KARIMI and R. LONGCHAMP. “A statistical analysis of certain iterative learning control algorithms”. To appear in the international journal of control.

[7] R. LONGCHAMP, “Commande numérique de systèmes dynamiques”. Presses Polytechniques et universitaires romandes. 2006

[8] R.W. LONGMAN. “Iterative learning control and repetitive control for engineering practice”. International Journal of Control, 73(10) : 930–954, 2000.

[9] S. GUNNARSSON, M. NORROLF “On the disturbance properties of high order iterative learning control algorithms”. Automatica, 42 (2006) 2031 – 2034

Acknowledgements:

Thanks to the Automatic Control Laboratory for this project and especially to Mark Butcher for his unconditional help and advice.