Estimation-based Norm-optimal Iterative Learning Control

Patrik Axelsson, Rickard Karlsson and Mikael Norrlöf

Original Publication:

Patrik Axelsson, Rickard Karlsson and Mikael Norrlöf, Estimation-based Norm-optimal Iterative Learning Control, 2014, Systems & control letters (Print), (73), 76-80.
http://dx.doi.org/10.1016/j.sysconle.2014.08.007

Estimation-based Norm-optimal Iterative Learning Control

Patrik Axelsson[a,*], Rickard Karlsson[b,a], Mikael Norrlöf[c,a]

[a]*Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden*
[b]*Nira Dynamics, Teknikringen 6, SE-583 30 Linköping, Sweden*
[c]*ABB AB – Robotics SE-721 68 Västerås, Sweden*

Abstract

The norm-optimal iterative learning control (ILC) algorithm for linear systems is extended to an estimation-based norm-optimal ILC algorithm where the controlled variables are not directly available as measurements. A separation lemma is presented, stating that if a stationary Kalman filter is used for linear time-invariant systems then the ILC design is independent of the dynamics in the Kalman filter. Furthermore, the objective function in the optimisation problem is modified to incorporate the full probability density function of the error. Utilising the Kullback-Leibler divergence leads to an automatic and intuitive way of tuning the ILC algorithm. Finally, the concept is extended to non-linear state space models using linearisation techniques, where it is assumed that the full state vector is estimated and used in the ILC algorithm. Stability and convergence properties for the proposed scheme are also derived.

Keywords: Iterative Learning Control, Estimation, Filtering, Non-linear systems

1. Introduction

The *iterative learning control* (ILC) method (Arimoto et al., 1984; Moore, 1993) improves performance, for instance trajectory tracking accuracy, for systems that repeat the same task several times. ILC for non-linear systems has been considered in e.g. Avrachenkov (1998); Lin et al. (2006); Xiong and Zhang (2004), where the ILC algorithm is formulated as the solution to a non-linear system of equations. Traditionally, a successful ILC control law is based on direct measurements of the control quantity. However, when the control quantity is not directly available as a measurement, the controller must estimate the control quantity from other measurements, or rely on measurements that indirectly relate to this quantity.

ILC in combination with estimation of the control quantity, has not been given much attention in the literature. In Wallén et al. (2009) it is shown that the performance of an industrial robot is significantly increased when an estimate of the control quantity is used instead of measurements of a related quantity. Performance of the ILC algorithm when combined with an estimator has previously been addressed in Axelsson et al. (2013). A related topic has been covered in Ahn et al. (2006); Lee and Lee (1998), where a state space model in the iteration domain is formulated for the error signal, and a KF is used for estimation. The difference to this paper is that in Ahn et al. (2006); Lee and Lee (1998) it is assumed that the control error is measured directly, hence the KF is merely a low-pass filter, with smoothing properties, for reducing the measurement noise.

Here, the estimation-based ILC framework, where the control quantity is not directly available as a measurement, is combined with an ILC design based on an optimisation approach, referred to as norm-optimal ILC (Amann et al., 1996). The estimation problem is formulated using recursive Bayesian methods. Extensions to non-linear systems, utilising linearisation techniques, are also presented. The contributions are summarised as

1. A separation lemma, stating that the extra dynamics introduced by the stationary KF is not necessary to include in the design of the ILC algorithm.
2. Extension of the objective function to include the full *probability density function* (PDF) of the estimated control quantity, utilising the Kullback-Leibler divergence. This provides an automatic and intuitive choice for one of the weights in the norm-optimal ILC algorithm.
3. Extensions to non-linear systems, including stability and convergence properties.

2. Iterative Learning Control (ILC)

The ILC-method improves the performance of systems that repeat the same task multiple times. The systems can be open loop as well as closed loop, with internal feedback. The ILC control signal $\mathbf{u}_{k+1}(t) \in \mathbb{R}^{n_u}$ for the next iteration $k+1$ at discrete time t is calculated using the error signal and the ILC control signal, both from the current iteration k. Different types of update algorithms can be found in e.g. Moore (1993).

*Corresponding author. *E-mail address*: patrik.axelsson@liu.se, *Phone*: +46 13 28 13 65, *Fax*: +46 13 13 92 82

Email addresses: patrik.axelsson@liu.se (Patrik Axelsson), rickard.karlsson@niradynamics.se (Rickard Karlsson), mikael.norrlof@se.abb.com (Mikael Norrlöf)

One design method is the *norm-optimal* ILC algorithm (Amann et al., 1996; Gunnarsson and Norrlöf, 2001). The method solves

$$\begin{aligned} \underset{\mathbf{u}_{k+1}(\cdot)}{\text{minimise}} \quad & \sum_{t=0}^{N-1} \|\mathbf{e}_{k+1}(t)\|^2_{\mathbf{W_e}} + \|\mathbf{u}_{k+1}(t)\|^2_{\mathbf{W_u}} \\ \text{subject to} \quad & \sum_{t=0}^{N-1} \|\mathbf{u}_{k+1}(t) - \mathbf{u}_k(t)\|^2 \leq \delta, \end{aligned} \tag{1}$$

where $\mathbf{e}_{k+1}(t) = \mathbf{r}(t) - \mathbf{z}_{k+1}(t)$ is the error, $\mathbf{r}(t)$ the reference signal, and $\mathbf{z}_{k+1}(t)$ the controlled quantity. The matrices $\mathbf{W}_e \in \mathbb{S}^{n_z}_{++}$, and $\mathbf{W}_u \in \mathbb{S}^{n_u}_{++}$ are weight matrices, used as design parameters, for the error and the ILC control signal, respectively[1].

Using a Lagrange multiplier and a batch formulation (see Appendix A) of the system from $\mathbf{u}_{k+1}(t)$ and $\mathbf{r}(t)$ to $\mathbf{z}_{k+1}(t)$ gives the solution

$$\bar{\mathbf{u}}_{k+1} = \boldsymbol{\mathcal{Q}} \cdot (\bar{\mathbf{u}}_k + \boldsymbol{\mathcal{L}} \cdot \bar{\mathbf{e}}_k) \tag{2a}$$

$$\boldsymbol{\mathcal{Q}} = (\mathbf{T}_{\mathbf{zu}}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_{\mathbf{zu}} + \boldsymbol{\mathcal{W}}_\mathbf{u} + \lambda \mathbf{I})^{-1} (\lambda \mathbf{I} + \mathbf{T}_{\mathbf{zu}}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_{\mathbf{zu}}) \tag{2b}$$

$$\boldsymbol{\mathcal{L}} = (\lambda \mathbf{I} + \mathbf{T}_{\mathbf{zu}}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_{\mathbf{zu}})^{-1} \mathbf{T}_{\mathbf{zu}}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e}, \tag{2c}$$

where λ is a design parameter and

$$\boldsymbol{\mathcal{W}}_\mathbf{e} = \mathbf{I}_N \otimes \mathbf{W}_e \in \mathbb{S}^{Nn_z}_{++}, \quad \boldsymbol{\mathcal{W}}_\mathbf{u} = \mathbf{I}_N \otimes \mathbf{W}_u \in \mathbb{S}^{Nn_u}_{++}, \tag{3}$$

$\mathbf{I}_N$ is the $N \times N$ identity matrix, $\otimes$ denotes the Kronecker product, $\mathbf{T}_{\mathbf{zu}}$ is the batch model from $\mathbf{u}$ to $\mathbf{z}$, and $\bar{\mathbf{e}}_k = \bar{\mathbf{r}} - \bar{\mathbf{z}}_k$. The reader is referred to Amann et al. (1996); Gunnarsson and Norrlöf (2001) for details of the derivation.

Under the assumption that there are no model uncertainties or noise present, the update equation (2a) is stable and monotone for all system descriptions $\mathbf{T}_{\mathbf{zu}}$, i.e., the batch signal $\bar{\mathbf{u}}$ converges to a constant value monotonically, see e.g. Barton et al. (2008); Gunnarsson and Norrlöf (2001).

3. Estimation-based ILC for Linear Systems

The error $\mathbf{e}_k(t)$ used in the ILC algorithm should be the difference between the reference $\mathbf{r}(t)$ and the controlled variable $\mathbf{z}_k(t)$ at iteration k. In general the controlled variable $\mathbf{z}_k(t)$ is not directly measurable, therefore an estimation-based ILC algorithm must be used, i.e., the ILC algorithm has to rely on estimates based on measurements of related quantities. The situation is schematically described in Figure 1.

3.1. Estimation-based Norm-optimal ILC

A straightforward extension to the standard norm-optimal ILC method is to use the error $\hat{\mathbf{e}}_k(t) = \mathbf{r}(t) - \hat{\mathbf{z}}_k(t)$

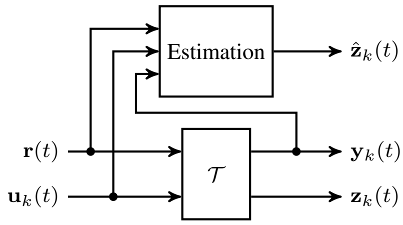

Figure 1: System $\mathcal{T}$ with reference $\mathbf{r}(t)$, ILC input $\mathbf{u}_k(t)$, measured variable $\mathbf{y}_k(t)$ and controlled variable $\mathbf{z}_k(t)$ at ILC iteration k and time t.

in the equations from Section 2, where $\hat{\mathbf{z}}_k(t)$ is an estimate of the controlled variable. The estimate $\hat{\mathbf{z}}_k(t)$ can be obtained using e.g., a Kalman filter (KF) for the linear case, or an extended Kalman filter (EKF) for the non-linear case (Kailath et al., 2000). Linear systems are covered in this section while Section 4 extends the ideas to non-linear systems. In both cases it must be assumed that i) the system is observable, and ii) the filter, used for estimation, converges.

The solution to the optimisation problem can be obtained in a similar way as for the standard norm-optimal ILC problem in Section 2, where the detailed derivation is presented in Amann et al. (1996); Gunnarsson and Norrlöf (2001). An important distinction, compared to standard norm-optimal ILC, relates to what models are used in the design. In the estimation-based approach, the model from $\mathbf{u}_{k+1}(t)$ and $\mathbf{r}(t)$ to $\hat{\mathbf{z}}_{k+1}(t)$ is used, i.e., the dynamics from the KF must be included, while in the standard norm-optimal design, the model from $\mathbf{u}_{k+1}(t)$ and $\mathbf{r}(t)$ to $\mathbf{z}_{k+1}(t)$ is used.

Let the discrete-time state space model be given by

$$\mathbf{x}_k(t+1) = \mathbf{A}(t)\mathbf{x}_k(t) + \mathbf{B_u}(t)\mathbf{u}_k(t) + \mathbf{B_r}(t)\mathbf{r}(t) + \mathbf{G}(t)\mathbf{w}_k(t), \tag{4a}$$

$$\mathbf{y}_k(t) = \mathbf{C_y}(t)\mathbf{x}_k(t) + \mathbf{D_{yu}}(t)\mathbf{u}_k(t) + \mathbf{D_{yr}}(t)\mathbf{r}(t) + \mathbf{v}_k(t), \tag{4b}$$

$$\mathbf{z}_k(t) = \mathbf{C_z}(t)\mathbf{x}_k(t) + \mathbf{D_{zu}}(t)\mathbf{u}_k(t) + \mathbf{D_{zr}}(t)\mathbf{r}(t), \tag{4c}$$

where the process noise $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$, and the measurement noise $\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t))$. A batch model (see Appendix A for definitions) for the output $\mathbf{y}_k$ and the estimate $\hat{\mathbf{z}}_k$ can be written as

$$\bar{\mathbf{y}}_k = \boldsymbol{\mathcal{C}}_\mathbf{y} \boldsymbol{\Phi} \mathbf{x}_0 + \mathbf{T_{yu}} \bar{\mathbf{u}}_k + \mathbf{T_{yr}} \bar{\mathbf{r}}, \tag{5a}$$

$$\bar{\hat{\mathbf{z}}}_k = \boldsymbol{\mathcal{C}}_\mathbf{z} \widetilde{\boldsymbol{\Phi}} \hat{\mathbf{x}}_0 + \mathbf{T_{\hat{z}u}} \bar{\mathbf{u}}_k + \mathbf{T_{\hat{z}r}} \bar{\mathbf{r}} + \mathbf{T_{\hat{z}y}} \bar{\mathbf{y}}_k, \tag{5b}$$

where $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are replaced by the corresponding expected values, which are both equal to zero, in the output model (5a). The KF batch formulation has been used in the model of the estimate in (5b). The optimal solution is now given by

$$\bar{\mathbf{u}}_{k+1} = \boldsymbol{\mathcal{Q}} \cdot (\bar{\mathbf{u}}_k + \boldsymbol{\mathcal{L}} \cdot \bar{\hat{\mathbf{e}}}_k) \tag{6a}$$

$$\boldsymbol{\mathcal{Q}} = (\mathbf{T}_\mathbf{u}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_\mathbf{u} + \boldsymbol{\mathcal{W}}_\mathbf{u} + \lambda \mathbf{I})^{-1} (\lambda \mathbf{I} + \mathbf{T}_\mathbf{u}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_\mathbf{u}) \tag{6b}$$

$$\boldsymbol{\mathcal{L}} = (\lambda \mathbf{I} + \mathbf{T}_\mathbf{u}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e} \mathbf{T}_\mathbf{u})^{-1} \mathbf{T}_\mathbf{u}^\mathsf{T} \boldsymbol{\mathcal{W}}_\mathbf{e}, \tag{6c}$$

[1] $\mathbb{S}^n_{++}$ denotes a $n \times n$ positive definite matrix.

where $\mathbf{T_u} = \mathbf{T_{\hat{z}u}} + \mathbf{T_{\hat{z}y}}\mathbf{T_{yu}}$ (see (A.6), (A.10) for details), and $\hat{\bar{\mathbf{e}}}_k = \bar{\mathbf{r}} - \hat{\bar{\mathbf{z}}}_k$. The expressions for $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{L}}$ in (6) are similar to (2). The only difference is that $\mathbf{T_u}$ is used instead of $\mathbf{T_{zu}}$. Theorem 1 presents a result for the special case of LTI-systems using the stationary KF[2].

Theorem 1 *(Separation lemma for estimation-based ILC): Given an LTI-system and the stationary KF with constant gain matrix $\mathbf{K}$, then the matrices $\mathbf{T_u}$ and $\mathbf{T_{zu}}$ are equal, hence the ILC algorithm (2) can be used for the estimation-based norm-optimal ILC.*

Proof. Assume that $\mathbf{D_{yu}} = \mathbf{0}$ and $\mathbf{D_{zu}} = \mathbf{0}$, then it holds that $\mathbf{T_{zu}} = \boldsymbol{\mathcal{C}}_\mathbf{z}\boldsymbol{\Psi}\boldsymbol{\mathcal{B}}_\mathbf{u}$ and $\mathbf{T_u} = \boldsymbol{\mathcal{C}}_\mathbf{z}\widetilde{\boldsymbol{\Psi}}\widetilde{\boldsymbol{\mathcal{B}}}_\mathbf{u} + \boldsymbol{\mathcal{C}}_\mathbf{z}\widetilde{\boldsymbol{\Psi}}_2\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{C}}_\mathbf{y}\boldsymbol{\Psi}\boldsymbol{\mathcal{B}}_\mathbf{u}$, see Appendix A. The structure of $\widetilde{\boldsymbol{\mathcal{B}}}_\mathbf{u}$ gives

$$\mathbf{T_u} = \boldsymbol{\mathcal{C}}_\mathbf{z}\left(\widetilde{\boldsymbol{\Psi}}\boldsymbol{\Gamma} + \widetilde{\boldsymbol{\Psi}}_2\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{C}}_\mathbf{y}\boldsymbol{\Psi}\right)\boldsymbol{\mathcal{B}}_\mathbf{u},$$
$$\boldsymbol{\Gamma} = \operatorname{diag}\left(\mathbf{I} - \mathbf{K}\mathbf{C_y}, \quad \ldots, \quad \mathbf{I} - \mathbf{K}\mathbf{C_y}, \quad \mathbf{0}\right).$$

It can now be shown algebraically that $\widetilde{\boldsymbol{\Psi}}\boldsymbol{\Gamma} + \widetilde{\boldsymbol{\Psi}}_2\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{C}}_\mathbf{y}\boldsymbol{\Psi} = \boldsymbol{\Psi}$, hence $\mathbf{T_{zu}} = \mathbf{T_u}$. The case $\mathbf{D_{yu}} \neq \mathbf{0}$ and $\mathbf{D_{zu}} \neq \mathbf{0}$ gives similar, but more involved, calculations. □

The result from Theorem 1 makes it computationally more efficient to compute the matrices $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{L}}$, since the matrix $\mathbf{T_{zu}}$ requires fewer calculations to obtain, compared to the matrix $\mathbf{T_u}$. Even if the iterative KF update is used, the Kalman gain $\mathbf{K}$ converges eventually to the stationary value for LTI systems. If this is done reasonably fast, then $\mathbf{T_u} \approx \mathbf{T_{zu}}$ can be a good enough approximation to use.

The stability results for the ILC algorithm in (6) is given in Theorem 2.

Theorem 2 *(Stability and monotonic convergence): The estimation-based ILC algorithm (6) is stable and monotonically convergent for all systems given by (5).*

Proof. Assuming that the KF is initialised with the same covariance matrix $\mathbf{P}_0$ at each iteration, makes the sequence of Kalman gains $\mathbf{K}(t)$, $t = 0, \ldots, N-1$ the same for different ILC iterations since $\mathbf{P}$ and $\mathbf{K}$ are independent of the control signal $\mathbf{u}$. The system matrices $\mathbf{T_{\hat{z}u}}$ and $\mathbf{T_{\hat{z}y}}$ are therefore iteration independent, hence the same analysis for stability and monotone convergence as for the standard norm-optimal ILC, presented in Barton et al. (2008); Gunnarsson and Norrlöf (2001), can be used. □

For norm-optimal ILC it is known how the performance and the convergence speed of the error depend on the tuning parameters $\mathbf{W_e}$, $\mathbf{W_u}$, and λ. In Barton et al. (2008) it is shown that the convergence speed depends strongly on λ. Moreover, the performance, i.e., $\mathbf{e}_k$, $k \to \infty$, is not dependent on λ and that the smallest error is achieved for $\mathbf{W_u} = 0$. It follows directly from Theorem 1 that the same properties must hold for LTI systems when the stationary Kalman filter is used for estimation. The performance and the convergence speed must also depend in the same way for linear time-varying systems using the Kalman filter since the equations for the ILC algorithm are the same. Only the model structure differ and that does not affect the analysis of the performance and the convergence speed.

Remark 1. Here it has been assumed that the KF takes the signals $\mathbf{r}(t)$ and $\mathbf{u}_k(t)$ as inputs. However, in general the estimation algorithm does not always use $\mathbf{r}(t)$ and $\mathbf{u}_k(t)$ as inputs. As an example, a system with a feedback loop usually uses the input to the controlled system for estimation, not the input to the controller. In this case $\mathbf{T_{\hat{z}u}}$, $\mathbf{T_{\hat{z}r}}$, and $\mathbf{T_{\hat{z}y}}$ will change accordingly.

3.2. Utilising the Complete PDF for the ILC Error

The objective of ILC is to achieve an error close to zero. Only considering the mean value is insufficient since a large variance means that there is a high probability that the actual error is not close to zero. Hence, the mean of the distribution should be close to zero and at the same time the variance should be small. To achieve this the PDF of the error is compared with a desired PDF with zero mean and small variance, using the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). The objective function (1) is modified by replacing the term $\|\mathbf{e}_{k+1}(t)\|^2_{\mathbf{W_e}}$ with the KL-divergence $D_{\text{KL}}\left(q(\mathbf{e}_{k+1}(t))||p(\mathbf{e}_{k+1}(t))\right)$, where $p(\mathbf{e}_{k+1}(t))$ is the actual distribution of the error given by the estimator, and $q(\mathbf{e}_{k+1}(t))$ is the desired distribution for the error, which becomes a design parameter.

For the linear case using a KF for estimation gives that the estimated state vector is Gaussian distributed (Kailath et al., 2000). Moreover, the error is an affine transformation of the estimated state vector, recall (4c), hence the error is also Gaussian distributed

$$\hat{\mathbf{e}}(t) \sim p(\mathbf{e}(t)) = \mathcal{N}\left(\mathbf{e}; \hat{\mathbf{e}}(t|t), \boldsymbol{\Sigma}_\mathbf{e}(t|t)\right). \tag{7}$$

It is now assumed that $q(\mathbf{e}(t)) = \mathcal{N}(\mathbf{e}; \mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a design parameter. The KL-divergence with the two Gaussian distributions $p(\mathbf{e}(t))$ and $q(\mathbf{e}(t))$ becomes (Arndt, 2001)

$$D_{\text{KL}}(q(\mathbf{e}(t))||p(\mathbf{e}(t))) = \frac{1}{2}\hat{\mathbf{e}}(t|t)^\mathsf{T}\boldsymbol{\Sigma}_\mathbf{e}^{-1}(t|t)\hat{\mathbf{e}}(t|t) + \xi, \tag{8}$$

where ξ is a term which does not depend on $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$. The objective function is finally modified by replacing the term $\|\mathbf{e}_{k+1}(t)\|^2_{\mathbf{W}_e}$ in (1) with $\|\hat{\mathbf{e}}_{k+1}(t|t)\|^2_{\boldsymbol{\Sigma}_\mathbf{e}^{-1}(t|t)}$. The interpretation of the result is that, if the estimated quantity is certain it will affect the objective function more than if it is less certain. The solution to the optimisation problem is the same as in (2) with

$$\boldsymbol{\mathcal{W}}_\mathbf{e} = \operatorname{diag}\left(\boldsymbol{\Sigma}_\mathbf{e}^{-1}(0|0), \quad \ldots, \quad \boldsymbol{\Sigma}_\mathbf{e}^{-1}(N-1|N-1)\right) \in \mathbb{S}^{Nn_z}_{++}.$$

Utilising the KL-divergence has provided a way to automatically choose the weight matrix $\mathbf{W_e}$ and, in addition, it provides an intuitive interpretation of $\mathbf{W_e}$.

[2] The stationary Kalman filter uses a constant gain $\mathbf{K}$ which is the solution to an algebraic Riccati equation, see Kailath et al. (2000).

Remark 2. The separation lemma in Theorem 1 and the stability result in Theorem 2 are not affected when the full PDF is included in the objective function.

4. Estimation-based ILC for Non-linear Systems

The ILC algorithm in Lin et al. (2006), which is based on the Newton method, utilise the fact that the solution can be rewritten as a two-step procedure. In the first step, a linearised system is obtained and standard linear ILC methods can be applied. In the second step, the ILC signal for the non-linear system is updated using the updated ILC signal from step one. However, nothing is stated about how to generally obtain the state trajectory for the linearisation step.

The method proposed here is to directly transform the non-linear system to a linear time-varying system, and then use the standard norm-optimal method. The linear state space model is obtained by linearising around an estimate of the complete state trajectory obtained from the previous ILC iteration. A necessary assumption is to have $\overline{\mathbf{u}}_{k+1}$ close to $\overline{\mathbf{u}}_k$, in order to get accurate models for the optimisation. It is possible to achieve $\overline{\mathbf{u}}_{k+1}$ close to $\overline{\mathbf{u}}_k$ by assigning λ a large enough value. The drawback is that the convergence rate can become slow.

4.1. Solution using Linearised Model

Linearisation of the non-linear model

$$\mathbf{x}_k(t+1) = f(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)), \tag{9a}$$
$$\mathbf{y}_k(t) = h_{\mathbf{y}}(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)), \tag{9b}$$
$$\mathbf{z}_k(t) = h_{\mathbf{z}}(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)), \tag{9c}$$

for iteration k and discrete time t, around $\hat{\mathbf{x}}_{k-1}(t|t)$, $\mathbf{u}_{k-1}(t)$, and $\mathbf{r}(t)$ gives a linear time-varying model, see Appendix B. The linearised quantities only depend on the previous ILC iteration, hence they are known at the current iteration. Using the linearised state space model gives, as before, the estimate $\hat{\mathbf{z}}_k(t)$ and the output $\mathbf{y}_k(t)$ in batch form as

$$\begin{aligned}\overline{\mathbf{y}}_k =& \boldsymbol{\mathcal{C}}_{\mathbf{y},k-1}\boldsymbol{\Phi}_{k-1}\mathbf{x}_0 + \mathbf{T}_{\mathbf{yu},k-1}\overline{\mathbf{u}}_k + \mathbf{T}_{\mathbf{ys_x},k-1}\overline{\mathbf{s}}_{\mathbf{x},k-1} + \overline{\mathbf{s}}_{\mathbf{y},k-1},\\ \hat{\overline{\mathbf{z}}}_k =& \boldsymbol{\mathcal{C}}_{\mathbf{z},k-1}\widetilde{\boldsymbol{\Phi}}_{k-1}\hat{\mathbf{x}}_0 + \mathbf{T}_{\hat{\mathbf{z}}\mathbf{u},k-1}\overline{\mathbf{u}}_k + \mathbf{T}_{\hat{\mathbf{z}}\mathbf{y},k-1}\overline{\mathbf{y}}_k\\ &+ \mathbf{T}_{\hat{\mathbf{z}}\mathbf{s_x},k-1}\overline{\mathbf{s}}_{\mathbf{x},k-1} + \mathbf{T}_{\hat{\mathbf{z}}\mathbf{s_y},k-1}\overline{\mathbf{s}}_{\mathbf{y},k-1} + \overline{\mathbf{s}}_{\mathbf{z},k-1},\end{aligned}$$

where all the matrices in the right hand side are dependent of k, and the vectors $\overline{\mathbf{s}}_{\mathbf{x},k}$, $\overline{\mathbf{s}}_{\mathbf{y},k}$, and $\overline{\mathbf{s}}_{\mathbf{z},k}$ are stacked versions of $\mathbf{s}_{\mathbf{x},k}(t)$, $\mathbf{s}_{\mathbf{y},k}(t)$, and $\mathbf{s}_{\mathbf{z},k}(t)$.

The norm-optimal ILC problem can be formulated and solved in the same way as before. Unfortunately, since the batch form matrices are dependent of k, the terms including $\overline{\mathbf{r}}$, $\hat{\mathbf{x}}_0$, $\mathbf{x}_0$, $\overline{\mathbf{s}}_{\mathbf{x},k}$, $\overline{\mathbf{s}}_{\mathbf{y},k}$, and $\overline{\mathbf{s}}_{\mathbf{z},k}$ cannot be eliminated. Note that the weight matrix for the error is also dependent of k, since it consists of the covariance matrices for the control error. The solution is therefore given by

$$\begin{aligned}\overline{\mathbf{u}}_{k+1} =& \left(\mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\mathbf{T}_{\mathbf{u},k} + \boldsymbol{\mathcal{W}}_{\mathbf{u}} + \lambda\mathbf{I}\right)^{-1}\Big[\lambda\overline{\mathbf{u}}_k + \mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\\ &\times\big(\overline{\mathbf{r}} - \boldsymbol{\mathcal{C}}_{\mathbf{z},k}\widetilde{\boldsymbol{\Phi}}_k\hat{\mathbf{x}}_0 - \mathbf{T}_{\hat{\mathbf{z}}\mathbf{s_x},k}\overline{\mathbf{s}}_{\mathbf{x},k} - \mathbf{T}_{\hat{\mathbf{z}}\mathbf{s_y},k}\overline{\mathbf{s}}_{\mathbf{y},k}\\ &- \overline{\mathbf{s}}_{\mathbf{z},k} - \mathbf{T}_{\hat{\mathbf{z}}\mathbf{y},k}(\boldsymbol{\mathcal{C}}_{\mathbf{y},k}\boldsymbol{\Phi}_k\mathbf{x}_0 + \mathbf{T}_{\mathbf{ys_x},k}\overline{\mathbf{s}}_{\mathbf{x},k} + \overline{\mathbf{s}}_{\mathbf{y},k})\big)\Big],\end{aligned} \tag{10}$$

where $\mathbf{T}_{\mathbf{u},k} = \mathbf{T}_{\hat{\mathbf{z}}\mathbf{u},k} + \mathbf{T}_{\hat{\mathbf{z}}\mathbf{y},k}\mathbf{T}_{\mathbf{yu},k}$ (see (A.6), (A.10) for details). The initial condition $\mathbf{x}_0$ is of course not known, hence $\hat{\mathbf{x}}_0$ must be used instead.

Remark 3. If the change in $\|\overline{\mathbf{u}}_{k+1} - \overline{\mathbf{u}}_k\|$ is sufficiently small, then the approximation $\mathbf{T}_{\mathbf{yu},k-1} \approx \mathbf{T}_{\mathbf{yu},k}$ and similar for the rest of the quantities is appropriate. Given the approximation the ILC algorithm (10) is simplified to

$$\overline{\mathbf{u}}_{k+1} = \boldsymbol{\mathcal{Q}}_k \cdot (\overline{\mathbf{u}}_k + \boldsymbol{\mathcal{L}}_k \cdot \hat{\overline{\mathbf{e}}}_k) \tag{11a}$$
$$\begin{aligned}\boldsymbol{\mathcal{Q}}_k =& (\mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\mathbf{T}_{\mathbf{u},k} + \boldsymbol{\mathcal{W}}_{\mathbf{u}} + \lambda\mathbf{I})^{-1}\\ &\times(\lambda\mathbf{I} + \mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\mathbf{T}_{\mathbf{u},k})\end{aligned} \tag{11b}$$
$$\boldsymbol{\mathcal{L}}_k = (\lambda\mathbf{I} + \mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\mathbf{T}_{\mathbf{u},k})^{-1}\mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}. \tag{11c}$$

Note that the change in $\overline{\mathbf{u}}$ depends strongly on the choice of λ.

4.2. Stability Analysis for the Linearised Solution

To analyse the stability of non-linear systems, utilising the linearisation technique, it is necessary to use the convergence results from Norrlöf and Gunnarsson (2002), which are based on theory for discrete time-variant systems. The stability properties for the iteration-variant ILC system (10) is presented in Theorem 3.

Theorem 3 *(Iteration-variant stability): If $f(\cdot)$, $h_{\mathbf{y}}(\cdot)$ and $h_{\mathbf{z}}(\cdot)$ are differentiable, then the system using the* ILC *algorithm* (10) *is stable.*

Proof. From Norrlöf and Gunnarsson (2002, Corollary 3) it follows that (10) is stable if and only if

$$\bar{\sigma}\left(\left(\mathbf{T}_{\mathbf{u},k}^{\mathsf{T}}\boldsymbol{\mathcal{W}}_{\mathbf{e},k}\mathbf{T}_{\mathbf{u},k} + \boldsymbol{\mathcal{W}}_{\mathbf{u}} + \lambda\mathbf{I}\right)^{-1}\lambda\right) < 1, \tag{12}$$

for all k. The construction of $\boldsymbol{\mathcal{W}}_{\mathbf{e},k}$ from the covariance matrices, gives that $\boldsymbol{\mathcal{W}}_{\mathbf{e},k} \in \mathbb{S}_{++}$ for all k. This fact, together with the fact that $\boldsymbol{\mathcal{W}}_{\mathbf{u}} \in \mathbb{S}_{++}$, and $\lambda \in \mathbb{R}_+$ guarantee that (12) is fulfilled for all k, hence the system with the ILC algorithm is stable. □

5. Conclusions and Future Work

An estimation-based norm-optimal ILC algorithm was derived and as a first contribution it was shown that the control law can be separated from the estimator dynamics in the case of LTI-systems and the stationary Kalman filter. The second contribution is the extension of the regular optimisation criteria for norm-optimal ILC, to include both the first and second order moments of the posterior PDF, enabling an automatic and intuitive way to choose the weight parameter $\mathbf{W_e}$ in the objective function. Finally, the third contribution is the extensions of the results, including stability and convergence, to non-linear systems using linearisation. The estimation-based norm-optimal ILC framework enables a systematic model-based design of ILC algorithms for linear as well as non-linear systems, where the controlled variables are not directly available as measurements.

Future work includes using smoothing instead of filtering, to obtain the estimates, and to include the smoother dynamics in the ILC design. Also, investigating the use of the KL-divergence when the estimates are obtained using a particle filter or particle smoother is a possible extension.

Appendix A. State Space Model and Kalman Filter in Batch Form

For the norm-optimal ILC algorithm it is convenient to describe the state space model over the whole time horizon in batch form. The discrete-time state space model

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B_u}(t)\mathbf{u}(t), \tag{A.1a}$$

$$\mathbf{y}(t) = \mathbf{C_y}(t)\mathbf{x}(t) + \mathbf{D_{yu}}(t)\mathbf{u}(t), \tag{A.1b}$$

has the following update formula for the state vector (Rugh, 1996)

$$\mathbf{x}(t) = \boldsymbol{\phi}(t,t_0)\mathbf{x}(t_0) + \sum_{j=t_0}^{t-1} \boldsymbol{\phi}(t,j+1)\mathbf{B_u}(j)\mathbf{u}(j), \tag{A.2}$$

for $t \geq t_0 + 1$, where the discrete-time transition matrix $\boldsymbol{\phi}(t,j)$ is

$$\boldsymbol{\phi}(t,j) = \begin{cases} \mathbf{A}(t-1)\cdot \ldots \cdot \mathbf{A}(j), & t \geq j+1 \\ \mathbf{I}, & t = j \end{cases}. \tag{A.3}$$

Using (A.1b) and (A.2), the output is given by

$$\begin{aligned}\mathbf{y}(t) =& \mathbf{C_y}(t)\boldsymbol{\phi}(t,t_0)\mathbf{x}(t_0) + \mathbf{D_{yu}}(t)\mathbf{u}(t) \\ &+ \sum_{j=t_0}^{t-1} \mathbf{C_y}(t)\boldsymbol{\phi}(t,j+1)\mathbf{B_u}(j)\mathbf{u}(j). \end{aligned}\tag{A.4}$$

Introducing the vectors

$$\overline{\mathbf{x}} = \begin{pmatrix}\mathbf{x}(t_0)^\mathsf{T} & \ldots & \mathbf{x}(t_0+N)^\mathsf{T}\end{pmatrix}^\mathsf{T} \tag{A.5a}$$

$$\overline{\mathbf{u}} = \begin{pmatrix}\mathbf{u}(t_0)^\mathsf{T} & \ldots & \mathbf{u}(t_0+N)^\mathsf{T}\end{pmatrix}^\mathsf{T} \tag{A.5b}$$

$$\overline{\mathbf{y}} = \begin{pmatrix}\mathbf{y}(t_0)^\mathsf{T} & \ldots & \mathbf{y}(t_0+N)^\mathsf{T}\end{pmatrix}^\mathsf{T} \tag{A.5c}$$

gives the solution from $t = t_0$ to $t = t_0 + N$ as $\overline{\mathbf{x}} = \boldsymbol{\Phi}\mathbf{x}(t_0) + \boldsymbol{\Psi}\boldsymbol{\mathcal{B}}_\mathbf{u}\overline{\mathbf{u}}$ and for the output as

$$\overline{\mathbf{y}} = \boldsymbol{\mathcal{C}}_\mathbf{y}\boldsymbol{\Phi}\mathbf{x}(t_0) + \underbrace{(\boldsymbol{\mathcal{C}}_\mathbf{y}\boldsymbol{\Psi}\boldsymbol{\mathcal{B}}_\mathbf{u} + \boldsymbol{\mathcal{D}}_\mathbf{yu})}_{\triangleq \mathbf{T_{yu}}}\overline{\mathbf{u}}. \tag{A.6}$$

Here $\mathbf{x}(t_0)$ is the initial value, and

$$\boldsymbol{\mathcal{B}}_\mathbf{u} = \operatorname{diag}\begin{pmatrix}\mathbf{B_u}(t_0), & \ldots, & \mathbf{B_u}(t_0+N-1), & 0\end{pmatrix} \tag{A.7a}$$

$$\boldsymbol{\mathcal{C}}_\mathbf{y} = \operatorname{diag}\begin{pmatrix}\mathbf{C_y}(t_0), & \ldots, & \mathbf{C_y}(t_0+N)\end{pmatrix} \tag{A.7b}$$

$$\boldsymbol{\mathcal{D}}_\mathbf{yu} = \operatorname{diag}\begin{pmatrix}\mathbf{D_{yu}}(t_0), & \ldots, & \mathbf{D_{yu}}(t_0+N)\end{pmatrix} \tag{A.7c}$$

$$\boldsymbol{\Phi} = \begin{pmatrix}\mathbf{I} \\ \boldsymbol{\phi}(t_0+1,t_0) \\ \boldsymbol{\phi}(t_0+2,t_0) \\ \vdots \\ \boldsymbol{\phi}(t_0+N,t_0)\end{pmatrix} \tag{A.7d}$$

$$\boldsymbol{\Psi} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \mathbf{I} & 0 & 0 & \cdots & 0 \\ \boldsymbol{\phi}(t_0+2,t_0+1) & \mathbf{I} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{\phi}(t_0+N,t_0+1) & \boldsymbol{\phi}(t_0+N,t_0+2) & \cdots & \mathbf{I} & 0 \end{pmatrix} \tag{A.7e}$$

The Kalman filter can be written in a similar batch form as described above. Let the state space model be given by

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B_u}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t), \tag{A.8}$$

$$\mathbf{y}(t) = \mathbf{C_y}(t)\mathbf{x}(t) + \mathbf{D_{yu}}(t)\mathbf{u}(t) + \mathbf{v}(t), \tag{A.9}$$

where $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$, and $\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t))$. From the filter recursions (Kailath et al., 2000) we get

$$\begin{aligned}&\hat{\mathbf{x}}(t+1|t+1) = (\mathbf{I} - \mathbf{K}(t+1)\mathbf{C_y}(t+1))\mathbf{A}(t)\hat{\mathbf{x}}(t|t) \\ &+ (\mathbf{I} - \mathbf{K}(t+1)\mathbf{C_y}(t+1))\mathbf{B_u}(t)\mathbf{u}(t) \\ &- \mathbf{K}(t+1)\mathbf{D_{yu}}(t+1)\mathbf{u}(t+1) + \mathbf{K}(t+1)\mathbf{y}(t+1) \\ &= \widetilde{\mathbf{A}}(t)\hat{\mathbf{x}}(t|t) + \widetilde{\mathbf{B}}_\mathbf{u}(t)\mathbf{u}(t) - \widetilde{\mathbf{D}}_\mathbf{yu}(t+1)\mathbf{u}(t+1) \\ &+ \mathbf{K}(t+1)\mathbf{y}(t+1),\end{aligned}$$

where, $\mathbf{K}(t)$ is the Kalman gain given by the recursion

$$\begin{aligned}\mathbf{P}(t|t-1) =& \mathbf{A}(t-1)\mathbf{P}(t-1|t-1)\mathbf{A}(t-1)^\mathsf{T} \\ &+ \mathbf{G}(t-1)\mathbf{Q}(t-1)\mathbf{G}(t-1)^\mathsf{T} \\ \mathbf{K}(t) =& \mathbf{P}(t|t-1)\mathbf{C_y}(t)^\mathsf{T} \\ &\times \left(\mathbf{C_y}(t)\mathbf{P}(t|t-1)\mathbf{C_y}(t)^\mathsf{T} + \mathbf{R}(t)\right)^{-1} \\ \mathbf{P}(t|t) =& (\mathbf{I} - \mathbf{K}(t)\mathbf{C_y}(t))\mathbf{P}(t|t-1).\end{aligned}$$

The update formula for the estimated state vector is finally given by

$$\begin{aligned}\hat{\mathbf{x}}(t|t) =& \widetilde{\boldsymbol{\phi}}(t,t_0)\hat{\mathbf{x}}(t_0|t_0) + \sum_{j=t_0}^{t-1} \widetilde{\boldsymbol{\phi}}(t,j+1)\widetilde{\mathbf{B}}_\mathbf{u}(j)\mathbf{u}(j) \\ &- \sum_{j=t_0+1}^{t} \widetilde{\boldsymbol{\phi}}(t,j)\widetilde{\mathbf{D}}_\mathbf{yu}(j)\mathbf{u}(j) + \sum_{j=t_0+1}^{t} \widetilde{\boldsymbol{\phi}}(t,j)\mathbf{K}(j)\mathbf{y}(j),\end{aligned}$$

where $\widetilde{\boldsymbol{\phi}}$ is defined as in (A.3), with $\widetilde{\mathbf{A}}(t)$ instead of $\mathbf{A}(t)$.

The KF recursion in batch form becomes

$$\hat{\bar{\mathbf{x}}} = \widetilde{\mathbf{\Phi}}\hat{\mathbf{x}}(t_0|t_0) + (\widetilde{\mathbf{\Psi}}\widetilde{\boldsymbol{\mathcal{B}}}_{\mathbf{u}} - \widetilde{\mathbf{\Psi}}_2\widetilde{\boldsymbol{\mathcal{D}}}_{\mathbf{yu}})\bar{\mathbf{u}} + \widetilde{\mathbf{\Psi}}_2\boldsymbol{\mathcal{K}}\bar{\mathbf{y}},$$

where $\widetilde{\mathbf{\Phi}}$, $\widetilde{\mathbf{\Psi}}$, and $\widetilde{\boldsymbol{\mathcal{B}}}_{\mathbf{u}}$ are given in (A.7) with $\widetilde{\mathbf{A}}(t)$ and $\widetilde{\mathbf{B}}_{\mathbf{u}}(t)$ instead of $\mathbf{A}(t)$ and $\mathbf{B}_{\mathbf{u}}(t)$. The remaining matrices are defined as

$$\begin{aligned}
\widetilde{\mathbf{\Psi}}_2 &= \begin{pmatrix} \mathbf{0}_{(N+1)n_x \times n_x} & \widetilde{\mathbf{\Psi}} \begin{pmatrix} \mathbf{I}_{Nn_x} \\ \mathbf{0}_{n_x \times Nn_x} \end{pmatrix} \end{pmatrix} \\
\widetilde{\boldsymbol{\mathcal{D}}}_{\mathbf{yu}} &= \operatorname{diag}\Big(\mathbf{0}, \quad \widetilde{\mathbf{D}}_{\mathbf{yu}}(t_0+1), \quad \ldots, \quad \widetilde{\mathbf{D}}_{\mathbf{yu}}(t_0+N)\Big) \\
\boldsymbol{\mathcal{K}} &= \operatorname{diag}\big(\mathbf{0}, \quad \mathbf{K}(t_0+1), \quad \ldots, \quad \mathbf{K}(t_0+N)\big).
\end{aligned}$$

Finally, the batch formulation for the variable

$$\mathbf{z}(t) = \mathbf{C}_{\mathbf{z}}(t)\mathbf{x}(t) + \mathbf{D}_{\mathbf{zu}}(t)\mathbf{u}(t),$$

is given by

$$\begin{aligned}
\hat{\bar{\mathbf{z}}} =& \boldsymbol{\mathcal{C}}_{\mathbf{z}}\widetilde{\mathbf{\Phi}}\hat{\mathbf{x}}(t_0|t_0) + \underbrace{\Big(\boldsymbol{\mathcal{C}}_{\mathbf{z}}(\widetilde{\mathbf{\Psi}}\widetilde{\boldsymbol{\mathcal{B}}}_{\mathbf{u}} - \widetilde{\mathbf{\Psi}}_2\widetilde{\boldsymbol{\mathcal{D}}}_{\mathbf{yu}}) + \boldsymbol{\mathcal{D}}_{\mathbf{zu}}\Big)}_{\triangleq \mathbf{T}_{\hat{\mathbf{z}}\mathbf{u}}}\bar{\mathbf{u}} \\
&+ \underbrace{\boldsymbol{\mathcal{C}}_{\mathbf{z}}\widetilde{\mathbf{\Psi}}_2\boldsymbol{\mathcal{K}}}_{\triangleq \mathbf{T}_{\hat{\mathbf{z}}\mathbf{y}}}\bar{\mathbf{y}},
\end{aligned} \tag{A.10}$$

where $\boldsymbol{\mathcal{C}}_{\mathbf{z}}$ and $\boldsymbol{\mathcal{D}}_{\mathbf{zu}}$ are given in (A.7) using $\mathbf{C}_{\mathbf{z}}(t)$ and $\mathbf{D}_{\mathbf{zu}}(t)$.

Appendix B. Linearisation of Non-linear Model

Consider the non-linear model

$$\mathbf{x}_k(t+1) = f(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)) \tag{B.1a}$$
$$\mathbf{y}_k(t) = h_{\mathbf{y}}(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)) \tag{B.1b}$$
$$\mathbf{z}_k(t) = h_{\mathbf{z}}(\mathbf{x}_k(t), \mathbf{u}_k(t), \mathbf{r}(t)) \tag{B.1c}$$

for iteration k and discrete time t. Linearisation around $\hat{\mathbf{x}}_{k-1}(t|t)$, $\mathbf{u}_{k-1}(t)$, and $\mathbf{r}(t)$ gives the following linear time-varying model

$$\begin{aligned}
\mathbf{x}_k(t+1) &= \mathbf{A}_{k-1}(t)\mathbf{x}_k(t) + \mathbf{B}_{k-1}(t)\mathbf{u}_k(t) + \mathbf{s}_{\mathbf{x},k-1}(t), \\
\mathbf{y}_k(t) &= \mathbf{C}_{\mathbf{y},k-1}\mathbf{x}_k(t) + \mathbf{D}_{\mathbf{y},k-1}(t)\mathbf{u}_k(t) + \mathbf{s}_{\mathbf{y},k-1}(t), \\
\mathbf{z}_k(t) &= \mathbf{C}_{\mathbf{z},k-1}\mathbf{x}_k(t) + \mathbf{D}_{\mathbf{z},k-1}(t)\mathbf{u}_k(t) + \mathbf{s}_{\mathbf{z},k-1}(t),
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A}_{k-1}(t) &= \frac{\partial f}{\partial \mathbf{x}}, & \mathbf{B}_{k-1}(t) &= \frac{\partial f}{\partial \mathbf{u}}, & \mathbf{C}_{\mathbf{y},k-1}(t) &= \frac{\partial h_{\mathbf{y}}}{\partial \mathbf{x}}, \\
\mathbf{C}_{\mathbf{z},k-1}(t) &= \frac{\partial h_{\mathbf{z}}}{\partial \mathbf{x}}, & \mathbf{D}_{\mathbf{y},k-1}(t) &= \frac{\partial h_{\mathbf{y}}}{\partial \mathbf{u}}, & \mathbf{D}_{\mathbf{z},k-1}(t) &= \frac{\partial h_{\mathbf{z}}}{\partial \mathbf{u}},
\end{aligned}$$

all evaluated at the point $\mathbf{x} = \hat{\mathbf{x}}_{k-1}(t|t)$, $\mathbf{u} = \mathbf{u}_{k-1}(t)$, and $\mathbf{r} = \mathbf{r}(t)$, with

$$\begin{aligned}
\mathbf{s}_{\mathbf{x},k-1}(t) =& f(\hat{\mathbf{x}}_{k-1}(t|t), \mathbf{u}_{k-1}(t), \mathbf{r}(t)) \\
&- \mathbf{A}_{k-1}(t)\hat{\mathbf{x}}_{k-1}(t|t) - \mathbf{B}_{k-1}(t)\mathbf{u}_{k-1}(t) \\
\mathbf{s}_{\mathbf{y},k-1}(t) =& h_y(\hat{\mathbf{x}}_{k-1}(t|t), \mathbf{u}_{k-1}(t), \mathbf{r}(t)) \\
&- \mathbf{C}_{\mathbf{y},k-1}(t)\hat{\mathbf{x}}_{k-1}(t|t) - \mathbf{D}_{\mathbf{y},k-1}(t)\mathbf{u}_{k-1}(t) \\
\mathbf{s}_{\mathbf{z},k-1}(t) =& h_z(\hat{\mathbf{x}}_{k-1}(t|t), \mathbf{u}_{k-1}(t), \mathbf{r}(t)) \\
&- \mathbf{C}_{\mathbf{z},k-1}(t)\hat{\mathbf{x}}_{k-1}(t|t) - \mathbf{D}_{\mathbf{z},k-1}(t)\mathbf{u}_{k-1}(t).
\end{aligned}$$

Acknowledgment

This work was supported by the Vinnova Excellence Center LINK-SIC, by the Excellence Center at Linköping-Lund in Information Technology, ELLIIT, and by the SSF project Collaborative Localization.

References

Ahn, H.-S., Moore, K. L., Chen, Y., June 2006. Kalman filter-augmented iterative learning control on the iteration domain. In: Proceedings of the American Control Conference. Minneapolis, MN, USA, pp. 250–255.

Amann, N., Owens, D. H., Rogers, E., March 1996. Iterative learning control for discrete-time systems with exponential rate of convergence. IEE Proceedings Control Theory and Applications 143 (2), 217–224.

Arimoto, S., Kawamura, S., Miyazaki, F., 1984. Bettering operation of robots by learning. Journal of Robotic Systems 1 (2), 123–140.

Arndt, C., 2001. Information Measures. Springer-Verlag, Berlin, Heidelberg, Germany.

Avrachenkov, K. E., December 1998. Iterative learning control based on quasi-Newton methods. In: Proceedings of the 37th IEEE Conference on Decision and Control. Tampa, FL, USA, pp. 170–174.

Axelsson, P., Karlsson, R., Norrlöf, M., November 2013. Estimation-based ILC using particle filter with application to industrial manipulators. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Tokyo, Japan, pp. 1740–1745.

Barton, K., van de Wijdeven, J., Alleyne, A., Bosgra, O., Steinbuch, M., December 2008. Norm optimal cross-coupled iterative learning control. In: Proceedings of the 47th IEEE Conference on Decision and Control. Cancun, Mexico, pp. 3020–3025.

Gunnarsson, S., Norrlöf, M., December 2001. On the design of ILC algorithms using optimization. Automatica 37 (12), 2011–2016.

Kailath, T., Sayed, A. H., Hassibi, B., 2000. Linear Estimation. Information and System Sciences Series. Prentice Hall Inc., Upper Saddle River, NJ, USA.

Kullback, S., Leibler, R. A., March 1951. On information and sufficiency. Annals of Mathematical Statistics 22 (1), 79–86.

Lee, K. S., Lee, J. H., 1998. Iterative Learning Control: Analysis, Design, Integration and Application. Kluwer Academic Publishers, Boston, MA, USA, Ch. Design of Quadratic Criterion-based Iterative Learning Control, pp. 165–192.

Lin, T., Owens, D. H., Hätönen, J., 2006. Newton method based iterative learning control for discrete non-linear systems. International Journal of Control 79 (10), 1263–1276.

Moore, K. L., 1993. Iterative Learning Control for Deterministic Systems. Advances in Industrial Control. Springer-Verlag, London, UK.

Norrlöf, M., Gunnarsson, S., 2002. Time and frequency domain convergence properties in iterative learning control. International Journal of Control 75 (14), 1114–1126.

Rugh, W. J., 1996. Linear System Theory, 2nd Edition. Information and System Sciences Series. Prentice Hall Inc., Upper Saddle River, NJ, USA.

Wallén, J., Gunnarsson, S., Henriksson, R., Moberg, S., Norrlöf, M., December 2009. ILC applied to a flexible two-link robot model using sensor-fusion-based estimates. In: Proceedings of the 48th IEEE Conference on Decision and Control. Shanghai, China, pp. 458–463.

Xiong, Z., Zhang, J., March 2004. Batch-to-batch optimal control of nonlinear batch processes based on incrementally updated models. IEE Proceedings Control Theory and Applications 151 (2), 158–165.