Design of Norm-Optimal Iterative Learning Controllers: The Effect of an Iteration-Domain Kalman Filter for Disturbance Estimation

Nicolas Degen, Autonomous System Lab, ETH Zürich
Angela P. Schoellig, University of Toronto Institute of Aerospace Studies

16 December 2014
53rd Conference on Decision and Control, Los Angeles
Quadrocopter Tracking Performance

**Problem**: Unsatisfactory tracking performance

**Solution**: Iterative Learning Control with Kalman Filter (K-ILC)


Goal: Analytic Comparison of ILC Algorithms

Compare QILC and K-ILC
What are the differences?

QILC
• Quadratic cost criterion ILC

K-ILC
• Kalman-Filter-Enhanced ILC


Outline of the Presentation

1. Detailed Presentation of **K-ILC Algorithm**

2. **Comparison** with Standard QILC

3. **Simulation** Example
Lifted-Domain Representation

Lifted vector notation for \textbf{j-th} Iteration:

$$u_j = [u_j[1], u_j[2], \ldots, u_j[N]]^T$$

Equivalent for all other signals

Nominal System Model:
Linear, Discrete, Iteration-Constant

$$y_j = \begin{bmatrix} CB & 0 & 0 & 0 \\ CAB & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ CA^{N-1}B & \cdots & CAB & CB \end{bmatrix} u_j$$

Time-constant linear system for illustration

Desired output
Control input
Tracking error

Measured tracking error:

$$e_j = y_j - y_{\text{desired}}$$
Disturbance Estimation of K-ILC Algorithm

Linearised system $F$ around desired trajectory:

$$y_j = Fu_j$$

System Model Including Modelled Disturbance as Stochastic Process:

$$d_{j+1} = d_j + \omega_j$$

$$y_j = Fu_j + d_j + \mu_j$$

$$\omega_j \sim \mathcal{N}(0, E_j), \mu_j \sim \mathcal{N}(0, H_j)$$

$$d_0 \sim \mathcal{N}(0, P_0)$$

Kalman filter equations:

$$S_j = P_j + E_j$$

$$K_j = S_j(S_j + H_{j+1})^{-1}$$

$$P_j = (I - K_j)S_j.$$
Input Update of K-ILC Algorithm

A  **Error prediction** of next iteration:

\[
\tilde{e}_{j+1} = Fu_{j+1} - y_d + \hat{d}_{j+1}
\]

nominal model error

Kalman filter used through Estimation of Disturbance:

\[
\hat{d}_{j+1} = \hat{d}_j + K_j(y_d - Fu_j - \hat{d}_j)
\]

B  **Updated input** as solution of **convex optimisation** of cost function:

\[
u_{j+1} = \arg\min_{u'_{j+1} \in C} \{ J_{j+1}(u'_{j+1}) \}
\]

\[
J_{j+1} = \tilde{e}_{j+1}^T W_e \tilde{e}_{j+1}
\]
Video of ILC in Action

Start Learning
Goal: Analytic Comparison of ILC Algorithms

Objective: Compare QILC and K-ILC

**QILC**
- Quadratic cost criterion **ILC**
- Deterministic system model

**K-ILC**
- Kalman-Filter-Enhanced **ILC**
- Modelling errors as stochastic disturbance
- Separated disturbance estimation and input update

---


Comparison of Input Update

QILC

A Error prediction

\[ \bar{e}_{j+1} = F \Delta u_{j+1} + e_j \]

\[ \Delta u_{j+1} = u_{j+1} - u_j \]

B Input update cost function

\[ J_{j+1} = \bar{e}_{j+1}^T \hat{W} e \bar{e}_{j+1} + \Delta u_{j+1}^T W \Delta u \Delta u_{j+1} \]

K-ILC

nominal model error

\[ \bar{e}_{j+1} = Fu_{j+1} - y_d + \hat{d}_{j+1} \]

noise filtering

\[ J_{j+1} = \bar{e}_{j+1}^T \hat{W} e \bar{e}_{j+1} \]
Parameters Defining the Algorithms

QILC

\[ J_{j+1} = \bar{e}_{j+1}^T W e \bar{e}_{j+1} + \Delta u_{j+1}^T W \Delta u \Delta u_{j+1} \]

noise filtering

2 Weighting Matrices

K-ILC

\[ d_{j+1} = d_f + \omega_j \]
\[ y_j = F u_j + d_j + \mu_j, \]
\[ \omega_j \sim \mathcal{N}(0, E_j), \mu_j \sim \mathcal{N}(0, H_j) \]
\[ d_0 \sim \mathcal{N}(0, P_0) \]

\[ S_j = P_j + E_j \]
\[ K_j = S_j(S_j + H_{j+1})^{-1} \]
\[ P_j = (I - K_j)S_j. \]

3 Covariance Matrices
Quadratic Norm Allows an Explicit Comparison

\[ QILC_{u_{j+1}} = u_{\text{nom}} - \sum_{i=1}^{j} QILC_{L} e_j \]

\[ K-ILC_{u_{j+1}} = u_{\text{nom}} - \sum_{i=1}^{j} K-ILC_{L_j} e_j \]

\[ QILC_{L} = (W_{\Delta u} + F^T W_{e} F)^{-1} F^T W_{e} = F^{-1} \]

\[ K-ILC_{L_j} = F^{-1} K_i = F^{-1} \]

**Explicit notation possible with quadratic norm and no constraints!**

For given iteration \( QILC \) can be made **equivalent** to \( K-ILC \)

➤ **K-ILC optimises** gain for **every iteration**
Mass-Spring-Damper Simulation Example

QILC equivalent of converged K-ILC robust, but converging slowly

QILC equivalent of initial K-ILC converging fast, but not robust once converged noise

K-ILC designed for the problem
QILC designed for the problem
QILC equivalent of converged K-ILC
QILC equivalent of initial K-ILC
Advantages of K-ILC Algorithm

Implications of Kalman filter usage:

1. **Separation** between disturbance estimation and input update
2. Straightforward **iteration-varying and optimal** input update behaviour:
   - Fast initial convergence behaviour
   - Noise-resilient converged behaviour

![Diagram showing ILC, Disturbance Estimator, Input Update, and System relationships]
Thank you!

Nicolas Degen, ETH Zurich
Angela P Schoellig, University of Toronto

53rd Conference on Decision and Control, 2014
Los Angeles