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

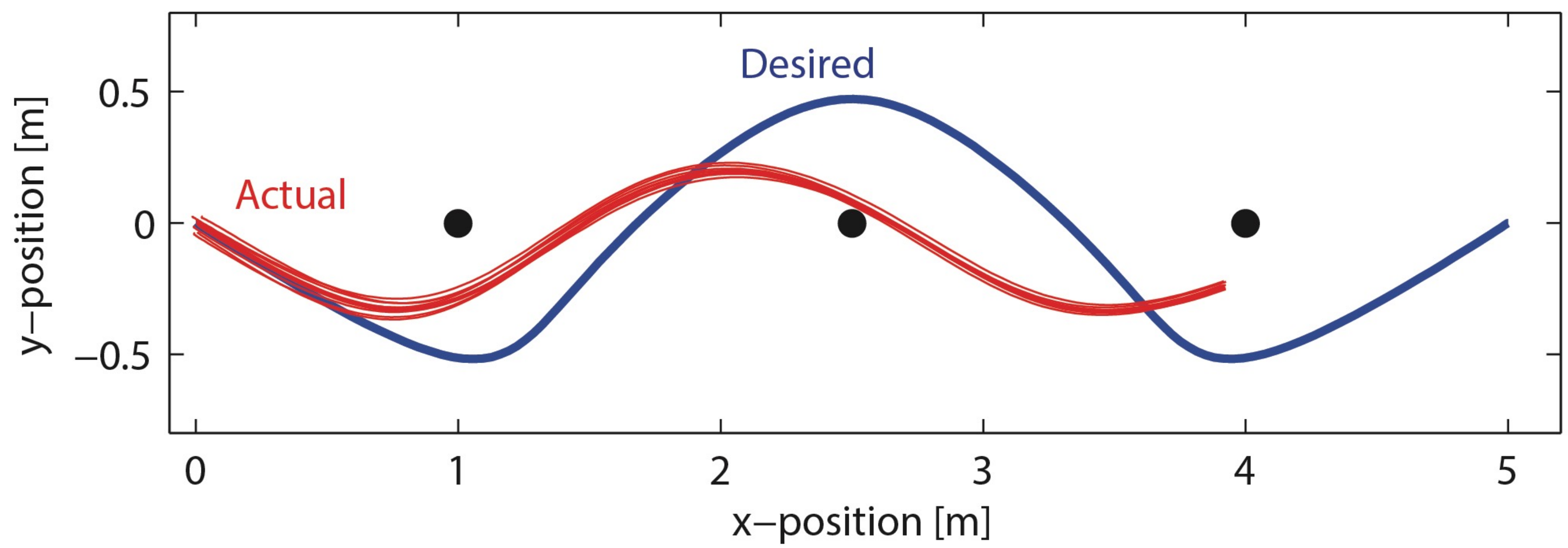
16 December 2014
53rd Conference on Decision and Control, Los Angeles

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

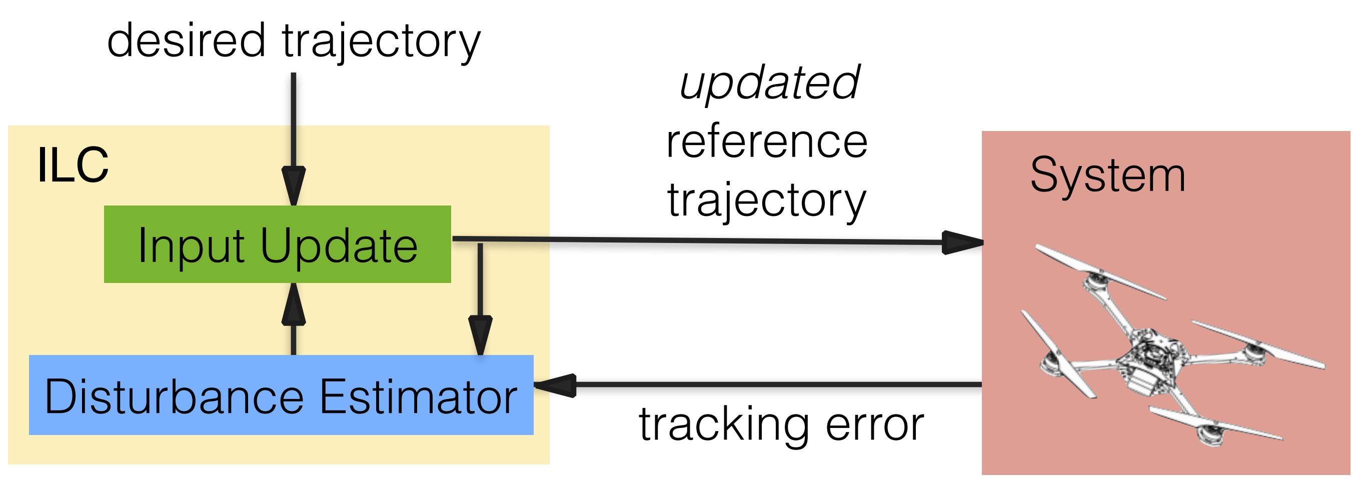
DYNAMIC
SYSTEMS LAB

Quadrocopter Tracking Performance

Problem: Unsatisfactory tracking performance



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



A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," Autonomous Robots, vol. 33, no. 1-2, pp. 103–127, 2012.

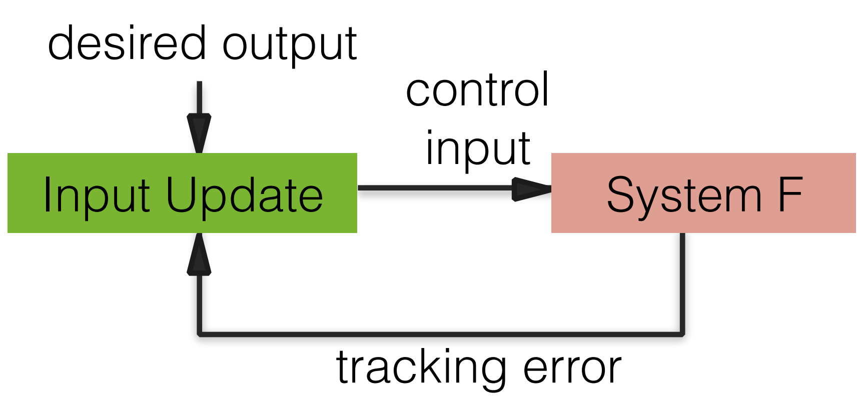
F. L. Mueller, A. P. Schoellig, and R. D'Andrea, "Iterative learning of feed-forward corrections for high-performance tracking," in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 3276–3281.

Goal: Analytic Comparison of ILC Algorithms

Compare QILC and K-ILC
What are the differences?

QILC

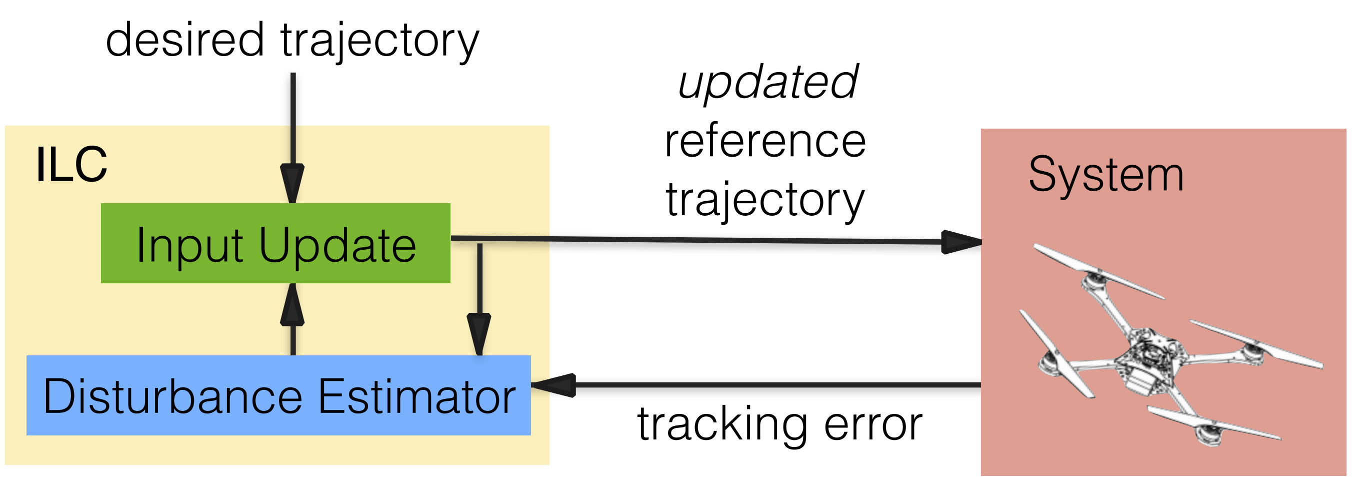
- **Q**uadratic cost criterion **ILC**



J. H. Lee, K. S. Lee, and W. C. Kim, “Model-based iterative learning control with a quadratic criterion for time-varying linear systems,” Automatica, vol. 36, pp. 641–657, 2000.

K-ILC

- **K**alman-Filter-Enhanced **ILC**



A. P. Schoellig, F. L. Mueller, and R. D’Andrea, “Optimization- based iterative learning for precise quadrocopter trajectory tracking,” Autonomous Robots, vol. 33, no. 1-2, pp. 103–127, 2012.

Outline of the Presentation

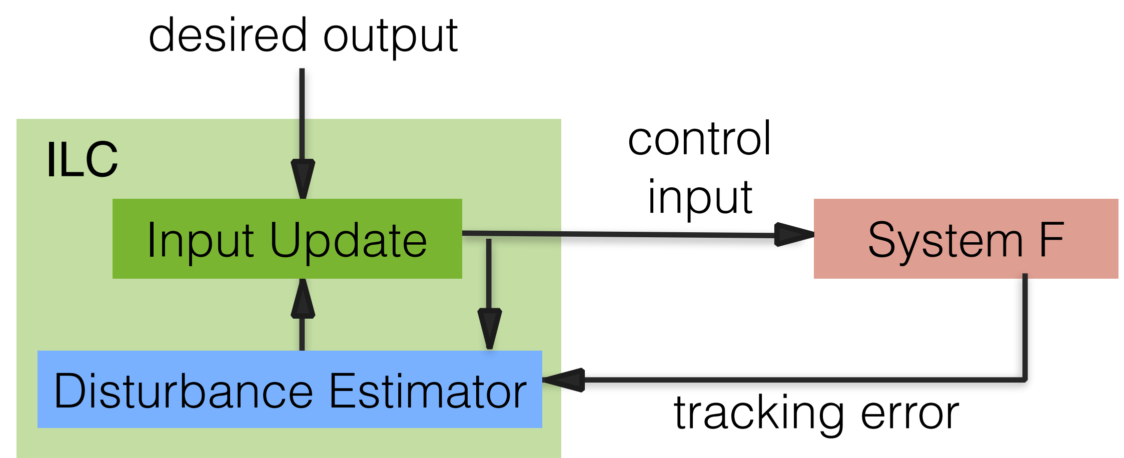
1. Detailed Presentation of **K-ILC Algorithm**
2. **Comparison** with Standard QILC
3. **Simulation** Example

Lifted-Domain Representation

Lifted vector notation for **j-th** Iteration:

$$u_j = [u_j[1], u_j[2], ..., u_j[N]]^T$$

Equivalent for all other signals



Nominal System Model:
Linear, Discrete, Iteration-Constant

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

Time-constant linear system for illustration

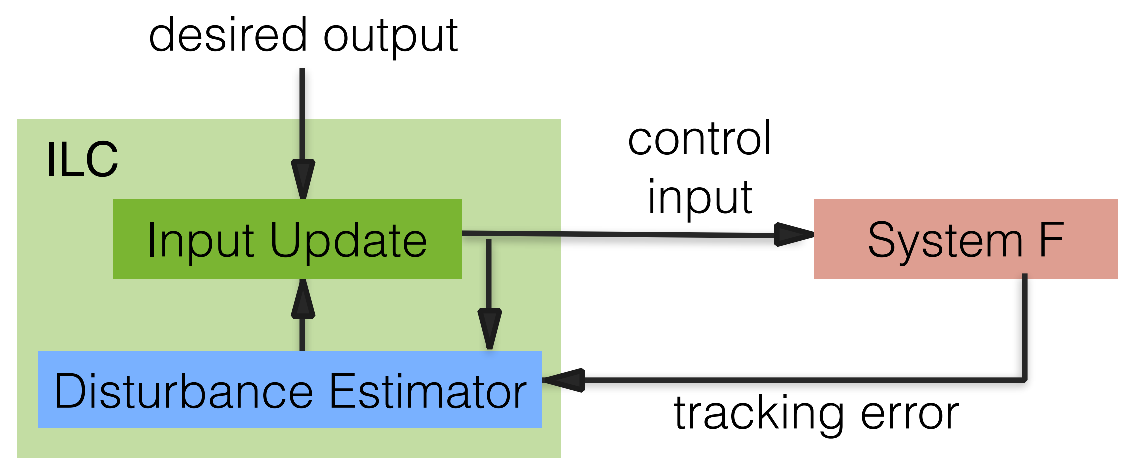
Measured tracking error:

$$e_j = y_j - y_{desired}$$

Disturbance Estimation of K-ILC Algorithm

Linearised system F around desired trajectory:

$$y_j = F u_j$$



System Model Including Modelled **Disturbance** as **Stochastic Process**:

$$\boxed{\begin{aligned} d_{j+1} &= d_j + \omega_j \\ y_j &= F u_j + d_j + \mu_j \end{aligned}}$$

stochastic disturbance representing modelling errors

modelled system output

$$\omega_j \sim \mathcal{N}(0, E_j), \mu_j \sim \mathcal{N}(0, H_j)$$
$$d_0 \sim \mathcal{N}(0, P_0)$$

random variable distributions

Kalman filter equations:
$$\begin{aligned} S_j &= P_j + E_j \\ K_j &= S_j (S_j + H_{j+1})^{-1} \\ P_j &= (I - K_j) S_j . \end{aligned}$$

iteration-varying Kalman gain K_j

Input Update of K-ILC Algorithm

A **Error prediction** of next iteration:

$$\bar{e}_{j+1} = \underbrace{F u_{j+1} - y_d}_{\text{nominal model error}} + \hat{d}_{j+1}$$

Kalman filter used through Estimation of Disturbance:

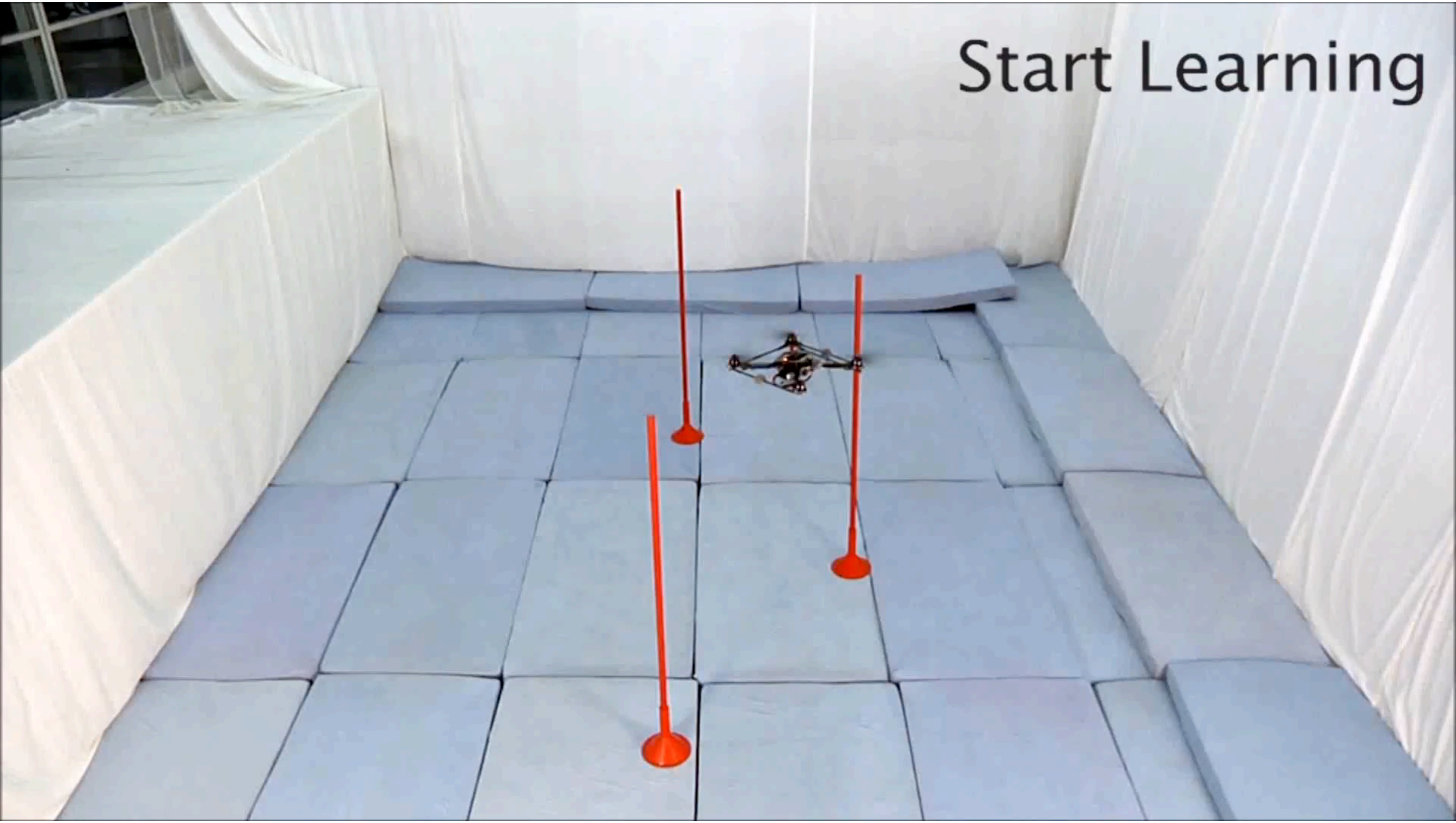
$$\hat{d}_{j+1} = \hat{d}_j + K_j (y_d - F u_j - \hat{d}_j)$$

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

$$u_{j+1} = \underset{u'_{j+1} \in C}{\operatorname{argmin}} \{ J_{j+1}(u'_{j+1}) \}$$

$$J_{j+1} = \bar{e}_{j+1}^T W_e \bar{e}_{j+1}$$

Video of ILC in Action



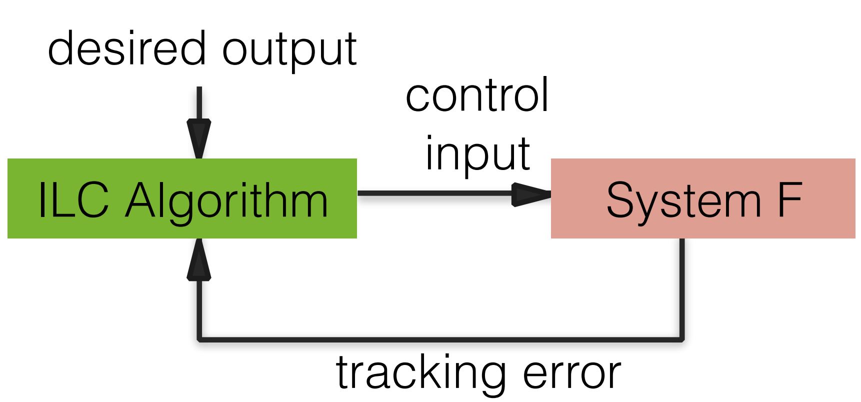
Goal: Analytic Comparison of ILC Algorithms

DYNAMIC SYSTEMS LAB

Objective: Compare QILC and K-ILC

QILC

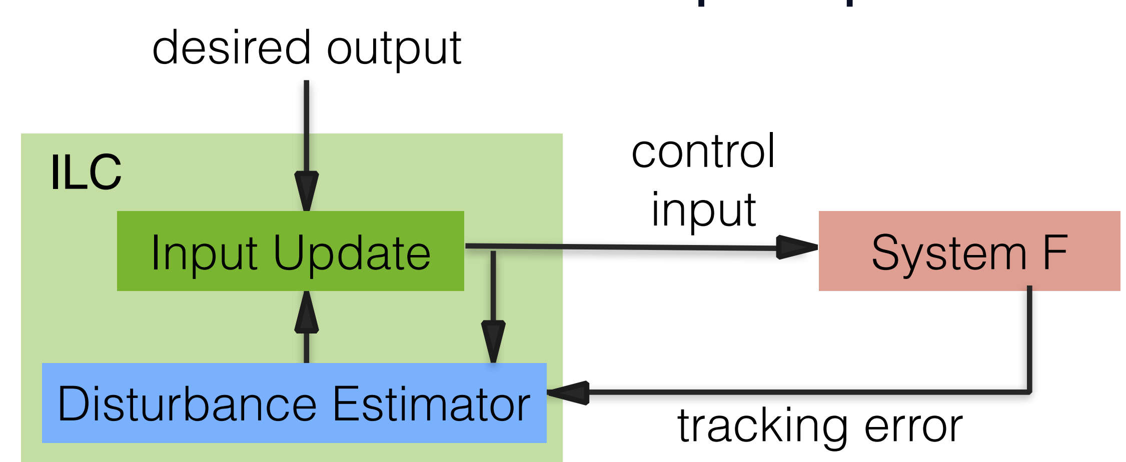
- **Q**uadratic cost criterion **ILC**
- Deterministic system model



J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," Automatica, vol. 36, pp. 641–657, 2000.

K-ILC

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



A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization- based iterative learning for precise quadrocopter trajectory tracking," Autonomous Robots, vol. 33, no. 1-2, pp. 103–127, 2012.

Comparison of Input Update

DYNAMIC SYSTEMS LAB

QILC	K-ILC

A Error prediction

QILC

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

➤ $\Delta u_{j+1} = u_{j+1} - u_j$

K-ILC

$$\bar{e}_{j+1} = \underbrace{F u_{j+1} - y_d}_{\text{nominal model error}} + \hat{d}_{j+1}$$

noise filtering

B Input update cost function

QILC

$$J_{j+1} = \bar{e}_{j+1}^T W_e \bar{e}_{j+1} + \Delta u_{j+1}^T W_{\Delta u} \Delta u_{j+1}$$

K-ILC

$$J_{j+1} = \bar{e}_{j+1}^T W_e \bar{e}_{j+1}$$

Parameters Defining the Algorithms

DYNAMIC SYSTEMS LAB

QILC | K-ILC

Parameters

$$J_{j+1} = \bar{e}^T_{j+1} \underline{W_e} \bar{e}_{j+1} + \Delta u^T_{j+1} \underline{W_{\Delta u}} \Delta u_{j+1}$$

noise filtering

$$d_{j+1} = d_f + \omega_j$$
$$y_j = F u_j + d_j + \mu_j,$$

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

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

$$\begin{aligned} S_j &= P_j + E_j \\ \underline{K_j} &= S_j (S_j + H_{j+1})^{-1} \\ P_j &= (I - K_j) S_j . \end{aligned}$$

2 Weighting Matrices

3 Covariance Matrices

Quadratic Norm Allows an Explicit Comparison

$$^{QILC}u_{j+1} = u_{nom} - \sum_{i=1}^{j} {}^{QILC}L e_j$$

$$^{K-ILC}u_{j+1} = u_{nom} - \sum_{i=1}^{j} {}^{K-ILC}L_j e_j$$

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

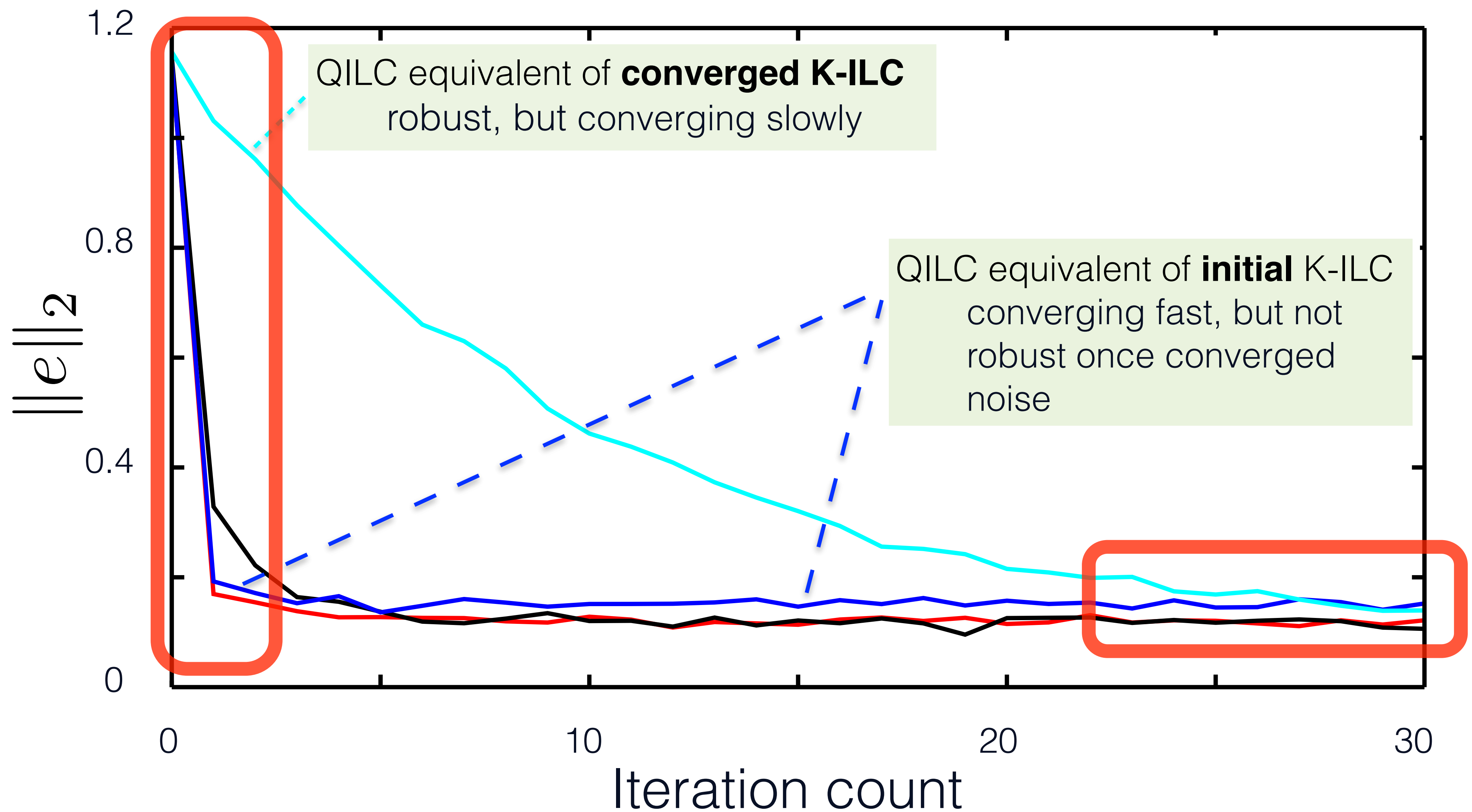
$$^{QILC}L = (\overset{0}{\cancel{W_{\Delta u}}} + F^{\mathrm{T}} W_e F)^{-1} F^{\mathrm{T}} W_e \; \boxed{= F^{-1}}$$

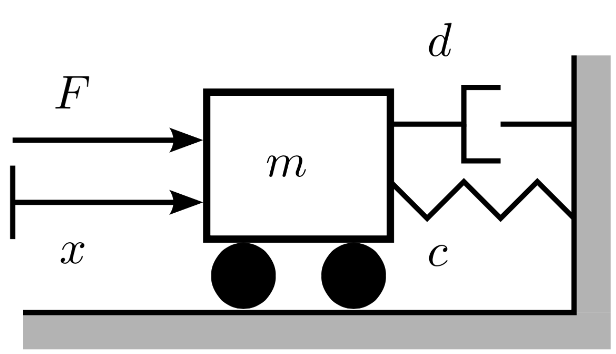
$$^{K-ILC}L_j = F^{-1} \overset{I}{\cancel{K_j}} \quad \boxed{= F^{-1}}$$

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

➤K-ILC **optimises** gain for **every iteration**

Mass-Spring-Damper Simulation Example





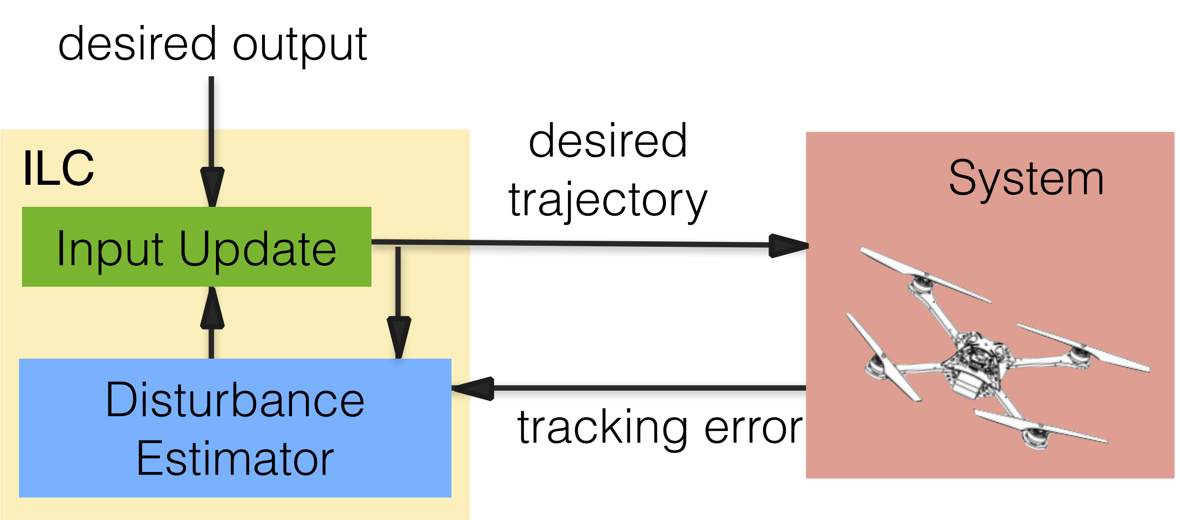
K-ILC designed for the problem
QILC designed for the problem
QILC equivalent of converged K-ILC
QILC equivalent of initial K-ILC

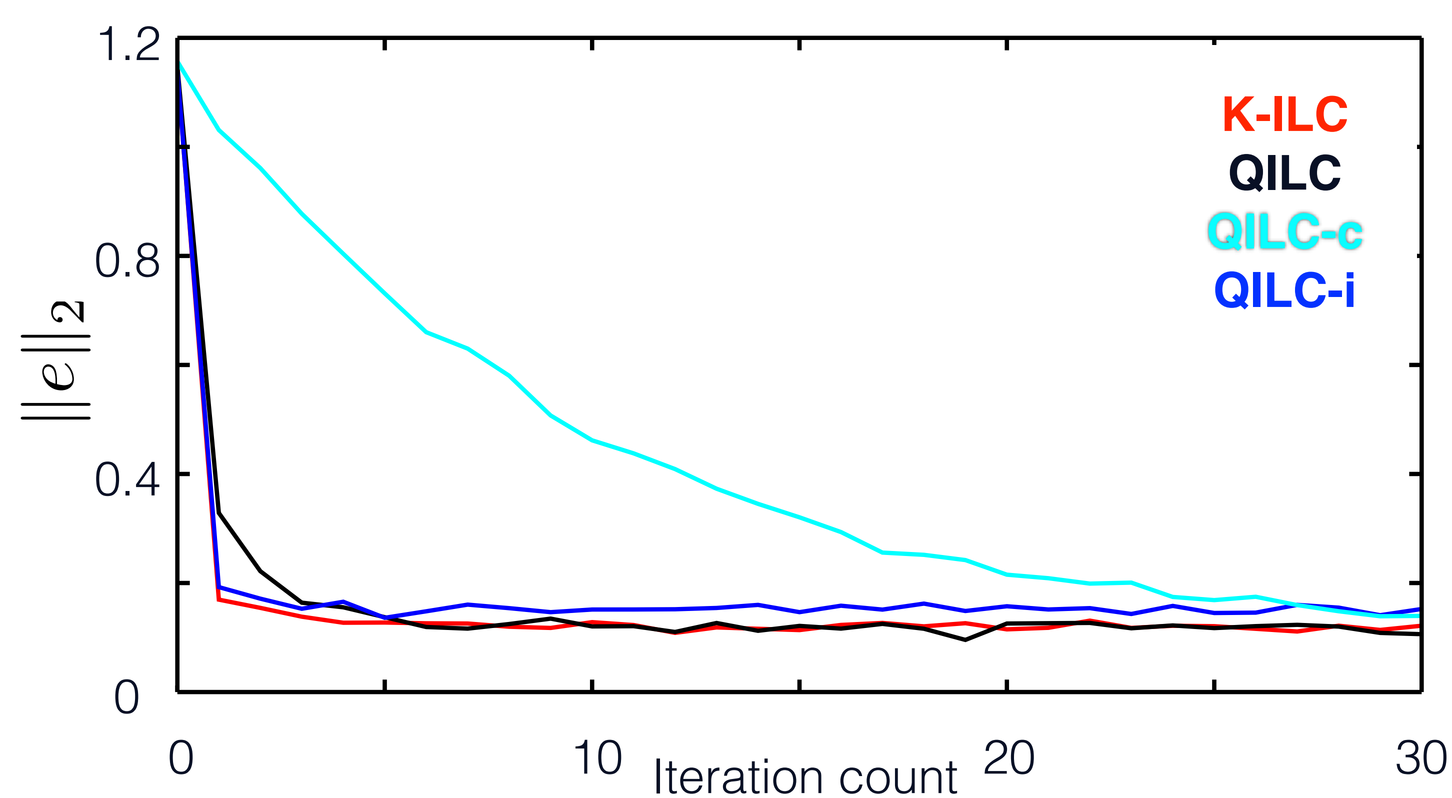
Advantages of K-ILC Algorithm

DYNAMIC
SYSTEMS LAB

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





Thank you!

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

53rd Conference on Decision and Control, 2014
Los Angeles

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

DYNAMIC
SYSTEMS LAB