MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Iterative Learning Control for Trajectory Tracking with Model Mismatch

Lihan Lian, Hangfei Li

Apr 23rd, 2024

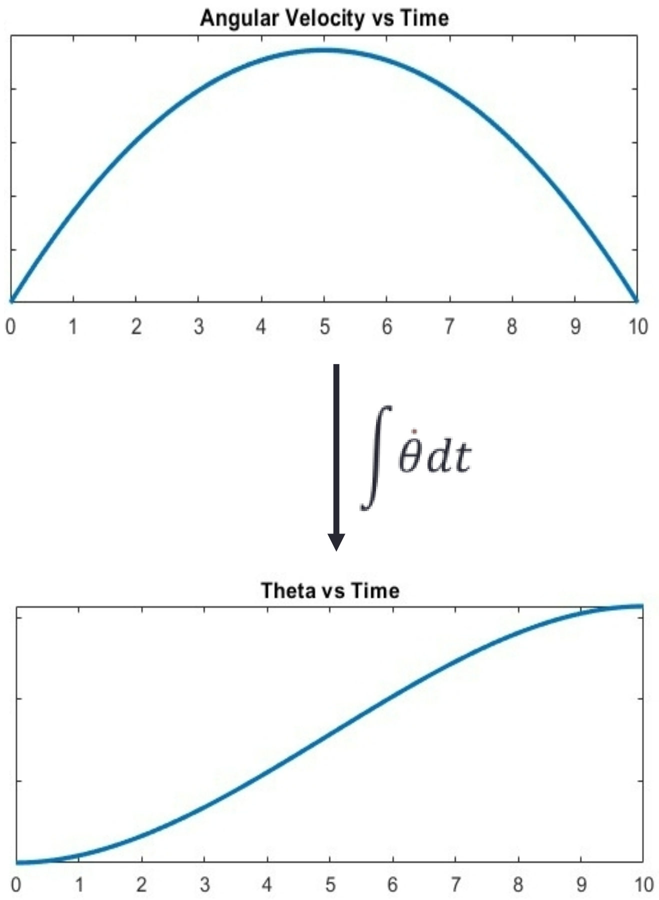
Introduction

- Mobile robots to transport materials, production parts and tools
- Start from warehouse to production line and collect empty carts back to warehouse
- Moving path can be designed and Trajectory tracking is important
- Explore the effects of different methods for trajectory tracking

Fig 1. Wheeled robot in the warehouse

Problem Statement

- Trajectory reference



$$\int \dot{\theta} dt$$

Fig 2.1. Reference Trajectory (ω, θ)

$$x = cos\theta$$
$$y = sin\theta$$

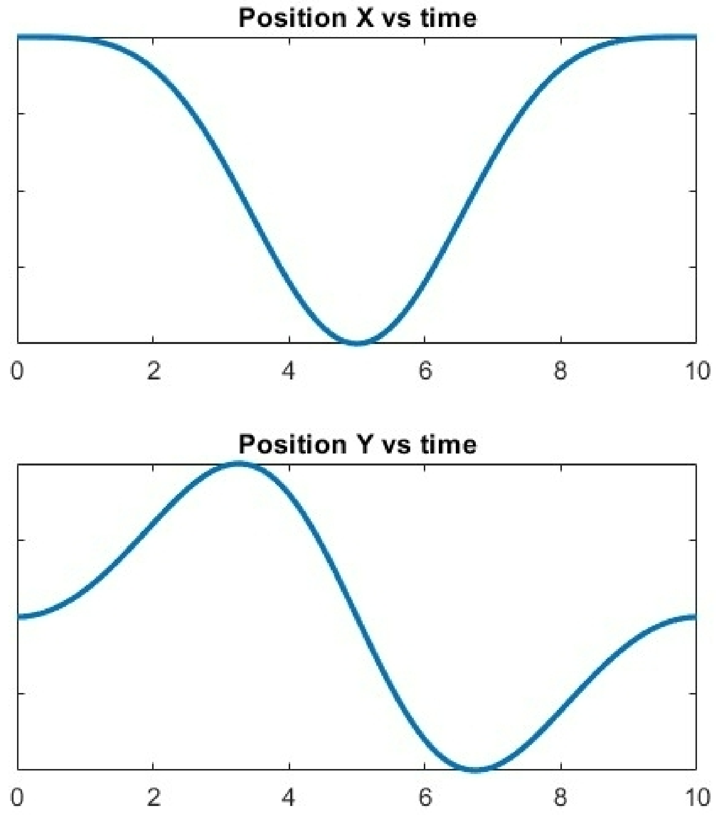
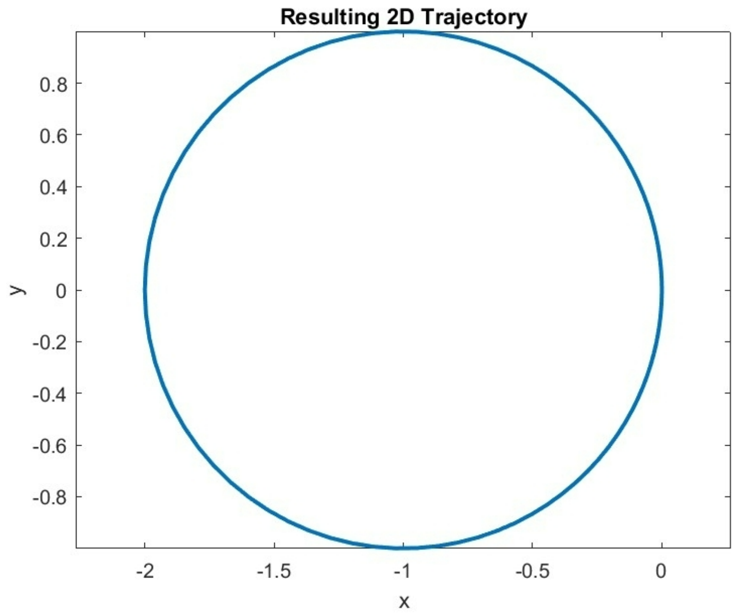


Fig 2.2. Reference Trajectory (XY)



$$(x+1)^2 + y^2 = 1$$

Fig 2.3. Reference Trajectory (X-Y Plane)

Problem Statement

- 2D double integrator model

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Fig 3.1. State Space Expression

- Actual model

$$A_{actual} = \begin{bmatrix} 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig 3.2. Actual Model Matrices

- Nominal Model (Idealized)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Fig 3.3. Nominal Model Matrices

Problem Statement

- Model Predictive Control (MPC)
- Direct collocation (DC) to generate initial control input
- Different strategies of Iterative Learning Control (ILC)

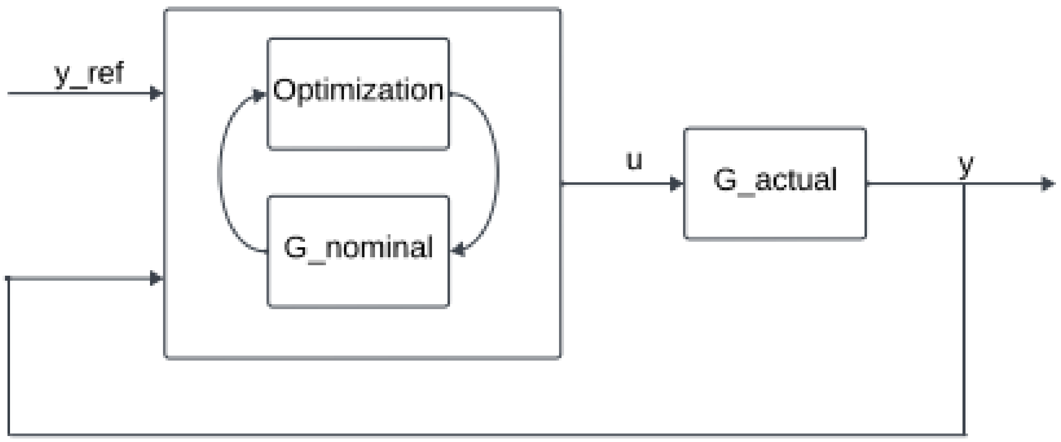


Fig 4. MPC Block Diagram

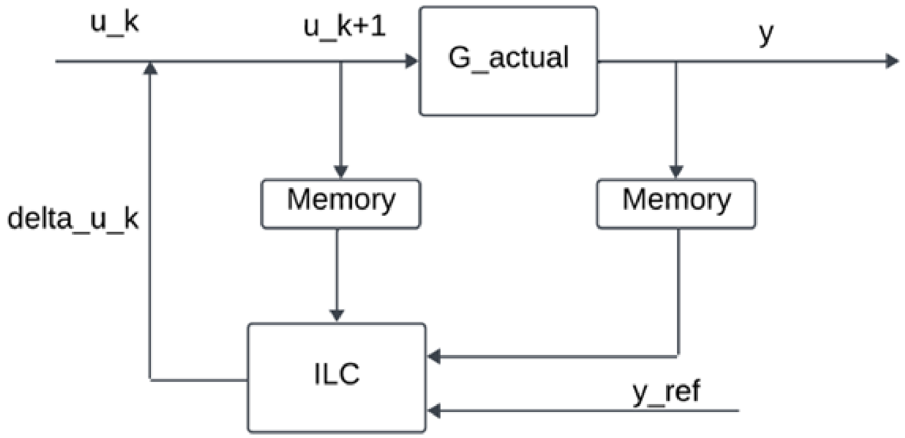


Fig 5. ILC Block Diagram

Method - MPC

- Require long prediction horizon and accurate model for good performance
- Cannot utilize history of control input and tracking error from previous iteration.
- Use nominal model for computing control input and use actual model for updating robot states.
- No error convergence.

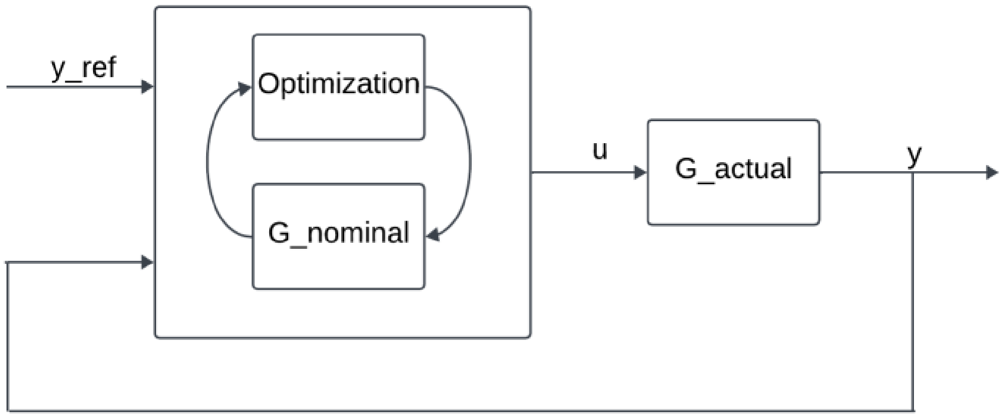


Fig 4. MPC Block Diagram

Algorithm 1 MPC

$x_k := x_0$

$J = (y_{ref} - y)^T Q (y_{ref} - y) + u^T R u$

for $k = 0 : N$ **do**

$y_k = C_{actual} x_k$

$u_k = mpc(A_{nominal}, B_{nominal}, C_{nominal}, J, y_{ref_k})$

$x_{k+1} = A_{actual} x_k + B_{actual} u_k$

$x_k = x_{k+1}$

$k = k + 1$

Fig 6. MPC Tracking Algorithm

Method - MPC

- Require long prediction horizon and accurate model for good performance
- Cannot utilize history of control input and tracking error from previous iteration.
- Use nominal model for computing control input and use actual model for updating robot states.
- No error convergence.

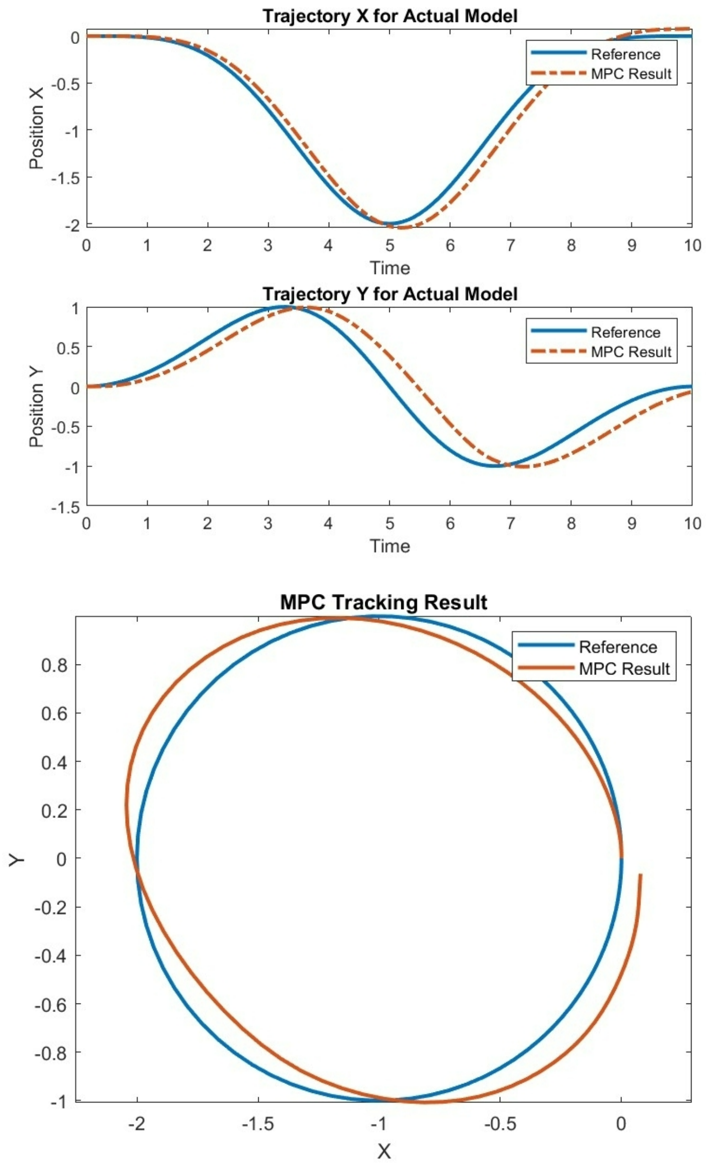


Fig 7. MPC Tracking Result

Method - ILC

- Use direct collocation to generate initial control input.
  - Computed off-line.
  - Use 5 waypoints for constraints.
- ILC is used to iteratively improve the control input u
  - provides the only control input (No feedback controller)
  - $u_k = u_1 + \Delta u_1 + \ldots \Delta u_{k-1}$

$$
\begin{aligned}
\min_{u_{k:N-1}} \quad & \sum_{k=0}^{N-1} (u_k^T u_k) \\
\text{s.t.} \quad & x_{k+1} = A_{nominal} x_k + B_{nominal} u_k, \quad k = 0, \ldots, N-1, \\
& u_{min} \leq u_k \leq u_{max}, \quad k = 0, \ldots, N-1, \\
& x_{N/4} = [-1, 1]^T, \\
& x_{N/2} = [-2, 0]^T, \\
& x_{3N/4} = [-1, -1]^T, \\
& x_0 = x_{start}, \\
& x_N = x_{goal},
\end{aligned}
$$

Fig 8. Direct Collocation Algorithm

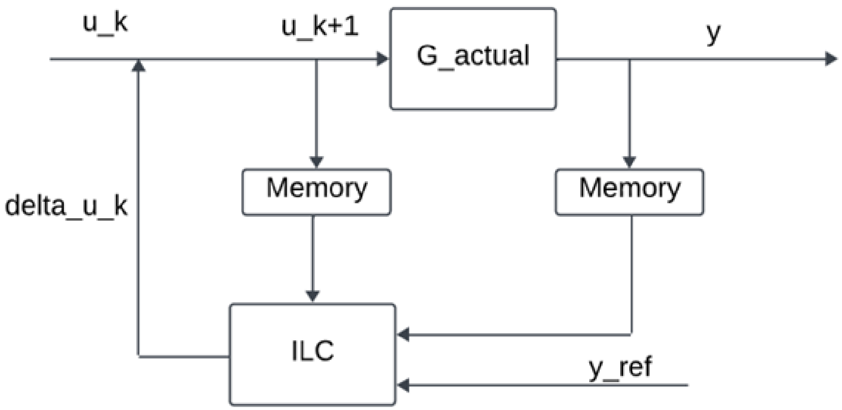


Fig 5. ILC Block Diagram

Method – ILC (Model-Based)

- Calculate Δu by solving constrained optimization problem.
  - Quadratic programming.
  - Linear quadratic regulator.
- Fast convergence rate and monotonically decreasing.
- Criterion of nominal model needs to be satisfied to guarantee convergence.

Algorithm 2 ILC-QP

$u := u_0$
$e_k = 1, k = 0$
$A_{eq} = G_{nominal}, B_{eq} = zeros$
$H = I$
while $e_k > 0.01$ **do**
  $y_k = G_{actual} u_k + d_{actual}$
  $e_k = y_{ref} - y_k$
  $\Delta(u) = quadprog(H, A_{eq}, B_{eq})$
  $u_k = u_k + \Delta(u)$
  $k = k + 1$

Fig 9. ILC-QP Algorithm

Algorithm 3 ILC-LQR

$u := u_0$
$e_k = 1, k = 0$
$A = I, B = -G_{nominal}$
$Q = 10 * I, R = 0.1 * I$
$K = dlqr(A, B, Q, R)$
while $e_k > 0.01$ **do**
  $y_k = G_{actual} u_k + d_{actual}$
  $e_k = y_{ref} - y_k$
  $\Delta(u) = -K e_k$
  $u_k = u_k + \Delta(u)$
  $k = k + 1$

Fig 10. ILC-LQR Algorithm

Method – ILC (Model-Based)

- Calculate Δu by solving constrained optimization problem.
  - Quadratic programming.
  - Linear quadratic regulator.
- Fast convergence rate and monotonically decreasing.
- Criterion of nominal model needs to be satisfied to guarantee convergence.

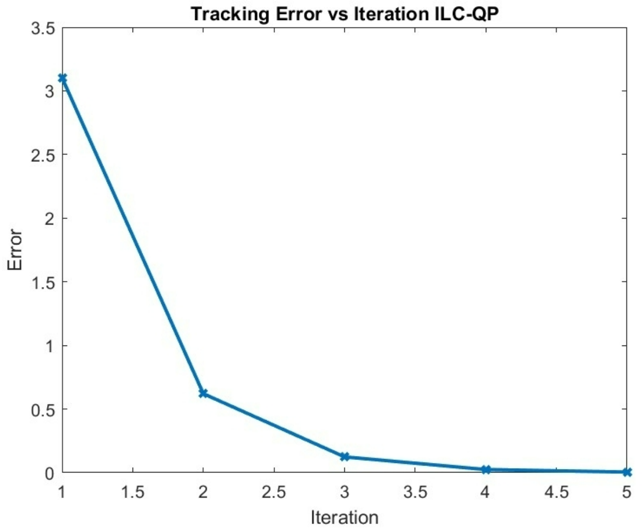


Fig 11. ILC-QP Convergence Result

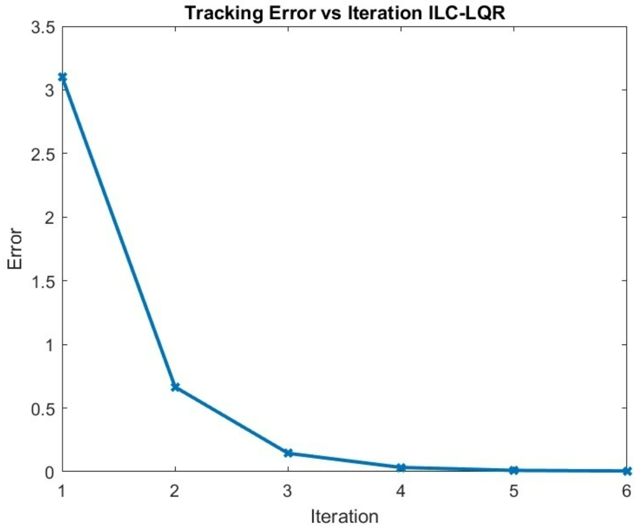


Fig 12. ILC-LQR Convergence Result

Method – ILC (Model-Free)

- Gradient-Based, require system adjoint G^*.
- Given the system is linear, vector d_{actual} is zero.
- τ is the inverse time operator to calculate the matrix adjoint, α is the learning rate.
- Lose monotonicity, slower convergence rate.

Algorithm 4 ILC-MF

$u := u_0, u_{prev} = 0$
$e = 1, k = 0$
$G^*e_{prev} = 0, \tau = fliplr(I)$
while $e > 0.01$ **do**
  $y_k = G_{actual}u_k + d_{actual}$
  $e = y_{ref} - y_k$
  $G^*e = \tau G_{actual}\tau e$
  $\Delta h_i = G^*e - G^*e_{prev}$
  $\Delta u_i = u - u_{prev}$
  if $\Delta h_i^T \cdot \Delta u_i \geq 0$ **then**
    $\alpha = 0.01$
  else
    $\alpha = -\Delta h_i^T \cdot \Delta u_i / (\Delta h_i^T \cdot \Delta h_i)$
  $u_{prev} = u$
  $G^*e_{prev} = G^*e$
  $u = u + \alpha G^*e$
  $k = k + 1$

Fig 13. ILC-MF Algorithm

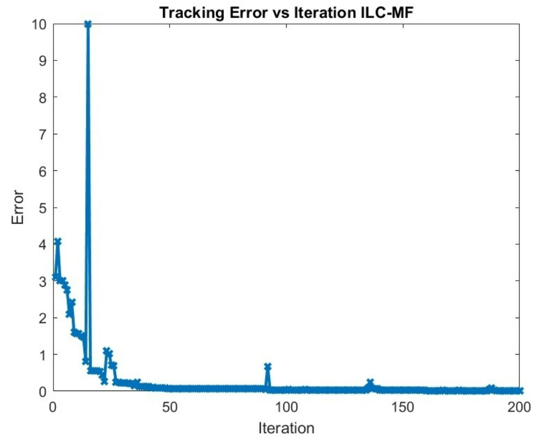


Fig 14. ILC-MF Convergence Result

Discussion

- Comparison on requirement of model accuracy and convergence rate.
- Share similarity with model-based policy gradient in the field of RL (DDPG)
- Criteria for guarantee of model-based ILC error convergence:

$$\left\| I - G_{actual} G_{nominal}^{-1} \right\| < 1$$

Comparison on performance

Controller	Model Required	Convergence Speed
MPC	Most Accurate	×
ILC-QP	Less Accurate	< 10
ILC-LQR	Less Accurate	< 10
ILC-MF	×	> 200

Table 1. Performance Comparison

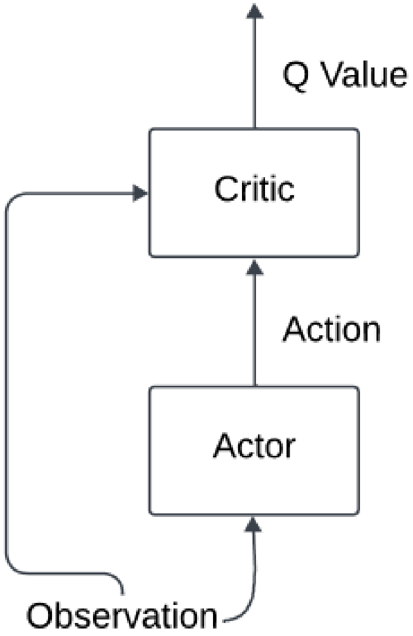


Fig 15. DDPG Block Diagram

Discussion

- Comparison on requirement of model accuracy and convergence rate.
- Share similarity with model-based policy gradient in the field of RL (DDPG)
- Criteria for guarantee of model-based ILC error convergence:

$$\left\| I - G_{actual} G_{nominal}^{-1} \right\| < 1$$

For actual system with plant model G_{actual}, the state update can be described as follows:

$$x_k = G_{actual} u_k + d_{actual} \quad (1)$$

$$x_{k+1} = G_{actual} u_{k+1} + d_{actual} \quad (2)$$

subtract (1) from (2) we can then get the error dynamics in iteration domain as shown in (5):

$$x_{k+1} - x_k = G_{actual}(u_{k+1} - u_k) \quad (3)$$

$$x_{k+1} - x_{ref} + x_{ref} - x_k = G_{actual} \Delta u_k \quad (4)$$

$$e_{k+1} = e_k - G_{actual} \Delta u_k \quad (5)$$

Here $\Delta u_k = u_{k+1} - u_k$, $e_k = x_{ref} - x_k$ and $e_{k+1} = x_{ref} - x_{k+1}$. In the previously discussed ILC-QP and ILC-LQR algorithms, they ultimately boil down

to satisfying the fundamental constraint.

$$e_k = G_{nominal} \Delta u \quad (6)$$

Assuming $G_{nominal}$ is invertible, then the follow relation can be obtained.

$$\Delta u = G_{nominal}^{-1} e_k \quad (7)$$

Substitute (7) into (5), the error dynamics in iteration domain can be rewrite as follows.

$$e_{k+1} = e_k - G_{actual} * G_{nominal}^{-1} * e_k \quad (8)$$

Thus, based on the theorem of contraction mapping, the condition for the error convergence is

$$||I - G_{actual} G_{nominal}^{-1}|| < 1 \quad (9)$$

and the error is guaranteed to be monotonically decreasing.

Fig 16. Model-Based ILC Convergence Proof

Conclusion

- Control input constraint (actuator limit) is not consider in the ILC iteration.
- Better model-free ILC can be investigated (use of learning function (L(q)) and Q-filter (Q(q)).
- Extension for nonlinear system or system with disturbance / noise.

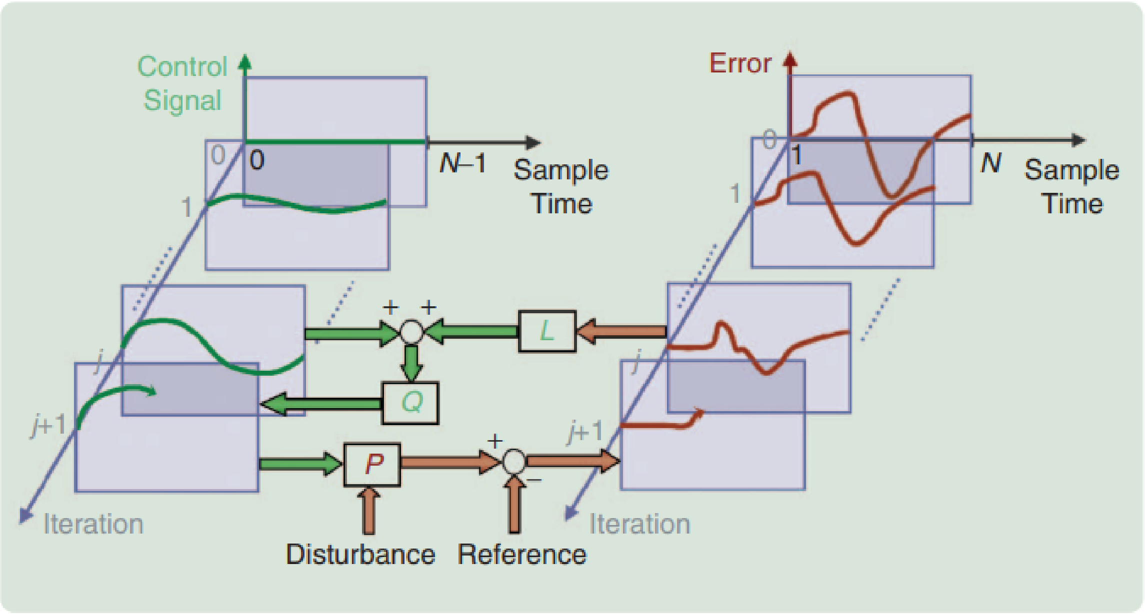


Fig 17. Schema of ILC Working Principle. (Bristow, Douglas A., Marina Tharayil, and Andrew G. Alleyne. "A survey of iterative learning control." IEEE control systems magazine 26.3 (2006): 96-114.)

Thank you !

Any questions ?