Iterative Learning of Feed-Forward Corrections for High-Performance Tracking

Fabian L. Mueller, Angela P. Schoellig, Raffaello D’Andrea
Institute for Dynamic Systems and Control
ETH Zürich, Switzerland
Iterative Learning of Feed-Forward Corrections for High-Performance Tracking

Fabian L. Mueller, Angela P. Schoellig, Raffaello D’Andrea
Institute for Dynamic Systems and Control
ETH Zürich, Switzerland
GOAL – Precise tracking of a desired output trajectory
GOAL – Precise tracking of a desired output trajectory

Example: Quadrotor vehicle
GOAL – Precise tracking of a desired output trajectory

Example: Quadrotor vehicle

Typical setup: Feedback control

Limitations of feedback control:
Disturbances and unmodelled dynamics (non-zero mean)
LEARNING APPROACH

Improve the performance over causal, feedback control by learning from a repeated operation.

Potential: Acausal action, anticipating repetitive disturbances.
LEARNING APPROACH

Do it again!

1. Dynamics model (here: from numerical simulation)
2. Disturbance estimation*
3. Update of input trajectory*

Prerequisites:

- **Coarse model** \( \mathcal{D} : U \rightarrow (Y, C) \)
- **Desired output trajectory** \( Y^* \) with corresponding nominal input \( (Y^*, C^*) = \mathcal{D}(U^*) \)
1 | DYNAMICS MODEL

Define:

- **Linear mapping** from *input deviations to changes in output and constrained variables*:
  
  \[ y = F u, \quad c = L u \]

  \[ u = U - U^*, \]
  
  \[ y = Y - Y^*, \quad c = C - C^* \]

From numerical dynamics simulation:

- Obtain \( F, L \) by running \( 2N \) identification runs
  
  - Apply \( U_{i+} = (u^*(0), \ldots, u^*(i) + \Delta u, \ldots, u^*(N - 1)) \rightarrow (Y_{i+}, C_{i+}) \)

  - Obtain

    \[ F(:, i) = (Y_{i+} - Y^*) / \Delta u \]
    
    \[ L(:, i) = (C_{i+} - C^*) / \Delta u \]
For each trial $j$, $j \in \{1, 2, \ldots \}$,

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

**Recurring disturbance** $d_j$.
Unknown. Only small changes between iterations:

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

**Noise** $\mu_j$.
Unknown. Changing from iteration to iteration.

$\mu_j, \omega_j$ — trial-uncorrelated, zero-mean Gaussian noise

From trial to trial our knowledge about $d_j$ improves.
UPDATE OF DISTURBANCE ESTIMATE via Kalman filter in the iteration domain:

Prediction step:
\[ d_{j+1} = d_j + \omega_j. \]

Measurement update step:
\[ y_j = F u_j + d_j + \mu_j. \]

Obtain \( \hat{d}_{j+1} \).
INPUT UPDATE via convex optimization:
minimizes the expected tracking error in the next trial:
\[ E[y_{j+1}|\text{all past measurements}] = F u_{j+1} + \hat{d}_{j+1}. \]

\[
\min_{u_{j+1}} \left\| F u_{j+1} + \hat{d}_{j+1} \right\|_p \quad p \in \{1, 2, \infty\}
\]
subject to
\[ L u_{j+1} \preceq c_{\max} \]

Obtain \( u_{j+1} \).
EXPERIMENTAL RESULTS

Desired position

CONTROL

Measured position

Angela Schoellig
VIDEO: https://youtu.be/zHTCsSkmADo?list=PLC12E387419CEAFF2

Quadrocopter Slalom Learning
CONCLUSIONS

• Learning algorithm combines **model data** with **experimental data**

• Convergence in around 5-10 iterations

**Repetitive** error components can be effectively compensated for by learning from past data.

Result is an **improved tracking performance.**