

機械学習とシステム同定の橋渡しによる繰り返し学習制御の新展開（延長）

代表研究者

大西 亘

東京大学大学院工学系研究科電気系工学専攻 准教授

1. 序論

ムーアの法則に牽引される形で近年の精密位置決め装置の位置決め精度と位置決め速度は目覚ましい発展を遂げている [1]。半導体露光装置は 2022 年時点で 3nm プロセスのチップが大量生産体制に入っているという報道 [2] があり、これからの制御にはより一層の精度と速度が求められる。また、半導体露光装置に限らず、より正確に・より速く・より安いコストで制御を行うことは次世代のモーションシステムに期待されている [3]。

制御を行う上で欠かせないこととして、フィードバック (FB) とフィードフォワード (FF) 制御が存在する。安定性と外乱抑圧を実現する上で FB 制御が重要な役割を果たす一方で、高速・高追従性を実現する上で大切になるのが FF 制御である。より高速で高精度な制御を実現するためには制御対象のより詳細なモデル化に基づく FF 制御が重要となる [4]。次世代のモーションシステムを扱うにあたり、剛体モードのみの考慮でも良い制御が実現できていた従来のモーションシステムよりも、変形モードを考慮した複雑なモデル化が必要である [3]。複雑なモデル化に伴いコントローラのハンドチューニングの難易度は飛躍的に上昇し、調整に多くの時間がかかる上最適な制御性能を実現することは難しい。そこでデータ駆動でコントローラのパラメータを自動調整する、「学習」と呼ばれる手法が近年注目を浴びている [5]。機械学習といったコンピュータサイエンスなどの分野で有名な手法を用いた例 [6-9] がある一方で、制御という分野の動的なダイナミクスを扱うことによるデータの少なさやロバスト性への厳しい要求水準により実際の適用事例は限定されている。そうした中、制御における活用事例の多いデータ駆動の学習手法は反復学習制御 (ILC) [10] と呼ばれる手法である。反復して同じ動作をする系に対象を限定することにより、極めて高速で高精度な制御を実現する。しかし、同じ動作が繰り返すという前提が崩れた際には性能が大幅に下落してしまう難点がある [11]。

ILC の研究が進み、弱点であったタスク柔軟性を解消する基底関数ベースのアプローチ (B-ILC) [11-13] も盛んに研究されている。しかし、同じ動作を反復する系に特化させた周波数領域ベースのアプローチ (F-ILC) と比較して追従性能が劣化してしまい、図 1 と図 2 に示されるタスク柔軟性と追従精度のトレードオフ関係が存在するのが現状であった [11]。本稿では基底関数ベースのアプローチと周波数領域ベースのアプローチを組み合わせた Combined ILC (C-ILC) を提案する。基底関数を用いることによりタスク柔軟性を確保した上で、基底関数によりモデリングしきれない部分を周波数領域ベースの ILC で補償することにより同じ動作の反復に対する高追従性を実現する。二次系のシステムのシミュレーションにおける既存の B-ILC と F-ILC との比較を通して提案手法のタスク柔軟性と高追従性の検討を行う。また、本報告書は文献 [14] から主に構成されていることを付記する。

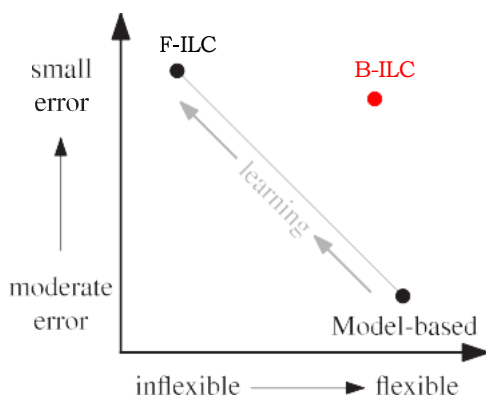


Fig.1: Comparison of F-ILC, B-ILC, model-based FF [11]

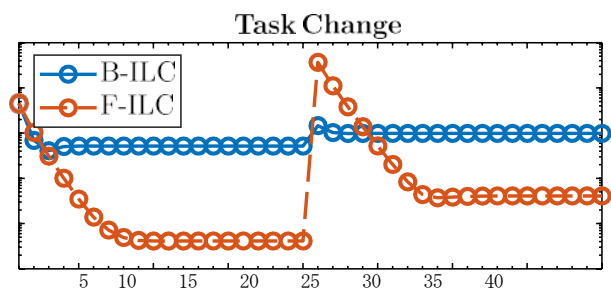


Fig.2: B-ILC performance robust to task change

2. 問題設定

問題設定のブロック線図を図 3 に示す。試行 j 回目における指令値 r_j 、制御入力 (FB 制御器) u_{fb} 、制御入力 (FF 制御器) f_j 、離散時間線形時不変システム G 、離散時間 FB 制御器 K とする。 G と K は試行間で不変であることを仮定し、事前に r_j を与えられているとした際の離散時間追従誤差 e_j を最小にすることを目的とする。

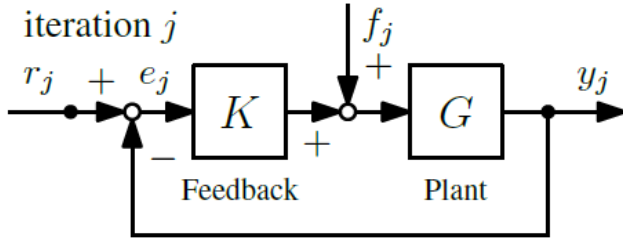


Fig.3: Block diagram of control configuration

3. 従来の反復学習制御 (ILC)

3-1. 周波数領域型反復学習制御 (F-ILC)

ILC の目的は、繰り返し行われる同じタスクに対し、サンプル点において出力を指令値に完全追従させることである。これは、タスクが行われるたびにそのタスクの出力誤差の情報から学習をし、FF 入力を調整することにより行われる。この学習行程が 3 試行目まで表したのが図 4 である。タスクの試行回数を j で表し一般化すると、FF 入力 f_j は

$$e_j = Sr - Gsf_j$$

$$f_{j+1} = Q(f_j + Le_j)$$

の式に従い逐次調整を行う。ここで、 S は感度関数 ($S = (1 + GK)^{-1}$) を表しており、 L は ILC の性能を決める学習フィルタと呼ばれる。

上式より、下記漸化式が得られるので、

$$e_{j+1} = Q(1 - GSL)e_j + (1 - Q)Sr$$

$$f_{j+1} = Q(1 - GSL)f_j + QLSr$$

全周波数領域内で $|Q||1 - GSL| < 1$ を満たしていれば、

$$e_\infty = \frac{(1 - Q)S}{1 - Q(1 - SGL)}r$$

$$f_\infty = \frac{QLS}{1 - Q(1 - SGL)}r$$

が成り立つ。ここで注目すべきことは $Q = 1$ とすることで最終的な追従誤差が 0 となり、なるべく $L = (GS)^{-1}$ となる様に設計することで学習の収束を早くすることができる点である。

ここで紹介した ILC は周波数領域ベースの Frequency domain ILC (F-ILC) と呼ばれる手法である。F-ILC は取得の容易な周波数応答データで試行領域の安定性を判別し、 L と Q を設計できることから、設計者の負担が軽く、学習のロバスト性を扱うことができる利点がある。一方で、タスクが試行毎に頻繁に変わってしまう対象とは相性が悪く、場合によっては FB 制御器 K のみを用いる場合よりも性能が落ちてしまう可能性がある。

3-2. 基底関数型反復学習制御(B-ILC)

B-ILC の背景

モデルベース FF 制御器は、高いプラントのモデル化が必要であるものの、高い追従特性を実現した上でタスクの変更に対し柔軟な応答を示すことが可能である [4]。そこで、この特徴に注目し、繰り返しのタスク毎

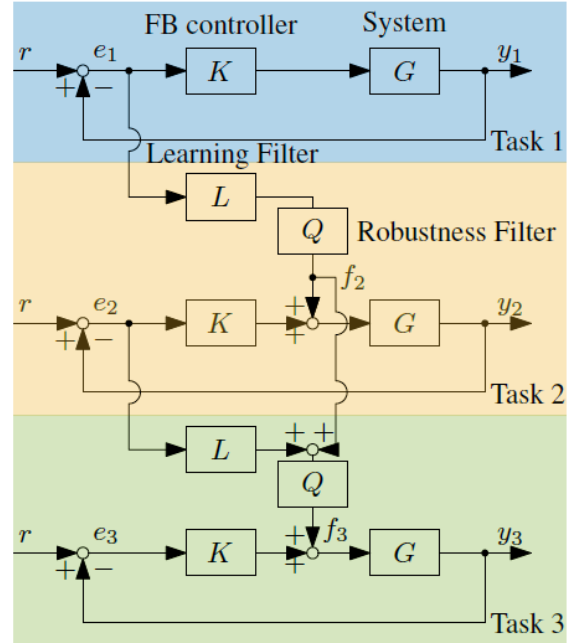


Fig.4: Frequency domain ILC update procedure

に FF 入力そのものではなく FF 制御器のパラメータをチューニングする基底関数ベースの Basis function ILC (B-ILC) が研究されている [11-13]。B-ILC の学習行程が 3 試行目まで表されているのが図 5 である。

実機に適用する際は、高次であるプラントを比較的低次にモデル化してしまうことから図 1 に図示するように最大のパフォーマンスは F-ILC に劣ってしまう。しかし、モデルベースの変化への柔軟性とデータ駆動の高性能・自動調整の両方の恩恵を受ける手法であり、図 2 の様にタスクの変化に影響されにくい結果を示す [11]。

B-ILC のアルゴリズム

B-ILC は試行 $j + 1$ における予測誤差の 2 ノルム $\|\hat{e}_{j+1}\|^2$ 等で表される下式の評価関数 $V(\theta_{j+1})$ を最小化することを目的として FF 制御器 $F(\theta_{j+1})$ ($f_{j+1} = F(\theta_{j+1})r_{j+1}$) のパラメータ θ_{j+1} をチューニングする。

$$V(\theta_{j+1}) = \|\hat{e}_{j+1}\|^2, \quad \theta_{j+1} = \arg \min_{\theta_{j+1}} V(\theta_{j+1})$$

上式より、評価関数を最小とする様な θ_{j+1} は $F(\theta_{j+1})$ がプラントの逆 G^{-1} に最も近づく時である。この際の FF 制御器の F の分子と分母の次元を決定するのが下式における基底関数 Ψ^A , Ψ^B である。

$$\begin{aligned} F(\theta_j) &= B^{-1}(\theta_j)A(\theta_j), & \theta_j &= [\theta_j^A, \theta_j^B]^T, \\ A(\theta_j) &= \Psi^A \theta_j^A, & B(\theta_j) &= I + \Psi^B \theta_j^B \end{aligned}$$

$A(\theta_j)$, $B(\theta_j)$ はそれぞれプラント G の予測モデル \hat{G} の分母と分子と捉えることもできる。

なお、本論文では基礎検討のため $\Psi^B = 0$, $r_{j+1} = r_j = r$ として議論する。 $\Psi^B \neq 0$ の場合には解析解を用いた議論を行うことができず、数値解を用いた議論を行う必要があるため詳細は [11] を参照されたい。

下式を定義すると、 \hat{e}_{j+1} を得られる。なお、ここにおける \widehat{SG} は SG の予測値とする。

$$\tilde{e}_j = e_j + \widehat{SG} f_j, \quad \Phi = \Psi^A \widehat{SG} r$$

$$\hat{e}_{j+1} = \tilde{e}_j - \Phi \theta_{j+1}$$

従って、最小値を満たすのは $\frac{\partial V(\theta_{j+1})}{\partial \theta_{j+1}} = 0$ の時であることから、

$$\frac{\partial \hat{e}_{j+1}}{\partial \theta_{j+1}} = -\Phi$$

より、

$$\begin{aligned} \frac{\partial V(\theta_{j+1})}{\partial \theta_{j+1}} &= \left(\frac{\partial \hat{e}_{j+1}}{\partial \theta_{j+1}} \right)^\top (\tilde{e}_j - \Phi \theta_{j+1}) \\ &= 0 \end{aligned}$$

であることから、

$$\theta_{j+1} = \left(\left(\frac{\partial \hat{e}_{j+1}}{\partial \theta_{j+1}} \right)^\top \Phi \right)^{-1} \cdot \left(\frac{\partial \hat{e}_{j+1}}{\partial \theta_{j+1}} \right)^\top \tilde{e}_j$$

以上より、

$$\tilde{e}_j = e_j + \Phi \theta_j$$

と表すことで B-ILC のパラメータ更新式を導くことができる。

$$\theta_{j+1} = Q(\theta_j) \theta_j + L(\theta_j) e_j$$

4. 基底関数と周波数領域型を組み合わせた反復学習制御(C-ILC)の提案

図6に示す基底関数と周波数領域型を組み合わせた反復学習制御(C-ILC)を提案する。以下、周波数領域分と基底関数分の設計を説明する。

4.1 周波数領域 ILC 成分(f^{ILC})の設計

通常の周波数領域 ILC と異なり、基底関数 ILC 成分 f^θ の更新分も加味した周波数領域 ILC 成分の FF 入力 f^{ILC} の更新を行う。

$$\begin{aligned} e_j &= Sr - GS(f_j^{ILC} + Q_\theta f_j^\theta) \\ f_{j+1}^{ILC} &= Q(f_j^{ILC} + Le_j) + Q_\theta(f_j^\theta - f_{j+1}^\theta) \end{aligned}$$

更新式において第2項の $Q_\theta(f_j^\theta - f_{j+1}^\theta)$ が追加されているのは、基底関数 ILC 成分 f^θ の学習した分の FF 入力を含むためである。そのため、実際に FF 入力を更新する際には $f^\theta \rightarrow f^{ILC}$ の順で更新を行う。

上式を用いることにより誤差と FF 入力の漸化式を得る。

$$\begin{aligned} e_{j+1} &= Q(1 - GSL)e_j + (1 - Q)S(1 - Q_\theta GF(\theta_j))r \\ f_{j+1}^{ILC} &= Q(1 - GSL)f_j^{ILC} + QLS(1 - Q_\theta GF(\theta_j))r \\ &\quad + Q_\theta(F(\theta_j) - F(\theta_{j+1}))r \end{aligned}$$

従って、全周波数領域内で $|Q||1 - GSL| < 1$ と $\lim_{j \rightarrow \infty} (F(\theta_j) - F(\theta_{j+1})) = 0$ を満たすことで、

$$\begin{aligned} e_\infty &= \frac{(1 - Q)S}{1 - Q(1 - GSL)}(1 - Q_\theta GF(\theta_\infty))r \\ f_\infty &= \frac{QLS}{1 - Q(1 - GSL)}(1 - Q_\theta GF(\theta_\infty))r \end{aligned}$$

が成り立つ。

この設計を用いることで C-ILC は F-ILC と比較して学習後の誤差が $|1 - Q_\theta GF(\theta_\infty)|$ 倍になり、 $Q_\theta \approx 1$ と $F(\theta_\infty) \approx G^{-1}$ を仮定することで $e_\infty \approx 0$ となることが分かる。

4.2 基底関数 ILC 成分(f^θ)の設計

周波数領域 ILC 成分 f^{ILC} の存在下では、誤差 e_j を最小化することは $f_j^\theta = G^{-1}r$ に繋がらない。つまり、FF 制御器 F はプラント G のパラメータを正しく学習することができない。そのため、分かりやすさのため $Q_\theta = 1$ とし、下式で表される e_j^θ を定義する。

$$\begin{aligned} e_j^\theta &= e_j + SGf_j^{ILC} \\ &= Sr - GSf_j^\theta \end{aligned}$$

これに伴い、下式で表される新たな評価関数を最小化することを目的とすることで FF 制御器 F_{j+1} のパラメータ θ_{j+1} をチューニングし、 $f_j^\theta = G^{-1}r$ の学習を実現する。

$$\begin{aligned} V(\theta_{j+1}) &= \|\hat{e}_{j+1}^\theta\|^2, & \theta_{j+1} &= \arg \min_{\theta_{j+1}} V(\theta_{j+1}) \\ \tilde{e}_j &= e_j + \widehat{S}G(f_j^\theta + f_j^{ILC}), & \Phi &= \Psi^A \widehat{S}G r \\ \hat{e}_{j+1}^\theta &= \tilde{e}_j - \Phi\theta_{j+1} \end{aligned}$$

以上の評価関数を最小にするのは $\frac{\partial V(\theta_{j+1})}{\partial \theta_{j+1}} = 0$ の時であることから、

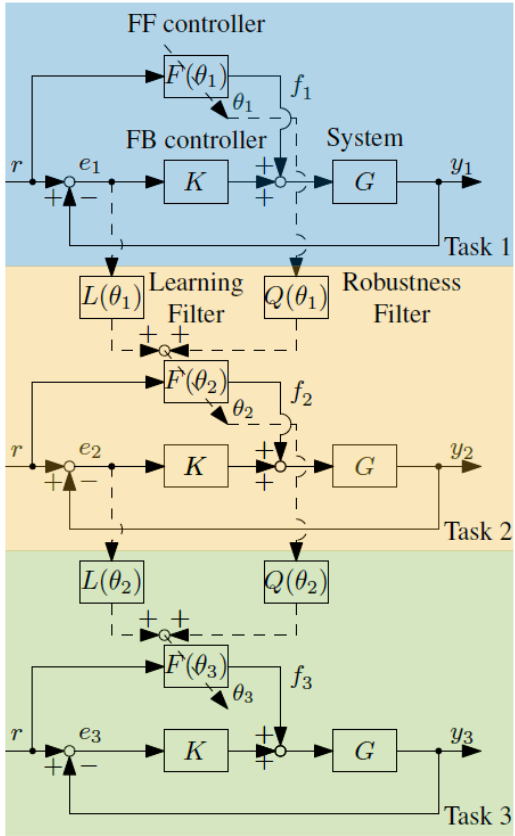


Fig.5: Basis function ILC update procedure. Realtime signals (bold line) and offline feedforward parameter update (dashed line) are depicted.

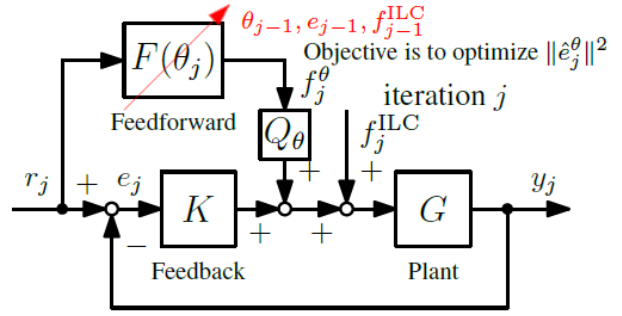
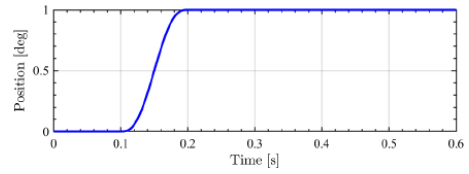
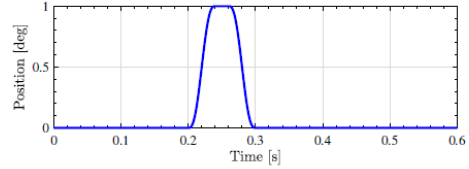


Fig.6: Proposed Combined ILC framework



(a) Trajectory 1



(b) Trajectory 2

Fig.7: Reference trajectory r_j . Trajectory 1 and 2 are used for iteration $j = 1 \sim 20$ and $j = 21 \sim 40$ respectively.

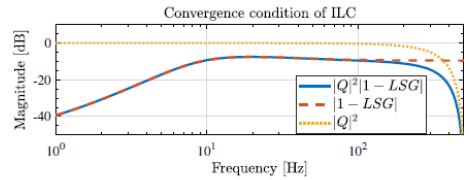


Fig.8: Employed Q-filter and convergence condition of F-ILC and C-ILC

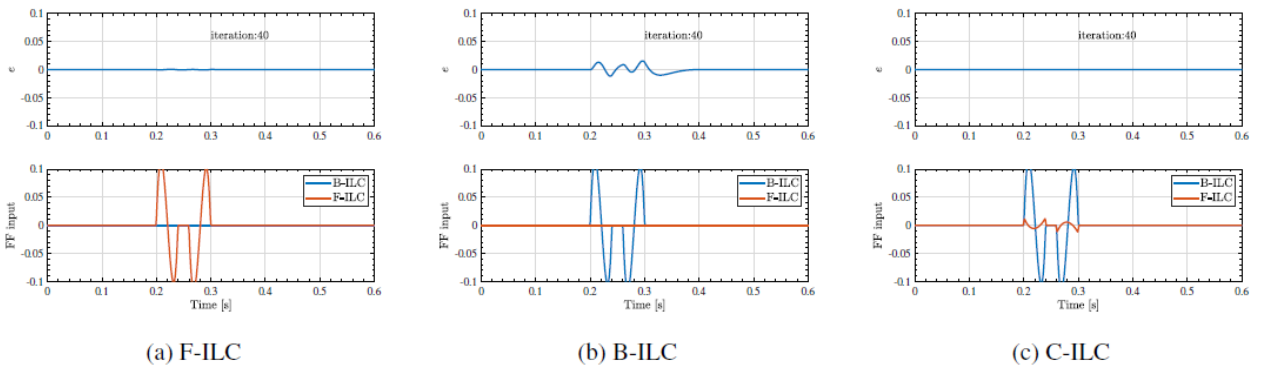


Fig.9: Error and feedforward value of final iteration for each method

$$\frac{\partial \hat{e}_{j+1}^{\theta}}{\partial \theta_{j+1}} = -\Psi^A \widehat{S}G r$$

を得る。導出の詳細は[14]を参照されたい。以上より、

$$\tilde{e}_j = e_j + \Phi \theta_j + \widehat{S}G f_j^{ILC}$$

と表すこと C-ILC のパラメータ更新式を得る。

$$\theta_{j+1} = Q(\theta_j)\theta_j + L(\theta_j)(e_j + \widehat{S}G f_j^{ILC})$$

5.2 次系の数値実験による C-ILC のタスク柔軟性・高追従性の検証

5.1 制御対象とシミュレーション条件

連続系で $G_c = \frac{1}{Js^2 + Ds}$ と表されるプラントを仮定し、制御周期 $T_s = 1 \text{ ms}$ で零次ホールドされる 2 次系のプラント G を制御対象とする数値実験を通し F-ILC, B-ILC, C-ILC の性能比較を行う。 $J = 3 \times 10^{-5}$, $D = 3 \times 10^{-4}$ とし、ノミナル値は $J_n = \frac{2}{3}J$, $D_n = \frac{2}{3}D$ とする。つまり、離散時間プラント G とそのノミナルプラント G_n を下式として各種制御器を構成する。

$$G = \frac{0.016611(z + 0.9967)}{(z - 1)(z - 0.99)}, \quad G_n = \frac{3}{2}G$$

FB 制御器は $K = \frac{0.4821(z - 0.9638)}{z - 0.7467}$ で統一し、最大の追従精度とタスク柔軟性を検証するため図 7 に示される 2 つの軌道のうち、試行回数 $j = 1 \sim 20$ の間は軌道 1 を、試行回数 $j = 21 \sim 40$ の間は軌道 2 を用いる。

5.2 制御器の学習条件

3 章と 4 章より、学習を始めるにあたり、F-ILC ではロバストフィルタ Q と学習フィルタ L , B-ILC では基底関数 Ψ^A と FF 制御器初期パラメータ θ_1 , C-ILC では上記全てに加え性能保証フィルタ Q_{θ} が必要である。

本数値例では、図 8 に示される Q を用い、 $\widehat{S}G = (1 + KG_n)^{-1}$ から $L = \widehat{S}G^{-1}$ を用いる。図 8 から全周波数領域内で $|1 - GSL| < 1$ を満たしているため本来は $Q = 1$ とすることで完全追従を実現できるが、F-ILC と C-ILC の性能差を比較するためにあえて $Q \neq 1$ とおく。残りの変数は下式の通り設定しシミュレーションを行う。

$$\Psi^A = \left[\left(\frac{1 - z^{-1}}{T_s} \right), \left(\frac{1 - z^{-1}}{T_s} \right)^2 \right], \quad \theta_1 = [J_n, D_n]^T, \quad Q_{\theta} = 1$$

なお、本シミュレーションでは C-ILC に確実にタスク柔軟性を持たせるため、タスク変更時には周波数領域 ILC 成分の信号を 0 にリセットした。

5.3 シミュレーション結果

以上の条件で数値実験を行った結果が図 10 であり、各試行 j における追従誤差 e_j の 2 ノルムの値がプロットされている。結果より、繰り返し同じ指令値が与えられている間は C-ILC, F-ILC, B-ILC の順に高い追従性能を示しており、指令値の変更があった際には C-ILC と B-ILC はタスク変更に対する同様の耐性を示したのに対し F-ILC では大きな誤差が生じてしまっている。

各手法の最後の試行における誤差と FF 制御入力基底関数 ILC 成分と周波数領域 ILC 成分をプロットしたものが図 9 である。図より、C-ILC では B-ILC の基底関数 ILC 成分のみでは補いきれなかった部分を周波数領域 ILC 成分で補うことにより高い追従性能を実現していることが分かる。

また、B-ILC と C-ILC の基底関数 ILC 成分の学習状況を示したものが図 11 である。シミュレーション上での FF パラメータ θ の学習が一致することにより図 14 において B-ILC と C-ILC が同等のタスク柔軟性を得ていると考えられる。

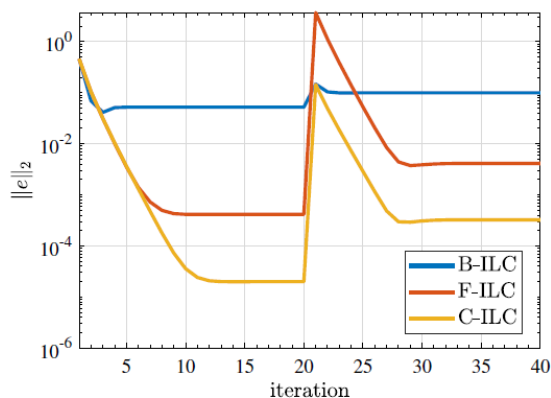


Fig.10: Comparison of F-ILC, B-ILC, and C-ILC tracking performance

最後に、図 10 における F-ILC と C-ILC の追従性能の違いについて、C-ILC の追従誤差は F-ILC の追従誤差の $|1 - GF(\theta_\infty)|$ 倍になる。図 12 から、概ね 20 dB の差があることが分かり、これが図 10 における F-ILC と C-ILC の誤差 2 ノルムの 10 倍ほどの性能差に繋がっていると考えられる。

6. 結論

本稿では、タスク柔軟で高精度な C-ILC の検討として基底関数と周波数領域設計を組み合わせ、従来の基底関数ベースのみの B-ILC、周波数領域ベースのみの F-ILC との比較検討を行った。C-ILC の基底関数 ILC 成分では周波数領域 ILC 成分の影響を除外する形で評価関数を組むことでタスク柔軟性を、周波数領域 ILC 成分では基底関数のみでは捉えきれないダイナミクスを補償することにより高追従性を確保した。また、二次系のシステムのシミュレーションにより C-ILC は B-ILC と同等のタスク柔軟性を持ち高精度であることに加え、F-ILC よりも高い追従性能を実現することが確認された。

[1] M. Steinbuch, T. Oomen, H. Vermeulen, “Motion Control, Mechatronics Design, and Moore’s Law,” *IEEJ Journal of Industry Applications*, vol. 11, no. 2, pp. 245–255, 2022.

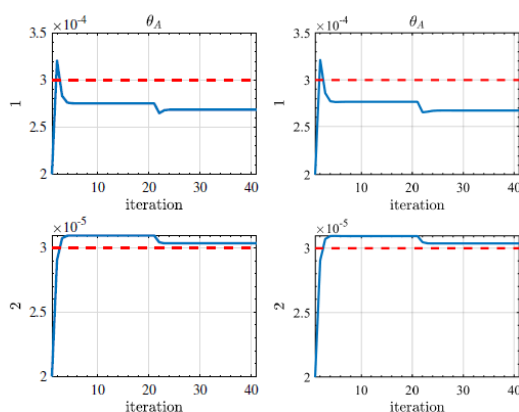
[2] “TSMC on track to move 3nm process to volume production in 2H22.” <https://www.digitimes.com/news/a20220415PD212.html> (2022 年 9 月 1 日閲覧)

[3] T. Oomen, “Advanced Motion Control for Precision Mechatronics: Control, Identification, and Learning of Complex Systems,” *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 127–140, 2018.

[4] W. Ohnishi, H. Fujimoto, and K. Sakata, “Model-based Control Techniques for Large-Scale High-Precision Stage,” *IEEJ Transactions on Electronics, Information and Systems*, vol. 140, no. 4, pp. 272–280, 2020.

[5] T. Oomen, “Learning in machines,” *Mikroniek*, vol. 6, pp. 5–11, 2018.

[6] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems—A survey,” *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.



(a) B-ILC (b) C-ILC

Fig.11: Parameter estimates of FF controller. Red dotted line for row 1 and row 2 each denote the theoretical value for continuous plant, $D = 3 \times 10^{-4}$ and $J = 3 \times 10^{-5}$.

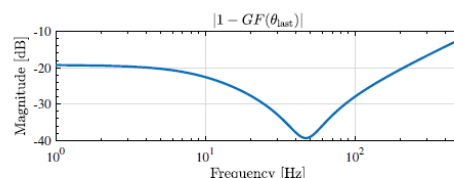


Fig.12: Theoretical tracking error improvement amount of using C-ILC instead of F-ILC

- [7] O. Sørensen, “Additive feedforward control with neural networks,” *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 1378–1383, 1999.
- [8] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, “Deep Learning and System Identification,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020.
- [9] J. Kon, D. Bruijnen, J. van de Wijdeven, M. Heertjes, and T. Oomen, “Physics-Guided Neural Networks for Feedforward Control: An Orthogonal Projection-Based Approach,” *2022 IEEE American Control Conference*, 2022, pp. 4377–4382.
- [10] A. Bristow, Douglas, M. Tharayil, and A. G. Andrew, “A survey of iterative learning control,” *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [11] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen, “Batch-To-Batch Rational Feedforward Control: From Iterative Learning to Identification Approaches, with Application to a Wafer Stage,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 826–837, 2017.
- [12] J. Van Zundert, J. Bolder, and T. Oomen, “Optimality and flexibility in Iterative Learning Control for varying tasks,” *Automatica*, vol. 67, pp. 295–302, 2016.
- [13] F. Boeren, L. Blanken, D. Bruijnen, and T. Oomen, “Optimal Estimation of Rational Feedforward Control via Instrumental Variables: With Application to a Wafer Stage,” *Asian Journal of Control*, vol. 20, no. 3, pp. 975–992, 2018.
- [14] 弦本健太郎, 大西亘, 古関隆章: 基底関数と周波数領域設計を組み合わせたタスク柔軟で高精度な ILC の基礎検討, 電気学会メカトロニクス制御研究会資料 (電気学会研究会資料), pp. 31-36, 2022.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
Non-causal state estimation for improved state tracking in iterative learning control	Modeling, Estimation, and Control Conference	2022/10
基底関数と周波数領域設計を組み合わせたタスク柔軟で高精度な ILC の基礎検討	電気学会メカトロニクス制御研究会	2022/9
Task flexible and high performance ILC: Preliminary analysis of combining a basis function and frequency-domain approach	IFAC World Congress	2023/7