Extension of Learnable Bandwidth in Iterative Learning Control

Danwei Wang* and Bin Zhang

Abstract—This paper describes frequency domain formulation of iterative learning control systems and study on their convergence along the operation axis. The concepts of learnable bandwidth and monotonic convergence are addressed and analyzed. It is shown that learnable bandwidth is a critical indicator for monotonic convergence and performance quality of the learning process. To achieve the good learning, various solutions are proposed to tune this learnable bandwidth along operation. There are two approaches, off line and online tuning along operation repetition axis. In this paper, some approaches are discussed to extend the learnable bandwidth in various domains are presented to show the potentials and effects of learnable bandwidth tuning. Some open problems are provided as well.

I. INTRODUCTION

Iterative Learning Control (ILC) is an approach to find appropriate control input for many industrial systems and machines that execute the same tracking task repeatedly. It aims to force the output of these systems to follow a trajectory $y_d(t)$ defined in a finite time duration $T$. Due to the presence of uncertainties and unmodeled dynamics, feedback control alone is difficult to fulfill this tracking requirement with high accuracy. ILC provides a feedforward channel to effectively solve this tracking problem.

Consider a single input single output linear time invariant system as follows:

$$\begin{cases} 
    x_j(k+1) = Ax_j(k) + Bu_j(k) + w_j(k) \\
    y_j(k) = Cx_j(k) + v_j(k)
\end{cases} \tag{1}$$

where the subscript $j$ indicates the operation iteration, $x(k) \in R^n$ the state vector, $y(k) \in R^p$ the output vector and $u(k) \in R^p$ the input vector, $w$ and $v$ the repeated state disturbance and output disturbance respectively.

The ILC update law has the general recursive form as:

$$u_{j+1}(t) = H(u_j(t), e_j(t)) \tag{2}$$

where $H$ is the ILC input update function; the tracking error $e_j(t) = y_d(t) - y_j(t)$ with $y_d(t)$ and $y_j(t)$ being the desired trajectory and practical trajectory of the $j$-th iteration, respectively. The objective is to make $e_j(t)$ converges to zero as iterations go to infinity. To describe $e_j(t)$ in an iteration with a finite time duration $T$, a certain norm of $e_j(t)$ is used. Therefore, ILC aims to achieve $\lim_{j \rightarrow \infty} ||e_j(t)|| \rightarrow 0$. Note that, we have two axis here. One is time axis in the range of $[0, T]$, while another one is trial axis or iteration axis $[1, \infty)$.

The convergence of tracking error in the sense of a given norm on the iteration axis is the focus of the study. It is well-known that ILC often shows bad learning behavior. That is, the tracking error goes down in the initial iterations but goes up again, usually to a very huge value, before it finally converges to zero. To overcome this bad learning behavior, the convergence of tracking error should be investigated in the $∞$-norm, which guarantees the monotonic decay of tracking error in the entire trajectory. Some examples include a learning in wave and the bisection method [1], an ILC with the adjustment of learning interval [2], a scheme based on unit step response data [3], an adaptive learning scheme with a learning gain matrices estimator based on the 2D model [4], an ILC together with an adaptive sliding mode control [5], an adaptive learning scheme with model uncertainties in a 2-dimension framework [6], basis function [7], model inverse based methods [8], high gain feedback schemes [9], etc.

In the frequency domain, a widely used method to achieve good learning behavior is to introduce a low-pass filter [10], [11], [12], [13]. However, ILC with such a filter no longer be able to achieve zero tracking since it cuts off high frequency components. If desired performance requires elimination of error components in high frequencies, this method results in poor tracking accuracy.

This paper formulates ILC problem in the time and frequency domains, as well as convergence in these two domains. We introduce the concept of learnable bandwidth and several schemes to extend the learnable bandwidth to achieve high tracking accuracy and good learning transient simultaneously. Experimental results of these schemes are presented to show the benefit of extension of learnable bandwidth. Finally, concluding remarks and open problems are mentioned for further research.

II. TIME DOMAIN FORMULATION OF ILC

The solution of (1) is:

$$y(t) = CA^t x(0) + \sum_{i=0}^{t-1} CA^{t-i-1} Bu_i + \sum_{i=0}^{t-1} CA^{t-i-1} w_i + v(t) \tag{3}$$

Considering a general one-step-ahead learning law

$$u_j(t) = u_{j-1}(t) + ke_{j-1}(t+1), \tag{4}$$

where $t \in [0, p - 1]$ with $p$ being the number of total sampling points of a given trajectory to be followed. Calculating the difference in two successive iterations and noting $y_j(t) - y_{j-1}(t) = -(e_j(t) - e_{j-1}(t))$, we have:

$$e_{j+1} = (I - kP)e_j = Qe_j \tag{5}$$

The authors are with School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. Corresponding author is Danwei Wang, phone: (65)-6790-5376; Fax: (65)-6795-3318; Email: edwwang@ntu.edu.sg
with 
\[ Q = \begin{bmatrix} 1 - kCB & 0 & \cdots & 0 \\ -kCAB & 1 - kCB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -kCAP^{-1}B & -kCAP^{-2}B & \cdots & 1 - kCB \end{bmatrix} \]
and \( e = [e(1), e(2), \cdots, e(p)]^T \).

Equation (5) represents a transfer function from the error at the iteration \( j \) to the error at the iteration \( j+1 \). Tracking error will converge to zero as iteration process provided \( Q \) has all eigenvalues less than one,

\[ |\lambda_Q(1 - kCB)| < 1 \]  

Condition (7) implies
\[ 0 < kCB < 2 \]  

Although it is easy to make this condition hold, this will, in most cases, cause bad learning behavior.

III. FREQUENCY DOMAIN FORMULATION OF ILC

Taking \( z \) transform of (1) for the \( j \)-th iteration:

\[ Y_j(z) = G(z)U_j(z) + T(z) \]  

where \( G(z) = C(zI - A)^{-1}B, \ T(z) = C(zI - A)^{-1}zxe(0) + C(zI - A)^{-1}W(z) + V(z) \).

The update law in the frequency domain has the form of

\[ U_j(z) = U_{j-1}(z) + kzE_{j-1}(z) \]  

After some simple algebraic operations, we have

\[ E_j(z) = (1 - kzG(z))E_{j-1}(z) \]  

Then, error decay at each frequency can be achieved if

\[ |1 - kzG(z)| < 1 \]  

holds for all frequencies. This frequency domain convergence condition has become a standard stability criterion. However, Longman et al. pointed out the difficulties to make this condition hold for all frequencies [14].

Definition: For an ILC system, the learnable-band collects all frequencies that ensure condition (12) be satisfied. The lowest frequency such that violates condition (12) is referred as the learnable-bandwidth.

For a desired trajectory, all components with frequencies inside the learnable-band can be learned with the proposed ILC. The learnable-bandwidth is normally used to set the filter frequency to divergency of tracking error with higher frequencies during the learning process.

To guarantee good learning transient, the frequency components entering the learning should be inside the learnable band. As mentioned earlier, a simple way to realize this goal is using a zero-phase low-pass filter with cutoff frequency lower than the learnable bandwidth. With a filter \( F(z) \), the learning law becomes

\[ U_j(z) = U_{j-1}(z) + kzF(z)E_{j-1}(z). \]  

Consequently, the condition for monotonic error decay is

\[ |1 - kzF(z)G(z)| < 1 \]  

Clearly, the introduction of low pass filter \( F(z) \), although guarantee the good learning behavior by preventing from high frequency components entering the learning law, causes the degradation of tracking accuracy. Therefore, it is desirable to widen the system learnable bandwidth such that good learning transient and high tracking accuracy can be achieved simultaneously. In the following sections, some effective and practical methods will be discussed.

IV. METHODS TO EXTEND LEARNABLE BANDWIDTH

A. Anticipatory-Type ILC

Frequency domain analysis shows that the main cause that (14) cannot hold true is the phase lag of \( G(z) \), especially in the high frequencies [15]. Therefore, phase compensation is important in learnable bandwidth extension. The concept of phase compensation has been used, at least implicitly, for many years. The D-type ILC [16], [17] provides a constant phase compensation of 90°. The limitation is that D-type ILC is insufficient to compensate phase lag in a wide frequency range. In addition, the digital derivative of error signal is often very noisy. This mitigates the benefit of phase compensation to a certain extent.

Another simple but efficient solution is to compensate for the phase lag by introducing a phase lead, which is realized by adding lead steps to the error signal [14], [16], [18], [15]. The learning law has the form of:

\[ u_j(t) = u_{j-1}(t) + kce_{j-1}(t + l), \]  

where \( l \) is the lead step and \( e_j(\Delta) = e_j(p) \) for \( \Delta > p \). The condition for monotonic error decay is:

\[ |1 - k\theta_l G(z)| < 1 \]  

Phase lead compensator \( z^l \) provides a phase lead \( \theta = \frac{l \omega}{2\pi} \times 180° \) to compensate for the system phase lag, especially at high frequencies. The closed-loop transfer function \( G(z) \) can be expressed as \( G(e^{j\omega}) = N_g(e^{j\omega}) \exp(j \theta_g(e^{j\omega})) \) with \( N_g(e^{j\omega}) \) and \( \theta_g(e^{j\omega}) \) being its magnitude characteristics and phase characteristics, respectively. Then, (16) becomes:

\[ 1 - kN_g(e^{j\omega})e^{j(\theta_g(e^{j\omega}) + \omega)} < 1 \]

Since \( k \) and \( N_g(e^{j\omega}) \) are both positive, we have:

\[ 0 < kN_g(e^{j\omega}) < 2 \cos(\theta_g(e^{j\omega}) + \omega) \]  

or, equivalently,

\[ 0 < k < \frac{2 \cos(\theta_g(e^{j\omega}) + \omega)}{N_g(e^{j\omega})} \]  

Condition (18) requires:

\[ |\theta_g(e^{j\omega}) + \omega| < 90° \]  

Note that at the frequency where phase angle crosses over \(-90°\) or \(90°\), the value of \( \cos(\theta_g(e^{j\omega}) + \omega) \) is zero, and
hence no positive gain can be found to meet the requirement of (18). To solve this problem, (19) is modified as follows:

\[ |\theta_\delta(e^{j\omega}) + k\omega| < 90^\circ - \epsilon \]  

(20)

where \(\epsilon\) is a positive constant for robustness. Since system model is normally not exactly known, it provides tolerance to model error. It can be seen that if condition (20) is satisfied, we can always find a \(k\) to satisfy (18).

Note that (19) involves only one design parameter \(l\) while the determine of gain \(k\) in (18) depends on \(l\). Thus, the lead step \(l\) can be determined first and then learning gain is chosen according to \(l\).

**B. Multi-Channel ILC**

![Fig. 1. Multi-channel learning control](image)

The anticipatory-type ILC usually only compensates for phase lag in a certain frequency range. For system with wider frequency range, it is almost always impossible to compensate for phase lag, especially in high frequencies, using a signal anticipatory type ILC. To compensate in a wider frequency range, a multi-channel method is presented here [19]. That is, more than one learning compensator work together in parallel. Each compensator compensates for phase lag in different frequency ranges and results in a wide-range phase compensator. Here, the phase compensator can be any ILC algorithm with phase compensation feature.

For this purpose, a new structure with \(n\) channels is proposed in Figure 1. The zero-phase channel filter \(F_i(z)\) defines a designated frequency band of the \(i\)-th channel, \(k_i\) a scalar learning gain for the \(i\)-th channel, and the \(i\)-th learning compensator \(\Phi_i(z)\) ensures the convergence of tracking error with phase compensation. The tracking error is separated into \(n\) parts corresponding to the designated bands/channels. These separated error parts are learned simultaneously in corresponding channels.

The overall ILC learning algorithm is:

\[ u_j(t) = \sum_{i=1}^{n} u_{i,j}(t) \]  

(21)

with the learning algorithm in the \(i\)-th channel as:

\[ u_{i,j}(t) = u_{i,j-1}(t) + k_i\Phi_i(F_i(e_{i,j-1}(t))) \]  

(22)

where error \(e_{i,j-1}(t)\) passing through channel zero-phase filter \(F_i(z)\) and channel compensator \(\Phi_i(z)\) before it is used to update the input for the channel.

In the frequency domain, this learning law is written as:

\[ U_j(z) = \frac{1}{n} \sum_{i=1}^{n} U_{i,j}(z) \]  

\[ = U_{j-1}(z) + \frac{1}{n} \sum_{i=1}^{n} k_iF_i(z)\Phi_i(z)E_{j-1}(z) \]  

(23)

The individual learning control laws in the \(i\)-th channel is

\[ U_{i,j}(z) = U_{i,j-1}(z) + k_i\Phi_i(z)F_i(z)E_{j-1}(z) \]  

(24)

Here, \(k_i\) and \(\Phi_i(z)\) must be chosen such that condition (14) satisfies in each channel individually. Then, the error contraction condition for multi-channel learning control is

\[ \left| 1 - G_p(e^{j\omega}) \sum_{i=1}^{n} k_i\Phi_i(e^{j\omega})F_i(e^{j\omega}) \right| < 1 \]  

(25)

In theory, learning can be done in an arbitrarily wide range of frequency if an appropriate learning compensator with an appropriate learning gain are chosen for each channel. In real applications, the learnable frequency will be limited by the frequency characteristics of the system, the desired trajectory and hardware limitation.

**C. Frequency Tuning ILC**

In practice, a trajectory may contain different frequency components at different time steps. For example, if the trajectory contains a sharp turn, the signal near the turning point contains many high frequency components and it is desirable to let this information enter the learning for a better performance. On the other hand, for those points only containing low frequency components, a low cutoff is suitable for better learning transient and long-term stability. According to the frequency components at each time step, an index dependent filter can be used [12].

In this method, a time- and iteration-varying filter \(F_j(t)\) is used in an anticipatory-type ILC as follows:

\[ u_{j+1}(t) = u_j(t) + kF_j(t)e_j(t + \beta) \]  

(26)

To illustrate the proposed method, an example is provided. Suppose signal \(e_j\) is preprocessed to eliminate frequency components above an estimated learnable bandwidth \(f_b\) to arrive at signal \(\tilde{e}_j\). This error signal \(\tilde{e}_j\) is decomposed by the wavelet packet algorithm on \(M\) levels to obtain \(2^M\) signals on different frequency regions. This series of signals is denoted as \(\tilde{e}_j^b\) with \(j\) being the cycle index and \(b \in [1, 2^M]\) being the index of frequency region. The frequency range \([0, f_b]\), which is the frequency bandwidth of signal \(\tilde{e}_j\), is evenly divided into \(2^M\) frequency regions. By comparison, the maximal frequency component of the decomposed signals at any time step \(\tilde{e}_{j,M}^b(t)\)

\[ \tilde{e}_{j,M}^b(t) = \max_{b \in [1, 2^M]} \tilde{e}_j^b(t) \]
D. Cutoff-Frequency Phase-In ILC

Different from the frequency tuning filter in section IV-C where cutoff frequency is determined according to time-frequency analysis of error signal, in cutoff-frequency phase-in method [12], the low pass filter \( F(z, t) \) has a cutoff frequency \( f(t) \) changing along the time axis in a trail follows a predefined profile. Here, subscript \( j \) is omitted because this profile does not change along iteration axis. The update law is:

\[
\begin{align*}
  u_{j+1}(t) &= u_j(t) + kF(z, t)e_j(t + 1) \quad (27)
\end{align*}
\]

The cutoff frequency of the low pass filter \( F(z, t) \) is illustrated in Figure 2 with a profile, which is designed with three parameters \( [f_1, f_2, f_3] \), referred to as initial cutoff frequency, estimated learnable bandwidth, and final cutoff frequency, respectively, on the frequency axis and two parameters \( [s_1, s_2] \) on the time axis. The parameters \( s_1 \) and \( s_2 \) divide an operation trial into three phases. Initial-phase is in \([0, s_1]\) and cutoff frequency changes from \( f_1 \) to \( f_2 \) along a parabola or line. Mid-phase corresponds to \([s_1 + 1, s_2]\) and cutoff frequency changes from \( f_2 \) to \( f_3 \) linearly. Final-phase lies in remaining steps of operation cycle \([s_2 + 1, p]\), and cutoff frequency stays at \( f_3 \) unchanged.

In the initial-phase, cutoff frequency \( f(t), t \in [1, s_1] \) has high values. This allows most error components, including of initial position offset and its influence in these early steps, enter learning. Hence, the influence of initial state error will be greatly suppressed in this phase. In the mid-phase, cutoff frequency \( f(t), t \in [s_1 + 1, s_2] \) is lower than the estimated learnable bandwidth. The influence of initial position offset in this phase is further suppressed. Because the estimated learnable bandwidth \( f_2 \) is inaccurate and the influence of initial condition decreases along the time axis, the linear decrease of cutoff frequency in this phase is helpful in maintaining good learning transient. In the final-phase, the influence of initial condition can be neglected. Cutoff frequency \( f(t), t \in [s_2 + 1, p] \) should make frequency domain condition hold for monotonic decay of error.

To describe cutoff frequency profile in a concise way, it is denoted as \( f_1 \text{Hz-}s_1\text{step-}f_2\text{Hz-}s_2\text{step-}f_3\text{Hz} \).

E. Cyclic Pseudo-Downsampled ILC

In this section, the extension of the learnable bandwidth is considered in the multirate signal processing domain.

1) Downsampled learning: Consider system given in (1) and a one-step-ahead learning update law given in (4). From previous analysis, the error transfer in two successive iterations is given by (5). To obtain the monotonic decay of error along the iteration axis, rather than using the \( \alpha \)-norm, we take the \( \infty \)-norm on both sides of (5) and arrive at:

\[
\|e_{j+1}\|_\infty \leq \|Q\|_\infty \|e_j\|_\infty \quad (28)
\]

Hence, the monotonic error decay requires

\[
\|Q\|_\infty \leq 1. \quad (29)
\]

If gain \( k \) is chosen such that \( (1 - kCB) > 0 \) and \( |1 - kCB| < 1 \), from (29) we have:

\[
|CB| \geq \sum_{i=1}^{p-1} |CA^iB| \quad (30)
\]

Condition (30) is only related to the system dynamics. For a discrete-time system with a given sampling rate, its Markov parameters are constants and condition (30) often cannot be satisfied. There is a hidden freedom - sampling rate - can be used to make (30) easier to be satisfied. For a continuous-time system \( A_c \), its zero order hold equivalent with the sampling period of \( T \) is [20]:

\[
A = e^{A_cT}
\]

If \( A_c \) is stable, all the eigenvalues of \( A \) are inside the unite circle [20]. If the sampling period \( T \to \infty \), then \( \lim_{T \to \infty} A \to 0 \). This makes condition (30) easier to satisfy. Therefore, suppose system with a sampling period \( T \) (feedback rate) cannot make (30) hold, the sampling period can be increased to \( mT \) (ILC rate), or downsampled \( m \) times (also known as sampling ratio), to force (30) hold.

Here, \( m \) is selected as an integer to simplify the processing of error signal. The limitation of downsampling learning is that it only takes care of the tracking accuracy on those downsampling points, while it ignores those sampling points in-between every two downsampling points [21]. Therefore, the tracking error on these in-between sampling points could be large and degrade the overall tracking performance.
2) Cyclic pseudo-downsampled ILC: A pseudo-downsampled ILC with time shift along the iteration axis is presented and is illustrated in Figure 3 to overcome the above mentioned limitations [22]. Note that, in different cycles, the first downsampling point has different sampling ratios. The sampling ratio for the first downsampling point \( r_j \) at the \( j \)-th iteration is calculated as:

\[
r_j = \begin{cases} 
\text{rem}(\frac{m}{mT}); & \text{if } \text{rem}(\frac{m}{mT}) \neq 0; \\
\text{rem}(\frac{m}{mT}); & \text{if } \text{rem}(\frac{m}{mT}) = 0
\end{cases}
\]

(31)

where \( \text{rem}(\cdot) \) is a function to get residual.

For instance, at the \( j \)-th iteration, suppose the first downsampling points has a sampling period of \( mT \) (i.e. \( r_j = 0 \)). Then, at the \( j + 1 \) iteration, the first downsampling point has a sampling period of \( T \) (i.e. \( r_{j+1} = 1 \)), and so forth. In every two consecutive cycles, all the downsampling points marked with solid points in Figure 3 have a time shift of \( T \). The set of downsampling points at each iteration always contains the first sampling point (the first solid point in the \( j \)-th and \( j + m \)-th iteration and circles in the \( (j+1) \)-th to \( (j+m-1) \)-th iterations). This way, the ILC output \( u(0) \) is always updated and this is desirable in the presence of initial state error.

With the above consideration, the number of downsampling points \( q_j \) at the \( j \)-th iteration is first given by:

\[
q_j = \begin{cases} 
\left\lfloor \frac{d}{m} \right\rfloor; & \text{if } \left\lfloor \frac{d}{m} \right\rfloor \neq \text{integer} \\
\left\lfloor \frac{d}{m} \right\rfloor; & \text{if } \left\lfloor \frac{d}{m} \right\rfloor \text{ is integer}
\end{cases}
\]

(32)

In Figure 3, note that downsampling points in iteration \((j+m)\) are the same with those in iteration \(j\). In this sense, it is cyclic on the iteration axis with a period of \( m \) iterations. The input for all sampling points is updated once every \( m \) iterations. The learning law is given in (33) in which 2) and 4) serve as zero-order holders. Other holder or interpolation method can be used as well.

- For the first input update point
  1. \( u_j(0) = u_{j-1}(0) + k\varepsilon_j^{-1}(r_j) \)
  2. \( u_j(t) = u_j(0) \) for \( r_j > 1 \) with \( i = 1, \ldots, r_j - 1 \)

- For the remaining input update points
  3. \( u_j(t) = u_{j-1}(t) + k\varepsilon_j^{-1}(t + m) \)
  4. \( u_j(t + i) = u_j(t) \) for \( i = 1, \ldots, m - 1 \) with \( i + j < p - 1 \)

(33)

The downsampled system dynamic of (1) with sampling period of \( gT \) \( (1 \leq g \leq m) \) is given by \( A_q = \sum_{g=1}^{d} \sum_{q=1}^{g} \frac{1}{T}A_q + B_q \). The system is available. A gain \( k \) is chosen such that

\[
k \leq \min_{1 \leq g \leq m} \frac{1}{|C_sB_g|}
\]

(34)

and \( m \) is chosen such that

\[
\min_{1 \leq g \leq m} |C_sAB_g| \geq |C_sB|,
\]

(35)

\[
|C_sB| \geq \max_{1 \leq g \leq m} \left( |C_sAq^{-1}B_g| \right) + \sum_{i=1}^{q-3} |C_sAq^{-3}B_i|, \quad \text{(36)}
\]

\[
|C_sAB| \geq \max_{1 \leq g \leq m} \left( |C_sA3q^{-3}B_g| \right) + \sum_{i=1}^{q-3} |C_sA3q^{-3}B_i| + \max_{(C_sB), |C_sAA3q^{-2}B_i|} \quad \text{(37)}
\]

In the implementation, suppose that a continuous-time system is discretized by a sampling period of \( T \) or an original discrete-time system with a sampling period of \( T \) is available. A gain \( k \) can always be selected to make (34) hold. Without downsampling, only (30) is applicable. If (30) does not hold, the sampling period is increased to \( 2T, 3T \), and so on until for a sampling period of \( mT \), conditions (35)-(37) are all satisfied. Then, update law (33) with this newly selected \( m \) is applied.

V. APPLICATIONS TO INDUSTRIAL ROBOT

Experiments are carried out on an industrial SCARA robot, SEIKO TT3000. One joint moving in the horizontal plane is used to test the proposed learnable bandwidth extension methods. The robot system has a sampling period of 0.01 second. The nominal model of the closed-loop joint is given as follows:

\[
G(z) = \frac{0.02277z}{z^2 - 1.659z + 0.683}
\]

(38)
A. Conventional ILC vs. Anticipatory-type ILC

To ensure the learning of high frequencies, condition (16) needs to be satisfied in high frequencies. When a signal anticipatory-type ILC is used, $l$ and $k$ are selected based on the design criteria in Section IV-A. The condition (18) with gain $k = 0.5$ and different lead steps is shown in Figure 4. It is clear to see that $l = 2$ has a wider learnable bandwidth up to 36Hz, comparing to the learnable bandwidth of 4Hz for a conventional one-step-ahead ILC with $l = 1$.

![Figure 4. Anticipatory-type ILC and Multi-channel ILC](image)

The Root Mean Square (RMS) errors of anticipatory-type ILC and a conventional one-step-ahead ILC are shown in Figure 5. It is clear to see the advantages of anticipatory-type ILC. For conventional ILC, the RMS error remains stable at 0.4° from the 30th iteration, while for anticipatory-type ILC, RMS error keeps reducing in the first 200 iterations and reaches about 0.037°. The improvement of tracking accuracy is remarkable.

![Figure 5. RMS error of Anticipatory-type ILC vs. Conventional ILC](image)

B. Anticipatory-type ILC vs. Multi-channel ILC

For channel 1, previous design suggests lead step of 2 with a learnable bandwidth of 36Hz. When the second channel is introduced, we choose lead step as 1. As shown in Figure 4, condition (17) is satisfied from 22Hz up. The cross point of curves $2 \cos(\theta_g(e^{j\omega}) + l\omega)$ for $l = 1$ and $l = 2$ is at 32Hz. Therefore, a low pass filter can be designed with a cutoff frequency of 32Hz. The error components in the frequency band $[0, 32]$Hz are suppressed by learning law in the channel 1, while the error components beyond that frequency band up to Nyquist frequency are suppressed by learning law in the channel 2.

![Figure 6. RMS error of anticipatory-type ILC vs multi-channel ILC](image)

The RMS tracking error of this two-channel ILC is shown in Figure 6. It shows that multi-channel ILC effectively improves the learning performance. This is because the single-channel anticipatory-type learns only the frequency components in the range $[0, 32]$Hz, compared with the wider learnable frequency band, $[0, 50]$Hz, of the proposed multi-channel learning scheme.

C. Anticipatory-type ILC vs. Frequency tuning ILC

In this experiment, the lead-step and gain are selected as 5 and 1, respectively. From Figure 4, the estimated learnable bandwidth is read as $f_b = 17$Hz. The decomposition level is set as 4. The experimental results are shown in Figure 7. We can see the RMS error of anticipatory ILC with fixed filter shows a very poor learning transient. It diverges from about the 100th cycle and we stop at the 600th cycle. On the contrary, the RMS error of the frequency tuning scheme remains stable and it continuously goes down after about 500 cycles. The tracking error finally reaches 0.0091°.

![Figure 7. RMS error compression of Anticipatory-type ILC vs. Frequency tuning ILC](image)
D. Conventional ILC vs. Cutoff-frequency Phase-in ILC

The parameter set is determined as \[ |k = 1, f_1 = 10Hz, f_2 = 4Hz, f_3 = 2Hz, s_1 = 10, s_2 = 200| \] [13]. Figure 8 shows that a conventional ILC with cutoff frequency 3Hz shows good behavior and final RMS error is about 0.100°. On the contrary, RMS error of our method converges monotonically and shows a much better tracking accuracy with final RMS error being about 0.0307°.

![RMS error comparison](image)

Fig. 8. RMS errors comparison

E. Conventional ILC vs. Cyclic Pseudo-downsampled ILC

The parameters learning gain \( k \) and sampling ratio \( m \) are determined as follows [22]:

- **Learning gain \( k \)** is selected as 0.5 in this experiment. With this learning gain, (34) is satisfied for sampling ratio \( m \) from 1 to 10.
- **Sampling ratio \( m \)**: Without downsampling, \( m = 1 \) (sampling period \( T = 0.01s \)), condition (30) does not hold. When downsamples to \( m = 5 \) (sampling period \( mT = 0.05s \)), all conditions (34)-(37) are satisfied. Hence, the value of \( m \) is chosen as 5.

The RMS errors are shown in Figure 9. It can be seen that for a conventional ILC, the RMS error reaches 0.134°. When the cyclic pseudo-downsampled ILC is employed, RMS error reduces to 0.007°. This is about one-twentieth of the RMS error of the conventional ILC.

![RMS errors for trajectory without offset](image)

Fig. 9. RMS errors for trajectory without offset

VI. CONCLUSIONS AND OPEN PROBLEMS

ILC is an efficient approach to achieve high accuracy in tracking. A major obstacle to its application, however, is the non-monotonic learning transient. In the frequency domain, this phenomenon is explained as high frequency error components beyond the system learnable bandwidth are amplified. Therefore, extension of the learnable bandwidth is critical to high tracking accuracy and monotonic learning transient. This paper surveys some methods for learnable bandwidth extension including linear phase compensation, multi-channel, cutoff frequency tuning, cutoff-frequency phase-in, and cyclic pseudo-downsampling. Their theoretical backgrounds and design criteria are discussed. Finally, experimental results are presented to demonstrate the efficiency of these approaches. It worth mentioning that these methods can be combined to achieve better performance.

With these efforts, learnable bandwidth extension still remains a challenging problem. First of all, the monotonic convergence condition (12) is a sufficient condition. It is desirable to develop necessary and sufficient conditions to further extend the learnable bandwidth. This is also true for the conditions in cyclic pseudo-downsampled ILC. Second, for frequency tuning ILC and phase-in ILC, the determination of cutoff frequency value is based on error signal analysis. The performance of learning can be further improved if associated design criteria or optimal cutoff frequency profile can be developed to maximize the learnable bandwidth. Third, except from cyclic pseudo-downsampled ILC methods, all other methods are developed in the frequency domain. The limitation of the frequency domain analysis is the requirement of linear system. New learnable bandwidth extension methods to deal with nonlinear systems are desired.

REFERENCES


