The Frontiers of Iterative Learning Control — II

Jian-Xin Xu*

1. Introduction

Practical tracking control tasks must be accomplished in a finite time interval. Asymptotic convergence in time domain without a specified convergence speed obviously does not meet the task requirements. On the other hand, it is a much harder problem in control theory to specify the system transient performance or achieve perfect tracking over a finite time interval. Most advanced control methods developed hitherto only ensure asymptotic convergence property. ILC complements the existing control methods in the sense that it targets at perfect tracking in a finite time interval, which is possible under a repeatable control environment and achieved through asymptotic convergence in iteration domain.

In Part I we have shown the limitation of CM-type (contraction mapping) ILC: the requirement of global Lipschitz condition (GLC) [1]. The example given in Section V of Part I tells us that a non-global Lipschitz nonlinear function may incur finite escape time in a simple dynamic system. Let us rewrite the example again

\[ \dot{x} = x^2 + u, \quad x(0) = 0.5 \]
\[ y = x + u. \]

The CM-type ILC design allows us to completely ignore the system dynamics part, whereas the finite time divergence shows how strongly could be such a dynamic impact. It is necessary to widen the learning control framework under which ILC can handle broader classes of system nonlinearities and uncertainties, including NGLC (non-global Lipschitz continuous) dynamics, time varying parametric and norm bounded uncertainties. In this paper we are going to exploit two aspects. The first is to study how the x-dynamics in state space can be incorporated in ILC. The second is to exploit the use of energy function approaches in ILC, such as Lyapunov function. Indeed, looking into the recent advances in control theories and applications, most progress was made in state space with Lyapunov direct method. It would be very meaningful to look into these control methods, henceforth derive the energy function based ILC (EF-based ILC).

There are two main streams in advanced control theories: adaptive control [2] and robust control [3], both highly depending on energy function approaches. The former mainly deals with parametric uncertainties and the latter deals more with the norm-bounded perturbations. We shall start from adaptive control, exhibit the first main characteristic of EF-based ILC — pointwise adaptation when dealing with time varying parametric uncertainties. When the norm-bounded perturbation is concerned, adaptive type control methods are no longer applicable. Robust control, characterized by high gain feedback, is able to secure the uniform bound of the system states, but in general is not able to obtain asymptotic convergence because of the lack of internal model which is nonlinear in nature. By incorporating EF-based ILC, it is possible to eliminate the tracking error asymptotically. Therefore we can show another main characteristic of EF-based ILC — nonlinear internal model control realized through embedding an integrator in the control input.

2. What We Can Learn from Adaptive Control

2.1 Adaptive Control Revisit

Consider a slightly more complicated dynamics

\[ \dot{x} = \theta x^2 + u, \quad x(0) = 0.5 \]

where \( \theta \in \mathcal{R} \) is an unknown constant. Using Lyapunov technology, it is easy to derive a nonlinear adaptive control law. Define a Lyapunov function candidate

\[ V = \frac{1}{2} x^2 + \frac{1}{2} \phi^2 \]

where \( e = x_d - x, \phi = \theta - \hat{\theta}(t) \) and \( \hat{\theta}(t) \) is an adjustable parameter. Differentiating \( V \) yields

\[ \dot{V} = e(\dot{x}_d - \theta x^2 - u) - \phi \hat{\theta}. \]

Choose the control law and adaptation law respectively as below

\[ u = ke + \dot{x}_d - \hat{\theta} x^2 \]
\[ \dot{\theta} = -x^2 e. \]

We have

\[ \dot{V} = -ke^2 \leq 0, \]
hence $x$ and $\dot{\theta}$ are bounded because $V$ is bounded. In the sequel $u$ and $\dot{e}$ are bounded. According to Barbalat Lemma[2], $e \to 0$ as $t \to \infty$.

It is noticeable that how the nonlinear term $x^2$ is incorporated into both control and adaptation laws. Let us summarize the features of adaptive control and try to extend the underlying idea to ILC. 1. It achieves asymptotic convergence in time domain. 2. It uses a nonlinear control mechanism even for a linear dynamics. 3. It only warrants the error convergence $e(t) \to 0$. 4. It is not necessary for the estimated parameter to converge to the true value. 5. The unknown parameter must be a constant.

2.2 Adaptive ILC

When the unknown system parameter $\theta$ is a constant, and the tracking task is repeated over the finite interval $[0, T]$ instead of $[0, \infty)$, applying the above adaptive control does not warrant tracking convergence. Repeating the same control for the same plant only generates the same performance. Let us see how adaptive ILC [4] solves this problem. Here the idea is straightforward: continuously update the parameter $\tilde{\theta}(t)$ along the time axis when the tracking task is repeated, i.e. link two consecutive iterations by

$$\dot{\theta}_{i+1}(0) = \tilde{\theta}_i(T).$$

Define the tracking error $e_i(t) = x_\theta(t) - x_i(t)$. The control law and the parameter updating law are analogous to (5) and (6) for each iteration, i.e.

$$u_i = ke_i + \dot{x}_d - \tilde{\theta}_i x_i^2,$$

$$\dot{\theta}_i = -x_i^2 e_i - \dot{\theta}_i(0) = 0.$$ (9)

Now let us see what kind of convergence property can be acquired. Still define the Lyapunov function candidate for the $i$-th iteration

$$V(e_i, \phi_i) = \frac{1}{2} e_i^2 + \frac{1}{2} \phi_i^2$$ (11)

where $\phi_i \triangleq \theta - \tilde{\theta}_i(t)$.

Note that the error dynamics at the $i$-th iteration, with the control law (9), is

$$\dot{e}_i = -\theta x_i^2 + \dot{x}_d - u_i = -ke_i - \phi_i x_i^2.$$ (12)

Differentiating $V$ with respect to time $t$, substituting the error dynamics (12) and adaptation law (10) yield

$$\dot{V} = -ke_i^2 \leq 0.$$ (13)

Since $t \in [0, T]$, we are not able to enjoy the asymptotic convergence as the adaptive control. Note that under the initial resetting condition, $x_i(0) = x_\theta(0)$ or $e_i(0) = 0$. When $i = 0$, there is no learning and the pure adaptive control is applied, hence $e_i(t)$ and $\dot{\theta}_i(t)$ are bounded in $[0, T]$. Now integrate the derivative of $V$ over $[0, T]$, and use the fact $|e_i(t)| = 0 \leq |e_i(T)|$,

$$V(0, \phi_i(T)) \leq V(e_i(0), \phi_i(0)) + \int_0^T \dot{V} dt = V(0, \phi_{i-1}(T)) - k \int_0^T e_i^2(\tau) d\tau.$$ (14)

Similarly we have

$$V(0, \phi_{i-1}(T)) \leq V(0, \phi_{i-2}(T)) - k \int_0^T e_i^2(\tau) d\tau.$$ (15)

Repeating the operation $i$ times leads to the following relationship

$$V(0, \phi(T)) \leq V(0, \phi_0(T)) - k \sum_{j=1}^i \int_0^T e_j^2(\tau) d\tau$$ (16)

or

$$k \sum_{j=1}^i \int_0^T e_j^2(\tau) d\tau \leq V(0, \phi_0(T)) - V(0, \phi(T)).$$ (17)

From the finiteness of $V(0, \phi_0(T))$ and the positive definiteness of $V(0, \phi(T))$, the sequence $e_j$ is convergent in $L^2$ norm.

We can clearly see that the adaptive ILC (9), (10) at each iteration are exactly the same as the typical adaptive control (5), (6). The major difference is the convergence property: adaptive ILC achieves $\|e_i\|_{L^2[0, T]} \to 0$ as $i \to \infty$, whereas the adaptive control achieves $\|e_i\|_{L^\infty} \to 0$ as $t \to \infty$.

2.3 Illustrative Example

Consider the system (2) where $\theta = 3$ is a constant. The target trajectory is

$$x_d(t) = \sin \pi t + 0.5, \quad t \in [0, 2].$$ (18)

Apply adaptive ILC and choose gain $k = 1$. The simulation result is shown in Fig. 1. The horizon is the iteration number and the vertical scale is the sup-norm $|e_i|$, $i.e.$ the maximum tracking error $|e_i(t)|$ in $[0, T]$. Excellent convergence can be observed.

The main limitation of adaptive ILC, like the adaptive control, is that unknown parameter $\theta$ must be a constant. Assume now $\theta = 3 + \sin \frac{\pi}{2} t$. Applying the
same adaptive ILC again, the asymptotic convergence may not hold, as shown in Fig. 2.

3. ILC with Composite Energy Function

3.1 EF-based ILC — Pointwise Adaptation

Adaptive ILC uses the conventional Lyapunov function because the unknowns are constant parameters. If however $\theta \in C^1[0, T]$ is time-varying, the parameter updating mechanisms (6) and (10), characterized as integrator, may not work well. According to the internal model principle, an integrator in time domain can only estimate or track a constant quantity when the process enters steady state, as we know well in classical control theory. Thus the traditional parameter adaptation mechanisms cannot deal with the time varying parameters. On the other hand, under the repeatable control environment, $\theta(t)$ is invariant with respect to the iterations. Hence for a particular $t \in [0, T]$, the corresponding $\theta(t)$ is considered as a constant in iteration domain $N' = \{0, 1, \ldots\}$. We can use a simple integrator to do the parameter updating job along the iteration axis. This leads to a new ILC approach — the pointwise adaptation over the entire interval $[0, T]$. Let us give the control law and parameter updating law. Consider the nonlinear dynamic system in $i$-th iteration

$$\dot{x}_i = \theta(t)x_i^2 + u_i, \quad x_i(0) = 0.5$$

where $\theta(t) \in C^1[0, T]$, and $x(t)$ is to track $x_d(t) \in C^1[0, T]$ with $x_d(0) = 0.5$. The error dynamics is

$$\dot{e}_i = -\theta(t)x_i^2 + \dot{x}_d - u_i, \quad e_i(0) = 0.$$  \hspace{1cm} (19)

The control law is

$$u_i = ke_i + \dot{x}_d - \dot{\theta}_i(t)x_i^2$$

and the parametric updating law is $\forall t \in [0, T]$

$$\dot{\theta}_i(t) = \dot{\theta}_{i-1}(t) - x_i^2(t)e_i(t) \quad \dot{\theta}_{-1}(t) = 0.$$  \hspace{1cm} (20)

It should be noted that the above control law is exactly the same as the previous case. The only difference arises in the parameter updating mechanisms. For each instant $t$, (21) describes a discrete-type parameter adaptation mechanism (integrator) in iteration domain.

3.2 Convergence with Composite Energy Function

Now let us derive the convergence property of the above ILC. For this purpose we need to find an appropriate "energy function" which plays the similar role as Lyapunov function in adaptive control. Can we directly use the preceding Lyapunov function $V(e_i, \dot{\theta}_i) = (1/2)e_i^2 + (1/2)\dot{\theta}_i^2$? Since the parametric updating is conducted in iteration domain, differentiating $V$ and $\dot{\theta}$ cannot capture the learning process in iteration domain. Can we consider the difference of $V$, i.e. $\Delta V_i = V(e_i, \dot{\theta}_i) - V(e_{i-1}, \dot{\theta}_{i-1})$? The difficulty lies in that the system dynamics (19) is described by differential equation instead of difference equation.

To overcome this difficulty, a composite energy function is used [5]

$$E_i(t) = \frac{1}{2}e_i^2(t) + \frac{1}{2} \int_0^t \phi_i^2(\tau)d\tau.$$  \hspace{1cm} (22)

Here we use $E_i$ to distinguish it from Lyapunov functions $V$ or $V_i$. The difference of $E_i$ is

$$\Delta E_i = E_i - E_{i-1} = \frac{1}{2}e_i^2 + \int_0^t (\phi_i^2 - \phi_{i-1}^2)d\tau - \frac{1}{2}e_{i-1}^2.$$  \hspace{1cm} (23)

Using the initial resetting condition, substituting the error dynamics (19) and the control law (20), the first term on the right hand side is

$$\frac{1}{2}e_i^2 = \int_0^t e_i\dot{e}_id\tau$$

$$= \int_0^t e_i[-\theta(t)x_i^2 + \dot{x}_d - u_i]d\tau$$

$$= \int_0^t [-\phi_i x_i^2 e_i - ke_i^2]d\tau.$$  \hspace{1cm} (24)

By substituting the parameter updating law (21), the second term on the right hand side of (23) can be expressed as

$$\frac{1}{2} \int_0^t (\phi_i^2 - \phi_{i-1}^2)d\tau$$

$$= \frac{1}{2} \int_0^t [(\dot{\theta}_{i-1} - \dot{\theta}_i)(2\theta - 2\dot{\theta}_i + \dot{\theta}_i - \dot{\theta}_{i-1})d\tau$$

$$= \int_0^t (\phi_i x_i^2 e_i - \frac{1}{2}x_i^4 e_i^2)dr.$$  \hspace{1cm} (25)

Clearly $\phi_i x_i^2 e_i$ appears in (24) and (25) with opposite signs. The difference of the composite energy function is

$$\Delta E_i = -\int_0^t ke_i^2 - \frac{1}{2} \int_0^t x_i^4 e_i^2 - \frac{1}{2}e_{i-1}^2 < 0.$$  \hspace{1cm} (26)
The function \( E_t \) is a monotonically decreasing sequence, hence is bounded if \( E_0 \) is bounded. The derivative of \( E_t \) is

\[
\dot{E}_0 = e_0 d_0 + \frac{1}{2} \dot{\phi}_0^2 \\
= -k \dot{e}_t^2 - \phi_0 x_0^2 e_0 + \frac{1}{2} \phi_0^2.
\]

At iteration number \( i = 0 \), \( \dot{\theta}_{-1} (t) = 0 \) \( \forall [0, T] \), thus

\[
\dot{\theta}_0 = -x_0^2 e_0,
\]

and \( \dot{E}_0 \) becomes

\[
\dot{E}_0 = -k \dot{e}_0^2 + \phi_0 \dot{\theta}_0 + \frac{1}{2} \phi_0^2 \\
= -k \dot{e}_0^2 - \frac{1}{2} \phi_0^2 + \phi_0 \theta.
\]

Using Young’s inequality, we have for any \( c > 0 \)

\[
\phi_0 \theta \leq c \phi_0^2 + \frac{1}{4c} \theta^2.
\]

Let \( 0 < c < \frac{1}{2} \),

\[
\dot{E}_0 \leq -k \dot{e}_0^2 - \left( \frac{1}{2} - c \right) \phi_0^2 + \frac{1}{4c} \theta^2.
\]

Since \( \theta (t) \in C^1 [0, T] \), there exists a finite bound \( \theta_m \geq \theta (t) \) \( \forall t \in [0, T] \). Thus \( E_0 \) is negative definite outside the region

\[
(e_0, \phi_0) \in \mathbb{R}^2 \bigg| k \dot{e}_0^2 - \left( \frac{1}{2} - c \right) \phi_0^2 \leq \frac{1}{4c} \theta_m^2
\]

which also specifies the bound of \( E_0 (t) \) in the finite interval \( [0, T] \).

Applying (26) repeatedly we have

\[
E_i (t) = E_0 (t) + \sum_{j=1}^{i} \Delta E_j \\
\lim_{i \to \infty} E_i (t) < E_0 (t) - \lim_{i \to \infty} \sum_{j=1}^{i} \int_{0}^{t} k \dot{e}_j^2 d\tau \\
- \lim_{i \to \infty} \sum_{j=1}^{i-1} \dot{e}_j^2 (t).
\]

Consider the positiveness of \( E_i \) and boundedness of \( E_0, e_i (t) \) converges to zero pointwise as \( i \to \infty \).

**Remark 1:** It is known mathematically that the pointwise convergence does not guarantee the convergent sequence to have a fixed upperbound, for instance

\[
e_i (t) = t^2 e^{-at}
\]

If possible, the uniform convergence should be targeted. However it is not so easy to always accomplish a uniform convergent tracking error sequence in ILC, due to its pointwise adaptation nature. Fortunately, in most practical control problems such a tracking error sequence as (30) can hardly appear, and the tracking performance will be sufficiently good with only a finite number of iterations, afterwards learning process ceases..

**Remark 2:** The learning effect can be maximized by treating the known term \( \dot{x}_d \) through learning, instead of cancelation in (20). In such case, there will be two time varying elements to be learned \( \hat{\theta}(t) = [\theta(t), \dot{x}_d (t)]^T \), and the corresponding regressor is \( \xi = [x_d^T, -1]^T \). The control law is

\[
u_i = k e_i - \dot{\theta}_i^T(t) \xi_i
\]

and the parametric updating law is \( \forall t \in [0, T] \)

\[
\dot{\theta}_i (t) = \dot{\theta}_{i-1} (t) - \xi_i (t) e_i (t) \quad \theta (t) = 0_{2 \times 1}.
\]

The convergence analysis is almost the same, except that \( \dot{\phi}_i^2 \) term in the CEF should be replaced by the vector valued \( \dot{\phi}_i^T \phi_i \) with \( \phi_i \equiv \theta - \dot{\theta}_i \).

**Remark 3:** In the pioneer work of EF-based ILC [6,7], a similar energy function has been used as below

\[
E_i (t) = \int_{0}^{t} \phi_i^2 d\tau
\]

which however is of \( L^2 \) only. By adding additional (1/2)\( e_i^2 \) or in general a Lyapunov function \( V \), CEF in (22) is obvious more general with both \( L^2 \) and \( L^\infty \).

### 3.3 Illustrative Example

Consider system (2) and target trajectory (17) again, but with an unknown time varying parameter \( \theta = 3 + \sin (\pi / 2) t \). Applying control law (20), updating law (21) and choosing gain \( k = 1 \), the learning convergence is shown in Fig. 3.

![Fig. 3 Learning convergence of ILC based on CEF](image)

Next the learning effect is maximized and control law (31) is applied. The learning convergence can be seen from Fig. 4.

### 4. On the Initial Condition

#### 4.1 Relaxation — Alignment Condition

The initial resetting condition has been with us from the CM-type ILC to the EF-based ILC. The necessity in CM-type ILC has been discussed when
perfect tracking is required [1]. From the differential equation theory, the initial condition will determine the solution trajectory of a nonlinear dynamics. A tiny discrepancy in initial conditions may lead to completely different solutions. However, a perfect initial resetting requires that the control system be equipped with a precise homing mechanism, which may not be possible for many practical engineering systems. Can we relax this condition to certain extent? Note that in CM-type ILC, the control objective is output tracking and the state variables are assumed neither available nor manoeuvrable. In EF-based ILC, however, we make full use of the system knowledge especially concerning state dynamics. This opens a new avenue: replacing the initial resetting condition with a less restricted initial condition – alignment condition – and meanwhile achieving the convergent property. The alignment condition is simply \( x_i(0) = x_{i-1}(T) \), i.e. the end state of preceding iteration becomes the initial state of the present iteration. In addition to this, we also need \( x_d(0) = x_d(T) \). The simplicity of the new condition can be easily perceived: we restart from wherever we stopped at. In the sequel we can avoid doing the extra task of bringing the system back to a specific place.

Under the framework of CEF, let us derive the convergence property with the alignment condition. Look into the procedure in deriving \( \Delta E_i(T) \) in the preceding section. Without the initial resetting equation the equation (24) is

\[
\frac{1}{2} e_i^2(t) = \int_0^t e_i e_d \, dt + \frac{1}{2} e_i^2(0). 
\]

Choosing \( t = T \) and using the alignment condition \( e_i(0) = e_{i-1}(T) \), the relationship (26) becomes

\[
\Delta E_i(T) = -\int_0^T k e_i^2 \, dt - \frac{1}{2} \int_0^T x_i^2 e_d^2 \, dt + \frac{1}{2} e_i^2(0) - \frac{1}{2} e_{i-1}^2(T). 
\]

Consequently

\[
E_i(T) = E_0(T) + \sum_{j=1}^i \Delta E_j(T) 
\]

\[
\lim_{i \to \infty} E_i(T) = E_0(T) - \lim_{i \to \infty} \sum_{j=1}^i \int_0^T k e_i^2 \, dt, 
\]

the tracking error sequence converges in \( L^2 \)-norm, instead of pointwise convergence.

### 4.2 Spatial Resetting vs Temporal Resetting

The initial resetting condition in ILC usually implies both spatial resetting and temporal resetting. While time resetting is natural for a task to be finished and repeated over a finite period, the spatial resetting is however not an easy job and not so imperative. Note that it is the spatial resetting which gives rise to extra implementation difficulty and incurs criticism.

Consider a target trajectory \( x_d(t) \in C^1[0, T] \), which forms a continuously spatial path. When do we need the spatial resetting? It is necessary only when the spatial path of the target trajectory is not completely closed, i.e. \( x_d(0) \neq x_d(T) \). For instance, \( x_d(t) = t, \quad t \in [0, 1] \). In such circumstance, a perfect tracking will lead to \( x_i(T) = x_d(T) \neq x_d(0) \). Hence an independent control mechanism must work appropriately between two consecutive iterations so as to bring back the system state to the initial position \( x_d(0) \).

For any trajectories spatially closed, i.e. \( x_d(0) = x_d(T) \), we can use the alignment condition and remove the spatial resetting requirement, as discussed in the preceding subsection.

### 4.3 Extention to Repetitive Tasks

By relaxing the spatial resetting to the alignment condition, we can now extend ILC to most repetitive control tasks – either tracking a periodic trajectory or reject a periodic disturbance over \([0, \infty)\). Let us exhibit how to convert a repetitive control task into an ILC task. Consider the target trajectory \( x_d(t) \in C^1[0, \infty) \) with the periodicity \( x_d(t) = x_d(t-T) \). Assume that \( \theta(t) \) is also periodic with the same period \( T \). Define the state \( x_i(t) = x((i-1)T+t) \), \( i = 1, 2, \ldots \). By virtue of the continuity, \( x(iT) \) is the end point of the \( i \)-th iteration defined over \([iT, (i+1)T)\), and also the initial point of the \((i+1)\)-th iteration defined over \([iT, (i+1)T)\). Note that the alignment condition is met because \( x_i(T) = x_{i+1}(0) \) is in fact the same point \( x(iT) \). Thus the original control problem is equivalent for \( x_i(t) \) to track \( x_d(t) \) over the period \([0, T]\), and the ILC can be directly applied.

What can we gain by converting a repetitive control problem into ILC problem? First of all, ILC is now able to handle periodic signals defined in infinite horizon, hence cover repetitive control problems. Second, ILC based on CEF is able to handle more general classes of system nonlinearities and uncertainties. Indeed, the convergence analysis of repetitive control is
mainly based on small gain theorem, quite similar to the contraction mapping, consequently the application is rather limited.

Remark 4: When \( e(0) = 0 \), the repetitive type ILC will generate a continuous control profile. If \( e(0) \neq 0 \), the repetitive type ILC may generate a piecewise continuous control profile, with the discontinuities occurring at each instant \( t = iT \).

### 4.4 Illustrative Example

Consider system (2) with the target trajectory (17), and unknown parameter \( \theta = 3 + \sin(\pi/2)t \). The initial values are \( x(0) = 1 \neq x_d(0) = 0.5 \). Instead of the initial resetting condition, the alignment condition \( x_i(0) = x_{i-1}(T) \) is used. Applying control law (20) and updating law (21), the learning convergence is shown in Fig. 5, which is close to the case with ideal initial resetting (Fig. 22).

Next, the ILC is extended to a repetitive case where the target trajectory is

\[
x_d = 0.5 + \sin \pi t, \quad t \in [0, \infty).
\]

The unknown time-varying parameter is \( \theta = 3 + \sin(\pi/2)t \). Thus the learning period should be \( T = 4 \). Applying the same ILC scheme with the alignment condition, the maximum error for each period is recorded in Fig. 6. The effectiveness is validated.

![Fig. 5 Learning convergence for system with alignment condition](image)

![Fig. 6 Learning convergence for repetitive tracking](image)

### 5. ILC with Optimality

#### 5.1 Underlying Idea

Since ILC evolves both in time domain and iteration domain, optimality should be pursued along both two directions. However so far little has been explored in iteration domain, thereafter we focus on time domain optimality. Aiming at balancing control performance vs control effort, nonlinear optimal control has been the active subject of considerable research work over the past few decades [8]. By applying the standard dynamic programming, the optimal control can be converted to the problem of solving partial differential equation known as Hamilton-Jacobi-Bellman (HJB) equation. There are however two obstacles in achieving optimal ILC. First, it is difficult to find the closed form optimal control for general nonlinear systems because of the difficulty in solving the nonlinear partial HJB differential equation. Second, ILC by default is supposed to handle systems with uncertainties.

Here we introduce the idea of quasi-optimal ILC[5] to attack the problem. To avoid the first obstacle, a suboptimal control strategy based on control Lyapunov function and Sontag’s formula[9] is used, which provides a suboptimal performance as well as stability along time horizon for a broad class of nonlinear dynamic systems. Regarding the second obstacle, we assume that the system dynamics can be separated into a nominal part and an uncertain part as below

\[
\dot{x}_i = f_i + \theta(t)x_i^2 + u_i
\]

where \( f_i = f(x_i, t) \) is a known smooth function, and considered as the nominal part. The suboptimal control law will be designed based on the system nominal part

\[
\dot{x}_i = f_i + u_i,
\]

and the ILC with pointwise adaptation mechanism is to address the uncertain part \( \theta(t)x_i^2 \) as before.

It should be noted that \( f_i \) can be easily canceled out by incorporating a term \(-f_i\) in the system input \( u_i \). Cancelation is nevertheless a passive way whereas optimal design is an active way of making use of the system knowledge \( f_i \). For example, if \( f_i = Ax_i \), we can construct an optimal controller accordingly. Besides, in many practical systems the nominal part consists of either linear or nonlinear damping term, which is a stabilizing force. If \( f_i \) is much smaller than \( \theta(t)x_i^2 \), the effect of optimality may be minor. Indeed, if the system is predominant by the uncertain part, other control methods such as robust control, adaptive control or ILC instead of optimal control should be used. On the contrary, if the nominal part dominates, the effect of optimality becomes obvious. Often we cannot tell which is dominant in practice, then it should be no harm to let ILC and optimal control coexist.
5.2 Quasi-optimal ILC

Consider a particular case $f_i = -x_i^2$ which is a nonlinear damping. The error dynamics under iteration is

$$\dot{e}_i = \dot{x}_d + x_i^3 - \theta(t)x_i^2 - u_i.$$  \hspace{1cm} (38)

The nominal part of the error dynamics is $\dot{e}_i = \dot{x}_d + x_i^3$. The objective function of optimal control is

$$J = \inf_{u_i} \int_0^T [q(e_i) + u_i^2]dt.$$  \hspace{1cm} (39)

If it is possible to solve the following HJB equation

$$q(e_i) - \frac{1}{4} \frac{\partial V^*}{\partial e_i}^2 + \frac{\partial V^*}{\partial e_i} \dot{f}_i = 0$$  \hspace{1cm} (40)

to find the closed form of the value function $V^*$, the optimal control is given as

$$u^*_i = -\frac{1}{2} \frac{\partial V^*}{\partial e_i}.$$  \hspace{1cm} (41)

However it is not an easy job to solve the HJB equation. Hence the Sontag's formula is introduced to provide a suboptimal solution to the nonlinear system (38) as follows

$$u_{opt,i} = \dot{f}_i + \sqrt{(\dot{f}_i)^2 + q(e_i)} \text{sign} \left( \frac{\partial V}{\partial e_i} \right).$$  \hspace{1cm} (41)

where $\text{sign}(\cdot)$ is the signum function, and $V$ is an arbitrary Lyapunov function (in general Control Lyapunov Function). It can be seen that the idea of the suboptimal control with Sontag's formula is to use a Lyapunov function $V$ to replace the value function $V^*$ which is hard to solve from the HJB equation. It is also worth to note how the nonlinear nominal part $\dot{f}_i$ is incorporated in the suboptimal control law (41), which is not merely a simple cancelation.

To show the boundedness of the tracking error, choose $V = (1/2)e_i^2$. Substituting the $u_{opt,i}$ into the nominal part of the error dynamics (38) yields

$$\dot{e}_i = -\sqrt{(\dot{f}_i)^2 + q(e_i)} \text{sign} \left( \frac{\partial V}{\partial e_i} \right).$$

Thus the time derivative of $V$, in the absence of the uncertainty, is

$$\dot{V} = \frac{\partial V}{\partial e_i} \dot{e}_i \leq -\sqrt{(\dot{f}_i)^2 + q(e_i)} |e_i| \leq 0.$$  \hspace{1cm} (41)

Quasi-optimal ILC is achieved by combining the suboptimal control and ILC simply in an additive form,

$$u_i = u_{opt,i} + u_{i,i},$$
$$u_{i,i} = ke_i - \dot{\theta}_i(t)x_i^2,$$
$$\dot{\theta}_i(t) = \dot{\theta}_{i-1}(t) - x_i^2 e_i$$  \hspace{1cm} (42)

where $u_{i,i}$ is the learning part with pointwise parameter adaptation.

The convergence analysis is much the same as part $B$ of Section III. By substituting the new control law (42) into the error dynamics (38) we have

$$\dot{e}_i = -ke_i - \sqrt{(\dot{f}_i)^2 + q(e_i)} \text{sign}(e_i) - \dot{\theta}_i x_i^2.$$  \hspace{1cm} (42)

In the sequel

$$\int_0^t e_i \dot{e}_i dt = \int_0^t e_i [\dot{x}_d + x_i^3 - \theta(t)x_i^2 - u_i] dt$$
$$= \int_0^t [-\dot{\theta}_i x_i^2 e_i - \sqrt{(\dot{f}_i)^2 + q(e_i)} |e_i| - ke_i] dt.$$  \hspace{1cm} (42)

Comparing with (24), the only difference is the additional term $-\sqrt{(\dot{f}_i)^2 + q(e_i)} |e_i|$ which is negative definite. The negativity of $\Delta E_i$ and boundedness of $E_0$ can thus be derived the same way.

5.3 Illustrative Example

Since $f_i = -x_i^2$ is known, it can be easily canceled by $u_i$. ILC with CEF can be applied directly. Choose $k = 1$ and the result is shown in Fig. 7 with the solid line. Next the quasi-optimal ILC (41) and (42) are employed with $q(e_i) = me_i^2$ and $m = 10$. The improved convergence can be seen from the dashed line in Fig. 7.

Next choose $m = 0.1$ and 100 respectively. As shown in Fig. 8, when $m$ increases the tracking error decreases.

Remark 5: For simplicity a simple SISO example is used throughout the paper. We know optimal control has been developed mainly for MIMO system. Thus
Institute of Systems, Control and Information Engineers
NII-Electronic Library Service

6. Can ILC Track Non-uniform Trajectories?

6.1 The Answer is Yes

That the target trajectory must be uniformly identical for all iterations is one of the fundamental conditions, hence one of the fundamental constraints, for all kinds of ILC schemes. Can we move one step forward – let ILC mechanism learn from different target trajectories? There were some pioneer work done by [10,11] etc., which are in essence a Least Square approach. The limitation of those methods are the demand for accurate information of control input and output signals of previous trials. Can we let ILC start from scratch and converge even if the target trajectory \( x_{d,i} \in C^1[0, T] \) may vary at each iteration? Let us deal with this problem under the framework of CEF. In fact, if the initial resetting condition is satisfied, i.e., \( x_i(0) = x_{d,i}(0) \), this problem is rather straightforward. Since the error dynamics (19) now is

\[
\dot{e}_i = -\theta(t)x_i^2 + \dot{x}_{d,i} - u_i, \quad e_i(0) = 0.
\]

The only change is from the identical trajectory \( x_d \) to the iteration dependent \( x_{d,i} \), which are nevertheless known to us. Accordingly the control law is

\[
u_i = k\epsilon_i + \dot{x}_{d,i} - \hat{\theta}(t)x_i^2
\]
and the parametric updating law remains the same as (21). It is immediately obvious that the preceding convergence analysis still holds.

Remark 6: Since \( x_{d,i} \) is now iteration i-dependent, generally speaking it cannot be treated through learning. The pointwise adaptation, as a functional approximation process, works only for iteration independent functions, such as \( \theta(t) \). This clearly shows the learnability of ILC.

6.2 Illustrative Example

The target trajectory is \( x_d = 0.5 + k_1 \sin k_2 t \) where \( k_1 \in [-1,1], k_2 \in [\pi/2,3\pi/2] \) and \( t \in [0, 2] \). In each iteration, the values of \( k_1 \) and \( k_2 \) are randomly generated from the given ranges. The simulation result is shown in Fig. 9, which confirms the learnability of ILC for non-uniform trajectories.

7. Can We Learn Norm-bounded Uncertainties?

7.1 Robust ILC

Up to now we focus on ILC with parametric uncertainties. What shall we do if the nonlinear uncertain term \( \theta(t)x^2 \) is only known a priori as a lumped disturbance with a known bounding function? Consider the following system with norm-bounded uncertainties \( \eta(x,t) \) which is local Lipschitz continuous

\[
\dot{x} = \eta(x,t) + u, \quad x(0) = 0.5
\]
or

\[
\dot{e} = \dot{x} - \eta - u, \quad e(0) = 0
\]

where \( |\eta(x,t)| \leq \bar{\eta}(x,t) \) and the bounding function \( \bar{\eta} \) is known a priori and continuously differentiable with respect to \( x \) and \( t \). Clearly we are unable to conduct pointwise adaptation for \( \eta \) as it is \( x \)-dependent, i.e. iteration dependent. Now let us study how robust control works. A typical robust control law can be chosen as

\[
u_i = u_{\tau,i} = (\rho_i \kappa_i + 1)\epsilon_i
\]

\[
\rho_i = \sqrt{2\bar{\eta}^2 + \epsilon + \bar{p}_i}
\]

\[
\kappa_i = \frac{\bar{\eta}^2 + 3\epsilon^2 + 8\epsilon}{(\sqrt{\bar{\eta}^2 + 3\epsilon^2 + \epsilon})^2}
\]

where \( \epsilon > 0 \) is a constant. Both \( \rho_i \) and \( \kappa_i \) are smooth functions of \( \epsilon_i \) and \( t \). This control law guarantees a tracking error bound \( |\epsilon_i| \leq \epsilon \). Unfortunately the tracking error cannot be eliminated perfectly by error feedback alone, as an infinitesimal \( \epsilon \) may lead to an unbounded gain \( \kappa_i \) when \( \epsilon_i \to 0 \).

When the control task repeats, ILC is again the appropriate candidate to improve control performance. The underlying idea is as follows. Since the system state is bounded in a set under robust control, the dynamic system is Lipschitz continuous on the set. Thus we can adopt the most widely applied CM-type ILC approach – embedding an integrator in the control input so as to learn the desired control profile directly. This leads to a new ILC strategy – robust ILC. Note that CEF contains a Lyapunov function, hence in this circumstance is suitable for robust ILC design and analysis. We also know that the Lipschitz constant is not used in CM-type ILC because of the introduction of the time weighted norm. We will do the same here by modifying the CEF (22) with a time weighting factor.

Choose the learning control law as below

\[
u_i = \text{proj}(u_{\tau,i}) + u_{\tau,i}
\]

where \( \text{proj}(\cdot) \triangleq \begin{cases} \cdot & |\cdot| \leq u^* \\ \text{sign}(\cdot)u^* & |\cdot| > u^* \end{cases} \)

\[
|u_{d}(t)| \sup_{t \in [0,T]} u^* \text{ is large such that } u^* \geq \text{sup}_{t \in [0,T]} |u_{d}(t)|. \text{ In practice, } u^* \text{ is}
\]

\[
\text{Fig. 9 Learning convergence in tracking non-uniform trajectories}
\]
either a physical process limitation or a virtual saturation bound which can be arbitrarily large but finite.

To analyze the convergence property of the proposed robust ILC, we define the following time weighted composite energy function

\[ E_i(t) = e^{-\lambda t}e_i^2 + \int_0^t e^{-\lambda \tau} \delta u_i^2 d\tau \]  

(48)

where \( \delta u_i = u_i - u_i \). Note the difference between two CEF (22) and (48). The former contains unknown parameters explicitly, whereas the latter contains the unknown but desired control input directly. The difference arise because the norm-bounded uncertainty is the lumped one and iteration dependent. Consequently the ILC law is also different. For norm-bounded case, the input integration along iteration axis appears again as shown in (47), analogous to CM-type ILC but now dealing with more general NGLC nonlinearities. (48) is the generalization of [12,13].

The ultimate objective of learning control is to find the desired control input \( u_d \) which realizes

\[ \dot{x}_d(t) = \eta(x_d, t) + u_d(t). \]  

(49)

According to the system dynamics (45), we can obtain

\[ \delta u_i = (\dot{x}_d - \eta_d) - (\dot{x}_i - \eta_i) = \dot{e}_i + \eta_i - \eta_d \]  

(50)

where \( \eta_d = \eta(x_d, t) \).

### 7.2 Convergence Properties

Let us show the convergence properties of the ILC scheme (47) and (46): the tracking error sequence \( e_i \) converges to zero uniformly, and the control input sequence \( u_i \) converges to the ideal \( u_d \) almost everywhere in \([0, T]\). First we will show that the boundedness of the system state is guaranteed by control laws (47) and (46). Define a Lyapunov function \( V_i = (1/2)e_i^2 \). When \( |e_i| \geq 1 \), the following inequality holds

\[ 1 - \kappa_i |e_i| = \frac{e^2_i + 3e^2_i + e^2_i + 2e\sqrt{e^2_i + 3e^2_i}}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} - \frac{\sqrt{e^2_i + 3e^2_i} |e_i| + 8e_i |e_i|}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} \leq \frac{e^2_i + 4e^2_i + 2e\sqrt{e^2_i + 3e^2_i}}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} - \frac{\sqrt{e^2_i + 8e_i |e_i|}}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} \leq \frac{4e^2_i + 4e\sqrt{e^2_i + 3e^2_i}}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} - \frac{8e_i |e_i|}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} \leq \frac{4(\epsilon - e_i |e_i|)}{\left(\sqrt{e^2_i + 3e^2_i}\right)^2} < 0. \]  

(51)

Consequently we have

\[ V_i = e_i \dot{e}_i = e_i(\dot{x}_d - \eta_d - u_i) \leq |e_i| u^* - e^2_i + (1 - \kappa_i |e_i|) (|e_i| + \bar{p}_i) |e_i| \]

\[ \leq |e_i| u^* - e^2_i = -|e_i| (|e_i| - u^*). \]

\( |e_i| \) is globally uniformly bounded by \( \max \{\epsilon, u^*\} \). Hence \( x_i \in X \) where \( X \) is a compact set.

Since \( x_i \) is bounded and \( \eta_i \) is local Lipschitz, there exists a Lipschitz constant \( \lambda_i \sup \left| \frac{\partial \eta_i}{\partial x_i} \right| < \infty \), \( \forall i \in \mathcal{N} \) and \( \forall (x_i, t) \in X \times [0, T] \), such that

\[ |\eta_i - \eta_d| \leq \lambda_i |x_i - x_d|. \]  

(52)

Moreover, according to the control law (46) and (47) the boundedness of \( x_i \) guarantees the finiteness of \( u_{r,i} \) and \( u_i \). Consequently, \( \dot{x}_i \) and \( \dot{e}_i \) are also finite on \( X \). From the definition of \( \kappa_i \) and \( \rho_i \), it can be derived that following two quantities are finite

\[ c_1 \equiv \sup_{(x_i, t) \in X \times [0, T]} \rho_i \kappa_i, \]

\[ c_2 \equiv \sup_{(x_i, t) \in X \times [0, T]} \frac{d\rho_i \kappa_i}{dt}. \]

Next let us calculate the difference of \( E_i(t) \)

\[ \Delta E_i = e^{-\lambda t}e_i^2 + \int_0^t e^{-\lambda \tau} (\delta u_i^2 - \delta u_{i-1}^2) d\tau \]

\[ -e^{-\lambda t}e_{i-1}^2. \]  

(53)

The first term on the right hand side of (53), with the initial resetting condition, can be expressed as

\[ e^{-\lambda t}e_i^2 = -\lambda \int_0^t e^{-\lambda \tau} e_i^2 d\tau + \int_0^t 2e^{-\lambda \tau} e_i e_i d\tau. \]  

(54)

The second term on the right hand side of (53) can be expressed as

\[ \int_0^t e^{-\lambda \tau} (\delta u_i^2 - \delta u_{i-1}^2) d\tau \leq \int_0^t e^{-\lambda \tau} ([u_i - \text{proj}(u_i - 1)]^2) d\tau \]

\[ = \int_0^t e^{-\lambda \tau} [-2(u_i - u_i)u_{i-1} - u_{i-1}^2] d\tau. \]  

(55)

Substitute (50) into (55) and drop the \( u_{i-1}^2 \) term, we have

\[ \int_0^t e^{-\lambda \tau} (\delta u_i^2 - \delta u_{i-1}^2) d\tau \]

\[ \leq -2 \int_0^t e^{-\lambda \tau} (\eta_i - \eta_d) (\rho_i \kappa_i + 1) e_i d\tau \]

\[ -2 \int_0^t e^{-\lambda \tau} (\rho_i \kappa_i + 1) e_i e_i d\tau \]

\[ \leq -2 \int_0^t e^{-\lambda \tau} (\eta_i - \eta_d) (\rho_i \kappa_i + 1) e_i d\tau \]

\[ + \int_0^t e^{-\lambda \tau} e_i^2 d\rho_i \kappa_i - 2 \int_0^t e^{-\lambda \tau} e_i e_i d\tau. \]  

(56)

Substituting (54) and (56) into (53) and considering (52), yield
\[ \Delta E_i = -\lambda \int_0^t e^{-\lambda \tau} e_i^2 d\tau + c_2 \int_0^t e^{-\lambda \tau} e_i^2 d\tau + 2 \int_0^t e^{-\lambda \tau} l_n x_d - x_i (c_1 + 1) e_i d\tau - e^{-\lambda \tau} e_i^2_{i-1} \]
\[ \leq -\lambda \lambda \int_0^t e^{-\lambda \tau} e_i^2 d\tau + [2 l_n (c_1 + 1) + c_2] \int_0^t e^{-\lambda \tau} e_i^2 d\tau - e^{-\lambda \tau} e_i^2_{i-1} \]
\[ = -\lambda - 2 l_n (c_1 + 1) + c_2 \int_0^t e^{-\lambda \tau} e_i^2 d\tau - e^{-\lambda \tau} e_i^2_{i-1}. \]

There exists a sufficiently large \( \lambda \) such that \( \lambda > 2 l_n (c_1 + 1) + c_2 \) to ensure that

\[ E_i(t) - E_{i-1}(t) \leq -e^{-\lambda T} e_i^2_{i-1}(t) \leq -e^{-\lambda T} e_i^2_{i-1}(t). \]

Consequently, \( E_i(t) \leq E_0(t) - e^{-\lambda T} \sum_{j=0}^{i-1} e_i^2(t). \)

Since both \( x_0 \) and \( u_0 \) are bounded under robust control, \( E_0(t) \) is bounded. From the positiveness of \( E_i(t), \) we can derive that \( \lim_{t \to \infty} e_i(t) = 0 \) pointwisely.

Next from (52), \( \lim_{t \to \infty} |\eta_i(t) - \eta_d(t)| \leq \lim_{t \to \infty} l_n |e_i| = 0. \) Thus using (50) and the boundedness of \( e_i \), we further derive

\[ \lim_{t \to \infty} E_i(t) = \lim_{t \to \infty} e^{-\lambda T} e_i^2 + \lim_{t \to \infty} \int_0^t e^{-\lambda \tau} \delta u_i^2 d\tau \]
\[ = \lim_{t \to \infty} \int_0^t e^{-\lambda \tau} (e_i + \eta_i - \eta_d)^2 d\tau \]
\[ = \lim_{t \to \infty} \int_0^t e^{-\lambda \tau} e_i^2 d\tau. \]

Since \( e_i \) is bounded, \( \lim_{t \to \infty} e_i = 0 \) leads to \( \lim_{t \to \infty} E_i(t) = 0 \) pointwisely. Hence, \( u_i \) converges to \( u_d \) almost everywhere as \( i \to \infty. \) On the other hand the boundedness of \( e_i(t) \) implies the uniform continuity of \( e_i(t). \) Hence, \( e_i(t) \) converges to zero uniformly.

Remark 8: Similar to Section IV, alignment condition can be applied to remove the initial resetting condition, and the ILC works for repetitive tasks. In such case the CEF (22) instead of (48) should be chosen, and the Lipschitz condition (52) will be used in the controller design.

Remark 9: Though the system (45) has relative degree of 1, the EF-based ILC does not need the derivative signal.

7.3 Illustrative Example

Consider system (45) with \( \eta = (3 + \sin t)^2. \) The target trajectory is \( x_d = \sin \pi t + 0.5, \) \( t \in [0, 2]. \) The known bounding function of \( \eta \) is \( \hat{\rho} = (16 \pi^2 + 1)^{1/2}. \) Choose \( \varepsilon = 0.3, \) \( v^* = 60 \) and apply the robust ILC scheme. The simulation result is shown in Fig. 10. At \( i = 0, \) the tracking error is the result of the robust control alone. Through comparison, the learning effect is obvious.

8. Concluding Remarks

To recap, in this paper we discussed a few important issues associated with ILC. First, we demonstrate how the concept of energy function, in particular the Lyapunov function and CEF, can be incorporated in ILC design and analysis. Second, the inherent relationship between adaptive control and EF-based ILC was illustrated. In a repeatable control environment, EF-based ILC achieves pointwise adaptation for time varying parameters. Third, synthesizing with robust control, EF-based ILC nullifies the influence from the norm-bounded disturbances, thus accomplishes the nonlinear internal model in general for repeated or periodic dynamic systems. Fourth, with the alignment condition, we are able to remove the initial resetting condition, facilitate learning control for repetitive control tasks in infinite time horizon, meanwhile still enjoy the advantages of EF-based ILC. Fifth, ILC for non-uniform trajectories and ILC with nonlinear optimality are exploited.

The paper covers a portion of the latest advances in ILC, but could by no means exhaust all. One important direction is how to construct ILC in a constrained input space[14]. From mathematical viewpoint ILC is a kind of well constructed functional approximation. The learnability of ILC is thus directly associated with the task space, input space, and various constraints from system dynamics. There also exist numerous open problems in ILC, such as output tracking with observer, design with guaranteed performance, application to infinite dimensional systems, the existence and uniqueness, etc. It is not exaggerated to say, that most sub-fields of control theory and application can find their counterparts in ILC field. Finally what is most interested to us is, many tough control problems may be revisited with the extra degree of freedoms offered by iterative learning.

References

– I, Systems, Control and Information, Vol. 46, No. 2,

著者略歴
Jian-Xin Xu
Vol. 46, No. 2, p. 72参照。

システム制御情報学会論文誌掲載別刷り料金
【別刷り 50 部・消費税別】

<table>
<thead>
<tr>
<th>原稿種別</th>
<th>ページ数</th>
<th>掲載別刷り料 50 部</th>
</tr>
</thead>
<tbody>
<tr>
<td>研究速報</td>
<td>3以内</td>
<td>40,000 円</td>
</tr>
<tr>
<td></td>
<td>6以内</td>
<td>70,000 円</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>90,000 円</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>115,000 円</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>145,000 円</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>180,000 円</td>
</tr>
<tr>
<td></td>
<td>11ページ以上の場合は1ページごとに加算</td>
<td>40,000 円</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>TEx原稿</th>
<th>非TEx原稿</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>55,000 円</td>
<td>90,000 円</td>
</tr>
<tr>
<td></td>
<td></td>
<td>115,000 円</td>
<td>145,000 円</td>
</tr>
<tr>
<td></td>
<td></td>
<td>180,000 円</td>
<td>220,000 円</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50部ごとに 10,000 円</td>
<td></td>
</tr>
</tbody>
</table>

注意：カラー印刷を必要とする場合には実費が加算されます