Iterative learning control and repetitive control in hard disk drive industry—A tutorial

YangQuan Chen¹.*, †, Kevin L. Moore², Jie Yu³ and Tao Zhang⁴

¹Center for Self-Organizing and Intelligent Systems (CSOIS), Department of Electrical and Computer Engineering, 4160 Old Main Hill, Utah State University, Logan, UT 84322-4160, U.S.A.
²Division of Engineering, Colorado School of Mines, Golden, CO 80401-1887, U.S.A.
³Western Digital Technologies 20511 Lake Forest Drive, Lake Forest, CA 92630, U.S.A.
⁴Seagate Technology, 1280 Disc Drive, Shakopee, MN 55379, U.S.A.

SUMMARY

This paper presents a tutorial on iterative learning control (ILC) and repetitive control (RC) techniques in hard disk drive (HDD) industry for compensation of repeatable runouts (RRO). After each tutorial, an application example is given. For ILC, a simple filtering-free implementation for written-in RRO compensation is presented. For the RC part, a new application of RC in dual-stage HDD servo is presented.

1. INTRODUCTION

Hard disk drives (HDDs) are ubiquitous and pervasive. HDDs are complex intelligent mechatronic systems/devices. Just hook up and plug-in HDDs and they will work for us reliably and efficiently. Just like electrical motors, today, we take HDDs for granted. To enhance our appreciation of the sophisticated HDD servo controls, Messner and Ehrlich [1] presented a nice tutorial on controls for disk drives. The history of HDDs servo controls has been reviewed in [2] and [3] is a textbook dedicated to HDD servo systems.

To further motivate the readers, let us indicate the first fact that, as early as the year 2000, HDDs were ‘nanotech products’. For example, if a HDD has TPI (tracks per inch) of 58 000. Then, the track pitch (TP) is $25.4 \text{ mm}/58000 = 497 \text{ nm}$. Assuming that the track following accuracy is

*Correspondence to: YangQuan Chen, Center for Self-Organizing and Intelligent Systems (CSOIS), Department of Electrical and Computer Engineering, 4160 Old Main Hill, Utah State University, Logan, UT 84322-4160, U.S.A.
†E-mail: yqchen@ece.usu.edu, yqchen@engineering.usu.edu
±10% of the TP, we get the nanopositioning of read/write (R/W) head with precision less than ±50 nm. Note that in [1], the highest TPI was 38 000 or 1500 tracks/mm. Today, the TPI has already been pushed beyond 100 K and steadily toward 200 K. Therefore, every HDD today is a nanotech product only from the nanopositioning point of view, not mentioning the R/W head material synthesis.

The second important fact is that, there is no conventional physical sensors for sensing displacement, velocity and acceleration in HDD servo control. This is quite magic and amazing as simply shown in Figure 1 where VCM (voice coil motor), spindle motor, disk (media), R/W head are clearly shown. The key enabling technology is called ‘embedded servo’ which uses a STW (servo track writer) to lay servo bursts on the media (disk) with track and sector information embedded on the servo burst, as illustrated in Figure 2. For example, a disk drive product in the year 2000 has 288 of servo sectors with spindle speed of 5400 RPM, i.e. 90 Hz. Then, it is clear that, the sampling frequency of this drive is $90 \times 288 \approx 26$ kHz. Note that, the R/W head can pick up the

![Figure 1](image1.png)  
**Figure 1.** An illustrative picture of an uncovered hard disk drive.

![Figure 2](image2.png)  
**Figure 2.** Embedded servo for position sensing.
The third important fact is that the areal density, which is defined as the product of BPI (bits per inch) and TPI, of HDDs is physically limited by the superparamagnetic effect in BPI side. There is a smallest allowable magnetic grain in the media. If the grain is smaller than critical size (e.g. less than 10 nm in diameter), random thermal effects will cause the grains to de-magnetize in tens of nanoseconds and the HDD will be volatile, not non-volatile for tens of years! So, increasing TPI, also known as radial density, or track density, is preferred. However, there is also a limit on TPI. That is, the TPI cannot be too high such that the neighboring tracks could be mis-registered. Enough TMR (track mis-registration) margin has to be ensured to guarantee reliable R/W. In simple words, increasing BPI will challenge head/media material scientists and this may incur big cost while increasing TPI will simply challenge servo mechanic engineers asking for smarter servo control algorithms.

This paper is an industrial tutorial on using these ‘smarter servo control algorithms’ to attack the high TPI challenge \cite{4}. The basic objective is to reduce the head positional error signal (PES) in face of external disturbances. There are various positional error (PES) sources. Basically, two types of errors are defined: NRRO (non-repeatable runout) and RRO (repeatable runout). NRRO may be caused by PES generation noise (demodulation noise), disk vibrations, actuator arm vibrations, disk enclosure vibrations, air turbulence (windage), etc. RRO may be caused by STW error (formatting errors, non-circular eccentric tracks, also known as WIRRO (written-in RRO)), synchronized vibrations (spindle motor imperfection). In this paper, we focus on how to compensate various types of RRO in HDD servo using ILC (iterative learning control) and RC (repetitive control) techniques.

This paper is organized as follows. After a brief tutorial on ILC in Section 2, we will present a simple, filtering-free type ILC application for WIRRO compensation in Section 3. Then, a brief introduction to RC is presented in Section 4 from ‘comb filter’ point of view. Section 5 is on
Y. Q. CHEN ET AL.

RC for periodic disturbance attenuation in dual-stage disk drive servo systems. Finally, Section 6 concludes this tutorial paper with some remarks on future research.

2. A TUTORIAL ON ILC

When we human perform a given task repeatedly, we can gain skills. Since many machines perform the same task again and again (e.g. robots, batch reactors, HDD servos, etc.), what we expect the machines to perform better and better? Can the repetitiveness be harnessed to improve the control performance of the machines? This is the main motivation of ILC. As indicated in [5], in general, a system under control may have uncertainties in its dynamic model and its environment. One attractive point in ILC is the utilization of the system repetitiveness to reduce such uncertainties and in turn to improve the control performance by operating the system repeatedly. ILC is a feedforward control strategy that updates, through iterative learning, control signals at every repeated operation. As the number of iterations increases, the system tracking error over the entire operation time period including the transient portion will decrease and eventually vanish. This may not be possible for conventional non-iterative learning control.

ILC is a relatively new type of control method compared with other control methods. For example, as of December 2006, in ieeeXplore.ieee.org, using search term ‘ILC’ for all fields generates 465 papers while using ‘robust control’ generates 10 075 and 11 654 for ‘adaptive control’ and 2800 for ‘distributed control’.

The term ‘ILC’ was coined by Arimoto and his associates [6] for a better control of repetitive systems. One way to describe the learning is as a process where the objective of achieving a desired result is obtained by experience when only partial knowledge about the plant is available. Therefore, as pointed out in [5], ‘learning is a bridge between knowledge and experience.’

So, roughly speaking, the purpose of introducing ILC is to utilize the system repetitions as experience to improve the system control performance even under incomplete knowledge of the system to be controlled. Based on Arimoto’s formulation, the mathematical description of ILC is as follows.

- **Dynamic system and control task:** A general nonlinear dynamic system is considered. The system is controlled to track a given desired output $y_d(t)$ over a fixed time interval. The system is operated repeatedly and the state equation at the $k$th repetition is described as follows:

\[
\begin{align*}
\dot{x}_k(t) &= f(x_k(t), u_k(t)) \\
y_k(t) &= g(x_k(t), u_k(t))
\end{align*}
\]

where $t \in [0, T]$; $x_k(t)$, $y_k(t)$ and $u_k(t)$ are state, output and control variables respectively. Only the output $y_k(t)$ is assumed to be measurable and the tracking error at the $k$th iteration is denoted by $e_k(t) \triangleq y_d(t) - y_k(t)$.

- **Postulates:**
  - **P1.** Every trial (pass, cycle, batch, iteration, repetition) ends in a fixed time of duration $T > 0$.
  - **P2.** A desired output $y_d(t)$ is given a priori over $[0, T]$. 

ILC AND RC IN HARD DISK DRIVE INDUSTRY—A TUTORIAL

- **P3.** Repetition of the initial setting is satisfied, that is, the initial state \( x_k(0) \) of the objective system can be set the same at the beginning of each iteration: \( x_k(0) = x^0 \), for \( k = 1, 2, \ldots \).
- **P4.** Invariance of the system dynamics is ensured throughout these repeated iterations.
- **P5.** Every output \( y_k(t) \) can be measured and therefore the tracking error signal, \( e_k(t) = y_d(t) - y_k(t) \), can be utilized in the construction of the next input \( u_{k+1}(t) \).
- **P6.** The system dynamics are invertible, that is, for a given desired output \( y_d(t) \) with a piecewise continuous derivative, there exists a unique input \( u_d(t) \) that drives the system to produce the output \( y_d(t) \).

- **Controller design task:** The controller design task is to find a recursive control law

\[
 u_{k+1}(t) = F(u_k(t), e_k(t))
\]

such that \( e_k(t) \) vanishes as \( k \) tends to infinity.

It should be noted that various research efforts can be found in the literature targeting on relaxing the above postulates (P1–P5). We do not give detailed references here, but refer the interested reader to several historical and recent surveys [7–12]. For extensive ILC tutorial slides, check [13, 14].

An illustrative toy example in the continuous-time domain is given in [15] using Arimoto’s D-type ILC updating law. Another illustrative example from HDD industry is presented in the next section where the simple ILC scheme has been practically used in mass production. A more detailed tutorial on ILC can be found in [15] with an excellent outlook on future research opportunities.

To end this tutorial section, a practically important issue on ILC ‘monotonic convergence’ must be explained. The reason is clear: ‘monotonic convergence’ ensures the ILC performance to be ‘better and better’. However, conventional ILC convergence is in the sense of ‘asymptotic convergence’ which may imply that the ILC performance could be ‘better, worse, … and better’.

Let us consider the following discrete-time single-input single-output (SISO) system:

\[
 x_k(t+1) = Ax_k(t) + Bu_k(t) \tag{3}
\]

\[
 y_k(t) = Cx_k(t) \tag{4}
\]

\[
 x_k(0) = x_0 \tag{5}
\]

where \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times 1} \) and \( C \in \mathbb{R}^{1 \times n} \) are the matrices describing the system in the state space; \( x_k(t) \in \mathbb{R}^n \), \( u_k(t) \in \mathbb{R} \) and \( y_k(t) \in \mathbb{R} \) are the state, input and output variables, respectively; \( t \) represents discrete-time points along the time axis; and the subscript \( k \) represents the iteration trial number along the iteration axis. Note that \( x_k(0) = x_0 \) for all \( k \) (ideal initial resetting). From the basic ILC theory, it is known that the ILC system is asymptotical-stable (AS) if and only if \( |1 - \gamma h_1| < 1 \) with the control updating law \( u_{k+1}(t) = u_k(t) + \gamma e_k(t + 1) \), where \( h_1 = CB \) with relative degree 1 and \( \gamma \) is the learning gain [16]. Thus, it is easy to design \( \gamma \) such that the condition \( |1 - \gamma h_1| < 1 \) is satisfied provided that \( C \) and \( B \) are known exactly. However, generally it is reasonable to assume that there exist model uncertainties in \( C \) and \( B \). In this case, it is necessary to select \( \gamma \) considering all possible model uncertainties [17–20].
However, the existing research results are almost all restricted to the asymptotic stability (AS) problem. In ILC, it has been observed that AS may not be acceptable in practical point, because it may have very high overshoot during the transient [21].

It is very interesting to note that the ILC AS condition

\[ |1 - \gamma h_1| = |1 - \gamma CB| < 1 \]

has nothing to do with the state matrix A. This is quite surprising in fact. Without knowing A, stable or unstable, ILC can still work (eventually)! This might offer a good reason to explain why ILC is getting more and more popular, but obviously there is something missing. That is the ILC transient and the related monotonic convergence issue. For technical details, please refer to [15, 17, 22].

3. A SIMPLE ILC IMPLEMENTATION IN HDD

In this section, based on [23], we present a simple ILC implementation in HDD servo for compensating the WI-RRO due to STW. Due to STW servo error, the written-in tracks are no longer perfectly circular, as illustrated in Figure 4. As shown in Figure 4, the track following control block diagram contains positional disturbances \( d = d_n + d_w \) where \( d_n \) is NRRO and \( d_w \) is the WIRRO. We wish to find a compensation table \( d_{ZAP} \) such that the zigzag real track can be corrected as close to the ‘straight’, perfectly circular track or ‘the zero acceleration path (ZAP)’, as possible.

Simple derivation from Figure 4 gives

\[
P_{ES}^k = \frac{1}{1 + P(s)C(z)} [r - d_n^k - d_w^k - d_{ZAP}^k] \tag{6}
\]

where \( k \) means the \( k \)th revolution of the disk. \( P(s) \) is the plant model that is uncertain and \( C(z) \) is the known feedback stabilization controller. \( r \) is the reference signal (\( r = 0 \) for track following case).

Figure 4. Embedded servo control loop with WIRRO \( d_w \).
Since the WIRRO is repeatable from revolution to revolution, it is possible to find the best \( d_{ZAP} \) table to compensate WIRRO \( d_w \). However, since \( P(s) \) is uncertain, we use the following ILC method to iteratively refine the compensation table \( d_{ZAP}^k \):

\[
d_{ZAP}^{k+1} = d_{ZAP}^k + \ell(PES_k(t))
\]

Now the question is how to decide the idealized learning operator. Let us iterate (6),

\[
PES^{k+1} = \frac{1}{1 + P(s)C(z)}[r - d_n^k - d_w - d_{ZAP}^k - \ell(PES_k(t))]
\]

\[
= \left(1 - \frac{\ell}{1 + PC}\right) PES^k - \frac{1}{1 + PC} (d_n^{k+1} - d_n^k)
\]

Denote the learning rate as \( \rho(\omega) = 1 - ((\ell)/(1 + PC))(j\omega) \). Obviously, the ideal learning operator

\[
\ell(j\omega) = 1 + PC
\]

Note that \( C(z)PES^k = u_{fb}^k \). The learning updating law could be simply using two available signals PES and \( u_{fb} \). In [23], a very simple ILC scheme is proposed as summarized in Figure 5. The ILC updating law is simply

\[
d_{ZAP}^{k+1} = d_{ZAP}^k + r_k ZPF(\omega_k, z, z^{-1})[PES^k + P_n(s)u_{fb}]
\]

where \( r_k \) is the learning gain at the \( k \)th iteration; ZPF is a low-pass zero phase filter (ZPF) with cutoff frequency \( \omega_k \) and \( P_n(s) \) is basically the nominal VCM model which is just a double integrator with a lumped actuator gain \( k_{lump} \). It should be remarked that the above abuse of notation in ‘\( z \)’ and ‘\( s \)’ domains is for convenience only so that continuous-time and discrete-time variables and systems can be identified. On the other hand, as indicated in [22], the supervector or lifting is the only correct framework for ILC analysis. Using frequency-domain analysis of ILC convergence either in ‘\( z \)’ or in ‘\( s \)’ transfer functions, is questionable since transfer functions are obtained from the steady-state responses to sine inputs but in ILC, each iteration has a finite time duration.

Note further that, \( PES^k \) should be understood as PESRRO part. Our ultimate goal is to use minimum possible revolutions (revs) to learn the ZAP table. We have tried two iterations and
3 revs averaging for each iteration and the results are good enough. Therefore, this simple method is attractive in the sense that only simple math operations are required. In addition, four parameters can be scheduled from iteration to iteration. They are (1) learning gain $r_k$; (2) cutoff frequency of the ZPF $\omega_k$; (3) phase advance and (4) servo loop gain change during PES collection for average. Detailed discussions on the scheduled parameters and the advantages can be found in [23].

The ILC algorithm is summarized in the following:

1. Lower the servo loop gain and then collect 3 revs PES and Udac. Do averaging. Get $PES_a$ and $Udac_a$.
2. Demean $Udac_a$ and then double integrate it. Get $Udac_{II}$.
3. $ZAP\_table\_1=(PES_a+Klump* Udac_{II})*Learning\_Gain\_1$.
4. ZPF for the first $ZAP\_table\_1$ (NOTE: $ZAP\_table\_0=0$).
5. Repeat (1) with an increased servo-loop gain.
6. Repeat (2).
7. $ZAP\_table\_2=ZAP\_table\_1+(PES_a+Klump*Uda_{II})*Learning\_Gain\_2$.
8. Set servo-loop gain to normal.

The drive level results are illustrated in Figure 6 where left four subplots are for the case of no WIRRO compensation and the right four subplots are for the case of after WIRRO compensation. In each case, four subplots are presented with (1) SNR: $1\sigma(RRO)/1\sigma(NRRO)$; (2) RRO: $(1\sigma)$ percent TP; (3) RV: total runout value (RV); and (4) NRRO. $(1\sigma)$ Obviously, NRRO remains almost the same before and after compensation. Typical time-domain traces are shown in Figure 7. Upper subplots are PES signals and the bottom subplots are for the actuator DAC signal. Clearly, after ILC compensation, the actuator follows a more straight path resulting less actuation.

To end this section, besides numerous RRO compensation-related patents, we wish to point out several recent papers on using ILC for RRO compensation in HDD servo [24–26] where new modifications and more systematic designs are presented.
4. A BRIEF TUTORIAL ON RC

In ieeeXplore.ieee.org, using search term ‘RC’ for all fields generates 258 papers. Compared with ILC, this number is about half. The basic idea of RC is to use a delay line to generate the internal model of the periodic disturbance of period $N$ as illustrated in Figure 8. In the ideal case, we can set $Q(z^{-1}) = 1$. Then, the gain block before $C(z^{-1})$ block is simply $1 + z^{-N}/(1 - z^{-N}) = 1/(1 - z^{-N})$. The Bode plot of $1/(1 - z^{-N})$ is shown in Figure 9, which is a ‘comb’. In fact, RC block diagram (Figure 8) can be equivalently drawn in the comb filter form, as shown in Figure 10. Therefore, as indicated in [27], RC is nothing but a kind of comb filter.

Observe that, the overall controller gain

$$C'(z^{-1}) = \frac{1}{1 - Q(z^{-1})z^{-N}C(z^{-1})}$$

has multiple peaks (high gain) at the integer multiples of the harmonic frequency of the periodic disturbance. The sensitivity function $S = 1/(1 + C'P)$ will have multiple deep dips or notches. Therefore, the harmonics in the disturbances can be ‘combed’ out. This is the desirable disturbance rejection or attenuation property from using RC in HDD servo systems.

Two good reviews of RC can be found in [28, 29]. In HDDs, the patent [30] offers many other variations on economic or parsimonious implementations. In [31], some excellent drive level RC results can be found. Note that, for HDD servo, it is normally believed that RC is for attacking harmonics of spindle frequency (90 Hz for 5400 RPM drives). However, as shown in [31] (slides...
Figure 8. Repetitive control with a $Q$-filter, internal-model form.

Figure 9. Bode magnitude plot of $1/(1-z^{-N})$ ($N=100$, sampling frequency is 9000 Hz).

192, 193), when implemented properly, arbitrary fundamental frequency can be achieved. Another remark is that, $Q$-filter is a low-pass filter to limit the effective bandwidth of the RC. $Q$-filter can be (and should be) non-causal.

Just a reminder that ILC and RC are mostly similar with only one important difference. In ILC, we need resetting to the same initial conditions. In RC, however, there is no resetting. RC is continuous along time axis with no extra mechanism to return to the same initial condition.
In the next section, we will present more technical details on RC with an emphasis on dual-stage HDD servo systems.

5. RC FOR PERIODIC DISTURBANCE ATTENUATION IN DUAL-STAGE DISK DRIVE SERVO SYSTEMS

5.1. Background

In many motion control systems, the desired references or external disturbances include periodic signals with a known time period. Iterative learning and repetitive control (IRC/RC) techniques are often used for the periodic signal learning to achieve zero tracking error. RC has been investigated extensively because of its practicality and applicability ([32–38], and the references therein). It is well known that the RC is based on Internal-Model Principle (IMP) ([39]. The algorithm repetitively applies a learning controller to a plant to achieve trajectory tracking or disturbance attenuation. The tracking error is reduced with each repetition and eventually, a perfect tracking control can be achieved. Many successful applications of iterative learning and RC have been found in motion control including HDDs and optical disk drives ([33–36] and many other servo control systems ([40–45].

In this section, we investigate the application of RC technique to the problem of periodic disturbance attenuation in dual-stage HDD control systems. Owing to windage, mechanical resonance, spindle motion and disc flutters, HDD servo systems contain both repeatable and non-repeatable disturbances appearing in the PES. In frequency domain, a RRO disturbance occurs at frequencies that are integer multiples of the disk rotation frequency. It is a considerable source position error with respect to the center of the data track. To meet the TMR requirement for high TPI products, the periodic disturbance must be attenuated to improve the servo tracking performance.

Figure 11 shows a dual-stage HDD servo system. Gvcm and Gpzt denote VCM and microactuator model, respectively. Cvcm and Cpzt are controllers for VCM actuator and Piezo microactuator, respectively. PES is the error between head position and the reference signal. d_nrro represents the total non-repeatable disturbances such as measurement noises, disk flutter, windage affects, etc., d_rro represents the repeatable disturbances due to spindle motion, and W_rro denotes the written-in error on servo tracks.

Generally speaking, there are two types of techniques for repeatable disturbance attenuation. For RRO caused by spindle motor and disk movement, adaptive feedforward compensation (AFC)
Figure 11. Dual-stage HDD control system.

technique is usually used to reject the disturbances [46]. For written-in errors generated in servo writing, WI-RRO, a possible approach is called the ZAP concept, which uses a feed-forward profile created by learning techniques such as ILC/RC schemes. The WI-RRO is compensated such that the servo system does not respond to the written-in disturbance.

5.2. Control structure

In HDDs, the RRO in low harmonic frequencies is caused by the disk vibration or spindle motor. The RRO amplitude and phase of the first several low-frequency harmonics are time varying and usually compensated by AFC. Hence, we separate the W\textsubscript{rro} into the following two parts:

\[ W\textsubscript{rro} = W + W_0 \quad (8) \]

where \( W_0 = W\textsubscript{rro}(\omega), \forall \omega \leq \omega_0 \) denotes the low-frequency repeatable disturbances to be handled by other feedforward methods; \( W \) represents the written-in error to be controlled by repetitive algorithm

\[ W = W\textsubscript{rro}(\omega) \quad \forall \omega > \omega_0 \quad (9) \]

Define

\[ d = d\textsubscript{nrro} + d\textsubscript{rro} + W_0 \quad (10) \]

\[ G = G\textsubscript{vc}G\textsubscript{vc} + G\textsubscript{pzt} \quad (11) \]

A RC structure is proposed in Figure 12. \( K(z) \) and \( Q(z) \) are the RC filters, \( N \) is the time period of one revolution and \( \hat{W} \) is the estimation of \( W \). The control objective is to design suitable filters \( K(z) \) and \( Q(z) \), such that the estimate \( \hat{W} \) converges to \( W \) and the tracking error \( e \) is minimized.

From Figure 12, we have

\[ \hat{W}(z) = Q(z)z^{-N}[\hat{W}(z) + e(z)K(z)] \quad (12) \]

Therefore,

\[ \hat{W}(z) = \frac{z^{-N}Q(z)K(z)}{1 - z^{-N}Q(z)} e(z) \quad (13) \]
In track following case, \( R(z) = 0 \), the closed-loop error system is
\[
e(z) = \hat{W}(z) - W(z) - d(z) - e(z)G(z)
\]
(14)

By using (13), we have
\[
e(z) = \frac{z^{-n}Q(z)K(z)}{[1 + G(z)][1 - z^{-N}Q(z)]}e(z) - \frac{W(z) + d(z)}{1 + G(z)}
\]
(15)

Hence
\[
e(z) = Q(z) \left[ 1 + \frac{K(z)}{1 + G(z)} \right] e(z)z^{-N} - \frac{1 - z^{-N}Q(z)}{1 + G(z)}[W(z) + d(z)]
\]
(16)

Since \( N \) is the one revolution period, we have \( e_{i-1}(z) = z^{-N}e_i(z) \) and \( d_{i-1}(z) = z^{-N}d_i(z) \) with the revolution number \( i \). Equation (16) can be expressed as
\[
e_i(z) = Q(z) \left[ 1 + \frac{K(z)}{1 + G(z)} \right] e_{i-1}(z) - \frac{d_{i} - d_{i-1}Q(z)}{1 + G(z)} - \frac{W_i - W_{i-1}Q(z)}{1 + G(z)}
\]
(17)

In order to cancel the repeatable disturbance \( W \), we select \( Q(z) \) as a ZPF with unity gain 1 at the harmonic frequencies for all \( \omega > \omega_0 \). This implies that
\[W_i - W_{i-1}Q(z) = 0\]

Define
\[
\alpha(\omega) = Q(\omega) \left[ 1 + \frac{K(\omega)}{1 + G(\omega)} \right]
\]
(18)

Then, (17) can be further written as
\[
e_i = \alpha(\omega)e_{i-1} - \frac{d_{i} - d_{i-1}Q(\omega)}{1 + G(\omega)}
\]
(19)

By iterating (19),
\[
e_i = \alpha^2(\omega)e_{i-2} - \alpha(\omega) \frac{d_{i-1} - d_{i-2}Q(\omega)}{1 + G(\omega)} - \frac{d_{i} - d_{i-1}Q(\omega)}{1 + G(\omega)}
\]
(20)
Further iterating (20), and defining $\bar{d}$ the upper bound of noise $d_i$, $\forall i \geq 0$, we have

$$
\| e_i \| \leq \| z(\omega) \| e_0 \| + \left[ \sum_{m=0}^{i} z^m(\omega) \right] \left\| \frac{1 + \| Q(\omega) \|}{1 + G(\omega)} \right\| \bar{d} \\
\leq \| z(\omega) \| e_0 \| + \frac{1 - z^{i+1}(\omega)}{1 - z(\omega)} \left\| \frac{1 + \| Q(\omega) \|}{1 + G(\omega)} \right\| \bar{d}
$$

(21)

To guarantee the convergence, the following condition is required:

$$
\| z(\omega) \| = \| Q(\omega) \left[ 1 + \frac{K(\omega)}{1 + G(\omega)} \right] \| < 1 \quad \forall \omega \leq \omega_s / 2
$$

(22)

where $\omega_s$ is the sampling frequency.

5.3. RC filter design using dual-stage loop-shaping

From (22), we can see that the convergence rate depends on the selection of filters $Q(z)$ and $K(z)$. In ideal case, if the filter $K(z)$ can be selected as

$$
K(\omega) = -[1 + G(\omega)] \quad \forall \omega_0 < \omega \leq \omega_s / 2
$$

(23)

Then,

$$
z(\omega) = 0 \quad \forall \omega_0 < \omega \leq \omega_s / 2
$$

(24)

This implies the convergence can be achieved in one iteration. In practice, it is not possible to design a filter satisfying this condition due to model uncertainties. Individual frequency components of the error signal can exhibit different convergence rates depending upon $z(\omega)$ evaluated at those frequencies [45]. Over the interested frequency range, a fast convergence is achieved with a learning filter $K(z)$ designed as an approximation to the inverse sensitivity function

$$
K(\omega) \approx -[1 + G(\omega)] \quad \forall \omega_0 < \omega \leq \omega_s / 2
$$

(25)

An interesting work in [23] shows a design using the knowledge of a nominal plant model and the control signal in order to construct the approximation of the inverse sensitivity function. There are several other methods for determining a stable approximation to the system inverse sensitivity [40, 47].

In HDD dual-stage servo systems, the additional actuator gives a significant freedom to shape the servo loop. This provides an opportunity to design a special dual-stage controller for improved RC ZAP process. The design concept is that synthesizing a servo controller such that the closed-loop sensitivity function satisfies

$$
\frac{1}{1 + G(\omega)} \approx 1 \quad \forall \omega_0 < \omega \leq \omega_s / 2
$$

(26)

This is highly feasible because the frequency $\omega_0$ is usually low and dual-stage servo system can achieve high bandwidth. A special servo loop can be selected by synthesizing a low bandwidth open-loop servo system with high gain and phase margin, which results in a flat gain response of closed-loop sensitivity such that (26) is satisfied. In this case, we can simply choose the RC filter
\( K(z) = -1 \). This implies that no modeling work is necessary. It greatly simplifies the repetitive controller design and implementation.

It should be mentioned that the above low bandwidth dual-stage system is running during repetitive learning process only. Once the desired RRO compensation profile is obtained, it will be frozen and written to the disk. During normal drive operation, the ZAP profile will be loaded from disc at each servo sample, and a normal bandwidth dual-stage servo system will be used to achieve better seek and tracking performance.

As we mentioned in Section 5.2, RRO at the first several harmonic frequencies are usually compensated by AFC scheme. In order to remove the components of these harmonic frequencies from the profile of repetitive controller, the ZPF \( Q(z) \) must have enough attenuation in the frequency range \( \omega \leq \omega_0 \). This also helps to solve the possible conflict of running AFC and RC algorithm in the same frequency range.

5.4. Drive level results

The proposed RC algorithm has been tested on Seagate 10K RPM Cheetah drives. The harmonic frequency of the product is 166.7 Hz. The sample rate is 48 kHz. Since the first three harmonic frequencies are handled by AFC, the RC algorithm will cover the frequency range 666.7 \( \leq f \leq 24 \) kHz. A high-pass filter \( Q(z) \) is selected as follows

\[
Q(z) = \frac{z^2 - 1.996z + 0.996}{z^2 - 1.825z + 0.8326}
\]  

(27)

where \( \beta = \frac{1}{3} \).

A low bandwidth dual-stage controller is designed to achieve \( 1/[1 + G(z)] \) close to 1 in the frequency range 666.7 \( \leq f \leq 24 \) kHz. Figure 13 plots the frequency response of the dual-stage open-loop \( G(z) \). It is shown that the bandwidth is around 670 Hz with a gain margin of 18 dB,
and a phase margin of 55°. The resulting sensitivity $1/(1+G)$ is plotted in Figure 14. It is shown that the sensitivity is very close to 1 in the RRO ZAP frequency range 667–24 000 Hz.

Figure 15 shows the time history of RRO reduction during the repetitive control process. It is shown that the RRO drops from 12% of TP to 3% after 10 iterations (2880 sectors). Figure 16 shows the frequency-domain RRO comparison. We see that most of the RRO harmonics decreases after applying the RRO compensation profile.

Figure 17 shows the experimental results of RRO improvement with and without RC ZAP. We can see that without ZAP, the RRO 3-sigma is 1.0µinches at OD tracks and 0.85µinches at ID
tracks. The RRO 3-sigma drops to 0.2596\(\mu\)inches in average when applying the RC algorithm. The iteration numbers used on each track are about 6.4 in average.

6. CONCLUDING REMARKS

In plain language, this paper explained why hard disk drive (HDD) servo control is challenging and exciting. We focused on how to reject the repeatable runouts in HDD servo. A tutorial on
iterative learning control (ILC) with a simple application/implementation in HDDs without any filtering operation. Drive-level results are presented for illustration. For RC, the tutorial, although brief, is presented from 'comb filter' point of view. Detailed RC techniques in HDD industry for compensation of repeatable runouts are presented for dual-stage HDD servo.

In summary, ILC/RC techniques have been playing important roles in modern HDD servos. Although there are many issues and challenges in implementations, ILC/RC can be considered as matured, established techniques for repeatable disturbance rejection. The rationale is clear: whenever the disturbance is deterministic and reoccurring, it can be rejected.

REFERENCES


27. Chen YQ. Iterative learning control (ILC) and repetitive learning control (RLC) are special applications of comb filter (CF). http://www.ece.usu.edu/csois/ilc/ILC/com_filter/index.htm, October 1999.


