2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Adaptive Inverse Control for Settling Performance Improvements

Brian Rigney, Lucy Pao, and Dale Lawrence

Abstract— **Single-track hard disk drive (HDD) seek performance is measured by settling time, t_s, defined as the time from the arrival of a seek command until the measured position reaches and stays within an acceptable distance from the target track. Previous work has shown feedforward dynamic inversion, coupled with an aggressive desired trajectory y_d, is capable of achieving high performance settling times when the closed-loop dynamics are time-invariant and accurately modeled. In contrast, we describe an adaptive inversion procedure in this paper which removes the requirement for accurate initial models and tracks the position-variant dynamics present in our Servo Track Writer (STW) experimental apparatus. The proposed indirect adaptive inversion algorithm relies on a recursive least squares (RLS) estimate of the closed-loop dynamics. Pre-filtering of the RLS input signals and covariance resetting are necessary additions to the baseline adaptive algorithm in order to achieve fast settling times. Compared to the nonadaptive solution with accurate system identification, we show the adaptive algorithm achieves a 22% reduction in average settling time and a 53% reduction in settling time standard deviation.**

I. Introduction

Settling time, t_s, is defined as the elapsed time from the start of the commanded motion until the measured position is contained within an acceptable distance from the target position. Minimizing t_s is desirable in many diverse applications, including automated manufacturing, where smaller t_s leads to reduced manufacturing time and cost, and space-based imaging, where smaller t_s enables increased coverage of the target and less distortion in the final image. This paper will focus on a third application: repetitive, unsaturated single-track *seeks* in hard disk drives (HDDs). Typical HDD operating modes where repetitive unsaturated single-tracks seeks are common include servo track writing [1], sequential data transfer [2], and scans for detecting media surface defects resulting from manufacturing [3]. In each of these operating modes, it is desirable to have each HDD within a large population achieve its minimum t_s to increase data throughput or decrease manufacturing time and cost. Standard practices within the HDD industry instead attempt to use a single robust seek control design that achieves a minimum level of performance across the entire population. Therefore, many of the HDDs sacrifice performance in order for the population outliers to perform adequately.

HDD dynamics can nominally be described by a rigid body mode with higher-order structural resonances. Over a population of plants, these resonances show structured uncertainty in lower frequency ranges, and much larger, unstructured variation at higher frequencies. It is extremely difficult for a low-order parametric model to capture the complexity of the population in this higher frequency range. While we simplify the focus of this paper to discrete-time, linear time-invariant (LTI), single-input single-output (SISO) descriptions of the HDD population dynamics, the population uncertainty provides a significant challenge to minimizing settling time.

A further complicating factor in the plant sets is nonminimum phase (NMP) zero dynamics. NMP dynamics can arise when the sensors and actuators are noncollocated [4, Ch. 8], a configuration common in HDDs where the magnetic reader position sensor and voice-coil actuator are on opposite ends of the flexible actuator arm. NMP zeros in discrete time dynamics can also result from fast sample rates and high relative degree [5]. NMP dynamics complicate the choice of settling time reduction algorithm, both for optimizing settling performance on a single HDD and an uncertain population of HDDs.

In previous work [6] [7], we have investigated the use of NMP dynamic inversion algorithms and architectures for settling time reduction on a single unit within a population. While this previous work produced aggressive settling performance on a single unit in a population, the results relied on accurate dynamics modelling. Small errors in the model used for the NMP inverse design led to undesirable settling performance. Further, slowly time- and position-variant dynamics caused the high performance results to deteriorate while seeking over many tracks, regardless of the accuracy of the fixed model.

Alternatively, we explore adaptive NMP inversion methods in this paper which exploit the persistently exciting nature of the HDD operating modes to:

1) Relax the accurate model requirement for each unit in the population.
2) Track slowly time- and position-variant dynamics such that we maintain consistent, aggressive settling performance over the entire range of motion.

While dynamic inversion is typically applied to trajectory *tracking* applications, we show that when combined with adaptive desired output trajectory, y_d, generation, it can be an effective *settling time* reduction approach. We favor this technique because it is computationally simple, lending itself to implementation on low-cost HDD digital signal processors (DSPs), and amenable to on-line adaptation across the population.

B. Rigney is an employee of Western Digital Corporation, Longmont, Colorado, 80503 brian.rigney@gmail.com

L. Pao is a Professor with the Electrical and Computer Engineering Dept., University of Colorado pao@colorado.edu

D. Lawrence is a Professor with the Aerospace Engineering Sciences Dept., University of Colorado dale.lawrence@colorado.edu

978-1-4244-4524-0/09/$25.00 ©2009 AACC

In the following section, we review previous results using NMP dynamic inversion for settling performance. Section III experimentally demonstrates the deterioration in settling performance due to both initial modeling errors and position-dependent dynamics. We review multiple adaptive approaches to NMP dynamic inversion in Section IV, ultimately selecting an indirect scheme based on simplicity and effectiveness. Section V discusses the adjustments required to the adaptive algorithm for high performance experimental results. With these adjustments, we show consistently fast settling times over many seeks using extremely limited initial closed-loop dynamics information. We summarize the results and discuss extensions in Section VI.

II. NMP Inversion for Settling Performance

Fig. 1 depicts our approach to achieving aggressive settling performance, as discussed in [6] and [7]. The closed-loop system H_{CL} is placed in series with a low-order LTI inverse system $\tilde{H}_{CL}^{-1}$ and a desired command profile y_d. The trajectory generator system selects the aggressiveness of the y_d profile to reduce t_s for each unit in the population. Based on the y_d trajectory and knowledge of the H_{CL} dynamics, the inverse system $\tilde{H}_{CL}^{-1}$ determines the required closed-loop input r.

Many related technologies have been applied to various settling performance applications. Time-optimal control strategies with saturated commands, as in [8], do not apply in this application because the actuator commands for single-track HDD seeks are far from the saturation limits. Traditional iterative learning control assumes a fixed desired trajectory [9]. If we choose a fixed y_d that every unit in the HDD population is capable of tracking, we would be sacrificing performance on many units. Classic input shaping techniques are most commonly concerned with the reduction of high frequency resonance mode excitation [10]. In order to achieve high performance HDD seeks, we must not only treat those resonance modes but also the uncertain lower frequency dominant second order dynamics. Optimal trajectory generation algorithms typically require complex calculations [11]. These calculations would be difficult to implement on-line for each unit in the HDD population. Finally, [12] and [13] propose a similar combination of dynamic inversion and reference trajectory generation to reduce t_s. Unfortunately, the complex off-line optimization procedures are not suited to the HDD application. We favor the scheme in Fig. 1 because it addresses the needs of the HDD operating modes with a realistic level of computational complexity for the HDD DSPs.

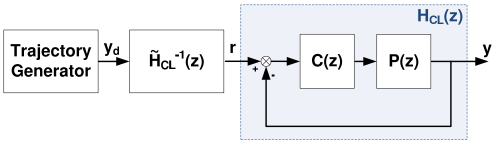

Fig. 1. Block digram describing NMP dynamic inversion for settling time reduction.

A. Desired Output Trajectory Generation

We need a parameterization of y_d trajectories that is computationally simple but able to produce a wide range of y_d's, from unaggressive to extremely aggressive. Motivated by the solution to the time optimal control problem for a rigid body, we use a family of y_d trajectories generated from the double integral of a bang-bang acceleration pulse. The $\mathcal{Z}$-transform of y_d can be written as

$$Y_D(z) \quad = \quad \frac{1}{2d^2}\frac{(z+1)}{(z-1)}\frac{\left(\sum_{i=0}^{d-1} z^i\right)^2}{z^{2d-1}} \ , \qquad (1)$$

where T is the sample time and $2d$ is the total duration of y_d. This simple parameterization provides a single scalar value, d, that can select the aggressiveness of the seek, while also being extremely computationally efficient. Although extensions of this work adaptively select d on-line for each unit in a population of HDDs [14], this paper uses a fixed extremely aggressive y_d. By fixing y_d, we simplify the scope of the paper and focus on the adaptive inversion algorithm. In the following sections, y_d is generated by setting $d = 3$ samples in (1).

B. NMP Inversion Algorithm

Dynamic inversion is complicated by the presence of NMP zeros, even for LTI SISO systems. The NMP zeros of the original system become unstable poles in the inverse system. For the HDD application, this causes the system in Fig. 1 to exactly track y_d (with possible delay) while r grows unbounded. Many techniques exist to compute exact and approximate solutions for the inverse dynamics, $\tilde{H}_{CL}^{-1}$, with bounded input signal r. There is a major division in NMP algorithms between those that stably approximate the exact unstable inverse of H_{CL}, and those that use the exact unstable inverse directly. While [7] quantifies the achievable settling performance of many NMP inversion algorithms, this paper simply reviews the best-performing stable approximate algorithm in the HDD application from [7].

In order to review stable approximate NMP inversion algorithms, it is helpful to first partition the closed-loop dynamics into minimum phase (MP) and NMP zero polynomials

$$H_{CL}(z) = \frac{B(z)}{A(z)} = \frac{B_m(z)B_n(z)}{A(z)} \ , \qquad (2)$$

where B is the closed-loop numerator polynomial, B_m contains all closed-loop MP zeros, B_n contains all closed-loop NMP zeros, and A is the closed-loop denominator polynomial. These polynomials can be described by

$$\begin{aligned} B(z) &= \left(b_0 z^m + b_1 z^{m-1} + \cdots + b_m\right) \ , && (3)\\ B_m(z) &= \left(b_{m0} z^{m_m} + b_{m1} z^{m_m-1} + \cdots + b_{mm_m}\right) \ , && (4)\\ B_n(z) &= \left(b_{n0} z^{m_n} + b_{n1} z^{m_n-1} + \cdots + b_{nm_n}\right) \ , && (5)\\ A(z) &= \left(z^n + a_1 z^{n-1} + \cdots + a_n\right) \ , && (6) \end{aligned}$$

where n is the order of the closed-loop dynamics, m is the total number of closed-loop zeros, m_m is the number of MP zeros, and m_n is the number of NMP zeros. We can then express the closed-loop inverse $\tilde{H}_{CL}^{-1}$ as

$$\tilde{H}_{CL}^{-1}(z) = \frac{A(z)\tilde{B}_n^{-1}(z)}{z^{k_p}B_m(z)} \,. \tag{7}$$

We use the variable $\tilde{B}_n^{-1}$ to denote the inverse of the NMP zero polynomial B_n. Also, k_p samples of delay have been added to $\tilde{H}_{CL}^{-1}$ to maintain causality and ensure $\tilde{H}_{CL}^{-1}$ is exactly proper. The $\tilde{}$ modifier on both $\tilde{H}_{CL}^{-1}$ and $\tilde{B}_n^{-1}$ signifies that these quantities are not the exact inverses of H_{CL} and B_n, respectively. When H_{CL} is NMP, $\tilde{B}_n^{-1}$ can take on numerous forms, including Hurwitz polynomial approximations of the exact inverse and stable rational polynomial transfer functions. The various stable approximate NMP inversion algorithms differentiate themselves in the way they compute $\tilde{B}_n^{-1}$ and the k_p samples of delay required [15]-[18].

A noncausal inverse can result from a strictly proper closed-loop transfer function H_{CL} with nonzero relative degree, where relative degree p is defined as $p = n - m_m - m_n$. Noncausal inverses can also result from the use of noncausal algorithms for approximating the unstable inverse of B_n. Referring to Fig. 1, a noncausal inverse system would require a change in r before a change in y_d, also known as *preactuation*. As discussed in [6] and [7], anticipation of a seek start command would be required to implement preactuation, which is unrealistic in the HDD operating modes of interest. We therefore must add k_p samples of delay to $\tilde{H}_{CL}^{-1}$. Unlike many dynamic inversion *tracking* applications, this negative effect of preactuation on settling time leads to interesting considerations when selecting an inversion algorithm.

The noncausal Taylor series approximation to the NMP inverse approximates $\frac{1}{B_n}$ with a noncausal polynomial

$$\tilde{B}_n^{-1}(z) = \frac{\sum_{i=0}^{m_T} \alpha_i z^i}{B_n(1)\sum_{i=0}^{m_T} \alpha_i} \,, \tag{8}$$

where m_T is the order of the series approximation and the α_i sequence is derived from the Taylor series expansion of $\frac{1}{B_n}$, as in [15]. The resulting transfer function from y_d to y is

$$\frac{Y(z)}{Y_d(z)} = \frac{B_n(z)\sum_{i=0}^{m_T} \alpha_i z^i}{z^{p+m_T+m_n}B_n(1)\sum_{i=0}^{m_T} \alpha_i} \,. \tag{9}$$

While (9) has a finite impulse response (FIR), which has desirable settling performance qualities, we have also added $k_p = p + m_n + m_T$ samples of delay to maintain causality of $\tilde{H}_{CL}^{-1}$. Increasing m_T improves the approximation to $\frac{1}{B_n}$ and yields better (delayed) tracking accuracy, while deteriorating the achievable settling performance. As in [7], the shortest series, a zero-order approximation with $m_T = 0$, provides the best settling performance in this HDD application. The tracking improvements that come from higher-order approximations are negated by the accompanying increase in t_s.

III. Nonadaptive Experimental Results

A. Experimental Hardware

A Servo Track Writer (STW) [1], provided by Maxtor Corporation, is the experimental testbed for our work. A population of Servo Track Writers is used to magnetically encode the initial servo position information on the magnetic media during high volume HDD manufacturing. The STW has its own voice-coil motor (VCM) and precision encoder that mechanically interface with the HDD actuator arm and HDD VCM through an opening in the HDD baseplate. Modern HDDs can require the STW to make over 500,000 single-track seeks per disk drive. The single-track seek distance is determined by the HDD track density; we use a single-track step size of 1 μrad as a representative angular track width for a modern HDD. The STW has an encoder sensor resolution of 0.5 nanorad, and a compensator sample time of $T = 68$ μs.

We experimentally identify the STW closed-loop dynamics via pseudo-random sinusoidal injection at 100 angular positions, with each position separated by 1 track. Fig. 2 shows the experimentally identified frequency response of H_{CL} at each angular location, with a weighted least-squares model fit to the compilation of all data sets. The weighted least squares model for H_{CL} is

$$H_{CL}(z) = \frac{0.10988(z+0.4947)}{(z-0.1541)(z^2-1.859z+0.8695)} \times \frac{(z^2-1.874z+0.8807)(z^2+2.389z+1.574)}{(z^2-1.24z+0.4409)(z^2+1.233z+0.878)} \,. \tag{10}$$

This 7^{th}-order model has a closed-loop bandwidth near 1 kHz, unity DC gain, a relative degree of 2, a high frequency structural mode near 5.3 kHz, and 2 NMP zeros outside the unit circle at $z = -1.1946 \pm j0.3838$.

While the 100 frequency responses are indistinguishable in Fig. 2, there are minor differences as a function of angular

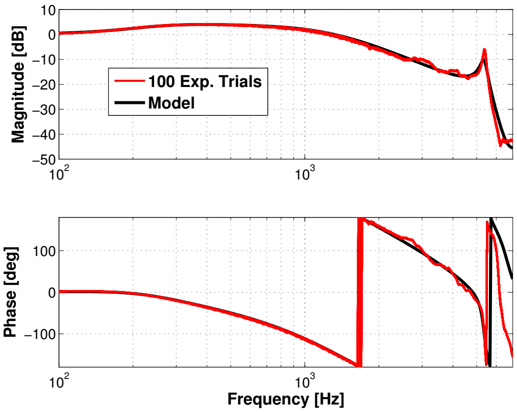

Fig. 2. Experimental and modeled frequency responses of STW closed-loop system.

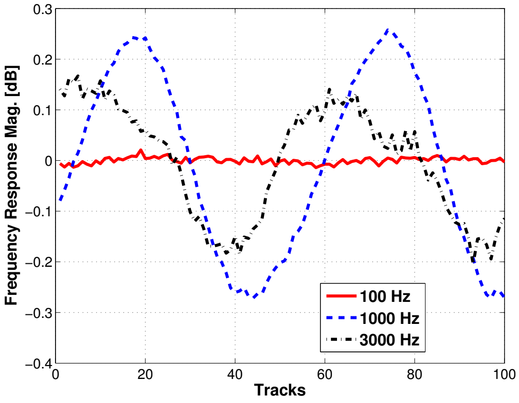

Fig. 3. Frequency magnitude response, with mean removed, of the closed-loop system at 100, 1000, and 3000 Hz over 100 tracks.

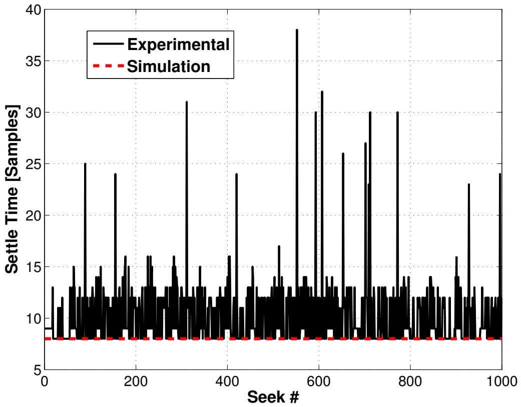

Fig. 4. Experimental and simulated settling performance over 1000 single-track seeks.

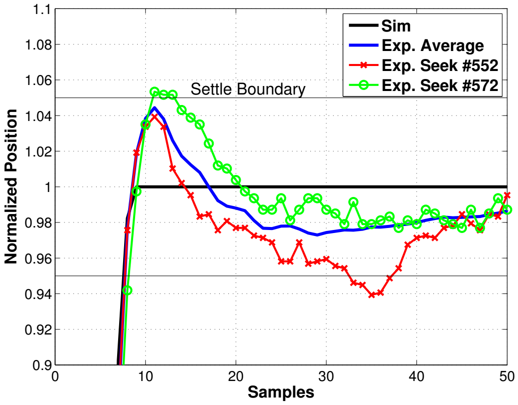

Fig. 5. Simulated and experimental output trajectories, including a 1000 seek average and individual trajectories for the 552^{nd} and 572^{nd} seeks.

location. Fig. 3 plots the frequency magnitude response at 100, 1000, and 3000 Hz versus track for each of the 100 data sets taken over 100 tracks. In each of the three data sets, the mean has been removed to better visualize the trends. There is an obvious periodicity that is most pronounced near the closed-loop bandwidth of 1000 Hz. The period of the oscillation is 56 tracks, which equates to the diffraction grating angular fringe spacing within the STW's optical encoder. The position-dependant fluctuations in the closed-loop dynamics result from optical encoder gain fluctuations as a function of encoder angle. While performing seeks over many tracks, the encoder gain and closed-loop dynamics vary periodically. As we will see, these small fluctuations in the closed-loop dynamics surprisingly lead to pronounced periodic fluctuations in t_s when using a fixed model for H_{CL} and aggressive y_d.

B. Experimental Settling Performance

Fig. 4 shows the simulated and experimental settling performance over 1000 single-track seeks when using (10) with $d = 3$ to derive a fixed model inverse. This is an extremely aggressive y_d given the system order is $n = 7$. Simulation predicts $t_s = 0.544$ ms, or 8 samples, for this choice of y_d. While some of the experimental seeks match this performance periodically, other seeks have settling times as large as 38 samples. In general, the experimental t_s shows a similar 56 tracks per cycle periodicity as the closed-loop gain variations in H_{CL}. This track-to-track variability in t_s is extremely undesirable in the HDD application.

Fig. 5 shows both simulated and experimental output trajectories. The simulated response illustrates the FIR behavior discussed in Section II-B. After 8 samples, the simulated response is within the settling boundary. After 9 samples, the response has perfectly achieved the desired final position. The experimental trajectories include the 1000 seek average response and individual responses from the 552^{nd} and 572^{nd} seeks. The average response is dominated by lower frequency overshoot without considerable high frequency resonance excitation, even for this aggressive y_d trajectory. The overshoot is consistent with misidentification of the dominant closed-loop dynamics. The 552^{nd} and 572^{nd} individual seeks are chosen near the extremes of the periodic variation in the closed-loop dynamics. Both individual seeks show overshoot and undershoot that exceed the settling boundary and dramatically increase t_s. The seemingly random large increases in t_s from Fig. 4 are actually caused by disturbances and noise pushing output trajectories, already close to the settling boundary because of overshoot and undershoot, across the threshold.

This experimental study highlights two serious issues with nonadaptive dynamic inversion for settling performance:

1) The inverse model is extremely sensitive to the off-line system identification of H_{CL}. While high frequency modeling errors become increasingly important with more aggressive y_d's, it is actually the low and mid frequency modelling errors that dominate our t_s results. Misidentification of the dominant 2^{nd}-order dynamics are the cause of the overshoot in the average trajectory in Fig. 5.
2) Time or position-variant dynamics in H_{CL} will cause any fixed model to be in error over many seeks. The t_s periodic fluctuations in Fig. 4 are the results of similar periodic fluctuations in the H_{CL} dynamics.

The following section presents adaptive extensions to dynamic inversion for settling performance in an attempt to address these two issues.

IV. NMP Adaptive Inversion

Adaptive control approaches can be generally classified as either direct or indirect. Indirect methods attempt to identify the relevant system dynamics parameters on-line, and then use those parameters for controller synthesis. The indirect self-tuning regulator (STR) [19, Sec. 3.3] is a well-known example of indirect adaptive control. In contrast, direct adaptive control approaches attempt to adjust the controller parameters on-line without first identifying the system dynamics. Model Reference Adaptive Control (MRAC) [19, Ch. 5] is a well-known direct adaptive feedback control technique.

NMP zero dynamics provide special challenges for adaptive control, especially for direct methods. In order to prohibit MRAC from cancelling the NMP zeros with unstable poles, the NMP zeros are typically assumed known and included in a reference model [19, pp. 118-119]. This is problematic for our HDD application where the NMP zeros may have significant uncertainty. An alternative direct adaptive *feedforward* control technique, specifically formulated for adaptive NMP inversion, uses an FIR approximation to the inverse of the closed-loop dynamics [20]. The inverse itself is constrained to be stable because it is FIR. Unfortunately, accurate approximations of H_{CL} in our application require high-order FIR filters to capture both the high and mid-frequency dynamics. High-order filters require more computation cycles and raise implementation issues given the limited computational hardware in the HDD application.

Given these concerns with direct adaptive control of NMP systems, we instead choose an indirect scheme to adaptively tune the $\tilde{H}_{CL}^{-1}$ system, as shown in Fig. 6. We rely on a recursive least squares (RLS) parameter estimation algorithm to estimate the closed-loop dynamics, and then use the inversion technique from II-B to synthesise an inverse system. A similar approach has been taken in [21], although the authors use a different inversion procedure designed for accurate tracking instead of settling performance.

We use the indices i and j to denote, respectively, seek number and sample number within a seek. Given this definition, we use the equation error formulation [22, Ch. 3] to write the output estimate, denoted $\hat{y}$, as the inner product of a regression vector ϕ and a parameter estimate vector $\hat{\theta}$

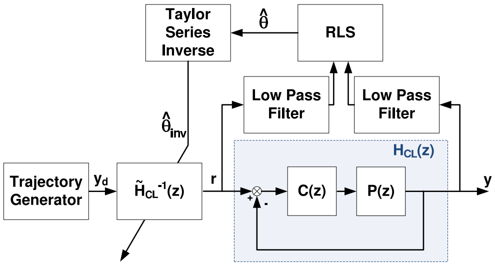

Fig. 6. Indirect adaptive dynamic inversion block diagram.

$$\hat{y}(i,j) = \phi^T(i,j)\hat{\theta}(i,j) \ , \tag{11}$$

where

$$\begin{aligned} \phi(i,j) &= [y(i,j-1)\cdots y(i,j-n) \\ &\qquad r(i,j+m-n)\cdots r(i,j-n)]^T \ , \quad (12) \\ \hat{\theta}(i,j) &= \left[\hat{a}_1\cdots\hat{a}_n\ \hat{b}_0\cdots\hat{b}_m\right]^T \ . \quad (13) \end{aligned}$$

Here, we use the definition of H_{CL} polynomials from (3) and (6). The ˆ modifier on the polynomial coefficients in (13) signifies that these values are approximations to the true closed-loop dynamics.

The recursive least squares algorithm form with forgetting (RLSF) for updating the parameter estimate [22, Ch. 4] is

$$\begin{aligned} \hat{\theta}(i,j) &= \hat{\theta}(i,j-1) + \\ &\qquad L(i,j)\left[y(i,j) - \phi^T(i,j)\hat{\theta}(i,j-1)\right] . (14) \end{aligned}$$

The time-varying vector update gain $L(i,j)$ is

$$L(i,j) = \frac{P(i,j-1)\phi(i,j)}{\lambda + \phi^T(i,j)P(i,j-1)\phi(i,j)} \ , \tag{15}$$

where the matrix P, also known as the covariance matrix, is

$$\begin{aligned} P(i,j) = \frac{1}{\lambda}&\left[P(i,j-1) - \right. \\ &\left. \frac{P(i,j-1)\phi(i,j)\phi^T(i,j)P(i,j-1)}{\lambda + \phi^T(i,j)P(i,j-1)\phi(i,j)}\right] . (16) \end{aligned}$$

λ is the forgetting factor and satisfies $0 < \lambda \leq 1$. $\lambda = 1$ uses no data forgetting in the RLS solution. Once the parameter estimate, $\hat{\theta}$, is computed, the zero-order Taylor series inverse approximation procedure from II-B is used to determine the inverse parameter estimate, $\hat{\theta}_{inv}$.

The parameter estimate, $\hat{\theta}$, from RLS can be shown to converge to the true parameter vector θ [22, Ch. 5], assuming noise-free and correct model structure conditions. The closed-loop input r must have "sufficiently rich" frequency content such that r and y, and consequently ϕ, contain information describing all the dynamics included in the model structure $\hat{\theta}$ [22, Ch. 6]. Exponential convergence can be achieved with RLSF if $\lambda < 1$. The downside of RLSF is that P can grow without bound if persistent excitation is not maintained. RLS without forgetting ($\lambda = 1$) does not suffer from this limitation because P will vanish over many iterations which effectively turns off the parameter updates in (14). While this ensures P remains bounded, time variations in the closed-loop dynamics, such as the periodic fluctuations in sensor gain from Fig. 3, will not be tracked. We choose to use RLS without forgetting in this paper and apply covariance resetting in Section V to overcome the inability to track time-varying parameters.

Persistent excitation can be a difficult requirement to fulfill in a realistic application with disturbances, noise, and unmodeled dynamics. It is further complicated when the adaptive dynamics are responsible for generating the excitation. While the repetitive seek trajectories are aggressive

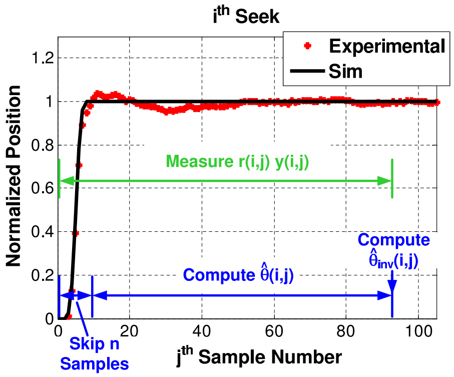

Fig. 7. Timing diagram showing when $\phi(i,j)$ is measured, and when $\hat{\theta}(i,j)$ and $\hat{\theta}_{inv}(i,j)$ are updated.

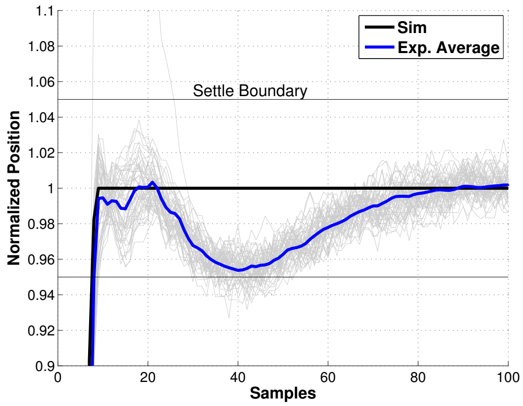

Fig. 8. Output trajectories for 1000 seeks using an adaptive $\tilde{H}_{CL}^{-1}$.

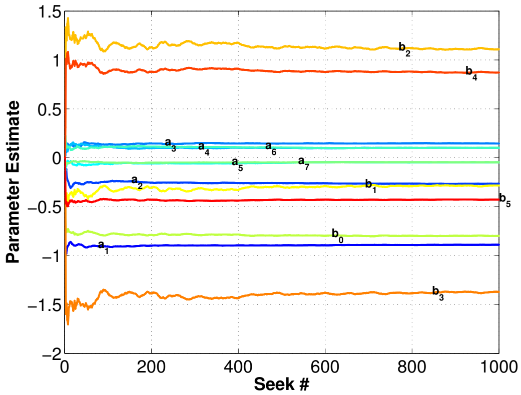

Fig. 9. $\hat{\theta}$ parameter trajectories for the baseline adaptive inversion algorithm.

and highly energetic in our HDD seek application, we are attempting to use zeros in the adaptive inverse to cancel poles in the closed-loop dynamics. If we do this well, we will be limiting the excitation energy in r near those closed-loop dynamics and limiting our ability to accurately identify. For our application, the feedback tracking controller is designed for precise set-point regulation, and thus has a well-damped dominant closed-loop mode. This relaxes the need to perfectly cancel the poles with zeros in the inverse. As we will show with results from our STW experimental apparatus, we are able to achieve the desired aggressive settling performance predicted with idealized noise-free simulations. We tolerate some identification errors in those well-damped closed-loop poles and do not have to resort to the addition of dithering excitation signals as in [20, Ch. 4] to improve identification.

We use a specific update procedure for $\hat{\theta}$ and $\hat{\theta}_{inv}$ to reduce computations and memory usage on the STW hardware, depicted in Fig. 7. First, while the closed-loop input r and output y are measured real-time during the i^{th} seek, the samples are cached and not immediately used. This allows the computations for $\hat{\theta}(i,j)$ to be implemented outside the real-time software. Second, we only calculate $\hat{\theta}_{inv}(i,j)$ when the desired action at a given track is complete, such as reading or writing data. We are only interested in tracking slow changes that occur over many seeks, and thus can use $\hat{\theta}_{inv}(i,j)$ as a fixed inverse over the $(i{+}1)^{th}$ seek. Finally, we do not compute $\hat{\theta}(i,j)$ for the first n samples of each seek (counting from zero), which allows us to populate $\phi(i,j)$ with data taken solely from the i^{th} seek. $\phi(i,j)$ can be discarded when the i^{th} action is complete. At the n^{th} sample of every seek, we require a past parameter vector and covariance matrix. We use the final values from the previous seek, as in $\hat{\theta}(i,n-1) = \hat{\theta}(i-1,k_f)$ and $P(i,n-1) = P(i-1,k_f)$, where k_f is the final sample when the reading or writing action is complete.

V. Adaptive Inverse Experimental Results

A. Baseline Adaptive Results

We begin the adaptive inversion experimental results by discussing the performance of the baseline adaptive algorithm. The algorithm is implemented exactly as described in Section IV and Fig. 6. The low-pass filters (LPFs) from Fig. 6 are set to 1, and we use an extremely poor initial parameter estimate with unity gain. We match the $\hat{\theta}$ order to the order of the model in (10), with $n = 7$ and $m = 5$. This yields an initial parameter estimate of $\hat{\theta}(1,6) = [0_{1\times7} \;\; 1 \;\; 0_{1\times5}]^T$. We also set the initial covariance matrix to the identity matrix $P(1,6) = I_{13\times13}$.

Fig. 8 plots both the average experimental output and each individual experimental trajectory for all 1000 seeks using the baseline adaptive inverse algorithm. The nonadaptive simulation output with an assumed perfect model is included for comparison. The experimental output trajectories show extreme undershoot causing the majority of the seeks to have settling times greater than 40 samples. This is much worse performance than the nonadaptive algorithm from Fig. 4. The errors between the average experimental trajectory and the ideal simulation are predominantly at lower frequencies, suggesting that our on-line identified model is not accurately capturing the dominant 2^{nd}-order dynamics.

Fig. 9 plots the parameter estimate vector components at the end of each seek, $\hat{\theta}(i,k_f)$. Not only do we require these parameters to capture the dominant mode in the closed-loop dynamics, we also expect them to track the known 56 tracks per cycle periodicity. While initially some parameters

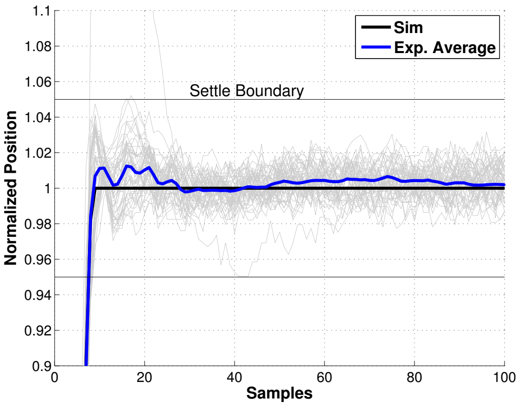

Fig. 10. Output trajectories for 1000 seeks using an adaptive $\tilde{H}_{CL}^{-1}$ and low frequency weighting.

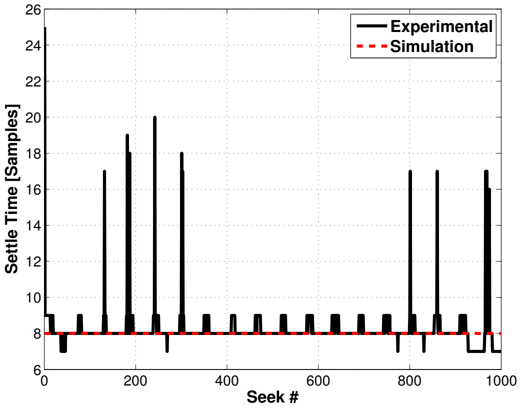

Fig. 11. Settling time over 1000 seeks for the adaptive inversion algorithm with low frequency weighting.

do show a 56 tracks per cycle oscillation, it quickly decays and all parameters show little motion after the 500^{th} seek. Together, both Figs. 8 and 9 clearly indicate that high performance settling times will require adjustments to the baseline adaptive algorithm.

B. Frequency Domain Weighting

Frequency domain weighting is a common technique prescribed to focus the RLS identification algorithm in specified frequency bands [22, Ch. 18]. In our application, the weighting is accomplished through the use of pre-filters on the signals r and y, as in Fig. 6. Motivated by the deficiencies in the baseline adaptive results, we use low-pass filters to more heavily weight the lower frequencies in r and y. We observe a wide range of LPFs correct the misidentification, and select a simple 2^{nd}-order low-pass Butterworth filter with a 100 Hz corner frequency to produce the remaining results in this paper.

Fig. 10 shows the experimental output trajectories when combining the baseline adaptive algorithm with the low-pass filter. We no longer see the extreme undershoot from Fig. 8. In fact, the average experimental response is accurately tracking the idealized simulation with less than 2% error. The LPFs have aided the RLS identification algorithm in accurately estimating the dominant lower frequency dynamics. The settling times as a function of seek number, plotted in Fig. 11, also show promise. After only twenty seeks, the initial settling time of 25 samples has converged to the predicted settling time of 8 samples. Given our poor initial model, this result is very encouraging. Unfortunately, the settling time results are still affected by the 56 tracks per cycle position-variant dynamics. The RLS parameter identification algorithm is clearly not tracking this variation in the closed-loop dynamics.

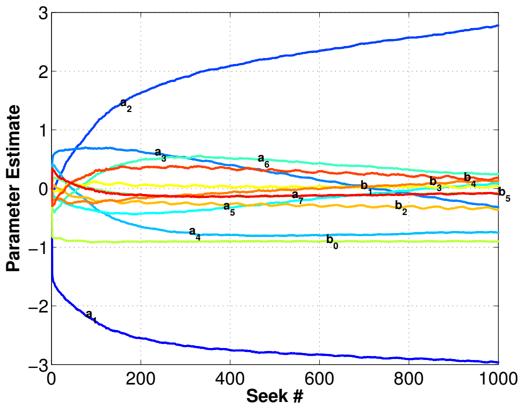

Fig. 12. $\hat{\theta}$ parameter trajectories for the adaptive inversion algorithm with low frequency weighting and covariance resetting.

C. Covariance Resetting

With an RLS algorithm without forgetting, the parameter estimate update gain, $L(i,j)$ in (14), rapidly decreases with each subsequent seek. New information contained in the r and y signals has little effect on the parameter update when $L(i,j)$ is small. This is a classic problem with RLS parameter identification and has multiple possible solutions. We use covariance matrix resetting [22, Sec 3.1] to keep $L(i,j)$ large and allow new information to influence $\hat{\theta}$. The covariance matrix, $P(i,j)$, is initialized and allowed to propagate over the first two seeks as in Section V-A. On the third and all subsequent seeks, the covariance matrix is preloaded to its final value from the second seek $P(i,6) = P(2,k_f)$ for all $i > 2$. While the particular choice of the 2^{nd} seek is not significant, we find this provides an adequately sized P and allows $\hat{\theta}$ to track time-varying dynamics.

Fig. 12 shows the parameter estimate $\hat{\theta}$ evolve over all 1000 seeks. Unlike Fig. 9 without resetting, multiple parameters show a consistent 56 tracks per cycle oscillation that does not decrease as seek number increases. The RLS identification algorithm is now responding to the closed-loop dynamics periodic variation. Fig. 13 plots the simulated and experimental settling times over all 1000 seeks. Similar to the low-pass filter results in Fig. 11, the settling time starts out large and quickly converges to the predicted 8 sample settling time over the first 20 seeks. Unlike the previous results, we no longer see periodic error in the settling performance. After the initial convergence, the settling times deviate from ideal simulation randomly by ±1 sample. The combination

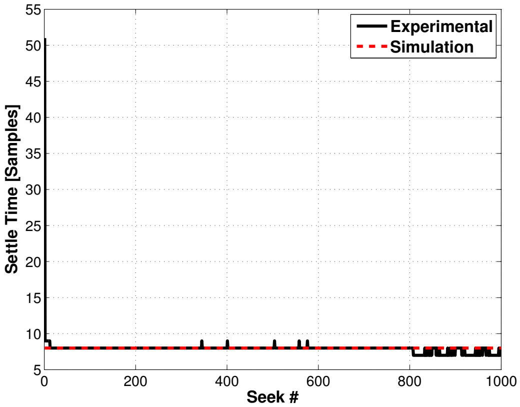

Fig. 13. Settling time over 1000 seeks for the adaptive inversion algorithm with low frequency weighting and covariance resetting.

of low-pass filtering and covariance resetting is required in this application to produce high performance settling times.

The benefits of the adaptive algorithm are clear when comparing with the nonadaptive solution. The average adaptive settling time is 7.9 samples, compared to 10.1 samples for the nonadaptive distribution. The settling time standard deviation for the adaptive algorithm is 1.4 samples versus 3.0 samples for the nonadaptive case. The adaptive solution clearly provides for faster and more consistent settling performance. It is also important to note that the nonadaptive results rely on an accurate off-line model. The settling time distribution for the nonadaptive case would likely be much worse for a different STW in the population.

VI. Conclusions

Adaptive dynamic inversion is an effective and practical means to achieve high performance settling times. With very poor initial models, the adaptive inverse algorithm is able to quickly identify the relative closed-loop dynamics over very few seeks and lower the settling time. Further, the adaptive algorithm tracks position-dependent dynamics as we seek over many tracks and maintains a consistent settling performance. We see substantial improvements in both the average settling time and settling time standard deviation when compared with a nonadaptive solution. The benefits of the adaptive algorithm are further strengthened when considering the nonadaptive results were taken with an accurate system identification model of the specific hardware.

The benefits of the adaptive inversion algorithm are not for free. In order to obtain the desired settling performance, both low-pass filtering and covariance resetting were required additions to the baseline adaptive algorithm. These two features add additional design parameters that may require changes for different families of closed-loop dynamics, different time- or position-variant dynamics, or different y_d trajectories. Extensions of this work focus on variable y_d trajectories and computational improvements through the use of fixed-covariance matrices.

References

[1] C. Lee. “Servowriters: A critical tool in hard disk manufacturing.” *Solid State Technol.*, vol. 34, no. 5, pp. 207-211, 1991.

[2] J. Guttmann, M. Heminger, M. Hicken, S. Howe, and T. Swatosh. “Method and apparatus for optimizing the data transfer rate to and from a plurality of disk surfaces.” U.S. Patent 6105104, Aug. 15, 2000.

[3] M. Blachek, M. Neumann, G. Smith, and P. Wachowiak. “Method and apparatus for detecting handling damage in a disk drive.” U.S. Patent 5935261, Aug. 10, 1999.

[4] D. Miu. *Mechatronics*. New York, NY: Springer-Verlag, 1993, pp. 114-155.

[5] K. Åström, P. Hagander, and J. Sternby. “Zeros of sampled-data systems.” *Automatica*, vol. 20, no. 1, pp. 31-38, 1984.

[6] B. Rigney, L. Pao, and D. Lawrence. “Discrete-Time Exact and Approximate Dynamic Inversion for Settle Performance,” in *Proc. IFAC World Congress*, July 2008, pp. 1778-1784.

[7] B. Rigney, L. Pao, and D. Lawrence. “Nonminimum phase dynamic inversion for settle time applications.” *IEEE Trans. Ctrl. Sys. Tech.*, vol. 17, 2009.

[8] H. Ho. “Fast servo bang-bang seek control.” *IEEE Trans. Magn.*, vol. 33, no. 6, pp. 4522-4527, Nov. 1997.

[9] D. Bristow, M. Tharayil, and A. Alleyne. “A survey of iterative learning control.” *Control Sys. Mag.*, vol. 26, no. 3, pp. 96-114, Jun. 2006.

[10] T. Singh and W. Singhose. “Tutorial on input shaping/time delay control of maneuvering flexible structures,” in *Proc. of the American Control Conf.*, May 2002, pp. 1717-1731.

[11] D. Iamratanakul and S. Devasia. “Minimum-time/energy output-transitions in linear systems,” in *Proc. of the American Control Conf.*, Jun. 2004, pp. 4831-4836.

[12] E. de Gelder, M. van de Wal, C. Scherer, C. Hol, and O. Bosgra. “Nominal and robust feedforward design with time domain constraints applied to a wafer stage.” *ASME J. Dyn. Sys., Meas., and Contr.*, vol. 128, pp. 204-215, Jun. 2006.

[13] A. Piazzi and A. Visioli. “Minimum-time system inversion for residual vibration reduction.” *IEEE/ASME Trans. Mechatronics*, vol. 5, no. 1, pp. 12-22, Mar. 2000.

[14] B. Rigney, “Adaptive settle-optimal control of servomechanisms,” Ph.D. dissertation, Dept. Elect. and Comp. Eng., Univ. of Colorado at Boulder, Boulder, CO, 2008.

[15] E. Gross and M. Tomizuka. “Experimental beam tip tracking control with a truncated series approximation to uncancelable inverse dynamics.” *IEEE Trans. Ctrl. Sys. Tech.*, vol. 2, no. 4, pp. 382-391, Dec. 1994.

[16] B. Potsaid, J. Wen, M. Unrath, D. Watt, and M. Alpay. “High Performance Motion Tracking Control for Electronic Manufacturing.” *ASME J. Dyn. Sys., Meas., and Contr.*, vol. 129, pp. 767-776, Nov. 2007.

[17] M. Tomizuka. “Zero phase error tracking algorithm for digital control.” *ASME J. Dyn. Sys., Meas., and Contr.*, vol. 109, pp. 65-68, Mar. 1987.

[18] D. Torfs, R. Vuerinckx, J. Swevers, and J. Schoukens. “Comparison of two feedforward design methods aiming at accurate trajectory tracking of the end point of a flexible robot arm.” *IEEE Trans. Ctrl. Sys. Tech.*, vol. 6, no. 1, pp. 2-14, Jan. 1998.

[19] K. Åström and B. Wittenmark. *Adaptive Control*. Menlo Park, CA: Addison-Wesley, 1995.

[20] B. Widrow and E. Walach. *Adaptive Inverse Control - A Signal Processing Approach*. Hoboken, NJ: Wiley, 2008.

[21] T. Tsao and M. Tomizuka. “Adaptive zero phase error tracking algorithm for digital control.” *ASME J. Dyn. Sys., Meas., and Contr.*, vol. 109, pp. 349-354, Dec. 1987.

[22] C. R. Johnson. *Lectures on Adaptive Parameter Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1988.