

# A Neural-Networks-Based Adaptive Disturbance Rejection Method and Its Application to the Control of Hard Disk Drives

Jason Levin<sup>1</sup>, Néstor O. Pérez-Arancibia<sup>2</sup>, Petros A. Ioannou<sup>1</sup>, *Fellow, IEEE*, and Tsu-Chin Tsao<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2560 USA

<sup>2</sup>Mechanical and Aerospace Engineering Department, University of California, Los Angeles, CA 90095-1597 USA

**This paper presents a neural-networks-based disturbance rejection adaptive scheme for dealing with repeatable and nonrepeatable runout simultaneously. The effectiveness of this method is demonstrated empirically on a commercial hard disk drive where the adaptive disturbance rejector is added to a baseline linear time-invariant (LTI) controller. The adaptive scheme can be broken into two subsystems: one subsystem is designed to suppress the repeatable runout (RRO) and the other to attenuate the residual disturbance and nonrepeatable runout (NRRO) by the use of radial basis functions. Two different methods for RRO suppression are employed in conjunction with the neural-networks-based NRRO rejector. The first one is an adaptive feedforward disturbance rejection scheme. The second is a repetitive controller. In both cases the neural modeled disturbance rejector is adapted online further increasing the track-following performance by as much as 6.4%. Experimental results of the schemes at various locations of the hard drive are included to demonstrate the general applicability of the approach on commercial drives. The total reduction of the error during track-following is measured to be as much as 25.4% respect to the baseline LTI controller.**

*Index Terms*—Adaptive disturbance rejection, hard disk drive, neural networks, real-time adaptive control.

## I. INTRODUCTION

**H**ARD DISK DRIVES (HDD) are a form of data storage that are present in just about every computer system. As the storage capacity grows so does the track density which puts tighter constraints on the servo control system. With tracks placed closer together in the radial direction there is a need to increase the positioning accuracy. There has been a large amount of research activity into two types of control problems: track-seeking and track-following [1]. The former deals with motion control of the head between tracks, and the latter with maintaining the head on the center of the HDD track. This paper deals with track-following which can be formulated as a disturbance rejection problem [2]–[4]. The disturbance can be separated into repeatable runout (RRO) and nonrepeatable runout (NRRO). The RRO is produced by imperfections and eccentricities on the tracks, while NRRO is produced by aggregated effects of disk drive vibrations, imperfections in the ball-bearings, and electrical noise. Research has been conducted over the years to cancel the effects of these disturbances and acquire better track following capabilities [2]–[12].

It has been shown that the RRO can be suppressed with adaptive feedforward methods [6], [13], and by repetitive control [7]. Both of these methods use the existing knowledge of the frequencies at which the RRO disturbance occurs to suppress its effect and obtain better tracking. The feedforward method employs the injection of the negative of an estimated sinusoidal disturbance model. The amplitude and phase of the sinusoidal disturbance are estimated online through a gradient update algorithm. An internal model is used to synthesize a linear time in-

variant controller that rejects sinusoidal disturbances of known frequencies in the repetitive control method.

The contribution of this paper is the experimental verification of a scheme that makes use of either the adaptive feedforward disturbance rejection scheme or the repetitive controller to eliminate the RRO in combination with another adaptive neural disturbance rejection scheme that focuses on reducing the tracking error even further. There has been work done in using neural networks for feedforward disturbance rejection [14]–[16]. Rejection of the disturbance torque for missile seekers using neural networks is presented in [14]. A multilayer neural network uses the measurable load disturbance to cancel to cancel the degrading effects through a feedforward controller. The conventional HDD does not have the luxury of extra sensors for the disturbance, so the only possible input to a neural network is the position error signal. In [15], a general approach with simulations show the benefit of adapting a dynamic neural network to cancel an unknown disturbance. Here, the idea of passing the disturbance estimate through an estimated plant inverse is used. The methods proposed in [16] use multilayer neural networks to model disturbances as outputs of dynamical systems and then expands the plant model to try to reduce the adverse effects.

Radial basis functions (RBF) have been used to model sea-clutter noise in radar applications [17]. The approach uses RBFs to remove the noise from radar signal data, since the clutter noise has been shown to be chaotic, thus providing the ability to detect small targets in the clutter. Training data is used to adapt the neural parameters before being implemented on actual test data. A similar approach was taken here except we add an extra term in the neural model to account for extra delays and there is no training set of data; the adaptive neural disturbance rejector is both adapted and implemented in real-time. Also the adaptation of the neural parameters uses a deadzone modification not present in [17], which allows adaptation to cease once the performance begins to degrade. The neural modeled disturbance rejection is added to either RRO rejection scheme to obtain better

Manuscript received June 27, 2008; revised November 05, 2008. Current version published April 17, 2009. Corresponding author: J. Levin (e-mail: levinj@usc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMAG.2009.2013339

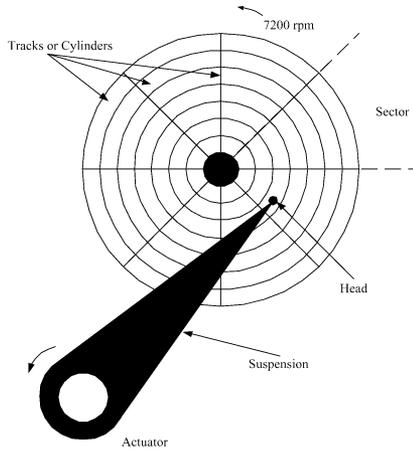


Fig. 1. Schematic idealization of the hard disk drive (HDD) system.

performance than the RRO scheme alone. This increase in performance is due to the ability of the adaptive neural network to provide a model for a dynamic nonlinear disturbance which is the residual from the RRO rejection schemes.

The paper is organized as follows. In Section II the experiment and real-time implementation issues are explained. Section III describes the baseline control to which the disturbance rejection schemes are added. These adaptive feedforward disturbance rejectors are explained in Section IV. Section V presents experimental results and conclusions are drawn in Section VI.

## II. DESCRIPTION OF THE EXPERIMENT

A HDD is a mechatronic device that uses rotating platters to store data. Information is recorded on, and read from concentric cylinders or tracks by read-write magnetic transducers called heads, that fly over the magnetic surfaces of the HDD platters. The position of the heads over the platters is changed by an actuator that consists of a coil attached to a link, which pivots about a ball bearing. This actuator connects to the head by a steel leaf called a suspension [18], [19]. This description of the HDD is shown in Fig. 1.

The control objective is to position the center of the head over the center of a data track. Thus, the typical measure of HDD tracking performance is the deviation of the center of the head from the center of a given track, which is often called track misregistration (TMR) [19]. There exist many indexes used to quantify TMR. Here we adopt

$$\text{TMR} = 3\sigma \quad (1)$$

where  $\sigma$  is the empirical standard deviation (STD) of the control error signal. It is common to express  $3\sigma$  as a percentage of the track pitch [4], [19], which must be less than 10% in order to be considered acceptable. TMR values larger than this figure will produce excessive errors during the reading and recording processes.

The experiment was performed with a two-platter (10 GB/platter), 4-head, 7200 rpm, commercial HDD, and a *Mathworks*<sup>®</sup> xPC Target system for control. The sample-hold rate of 9.36 KHz, used for communication, control and filtering, is

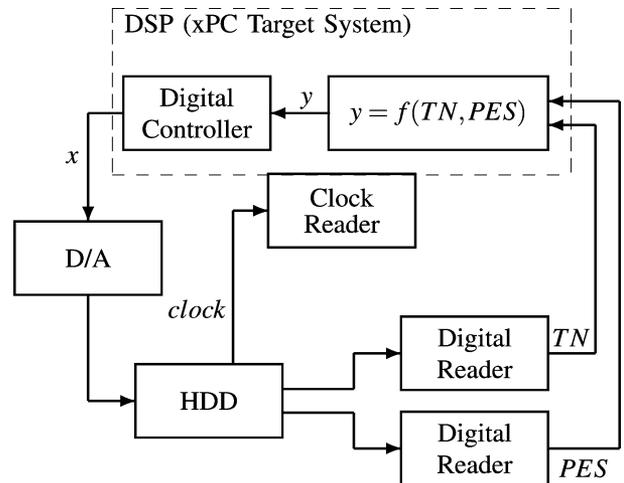


Fig. 2. Diagram of the experiment.

internally determined by the HDD and transmitted through a *clock* signal to the target PC used for control. Both systems must operate in a synchronized manner, as shown in the diagram of the experiment (Fig. 2).

The position of a given HDD head is digitally transmitted by the use of two signals. The first conveys the track number (TN) over where the head is positioned. The second is the position error signal (PES), which conveys the position of the head on the track pitch. Thus, the measured position  $y$  is a function of both the TN and PES signals.

The loop is closed when the digital controller outputs the sequence  $x$  which is converted into an analog signal to command the HDD actuator. At this stage, we pose the control problem in the discrete-time domain, defining the mapping from  $x$  to  $y$  as the open-loop plant  $P$ .

## III. BASELINE CONTROL

The work in this paper utilizes controllers that were previously developed and implemented in [3] as baseline controllers for adding the disturbance rejection schemes. This includes a simple LTI controller and a controller which is tuned using the inverse QR-RLS algorithm, both will be described briefly.

### A. Controller Design

An open-loop model of the HDD,  $\hat{P}$ , is first found by the method described in [3]. A simple LTI feedback controller  $C$  shown in Fig. 3 was designed using discrete-time domain classical techniques. It consists of a digital integrator and a digital notch filter. The integrator gain and notch parameters were tuned to maximize the output-disturbance rejection bandwidth.

The controller tuned with the inverse QR-RLS is developed using a model of the closed-loop plant  $G_1$  shown in Fig. 3. An identified model of this closed-loop plant,  $\hat{G}_1$ , is found using the *n4sid* algorithm and truncated to a 4th-order model. Now the control objective is to minimize the RMS value of the position error. The control problem is posed as a least squares problem and solved using the inverse QR-RLS algorithm in [20]. The algorithm is allowed to converge to steady-state and the controller is denoted as  $U(z)$  in Fig. 3.

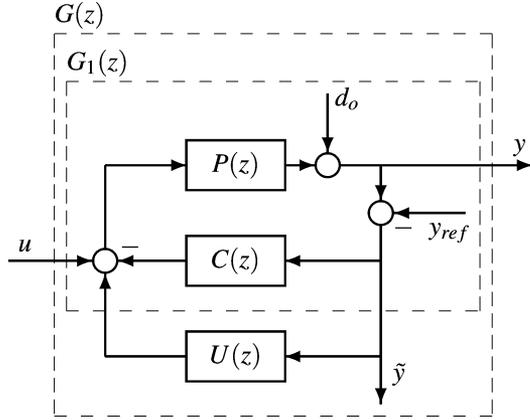


Fig. 3. Block diagram of the control system.  $P(z)$  = open-loop plant;  $C(z)$  = simple LTI controller;  $U(z)$  = converged inverse QR-RLS controller;  $y$  = position of the head;  $d_o$  = aggregate disturbance;  $y_{ref}$  = position reference;  $\tilde{y}$  = PES;  $u$  = control signal;  $G_1(z)$  = closed-loop plant with  $C(z)$ ;  $G(z)$  = closed-loop plant with  $C(z)$  and  $U(z)$ .

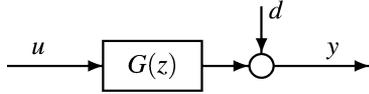


Fig. 4. Block diagram of the system used for disturbance rejection.  $G(z)$  = closed-loop system with baseline controllers;  $d$  = disturbance;  $u$  = control signal.

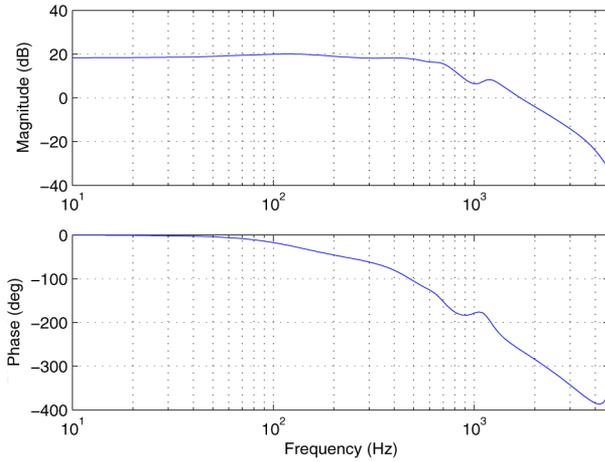


Fig. 5. Bode plot of identified close-loop system  $\hat{G}$ .

### B. Closed-Loop Model

The baseline LTI and inverse QR-RLS controllers are placed in the loop with the HDD dynamics and a closed-loop model from  $u$  to  $\tilde{y}$  is formed, denoted as  $G$ . The new system diagram is shown in Fig. 4, the output  $y$  of this system is the PES, which is the same as  $\tilde{y}$  in Fig. 3 since  $y_{ref}$  is constant. An identified model of this system,  $\hat{G}$ , is again found by the  $n4sid$  algorithm and truncated to a 10th-order model. The bode plot of the identified  $\hat{G}$  is seen in Fig. 5. This is the system that will be used for disturbance rejection throughout the rest of this paper.

## IV. DISTURBANCE REJECTION

The goal of the disturbance rejection schemes that will be added to the baseline control, explained above, is to reduce the PES which is  $y$  in Fig. 4. This will be done in two parts. First, the RRO will be suppressed with either the adaptive feedforward scheme demonstrated in [6] or a repetitive controller similar to the one demonstrated in [7]. Once the repeatable runout disturbance rejection is applied there is still disturbance that creates nonperfect tracking. The second part of the disturbance rejection scheme models the remaining disturbance using neural techniques and is adaptively updated online. This technique was demonstrated in [21].

### A. Adaptive Feed Forward RRO Disturbance Rejection

This RRO disturbance rejection scheme will be injecting a negative of the estimated disturbance into the system and has proven experimental results as shown in [6], [13]. This paper makes no modification to the algorithm other than applying it to a large number of frequencies. The repeatable runout (RRO) disturbance occurs at frequencies  $120m$  Hz where  $m = 1, 2, \dots, n$  due to the 7200 rpm speed of the disk. A control input is designed such that it will adaptively cancel this disturbance. The disturbance,  $d(k) = d_{RRO}(k)$  can be modeled as

$$d_{RRO}(k) = \sum_{i=1}^n a_i(k) \sin\left(\frac{2\pi ik}{N_{rev}}\right) + b_i(k) \cos\left(\frac{2\pi ik}{N_{rev}}\right) \quad (2)$$

where  $i$  is the index for the harmonic and  $N_{rev}$  is the number of samples per revolution.

If the system is modeled as in Fig. 4 then the output is

$$y(k) = G[u(k)] + d_{RRO}(k). \quad (3)$$

To cancel the disturbance the control signal should be  $u(k) = -\hat{G}^{-1}[\hat{d}_{RRO}(k)]$ . The identified inverse,  $\hat{G}^{-1}$ , will have an effect on the magnitude and phase of the disturbance estimate,  $\hat{d}_{RRO}(k)$ . Since the magnitude and phase of the sinusoidal disturbance is being estimated, the system inverse can be ignored and the new control signal becomes  $u(k) = -\hat{d}_{RRO}(k)$ . The disturbance estimate is

$$\hat{d}_{RRO}(k) = \sum_{i=1}^n \hat{a}_i(k) \sin\left(\frac{2\pi ik}{N_{rev}}\right) + \hat{b}_i(k) \cos\left(\frac{2\pi ik}{N_{rev}}\right). \quad (4)$$

The update equations for the estimated parameters are

$$\hat{a}_i(k) = \hat{a}_i(k-1) + \gamma_i y(k-1) \sin\left(\frac{2\pi ki}{N_{rev}} + \phi_i\right) \quad (5)$$

$$\hat{b}_i(k) = \hat{b}_i(k-1) + \gamma_i y(k-1) \cos\left(\frac{2\pi ki}{N_{rev}} + \phi_i\right) \quad (6)$$

where the  $\gamma_i$  are adaptation gains, chosen differently for each harmonic. A phase advance modification is added to reduce the sensitivity and allow for more harmonics to be canceled as was

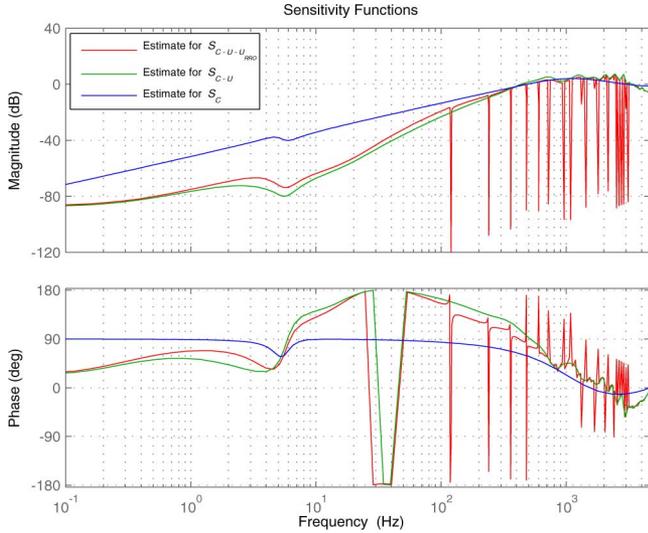


Fig. 6. Bode plots for the computed output sensitivity transfer functions  $\hat{S}_C = (1 + \hat{P}C)^{-1}$ ,  $\hat{S}_{C-U} = [1 + \hat{P}(C-U)]^{-1}$ , and  $\hat{S}_{C-U-U_{RRO}} = [1 + \hat{P}(C-U-U_{RRO})]^{-1}$ .

done previously in [6]. The  $\phi_i = \angle G(j\omega_i)$  and  $\omega_i$  is the angular frequency of the  $i$ th harmonic.

It was shown previously in [13] that the adaptive feed forward disturbance rejection scheme has an LTI equivalent representation. By treating this RRO rejection scheme as LTI, the sensitivity function  $S_{C-U-U_{RRO}} = [1 + P(C-U-U_{RRO})]^{-1}$  from  $d_o$  to  $y$  can be computed. Also, the shape of  $N_{C-U-U_{RRO}} = P(C-U-U_{RRO})[1 + P(C-U-U_{RRO})]^{-1}$  from  $y_{ref}$  to  $y$  is computed, where  $U_{RRO}$  is the LTI representation for the complete adaptive feedforward scheme with  $n = 33$  harmonics. Estimates for both transfer functions, denoted with  $\hat{\bullet}$ , are shown *in red* in Figs. 6 and 7 respectively. The attenuation at the RRO harmonic frequencies is very pronounced as is some amplification in the low frequency region. However the experimental results will show that this amplification is in fact an attenuation, contrary to what the sensitivity plot shows.

### B. Repetitive Control

Two prominent methods for dealing with RRO disturbances have been described in the literature. One is the adaptive feedforward rejection method described in the previous subsection. Another one is repetitive control, which uses the concept of *internal model* in [22] for synthesizing *linear time invariant* (LTI) controllers. In this paper, we employ a repetitive control scheme presented in [7], which has been demonstrated to be suitable for integrating repetitive and adaptive elements simultaneously.

The design is carried out as follows. First, we consider  $G$  to be stable and that  $\hat{G} = G$ , and we represent the aggregated effects of all the disturbances acting on the system by the output disturbance  $d$ , i.e.,  $d = [1 + P(C-U)]^{-1}d_o$ . This is illustrated in Fig. 4. Then we choose the internal model  $D = 1 - q(z, z^{-1})z^{-N}$ , where  $q$  is a zero-phase low-pass filter and  $N$  is the period of the periodical disturbance to be attenuated. Notice that the operator  $q$  allows us some flexibility over the frequency range of disturbances to be canceled while maintaining stability. The filter  $D$  has a combed shape with notches

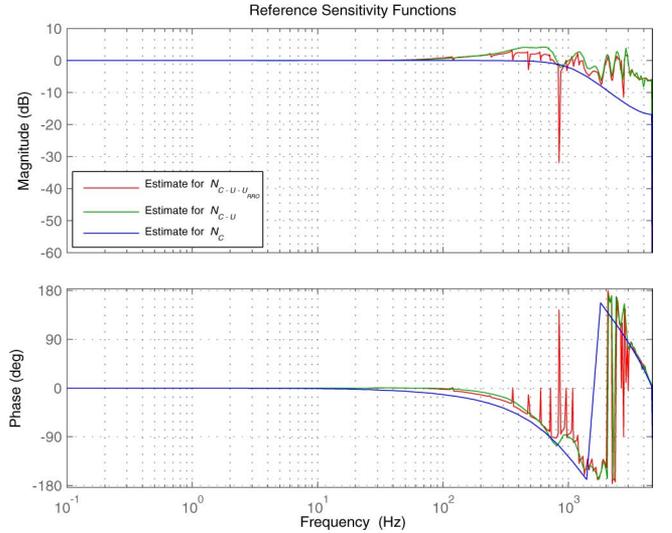


Fig. 7. Bode plots for the computed mappings  $\hat{N}_C$ ,  $\hat{N}_{C-U}$ , and  $\hat{N}_{C-U-U_{RRO}}$  from  $y_{ref}$  to  $y$ , for controllers  $C$ ,  $C-U$ , and  $C-U-U_{RRO}$ , respectively.

matching the frequencies of the periodic disturbance signals. Thus, ideally we would like to search for a filter  $K$  that makes the frequency response of the LTI system  $1 - KG$  close to zero at the same periodic frequencies. This is achievable by solving the Bézout identity

$$RD + KG = 1, \quad (7)$$

where  $R$  and  $K$  are the unknowns.

For (7) the existence of stable solutions for  $R$  and  $K$  will be assured if the numerators and denominators of  $G$  and  $D$  can be arranged to have a polynomial Diophantine equation satisfying the coprimeness condition in [23]. Furthermore, (7) characterizes a whole family of stabilizing internal model type repetitive controllers, because the system  $1 - GK_o$  is stable as long as  $G$  and  $K_o$  remain stable. Following the general guidelines in [24] and [25] a particular solution is presented here. The method starts by separating  $G$  into its minimum and nonminimum phase parts  $G_o$  and  $G_i$ , respectively. Thus,

$$G = \frac{B}{A} = \frac{B^+B^-}{A} = G_iG_o, \quad (8)$$

$$G_o = \frac{B^+}{A}, \quad G_i = B^-.$$

Where  $B^+$  and  $B^-$  are the cancelable and uncancelable parts of the numerator  $B$  of  $G$ . Now, substituting (8) into (7) we can write

$$RD + K'G_i = 1, \quad K' = KG_o. \quad (9)$$

Among the infinity many solutions to (9) it is verifiable by simple algebraic manipulations that one of the solutions is given by

$$R_o = \frac{1}{1 - (1 - \gamma G_i^* G_i) q z^{-N}}, \quad (10)$$

$$K'_o = q \gamma G_i^* z^{-N} R_o, \quad K_o = K'_o G_o^{-1}.$$

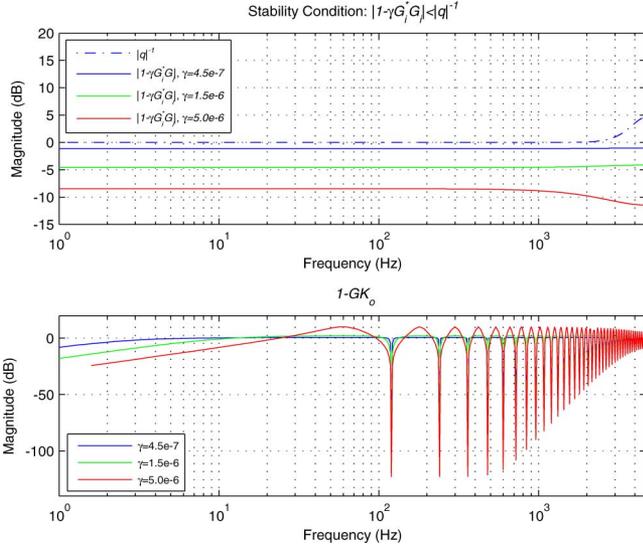


Fig. 8. *Top plot*: Sufficient stability condition for different values of  $\gamma$ . *Bottom plot*: Estimated sensitivity function from  $d$  to  $y$ ,  $1 - GK_o$ , for different values of  $\gamma$ .

Here,  $G_i^*$  is defined as  $G_i^*(z^{-1}) = G_i(z)$ , and  $\gamma$  as a positive real number.

At this point, questions on the causality and the stability of the controller  $K_o$  arise. The zero-phase filter  $q$  is noncausal and the plants  $G_i$  and  $G_i^*$  might not be causal as well. Nonetheless, the causality of  $K_o$  is guaranteed for a sufficiently large  $N$ , since  $z^{-N}$  is a factor of both  $(1 - \gamma G_i^* G_i)qz^{-N}$  and  $\gamma q G_i^* z^{-N}$ . Also, it is verifiable, by the use of the *small gain* theorem [26] that the stability of  $K_o$  and the stability of  $R_o$  are ensured by the sufficient condition

$$|1 - \gamma G_i^*(e^{j\theta})G_i(e^{j\theta})| < \frac{1}{|q(e^{j\theta})|}, \quad \forall \theta \in [0, \pi]. \quad (11)$$

In (11), the real number  $\gamma$  can be thought of as a stability and performance tuning parameter. Fig. 8 shows the fulfilment of the condition (11) and the achievable performances for three different values of  $\gamma$ :  $4.5 \times 10^{-7}$ ,  $1.5 \times 10^{-6}$  and  $5.0 \times 10^{-6}$ . Clearly, when considering the three cases in Fig. 8, there exists a trade-off between performance and stability robustness. For example, for the case  $\gamma = 5.0 \times 10^{-6}$  the condition (11) is amply satisfied, however, there exists a noticeable internotch amplification in the sensitivity function  $1 - GK_o$ , from  $d$  to  $y$ , shown on the bottom of Fig. 8. On the other hand, the converse is true for the case  $\gamma = 4.5 \times 10^{-7}$ , i.e., the internotch amplification is almost unnoticeable and the condition (11) is satisfied with a smaller amplitude. It is important to remark that there is no a linear relation between the magnitude of  $\gamma$  and performance or stability robustness and that the purpose of Fig. 8 is simply exemplify that  $\gamma$  can be thought of as a tuning parameter. Here, we choose  $\gamma = 4.5 \times 10^{-7}$  because this value is a good compromise.

The bottom plot in Fig. 8 shows the feedforward  $1 - GK_o$ , which is the sensitivity function from  $d$  to  $y$ . However, what is more interesting at this point is the shape of the overall sensitivity function  $\hat{S}_{C-U-U_{\text{REP}}} = [1 + P(C - U - U_{\text{REP}})]^{-1}$  from  $d_o$  to  $y$ , and also the shape of  $N_{C-U-U_{\text{REP}}} = P(C -$

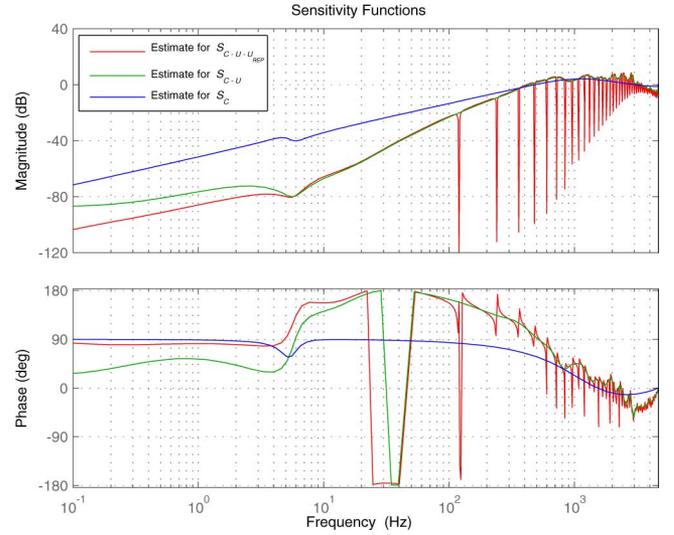


Fig. 9. Bode plots for the computed output sensitivity transfer functions  $\hat{S}_C = (1 + \hat{P}C)^{-1}$ ,  $\hat{S}_{C-U} = [1 + \hat{P}(C - U)]^{-1}$ , and  $\hat{S}_{C-U-U_{\text{REP}}} = [1 + \hat{P}(C - U - U_{\text{REP}})]^{-1}$ .

$U - U_{\text{REP}}][1 + P(C - U - U_{\text{REP}})]^{-1}$  from  $y_{\text{ref}}$  to  $y$ , with  $U_{\text{REP}} = -K_o(1 - K_o G)^{-1}$  (recall that  $d_o$  is the original open-loop output disturbance and  $y_{\text{ref}}$  is the reference signal). Estimates for both transfer functions, denoted with a hat  $\hat{\cdot}$ , are shown *in red* in Figs. 9 and 10, respectively. Notice that this internal-model-based controller is not only able to create deep notches but also to improve the rejection over low frequencies at the expense of some amplification on internotch regions that can be attenuated using the neural-networks method.

### C. Neural Modeled Disturbance Rejection

In the above sections, there have been several controllers developed. Initially there is a simple LTI controller designed to stabilize the HDD, then a controller tuned with the inverse QR-RLS is added to achieve better tracking performance. This control scheme can be yet improved using available information of the frequency of the RRO disturbance. Using this a priori knowledge the adaptive feedforward RRO disturbance rejection scheme and the repetitive controller are designed. With the RRO eliminated there still exists a disturbance that is nonlinear and dynamic. A neural model of this dynamic nonlinear disturbance is then created to reduce the tracking error even further. The new disturbance is modeled as

$$d(k) = d_{\text{RRO}}(k) + d_{\text{NN}}(k). \quad (12)$$

So the system output now becomes

$$y(k) = G[u(k)] + d_{\text{RRO}}(k) + d_{\text{NN}}(k). \quad (13)$$

The neural modeled disturbance rejector assumes that the disturbance from the RRO,  $d_{\text{RRO}}(k)$  is suppressed with either the adaptive feedforward rejection of repetitive controller. This means the only disturbance of concern at this point is the  $d_{\text{NN}}(k)$ . To cancel the disturbance the control signal should be  $u(k) = u_{\text{RRO}}(k) - \hat{G}^{-1}[d_{\text{NN}}(k)]$ , where  $u_{\text{RRO}}(k)$  is the input generated by either RRO suppression scheme. Since the identified model of the HDD is nonminimum phase the

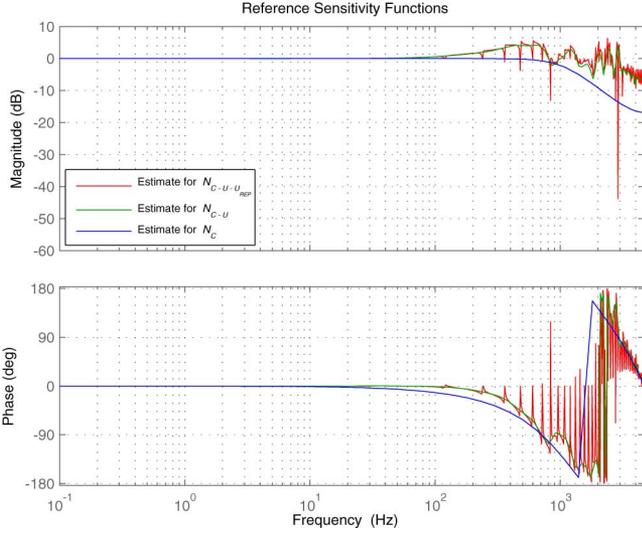


Fig. 10. Bode plots for the computed mappings  $\hat{N}_C$ ,  $\hat{N}_{C-U}$  and  $\hat{N}_{C-U-U_{\text{REP}}}$  from  $y_{\text{ref}}$  to  $y$ , for controllers  $C$ ,  $C-U$ , and  $C-U-U_{\text{REP}}$ , respectively.

inverse is unstable. The unstable zero of  $G$  is reflected across the unit circle and the inverse is taken. This new inverse,  $\bar{G}^{-1}$ , is used in the computation of the control signal, making it  $u(k) = u_{\text{RRO}}(k) - \bar{G}^{-1}[\hat{d}_{\text{NN}}(k)]$ , and therefore causes an extra delay that will be dealt with.

The following disturbance rejection scheme uses Gaussian radial basis functions (RBF) from neural networks to attempt to model the disturbance. The disturbance estimate takes the form

$$\hat{d}_{\text{NN}}(k) = \sum_{q=1}^L \sum_{i=1}^M \theta_{q,i}(k) R_{q,i} \hat{d}(k - \delta \cdot (q-1) - 1) \quad (14)$$

$$R_{q,i} = \Psi_i(\hat{d}(k - \delta \cdot (q-1) - 1)) \quad (15)$$

where  $R_{q,i}$  is computed using an RBF and the  $i$ th Gaussian RBF is

$$\Psi_i(x) = \exp \left[ - \left( \frac{x - c_i}{\beta} \right)^2 \right]. \quad (16)$$

The parameters that specify the shape of the  $i$ th Gaussian RBF are the center  $c_i$  and the width  $\beta$ . There are a total of  $M$  Gaussian RBFs, and their centers are linearly spaced across the range of input. The current disturbance estimate,  $\hat{d}_{\text{NN}}(k)$ , is a function of  $L$  previous disturbances that are spaced  $\delta$  samples apart.

The reason for the spacing  $\delta$  is the delay associated with passing the disturbance estimate through the system inverse. One method of coping with the delay would be to estimate the disturbance at the next sample, and then use this estimate to create another future estimate, and continue iterating to find some  $\hat{d}_{\text{NN}}(k+\Delta)$  in the future [15]. This method did not work as the estimation error grew with each future estimate. Instead, the disturbance is thought of as a function of previous evenly spaced disturbances. The  $\hat{d}_{\text{NN}}(k)$  can be viewed as a future disturbance estimate when compared to the HDD sample rate. It should be noted that the algorithm still creates a new disturbance estimate at every sample of the HDD output.

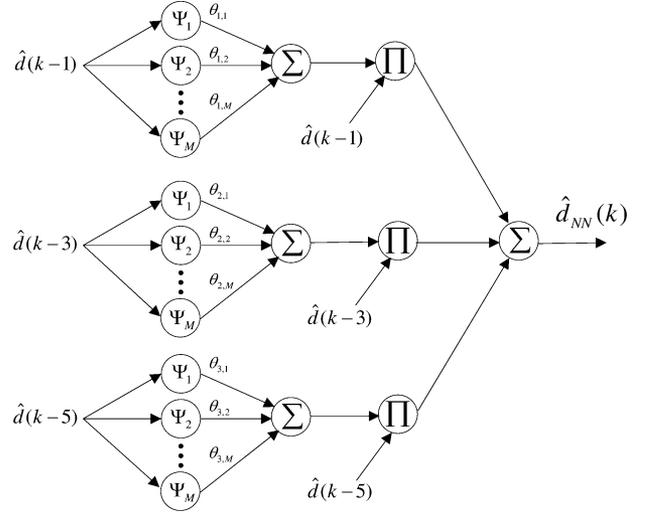


Fig. 11. Simple example of how  $\hat{d}_{\text{NN}}(k)$  is computed from previous values of the disturbance. Here  $\delta = 2$  and  $L = 3$ .

The model of the disturbance is motivated by the assumption that the disturbance is a nonlinear function of previous disturbance values. The  $\hat{d}(k - \delta \cdot (q-1) - 1)$  term that is multiplied by the output of the RBF in (14) is the added term that allows the model to work for the disturbance in this HDD. This added term makes the model different from previously used RBF neural predictors [17], [15]. Now the disturbance can be thought of as an autoregressive filter with spacing  $\delta$  and nonlinear coefficients that are modeled with the RBF's. A simple example with  $\delta = 2$  and  $L = 3$  is shown in Fig. 11 to help view the modeling of the disturbance. The  $\hat{d}(k - \delta \cdot (q-1) - 1)$  used in the estimate is not produced by the estimator but rather is a measured disturbance estimate which can be calculate from an identified model of the system,  $\hat{G}(z)$ , as follows:

$$\hat{d}(k-1) = y(k-1) - \hat{G}[u(k-1)]. \quad (17)$$

The unknown parameters should be updated with the current modeling error, which is the plant output  $y(k-1)$ , and the parameters that caused that error,  $\hat{d}(k - \delta q - 1)$ . This leads to the update equations

$$\theta_{q,i}(k) = \begin{cases} \theta_{q,i}(k-1) + \alpha_{q,i}(k) & \varepsilon_{\text{new}} \leq \rho \cdot \varepsilon_{\text{old}} \\ \theta'_{q,i} & \text{otherwise} \end{cases} \quad (18)$$

$$\alpha_{q,i}(k) = \frac{l_{\text{NN}}}{m^2(k)} y(k-1) \bar{R}_{q,i} \hat{d}(k - \delta q - 1) \quad (19)$$

$$\bar{R}_{q,i} = \Psi_i(\hat{d}(k - \delta q - 1)) \quad (20)$$

$$m^2(k) = 1 + m_s(k) \quad (21)$$

$$m_s(k) = \delta_0 m_s(k-1) + \hat{d}^2(k-1). \quad (22)$$

Every  $N_\varepsilon$  samples the following are computed

$$\varepsilon_{\text{new}} = \sum_{n=1}^{N_\varepsilon} y^2(k-n) \quad (23)$$

$$\varepsilon_{\text{old}} = \begin{cases} \varepsilon_{\text{new}} & \varepsilon_{\text{new}} < \varepsilon_{\text{old}} \\ \varepsilon_{\text{old}} & \text{otherwise} \end{cases} \quad (24)$$

$$\theta'_{q,i} = \begin{cases} \theta_{q,i}(k) & \varepsilon_{\text{new}} < \varepsilon_{\text{old}} \\ \theta'_{q,i} & \text{otherwise.} \end{cases} \quad (25)$$

The update is an instantaneous gradient algorithm with a couple robustness modifications. The adaptation, or learning, rate is  $l_{\text{NN}}$  and is greater than zero. The update term is normalized with a dynamic term to add robustness, this term is calculated in (22). The parameter  $\delta_0$  is chosen between 0 and 1.

The other robustness modification is one that is added to stop adaptation when the performance starts to degrade. In practice the estimation error will never become zero and so the parameters will continue to update. There will be a point at which the estimation error is small and on the same level as the noise and modeling error. Usually, a simple deadzone is added to stop adaptation when the current error is below some threshold. In the HDD application the estimation error, which is the HDD output, is noisy and must be averaged over  $N_\varepsilon$  samples. Instead of averaging, the sum squared error is easier to calculate online via (23). The adaptation will continue as long as this new sum squared error,  $\varepsilon_{\text{new}}$ , is less than the old sum squared error,  $\varepsilon_{\text{old}}$ , to within some small range. The  $\rho$  term is selected to be greater than 1 to allow adaptation even if the sum squared error did not decrease. This gives some room for noisy measurements and lets the algorithm continue. If the algorithm is doing a good job and the new sum squared error is strictly less than the old, the current  $\theta_{q,i}(k)$ 's are saved and the  $\varepsilon_{\text{old}}$  is updated. The  $\theta_{q,i}(k)$ 's are saved so that when the sum squared error is too large, the algorithm can revert back to the best known parameters.

#### D. Parameter Tuning for Neural Model

The parameters were first tuned in an offline simulation. Experimental disturbance data was collected by allowing the LTI and converged inverse QR-RLS controllers to run and measuring the PES. This PES data is the disturbance for both of the adaptive disturbance rejection schemes. The number of harmonics and adaptive gain for each harmonic of the RRO rejection scheme were tuned first. This was done through Monte-Carlo simulations while varying the adaptive gain. The number of harmonics was simply increased until no performance benefit was seen. Once these values were fixed the parameters of the neural model disturbance rejection could be tuned. Monte-Carlo simulations were run while varying the number of basis functions,  $M$ , adaptive gain,  $l_{\text{NN}}$ , and the number of past parameters,  $L$ .

The  $N_\varepsilon$  and  $\rho$  needed to be tuned online once at a single head/track combination to ensure adaptation stopped at the appropriate time to maximize performance. It should be noted that each time the algorithm begins  $\varepsilon_{\text{old}}$  is initialized by  $\varepsilon_{\text{old}} = \varepsilon_0$ , where  $\varepsilon_0$  is a design parameter that should be chosen large at first. This way after the first  $N_\varepsilon$  samples  $\varepsilon_{\text{old}}$  can be updated via (24).

#### E. Stability Analysis of Neural Model

The adaptive neural modeled disturbance rejection scheme can be placed in a form that follows the framework of adaptive estimators in [27]. Starting with the modeled disturbance

$$d_{\text{NN}}(k) = \sum_{q=1}^L \sum_{i=1}^M \theta_{q,i}^*(k) R_{q,i} \hat{d}(k - \delta \cdot (q-1) - 1) \quad (26)$$

$$R_{q,i} = \Psi_i(\hat{d}(k - \delta \cdot (q-1) - 1)) \quad (27)$$

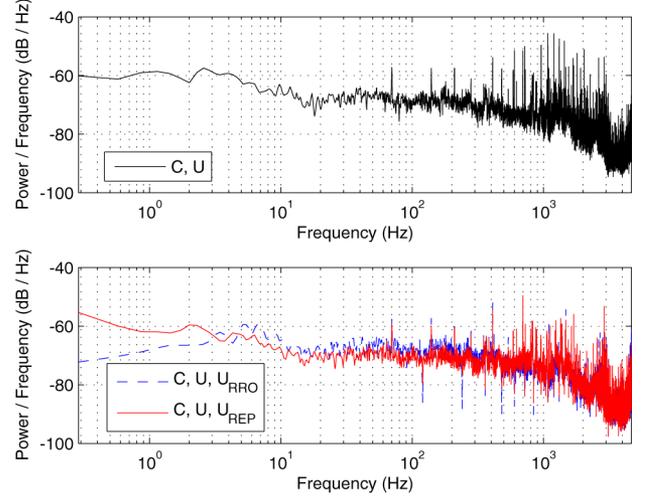


Fig. 12. Experiment performed on head 0 and track 15 000. **Top Plot:** PSD with no adaptive disturbance rejection ( $C$  and  $U$ ). **Bottom Plot:** PSD with adaptive feedforward RRO disturbance rejection ( $C$ ,  $U$ , and  $U_{\text{RRO}}$ ) and PSD with repetitive control ( $C$ ,  $U$ , and  $U_{\text{REP}}$ ).

and placing it in the form of

$$z(k) = \theta^{*T} \phi(k) + \eta(k) \quad (28)$$

where

$$z(k) = d(k) \quad (29)$$

$$\theta^* = [\theta_{1,1}^*, \dots, \theta_{L,M}^*]^T \quad (30)$$

$$\phi(k) = [f_{1,1}, \dots, f_{L,M}]^T \quad (31)$$

$$f_{q,i} = R_{q,i} \hat{d}(k - \delta \cdot (q-1) - 1) \quad (32)$$

and  $\eta(k)$  is the modeling error. The estimation model and estimation error are given as

$$\hat{z} = \theta^T(k) \phi(k) \quad (33)$$

$$\varepsilon(k) = \frac{z(k) - \hat{z}}{m^2(k)} = \frac{z(k) - \theta^T(k) \phi(k)}{m^2(k)} \quad (34)$$

where  $m(k)$  is the normalizing signal designed to bound  $|\phi(k)|$  and  $|\eta(k)|$  from above. Only the  $\hat{d}(k - \delta \cdot (q-1) - 1)$  part of the  $\phi(k)$  needs to be bounded by  $m(k)$  since the RBF's,  $R_{q,i}$ , are bounded by definition. Using the gradient law in (18) the parameters are adaptively updated. As shown in [27] the adaptive laws defined above will guarantee that  $\theta_{q,i}(k), \varepsilon(k) \in \ell_\infty$  and  $\varepsilon(k), |\theta_{q,i}(k) - \theta_{q,i}(k-1)| \in S(g_0 + \eta_0^2)$ , where  $\eta_0$  is an upper bound of  $(|\eta(k)|/m(k)) \leq \eta_0$ , and  $g_0$  is bounded by  $\rho \cdot \varepsilon_0 \geq g_0 \geq \rho \cdot \varepsilon_{\text{old}}$ .

## V. EXPERIMENTAL RESULTS

In the experiments described here, the sample-and-hold rate for control was 9.36 KHz, externally determined by the HDD clock. The controllers used in this section include the LTI controller, denoted by  $C$ . The  $U$  that is used is a 36th-order tuned inverse QR-RLS, tuned using the head 0 over track 15 000. The  $U_{\text{RRO}}$  is the adaptive feedforward disturbance rejection scheme used to cancel  $n = 33$  harmonics and the gains were tuned offline using a previously acquired PES signal.  $U_{\text{REP}}$  is the repetitive controller which is designed entirely offline as well. Lastly,

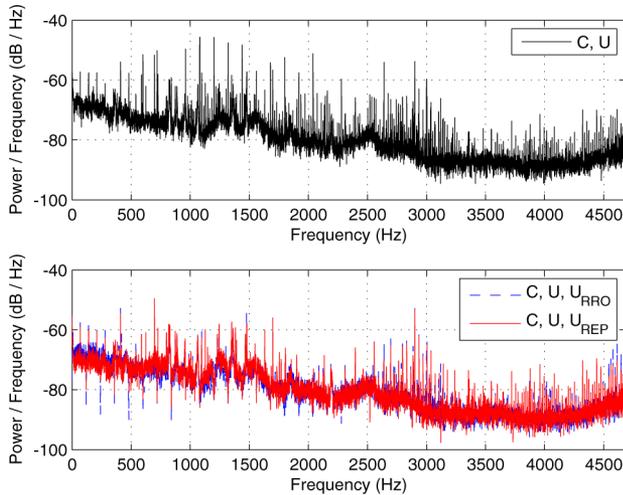


Fig. 13. Experiment performed on head 0 and track 15 000. **Top Plot:** PSD with no adaptive disturbance rejection ( $C$  and  $U$ ). **Bottom Plot:** PSD with adaptive feedforward RRO disturbance rejection ( $C$ ,  $U$ , and  $U_{RRO}$ ) and PSD with repetitive control ( $C$ ,  $U$ , and  $U_{REP}$ ).

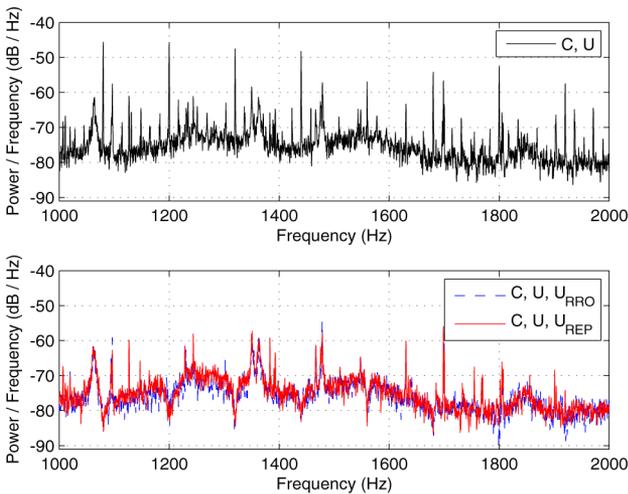


Fig. 14. Closeup of PSD in Fig. 13. **Top Plot:** PSD with no adaptive disturbance rejection ( $C$  and  $U$ ). **Bottom Plot:** PSD with adaptive feedforward RRO disturbance rejection ( $C$ ,  $U$ , and  $U_{RRO}$ ) and PSD with repetitive control ( $C$ ,  $U$ , and  $U_{REP}$ ).

the  $K_{NN}$ , which is the adaptive neural disturbance rejection scheme with  $M = 15$ ,  $L = 9$ ,  $l_{NN} = 1$ ,  $N_\epsilon = 500$ ,  $\rho = 1.5$ , and  $\delta = 2$ . The  $a_i(k)$  and  $b_i(k)$  of the RRO disturbance rejector, and the  $\theta_{q,i}(k)$  of the neural scheme are all initialized to zero and adapted online.

The adaptive feedforward RRO disturbance rejection scheme and repetitive controller are first tested in the experiment. Each RRO rejection scheme was tested with the  $C$  and  $U$  controllers but independently of one another. A 10-s period of HDD PES was collected using the head 0 over track 15 000 and the PSDs are calculated. The logarithmic and linear PSDs are shown in Figs. 12 and 13, respectively. The *top plot* in each shows the baseline control with no RRO disturbance rejection, and the *bottom plot* is the case when the RRO is eliminated. There are a couple differences between the two experimental results that should be pointed out. The  $U_{REP}$  appears to have better low

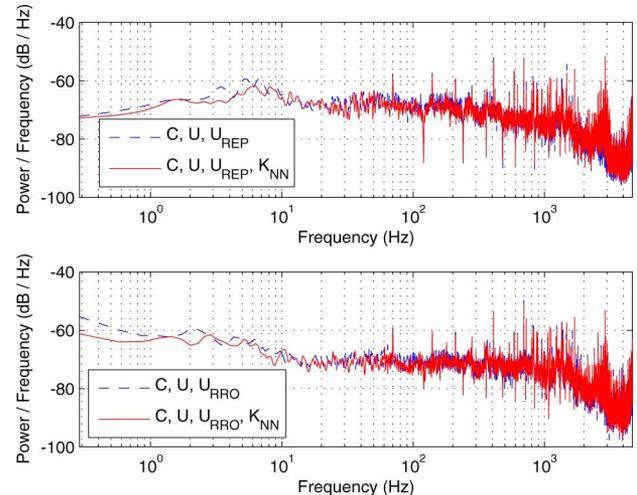


Fig. 15. Experiment performed on head 0 and track 15 000. **Top Plot:** PSD with repetitive control and neural modeled disturbance rejection ( $C$ ,  $U$ ,  $U_{REP}$ , and  $K_{NN}$ ) and PSD with repetitive control ( $C$ ,  $U$ , and  $U_{REP}$ ). **Bottom Plot:** PSD with adaptive feedforward RRO and neural modeled disturbance rejection ( $C$ ,  $U$ ,  $U_{RRO}$ , and  $K_{NN}$ ) and PSD with adaptive feedforward RRO disturbance rejection ( $C$ ,  $U$ , and  $U_{RRO}$ ).

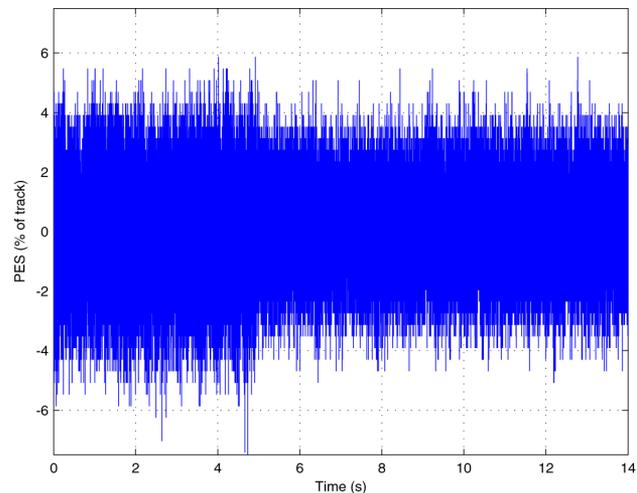


Fig. 16. Time series data from experiment performed on head 0 and track 15 000. At 5 s, the adaptive disturbance rejection is switched on ( $C$ ,  $U$ ,  $U_{RRO}$ , and  $K_{NN}$ ).

frequency attenuation and this is because the zero frequency is eliminated in the repetitive controller but not in the  $U_{RRO}$  scheme. Although the zero frequency can also be eliminated with the  $U_{RRO}$  scheme this will not help our ultimate goal of improving our TMR metric which is  $3\sigma$ . Also both figures show that the  $U_{REP}$  has deeper notches at lower frequency than the  $U_{RRO}$ , which is true in this design but may not be in general. The  $U_{RRO}$  does attenuate the overall disturbance in the frequency range from 5 to 400 Hz, and not just RRO harmonics like the  $U_{REP}$ . Fig. 14 clearly shows the rejection of the RRO harmonics at multiples of 120 Hz. Both RRO rejection schemes do a similar job in this frequency region of canceling the repeatable-runout but at the same time amplifying the disturbance in between the harmonics.

Both RRO rejection schemes perform well and some tabulated results are in Fig. 17 and Table I where the  $3\sigma$  value of the

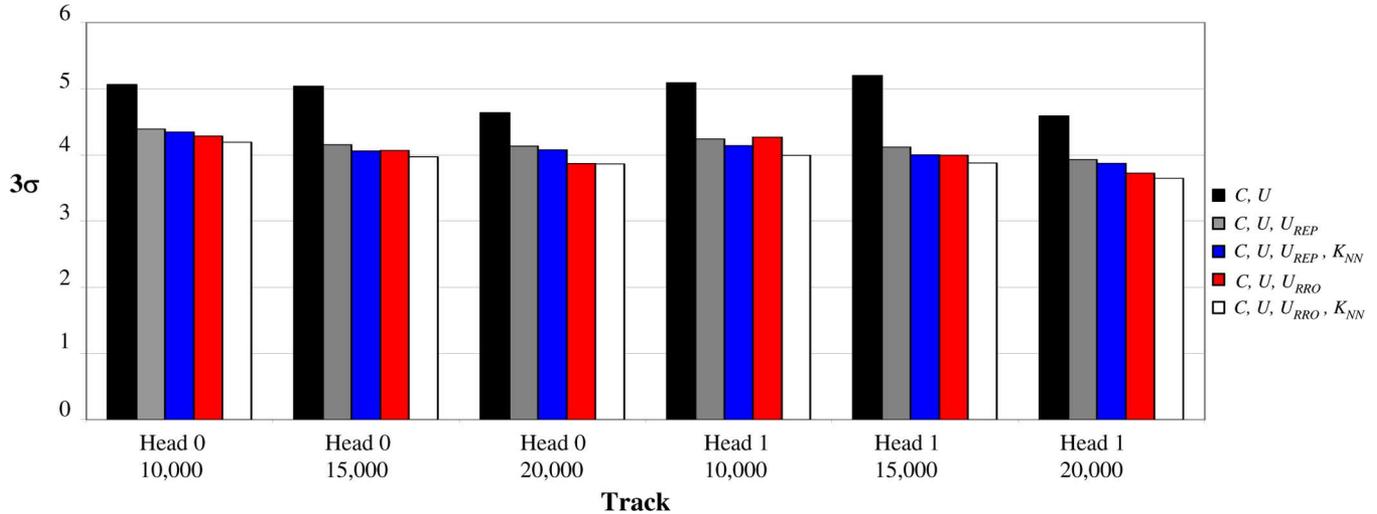


Fig. 17.  $3\sigma$  value of the position error signal as a percentage of the track width for various head and track locations.

TABLE I  
 $3\sigma$  VALUE OF THE POSITION ERROR SIGNAL (PES) AS A PERCENTAGE OF THE TRACK WIDTH

	Head 0			Head 1		
	$y_{ref} = 10^4$	$y_{ref} = 1.5 \cdot 10^4$	$y_{ref} = 2 \cdot 10^4$	$y_{ref} = 10^4$	$y_{ref} = 1.5 \cdot 10^4$	$y_{ref} = 2 \cdot 10^4$
$C$ and $U$	5.0641	5.0371	4.6399	5.0905	5.1992	4.5877
$C$ , $U$ , and $U_{REP}$	4.3918	4.1531	4.1311	4.2409	4.1162	3.9287
$C$ , $U$ , and $U_{RRO}$	4.2809	4.0642	3.8694	4.2666	3.9954	3.7213
$C$ , $U$ , $U_{REP}$ , and $K_{NN}$	4.3452	4.0574	4.0770	4.1402	3.9981	3.8746
$C$ , $U$ , $U_{RRO}$ , and $K_{NN}$	4.1926	3.9714	3.8634	3.9930	3.8760	3.6445

PES as a percentage of the track width is calculated for different heads and tracks of the HDD. In just about every scenario the adaptive feed forward RRO disturbance rejection scheme outperforms the repetitive controller, which could be due to the tuning in this case. However, the neural modeled disturbance rejection scheme improves the TMR in every case tested and is able to work in conjunction with either RRO rejection scheme to make a mean improvement in of 2.3%. The PSD of the neural scheme in combination with the RRO rejection schemes is displayed in Fig. 15, where the low frequency disturbance rejection from 0–400 Hz is improved in both cases. The impact of the adaptive schemes is most noticeable in Fig. 16, where the time series data from the HDD at head 0 and track 15 000 has been plotted. Initially only the baseline controllers are active and then at 5 s the adaptive disturbance rejectors,  $U_{RRO}$  and  $U_{NN}$ , are switched on.

The results compiled in Table I come from experiments performed on the same day around the same time. This was done for comparison purposes, however the different schemes were run hundreds of times on various days to assure the ability to work under various conditions. The disturbance was quantitatively different during various tests and the improvement from the RRO schemes varied. However, in all of the tests completed the neural disturbance rejector was able to improve the performance by similar amounts seen in this paper. In all the cases tested the RRO rejection schemes performed quite well even though the schemes were tuned on a single track and head combination. Different tracks and heads result in different RRO dis-

turbance spectrums, but the measurable tracking performance of the schemes presented points to the robustness of the design.

Each of the disturbance rejection schemes possesses implementation issues. As discussed in [13] the adaptive feedforward disturbance rejector will in general require more computation than the repetitive controller. However, in this experiment there are 66 parameters that are updated online using sine lookup tables, while the repetitive controller is a filter with 192 internal states. The repetitive controller requires a mathematical model of the plant, while the adaptive feedforward method can be implemented with just experimental frequency response data. The most computation intensive part of this scheme is the neural modeled disturbance rejector. For this experiment it requires a model of the plant, a model of stable inverse, and the online updating of 270 neural model parameters. The identified plant and identified stable inverse both use 10 states and their outputs must also be computed online. This large amount of computation required may be too large for production disk drives, where computation and memory budgets are of importance. However, this is one application of the scheme where the benefit in tracking to high performance systems can be experimentally verified. The disturbance rejection scheme can possibly be applied to other systems where computational power is not of paramount concern.

## VI. CONCLUSION

This paper presented an adaptive disturbance rejection scheme for a HDD. The RRO of the disturbance is attenuated

through the use of an adaptive feedforward model or a repetitive controller and the remaining disturbance is modeled with neural techniques using radial basis functions. The complete control scheme was experimentally tested without retuning at various head and track positions on a HDD to show the improvement in TMR by as much as 25.4%. However, the addition of the neural disturbance rejector improved the tracking by as much as 6.4%, but the performance increase was measurable on all track and head combinations tested. It is difficult to quantitatively compare the results presented here to other published results since variations in HDDs and disturbances will have a large impact, but another adaptive repetitive control scheme was done on the same HDD and published in [7]. The NRRO suppression ability of the neural scheme is superior to the adaptive scheme tested in [7], while both add some complexity in terms of computation. The increase in computation for the neural scheme results in better performance while remaining implementable on the experimental setup. Unlike the QR-RLS scheme in [3], where the order can be increased by 100 resulting in simulated performance benefits but the computation time required is too great for the computing power of the experimental setup.

The performance benefit of the neural scheme is clear from the experimental results provided, but the added complexity may make implementation on commercial disk drives difficult. As technology and research improves, this scheme presented may be simplified and the computations accelerated with new algorithms. Examples from the past are RLS filters, Kalman filters, and matrix operations, which have all progressed in terms of their ability to utilize modern processing power. Future research should focus on improving the speed of the neural algorithm while also expanding the range of applications. The HDD is one application of the disturbance rejection scheme that demonstrates the ability to reduce tracking error through innovative techniques. This scheme can also be applied to other systems where computational power is not the driving constraint.

#### ACKNOWLEDGMENT

This work was supported in part by NSF Award CMS 0510921, NASA Grant NCC4-158, and the NSF Nanoscale Science and Engineering Center for Scalable and Integrated Nanomanufacturing (SINAM) under Grant DMI 0327077.

#### REFERENCES

- [1] B. Chen, T. Lee, and V. Venkataramanan, *Hard Disk Drive Servo Systems*. London, U.K.: Springer-Verlag, 2002.
- [2] R. Horowitz and B. Li, "Adaptive control for disk file actuators," in *Proc. 34th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 1995, pp. 655–660.
- [3] N. O. Pérez Arancibia, T.-C. Tsao, and S. Gibson, "Adaptive tuning and control of a hard disk drive," in *Proc. of the American Control Conf.*, New York, NY, Jun. 2007.
- [4] K. Krishnamoorthy and T.-C. Tsao, "Adaptive-Q with LQG stabilization feedback and real time computation for disk drive servo control," in *Proc. American Control Conf.*, Boston, MA, June 2004, pp. 1171–1175.
- [5] M. White, M. Tomizuka, and C. Smith, "Improved track following in magnetic disk drives using a disturbance observer," *IEEE/ASME Trans. Mechatronics*, vol. 5, no. 1, pp. 3–11, Mar. 2000.
- [6] A. Sacks, M. Bodson, and P. Khosla, "Experimental results of adaptive periodic disturbance cancellation in a high performance magnetic disk drive," *J. Dyn. Syst., Meas. Control*, vol. 118, pp. 416–424, Sep. 1996.
- [7] N. O. Pérez Arancibia, C.-Y. Lin, T.-C. Tsao, and S. Gibson, "Adaptive-repetitive control of a hard disk drive," in *Proc. 46th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 4519–4524.
- [8] S.-C. Wu and M. Tomizuka, "Repeatable runout compensation for hard disk drives using adaptive feedforward cancellation," in *Proc. of the American Control Conf.*, Minneapolis, MN, Jun. 2006, pp. 382–387.
- [9] J. Li and T.-C. Tsao, "Rejection of repeatable and non-repeatable disturbances for disk drive actuators," in *Proc. of the American Control Conf.*, San Diego, CA, Jun. 1999, pp. 3615–3619.
- [10] J. Zhang, R. Chen, G. Guo, and T.-S. Low, "Modified adaptive feedforward runout compensation for dual-stage servo system," *IEEE Trans. Magn.*, vol. 36, no. 5, pp. 3581–3584, Sep. 2000.
- [11] Q.-W. Jia, Z.-F. Wang, and F.-C. Wang, "Repeatable runout disturbance compensation with a new data collection method for hard disk drive," *IEEE Trans. Magn.*, vol. 41, no. 2, pp. 791–796, Feb. 2005.
- [12] H. Melkote, Z. Wang, and R. J. McNab, "An iterative learning controller for reduction of repetitive runout in disk drive," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, no. 3, pp. 467–473, May 2006.
- [13] A. Sacks, M. Bodson, and W. Messner, "Advanced methods for repeatable runout compensation," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1031–1036, Mar. 1995.
- [14] C.-L. Lin and Y.-H. Hsiao, "Adaptive feedforward control for disturbance torque rejection in seeker stabilizing loop," *IEEE Trans. Contr. Syst. Technol.*, vol. 9, no. 1, pp. 108–121, Jan. 2001.
- [15] G. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Trans. Neural Networks*, vol. 14, no. 2, pp. 360–376, Mar. 2003.
- [16] S. Mukhopadhyay and K. Narendra, "Disturbance rejection in nonlinear systems using neural networks," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 63–72, Jan. 1993.
- [17] G. Hennessey, H. Leung, A. Drosopoulos, and P. C. Yip, "Sea-clutter modeling using a radial-basis-function neural network," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 3, pp. 358–372, July 2001.
- [18] D. Abramovitch and G. Franklin, "A brief history of disk drive control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 28–42, June 2002.
- [19] W. Messner and R. Ehrlich, "A tutorial on controls for disk drives," in *Proc. American Control Conf.*, Arlington, VA, Jun. 2001, pp. 408–420.
- [20] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.
- [21] J. Levin, N. O. Pérez Arancibia, P. Ioannou, and T.-C. Tsao, "Adaptive disturbance rejection for disk drives using neural networks," in *Proc. American Control Conf.*, Seattle, WA, Jun. 2008.
- [22] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, Sept. 1976.
- [23] K. J. Åström and B. Wittenmark, *Computer Controlled Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [24] M. Tomizuka, T.-C. Tsao, and K.-K. Chew, "Analysis and synthesis of discrete-time repetitive controllers," *ASME J. Dyn. Syst., Meas., Contr.*, vol. 111, no. 3, pp. 353–358, Sept. 1989.
- [25] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *ASME J. Dyn. Syst., Meas., Contr.*, vol. 111, no. 1, pp. 65–68, Mar. 1987.
- [26] M. A. Dahle and I. J. Diaz-Bobillo, *Control of Uncertain Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [27] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA: SIAM, 2006.

**Jason Levin** (S'02) received the B.S. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2004 and the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 2005. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering-Systems, University of Southern California.

He has been an academic part time employee of the Jet Propulsion Laboratory, Pasadena, CA, working as an Engineer since 2005. His research interests include nonlinear and adaptive control, intelligent disturbance rejection, and the control of hard disk drives, high performance aircraft, optical systems, and hypersonic vehicles.

**Néstor O. Pérez-Arancibia** (S'05–M'08) received the Ingeniero and M.Eng. degrees from the Pontificia Universidad Católica de Chile in 2000 and the Ph.D. degree from the Mechanical and Aerospace Engineering Department at the University of California, Los Angeles (UCLA) in 2007.

Since October 2007, he has been a Postdoctoral Scholar with the Laser Beam Control Laboratory and also with the Mechatronics and Controls Laboratory in the Mechanical and Aerospace Engineering Department at UCLA. His current interests include feedback control, adaptive filtering, adaptive optics, and mechatronics.

**Petros A. Ioannou** (S'80–M'83–SM'89–F'94) received the B.Sc. degree in mechanical engineering with "first class honors" from the University College London, London, U.K., in 1978 and the M.S. in mechanical engineering and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1980 and 1982, respectively.

In 1982, he joined the Department of Electrical Engineering—Systems, University of Southern California, Los Angeles. He is currently a Professor in the same department and the Director of the Center of Advanced Transportation Technologies. He also holds a joint appointment with the Department of Aerospace and Mechanical Engineering. He is the author/coauthor of five books and over 150 research papers in the area of controls, neural networks, nonlinear dynamical systems, and intelligent transportation systems. His research interests are in the areas of adaptive control, neural networks, nonlinear systems, vehicle dynamics and control, intelligent transportation systems, and marine transportation.

Dr. Ioannou was a recipient of the Outstanding Transactions Paper Award from IEEE Control Systems Society and the recipient of a 1985 Presidential Young Investigator Award for his research in Adaptive Control in 1984. He has been an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the *International Journal of Control, Automatica*, and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He also served as a member of the Control System society on IEEE Intelligent Transportation Systems (ITS) Council Committee, and his center on advanced transportation

technologies was a founding member of Intelligent Vehicle Highway System (IVHS) America, which was later renamed ITS America. He is currently an Associate Editor at Large of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and a Chairman of the International Federation of Automatic Control (IFAC) Technical Committee on Transportation Systems.

**Tsu-Chin Tsao** (S'86–M'88) received the B.S. degree in engineering from National Taiwan University, Taiwan, in 1981, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California, Berkeley, in 1984 and 1988, respectively.

In 1999, he joined the faculty of the University of California, Los Angeles (UCLA), where he is currently a Professor with the Mechanical and Aerospace Engineering Department. He served 11 years on the faculty of the Mechanical and Industrial Engineering Department at the University of Illinois, Urbana-Champaign. His research interests include control systems and mechatronics.

Dr. Tsao was a recipient of the *ASME Journal of Dynamic Systems, Measurement, and Control* Best Paper Award for papers published in the journal in 1994, the Outstanding Young Investigator Award from ASME Dynamic Systems and Control Division, in 1997, and the Hugo S. Shuck Best Paper Award from American Automatic Control Council in 2002.