2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Adaptive Disturbance Rejection for Disk Drives Using Neural Networks

Jason Levin, Néstor O. Pérez Arancibia, Petros A. Ioannou, and Tsu-Chin Tsao

Abstract— **This paper presents the experimental verification of an adaptive feedforward disturbance rejection scheme for hard disk drives. The control scheme is shown to reduce the error during track-following by as much as 11.8%. The adaptive disturbance rejector is added to a baseline controller and broken into two parts: a part to suppress the repeatable runout and another to attenuate the residual disturbance by using radial basis functions. Both parts are adapted online to ensure good disturbance rejection. The experimental results are included to demonstrate the effectiveness on a commercial hard disk drive.**

I. INTRODUCTION

Hard 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 non-repeatable 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], and by adaptive repetitive control [7]. In this paper we use the adaptive feedforward disturbance rejection scheme to eliminate the RRO and then focus on reducing the disturbance even further. There has also been work done in using neural networks for feedforward disturbance rejection [13]–[15]. Radial basis functions (RBF) have been used to model sea-clutter noise in radar applications [16], a similar approach was taken here. We experimentally verify a neural model of the disturbance that is adapted online and used for disturbance rejection to obtain more precise track-following. 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 feedfoward 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 [17], [18]. 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) [18]. There exist many indexes used to quantify TMR. Here we adopt

$$TMR = 3\sigma. \tag{1}$$

Where σ is the empirical standard deviation (STD) of the control error signal. It is common to express 3σ as a percentage of the track pitch [4], [18], 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 2-platter (10 GB/platter), 4-head, 7200 rpm, commercial HDD, and a Mathworks xPC Target system for control. The sample-hold rate of 9.36 KHz, used for communication, control and filtering, is 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

This work has been supported in part by NSF Award No. CMS-0510921

J. Levin and P. Ioannou are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089-2560, USA. levinj@usc.edu, ioannou@usc.edu

N. O. Pérez Arancibia and T.-C. Tsao are with the Mechanical and Aerospace Engineering Department, University of California, Los Angeles, CA, 90095–1597, USA. nestor@seas.ucla.edu, ttsao@seas.ucla.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

control problem in the discrete-time domain, defining the mapping from x to y as the open-loop plant P.

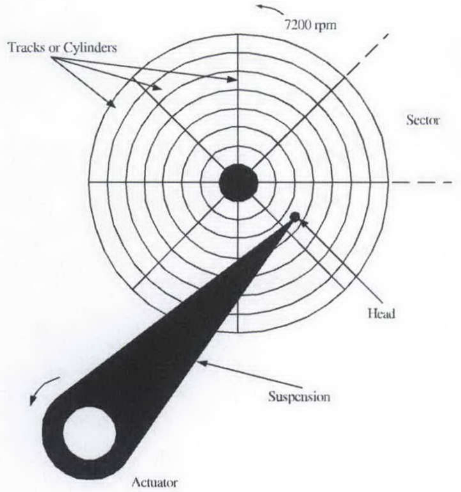

Fig. 1. Schematic idealization of the hard disk drive (HDD) system.

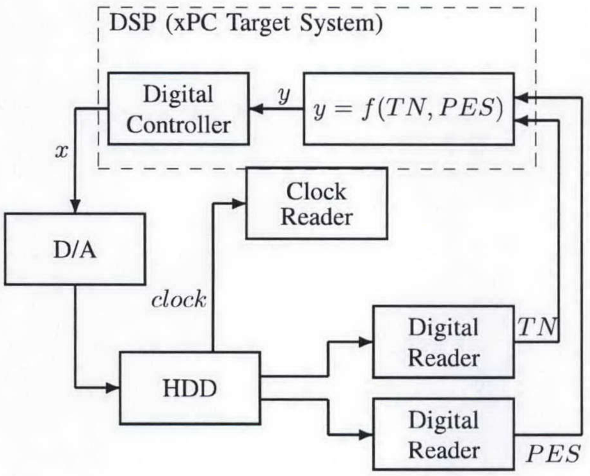

Fig. 2. Diagram of the experiment.

III. Baseline Control

The work in this paper utilizes controllers that were previously developed and implemented in [3] as baseline controllers for adding adaptive 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 4^{th} 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 [19]. The algorithm is allowed to converge to steady-state and the controller is denoted as $U(z)$ in Fig. 3.

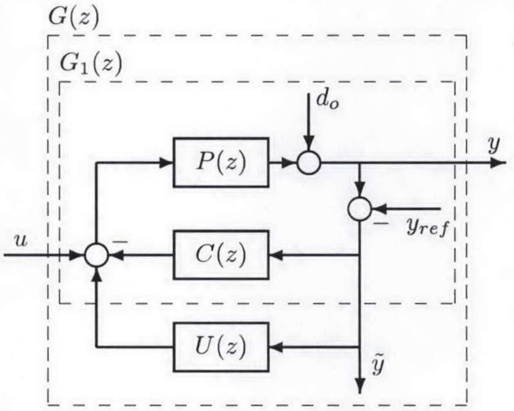

Fig. 3. Block diagram of the control system. $P(z)$ = open-loop plant; $C(z)$ = simple LTI controller; $U(z)$ = converged inverse QR-RLS controller; y = position of the head; d_0 = aggregate disturbance; y_{ref} = position reference; $\tilde{y}$ = PES; u = control signal; $G_1(z)$ = closed-loop plant with $C(z)$; $G(z)$ = closed-loop plant with $C(z)$ and $U(z)$.

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 10^{th} 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.

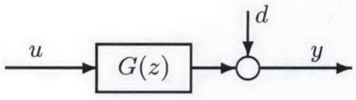

Fig. 4. Block diagram of the system used for disturbance rejection. $G(z)$ = closed-loop system with baseline controllers; d = disturbance; u = control signal.

IV. Adaptive Feedforward Rejection

The goal of the disturbance rejection problem is to reduce the PES, y in Fig. 4, by injecting the negative of an estimate of the disturbance. The problem is divided into two parts:

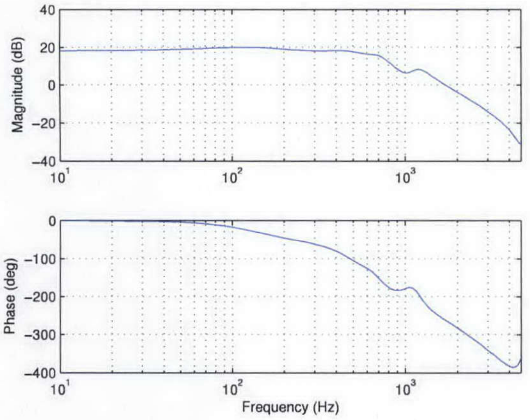

Fig. 5. Bode plot of identified close-loop system $\hat{G}$

repeatable runout disturbance rejection and neural modeled disturbance rejection. The former has proven experimental results as shown in [6]. This paper makes no modification to the algorithm other than applying it to a large number of frequencies. Once the repeatable runout disturbance rejection is applied there is still disturbance that creates non-perfect tracking. This remaining disturbance is modeled using neural techniques and adaptively updated online.

A. Repeatable Runout Disturbance Rejection

The repeatable runout (RRO) disturbance occurs at frequencies $120m$ Hz where $m = 1, 2, ..., n$ 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

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

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 figure 4 then the output is

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

To cancel the disturbance the control signal should be $u(k) = -\hat{G}^{-1}(z)[\hat{d}_{RRO}(k)]$. The identified inverse, $\hat{G}^{-1}(z)$, 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

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

The update equations for the estimated parameters are

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

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

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

B. Neural Modeled Disturbance Rejection

Since the disturbance does not consist entirely of harmonics from the rotation of the disk, another disturbance rejection algorithm is added. The new disturbance is modeled as

$$d(k) = d_{RRO}(k) + d_{NN}(k). \tag{7}$$

So the system output now becomes

$$y(k) = G(z)[u(k)] + d_{RRO}(k) + d_{NN}(k) \tag{8}$$

To cancel the disturbance the control signal should be $u(k) = -\hat{d}_{RRO}(k) - \hat{G}^{-1}(z)[\hat{d}_{NN}(k)]$. Since the identified model of the HDD is non-minimum phase the inverse is unstable. The unstable zero of $G(z)$ is reflected across the unit circle and the inverse is taken. This new inverse, $\bar{G}^{-1}(z)$, is used in the computation of the control signal, making it $u(k) = -\hat{d}_{RRO}(k) - \bar{G}^{-1}(z)[\hat{d}_{NN}(k)]$, 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

$$\hat{d}_{NN}(k) = \sum_{q=1}^{L} \sum_{i=1}^{M} \theta_{q,i}(k) R_{q,i} \hat{d}(k - \delta \cdot (q-1) - 1) \tag{9}$$

$$R_{q,i} = \Psi_i\left(\hat{d}(k - \delta \cdot (q-1) - 1)\right) \tag{10}$$

Where $R_{q,i}$ is computed using an RBF and the ith gaussian RBF is

$$\Psi_i(x) = exp\left[-\left(\frac{x - c_i}{\beta}\right)^2\right] \tag{11}$$

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

The reason for the spacing δ 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 $\hat{d}_{NN}(k + \Delta)$ in the future [14]. 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 $\hat{d}_{NN}(k)$ can be viewed as a future disturbance estimate when compared to the HDD sample rate. It should be noted that the algorithm

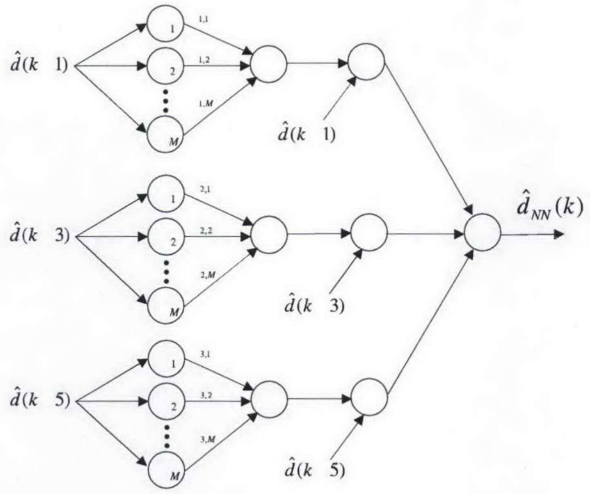

Fig. 6. Simple example of how $\hat{d}_{NN}(k)$ is computed from previous values of the disturbance. Here $\delta = 2$ and $L = 3$.

still creates a new disturbance estimate at every sample of the HDD output.

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 (9) 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 [14], [16]. Now the disturbance can be thought of as an autoregressive filter with spacing δ and nonlinear coefficients that are modeled with the RBF's. A simple example with $\delta = 2$ and $L = 3$ is shown in Fig. 6 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

$$\hat{d}(k-1) = y(k-1) - \hat{G}(z)[u(k-1)] \tag{12}$$

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

$$\theta_{q,i}(k) = \begin{cases} \theta_{q,i}(k-1) + \alpha_{q,i}(k) & \varepsilon_{new} \leq \rho \cdot \varepsilon_{old} \\ \theta'_{q,i} & otherwise \end{cases} \tag{13}$$

$$\alpha_{q,i}(k) = \frac{l_{NN}}{m^2(k)} y(k-1)\bar{R}_{q,i}\hat{d}(k-\delta q-1) \tag{14}$$

$$\bar{R}_{q,i} = \Psi_i\left(\hat{d}(k-\delta q-1)\right) \tag{15}$$

$$m^2(k) = 1 + m_s(k) \tag{16}$$

$$m_s(k) = \delta_0 m_s(k-1) + \hat{d}^2(k-1) \tag{17}$$

Every N_ε samples the following are computed

$$\varepsilon_{new} = \sum_{n=1}^{N_\varepsilon} y^2(k-n) \tag{18}$$

$$\varepsilon_{old} = \begin{cases} \varepsilon_{new} & \varepsilon_{new} < \varepsilon_{old} \\ \varepsilon_{old} & otherwise \end{cases} \tag{19}$$

$$\theta'_{q,i} = \begin{cases} \theta_{q,i}(k) & \varepsilon_{new} < \varepsilon_{old} \\ \theta'_{q,i} & otherwise \end{cases} \tag{20}$$

The update is an instantaneous gradient algorithm with a couple robustness modifications. The adaptation, or learning, rate is l_{NN} and is greater than zero. The update term is normalized with a dynamic term to add robustness, this term is calculated in (17). The parameter δ_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_ε samples. Instead of averaging, the sum squared error is easier to calculate online via (18). The adaptation will continue as long as this new sum squared error, ε_{new}, is less than the old sum squared error, ε_{old}, to within some small range. The ρ 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 ε_{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.

C. Parameter Tuning

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_{NN}, and the number of past parameters, L.

The N_ε and ρ 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 ε_{old} is initialized by $\varepsilon_{old} = \varepsilon_0$, where ε_0 is a design parameter that should be chosen large at first. This way after the first N_ε samples ε_{old} can be updated via (19).

D. 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 [20]. Starting with the modeled disturbance

$$d_{NN}(k) = \sum_{q=1}^{L}\sum_{i=1}^{M} \theta^*_{q,i}(k) R_{q,i}\hat{d}(k-\delta\cdot(q-1)-1) \tag{21}$$

$$R_{q,i} = \Psi_i\left(\hat{d}(k-\delta\cdot(q-1)-1)\right) \tag{22}$$

And placing it in the form of

$$z(k) = \theta^{*T}\phi(k) + \eta(k) \tag{23}$$

Where

$$z(k) = d(k) \tag{24}$$

$$\theta^* = [\theta^*_{1,1}, \ldots, \theta^*_{L,M}]^T \tag{25}$$

$$\phi(k) = [f_{1,1}, \ldots, f_{L,M}]^T \tag{26}$$

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

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

$$\hat{z} = \theta^T(k)\phi(k) \tag{28}$$

$$\epsilon(k) = \frac{z(k)-\hat{z}}{m^2(k)} = \frac{z(k)-\theta^T(k)\phi(k)}{m^2(k)}, \tag{29}$$

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 (13) the parameters are adaptively updated. As shown in [20] the adaptive laws defined above will guarantee that $\theta_{q,i}(k), \epsilon(k) \in \ell_\infty$ and $\epsilon(k), |\theta_{q,i}(k) - \theta_{q,i}(k-1)| \in S(g_0 + \eta_0^2)$, where η_0 is an upper bound of $\frac{|\eta(k)|}{m(k)} \leq \eta_0$, and g_0 is bounded by $\rho\cdot\varepsilon_0 \geq g_0 \geq \rho\cdot\varepsilon_{old}$.

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 36^{th} order converged inverse QR-RLS, tuned using the head 0 over trhe track 15,000. The K_{RRO} is the adaptive feedforward disturbance rejection scheme used to cancel $n = 33$ harmonics. Lastly the K_{NN}, which is the adaptive neural disturbance rejection scheme with $M = 15, L = 9, l_{NN} = 1, N_\varepsilon = 500, \rho = 1.5$, 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 performance of the algorithms can be seen in Fig. 7 where the metric for comparison is the 3σ value of the PES as a percentage of track width. The controllers were implemented on 3 different tracks on 2 heads of the disk drive. Although the adaptive controllers were tuned on head 0 and track 15,000 they still perform very well on other head/track combinations. In all cases the addition of the adaptive neural disturbance rejector, K_{NN}, provided the best tracking performance. The K_{NN} improves the PES tracking performance on head 0 by 6%, 4.8%, and 10% at track 10,000, 15,000, and 20,000 respectively. The improvement on head 1 is 11.8%, 6.6%, and 7.7% on track 10,000, 15,000, and 20,000 respectively.

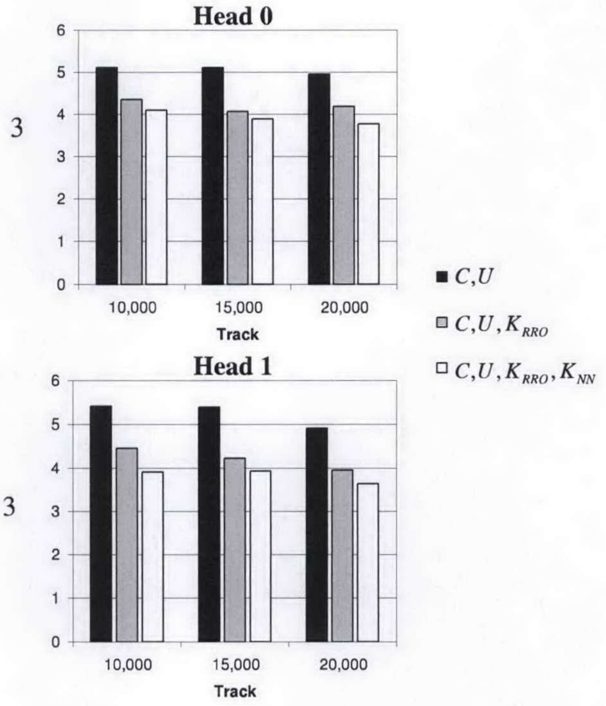

Fig. 7. 3σ value of the position error signal as a percentage of the track width.

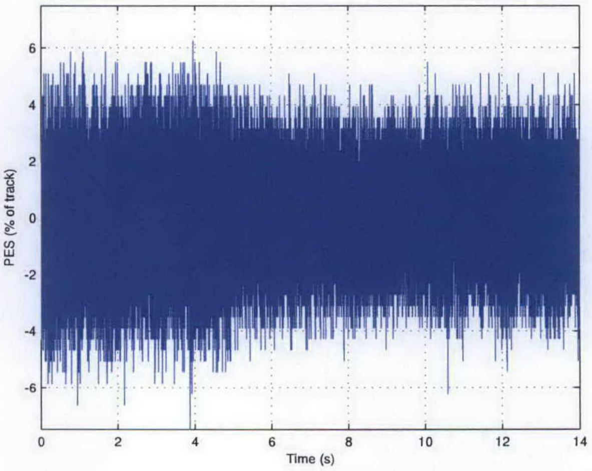

Fig. 8. Time series data from experiment performed on head 0 and track 15,000. At 5 seconds the adaptive disturbance rejection is switched on (K_{RRO} and K_{NN}).

The impact of the adaptive schemes can be clearly seen in Fig. 8, where the time series data from the HDD at head 0 and track 15,000 has been plotted. Initially only the baseline

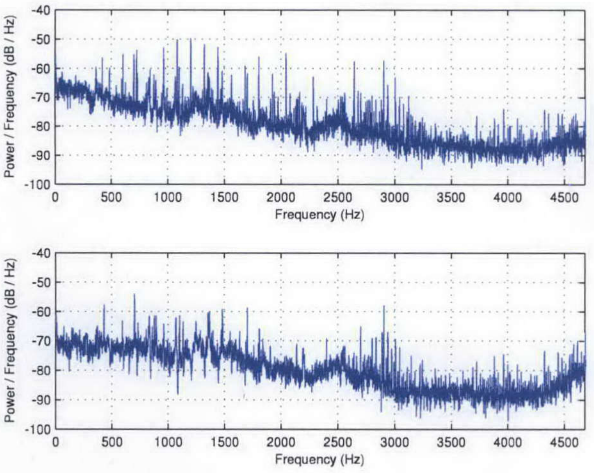

Fig. 9. Experiment performed on head 0 and track $15,000$. *Top Plot:* PSD with no adaptive disturbance rejection (C and U). *Bottom Plot:* PSD with adaptive disturbance rejection. (C, U, K_{RRO}, and K_{NN}).

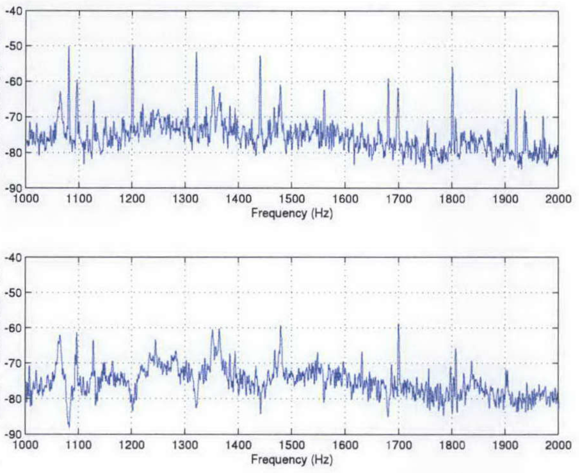

Fig. 10. Closeup of PSD in Fig. 9. *Top Plot:* PSD with no adaptive disturbance rejection (C and U). *Bottom Plot:* PSD with adaptive disturbance rejection. (C, U, K_{RRO}, and K_{NN}).

controllers are active and then at 5 seconds the adaptive disturbance rejectors, K_{RRO} and K_{NN}, are turned on. The PSDs are shown in Fig. 9, where low frequency attenuation and high frequency amplification is present. The closeup view in Fig. 10 displays the elimination of the RRO which occurs at frequencies of $120m$ Hz where $m = 1, 2, ..., 33$.

VI. Conclusion

This paper presented an adaptive feedforward disturbance rejection scheme for a HDD. The RRO of the disturbance is attenuated through the use of an adaptive feedforward model and the remaining disturbance is modeled with neural techniques using radial basis functions. The control scheme was experimentally tested without retuning at various head and track positions on a HDD to show the improvement in TMR.

References

[1] B. Chen, T. Lee, and V. Venkataramanan, *Hard Disk Drive Servo Systems*. London: 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, "Adative tuning and control of a hard disk drive," in *Proc. of the American Control Conference*, 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. of the American Control Conference*, Boston, MA, Jun. 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 Transactions on 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," *Trans. ASME, J. Dyn. Syst., Meas. Control*, vol. 118, pp. 416–424, Sept. 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.* 46^{th} *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 Conference*, Minneapolia, MN, Jun. 2006, pp. 382–387.

[9] J. Li and T.-C. Tsao, "Rejection of repeatable and non-repeatbale disturbances for disk drive actuators," in *Proc. of the American Control Conference*, San Diego, CA, Jun. 1999, pp. 3615–3619.

[10] J. Zhang, R. Chen, G. Guo, and T.-S. Low, "Modified adaptive feedforward runout compenation for dual-stage servo system," *IEEE Transactions on Magnetics*, vol. 36, no. 5, pp. 3581–3584, Sept. 2000.

[11] Q.-W. Jia, Z.-F. Wang, and F.-C. Wang, "Repeatable runout disturbance compenation with a new data collection method for hard disk drive," *IEEE Transactions on Magnetics*, vol. 41, no. 2, pp. 791–796, Feb. 2005.

[12] H. Melkote, Z. Wang, and R. J. McNab, "An iterative learning controller for reducation of repetitive runout in disk drive," *IEEE Transactions on Control systems technology*, vol. 14, no. 3, pp. 467–473, May 2006.

[13] C.-L. Lin and Y.-H. Hsiao, "Adaptive feedforward control for disturbance torque rejection in seeker stabilizing loop," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 108–121, Jan. 2001.

[14] G. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 360–376, Mar. 2003.

[15] S. Mukhopadhyay and K. . Narendra, "Disturbance rejection in nonlinear systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 4, no. 1, pp. 63–72, Jan. 1993.

[16] 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.

[17] 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.

[18] W. Messner and R. Ehrlich, "A tutorial on controls for disk drives," in *Proc. of the American Control Conference*, Arlington, VA, Jun. 2001, pp. 408–420.

[19] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York, NY: Wiley, 2003.

[20] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA: SIAM, 2006.