2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Fast Feedforward Vibration Rejection Algorithms based on Orthogonalization

Jinwen Pan[1,2], Zhi Chen[2], Yong Wang[1] and Roberto Horowitz[2]

***Abstract*— Vibration rejection using feedforward control architecture is presented in this paper showing that both input and output vibration rejection can be utilized as one problem. Wiener solution to this problem is derived by using numerical data. Two fast convergence feedforward control algorithms are developed and analyzed to solve the slow convergence rate of Fx-LMS algorithm when underlying inputs are highly correlated. Theoretical results show that the Orth.-X algorithm can always improve the convergence rate over Fx-LMS algorithm while the Orth.-A algorithm outperforms the Fx-LMS algorithm conditionally. Then numerical examples are given in details to show the procedure of analyzing the provided algorithms. Moreover, comprehensive simulation results coincide with the numerical analysis.**

***Index Terms*— Vibration rejection, Fx-LMS, orthogonal transform, eigenvalue spread, fast convergence, autocorrelation function, cross-correlation function**

I. INTRODUCTION

Control methods for disturbance rejection have attracted great interest from researchers both in controls and signal processing communities ever since 1930's [1]. The term "disturbance" can be "noise" or "vibration" in reality, resulting in active noise cancellation or vibration rejection [2], [3]. Due to the knowledge of the disturbance, the controller can be designed either in a feedback, or a feedforward fashion. A pictorial example given in [4], where non-negligible non-repeatable runout (NRRO) exists in a hard disk drive (HDD). Usually NRRO cannot be measured and what is available to control design is its spectrum, therefore, only feedback controller can be considered to have the ability to suppress the vibration within its bandwidth. However, if we somehow know the disturbance, a feedforward controller can be designed to reject the disturbance. In [5], frequencies of the disturbance are available, therefore, both direct [6] and indirect [7] adaptive methods are developed based on the known frequencies, known as narrow-band disturbance rejection. Meanwhile, if we can measure the vibration with a sensor, then a disturbance rejection can be considered no matter the disturbance is narrow- or broad-band [8]. In this case, the filtered-x least meas square (Fx-LMS) is known as an effective method to reduce the disturbance effect at the performance side [2].

*This work was supported by ASTC.

[1]Jinwen Pan and Yong Wang are with the Department of Automation, University of Science and Technology of China, Hefei, 230027, Anhui Province, China `yongwang@ustc.edu.cn`

[2]Zhi Chen and Roberto Horowitz are with the Department of Mechanical Engineering, University of California, Berkeley, Berkely, 94720, California, United States `horowitz@me.berkeley.edu`

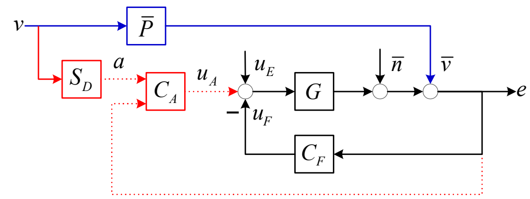

Fig. 1. Vibration at output side.

When the eigenvalues of the input signal underlying correlation matrix are widely spread, LMS type algorithms that use a transversal tapped delay line (TDL) structure gives poor convergence characteristics even though they are simple and efficient [1]. The recursive least square (RLS) was developed as a method to extract past information to dissociate the present input signal, but suffers from high computational cost [9]. Another choice to decorrelate the input signal is to use a transform such as the discrete Fourier transform (DFT), the discrete cosine transform (DCT) or the lattice predictor [10]. These approaches considered as self-orthogonalizing adaptive filters or transform domain adaptive filters (TDAF) usually save computational cost and are independent of the input signal characteristics. They can be analyzed by comparing the eigenvalue spread of underlying correlation matrix before and after the transform.

This paper discuss the vibration rejection in an feedforward fashion, meaning that a signal that is correlated to the vibration, is available to use. Two lattice predictor based algorithms are developed and analyzed to outperform the Fx-LMS algorithm in regarding the convergence rate.

Firstly, cases that vibration is either at output side or the input side are addressed and utilized as one problem. The Wiener solution (optimal) of the vibration rejection problem is given in Section.II. Two TDAF algorithms based on lattice orthogonal transform are provided in Section.**??** aimed to speed up the convergence rate over conventional Fx-LMS algorithm. The adaptive lattice-based orthogonalization and analysis are given in Section.IV. Numerical examples and simulations are given in Section.V while Section.VI draws the conclusions.

II. PROBLEM STATEMENT

A. Vibration at Output

The blocks G and C_F in Fig. 1 respectively denote a linear-time-invariant (LTI) plant and an LTI feedback compensator that form a stable closed–loop system. This nominal controller can be continuous or discrete time, and

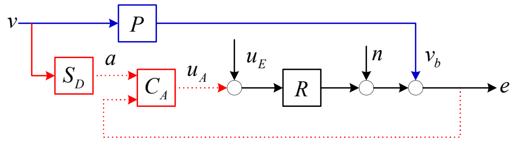

Fig. 2. General case.

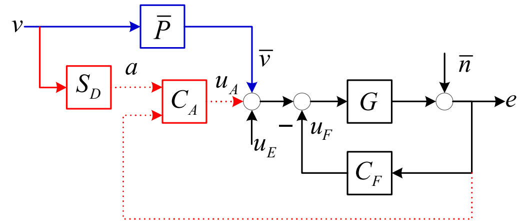

Fig. 3. Vibration at input side.

it generally provides disturbance rejection across a broad frequency spectrum. On the other hand, the feedforward adaptive controller, denoted by C_A, provides compensation for the special disturbance denoted by $\bar{v}(k)$. The source of this disturbance, $v(k)$ pollutes the system after passing through an unknown transfer function (primary path) $\bar{P}$. The signal $a(k)$ is an implicit measurement of $v(k)$ by using an additional sensor represented by S_D. Since our design does not depend on whether the plant/nominal controller are continuous or discrete time, we assume that both G and C_F are discrete time systems to make notations simpler and z is the discrete time frequency domain variable.

The nominal feedback controller C_F is designed to attenuate broad–band disturbance $\bar{n}$ that contaminates the output. The special disturbance that should be compensated by the adaptive controller is denoted by $\bar{v}$, and without loss of generality, we assume that it contaminates the plant output. The case that the disturbance contaminates the plant input will be discussed rigorously later and will be shown that for both cases, the controller design procedure is the same. From Fig.1, we have

$$e = \frac{G}{1+GC_F}(u_E+u_A) + \frac{1}{1+GC_F}\bar{n} + \frac{1}{1+GC_F}\bar{v}.$$

The stable closed-loop system and the sensitivity function are

$$R(z) := \frac{G(z)}{1+G(z)C_F(z)}, \quad S(z) := \frac{1}{1+G(z)C_F(z)}. \tag{1}$$

By defining

$$n := S(z)\bar{n}, \tag{2a}$$
$$v_b := S(z)\bar{v}, \tag{2b}$$

we have

$$e = R(u_E+u_A) + n + v_b, \tag{3}$$

which is shown in Fig.2 with u_E an exogenous excitation signal that needs to be carefully designed when R is unknown.

Our design is based on an abstract LTI dynamics from the adaptive control (u_A) injection point to the *error* signal (e) which is denoted by $R(z)$ as shown in Fig.2. The only requirement for $R(z)$ is that it should be stable which is always true in this example for that C_F stabilizes G. It should be emphasized that $R(z)$ can be non-minimum phase.

B. Vibration at Input

For the case that the disturbance contaminates the plant input as shown in Fig.3, we have

$$e = \frac{G}{1+GC_F}(u_E+u_A) + \frac{1}{1+GC_F}\bar{n} + \frac{G}{1+GC_F}\bar{v}.$$

By using the notations in (1) and (2a), and defining

$$v_b := R(z)\bar{v}, \tag{4}$$

we have the same final equation shown in (3). Since v_b in (2b) and (4) are both unknown, and we don't use any information from v_b, it doesn't matter whether the vibration is injected at the input side or the output side for that the feedforward controller design procedure is exactly the same. Without loss of generality, we will discuss the feedforward controller design for the general case shown in Fig.2.

C. Design Objective

Our objective is to synthesize an adaptive controller that only uses the scalar–valued error signal $e(k)$ and $a(k)$ to generate a feedforward control $u_A(k)$ such that it minimizes the effect of $v_b(k)$ on the error signal $e(k)$. We call it a feedforward controller because when the system dynamics and disturbance profile are time–invariant, C_A will not depend on the error signal $e(k)$ once the control law is learned.

The transfer function $R(z)$ in time–domain is described by

$$R(q^{-1}) := \frac{B(q^{-1})}{A(q^{-1})},$$

where q^{-1} denotes the one–step delay operator. The polynomials $A(q^{-1})$ and $B(q^{-1})$ are

$$A(q^{-1}) := 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_A} q^{-n_A},$$
$$B(q^{-1}) := b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_B} q^{-n_B},$$

and their coefficients are either known or unknown. Notice that here we assume that the system $R(z)$ has one step time delay, however, for system with no or multi-step delay, the algorithms developed in this paper can be easily adjusted.

Throughout this paper we assume that n can be whitened through $A(z)$ which implies that a white noise $w(k)$ can be defined such that

$$w(k) = A(q^{-1})n(k). \tag{5}$$

The vibration v_b in the whole paper is modeled as

$$v_b(k) = \frac{D(q^{-1})}{A(q^{-1})}a(k), \tag{6}$$

indicating that the vibration shows up in the final error can be modeled by passing the acceleration signal through the unknown filter $\frac{D(q^{-1})}{A(q^{-1})}$. Therefore from (3), (5) and (6), we have the following representation

$$A(q^{-1})e(k) = B(q^{-1})u(k) + D(q^{-1})a(k) + w(k). \tag{7}$$

Since $a(k)$ is available, our goal is to find a controller $H(q^{-1})$ and generate the control input by passing $a(k)$ through an finite impulse response (FIR) filter or equivalently

$$u(k) = H(q^{-1})a(k) = \phi_a^T(k)\theta_H, \tag{8}$$

such that the output $e(k)$ is minimized, where the regressor $\phi_a(k)$ and the controller parameter vector θ_H are

$$\begin{aligned} \phi_a^T(k) &:= [a(k), a(k-1), \cdots, a(k-N)], \\ \theta_H^T &:= [h_0, h_1, \cdots, h_N]. \end{aligned} \tag{9}$$

III. WIENER SOLUTION

Here in this paper, we only consider the case when the system function $R(z)$ is known, therefore, $u_E(k) = 0$ and $u(k) = u_A(k)$. The control objective consists in finding the optimal FIR filter $H^\star$ that minimizes the following cost function,

$$\begin{aligned} H^\star &= \arg\left\{\min_{H\in\mathcal{R}^{n_H+1}} \quad E\{e^2\}\right\} \\ &= \arg\left\{\min_{H\in\mathcal{R}^{n_H+1}} \left\|\frac{D(z)+B(z)H(z)}{A(z)}\right\|_2\right\}, \end{aligned} \tag{10}$$

where

$$E\{e^2\} = E\{e^2(k)\} = \lim_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K} e^2(k),$$

is the variance of the output signal and the symbol $\|\cdot\|_2$ denotes the H_2 norm of its inside elements.

The optimization problem in (10) can be solved by the classical Wiener filtering theory when the plant and vibration models are known. Define the variable $x(k)$ which satisfies

$$A(q^{-1})x(k) = -B(q^{-1})a(k), \tag{11}$$

and from (6) and (7) we will have

$$\begin{aligned} e(k) &= v_b(k) - H(q^{-1})x(k) \\ &= v_b(k) - \phi_x^T(k)\theta_H, \end{aligned} \tag{12}$$

where $\phi_x^T(k) := [x(k), x(k-1), \cdots, x(k-N)]$. By applying the orthogonality principle,

$$E\{\phi_x(k)e(k)\} = 0,$$

the optimal FIR parameter vector $\theta_H^\star$ is the solution of the following equation,

$$E\{\phi_x(k)v_b(k)\} - E\{\phi_x(k)\phi_x^T(k)\}\theta_H^\star = 0,$$

which is

$$\theta_H^\star = \frac{E\{\phi_x(k)v_b(k)\}}{E\{\phi_x(k)\phi_x^T(k)\}}. \tag{13}$$

In application, $\theta_H^\star$ can be estimated using least mean squares (LMS) or recursive least-squares (RLS) algorithms.

The basic algorithm to find the FIR parameter adaptively is the Fx-LMS. The filtered-x recursive least squares (Fx-RLS) can also be used when the controller order is low or the digital signal processor (DSP) is fast enough to handle high computational complexity. As is known, Fx-LMS results in low convergence rate when the input signals to the tap weights are highly correlated while Fx-RLS is too computational intensive. There are methods known as orthogonalization transform that slightly increase the computation but result in faster convergence rate than LMS. Here in this paper, two orthogonalization transform algorithms are considered to improve the convergence rate of the adaptive vibration rejection processing.

A. ORTHOGONALIZED-X BASED ALGORITHM

From (8), the feedforward control is updated using

$$u(k) = \phi_a^T(k)\hat{\theta}_H(k-1). \tag{14}$$

We know that the orthogonalization is nothing more than a linear transform. In order to simplify the procedure of developing lattice predictor based algorithms, for the signal $x(k)$ computed from (11), there exists a non-singular matrix L_x, such that

$$\boldsymbol{b}_x(k) = L_x\phi_x(k), \tag{15}$$

with the orthogonal property

$$E\{b_{x,i}(k)b_{x,j}(k)\} = 0, \text{ if } i \neq j, \text{ for } i, j = 0, 1, \cdots, N, \tag{16}$$

where $\boldsymbol{b}_x^T(k) = [b_{x,0}(k), b_{x,1}(k), \cdots, b_{x,N}(k)]$.

Parameters $\hat{\theta}_x(k)$ related to $\boldsymbol{b}_x(k)$ can be updated using the step-normalized LMS algorithm

$$\begin{aligned} \sigma_{x,i}^2(k) &= \beta\sigma_{x,i}^2(k-1) + (1-\beta)b_{x,i}^2(k), \\ \hat{\theta}_{x,i}(k) &= \hat{\theta}_{x,i}(k-1) + \frac{2\mu_0}{\sigma_{x,i}^2(k)}b_{x,i}(k)e(k), \end{aligned} \tag{17}$$

where β is the forgetting factor. (17) is equivalent to using different step-size parameters at different taps. Especially, each step-size parameter is chosen proportional to the inverse of its associated input signal power. Observing this, (17) is referred to be a step-normalized LMS recursion to prevent confusion between normalization applied to TDAFs and the standard NLMS algorithm [11]. Parameters before and after the transform has the following relationship,

$$\hat{\theta}_H(k) = L_x^T\hat{\theta}_x(k). \tag{18}$$

The algorithm developed here will be referred as Orth.-X algorithm which means this algorithm is based on the orthogonalization of the filtered signal $x(k)$. A brief description of the Orth.-X algorithm is illustrated in **Algorithm.1**.

For an adaptation like (17) where each element in $\hat{\theta}_x(k)$ is estimated individually with a step-normalized LMS, the convergence rate of the algorithm can be justified by the eigenvalue spread of the following normalized matrix

$$R_b^{\mathrm{n}} = D^{-\frac{1}{2}}R_bD^{-\frac{1}{2}}, \tag{19}$$

where $R_b = E\left\{\boldsymbol{b}_x(k)\boldsymbol{b}_x^T(k)\right\}$,. and D is formed by the diagonal elements of R_b i.e., $D = \textbf{diag}\{R_b\}$. The eigenvalue spread of a matrix M is noting as

$$\rho(M) = \frac{\lambda_{max}(M)}{\lambda_{min}(M)}, \tag{20}$$

Algorithm 1 Orthogonalize $x(k)$

procedure ORTH_X(N)
  Read $a(k)$ and update $\phi_a(k)$ ▷ (9)
  Compute $u(k)$ and inject it to the system ▷ (14)
  Filter $a(k)$ to obtain $x(k)$ ▷ (11)
  Orthogonalized $x(k)$ to obtain $\boldsymbol{b}_x(k)$
  Update $\hat{\theta}_x(k)$ ▷ (17)
  Compute $\hat{\theta}_H(k)$ ▷ (18)
end procedure

Algorithm 2 Orthogonalize $a(k)$

procedure ORTH_A(N)
  Read $a(k)$
  Orthogonalized $a(k)$ to obtain $\boldsymbol{b}_a(k)$
  Compute $u(k)$ and inject it to the system ▷ (26)
  Filter $b_a(k)$ to obtain $r_a(k)$ ▷ (23)
  Update $\hat{\theta}_r(k)$ ▷ (25)
end procedure

with $\lambda_{max}(M)$ and $\lambda_{min}(M)$ the maximum and minimum eigenvalue of the matrix M, respectively.

The advantage of Orth.-X algorithm is that elements in $\boldsymbol{b}_x(k)$ are uncorrelated. However, we need to find $\hat{\theta}_H(k)$ from $\hat{\theta}_x(k)$ in order to compute the control input $u(k)$ in (14).

B. ORTHOGONALIZED-A BASED ALGORITHM

For the signal $a(k)$, there exists a nonsingular matrix L_a, such that

$$\boldsymbol{b}_a(k) = L_a \phi_a(k), \tag{21}$$

with the property

$$E\{b_{a,i}(k) b_{a,j}(k)\} = 0, \text{ if } i \neq j, \text{ with } i, j = 0, 1, \cdots, N, \tag{22}$$

where $\boldsymbol{b}_a^T(k) = [b_{a,0}(k), b_{a,1}(k), \cdots, b_{a,N}(k)]$. The regressor we used for updating the weights will be $\boldsymbol{r}_a(k)$ with $\boldsymbol{r}_a^T(k) = [r_{a,0}(k), r_{a,1}(k), \cdots, r_{a,N}(k)]$, where

$$A(q^{-1}) r_{a,i}(k) = B(q^{-1}) b_{a,i}(k), \text{ for } i = 0, 1, \cdots, N. \tag{23}$$

However, usually the orthogonality for elements in $\boldsymbol{r}_a(k)$ is not being preserved, i.e., usually

$$E\{r_{a,i}(k) r_{a,j}(k)\} \neq 0, \text{ for } i, j = 0, 1, \cdots, N. \tag{24}$$

Parameters $\hat{\theta}_r(k)$ related to $\boldsymbol{r}_a(k)$ can be updated using the above mentioned step-normalized LMS,

$$\begin{aligned} \sigma_{r,i}^2(k) &= \beta \sigma_{r,i}^2(k-1) + (1-\beta) r_{a,i}^2(k), \\ \hat{\theta}_{r,i}(k) &= \hat{\theta}_{r,i}(k-1) + \frac{2\mu_0}{\sigma_{r,i}^2(k)} r_{a,i}(k) e(k). \end{aligned} \tag{25}$$

The control input can be directly generated by utilizing $\boldsymbol{b}_a(k)$ and $\hat{\theta}_r(k)$ as

$$u(k) = \boldsymbol{b}_a^T(k) \hat{\theta}_r(k-1). \tag{26}$$

It is easy to show that (14) and (26) are equivalent.

The algorithm based on the orthogonalization of signal $a(k)$ is described as **Algorithm.2**.

The advantage of Orth.-A algorithm here is we don't need to find $\hat{\theta}_H(k)$ from $\hat{\theta}_r(k)$ when we compute the feedforward control input $u(k)$ in (26). However, we need to filter each element in $\boldsymbol{b}_a(k)$ in order to obtain $\boldsymbol{r}_a(k)$ as shown in (23).

Remark 1
*For **Algorithm.1** and **Algorithm.2**, we expect that the eigenvalue spreads of $E\{\boldsymbol{r}_a(k)\boldsymbol{r}_a^T(k)\}$ and $E\{\boldsymbol{b}_x(k)\boldsymbol{b}_x^T(k)\}$ are much smaller than that of $E\{\phi_x(k)\phi_x^T(k)\}$ when elements of $\phi_x(k)$ are highly correlated with one another. Therefore, fast convergence rate can be achieved. In the following section, we will show that for **Algorithm.1**, faster convergence rate can be ensured, however, no faster convergence rate for **Algorithm.2** can be ensured.*

IV. ADAPTIVE LATTICE PREDICTOR

There are various orthogonal transforms like DFT, DCT, DST and Lattice Predictor. Here we use adaptive lattice predictor to perform orthogonalization.

Consider a lattice predictor of order $N+1$. The lattice predictor offers a highly efficient structure for generating the sequence of forward prediction errors $f_i(k)$ and the corresponding sequence of backward prediction errors $b_i(k)$ simultaneously with the order-update recursions.

The backward prediction error vector $\boldsymbol{b}(k)$ is then obtained

$$\boldsymbol{b}^T(k) = [b_0(k), b_1(k), \cdots, b_N(k)], \tag{27}$$

with

$$E\{b_i(k) b_j(k)\} = 0, \text{ if } i \neq j, \text{ with } i, j = 0, 1, \cdots, N, \tag{28}$$

For analysis, we have

$$\boldsymbol{b}(k) = L \phi_s(k), \tag{29}$$

where $\phi_s^T(k) := [s(k), s(k-1), \cdots, s(k-N)]$, $s(k)$ is the signal we want to orthogonalize and L is a lower triangle matrix with the following form

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -a_{1,1} & 1 & 0 & \cdots & 0 & 0 \\ -a_{2,2} & -a_{2,1} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_{N,N} & -a_{N-1,N-1} & -a_{N-2,N-2} & \cdots & -a_{N,1} & 1 \end{bmatrix},$$

with $a_{n,i}$, for $n = 1, 2, \cdots, N$, and $i = 1, 2, \cdots, n$ the optimal coefficients that minimize the forward prediction error $f_n(k)$.

In the following part, we will explore the stochastic of $\boldsymbol{r}(\boldsymbol{k})$ obtained by passing backward prediction error $\boldsymbol{b}(k)$ through a stable systems $R(z)$. We first introduce a lemma from reference [12].

Lemma 1
For a LTI system with input sequence $\{x(n)\}$, system function $H(z)$ and output $\{y(n)\}$, then

$$\Phi_{x,y}(z) = H^*(\frac{1}{z^*}) \Phi_{x,x}(z),$$
$$\Phi_{y,x}(z) = H(z) \Phi_{x,x}(z).$$

Moreover, if $\{d(n)\}$ *is a third process,*

$$\Phi_{y,d}(z) = H(z)\Phi_{x,d}(z),$$
$$\Phi_{d,y}(z) = H^*(\frac{1}{z^*})\Phi_{x,d}(z).$$

From **Lemma**.1, the following theorem can be obtained. The theorem gives the way to compute the cross-correlation of two signals that are generated from the same source but through different LTI systems.

Theorem 1
For a LTI system with input sequence $\{x(n)\}$*, system function* $H_1(z)$ *and output* $\{y_1(n)\}$*, and another LTI system with the same input sequence but a different system function* $H_2(z)$ *and output* $\{y_2(n)\}$*, then*

$$\Phi_{y_1,y_2}(z) = H_1(z)H_2^*(\frac{1}{z^*})\Phi_{x,x}(z),$$
$$\Phi_{y_2,y_1}(z) = H_2(z)H_1^*(\frac{1}{z^*})\Phi_{x,x}(z).$$

Moreover, if both $H_1(z)$ *and* $H_2(z)$ *are of real coefficients, then*

$$\Phi_{y_1,y_2}(z) = H_1(z)H_2(z^{-1})\Phi_{x,x}(z),$$
$$\Phi_{y_2,y_1}(z) = H_2(z)H_1(z^{-1})\Phi_{x,x}(z).$$

The proof can be completed by exploring **Lemma**.1, which is omitted here.

From (29), by defining, we have

$$\begin{aligned} &H_0(z) = 1,\\ &H_1(z) = z^{-1} - a_{1,1},\\ &H_2(z) = z^{-2} - a_{2,1}z^{-1} - a_{2,2},\\ &\quad\vdots\\ &H_N(z) = z^{-N} - a_{N,1}z^{-N+1} - \cdots - a_{N,N-1}z^{-1} - a_{N,N}. \end{aligned} \tag{30}$$

Since here $H_i(z)$ and $R(z)$ are both have real coefficients, by applying **Theorem.1**, we have

$$\Phi_{r_i,r_j}(z) = R(z)R(z^{-1})H_i(z)H_j(z^{-1})\Phi_{s,s}(z), \text{ for } i,j = 0,1,2,\cdots,N.$$

Noticing,

$$\Phi_{b_i,b_j}(z) = H_i(z)H_j(z^{-1})\Phi_{s,s}(z), \text{ for } i,j = 0,1,2,\cdots,N,$$

we have

$$\Phi_{r_i,r_j}(z) = R(z)R(z^{-1})\Phi_{b_i,b_j}(z), \text{ for } i,j = 0,1,2,\cdots,N,$$

Since

$$\phi_{b_i,b_j}(k) = 0, \text{ for } i,j = 0,1,2,\cdots,N, \text{ and } i \neq j,$$

and the $R(z)$ is arbitrarily stable causal system, we know that the autocorrelation matrix for $\boldsymbol{r}(k)$ will be

$$(R_r)_{i,j} = E\Big\{r_i(k)r_j(k)\Big\} = \phi_{r_i,r_j}(0) \neq 0,$$

for $i \neq j$ and $i,j = 0,1,2,\cdots,N$, which implies that the presence of $R(z)$ destroy the orthogonality of $\boldsymbol{b}(k)$.

V. NUMERICAL EXAMPLES AND SIMULATIONS

In this section, we give a simple vibration rejection example based on the developed scheme shown in Fig.2 and algorithms proposed in this paper.

Consider the case where $a(k)$ is a first-order autoregressive process generated by passing a white noise process $v(k)$ through the system,

$$S_D(z) = \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}}, \tag{31}$$

where α is a constant and $\alpha \in (-1,1)$. The transfer function from $a(k)$ to $x(k)$ is assumed to be

$$R(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1}}, \tag{32}$$

where a_1 in a constant and $a_1 \in (-1,1)$. Supposing that the vibration $v_b(k)$ is generated by passing the same white noise $v(k)$ through system

$$P(z) = \frac{\sqrt{1-a_1^2}}{1-a_1 z^{-1}}. \tag{33}$$

The autocorrelation of $a(k)$ and $x(k)$ in Z-domain are respectively

$$\begin{aligned} &\Phi_{a,a}(z) = S_D(z)S_D(z^{-1})\Phi_{v,v}(z),\\ &\Phi_{x,x}(z) = R(z)R(z^{-1})\Phi_{a,a}(z). \end{aligned} \tag{34}$$

Since $\Phi_{v,v}(z) = 1$, we have

$$\Phi_{a,a}(z) = \frac{1-\alpha^2}{(1-\alpha z^{-1})(1-\alpha z)},$$
$$\Phi_{x,x}(z) = \frac{(1-\alpha^2)(b_1 + b_2 z^{-1})(b_1 + b_2 z)}{(1-\alpha z^{-1})(1-\alpha z)(1-a_1 z^{-1})(1-a_1 z)}.$$

By doing the inverse Z-transform, the autocorrelation of $a(k)$ and $x(k)$ in time domain can be obtained,

$$\begin{aligned} &\phi_{a,a}(k) = \alpha^{|k|}, \text{ for } k = \cdots,-2,-1,0,1,2,\cdots,\\ &\phi_{x,x}(k) = c_1\alpha^{|k|} + c_2 a_1^{|k|}, \text{ for } k = \cdots,-2,-1,0,1,2,\cdots, \end{aligned} \tag{35}$$

where c_1 and c_2 are constants with $c_1 = \frac{(b_1+b_2\alpha^{-1})(b_1+b_2\alpha)}{(1-a_1\alpha^{-1})(1-a_1\alpha)}$ and $c_2 = \frac{(1-\alpha^2)(b_1+b_2a_1^{-1})(b_1+b_2a_1)}{(1-\alpha a_1^{-1})(1-\alpha a_1)(1-a_1^2)}$.

Supposing that the FIR filter is of order 3,when $\alpha = 0.9$, $a_1 = 0.5$, $b_1 = 1$, $b_2 = 1.2$, we have $c_1 = \frac{1092}{55}$, $c_2 = -\frac{2584}{825}$. Then the eigenvalue spread of R_a is $\rho(R_a) = 39.5331$ and the eigenvalue spread of R_x is $\rho(R_x) = 607.1014$. Therefore, when use $a(k)$ or $x(k)$ as tap-inputs, very slow LMS adaptation can be predicted.

For signal $a(k)$, the autocorrelation $\phi_{a,a}(0) = 1$, $\phi_{a,a}(1) = \alpha = 0.9$, $\phi_{a,a}(2) = \alpha^2 = 0.81$. By applying the Levinson-Durbin algorithm [10], we have $a_{1,1} = a_{2,1} = 0.9$, and $a_{2,2} = 0$, therefore, the autocorrelation matrix of $\boldsymbol{b}_a(k)$, $R_{L_a} = \mathbf{diag}\{[1, 0.19, 0.19]\}$. For signal $x(k)$, similarly we have the autocorrelation matrix of $\boldsymbol{b}_x(k)$, $R_{L_x} = \mathbf{diag}\{[16.7224, 0.8283, 0.4008]\}$. As a result, by applying (19), both $R^{\mathrm{n}}_{L_x}$ and $R^{\mathrm{n}}_{L_a}$ are with the eigenvalue spread $\rho(R^{\mathrm{n}}_{L_a})$ and $\rho(R^{\mathrm{n}}_{L_x})$ equal to one. Therefore, when $\boldsymbol{b}_a(k)$ or $\boldsymbol{b}_x(k)$

being used as the regressors with step-normalized LMS update, fast convergence rate can be predicted.

In the following, we will compute the autocorrelation matrix R_r for the input vector $\boldsymbol{r}(k)$ generated by passing $\boldsymbol{b}_a(k)$ through $R(z)$. First of all, we have the transfer function defined in (30) as

$$\begin{aligned} H_0(z) &= 1, \\ H_1(z) &= z^{-1} - a_{1,1}, \\ H_2(z) &= z^{-2} - a_{2,1}z^{-1} - a_{2,2}. \end{aligned}$$

Note that here $a_{1,1}$, $a_{2,1}$, $a_{2,2}$ are related to $a(k)$, which here are $a_{1,1} = a_{2,1} = 0.9$, and $a_{2,2} = 0$, respectively. According to **Theorem.1**, and applying the inverse Z-transform, the autocorrelation and cross-correlation in time domain can be obtained. When $\alpha = 0.9$, $a_1 = 0.5$, $a_{1,1} = 0.9$, $a_{2,1} = 0.9$ and $a_{2,2} = 0$, we have

$$R_r = \begin{bmatrix} 16.7224 & 1.2528 & 0.6264 \\ 1.2528 & 0.9221 & 0.6891 \\ 0.6264 & 0.6891 & 0.9221 \end{bmatrix},$$

and its diagonal normalized matrix

$$R_r^{\mathrm{n}} = \begin{bmatrix} 1.0000 & 0.3190 & 0.1595 \\ 0.3190 & 1.0000 & 0.7473 \\ 0.1595 & 0.7473 & 1.0000 \end{bmatrix},$$

which is obviously not diagonal with the eigenvalue spread of $\rho(R_r) = 8.0183 < \rho(R_x)$, indicating a faster convergence when $\boldsymbol{r}(k)$ is used as the regressor than that of $\phi_x(k)$. However, $\rho(R_r)$ cannot be forced to be as close as possible to one. In fact, $\rho(R_r)$ is determined by the system dynamics $R(z)$. In other words, the uncorrelation of elements in $\boldsymbol{b}_a(k)$ cannot be preserved when passing uncorrelated signal to arbitrarily LTI system. For instance, here in this example, $a_1 = 0.95$ results in $\rho(R_r) = 118.6989$ which is too large while $a_1 = 0.1$ gives $\rho(R_r) = 3.6476$ which is good for step-normalized update.

The LMS algorithm, Orth.-A based and Orth.-X based algorithms are all implemented using MATLAB simulation. In this example, the optimal controller is obtained using (13) as

$$H^\star(q^{-1}) = 0.6595 - 0.8709q^{-1} + 0.2538q^{-2}.$$

The step-size parameter μ_0 is 0.0005 for all three configurations. The coefficients revolution of $H(q^{-1})$ are shown in Fig.4 where the black solid line shows the optimal coefficients, the blue dotted line gives coefficients revolution for Fx-LMS,the red dashed dot line for **Algorithm.1** and magenta dashed lien for **Algorithm.2**. Among these three algorithms, all the coefficients converge to their optimal value finally in our simulation. Here in Fig.4, only the first 12,000 iterations are given. As predicted, the LMS algorithm is the slowest one, and the Orth.-X based algorithm is the fastest one while Orth.-A has improvement in convergence rate over LMS in our simulation configuration but not as fast as Orth.-X based algorithm.

VI. CONCLUSIONS

In this paper, vibration rejection using feedforward control architecture is presented. Two fast convergence feedforward control algorithms are developed and analyzed. The numerical examples show that Orth.-A algorithm can improve the convergence rate over Fx-LMS algorithm under certain condition, while the Orth.-X algorithm is never worse than the Fx-LMS algorithm. Simulations coincide with the numerical results. In the following, we will do simulation using real HDD models and implement the algorithms in real HDD setup to see how many revolutions are sufficient for algorithms to converge. After that, efficient direct adaptive algorithm based on lattice predictor can be developed to estimate the system $R(z)$ as well as the optimal filter $H^\star(z)$ simultaneously.

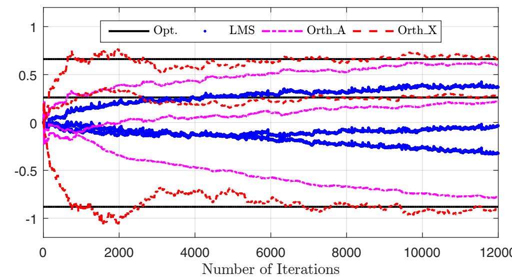

Fig. 4. Coefficients revolution of various algorithms.

REFERENCES

[1] B. Widrow and S. D. Stearns, "Adaptive signal processing," *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, vol. 1, 1985.

[2] J. C. Burgess, "Active adaptive sound control in a duct: A computer simulation," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 715–726, 1981.

[3] J. Pan and Y. Wang, "Internal model based active disturbance rejection control," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 6989–6994.

[4] J. Pan, O. Bagherieh, B. Shahsavari, and R. Horowitz, "Triple-stage track-following servo design for hard disk drives using μ-synthesis," in *ASME 2016 Dynamic System and Control Conference*. American Society of Mechanical Engineers, 2016.

[5] J. Pan, P. Shah, and R. Horowitz, "Dsp implementation of a direct adaptive feedfoward control algorithm for rejecting repeatable runout in hard disk drives," in *ASME 2016 Information Storage and Processing Systems Conference*. American Society of Mechanical Engineers, 2016.

[6] B. Shahsavari, J. Pan, and R. Horowitz, "Adaptive rejection of periodic disturbances acting on linear systems with unknown dynamics," in *55th IEEE Conference on Decision and Control*. IEEE, 2016.

[7] B. Shahsavari, E. Keikha, F. Zhang, and R. Horowitz, "Adaptive repetitive control using a modified filtered-x lms algorithm," in *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2014, pp. –.

[8] B. Shahsavari, J. Pan, and R. Horowitz, "An adaptive feedforward control approach for rejecting disturbances acting on uncertain linear systems," in *ASME 2016 Dynamic System and Control Conference*. American Society of Mechanical Engineers, 2016.

[9] K.-A. Lee, W.-S. Gan, and S. M. Kuo, *Introduction to Adaptive Filters*. John Wiley & Sons, Ltd, 2009, pp. 1–40.

[10] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*. John Wiley & Sons, 2013.

[11] ——, *Transform Domain Adaptive Filters*. John Wiley & Sons, Ltd, 2013, pp. 207–250.

[12] ——, *Discrete-Time Signals and Systems*. John Wiley & Sons, Ltd, 2013, pp. 28–47.