Advanced Engineering Forum *Online: 2011-12-22*
ISSN: 2234-991X, Vols. 2-3, pp 640-644
doi:10.4028/www.scientific.net/AEF.2-3.640

Implementation of FxLMS Algorithm Based on Level-2 S-Function and Behavior Analysis

Feng Zhang[1,a], Yinong Li[1,b] , Lei Wang[1,c]

[1]State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, 400044, China

[a]alwaysqing@126.com, [b]ynli@cqu.edu.cn, [c]wangl2000@yahoo.cn

Key words: FxLMS, level-2, S-function, adaptive filter, Active vibration control

Abstract. Active vibration control use an active force from secondary path to suppress or attenuate periodic or quasi-periodic vibration and noises, one of the popular control algorithms is filtered-x least mean square (FxLMS) adaptive method. Based on structure of FxLMS algorithm, the paper uses Level-2 S-fun to create a new FxLMS blocks in Matlab/Simulink and apply it on vibration active control system simulator. Results of computer simulation showed the custom FxLMS block's feasibility and control algorithm's efficiency. On the condition of stabilization, behavior analysis was also taken out by adjusting the interior parameters, including the number of Weights and the Step size.

Introduction

Rotating machines such as gear boxes, motors, cutting machines, aircrews etc, generate noise and vibration signals that can usually be modeled as sinusoidal signals in additive white noises. Suppressing or reducing these noise signals, especially their lower frequency portion, is of great importance in various engineering and environmental systems [1]-[3], Recently active vibration control (AVC) systems have been developed to suppress these annoying vibration sources[4]. A large number of AVC systems have been proposed in the literature, and some of them have been implemented in real-life applications [5]. Usually, FIR filters are applied, which are adapted by the filtered-X least-mean square (FxLMS) algorithm or its variants. Other techniques using recursive least squares (RLS) and Kalman filter based algorithms have also been developed for many ANC systems. Normally, a well-designed controller will be able to provide sufficient reduction to achieve the desired response under certain conditions. This paper aims the implementation of FxLMS Algorithm based on level-2 s-function and behavior analysis. Because the Level-2 S-function provides access to a more extensive set of the S-function API and supports code generation that facilitates the next experiments and practical engineering applications.

Structure of FxLMS Algorithm

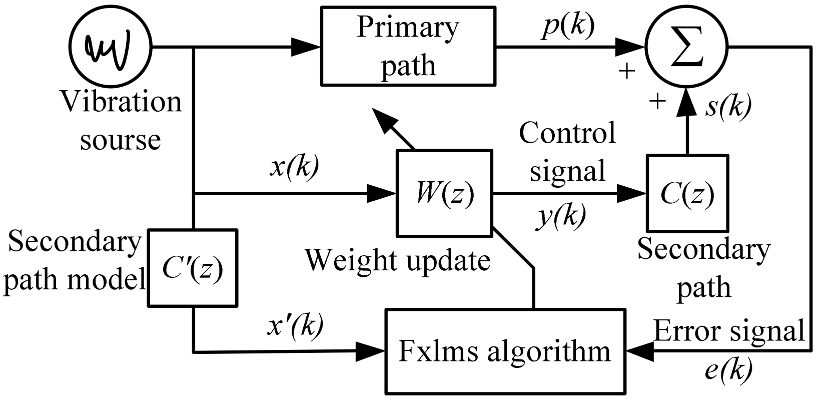


Fig. 1. Block diagram of active vibration control system using the FxLMS algorithm

Figure 1 shows the block diagram of the active vibration control system using FxLMS algorithm. The purpose of the control system can divide into two effectively sections: the export of the control signal and the weight adaptation of control filter. $p(k)$ is original vibration to be controlled and $C(z)$

represents the secondary path, which accounts for the transducer response, including the A/D and D/A converters, and the acoustical or structural propagation. $C'(z)$ is a model of the secondary-path transfer function. Error signal $e(k)$ equal to the sum of initial vibration signal $p(k)$ and anti-vibration signal created by control source $s(k)$.

$$e(k) = p(k) + s(k) \tag{1}$$

At time k, the output of N orders FIR control filter, $y(k)$, equals to the following convolution:

$$\begin{aligned} y(k) &= \sum_{i=0}^{N-1} w_i(k)x(k-i) \\ &= w^T(k)x(k) = x^T(k)w(k) \end{aligned} \tag{2}$$

As mentioned above, the signal of feedforward control obtained by FIR control filter, $y(k)$, will doesn't equal to the control source of error signal $s(k)$, for during the active control of noise and vibration, the objective of control is actual signal, including sound interference and vibration disturbance, differ from the electric unit that can use electric signal directly. Therefore, sensors are necessary to transfer between the two states, these sensors (actuators and error sensors) will have characteristic frequency response and transfer functions, that was called the secondary path's transfer function which can be modeled by FIR filters in time domain.

$$c = [c_0\ c_1 \cdots c_{m-1}]^T \tag{3}$$

The anti-vibration components $s(k)$ generated by control channel equal to the convolution of the output of LMS control filter $y(k)$ and the transfer function of secondary path:

$$s(k) = y(k) * c = \sum_{i=0}^{m-1} (k-i)c_i = y^T(k)c \tag{4}$$

and $y(k)$ is a vector of m×1 order:

$$y(k) = X^T(k)w \tag{5}$$

x (k) is the input of control filter and f(k) is the filtered reference signal.

$$f(k) = X^T(k)c \tag{6}$$

The update formula of the control FIR filter's weight is:

$$w(k+1) = w(k) - 2\mu e(k) f(k) \tag{7}$$

Formula (7) is so called FxLMS algorithm. During the actual applications, if the input signal $x(k)$,the transfer function of secondary path, c, and the error signal $e(k)$ are known, take the initial weight w to 0, then the FxLMS algorithm process could be designed by the following steps:

1. Raise the value of delay chain of FIR control filter to next period and input a new reference sample.
2. Compute a new output of FIR control filter by formula (5).
3. Obtain the anti-vibration components s (k) by formula (4).
4. Obtain the error signal e (k), which equal to the sum of the primary path and the secondary path.
5. Compute the new weight coefficient by formula (7)
6. Repeat the above steps.

Create an FxLMS block using level-2 S-functions

Matlab/Simulink provides Signal Processing Blockset for the design and simulation of signal processing systems, which contain a variety of LMS filters. However, there is no FxLMS block in the Blockset, and the FxLMS algorithm can't realize by LMS filter blocks because the difference in error signals and process structure of the weight iterate. In order to realize the simulation of FxLMS algorithm in Simulink environment, we introduce S-functions (System-functions) to create a new user-defined FxLMS block. S-function is a computer language description of a Simulink block written in MATLAB, C, C++, or Fortran. There are 2 kinds of M-file S-functions: the Level-1 M-file S-function and the Level-2 M-file S-function. The Level-1 M-file S-function is simple and quickly operates, Note that Level 1 M-file S-functions are provided only to ensure Simulink models with predecessor S-functions simulate properly. Use Level-2 M-file S-functions to develop new S-functions. The Level-2 M-file S-function application programming interface (API) allows us to create blocks that have many of the features and capabilities of Simulink built-in blocks, including:

1. FxLMS algorithm has an inherent feature of multiple input and output ports, level-2 S-functions just have the advantages of support multiple input and output ports.

2. The Level-2 M-file S-function API corresponds closely to the API for creating C MEX-file S-functions. Much of the documentation for creating C MEX-file S-functions applies also to Level-2 M-file S-functions.

3. The Level-2 M-file S-function support Dwork vectors re blocks of memory that an S-function asks the Simulink engine to allocate to each instance of the S-function in a model. If multiple instances of your S-function can occur in a model, your S-function must use DWork vectors instead of global or static memory to store instance-specific values of S-function variables. Otherwise, your S-function runs the risk of one instance overwriting data needed by another instance, causing a simulation to fail or produce incorrect results.

4. Providing a corresponding Target Language Compiler (TLC), the Level-2 M-file S-function can generate target code for a model, which can directly download to DSP or DSpace You do not need a TLC file to accelerate a model containing a Level-2 M-file S-function. The Simulink Accelerator software can run Level-2 M-file S-functions in interpreted mode.

The Level-2 M-file S-function API defines the signatures and general purposes of the callback methods that constitute a Level-2 M-file S-function. The S-function itself provides the implementations of these callback methods. The implementations in turn determine the block attributes (e.g., ports, parameters, and states) and behavior (e.g., the block outputs as a function of time and the block inputs, states, and parameters). During the course of built the FxLMS block, the following callback methods was necessary.

1. The setup Method, initializes the instance of the corresponding Level-2 M-File S-Function block, the setup method is similar to the mdlInitializeSizes callback methods implemented by C MEX S-functions. In this paper, the setup method performs such tasks including:

Initializing the number of input and output ports of the block, 3 inputs are reference signal, filtered reference signal and error signal, 2 outputs are control signal and weight coefficients. Invoke the run-time object's SetPreCompInpPortInfoToDynamic and SetPreCompOutPortInfoToDynamic methods to indicate that the input and output ports inherit their compiled properties.

Set the run-time object's NumDialogPrms property to 1 to initialize one S-function dialog parameter.

Registering S-function callback methods by passing the handles of local functions in the M-file S-function to the RegBlockMethod method of the S-Function block's run-time object

2. The CheckPrms Method, Check the validity of an S-function's parameters,

3. The PostPropagationSetup method, called DoPostPropSetup in this S-function, set up two DWork vectors, The first DWork vector stores the reference signal value, $f(k)$, which is modified at every time step in the Update method. The second DWork vector stores the weight value, W. The value of this DWork vector does not overwrite during the course of the simulation.

4. The Start Method, initialize the state vectors of the S-function, set initial value of Dwork vectors to 0.

5. The Output Method, Compute the signals that the FxLMS block emits, and export the results to output ports.

Simulation of FxLMS Algorithm

To examine the custom block of FxLMS algorithm, we apply the block built by the Level-2 S function on the active vibration control. Figure 2shows the simulation with custom FxLMS block used in simulation. In this simulation, the sinusoidal excitation depart 4 part: 90Hz,100 Hz, 110Hz with an amplitude of 1, and a white noise. Both the primary and secondary path were modeled by FIR filters, the transfer function of primary path is:

$$p(z) = 0.001 + 0.005z^{-1} - 0.005z^{-2} + 0.75z^{-3} + 0.5z^{-4} - 0.35z^{-5} - 0.4z^{-6} + 0.2z^{-7} - 0.4z^{-8} - 0.1z^{-9}$$

and the secondary path is:

$$c(z) = 0.01 + 0.01z^{-1} + 0.9z^{-2} - 001z^{-3} - 0.75z^{-5}$$

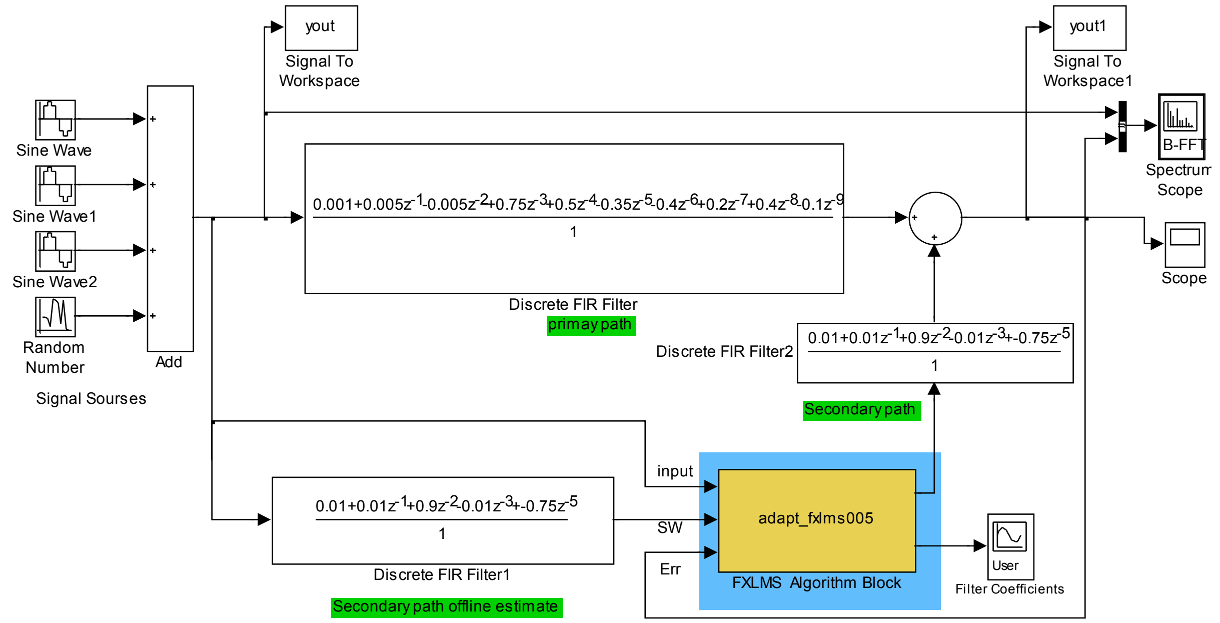


Fig. 2. Computer simulation of the active vibration control system using FxLMS algorithm

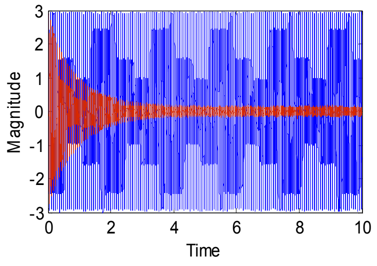


Fig. 3. Results of active control in time domain

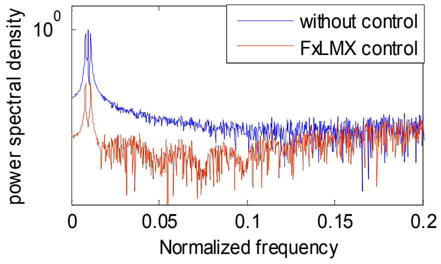


Fig. 4. Weight iteration in time domain

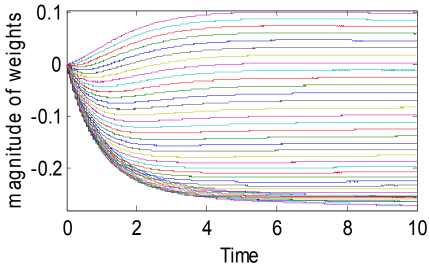
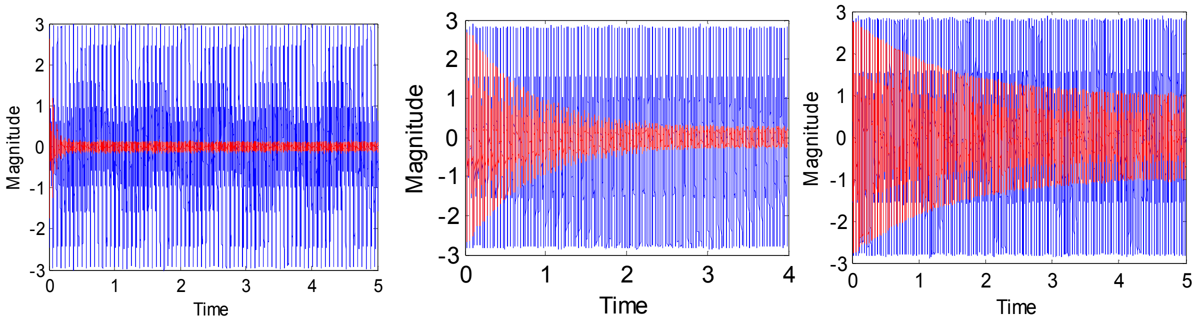


Fig. 5. Weight iteration

Figure 3 shows the active vibration control results applying the proposed FxLMS algorithm , From the figure, under ideal circumstances(Reference signal and the initial vibration signal is identical, the estimation of secondary path and the actual secondary path is identical), the initial vibration almost be attenuated. Figure 4 show the control results in frequency domain, the blue line is the vibration spectrum without control and the red line is the vibration control with the control turn on. Figure 5 shows the update of the weight coefficient; the weight will turn to be fixed when the vibration sources are a stable period signal.

Performance analyses of the algorithm blocks



a μ=0.001 n=80 b μ=0.0001 n=80 c μ=0.0001 n=10

Fig. 6. Results of modified different parameters of vibration active control

On the condition of stabilization, there are 2 main interior parameters that affect the performance of the FxLMS algorithm: the step size and the number of weights. Reference [6] has pointed out the range of step size to ensure the convergence of the algorithm. To verify the performance of the custom FxLMS block, change the step size and the number of weights, obtain results separately.

Firstly, maintain the order equal to 80 and change the step size, the results can be seen in figure 6(a) and 6(b). Secondly, keep step size to 0.0001 and change the number of weights from 80 to 10, then get the result in 6(c). Compare fig. 6 and fig. 3 we can obtained that when the Step size (μ) is increased from 0.0001 to 0.001, FxLMS algorithm converges more quickly; when the number of Weights is decrease from 80 to 10, the filter of the algorithm will not remove the noise properly and then the control effect were weakened largely.

Conclusions

1. Uses Level-2 S-functions create a new FxLMS blocks in Matlab/Simulink and apply it on vibration active control system simulator. Results of computer simulation showed the custom FxLMS block's feasibility and control algorithm's efficiency.
2. Conclusions could be drawn that if the number of weights is large, the algorithm will be slow to run, but have better accuracy, if the number of weights is decreased, the filter will not remove the vibration properly. When the Step size is increased, algorithm converges more quickly, when the step-size exceeds a max threshold to ensure stability, the algorithm will be unstable. These results will provide guidance to obtain optimum performance in practical applications.

Acknowledgement

This project is supported by National Natural Science Foundation of China (Grant No.50875270), Ministry of Education Foundation (Grant No.108108) and Chongqing Municipal Natural Science Foundation (CSTC, Grant No.2008BA6025)

References

[1] C.H.Hansen, S. D. Snyder. Active Control of Noise and Vibration, E and FN Spon, 1997

[2] Gary L. Anderson, Devendra P. Garg, Active Vibration Control in Rotorcraft Systems Using Smart Actuators. 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado, Apr. 22-25

[3] JJ Carnahan, CM Richards, A Modification to Filtered-X LMS Control for Airfoil Vibration and Flutter Suppression, Journal of Vibration and Control, Vol. 14, No. 6, (2008) 831-848

[4] Elias Bjamason, analysis of the Filtered -X LMS Algorithm, IEEE transactions on speech and audio processing, vol.3 No 6 11.1995

[5] B.Widrow, S.D.Stearns, Adaptive Signal Processing, China Machine Press, Beijing, 2008

[6] Luis Vicente, Enrique Masgrau, Novel FxLMS Convergence Condition With Deterministic Reference, IEEE Transactions on Signal Processing. 54-10(2006) 3768-3774