

DISTURBANCE REJECTION USING DECENTRALIZED SELF-TUNING ARMARKOV ADAPTIVE CONTROL WITH SIMULTANEOUS IDENTIFICATION

Ravinder Venugopal*, Harshad S. Sane†, Daniel P. Scharf‡,
Dennis S. Bernstein§ and David C. Hyland ¶

Abstract

In this paper we experimentally investigate the issues of decentralized implementation of the extended ARMARKOV adaptive control (EAAC) algorithm with simultaneous identification for disturbance rejection. The test-bed is the Multi-Hex Prototype Experiment (MHPE) which is constructed to emulate the dynamics of a large space-based optical telescope, and the application is active vibration control. The ARMARKOV adaptive algorithm requires a model of only the secondary path (control input to performance variable) transfer function which is identified on-line using the time-domain ARMARKOV/Toeplitz identification technique in the EAAC. Two decentralized EAAC controllers, each connected to three sensors and two actuators, are implemented and experimental results which show broadband disturbance rejection are presented.

Introduction

Both robust control and adaptive controllers seek to achieve system performance without excessive reliance on plant models. While robust controllers desensitize the control system to plant uncertainty, the gains of robust controllers are fixed. On the other hand, adaptive controllers adjust gains during operation in order to permit greater uncertainty levels than can be tolerated by robust control and to improve system performance during operation, which is not possible using robust controllers. Another distinction between robust and adaptive controllers is the fact that robust controllers are generally linear, while adaptive controllers are inherently nonlinear.

Since adaptive controllers are inherently nonlinear, the rigorous analysis of their convergence properties in the presence of unknown disturbances and plant dynamics is generally more difficult than the analysis of robust controllers. Nevertheless, the analysis of both direct and indirect adaptive control algorithms has reached a fairly mature stage [1, 2, 3, 4].

In this paper we consider the ARMARKOV adaptive control (AAC) algorithm developed in [4] and extended in [5]. The underlying model structure of AAC is the ARMARKOV model, which is a structurally constrained ARMA model with explicit impulse response (Markov) parameters. The results reported in [3, 4, 5, 6, 7] demonstrate the ability of the algorithm to suppress single-tone, dual-tone, and broadband disturbances without prior knowledge of the spectral characteristics of the disturbance. These results depend upon the availability of a model of only the secondary path transfer function from the control input to the error variables, represented by the Toeplitz matrix B_{zu} .

In [5] the AAC algorithm is extended by including simultaneous identification of the secondary path. To do this, the secondary path matrix B_{zu} is updated *at each time step* by means of the ARMARKOV/Toeplitz recursive identification method of [8]. Thus, the extended ARMARKOV adaptive control (EAAC) algorithm starts out with no prior knowledge of the plant dynamics *and* no measurement of the disturbance or knowledge of its spectrum.

To oversee the proper functioning of simultaneous control and identification, a supervisory controller is used to make mode-switching decisions. These decisions include 'toggling controller adaptation', 'switching control signal ON/OFF', 'resetting controller parameters to zero' and 'toggling simulta-

*dSPACE Inc., Northville, MI,

†Aerospace Engg. Dept., Univ. of Michigan, Ann Arbor

‡Aerospace Engg. Dept., Univ. of Michigan, Ann Arbor

§Aerospace Engg. Dept., Univ. of Michigan, Ann Arbor

¶Aerospace Engg. Dept., Univ. of Michigan, Ann Arbor

Copyright © 2000 The American Institute of Aeronautics and Astronautics Inc. All rights reserved

neous identification'.

The increased performance and robustness attained using the EAAC comes at the price of a significant computational burden on the real-time processor. While advances in processor speed have provided new opportunities for implementing adaptive controllers [2, 4, 9, 11], the computational complexity of the EAAC grows significantly with the number of sensors and actuators that are used to control the system. One solution to this problem is to spread the computational effort over several processors, each running a decentralized adaptive controller connected to a small set of sensors and actuators.

This paper presents the results of an experimental study of decentralized EAAC on the Multi-Hex Prototype Experiment (MHPE) at the University of Michigan. The MHPE emulates the support structure for a large space-based optical telescope. Two decentralized EAAC controllers were implemented on a dSPACE real-time multiprocessor system with four Alpha/C40 combination processors.

This study has the following specific objectives. First, it is of interest to investigate the ability of the EAA controllers to independently reject tonal and broadband disturbances on this test bed. Next we investigate the ability of each controller to identify its secondary path while the other controller is operational. Based on the results of these tests we formulate a mode switching sequence for both controllers which achieves the objective of disturbance attenuation.

Disturbance Rejection Problem

Consider the linear discrete-time two vector-input, two vector-output (TITO) system shown in Figure 1. The *disturbance* $w(k)$, the *control* $u(k)$, the *measurement* $y(k)$ and the *performance* $z(k)$ are in \mathcal{R}^{m_w} , \mathcal{R}^{m_u} , \mathcal{R}^{l_y} and \mathcal{R}^{l_z} , respectively. The system can be written in state space form as

$$x(k+1) = Ax(k) + Bu(k) + D_1w(k), \quad (1)$$

$$z(k) = E_1x(k) + E_2u(k) + E_0w(k), \quad (2)$$

$$y(k) = Cx(k) + Du(k) + D_2w(k), \quad (3)$$

or equivalently in terms of transfer matrices

$$z = G_{zw}w + G_{zu}u, \quad (4)$$

$$y = G_{yw}w + G_{yu}u. \quad (5)$$

The controller G_c generates the control signal $u(k)$

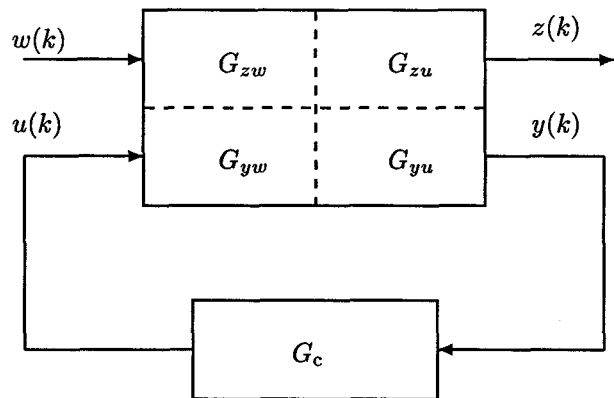


Figure 1: Standard problem with fixed-gain controller

based on the measurement $y(k)$, that is,

$$u = G_c y. \quad (6)$$

The objective of the standard problem is to determine a controller G_c that produces a control signal $u(k)$ based on the measurement $y(k)$ such that a performance measure involving $z(k)$ is minimized.

ARMAKOV/Toeplitz Model of TITO Systems

We now develop the ARMAKOV/Toeplitz [4, 8] model of (1)-(3). Defining the Markov parameters of the system by

$$H_{yu,-1} \triangleq D, \quad H_{yu,j} \triangleq CA^j B, \quad j \geq 0, \quad (7)$$

$$H_{yw,-1} \triangleq D_2, \quad H_{yw,j} \triangleq CA^j D_1, \quad j \geq 0, \quad (8)$$

$$H_{zu,-1} \triangleq E_2, \quad H_{zu,j} \triangleq E_1 A^j B, \quad j \geq 0, \quad (9)$$

$$H_{zw,-1} \triangleq 0, \quad H_{zw,j} \triangleq E_1 A^j D_1, \quad j \geq 0, \quad (10)$$

the ARMAKOV model of (1)-(3) with μ Markov parameters is given by

$$\begin{aligned} z(k) = & \sum_{j=1}^n -\alpha_j z(k - \mu - j + 1) \\ & + \sum_{j=1}^{\mu} H_{zw,j-2} w(k - j + 1) \\ & + \sum_{j=1}^n B_{zw,j} w(k - \mu - j + 1) \\ & + \sum_{j=1}^{\mu} H_{zu,j-2} u(k - j + 1) \\ & + \sum_{j=1}^n B_{zu,j} u(k - \mu - j + 1), \quad (11) \end{aligned}$$

$$\begin{aligned}
y(k) &= \sum_{j=1}^n -\alpha_j y(k - \mu - j + 1) \\
&+ \sum_{j=1}^{\mu} H_{yw,j-2} w(k - j + 1) \\
&+ \sum_{j=1}^n B_{yw,j} w(k - \mu - j + 1) \\
&+ \sum_{j=1}^{\mu} H_{yu,j-2} u(k - j + 1) \\
&+ \sum_{j=1}^n B_{yu,j} u(k - \mu - j + 1), \quad (12)
\end{aligned}$$

where $\alpha_j \in \mathcal{R}$, $B_{zw,j}, H_{zw,j} \in \mathcal{R}^{l_z \times m_w}$, $B_{zu,j}, H_{zu,j} \in \mathcal{R}^{l_z \times m_u}$, $B_{yw,j}, H_{yw,j} \in \mathcal{R}^{l_y \times m_w}$ and $B_{yu,j}, H_{yu,j} \in \mathcal{R}^{l_y \times m_u}$.

Next, define the *extended performance vector* $Z(k)$, the *extended measurement vector* $Y(k)$ and the *extended control vector* $U(k)$ by

$$Z(k) \triangleq [z(k) \ \cdots \ z(k - p + 1)]^T, \quad (13)$$

$$Y(k) \triangleq [y(k) \ \cdots \ y(k - p + 1)]^T, \quad (14)$$

$$U(k) \triangleq [u(k) \ \cdots \ u(k - p_c + 1)]^T, \quad (15)$$

where p is a positive integer and $p_c \triangleq \mu + n + p - 1$, and the *ARMARKOV regressor vectors* $\Phi_{zw}(k)$ and $\Phi_{yw}(k)$ by

$$\Phi_{zw}(k) \triangleq \begin{bmatrix} z(k - \mu) & \cdots & z(k - \mu - p - n + 2) \\ w(k) & \cdots & w(k - \mu - p - n + 2) \end{bmatrix}^T, \quad (16)$$

$$\Phi_{yw}(k) \triangleq \begin{bmatrix} y(k - \mu) & \cdots & y(k - \mu - p - n + 2) \\ w(k) & \cdots & w(k - \mu - p - n + 2) \end{bmatrix}^T. \quad (17)$$

Then (11) and (12) can be written as an AR-MARKOV/Toeplitz model in the form

$$Z(k) = W_{zw} \Phi_{zw}(k) + B_{zu} U(k), \quad (18)$$

$$Y(k) = W_{yw} \Phi_{yw}(k) + B_{yu} U(k), \quad (19)$$

where the block-Toeplitz matrices W_{zw} , B_{zu} , W_{yw} and B_{yu} contain the parameters α_j , $B_{zw,j}$, $H_{zw,j}$, $B_{zu,j}$, $H_{zu,j}$, $B_{yw,j}$, $H_{yw,j}$, $B_{yu,j}$ and $H_{yu,j}$. These matrices are as defined in [4].

Adaptive Disturbance Rejection Algorithm

In this section we review the ARMARKOV adaptive disturbance rejection feedback algorithm for the TITO system represented by (18) and (19) [4]. We use a strictly proper controller in ARMARKOV form

of order n_c with μ_c Markov parameters, so that the control $u(k)$ is given by

$$\begin{aligned}
u(k) &= \sum_{j=1}^{n_c} -\alpha_{c,j}(k) u(k - \mu_c - j + 1) \\
&+ \sum_{j=1}^{\mu_c - 1} H_{c,j-1}(k) y(k - j + 1) \\
&+ \sum_{j=1}^{n_c} B_{c,j}(k) y(k - \mu_c - j + 1), \quad (20)
\end{aligned}$$

where $H_{c,j} \in \mathcal{R}^{m_u \times l_y}$ are the Markov parameters of the controller. Next, define the *controller parameter block vector*

$$\theta(k) \triangleq \begin{bmatrix} -\alpha_{c,1}(k) I_{m_u} & \cdots & -\alpha_{c,n_c}(k) I_{m_u} \\ H_{c,0}(k) & \cdots & H_{c,\mu_c-2}(k) & B_{c,1}(k) & \cdots & B_{c,n_c}(k) \end{bmatrix}.$$

Now from (13) and (20) it follows that $U(k)$ is given by

$$U(k) = \sum_{i=1}^{p_c} L_i \theta(k - i + 1) R_i \Phi_{uy}(k), \quad (21)$$

and the control input to the system $u(k)$ at the instant k is given by

$$u(k) = \theta(k) R_1 \Phi_{uy}(k), \quad (22)$$

with

$$\Phi_{uy}(k) \triangleq \begin{bmatrix} u(k - \mu_c) & \cdots & u(k - \mu_c - n_c - p_c + 2) \\ y(k - 1) & \cdots & y(k - \mu_c - n_c - p_c + 2) \end{bmatrix}^T,$$

and where L_i and R_i are constraint matrices that maintain the block-Toeplitz structure of the control weight matrix in (21) [4]. Thus, from (18) and (21) we obtain

$$\begin{aligned}
Z(k) &= W_{zw} \Phi_{zw}(k) \\
&+ B_{zu} \sum_{i=1}^{p_c} L_i \theta(k - i + 1) R_i \Phi_{uy}(k). \quad (23)
\end{aligned}$$

Next, we define a cost function that evaluates the performance of the current value of $\theta(k)$ based upon the behavior of the system during the previous p_c steps. Therefore, we define the *estimated performance* $\hat{Z}(k)$ by

$$\hat{Z}(k) \triangleq W_{zw} \Phi_{zw}(k) + B_{zu} \sum_{i=1}^{p_c} L_i \theta(k) R_i \Phi_{uy}(k), \quad (24)$$

which has the same form as (23) but with $\theta(k - i + 1)$ replaced by the current parameter block vector $\theta(k)$.

Using (24) we define the *estimated performance cost function*

$$J(k) = \frac{1}{2} \hat{Z}^T(k) \hat{Z}(k). \quad (25)$$

The gradient of $J(k)$ with respect to $\theta(k)$ is given by

$$\frac{\partial J(k)}{\partial \theta(k)} = \sum_{i=1}^{p_c} L_i^T B_{zu}^T \hat{Z}(k) \Phi_{uy}^T(k) R_i^T. \quad (26)$$

Note that $\hat{Z}(k)$ cannot be evaluated using (24) since $w(k)$ is not available which implies that $\Phi_{zu}(k)$ is unknown. However, it follows from (18) and (24) that

$$\hat{Z}(k) = Z(k) - B_{zu} \left(U(k) - \sum_{i=1}^{p_c} L_i \theta(k) R_i \Phi_{uy}(k) \right),$$

which can be used to evaluate (26).

The gradient (26) is used in the update law

$$\theta(k+1) = \theta(k) - \eta(k) \frac{\partial J(k)}{\partial \theta(k)}, \quad (27)$$

where $\eta(k)$ is the *adaptive step size* given by

$$\eta(k) = \frac{1}{p_c \|B_{zu}\|_F^2 \|\Phi_{uy}(k)\|_2^2}. \quad (28)$$

It is shown in [4] that the update law (27) with the step size (28) brings $\theta(k)$ closer to the minimizer of $J(k)$ with each time step. Note that for implementing the algorithm in practice (26, 27, 28), we only need to know the secondary feedback matrix B_{zu} apart from the measurements z and y .

Extended AAC Algorithm

In this section we discuss the self-tuning AR-MARKOV/Toeplitz controller along with simultaneous identification. The secondary path matrix B_{zu} can be obtained on-line using the time domain identification technique discussed in [8]. In order to identify B_{zu} in the presence of the disturbance $w(k)$, an uncorrelated signal u_{ID} is added to the control signal. The signal u_{ID} is small enough not to deteriorate the performance beyond acceptable limits. An estimate $\hat{W}_{zu}(k)$ can be obtained at every time instant k using the identification method of [8] with $u(k)$ replaced by $u_{ID}(k)$. An estimate of B_{zu} , namely $\hat{B}_{zu}(k)$ can thus be extracted from $\hat{W}_{zu}(k)$ and passed on to the AAC algorithm for $\theta(k)$ gradient update. Hence for practical implementation,

B_{zu} in equations (26 - 28) is replaced by the current estimate $\hat{B}_{zu}(k)$.

Figure 2 shows the schematic of the method employed for on-line identification. A supervisory controller oversees the operation of simultaneous identification and control by making higher level decisions such as switching ON/OFF control signal u , toggling controller adaptation, resetting controller parameter vector $\theta(k)$ to zero and switching ON/OFF the identification process. The additional signal u_{ID} is turned OFF when the identification process is OFF. The decisions of the supervisory controller are based on a measure of performance involving the RMS value of z data window. The supervisor has 'binary' states Z-grows, Z-reduces, Z-low which are updated at the end of the current time window by comparing the performance during the current time window to the performance during the previous data window. A well-defined set of rules then update the control variables Cont, Adap, Cont-reset, ID to their respective ON/OFF values depending on the states and previous values of control variables.

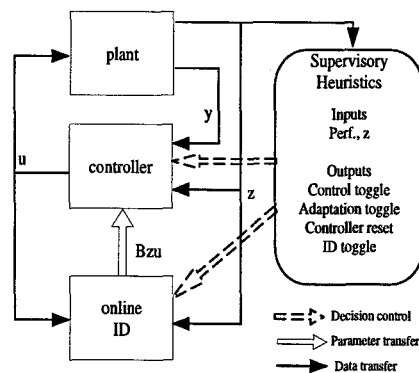


Figure 2: Simultaneous identification and control with the supervisory controller

MHPE Test-Bed

The decentralized EAAC algorithm was tested on the MHPE. The MHPE (Figure 3) emulates the support structure for a large optical telescope. Seven hexagonal, graphite-epoxy box trusses form the primary reflector support structure with a secondary tower extending above. Graphite-epoxy was used since it is a space-qualified material, giving the MHPE an even greater similarity to real flight systems. The structure is modally dense and lightly damped.

The MHPE is connected to a shaker base plate via six aluminum struts with in-line Linear

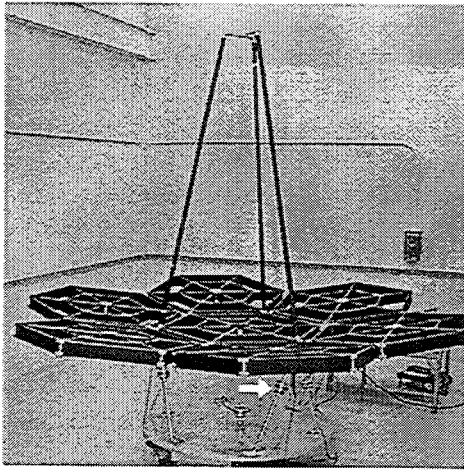


Figure 3: MHPE Test Bed

Precision Actuators (LPACTs) (arrow in Figure 3) mounted on them. The LPACTs are bearingless, linear voice coil actuators and are shown in Figure 4. An LPACT at the center of the base plate serves as a disturbance actuator. Hybrid accelerometers (which are a combination of servo and piezoelectric devices) mounted on each strut serve as performance sensors. Each accelerometer is located close to the LPACT on the strut, to obtain approximate collocation of performance sensor and control actuator. An accelerometer mounted on the disturbance actuator serves as the measurement sensor. This sensor-actuator arrangement is chosen to maximize achievable broadband performance [10].

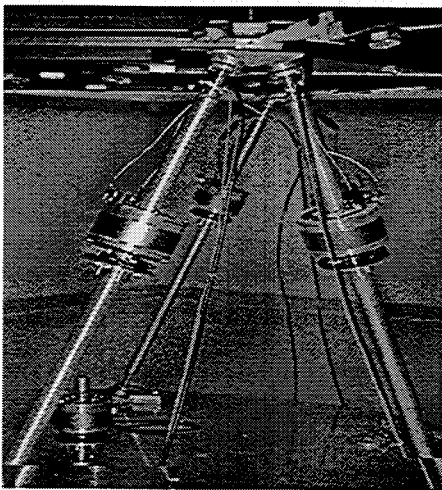


Figure 4: LPACT

Of the six control LPACTs and performance accelerometers, two LPACTs and accelerometers were used for each of the two decentralized controllers. Thus, four control ($u(k)$) actuators and four performance ($z(k)$) sensors were used in total. In addition, an accelerometer mounted near the disturbance LPACT was used by both controllers as a measurement ($y(k)$) sensor. For each decentralized adaptive controller, two dSPACE DS1004/DS1003 Alpha/C40 combination processors were used; one for implementing the adaptive control algorithm, and the other for implementing the identification and supervisory algorithms. The four Alpha/C40 pairs were mounted in a 20 slot expansion box along with a DS2003 32 channel A/D board and a DS2103 32 channel D/A board. The controllers were built as SIMULINK models with the algorithms coded as C S-functions. dSPACE Real-Time Implementation software for Multi-Processors (RTI-MP) was used to implement the decentralized controllers from the SIMULINK level, thus eliminating the need to write additional software for inter-processor communication. Figure 5 shows a schematic of the decentralized EAAC implementation structure.

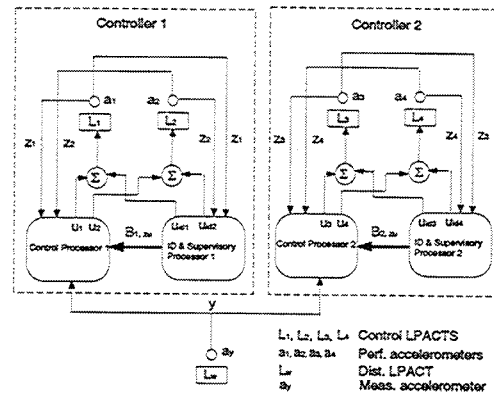


Figure 5: Decentralized EAAC Implementation Structure

Results

For the first experimental test, each decentralized controller was run separately with two performance sensors and two actuators. The disturbance signal was a single tone at 95 Hz. The objective was to ensure that each EAAC controller worked independently. An average attenuation level of 42.3 dB over the four performance sensors was observed.

For the second test, the two decentralized controllers were run simultaneously. Both con-

| Step | Action | Step | Action |
|------|------------------|------|---------------------|
| 1 | Reset C1,C2 | 5 | Freeze C1, Adapt C2 |
| 2 | ID C1 | 6 | Freeze C2, ID C1 |
| 3 | Adapt C1 | 7 | Freeze C2, Adapt C1 |
| 4 | Freeze C1, ID C2 | 8 | Adapt C1, C2 |
| | | 9 | Freeze C1, C2 |

Table 1: Switch State Sequence

trollers had their initial control parameters set to zero, and secondary path model parameters were chosen randomly. Bandlimited (0-250 Hz) white noise was used to drive the disturbance actuator and the controllers were started with the disturbance on. The following limitations on the identification process were observed:

- Poor models were obtained when both controllers attempted to identify their respective secondary paths simultaneously by driving their actuators with white noise. This is because the injected identification signals had to be of sufficient amplitude to be “heard” over the disturbance, and thus, the signals generated by Controller 1 corrupted the output of the sensors for Controller 2 and vice versa. Thus, identification had to be performed on only one controller at a time.
- Poor identified models were obtained when one controller performed identification while the other was adapting. This is because the adaptation continuously changed the closed-loop dynamics of the system in response to the identification signal introduced by the other controller. Thus, identification for one controller had to be performed with the other controller’s parameters frozen (adaptation off).

Taking these limitations into account, the switching sequence shown in Table 1 for the two controllers was implemented manually with the supervisors that control the switching modes of each of the two controllers disabled. C1 and C2 denote Controller 1 and Controller 2 respectively. Step 1 resets both controllers. Steps 4 through 7 were repeated 3 times and then both controllers were allowed to adapt simultaneously in Step 8. Finally, the parameters of both controllers were frozen in Step 9 and data for closed-loop analysis were captured. Figures 6 - 9 show that significant broadband attenuation was obtained on all four sensors.

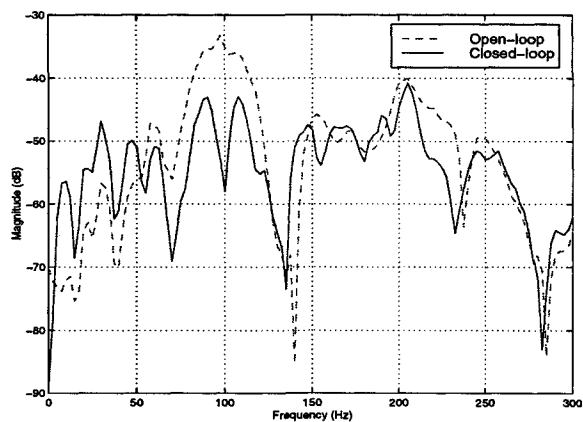


Figure 6: Open-loop and closed-loop performance (LPACT 1)

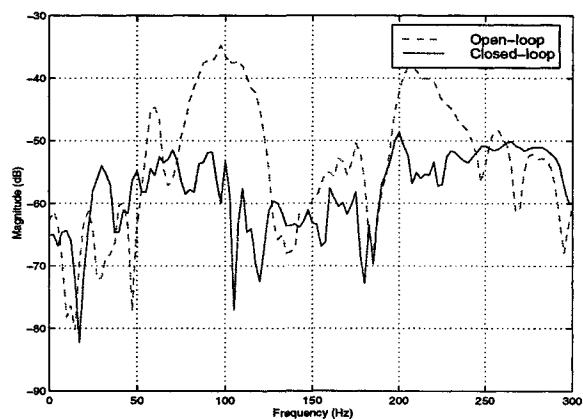


Figure 7: Open-loop and closed-loop performance (LPACT 2)

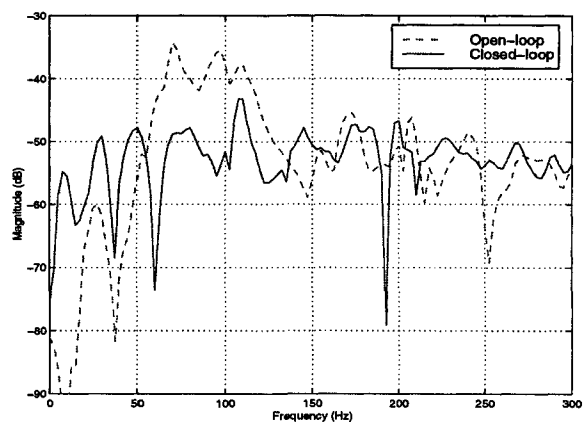


Figure 8: Open-loop and closed-loop performance (LPACT 3)

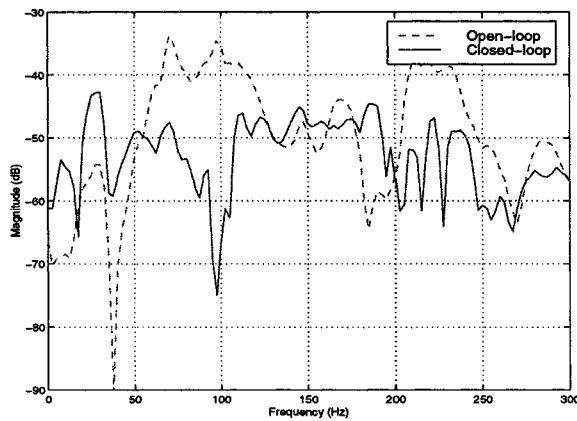


Figure 9: Open-loop and closed-loop performance (LPACT 4)

Conclusions

In this paper, the implementation of decentralized EAA controllers for MIMO active vibration control was studied experimentally on the MHPE test bed. Two decentralized EAA controllers were implemented, and attenuation of harmonic and broadband disturbances at four sensor locations was achieved using four actuators. However, performance was found to depend on how the on-line identification of the secondary paths was implemented with respect to operational modes of the controllers. A switching sequence which accounts for the limitations in the identification process was developed and implemented. The results of this study indicate that a centralized supervisor that controls the mode switches in the independent adaptive controllers is required, and future research will include the design and implementation of such a supervisor.

References

- [1] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd edition, Reading, MA: Addison-Wesley, 1995.
- [2] R. L. Clark, W. R. Saunders, and G. P. Gibbs, *Adaptive Structures Dynamics and Control*, John Wiley and Sons, New York, 1998.
- [3] T. Van Pelt, R. Venugopal, and D. S. Bernstein, "Experimental Comparison of Adaptive Cancellation Algorithms for Active Noise Control," *Proc. Conf. Contr. Appl.*, Hartford, CT, pp. 559-564, October 1997.
- [4] R. Venugopal and D. S. Bernstein, "Adaptive Disturbance Rejection using ARMARKOV Sys-

tem Representations," *IEEE Trans. Contr. Sys. Tech.*, Vol. 8, pp. 257-269, March 2000.

- [5] H. Sane, R. Venugopal, and D. S. Bernstein, "Disturbance Rejection Using Self-Tuning ARMARKOV Adaptive Control with Simultaneous Identification," *Proc. Amer. Contr. Conf.*, pp. 2040-2044, San Diego, CA, June 1999.
- [6] S. L. Lacy, R. Venugopal, and D. S. Bernstein, "ARMARKOV Adaptive Control of Self-Excited Oscillations of a Ducted Flame," *Proc. Conf. Dec. Contr.*, pp. 4527-4528, Tampa, FL, December 1998.
- [7] H. Sane and D. S. Bernstein, "Active Noise Control Using an Acoustic Servovalve," *Proc. Amer. Contr. Conf.*, pp. 2621-2625, Philadelphia, PA, June 1998.
- [8] J. C. Akers and D. S. Bernstein, "Time-Domain Identification Using ARMARKOV/Toeplitz Models," *Proc. Amer. Contr. Conf.*, Albuquerque, NM, pp. 191-195, June 1997.
- [9] S. J. Elliot, I. M. Stothers and P. A. Nelson, "A Multiple Error LMS Algorithm and its Applications to the Active Control of Sound and Vibration," *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. ASSP-35, pp. 1423-1434, 1987.
- [10] J. Hong and D. S. Bernstein, "Bode Integral Constraints, Colocation, and Spillover in Active Noise and Vibration Control," *IEEE Trans. Contr. Sys. Tech.*, Vol. 6, pp. 111-120, 1998.
- [11] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems*, Wiley, New York, 1996.