Tr. J. of Engineering and Environmental Science
23 (1999) , 209 – ??.

Gain Scheduling Adaptive Model Control

Nafiz Aydın HIZAL
*İstanbul Technical University, Faculty of Mechanical Engineering,
Automatic Control Division, İstanbul-TURKEY*

Received 09.07.1998

Abstract

Adaptive Model Control (AMC) is an adaptive control methodology that makes use of FIR (Finite Impulse Response) adaptive modeling and that requires a minimum of a priori knowledge about the plant to be controlled. The control relies on the inversion of the adaptively obtained model. In this work, a Gain Scheduling version of this methodology is proposed and tested on the simulated pitch dynamics of the A-4D aircraft, which is a time varying plant. The proposed system is formed by combining the Adaptive Model Control system with a learning algorithm that learns and stores the AMC weight vectors under various operating conditions in a "gain scheduling matrix".

Key Words: Adaptive Control, Adaptive Model Control, Gain Scheduling, Time Varying Plants.

Kazanç Programlamalı Uyarlamalı Model Kontrolü

Özet

Uyarlamalı Model Kontrolü (UMK), sonlu impuls cevaplı filtreler ile yapılan modellemeye dayanan ve kontrol edilecek sistem hakkında minimum ön bilgi gerektiren bir uyarlamalı kontrol yöntemidir. Kontrol işlemi, uyarlamalı olarak elde edilmiş olan modelin ters çevrilmesi temeline dayanır. Bu çalışmada, bu yöntemin Kazanç Programlamalı bir şekli önerilmekte ve A-4D uçağının, zamanla değişen bir yapıda olan yunuslama dinamiğine simülasyon ile uygulanarak test edilmektedir. Önerilen sistem, Uyarlamalı Model Kontrolü sistemi ile bir öğrenme algoritmasının birleştirilmesi ile oluşmaktadır. Öğrenme algoritması, değişik şartlar altındaki UMK ağırlık vektörlerini bulup saklayarak, bir "kazanç programlama matrisi" oluşturur.

Anahtar Sözcükler: Uyarlamalı Kontrol, Uyarlamalı Model Kontrolü, Kazanç Programlama, Zamanla Değişen Sistemler.

Introduction

Adaptive Model Control (AMC) which dates from early 1970s is an adaptive control methodology proposed by Widrow and his coworkers. A significant application of Adaptive Model Control was the control of the blood pressure of a dog, which was experimentally realized (Widrow 1971, Widrow and Stearns 1985). On the other hand, Adaptive Inverse Control was proposed in the mid 1980s (Widrow 1986), and has been under development since then (Widrow and Bilello 1993, Widrow and Walach 1996, Widrow et al. 1996).

These two methodologies require a minimum amount of a priori knowledge about the plant to be controlled, and they have many attributes in common. For command tracking, they do not use feedback in the usual sense (unless the plant is unstable), but they "close" the loop through the adaptation process. They rely on adaptively obtaining the effective inverse of the plant, so that the cascade connection of this effective inverse and the plant practically emulate a transfer of unity, providing what is termed a "deconvolution" of the plant. For adaptive modeling, both of them use FIR (Finite Impulse Response) adaptive filters comprised of a tapped delay line, a linear combiner and an adaptation algorithm such as the LMS (Least Mean Squares) algorithm to adapt the weights of the linear combiner. The two methodologies differ in the manner the effective inverse of the plant is obtained; in Adaptive Model Control, a direct model of the plant is obtained adaptively, and it is inverted algebraically to generate an algorithm to be used as the cascade controller, while in Adaptive Inverse Control, the adaptively obtained model is either the inverse model which is then copied to the controller position in the system if no "disturbance cancelation" is used, or the direct model if the disturbance cancelation scheme is to be mechanized. In this latter case, the inverse model is derived from the direct model by a second adaptive process that runs offline.

Since the time it was proposed in the mid 1980s, Adaptive Inverse Control has been studied and developed to include a "reference model" for specifying the system behavior, in addition to the scheme for disturbance cancelation. It is favored for its structure that can accomodate neural nets, so that effective inverses for nonlinear plants can be obtained also. On the other hand, the algebraically inverting Adaptive Model Control methodology can also be used with the advanced features above, combined with its potential for faster adaptation. This faster adaptation potential is due to the simpler structure of an AMC system compared to that of an AIC system, and in particular, due to the fact that AMC has a single adaptive process running, while AIC has two such processes running interdependently, limiting the adaptation speeds considerably. It is shown in (Hızal 1998c) that the plant variation rates that can be handled successfully by the AMC system is much greater than those that can be handled by the AIC system. Therefore, it is concluded in the above reference that, especially for fast plant variation rates and where considerable nonlinearities do not exist, an AMC system can be superior to an AIC system.

In the present work, a gain scheduled version of the AMC system is proposed and tested, using the A-4D aircraft as a testing platform in simulations. The addition of this gain scheduling capability renders the adaptation speed irrelevant with regard to plant variations, but nevertheless an AMC system is used due to its simplicity advantage, as the plant under consideration is linear. With the Mach number as the measurable environmental parameter, a gain scheduling matrix is formed with one of its two dimensions representing the Mach number, and the other one representing the weight vector index. The adaptive process has to handle only the long term plant variations, modifying the gain scheduling matrix as required, in addition to the correction of any possible initial errors in the matrix. A learning algorithm is used for gain scheduling matrix training. This algorithm trains the matrix elements (called the "support values" in the terminology of this algorithm) according to their contributions to the result of the interpolation that produces the plant impulse response vector (the "weight vector") for any intermediate Mach number.

Adaptive Model Control

The operating principle of the Adaptive Model Control system with the "disturbance cancelation" and the "reference model" features added as proposed in (Hızal 1998c), is depicted in Figure 1. The direct modeler adaptively forms the discrete time impulse response of the plant by varying the weights of the linear combiner which receives its inputs from a tapped delay line, comprising a transversal filter (Figure 2). After convergence, the weights contain the identification information about the plant dynamics in the form of an impulse response shape. The inversion of the plant dynamics is effected algebraically by using this weight vector, and this process is named "Forward Time Calculation". The Forward Time Calculation block is placed between the command signal to be followed by the plant, and the plant input, effectively canceling the plant dynamics. Ideal inversion would result in the plant output following the command signal exactly, while the causality principle prevents this ideal behavior for plants with a deadtime. In practice, there may be further limitations due to saturation of the actuator and/or the plant input.

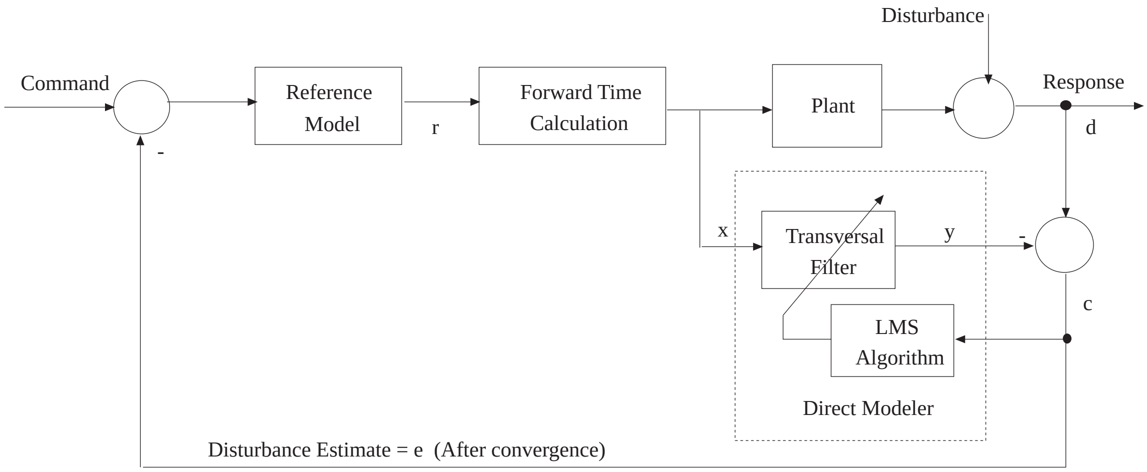

Figure 1. Block diagram of the AMC system, with the disturbance cancelation and the reference model features added.

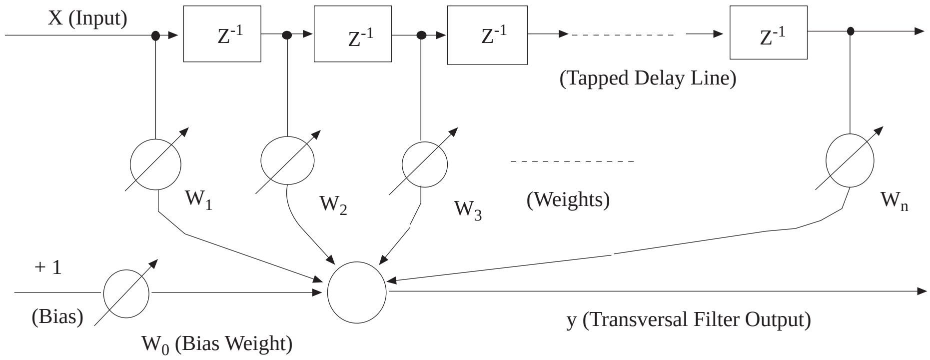

Figure 2. The transversal filter used in adaptive FIR modeling.

Having chosen an appropriate sampling period T_s for the system, the tapped delay line length n is chosen to cover the "memory time" of the plant to be modeled, i.e., a time span in which the impulse response of the plant, which is assumed to be stable, reaches an insignificantly small value. If this memory time is not known a piori for the plant in question, it is possible to make the delay line length large enough to be on the safe side. This would result in a certain number of the weights at the end of the weight vector to assume negligible values after convergence, which can then be removed in a final design. In addition to the n+1 weights that result from n delay units, there is also a "bias" weight that can model nonzero plant outputs which may be present in spite of zero inputs, and which may be due to slow, drift type disturbances and/or nonlinearities. The bias weight has a constant and unity input (Widrow and Stearns 1985).

The LMS (Least Mean Squares) adaptation algorithm devised by Widrow and Hoff (Widrow and Hoff 1960, Widrow and Stearns 1985, Widrow and Walach 1996) is a steepest descent algorithm using an approximation for the gradient. The descent is on a "performance surface" defined by the expected value of the error e, which is the difference between the actual modeler response and the desired response. The desired response is the response the modeler is expected to give after convergence. In case of adaptive modeling, the modeler receives the same input signal as the system to be modeled, and the desired response is the output of the plant to be modeled. Denoting the weights as the vector **W**, the LMS algorithm gives the weight vector **W**(k+1) at the k+1 st instant, as

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\mu e(k)\mathbf{X}(k) \tag{1}$$

where the vector $\mathbf{X}$ denotes the linear combiner input (tapped delay line output), and μ is a convergence constant. The convergence constant μ controls the stability, the convergence speed, and the convergence behavior with regard to the presence and the nature of adaptation oscillations. It also controls the "misadjustment", which is extra error due to the fact that the gradient used by the algorithm is an approximation. The difference with the actual gradient may be interpreted as noise which is low-pass filtered through the slowness of the adaptation whose speed can be controlled by this convergence constant. Using the expectation operator E over the time index k, the autocorrelation matrix of the input is defined as

$$\mathbf{R} = E_k[\mathbf{X}\mathbf{X}^T] \tag{2}$$

Stability of the LMS algorithm is guaranteed if the convergence constant μ is selected within the range (Widrow and Stearns 1985)

$$0 < \mu < (1/trace(\mathbf{R})) \tag{3}$$

The expected value of the weight vector converges to the optimal, or Wiener, vector $\mathbf{W}^\star$ which is given by

$$\mathbf{W}^\star = \mathbf{R}^{-1}\mathbf{P} \tag{4}$$

where the crosscorrelation vector $\mathbf{P}$ between the desired response d and the input vector $\mathbf{X}$ is defined as

$$\mathbf{P}^T = E_K[d\mathbf{X}^T] \tag{5}$$

The purpose of giving Equations (2) and (5) here (which are not used in the present methodology) is to make a brief introduction to the LMS algorithm. The LMS Algorithm and transversal filter combination has been used for adaptive FIR modeling of many linear, and some nonlinear systems (Hızal 1982, Hızal 1984). Extensive reference lists on this method of modeling are available in (Widrow and Walach 1996).

For control purposes, the model has to be used to generate a controller, and this is achieved by the Forward Time Calculation process. The linear combiner, with its weights w_i and the bias weight w_0, leads to the equation.

$$\sum_{i=1}^{n+1} x(k-i+1)w_i(k) + w_0(k) = r(k) \tag{6}$$

if its output is assumed to be equal to the command signal r(k), which is practically the case after convergence of the model. This gives the required input to the plant at the k th instant, x(k), as

$$x(k) = \frac{1}{w_1(k)}\Bigg[r(k) - w_0(k) - \sum_{i=2}^{n} x(k-i+1)w_i(k)\Bigg] \tag{7}$$

and this constitutes the Forward Time Calculation algorithm.

A problem is encountered if the first weight of the model, namely w_1, is zero or very small. For second or higher order plants, the impulse response starts from zero, therefore w_1 is zero. If the plant has deadtime, then any number of the initial weights can be zero. Denoting the first nonzero weight as w_p (or, in an actual application, the first one that is large enough not to cause any numerical computation problems), the forward time calculation takes the form

$$x(k) = \frac{1}{w_p(k)}\Bigg[r(k+p-1) - w_0(k) - \sum_{i=p+1}^{n} x(k-i+p)w_i(k)\Bigg] \tag{8}$$

The appearance of r(k+p-1) in this expression indicates that p-1 future command values will be required. For some applications, this may be possible. If not, r(k) is used instead, with the assumption that r varies slowly compared to the sampling period, leading to the plant's following the command signal with a p-1 step delay, which is inevitable with a plant having deadtime.

The difference between the plant output and the direct modeler output is an estimate of the disturbance. After changing its sign, it is injected at the summing point on the left, modifying the command signal (Figure 1). The reference model is placed in cascade with the plant. Effectively canceling the plant dynamics by the forward time calculation, there remains the reference model dynamics as the dominant dynamics of the entire system.

Table 1.

Flight Condition	K [s^{-1}]	τ[s]	$\omega_n[rads^{-1}]$	ζ
Mach 0.4	-0.1114	11.98	0.933	0.1301
Mach 0.6	-0.2052	7.494	1.428	0.1370
Mach 0.8	-0.2890	5.108	1.943	0.1524

The Gain Scheduling Matrix Learning Algorithm

The gain scheduling matrix training algorithm proposed for use here allows the use of a finite number of "support points" in the Mach number direction, thus enabling the definition of such a matrix. This algorithm was originally used for building heating automation (Leimgruber et al. 1984, Leimgruber et al. 1988) which required learning of the building thermal dynamics. In its original form, it has a two dimensional input space, but its dimensions can be increased as in (Hızal 1997) where wind speed effects are taken into account also. In (Hızal 1998a), properties of this algorithm was studied, and being a learning algorithm, it was applied to a jet engine characteristics monitoring problem (Hızal 1998b).

The algorithm has the estimation and the training stages. Estimation is performed by multidimensional linear interpolation, using a number of "support" values, represented by the gain scheduling matrix in the present problem. Training is based on the correction of these support values according to their contributions to the interpolation result, and according to the estimation error.

In the original two dimensional form of the algorithm, the ranges of the input variables X and Y are subdivided into m_x and m_y regions respectively, thus generating and (m_x+1) by (m_y+1) dimensioned grid, at the intersections of which the support values will be located. The estimation at a particular input point (X_1, Y_1) gives the estimate $Z_E = f_E(X_1, Y_1)$. The actual Z value, Z_M, must be measured at the time of the estimation (as in the engine characteristics application) or after using the estimate (as in the building automation application) so that the error can be used in the training stage following this estimation.

A single cell of the input space grid is shown in Figure 3. If the present input point (X_1, Y_1) is within the indicated cell, the four surrounding support values are to be used for a bilinear interpolation, completing the estimation phase. This procedure at a particular time step k is represented by the equations

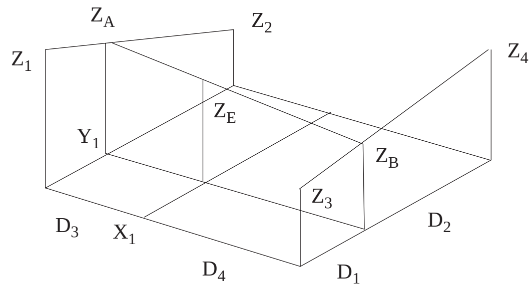

Figure 3. One cell of the learning algorithm. (X_1, Y_1) is the input point.

$$Z_A = Z_1 + (Z_2 - Z_1)D_1/(D_1 + D_2) \quad (9)$$

$$Z_B = Z_3 + (Z_4 - Z_3)D_1/(D_1 + D_2) \quad (10)$$

$$Z_E = Z_A + (Z_B - Z_A)D_3/(D_3 + D_4) \quad (11)$$

where the time index k is omitted, D_i are the distances of the input point from the cell edges, Z_i are the neighboring support values, Z_A and Z_B are the auxiliary estimates, and Z_E is the estimate.

To train the system, the actual value Z_M is used to calculate the "error factor" F which is the relative error.

$$F = (Z_M - Z_E)/Z_E \quad (12)$$

The four support values Z_i are corrected according to the error factor F and with varying degrees that depend on their contributions to the estimate, represented by the correction weights K_i which are calculated as functions of the distances D_i.

$$K_1 = [D_2/(D_1 + D_2)][D_4/(D_3 + D_4)] \quad (13)$$

$$K_2 = [D_1/(D_1 + D_2)][D_4/(D_3 + D_4)] \quad (14)$$

$$K_3 = [D_2/(D_1 + D_2)][D_3/(D_3 + D_4)] \quad (15)$$

$$K_4 = [D_1/(D_1 + D_2)][D_3(D_3 + D_4)] \quad (16)$$

Equation (17) below gives the correction expression for the i'th support value, its new value $Z_i(k+1)$ being calculated from the quantities at the time step k.

$$Z_i(k+1) = Z_i(k)[1 + K_i(k)F(k)] \quad (17)$$

In (Hızal 1998a, Hızal 1998b), the effects of including a concergence constant on stability and performance are investigated, as well as the effects of the number of support values on the accuracy.

If the nonlinear function to be learned has zero values, then Equation (12) would mean division by zero. In this case a bias can be added to the function to move it away from zero. It is also possible to use the absolute rather than the relative error, and this is done in the present work.

Although the gain scheduling matrix has two dimensional data in the present problem, one of its dimensions, being the weight vector index, is not subject to intermediate inputs, therefore the use of this algorithm here has a one dimensional input space only. If the measurable environmental parameters resulting in plant variations were more in number, the dimensionality of the algorithm would increase accordingly.

The Combination of the Two Algorithms

The combined use of the LMS dynamic modeling algorithm and the gain scheduling matrix training algorithm leads to the following procedure: At each time step, a linear interpolation is performed in the Mach number direction of the gain scheduling matrix to obtain the weight vector for the present Mach number, to be used in the direct modeler and the forward time calculation block. The modeling error that results at this step is used in the LMS algorithm to obtain a weight correction vector δw, as

$$\delta w_k = 2\mu e_k \mathbf{X}_k \tag{18}$$

with the notation used in Eqn. (1), and where e_k replaces F in (12). This δw_k is divided into two parts, to be applied to the two neighboring support vectors that were used in the interpolation. The division is according to the linear relationships

$$\delta w_{k1} = K_1 \ \delta w_k, \quad \delta w_{k2} = K_2 \ \delta w_k \tag{19}$$

where

$$\begin{aligned} K_1 &= (M - M_1)/(M_2 - M_1), \\ K_2 &= (M_2 - M)/(M_2 - M_1) \end{aligned} \tag{20}$$

with M being the current Mach number, and M_1 and M_2 standing for the two neighboring support Mach values.

In the algorithm generated as a combination of the two algorithms, the convergence constants of the two algorithms appear as a product, hence a single convergence constant μ is defined as this product. The fact that the constant K_i <1 are included in the training of the support values leads to a decrease in the convergence constant values from the point of view of the LMS algorithm. Although the amount of this decrease varies depending on the Mach value (training point) at a particular instant, by assuming a uniform distribution of the training points, it is clear that the mean reduction factor is 1/2, since the constants K_i vary within the range [0,1] (Eqn. 20). However, for a learning algorithm with a two dimensionla input space, the constants K_i are found as the product of the normalized coordinates of the training point within a cell defined by four neighboring support points (Eqs. 13-16), in which case the reduction factor is 1/4 as can be seen from the double integral

$$\int_{y=0}^{1} \int_{x=0}^{1} x \ y \ dx \ dy = 1/4 \tag{21}$$

where the normalized coordinates within a cell are called x and y for convenience, and K=x y. This entails the use of a larger convergence constant μ for the same LMS algorithm performance, and therefore the stability limits with regard to μ are increased inversely proportionally with this factor. This was observed in the present study.

The Testing Platform

The plant model used for testing the proposed gain scheduling adaptive scheme is that of the A-4D fighter aircraft with regard to its longitudinal short-period dynamics. It was not the aim of this work to design a control system for A-4D, but this plant was used as a testing platform, as it has time variability in addition to being a rather difficult plant to model by a FIR filter, as explained below.

The aircraft short period equations (Blakelock 1991), combined with the numerical values for the A-4D aircraft (Nelson 1990) gives the block diagram in Figure 4, where the typical dynamics of an elevator servo is added also. The parameter values for three flight conditions are given in Table 1. Polynomial fits are used for intermediate Mach numbers within the range [0.4, 0.8].

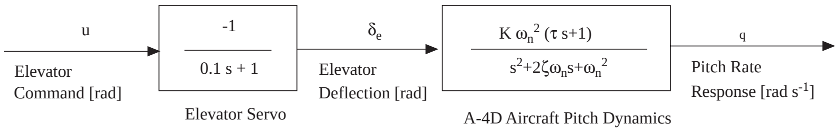

Figure 4. Transfer functions of the A-4D pitch dynamics and the elevator servo.

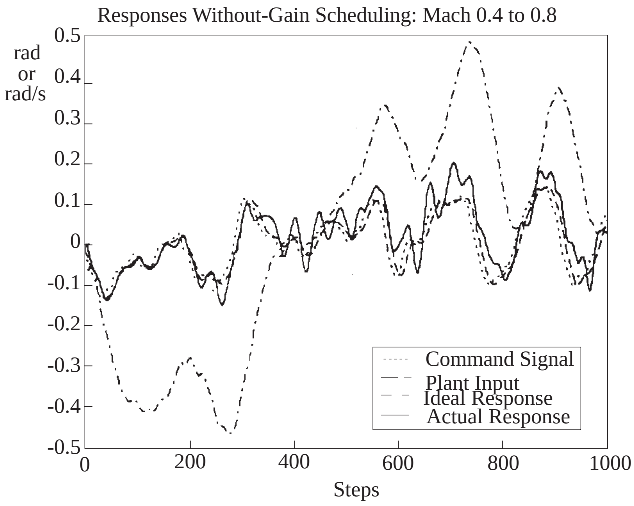

Figure 5. Time response of the AMC system without gain scheduling.

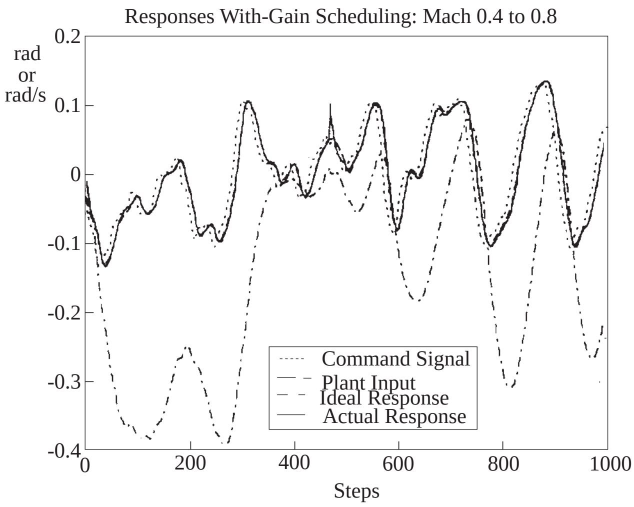

Figure 6. Time response of the AMC system with gain scheduling. There is a step disturbance at time step 470.

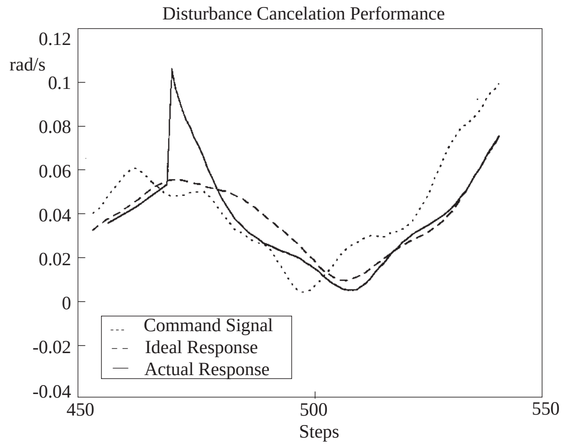

Figure 7. Details of the disturbance cancelation response in Figure 6.

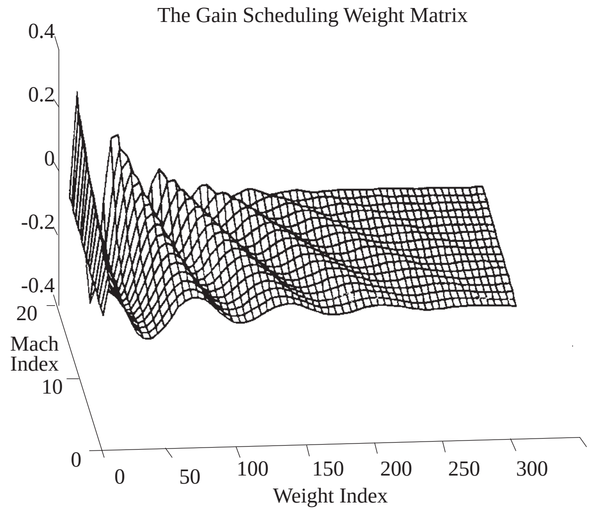

Figure 8. The gain scheduling matrix for A-4D, also depicting the variation of its impulse response with the Mach number.

Figure 8, being the gain scheduling matrix, also shows the pitch rate response shapes of A-4D for various Mach numbers. It can be observed that the response curves change considerably with the Mach number, and generally, the damping is seen to be too low. The low damping makes this plant a rather difficult one to model by a FIR filter, as the sampling period is to be selected according to the highest frequency component in the response, while the time span of the FIR filter is to be selected according to the "memory time" of the system, which is long due to the low damping, resulting in a large number of weights in the adaptive modeler. In accordance with Eqn. (3), the large size of the weight vector limits the useful range of convergence constants μ by stability considerations. The aircraft was considered to be an appropriate testing platform because of this high dimensionality in its modeled dynamics. An appropriate sampling period was T_s=0.1 s, and the resulting weight vector size to cover a time span of 30 seconds was 300 elements.

Performance Evaluations

The main parameters of the proposed system that a designer needs to consider are the locations of the support points in the Mach number direction, the convergence constant, the reference dynamics to be represented by the reference model, in addition to the two parameters that are directly apparent from the plant dynamics, namely, the sampling period and the weight vector (or tapped delay line) length.

In this work, a third order reference model was used, with the transfer function

$$G(s) = \frac{1}{a_3 s^3 + a_2 s^2 + a_1 s + 1}, \quad a_3 = 1/27,$$
$$a_2 = 9/27, a_1 = 22/27. \qquad (22)$$

The selection of the reference model is primarily dependent on the specific application area. In case of an aircraft control system for instance, the reference model dynamics would depend on the performance expected from the aircraft, also considering e.g. the control effort and the structural limitations. On the other hand, the selected reference model affects the performance of the gain scheduling system. A faster reference model is more demanding with regard to support values densities of the gain scheduling matrix. In the present study, however, the A-4D aircraft acted only as a testing platform, so a reasonable reference model was appropriate, without further considerations due for a complete pitch control system design.

Time response curves with a random command input signal are given in Figure 5 and Figure 6. Figure 5 depicts the case with no gain scheduling, the AMC weights reamining as those for Mach 0.4, while actually the Mach number changes from 0.4 to 0.8 along the horizontal axis in the diagram. Compared to the ideal response which is that of the reference model, the actual response is satisfactory at the beginning, but deteriorates as the Mach number increases in time. On the other hand, in Figure 6, gain scheduling is active, and the performance remains uniformly satisfactory in time as the Mach number range from 0.4 to 0.8 is covered. At time step 470, there is a step disturbance applied on the plant output. It can be seen that the "disturbance cancelation" mechanism of the AMC eliminates its effects within a short time, returning the response close to that of the reference model. This disturbance cancelation response region is enlarged and given in Figure 7.

Figure 8 shows the gain scheduling matrix in three dimensions, with the horizontal input plane having the components "Mach number index" and the "weight vector index". The support point locations in the Mach number direction need not be uniformly distributed, and an appropriate distribution reduces the required number of points, hence the gain scheduling matrix size, for a given level of accuracy. Denser placement of the support points where the curvature of the surface to be generated is greater, leads to a more efficient arrangement regarding the relationship between the gain scheduling matrix size and a required accuracy level. The 17 support points used in the Mach number direction are given in Figure 9. With a weight vector size of 300 elements, the gain scheduling matrix size becomes 17×300 in this case. However, coarser Mach resolution was also satisfactory, down to 5 supports in this direction, shown as full circles in Figure 9. To investigate this, a "mean square error" study was conducted, with its results given in Figure 10. With only two supports (i.e. a single interpolation region covering the entire Mach range under consideration), the response was excessively oscillatory and totally unacceptable.

The constant μ in this case is a combined convergence constant of the two algorithms. Though it determines, among other things, the convergence speed, in this application the speed requirements are not determined by the plant time variation rates, because of the presence of gain scheduling. It is only

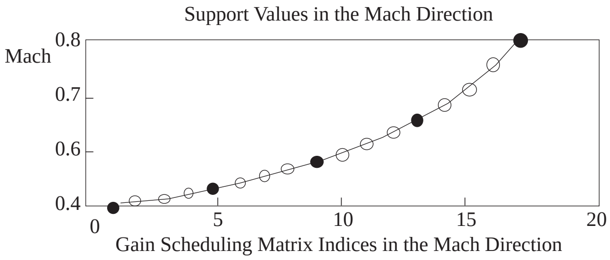

Figure 9. The support values in the Mach direction, determining the resolution of the gain scheduling matrix in that direction.

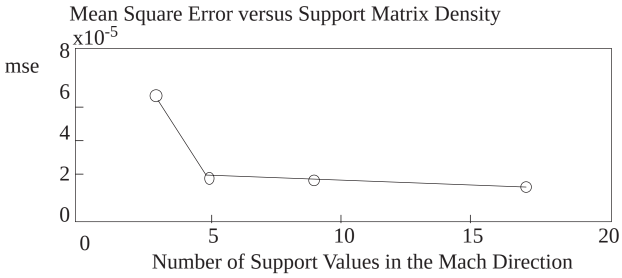

Figure 10. Mean square errors versus the number of support values in the Mach direction.

necessary to achieve a learning rate that will be able to follow any long term plant variations manifested as a change in the gain scheduling matrix support values. This makes it possible to use a relatively small convergence constant μ, leading to better performance in other respects, namely, being away from instability conditions by a large margin, low sensitivity to gradient noise which is inherent in the LMS algorithm, and low sensitivity to external noise sources. A combined convergence constant value of μ=0.002 was successfully used throughout the investigations, and this resulted in the correction of any initial errors in the gain scheduling matrix, whose initial values were determined analytically by the theoretical impulse responses at various Mach numbers.

Conclusions

The results indicate that the proposed combination of the adaptive modeling algorithm and the learning algorithm produces a viable gain scheduling system that exhibits the desirable behavior of gain sheduling systems, namely, the lack of a speed limit for the plant variations that can be handled succesfully, in addition to the desirable feature of adaptivity in the form of a capability for correction of the gain scheduling matrix in case of long term changes in the plant characteristics, and also for the correction of any possible initial errors in that matrix.

The selection of a convergence constant μ is not critical for this medhodology, since the adaption speed requirements are relaxed and a relatively small μ value will suffice, eliminating other problems related to the adaptation speed, namely the stability of the system, the gradient noise and the external measurement noise sensitivities.

An appropriate distribution of support points minimizes the gain scheduling matrix size for a given accuracy, though with modern digital equipment, neither the memory capacities nor the operation speeds are likely to be a limiting factor for the size of this matrix. Although the matrix size affects also the adaptation speeds, this latter is not critical in the present methodology.

References

Blakelock, J. H., “Automatic Control of Aircraft and Missiles”, 2nd ed., Wiley, 1991.

Hızal, N. A., “Simulation Study on Adaptive Modeling in a Heat Transfer Channel “(in Turkish), Ph D Thesis, Faculty of Electrical and Electronics Engineering, İstanbul Technical University, 1982.

Hızal, N. A., “Simulation Study on Adaptive Modeling in a Reactor Heat Transfer Channel”, Annals of Nuclear Energy, Pergamon Press, 11/5: 207-211, 1984.

Hızal, N. A., “Control of Building Heating Turn-on and Turn-off Times by a Learning System (in Turkish)”, Proc. MAMKON ‘97, Istanbul Technical University, Faculty of Mechanical Engineering, 1997.

Hızal, N. A. “A Study on a Learning Algorithm”, (in Turkish), TÜBİTAK Journal of Engineering and Environmental Sciences, Vol.22, No.2, pp.109-116 , 1998a.

Hızal, N. A., “Monitoring Jet Engine Characteristics by a Learning Algorithm”, ASME Journal of Gas Turbines and Power, Submitted, 1998b.

Hızal, N. A., “Improved Adaptive Model Control”, ARI, Springer-Verlag Accepted, 1998c.

Leimgruber, J., Hribar, R., Bieri, R., “New Approaches in Building Automation: Control system SICOS 2000”, Sulzer Technical Review, 66/4:28-30, 1984.

Leimgruber, J., Lipsky, P., Reichlin, A., "Gradient Method for Calculating Optimal Switch Timings in Heating and Air Conditioning Plants", Sulzer Technical Review, 70/1:19-21, 1988.

Nelson, R. C., "Flight Stability and Automatic Control", McGraw-Hill Aerospace Series, 1990.

Widrow, B., Hoff, M. E., "Adaptive Switching Circuits", IRE Wescon Convention Record, Part 4, pp 96-104, 1960.

Widrow, B., "Pattern Recognition and Machine Learning", Plenum Press, pp 310-324, 1971.

Widrow, B., Stearns, S. D., "Adaptive Signal Processing", Prentice-Hall, 1985.

Widrow, B., "Adaptive Inverse Control", Proc. of the 2nd IFAC Workshop, Lund, Sweden, pp 1-5, 1986.

Widrow, B., Bilello, M., "Adaptive Inverse Control", Proc. of the 1993 IEEE Int. Symp. on Intelligent Control, Chicago, USA, pp 1-6, 1993.

Widrow, B., Walach, E., "Adaptive Inverse Control", Prentice-Hall, 1996.

Widrow, B. et al., "Adaptive Inverse Control Based on Linear and Nonlinear Adaptive Filtering", Proc. of Int. Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, NICROSP 1996, IEEE, Los Alamitos, USA, pp 30-38, 1996.