M.'D. Peek and P. J. Antsaklis, "Parameter Learning for Performance Adaptation," I EEE C ontrol S ystems M agazine , pp.3-11, December 1990.

Parameter Learning for Performance Adaptation

Mark D. Peek and Panos J. Antsaklis

A parameter learning method is introduced and used to broaden the region of operability of the adaptive control system of a flexible space antenna. The learning system guides the selection of control parameters in a process leading to optimal system performance. The method is applicable to any system where performance depends on a number of adjustable parameters. A mathematical model is not necessary, as the learning system can be used whenever the performance can be measured via simulation or experiment.

Introduction

The original motivation for this work was the desire to enhance the control of the transient behavior of a flexible space antenna. Flexible space structures pose unique control problems because of the complexity of their dynamic behavior, the limited knowledge of the model, and the uncertainty of the environment about the types of disturbances that will be encountered. While adaptive control has shown potential in controlling such systems [14] and offers good disturbance rejection, the region of operability of the control system is rather restrictive. This is because the adaptive controller parameters are typically designed based on convergence and stability analysis only, and this may place severe limitations on the performance of the compensated system. These limitations can be seen as restrictions on the acceptable operating region of the controller, since outside this region the system, although it may be stable, can exhibit such completely unacceptable behavior as excessive overshoot and oscillations. By enhancing the conventional controller with learning, one effectively expands the region of operability of the controller and creates a more robust controller. The control system can then compensate for a larger number of changes in the plant and its environment. Future control systems will have to be more robust to changes and will need a higher degree of autonomy than the systems of today. Autonomous systems require a high degree of flexibility to adapt to situations which cannot be predicted [3], and adaptive behavior of this type is not offered by conventional adaptive control systems. Learning methods appear to be useful in expanding the region of operability of the controller, and thus offer a higher degree of autonomy.

It is perhaps appropriate at this point to briefly discuss learning and its application to control problems. We are interested in the ability of man-made systems to learn from experience and, based on that experience, to improve their performance. Thus, we start with a working definition of such learning. Learning is the process whereby a system can alter its actions to perform a task more effectively the next time, due to increases in knowledge related to the task. For example, a control system may change the type of controller used, or vary the parameters of the controller, after learning that the current controller does not perform satisfactorily within a changing environment. The ability to learn entails such issues as knowledge acquisition, knowledge representation, and some level of inference capability.

Learning, considered fundamental to intelligent behavior, has been the subject of research in the field of machine learning for over twenty years (see [15]) and has gained a renewed interest in the artificial intelligence community. Machine learning has also attracted renewed attention in the control community [1],[2],[7],[12],[13]. Alternate learning vehicles, in the form of expert system technology, and rule-based programming in particular, are providing new methods to implement learning strategies. Neural networks with their massive parallelism and their ability to learn also offer exciting possibilities for learning in control systems [20].

The goal of the machine learning method proposed in this article, as applied to the flexible space antenna, is to broaden the region of operability of the adaptive control system by allowing the controller parameters to better adapt to different plant and environmental conditions. These operating conditions may cause the nominal adaptive system to exceed the tolerances of its design. The parameter learning system determines parameter values for optimal performance for given operating conditions and then stores them in memory. In this way, the controller is able to operate effectively over a wider region. It should be noted that, at the local level, the parameter learning system performs parameter auto-tuning. The overall system, however, does more than this: it uses the results of auto-tuning to expand the region of operability of the control system.

It is very difficult to control the transient response and the performance of an adaptive control system. This is due to the mathematical complexity of the nonlinear relationship between the design parameters in the adaptive controller and the output of the compensated system, even for quite simple plants. Only recently, some progress has been made for low order plants and particular adaptive controllers [4],[5]. The approach introduced in this article is general and it offers a viable alternative to analytical approaches. It can be used either when no analytical methods exist or when existing methods are too cumbersome.

The learning approach proposed in this article is applicable to any control system where performance depends on a number of adjustable parameters. The mathematical relation between the performance and the parameters does not need to be known. Given particular values for the parameters, a performance index is evaluated via computer simulation or physical experiment. If the mathematical relation is known, the performance evaluation can be done directly.

The learning system determines the next set of parameters in a process leading to an optimum performance. In effect, the learning system guides the selection of parameters for optimization; this procedure seems to be a form of learning by observation and discovery rather than learning by example, since the experiments are generated by the system itself. After the best parameter values have

Mark D. Peek is with Tellabs Research Center, 3702 N. Main St, Bldg. 2, Suite 101, Mishawaka, IN 46545. Panos J. Antsaklis is with the Dept. of Electrical & Computer Eng., University of Notre Dame, Notre Dame, IN 46556. This work was partly supported by the Jet Propulsion Laboratory, Pasadena, CA, under Contract 957856.

0272-1708/90/1200-0003$01.00©1990IEEE

been discovered, they are stored in memory along with the corresponding operating conditions. This provides the memory which is a necessary element for learning. When the same operating conditions recur (here it is assumed that information about the operating conditions is provided to the learning system), the system selects the appropriate controller parameter values it has learned. Interpolation is used to select values when similar, but not the same, conditions occur.

In this article, a general format of the parameter learning system is first outlined. Next, the model of the antenna and the antenna parameter learning system are described. The results of two experiments, the transient regulation and the command following experiment, are then presented. Note that several of the above results have appeared in [11],[16],[17].

The Parameter Learning System

The role of the parameter learning system described here is first to determine the best parameter values given certain system operating conditions, and then to store these values in memory for future use. This parameter learning method is applicable to any system where performance depends on a number of adjustable parameters. Furthermore, a mathematical model is not necessary, as the learning system can be used whenever performance can be measured via simulation or experiment. A particular application of this learning system to the control of a space antenna is described in the next sections of this article.

The functional diagram of the parameter learning system is given in Fig. 1. First, initial parameter values are assigned; these values can be assigned randomly or based on information about the system's behavior from data stored in memory, or by some other method as discussed below. This current set of parameter values X_k is sent to the system and system performance is evaluated by computer simulation or physical experiment. Here the performance is measured via a performance index J and it is assumed that the parameters X_k and the performance index J_k are related by the following where the function $f(\cdot)$ is typically unknown:

$$J_k=f(X_k). \tag{1}$$

The performance of the system is actually evaluated using measurable quantities Y_k via a different equation where $g(\cdot)$ is a known function:

$$J_k = g(Y_k). \tag{2}$$

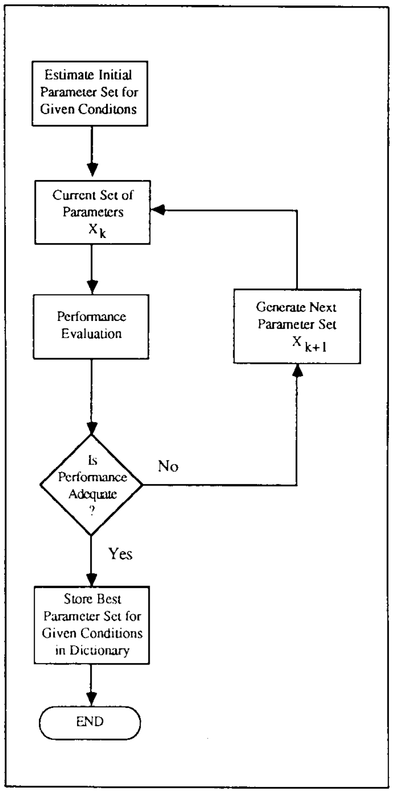
Estimate Initial Parameter Set for Given Conditons
Current Set of Parameters X_k
Performance Evaluation
Generate Next Parameter Set X_{k+1}
Is Performance Adequate ?
No
Yes
Store Best Parameter Set for Given Conditions in Dictionary
END

Fig. 1. Functional diagram of the parameter learning system.

This is accomplished as follows: as X_k vary, the measurable quantities Y_k reflect the changes in system performance, and J_k is then evaluated via (2). To illustrate, in the antenna parameter learning system described below, Y_k are measurable quantities such as settling time and maximum output error, while $g(\cdot)$ of (2) is chosen to be a weighted sum of these quantities.

It should be stressed that in a particular problem, given the adjustable parameters X_k, there are typically many appropriate choices for Y_k and J_k. It is up to the designer to select Y_k so that they are good measures of the changes in performance and at the same time easy to determine. In the parameter learning system of Fig. 1, the performance of the system is then judged to be adequate or inadequate. If inadequate, a new set of parameter values X_{k+1} is generated to improve the performance. Since the function f(·) in (1) is not known, an optimization method that does not require a mathematical model is used to generate X_{k+1}. Of course, if the function f(·) is known, other optimization algorithms may be used and (2) may not be necessary. This process continues until the performance is judged adequate. At that time, the best parameter values found are stored in memory, which here is taken to be a dictionary containing, in each entry, the given system operating conditions and the corresponding best parameter values.

The main objectives in developing this approach and applying it to the space antenna control problem have been the effective use of all available information and the learning system's speed of response. This task appears plausible because the interest is in developing a learning method for a rather specific class of problems where the available information is well defined. It is important to utilize the available information about the system dynamical behavior because, and this needs to be stressed, the more the system knows the faster it can learn.

Whether or not the parameter learning system is invoked will depend upon the time restrictions placed on determining a new parameter set and whether learning is necessary. After the operating conditions have been presented to the learning system, the dictionary containing information about such conditions is consulted to determine if they are known by the system. If these conditions are known, the parameters can be set appropriately, and learning is not required. If the conditions are unknown, a decision to enable learning is made. If time allows, learning is enabled. Otherwise, the parameter values are estimated from known conditions in the dictionary using, for example, an interpolation method. Or the parameter values may be left unchanged. The decision-making mechanism to determine whether or not the learning system will be invoked is beyond the scope of this project and thus has not been implemented.

The Antenna Parameter Learning System

The parameter learning system described above is now used for the selection of parameter values for the adaptive control of a large flexible space antenna. The adaptive control system to be considered, shown in Fig. 2, is taken from [14]. For the design of the adaptive controller, a model of the antenna including only the twelve boom-dish modes was considered. These modes involve motion of the boom, hub, and dish structures of the antenna together. This model is decoupled into two single-input single-output subsystems, each with six modes; one contains the six boom dish modes in the rib axis 1-7 of the antenna, and the other the six modes in axis 4-10, orthogonal to axis 1-7 [14]. For simplicity, only one of these

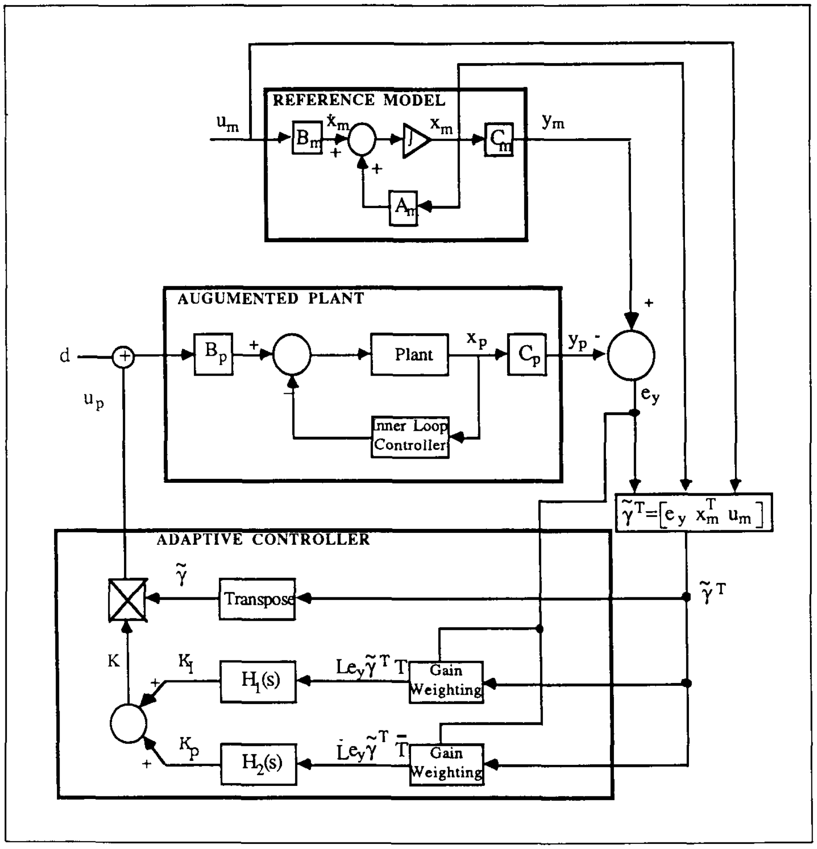

Fig. 2. Adaptive control system block diagram.

subsystems is considered, the one containing the boom-dish modes in axis 1-7. The input and the output of this system are a hub torquer and a hub angle sensor, respectively. This is a single-input single-output system of order twelve, represented by the standard equations, where the state feedback gain matrix G was chosen to satisfy the conditions of the adaptive controller design method [14]:

$$x_p(k + 1) = (\Phi_p + \Gamma_p G)x_p(k) + \Gamma_p u_p(k) \tag{3}$$

$$y_p(k) = C_p x_p(k) \tag{4}$$

Further details of the plant model are found in [4],[10],[14], and [19].

The adaptive control action u_p is defined as shown, where

$$\tilde{\gamma}^T = [\,e_y x_m^T u_m\,]$$

with x_m of dimension r, the dimension of the modeled system, and e_y, u_m scalars; and k is a vector gain of dimension $r + 2$:

$$u_p = k^T\tilde{\gamma} \tag{5}$$

The vector gain k has the following form, where k_I and k_P, the integral and proportional gains, respectively, are given by [14]:

$$k = k_P + k_I \tag{6}$$

$$\dot{k}_I = -\sigma_1 k_I + Le_y\tilde{\gamma}^T T \tag{7}$$

$$\dot{k}_P = -\sigma_2 k_P + \bar{L}e_y\tilde{\gamma}^T\bar{T} \tag{8}$$

The adjustable controller parameters are $S = \{\sigma_1, \sigma_2, L, \bar{L}, T, \bar{T}\}$. They were originally selected in an *ad hoc* fashion, mainly to satisfy stability requirements. It is desirable to select their values in the design process to optimize system performance. However, there is no systematic, analytical method available for designers to make the best possible choice. This is due to the complexity of the transient response analysis of (nonlinear) adaptive control systems, which makes the analytical relation between performance and the actual parameter values extremely difficult to determine. Here we shall determine the optimal performance via a non-analytical, systematic method.

It is desirable to select the control parameters to optimize performance for pulse disturbances and step command inputs. Note that the main purpose of the work is to demonstrate the applicability of the method, and these conditions along with the particular antenna model were chosen for simplicity; more complicated model and operating conditions could have been used. The search procedure used to generate a new parameter set X_{k+1} is a modified version of the Hooke and Jeeves multidimensional search algorithm [6]. The performance of the system is evaluated using measurable system quantities and is defined as shown with w_1, w_2, and w_3 as weighting factors:

$$J = w_1 * RMS + w_2 * ME + w_3 * ST \tag{9}$$

The RMS is defined in the usual way where N is the number of iterations in the simulation:

$$RMS = \left| (1/N) \sum_{k=1}^{N} e_y^2(k) \right|^{1/2} \tag{10}$$

ME is the largest absolute value of the output error e_y, and ST is the time it takes the output to settle to 4% of its maximum value. In the simulations, w_1, w_2, and w_3 were chosen as 100, 10, and 0.1 to equally weight each term of the performance index so that no particular term is favored in the determination of the best solution.

In the remainder of this section, the components of the antenna parameter learning system are discussed in detail.

The Grid Search Procedure

The grid search procedure is used to estimate an initial set of parameter values and it is optional. The goal of the procedure is to approximately characterize the performance surface and to extract information about the system behavior for the learning system. Grid search is typically done when searching for global instead of local minima, and is performed only once, when the dictionary (memory) is empty. The procedure uses information about values and range of the parameters, given by the original control design of the antenna, and obtains an approximate mapping of the performance surface. The data collected during the evaluation process includes values for each parameter, the performance index, and all components of the performance index (RMS, ME, and ST). This data is sorted and stored.

M. D. Peek and P. J. Antsaklis, "Parameter Learning for Performance Adaptation," I EEE C ontrol S ystems M agazine , pp.3-11, December 1990.

The three performance indices of lowest value and the associated controller parameters are then used by the optimization search procedure to determine an optimal set of controller parameters. The final three parameter sets from the optimization search are stored in memory for future use. After this initialization, memory is consulted for a starting point anytime a new operating condition is encountered.

The Optimization Search Procedure

The optimization search procedure uses a variation of the Hooke and Jeeves multidimensional search algorithm [6] described below. Formulated as an unconstrained, nonlinear optimization problem, the search for the optimal parameters of the adaptive controller is based on the performance index (9). In the optimization search procedure, it is typically assumed that the performance surface is strictly quasiconvex. The local versus global minimum problem is not addressed at this level. It is also assumed that there is no magnitude constraint on the values of the parameters, although precautions are taken to guard against sudden increases in parameter values, thus keeping the system in a desirable parameter region. The search procedure alternates between an exploratory search and a multidimensional pattern search of the performance surface until an optimal controller parameter set is obtained. It is important to note that the mathematical relation between the parameters of the adaptive control system and the performance index is unknown. A rule-based system monitors the optimization process. It keeps track of the parameter being varied, its current step size and direction, the value of the performance index, and the next parameter of the search process.

In the following, details of the modified Hooke and Jeeves optimization algorithm are briefly presented. A modified version of the Hooke and Jeeves algorithhn was used because it has shown faster convergence than the standard Hooke and Jeeves algorithm; it has also been shown to be much faster than the cyclic search method, which was implemented as well. The modification involves different logic and step sizes in changing the parameters.

The first step in the Hooke and Jeeves algorithm is an exploratory search. This search explores along the coordinate axes of the performance surface in the space of each parameter individually. Under the assumption that the surface is strictly quasiconvex, there is only one minimum along each of these coordinates in the parameter space. Assigned an initial value for each parameter, either from the grid search or from the dictionary, the simulation of the closed-loop adaptive control system produces a value for the performance index. The exploratory search rule-base uses this first set of parameter values as a starting point on the performance surface.

A second set of values is obtained by changing a single parameter. The change in the parameter value is initially taken to be 30%. After this parameter change, the simulation is repeated to determine the effect of the change. If the performance index decreases on the second iteration, the parameter is again changed and the simulation is rerun. If the performance index decreases three consecutive times, it is assumed that the appropriate direction has been found and the step size is increased by a factor of ten in an attempt to reach the minimum on the performance surface faster. This process continues until a parameter change causes the performance index to increase. Whenever the performance index begins to increase, the rule-base system assumes that the parameter is moving off in the wrong direction. The rule-base system resets the parameter to the last known good point on the performance surface and reverses the direction of the parameter change. If this again causes the index to increase, the step size of the parameter is decreased by a factor of ten and the search resumes. This last rule is executed each time the performance index increases in either direction.

Once the changes in a parameter become small, less than 0.1%, it is concluded that the parameter will no longer contribute to a significant decrease in the performance index, and the exploratory search for the next parameter begins. If at any time during the exploratory search the parameter changes cause the parameter to exceed a proportional maximum or minimum, the exploratory search on that parameter is stopped. This prevents the parameter from changing in magnitude too rapidly and causing the system to exhibit undesirable behavior.

The second step of the Hooke and Jeeves algorithm is a pattern search on the performance surface. Once the exploratory search for all parameters is completed, and changes in the performance index at the start and the end of the search indicate that progress on the performance surface still appears possible, the pattern search is initiated. The rules that control the pattern search are similar to those for the exploratory search. Here, however, the direction of the search is along a vector of parameter changes. When the pattern search is completed, the exploratory search will start from the last position obtained during the pattern search. This cyclic process will continue until it is decided that further parameter changes will not offer any significant performance improvement.

The results obtained by these search methods are very dependent on the initial step size, the starting points, and the stopping criteria. If the initial step size is increased, a larger portion of the performance surface is explored. However, this may lead to undesirable behavior of the closed-loop adaptive system. The effect of the stopping criterion is similar. When it is too large, the search for the minimum performance index will be cut short. Too small a stopping criterion, however, does not yield a significant decrease in the performance index. Starting points also affect the relative coverage of the performance surface. It is possible with a different starting point to find a better minimum on the performance surface, since the global minimum problem is not being addressed at this level. This is the reason for implementing the grid search procedure. While an exhaustive search of the entire performance surface is unwieldy, it is possible to get a general impression of the performance surface, and thus narrow the search space.

It should be noted that the Hooke and Jeeves method has been used for similar purposes in [8],[9] to auto-tune the control parameters of a robotic arm. The method presented here, although similar to some aspects of Chen's work, differs in the addition of the grid search procedure and the dictionary to assist in the selection of initial starting points for the optimization search procedure.

The Dictionary

The dictionary, a specific memory structure, is initially empty and is gradually built from data consisting of operating conditions and corresponding best values of controller parameters. It is assumed that the operating conditions are given to the learning system. When it has been determined that there is need for improved performance, the dictionary is used to determine a starting set of controller parameters for the optimization search procedure. Once the optimal set of parameters is found, this information, along with the corresponding operating conditions, is added to the dictionary for future use.

Implementation

The antenna parameter learning system was initially designed to run on a PC-AT

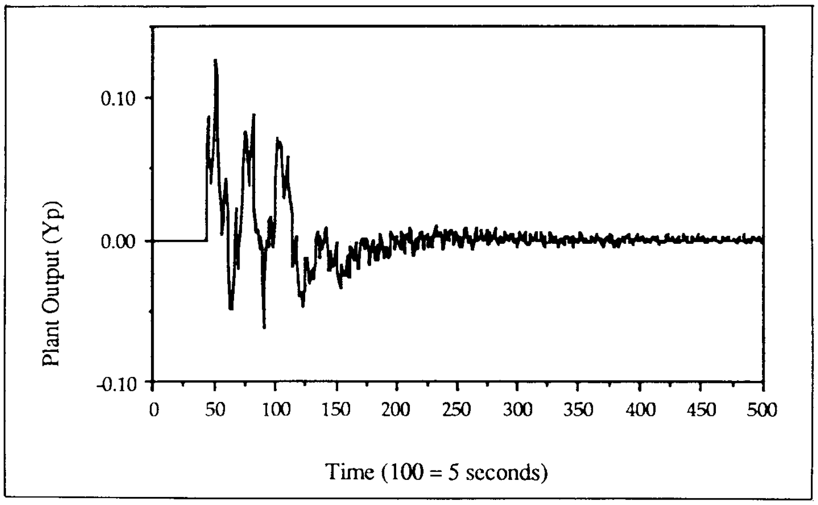

Fig. 3. Plant output of original control system with disturbance magnitude = 2.0, duration = 2 s.

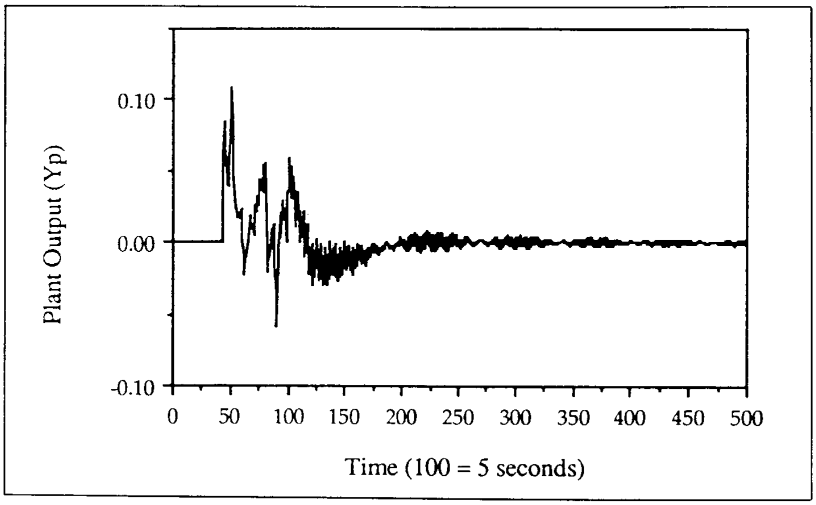

Fig. 4. Plant output after learning applied with disturbance magnitude = 2.0, duration = 2 s.

compatible computer with a math coprocessor. Both the antenna and the adaptive controller were implemented in the C programming language, and the learning system was coded using a commercial expert system shell called CxPERT. CxPERT, based on the C programming language, allows easy interaction between the rule-based system and the model of the antenna and adaptive controller.

The most time-consuming portion of the system runtime was the simulation of the plant and controller. When system runtimes had become prohibitively long on the AT compatible machine—on the order of 3 h—the learning system was recoded in C, using if–then statements, and ported to a SUN 3/50 computer. This allowed results to be obtained in several minutes. Later, the addition of the grid search procedure increased the number of required system simulations, which prompted the porting of the system once again, this time to a Convex computer. This allowed results to be obtained in minutes, instead of about an hour on the SUN computer.

Results

The controller parameters in the set

$$S = \{\sigma_1, \sigma_2, L, \bar{L}, T, \bar{T}\}$$

of (7) are to be chosen in the design process to optimize system performance when certain external inputs are present. Without loss of generality, T and $\bar{T}$ are fixed and set to 0.05 [14]. The parameters σ_1, σ_2, L, and $\bar{L}$ are then optimized with respect to the performance index (9). For the remaining discussion, S will denote the parameter values $\{\sigma_1, \sigma_2, L, \bar{L}\}$ and S_{opt} will denote the values of the parameters that optimize system performance.

It is important to stress at this point that the emphasis here is on illustrating the learning method. The particular choice for the performance index and the actual values of the parameters are of secondary importance. Other performance criteria leading to parameter values which satisfy additional constraints could have been chosen.

The simulation results of the antenna parameter learning system are presented for two experiments: the transient regulation and the command following experiments. In the simulation, the 12th-order model of the antenna described above was used for numerical simplicity. The full 164th-order model [19] could have been used, but this would have significantly increased the time required for the simulations. The actual physical antenna could also be used, with appropriate sensors, without requiring significant changes in the learning system software.

The Transient Regulation Experiment

In this experiment, the reference model of the adaptive system in Fig. 2 is set to zero throughout the simulation, i.e., it has zero input and zero initial states. With the initial parameters given in the design of the adaptive control system [15], $\sigma_1 = 0.5$, $\sigma_2 = 21.99$, $L = \bar{L} = 1.0 \times 10^4$, and introducing a pulse disturbance with a magnitude of 2.0 and duration of 2 s, the closed-loop adaptive control system was able to track the zero reference model output as expected. The plant output is presented in Fig. 3. The corresponding performance index and its components are given in Table I.

Note that in all experiments, the controller is switched on and the output is measured starting at time 2 s.

Next, the parameter learning system was used to enhance the performance of the adaptive system. The plant output of this system after learning is shown is Fig. 4. The corresponding performance index and its components are given in Table I. The optimal parameter set S_{opt} is $\sigma_1 = 0.093$, $\sigma_2 = 10.05$, $L = 49\ 000$ and $L = 119\ 703$. These values provide a 30% improvement over the original system design. The total number of control system simulations performed was 8812, which includes 8712 iterations for the grid search procedure. The grid search does add considerable computational time in finding the optimal parameter set. However, the grid search procedure increases the confidence that the parameter set found is near optimal. In addition, the grid search is run only once, if needed, when the dictionary does not contain knowledge about the performance of the system.

Table II contains the results of training, using a number of pulse disturbances of different amplitudes. The table represents the contents of the dictionary after learning. Additional disturbances, known to cause excessive output values in the original adaptive control system, were introduced to validate the usefulness of the dictionary. These disturbances are listed in Table III. The resulting optimal parameter sets S_{opt} found by the antenna parameter learning system are also presented in Table III, along with the respective performance indices. Figs. 5 and 6 show the plant output for the disturbance of magnitude 8.0, both before and after learning.

The results show that, besides learning the values of controller parameters that improve system performance, the parameter learning system is able to extend the region of operability through training. Note that training is has to be incremental, building upon previous knowledge. Since the relationship between the performance index and the controller parameter set is nonlinear, the learning process needs to take steps small enough to determine the effects of different size disturbances on the plant.

The Command Following Experiment

In this experiment, the reference model is a standard highly damped second-order transfer function $H(s)$, with $\zeta = 0.9$ and $\omega_n = 1.885$. $H(s)$ was discretized for simulation purposes, using a sampling frequency of 20 Hz. Notice that in Fig. 2 the command input equals u_m and it is added to u_p, the input to the plant.

Using $\sigma_1 = 0.5$, $\sigma_2 = 6.28$, $L = \bar{L} = 0.15$, and this time introducing a command step input applied at time 2 s, the closed loop adaptive control system was able to track the second-order reference model output. The plant output is present in Fig. 7. The corresponding performance index of the system and its components are given in Table IV.

Next, the parameter learning system was used to enhance the performance of the adaptive system. The plant output of this system after learning is shown is Fig. 8. The corresponding performance index is given in

Table I
Performance Indices for Transient Regulation Experiment

	RMS	ME	ST	J
Original control system:	0.195	0.087	19.1	4.7315
After learning:	0.015	0.058	12.15	3.3240

Table II
Training Disturbances and Parameter Sets After Learning

Disturbance				S_{opt}		Performance
Amplitude	**Duration**	σ_1	σ_2	L	$\bar{L}$	**Index (J)**
2.0	2	0.093	10.05	49000.0	119703.96	3.324
2.25	2	0.213	7.04	39200.0	131674.35	3.969
2.5	2	0.302	7.35	44046.1	145910.16	4.509
3.0	2	0.307	15.79	37325.7	183327.84	5.390
3.5	2	0.876	9.56	44046.1	175092.19	5.956
4.0	2	0.808	10.48	29114.0	203493.90	6.415
4.5	2	1.767	10.48	32025.4	203493.90	6.897
5.0	2	3.924	8.70	48450.7	140073.75	7.394
5.5	2	6.928	7.73	41567.5	138395.55	7.871
6.0	2	11.08	10.05	41567.5	138395.55	8.437
7.0	2	14.41	19.10	41567.5	110716.44	9.723

Table III
Disturbances Causing Excessive Output Values in Original System and Parameter Sets After Learning

Disturbance				S_{opt}		Performance
Amplitude	**Duration**	σ_1	σ_2	L	$\bar{L}$	**Index (J)**
8.0	2	23.06	19.1	41567.48	88573.15	10.81
9.0	2	1383.5	11.33	48450.71	77923.24	11.526

Table IV
Performance Indices for Command Following Experiment

	RMS	ME	ST	PI
Original control system:	0.142	0.255	24.95	19.26
After learning:	0.039	0.185	20.10	7.72

Table V
Training Commands and Parameter Sets After Learning

Command			S_{opt}		Performance
Amplitude	σ_1	σ_2	L	$\bar{L}$	**Index (J)**
1.00	−0.098	−15.0	0.409	0.0	7.72
1.10	−0.091	−3.0	0.383	0.0	7.89
1.15	−0.084	−3.0	0.350	0.0	8.15
1.20	−0.078	13.0	0.322	0.0	8.44
1.25	−0.072	−15.0	0.297	0.0	8.71
1.30	−0.066	−15.0	0.274	0.0	8.97
1.35	−0.062	13.0	0.254	0.001	9.27
1.40	−0.057	−3.0	0.236	0.0	9.52
1.45	−0.053	437.3	0.220	0.001	9.79

Table VI
Commands Causing Excessive Output Values in Original System and Parameter Sets After Learning

Command			S_{opt}		Performance
Amplitude	σ_1	σ_2	L	$\bar{L}$	**Index**
1.40	−0.057	−3.0	0.236	0.0	9.52
1.45	−0.053	437.3	0.220	0.001	9.79
1.50	−0.049	2536.5	0.206	0.001	10.06

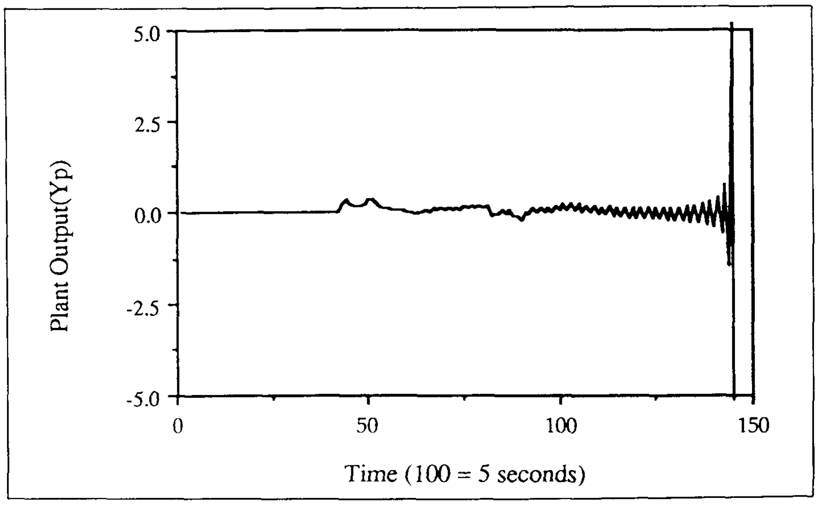

Fig. 5. Plant output of original controller with disturbance magnitude = 8.0, duration 2 s.

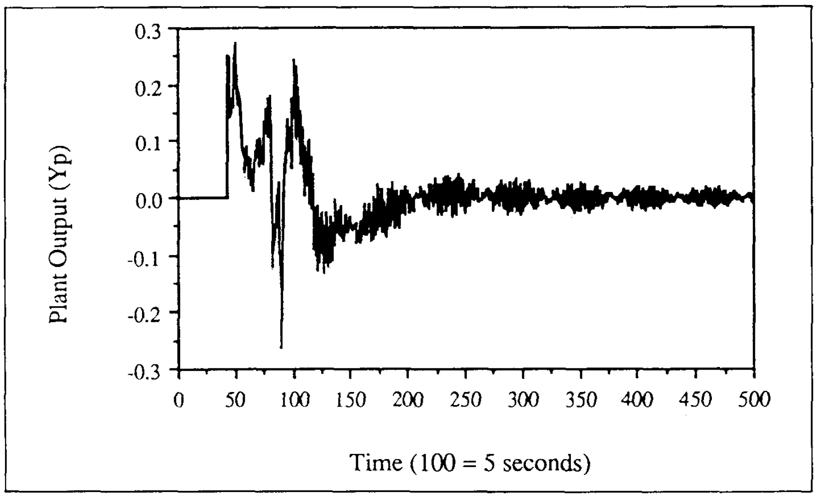

Fig. 6. Plant output with disturbance magnitude = 8.0, duration = 2 s after learning applied.

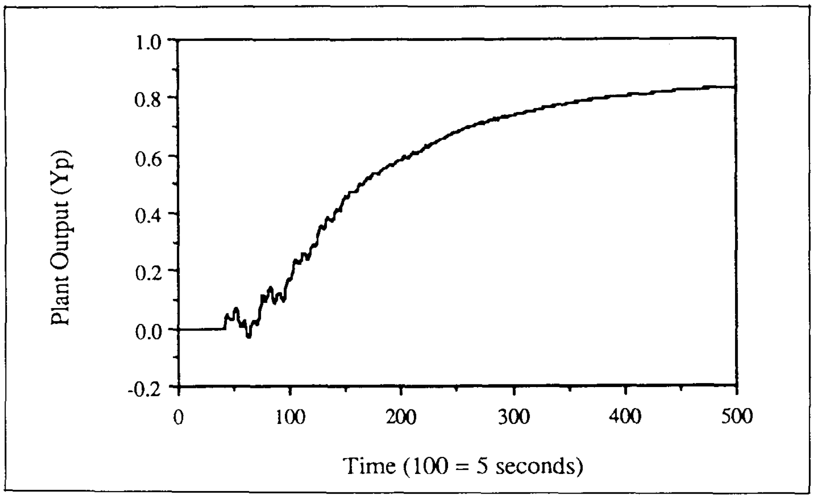

Fig. 7. Plant output of original control system with command unit step input.

Table IV. The optimal parameter set S_{opt} is $\sigma_1 = -0.098$, $\sigma_2 = -15.0$, $L = 0.409$ and $\bar{L} = 0.0$, which provides a 60% improvement over the original system design.

Table V contains the results of training, using a number of command inputs of different amplitudes. The table represents the contents of the dictionary after learning. Additional commands, known to cause excessive output values in the original adaptive control system, were introduced to validate the usefulness of the dictionary. These commands are listed in Table VI. The resulting optimal parameter sets S_{opt} found by the antenna parameter learning system are also presented in Table VI, along with the respective performance indices. Figs. 9 and 10 show the plant output for the command of magnitude 1.4, both before and after learning.

Again the results show that, besides being able to learn controller parameters that improve system performance, the parameter learning system is able to extend the region of operability through training. The second experiment also shows the flexibility of the learning system. The learning system can use different plant models, reference models, performance indices, or optimization methods, depending on its goals.

Conclusions

The learning method presented here provides performance adaptation for adaptive systems. This appears to be a novel approach to the problem. This method also deals with the question of boundedness of adaptive control systems. While analytical tools exist to determine whether a system variable will be bounded, the analysis typically does not indicate how large the bound will be. It is possible to exceed the system tolerances and yet be analytically stable. The learning method can determine this bound, and use the information in the process of controlling the system.

This method is general and it can be used in any system where performance depends on a number of adjustable parameters. As a matter of fact, the method was also successfully applied for verification purposes to determine the optimum gain in an LQR problem. However, specialized methods, when they exist, are obviously more efficient to solve specific problems. General methods, like the one presented here, are recommended for complicated problems when traditional methods fail. The method presented is also modular, since both the functional evaluation and the optimization search procedures can be modified to match the particular problem

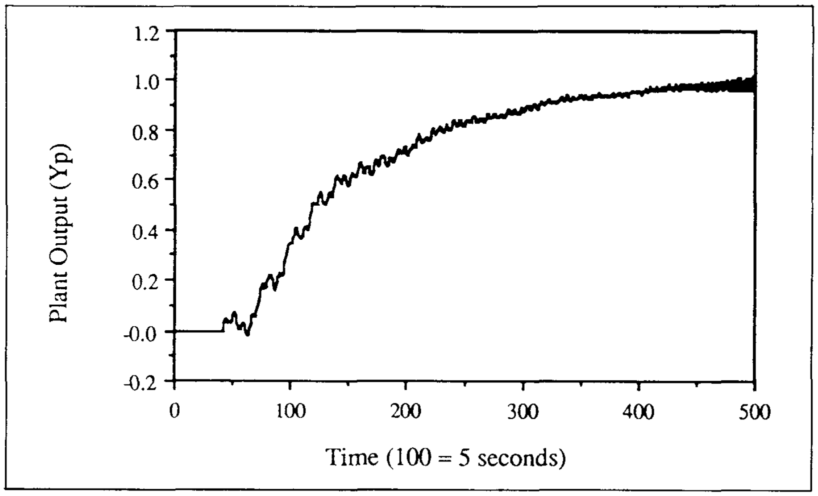

Fig. 8. Plant output with command unit step input after learning applied.

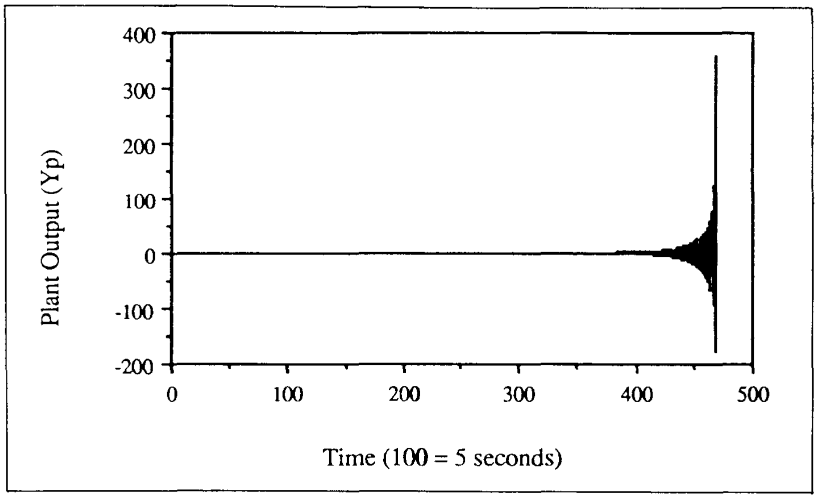

Fig. 9. Plant output of original controller with command input of magnitude 1.4.

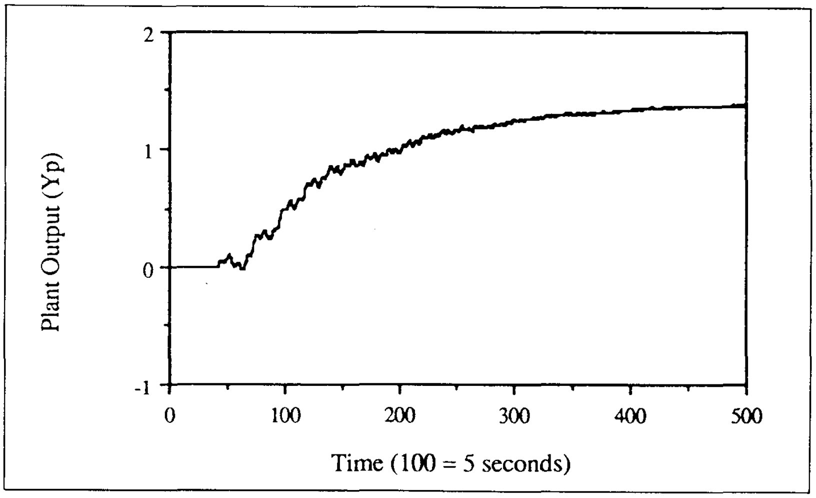

Fig. 10. Plant output with command input of magnitude 1.4 after learning applied.

at hand. Possible modifications include, for example, the gradient approach and other alternate methods. In addition, functional evaluation can be performed via computer simulation, physical experiment or mathematical calculation.

It is noted that the learning method presented here uses *a priori* information on what is known about the system. This is in contrast to many machine learning applications, where learning is usually accomplished with little *a priori* information. In engineering applications, it is recommended to pay particular attention to utilizing all available information. The more the system knows, the faster it will learn.

References

[1] C. W. Anderson, "Learning to control an inverted pendulum with connectionist networks," in *Proc. 1988 Amer. Control Conf.*, Atlanta, GA, June 15-17, 1988, pp. 2294-2298.

[2] P. J. Antsaklis, "Learning in control," in *Proc. 3rd IEEE Int. Symp. Intelligent Control*, Arlington, VA, August 24-26, 1988, pp. 500-507.

[3] P. J. Antsaklis, K. M. Passino, and S. J. Wang, "Towards intelligent autonomous control systems: Architecture and fundamental issues," *J. Intelligent Robotic Syst.*, vol. 1, pp. 315-342, 1989.

[4] D. S. Bayard, "Averaging methods for transient analysis in adaptive control of large space structures," Jet Propulsion Laboratory, Engineering Memo EM 347-88-228, Feb. 4, 1988.

[5] D. S. Bayard, "Optimal adaptive control design for large space structures," Jet Propulsion Laboratory, Engineering Memo EM 343-1107, Nov. 21,1988.

[6] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming Theory and Algorithms*. New York: Wiley, 1979.

[7] K. Chen and S. C.-Y. Lu, "A machine learning approach to the automatic synthesis of mechanistic knowledge for engineering decision-making," in *Proc. 1988 Fourth Conf. Artificial Intelligence Application*, San Diego, CA, Mar. 14-18, 1988, pp. 306-311.

[8] Y. Chen, "A formulation of automatic parameter tuning for robot arms," in *Proc. 26th Conf. Decision and Control*, Los Angeles, CA, Dec. 9-11, 1987, pp. 595-598.

[9] Y. Chen, "Automatic parameter tuning and its implementation for robot arms," in *Proc. 1988*

Conf. Robotics and Automation, Philadelphia, PA, Apr. 24-29, 1988, pp. 691-698.

[10] D. B. Eldred, "System model for JPL/AFAL experimental structure," private communication, Jan. 1988.

[11] Z. Gao, M. Peek, and P. J. Antsaklis, "Learning for the adaptive control of large flexible structures," *Proc. 3rd IEEE Int. Symp. Intelligent Control*, Arlington, VA, Aug. 24-26, 1988, pp. 508-512.

[12] D. E. Goldberg, "Computer-aided gas pipeline operation using genetic algorithms and rule learning," Ph.D. dissertation, Dept. of Civil Eng., Univ. Michigan, Ann Arbor, 1983.

[13] D. A. Handelman and R. F. Stengel, "Rule-based mechanisms of learning for intelligent adaptive flight control," in *Proc. 1988 Amer. Control Conf.*, Atlanta, GA, June 15-17, 1988.

[14] C.-H.C. Ih, D. S. Bayard, S. J. Wang, and D. B. Eldred, "Adaptive control experiment with a large flexible structure," in *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Aug. 1988, pp. 832-851.

[15] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning—An Artificial Intelligence Approach*. Palo Alto, CA: Tioga, 1983.

[16] M. D. Peek, "Machine learning of parameter values for system performance adaptation," M.S. thesis, Dept.of Electrical and Computer Eng., Univ. Notre Dame, Notre Dame, IN, Sept. 1989.

[17] M. D. Peek and P. J. Antsaklis, "Parameter learning for performance adaptation in large space structures," in *Proc. 4th IEEE Int. Symp. on Intelligent Control*, Albany, NY, Sept. 25-27,1989.

[18] K. Sobel, K. Kaufman, and L. Mabius, "Implicit adaptive control systems for a class of multi-input multi-output systems," *IEEE Trans. Aerospace Elec. Syst.*, vol. AES-18, no. 5, pp. 576-590, 1982.

[19] H. C. Vivian *et al.*, "Flexible structural control laboratory development and technical demonstration: Final report," JPL, Pub. 88-29, Oct. 1, 1987.

[20] P. J. Antsaklis, Guest Ed., "Special Issue on Neural Networks in Control," *IEEE Control Syst. Mag.*, Apr. 1990.

Mark D. Peek was born in Camden, NJ, in 1963. He received the B.S. degree in 1985 and the M.S. degree in 1989 from the University of Notre Dame. He has worked for RCA Missile and Surface Radar Division, Moorestown, NJ. He is currently a Member of the Research Staff at Tellabs Research Center, Mishawaka, IN. His research interests include machine learning, adaptive systems, and network sychronization.

Panos J. Antsaklis received the diploma of mechanical and electrical engineering from the National Tech-nical University of Athens, Greece, in 1972, and the M. S. and Ph. D. degrees in electrical engineering from Brown University, Providence, RI, in 1974 and 1977, respectively. After holding faculty positions at Brown University, Rice University, and Imperial College, University of London, he joined the University of Notre Dame where he is currently Professor in the Department of Electrical and Computer Engineering. In the summer of 1986 he was a NASA Faculty Fellow at the Jet Propulsion Laboratory, Pasadena, CA. He was a Senior Visiting Scientist at the Laboratory for Information and Decision Systems of the Massachusetts Institute of Technology during a sabbatical leave in 1987. His research interests are in multivariable system and control theory, discrete event systems, adaptive, learning, and reconfigurable control, autonomous systems, and neural networks. He has published a number of technical results in those areas. He has served as Associate Editor of the *IEEE Transactions on Automatic Control*, and is currently chairman of the Technical Committee on Theory and group leader of Control Systems in the Technical Committee on Intelligent Control of the IEEE Control Systems Society. He is also an Associate Editor of the *IEEE Transactions on Neural Networks*, and Guest Editor for Neural Networks for *IEEE Control Systems Magazine*.

Doctoral Dissertations

The information about doctoral dissertations should be typed double-spaced using the following format and sent to:

Prof. Bruce H. Krogh
Dept. of Electrical and Computer Eng.
Carnegie-Mellon University
Pittsburgh, PA 15213.

Massachusetts Institute of Technology
Papastavrou, Jason D., "Decentralized Decision Making in a Hypothesis Testing Environment."
Date: May 1990.
Supervisor: Prof. Michael Athans.
Current Address: Department of Industrial Engineering, Purdue University, West Lafayette, IN, 47907.

Rensselaer Polytechnic Institute, Troy, NY
Wang, Fei-Yue, "A Coordination Theory for Intelligent Machines."
Date: July 1990.
Supervisor: George N. Saridis.
Current Address: Dept. Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721.

Rensselaer Polytechnic Institute.
Imamura, Fumihiko, "Integrated Optimization of the Command and Control Systems for Computerized Machine Tools."
Date: May 1990.
Supervisor: Howard Kaufman.
Address: ECSE Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590.

Rensselaer Polytechnic Institute
Neat, Gregory W. "Expert Adaptive Control: Method and Medical Application."
Date: May 1990.
Supervisor: Howard Kaufman.
Address: ECSE Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590.