This article was written by Rich Sutton and published as part of a large technical report prepared by him and by Andrew Barto in 1981. It should be cited as follows:

Sutton, R.S. (1981). "Adaptation of learning rate parameters." In: *Goal Seeking Components for Adaptive Intelligence: An Initial Assessment*, by A. G. Barto and R. S. Sutton. Air Force Wright Aeronautical Laboratories Technical Report AFWAL-TR-81-1070. Wright-Patterson Air Force Base, Ohio 45433.

APPENDIX C

ADAPTATION OF LEARNING RATE PARAMETERS

C.1 Preface

The work presented in this appendix is directed toward developing an algorithm for adjusting the learning rate parameter c of each synapse individually. Consider a single synapse in one of the learning elements, such as a Widrow-Hoff element or the "classical conditioning" element discussed in Section 4. This synapse is trying to use the information in its presynaptic signal to contribute to the prediction of subsequent input. One problem is that all the other synapses will also be trying to do this. If each changed itself independently of the others so that its contribution would make up the difference or error in prediction, then the next time the situation occurred, there would probably be a huge overshoot as the hundreds of active synapses each provided enough to correct the original error. In this sort of situation each synapse must proceed cautiously, changing its weight but little to prevent overshoot, yet not so little as to make learning unnecessarily slow (undershoot).

A second and similar problem is that the signal may only provide information in a statistical sense; i.e., its presence may indicate that the input will probably be higher (or lower), but not that it definitely will be. In this case the synapse must average out the cases in which the synapse is right and wrong to arrive at a compromise measure combining both the size of the change in input predicted and the probability with which it is predicted. Again, this averaging means a slowing in the learning rate for the synapse, which must be counterbalanced against the need for speedy learning (which requires a high learning constant). How then is this learning constant to be set?

The above discussion suggests the general form of the answer: Each synapse can determine from its local measure of success in prediction - its overshoots or undershoots - whether its learning rate is too large or too small. Thus, each synapse should set its learning rate parameter as the adaptation proceeds, according to some iterative algorithm. The work presented in this appendix is the beginning of the search for, and formalization of, that algorithm.

It should be clear from the discussion of the problem facing the individual synapse that it is basically a tracking task. The synapse is trying to track the actual input with its prediction of that input by changing its prediction proportionally to the difference between predicted and actual input in those cases in which the synapse is involved, i.e., in those cases in which the synapse is presynapticaly active. In the terminology of servo-mechanism tracking, that constant of proportionality, the learning rate constant, is known as the gain. Thus, this appendix considers the problem of setting the gain of a simple tracking servo-mechanism. It is felt that the results are highly relevant to the learning rate parameter setting problem for synapses, but the work has not yet progressed to the point where it can be directly translated into this form. Further work is necessary both on the abstracted tracking problem and on mapping the results back into a learning rate parameter adaptation algorithm for a neuron-like adaptive element.

The rest of this appendix was originally a self contained paper entitled "A Method for the Automatic Selection of Gain for Discrete-Time Algorithms."

C.2 Introduction

Consider a one-dimensional, discrete-time tracking problem and its solution by a simple servomechanism (see Figure C.1). The pursuing function y(t) and the target function Y(t) are related according to the classic servomechanism equation:

$$y(t+1) = y(t) + G [Y(t)-y(t)] , \qquad (C.1)$$

where G is called the gain. In general, the target function Y(t) and the gain G will determine the quality of performance. If Y(t+1) is determined from Y(t) by the addition of a random variable chosen according to a symmetric probability distribution with an expected value of zero, then the optimal gain will be G=1.0, since then y(t) will equal Y(t-1), the best guess for Y(t). If the target function Y has inertia, the optimal gain will lie between 1.0 and 2.0, and if Y(t) is a noise corrupted version of an inertialess function z(t), then the optimal gain will lie between 0 and 1.0. In this context the problem considered in this appendix is the automatic selection of a gain parameter through experience with attempts to track a target function Y.

An adaptive tracking system should have both the

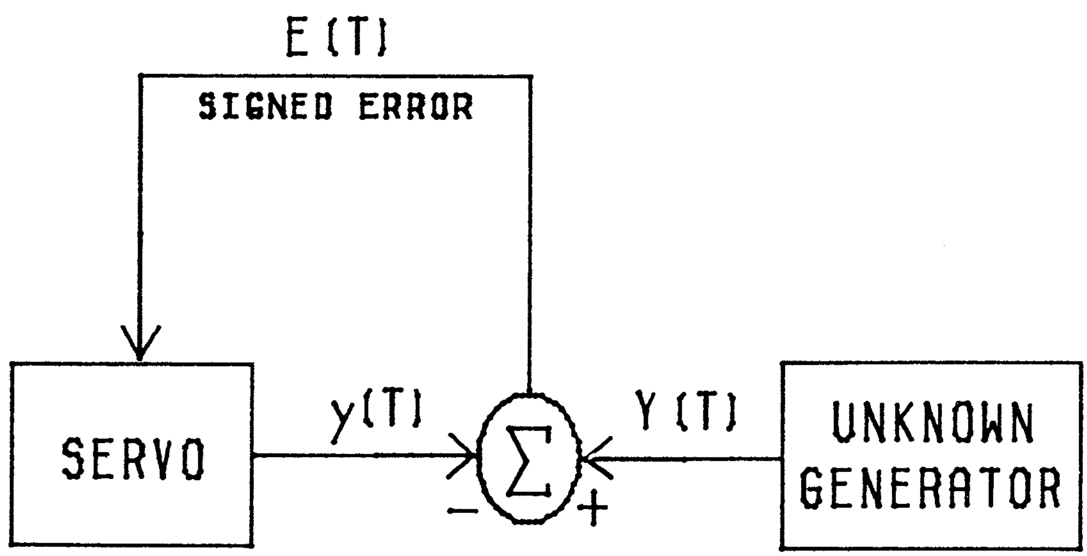

FIGURE C.1. A block diagram of a simple tracking servomechanism. y(t) is the pursuing function, Y(t) the target function, and E(t) the signed error.

property of refinement, meaning the ability to carefully zero in on the target function by averaging out noise, and the property of responsiveness, meaning the ability to stop converging and follow the target closely if it begins to move rapidly. To have both of these properties in a tracking servomechanism requires a method of adaptively modifying the gain. Previous work on this problem apparently has not found a satisfactory solution (e.g., Eisenstein, 1972).

C.3 The Gradient Descent Approach

To optimize some parameter or vector C(t) according to some evaluation function J(t) to be minimized, a straightforward approach is that of gradient descent with fixed increment:

$$C(t+1) = C(t) - a \nabla_{C(t)} J(t)$$

where a is the fixed positive increment size. Ideally, one can analytically compute an expression for the gradient to get the desired algorithm. For example, this technique can be used to derive the servo equation (Equation C.1). Here the parameter to be optimized is $y(t)$, the evaluation function $J(t)$ to be minimized is $[Y(t)-y(t)]^2$, and the positive increment is G/2:

$$y(t+1) = y(t) - G/2 \; \nabla_{y(t)} J(t)$$

$$= y(t) - G/2 \; \frac{d\,[Y(t)-y(t)]}{d\,y(t)}$$

$$= y(t) + G\,[\,Y(t)-y(t)\,]$$

Yielding the servo-mechanism equation (Equation C.1).

Now let us apply the same methodology to derive an algorithm for optimizing the gain term G which we now vary as a function of time:

$$y(t+1) = y(t) + G(t+1)\,[\,Y(t)-y(t)\,]$$

$$G(t+1) = G(t) - a \; \nabla_{G(t)} J(t)$$

$$= G(t) - a \frac{d}{dG(t)} [Y(t)-y(t)]^2$$

$$= G(t) - a \frac{d}{dG(t)} \{\, Y(t) - y(t-1)+G(t)[Y(t-1)-y(t-1)] \,\}^2$$

$$= G(t) + 2a\,[Y(t)-y(t)]\,[Y(t-1)-y(t-1)]$$

$$= G(t) + b\,E(t)\,E(t-1) \qquad \text{(C.2}$$

for $b = 2a$ and $E(t) = Y(t) - y(t)$.

The intuition behind the workings of this algorithm is fairly straightforward: If the gain is too large, there will be a tendency for the pursuing function y(t) to overshoot the target, which causes oscillation in the error, and thus via this algorithm will cause a decrease in the gain. If the gain is too small, on the other hand, then the pursuer will tend to undershoot, and successive errors will usually be of the same sign, and this algorithm will cause the gain to decrease. Previous approaches to this problem and its relatives have been based only on the signs of the succesive errors, completely ignoring the sizes of the errors (Kesten, 1958; Sardis, 1970; Perel'man, 1967). That the algorithm presented here utilizes more of the information available in the successive errors suggests that it may be an improvement over these earlier methods.

C.4 Analysis of a Special Case

For the purposes of analysis, we now consider a special case of the general problem. Assume Y(t) is a noise corrupted version of a random variable z(t), and that z(t) is varying as in a "random walk":

$$Y(t) = z(t) + B(t) \tag{C.3}$$

$$z(t+1) = z(t) + A(t), \tag{C.4}$$

for movement and noise random variables A(t) and B(t). Let us assume that the random variables A(t) and B(t) are chosen according to normal probability distributions with zero means and variances sA and sB respectively. (Since the movement of z is an inertialess random walk, for this special case the optimal gain will never be greater than 1.0.) For this case, we can prove that algorithm (C.2) converges to the gain that minimizes the expected mean square error $EXP\{[Y(t)-y(t)]^2\}$. The proof has two main steps: 1) find an expression for the optimal gain in terms of sA and sB, and 2) proves that Equation C.2 converges to that optimal gain. To find an expression for the optimal gain, first we find an expression for the expected asymptotic mean square error (MSE) in terms of sA, sB, and the gain G.

Let $e(t) = z(t) - y(t)$

Then note that the total error can be written

$$E(t) = e(t) + b(t) . \tag{C.5}$$

Now we solve for asymptotic e(t):

$$\begin{aligned} e(t+1) &= z(t+1) - y(t+1) \\ &= z(t) + A(t) - y(t) - G\ E(t) \\ &= e(t) + A(t) - G\ [e(t)+B(t)] \\ &= (1-G)e(t) + A(t) - G\ B(t) \end{aligned}$$

or

$$e(t) = (1-G)^{t}\ e(0) + \sum_{n=0}^{t-1} (1-G)^{n} \ [\ A(t-1-n) - G\ B(t-1-n)\ ]$$

Let $e(\infty)$ denote the limit of this expression as t goes to infinity. Since $e(\infty)$ is a sum of independent identically normally distributed random variables, it will also be normally distributed, will have mean zero, and will have variance the sum of the variances of the summands:

$$s^2_{e(\infty)} = \lim_{t\to\infty} \sum_{n=0}^{t} s^2 \{(1-G)^n\ [\ A(t-n)-GB(t-n)]\}$$

$$= \lim_{t\to\infty} \sum_{n=0}^{t} [(1-G)^2]^n\ [sA^2 + G\ sB^2]$$

where s^2_X denotes the variance of the random variable X.

This geometric series is convergent for $0 \leq G \leq 2.0$:

$$s^2_{e(\infty)} = \frac{sA^2 + G^2\ sE^2}{1 - (1-G)^2}$$

By (C.5), and since $e(\infty)$ is normally distributed with mean zero, $E(\infty)$ is also normally distributed with mean zero and of variance

$$s^2_{E(\infty)} = \frac{sA^2 + G^2\ sB^2}{1 - (1-G)} + sB^2 \qquad (C.6)$$

Which is just the desired equation for the mean square error in terms of sA, sB, and G. The value of G which minimizes this MSE can be found by the straightforward but tedious process of differentiating Equation C.6 with respect to G and setting it to zero. After simplification and solving a quadratic, a single positive root is found:

$$G_{opt} = \frac{-sA^2 + \sqrt{sA^4 + 4\ sA^2\ sB^2}}{2\ sB^2} \qquad (C.7)$$

For the second part of the proof we must show that Equation C.2 converges to the optimal gain (C.7). From (C.2) and (C.6):

$$G(t+1) = G(t) + b\ E(t)\ E(t-1) \qquad (C.8)$$

We will assume that if the constant b is chosen properly, G(t) will (nearly) converge to the fixedpoint of (C.8), and only prove that that fixedpoint is (C.7). (Note: to really complete the convergence proof it is necessary to let the increment b become an decreasing sequence and prove a contraction property on the expected change in G(t) as t goes to infinity.) At the fixedpoint G of (C.8)

$$
\begin{aligned}
0 &= \mathrm{EXP}\{\ E(t+1)\ E(t)\ \} \\
&= \mathrm{EXP}\{\ [\ \epsilon(t+1) + B(t+1)\ ]\ E(t)\ \} \\
&= \mathrm{EXP}\{\ [\ (1-G)E(t) - B(t) + A(t) + B(t+1)\ ]\ E(t)\ \} \\
&= \mathrm{EXP}\{\ (1-G)E(t)^2 - E(t)B(t) + E(t)A(t) + E(t)B(t+1)\ \} \\
&= (1-G)\mathrm{EXP}\{E(t)^2\} - \mathrm{EXP}\{\ [e(t)+B(t)]\ B(t)\ \} \\
&= (1-G)\mathrm{EXP}\{\ E(t)^2\ \} - \mathrm{EXP}\{\ B(t)^2\ \}
\end{aligned}
$$

$$
= (1-G)\ s^2_{E(\infty)} - sB^2
$$

Substituting in with (C.8), and simplifying yields

$$
0 = sB^2\ G^2 + sA^2\ G - sA^2,
$$

whose only positive root is the same as (C.5), the expression for the optimal gain.

C.5 Computer Simulation

The algorithm (Equations C.1, C.2, C.3, C.4) was programmed on a digital computer, with the distributions of the random variables approximated by pseudo-random number generating programs. Figure C.2 reports the results of an experiment in which the observation noise standard deviation sB was step changed from sB=0.3 to sB=2.0 and back again. Figure C.2a shows the optimal gain compared to the actual gain, both versus time. This figure demonstrates that the gain adaptation algorithm can both increase and decrease the gain, whichever is appropriate. Figure C.2b shows the analytic asymptotic error for the actual and optimal gains plotted versus time. This figure illustrates that the algorithm can keep the error of the tracking system at very nearly the optimal theoretical limit despite occasional or slow changes in the unknown system and thus in the optimal gain.

C.6 Further Levels of Adaptation

One nice aspect of the algorithm presented here is that only one parameter, the gain increment parameter b, need be

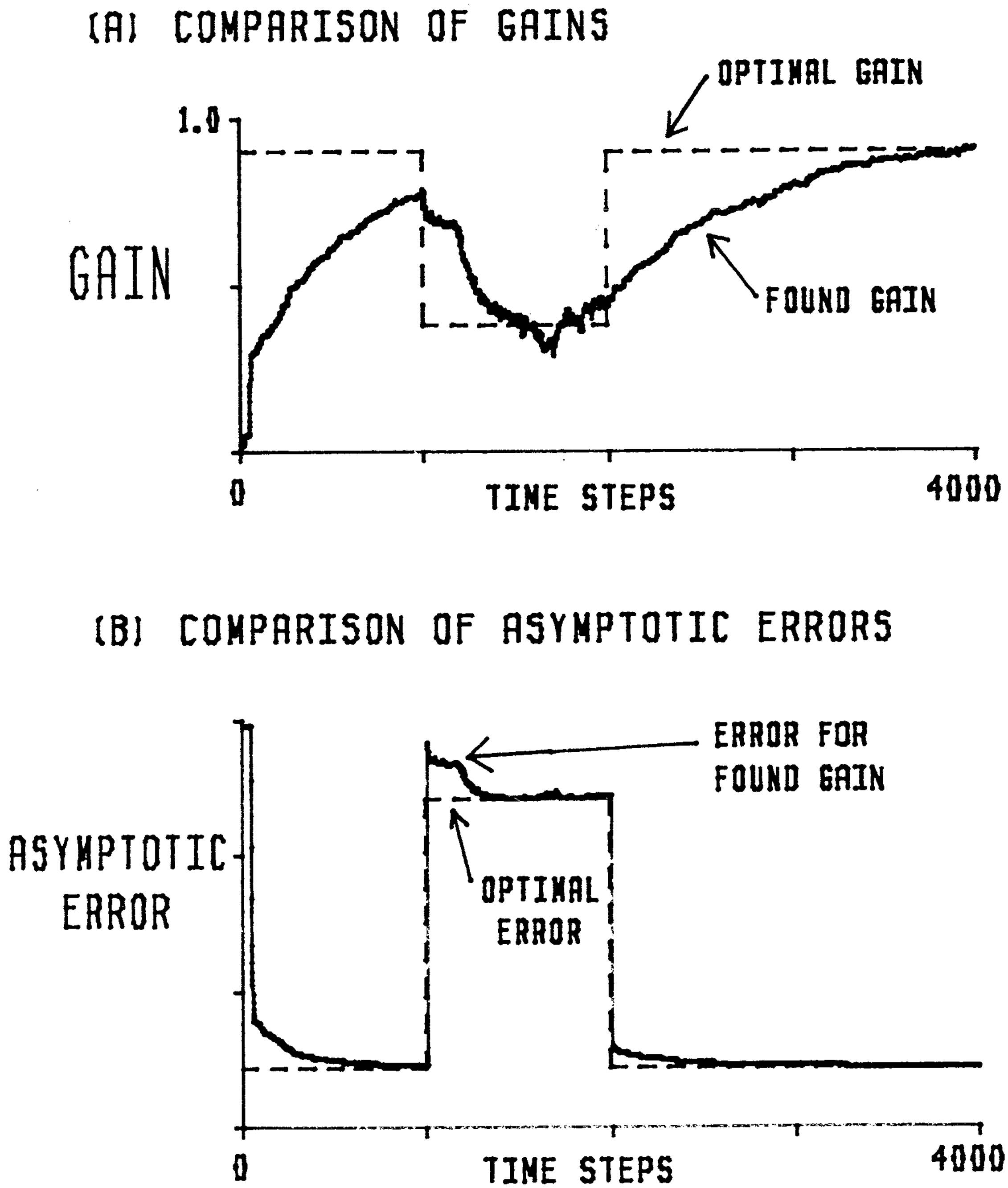
(A) COMPARISON OF GAINS
OPTIMAL GAIN
1.0
GAIN
FOUND GAIN
0
TIME STEPS
4000
(B) COMPARISON OF ASYMPTOTIC ERRORS
ERROR FOR FOUND GAIN
ASYMPTOTIC ERROR
OPTIMAL ERROR
0
TIME STEPS
4000

FIGURE C.2

FIGURE C.2. Computer simulation of the single level adaptive gain selection algorithm. In this experiment the observation noise standard deviation parameter was step changed from sB=0.3 to sB=2.0 and then back again while the random movement standard deviation parameter remained constant at 1.0. These changes were made at the 1000th and 2000th time steps respectively.

FIGURE C.2a compares the analytic optimal gain (dashed line) with that found by the single-level gain adaptation algorithm (solid line). Note that the algorithm can both increase and decrease the gain.

FIGURE C.2b compares the analytic asymptotic error (MSE) levels under the optimal (dashed line) and actual gains (solid line). The changes in actual gain keep the error nearly at the theoretical minimum despite the changes in the observation noise. The gain change rate parameter was b=0.001 in this experiment.

chosen arbitrarily by the designer or user of the technique to fit the characteristics of the particular application. This is in sharp contrast to the methods of Perel'man (1967) and Kesten (1958), whose performance is dependent on a series of possible gain parameters that need to be specified by the user. In the algorithm here, even the dependence on the b parameter can be reduced, i.e., can be made automatically adaptive to the environment, by extending the scheme to additional levels of adaptation. Applying the same methodology we used twice above, we let b become a function of time and change it in proportion to the gradient of the evaluation function J(t) with respect to b(t):

$$b(t+1) = b(t) - a \nabla_{b(t)} J(t)$$

Solving this analyticaly results in an algorithm for the optimal rate of change of gain parameter. This algorithm will in turn have a rate paramter, and an optimizing algorithm can be derived for that. The result is an arbitrarily deep hierarchy of rate of change or gain algorithms. A pattern in these algorithms quickly becomes apparent. We change notation slightly at this point to allow a statement of the multiple-level adaptive gain selection scheme which makes this pattern more apparent. For a gain selection algorithm with n levels of adaptation:

$$y(t+1) = y(t) + G_1(t+1)\, E_0(t)$$

where $E_0(t) = Y(t) - y(t)$

$$G_i(t+1) = G_i(t) + G_{i+1}(t+1)\, E_i(t) \qquad i=1,\ldots,n-1$$

where $E_i(t) = E_{i-1}(t)\, E_{i-1}(t-1) \qquad i=1,\ldots,n-1$

and $G_n(t+1) = G_n$, a small positive constant for the last level of adaptation.

This multiple-level algorithm was also programmed on a digital computer for the special case of normally distributed movement and noise random variables. An experiment was run comparing the previous two level system (the first level of adaptation was just the simple servo itself) to a three level system for a case in which the optimal gain remained constant. We see from Figure C.3a that while the two level system found the optimal gain very quickly, there was no tendency for the gain to converge to that value. The three level system, on the other hand, was able to detect that the optimal gain itself was not changing, and reduced the rate at which it changed the gain, resulting in the convergence of the gain to its optimal value (Figure C.3b). However, simulation results also revealed that the multiple-level algorithm can become

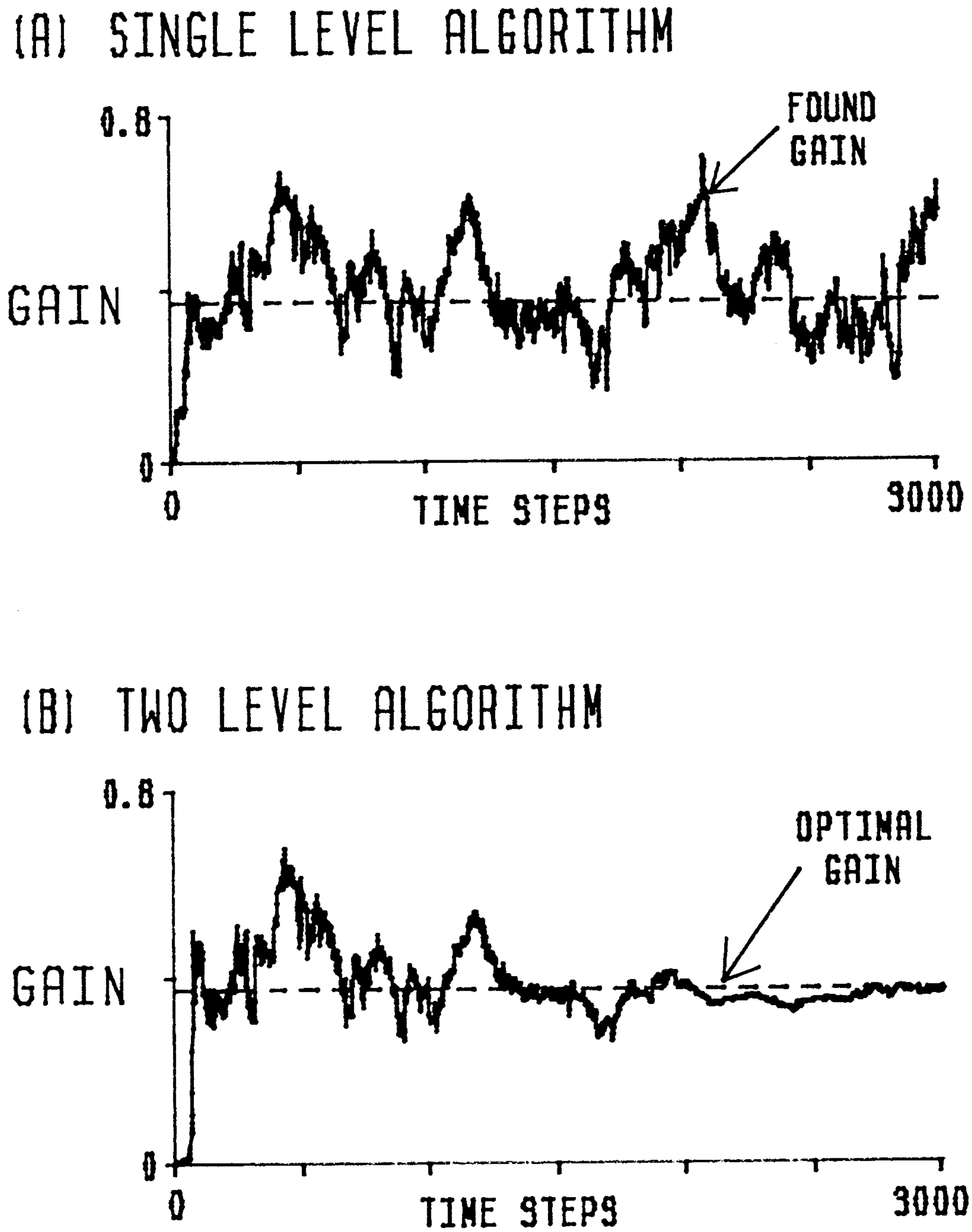
(A) SINGLE LEVEL ALGORITHM
0.8
FOUND
GAIN
GAIN
0
0
TIME STEPS
3000
(B) TWO LEVEL ALGORITHM
0.8
OPTIMAL
GAIN
GAIN
0
0
TIME STEPS
3000

FIGURE C.3

FIGURE C.3. Comparison of a single level gain adaptation algorithm with a two level algorithm for the case of a constant optimal gain. While the single level system (Figure C.3a) finds the optimal gain (dashed line), there is no tendency for the gain to converge to it. The two level gain adaptation algorithm (Figure C.3b), on the other hand, can adjust the rate with which it varies its gain, and does converge to the optimal gain. In this experiment sA=1.0, sB=3.0, and the rate or gain constants for the last levels of adaptation were gain(2)=0.001 for Figure C.3a and gain(C.3)=5.0e-8 for Figure C.3b.

unstable for some unknown systems and for some settings of the rate or gain constants for the last level of adaptation. This is probably due to the fact that the gradient descent analysis technique is due to a linear approximation of the gradient of the evaluation function J(t). If the increment is small, this approximation is a good one, but if the increment is large, it can be a very poor approximation to the actual gradient. In the multiple-level algorithm this increment is under adaptive control, and thus there is no guarantee that the increment will remain sufficiently small, and instability can result. Further work is needed to solve this problem with the otherwise promising multiple-level algorithm.

C.7 Conclusions

The multiple-level gain selection algorithm presented here seems to be applicable to any case of discrete-time adaptation involving a signed error and an associated gain or rate parameter. This algorithm is able to both increase and decrease gain in response to changes in the target function's behavior, utilizes all the information in the error signal, and is extremely simple. Comparisons are dificult to make between disimilar algorithms, but the above properties suggest that this algorithm may be a significant improvement over other gain or rate parameter selection algorithms in the literature.

Finally, a multiple-level version of this algorithm was presented. Its particular advantages will be most important in systems which must handle with high performance a wide range of uncertain environments. Although the approach seems promising, further work is necessary on the multiple-level algorithm.

BIBLIOGRAPHY TO APPENDIX C
ADAPTATION OF LEARNING RATE PARAMETERS

Kesten, H. : Accelerated stochastic approximation. Ann. Math. Statist. 29, 41-59 (1958)

Sardis, G. S. : Learning applied to successive approximation algorithms. IEEE Trans. Syst. Sci. Cybern. SMC-6, 2, 97-103 (1970)

Perel'man, I. I. : Method of self-adjustment of step search systems. Translated from Avtomatika i Telemekhanika 4, 80-93 (April, 1967)

Eisenstein, B. A. : A self-learning estimator for tracking. IEEE Trans. Syst., Man, Cybern. SMC-2, 2, 281-284 (1972)