Key words: *adaptive algorithms, model validation, physiological systems, predictors, sensitivity analysis, smooth pursuit eye movements, time-delay systems*

Development and Sensitivity Analysis of Adaptive Predictor for Human Eye Movement Model

DEBORAH R. HARVEY* AND A. TERRY BAHILL

Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721

ABSTRACT

The human can overcome a time delay and track visual targets with zero latency. This is nicely demonstrated by the smooth pursuit eye movement system. We found that if our model was to emulate the human, it had to predict target velocity and compensate for system dynamics. The model accomplished this using a least means-square prediction algorithm. To help validate the model, a sensitivity analysis and a parameter estimation study were performed.

1. INTRODUCTION

The control of movement has long been an enigma for scientists as well as for parents who marvel at the miracle of seeing their children take their first steps. The control of muscles that we often take for granted is so complex that it is difficult to comprehend the intricacies involved. To develop an understanding of such complex movement control systems, we started with a study of a simple neuro-muscular system, developed physiologically realistic models, and then refined these models.

The eye movement system is a good starting point because of its simplicity, relative to other neuro-muscular systems. This system has primarily two degrees of freedom, horizontal and vertical, and only two muscles are involved in horizontal eye movements, as compared with six or more degrees of freedom and about thirty major muscles for each leg involved in locomotion. The study of the eye movement system is also aided by the ease with which the movements can be measured. Any knowledge gained in the control of eye movements will contribute not only to the understanding of the oculomotor system but also to the understanding of larger, more complex neuro-muscular systems.

*Present address: Mitre Corp., MS W454, 1820 Dolley Madison Blvd., McLean, VA 22102.

TRANSACTIONS of The Society for Computer Simulation
ISSN 0740-6797/85$2.00 + .10

Volume 2, Number 4, pp. 275-292

Figure 1 shows our model for the human eye movement systems. Like the human, this model can overcome the time delay and track a target without latency. To do this, the model must be able to predict future target velocity, and compensate for system dynamics. In this paper a least means-square technique for predicting target velocity is described. After incorporating this prediction algorithm into the model, the model was studied to learn more about the model, and hopefully about the human. In particular, we performed a sensitivity analysis of the predictor and then investigated how parameter variations effected the mean-square error between the predicted output and the actual target waveform.

2. EYE MOVEMENT SYSTEMS

The purpose of the eye movement system is to keep the fovea, the region of the retina with the greatest visual acuity, on the object of interest. To accomplish this task, the following four types of eye movements work in harmony: *saccadic* eye movements, which are used in reading text or scanning a roomful of people; *vestibulo-ocular* eye movements, used to maintain fixation during head movements; *vergence* eye movements, used when looking between near and far objects; and *smooth pursuit* eye movements, used when tracking a moving object. These four types of eye movements have four independent control systems involving different areas of the brain. Their dynamic properties, such as latency, speed, and high-frequency cutoff values, are different, and they are affected differently by fatigue, drugs, and disease (Bahill 1981, Bahill and Stark 1975, Leigh and Zee 1983). For simplicity, none of the other neural systems associated with vision or movement will be discussed in this paper.

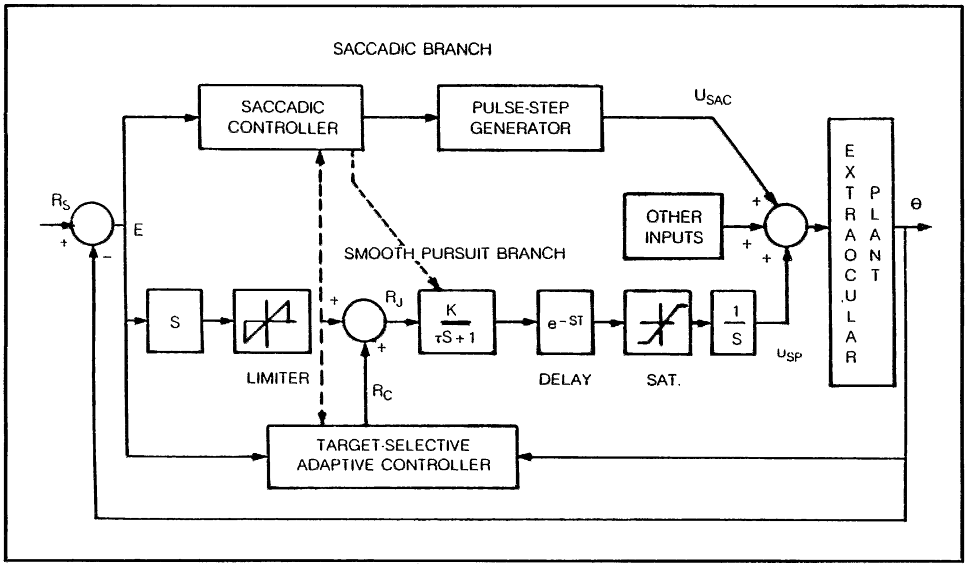

Figure 1. The Target Selective Adaptive Control Model (TSAC) of Bahill and McDonald (1983) has three branches: smooth pursuit, saccadic, and the adaptive predictor.

The specific actions of these four systems can be illustrated by the example of a duck hunter sitting in a rowboat on a lake. He scans the sky using saccadic eye movements, jerking his eyes quickly from one fixation point to the next. When he sees a duck, he tracks it using smooth pursuit eye movements. If the duck comes close to him, he moves his eyes toward each other with vergence eye movements. Throughout all this, he uses vestibulo-ocular eye movements to compensate for the movement of his head caused by the rocking of the boat. Thus, all four systems are continually used to move the eyes.

3. THE TSAC MODEL

This paper primarily examines the smooth pursuit eye movement system. The earliest model for the smooth pursuit system is the sampled data model developed by Young and Stark (1963). As a result of later evidence presented by Robinson (1965) and Brodkey and Stark (1968), the pursuit branch is no longer viewed as a sampled data system, but rather as a continuous one.

There is one physically realizable model capable of overcoming the time delay in the smooth pursuit branch, the Target Selective Adaptive Control (TSAC) model developed by Bahill and McDonald (1983) and McDonald and Bahill (1983). This model with the saccadic and smooth pursuit branches and their interactions is shown in Fig. 1.

Referring to Fig. 1, the input to the smooth pursuit branch is retinal error, which is converted to velocity by the differentiator. The limiter prevents any velocities greater than 70 degrees per second from going through this branch. (The numbers given in this section are only typical values, the standard deviations are large, *e.g.*, Bahill and LaRitz (1984) showed smooth pursuit velocities of over 150 deg/sec for a baseball player.) The leaky integrator $K/(\tau s + 1)$ is suggested from experimental results showing that humans can track ramps with zero steady-state error (Rashbass 1961), and open-loop experiments that showed a slope of -20 decibels per decade for the pursuit branch's frequency response (Bahill and Harvey 1986). The gain, K, for the pursuit branch must be greater than unity, since the closed-loop gain is almost unity. Currently used values for the gain are between two and four (Bahill and Harvey 1986, Young 1971). The e^{-sT} term represents the time delay, or latency, between the start of the target movement and the beginning of pursuit movement by the subject. A time delay of 150 msec is currently accepted (Bahill and Harvey 1986). The saturation element prevents the output of any velocities greater than 60 degrees per second, the maximum velocity produced by the human smooth pursuit system.

The model must be able to overcome the 150 msec time delay and track with zero latency. Because the smooth pursuit system is a closed loop system, the model's time delay appears in the numerator and the denominator of the closed-loop transfer function,

$$\frac{\dot{\theta}_E}{\dot{\theta}_T} = \frac{Ke^{-sT}}{\tau s + 1 + Ke^{-sT}}$$

An adaptive predictor using a Least Means-Square adaptive filter was designed to overcome the time delay in the numerator. Compensation for the model's dynamics overcomes the time delay in the denominator.

4. THE LEAST MEANS-SQUARE ADAPTIVE FILTER

The Least Means-Square (LMS) adaptive filter, popularized by Widrow 1971, Widrow *et al.* 1976, Widrow and Stearns 1985, is a "self-designing" filter composed of a tapped delay-line, variable weights, a summing junction to add the weighted signals, and machinery to adjust the weights. Two processes occur in the adaptive filter: the adaptation process and the operation process.

The *adaptation process* handles weight adjustment. The values of the weights are determined by estimating the statistical characteristics of the input and output signals. The heart of the adaptation process is the weight adjustment algorithm. As each new input sample is received the weights are updated by the algorithm,

$$\mathbf{W}(j+1) = \mathbf{W}(j) - k_s \nabla[E^2(j)]$$

with

$\mathbf{W}(j+1)$ = the weight vector after adaptation
$\mathbf{W}(j)$ = the weight vector before adaptation
k_s = the proportionality constant controlling stability and the rate of convergence
$E(j)$ = the difference between the desired response and the filter's output, the error
$\mathbf{X}(j)$ = the vector of input signals.
$\nabla[E^2(j)]$ = the gradient of the error squared with respect to $\mathbf{W}(j)$

In order to find the best possible weights, we computed the gradient (with respect to $\mathbf{W}$) of the squared error, set this equal to zero, and solved for the optimum weights. The result is the Weiner-Hopf equation.

$$\mathbf{W}_{\text{LMS}} = \phi^{-1}(x,x)\phi(x,d)$$

where

$\mathbf{W}_{\text{LMS}}$ = the vector of weights that would give the least mean-square error
$\phi(x,x)$ = autocorrelation matrix of the input signals
$\phi(x,d)$ = covariance matrix between the input signal and the desired output signal.

To solve the Wiener-Hopf equation it is necessary to compute the correlation matrices. However this would require a lot of computer time; furthermore these matrices cannot be computed in advance, because this would require *a priori* knowledge of the statistics of the input signal.

Because it is difficult to compute the true gradient, we use an estimate of the gradient, which is equal to $-2E(j)\mathbf{X}(j)$. Our algorithm is a form of the method of steepest descent using estimated gradients instead of measured gradients. Using this estimated gradient, the adjustment algorithm can be written,

$$\mathbf{W}(j+1) = \mathbf{W}(j) + 2k_s\, E(j)\mathbf{X}(j)$$

Figure 2 illustrates the implementation of this weight adjustment algorithm. If the input signals are uncorrelated, then the expected value of the estimated gradient converges to the true gradient without any knowledge of the input signal's statistics.

During the *operation process* of the LMS filter, illustrated in Fig. 3, the tapped delay-line input signals are weighted, using the gains from the adaptation process and summed to form the output signal. The difference between the desired output signal and the actual output of the filter is the error that is fed back to the weight adjustment algorithm.

The speed and accuracy of the filter while converging to the optimal solution depends on several factors. Because noise is introduced into the weight vector from the gradient estimates, it follows that if the filter is allowed to converge slowly, less noise will be introduced during each adaptation cycle and the convergence will be smoother. Regardless of the speed with which the filter converges, some noise will be introduced. This noise prevents the filter from converging to the minimum mean-square error. The ratio of the excess mean-square error to the minimum mean-square error gives a measure of the misadjustment of the filter compared to the optimum system. The misadjustment depends on the time constant of the filter's weights, where the time constant is equal to the time it takes for the weights to fall within 2 percent of their converged value. A good approximate formula for the misadjustment, M, is

$$M = \frac{n}{4\tau_{mse}} \qquad (1)$$

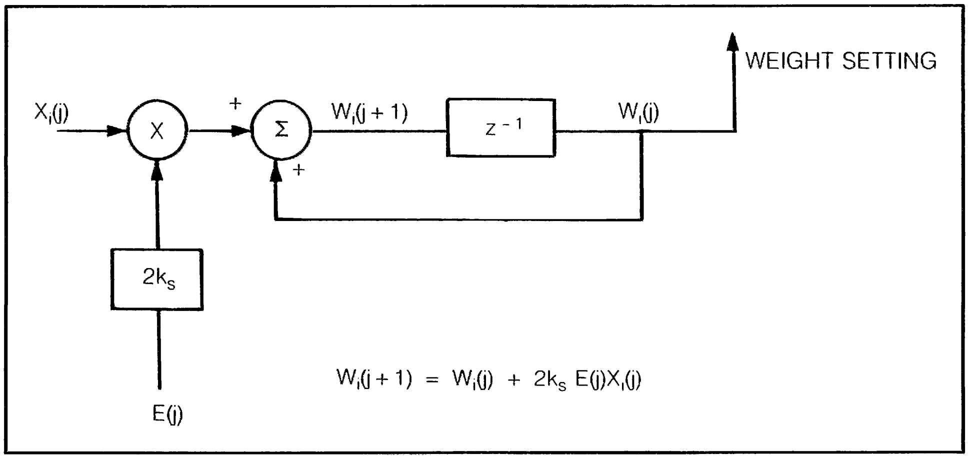

Figure 2. Implementation of the weight adjustment algorithm.

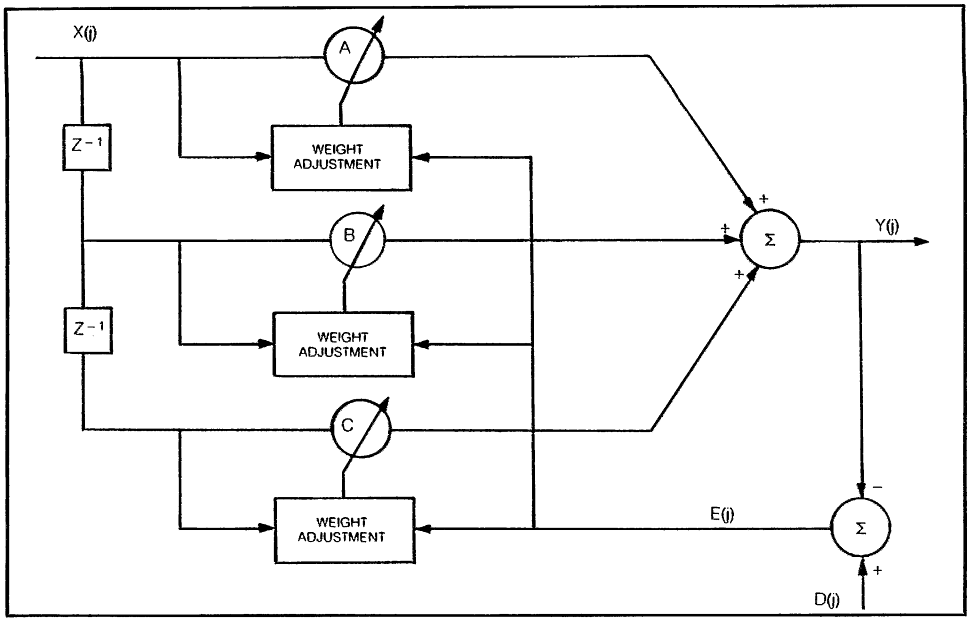

Figure 3. The Least Means-Square (LMS) adaptive filter. The boxes labeled weight adjustment contain systems like Fig. 2.

This algorithm shows that M is proportional to the number of weights, n, and inversely proportional to the time constant, τ_{mse}.

To insure convergence the proportionality constant, k_s, in the weight adjustment algorithm must be within the following bounds

$$0<k_s<\frac{1}{\Sigma_{j=1}^{n} E[X_j^2]}$$

where $E[X_j^2]$ is the expected value of the square of the jth input. For slow and precise convergence k_s should be within the following more restrictive bounds

$$0<k_s\ll\frac{1}{\Sigma_{j=1}^{n} E[X_n^2]}$$

According to Widrow (1971) and Widrow *et al.* (1976), for a filter using tapped delay-line input signals, the time constant is related to the proportionality constant by

$$\tau_{mse} = \frac{1}{4k_s\, E[x^2]} \qquad (2)$$

In summary, an adaptive filter is made up of a tapped delay-line, variable weights, a summing junction, and the weight adjustment algorithm. The filter adjusts its own internal settings to converge to the optimal solution. Due to noise from the gradient estimate, the accuracy and speed of convergence depends on the number of weights and the proportionality constant, k_s.

5. THE ADAPTIVE PREDICTOR

The adaptive predictor is an application of the LMS adaptive filter. We used this predictor to overcome the 150 msec time delay in the smooth pursuit model.

Figure 4 shows the design of the adaptive predictor. Two filters are used: an adaptive filter and a slave filter. The adaptive filter determines the appropriate weights. It does this by predicting the value of the input signal 150 msec into the future, $\hat{D}(j + T)$. To accomplish this, the input signal, $D(j)$, is delayed by an amount of time equal to the time to be predicted, in this case 150 msec. This delayed signal, $X(j)$, then serves as the input to the adaptive filter. The filter's weights converge to values that give an output signal, $Y(j)$ or $\hat{D}(j)$, which ideally matches the undelayed input signal.

The slave filter is responsible for predicting. The input to the slave filter is the undelayed signal, $D(j)$. The slave filter is organized like the adaptive filter except there is no automatic adaptation process, *i.e.*, no weight adjustment. The weights from the adaptive filter are copied into the slave filter after each adaptation cycle. The output of the slave filter, $\hat{D}(j + T)$, is the predicted value of the input signal at the desired future time.

For the TSAC model, the velocity of the target must be predicted 150 msec into the future to overcome the smooth pursuit system's time delay. Therefore, the target's velocity, the input signal to the predictor, was delayed by 150 msec and used as the adaptive filter's input. Our adaptive filter used between 15 and 150

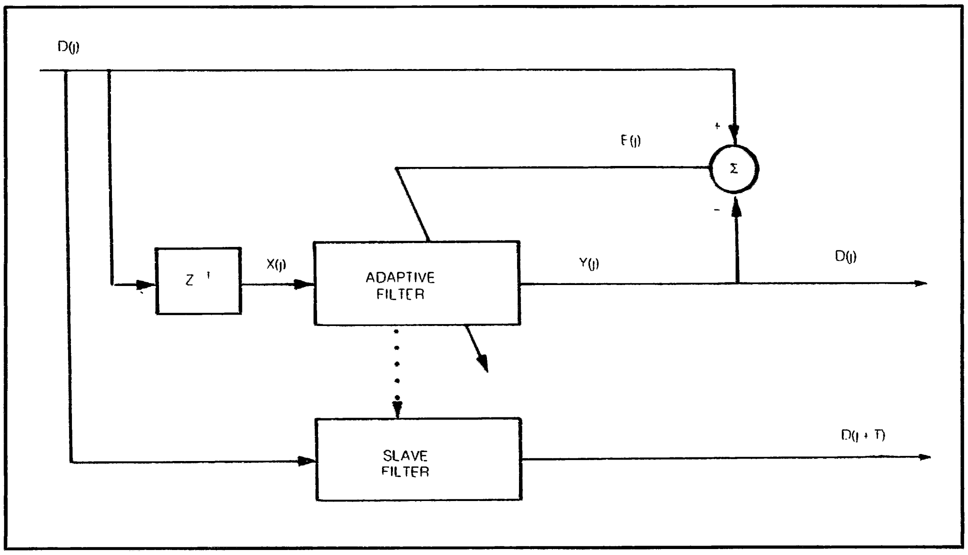

Figure 4. The Adaptive Predictor. The boxes labeled adaptive filter and slave filter contain systems similar to Fig. 3.

weights and a proportionality constant of 0.00001. Figures 5 and 6 show the behavior of the predictor with 150 weights. Figure 5 shows the output of the predictor, $\hat{D}(j + T)$, for various waveforms superimposed on the signal to be predicted. The filter's transients die out within 2.5 seconds of each abrupt change in velocity.

Figure 6 shows the predictor's mean-square error, $\|E(j)\|^2$, plotted against the number of iterations of the filter; an iteration was completed every 5 msec. After

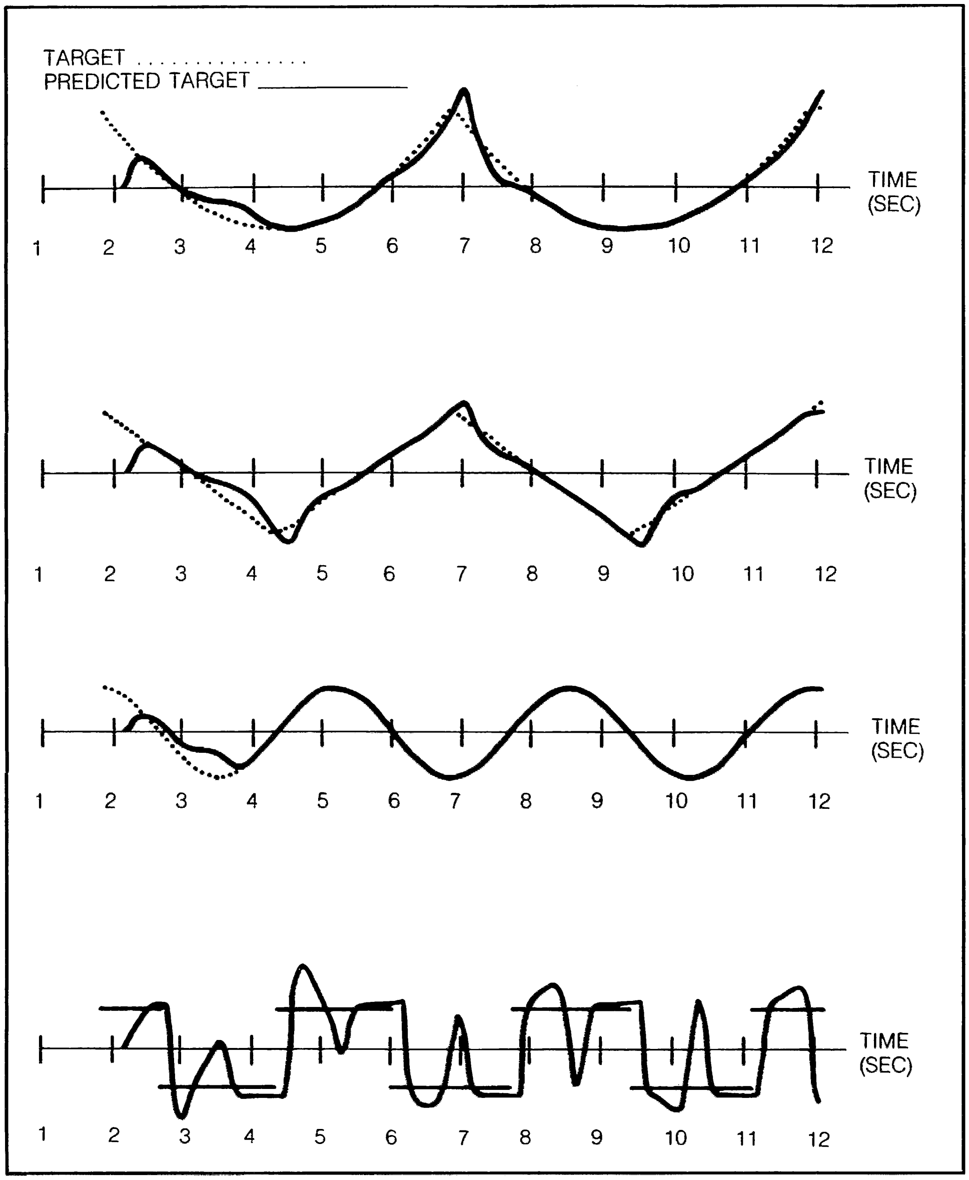

Figure 5. The predictor's output superimposed on the signal it is predicting. The target velocity waveforms are from top to bottom: parabolic, triangular, sinusoidal, and square.

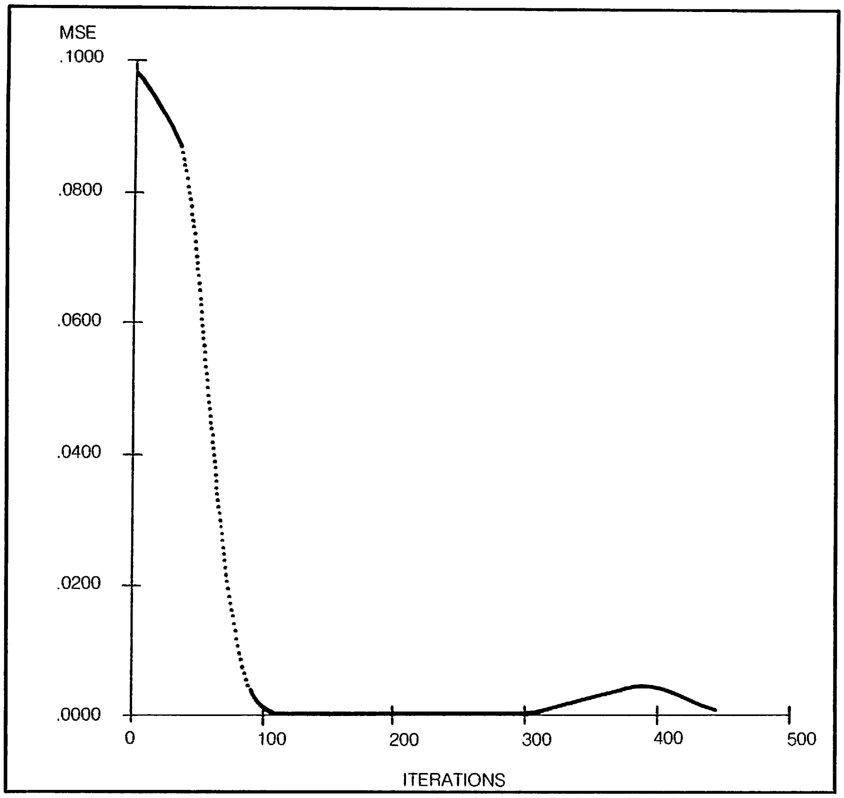

Figure 6: The learning curve for the adaptive predictor.

450 iterations the mean-square error was effectively zero; this corresponds to 2.25 seconds, which agrees with the predictor's output in Figure 5. The settling time of the filter, 450 iterations, is approximately $4\tau_{mse}$, where τ_{mse} is the average time constant for the weights. This gives a τ_{mse} of 112.5 iterations. Using Eq. (5), $M = n/4\tau_{mse}$, the predictor has a misadjustment of approximately 33.3 percent.

6. COMPENSATING FOR PLANT DYNAMICS

The predicted target velocity from the adaptive predictor compensates for the effects of the time delay in the numerator of the transfer function of Eq. (1). To overcome the effects of the time delay in the denominator, compensation for the model's dynamics must be done. This means that the brain must have a model for itself and the rest of the physiological system, and that it uses this model to generate the required compensation signal.

When linear state-variable feedback notation is used for a system its closed-loop transfer function is

$$\frac{Y(s)}{R_t(s)} = \frac{\mathbf{h}'(s\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{b}Ke^{-sT}}{1+\mathbf{k}'(s\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{b}Ke^{-sT}} \tag{3}$$

where

Y = system output, θ in Fig. 1
R_i = system input
T = time delay
$\mathbf{A}$ = system matrix
$\mathbf{b}$ = input coefficient vector
$'$ = vector transpose operation
$\mathbf{k}'$ = transposed control vector
$\mathbf{h}'$ = transposed output coefficient vector
K = the gain.

The general method of compensating for model dynamics is complex. It involves computing an adaptive signal R_a, which, when added to the target position R_s, produces a system input R_i that will produce zero-latency tracking. This method is discussed in detail by McDonald and Bahill (1983). We will now briefly show how we used it.

For the human eye movement system the order of the system and the control and output vectors are one so that the following values are appropriate.

$$A = -\frac{1}{\tau}$$

$$b = \frac{1}{\tau}$$

$$h = 1$$

$$k = 1.$$

The system's input, $r_i(t)$, is the sum of the target reference signal, $r_s(t)$, and the adaptive signal, $r_a(t)$ that must be computed. To obtain zero latency tracking $y(t)$ must equal $r_s(t)$. Putting all of this information into Eq. (3) gives

$$r_s = \frac{(s + 1/\tau)^{-1}(1/\tau)\ K\ e^{-sT}}{1 + (s + 1/\tau)^{-1}(1/\tau)\ K\ e^{-sT}}(r_s + r_a)$$

Solving for r_a gives

$$r_a = \frac{e^{+sT}}{K}(\tau s + 1)r_s \qquad (4)$$

The e^{+sT} term shows that predictions must be made. However, the smooth pursuit system is a velocity tracking system, not a position tracking system, so the controller must be able to predict future values of target velocity. For example, if $\dot{r}_s(t)$ is the present target velocity, it must be able to produce $\dot{r}_s(t + T)$, where T is the time delay of the smooth pursuit system. And the controller must modify this prediction to compensate for the dynamics of the system in accordance with Eq. (4). Therefore, the compensation signal, R_c of Fig. 1 becomes

$$r_c(t) = \frac{1}{K}[\frac{d}{dt}\tau\dot{r}_s(t+T) + \dot{r}_s(t+T)] \tag{5}$$

This compensation signal allows the smooth pursuit system to overcome the time delay. To synthesize this signal the adaptive controller must be able to both predict future values of the target velocity, and compute first derivatives. These are reasonable computations for the human brain. Therefore, Eq. (5) is the algorithm that is in the box of Fig. 1 labeled Target Selective Adaptive Controller.

7. THE SENSITIVITY OF THE PREDICTOR TO PARAMETER CHANGES

To determine which parameters have the greatest effect on the model and when they exert their influence, we computed the semirelative sensitivity function, $\tilde{S}^y_\beta$, for each parameter (Bahill 1981, Bahill *et al.* 1980).

$$\tilde{S}^y_\beta = \beta_o \frac{\partial y}{\partial \beta}$$

where y is the output of the system and β is the parameter that is varied. For this study we used a fixed pertubation size of +5%. While tracking slowly moving targets the model is linear and therefore only one perturbation step size was needed (Bahill *et al.* 1980).

The smooth pursuit model developed in this study is not independent of other systems. The saccadic system and the adaptive predictor interact with the smooth pursuit branch. Therefore, we performed the sensitivity analysis both with the saccadic system and the predictor turned on, and with the saccadic system and the predictor turned off. Eliminating the saccadic system and predictor allowed us to isolate the pursuit branch and study it independently.

The sensitivity of the predictor was studied for three parameters: k_s, the proportionality constant; the number of weights; and the time to be predicted. For k_s and the number of weights, the target waveforms were also changed to determine if the predictor was sensitive to different input signals.

The effect of k_s was found to be the greatest after points of acceleration discontinuities. We performed a sensitivity analysis for many target waveforms, including the four shown in Fig. 5. The influence of k_s is most apparent for the analyses done with the cubic position waveforms. In Fig. 7 we show the results for the cubical target position waveform, which has the parabolic velocity waveform shown in this figure. $\tilde{S}_{k_s}$ peaks at the turnaround points and then begins to taper off to a steady state value.

Similar results were found for the sensitivity analyses when the number of weights was changed for each target waveform. Figure 8 shows the results of the sensitivity analysis for the weights. $\tilde{S}_n$ is similar for the two parameters, but the time of greatest sensitivity occurs a little later for the weights. This similarity of the two sensitivity functions is reasonable if the misadjustment algorithm for the adaptive filter from Eqs. (1) and (2) is recalled,

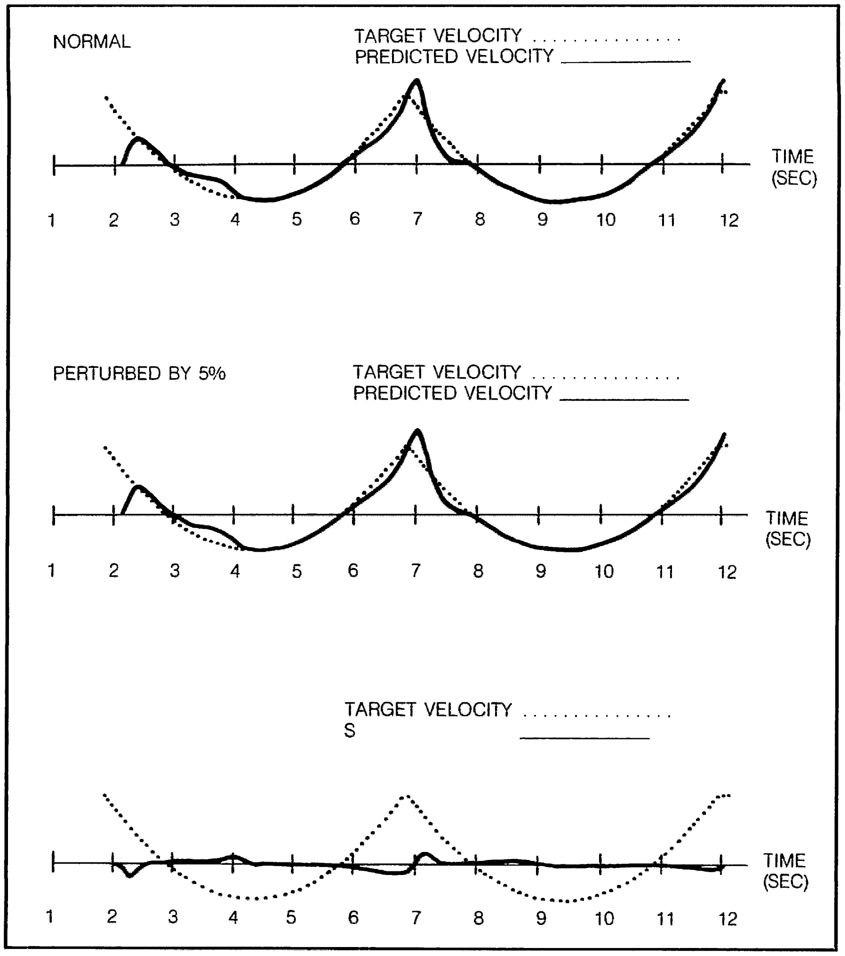

Figure 7. Semirelative sensitivity function of the predictor for changes in the proportionality constant, k_s, for a cubic waveform.

$$M = \frac{n}{4\tau_{mse}} = nk_s E[x^2]$$

This equation shows that a 5 percent change in either the proportionality constant, k_s, or the number of weights, n, will change the misadjustment of the predictor in a similar manner.

The other parameter changed for the predictor was the prediction time, the desired time to be estimated. The $\tilde{S}$ curve for this case also had the same shape as the curve for the number of weights and the proportionality constant, but its magnitude was smaller.

From these curves, the effect of the predictor can be determined. Changing each parameter by 5 percent showed that all of them exert their greatest influence right after acceleration discontinuities. Therefore, the predictor's influence will be the greatest at those points.

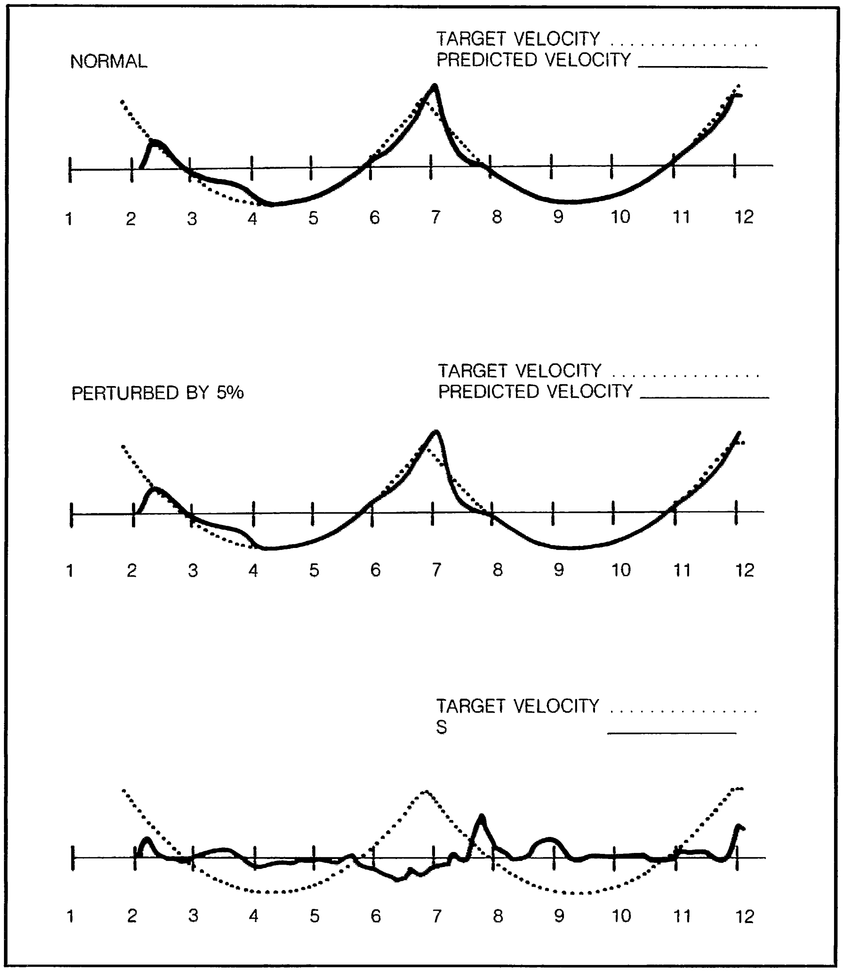

Figure 8. Semirelative sensitivity function of the predictor for changes in the number of weights for a cubic waveform.

8. THE EFFECT OF PARAMETER CHANGES ON THE MEAN-SQUARE ERROR

Our semirelative sensitivity analysis gives a measure of how changing a parameter affects the model and it shows when the parameter exerts its greatest influence. For our second sensitivity analysis, we considered the effect on the model's performance of changing each parameter over a range of values. Each parameter was given values above and below the nominal values; the velocity mean-square error between the model's output and the target was computed for each change. For the predictor, the filter's mean-square error, FMSE, was computed between the velocity of the target 150 msec in the future and the velocity predicted by the adaptive predictor. The mean-square errors were then plotted against the parameter values.

9. THE PREDICTOR'S SENSITIVITY TO CHANGES IN PARAMETERS

The effect of changes in the proportionality constant on the predictor was studied first. As the proportionality constant in Fig. 9 became larger, the filter's mean-square error became smaller. According to the misadjustment algorithm, the larger k_s, the larger the misadjustment. This appears to disagree with this figure. However, the mean-square error for the figure was taken during the first 12 seconds of the simulation; therefore, the start-up transients are influencing the error. The larger k_s the faster the filter adapts; for smaller k_s the filter takes longer to converge, but does converge to a solution with smaller error. Therefore, in the figure, the large mean-square error for a small k_s is because the filter takes longer to converge to the optimal solution. With the larger k_s values, the filter is converging rapidly and appears to have a smaller error. If k_s were increased even more, the error would also begin to increase. When we made the filter's task easier, by eliminating the startup transient and only studying the steady-state behavior, we found the FMSE increased with k_s as expected.

Our detailed analysis also showed a larger mean-square error for the cubic waveform compared to the sinusoidal waveform. This result is not unexpected since the cubic is a higher order waveform than the sine wave and the misadjustment is proportional to the expected value of the input signal.

Referring to Fig. 10, the mean-square error of the predictor is shown as a function of the number of weights in the adaptive filter. According to the misadjustment algorithm, as the number of weights increases so does the misadjustment of the filter. The curves here show the filter's error decreasing until 15 weights and

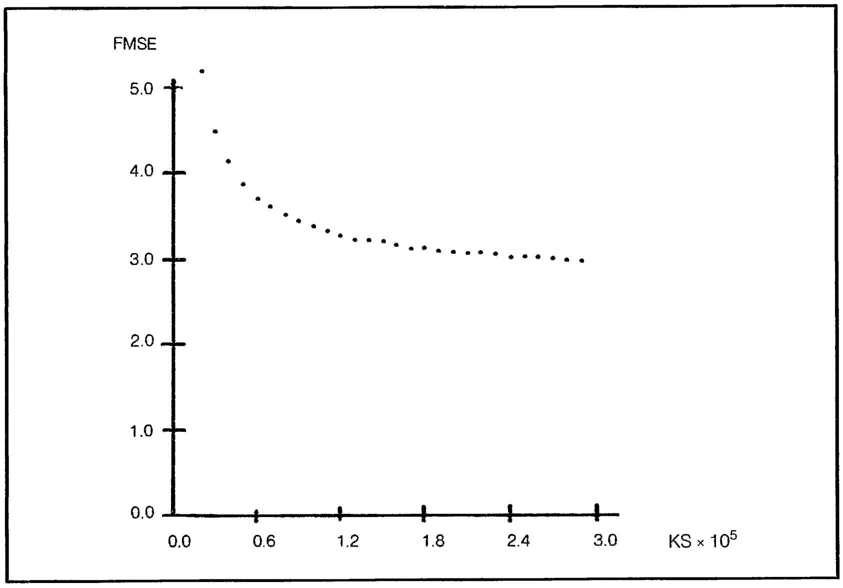

Figure 9. The mean-square error of the predictor as a function of changes in the proportionality constant, k_s. The top figure shows the changes in the error for a cubic waveform. The bottom figure is the mean-square error as a function of changes in k_s for a parabolic waveform.

then rising slightly before falling off after 40 weights. Because the adaptive filter is using a tapped delay-line input signal, as the number of weights is increased the input signals for the adaptive and slave filters begin to overlap. This improves the predictor's performance because the statistics of the two input signals are the same since the input signals are the same.

The increase in error between 15 and 40 weights shows the rise in error predicted by the misadjustment algorithm. However, after 40 weights the statistics of the input signals for the two filters begin to get close enough that the error drops off. The input signals for the two filters begin overlapping after 30 weights, which is approximately where the curves peak.

The effect of changing the prediction time and the signal's frequency were also studied. Figure 11 shows the predictor's error as a function of prediction time. The error appears to be a linear function of the prediction time. The further into the future that is to be predicted, the worse the predictor does. We also computed that for changes in frequency, the faster the target moves the poorer the predictor does.

Summarizing, the predictor's performance is poorer as the proportionality constant is increased, although the error is a function of the time when the measurements

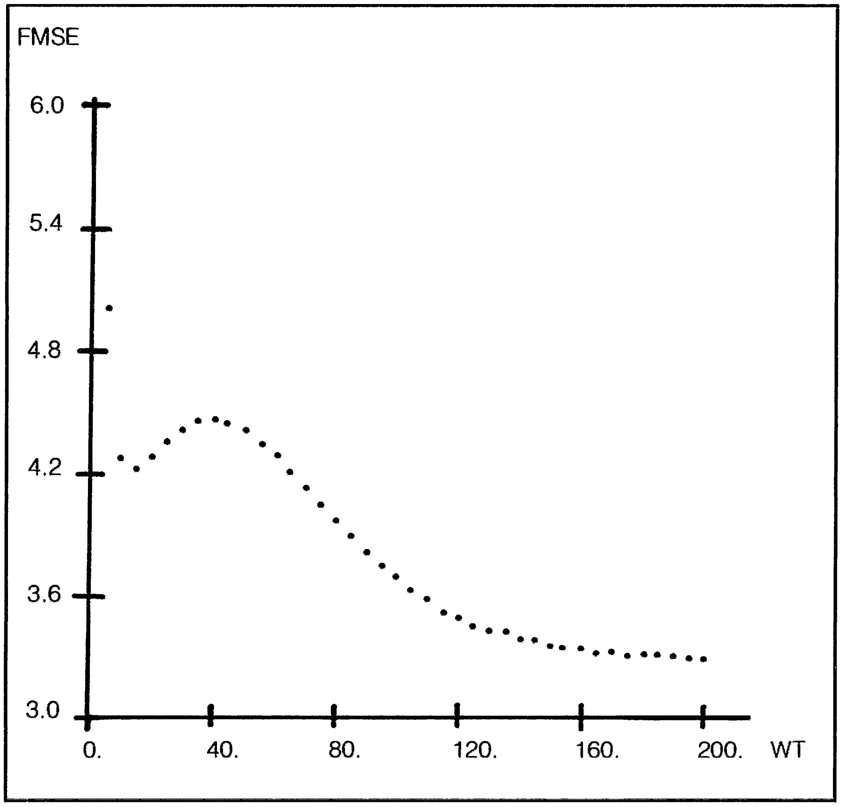

Figure 10. The mean-square error of the predictor as a function of changes in the number of weights for a cubic waveform.

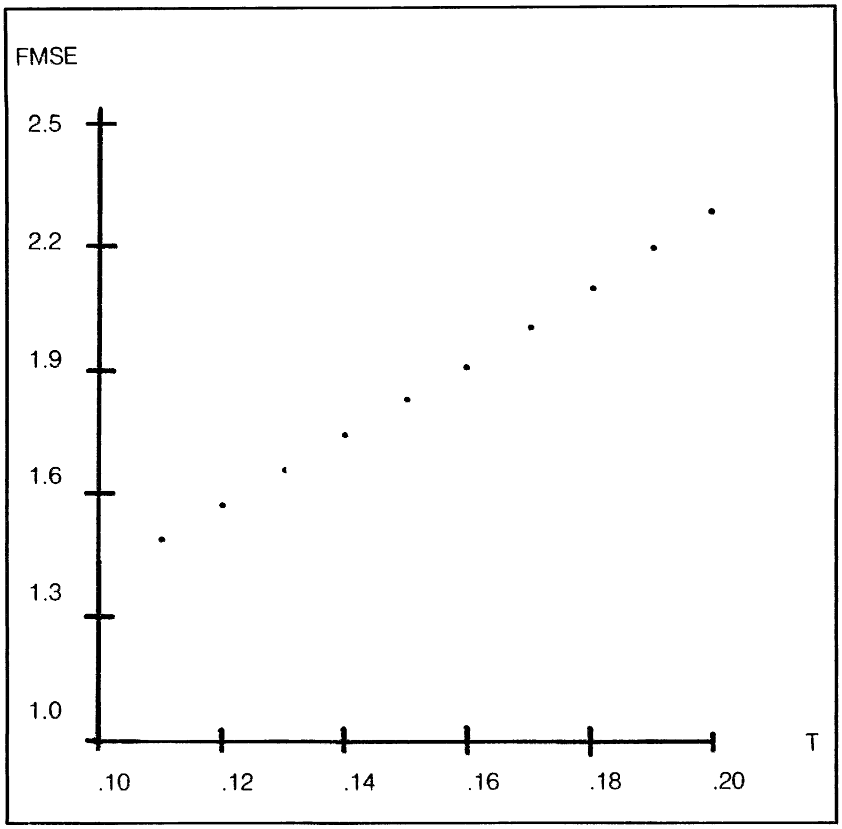

Figure 11. The change in the mean-square error of the predictor as the prediction time is changed.

were taken. For instance, in this study, the start-up transients have not died down so the reverse statement appears true. For the weights, as the number of weights increased, the error also increased. The exception, seen in this work, is when a tapped delay-line input signal is used and the statistics of the input signals to the adaptive and slave filters are similar. The error of the predictor increases as the prediction time increases and as the input signal's frequency increases.

10. DISCUSSION

To create a model, we first determine the form, then derive parameter values. When possible we use physiological data to derive these values. A sensitivity analysis shows which parameters are the most and the least important so we can focus our efforts appropriately. In one of our final modeling stages we run a function minimization routine to adjust parameter values so that we get the least squared error between the human and the model outputs.

When we applied our parameter estimation program to this present model it failed, because the number of weights was discrete, and our function minimization routine used a modified gradient technique, which only works with continuous functions (Latimer and Bahill 1979). Therefore, we used a manual function minimization scheme as shown in Figs. 9 and 10. We are presently modifying our parameter estimation program so that it will work on discrete parameters.

Our model shown in Fig. 1 is an approximation to the human smooth pursuit system. Similarly our simulation is only an approximation of the model of Fig. 1. For example, the model of Fig. 1 should be stable for any gain up to 2.3. But our simulation started to oscillate at 1.8. We found that we were getting 5 to 10 degrees of artifactual phase shift from the differentiators, the integrators, and even the summers. A smaller simulation step size would have obviously solved the problem, however, just being aware of the problem was also sufficient.

11. CONCLUSIONS

Our least mean-squares predictor worked well except when discontinuities in the target waveform were present. For any desired accuracy, trade-offs could be made between the predicted gain and the number of weights. When this predictor was incorporated into our full eye movement model, the model was able to overcome its 150 msec delay and track targets with no latency, just like the human.

For optimal performance, 150 weights were used. Because the model gets a new target position every 5 msec, this means it uses the previous 750 msec of data for each calculation. We are not sure that the human uses this large a data window. Therefore, we also ran the model with only 15 weights. Even with this reduced number of weights, the model still performed as well as the human. We are presently designing experiments to be performed on the model and on the human that will allow us to differentiate between these possibilities.

ACKNOWLEDGMENTS

This research was partially supported by NSF grant ECS-8121259, and a contract from Canadian Aviation Electronics (CAE).

REFERENCES

Bahill, A. T. 1981. *Bioengineering: Biomedical, Medical and Clinical Engineering.* Englewood Cliffs, New Jersey: Prentice-Hall Inc.

Bahill, A. T. and D. R. Harvey. 1986. "Open-loop experiments for modeling the human eye movement system," *IEEE Transactions on Systems, Man, and Cybernetics,* **SMC-16** (in press).

Bahill, A. T. and T. LaRitz. 1984. "Why can't batters keep their eyes on the ball," *American Scientist,* 72:249-253.

Bahill, A. T., J. R. Latimer, and B. T. Troost. 1980. "Sensitivity analysis of linear homeomorphic model for human movement," *IEEE Transactions on Systems, Man, and Cybernetics,* **SMC-10**:924-929.

Bahill, A. T. and J. D. McDonald. 1983. "Model emulates human smooth pursuit system producing zero-latency target tracking," *Biological Cybernetics,* 48:213-222.

Bahill, A. T. and L. Stark. 1975. "Overlapping saccades and glissades are produced by fatigue in the saccadic eye movement system," *Experimental Neurology,* 48:95-106.

Brodkey, J. S. and L. Stark. 1968. "New direct evidence against intermittency or sampling in human smooth pursuit eye movements," *Nature,* 218:273-275.

Latimer, J. R. and A. T. Bahill. 1979. "Parameter estimation by function minimization using a modified steepest descent method," In *Modeling and Simulation, Proc. Tenth Annual Pittsburgh Conference* edited by W. Vogt and M. Mickle, pp. 683-690. Pittsburgh, Instrument Society of America.

Leigh, R. J. and D. S. Zee. 1983. *The Neurology of Eye Movements.* Philadelphia: F. A. Davis Co.

McDonald, J. D. and A. T. Bahill. 1983. "Zero-latency tracking of predictable targets by time-delay systems," *International Journal of Control,* 38:881-893.

Rashbass, C. 1961. "The relationship between saccadic and smooth tracking eye movements," *Journal of Physiology,* **159**:326-338.

Robinson, D. A. 1965. "The mechanics of human smooth pursuit eye movements," *Journal of Physiology,* **180**:569-591.

Widrow, B. 1971. "Adaptive filters," In *Aspects of Network and System Theory,* edited by R. E. Kalman and N. DeClaris, pp. 563-587. New York: Holt Rinehart and Winston Inc.

Widrow, B., J. M. McCool, M. G. Larimore, and C. R. Johnson. 1976. "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE,* **64**:1151-1162.

Widrow, B. and S. D. Stearns. 1985. *Adaptive Signal Processing.* Englewood Cliffs: Prentice Hall.

Young, L. 1971. "Pursuit eye tracking movements," In *The Control of Eye Movements,* edited by P. Bach-y-Rita, C. C. J. Collins, and J. Hyde, pp. 429-443. New York: Academic Press.

Young, L. R. and L. Stark. 1963. "Variable feedback experiments testing a sampled data model for eye tracking movements," *IEEE Transactions on Human Factors Electronics,* **HFE-4**:38-51.

Deborah R. Harvey received the A.B. degree in mathematical physics from Sweet Briar College, Sweet Briar, VA, and the M.S. degree in electrical engineering from Carnegie-Mellon University, Pittsburgh, PA.

During 1984, Ms. Harvey was a member of the Technical Staff at TRW, Inc., McLean, VA, where she was involved in studies of fiber optic sensor systems. In January 1985, she joined the MITRE Corporation as a member of the technical staff. Since then she has been involved in the research and development of fiber optic local area networks.

Ms. Harvey is a member of Phi Beta Kappa.

A. Terry Bahill received a B.S. in electrical engineering from the University of Arizona, Tucson, an M.S. in electrical engineering from San Jose State University, and a Ph.D. in electrical engineering and computer science from the University of California, Berkeley.

From 1967 to 1971 he served as a Lieutenant in the U.S. Navy teaching mathematics and electrical engineering to the students of the Navy Nuclear Power School, Mare Island, CA. From 1976 to 1984 he was an Assistant and subsequently an Associate Professor of Electrical and Biomedical Engineering at Carnegie-Mellon University, and concurrently of Neurology at the University of Pittsburgh. He is now Professor of Systems and Industrial Engineering at the University of Arizona in Tucson. His research interests include systems theory, modeling physiological systems, head and eye coordination of baseball players, expert systems, and computer text and data processing. He is the author of Bioengineering: Biomedical, Medical, and Clinical Engineering (Prentice-Hall, 1981).

Dr. Bahill is a member of the IEEE societies; Systems, Man, and Cybernetics, Engineering in Medicine and Biology, Automatic Controls, and Professional Communications. He was Vice President for Publications and is now Vice President for Meetings and Conferences and an associate editor for the Systems, Man, and Cybernetics Society. He is an associate editor of IEEE Expert. He is a member of Tau Beta Pi, Sigma Xi, Psi Chi, and a Registered Professional Engineer.