

01 Jan 1999

## Predictive Head Tracking for Virtual Reality

Donald C. Wunsch

*Missouri University of Science and Technology, dwunsch@mst.edu*

Emad W. Saad

T. P. Caudell

Follow this and additional works at: [https://scholarsmine.mst.edu/ele\\_comeng\\_facwork](https://scholarsmine.mst.edu/ele_comeng_facwork)



Part of the [Electrical and Computer Engineering Commons](#)

---

### Recommended Citation

D. C. Wunsch et al., "Predictive Head Tracking for Virtual Reality," *Proceedings of the International Joint Conference on Neural Networks, 1999. IJCNN '99*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1999.

The definitive version is available at <https://doi.org/10.1109/IJCNN.1999.830785>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# Predictive head tracking for virtual reality

E. W. Saad<sup>1</sup>, T. P. Caudell<sup>2</sup>, and D. C. Wunsch II<sup>1</sup>

<sup>1</sup>Applied Computational Intelligence Laboratory, Dept. of Electrical Engineering,  
Texas Tech University, Lubbock, TX 79409-3102

<sup>2</sup>Dept. of EECE

University of New Mexico, Albuquerque, NM 87131  
saade@ttu.edu, tpc@eece.unm.edu, and dwunsch@coe.ttu.edu  
http://www.acil.ttu.edu, http://www.eece.unm.edu/faculty/tpc/

## Abstract

In Virtual Reality (VR), head movement is tracked through inertial and optical sensors. Computation and communication times result in delays between measurements and updating of the new frame in the head mounted display (HMD). These delays result in problems, including motion sickness. We use recurrent and time delay neural networks to predict the head location and use it to calculate the new frame. A predictability analysis is used in designing the prediction system

## Introduction

In virtual reality systems, different optical and inertial sensors are used to track the movement of the user's head. Measured variables are the sampling time  $t$ , three coordinates  $x$ ,  $y$ , and  $z$ , and head angles  $\alpha$ ,  $\beta$ , and  $\gamma$ . Processor as well as communication delays result in a delayed update of the scene on the head-mounted display (HMD). This delayed display may produce dizziness and motion sickness of the user. A model which can predict the next head position and orientation can help computing and updating the display faster, and reducing or eliminating dizziness.

Many linear autoregressive models have been used in time series prediction. Neural networks have been shown to be powerful nonlinear models in predicting time series in various applications [1], [2]. In particular, time-delay neural networks (TDNN) and recurrent neural networks (RNN), have been shown to be most suited for dynamic modeling of time series [3], [4]. In this case the network input is the value of the variable to be predicted at one or more previous time steps. The output is the prediction of its value at the next time step. Such one-step time series prediction can be iterated for predicting multisteps in the future. Such multistep time series prediction is extremely hard because of the accumulation of the prediction error at

every step. In our case, we are only interested in single step prediction of the six position and orientation variables above.

One question is whether the time series is predictable or not. For example, in a stock market, the random walk principle suggests that the stock price is random, and does not depend on the historical values of the stock. This may not always be true. Chaotic time-series look like random, but actually represent a deterministic dynamic system and can be modeled and hence predicted [3]. In this paper we use predictability analysis tools in order to estimate the degree of determinism of the different series, and design the prediction system.

## Problem Description

In our VR system, the display is computed based on the position coordinates  $x$ ,  $y$ , and  $z$ , as well as head Euler angles  $\alpha$ ,  $\beta$ , and  $\gamma$ . Also, the sampling time of these six variables is not constant, but is affected by the processor load and communication delays. Therefore, the prediction model needs to first predict the next sampling time step, then predict the other six variables and use them in calculating the new frame in the HMD.

## Approach

Our approach is to train a neural network to learn an individual motion profile. Thus, a different network would be used for every user. The system will engage the neural network predictor only after it has learned to predict better than some threshold.

Three different approaches were considered. One approach is to use a unified network for multidimensional time series prediction. All seven variables are used as inputs to the network, and the network is trained to predict the next value of the all variables simultaneously. The advantage of this approach is to exploit the mutual information between the variables. On the other hand, if the variables are not correlated, doing a multidimensional time series prediction will be a harder task on the network.

The second approach is to use two networks, one for predicting the position, and the other for predicting the angles. This approach should work better, if the angles are not correlated with the position.

The last approach is to use a separate decoupled network for every time series. Thus, seven networks are used. Predicting a single time series should be an easier task for the neural network. The performance should be better, if the different series are not correlated.

The predictability analysis described below was used to choose one of the three designs.

### Predictability Analysis

Predictability analysis tools serve in determining the degree to which a time series is random or chaotic. A chaotic series looks like a random one, but is governed by a deterministic system, so is predictable. The head tracking variables can be a mixture of random and chaotic series, and thus predictable to different degrees. Different predictability analysis tools have been useful in analyzing the time series, and designing the prediction system, for example by choosing the delay between the TDNN inputs as well as the number of taps [3].

#### A. Phase Space Diagrams

A phase space diagram (phase diagram) is the easiest test of chaotic behavior. It is a scatter plot where the independent variable is the value of a time series  $x(t)$  at time  $t$ , and the dependent variable is  $x(t+\tau)$ . The delay  $\tau$  can be chosen as the first zero of the series autocorrelation coefficient.

The phase diagram of a deterministic system is identified by its regularity. The trajectory is contained in a limited area of the range of the series called an *attractor*. This is in contrast to a random series where the trajectory covers all the range of the diagram. Phase diagrams can be plotted only in two or three dimensions, which is the main shortcoming of this technique

#### B. Lyapunov Exponent

Chaos is characterized by sensitivity to initial conditions. The *Lyapunov Exponent* measures divergence of two orbits starting with slightly different initial conditions [5]. If one orbit starts at  $x_o$  and the other at  $x_o + \Delta x_o$ , after  $n$  steps, the divergence between orbits becomes

$$\Delta x_n = |f^n(x_o + \Delta x_o) - f^n(x_o)|, \quad (1)$$

where  $x_{n+1} = f(x_n)$ . For chaotic orbits,  $\Delta x_n$  increases exponentially for large  $n$ :

$$\Delta x_n = \Delta x_o e^{\lambda n}, \quad (2)$$

where  $\lambda$  is the Lyapunov Exponent:

$$\lambda = \lim_{n \rightarrow \infty} [ (1/n) \ln (\Delta x_n / \Delta x_o) ]. \quad (3)$$

A positive exponent indicates chaotic behavior. If the exponent is very small or negative, this means that the series is either random or periodic.

This test is practical, and does not have the limitations of other tests such as the correlation dimension [6] which is limited by the number of available data points.

### Time Delay and Recurrent Neural Networks

The Time-Delay Neural Networks (TDNN) used in this study are feedforward Multilayer Perceptrons, where the internal weights are replaced by finite impulse response (FIR) filters. This builds an internal memory for time series prediction [7]. The input of the network consists of a delay line corresponding to each time series. The delay between each tap has been estimated using the first zero of the autocorrelation. This is useful in minimizing the redundancy between the different taps.

The Recurrent Neural Network (RNN) considered in this paper (Fig. 1) is a type of Discrete-Time Recurrent Multilayer Perceptrons [8]. Temporal representation capabilities of this RNN can be better than those of purely feedforward networks, even with tapped-delay lines. Unlike other networks, RNN is capable of representing and encoding deeply hidden states, in which a network's output depends on an arbitrary number of previous inputs.

Among many methods proposed for training RNNs, Extended Kalman Filter<sup>1</sup> (EKF) training stands out [9]. EKF training is a parameter identification technique for a nonlinear dynamic system (RNN). This method adapts weights of the network pattern-by-pattern accumulating training information in approximate error covariance matrices and providing individually adjusted updates for the network's weights.

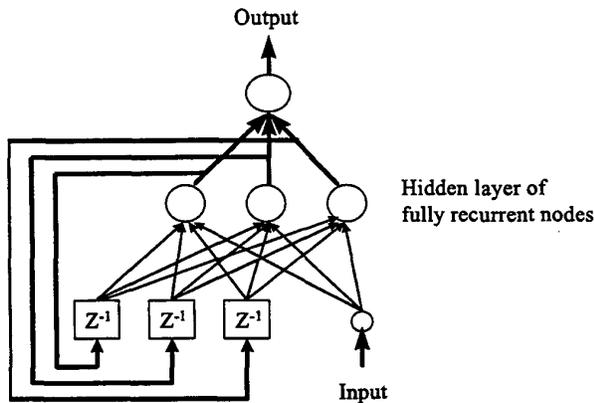


Fig. 1. Recurrent network architecture.  $Z^{-1}$  represents a one time step delay unit. This network has a compact memory structure. The EKF described is well-suited for this architecture.

Unlike TDNN, RNN is easier to implement, since there is no need of choosing the number of delays. The recurrence creates an internal memory in the network.

## Experimental Results

By plotting the sampling time, the sampling interval was found almost constant at about 20 ms, most of the time, except for some spikes of constant amplitude due to network communication, not processor load. Therefore, we decided to predict the other 6 variables independently from the sampling interval.

### A. Lyapunov Exponent

The Lyapunov exponent has been calculated for all 6 variables. Table 1 shows the calculated values for every variable. We notice that while the position has negative and small exponents, the angles have larger positive one.

<sup>1</sup> We note that the full name of the EKF method described here is parameter-based node-decoupled EKF.

This suggests that they are more predictable than the position.

Table 1 Lyapunov exponent for position and angles of head.

Variable	Lyapunov exponent
$x$	-0.0074
$y$	-0.0018
$z$	-0.027
$\alpha$	0.417
$\beta$	0.17
$\gamma$	0.21

### B. Correlation Coefficient

The correlation between the different coordinates as well as between the different angles has been calculated. The following values have been obtained:  $\rho_{xy}=-0.51$ ,  $\rho_{xz}=-0.72$ ,  $\rho_{yz}=0.83$ ,  $\rho_{\alpha\beta}=0.15$ ,  $\rho_{\alpha\gamma}=-0.0074$ ,  $\rho_{\beta\gamma}=0.44$ . This shows that generally the angles are less correlated than the position coordinates. This result agrees with the prediction results shown below, where decoupling the prediction of the angles performed better than the unified network.

### C. Prediction

We started by predicting only the angles for two reasons. First, rotations produce the greatest amount of scene change in the graphics, since seated persons can only translate their head in a limited range. Second, these are more predictable than the coordinates according to the calculation of Lyapunov exponent.

Comparing the unified and decoupled neural networks, we found that using a separate network for each angle resulted in a more accurate prediction. This agrees with the low correlation coefficients calculated above. Fig. 2 shows the predictions of the  $\alpha$  angle using RNN and TDNN. In this case RNN provided better quality predictions.

## Conclusion

Time series prediction with neural networks is used to minimize head tracking delay in VR. Recurrent and time delay neural networks are chosen for the internal memory.  $M$ -dimensional time series analysis is investigated using recurrent and time delay neural networks and applied to head tracking in VR systems. A predictability analysis is done, and the results are used in designing the prediction system. The resulting system achieved adequate performance using both techniques, although the RNN results were the most accurate.

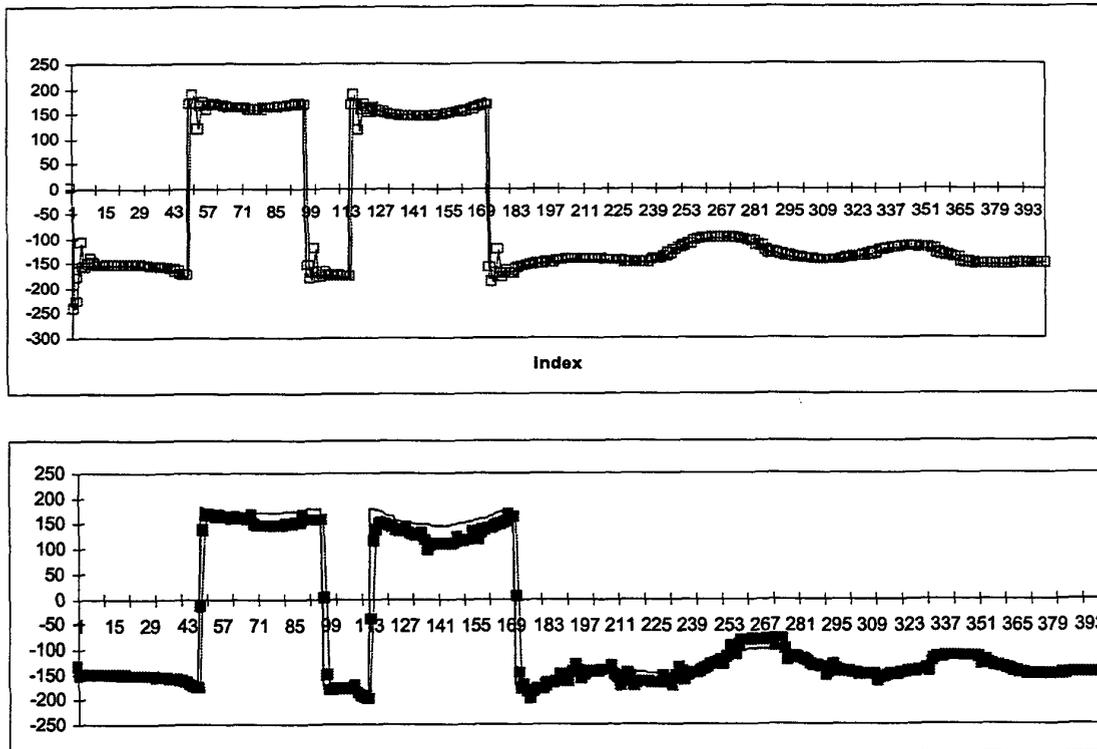


Fig. 2 Prediction of the  $\alpha$  angle using RNN and TDNN respectively. The desired signal is indicated by the solid line. The prediction is indicated by the line with square markers. The predictions are accurate enough to improve head tracking performance, especially, in this case, for the RNN approach.

### References

- [1] N. Gershenfeld, and A. Weigend, "The Future of Time Series: Learning and Understanding," *Time Series Prediction: Forecasting the Future and Understanding the Past* (A.S. Weigend and N.A. Gershenfeld).
- [2] A. Weigend, B. Huberman, and D. Rumelhart, "Predicting the Future: A Connectionist Approach," *International Journal of Neural Systems*, vol. 1, pp. 193-209, 1990.
- [3] E. W. Saad, D. V. Prokhorov and D. C. Wunsch II, "Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks," *IEEE Trans. Neural Networks*, vol. 9, no. 6, 1998.
- [4] E. Saad, D. Prokhorov, and D. Wunsch, "Advanced Neural Network Training Methods for Low False Alarm Stock Trend Prediction," In *Proc. IEEE International Conference on Neural Networks*, Washington, DC, June 1996, pp. 2021-2026.
- [5] H. Korsch, and H. Jodl, *Chaos: A Program Collection for the PC*. Berlin: Springer-Verlag, 1994.
- [6] P. Grassberger, and Procaccia, "Characterization of Strange Attractors," *Phys. Rev. Lett.*, vol. 50, pp. 346-349, 1983.
- [7] E. Wan, "Time Series Prediction by Using a Connectionist Network with Internal Delay Lines," *Time Series Prediction: Forecasting the Future and Understanding the Past* (A.S. Weigend and N.A. Gershenfeld, eds.), pp. 195-217. Reading, MA: Addison-Wesley, 1993.
- [8] D. Hush, and B. Horne., "Progress in Supervised Neural Networks," *IEEE Signal Processing Mag.*, Jan. 1993.
- [9] G. Puskorius, and L. Feldkamp, "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 279-297, 1994.