1998

Determination of Head Kinematics from Impact Acceleration Test Data Using Neural Networks

Edit J. Kaminsky
University of New Orleans, ebourge@uno.edu

Anwer Bashi
University of New Orleans

Recommended Citation

Kaminsky, Edit J. and Anwer S. Bashi, "Determination of Head Kinematics from Impact Acceleration Test Data using Neural Networks," 1998 IASTED Internal. Conf. on Signal Processing and Communications (ICSPC'98), (Canary Islands, Spain), Feb. 11-14, 1998, pp. 224-227.

Proceedings of the IASTED International Conference
Signal Processing and Communications
February 11-14, 1998, Canary Islands - Spain

DETERMINATION OF HEAD KINEMATICS FROM IMPACT ACCELERATION TEST DATA USING NEURAL NETWORKS

EDIT KAMINSKY and ANWER BASHI
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148
U.S.A.

ABSTRACT

This paper presents a study of feed-forward neural network (NN) systems developed to determine the head kinematics of subjects who are exposed to impact accelerations. The neural networks process accelerometer data collected during short-duration impact acceleration tests conducted at the National Biodynamics Laboratory[1] of the University of New Orleans. During an impact acceleration experiment, the subject sits on the sled chair and a piston gives impetus to the sled to travel down a track. Head data is gathered by an array of nine accelerometers. Two more accelerometers are mounted on the sled.

The neural processing systems produce the history of the rotational and translational position, velocity, and acceleration of the origin of the accelerometer array mounted on the mouth. Output produced by a least squares algorithm that uses both photographic and accelerometer raw data are used as a baseline and to provide training data for the neural networks. The main advantages of the NNs are their speed, and that statistical information and accurate modeling of the testing system are not required. Results show that the neural networks provide accurate information about the kinematics of the subject even when no photographic data are used.

Keywords: *Neural networks, kinematics, quaternions, accelerometers, impact acceleration*.

INTRODUCTION

The National Biodynamics Laboratory (NBDL) of the University of New Orleans (UNO) conducts short-duration sled impact experiments with human subjects or anthropomorphic mannequins sitting on a sled. A piston suddenly impacts on the sled (at up to about 16 g's for humans) and forces the sled and subject to travel down a 700-ft. track. During these impact acceleration tests, NBDL collects linear inertial acceleration data [1] every 0.5 ms with arrays of accelerometers mounted on a T-plate and attached to the subject's mouth; sled sensor data is also gathered. Photographic data [2] is collected as well, but is not currently being used in the neural system.

In this paper we present the development of neural network (NN) systems used to process accelerometer input data and produce rotational and translational kinematics (displacement, velocity, and acceleration) of the subject's head. Results of the NNs are compared to those of an extended Kalman filter [3,4] and EZFLOW, a least-squares processing algorithm developed by NBDL and currently used to process all photographic and sensor data collected.

In what follows we present the description of the system, results of the neural network, a comparison to the results from other processing systems, and conclusions and suggestions for further work.

SYSTEM DESCRIPTION

The desired output variables are the kinematics of the center of the accelerometer array mounted on the mouth. The 24-element state vector is

$$x=[y_1,y_2,y_3,z_1,z_2,\cdots,z_9,q_1,q_2,\cdots,q_{12}]$$

and consists of the displacement, velocity, and acceleration of the sled in the x (thrust) direction with respect to the fixed laboratory coordinate system (y_1 through y_3); the x, y, and z components of the displacement, velocity, and acceleration of the origin of the mouth accelerometer array with respect to the sled coordinate system (z_1 through z_9); and the rotational displacement, velocity, and acceleration given in quaternion notation (q_1 through q_{12}).

The observable vector, ζ, consists of the acceleration measured by each of the nine sensors in the mouth array and the two sled accelerometers, and is defined as

$$\zeta=[\alpha_1,\alpha_2,\cdots,\alpha_9,\gamma_1,\gamma_2]$$

The observable vector ζ, or at least some of its components, is the input to the neural network. These measurements are, of course, contaminated by measurement noise. The state vector x or, again, some of its components, is what the neural network produces as output. The instrumentation data consists of 2000 samples from each of the nine accelerometers; the sampling rate is 0.5 ms.

THE NEURAL NETWORKS

Several three-layer, fully-connected, feed-forward neural networks were designed, implemented, and tested to produce the desired output kinematics. All of them use the backpropagation training paradigm. Noisy input data are available from the impact acceleration system, and output data (kinematic variables) from a processing system currently used by NBDL, called EZFLOW [5], are also available and used to train the neural nets.

Eight sets of files containing instrumentation data and the corresponding (assumed) correct mappings were used to train each neural network. Three more sets of files (that the network had not been trained on) were used to test the neural networks' generalization abilities. All of the data contained in the files were obtained from independent runs performed by NBDL (i.e. different subjects, different "g-levels", etc.)

There are 9 accelerometers, each providing 2000 samples of observable data. The EZFLOW training data consists of 1600 samples of the output kinematic variables and are used as the desired response while training. We determined that using six of these nine accelerometers was sufficient for training the network and also for generalizing in response to new input data. There are only six degrees of freedom in the system.

To further reduce the size of the networks we only output the rotational and translational acceleration kinematics. If the network can map these states, then the other states can also be obtained. This is due to the fact that the remainder of the kinematics are scaled summations (or integrations) of the acceleration kinematics set.

Our networks will have to consist of at least three layers since we do not have a linearly separable solution. The transfer function of the output layer is purely linear so that a wide range of values can be represented; the transfer function of the hidden layer was chosen to be a symmetric sigmoid. The networks are trained using a back-propagation algorithm with momentum term and variable learning rate.

The general method by which temporal or sequential patterns are recognized by a neural network is by applying the input vector (let us call the k-time input vector x_k) to a tapped delay line. The set of resulting vectors is then fed into the network as the new input. Using this technique, a neural network can recognize the time dependencies of the data as well as such non-localized properties as rate of change, variance, mean, etc. We tried two techniques of presenting the training data: Piecewise presentation, and complete presentation.

Piecewise Presentation

We can provide a local "windowed" version of the data by using a tapped delay line with a length that is shorter than the length of the data to be presented:

$$< x_{k-m}, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_{k+m} >$$

For convenience, we will refer to this as the M^{th}-order model. We will use a slight variation of this window that has a broader selective window:

$$< x_{k-mc}, \ldots, x_{k-1c}, x_k, x_{k+1c}, \ldots, x_{k+mc} >$$

Where c is a positive constant greater than one that stretches the window's samples in order to get a better look at the "big picture".

We can also assume that our system has discrete operating modes corresponding to the following: Pre-impact, impact acceleration, impact deceleration, and settling down (see Fig. 1).

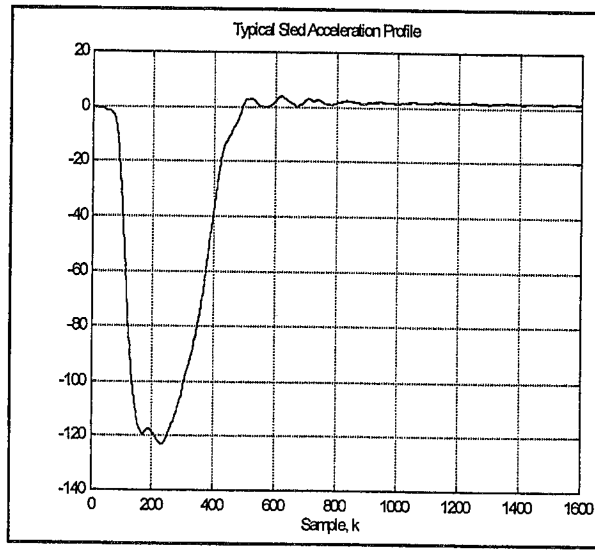

Fig. 1: Sled acceleration showing pre-impact, impact acceleration, deceleration, and settling (m/s²).

This leads us to suggest that the input can be presented for piece-wise analysis since the network can deliver a response at the presentation of data corresponding to each mode.

[1]Formerly the Naval Biodynamics Laboratory.

Training signals are generated by applying a sliding window both to the instrumentation data and to the processed data. This "local image" of the instrumentation data is then fed to the network as the input and the image of the processed data presented as the desired response. We can thus extract training sets from each run.

Complete Presentation

This technique presents all the input-output data at once. The difficulty to be overcome here is that 1600 samples for each of the six selected accelerometers and each output are to be presented. The size of the resultant network in this case would be massive. Obviously, some down-sampling is appropriate here. The output of the network will be compared to the output of a down-sampled version of the desired output to produce the error vector.

The most obvious and simple method is to use a uniform sampling rate applied to both the input and the desired response to down-sample the original data stream. We apply a low-pass filter to the input to remove the high-frequency noise components from the signal before subsampling.

RESULTS

When presented with the data through a sliding window, the results were not as good as those obtained with a down-sampled version of the complete history; we present these results for completeness. Several runs were performed with varying number of samples, model order, hidden neurons, and training epochs. When M=1, poor results were obtained. The results were better when M=2 and slightly better than that when M=3. When the model order was increased to M=4, no significant improvement took place. The peaks of the estimate of the linear accelerations seem to start at the same time as the peaks for the desired response, however the magnitudes don't match well, with errors as large as about 30 %.

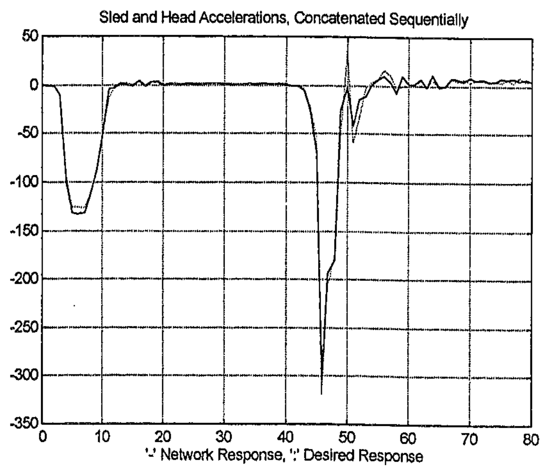

Fig. 2: Sled and head NN accelerations (complete pres.) (m/s^2)

The best results were obtained when the neural network was presented with the complete (un-windowed) down-sampled signal vector; in this case the networks response almost overlaps the desired response, both in the case of the linear accelerations and in the case of the rotational accelerations. In Figs. 2 and 3 we show examples of the results obtained with the neural network as well as the results from EZFLOW which are used to validate the networks performance. Fig. 2 presents the acceleration of the head and the sled in the x-direction concatenated, while Fig. 3 presents each of the four components of the acceleration quaternion, again concatenated into one vector.

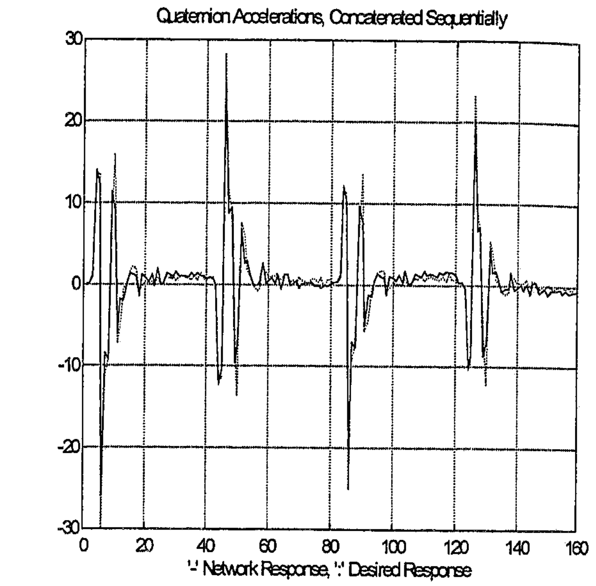

Fig. 3: Quaternion NN accelerations (complete pres.) (s^{-2})

Our results show that it is possible to solve the non-linear mapping problem with the use of a simple neural networks. We trained our networks with 40, 80, and 160 samples (out of a total of 1600), all with similar results.

Some numerical results are displayed in Table I. The column labeled "Total Error" is obtained by calculating the sum of absolute error between the network's response to a previously unseen stimulus and the desired (EZFLOW) response to that stimulus; it is the sum of the errors from each of the network's output neurons for all time. The column TE/Sample indicates the sum of the errors for the six output states, averaged over discrete time.

The optimal size of the network's hidden layer seems to be around

$$\sqrt{n+m}$$

where n is the number of inputs to the network, and m is the number of outputs.

CONCLUSIONS

We have shown that it is possible to solve the non-linear mapping problem of inertial accelerometer array data to head kinematics data with the use of a neural network by sub-sampling the original sequences and training three-layer backpropagation networks. Both rotational and translational kinematics were produced with very small differences when compared to output from a least squares processing system. Notice that only accelerometer data was used with the neural networks, while EZFLOW requires both accelerometer and photographic data to produce results.

In the future, tests using recurrent networks to solve this problem with windowed data might prove to be even more fruitful than the technique presented here. Also, we may take advantage of the availability of photographic data and improve the performance of the neural networks by using these data along with the sensor data.

We can also use specialized sub-nets for each of the four modes (pre-impact, acceleration, deceleration, and settling). While more networks would have to be used, each one would probably be smaller than one single network that works for all modes.

REFERENCES

[1] Padgaonkar, A. J., K. W. Krieger and A. I. King, "Measurement of Angular Acceleration of a Rigid Body Using Linear Accelerometers", *Journal of Applied Mechanics*, Paper No. 75-APMB-3, March, 1975.

[2] Kaminsky, E. J., R. E. Trahan, P. M. Chirlian, and B. King, "Extended Kalman Filter for Photographic Data from Impact Acceleration Tests", in *Proc. of IASTED SIP'96, Signal and Image Processing Conf.* (Orlando, FL), Nov. 11-14, 1996, pp. 59-63.

[3] Bashi, A., and E. J. Kaminsky, "Comparison of Neural Network and Extended Kalman Filter Determination of Kinematics from Impact Acceleration Tests," in *Proc. of 1997 IEEE Internat. Conf. on Systems, Man, and Cybernetics*, (Orlando, FL), Oct. 12-15, 1997, vol.4, pp. 3501-3506.

[4] Kaminsky, E. J., R. E. Trahan, P.M. Chirlian, and D. Malley, "Extended Kalman Filter for Accelerometer Data from Impact Acceleration Tests," in *Proc. IASTED Signal and Image Process. Conf. (SIP'96)*, (Orlando, FL), Nov. 11-14, 1996, pp. 147-149.

[5] Becker, E. and G. Willems, "An Experimentally Validated 3-D Inertial Tracking Package for Application in Biodynamic Research", in *Proc. Nineteenth Stapp Car Conf.,* (San Diego, CA), Nov. 17-19, 1975, pp. 899--930

Table I: Neural network configurations and results.

Number Samples	Input nodes	Hidden nodes	Output nodes	Total Error	TE/Sample (sum of 6)
160	960	22	960	15696	98.10
80	480	31	480	7775	97.19
40	240	240	240	3519	87.98
40	240	255	240	1974	49.35
40	240	480	240	1995	49.88