NEURAL NETWORK CONTROLLER DEVELOPMENT FOR A MAGNETICALLY SUSPENDED FLYWHEEL ENERGY STORAGE SYSTEM

Roger L. Fittro
University of Virginia
Charlottesville, VA

Da-Chen Pang
Davinder K. Anand
University of Maryland
College Park, MD

ABSTRACT

A neural network controller has been developed to accommodate disturbances and nonlinearities and improve the robustness of a magnetically suspended flywheel energy storage system. The controller is trained using the back-propagation-through-time technique incorporated with a time-averaging scheme. The resulting nonlinear neural network controller improves system performance by adapting flywheel stiffness and damping based on operating speed. In addition, a hybrid multi-layered neural network controller is developed off-line which is capable of improving system performance even further. All of the research presented in this paper was implemented via a magnetic bearing computer simulation. However, careful attention was paid to developing a practical methodology which will make future application to the actual bearing system fairly straightforward.

INTRODUCTION

Artificial neural networks are massively parallel systems of densely interconnected simple processing elements which work together to adaptively produce a complex input/output functional relationship through a learning process. Since they were developed based on the understanding of how the human brain functions, physically neural networks can be visualized as a very simple model of the massive interconnections of neurons which make up the human brain.

Neural networks have been developed over the past 40 years. In the 1950's, research was carried out utilizing neural networks composed of individual neurons (nodes) called perceptrons. A considerable amount of progress was made with these networks and associated learning algorithms; however, they were only capable of learning linear relationships [1,2]. Because the ability to learn and represent nonlinear relationships is highly desirable, in ensuing years many researchers pursued methods to surpass the capabilities of perceptrons. In the 1960's, for example, Bernard Widrow developed networks composed of adalines and madalines which improved the overall capabilities of neural networks even in the area of representing nonlinear functions [1]. However, a practical, universally applicable network and training scheme which could accurately learn highly nonlinear relationships alluded Widrow and others. It was not until 1974 that the long awaited breakthrough was made by Werbos [3]. His work was popularized by Rumelhart, et al. [4] in 1986 and is now commonly know as back-propagation. Since that time,

many researchers have made further advancements in artificial neural networks. Numerous new network configurations and training algorithms have evolved as well as countless practical applications [1,2].

The building blocks (nodes) of all artificial neural networks perform a very simple two-step procedure. First, the inputs to a node are multiplied by independent weighting factors and then added together. Second, this weighted sum is passed through some sort of function, usually a nonlinear sigmoid function (see Figure 1). By connecting a large number of these elements together and choosing the correct weighting factors, very complex input/output relationships can be represented. However, to successfully achieve this goal, an appropriate network configuration and training algorithm are necessary.

Primarily there have been two main neural network configurations which have developed over the years. First, there are feed-forward networks which are composed of nodes fully interconnected from one layer to the next, beginning with the input and culminating at the output layer (see Figure 2a). Recurrent networks are the second predominant type of network configuration. In these networks, all or a portion of the nodes are fully interconnected to every other node and a few or all of the nodes are chosen as inputs and/or outputs (see Figure 2b).

In order to choose the proper weighting values to correctly represent a given relationship, a network learning technique is necessary. Neural network learning algorithms can also be broken down into two main categories: supervised and unsupervised. In supervised learning, desired input/output pairs are provided and the network adapts in such a way as to learn the associated relationship. Unsupervised learning, on the other hand, does not have desired input/output pairs available to learn from. Instead, a performance criterion is utilized to judge whether the correct relationship has been learned to a desired accuracy.

NEURAL NETWORK CONTROL SYSTEMS

Neural networks have been successfully implemented as system controllers in a number of different ways. Primarily, existing control system designs have been utilized with neural networks replacing various system components. For example, neural networks have been used as inverse plant controllers, self-tuning regulators, and as part of model reference adaptive control systems [5-7]. For this research, the existing PD controller in the magnetic bearing system was chosen to be replaced with a neural network. A PD-like neural network control system can be developed by feeding time-delayed inputs and direct inputs into the input nodes of a neural network. With these inputs, a derivative can be developed as well as a nonlinear proportionality resulting in a PD-like controller configuration.

Specific research in the area of neural network controller development for magnetic bearings has been limited to date. The main contribution has come from the Swiss Federal Institute of Technology. Researchers there have completed two stages of research. First, they have successfully developed a neural network controller capable of suspending a one degree-of-freedom iron sphere computer simulation. Second, they have experimentally suspended a single degree-of-freedom floating ball using a control system consisting of a standard linear controller and a neural network running in parallel [8]. This research produced significant results and provided the motivation to pursue further developments in this area. There are, however, a number of limitations in this research that need to be overcome. First, only a very small scale application was dealt with (i.e. mass = 0.013 kg). Second, only self-suspension was investigated. Rotation, especially at high speeds, produces a great deal of complications. And third, inefficient training algorithms based on random adaptation methods were used. The inherent instability of magnetic bearings makes these methods much less than ideal. Therefore, in order to address these areas and

pursue the development of a neural network controller for magnetic bearings further, a more thorough analysis is considered in this research.

MAGNETIC BEARING FLYWHEEL ENERGY STORAGE SYSTEM

The University of Maryland has developed a combination electro/permanent magnet bearing system for use in flywheel energy storage. The design consists of a motor/generator sandwiched between two pancake magnetic bearings which support a composite flywheel (see Figure 3). Each degree of freedom in the system is controlled by an independent controller, and a SISO model of the pancake magnetic bearing and control system is shown in Figure 4. In the bearing, a position transducer senses the position of the flywheel and generates a control signal to drive the electromagnetic coils producing the appropriate stabilizing force. Unfortunately from a system design standpoint, the resulting control and stabilization of the system based on linear control theory is not robust due to a number of nonlinearities and disturbances [9]. The following nonlinearities and disturbances were incorporated into the magnetic bearing computer simulation which was developed using Butcher's fifth-order Runge-Kutta method [10]. First, the nonlinearity associated with the power amplifier saturation was included. The nonlinear relationship (K_i) between the current supplied to the electromagnets and the resulting corrective force (F_c) was also taken into account in the analysis. (see Figure 5) And finally, the nonlinearity associated with the touch-down gap due to the back-up mechanical bearing was considered. The disturbances included in this research are those resulting from the mass imbalance of the flywheel, geometric error due to manufacturing and assembly tolerances, and error attributable to the sensors.

NEURAL NETWORK CONTROLLER TRAINING TECHNIQUE

Out of all of the numerous training techniques developed for neural networks, only a fraction of them are capable of addressing the unique problems encountered in controller training. A number of candidates were evaluated [2], and the back-propagation-through-time method was chosen because of its relative simplicity and proven performance.

The Back-propagation-through-time technique is based on the well documented learning algorithm: back-propagation [1,2,4]. Back-propagation is a supervised training technique which performs a gradient descent search for the optimal network weights. In control system applications, the proper input/output relationship necessary to produce the desired response of the plant is to be learned. Since back-propagation is a supervised learning technique which requires sample input/output pairs of this unknown relationship, it cannot be used directly as a controller training method.

In D.H. Nguyen and B. Widrow's paper [5], the following two-step procedure was outlined to circumvent this problem. First, a neural network is trained to emulate the plant. To train this network, appropriate plant input/output pairs need to be developed. Using these pairs, the network can very easily be trained via back-propagation or some other technique. This procedure closely parallels the plant identification procedure commonly performed in linear control system design. The second step of the procedure is developing a relationship between the plant output error and the error in the control signal. Because the plant emulator is a neural network, the error associated with the plant output can be back-propagated through it. This back-propagated error produces the desired relationship between the output and the control signal errors. The control signal error in turn can be back-propagated through the controller and used to adapt the controller weights

appropriately (see Figure 6). Because the plant output error cannot be back-propagated through the plant, the neural network emulator had to be developed.

The exact step by step desired plant output is not known for the magnetic bearing system. Therefore, an operating range is used as a more general system constraint in this work. The neural network controller and plant are allowed to progress unaltered for a predetermined period of time or until the operating range is exceeded. After this time, a performance criterion is calculated and used to back-propagate the associated plant error through the neural network emulator and controller. Since the controller and plant progressed through a number of time steps before resulting in the output error and since the final controller output is not solely responsible for the final state of the plant, this back-propagation procedure is continued through as many time iterations as the system progressed in the forward mode. In this way, there is an error and weight update associated with each controller output. It is this back-propagation of the output error through numerous time steps that gives back-propagation-through-time its name. A graphical representation of this procedure can be seen in Figure 7, where the C's, E's, and P's represent the neural network controller, emulator, and plant respectively.

MAGNETIC BEARING SYSTEM IMPLEMENTATION

For the magnetic bearing system, the plant was defined as the power amplifier, voltage supply, actuator coils, flywheel dynamics, and position transducer (see Figure 4). The remainder of the system was replaced by the neural network controller.

Emulator Development

The first step in the back-propagation-through-time technique is the training of a neural network plant emulator. In order to train the emulator, the appropriate inputs and outputs for modeling the plant need to be determined. Utilizing discrete linear control system design representation, the plant's dynamics in state-space notation can be expressed as:

$$x(k+1)=\mathbf{A}x(k) + \mathbf{B}u(k). \qquad (1)$$

Utilizing this representation, the present state of the plant and the plant inputs were chosen as the inputs to the neural network emulator. The network outputs were chosen to be the future states of the plant. The appropriate plant states were determined to be the coil current and the position and velocity of the flywheel as sensed by the position transducer. The control signal and disturbance force were chosen as the plant inputs. The corresponding inputs to the emulator are the control signal, disturbance force, and present states of the plant (coil current, position, and velocity). The outputs are the resulting states of the plant one time step later.

Because the magnetic bearing system is inherently unstable, it is not practically possible to obtain the sample input/output pairs necessary for training the emulator without having a controller in place. It would also be very expensive and impractical to build a controller just to obtain the data necessary to develop another control system. Therefore, an alternate method of training the emulator is necessary. One of the practically useful characteristics of the back-propagation-through-time technique is its relatively high tolerance to emulator error. Because it is a one-step predictor, as opposed to a multi-step predictor, it only requires relatively accurate modeling over one time step. Linearization of a system over a single time step produces satisfactory accuracy in most cases. Therefore, sample I/O pairs were generated based on an easily developed linearized model of the plant. Following this approach, a suitable magnetic bearing emulator was trained. A

conjugate gradient learning algorithm was used for training the emulator instead of back-propagation because of its superior speed and accuracy. These advantages can be demonstrated by comparing the results obtained from learning a sine wave. The conjugate gradient method learned the proper relationship approximately 17 times faster while achieving an RMSE value one order of magnitude better than the back-propagation routine. When applied to training the emulator, the conjugate gradient method achieved an RMSE value of 3.17×10^{-8} on a training set consisting of 600 I/O pairs. The resulting network was subsequently testing on a set of 400 I/O pairs producing a comparable RMSE value of 3.94×10^{-8}.

Controller Training

Because network training is a trial and error procedure and the magnetic bearing is an inherently unstable system, bearing damage is very possible due to excessive failures. To overcome this problem, a two-stage training procedure was implemented. The first stage was to train the network to perform self-suspension. Once this was accomplished, the bearing was allowed to rotate at increasingly higher speeds while the controller continued to learn. If the speed is not increased too quickly and appropriate operating ranges are chosen, learning is able to continue without any failures taking place after self-suspension is accomplished. This is because learning takes place when the designated operating range is exceeded; and if this range is chosen carefully, the system will remain stable throughout this phase of training.

Before this two-stage training procedure was started, one more concern was dealt with. Neural networks produce an instantaneous response while a time delay exists in the actuator coil. This combination results in an overly sensitive controller which in turn causes system instability. The neural network continually attempts to instantaneously bring the system to a point of equilibrium without being influenced in any way by previous outputs. Therefore, the network does not allow sufficient time for the actuator to respond properly to counteract the destabilizing force before a new and possibly contradictory command is given.

Based on this insight, a time-averaging scheme for the neural network control signal was developed. This time-averaging method takes the instantaneous output of the network and averages it with previous outputs over a specified time period. This produces a control signal which takes into account previous outputs and also causes a smooth signal to result. As long as the time-averaging period is chosen appropriately, this technique tends to produce a stable system (see Table I).

A number of different network configurations were successfully trained using the modified back-propagation-through-time technique. The performance criterion (E) used during training was

$$E = \alpha_1 x + \alpha_2 x; \tag{2}$$

where α_1 and α_2 were calculated based of the emulator such that the position and velocity of the flywheel had comparable effects on the weight updating procedure. The values calculated were as follows: $\alpha_1 = 1.0$ and $\alpha_2 = 0.00036$ [2].

In the first stage of controller training, self-suspension, the networks learned quickly allowing less than ten failures on average. The operating range specified for this stage was the entire touch-down gap (0.00015 m). The resulting neural networks were capable of controlling the system at 1000 rpms. Operating at this speed, the networks went through an additional training procedure, after which the weights were fixed. For this stage of training, an operating range of 0.00002 m

was specified in conjunction with a moderate mass imbalance corresponding to a sinusoidal disturbance of 10 Newtons. A summary of the training run for the network with the best performance is shown in Table II.

The best performing controller unexpectedly turned out to be a single-layer network. Even though theoretically a nonlinear controller should be able to produce better results than a linear controller, this is only true if the proper nonlinear function is developed. It turns out that an optimal control system for a magnetic bearing has a nonlinear relationship with respect to flywheel rotational speed. This means that in order for a controller training algorithm to produce an appropriate nonlinear network, a relationship involving the rotational speed of the plant needs to be included. This was not the case for this training procedure; therefore, the resulting nonlinear relationships developed by the multi-layered neural networks cannot be expected to be correct, and neither can they be expected to out perform a single-layer network. This analysis is also confirmed by the results of Krodiewski, et al [11].

In addition to the training of these networks using back-propagation-through-time, a hybrid multi-layered neural network controller was developed off-line utilizing experimental results of the previously trained networks. In this network (see Figure 8), the speed of the flywheel and a + 1 bias act as the only inputs. Experimentally determined controller stiffness and damping factors appear as weights in the second layer. These cause the effective stiffness and damping values to vary based on the speed of the flywheel (see Figure 9). The outputs of the third layer of nodes are the effective stiffness and damping for the given speed. In the fourth layer of nodes, these values are taken as inputs in addition to the flywheel's position and velocity. This layer of nodes is made up of two product units which multiply the inputs together rather than calculating a weighted sum. Finally, the outputs of these two nodes are fed into the fifth layer of nodes producing the control signal (v_o).

Performance Results

Comparison of the best performing trained neural network controller (NN) and the linear control system (LCS) produced another unexpected result. As can be seen from Figure 10, the performance envelopes for the linear controller and the neural network are very similar in the low to mid-speed range. However, in the higher speed range, the neural network performance far exceeds that of the linear controller. A single-layer neural network is only capable of producing a linear relationship; therefore, the question as to why it is able to outperform a linear control system design needs to be answered. Upon closer analysis, it turns out that the neural network in conjunction with the time-averaging scheme is able to produce a nonlinear relationship with respect to the rotational speed of the flywheel. At low speeds, the time-averaging period is negligible compared to the time period associated with the mass imbalance. However, as the flywheel speed increases, the time period of the mass imbalance related disturbance force decreases. Meanwhile, the time-averaging period remains fixed. As the speed of the flywheel increases above a certain level, the time-averaging period becomes significant and causes lower effective controller stiffness and damping. This phenomenon becomes more pronounced as the flywheel speed increases, and results in the desired nonlinear control system relationship necessary to improve performance. Hence, the combination of the trained single-layer neural network controller and the time-averaging technique actually results in a nonlinear rather than a linear controller. Further performance improvements can be seen to be achieved through the use of the hybrid multi-layered neural network controller (HMLNN).

The disturbance force due to a mass imbalance can be written in the form:

$$F_d = m\, e\, \omega^2 \sin(\omega t); \quad (3)$$

where m is the mass of the flywheel and e is the mass imbalance distance. Using this relationship, new performance curves can be generated. As can be seen from Figure 11, performance measured by this new index is also improved through the use of the neural network controllers. This result is very significant because it demonstrates that the neural network is much more likely to be able to control the system at higher speeds. It also suggests that comparable performance can be obtained with a neural network controlled system subjected to reduced manufacturing tolerances and a system using a linear controller and much tighter tolerances.

CONCLUSIONS/RECOMMENDATIONS

In this research, the historical developments and basic operating principles of artificial neural networks have been discussed. The back-propagation-through-time technique has been described and modified to produce a very practical controller training technique for the magnetic bearing system. The training procedure has been implemented on a computer simulation, and the neural network controllers were successfully trained. The resulting system performance characteristics were compared with those of the existing linear control system. Significantly improved performance was achieved for both the single-layer and hybrid multi-layered neural network controllers. These improvements demonstrate the advantages of using neural networks in control system design especially when the desired controller response is nonlinear.

In order to produce even better performing nonlinear controllers, the developments of this research can be extended. The on-line training technique could be modified to include the rotational speed of the flywheel as a training parameter so that the appropriate nonlinear relationship would be learned by a multi-layered neural network. The inclusion of the coil current as an input to the control system should also be investigated. This added input may be able to improve controller performance for both standard linear control systems and neural networks. In addition, an even more accurate computer simulation could be developed. The simulation used in this research includes most of the important system characteristics. However, there are a few areas that still need to be addressed [9]. At high speeds, a number of factors begin to play a significant role. For example, gyroscopic effects at high speeds make the assumption of independent control axes incorrect. Material deformation at high speeds also alters system performance due to growth of the air gap. Additionally, magnetic material properties alter due to eddy currents and hysteresis. Therefore, all of these characteristics need to be taken into account in order to obtain an accurate model. However, no simulation is ever perfect. Therefore, controller training should be performed on the actual magnetic bearing flywheel energy storage system in order to practically prove and improve on the results obtained through simulation.

REFERENCES

1. Dayhoff, J., Neural Network Architectures: An Introduction, Van Nostrand Reinhold, 1990.

2. Fittro, R., "Neural Network Controller Design for a Magnetic Bearing Flywheel Energy Storage System", Master's Thesis, University of Maryland, College Park, MD, 1993.

3. Werbos, P., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences", Ph.D. Thesis, Harvard University, Cambridge, MA, 1974.

4. Rumelhart, D., Hinton, G., Williams, R., "Learning Internal Representation by Error Propagation", In Rumelhart, D., McClelland, J., editors, Parallel Distributed Processing - Explorations in the Microstructure of Cognition, chapter 8, pp. 318 - 362, MIT Press, 1986.

5. Nguyen, D., Widrow, B., "Neural Networks for Self-Learning Control Systems", IEEE Control Systems Magazine, April, 1990.

6. Chen, F., "Back Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control", IEEE Control System Magazine, April, 1990.

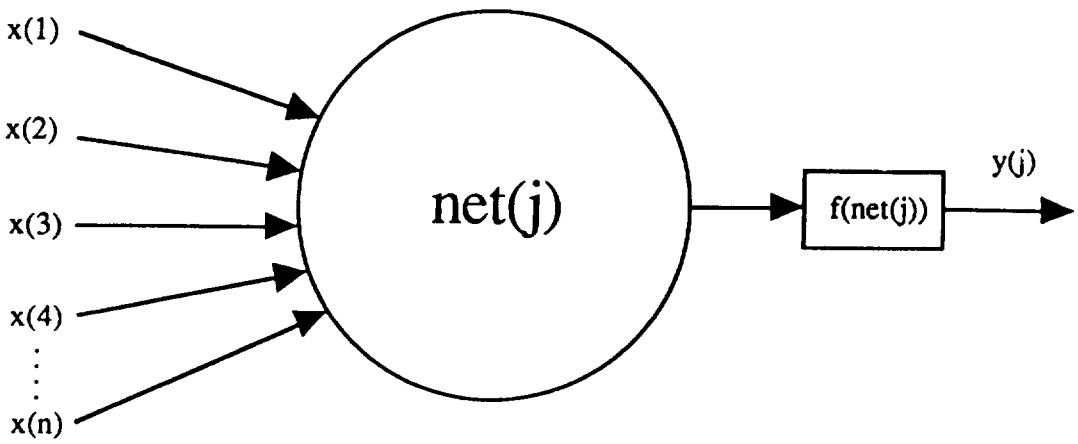
7. Narendra, K., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Trans. on Neural Networks, Vol. 1, No. 1, 4 - 27, March 1990.

8. Bleuler, H., Diez, D., Lauber, G., Meyer, U., Zlatnik, D., "Nonlinear Neural Network Control with Application Example", International Neural Network Conference (INNC), 1990.

9. Johnson, R.G., Pang, D., Kirk, J.A., Anand, D.K., "Physical Modeling of High Speed Magnetic Bearing Systems", International Symposium on Magnetic Bearings, 1992.

10. Chapra, Canale, Numerical Methods for Engineers with PC Applications, McGraw Hill, 1985.

11. Krodiewski, J., Zmood, R., Kirk, J., Lashley, C., "Influence of Magnetic Bearing Operating Constraints on Rotor-Bearing System Performance", Proc. of the 27th Intersociety Energy Conversion Engineering Conference, Vol. 4, pp. 15 - 22, 1992.



$$net(j) = \sum_i w(i,j)\, x(i)$$

$$y(j) = f(net(j))$$

Figure 1. Single neural network node and equations.

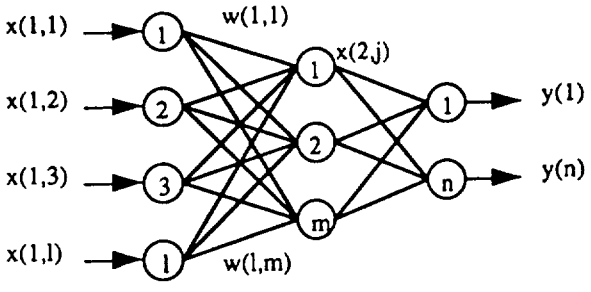


Figure 2a. Feed-forward network.

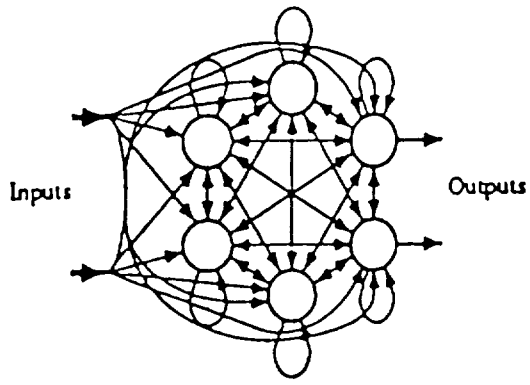


Figure 2b. Recurrent network.

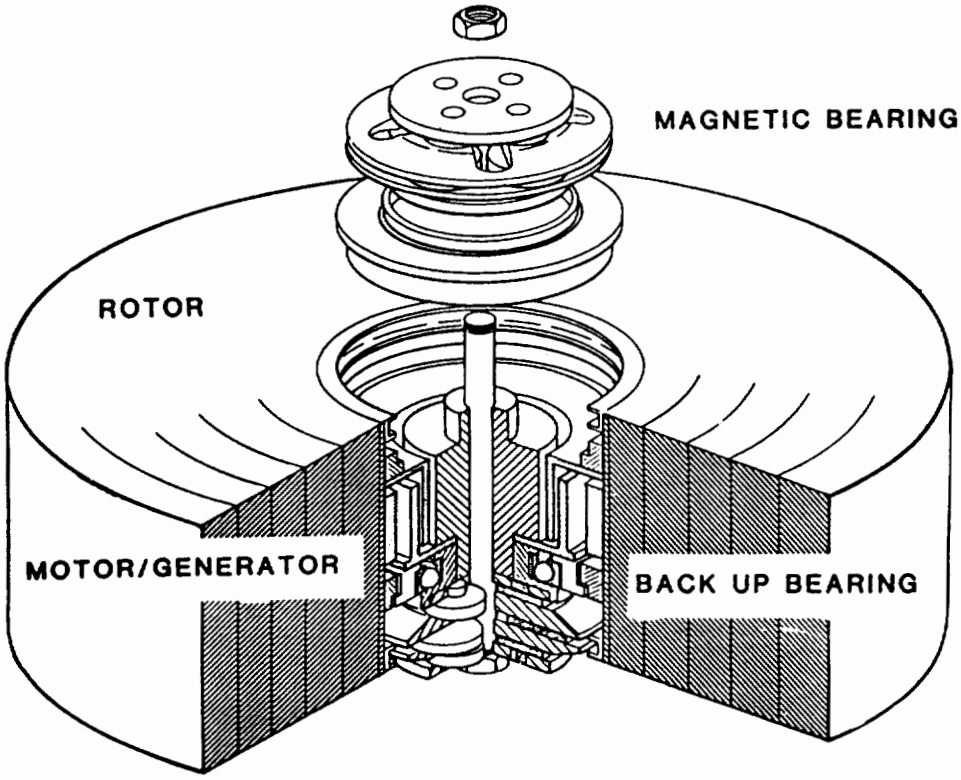


Figure 3. Stack design magnetic bearing flywheel energy storage system.

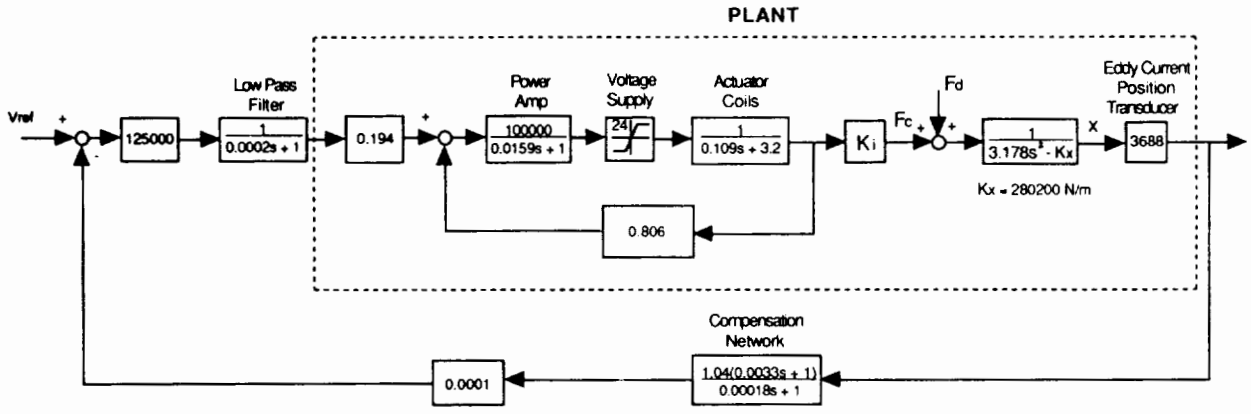


Figure 4. Nonlinear pancake bearing system block diagram.

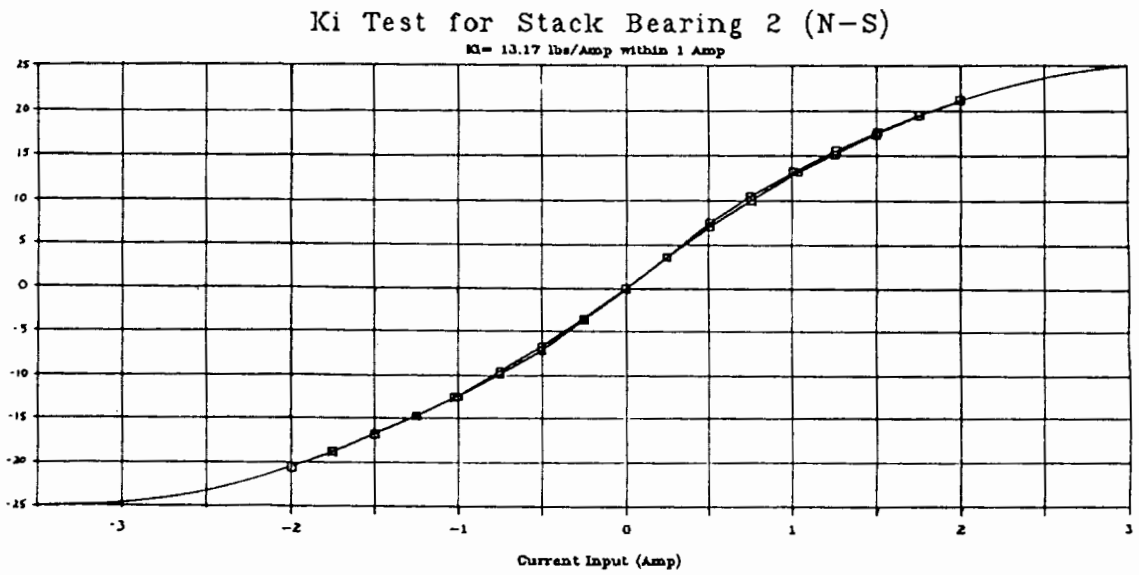


Figure 5. Current/Force Sensitivity.

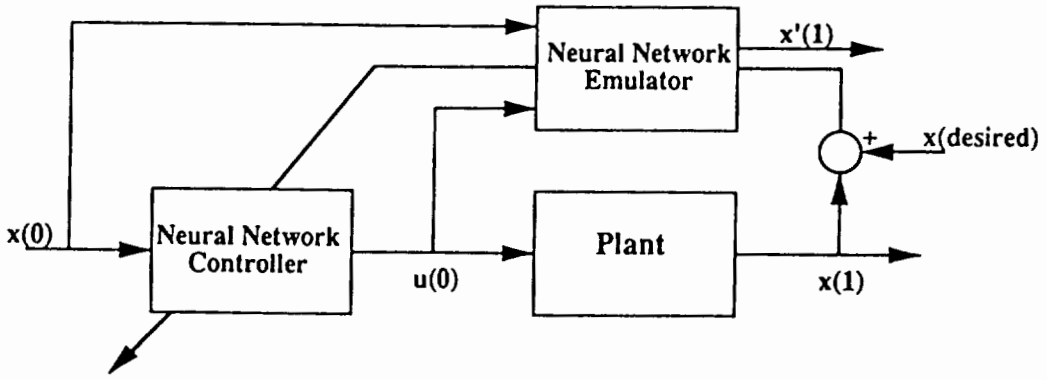


Figure 6. Single step of the back-propagation-through-time procedure.

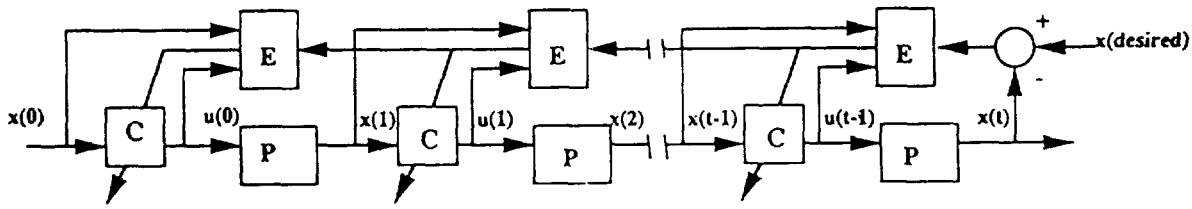


Figure 7. Back-propagation-through-time procedure.

Table I. Time-averaging Period Results

Time-averaging Period	Results
0.0002 sec.	unstable
0.0005 sec.	stable - better performance at low to medium speeds
0.0010 sec.	stable - better performace at higher speeds

Table II. Neural Network Controller Training Results

Training	Network weights (normalized)	Adjusted weights
Self-suspension	w(1,1) = -0.636254 w(1,2) = -0.455918	w'(1,1) = -6.36254 w'(1,2) = -0.022796
1000 rpm	w(1,1) = - 1.31773 w(1,2) = - 0.777021	w'(1,1) = - 13.1773 w'(1,2) = - 0.038851

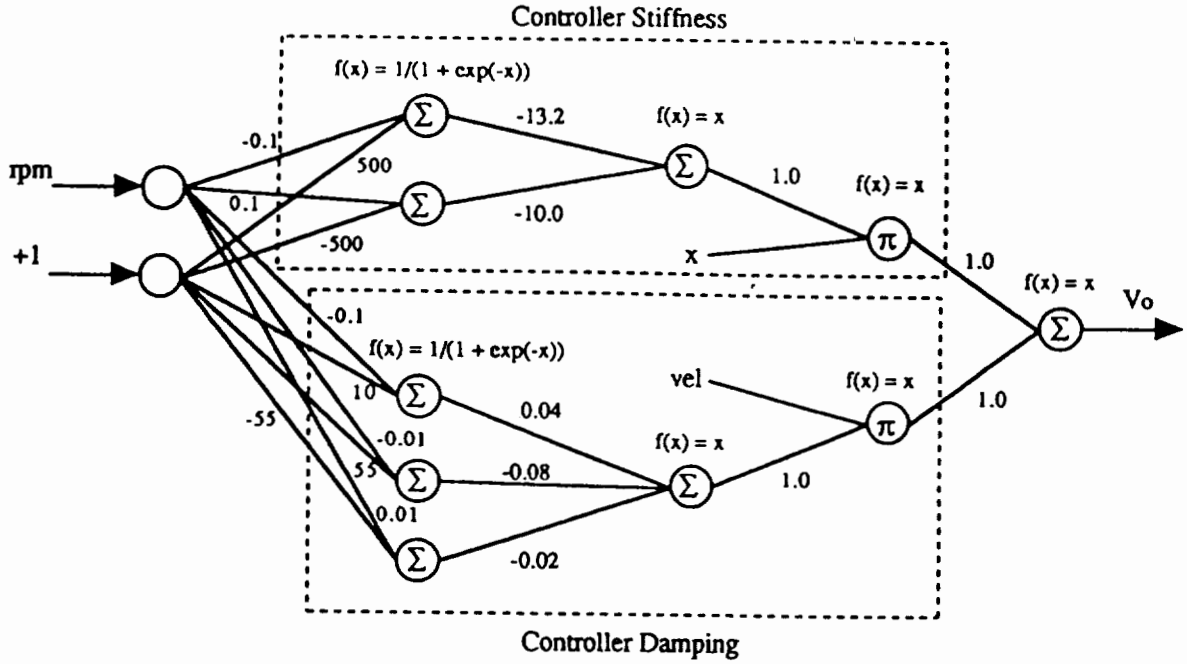


Figure 8. Hybrid multi-layered neural network controller.

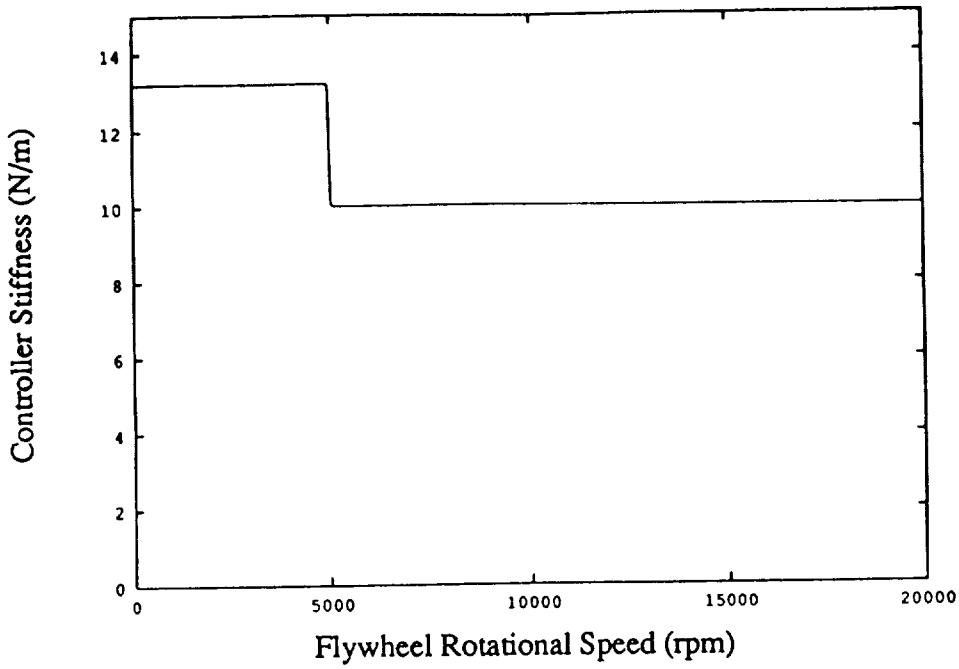


Figure 9a. Neural network controller stiffness.

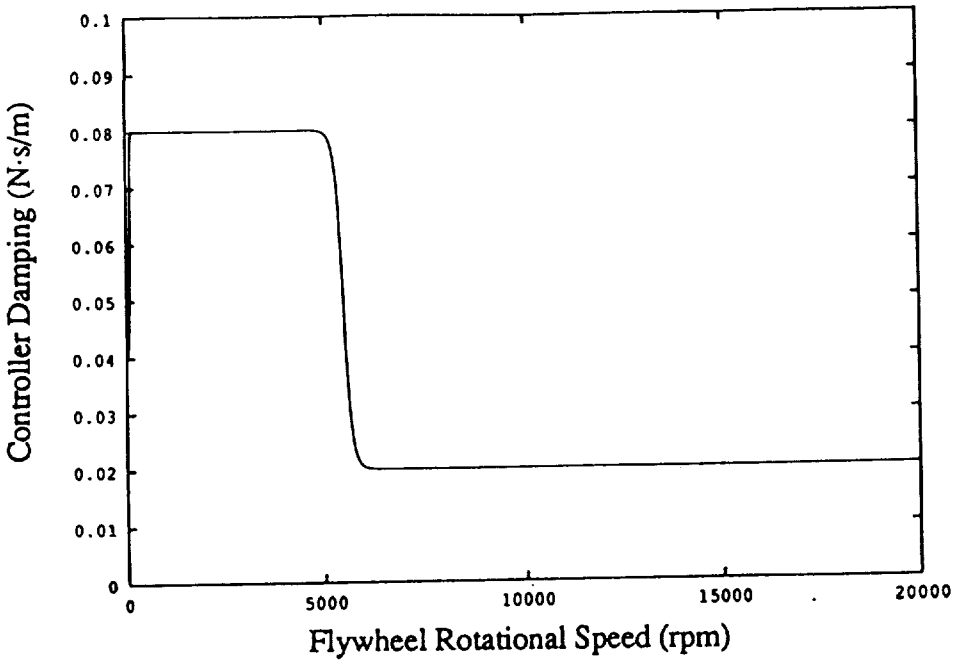


Figure 9b. Neural network controller damping.

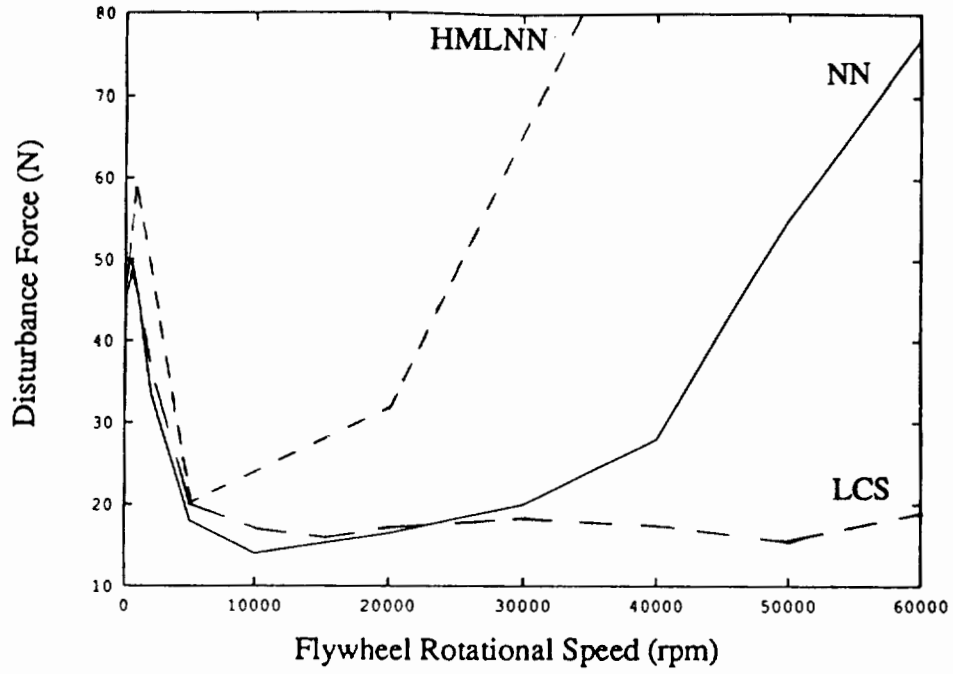


Figure 10. Control system performance.

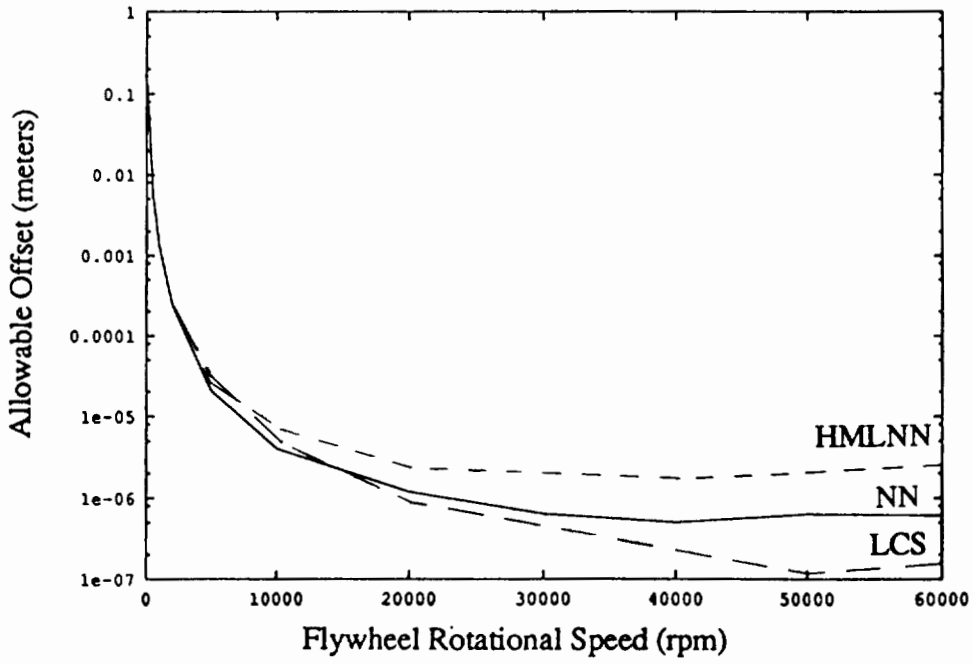


Figure 11. Control system performance.