1-1-2010

Experimental implementation of artificial neural network-based active vibration control & chatter suppression

Yong Xia
Ryerson University

EXPERIMENTAL IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK-BASED ACTIVE VIBRATION CONTROL & CHATTER SUPPRESSION

by

Yong Xia

MASc – Mechanical Engineering, Ryerson University, Canada, 2005

BEng – Instrumentation, Tianjin University, China, 1992

A dissertation

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Program of

Mechanical Engineering

Toronto, Ontario, Canada, 2010

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institution or individuals for the purpose of scholarly research.

Signature: ______________________

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature: ______________________

ABSTRACT

EXPERIMENTAL IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK-BASED ACTIVE VIBRATION CONTROL & CHATTER SUPPRESSION

Yong Xia
Doctor of Philosophy, 2010
Mechanical Engineering, Ryerson University

Vibration control strategies strive to reduce the effect of harmful vibrations such as machining chatter. In general, these strategies are classified as passive or active. While passive vibration control techniques are generally less complex, there is a limit to their effectiveness. Active vibration control strategies, which work by providing an additional energy supply to vibration systems, on the other hand, require more complex algorithms but can be very effective. In this work, a novel artificial neural network-based active vibration control system has been developed. The developed system can detect the sinusoidal vibration component with the highest power and suppress it in one control cycle, and in subsequent cycles, sinusoidal signals with the next highest power will be suppressed. With artificial neural networks trained to cover enough frequency and amplitude ranges, most of the original vibration can be suppressed. The efficiency of the proposed methodology has been verified experimentally in the vibration control of a cantilever beam. Artificial neural networks can be trained automatically for updated time delays in the system when necessary. Experimental results show that the developed active vibration control system is real time, adaptable, robust, effective and easy to be

implemented. Finally, an experimental setup of chatter suppression for a lathe has been successfully implemented, and the successful techniques used in the previous artificial neural network-based active vibration control system have been utilized for active chatter suppression in turning.

ACKNOWLEDGEMENTS

The author would first and foremost like to thank his thesis supervisor, Dr. Ahmad Ghasempoor, for his guidance and support in this research.

The author would also like to thank the Department of Mechanical and Industrial Engineering and the School of Graduate Studies of Ryerson University for their support in terms of department facilities, research stipend and scholarship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

NOMENCLATURE

x	input signal
Y	output signal
e	error signal
d	disturbance signal
p	primary disturbance
u	control disturbance
w	weight
b	bias
v	activation potential
t	target output
μ	adaptation parameter/step-size/convergence factor
Z^{-1}	a unit sample of delay
E	error criterion/total error/cost function
α	learning rate parameter or step size in ANN
$\lambda_{\max}$	maximum curvature, or the maximum eigenvalue of Hessian matrix
c_i^*	coefficients of the estimated FIR filter model of the forward path
ζ	damping ratio
ω	natural frequency radians/sec
f	natural frequency in Hz
x_{scaled}	normalized value
$x_{\min}$	minimum value

$x_{\max}$	maximum value
V1	signal amplitude of the primary shaker
V2	signal amplitude of the control shaker
F1	signal frequency of the primary shaker
F2	signal amplitude of the control shaker
PD	phase difference between the input signal and the output signal
dB	decibel
AVC	active vibration control
DSP	digital signal processing
FIR	finite impulse response
IIR	infinite impulse response
LMS	least mean square
MSE	mean square error
RMS	root mean square
ANN	artificial neural network
TDL	tapped delay line
NNI	neural network identifier
NNC	neural network controller
A/D	analog-to-digital
D/A	digital-to-analog
ADC	analog-to-digital converter
DAC	digital-to-analog converter
FFT	fast Fourier transform

DAQ	data acquisition
NI	National Instruments Co.
VI	virtual instrument
FEM	finite element methods
RPROP	resilient propagation
PID	proportional-integral-derivative
S/N	signal-to-noise ratio
f_{Nx}	measured first natural frequency of the tool system in the horizontal direction
f_{Ny}	measured first natural frequency of the tool system in the vertical direction
f_x	horizontally oriented chatter frequency
f_y	vertically oriented chatter frequency
A_x	horizontally oriented chatter amplitude
A_y	vertically oriented chatter amplitude
P_x	horizontally oriented chatter phase
P_y	vertically oriented chatter phase
γ_{xy}^2	the coherence function of *x*-digital signal and *y*-digital signal
$(\gamma_{xy}^2)_{\max}$	the maximum of γ_{xy}^2
f_m	the corresponding frequency of the $(\gamma_{xy}^2)_{\max}$ value
T	a threshold
f_{min}	the lower end of possible chatter frequency range
f_{max}	the upper end of possible chatter frequency range

γ	the value of the chatter index
f_{xu}	horizontal control signal frequency
f_{yu}	vertical control signal frequency
A_{xu}	horizontal control signal amplitude
A_{yu}	vertical control signal amplitude
P_{xu}	horizontal control signal phase
P_{yu}	vertical control signal phase

CHAPTER 1

INTRODUCTION

1.1 Vibration Control

Vibration is present in many dynamic mechanical systems. Vibration control is the effort to reduce the negative effects of vibration effectively.

Two main groups of vibration control methods are passive and active methods [1, 2]. Other vibration control methods, e.g., semi-active and other hybrid methods, can be considered as a combination of passive and active methods.

1.1.1 Passive Methods

Passive vibration control methods include elimination of additional energy sources, eliminating or decreasing input forces and isolation from external disturbances [1]. Examples include balancing, reduction of mass, substitution of rolling bearings by sliding ones, etc. The parametric modifications lead mainly to changes of mass and stiffness elements. The structural modifications deal with introducing additional constraints to the system or modification of existing ones (i.e., continuity interruption of vibrator structure via introduction of intermediate elements), or addition of vibration eliminators [1]. Damping is also an important parameter but usually takes secondary consideration. It works by dissipation of mechanical energy that is converted to heat [1]. Therefore, it may cause the decrease of general working efficiency in some areas, e.g., for machines. The ideal device should run with minimal damping values. In cases when undesirable vibrations cannot be eliminated via structural or parameteric changes, damping may be introduced. Additional damping is usually provided by materials with appropriate damping values, frictional joints, additional dampers, etc.

Unfortunately, the use of passive vibration control methods is not effective because of inefficiency in the range of low frequencies, sensitivity on application conditions, lack of robustness, and increased size and weight. On the other hand, these methods allow dissipation of a great deal of vibration energy in the range of sufficiently high frequencies (energy dissipation increases with decrease of passive elements stiffness) [1, 2].

Better vibration control results may be achieved by using active methods.

1.1.2 Active Methods

Active vibration control (AVC) methods work by providing an additional energy supply to vibration systems. These methods alleviate the problems of contradictory requirements imposed on passive vibration control techniques, such as efficiency of device operation, low frequency vibration, dynamic stability, stiffness, etc. The additional energy supply can produce a force that compensates the forces that account for vibrations. To reduce the vibration, the additional energy should have appropriate power, frequency and phase, relative to existing vibration forces. AVC methods can change the system parameters in an active way. Some commonly utilized AVC methods include adaptive filter control, Proportional-Integral-Derivative (PID) control, fuzzy logic control and artificial neural network (ANN)-based control. As the vibration control method used in this research, the AVC method will be introduced in detail in chapter 2.

AVC is a highly multidisciplinary field with elements from structural dynamics, signal processing, materials science, mechanical vibration, actuator and sensor technology. The applications of AVC are also diverse. The following are some example areas for AVC application.

Chatter is a machining process instability resulting from self-excited vibration caused by the interaction of the chip removal process and the structure of the machine tool. Chatter occurs especially when machining flexible parts such as turbine blades, or when machining deep features using slender end mills. Chatter can limit metal removal rates (productivity), cause poor surface finish, and accelerate tool wear. There has been a considerable research effort concerned with the suppression of chatter. Applying ANN-based AVC in machining chatter suppression is discussed in much more detail later in this thesis.

Active noise control is an attractive means to achieve large amounts of noise reduction in a small package, particularly at low frequencies. Current applications include: control of aircraft interior noise by use of lightweight vibration sources on the fuselage and acoustic sources inside the fuselage; reduction of helicopter cabin noise by active vibration isolation of the rotor and gearbox from the cabin; attenuation of unavoidable noise in automobiles (electronic mufflers, engine mounts, and so on).

AVC can be used for suppressing the vibrations within a building generated by high winds or an earthquake, for reduction of sway in tall buildings and vibration of construction machinery, and to prevent large bridges from damage by flutter [1, 2].

AVC has also been used in vibration-sensitive machines. Examples include surgical microscopes, electronic equipment, lasers, MRI units, scanning electron microscopes, and computer disk drives.

AVC can also be used in transportation equipment, life sciences, marine, semiconductor industry, and many other areas as well.

1.1.3 Hybrid Methods

There are other vibration control methods that can be considered as a combination of passive and active methods. One of these methods is called semi-active vibration control and includes passive elements where elastic and damping forces may be changed depending on the conditions. Such methods require an external energy source with much lower power requirements.

Another hybrid method uses both active and passive elements. The active part operates in the low frequency range and the passive one in the higher frequency range.

Considering the superior capabilities of active methods over passive methods, in this research, AVC will be utilized. However, there are some technical challenges associated with the implementation of AVC. For example, adaptive filter control and PID control can only work effectively for linear vibration problems; most implemented artificial neural network (ANN)-based control methods rely on the plant output as the feedback signal and therefore have to solve the time-delay problem, which leads to the complexity of the control systems (see chapter 2 for details). This AVC research is working on dealing with these challenges.

1.2 Chatter Suppression

Machine tool vibration suppression techniques have been studied for many years because excessive vibrations often result in poor surface finish, reduced tool life and severe acoustic noise in work environment. Among those vibrations, chatter is the most problematic limiting factor of machining, especially with high spindle speeds and long reach cutters, such as a boring bar.

Chatter is a violent relative vibration between a cutting tool and a workpiece. Chatter falls into two categories, forced and self-excited. Forced chatter is due to the unbalance of rotating members, such as unbalanced driving system, servo instability, or impacts from a multi-tooth cutter. Self-excited chatter consists of two types, i.e., primary (or non-regenerative) type and regenerative type [3]. Regenerative chatter frequently occur in boring operation due to the low stiffness and low damping property of a slender boring bar itself.

The regenerative type of self-excited chatter is due to the interaction of the cutting force and the workpiece surface undulations produced by previous tool passes. Its amplitude increases with the progress of cutting [3]. Regenerative chatter is found to be the most detrimental phenomena in most machining process [4]. Hence, methods to suppress it have been the focus of many studies.

Generally, chatter suppression methods fall into two categories, passive and active methods. Other methods, e.g., semi-active and hybrid methods, can be considered as a combination of passive and active methods.

Passive methods include enhancing the system’s dynamic stiffness and damping, elimination of additional energy sources, eliminating or decreasing input forces and isolation from external disturbances [5].

Active methods work by providing an additional energy supply to vibration systems. To suppress chatter, the additional energy should have appropriate power, frequency and phase, relative to existing chatter forces. The actual active chatter suppression methods are diverse and some examples will be introduced in the next chapter.

1.3 Objective

The objective of the current work is to develop a robust real-time adaptable AVC system to detect and suppress the vibration of a cantilever beam, and utilize similar techniques used in the AVC system in an active chatter suppression (ACS) system for turning. The effectiveness of both systems is verified through experiments.

CHAPTER 2

LITERATURE REVIEW

2.1 Active Vibration Control

The research in active vibration control (AVC) has been expanding since Lueg's work in 1930s [6], and especially rapidly in the past three decades.

AVC is achieved by using a control source to introduce a secondary (control) disturbance into a system to cancel the existing (primary) disturbance, thus resulting in an attenuation of the original vibration [1]. These secondary sources are interconnected through an electronic system using a specific signal-processing algorithm for a particular cancellation scheme. To explain the concept of AVC, a classical application of active noise control is always used as an example as in [2]. The principle of destructive interference used is not only limited to the control of acoustic waves, but also successfully applied to the control of other vibration.

The classical application is the active control of sound waves in a small duct, which is shown in Figure 2-1. An actuator (a loudspeaker) and two sensors (two microphones in this case) are used. Sensor A, which is called a reference sensor, is used to measure the advanced information on the disturbance sound wave that propagates in the duct, and sensor B, which is called an error sensor, is used to monitor the performance of the active sound control system, thus providing feedback to a control algorithm. To keep this example simple, sensor A is assumed to be not coupled with the actuator, so sensor A only measures the disturbance sound field. The control structure of Figure 2-1 is called "feedforward", because the controller feeds the actuator with a signal based on the advanced information obtained from sensor A. If the controller works properly, the signal sent to the actuator will generate a sound wave, which will cancel the disturbance sound wave at the location of sensor B. Because only plane wave propagation is considered in such a small duct commonly, the sound field will be uniform in any section of the duct, and the sound will be reduced from sensor B to the end of the duct. Details of active sound control theory and applications can be found in [2].

Compared with AVC, passive vibration control methods suppress vibration by using energy absorbing dampers to consume energy input or by changing the system structures or conditions to reduce the energy inputs or generated energy.

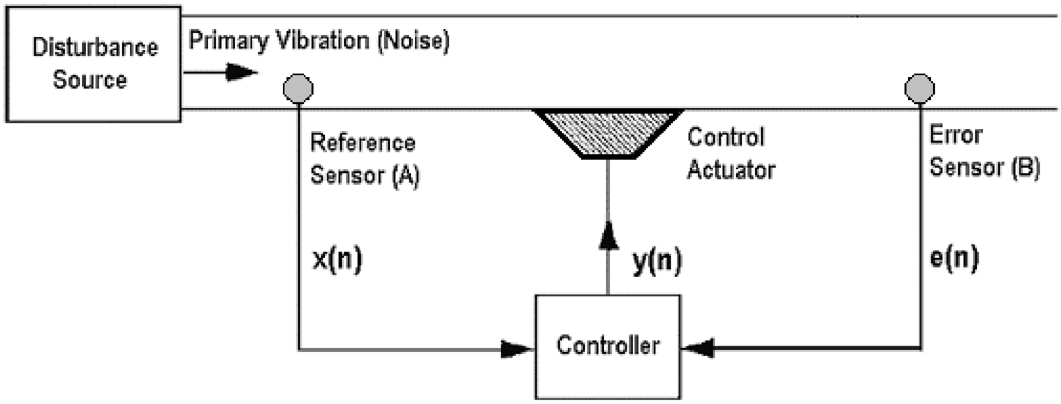


Figure 2-1: Active control of sound waves in a duct [2]

The concept of AVC has been known for more than 60 years. The basic ideas of AVC were proposed in 1936, when Paul Lueg first described the design of active noise control in a patent published in the United States. Even though the concept is simple, it is only with the development of low-cost fast digital signal processing (DSP) systems during the last 20 years that the implementation of practical active sound and vibration control systems has become feasible. It is desirable for the vibration canceller to be digital, where signals from electroacoustic or electromechanical transducers are sampled and processed in real time using DSP systems. Digital technology is well suited for adaptive control systems or control configurations where a lot of precision is required in order to achieve good performance.

The continuous progress of AVC involves the development of improved adaptive signal processing algorithms, transducers, and DSP hardware. Since the characteristics of the vibration source and the environment are time varying, the frequency content, amplitude, phase, and vibration velocity of the undesired vibration are not stationary. An AVC system must therefore be adaptive in order to cope with these variations. AVC systems can be used in linear or nonlinear control problems.

For linear control problems, adaptive filters combined with some algorithms are always used. Adaptive filters adjust their coefficients to minimize an error signal and can be realized as finite impulse response (FIR), infinite impulse response (IIR), lattice, and transform-domain filters [6]. The FIR filter is also called a transversal filter and the IIR filter is also called a recursive filter. The algorithms used for adaptive filters are generally based on gradient descent algorithms. Examples are least-mean-square (LMS) and filtered-X LMS algorithms. For nonlinear control problems, many researchers have focused on neural networks combined with some specific algorithms, such as the standard gradient descent backpropagation algorithm. Moreover, neural networks proved experimentally to be robust for not only nonlinear control but also linear control [7]. Some commonly used architectures and algorithms in AVC will be introduced in a latter section.

More sophisticated algorithms allow faster convergence and greater vibration suppression and are more robust with respect to interference. A good review of the different control techniques that have been used for the active control of sound and vibration can be found in [6, 8-11]. The development of improved DSP hardware has allowed these more sophisticated algorithms to be implemented in real-time to improve system performance.

2.2 Control Strategies

Basically, in the area of AVC, there are two kinds of control strategies: feedforward and feedback [1, 2, 6].

2.2.1 Feedforward Control

The noise suppression example in the preceding section is a feedforward control strategy application. The principle of feedforward control is presented in Figure 2-2. Feedforward controllers rely on the availability of a reference signal correlated to the primary disturbance. This signal is passed through an adaptive controller. The output of the controller is applied to the system by secondary sources. The filter coefficients are adapted in such a way that the error signal at one or several critical points is minimized. The idea is to produce a secondary disturbance such that it cancels the effect of the primary disturbance at the location of the error sensor.

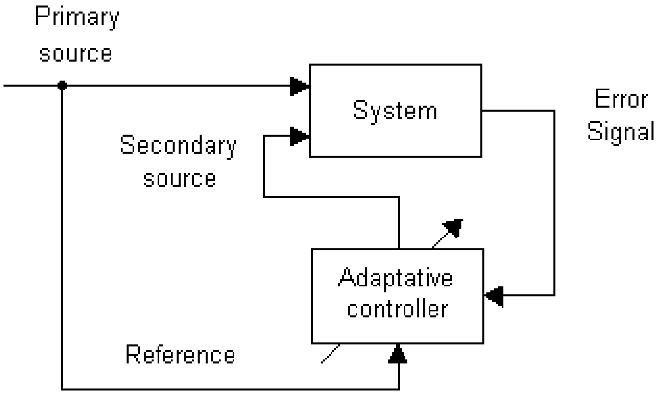


Figure 2-2: Principle of feedforward control [12]

One important point to note is that the signal must be received by the controller in sufficient time for the required control signal to be generated and output to the control/secondary source when the disturbance (from which the reference signal was generated) arrives. Systems, for which the active control system produces the control signal at the downstream location at the same time that the primary signal arrives, are referred to as “causal”. Causality is a condition that all feedforward designs must satisfy if the vibration to be controlled is not periodic [8]. As will be discussed in section 2.3, the reason why most transfer function models utilize tapped delay lines is to enable the modeling of the explicit system time delays to maintain causality within the control schemes. If the vibration to be controlled is periodic, it is possible to get a similar outcome without satisfying causality if the control signal output and the arriving disturbance have a phase difference equal to an integer number of periods.

Another point to note is that there is no guarantee that the global response is also reduced at other locations and, unless the response is dominated by a single mode, there are places where the response can be amplified; the method can therefore be considered as a local one, in contrast to feedback which is global (as will be discussed in the following section).

2.2.2 Feedback Control

Feedback control is a control system which monitors its effect on the system it is controlling and modifies its output accordingly. The principle of feedback control is presented in Figure 2-3. The output *y* of the system is compared to the reference input *x* and the error signal, *e* = *x* - *y*, is passed into a compensator *H(s)* and applied to the system *G(s)*. The design problem consists of finding the appropriate compensator *H(s)* such that the closed-loop system is stable. Feedback on how the system is actually performing allows the compensator (controller) to dynamically compensate for disturbances, *d*, to the system.

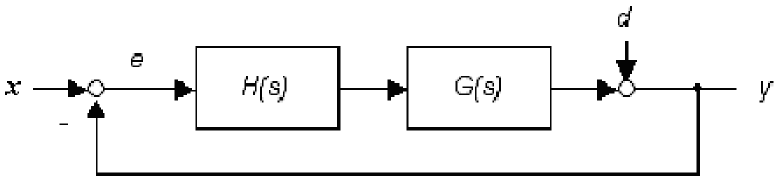


Figure 2-3: Principle of feedback control [12]

2.2.3 Feedforward vs. Feedback

Generally, a feedforward control system should be implemented whenever it is possible to obtain a suitable reference signal because of its inherent stability characteristics and usually superior performance to a feedback control system [1, 6]. The use of feedback control has been popular in the area of structural vibration because of its ability to damp structural vibrations without the need to be able to measure a reference signal in advance [1, 2, 6, 8].

There is no direct connection between the output of the system and the actual conditions encountered in feedforward control, which means the system does not and cannot compensate for unexpected disturbances. Moreover, a suitable reference signal is not always available.

2.3 Control System Design

Many of the electronic systems utilized in control schemes derive control inputs for secondary sources (as in Figure 2-2) via modified adaptive signal processing architecture/algorithm combinations. Since the characteristics of input signal and system response are unknown, or may be slowly changing with time, the controller must "learn", and be able to "relearn" to cope with changes in the signal structure. Basically, there are two kinds of such architecture/algorithm combinations: linear filter combinations and nonlinear combinations.

2.3.1 Linear Filter Architecture/Algorithm Combinations

Many of today's implementations of active control use adaptive linear filtering techniques. With the advances in digital technology over the past several decades, adaptive DSP has become a firmly established field, encompassing a wide range of applications. One of the most common forms of adaptive architecture/algorithm linear filter combinations is the filter-based controller adapted by using Least Mean Square (LMS)-based algorithms (discussed in section 2.3.1.2) [8]. The aim of the adaptation algorithm is to adapt the filter coefficients such that the error signal is minimized.

2.3.1.1 Adaptive Filters

Adaptive filters can be realized as (transversal) finite impulse response (FIR), (recursive) infinite impulse response (IIR), lattice, and transform-domain filters [8-11] and [13-14]. Two primary types of digital filters used in (DSP) applications are FIR filters and IIR filters [8].

2.3.1.1.1 FIR Filters

Figure 2-4 shows the structure of a transversal FIR filter with N taps adjustable weights. The Z^{-1} block represents a unit sample of delay. The impulse response is "finite" because there is no feedback in the filter. The FIR filter is obtained by combining a tapped delay line with a linear network [8].

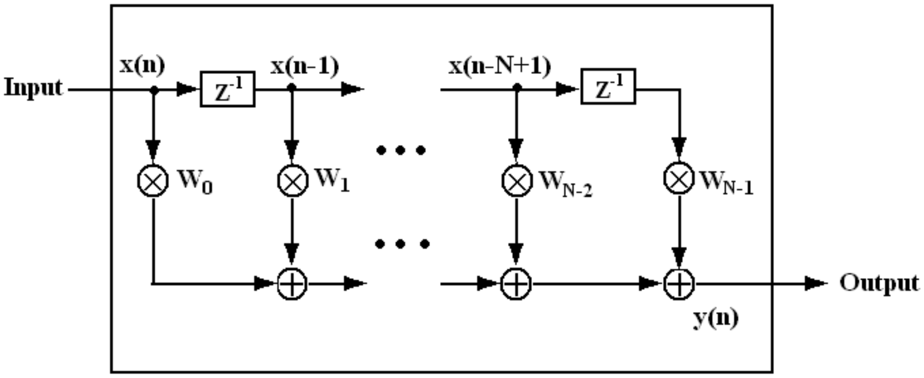


Figure 2-4: Transversal FIR Filter structure

The tap-weight vector, *w(n),* is represented as $w(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T$; the tap-input vector, *x(n)*, is represented as $x(n) = [x(n), x(n-1), \ldots, x(n-(N-1))]^T$; the FIR filter output, *y(n),* can then be expressed as

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) \tag{2-1}$$

where *n* is the time index and *N* is the order of the filter.

2.3.1.1.2 IIR Filters

Another primary digital filter used in DSP application is IIR. The impulse response is "infinite" because there is feedback in the filter. An IIR filter produces an output, *y*(*n*), which is the weighted sum of the current and past inputs, *x*(*n*), and past outputs. Figure 2-5 is a generic IIR digital filter and equation (2-2) shows how to calculate the output of the filter [15].

$$y(n) = g[x(n) + a_1\, x(n-1) + \ldots + a_N\, x(n-N) + b_1\, y(n-1) + \ldots + b_M\, y(n-M)] \tag{2-2}$$

If the generic IIR filter in Figure 2-5 does not operate on the past values of the output, i.e., it would only have non-zero a_i coefficients in the above equation, but all b_j coefficients would be zero, then it changes to a FIR filter as shown in Figure 2-4.

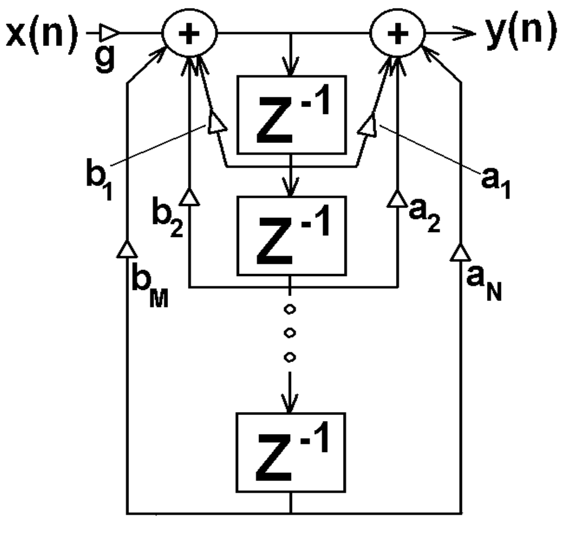


Figure 2-5: A generic IIR digital filter [31]

2.3.1.1.3 FIR Filters vs. IIR Filters

The FIR filter and The IIR filter have their separate advantages and disadvantages. Overall, though, the advantages of the FIR filter outweigh its disadvantages, so it is used much more than the IIR filter [8]. Compared to the FIR filter, the IIR filter can achieve a given filtering characteristic using less memory and calculation than a similar FIR filter. However, compared to IIR filters, FIR filters have more advantages as listed below [16]:

- They can easily be designed to be "linear phase";
- They are simple to implement;
- They are suited to multi-rate applications;
- They have desirable numeric properties; and
- They can be implemented using fractional arithmetic.

2.3.1.2 Adaptation Algorithms

There are many possible adaptation algorithms, and one of the most useful algorithms is the LMS algorithm [8, 10, 13]. Many LMS-based algorithms and their variants exist. This section focuses on the LMS algorithm and an LMS-based algorithm, i.e., the filtered-x LMS algorithm for FIR filters.

2.3.1.2.1 FIR Filters / LMS Algorithm Combination

Figure 2-6 shows a block diagram of an adaptive filter system identification model. The unknown system is modeled by an FIR filter with adjustable coefficients. Both the unknown system and the FIR filter model are excited by an input sequence *x(n).* The adaptive FIR filter output *y(n)* is compared with the unknown system output *d(n)* to produce the error signal *e(n).* The error signal represents the difference between the unknown system output and the model output. The error *e(n)* is then used as the input to an adaptive control algorithm, which corrects the individual tap weights of the filter. This process is repeated through some iterations until the error signal *e(n)* becomes sufficiently small. The resultant FIR filter response now represents that of the previously unknown system.

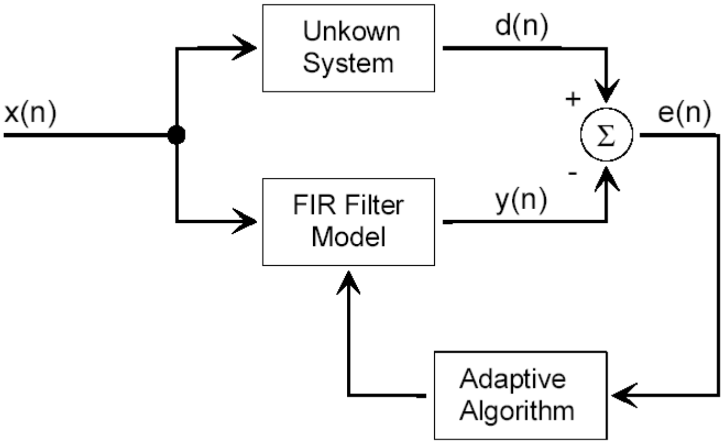


Figure 2-6: Adaptive filter system with a controller based on the LMS [11]

The LMS algorithm adjusts the weights and biases of the FIR filter so as to minimize the mean square error. LMS algorithm is an example of supervised training, in which the learning rule is provided with a set of examples of desired network behavior: $\{x_1, t_1\}, \{x_2, t_2\}, ..., \{x_{q,} t_q\}$. Here x_q is an input to the network, and t_q is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. The Mean Square Error (MSE) to be minimized is

$$MSE = \frac{1}{Q}\sum_{k=1}^{Q} e(n)^2 = \frac{1}{Q}\sum_{k=1}^{Q} (t(n) - y(n))^2 \qquad (2\text{-}3)$$

The LMS algorithm is initialized by setting all the weights to zero at time *k=0*. Tap weights and bias are updated using the relationship [11, 13]

$$w(k+1) = w(k) + 2\,\mu\, e(k)x(k) \qquad (2\text{-}4)$$

$b(k+1) = b(k) + 2\,\mu\, e(k)$ (2-5)

where *w(k)* represents the tap weights of the transversal filter, *e(k)* is the error signal, *x(k)* represents the tap inputs, *b(k)* is the bias, and the factor μ is a convergence factor (also called the learning rate or the step length) whose value influences the amount by which the weight vector is altered at each iteration. To ensure convergence, μ should satisfy the condition [11]:

$0 < \mu < (1 / \lambda_{\max})$ (2-6)

where $\lambda_{\max}$ is the maximum eigenvalue of the input correlation matrix.

2.3.1.2.2 FIR Filters / Filtered-x LMS Algorithm Combination

Another popular adaptive architecture/algorithm combination is the FIR filter/filtered-x LMS algorithm combination. The filtered-x LMS algorithm is also called the multi-error LMS algorithm or multi-channel LMS algorithm, which is an extension of the LMS algorithm.

In Figure 2-6, the output of the FIR filter is an estimate of the unknown system. However, in real control applications, the adaptive filter works as a controller controlling dynamic systems, which contain actuators, amplifiers, etc. The estimate in this case can thus be considered as the output signal from a dynamic system, i.e., a forward path as in Figure 2-7. Since there is a dynamic system between the filter output and the estimate, the direct LMS algorithm is likely to be unstable in this application due to the phase shift (delay) introduced by the forward path [2]. In this case, a model of the forward path can be introduced to filter the reference signal to the adaptive algorithm as in Figure 2-7. The compensated adaptive algorithm obtained is the filtered-x LMS algorithm [13].

As shown in section 2.3.1.1.1, the FIR filter output, *y(n),* can be expressed as equation (2-1). Equation (2-1) can also be written as

$$y(n) = w^T(n)x(n) \tag{2-7}$$

where

$$x(n) = [x(n), x(n-1), ..., x(n-N+1)]^T \tag{2-8}$$

is the input signal vector to the adaptive filter and

$$w(n) = [w_0(n), w_1(n), ..., w_{N-1}(n)]^T \tag{2-9}$$

is the adjustable filter coefficient vector. In control applications, the estimation error $e(n)$ is defined by the difference between the desired signal (desired response) $d(n)$ and the output signal from the forward path or plant under control $y_c(n)$:

$$e(n) = d(n) - y_c(n) \tag{2-10}$$

$$x_{C^*}(n) = \begin{bmatrix} \sum_{i=0}^{I-1} c_i^* x(n-i) \\ \sum_{i=0}^{I-1} c_i^* x(n-i-1) \\ . \\ . \\ . \\ \sum_{i=0}^{I-1} c_i^* x(n-i-N+1) \end{bmatrix} \tag{2-11}$$

where n is the time index, i is the order and c_i^* is the coefficients of the estimated FIR filter model of the forward path [10].

$$h_{C^*}(n) = \begin{cases} c_n^* \rightarrow n \in \{0, ..., I-1\} \\ \quad 0 \rightarrow else \end{cases} \tag{2-12}$$

where $h_{C^*}(n)$ is the estimated FIR filter model of the forward path.

$$w(n+1) = w(n) + \mu \cdot x_{C^*}(n)e(n) \tag{2-13}$$

where μ is a convergence factor (also called learning rate or step length).

In order to ensure that the filtered-x LMS algorithm is stable, the maximum value for the convergence factor μ should be given approximately by [13]:

$$\mu < \frac{2}{NE[x^2(n)]} \tag{2-14}$$

where $E[x^2(n)]$ is the mean square value of $x(n)$.

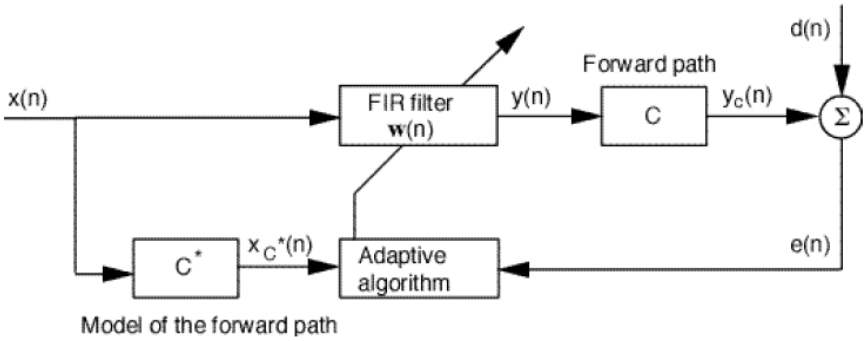


Figure 2-7: Adaptive filter system with a controller based on the filtered-x LMS algorithm [13]

2.3.2 Nonlinear Architecture /Algorithm Combinations

Even though the combination of a transversal filter using modified LMS algorithms has been widely demonstrated to be very useful, it has a potential limitation because it is designed for linear control problems. In other words, the control signal, as well as the associated measured error signal used in the adaptation process, must be linear functions of the reference signal used by the adaptive filter to derive the control signal [6]. The linear filter architectures may not perform well in cases where nonlinearities are found in an active control system. One of the most common sources of nonlinearity in the field of active control of vibration is the actuator. An actuator has typically a nonlinear response when it operates with an input signal having an amplitude close to (or above) the nominal input signal value, or when it operates at a frequency outside of the normal frequency range of operation (or close to the limits) [18]; a control actuator also may have some nonlinear performance characteristic, such as where it generates some harmonics and introduces the harmonics into the system, which must be compensated for; and where a power based (intensity) error signal is used, which will be twice the frequency of the reference signal [6], for example, when some band-pass filters are used for the error signal. Another example of nonlinearity in active vibration control is when a sinusoidal reference signal is used to derive a signal to control a disturbance containing both the reference tone and several harmonics. Nonlinear behaviors can also occur when the dynamics of the system to be controlled are nonlinear.

Therefore, what is desired in these situations is a nonlinear controller, which can improve the control performance of a system associated with some form of nonlinearity. One such controller arrangement, which has received increased attention in recent years, is the artificial neural network (ANN). ANN-based AVC is the focus of this research, so ANN architectures and algorithms, and ANN application in AVC will be introduced in detail.

2.4 Artificial Neural Network

An ANN is a system whose architecture is inspired by the arrangement of nerves in biological systems and by their operation. An ANN is an extensively parallel interconnection of simple neurons that has the ability to learn from its environment and store the acquired knowledge for future use [10]. ANNs are used for pattern recognition or function approximation. In AVC, ANNs are mostly used for their function approximation capability. Properly designed and trained neural networks are capable of approximating any linear or nonlinear function to the desired degree of accuracy [10]. A strong case can be made that neural network implementation is simply a form of multivariate statistical analysis [6].

There are a variety of design and learning techniques to choose from when using an ANN. The pattern of connections between the neurons, called network architecture, and the method in which the weights of the connections are determined, known as learning algorithm, are the elements that characterize an ANN.

2.4.1 Multilayer Feedforward Neural Network

The most common ANN architecture used in AVC is the multilayer feedforward neural network. Figure 2-8 illustrates an example of the structure of a multilayer feedforward neural network, which comprises a layer of input signals, 2 hidden layers (layer 1 and layer 2) of neurons, and an output layer (layer 3) of neurons.

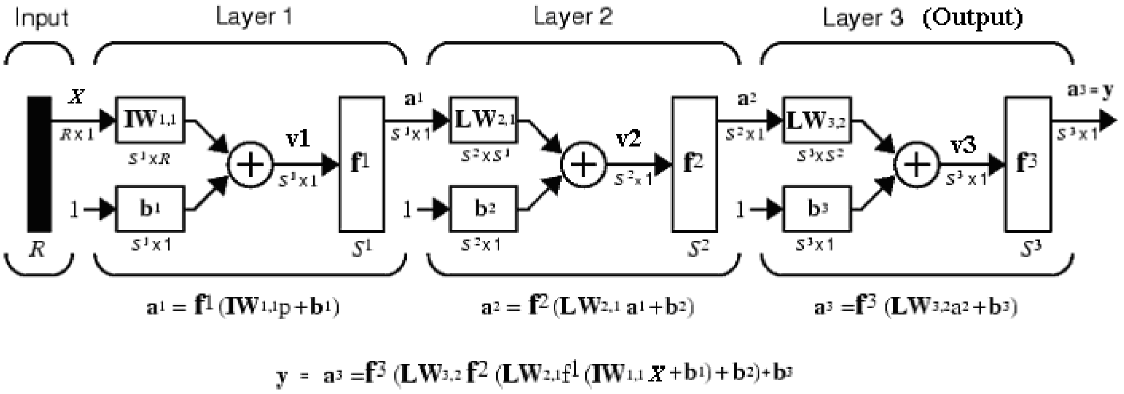


Figure 2-8: An example of the structure of a multilayer feedforward neural network [19]

A multilayer feedforward neural network (see Figure 2-8) comprises a layer of input signals, one or more hidden layers of neurons, and an output layer of neurons. A layer consists of a single or multiple neurons. Neurons in each layer are connected to all neurons in adjacent layers. The network in Figure 2-8 is of the feedforward type, wherein the effects of the input signals are propagated through the networks layer by layer. Differences between the desired outputs (targets) and the network outputs are the "errors". The connection strengths ('weights') and 'biases' are updated during training (learning) such that the network produces the desired output for the given input. The multilayer feedforward neural networks trained with a back-propagation algorithm are compact and provide excellent generalization (i.e., accurate outputs for inputs not encountered during training) [10]. Mathematically, neuron *j* having *m* inputs is described as follows [10]:

$$y_j = f_j(v_j)$$

$$v_j = \sum_{i=1}^{m} w_{ji} x_i + b_j \qquad (2\text{-}15)$$

where x_i are input signals, w_{ji} are weights from neuron *i* to neuron *j*, b_j is bias, v_j is the activation potential, f_j is the activation/transfer function and y_j is the output signal of the neuron. The bias applies an affine transformation to the linear combination of inputs and weights. The type of activation function used determines the neuron output for the given weights, inputs and bias. For example, the following activation functions are always used [10]:

$$f_j(v_j) = v_j, \qquad \text{linear}$$

$$f_j(v_j) = \frac{1}{1+e^{-v_j}}, \qquad \text{log-sigmoid}$$

$$f_j(v_j) = \tanh(v_j) = \frac{e^{v_j} - e^{-v_j}}{e^{v_j} + e^{-v_j}}, \qquad \text{tan-sigmoid} \qquad (2\text{-}16)$$

Feedforward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1. However, if the outputs of a network need to be constrained (such as between 0 and 1), the output layer should use a sigmoid transfer function (such as log-sigmoid).

Theoretically, a two-hidden-layer network having a sigmoid first hidden layer and a linear second layer can be trained to approximate most functions (linear or nonlinear) arbitrarily well [10].

2.4.2 Backpropagation Algorithms

There are many variations of the backpropagation algorithm. The simplest one is the gradient descent algorithm, which updates the network weights and biases in the direction in which the performance function decreases most rapidly -- the negative of the gradient.

For a given set of inputs to the network, outputs are computed for each neuron in the first layer and forwarded to the next layer. The signals propagate on a layer-by-layer basis until the output layer is reached. The weights and biases remain unchanged during the “forward pass”. The output of the network is compared with the desired value (t_j), and the difference gives the error:

$$e_j = t_j - y_j \tag{2-17}$$

Applying the delta rule [10], which adjusts the weights so as to minimize the mean square error, the total error (or “error criterion” in some papers) is defined as:

$$E = \frac{1}{2}\sum_{j=1}^{c} e_j^2 \tag{2-18}$$

where c is the number of neurons in the output layer.

The error E represents the cost function, and the weights and biases are updated to minimize it. The backpropagation algorithm minimizes the cost function in a manner similar to the steepest descent method [10]. The computed partial derivatives (sensitivity)

$\partial E/\partial w_{ji}$ determine the search direction for updating the weights w_{ji} as

$$w_{ji}(k+1) = w_{ji}(k) - \alpha \frac{\partial E(k)}{\partial w_{ji}(k)} \qquad (2\text{-}19)$$

where α is the learning rate parameter (step size), k is the current time and (k + 1) is the next time step. For stable learning,

$$\alpha < \frac{2}{\lambda_{\max}} \qquad (2\text{-}20)$$

where $\lambda_{\max}$ is the maximum curvature, or the maximum eigenvalue of the Hessian matrix [10]. The weights and biases are updated during the "backward pass" starting from the output layer, and recursively computing the local gradient for each neuron.

The gradient descent algorithm can be refined using a 'momentum term' that has a stabilizing effect on the backpropagation algorithm [10]. By the use of momentum, a larger α can be used, while maintaining the stability of the algorithm. Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. It also allows the network to ignore small features in the error surface. Without momentum, a network may get stuck in a shallow local minimum. With momentum, a network can slide through such a minimum [10].

In practical application, the two backpropagation algorithms (gradient descent, and gradient descent with momentum) are often too slow. Modifications fall into two main categories [19]. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. One heuristic modification is the momentum technique, such as variable learning rate backpropagation, and resilient backpropagation [19]. The second category of fast algorithms uses standard numerical optimization techniques, such as conjugate gradient, quasi-Newton, and Levenberg-Marquardt [10].

2.4.3 Learning Methods

The learning methods include supervised learning and unsupervised learning. In supervised learning neural networks are adjusted, or trained, so that a particular input leads to a specific target output. In unsupervised learning, there is not a specific target output. Unsupervised networks can be used, for instance, to identify groups of data. However, in AVC, supervised training methods are commonly used.

2.4.4 Training Styles

The training of a neural network is complete when the error (or change in the error) reduces to a predetermined small value. ANN training methods comprise batch training and incremental training. Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as “on-line” or “adaptive” training.

2.4.5 Normalization

Before training, it is often useful to scale the inputs and targets so that they always fall within a specified range [19]. Neural networks are very sensitive to absolute magnitudes. For this reason, the inputs and targets are usually scaled to give each of them equal importance and to prevent premature saturation of activation functions. All data to an ANN are normalized so that they correspond to roughly the same range of values. Normalization has the advantage of mapping the desired range of variable to a full working range.

2.4.6 Improving Generalization

One problem that occurs during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations [19].

One method for improving network generalization is to use a network that is just large enough to provide an adequate fit [10]. Unfortunately, it is difficult to know beforehand how large a network should be for a specific application. The standard ways to limit the capacity of an ANN including limiting the number of hidden units, limiting the size of weights and stopping the learning before it has time to overfit (early stopping).

In the early stopping method, the available data are divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise (see Figure 2-9, which is an example drawn by using MATLAB). When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. The test set error is not used during the training, but it is used to compare different models. It is also useful to plot the test set error during the training process. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this may indicate a poor division of the data set [19].

Another way recommended in [19] is Bayesian regularization. Bayesian regularization generally provides better generalization performance than early stopping, when training function approximation networks [19]. This is because Bayesian regularization does not require that a validation data set be separated out of the training data set. It uses all of the data. This advantage is especially noticeable when the size of the data set is small.

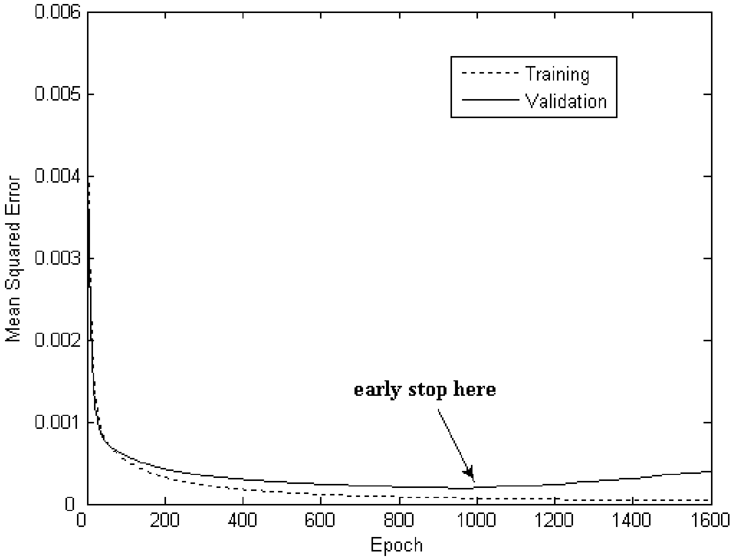


Figure 2-9: Early stopping method (drawn by using MatLab)

2.4.7 Dynamic Neural Network

Dynamic neural networks are always used to model dynamic systems [10]. A dynamic ANN has some time-delayed values of the outputs as its inputs by using tapped delay lines (TDL). A TDL is a delay line with at least one “tap”. A delay-line tap extracts a signal output from somewhere within the delay line, optionally scales it, and usually sums with other taps from the signal output. A tap may be interpolating or non-interpolating. A non-interpolating tap extracts the signal at some fixed integer delay relative to the input. Tapped delay lines efficiently simulate multiple echoes from the same source signal.

One example of using TDL in a neural network model of a system is shown in Figure 2-10 (a). In this example each “Time Delay” operator yields a one-time-step delayed version of the input signal, and thereby builds a short-term memory into the system. This feature transforms a static ANN to a dynamic ANN whose output is a function of time [9, 20].

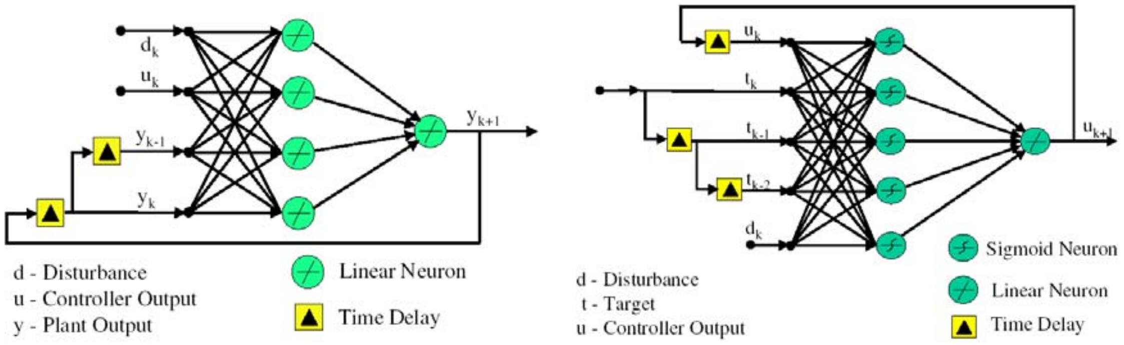
2.5 ANN Applications in AVC

The idea of using ANNs for nonlinear feedforward control problems was first proposed by K. S. Narendra and K. Parthasarathy [21], and then, the investigations into it by some other people have been reported (see, for example, [7, 17, 22-25]). Generally, multilayer feedforword neural networks adapted using some extensions of the standard gradient descent backpropagation algorithm has received the most attention from the control community as a potential nonlinear filtering tool [6].

In the AVC area, the linear function is generally used for neurons in the output layer, and nonlinear functions (such as the sigmoid function) are used for neurons in hidden layers. The linear activation function used for output neurons can provide the control signals with the capacity to vary over the positive/negative range required for control; the nonlinear activation functions used for hidden neurons can provide nonlinear control.

2.5.1 Batch-Training Dynamic ANN-Based AVC

One example of batch-training dynamic ANN-based AVC is discussed in [25]. A Neural Network Identifier (NNI) and a Neural Network Controller (NNC) are used in this example (Figure 2-10). The NNI is a model of the system, which is a dynamic neural network and used to simulate the response and design the controller. The inputs to the NNI include excitation signals, control signals and two delayed values of plant output. The activation function is purely linear. The design of the NNC is based on the inversion of the plant model (NNI). The NNC has five hidden neurons and a single output neuron, which produces the controller voltages. The inputs consist of excitation signals, control signals and time-delayed target values. The hidden layer uses the tangent sigmoid activation function (as in equation 2-16), which limits the output to ±1 for large values of the activation potential. This has a stabilizing effect on the controller signals. For the output neuron, the activation function is purely linear, which provides the control signals with the capacity to vary over the positive/negative range required for control.



(a) Neural network identifier (NNI).

(b) Neural Network Controller (NNC).

Figure 2-10: Neural network identifier and Neural Network Controller in [25]

The significant feature in this example is the way to train the NNC. The NNI is trained in a batch mode, and then is used to train the controller through off-line simulation. An adaptive scheme was used for the controller, as shown in Figure 2-11.

The problem in training the NNC is that no target values (in terms of control signals, which are the output of the NNC) are known that would minimize the tip acceleration (which is the goal). The significant idea of [25] to solve this problem is by defining the target as the sum of control signal and tip acceleration (NNI output, which is desired to be zero for vibration suppression). The error is the difference between the target value and the NNC predicted value (control signal). The error (the tip acceleration in this case) is backpropagated to train the NNC. The NNC weights and biases are adjusted after every ten samples of data, which resulted in updated control signals. Once the NNC training was complete, it was connected to the plant to obtain controlled responses.

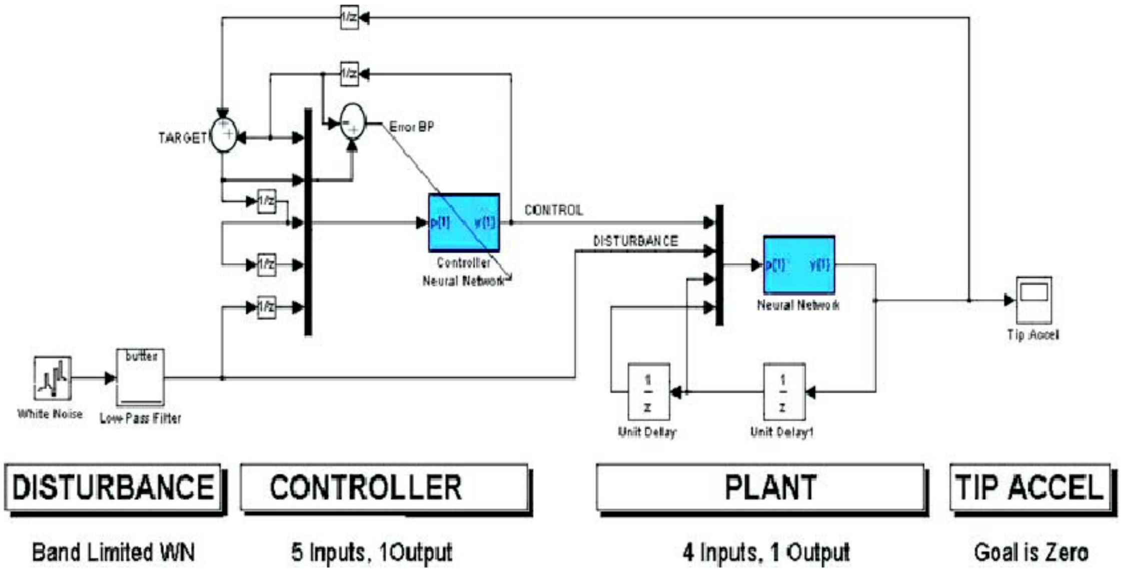


Figure 2-11: Training of Neural Network Controller (NNC) in [25]

In another example, the simplest concept called “Direct Inverse Control” is implemented in an ANN-based AVC [24]. The principle of “Direct Inverse Control” is that if a process can be described by a function dependent on the past states of the system that may be dependent on past inputs and outputs, a network is trained as the inverse of the process since that plant can be approximated by an inverse function. The inverse model is subsequently applied as the controller for the process by inserting the desired output. [24].

To design a plant inverse model, input/output data must be collected for the operating range and input conditions of the system. To construct a direct inverse neural network controller, an ANN to model the inverse of the plant needs to be trained (see Figure 2-12). The inputs to the ANN inverse model are the state: *x(k)* and the change in state: *dx*. The output is the input control signal going to an actuator or shaker.

After the ANN is trained, it is put into the direct inverse control framework. The input to the inverse plant model controller is the current state and the desired state. The output of the controller is the input control signal going to an actuator or shaker (see Figure 2-13, where D means one time step delay here).

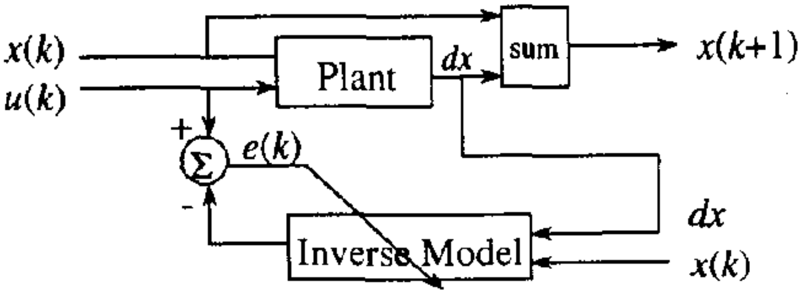


Figure 2-12: Inverse system identification in [24]

Both of the above two examples are claimed to demonstrate the efficiency and robustness of batch-training dynamic ANN-based AVC mechanism.

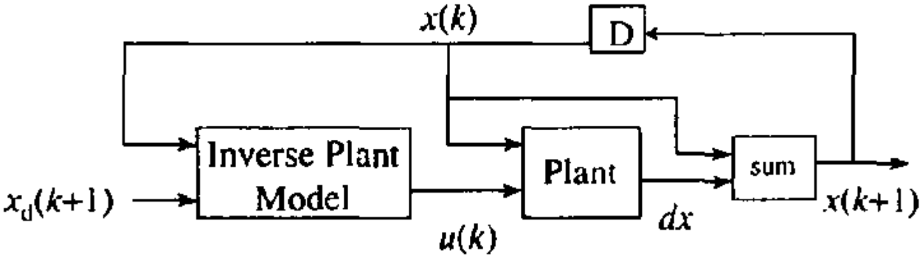


Figure 2-13: Direct inverse control of a plant in [24]

2.5.2 On-Line-Training Dynamic ANN-Based AVC

There has been an intensive interest in on-line-training dynamic ANN-base AVC since the idea of using ANN for nonlinear feedforward control problems was first proposed by K. S. Narendra and K. Parthasarathy ([7, 17-18, 21]). The commonly used feedforward AVC arrangements are similar to the one shown in Figure 2-14. “ANN 1” is a neural network controller and “ANN 2” is a neural network model of the plant. Reference signals are sent to “ANN 1”. Tapped delay lines are always used as the input layer in feedforward neural networks. Tapped delay lines yield time-step delayed versions of the input signal, and thereby built short-term memories into systems. This feature transforms a static network into a dynamic network whose output is a function of time (as discussed in section 2.4.7). Transfer function models utilize tapped delay lines to enable the modeling of the explicit system time delays to maintain causality within the control scheme (as discussed in section 2.2.1).

The reference signals are used by “ANN 1” (neural network controller) to derive a set of control signals. Each control signal is modified by some system dependent cancellation path transfer function, i.e., “ANN 2” (plant model), before sent to the plant through actuators. Each error signal is then the sum of the primary and control components (superposition of the signals in the plant environment) and measured by a sensor. The error signals are then used by the neural networks for weight adjusting.

The training of “ANN 2” can be done with classical neural networks algorithms, including backpropagation algorithms (with or without momentum), nonlinear optimization algorithms (quasi-Newton algorithms, conjugate gradient algorithms) or nonlinear identification techniques (nonlinear extended Kalman filtering or recursive-least-squares algorithms). A review of these algorithms can be found in [18]. The training of “ANN 1” can not be done with those classical algorithms, because of the tapped delay lines between the two neural networks. For this kind of on-line-training dynamic ANN-based AVC, finding the effective algorithms to train “ANN 1” has been a focus of many researches. The combination of multilayer feedforword neural networks adapted using some extensions of the standard gradient descent algorithm, together with the developments of those modified algorithms, can be found in the literature.

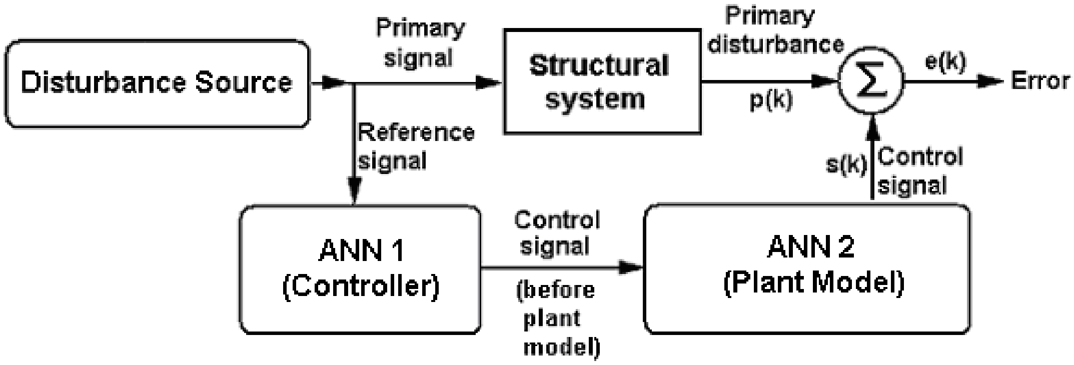


Figure 2-14: Commonly used feedforward active control arrangement

One example of such a kind of on-line-training dynamic ANN-based AVC is shown in [7], in which two multilayer feedforword neural networks, each having a tapped delay line, are used; a modified gradient descent backpropagation algorithm for adapting the feedforward neural network which has a tapped delay line is developed, and the performance characteristics are assessed experimentally.

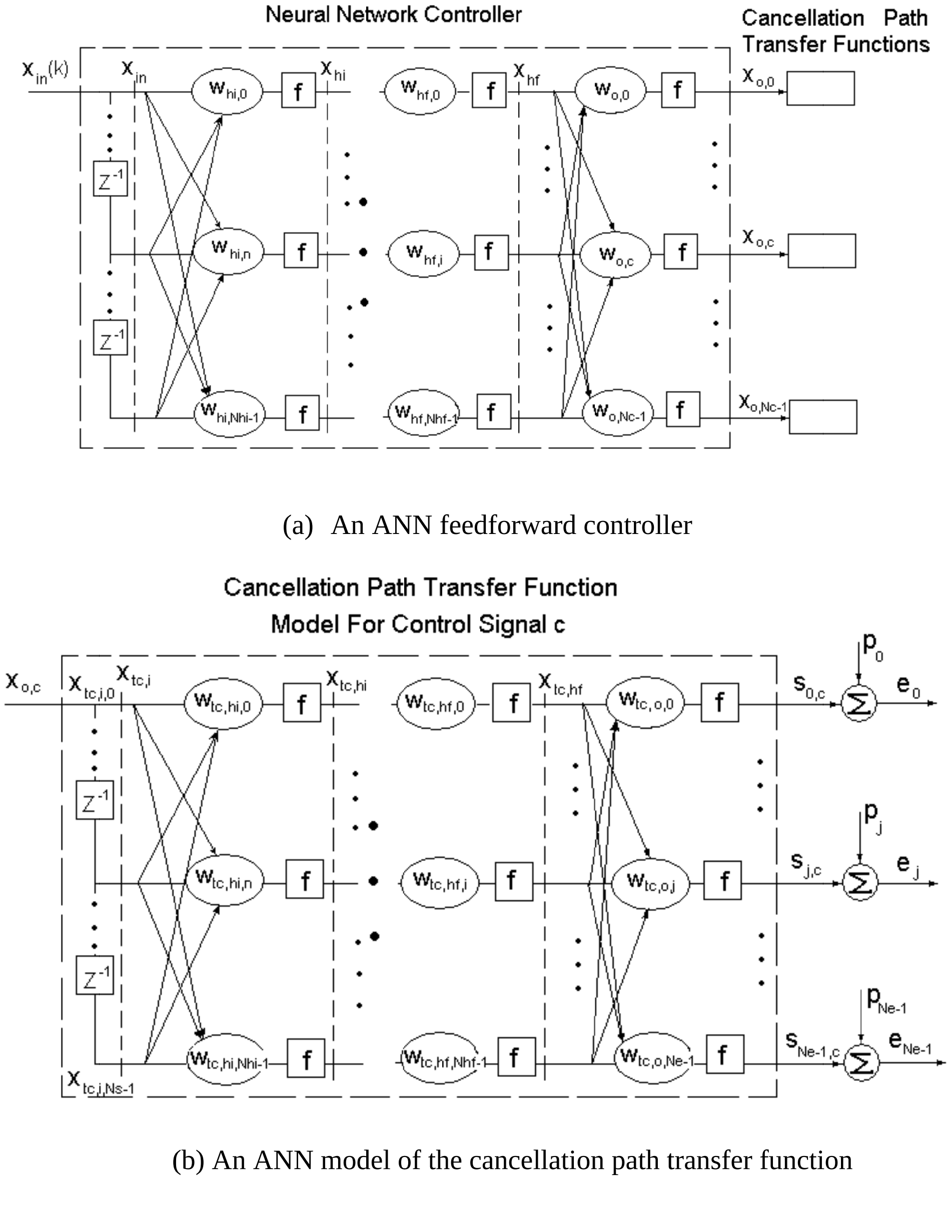
Figure 2-15 represents the two multilayer feedforword neural networks used in [7]. Figure 2-15 (a) is an ANN feedforward controller, i.e. "ANN 1" in Figure 2-14; and Figure 2-15 (b) is an ANN model of the cancellation path transfer function, i.e. "ANN 2" in Figure 2-14. In [7], the control achieved by using a 6x6x1 "ANN 1" (6 inputs, six nonlinear hidden layer nodes, and one linear output node), together with a 6x1 "ANN 2" (six inputs and one linear output node).

As discussed before, in applications of on-line-training dynamic ANN-based AVC, in order to derive an algorithm which will facilitate stable adaptation of an ANN-based feedforward active control system, the plant model is used to incorporate the frequency response characteristics of the control actuator and error sensor, as well as the response characteristics of the structural system which separates them, including delays due to the finite distance between the source and the sensor. In [7], this plant model is modeled as a second neural network, the input of which is the control signals, $x_0(k)$, and the output of which is the feedforward control signals, $s(k)$, measured at the output of the error sensor. A reference input sample at time *k*, $x_{in}(k)$, which is in some way related to (but not necessarily linearly correlated with) in impending primary disturbance, $p(k)$, is used to derive the set of control signals, $x_0(k)$. Each error signal is then the sum (superposition) of the primary and control components [7]:

$$e_j(k) = p_j(k) + s_j(k) \tag{2-21}$$

The error criterion that the controller is to minimize is the sum of the mean square value of error signals from $N(e)$ error sensors [7]:

$$ErrorCriterion = \sum_{n=0}^{N_e-1}\xi_n(k) = \sum_{n=0}^{N_e-1}E\left\{e_n^2(k)\right\} \approx \sum_{n=0}^{N_e-1}e_n^2(k) \qquad (2\text{-}22)$$



(a) An ANN feedforward controller

(b) An ANN model of the cancellation path transfer function

Figure 2-15: Two multilayer feedforword neural networks used in [7]

The gradient descent algorithm acts to adjust the weights of the control system [7]:

$$w(k+1) = w(k) - \alpha\Delta w(k) \qquad (2\text{-}23)$$

The gradient estimate used in the adaptive algorithm is [7]:

$$\Delta w \approx \sum_{n=0}^{N_e-1} \frac{\partial e_n^2(k)}{\partial w} \tag{2-24}$$

The impediment to utilizing the standard backpropagation algorithm for adapting the controller neural network is the inclusion of the tapped delay line input to the transfer function model. The standard backpropagation algorithm cannot be used directly in this arrangement since it must backpropagate through a tapped delay line. Therefore, the standard gradient descent backpropagation algorithm must be modified to enable adaptation of the neural controller for use in feedforward control systems. In [7], steepest descent algorithms based on the filtered-x approach were introduced for the training of the controller network. In order to derive a modified algorithm, the error signals can first be backpropagated from the transfer function model output to the tapped delay line if the transfer function model input uses the standard algorithm [7]. The outcome shows that it is in fact past and present versions of the nodal outputs which are used in updating the controller network weights, and not past and present values of the error signals (see [7] for details of the algorithm development).

One point should be stressed here is that the neural network controller and the cancellation path transfer function neural network are inherently different in their function, so they must be adapted separately [7]. Once converged, the cancellation path transfer function neural network is then simply used as a tool to facilitate stable adaptation of the neural network controller and is not modified itself in this process. The neural network controller is a “phase inverse” model, whose error signal is defined as the sum of its output and the signal whose inverse signal is desired (the superposition of the control signal and the primary disturbance in the structural domain); while the cancellation path transfer function neural network is a model, whose error signal is based on the difference between its output and some desired signal (the system response measured at the error senor output to the control signal input) [7].

For this kind of on-line-training dynamic ANN-based AVC, finding the effective algorithms to train the ANN controller has been a focus of many researches. For example, in [17], an adjoint approach is introduced; and in [18], a heuristic procedure is introduced for the development of recursive-least-squares algorithms based on the filtered-x and the adjoint gradient approaches, which leads to the development of new recursive-least-squares algorithms for the training of the ANN controller.

2.6 Real-Time Concepts for Dynamic ANN-Based AVC

The following real-time concepts are very important for dynamic ANN-based AVC experimental implementation.

2.6.1 Real-Time Performance

The most common misconception associated with real-time performance is that it increases the execution speed of a program. While this is true in some cases, it actually enhances the application by providing more precise and predictable timing characteristics. With these enhancements, the exact time when certain events will occur can be determined [26]. Real-time performance can be achieved through either hardware or software.

2.6.2 Real-Time Operating Systems

A real-time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation, but also upon the time at which the result is produced. If the timing constraints of the system are not met, system failure is said to have occurred [27]. Microsoft Windows 2000, XP and Vista are all general-purpose operating systems, but they are not real-time operating systems [26].

One of the main differences between real-time operating systems and general-purpose operating systems is the ability to guarantee a worst-case latency. On a general-purpose operating system, an external interrupt could be put into a queue and then serviced later after the operating system has finished its current operation and any other interrupts in the queue. On the other hand, a real-time operating system can halt its current process to handle an interrupt immediately. In essence, the real-time operating system guarantees event response within a certain interval [26].

2.6.3 Real-Time Control

With real-time control, a physical system can be monitored and simulated continually. Real-time control applications repeatedly perform a user-defined task with a specified time interval separating them [26].

2.6.4 Real-Time Signal Processing

Real-time signal processing has many of the same characteristics as real-time control. It requires deterministic time intervals between repetitive events [26]. But instead of calculating a response, it performs signal processing on the acquired data. In dynamic ANN-based AVC applications, point-by-point analysis routines provide much better performance. Instead of analyzing blocks of data, these routines maintain a memory of previous data and calculate a new output based on the history of the data and the current value. Hard real-time performance is necessary in these systems because missing input values or even acquiring these values after a small time delay destroys the integrity of the historical data for future calculations.

2.7 Challenges in Current AVC Applications

There are some technical challenges associated with the current AVC applications. As introduced in section 2.3, adaptive filter control systems can only work effectively for linear vibration problems. As introduced in section 2.5, most implemented ANN-based control methods lead to the complexity of the control systems.

The previous work tried to deal with these challenges. The proposed methodology is introduced in the following section.

2.8 Previous Work in AVC

The previous work in AVC is presented in [28]. Feedforward control strategy (as introduced in section 2.2.1) was utilized in the previous work of AVC design and the system did not have to deal with the time-delay problem directly. In a feedforward AVC strategy (see Figure 2-16), a controller relies on the availability of a reference signal correlated to the primary disturbance, i.e. *x(p)* in Figure 2-16. This signal is passed through the controller. And then, the output of the controller, i.e. *x(u)*, is applied to the plant by a secondary source, i.e. Actuator in Figure 2-16. The plant output, i.e. *Y* in Figure 2-16, is the vibration response of the plant measured at some point.

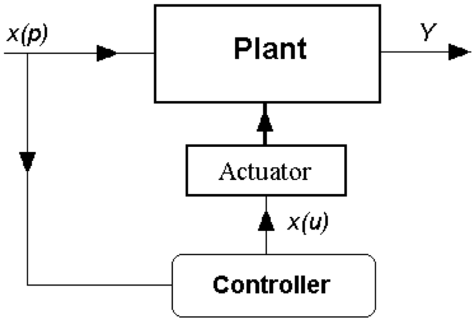


Figure 2-16: Feedforward AVC strategy in previous work

An ANN was used as the main part of the controller in the previous work. The ANN was used for identification of the system, i.e. as a system model. The system model was trained off-line using the system inputs as inputs and the measured vibration responses of the plant at some point of interest as the output. The input signal of the plant was also the reference signal to the controller as shown in Figure 2-16.

Figure 2-17 shows the training process of the ANN model, where *x(p)* represents the primary inputs to the plant, *x(u)* represents the control inputs to the system and *Y* is the vibration response of the plant measured at some point of interest, which also is the expected output of the ANN model. The error, i.e. *e*, was the difference between the measured plant output values, i.e. *Y*, and predicted values of the ANN model, i.e. *Y*´. The error was backpropagated to train the ANN model. After training the ANN model, the model was used in the controller for AVC of the plant.

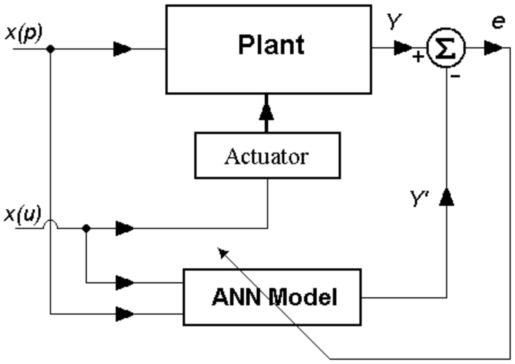


Figure 2-17: Training of the ANN model in previous work

To make the AVC system simple to implement, in the previous work, the controller was a static ANN controller, which included two main parts (see Figure 2-18). The first part was the trained ANN model. The second part was a minimization module which generated random control inputs *x(u)* ´ to the ANN model and found the minimum ANN model output value. Literally, the controller worked in this way: for given primary inputs to the plant, i.e. *x(p)*, the ANN controller generated random control inputs *x(u)* ´; the combination of *x(p)* and *x(u)* ´ were sent to the ANN model as inputs; for each combination of *x(p)* and *x(u)* ´, an output, i.e. *y* ´, was obtained and sent to the minimization module; the minimization module found the minimum ANN model output, i.e. Y'_{min}, based on the given primary input to the plant and sent the corresponding signal, i.e. *x(u)* ´, to the plant as control signal, i.e. *x(u)*.

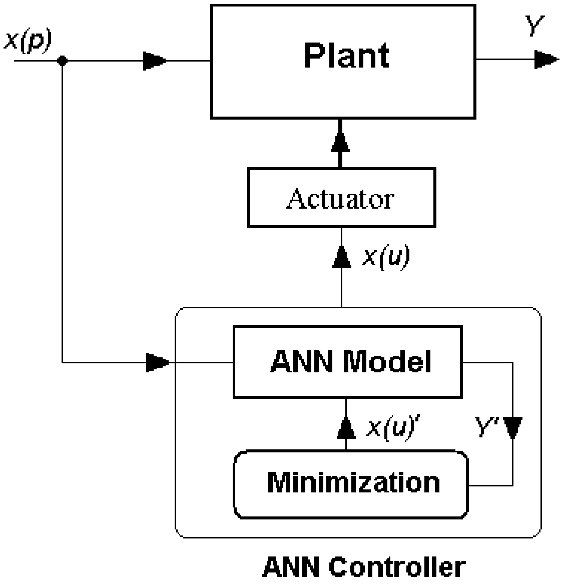


Figure 2-18: Static ANN-based AVC methodology in previous work

The inputs of the ANN included the primary inputs to the plant, i.e. *x(p)*, and the control inputs to the system, i.e. *x(u)*. The output of the ANN model was the vibration responses of the plant measured at some point of interest, i.e. *Y*. The training data, i.e. *x(p)*, *x(u)* and *Y*, were obtained by experiments.

To get a robust neural network model, which means a model affected minimally by external sources of variability, the experiments need to be designed first. In the previous work, the fractional factorial design [59] was used for the design of experiments to obtain training data for the ANN model.

The minimization module in the ANN controller used simulated annealing and resilient propagation algorithms.

2.9 Machining Chatter Suppression

In the chatter control area, many papers concentrate on chatter control while some focus on chatter detection. This research will comprise both parts but focus on chatter control. Therefore, both chatter detection and suppression methods, together with regenerative chatter mechanisms, will be reviewed in this section.

2.9.1 Regenerative Chatter

The mechanisms that cause regenerative chatter are called "Regeneration of Waviness". Tobias [31] was the first to identify the mechanisms known as regeneration. Figure 2-19 shows a two-dimensional metal cutting in turning. Although in many papers nowadays the turning process is described as two-degree-of-freedom systems (early investigated by Salje [32]), for the sake of simplification, in this figure the dynamic model of the turning tool is assumed to be a single-degree-of-freedom lumped parameter system whose stiffness and mass are the effective stiffness and mass of the turning tool reduced to the point of attachment of the cutting insert. The flexible tool engages the workpiece and, due to the cutting force, begins vibrating. This vibration is imprinted on the machined surface to leave a specific "wavy" surface on the workpiece. This wavy surface varies the instantaneous chip thickness which, in turn, modulates the cutting force and the cutter vibration (i.e., a feedback mechanism is produced that can lead to self-excited vibrations, or chatter) [33]. Depending on the relationship between the wavy surface left by the previous tooth and the current cutter vibration, the resulting deflections and forces can grow very large (chatter) or diminish (stable cutting) [33].

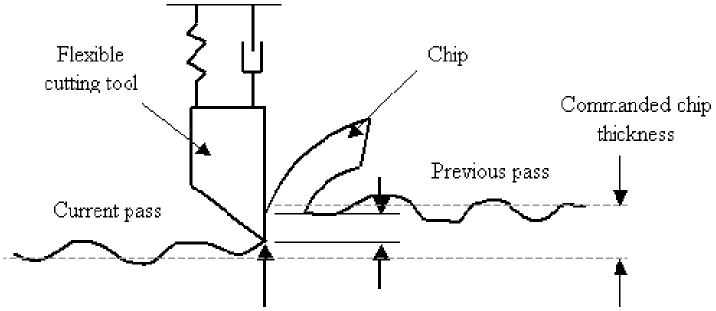


Figure 2-19: Chip thickness variation due to cutter vibrations [33]

An example of a stable turning operation is shown in Figure 2-20, where the tool vibrations are in phase with the surface left in the previous revolution. This leads to very little variation in the instantaneous chip thickness and, therefore, a stable cut even though the tool is vibrating. If the new cut leads to a chip with variable chip thickness (i.e. waves are out of phase), this would translate as variable forces on the cutting edge and eventually as vibration, i.e., regenerative chatter [33].

Favorable spindle speeds and chip widths can be selected to avoid chatter by using stability lobe diagrams, provided the system dynamics have been characterized beforehand. A typical stability lobe diagram example shown in Figure 2-21 is a plot that separates unstable combinations of chip width and spindle speed (i.e., those that produce chatter) from stable combinations [33]. Stable cuts occur in the region below the stability boundary (or combination of all the stability "lobes"), while unstable cuts occur above the stability boundary.

It is often possible to increase the allowable chip width without chatter by increasing the spindle speed, rather than slowing down. This counterintuitive behavior is one reason that is important to characterize the dynamic response of the cutting tool and produce the corresponding stability lobe diagram [33].

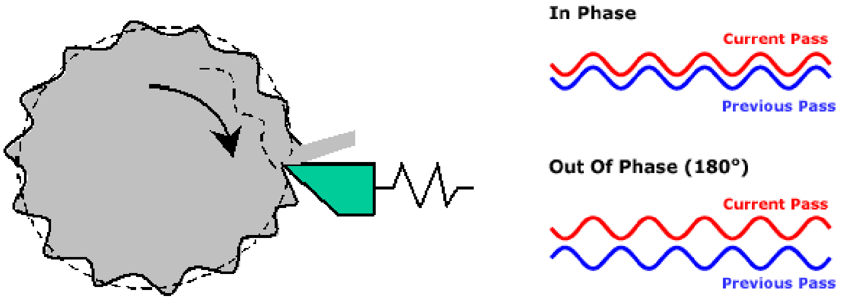


Figure 2-20: Tool vibrations are in phase with previously cut surface [33]

Upon closer observation of the Stability Lobes, it is evident that maintaining a chip width below the b_{limit} will always yield a stable cutting region. The value of b_{limit} depends on the dynamic characteristics of the structure, the work piece material, cutting speed and feed, and the geometry of the tool [34].

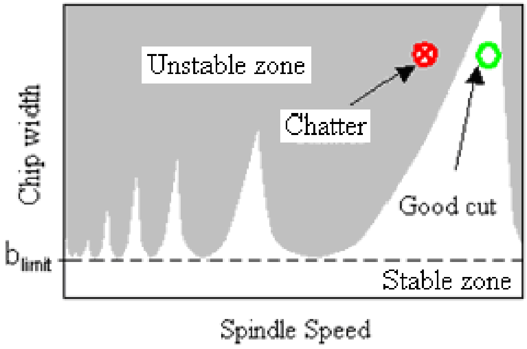


Figure 2-21: Example stability lobe diagram [33]

It is widely known that machining chatter signals have harmonic shapes [32], and their frequencies are around the respective natural frequencies of the machining systems [32]. As observed by experimental results of many papers, such as [32] and [35], the frequency of a regenerative chatter is around the first natural frequency of the machining system and in most cases slightly lower than the natural frequency.

2.9.2 Chatter Detection

Over the years, various techniques for on-line detection of chatter have been studied to detect chatter rapidly and accurately. In the frequency domain, some studies have focused on setting up a proper threshold value for the power spectrum of measured dynamic force, sound, acceleration or displacement to detect chatter occurrence [36]. The difficulty in determining suitable threshold values has led to artificial neural network-based techniques (e.g., [37]), multi-sensor based techniques (e.g., [38]) or the combination of both (e.g., [39]). In the transition of cutting dynamics domain, [40] proposed that the cutting process contained chaotic dynamics and utilized the premise in chatter detection using coarse-grained entropy rate, based on a transition from high dimensional to low dimensional dynamics of cutting at the onset of chatter; [41] and [36] applied the maximum likelihood (ML) algorithm to estimation of fractal dimension using wavelets.

However, some methods mentioned above are not suitable for the current research objective. In the frequency domain, although ANNs are a potentially powerful tool for classification owing to their ability to represent complex patterns by learning, the successful application of an ANN is strongly dependent on the proper selection of the type of network structure as well as the adequacy of the training data, which are not always available for the constantly changing machining environment, such as the machining of super alloys in small batches. In the transition of cutting dynamics domain, calculations are always complicated and sometimes are not suitable for on-line application.

2.9.3 Chatter Suppression

Generally, chatter suppression methods fall into two categories, passive and active methods.

2.9.3.1 Passive Methods

Passive control, compared with active control, exhibits the advantages of easy implementation, low cost, and no need for external energy. More importantly, passive control methods never drive the controlled system to instability, while the active control methods might [5].

Passive methods include enhancing the system’s dynamic stiffness and damping, elimination of additional energy sources, eliminating or decreasing input forces and isolation from external disturbances [5]. Usually passive chatter suppression methods include using energy absorbing dampers to consume the energy input (such as using damping treatment on a workshop floor), changing the cutting conditions to reduce the energy input or by changing the cutting conditions to reduce chatter energy generated during the machining process (e.g., detecting and calculating chatter “stability lobes” and then adjusting the process parameters, such as speed and feed, to produce a stable cut, as utilized in many papers), increasing of the rigidity of the machine tool structure by redesigning or through the use of dampers [43], changed tool geometry, variation of directional factors, etc. (e.g., modifications were made to the tool holders for adding dynamic stiffness and damping in [42]). Based on the dynamics of stability lobe diagrams, some tunable stiffness, damping [44] or vibration absorber [49], spindle speed variation [45] and spindle speed selection [46] strategies have been proposed for regenerative chatter suppression. Also, In the case of conventional cutting, chatter is very sensitive to the tool geometry, such as the rake angle and the clearance angle [47] [48].

The use of passive vibration control methods is restricted because of small effectiveness in the range of low frequencies, sensitivity-dependent on application conditions, lack of robustness, reduction of efficiency, increased size and weight, etc. Damping refers to mechanical energy dissipation that is converted to heat [5], so it may cause the decrease of general working efficiency in machining. The ideal device should run with minimal damping values. On the other hand, because of sensitivity-dependent on application conditions and lack of robustness, passive methods suffer from the fact that when the machine tool-workpiece configuration changes, the machines have to be retuned. However, an online or self-tuning is difficult to achieve with passive methods. All these lead to active control consideration.

2.9.3.2 Active Methods

Active methods work by providing an additional energy supply to vibration systems. The additional energy supply can produce forces that compensate the forces that account for chatters. As mentioned, active methods can overcome the limitations discussed before. In this section, some recent examples of active chatter control will be introduced.

Active vibration control is typically achieved by incorporating sensor and actuator pairs in the structural design to modify the response via feedback control [35]. Once active elements are incorporated into the structure, any type of feedback control may be used. The experimental setup of [35] (as shown in Figure 2-22) for a boring chatter suppression is the nearest to the experimental setup that will be utilized in this research.

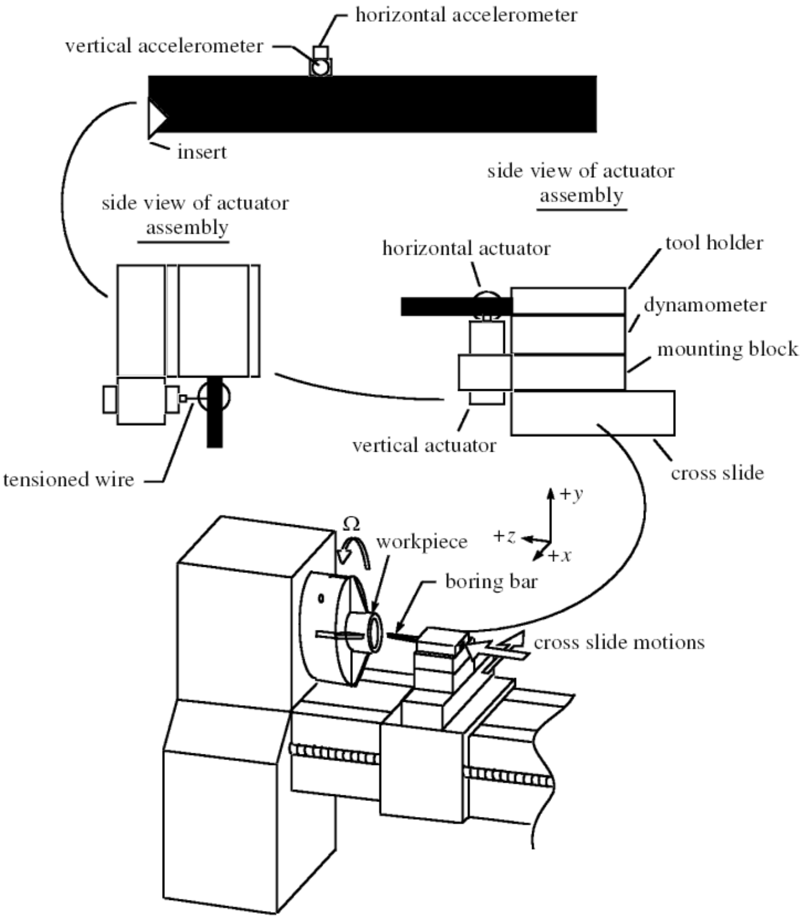


Figure 2-22: Location of the actuators and sensors along the boring bar [35]

In [50], a method based on a variable-stiffness boring bar containing an electrorheological (ER) fluid is utilized to suppress chatter in boring. ER fluids undergo a phase change when subjected to an external electrical field, the deformation modes of which are dependent on applied electrical field strength and strain amplitude. This phenomenon permits the global stiffness and energy-dissipation properties of the bar to be tuned on line by varying the electrical field strength for chatter suppression.

The chatter suppression method is based on the application of active dampers to a slender boring bar in [51]. Chatter vibration signals detected by a sensor are fed to a computer. After calculating the chatter frequency and the corresponding phase shift parameter, the computer supplies the amplified signals to piezoelectric actuators with the same phase as that of the vibration velocity of the boring bar. As a result of this, the actuators generate damping forces; that is, they act as active dampers. The experimental setup of [51] (as shown in Figure 2-23) can get more working range for the boring bar than [35]'s, so it deserves a closer scrutiny in the future. In this setup, eight piezoelectric actuators are attached to the boring bar as the active dampers.

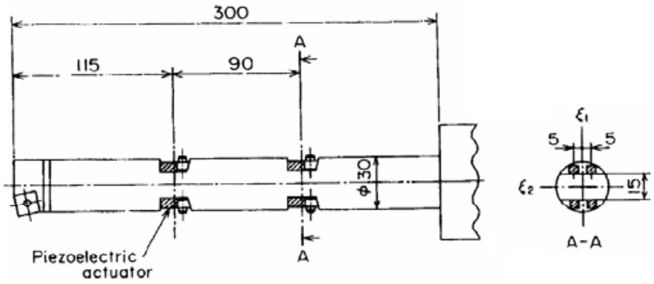


Figure 2-23: Boring bar with active dampers [51]

A unique boring bar system has been developed in [52]. The system consists of three principle subsystems: active clamp (tool holder), instrumented boring bar and control electronics. The active clamp is a lathe-mountable body capable of supporting bars of varying sizes and articulating them in orthogonal directions from the base of bar shank. The instrumented bar consists of a steel shank, standard insert head and imbedded accelerometers. Wire harnesses from both the bar and the clamp connect to control electronics comprised of amplifiers, a PC-based program manager and two digital signal processing boards. All real-time signal processing is based on the principles of adaptive filter minimization. The active clamp design of [52] can also provide more working range for the boring bar and be able to support bars of varying size, but it is complicated and delicate.

CHAPTER 3

METHODOLOGY OF AVC

As stated previously, the first part of the objective of this research is to develop an effective adaptable real-time online AVC system to detect and suppress the noisy sinusoidal vibration of a cantilever beam. Because of their many advantages, artificial neural networks (ANNs) are used to fulfill these requirements. The methodology to be introduced in this chapter does not utilize a reference signal correlated to the primary disturbance and as such is different from previous work [28], which utilized feedforward control strategy. The rationale is that a suitable reference signal is not always available and a robust AVC system should be able to monitor its effect on the system it is controlling and compensate for unexpected changes. Therefore, feedback control strategy is utilized in the current work. As a consequence, the vibration signal needs to be detected in the current work.

The general idea of the AVC methodology in the current work is based on Fourier theory. Fourier theory states that any signal, in the current case vibration signals, can be expressed as a sum of a series of sinusoids. If a vibration control system can detect the sinusoid with the highest power and control this sinusoid in one control cycle, and repeat the control cycle to control the sinusoid with the next highest power, then, after enough control cycles, most of the original vibration should be controlled. Therefore, the general methodology is divided into vibration detection, which detects sinusoid parameters, and vibration control, which sends out an accumulated control signal to control vibration sinusoids detected in different control cycles.

The vibration detection methods and the AVC strategy used in the current work will be presented. The resilient propagation algorithms used in the AVC system will also be introduced.

3.1 Vibration Detection

As mentioned at the beginning of this chapter, vibration detection is to detect sinusoid parameters because the general methodology in the current work is based on Fourier theory, which states that any signal can be expressed as a sum of a series of sinusoids. A summary of different methods for detecting sinusoid vibration parameters can be found in [53]. Classical methods include the maximization of periodogram (MP) and the minimization of the sum of squared error by non-linear least squares (NLS) regression. In [54], an algebraic approach is proposed for the fast and reliable, on line, identification of the amplitude, frequency and phase parameters in unknown noisy sinusoidal signals.

Generally, the algebraic method uses the algebraic derivative method in the frequency domain yielding exact formulae, when placed in the time domain, for the unknown parameters. Considering an uncertain sinusoidal signal of the form:

$$x(t) = A\sin(\omega t + \phi) + K \tag{3-1}$$

where A is the unknown amplitude, ω is the unknown frequency, ϕ is the unknown phase, and K is an unknown constant bias perturbation term, the Laplace transform of this signal is given by [54]:

$$x(s) = \frac{A\omega\cos\phi}{s^2+\omega^2} + \frac{sA\sin\phi}{s^2+\omega^2} + \frac{K}{s} \tag{3-2}$$

where s is the complex frequency. After some differentiations, integrations, and integral convolutions as shown in [54], the unknown A, ω, and ϕcan be obtained. If used with appropriate filters, the algebraic method can deal with noise very well.

Since the algebraic approach is fast (can be performed in a quite small time interval which is only a small fraction of the first full cycle of the measured sinusoid signal), robust with respect to signal measurement noises and able to do the computation of amplitudes, frequencies and phases of a linear combination of sinusoids [54], it can be utilized in the current work. To get more accurate parameters, especially for frequency, the outcomes of the algebraic approach can be applied to classic methods, which require extremely precise initial values to ensure convergence.

3.2 The General Vibration Control Strategy

The general proposed AVC strategy utilized in the current work is shown in Figure 3-1 [30]. In this strategy, a vibration suppression module relies on the availability of detected vibration parameters from the vibration detection module to generate control signals, i.e., *x(u)*, which are applied to the plant by secondary sources, i.e., actuators, to suppress the vibration. In Figure 3-1, *x(p)* represents the primary disturbance. The plant output, i.e., *Y* in Figure 3-1, is the vibration response of the plant measured at the location of interest.

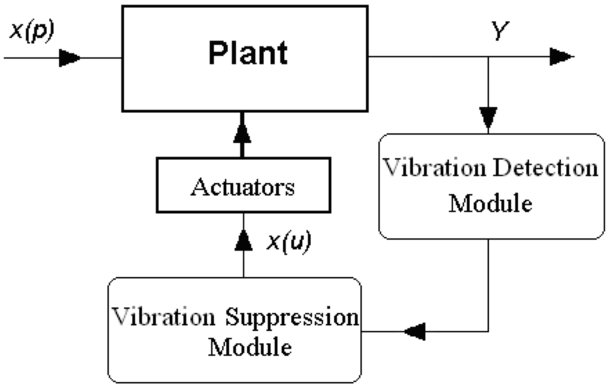


Figure 3-1: The general AVC strategy

In this strategy, the following relation exists:

$$Y = F(x(p), x(u), t) \tag{3-3}$$

The vibration suppression module's task is to synthesize *x(u)* such that it minimizes *Y*. If a comprehensive, differentiable physical model of plant is available, the control signal to the actuator, i.e., *x(u)*, could be determined through an optimization method in order to minimize *Y*. One such optimization method is steepest decent, where:

$$x(u)^{k+1} = x(u)^k - \alpha^k \frac{\partial Y^k}{\partial x(u)^k} \tag{3-4}$$

Here $\frac{\partial Y^k}{\partial x(u)^k}$ is the gradient of the dynamic model of the plant Y; $x(u)^k$ and $x(u)^{k+1}$ are the values of the control signal in the *k* and *k+1* iterations respectively; and *α* is the size of the steps in the direction of minimization.

The calculation of the gradient requires the availability of a differentiable physical model. However, comprehensive, differentiable physical models of complex systems usually do not exist. In this case, using artificial neural networks (ANN) in the chatter suppression module is a good choice because ANNs are known for their function approximation capability. As introduced before, properly designed and trained neural networks are capable of approximating any linear or nonlinear function to the desired degree of accuracy and they are noise tolerant.

In this work, a vibration suppression module is used to generate a control signal to suppress the original vibration at the location of interest. The ideal generated control vibration should have the same amplitude and frequency of the original one at that location but with a 180-degree phase difference in dominant vibration frequencies.

3.3 Vibration Control Subsystem Design

The vibration control module is the most critical part of this control system. Figure 3-2 shows some details of the proposed vibration control subsystem design.

To generate an “opposite” vibration at the location of interest to suppress the original one, an ANN is utilized as an identification model of the plant based on the function approximation capability of ANNs.   To make the proposed AVC system robust, the ANN model should be used for a relatively stable part of the plant. To generate control signals, the ANN model should work as an inverse model, which means the inputs of the ANN model are actually the outputs of the plant, i.e., the parameters of the vibration signal, which include amplitude (*AI*), frequency (*FI*) and phase (*PI*), while the outputs of the ANN are the parameters of the control signal, which include amplitude (*AO*), frequency (*FO*) and phase (*PO*).

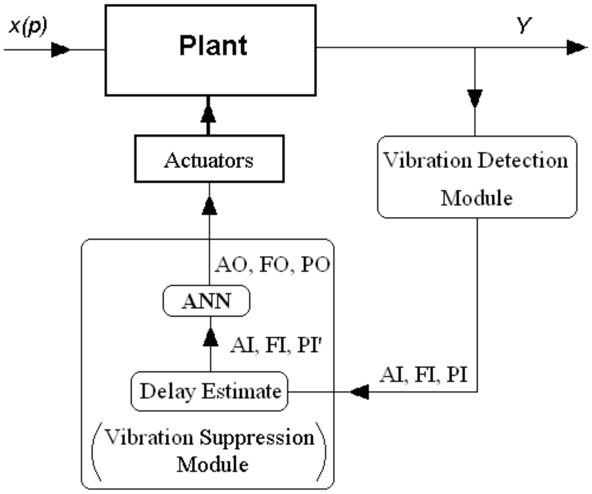


Figure 3-2: Some details of the vibration control module

Time delay in AVC is very critical. To satisfy causality of different iterations, the time delay between the iteration to collect vibration signal parameters and the iteration to send out control signal should be considered to get the actual phase input (PI') to the ANN.

Real-time digital signal processing provides precise and predictable timing characteristics. Because of the deterministic property of a real-time system, the accuracy of running time of a control iteration, or a while loop, can be expected. In the current work, if the running time for each iteration is t, considering the time delay of one control iteration and the 180-degree phase difference, the actual phase input (PI' in Figure 3-2) of the ANN model should be,

$$PI' = PI + 180 + (FI \times t - \text{int}(FI \times t)) \times 360 \qquad (3\text{-}5)$$

3.4 Design of the Inverse ANN Model

As mentioned before, in the proposed AVC system, the ANN is used for function approximation and works as an inverse identification model of a relatively stable part of the plant. The design of the ANN model is based on the applied AVC strategy and the actual experimental setup. Generally, design steps are as follows: First, training data for the ANN models are collected via experiments according to the AVC strategy presented in previous sections. Then, the training data are analyzed in order to choose a proper normalization method. The general network architectures of the ANN models are then designed and the suitable learning algorithm is chosen. Finally, the ANN models are trained to avoid overfitting. The network architectures may be modified for better function approximation based on experimental results.

An ANN model example based on the proposed vibration suppression subsystem design is shown in Figure 3-3. In this example, a multilayer feedforward ANN is utilized.

The ANN architecture used here has three inputs, one hidden layer of log-sigmoid neurons and one output layer of three log-sigmoid neurons.

The ideal control signal frequency (*FO*) should be the same as vibration frequency (*FI*) [29]. Moreover, the input *PI* can be cancelled if the phase difference (*PD*) between the control signal and the vibration signal is utilized (*PD = PI-PO*). In this case, the ANN can be simplified as shown in Figure 3-4. In the detailed design, the number of hidden layers and the number of neurons in each hidden layers are decided by finding out what the best numbers are to obtain the smallest mean square error (MSE) for validation data sets.

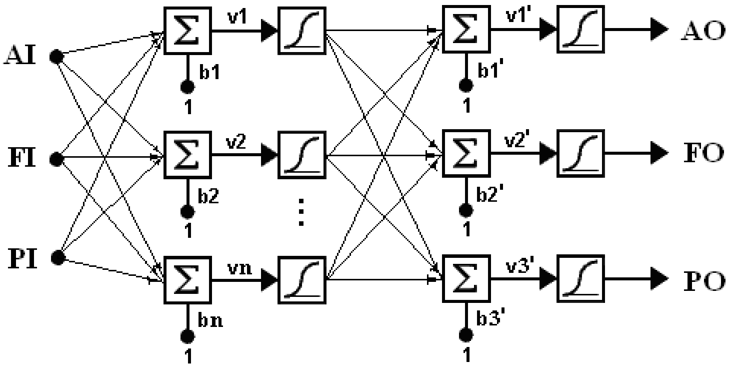


Figure 3-3: ANN model example

In experiments to collect training data for the ANN models, only the control actuator is utilized to generate the plant vibration, i.e., the primary disturbance *x(p)* =0. Therefore, the inputs of the ANN are *AI* and *FO* (*FI* should be the same as *FO*). To get a robust training, which means a training affected minimally by external sources of variability, the experiments to collect training data need to be designed first. In this work, the fractional factorial design [59] is used for the design of experiments to obtain the training data for the ANN models as shown in [29].

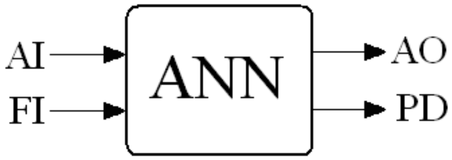


Figure 3-4: Simplified ANN model

Considering the time delay between the iteration to collect vibration signal parameters and the iteration to send out control signal, the phase of the control signal should be:

$$PO = PI' - PD \tag{3-6}$$

where PI' can be calculated from equation (3-5), in which PI is known from experiments.

3.5 Resilient Propagation

In the current work, Resilient Backpropagation is used as the training algorithm for the ANN model because, although it is not the fastest one, theoretically, it can also help to reduce squashing effect of the magnitudes of partial derivatives.

Resilient propagation (RPROP) performs a local adaptation of the weight updates according to the behavior of the error function [55]. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update-value, Δ_{ij}. The update-value for each weight and bias is increased by calculated factors. The adaptive update-value, i.e., Δ_{ij}, evolves during the learning process based on its local sight on the error function *E* according to the following learning rule [55]:

$$\Delta_{ij} = \begin{cases} \eta^{+} * \Delta_{ij}^{(t-1)}, & if \quad \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^{-} * \Delta_{ij}^{(t-1)}, & if \quad \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)}, & else \end{cases} \tag{3-7}$$

where $0 < \eta^{-} < 1 < \eta^{+}$, and w_{ij} is the weight from neuron *j* to neuron *i.*

Described in words, the adaptation rule works as follows [55]: every time the partial derivative of the corresponding weight w_{ij} with respect to *E* changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value is decreased by the factor η^{-}. If the derivative retains its sign, the update-value is slightly increased by factor η^{+} in order to accelerate convergence in shallow regions [55].

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value; if the derivative is negative, the update-value is added [55]:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & if \quad \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)}, & if \quad \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0, & else \end{cases} \tag{3-8}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \tag{3-9}$$

However, there is one exception: if the partial derivative changes sign, i.e., the previous step was too large and the minimum was missed, the previous weight-update is reverted [55]:

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \quad if \quad \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \tag{3-10}$$

Due to that “backtracking” weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update-value, there should be no adaptation of the update-value in the succeeding step.

CHAPTER 4

EXPERIMENTAL SETUP OF AVC

The methodology presented in Chapter 3 was evaluated experimentally through active vibration control (AVC) of a cantilever beam. The experimental setup is described in this chapter.

4.1 Hardware Setup

Figure 4-1 shows the top view of the hardware setup developed for the experiments. The beam is made of plain carbon steel and has the dimensions: 550 mm x 25 mm x 4.5 mm. Two electromagnetic shakers are used to provide the primary disturbance force (shaker 1) and the control force (shaker 2) to the beam. These shakers are located at 145 mm and 373 mm from the clamped end, on each side of the beam, respectively. Since the shakers have significant mass, and mass loading will lower the apparent measured frequency ($\omega = \sqrt{\frac{k}{m}}$), to minimize the effect of the shakers on the structure, the shakers are attached to the beam through stingers. The stingers serve to isolate the shakers from the structure, reduce the added mass, and cause the force to be transmitted axially along the stingers. The primary shaker is attached to the beam firmly; but the control shaker simply pushes up against the beam. The resulting preload is used to maintain contact between the control shaker and the beam. The objective is to minimize the vibration of the beam at the proximity sensor location, which is 498 mm from the clamped end of the beam.

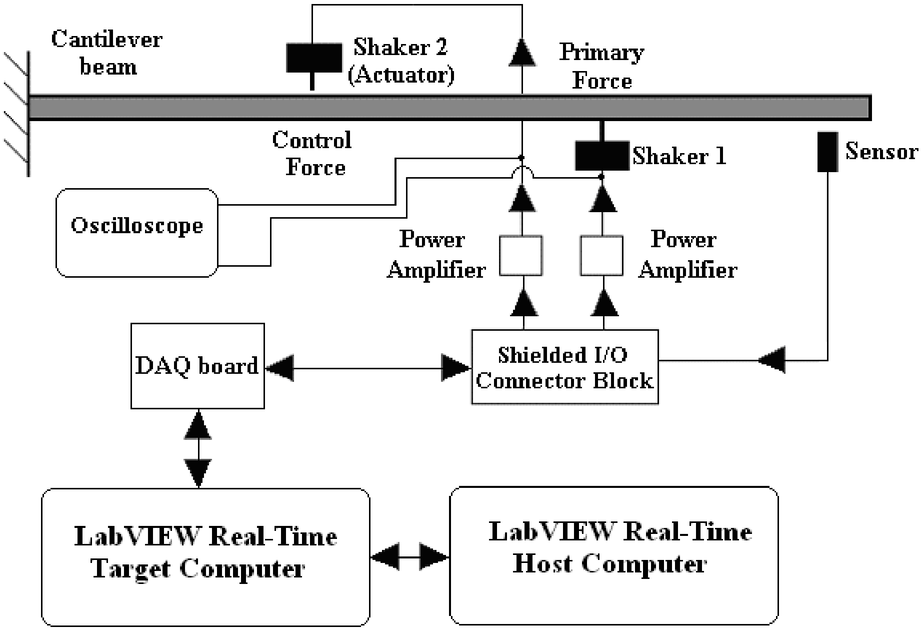


Figure 4-1: Hardware setup for implementing AVC design

Two computers are used in the system because LabVIEW Real-Time Developing System of National Instruments is used as the main developing software in this project. One computer works as a Dedicated Real-Time Target, which integrate the DAQ board for analog, digital, counter/timer, and vibration signals. Programs are developed on a Windows XP host computer and downloaded to the real-time target via Ethernet. Moreover, some time-consuming applications, such as graphing and data recording, are implemented on the host computer instead of the target computer. The target computer runs a single-kernel RTOS (Real-Time Operating System) for maximum reliability.

The data acquisition (DAQ) board has four 12-bit analog inputs with a maximum sampling rate of 5 Mega-Sample (MS)/s. It also provides two 16-bit analog outputs with a maximum update rate of 2.5 MS/s (see Appendix A for the DAQ board specifications). Figure 4-2 is a sketch of the experimental hardware setup, which shows some functions of the DAQ board. Two analog outputs, i.e., “AO 1” and “AO 2”, and one analog input, i.e., “AI”, are used in the experiments.

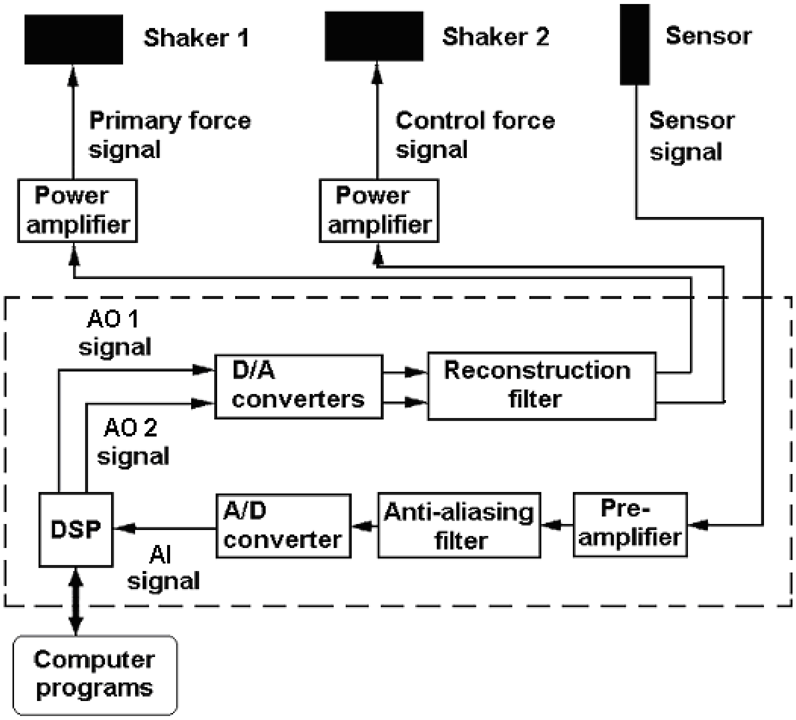


Figure 4-2: Schematic diagram of experimental setup

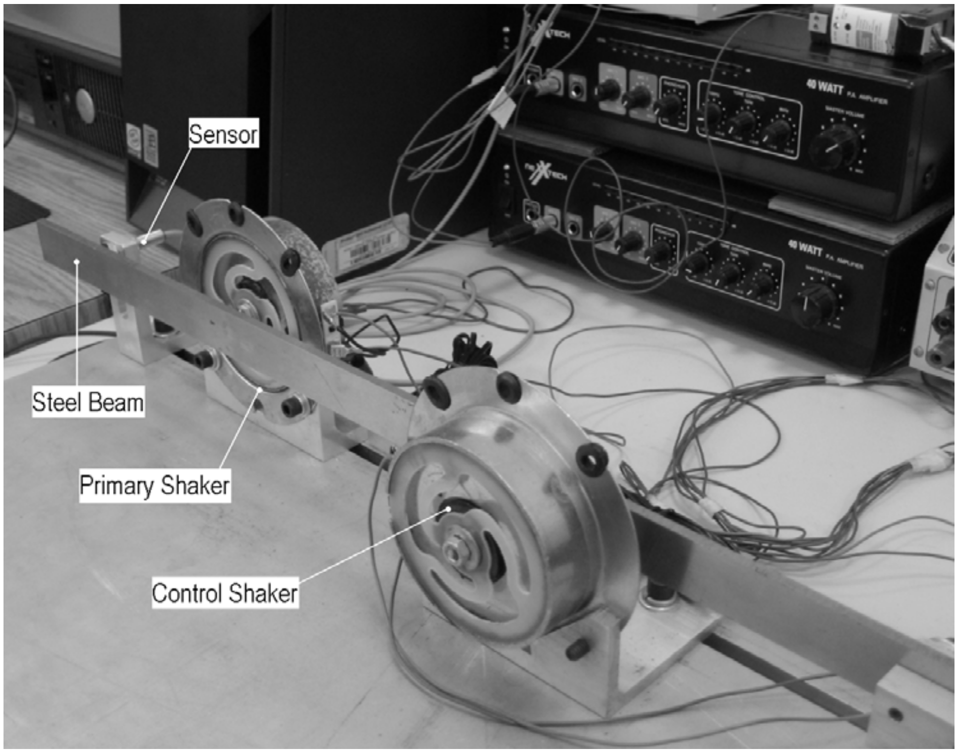


Figure 4-3: Photograph of the experimental hardware setup

The vibration is measured using an inductive proximity sensor (see Appendix B for sensor specifications). The two electromagnetic shakers are driven by two 40W amplifiers. Figure 4-3 shows the photograph of the experimental setup as described.

4.2 Software Environment

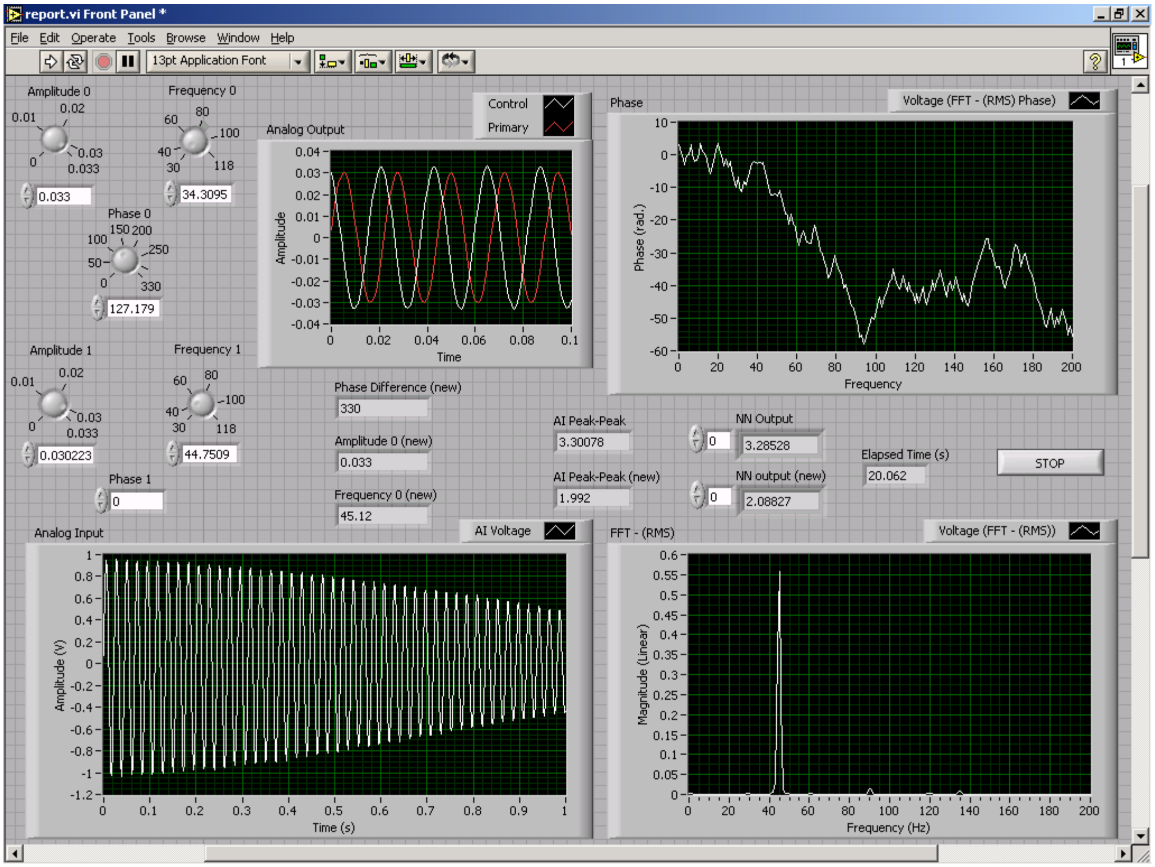
In this work, the operation system on the host computer was Microsoft Windows XP and LabVIEW Real-Time version 8.5.1 on the target computer. LabVIEW Real-Time version 8.5.1 is used for measurement, signal processing and the development of user interfaces. MATLAB 2008 is used for neural network design, implementation, visualization and simulation, as well as data analysis and some graphing. Microsoft Visual C++ 6.0 is used for algorithm implementation and for implementing neural networks in the form of Dynamic Link Library (DLL) files. ANSYS 7.0 is used for the Finite Element Method (FEM) analysis of the beam vibration.

Figure 4-4 shows one example of user interfaces developed in the current work. In this user interface, the amplitudes, frequencies and phases of the primary and control signals can be input by typing or turning the small nubs. The calculated control signal parameters, measured analog input maximum peak-peak amplitudes, an artificial neural network (ANN) model outputs and the elapsed time of the system are displayed numerically. The control, primary and analog input signals are displayed graphically. The two FFT graphs show the phases and magnitudes of the analog input signal.

4.3 Signal Processing

In this work, the Hanning window function is utilized for sinusoidal signals and the uniform window function is utilized for white noise signals to correct leakage.

The sampling rate can be set to up to 2 MS/s while 1000 S/s is set in some cases, e.g., measuring natural frequencies. High sampling rate is good for fast FFT calculation [26] and avoiding aliasing and improving measurement accuracy even with very a short sampling time, such as 25 ms. According to Shannon's sampling theorem, to avoid aliasing, the signal should be sampled at a rate at least two times of the highest frequency in the signal [1]. As will be discussed in the following chapter, the highest frequency in the signals of the current work is less than 100 Hz. Therefore, the sampling rate is good enough to avoid aliasing and improve measurement accuracy even with a very short sampling time. Aliasing can also be avoided in signals containing many frequencies by subjecting the analog signal to an antialiasing filter [5].



Amplitude 0: 0.033
Frequency 0: 34.3095
Phase 0: 127.179
Amplitude 1: 0.030223
Frequency 1: 44.7509
Phase 1: 0
Phase Difference (new): 330
Amplitude 0 (new): 0.033
Frequency 0 (new): 45.12
AI Peak-Peak: 3.30078
AI Peak-Peak (new): 1.992
NN Output: 0 3.28528
NN output (new): 0 2.08827
Elapsed Time (s): 20.062

Figure 4-4: An example of user interfaces used in the current work

4.4 Bending Vibration Analysis of the Beam

In the proposed experimental setup, the vibration of the cantilever beam in the direction perpendicular to its length is considered. Such vibrations are known as transverse or flexural vibrations.

The cantilever beam of this experiment can be considered an Euler-Bernoulli beam because its length/width≥ 10 and length/thickness≥ 10 [1]. Therefore, the effects of shear deformation and rotary inertia are ignored in the vibration analysis.

When the energy dissipation of the beam is taken into consideration, the damping ratios ζ_n (the *n*th modal damping ratio) are chosen based on experience or on experimental measurements. Usually, ζ_n is a small positive number between 0 and 1, with most common values of $\zeta_n \leq 0.05$ [5]. The damped natural frequency is:

$$\omega_{dn} = \omega_n \sqrt{1-\zeta_n^2} \tag{4-3}$$

where ω_n is the undamped natural frequency.

If the energy dissipation of the beam (see Figure 4-5) is ignored, the natural frequency ω_n and the mode shape $X_n(x)$ of the beam can be calculated as [5]:

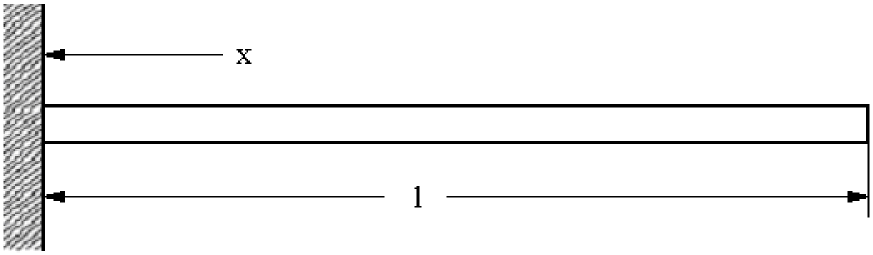


Figure 4-5: A cantilever beam

$$\omega_n = \beta_n^2 \sqrt{EI / \rho A} \tag{4-4}$$

$$X_n(x) = \cosh\beta_n x - \cos\beta_n x - \sigma_n(\sinh\beta_n x - \sin\beta_n x) \qquad (4\text{-}5)$$

$$\sigma_n = \frac{\sinh\beta_n l - \sin\beta_n l}{\cosh\beta_n l + \cos\beta_n l} \qquad (4\text{-}6)$$

where $\beta_n l$ are called the weighted frequencies, l is the length of the beam, E is the Young's elastic modulus for the beam, I is the cross-sectional area moment of inertia, and ρ is the density of the beam.

From [5], the following parameters can be found:

$\beta_1 l$ =1.87510407; $\beta_2 l$ =4.69409113; $\beta_3 l$ =7.85475744; …

σ_1=0.7341; σ_2=1.0185; σ_3=0.9992; …

For the beam used in the current work,

l = 0.55m;

$E \approx 2\text{x}10^{11}\text{N/m}^2$; [56]

I = $(1/12)(2.5\text{x}10^{-2})(4.5\text{x}10^{-3})^3 \approx 1.9\text{x}10^{-10}\ \text{m}^4$; [58]

$\rho \approx 7.84\text{x}10^3\ \text{Kg/m}^3$; [57]

$A = (2.5\text{x}10^{-2})(4.5\text{x}10^{-3}) = 1.125\text{x}10^{-4}\text{m}^2$.

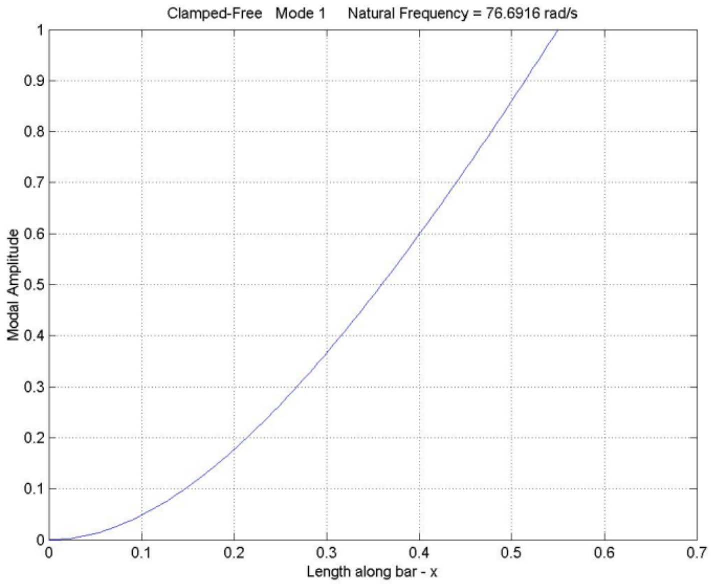
So the first three natural frequencies can be calculated by equation 4-4 as:

$\omega_1 \approx 76.7$ rad/s, $f_1 = \dfrac{\omega_1}{2\pi} \approx 12.20$ Hz

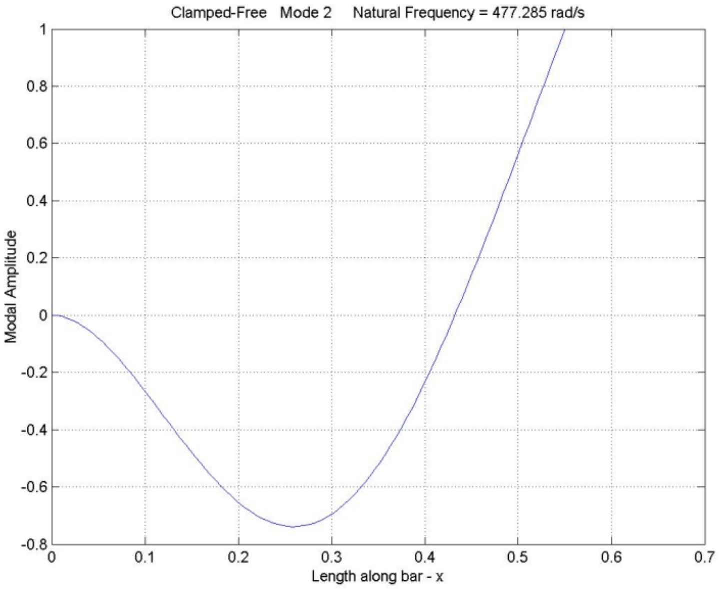
$\omega_2 \approx 477.3$ rad/s, $f_2 = \dfrac{\omega_2}{2\pi} \approx 75.96$ Hz

$\omega_3 \approx 1337.1$ rad/s, $f_3 = \dfrac{\omega_3}{2\pi} \approx 212.80$ Hz

The first three mode shapes are plotted in Figure 4-6.

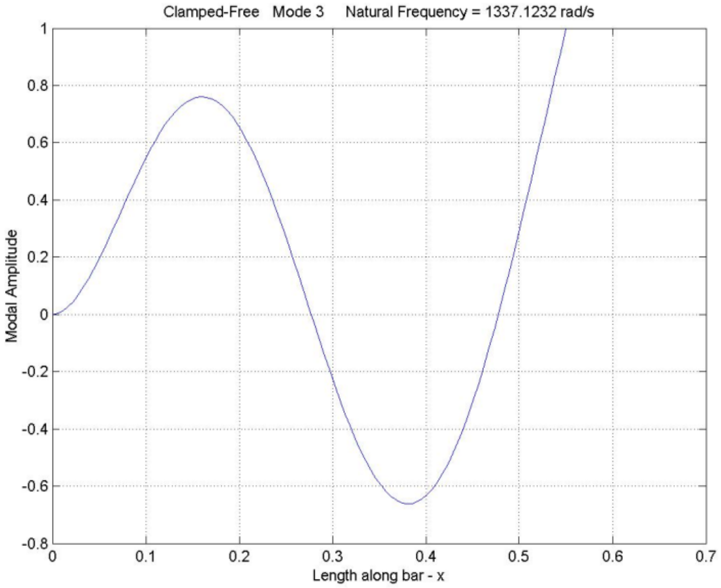


(a) Mode 1



(b) Mode 2

Figure 4-6 (a-b): The first three mode shapes of the cantilever beam



(c) Mode 3

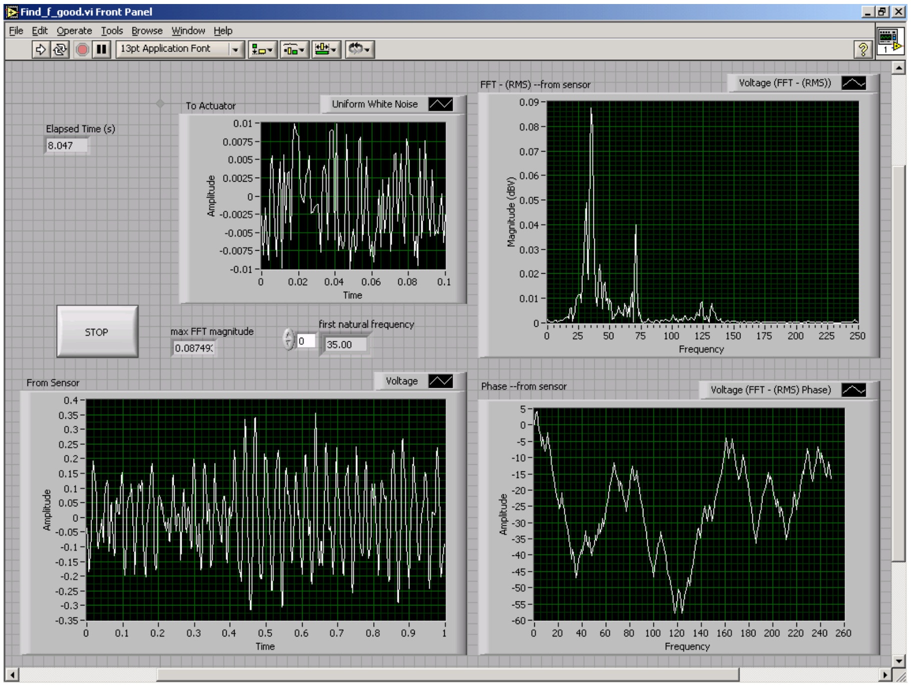
Figure 4-6 (c): The first three mode shapes of the cantilever beam

4.5 Natural Frequencies of the Overall System

The method used to measure the natural frequencies of the whole experimental system including the beam and two shakers is via sending uniform white noise signals to the shakers and analyzing the corresponding Fast Fourier Transform (FFT) responses of the system at the sensor location, as shown in Figure 4-7.

Uniform white noise generates a signal that contains a uniformly distributed, pseudorandom pattern whose values are in the range [-*a*: *a*], where *a* is the absolute value of amplitude [26]. Ideal white noise has equal power per unit bandwidth, resulting in a flat power spectral density across the frequency range of interest. Figure 4-8 shows the FFT response of the system output when uniform white noise (*a* = 0.01V) is sent to the primary shaker alone; to the control shaker alone; and to both shakers respectively. There is almost no jitter for the spectra of (a); but there are many jitters for the spectra of

(b) and some jitters for the spectra of (c). However, the graphs show that the first natural frequency is around 34 Hz and the second natural frequency is around 130 Hz. Because of spectral leakage, 34 Hz and 130 Hz are approximate values of the first and second natural frequencies.



Elapsed Time (s) 8.047

max FFT magnitude 0.08749

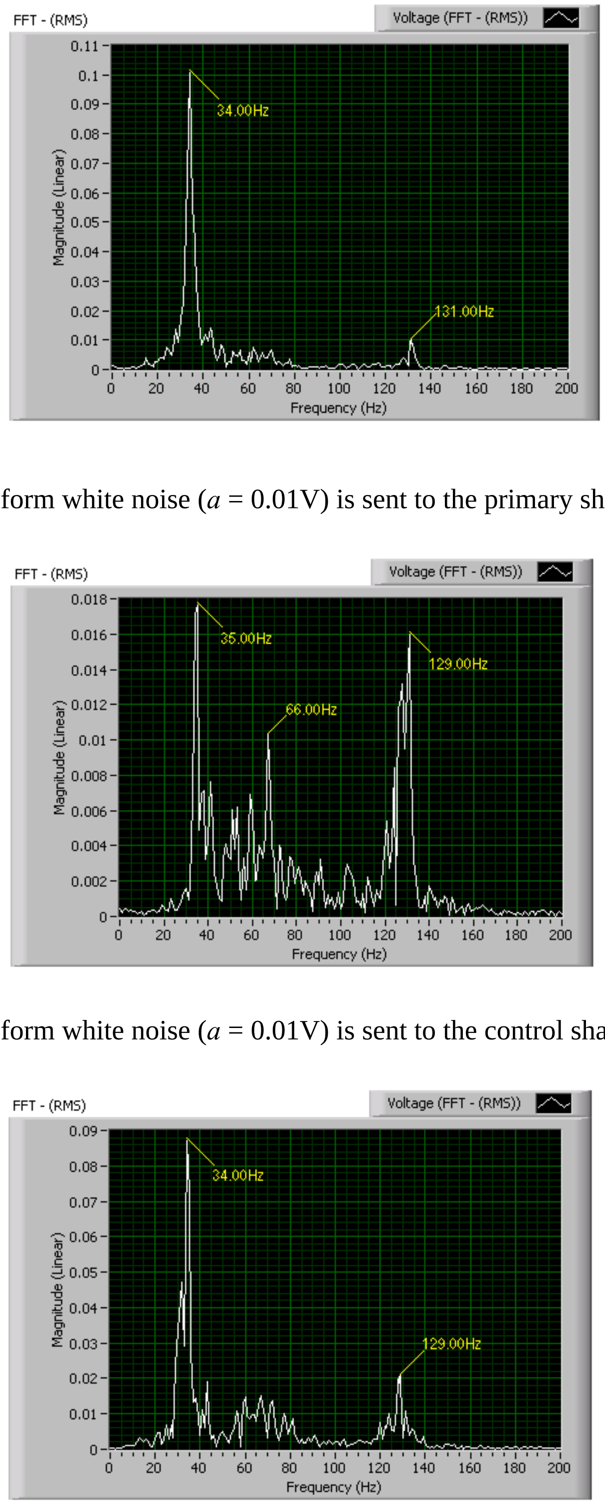
0

first natural frequency 35.00

Figure 4-7: Interface of system natural frequency measurement

The main reason for the complex behavior of the system is the control shaker (actuator). The control shaker makes the FFT spectra of the system output jitter, because it is not firmly attached to the beam. It simply pushes up against the beam (see section 4.1). When the amplitude of vibration of the beam is large, the control shaker may lose contact with the beam, and therefore makes the system more complicated by introducing nonlinearity.

To test the efficiency of this natural frequency detection system, a magnet of 145 grams was attached to the beam at 280 cm and 518 cm from the clamped end of the beam separately. After running the program, the new first natural frequencies changed to 32Hz and 25 Hz respectively, as shown in Figure 4-9.



(a) Uniform white noise (*a* = 0.01V) is sent to the primary shaker only

(b) Uniform white noise (*a* = 0.01V) is sent to the control shaker only

(c) Uniform white noise (*a* = 0.01V) is sent to both shakers

Figure 4-8: Measuring the system natural frequencies via experiments

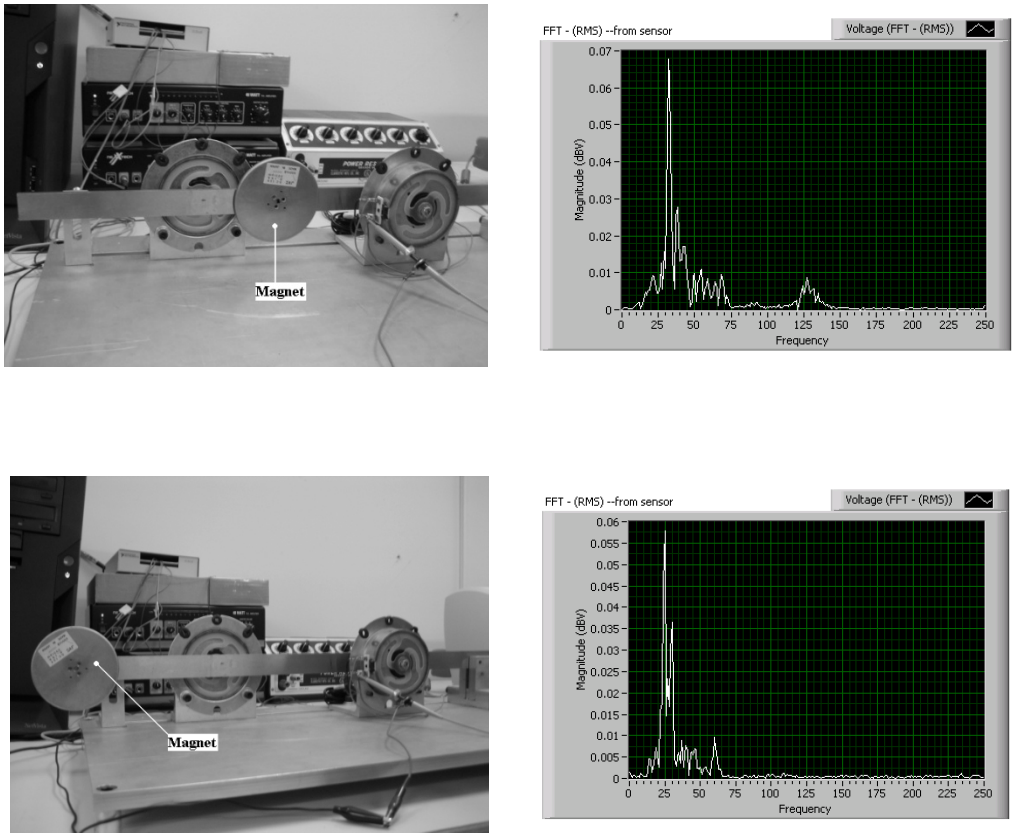


Figure 4-9: First natural frequency measurement outcomes

4.6 Introducing Nonlinearity to the System

For the purpose of testing performance of the system for nonlinear control problems, nonlinearity is introduced into the experimental arrangement. This can be done in two different ways (see Figure 4-10) in the experimental setup. The first way is by not attaching the control source shaker to the beam, but simply pushing it up against the beam. The resulting preload is used to maintain contact between it and the beam. By increasing the driving force of the primary disturbance, the control shaker must also drive harder to suppress the primary disturbance, which in this case will cause the control shaker to rattle as it loses the contact with the beam, and therefore will make the resultant error signal spectrum “noisier”.

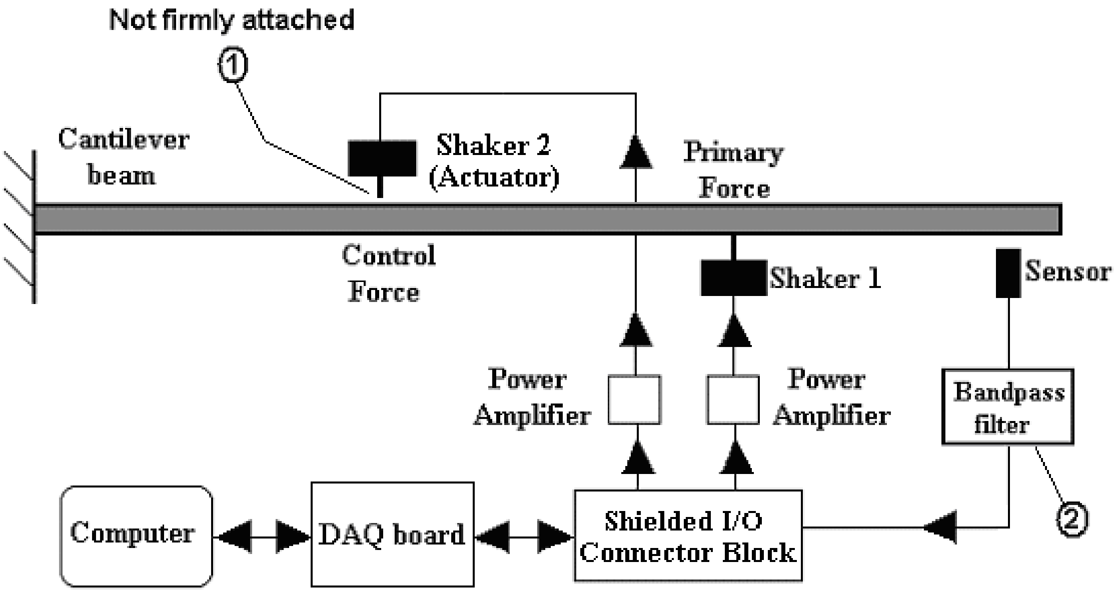
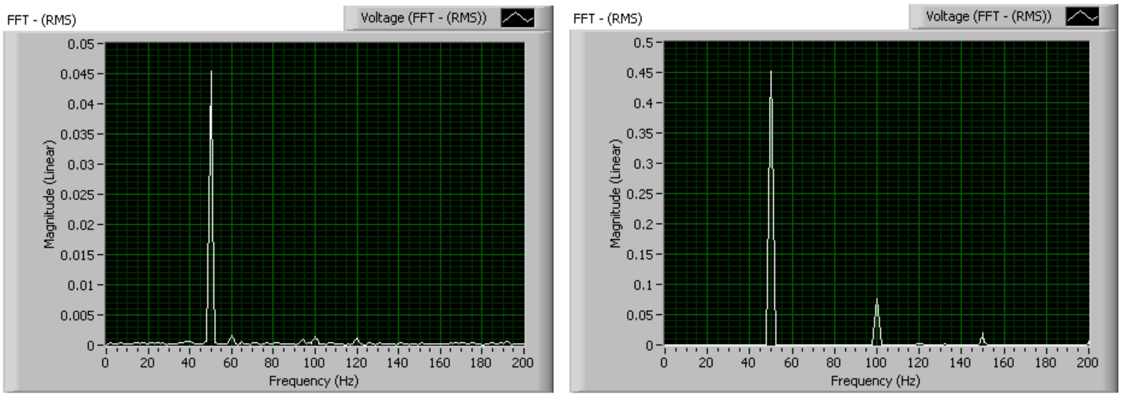


Figure 4-10: Two ways to introduce nonlinearity

The effect of the first way is shown by the two graphs in Figure 4-11, which are produced when a sinusoidal signal (50Hz) is sent to the control shaker only. When the control shaker is driven soft (0.005V), the resultant error signal spectrum is very clean, as shown in Figure 4-11 (a). Only the 50Hz peak (magnitude > 0.046) is evident, and all the other peaks are too small (magnitudes < 0.002) to be compared with this one. The small 60Hz peak is due to electrical noise, the small 100Hz peak is the first harmonic of 50Hz signal and the small 120Hz peak is the first harmonic of the electrical noise. However, when the control shaker is driven harder (0.1V), the resultant error signal spectrum becomes “noisier” with the comparable harmonics of the signal, as shown in Figure 4-11 (b).

The second way for introducing nonlinearity is by bandpass filtering the analog input signal from the sensor to provide a slight bias to the higher frequency harmonics, e.g., 100 Hz, thus exaggerating the relative importance of the harmonics in the spectrum.



(a) driven soft (*a* = 0.005V)

(b) driven hard (*a* = 0.1V)

Figure 4-11: Spectrum comparison between driving the control shaker soft and hard

CHAPTER 5

EXPERIMENTAL IMPLEMENTATION OF AVC

5.1 AVC System Design

Based on the methodology and the experimental setup, the designed AVC system to generate a control signal is shown in Figure 5-1. This control system can modify the control signal online in every control cycle (one control cycle could include one or more than one control iterations) in the following way:

1. The system can repeat all calculations in one control iteration and generate a current control signal (e.g., with parameters of FO, AO, and PO) based on the detected vibration situation (e.g., FI, AI, and PI) of this current iteration;

2. As shown in Equation (5-1), the current control signal is added to the accumulated control signal, i.e., (control signal)$_n$ of the nth control cycle, which is a combination of all previous continuous control signals, to get an updated control signal, i.e., (control signal)$_{n+1}$, in the Signal Combination module (the control signal of the first control cycle is zero);

$$(\text{control signal})_{n+1} = (\text{control signal})_n + \text{current control signal} \tag{5-1}$$

3. The new updated control signal, i.e., (control signal)$_{n+1}$, is sent out to the actuator at the beginning of the next iteration; and

4. At the same time, this updated control signal becomes the "accumulated control signal" in the next control cycle.

Therefore, the actual control signal sent to the actuator is an accumulation of all previous generated control signals, which are all continuous. One control cycle should include more than one control iterations to avoid unstable transient conditions after the modification of the actual control signal and to get more accurate measurements of vibration. In current work, a control cycle could be 25 ms, which means a new control signal could be generated as fast as in every 25 ms in the current setup.

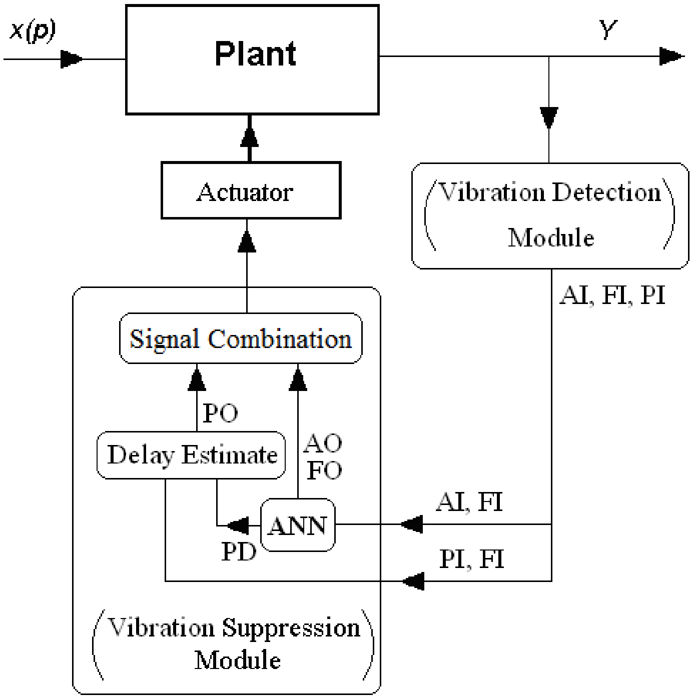


Figure 5-1: The AVC system in current work

The parameters of the control signal, i.e., AO, FO, and PO, can be obtained from ANN outputs and the equations in the previous section. The phase difference between the control signal in programming and the actual control signal at the connector block is considered because the difference may not be the same when the system restarts.

5.2 ANN Training Data

As discussed in Chapter 3, after the control strategy is set up, the first step to design the ANN model is to collect training data based on the control strategy and experimental setup. The training data should be collected via experiments based on the control strategies and the experimental setup. To get a robust training, which means a training affected minimally by external sources of variability, the experiments to collect training data need to be designed first.

Generally, experiments are used to study the performance of processes and systems. The process or system can be represented by the model shown in Figure 5-2. The process transfers some inputs into an output that has one or more observable responses [59]. Some of the process variables (factors) *x*1, *x*2, …, *xp* are controllable, whereas other variables *z*1, *z*2, …, *zq* are uncontrollable.

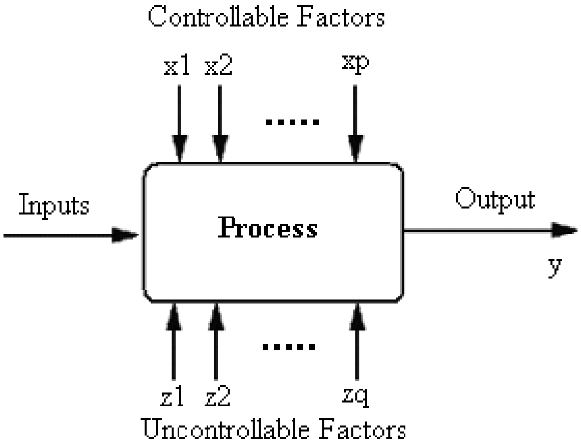


Figure 5-2: General model of a process or system [59]

The objectives of the experiment may include the following:

1. Determining which factors are most influential on the response *y*;
2. Determining where to set the influential *x*’s so that the response *y* is almost always near the desired nominal value;
3. Determining where to set the influential *x*’s so that variability in *y* is small;
4. Determining where to set the influential *x*’s so that the effects of the uncontrollable variables are minimized.

The general approach to planning and conducting the experiment is called the experimental design. There are several strategies that an experimenter could use, such as best-guess approach, one-factor-at-a-time, factorial design and fractional factorial design [59].

The fractional factorial design is a variation of basic factorial design in which only subsets of the runs are made and it considers the interaction among the factors [59]. In this project, the fractional factorial design is used for the design of experiments to obtain the training data for the ANN model. The objective of the experiments in the current work is to determine the influence of the controllable factors, i.e., control signal parameters, i.e., AO, FO and PO, to the response, i.e., the sensor signal parameters, i.e., AI, FI and PI. Therefore, there should be three controllable factors, i.e., AO, FO and PO, with different levels. In this case, an orthogonal array for three factors with different levels should be used in the experiment design. However, as introduced in section 3.4, by using phase difference, i.e., PD, the controllable factors can be reduced to two, which are AO and FO. For only two factors with different levels, the design of experiments is straight, which is to find all combinations of the two factors with different levels.

The data ranges are decided by the regions of interest for each variable and hardware performance limitations. After some modifications to strengthen stingers of the experimental setup (as shown in Figure 4-1), the first natural frequency of the system was found to be around 38 Hz and the second natural frequency was around 137 Hz. Since the frequency response ranges for the two amplifiers are 20 Hz to 30 KHz, and the vibration frequency range of interest is 25 Hz to 55 Hz in the current work, the frequency range 25 Hz to 55 Hz is used for the control signal. To find out the proper amplitude ranges for the signals sent to the shakers, many experiments were undertaken. Considering the measurement range of the sensor and the hardware setup, according to the results of these experiments, the peak amplitude range for the control signal sent to the actuator is set to from 0.002 V to 0.038 V for frequencies around the first natural frequency of the system and from 0.002 V to 0.1 V for other frequencies. For frequencies around the first natural frequency of the system, the vibration amplitude of the beam at the sensor location increases much faster with the increase of the control signal amplitude and the steel beam may hit the sensor if the control signal amplitude is larger than 0.038 V. The phase difference range can be set to from -180 degree to 180 degrees.

To reduce harmful effects, e.g., the squashing effect, of using sigmoid transfer functions in the hidden layer and the output layer of the ANN model, and normalization, the above data ranges can be divided into several sub-ranges, e.g., the original frequency range can be divided into three smaller sub-ranges: 25 Hz to 35 Hz, 35 Hz to 40 Hz and 40 Hz to 55 Hz. The control signal amplitude range sent to the actuator is set to from 0.002 V to 0.038 V for the frequency range 35 Hz to 40 Hz and from 0.002 V to 0.1 V for the other two frequency ranges. ANN models are trained separately for different sub-ranges. Moreover, Resilient Backpropagation (RPROP) algorithm is utilized to train ANN models because, although it is not the fastest one, theoretically, it can also help to reduce squashing effect of the magnitudes of partial derivatives.

5.3 Normalization

The values of all the data must be normalized for efficient processing by the ANN (see section 2.4.5). In the current work, all the experimental input and output data are positive values. The data are normalized to a range of 0.1 to 0.9 by using the following equation:

$$x_{scaled} = 0.8 * \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \tag{5-2}$$

where x is the real value, x_{scaled} is the normalized value, $x_{\min}$ is the minimum value and $x_{\max}$ is the maximum value of one input or output.

5.4 ANN Architecture

In this work, multilayer feedforward ANNs are utilized (see section 2.4.1). To find the suitable number of neurons in the hidden layer, many experiments have been done to see how many neurons in the hidden layer can provide the smallest Mean Square Error (MSE) for validation data sets (see section 2.4). Early stopping and Bayesian regularization methods are used in MATLAB to improve generalization in these experiments (see section 2.4.6). Test data sets are also used to compare different models.

The best ANN architecture found via experiments for the AVC system is similar to the example shown in Figure 3-3. It is a multilayer feedforward ANN, which has two inputs (i.e., AI and FI), one hidden layer of 12 log-sigmoid neurons and one output layer of two log-sigmoid neurons for two outputs (i.e., AO and PD). The smallest MSE obtained for validation data sets of this architecture is 0.058% in the current work. The output layer uses a log-sigmoid transfer function because the outputs of the ANNs are supposed to be constrained to a range of 0 to 1 and it is a good choice in the architecture for the current experiment setup [5]. This ANN architecture provided the smallest Mean Square Error (MSE) and has very good performance for generalization in experiments. For the same experimental setup, the ANN architecture did not change, but the weights between neurons changed for different data sub-ranges after training.

5.5 Training Algorithms

A number of different training algorithms were examined. The training speed is not very critical in the current work, because after the ANN is trained, it is then used as a part of the controller and is not modified in the control process. Moreover, when using early stopping to improve generalization, an algorithm that converges too rapidly should not be used [19].

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, since they compress an infinite input range into a finite output range [19]. Sigmoid functions are characterized by the fact that their slope must approach zero, as the input gets large. This causes a problem when using steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values [19]. In MATLAB, the resilient backpropagation training algorithm is used to eliminate these harmful “squashing” effects of the magnitudes of the partial derivatives.

The resilient backpropagation training algorithm was found to be a good choice since it can help to reduce the harmful effects of using sigmoid transfer functions in the hidden layer and output layer of the ANN model.

After the ANN model is trained, it works in the controller of the AVC system as shown in Figure 5-1. The performance of this AVC system has been evaluated experimentally.

5.6 Experimental Results

In order to evaluate the performance of the AVC system experimentally, a noisy sinusoidal signal was sent to the primary shaker to generate beam vibration. The controller was turned on several seconds after the start of the vibration to allow steady state to prevail.

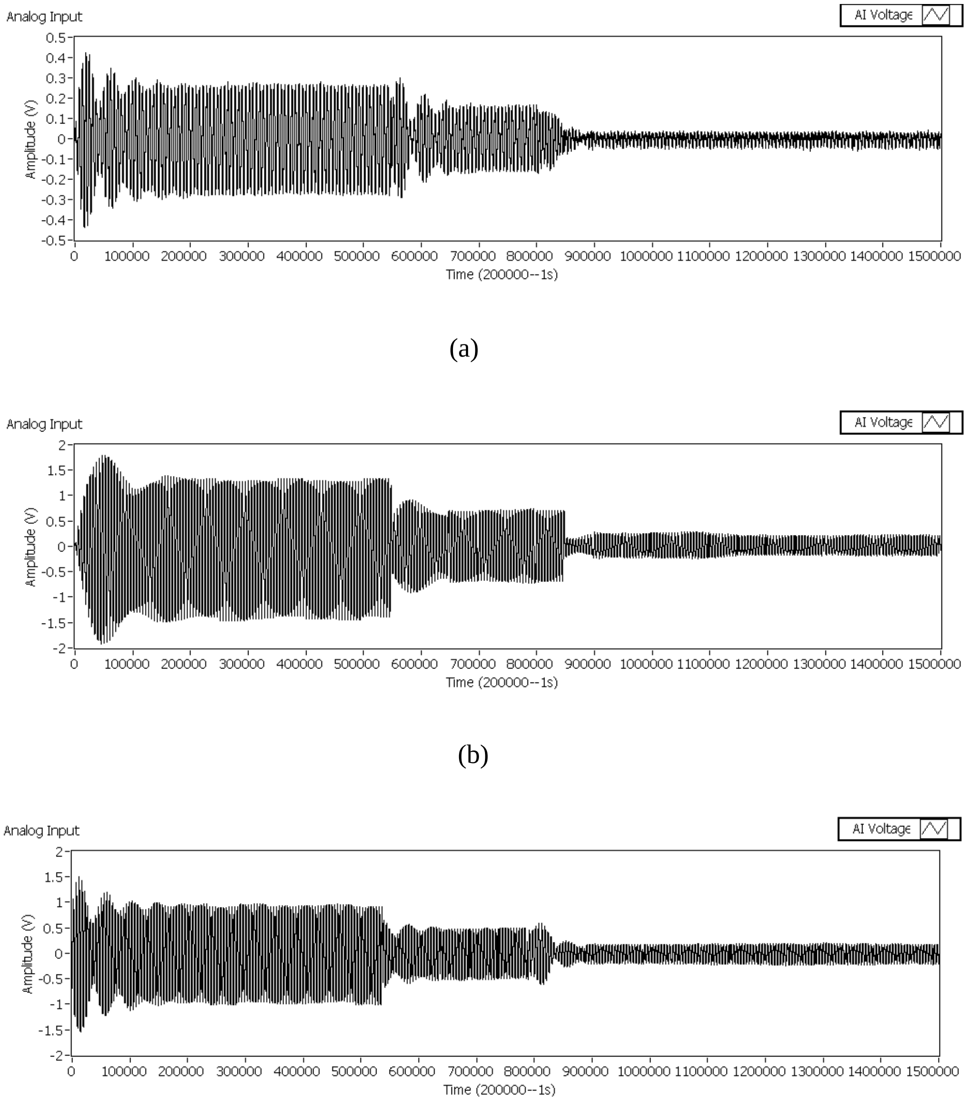
All the analog input and analog output signals, and FFT (magnitude and phase) are displayed on user interfaces graphically only on a host computer and let the target computer work as a dedicated real-time system. The sampling rate for data analysis was 200000 Hz.

Figure 5-3 shows five examples of the beam vibration at the sensor location in the first 7.5 seconds. The figures are grabbed from a user interface directly. The Y-axis of Figure 5-3 shows amplitudes of sensor signals, which are analog inputs of the data acquisition system. In all the experiments, the primary shaker was driven with a primary noisy sinusoidal signal from the beginning. After about 2.75 seconds, a control signal was generated and sent to the control shaker, but with only about a fraction, e.g., around 70%, of the calculated amplitude to get some vibration remained for a second control signal to check out the adaptability of the AVC. Then, after about 1.5 seconds a new control signal was generated based on the current vibration status and added to the original control signal sent to the control shaker.

In Figure 5-3 (a), for the primary signal, the frequency is about 32.33Hz, the amplitude is about 0.038V, and the signal-to-noise ratio (S/N) is about 40; In Figure 5-3 (b), for the primary signal, the frequency is about 38.38Hz, which is close to the first natural frequency, the amplitude is about 0.028V, and the S/N is about 40; In Figure 5-3 (c), for the primary signal, the frequency is about 43.58Hz, the amplitude is about 0.070V, and the S/N is about 35; In Figure 5-3 (d), for the primary signal, the frequency is about 33.07Hz, the amplitude is about 0.099V, and the S/N is about 38; In Figure 5-3 (e), for the primary signal, the frequency is about 28.88Hz, the amplitude is about 0.059V, and the S/N is about 35.

Figure 5-3 (a) shows that the amplitude of the vibration was reduced from about 0.58V (peak to peak) to about 0.1V, which represents about 82.7% reduction of the beam vibration at the sensor location; Figure 5-3 (b) shows that the amplitude of the vibration was reduced from about 2.88V to about 0.5V, which represents about 82.6% reduction of the beam vibration at the sensor location; Figure 5-3 (c) shows that the amplitude of the vibration was reduced from about 2V to about 0.45V, which represents about 77.5% reduction of the beam vibration at the sensor location; Figure 5-3 (d) shows that the amplitude of the vibration was reduced from about 1.8V (peak to peak) to about 0.32V, which represents about 82.2% reduction of the beam vibration at the sensor location;

Figure 5-3 (e) shows that the amplitude of the vibration was reduced from about 0.65V (peak to peak) to about 0.13V, which represents about 80% reduction of the beam vibration at the sensor location. Generally, around 80% reduction in vibration amplitudes can be achieved in all experimental results in this work.

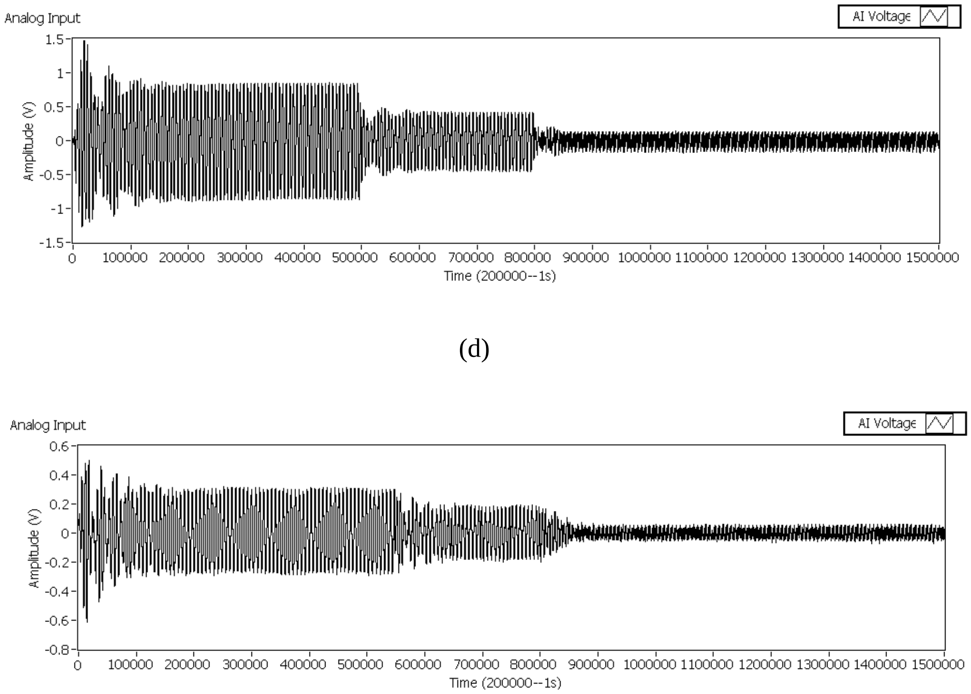


(a)

(b)

(c)

Figure 5-3 (a)-(c): Beam vibration control outcome examples



(d)

(e)

Figure 5-3 (d)-(e): Beam vibration control outcome examples

For the same primary noisy sinusoidal signal as in the above five examples, without reducing the first calculated control signal amplitude, the system can get almost the same vibration amplitude reduction right after the first control signal. These examples show the online adaptive ability of the system.

Experimental results have also shown that the designed AVC system eliminates the sensitivity to time delays. Some experiments have been executed by changing the position of the control shaker or the point of interest, i.e., the sensor location, and therefore changing the time delays. After each modification, an initialization program can be run to collect training data and train the ANN model automatically based on the new experimental setup, and therefore absorbs the information of new time delays. After retraining, the AVC can work as well as before. For example, Figure 5-3 (a) shows a reduction of 82.7% in vibration amplitude when the primary vibration of the beam was at 32.33 Hz. After moving the sensor to another location (478 mm from the clamped end of the beam) and retraining the ANN, the AVC system can still get a reduction around 82%. The experimental outcomes did not show reduction of the AVC system ability caused by time delay changes. Some other random changes in time delay, i.e., the position change of the primary shaker, do not reduce the control ability of a given AVC system, which means the ANN model does not need to be retrained.

Many papers and books, e.g., [6], have already demonstrated the ability of ANN control systems to deal with nonlinearity because of the nature of ANNs. Although the ANN used in the current work is not designed for dealing with nonlinearity, the AVC system has proved to be able to deal with nonlinearity as long as the vibration frequency can be measured accurately. In the above experiments, as mentioned before, the primary shaker can also be simply pushed up against the beam to introduce nonlinearity into the experimental setup. By increasing the driving force of the primary disturbance, the primary shaker rattles as it loses the contact with the beam, and therefore will make the resultant error signal spectrum noisier. To check the ability of the AVC system in dealing with nonlinearity, some experiments were completed with the primary shaker pushed up against the beam. The results of these experiments are almost the same as the results shown above. In most cases, for the same inputs, the outcome differences of the two different setups are within 8%. For example, when the frequency of the primary signal is about 45.88Hz and the amplitude is about 0.27V, a reduction of 57.9% of the beam vibration at the sensor location was obtained when the primary shaker was simply pushed up against the beam; when the primary shaker was attached to the beam, for the same primary signal, the reduction was around 59%.

5.7 Discussion

The experimental results show that the proposed AVC system works effectively. The ANN controller of the modified AVC system can reduce the root mean square (RMS) vibrations by about 80%. The reductions in the RMS vibrations have a very significant effect on the fatigue life of a structure in practical application. Generally, reducing the RMS vibrations by just 10% doubles the fatigue life [25].

By using a real-time environment, the designed AVC system was used to repeat the vibration detection and control loop in every 25 ms and worked very well. The system is real-time adaptable. The repetition of adding new control signals can also be set up at any specific time during online control.

The AVC system is also robust when the experimental setup changes. When the setup changes, the AVC system can collect training data and train the ANN model automatically via running a calibration or initialization program and then the system is prepared for AVC of the new setup.

At the present time, the AVC system has proved to be able to deal with noisy sinusoidal vibrations. Its ability to deal with more complicated signals can be tested in the future.

CHAPTER 6

APPLICATION TO CHATTER SUPPRESSION

As mentioned previously, the objective of the current work is to develop an effective adaptable real-time online AVC system to detect and suppress the noisy sinusoidal vibration of a cantilever beam, and utilize the similar techniques in machining chatter control. The designed AVC system has proved to be able to deal with noisy sinusoidal vibrations effectively, as introduced in previous chapters. In the next few chapters, the methodology for design and implementation of such an active chatter suppression (ACS) system will be presented.

6.1 Chatter Detection Methodology

As mentioned in Chapter 2, it is widely known that machining chatter signals have harmonic shapes, and their frequencies are around the respective natural frequencies of the machining systems [32]. As observed by experimental results of many researchers, the frequency of a regenerative chatter is around the first natural frequency of the machining system and in most cases slightly lower than the natural frequency [32, 35]. Therefore, the vibration detection techniques used in the designed AVC system can be utilized as a part of the chatter detection sub-system of the proposed ACS system.

As reviewed in section 2.9.2, over the years, various techniques for on-line detection of chatter have been studied to detect chatter rapidly and accurately. In most cases, suitable threshold values need to be determined based on experimental results. The chatter detection sub-system should be able to decontaminate the sensor signal and recognize the chatter omen.

6.1.1 Chatter Detection Sub-System for Two Actuator/Sensor Pairs

Considering current research objectives, the accelerations of both horizontal and vertical directions from the bending vibration of the cutting tool should be measured. Using the coherence function between the two accelerations provides a good choice because it is simple and yet effective [38]. This approach provides an easy way of setting threshold values for chatter detection because it has been found that the two accelerations are highly correlated at the chatter frequency, resulting in a sharp increase in their coherence function to a value approaching unity [38].

As introduced in Chapter 1, the proposed ACS system could utilize two actuator/sensor pairs: one pair to control motions parallel to the axis of the workpiece (x-direction) and the other pair to control motions in the direction tangent to the machined surface (y-direction). The horizontal (x-direction) & vertical (y-direction) accelerometers (the sensors) would be glued to the cutting tool at the location of the insert, i.e., the location of interest. By using two sensors, two crossed accelerations, i.e., horizontal and vertical accelerations, from the bending vibration of the cutting tool at the location of the insert will be measured.

The proposed chatter detection sub-system is shown in Figure 6-1. There are four inputs of the chatter detection sub-system:

1. Measured horizontal acceleration signal: x-signal;
2. Measured vertical acceleration signal: y-signal;
3. Measured first natural frequency of the tool system in the horizontal direction: f_{Nx} ;
4. Measured first natural frequency of the tool system in the vertical direction: f_{Ny} .

The outputs of the chatter detection sub-system include the following:

1. Horizontally oriented chatter frequency f_x ;

2. Vertically oriented chatter frequency f_y;

3. Horizontally oriented chatter amplitude: A_x;

4. Vertically oriented chatter amplitude: A_y;

5. Horizontally oriented chatter phase: P_x;

6. Vertically oriented chatter phase: P_y.

As shown in Figure 6-1, the measured horizontal and vertical acceleration signals, i.e., *x*-signal and *y*-signal, have been converted to corresponding digital signals, i.e., *x*-digital signal and *y*-digital signal. Then, after Digital Signal Processing (DSP), the value of the coherence function of *x*-digital signal and *y*-digital signal, i.e., γ_{xy}^2 and the amplitude and phase FFTs of both *x*-digital signal and *y*-digital signal can be obtained. The system will compare the maximum of γ_{xy}^2, i.e., $(\gamma_{xy}^2)_{\max}$, in the whole frequency range, with a threshold, e.g., *T* (based on experimental results, *T* should be close to unity), in each control iteration, as shown in Figure 6-1. If $(\gamma_{xy}^2)_{\max}$ is bigger than the threshold, which means that $(\gamma_{xy}^2)_{\max} \approx 1$, the system will find out the corresponding frequency of the $(\gamma_{xy}^2)_{\max}$ value, i.e., f_m.

The range of the chatter frequency is estimated by using the measured first natural frequencies of the tool system in the horizontal and vertical directions, i.e., f_{Nx} and f_{Ny}. In Figure 6-1, the range is ($f_{min,}$ f_{max}), where f_{min} is M_l times the smaller one of f_{Nx} and f_{Ny}, and f_{max} is M_h times the bigger one. M_l and M_h are the coefficients chosen based on experimental results, e.g., $M_l = 0.7$ and $M_h = 1.1$.

Then, if f_m falls in the range, which means $f_{min}< f_m < f_{max}$, the system will calculate the six outputs, i.e., f_x, f_y, A_x, A_y, P_x and P_y. f_x and f_y are the corresponding frequencies in x-direction and y-direction which have the maximum FFT amplitudes. A_x, A_y, P_x and P_y

can be obtained based on FFT calculations.

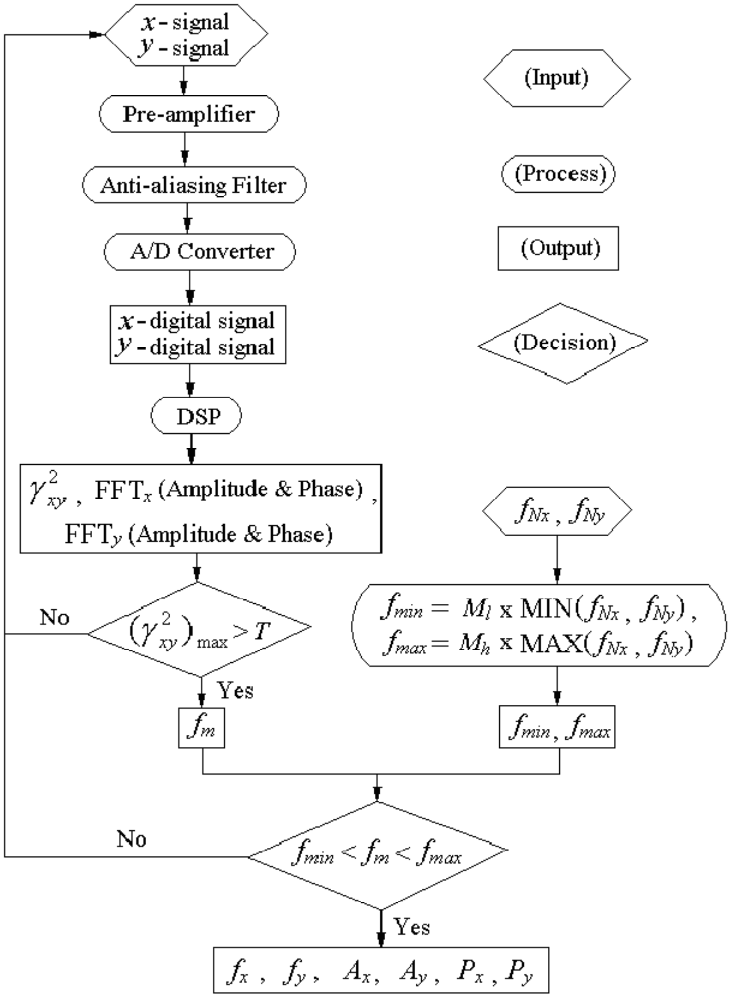

x - signal
y - signal
Pre-amplifier
Anti-aliasing Filter
A/D Converter
x - digital signal
y - digital signal
DSP
γ^2_{xy}, FFT$_x$ (Amplitude & Phase), FFT$_y$ (Amplitude & Phase)
No
$(\gamma^2_{xy})_{\max} > T$
Yes
f_m
f_{Nx}, f_{Ny}
$f_{min} = M_l$ x MIN(f_{Nx}, f_{Ny}), $f_{max} = M_h$ x MAX(f_{Nx}, f_{Ny})
f_{min}, f_{max}
No
$f_{min} < f_m < f_{max}$
Yes
f_x, f_y, A_x, A_y, P_x, P_y
(Input)
(Process)
(Output)
(Decision)

Figure 6-1: The proposed chatter detection sub-system for two actuator/sensor pairs

6.1.2 Chatter Detection Sub-System for One Actuator/Sensor Pair

When there is only one actuator/sensor pair, chatter can be detected when a pre-specified threshold is reached. The parameters of the chatter signals will be obtained and then used by the chatter suppression sub-system. However, it is difficult to determine the proper threshold values beforehand.

In [36], an on-line chatter detection methodology is proposed based on the maximum likelihood algorithm using wavelets. A cutting process is assumed as a nearly-1/f process and γ, which is the spectral parameter related to the self-similarity parameter, is used as a chatter index. To find the proper threshold value of γ, surface roughness data were recorded with a profilometer and used as a basis for determining the chatter occurrence. The proposed proper threshold value of γ for reliable chatter detection for turning processes is -0.5. Moreover, it is claimed that the proposed index is independent of cutting conditions, and is accurate and appropriate for on-line chatter detection.

The on-line chatter detection methodology in [36] can be utilized when only one actuator/sensor pair is used in one direction, e.g., the vertical direction, as shown in Figure 6-2. The only input of the chatter detection sub-system is the measured vertical acceleration signal: y-signal. The outputs of the chatter detection sub-system are the measured frequency f_y, amplitude A_y and phase P_y when chatter is detected.

As shown in Figure 6-2, in every control iteration, the measured vertical acceleration signal, i.e., y-signal, has been converted to a corresponding digital signal, i.e., y-digital signal. After Digital Signal Processing (DSP), which includes the applied algorithm in [36] and FFT analysis, the value of the chatter index γ of y-digital signal and the amplitude and phase FFT's of y-digital signal can be obtained. When γ is bigger than -0.5, the sub-system will send out the measured frequency f_y, amplitude A_y and phase P_y, which are the corresponding parameters with the maximum value in the amplitude FFT of y-digital signal in that control iteration.

There are other methods to detect chatter, as reviewed in section 2.9.2. For most of these, the critical point is to choose a threshold based on experiments. The researchers provided some suitable threshold values based on their experiment results. Most proposed threshold values are also claimed to be independent of cutting conditions, accurate and appropriate for on-line chatter detection.

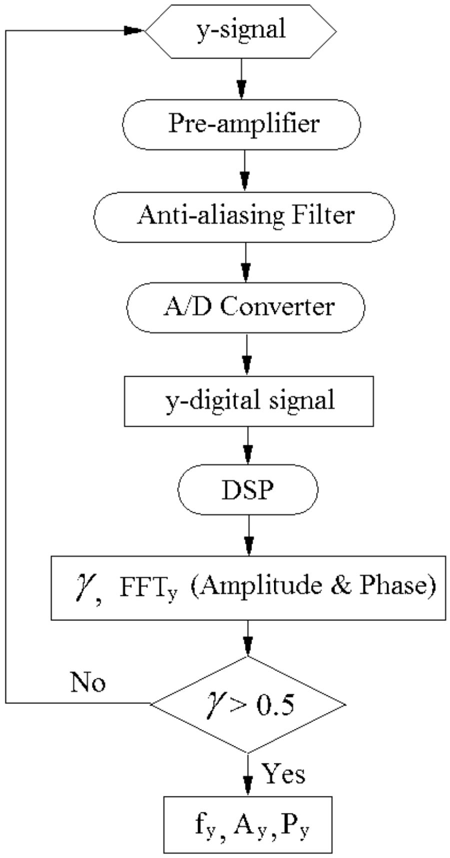


Figure 6-2: The proposed chatter detection sub-system for one actuator/sensor pair

6.2 Chatter Suppression Methodology

As stated previously, the objective of the current work is to develop a real-time online active chatter suppression (ACS) system, which would be effective, adaptable, simple to implement, noise tolerant and robust. Control forces could be applied in two-orthogonal directions of a cutting tool to insure a robust and effective control system. Because of their many advantages, artificial neural networks (ANNs) will be used to fulfill these requirements. The methodology to design and implement such an ANN-based ACS system will be presented in this section. First, the general ACS strategy used in the current work will be presented. Then the design of the chatter suppression sub-system used in the ACS system will be introduced. Finally, the general methods to design the ANN models will be discussed.

6.2.1 General ACS Strategy

The proposed ACS control strategy utilized in the current work is shown in Figure 6-3. In this strategy, a chatter suppression module relies on the availability of a detected chatter parameters from the chatter detection module to generate control signals, i.e., *x(u)*, which are applied to the plant by secondary sources, i.e., actuators, to suppress the chatter. In Figure 6-3, *x(p)* represents the primary disturbance, which includes the sharpness and parameters of the insert, the workpiece shape and material, depth of cut, cutting speed, as well as other factors. The plant output, i.e., *Y* in Figure 6-3, is the vibration response of the cutting tool measured at the sensor location.

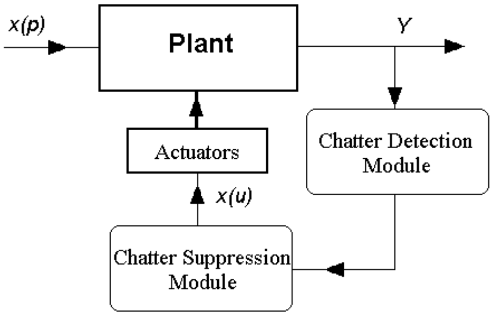


Figure 6-3: The proposed ACS strategy

The chatter detection module was introduced in the previous section. There are two sets of inputs for this module, i.e., the detected signals (i.e., *x*-signal and *y*-signal in Figure 6-1) from the sensors and the measured first natural frequencies of the tool system in the horizontal and vertical directions (i.e., f_{Nx} and f_{Ny} in Figure 6-1). The plant output, i.e., *Y* in Figure 6-3, includes both *x*-signal and *y*-signal, which are coming from the sensors of the proposed experimental setup. Another set of inputs, i.e., f_{Nx} and f_{Ny}, are obtained before cutting. The strategy for the chatter detection module to obtain f_{Nx} and f_{Ny}

is shown in Figure 6-4.

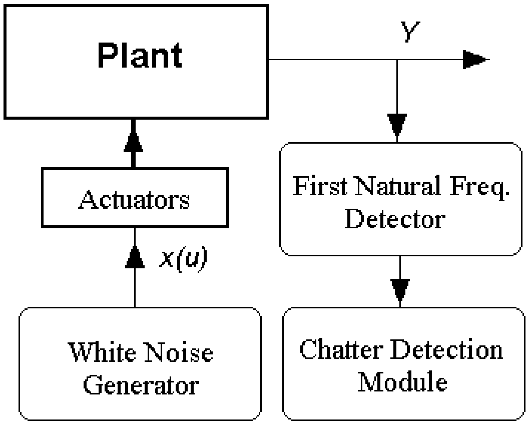


Figure 6-4: The proposed first natural frequency detection strategy

The natural frequencies of the tool system are measured by using the system natural frequency measurement techniques used in the AVC system (see section 4.5). The white noise generator in Figure 6-4 generates uniform white noise signals that contains a uniformly distributed, pseudorandom pattern whose values are in the range [-*a*: *a*], where *a* is the absolute value of amplitude. Ideal white noise has equal power per unit bandwidth, resulting in a flat power spectral density across the frequency range of interest. Therefore, when a uniform white noise signal is sent to an actuator, in the corresponding amplitude FFT response of the tool system at the sensor location, the frequency corresponding to the maximum amplitude should be the first natural frequency because of the resonance effect. In this way, f_{Nx} and f_{Ny} can be obtained. Then, the chatter detection module will save them and their values will not change unless the tool system is modified.

Based on the two sets of inputs, the chatter detection module can monitor the vibration status of the cutting tool at the sensor location and send the chatter parameters to the chatter suppression module when the onset of chatter is detected, as shown in Figure 6-3.

In this ACS strategy, the following relation exists:

$$Y = f(x(p), x(u), t) \tag{6-1}$$

The chatter suppression module is used for minimizing *Y*. In order to minimize *Y*, the control signal to the actuator, i.e., *x(u)*, can be determined through an optimization method such as steepest decent, where

$$x(u)^{k+1} = x(u)^k - \alpha^k \frac{\partial Y^k}{\partial x(u)^k} \tag{6-2}$$

where $\frac{\partial Y^k}{\partial x(u)^k}$ is the gradient of the dynamic model *Y*; $x(u)^k$ and $x(u)^{k+1}$ are the values of the control signal in the *k* and *k+1* iteration; and α is the size of the steps in the direction of minimization. The calculation of $\frac{\partial Y^k}{\partial x(u)^k}$ depends on availability of a differentiable model.

However, a comprehensive physical model of a complex system usually does not exist. In this case, using artificial neural networks in the chatter suppression module is a good choice because ANNs are known for their function approximation capability. As introduced before, properly designed and trained neural networks are capable of approximating any linear or nonlinear function to the desired degree of accuracy and they are noise tolerant.

6.2.2 Chatter Suppression Subsystem Design

The design of the chatter suppression subsystem is based on the techniques utilized in the AVC subsystem design (see section 3.3).

Figure 6-5 shows some details of the proposed chatter suppression sub-system. The inputs of this sub-system, i.e., f_x, f_y, A_x, A_y, P_x and P_y, are the outputs of the proposed chatter detection sub-system, as shown in Figure 6-1. The outputs of the proposed chatter suppression sub-system, i.e., f_{xu}, f_{yu}, A_{xu}, A_{yu}, P_{xu} and P_{yu}, which are also the parameters of control signals, i.e., *x(u)* in Figure 6-3, are sent to corresponding actuators to suppress the detected chatter. PD is the phase difference as described in the next section. The purpose of the chatter suppression subsystem is to generate an "opposite" vibration of the cutting tool at the location of interest to suppress the original one. Therefore, the vibration generated by each control signal at the sensor location should be out of phase with the chatter signal in each direction, while the amplitude and the frequency of the vibration generated by the control signal at the sensor location should be the same as the chatter's in this direction. In this way, the chatter in this direction will be suppressed. The same control strategy will be applied in both directions.

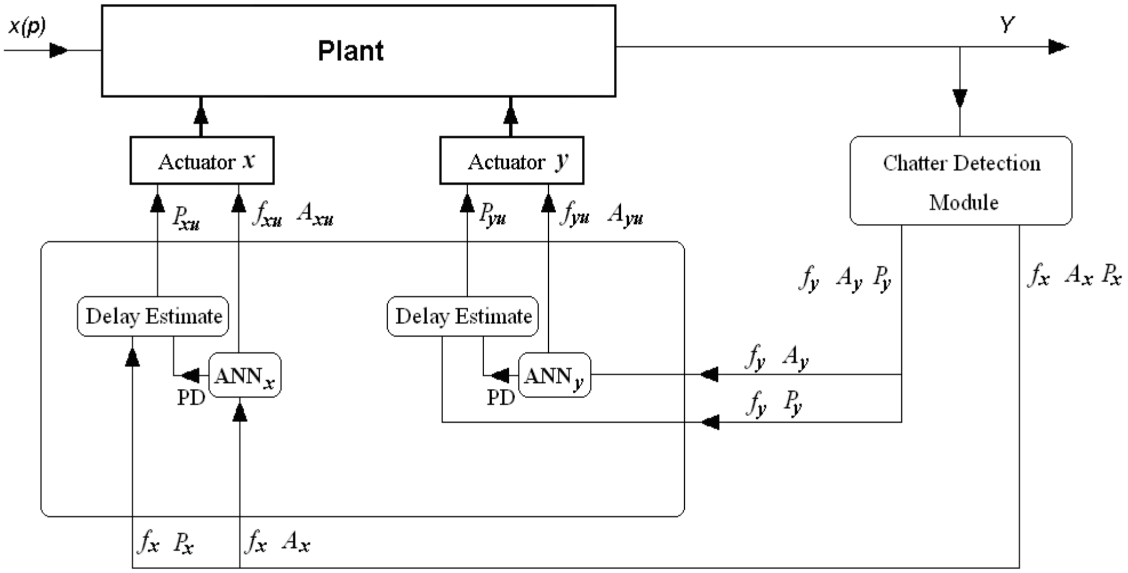


Figure 6-5: Proposed chatter suppression sub-system

To generate an "opposite" vibration at the location of interest to suppress the original one, an ANN is utilized at each direction as an identification model of the plant based on the function approximation capability of ANNs. Some current active vibration control systems use ANNs as the whole system identification models and use offline training method to train the ANNs. However, as mentioned previously, the primary disturbance, i.e., *x(p)* in Figure 6-3, includes the sharpness and parameters of the insert, the workpiece shape and material, depth of cut, cutting speed, as well as many other factors. Among these factors, some keep changing during cutting. Therefore, using such an ANN that is offline-trained to be an identification model of such a changeable system cannot be robust. To make the proposed ACS system robust, the ANN model should be used as the identification model of a relatively stable part of the system during cutting. In the proposed experimental setup, which will be presented in the next chapter, the tool system (which includes the tool holder, the boring bar, the actuators and the sensor) is very stable. Therefore, this tool system is chosen as the model for the ANNs in the current work.

Time delay in ACS is very critical. As discussed in section 3.3, to satisfy causality of different iterations, the time delay between the iteration to collect vibration signal parameters and the iteration to send out control signal should be considered.

To generate control signals, the ANN model should work as an inverse model, which means the inputs of the ANN model are actually the outputs of the plant, as discussed in section 3.3.

6.2.3 Design of the Inverse Artificial Neural Network Models

The design of the inverse ANN models is very similar to design for the AVC system, as presented in section 3.4.

The ANNs are used for function approximation and work as the inverse identification models of the cutting tool system. The design of the ANN models is based on the applied ACS strategy and the actual experimental setup. Generally, the design steps are as follows: First, training data for the ANN models are collected via experiments according to the ACS strategy presented in the previous section. Then, the training data are analyzed in order to choose a proper normalization method. The general network architectures of the ANN models are then designed and the suitable learning algorithm is chosen. Finally, the ANN models are trained to avoid overfitting. The network architectures may be modified for better function approximation based on experimental results.

As the first step to design ANN models after the control strategy is set up, training data should be collected based on the control strategy and experimental setup. In experiments to collect training data for the ANN models, only the control actuators work to generate the tool vibration. To get a robust training, which means a training affected minimally by external sources of variability, the experiments to collect training data need to be designed first. In this project, the fractional factorial design [59] is used for the design of experiments to obtain the training data for the ANN models as shown in [29]. Moreover, the experiments are carried out at frequencies around the first natural frequencies of the tool system because, as discussed before, the chatter frequencies are always around the first natural frequencies of the tool system.

An ANN model for chatter suppression in horizontal direction is shown in Figure 6-6. This model shows an example based on the results in [29, 30]. The ANN architecture used here has three inputs, one hidden layer of log-sigmoid neurons and one output layer of three log-sigmoid neurons (because there are three outputs). The output layer uses a log-sigmoid transfer function because the outputs of the ANNs are constrained to a range of 0 to 1. In the detailed design, the number of hidden layers and the number of neurons in each hidden layers are decided by finding out the optimal number to obtain the smallest mean square error (MSE) for validation data sets.

In experiments undertaken to collect training data for the ANN models, only the control actuators are active to generate the plant vibrations, i.e., the primary disturbance $x(p)$ =0. Therefore, the ANN model can provide the information about the kind of control signal (f_{xu}, A_{xu}, and P_{xu}) needed to get the specific vibration signal (f_x, A_x, and P_x) from the sensor. The ideal control signal frequency (f_{xu}) should be the same as vibration frequency (f_x). Moreover, the input P_x can be cancelled if the phase difference (PD) between the control signal and the vibration signal is utilized (PD = $P_{xu} - P_x$). In this case, the ANN can be simplified as shown in Figure 6-7. There are only two inputs (i.e., A_x and f_x) and two outputs (i.e., A_{xu} and PD) for the ANN. In the detailed design, the number of hidden layers and the number of neurons in each hidden layers are decided by finding out what the best number of hidden nodes are to obtain the smallest Mean Square Error (MSE) for validation data sets.

The ANN model for chatter suppression in vertical direction can be designed in the same way.

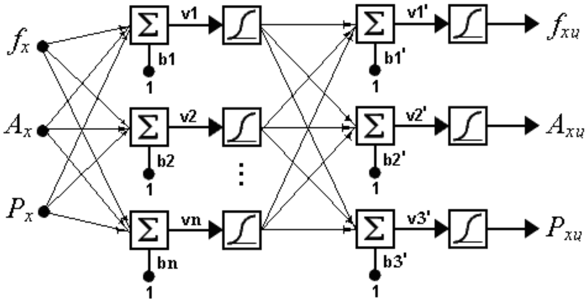


Figure 6-6: The proposed ANN model for chatter suppression in horizontal direction

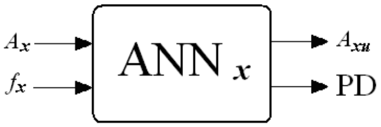


Figure 6-7: Simplified ANN model

Based on the similarity of the ACS subsystem design and the AVC subsystem design (section 3.3 - 3.4), by using equation (3-5) and (3-6), in Figure 6-5, P_{xu} and P_{yu} can be obtained as:

$$P_{xu} = P_x + 180 + (f_x \times t - \mathrm{int}(f_x \times t)) \times 360 - PD \tag{6-4}$$

$$P_{yu} = P_y + 180 + (f_y \times t - \mathrm{int}(f_y \times t)) \times 360 - PD \tag{6-5}$$

where t is the running time of each control iteration. f_{xu} is set to the same as f_x and f_{yu} is set to the same as f_y. A_{xu} and A_{yu} can be obtained from the ANN model directly.

CHAPTER 7

EXPERIMENTAL SETUP OF ACS

The methodology presented in Chapter 6 will be evaluated experimentally through active chatter suppression (ACS) in turning on a lathe. The experimental setup is described in this chapter.

7.1 Hardware Setup

Figure 7-1 shows the proposed hardware setup for the experiments. The proposed lathe is a two-axis engine lathe. The square tool bar (8 mm x 8 mm) is made of plain carbon steel and has a total length of 208 mm with one turning insert on one end and one boring insert on another end. In Figure 7-1, the turning insert is used and the overhang length of the tool bar is 108 mm, as measured from clamped end to the cutting edge of the tool bar. As discussed in the previous chapter, the proposed control system should be comprised of two actuator/sensor pairs: one pair to control motions parallel to the axis of the workpiece (x-direction) and the other pair to control motions in the direction tangent to the machined surface the revolving workpiece (y-direction). The horizontal (x-direction) & vertical (y-direction) accelerometers (sensors) can be glued to the cutting tool at the location of the insert because this is the location of interest. By using two sensors, two crossed accelerations, i.e., horizontal and vertical accelerations, from the bending vibration of the cutting tool at the location of the insert will be measured. The actuators should be attached to the boring bar at the location close to the clamped end of the tool bar to get maximum working range of the boring bar. The actuators simply push up against the tool bar. The resulting preload is used to maintain contact between the control shaker and the beam.

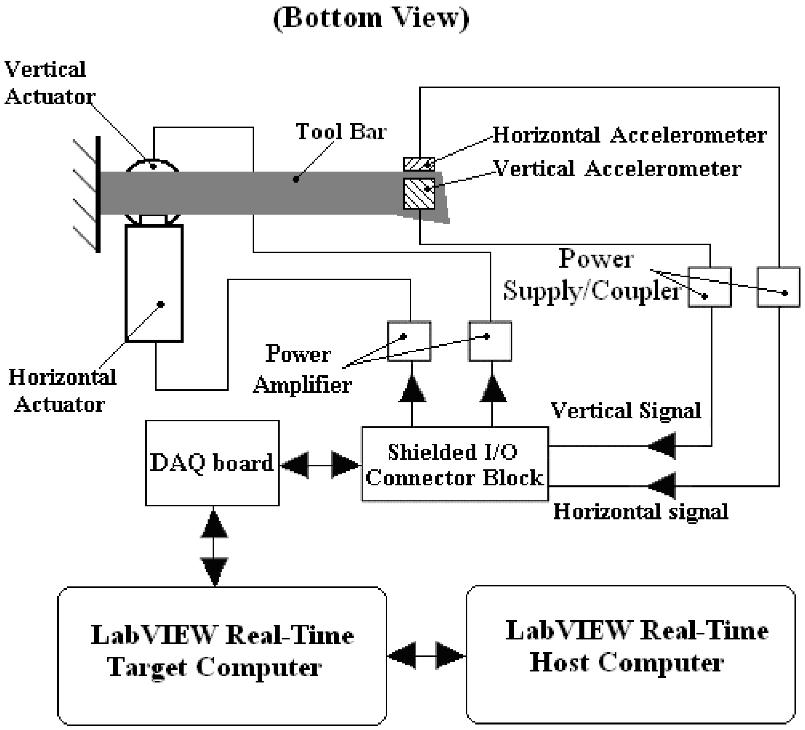


Figure 7-1: The hardware setup for implementing active chatter suppression

Since the proposed location for the actuators is close to the clamped end of the tool bar, the force required to actuate the bar near its base is substantial while the travel range required is small. Therefore, induced strain actuators are good choices for the current application. On today's market, among the available induced strain actuators, both those fashioned from magnetostrictive materials, such as Terfenol-D, and piezo-stack actuators can work in this setup. Terfenol-D actuators have the advantages that they can be driven using standard audio-type power amplifiers [25], have a higher energy density and more durable. However, Terfenol-D actuators are always much bigger than piezoelectric actuators. Considering the small size of the lathe and the boring bar used in the current work, to get enough working range of the tool, piezoelectric actuators are the better choice if they can provide enough force and travel range. Based on calculation, P-830.30 piezo actuator (see Appendix C for the actuator specifications) is chosen for the ACS system.

There are two choices for the amplifier for the selected piezo actuator (P-830.30 of Physik Instrumente GmbH & Co.), which are E-610 Piezo Amplifier/Controller and E-617 High-Power Piezo Amplifier. Considering the probable working frequencies and the electrical capacitance of the actuator, the more versatile one, i.e., E-617 High-Power Piezo Amplifier, was chosen.

Also considering the small size of the lathe and the boring bar used in the current work, a small, adhesive mounted piezoelectric accelerometer was chosen for measuring the tool vibration. After investigation, choices were narrowed down to several accelerometers with acceleration range of ± 5G (1 G = 9.80665 m/s^2), ± 50G and $\pm$ 500G. Accelerometers with acceleration range of ± 5G and ± 50G were originally favored because based on the experimental setup it was assumed that the maximum acceleration should be several G only. In this case accelerometers with acceleration range of ± 5G and ± 50G would provide higher measurement accuracy. However, accelerometers with acceleration range of ± 5G and ± 50G are too big for the current experimental setup. Calculation also showed that the maximum acceleration could be larger than the assumption. Therefore, an accelerometer with acceleration range of $\pm$ 500G (8778A500M14 of Kistler Co., see Appendix D for the accelerometer specifications) was chosen for measuring the vibration. This accelerometer is only about ϕ5.8mm x 4.3mm in size and weighs only 0.4 gm, which is suitable for the size of the boring bar and can reduce unwanted mass-loading effects.

The capabilities of the accelerometer, the actuator and amplifiers in the current experimental setup still needed to be verified. Therefore, only one pair of accelerometer/actuator was purchased for the current work.

The data acquisition (DAQ) board (see Appendix A for the DAQ board specifications) is the same board used in the AVC system, which has four 12-bit analog inputs with a maximum sampling rate of 5 MS/s. It also provides two 16-bit analog outputs with a maximum update rate of 2.5 MS/s. Figure 7-2 is a sketch of the experimental hardware setup, which also shows some functions of the DAQ board. Two analog outputs, i.e., “AO-1” and “AO-2”, and two analog inputs, i.e., “AI-1” and “AI-2”, can be used in the experiments.

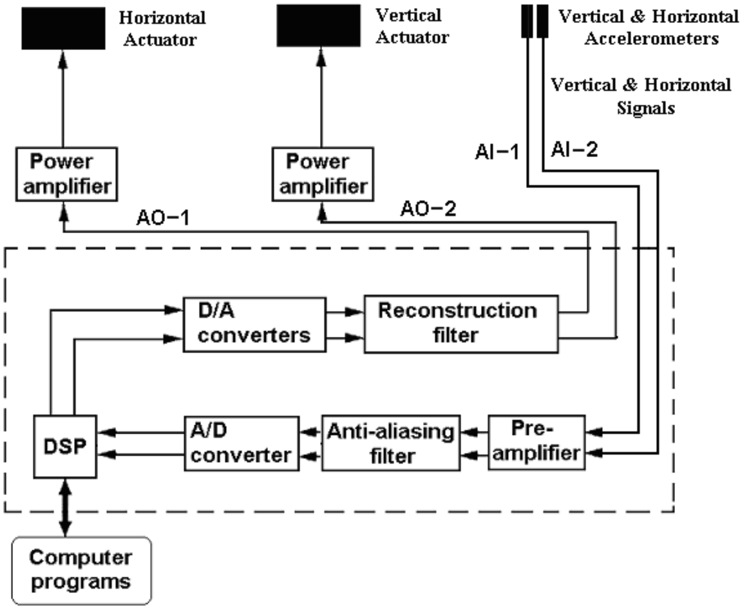
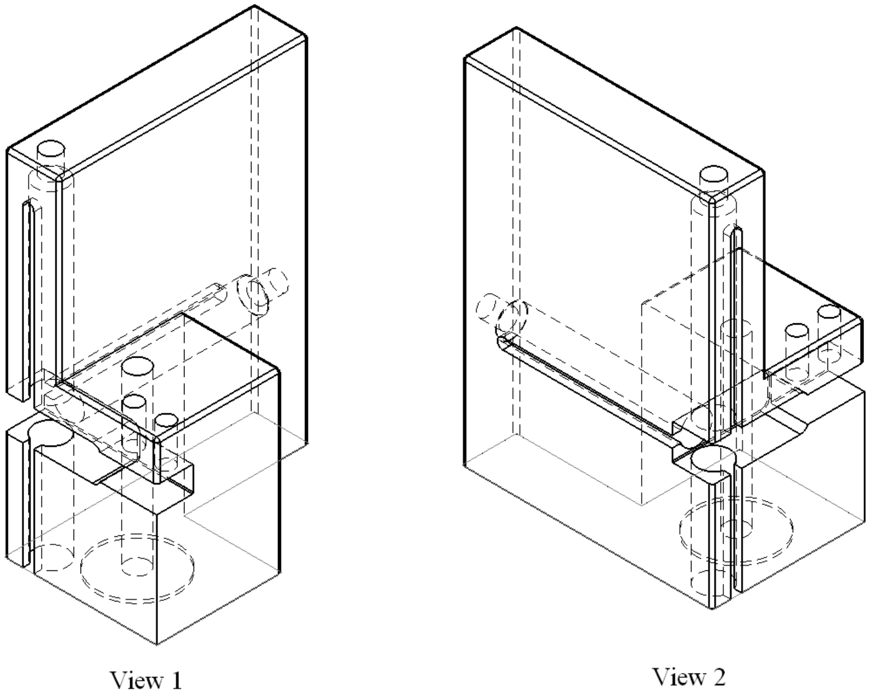


Figure 7-2: Schematic diagram of experimental setup

Two computers were used in the system because LabVIEW Real-Time was used as the main developing software in this work. One computer worked as a dedicated real-time target, which integrated the DAQ board for analog, digital, counter/timer, and vibration signals. Programs were developed on a Windows XP host computer and downloaded to the real-time target via Ethernet. Moreover, some time-consuming applications, such as graphing and data recording, were implemented on the host computer instead of the target computer. The target computer ran a single-kernel RTOS (real-time operating system) for maximum reliability.

To mount the P-830.30 piezo actuators properly, a toolholder was designed and built. Figure 7-3 shows two views of the designed toolholder. The actuator can be mounted in both vertical and horizontal directions. Two setscrews were used to hold the tool bar and a set of bolt/nut/lock washer was used to add preload on the actuator in each direction.



View 1

View 2

Figure 7-3: Two views of the designed toolholder

In the toolholder design, the distance between the setscrew and the actuator is 25 mm. The overhang length of the tool bar is 108 mm, as measured from the setscrew that holds the bar in the tool holder to the cutting edge of the turning insert (see Figure 7-1). To check the ability of the actuator for the designed toolholder some calculations have been done to estimate the properties. The tool bar is considered as the cantilever beam in Figure 7-4, where

L=108 mm (cutting point);

l = 25 mm (actuator point);

E (Young’s modulus) = 2x10^{11} N/m^2 (30000 kpsi);

b = d = 8 mm (square tool bar: 8 mm x 8 mm);

I (Moment of Inertia of the cross section) = $\frac{bd^3}{12} \cong$ 3.4133 x10^{-10} m^4 [5];

P = 1000 N (see Appendix C for the actuator specifications).

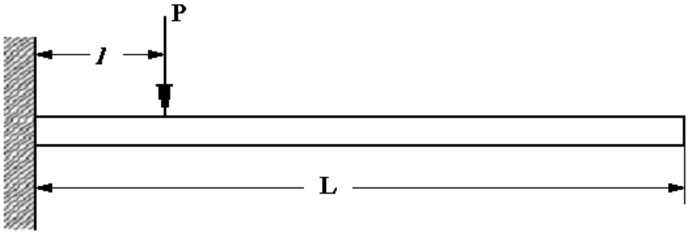


Figure 7-4: Schematic diagram of the tool bar

To find out the maximum displacement, δ_b, at the cutting point that can be obtained by the force applied by the actuator, the following equation can be used [5]:

$$P = \frac{\delta_b 6EI}{l^2(3L-l)} \tag{7-1}$$

By solving the equation, the maximum displacement can be obtained, which is about 0.456 mm. This is larger than the displacement at the cutting point during turning in most cases [3].

Figure 7-5 shows two photographs of the experimental setup as described.

In the current work MAXNC T2 lathe was used. This is a two-axis mini-CNC engine lathe with a 1/2 HP Spindle Motor and Spindle speeds from 0 to 1500 RPM. Only one piezo accelerometer and one piezo actuator was utilized in the current setup. The workpieces were aluminum with the size of about ϕ25 mm x 118 mm.

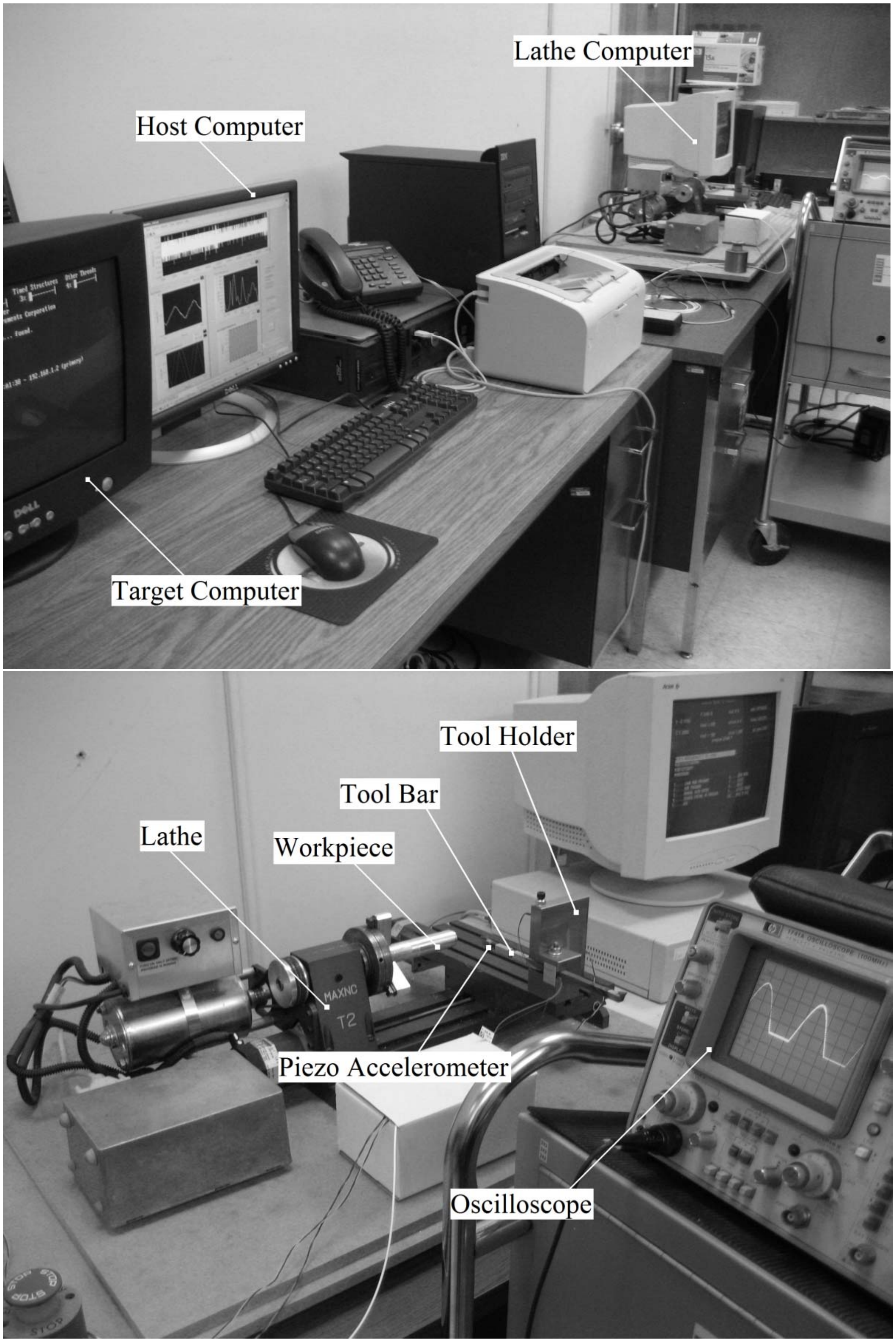
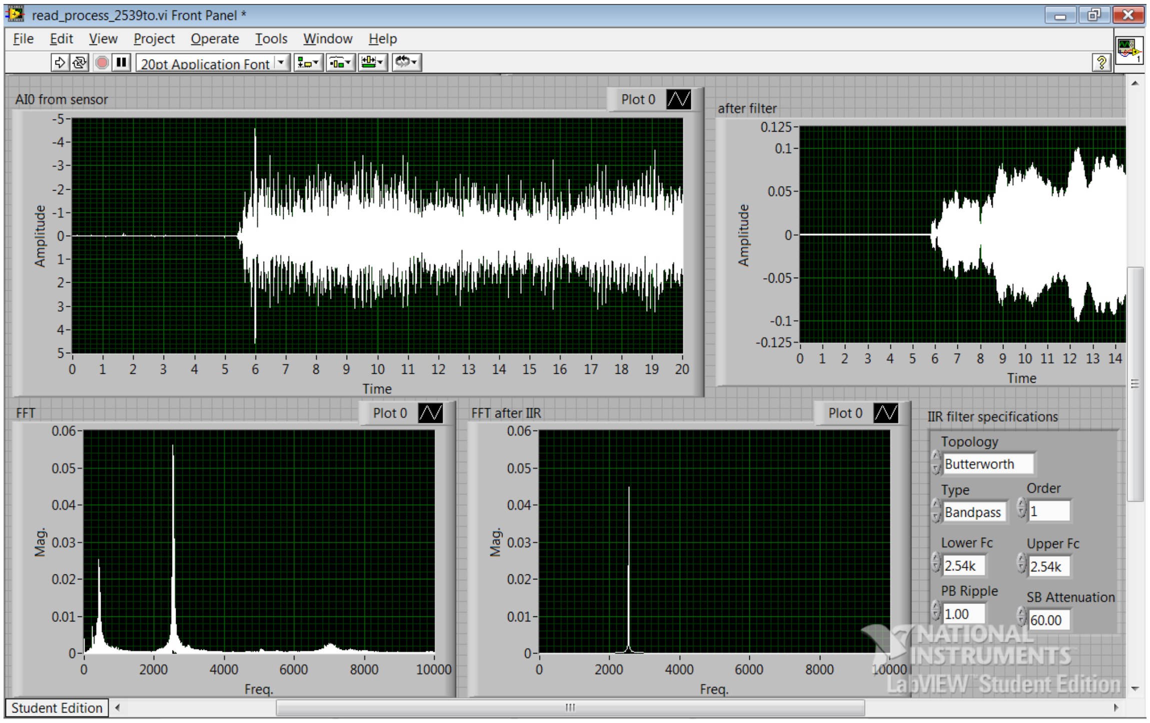


Figure 7-5: Two photographs of the experimental hardware setup

7.2 Software Environment

The operation system on the host computer was Microsoft Windows XP and LabVIEW Real-Time version 8.5.1 on the target computer. LabVIEW Real-Time version 8.5.1 was used for measurement, signal processing and the development of user interfaces. MATLAB 2008 was used for some neural network design, implementation, visualization and simulation, as well as data analysis and some graphing. Microsoft Visual C++ 6.0 was used for algorithm implementation and for implementing neural networks in the form of dynamic link library (DLL) files.

Figure 7-6 shows one example of user interfaces developed in the current work. The interfaces can only be shown on the host computer.



IIR filter specifications

Topology: Butterworth

Type: Bandpass

Order: 1

Lower Fc: 2.54k

Upper Fc: 2.54k

PB Ripple: 1.00

SB Attenuation: 60.00

Figure 7-6: An example of a part of a user interface used in the current work

CHAPTER 8

EXPERIMENTAL IMPLEMENTATION OF ACS

8.1 AVC System Design

Based on the methodology and the experimental setup, the designed AVC system to generate a control signal is shown in Figure 8-1. This control system can modify the control signal online in every control cycle (one control cycle could be one or more than one control iterations) in the following way:

1. The system can repeat all calculations in one control iteration and generate a current control signal (e.g., with parameters of $f_{xu,}$ $A_{xu,}$ and P_{xu}) based on the detected vibration situation (e.g., $f_{x,}$ $A_{x,}$ and P_x) of this current iteration;

2. The current control signal is added to the accumulated control signal, which is a combination of all previous continuous control signals, to get an updated control signal in the Signal Combination module (the control signal of the first control cycle is zero);

3. The new updated control signal is sent out to the actuator at the beginning of the next iteration; and

4. At the same time, this updated control signal becomes the “accumulated control signal” in the next control cycle.

Therefore, the actual control signal sent to the actuator is an accumulation of all previous generated control signals, which are all continuous. One control cycle should include more than three control iterations to avoid unstable transient conditions after the modification of the actual control signal, and therefore to get more accurate measurements of vibration. In current work, a control cycle includes five or more iterations. One control cycle could be 25 ms, which means a new control signal could be generated as fast as in every 25 ms in the current setup.

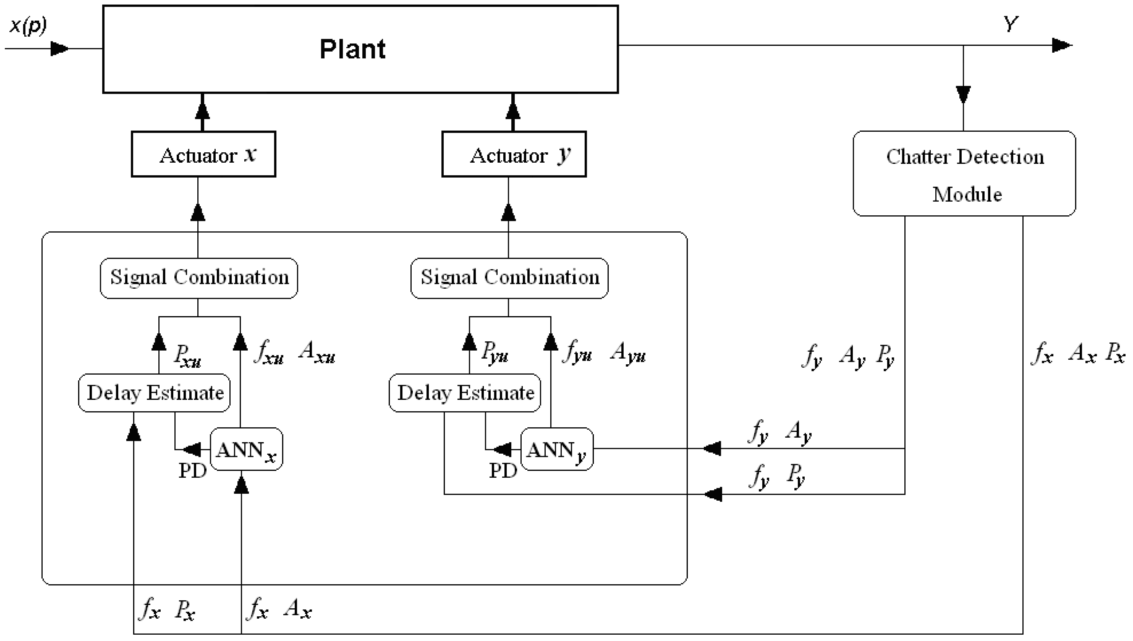


Figure 8-1: The AVC system in current work

The parameters of the control signal, e.g., $f_{xu},$ $A_{xu,}$ and P_{xu}, can be obtained from ANN outputs and the “delay estimate” module by using the equations in section 6.2.

Before the ACS system can work during cutting, the first natural frequencies of the system should measured by using the system natural frequency measurement techniques used in the AVC system, as introduced in section 4.5, and the ANN models need to be trained. The system natural frequency measurement and ANN training can be done automatically by running a preliminary program before cutting. One piezo accelerometer and one piezo actuator is utilized in the current work. The first natural frequency of the tool system in the vertical direction is about 488 Hz and in the horizontal direction is about 423 Hz.

8.2 Artificial Neural Network Training

The detailed steps to train the ANN models of the ACS system are presented in section 6.2.3. The ANNs are used for function approximation and work as the inverse identification models of the tool system. The design of the ANN models is based on the applied ACS strategy and the actual experimental setup. Generally, the design steps are as follows: First, training data for the ANN models are collected via experiments according to the ACS strategy presented in the previous section; Then, the training data are analyzed in order to choose a proper normalization method; The general network architectures of the ANN models are then designed and the suitable learning algorithm is chosen; Finally, the ANN models are trained to avoid overfitting. The network architectures may be modified for better function approximation based on experimental results.

The Simplified ANN models are shown in Figure 8-2. As the first step to design ANN models after the control strategy is set up, training data should be collected based on the control strategy and experimental setup. In experiments to collect training data for the ANN models, only the control actuator work to generate the tool vibration. To get a robust training, which means a training affected minimally by external sources of variability, the experiments to collect training data need to be designed first. In this project, the fractional factorial design [59] is used for the design of experiments to obtain the training data for the ANN models as shown in [29]. Moreover, the experiments will be done at frequencies around the first natural frequencies of the tool system because, as discussed before, the chatter frequencies are always around the first natural frequencies of the tool system.

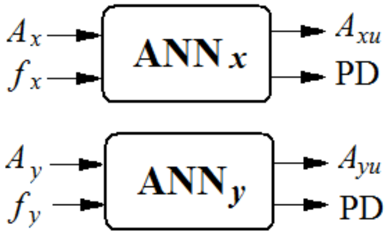


Figure 8-2: Simplified ANN models in ACS

As discussed in Chapter 6, a multilayer feedforward ANN is utilized, as shown in Figure 6-8, which shows the ANN model for horizontal direction. The ANN architecture used here has three inputs, one hidden layer of log-sigmoid neurons and one output layer of three log-sigmoid neurons (because there are three outputs). The output layer uses a log-sigmoid transfer function because the outputs of the ANNs are supposed to be constrained to a range of 0 to 1. The number of hidden layers and the number of neurons in each hidden layers are decided by finding out what the best numbers are to obtain the smallest Mean Square Error (MSE) for validation data sets. The ANN model for chatter suppression in vertical direction will be designed in the same way.

The objective of the experiments in the current work is to determine the influence of the controllable factors, e.g., the horizontal chatter frequency (f_x) and the horizontal chatter amplitude (A_x), to the response, e.g., the horizontal control signal amplitude (A_{xu}) and the phase difference (PD). The data ranges are decided by the regions of interest for each variable and hardware performance limitations. Since the first natural frequency of the tool system in the vertical direction is about 488 Hz and in the horizontal direction is about 423 Hz, based on the results of many other papers, such as [35, 38], the chatter frequency should be around 423 Hz to 488 Hz. Therefore, the region of interest for frequency can be set to 400 Hz to 500 Hz, and the frequency response range of the amplifier of the piezo actuator can cover this range with maximum work load (as shown in Figure 8-3, which shows the operating limits of E-617 High-Power Piezo Amplifier [60], while the Electrical Capacitance of the piezo actuator, i.e., P-830.30, is 4.5 µF). When two pairs of accelerometer/actuator are utilized, the frequency range 400 Hz to 500 Hz can be used to train the ANN models and also used in the Chatter Detection Module, i.e., setting f_{min} = 400 Hz and f_{max} = 500 Hz, as discussed in section 6.1.1. The analog output amplitude range is 0 V to 10 V for safety reasons. To find out the proper amplitude ranges to train the ANN models, cutting experiments should be undertaken. Considering the measurement range of the sensor and the hardware setup, according to results of experiments, the amplitude range for the control signal sent to the actuator was set to be from 0.01 V to 5 V.

As mentioned before, the values of all the data must be normalized for efficient processing by the ANN (see section 2.4.5). In the current work, all the experimental input and output data are positive values. The data are also normalized to a range of 0.1 to 0.9 by using equation (5-2).

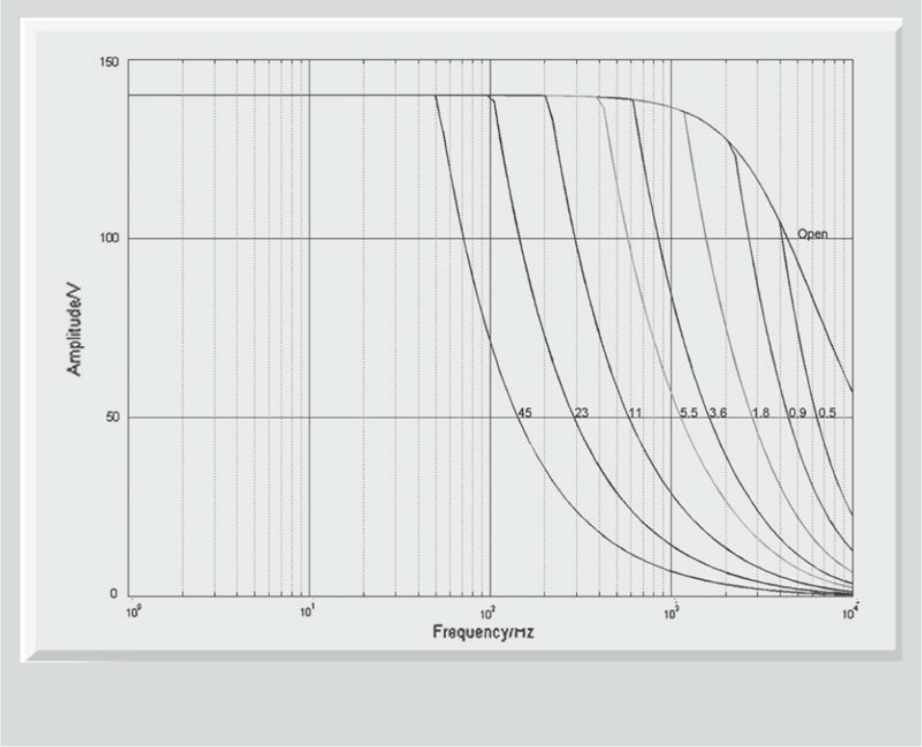


Figure 8-3: Operating limits with various PZT loads (capacitance is measured in µF) [60]

As discussed in [29], to reduce harmful effects, e.g., the squashing effect, of using sigmoid transfer functions in the hidden layer and the output layer of the ANN model, and normalization, the above data ranges can be divided into several sub-ranges, e.g., the original frequency range can be divided into four smaller sub-ranges: 400 Hz to 420 Hz, 420 Hz to 440 Hz, 440 Hz to 480 Hz, and 480 Hz to 500 Hz. About 100 sets of data are collected for each sub-range. ANN models are trained separately for different sub-ranges. Moreover, Resilient Backpropagation (RPROP) algorithm is utilized to train ANN models because, although it is not the fastest one, theoretically, it can also help to reduce squashing effect of the magnitudes of partial derivatives (see Chapter 5 for details).

To find the best ANN architecture (the number of hidden layers and the number of neurons in each hidden layer), many experiments have been done. The results and experiences in [29, 30] were used as guidelines. The best ANN architecture can provide the smallest Mean Square Error (MSE) for validation data sets (see section 2.4). Early stopping and Bayesian regularization methods were used in MATLAB to improve generalization in these experiments (see section 2.4.6). Test data sets were also used to compare different models. The best ANN architecture found via experiments for the ACS system was similar to the example shown in Figure 6-8. It is a multilayer feedforward ANN, which has two inputs, one hidden layer of 15 log-sigmoid neurons and one output layer of two log-sigmoid neurons. The MSE of validation data sets for this architecture was about 0. 057%. The output layer uses a log-sigmoid transfer function because the outputs of the ANNs are supposed to be constrained to a range of 0 to 1. This ANN architecture provides the smallest Mean Square Error (MSE) and has very good performance for generalization in experiments. For the same tool setup, the ANN architecture does not change, but the weights between neurons change for different data sub-ranges after training.

8.3 Experimental Results

In the current work, for the reasons mentioned in section 7.1, only one pair of piezo accelerometer/actuator was utilized. The original workpieces were aluminum with the dimensions of ϕ25 mm x 118 mm.

8.3.1 Chatter Frequencies and Amplitudes

As mentioned in previous chapters, it is widely known that machining chatter signals have harmonic shapes and their frequencies are around the respective first natural frequencies of the machining systems [32]. Also, in some chatter suppression papers with experimental data, e.g., [35, 36, 38], the chatter frequencies are in the range of 120 Hz to 550 Hz, which is around the respective first natural frequencies of the machining systems studied. However, in the current work, the experiment results show that the chatter frequencies are very different from the respective first natural frequencies of the machining system.

To measure the first natural frequencies of the tool system in vertical direction and in horizontal direction, the techniques used in section 4.5 are utilized and the frequencies are measured while the lathe is not running. The first natural frequency of the tool system in the vertical direction was about 488 Hz and in the horizontal direction was about 423 Hz. Then, the frequencies of the lathe, while its spindle was turning at different speeds from about 200 RPM to 1500 RPM but without cutting, were measured. The frequency measured from the sensor at the insert location of the tool in the vertical direction was about 470 Hz to 490 Hz and in the horizontal direction about 400 Hz to 420 Hz in most cases and in most iterations (the measured frequency changes in different iteration in the same experiment too). As an example, Figure 8-4 shows the signal from the sensor and the amplitude FFT measured in vertical direction when the spindle of the machine is turning without cutting. In the figures, the unit for time is second, for amplitude is volt, for frequency is Hz and for magnitude is volt.

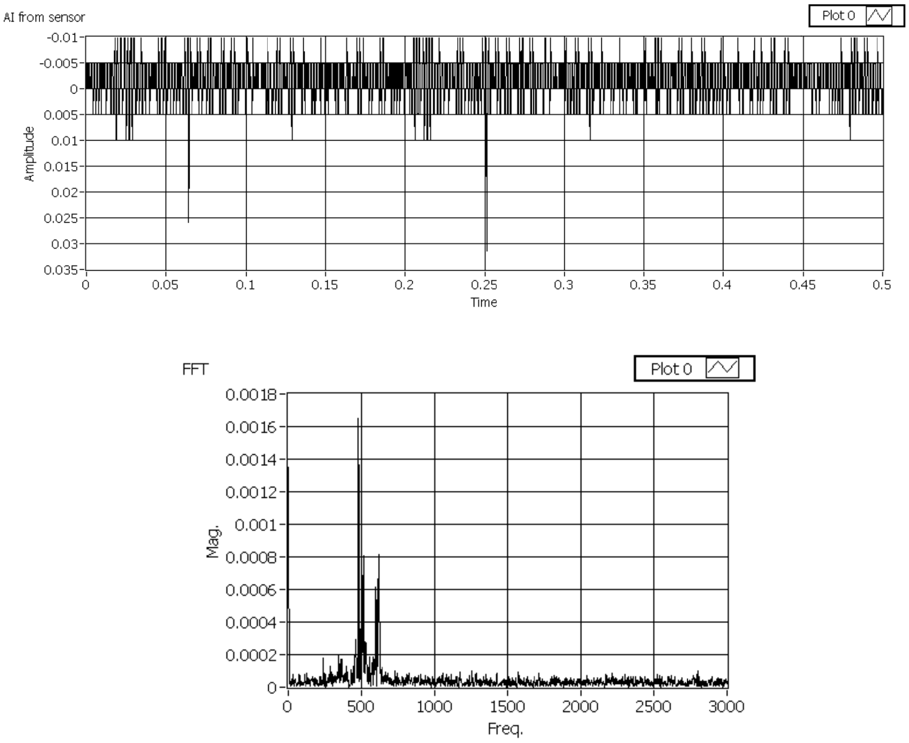
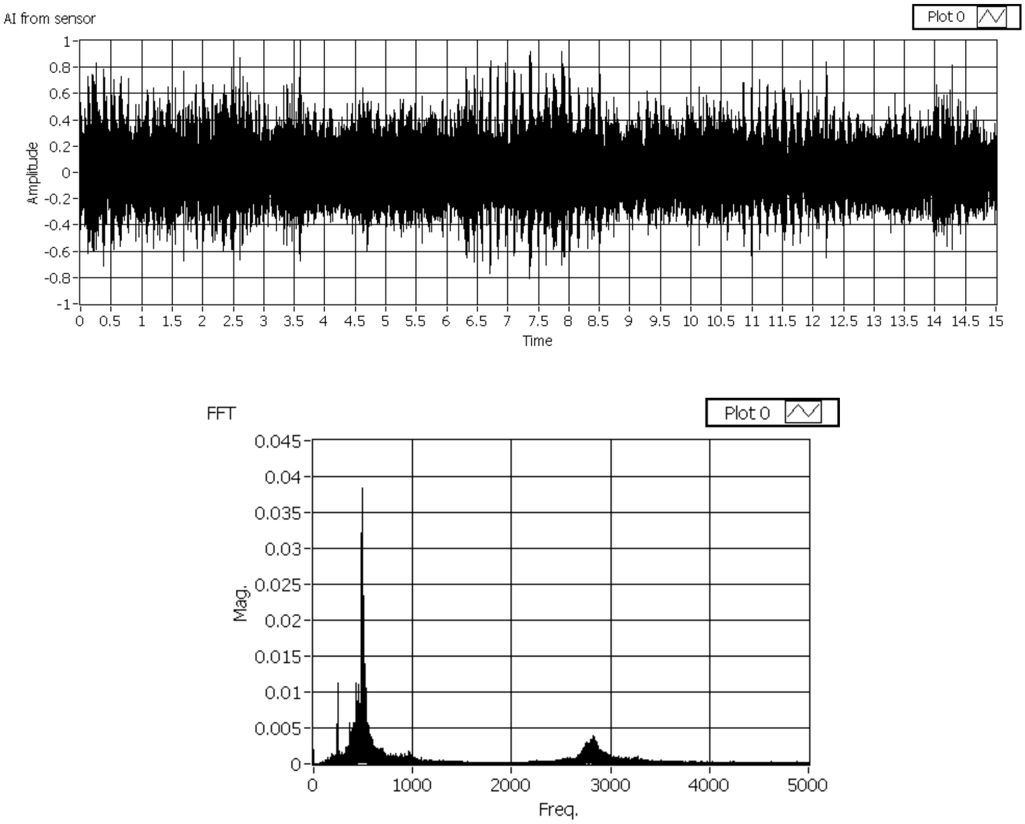


Figure 8-4: Signal from the sensor and the amplitude FFT measured in vertical direction when the spindle of the machine is turning without cutting

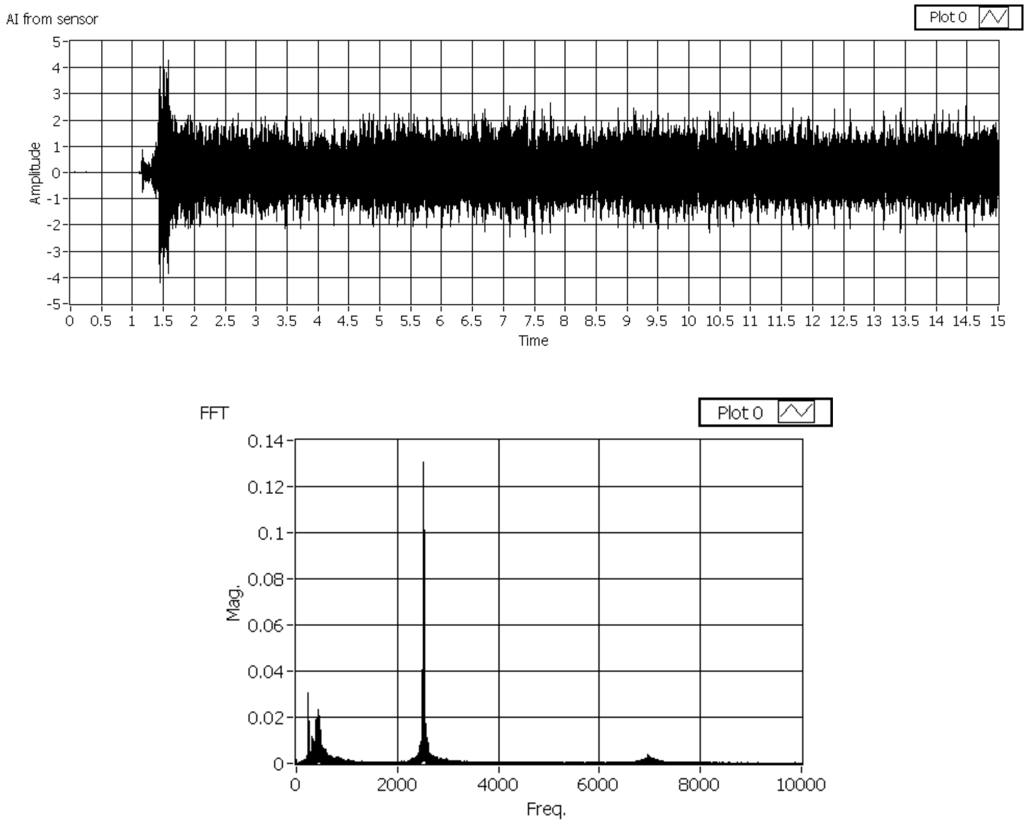
From these measurements and experiments from literature (e.g., [35, 36, 38]), the chatter frequency is expected to be in a range of 400 Hz to 500 Hz. However, when chatter happened, the measured frequency in vertical direction was between 2500 Hz and 2600 Hz in most cases. Figure 8-5 compares the signal from the sensor and the amplitude FFT measured in vertical direction when the machine was cutting with and without chatter. The scales of axes are different to provide as much information for signals as possible. In the experiments shown in Figure 8-5, the spindle speed was about 500 RPM and Feed=1 (setting of MAXNC T2, about 0.015 inch / second without relation to the spindle speed), while the cutting depth was 0.01 inch for Figure 8-5 (a) and 0.02 inch for Figure 8-5 (b).

In horizontal direction, when chatter happens, the measured frequency was more scattered than in vertical direction, but a large part of the measurements is between 2600 Hz and 2800 Hz. As an example, Figure 8-6 shows the signal from the sensor and the amplitude FFT measured in horizontal direction when the machine was cutting with chatter. In the experiment shown in Figure 8-6, the spindle speed was about 500 RPM and Feed=1, while the cutting depth was 0.02 inch. Experiment results show that when the cutting depth is bigger, a larger part of the power is between 2600 Hz and 2800 Hz.



(a) Without chatter

Figure 8-5 (a): Comparison of vertical signals when the machine is cutting without and with chatter



(b) With chatter

Figure 8-5 (b): Comparison of vertical signals when the machine is cutting without and with chatter

From experiments, as shown in the amplitude FFT figures above, it can be seen that most power falls into the range of 2500 Hz to 2800 Hz when chatter happens, especially in vertical direction (2500 Hz to 2550 Hz). And there is much more power in vertical direction than in horizontal direction when chatter happens, which can be seen by comparing Figure 8-5 (b) and Figure 8-6. The scales of Y-axis in these two figures are very different because “auto scale” was used in programming to provide as much information for signals as possible.

In all the experiments in this work, although the proportion of the power in horizontal direction to the power in vertical direction may change with different cutting conditions, for the same cutting conditions, when chatter happened, there was always more power in vertical direction than in horizontal direction.

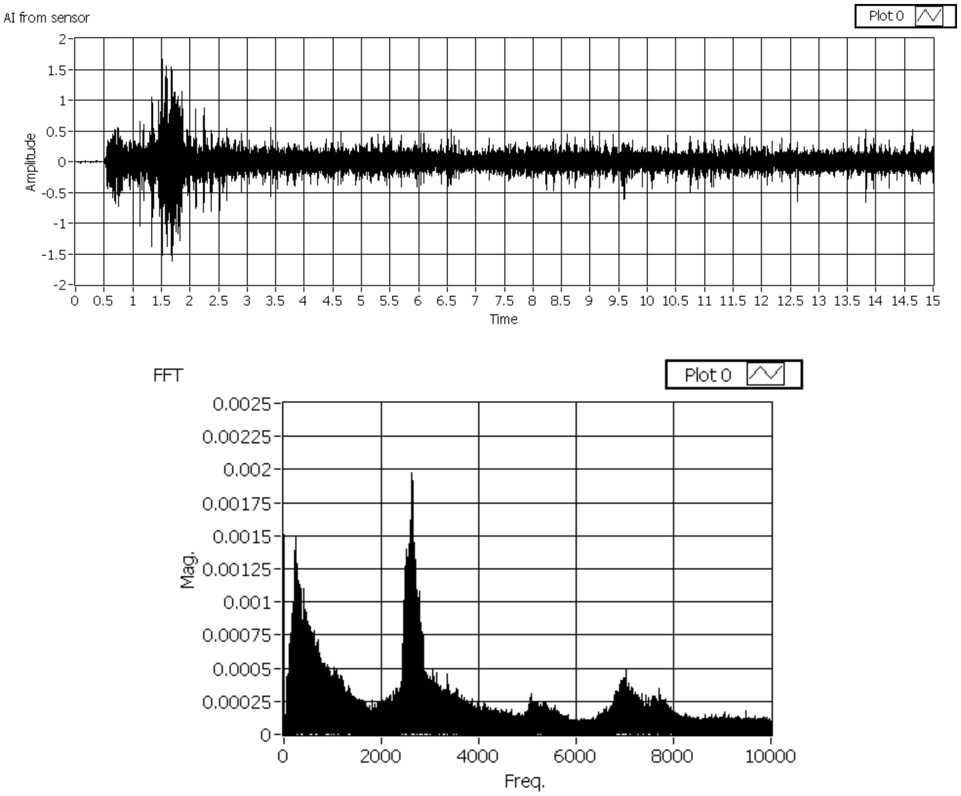


Figure 8-6: Signal from the sensor and the amplitude FFT in horizontal direction with chatter

8.3.2 Chatter Signal Waveforms

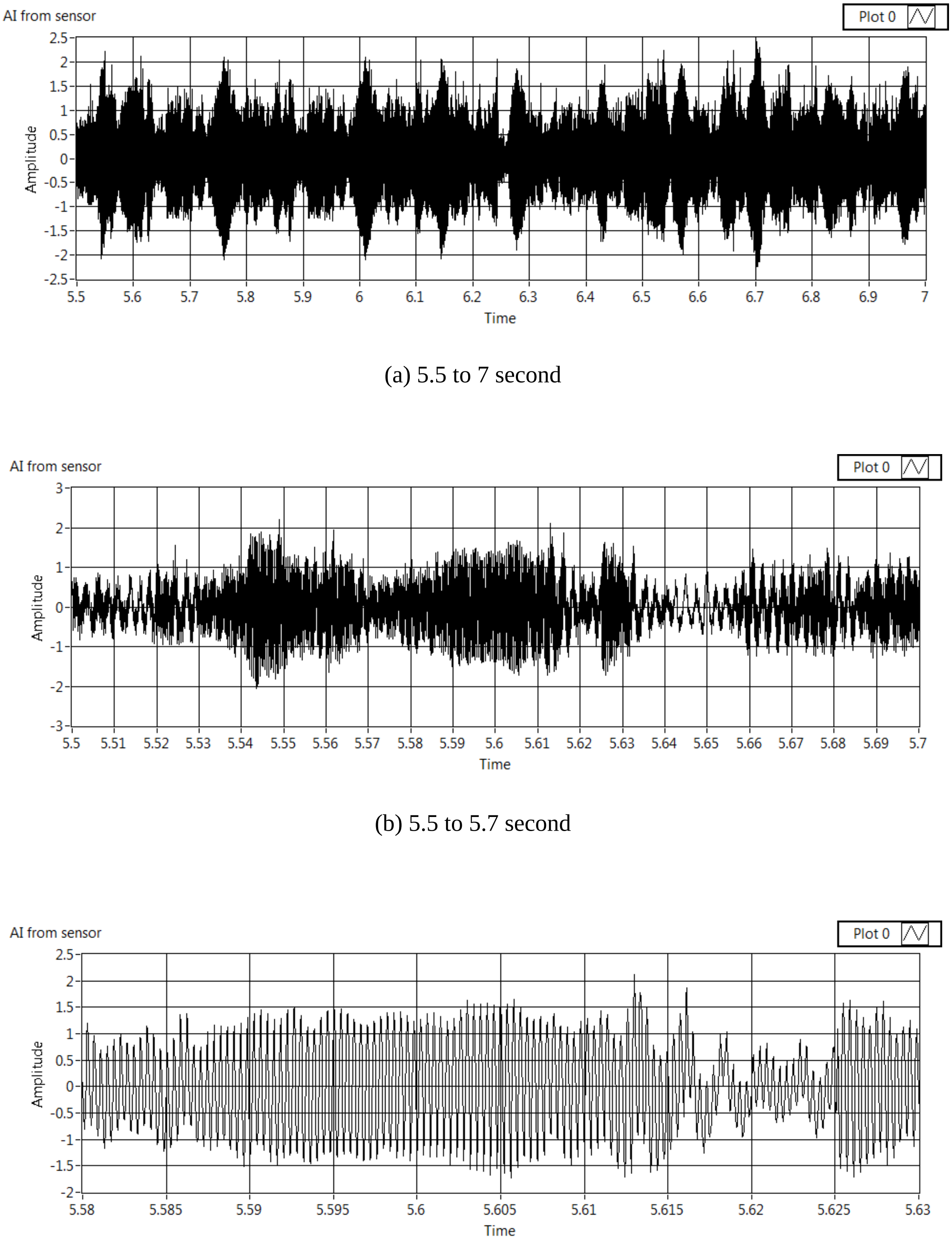
The signals from the sensor (accelerometer) in the experiments when chatter happens have shown more complexity than expected: First, as shown in the previous section, the chatter frequencies in vertical direction and in horizontal direction are very different from the first natural frequencies of the machine in the respective directions, and they are also different from each other. Second, the chatter frequencies are changing in different iterations, which means from time to time, within ranges, as shown in Table 8-1, which is a part of the recorded data of the same experiment to obtain data for Figure 8-5 (b). In that experiment, iteration time was 0.25 second. The measured frequencies are the highest amplitude FFT frequencies. Moreover, the waveforms also show complexity. As an example, Figure 8-7 shows some details of Figure 8-5 (b). As shown in the detailed waveforms, the chatter signal waveforms include many big “waves”.

These big waves are irregular with changing amplitudes, shapes and sizes. One big wave can include many cycles of the measured frequencies shown in Table 8-1. As mentioned before, when chatter happens, the measured frequency was between 2500 Hz and 2600 Hz in most cases, while the measured frequency carries the most of the vibration power. Therefore, these big waves are not chatter signals. The chatter frequency should be the measured frequency, and this was verified by experiments as will be discussed later in this chapter (see the chatter pattern in Figure 8-10).

The irregular big waves are a problem for chatter suppression. The possible reasons for the big waves could be using an unstable mini lathe and the bending of the workpiece without using a tailstock. Experiment results also show that when cuttings were made close to the clamped end of the workpiece, the sizes of big waves were reduced as will be discussed later in this chapter (see Figure 8-8 and Figure 8-9). Based on the current experimental setup, the effect of the big waves was considered in chatter suppression.

Measured Frequency	Measured Amplitude	Measured Phase	Time
480.6740677	0.0023948	84.9828755	0.4389858
480.3943971	0.0027641	43.839986	0.6887331
480.3387916	0.0026016	29.248589	0.9387803
481.0432105	0.0028186	-38.9911723	1.1887274
507.8809118	0.1338728	94.2326464	1.4387722
2514.916098	1.3256609	13.8462919	1.6887398
2523.018426	0.9135309	70.9902876	1.9388185
2523.018426	0.9135309	70.9902876	2.1887345
2525.412902	0.6949867	-113.4761672	2.4387789
2525.412902	0.6949867	-113.4761672	2.6887593
2525.734173	0.6156788	99.7905283	2.938765
2530.430332	0.6663986	132.4818191	3.1887803
2534.152818	0.4751391	-70.2069631	3.4387655
2521.163563	0.5687808	163.9875793	3.6887302
2521.163563	0.5687808	163.9875793	3.9387593
2523.215564	0.4669578	-2.6812284	4.1887589
2517.355924	0.574584	-8.6912755	4.4387722
2502.732945	0.3639981	27.3728908	4.6887398
2519.134118	0.5924571	-129.3306842	4.9387846
2522.383833	0.684596	143.4411053	5.1887531
2518.965176	0.5735396	-234.470592	5.4387703
2518.965176	0.5735396	-234.470592	5.6887903
2522.097605	0.6198768	151.0523919	5.938786
2522.097605	0.6198768	151.0523919	6.1887598
2517.740162	0.5237959	-105.9861235	6.4388204
2521.381382	0.7792558	24.5645912	6.6887798
2522.588411	0.6057447	-134.1329803	6.9388089
2523.109188	0.7111996	-107.569697	7.1887765
2523.109188	0.7111996	-107.569697	7.4388022
2514.060218	0.5959324	61.2454347	7.688756
2521.775586	0.7511755	50.7832018	7.9387913
2520.333803	0.4253547	143.2432826	8.1887226
2523.011076	0.4524108	44.6645453	8.4388232
2536.693248	0.4040842	46.1623357	8.6887689

Table 8-1: A part of the recorded data of the same experiment to get Figure 8-5 (b)



(a) 5.5 to 7 second

(b) 5.5 to 5.7 second

(c) 5.58 to 5.63 second

Figure 8-7: Some details of Figure 8-5 (b)

8.3.3 Chatter Detection

The chatter detection module has been developed based on the methodologies presented in section 6.1. However, as shown in the section 8.3.1, the chatter frequencies in vertical direction and in horizontal direction are very different from the first natural frequencies of the machine in the respective directions, and they are also different from each other. The actual chatter frequencies in the two directions, while they are still close to each other, are much larger than the first natural frequencies of the machine in respective directions. Therefore, there should be some modifications in the original design as in section 6.1.1 if two pairs of accelerometer/actuator are used.

First, in Figure 6-1, f_{min} and f_{max} should not be decided by f_{Nx} and f_{Ny} any more. Based on the experiment results, f_{min} and f_{max} should be decided by the range of 2500 Hz to 2800 Hz (substituting f_{Nx} and f_{Ny} with 2500 Hz and 2800 Hz).

Second, in Figure 6-1, the threshold T of $(\gamma_{xy}^{2})_{\max}$ must be set based on experiment results since chatter frequencies are different in the two directions, although they are close to each other.

The developed chatter detection module works well for simulated signals. However, modifications need to be made based on experiments. Since only one set of piezo accelerometer and piezo actuator was utilized in the current work, the chatter detection methods in frequency domain, as introduced in section 2.9.2 and section 6.1.2, were more suitable.

8.3.4 Chatter Suppression

Based on the experiment results, some modifications must be made for the chatter suppression sub-system.

8.3.4.1 Modifications in ANN Model Training

Since the chatter frequencies in vertical direction and in horizontal direction are very different from the first natural frequencies of the machine in the respective directions, the ANN models need to be re-trained. Compared with the training in section 8.2, in which the region of interest for frequency was set to 400 Hz to 500 Hz, based on the experiment results, the range of 2300 Hz to 3000 Hz is used as the region of interest for frequency or added to the region of interest for frequency.

In section 8.2, the amplitude range for the control signal sent to the actuator was set to from 0.01 V to 5 V while the available safe range was from 0 V to 10 V. However, for a frequency range of 2300 Hz to 3000 Hz, as shown in Figure 8-3, the available safe range of the control signal is only 0 V to about 1.5 V. To find out the actual proper amplitude ranges to train the ANN models, cutting experiments should be undertaken. According to the results of some experiments, the amplitude of the control signal sent to the actuator may need to be larger than 1.5 V, which means that a better amplifier or an actuator with electrical capacitance less than 0.9 µF should be used. Therefore, the amplitude range of the control signal sent to the actuator is set to from 0.01 V to 1.5 V for the new frequency range of interest because of the limitation of current hardware.

The ANN models were trained in the same way as introduced in section 8.2. As mentioned before, the values of all the data are normalized to a range of 0.1 to 0.9 by using equation (5-2). To reduce harmful effects, e.g., the squashing effect, of using sigmoid transfer functions in the hidden layer and the output layer of the ANN model, and normalization, the above data ranges can be divided into several sub-ranges, e.g., the original frequency range can be divided into five smaller sub-ranges: 2300 Hz to 2500 Hz, 2500 Hz to 2600 Hz, 2600 Hz to 2700 Hz, 2700 Hz to 2800 Hz and 2800 Hz to 3000 Hz. ANN models are trained separately for different sub-ranges. Moreover, Resilient Backpropagation (RPROP) algorithm is utilized to train ANN models. The best ANN architecture found via experiments is also a multilayer feedforward ANN, which has two inputs, one hidden layer of 15 log-sigmoid neurons and one output layer of two log-sigmoid neurons. The MSE of validation data sets for this architecture was about 0.07%. Also, for the same tool setup, the ANN architecture does not change, but the weights between neurons change for different data sub-ranges after training.

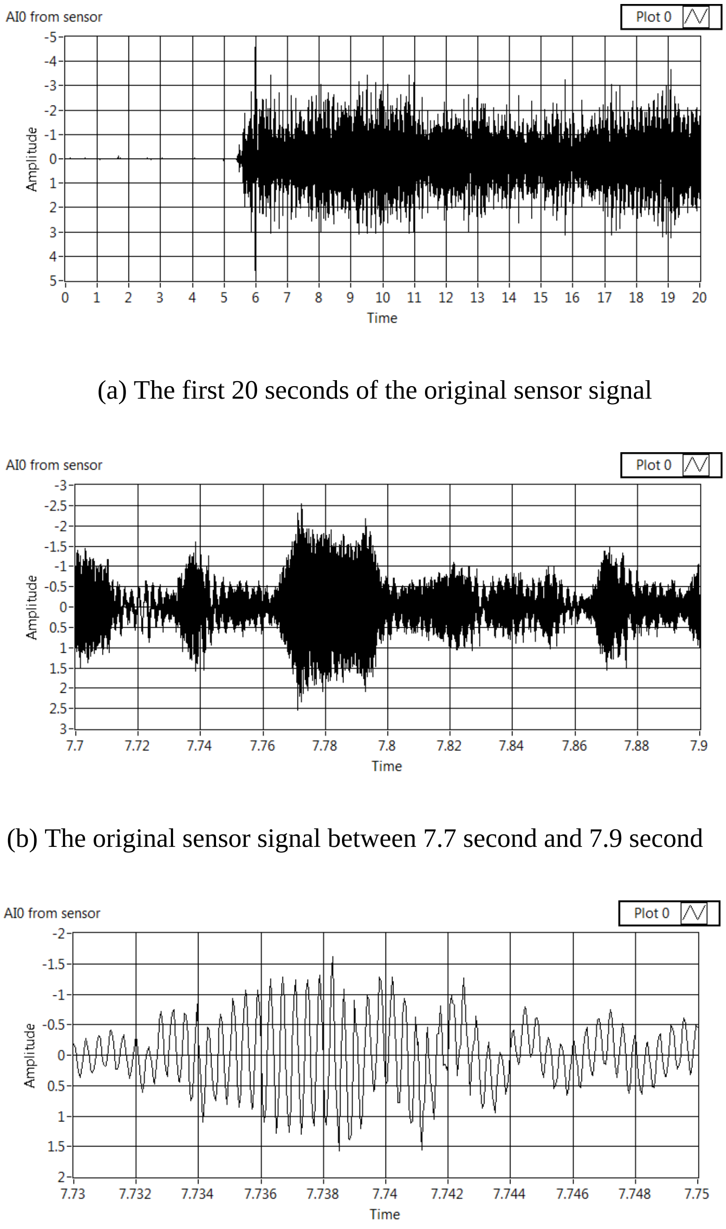
8.3.4.2 Chatter Suppression Analysis and Processing

The piezo actuator and the designed toolholder worked very well, as shown in the ANN model training data collection. When a sinusoidal signal is sent out to the actuator, a sinusoidal signal with excellent shape of the same frequency can be obtained from the sensor, which means a sinusoidal vibration of the tool is generated. However, actual chatter control experiments did not show satisfying outcomes.

One possible reason is the complexity of the signals from the sensor when chatter happens in the current work, as shown in section 8.3.2. Figure 8-8 also shows this complexity. In the experiment to obtain data for Figure 8-8, the spindle speed was about

500 RPM and Feed=1 (setting of MAXNC T2, about 0.015 inch/second without relation to the spindle speed here), while the cutting depth was about 0.02. Figure 8-8 (a) shows the first 20 seconds of the sensor signal; Figure 8-8 (b) shows the sensor signal between 7.7 second and 7.9 second; Figure 8-8 (d) shows the first 20 seconds of the filtered signal of the original sensor signal by using a 2539-2539.1 Hz bandpass IIR filter (the rationale to use this band was that this is the range with highest FFT amplitude in the general amplitude FFT at specific time); while Figure 8-8 (c) and Figure 8-8 (e) are the 7.73-7.75 second signals of the original sensor signal and the filtered signal. Figure 8-8 (b) shows clearly that the sensor signal includes big waves. With these big waves, the amplitudes always change between very large and very small rapidly, even within the same iteration. However, in the chatter suppression system, within the same iteration, the control signal does not change, while the control signal could be a combination of many sinusoidal signals. Therefore, the signal complexity is one reason for the unsatisfactory chatter control experiment outcomes. This problem cannot be solved by just setting a very small iteration time because, first, if iteration time is too small, measurement accuracy will become worse; second, the machine system needs time to reach relatively stable status.

The inconsistency of the chatter signals for the experimental setup can be traced to the lathe used for data collection. The lathe used in the current work is a mini lathe without a tailstock. The cutting conditions required to generate chatter were too aggressive for the construction of the lathe as shown in high frequency vibrations. Without the support of a tailstock, the force of the tool on the workpiece would cause it to bend away from the tool, producing unexpected signal signatures. Experiments also show that when the cutting is closer to the clamped end of the workpiece, the waves in the chatter signal are smaller.

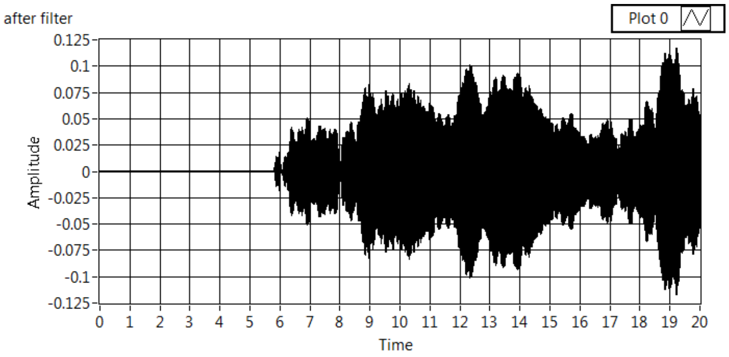


(a) The first 20 seconds of the original sensor signal

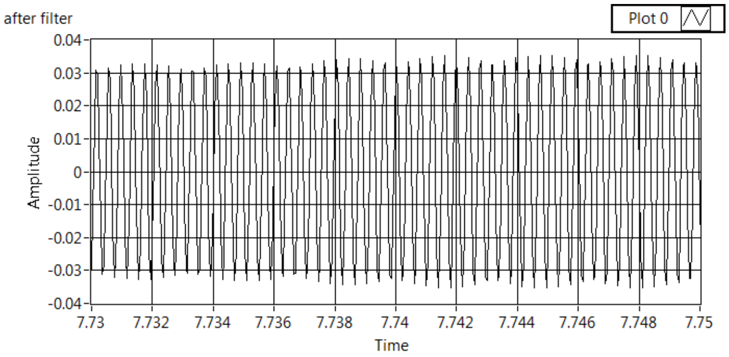
(b) The original sensor signal between 7.7 second and 7.9 second

(c) The original sensor signal between 7.73 second and 7.75 second

Figure 8-8 (a)-(c): Waveforms of the original signal and the filtered signal of a chatter example



(d) The first 20 seconds of the filtered signal



(e) The filtered signal between 7.73 second and 7.75 second

Figure 8-8 (d)-(e): Waveforms of the original signal and the filtered signal of a chatter example

By comparing Figure 8-8 (c) and Figure 8-8 (e), which are the 7.73-7.75 second signals of the original sensor signal and the filtered signal, it is obvious that the filtered signal is “cleaner”. The original signal can be considered as a combination of many sinusoidal signals, while it has big waves, big changes in amplitude, frequency and phase. However, the filtered signal is an excellent continuous sinusoidal signal without noticeable phase change, and in a small duration, its amplitude does not change as much as the original signal. Moreover, the waves in the filtered signal could be smaller on a regular lathe with a tailstock. In that case, the filtered signal will be similar to the vibration signal in the AVC system, which is a noisy sinusoidal signal as discussed in Chapter 3. Therefore, this kind of signal can be suppressed by using the designed chatter suppression system.

As shown in Table 8-1, the measured frequency changes in different iteration in the same experiment; also as shown in amplitude FFT graphs, the highest amplitude covers a range. This means only suppressing the filtered signal shown in Figure 8-8 is not enough. However, this should not be a problem for the designed ACS system because one strength of the system is that the system can generate a current control signal based on the current iteration measurements and add the current control signal to the original control signal to send to the actuator at the beginning of the next iteration as a new control signal (as discussed in Chapter 6).

8.3.4.3 Chatter Suppression Experiment Results

To reduce the effect of large amplitude vibrations resulting from excessive shaking of the lathe, cuttings were made close to the clamped end of the workpiece. Some chatter signals show that the sizes of big waves were reduced. Then, the designed chatter suppression system was used to control chatter signals with relatively regular pattern offline. Figure 8-9 shows two examples of offline control outcomes (the control system started to send out control signals after 10 seconds). These examples show that about 50% reduction in the vibration amplitude at the sensor location can be achieved.

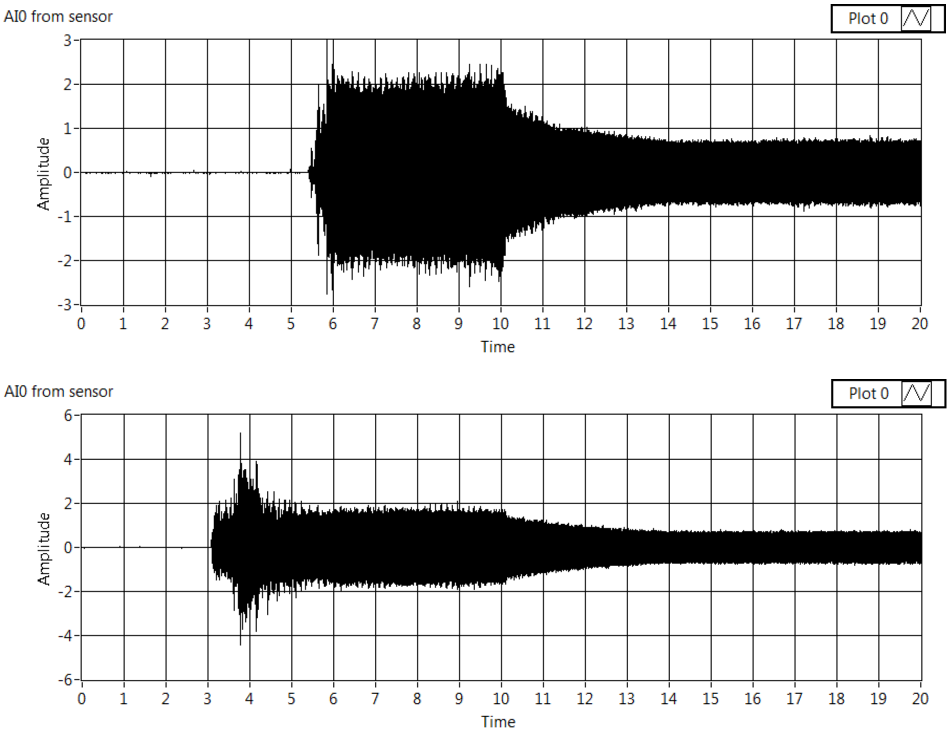


Figure 8-9: Examples of control outcomes

Next, the chatter suppression system was applied during cutting when chatter happened. As mentioned before, there is only one set of piezo accelerometer and piezo actuator. Since there is more power in vertical direction than in horizontal direction when chatter happens, which can be seen by comparing Figure 8-5 (b) and Figure 8-6, as discussed in section 8.3.2, the set of piezo accelerometer and piezo actuator is mounted in the vertical direction. The cuttings are close to the clamped end of the workpiece. Figure 8-10 shows that some outcomes are acceptable when chatter patterns are relatively regular. Before applying control, many small horizontal chatter stripes are visible on the workpiece surface. By counting the number of the small chatter stripes for one revolution, the chatter frequency can be calculated. In the experiment to get Figure 8-10, cutting is very close to the clamped end of the workpiece; the spindle speed was about 500 RPM; Feed=1 (setting of MAXNC T2, about 0.015 inch / second without relation to the spindle speed here); and the cutting depth was about 0.02 inch. Since there are about 300 small stripes for one revolution, the chatter frequency can be calculated as about 2500 Hz. After control, on the workpiece surface, the small chatter stripes are almost gone and only trails of the tool are obvious.

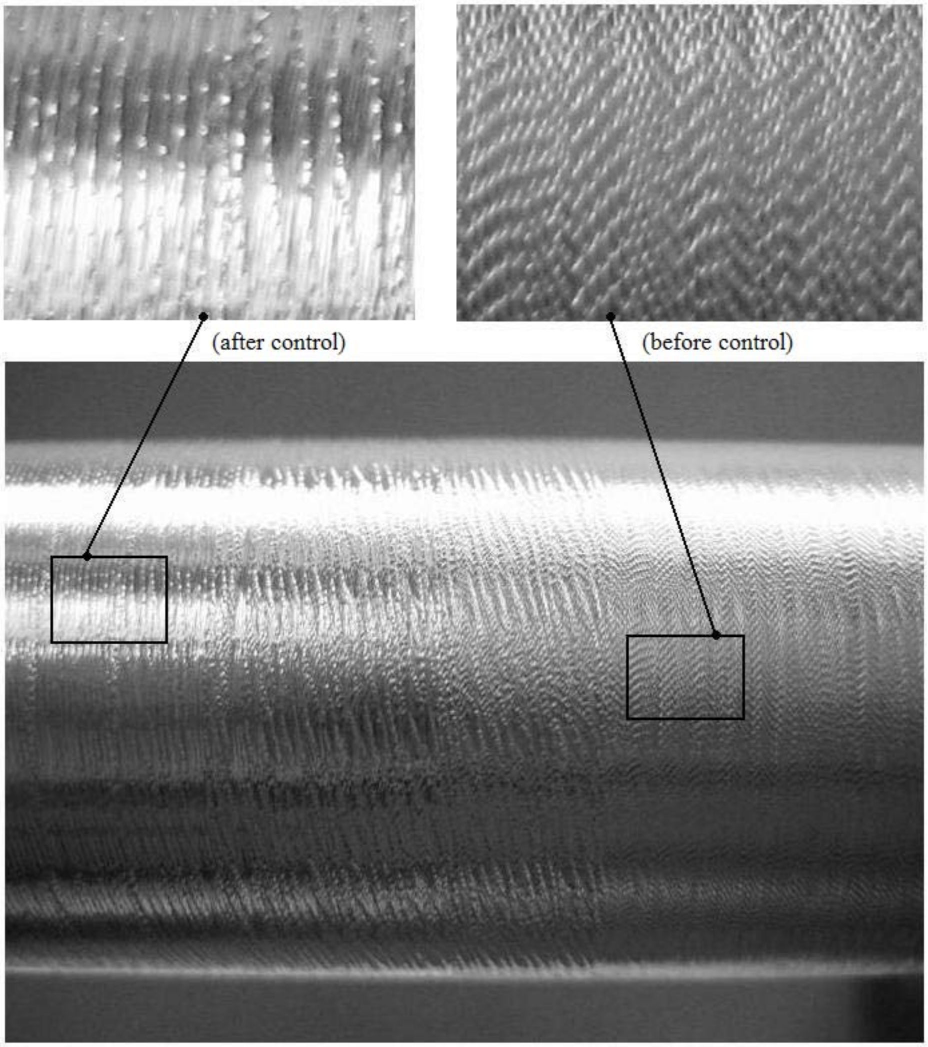


Figure 8-10: An example of good control outcome

Taking a close look at Figure 8-10, the depth of the grooves after control appear deeper than the chatter patterns before control was activated. Without chatter there is more power available for the cutting edge to cut deeper grooves.

Figure 8-11 shows that some outcomes are unsatisfactory when the chatter pattern is irregular, which means the waves in the chatter signal, as mentioned in section 8.3.2, are very irregular and choppy. In the experiment to get Figure 8-11, cutting is closer to the free end of the workpiece; the spindle speed was about 500 RPM; Feed=1.5 (setting of MAXNC T2, about 0.02 inch / second without relation to the spindle speed here); and the cutting depth was about 0.03 inch.

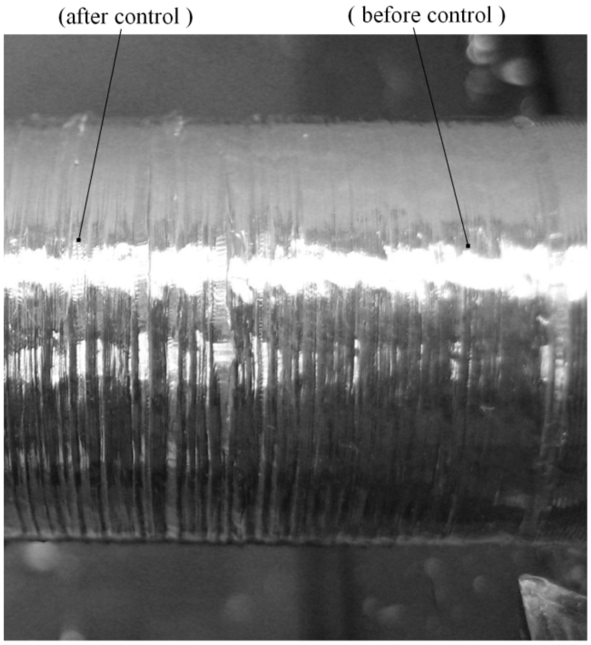


Figure 8-11: An example of unsatisfying control outcomes

8.4 Discussion

The experimental results show that the proposed ACS system can work when the chatter signals are regular, i.e., when cuts were made close to the clamped end of the workpiece and the big waves in chatter signals are not significant. By using a real-time developing environment, the designed ACS system can repeat the vibration detection and control loop as fast as in every 25 millisecond. The designed ACS system is a real-time online control system.

The ACS system is also robust when the experimental setup changes. When the setup changes, the ACS system can collect training data and train the ANN model automatically via running a calibration or initialization program before cutting and then the system is ready for the ACS of the new setup.

The system can always modify the control signal based on the feedback from the sensor after a specific time (one or more control iterations). As presented in Chapter 6 and section 8.1, the control system can repeat all calculations and generates a current control signal based on the detected vibration situation of current control iteration. This current control signal is added to the accumulated control signal, which is a combination of all previous continuous control signals, to get an updated control signal. The new updated control signal is sent out to the actuator at the beginning of the next iteration. This updated control signal becomes the “accumulated control signal” in the next control cycle. Therefore, the actual control signal sent to the actuator is an accumulation of all previous generated control signals, which are continuous. One control cycle may include more than one control iteration to avoid unstable transient conditions after the modification of the actual control signal, and therefore to get more accurate measurements of vibration.

In the current work, the experiment results show that the chatter frequencies were different from the respective first natural frequencies of the machining system (see section 8.3 for details). While the first natural frequency of the tool system in the vertical direction is about 488 Hz and in the horizontal direction is about 423 Hz, when chatter happens, in most cases, the measured frequency in vertical direction is between 2500 Hz and 2600 Hz, and in horizontal direction is between 2600 Hz and 2800 Hz. More complexities of chatter signals of the current experimental setup are shown in section 8.3.

The reasons for the complexities could be investigated by comparing the performances of the current lathe and a regular lathe with a tailstock. Moreover, the designed ACS system can be tested on a regular lathe with a tailstock. For a mini lathe without a tailstock, the cutting conditions required to generate chatter may be too aggressive. Without the support provided by using a tailstock, the force of the tool on the workpiece would cause it to bend away from the tool, producing unexpected signal signatures.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1 Contributions

A novel artificial neural network (ANN)-based active vibration control (AVC) technique has been developed. The developed AVC system can detect the vibration frequency with the highest power and suppress this sinusoidal signal in one control cycle. In subsequent cycles, vibration frequencies of next level of power will be suppressed. Technically, if the ANNs can be trained to cover enough frequency and amplitude ranges, after enough control cycles, most of the original vibration could be suppressed. The ANNs can be trained automatically for updated time delays for any changes in the system. The AVC system is experimentally verified to be effective, real-time, adaptable, robust, and easily implemented.

When applied to a machining chatter suppression system, the developed methodology can provide advantages over many other chatter suppression methods. Since it works without increasing the damping of the machine tool, it does not cause a decrease of efficiency. Unlike many other methods where a change in tool-workpiece configuration requires off-line changes in the system, the designed system in this work is online, adaptable and can self-tune. Using a feedback structure, the proposed methodology can deal with unexpected disturbances. Some active suppression methods use linear adaptive architecture/algorithm combinations and therefore cannot deal with nonlinearities while more capable ANNs are used in the proposed methodology. Moreover, compared with some other ANN-based active control methods, the proposed methodology can deal with time delay and be implemented in an easier way.

The proposed methodology can be used in lights-out machining systems where machines are kept running through a second or third shift without adding manpower.

9.2 Conclusions

In this work, an experimental AVC system, which includes hardware setup and real-time software development environment, has been successfully implemented. The efficiency, real-time execution, adaptability and robustness of the proposed methodology have been verified experimentally in the vibration control of a cantilever beam.

In addition, an experimental setup for the chatter suppression of a turning tool has been successfully implemented. The ANN-based AVC techniques have been utilized for active chatter suppression (ACS) for a mini-lathe. While chatter signals can be very irregular in the current work, experimental results show that the proposed ACS system can work for chatter signals with regular pattern.

Although the efficiency of the designed methodology has been verified experimentally, there are some limitations for it in applications. One limitation comes from ANNs used in the system. Since the errors of ANNs cannot be eliminated completely, there will always be some residual vibrations after control. Some limitations also come from hardware. For example, actuators and amplifiers usually have their working frequency and amplitude ranges. Therefore, if components of vibrations to be controlled are out of the working ranges, the vibrations cannot be eliminated completely.

9.3 Future Work

The ability of the proposed methodology should be tested on a production lathe for turning and boring operations. This will require much stronger actuators and custom built toolholders.

The chatter problems in milling operations are more complex because of the discontinuity of the cutting action. Nevertheless, the proposed methodology in this work can be adapted to this problem with some modifications including signal processing.

APPENDICES

S Series Multifunction DAQ Specifications

Specifications – PCI-6110, PCI-6111

These specifications are typical for 25 °C unless otherwise noted.

Analog Input

Accuracy specifications	See Table 2, page 199

Input Characteristics

Number of pseudodifferential inputs:	
PCI-6110	4
PCI-6111	2
Resolution	12 bits, 1 in 4,096
Maximum Sampling Rate	
6110/6111	5 MS/s
Minimum sampling rate	1 kS/s
Streaming-to-disk rate 6110/6111	8 MS/s, system dependent

Input signal ranges

Input Range (Software, Selectable per Channel)	Bandwidth (MHz)
±42 V	5.5
±20 V	4.4
±10 V	7.2
±5 V	4.8
±2 V	4.8
±1 V	4.4
±500 mV	4.4
±200 mV	4.1

Input coupling	DC or AC (software selectable)

Maximum working voltage for all analog input channels

Input Channels	Range	Maximum Working Voltage (Signal + Common Mode)
ACH<0..3>+	200 mV to 10 V	Should remain within ±11 V of ground
	20 to 50 V	Should remain within ±42 V of ground
ACH<0..3>-	All	Should remain within ±11 V of ground

Overvoltage protection	
On/Off	±50 V
Inputs protected	ACH<0..3>+, ACH<0..3>-
FIFO Buffer Size	
6110/6111	8,192 Samples
Data transfers	DMA, interrupts, programmed I/O
DMA modes	Scatter-gather (single transfer, demand transfer)

Transfer Characteristics

Relative accuracy	±0.5 LSB typical, ±1 LSB maximum
DNL	±0.3 LSB typical, ±0.75 LSB maximum
No missing codes	12 bits, guaranteed

Spurious free dynamic range (SFDR), DC to 100 kHz

Range	SFDR (dB)
200 mV to 10 V	75
20 to 42 V	70

Effective number of bits (ENOB)	11.0 bits, DC to 100 kHz

Amplifier Characteristics

Input impedance	
ACH<0..3>+ to ACH <0..3> -	
Normal powered on	1 MΩ in parallel with 100 pF
Powered off	1 MΩ minimum
Overload	1 MΩ
Impedence to ground	
ACH<0..3>- to ground	10 nF
Input bias current	±200 pA
Input offset current	±100 pA

CMRR, DC to 60 Hz

Range	CMRR (dB)
200 mV	72
500 mV	70
1 V	67
2 V	62
5 V	56
10 V	50
20 to 42 V	35

Dynamic Characteristics

Interchannel skew	1 ns typical; f_{in} = 100 kHz, 10 V range

System noise (LSB_{rms}, not including quantization)

Range	Noise
200 mV	1.0
500 mV	0.6
1 to 50 V	.05

Crosstalk	-80 dB, DC to 100 kHz

Analog Output

Output Characteristics

Number of channels	2 voltage outputs
Resolution	
6110/6111	16 bits, 1 in 65,536
Maximum update rate	
1 channel	4 MS/s, system dependent
2 channel	2.5 MS/s, system dependent
Type of DAC	Double buffered, multiplying
FIFO Buffer Size	
6110/6111	2,048 Samples
Data transfers	DMA, interrupts, programmed I/O
DMA modes	Scatter-gather (single transfer, demand transfer)

Transfer Characteristics

Relative accuracy	±4 LSB typical, ±8 LSB maximum
DNL	±2 LSB typical, ±8 LSB maximum

Voltage Output

Ranges	±10 V
Output coupling	DC
Output impedance	50 Ω ±5% Short
circuit current	±27 mA typical Output
stability	Any passive load
Protection	Short-circuit to ground
Power-on state	0 V, ±400 mV

Dynamic Characteristics

Settling time and slew rate

Setting Time for Full-Scale Step	Slew Rate
300 ns to ±0.01%	300 V/µs

Noise	1 mV_{rms}, DC to 5 MHz
Glitch energy (at mid-scale transition)	
Magnitude	350 pV

Digital I/O

Number of channels	8 input/output
Compatibility	5 V TTL/CMOS
Power-on state	Input (high impedance)
Data Transfers	
6110/6111	Programmed I/O

S Series Multifunction DAQ Specifications

Specifications – PCI-6110, PCI-6111 (continued)

Digital logic levels

Level	Minimum (V)	Maximum (V)
Input low voltage	0	0.8
Input high voltage	2.0	5.0
Output low voltage (Iout = 24 mA)	–	0.4
Output high voltage (Iout = 13 mA)	4.35	–

Timing I/O

General-Purpose Up/Down Counter/Timers

Number of channels	2
Resolution	24 bits
Compatibility	5 V TTL/CMOS

Digital logic levels

Level	Minimum (V)	Maximum (V)
Input low voltage	0	0.8
Input high voltage	2.0	5.0
Output low voltage (Iout = 5 mA)	–	0.4
Output high voltage (Iout = 3.5 mA)	4.35	–

Base clocks available	20 MHz and 100 kHz
Base clock accuracy	±0.01%
Maximum source frequency	20 MHz
External source selections	PFI <0..9>, RTSI <0..6>, analog trigger; software selectable
External gate selections	PFI <0..9>, RTSI <0..6>, analog trigger; software selectable
Minimum source pulse duration	10 ns, edge-detect mode
Minimum gate pulse duration	10 ns, edge-detect mode
Data transfers	DMA, interrupts, programmed I/O
DMA modes	Scatter-gather (single transfer, demand transfer)

Frequency Scaler

Number of channels	1
Resolution	4 bits
Compatibility	5 V/TTL

Digital logic levels

Level	Minimum (V)	Maximum (V)
Input low voltage	0	0.8
Input high voltage	2.0	5.0
Output low voltage (Iout = 5 mA)	–	0.4
Output high voltage (Iout = 3.5 mA)	4.35	–

Base clocks available	10 MHz, 100 kHz
Base clock accuracy	±0.01%
Data transfers	Programmed I/O

Triggers

Analog Triggers

Number of triggers	1
Purpose	
Analog input	Start and stop trigger, gate, clock
Analog output	Start trigger, gate, clock
General-purpose counter/timers	Source, gate

Source

PCI-6110	ACH<0..3>, PFI0/TRIG1
PCI-6111	ACH<0..1>, PFI0/TRIG1

Level	
Internal source, ACH<0..3>	±Full-scale
External source, PFI0/TRIG1	±10 V
Slope	Positive or negative; software-selectable
Resolution	8 bits, 1 in 256
Hysteresis	Programmable Bandwidth (-3 dB)
Internal source, ACH<0..3>	5 MHz
External source, PFI0/TRIG1	5 MHz

Digital Triggers

Number of triggers	2
Purpose	
Analog input	Start and stop trigger, gate, clock
Analog output	Start trigger, gate, clock
General-purpose counter/timers	Source, gate
Source	PFI<0..9>, RTSI<0..6>
Slope	Positive or negative; software selectable
Compatibility	5 V TTL
Response	Rising or falling edge
Pulse width	10 ns minimum

External input for digital or analog trigger (PFI0/TRIG1)

Impedance	10 kΩ
Coupling	DC or AC
Protection	
Digital trigger	-0.5 to V_{cc} + 0.5 V
Analog trigger	
On/off/disabled	±35 V

Calibration

Recommended warm-up time	15 minutes
Calibration Interval	1 year
Onboard calibration reference	
DC Level	5.000 V (±3.5 mV); actual value stored in EEPROM
Temperature coefficient	±0.6 ppm/°C maximum
Long-term stability	±6 ppm/$\sqrt{1{,}000h}$

RTSI

Trigger lines	7

Bus Interface

PCI	Master, slave

Power Requirements

Device	+5 VDC (±5%)	Power Available at I/O Connector
PCI-6110	3.0 A	+4.65 to 5.25 VDC, 1 A
PCI-6111	2.5 A	+4.65 to 5.25 VDC, 1 A

Physical

Dimensions	
Not including connectors	31.2 by 10.6 cm (12.3 by 4.2 in.)
I/O connector	68-pin male SCSI type

Environment

Operating temperature	0 to 45 °C
Storage temperature	-20 to 70 °C
Relative humidity	10 to 90%, noncondensing

Certifications and Compliances

CE Mark Compliance CE

[1] Bandwidth specifications are for signals on the (+) input with the (-) input at DC ground. The (-) input is slew rate limited to 24 V/µsec and has an additional 10 nF capacitance to ground.

AE Series Analog Inductive Proximity Sensors

M8 (8mm) metal – analog output

- 4 models available
- Compact metal housing
- Axial cable or M8 quick-disconnect models
- IP67 rated

AE Series M8 Analog Inductive Prox Selection Chart

Part Number	Price	Sensing Range	Housing	Output	Connection	Dimensions
AE9-10-1A	check	0 to 4.0mm (0-0.157in)	Shielded	0-10VDC	2m (6.5') axial cable	Figure 1
AE9-10-1F	check				M8 (8mm) connector	Figure 2

Specifications

	AE9-10-1*
Output Type	0-10VDC
Resolution	1µm
Repeat Accuracy	±0.01mm
Material Correction Factors	See Proximity Sensor Terminology
Operating Voltage	15-30VDC
Ripple	20%
No-load Supply Current	10mA
Voltage Output Minimum Load	1kO
Voltage Drop	2.0 V
Time Delay Before Availability	50ms
Input Voltage Transient Protection	Up to 30VDC
Input Power Polarity Reversal Protection	Yes
Output Power Short-Circuit Protection	Yes (switch autoresets after overload is removed)
Temperature Range	-25° to +70° C (-13° to 158° F)
Temperature Drift	10% Sr
Protection Degree (DIN 40050)	IEC IP67
Housing Material	Chrome-plated brass
Sensing Face Material	PBT
Tightening Torque	4Nm (0.71lb./in.)
Weight (cable/M8 connector)	50g (1.76 oz.) / 20g (0.71 oz.)

Wiring diagrams

Connector

Dimensions

Figure 1

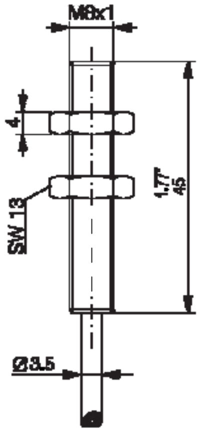
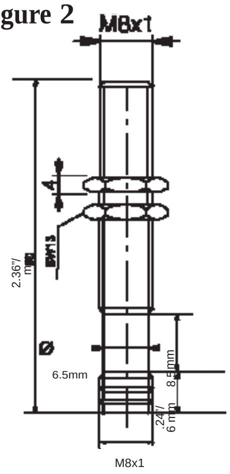
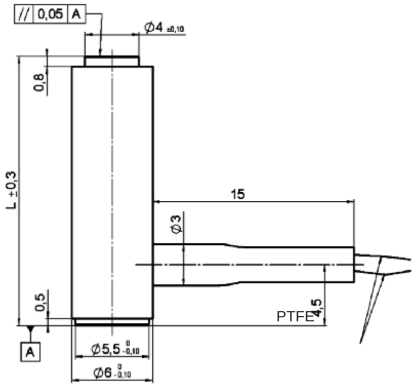
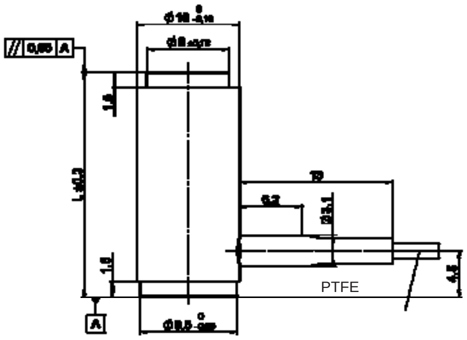


Figure 2





P-810 dimensions in mm. Two 0.1 m long teflon insulated pigtails. Length L: see table. Max. torque at top piece: 0.08 Nm.



P-830 dimensions in mm. Two 0.1 m long teflon insulated pigtails. Length L: see table. Max. torque at top piece: 0.2 Nm.

Technical Data and Product Order Numbers

Order number	Travel range for 0 to 100 V [µm] ±20 %	*Resolution [nm]	**Static large-signal stiffness [N/µm] ±20 %	Push- / pull force capacity [N]	Electrical capacitance [µF] ±20 %	Dynamic operating current coefficient [µA / (Hz • µm)]	Resonant frequency (unloaded) [kHz] ±20 %	Mass [g] ±5 %
P-810.10	15	0.15	14	50 / 1	0.3	3.0	22	4
P-810.20	30	0.3	7	50 / 1	0.7	3.0	15	6
P-810.30	45	0.45	4	50 / 1	1.0	3.0	12	8
P-830.10	15	0.15	57	1000 / 5	1.5	12.5	23	10
P-830.20	30	0.3	27	1000 / 5	3.0	12.5	14	16
P-830.30	45	0.45	19	1000 / 5	4.5	12.5	10	21
P-830.40	60	0.6	15	1000 / 5	6.0	12.5	8.5	27

Ceramic Shear Accelerometer

Type 8778A500...

Ultra Miniature, 0.4 Gram Weight, Voltage Mode Accelerometer

Small, light weight general purpose accelerometer for vibration measurements in wide range of applications. Available in two cable versions, the standard with a permanent attached cable and the M14 with a field replaceable twisted wire pair.

- Low impedance, voltage mode
- Ultra low base strain and thermal transient response
- Wide frequency response, 2 ... 9000 Hz (±5 %)
- Ground isolated assembly
- High 10 mV/g sensitivity
- Conforming to CE

Type 8778A500sp

Type 8778A500M14sp

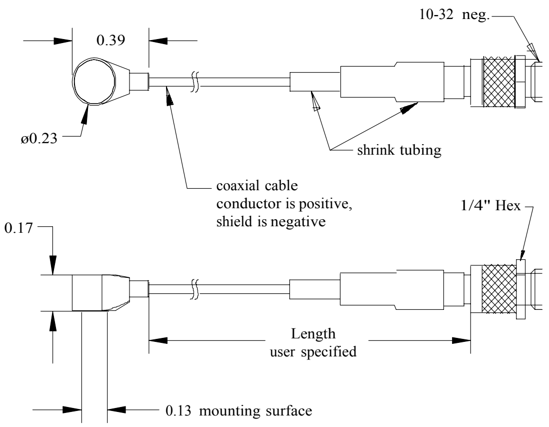
Description

The Type 8778A500... and 8778A500M14 are high frequency, ultra miniature, light weight accelerometers that contain uniquely designed ceramic shear sensing element. The shear mode element design provides an immunity to thermal transients, base strain and transverse motion.

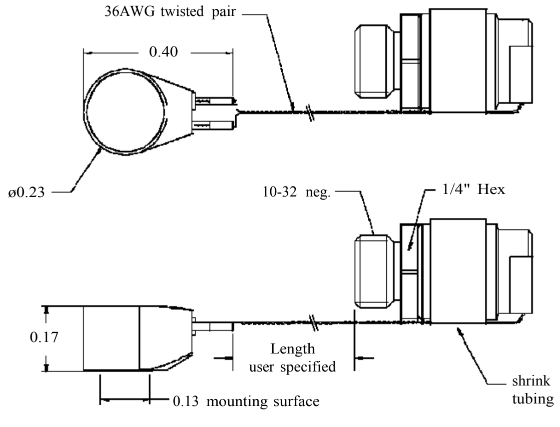
An internal microelectronic Piezotron® signal conditioning circuit converts the charge developed in the ceramic element as a result of the accelerometer being subjected to a vibration, into a useable high level voltage output signal at a low impedance level. The standard Type 8778A500... accelerometer includes an integral Teflon® jacketed 3 ft long cable terminated with a 10-32 neg. connector while the M14 version features a field replaceable twisted wire pair and connector. The units are designed for wax or adhesive mounting and is supplied with a custom wrench to facilitate removal after testing. Power to the Type 8778A500… accelerometers can be provided by any Kistler coupler Type 51... or by any industry standard voltage mode IEPE (Integral Electronic Piezo-Electric) power supply/coupler.

Application

The light weight, low profile and small size of the accelerometer Type 8778A500… makes it ideal for: precision vibration measurements; modal analysis on small, thin walled structures or where space is limited and mass loading is of primary concern.
Typical applications included PC Board stress screening and critical component evaluation on disk drive assemblies.



Type 8778A500sp



Type 8778A500M14sp

Technical Data

Specification	Unit	Type 8778A500...
Acceleration range	g	±500
Acceleration limit	gpk	±750
Threshold (noise 100 µVrms), nom.	grms	0.01
Sensitivity, ±5 %	mV/g	10
Resonant frequency mounted, nom.	kHz	70*
Frequency response, ±5 %	Hz	2 … 9000*
Amplitude non-linearity	%FSO	±1
Time constant, nom.	s	≥0.3
Transverse sensitivity, nom. (max. 5)	%	3

Environmental

Base strain sensitivity @ 250 µε	g/µε	0.009*
Shock limit (1 ms pulse)	gpk	5000
Temperature coefficient of sensitivity	%/°F	–0.08
Operating temperature range (4 mA supply current)	°F	–65 … 250

Output

Bias, nom.	VDC	11
Impedance	Ω	≤100
Voltage full scale	V	±5

Source

Voltage	VDC	18 … 30
Constant current	mA	2 ... 20

Construction

Sensing element	Type	ceramic-shear
Case/base	material	Aluminum/hard anodized/Titanium
Degree of protection case/connector	Type	epoxy
Connector-terminates	Type	10-32 neg.
Ground isolation, min.	MΩ	10
Weight (excluding cable)	grams	0.4
Mounting	Type	adhesive/wax

The Type 8778A500… can be attached to the test structure by adhesive or wax. The accelerometer's side cable facilitates orientation in confined areas. Reliable and accurate measurements require that the mounting surface be clean and flat. The operating instruction manual for the accelerometer Type 8778A... (002-085) provides detailed information regarding mounting surface preparation.

The recommended adhesives to be placed between the accelerometer's base and the test object surface include:

- Petro wax, Type 8432
- Loctite 430 general purpose for adhesion to metals
- Loctite 495 general purpose for adhesion to other materials

Note: Removal of an adhesively mounted unit is extremely difficult and care should be exercised during the removal process. An appropriate adhesive solvent and the Type 1378 custom designed removal wrench should be used to twist the accelerometer off of the test object.

Included Accessories	Type
• Petro wax	8432
• Removal wrench	1378

Optional Accessories	Type
• Connecting cable for low impedance sensors	1761B
• Cable kit (needed to connect Type 8778A500... to Kistler couplers)	1764A

Ordering Key

Type 8778A500 □

Connector/Cable

Solder pins/no cable	**M14**
10-32 neg./integral coax cable	**sp**
10-32 neg./repairable twisted pair cable	**M14sp**

Related Accelerometers	Type
• integral cable, 1.6 gram weight	8728A500
• top 10-32 connector, 1.9 gram weight	8730A500
• integral cable, 1.1 gram weight	8732A500
• integral cable, 1.1 gram weight with mounting flange	8734A500

REFERENCES

1. C. R. Fuller, S. J. Elliott, and P. A. Nelson, “Active Control of Vibration,” Academic, New York, 1995.

2. S. J. Elliott and P. A. Nelson, “Active noise control,” IEEE Signal Processing Mag., vol. 10, pp. 12–35, 1993.

3. G. Tlusty, “Manufacturing Processes and Equipment”, Prentice Hall Inc., NJ, 2000.

4. S. A. Tobias, “Machine-Tool Vibration”, Blackie and Sons Ltd., Scotland, 1965.

5. D. J. Inman, “Engineering Vibration,” Prentice-Hall, Englewood Cliffs, NJ, 1994.

6. C. H. Hansen and S. D. Snyder, “Active Control of Noise and Vibration,” E&FN Spon, London, U.K., 1997.

7. S. D. Snyder and N. Tanaka, “Active Control of Vibration Using a Neural Network,” IEEE Transactions on Neural Networks, Vol. 6, No. 4, pp. 819-828, July 1995.

8. B. Widrow and S. D. Stearns, “Adaptive Signal Processing,” Prentice-Hall, Englewood Cliffs, NJ, 1985.

9. S. Haykin, “Adaptive Filter Theory,” 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1996.

10. M. Hagan and H. Demuth, “Neural Network Design,” PWS Publishing Co., Boston, MA, 1996.

11. S. M. Kuo and D. R. Morgan, “Active Noise Control Systems: Algorithms and DSP Implementations,” Wiley, New York, 1996.

12. http://www.micromega-dynamics.com/vibration-strategies.htm

13. L. H°akansson, “The Filtered-x LMS Algorithm,” University of Karlskrona/Ronneby, Sweden.

14. J. J. Shynk, “Adaptive IIR Filtering,” IEEE Acoust. Speech Signal Processing Mag., pp. 4–21, 1989.

15. http://soundlab.cs.princeton.edu/learning/tutorials/DSP/DSP.html

16. http://www.dspguru.com/info/faqs/fir/basics.htm

17. M. Bouchard, B. Paillard, and C. T. LeDinh, “Improved Training of Neural Networks for the Nonlinear Active Control of Sound and Vibration,” IEEE Transactions on Neural Networks, Vol. 10, No. 2, pp. 391-401, March 1999.

18. M. Bouchard, “New Recursive-Least-Squares Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks,” IEEE Transactions on Neural Networks, Vol. 12, No. 1, pp. 135-147, January 2001.

19. http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/

20. S. Haykin, “Neural Networks: A Comprehensive Foundation,” 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.

21. K.S. Narendra and K. Parthasarathy, “Gradient methods for the optimization of dynamical systems containing neural networks,” IEEE Trans, Neural Networks, Vol. 2, pp. 252-262, 1991.

22. T. Insperger, G. Stepan, P.V. Bayly, and B.P. Mann, “Experimental Investigation of Active Vibration Control Using Neural Networks and Piezoelectric Actuators,” Institute of Physics Publishing, February 2001.

23. W. C. Choi and N. W. Kim, “Experimental Study on Active Vibration Control of a Flexible Cantilever Using an Artificial Neural-network State Predictor,” Multimedia Institute, LG Electronics Ltd, 16 Umyon-dong, Sucho-go, 137-240, August 1996.

24. G. C. M. Deabreu, R. L. Teixeira and J. F. Ribeiro, “A Neural Network-Based Direct Inverse Control for Active Control of Vibrations of Mechanical Systems,” IEEE 1522-4899, 2000.

25. R. Jha and J. Rower, “Experimental investigation of active vibration control using neural networks and piezoelectric actuators,” Institute of Physics Publishing, November 2002.

26. http://ni.com/

27. http://msdn.microsoft.com/library/

28. Y. Xia and A. Ghasempoor, “Neural network-based active vibration control,” Dig. 21st Canadian Congress of Applied Mechanics, Canada, 2007.

29. Y. Xia and A. Ghasempoor, “Adaptive active vibration suppression of flexible beam structures,” Pro. of the Ins. of Mech. Engineers, Part C: Journal of Mechanical Engineering Science, Vol. 222, No. 3, pp.357-364, 2008.

30. Y. Xia and A. Ghasempoor, “Active Vibration Suppression Using Neural Networks,” Proceedings of the World Congress on Engineering 2009, WCE 2009, London, U.K. July, 2009.

31. S. A. Tobias, “Machine-Tool Vibration”, Blackie and Sons Ltd., Scotland, 1965.

32. E. Salje, “Self-Excited Vibrations of Systems with Two Degrees of Freedom”, Transactions of the ASME, Vol. 78, pp. 737–748, 1956.

33. http://highspeedmachining.mae.ufl.edu

34. G. Tlusty, “Manufacturing Processes and Equipment”, Prentice Hall Inc., NJ, 2000.

35. J. R. Pratt and A. H. Nayfeh, “Chatter Control and Stability Analysis of a Cantilever Boring Bar Under Regenerative Cutting Condition”, Phil. Trans. R. Soc. Lond. A 359, pp. 759–792, 2001.

36. T. Choi and Y. C. Shin, “On-Line Chatter Detection Using Wavelet-Based Parameter Estimation”, Journal of Manufacturing Science and Engineering, Vol. 125, pp. 21-28, 2003.

37. M. Wang and R. Fei, “On-Line Chatter Detection and Control in Boring Based on an Electrorheological Fluid”, Mechatronics, Vol. 11, pp. 779-792, 2001.

38. X. Q. Li, Y. S. Wong and A. Y. C. Nee, “Tool Wear and Chatter Detection Using the Coherence Function of Two Crossed Accelerations”, Int. J. Mach. Tools Manuf., Vol. 37, No. 4, pp. 425-435, 1997.

39. I. N. Tansel, A. Wagiman and A. Tziranis, “Recognition of chatter with neural networks”, Int. J. Mach. Tools Manufact. 31: 539-552, 1991.

40. J. Gradisek, E. Govekar and I. Grabec, “Using Coarse-grained Entropy Rate to Detect Chatter in Cutting”, J. Sound Vib., Vol. 214, No. 5, pp. 941–952, 1998.

41. G. W. Wornell, “Signal Processing with Fractals; A Wavelet-based Approach”, Prentice Hall, Upper Saddle River, NJ, 1996.

42. Z. Y. Kelson, and W. C. Hsueh, “Suppression of Chatter in Inner-Diameter Cutting”, JSME Int. J. Series C: Dyn., Cont. Robot., Design, Manuf., Vol. 39, No. 1, pp. 25-36, 1996.

43. Satoshi Ema and Etsuo Marui, “Suppression of chatter vibration of boring tools using impact dampers”, International journal of machine tools & manufacture, Vol. 40, pp. 1141–1156, 2000.

44. A. H. Slocum, E. R. Marsh and D. H. Smith, “A new damper design for the machine tool structure: the replicated internal viscous”, Precision Engineering, Vol. 16, No. 3, pp. 174-183, 1994.

45. Y. S. Liao and Y. C. Young, “A new on-line spindle speed regulation strategy for chatter control”, International journal of machine tool & manufacturing, Vol. 36, No. 5, pp. 651-660, 1996.

46. S. Smith and T. Delio, “Sensor-based chatter detection and avoidance by spindle speed selection”, ASME, Journal of Engineering for Industry, Vol. 114, pp. 486–492, 1992.

47. C. R. Liu and T. M. Liu, “Automated Chatter Suppression by Tool Geometry Control”, ASME, Journal of Engineering for Industry, Vol. 107, pp. 95–98, 1985.

48. Y. S. Tarng, H. T. Young and B. Y. Lee, “An Analytical Model of Chatter Vibration in Metal Cutting”, International Journal of Machine Tools & Manufacture, Vol. 34, pp. 183–197, 1994.

49. Y. S. Tarng, J. Y. Kao and E. C. Lee, “Chatter suppression in turning operations with a tuned vibration absorber”, Journal of Materials Processing Technology, Vol. 105, pp. 55-60, 2000.

50. M. Wang and R. Fei, “Improvement of machining stability using a tunable-stiffness boring bar containing an electrorheological fluid”, Smart Mater., Struct., Vol. 8, pp. 511–514, UK, 1999.

51. H. Tanaka and F. Obata, “Active Chatter Suppression of Slender Boring Bar Using Piezoelectric Actuators”, JSME International Journal, Series C: Dynamics, Control, Robotics, Design and Manufacturing, Vol. 37, pp. 601-606, 1994.

52. D. R. Browning, I. Golioto, and N. Thompson, “Active Chatter Control System for Long-Overhang Boring Bars”, Lucent Technologies/Bell Labs, Whippany N.J., 2004.

53. P. Stoica, “List of references on spectral line analysis,” Signal Processing, Vol. 31, pp. 329–340, 1993.

54. J. R. Trapero, H. Sira-Ramirez, and V. Feliu-Batlle, “An algebraic frequency estimator for a biased and noisy sinusoidal signal,” Signal Processing, Vol. 87, pp. 1188–1201, 2007.

55. M. Riedmiller, and H. Braun, “A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm,” Proceedings of the IEEE International Conference on Neural Networks, San Francisco, 1993.

56. http://www.engineeringtoolbox.com/

57. http://www.mcelwee.net/html/densities_of_various_materials.html

58. http://darkwing.uoregon.edu/~struct/courseware/461/461_lectures/461_lecture28/461_lecture28.html

59. D. C. Montgomery, “Design and Analysis of Experiments,” 5th ed., John Wiley & Sons, 2001.

60. http://www.physikinstrumente.com/