DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Active Vibration and Noise Control in Wood Machining Process

Monograph submitted to the Federal University of Santa Catarina

As a requirement for approval in the subject:

DAS 5511: End Couse Project

Leonardo Mortean Ono

Braunschweig, April 2009

Active Vibration and Noise Control in Wood Machining Process

Leonardo Mortean Ono

This monograph was judged in the context of the subject
DAS 5511: End Course Project
And approved in its final form by the
Control and Automation Engineering Course

Examining Committee:

Dr.-Ing. Hans-Werner Hoffmeister
Local Advisor

Prof. Dr. Eng. Rodrigo Lima Stoeterau
University Advisor

Prof. Augusto Humberto Bruciapaglia
Program Coordinator

Prof. Eugênio de Bona Castelan, Evaluator

Diego Machado Vieira, Discussant

Bruno Burigo Brandl, Discussant

Resumo

O presente trabalho estuda diferentes abordagens para o controle ativo de vibrações e ruído através de um controle adaptativo aplicado a um processo de usinagem de madeira. O algoritmo de controle ativo testado foi o LMS.

A escolha do algoritmo LMS foi feita porque é o algoritmo mais utilizado em aplicações de ANC, além de ser relativamente simples de projetar e implementar.

O objeto de estudo deste projeto é uma máquina CNC de usinagem de madeira, situada no Instituto de Máquinas-Ferramentas e Tecnologia de Produção na cidade de Braunschweig, Alemanha. A máquina em questão apresentava um ruído muito alto durante o seu funcionamento, tornando-se um inconveniente para seus usuários.

O trabalho está basicamente dividido em duas partes. Primeiro, algumas abordagens de controle são testadas em simulação através do programa Matlab para verificar a eficiência de cada abordagem.

Finalmente, o algoritmo e os métodos de controle são implementados no processo real de usinagem de madeira. Atuadores piezoelétricos, acelerômetros e microfones foram usados no controle e monitoramento das vibrações e ruídos, respectivamente. Resultados satisfatórios de redução de vibração e ruído são obtidos, mostrados e discutidos para enfim escolher a abordagem mais eficiente.

Palavras-chave: Controle Ativo, Vibração, Ruído, Algoritmo Adaptativo

Abstract

The present work studies different approaches for active vibration and noise control through an adaptive control applied in a wood machining process. The active control algorithm tested was the LMS.

The choice of LMS algorithm was made because it is the most commonly used in ANC applications, besides it is relatively simple to design and implement.

The object of study of this project is a CNC programmable stationary wood machine system, located at the Institute of Machine Tools and Production Technology, in Braunschweig, Germany. The machine presented a very loud noise while working, becoming an inconvenient for its users.

The work is basically divided in two parts. First, some control approaches are tested in simulation through the program Matlab in order to verify the performance of each approach.

Finally, the algorithm and the control methods are implemented in the real process of wood machining. Piezoactuators, accelerometers and microphones were used for control and monitoring of vibration and noise, respectively. Satisfactory results of vibration and noise reduction are obtained, showed and discussed to finally choose the most efficient approach.

Keywords: Active Control, Vibration, Noise, Adaptive Algorithm

Resumo estendido

Problemas de ruído no ambiente têm ganhado muita atenção devido ao grande crescimento tecnológico que levou a criação de motores ruidosos, máquinas pesadas, bombas e muitas outras fontes de ruídos. A exposição a ruídos muito altos provou-se prejudicial para os seres humanos em ambos aspectos físico e psicológico. Devido a isso, o problema do controle do nível de ruído em um ambiente tem sido foco de um grande número de pesquisas no decorrer dos anos.

A abordagem clássica de cancelamento de ruído é a abordagem passiva. Técnicas passivas de redução como absorção do som ou isolamento são inerentemente estáveis e eficientes sobre uma grande faixa de freqüências. Contudo, essa técnica tende a ser cara, ocupa um espaço considerável e geralmente é ineficiente no cancelamento de ruídos de baixa freqüência. O desempenho desses sistemas é também limitado a uma estrutura fixa e tem mostrado inviável em um número de situações onde o espaço é valorizado e a adição de mais um volume pode se tornar um empecilho. As limitações dos métodos passivos de redução de ruído têm dado impulso para pesquisas e aplicações de métodos alternativos de controle de ruído no ambiente, como o Controle Ativo de Ruído.

O Controle Ativo de Ruído envolve um sistema eletroacústico ou eletromecânico que cancela um ruído primário (indesejável) baseado no princípio da superposição; mais especificamente, um antiruído com igual amplitude porém com fase oposta é gerado e combinado com o ruído primário, resultando no cancelamento de ambos os ruídos.

O Controle Ativo consegue atenuar com eficiência ruídos de baixa freqüência, e devido a isso tem se desenvolvido rapidamente permitindo uma melhoria no controle de ruído, juntamente com potenciais benefícios em tamanho, peso, volume e custo.

Várias técnicas de processamento de sinal têm sido propostas no decorrer dos anos para redução de ruído no ambiente. O crescimento explosivo dos algoritmos e tecnologias de processamento digital têm ajudado na aplicação dessas técnicas no mundo real. Processadores Digitais de Sinal têm encolhido significativamente em tamanho enquanto suas capacidades de processamento têm crescido exponencialmente.

Uma vez que as características da fonte de ruído acústico no ambiente são variantes no tempo, freqüência, amplitude, fase e velocidade do ruído gerado são não estacionários. Desta forma o sistema do Controle Ativo deve ser adaptativo para poder enfrentar essas variações. Filtros adaptativos ajustam seus coeficientes com o objetivo de minimizar o sinal de erro e pode ser projetado de diversas maneiras. O filtro adaptativo mais comum e o que será usado neste projeto é o algoritmo LMS, que se destaca pela sua simplicidade de projeto e implementação.

Muitos autores obtiveram com sucesso uma boa redução de ruído usando alto falantes. Contudo, o presente projeto visa a diminuição do ruído usando técnicas de controle de vibrações. Neste caso, os alto falantes darão lugar a atuadores piezo elétricos.

Deste modo, o objetivo deste projeto é estudar a influência da Redução Ativa de Vibrações sobre a redução do ruído no ambiente de um processo de usinagem de madeira.

Uma motivação para esta pesquisa é que, uma vez tendo conseguido a redução de ruído através do controle de vibrações, ruídos indesejáveis do sistema serão reduzidos, garantindo assim um melhor ambiente de trabalho. Além disso, a redução de vibração é responsável por proporcionar um uso mais eficiente das ferramentas de usinagem e dos recursos de madeira disponíveis.

Contents

Resumo ii

Abstract iii

Resumo estendido iv

Contents vi

Chapter 1: Introduction 1

1.1: Problem Definition 1

1.2: Objectives 1

1.3: Motivation 2

1.4: Outline of the Monograph 2

Chapter 2: Literature Review 3

2.1: Noise and Vibration 3

2.1.1: Vibration 3

2.1.2: Noise 4

2.2: Introduction to Active Noise and Vibration Control 4

2.2.1: Adaptive Transversal Filters 5

2.2.2: Feedforward AVC Systems 6

2.2.3: Secondary Path effects 7

2.2.4: Feedback AVC Systems 10

2.2.5: Hybrid AVC Systems 11

Chapter 3: Control Simulation in Matlab 13

3.1: Feedforward AVC 14

3.1.1: Sin reference 14

3.1.2: Impulse reference 16

3.2: Feedback AVC ... 18

3.3: Hybrid AVC ... 19

Chapter 4: Physical Experimental Set Up ... 22

4.1: The Wood Machining System ... 22

4.2: Adaptronics for machine tools ... 24

4.3: Graphical User Interface ... 26

Chapter 5: Experimental Results with AVC ... 29

5.1: Experiment using shaker ... 29

5.1.1: Feedforward Control ... 31

5.1.2: Feedback Control ... 41

5.2: Experiment with real process ... 41

5.2.1: Sine Wave as a Reference ... 43

5.2.2: Impulse as Reference ... 47

5.3: Discussion of results and final considerations ... 49

Chapter 6: Conclusions and Future Works ... 51

References ... 52

Appendix A – FxLMS Control Algorithm ... 53

Chapter 1: Introduction

1.1: Problem Definition

Acoustic problems in the environment have gained attention due to the tremendous growth of technology that has led to noisy engines, heavy machinery, pumps, high speed wind buffeting and a myriad other noise sources. Exposure to high decibels of sound proves damaging to humans from both a physical and a psychological aspect. The problem of controlling the noise level in the environment has been the focus of a tremendous amount of research over the years.

In the case of a stationary wood machining system, the object of study of this project, the noise produced by the machine while it is working is too loud for the environment, becoming an inconvenient for the users.

The classical approach to noise cancellation is a passive acoustic approach. Passive silencing techniques such as sound absorption and isolation are inherently stable and effective over a broad range of frequencies. However, these tend to be expensive, bulky and generally ineffective for canceling noise at the lower frequencies. The performance of these systems is also limited to a fixed structure and proves impractical in a number of situations where space is at a premium and the added bulk can be a hindrance. The shortcomings of the passive noise reduction methods have given impetus to the research and applications of alternative methods of controlling noise in the environment, such as the Active Noise Cancellation (ANC).

1.2: Objectives

Various signal processing techniques have been proposed over the years for noise reduction in the environment. The explosive growth of digital processing algorithms and technologies has given an impetus to the application of these techniques to the real world. Digital Signal Processors (DSPs) have shrunk tremendously in size while their processing capabilities have grown exponentially.

Many authors have successfully obtained a good noise reduction using loud speakers. However, the present project aims to obtain the noise reduction using techniques of vibration control. In this case the loud speakers are replaced by piezo actuators.

Thus, the objective of this project is to study the influence of the Active Vibration Control over the noise reduction around the environment in wood milling process.

1.3: Motivation

One motivation for this research is that, once the noise reduction is achieved through vibration control, undesirable system's noise will be reduced, ensuring a better work environment. Also, vibration reduction is responsible to provide more efficient uses of available wood resources.

1.4: Outline of the Monograph

The monograph is organized in the following way:

- Chapter 1 presents the problem, objectives and motivation of the project;
- Chapter 2 discusses the literature review about Active Noise and Vibration Control;
- Chapter 3 presents the simulations made for this project;
- Chapter 4 brings the implementation of the solution in the physical system;
- Chapter 5 discusses the practical results reached of this work;
- Chapter 6 presents the conclusions obtained.

Chapter 2: Literature Review

2.1: Noise and Vibration

2.1.1: Vibration

Milling processes are, because of their wide range of applications, the most commonly used machining processes in woodworking. In peripheral milling (Figure 2.1), vibrations are often related to the periodic changes of the relative position between the workpiece and the tool. If uncontrolled, vibrations and chatter can reduce surface finish, limit dimensional accuracy, increase tool wear, create high levels of noise and can even be the cause for tool breakage. These undesirable performance attributes can, in turn, lead to increased machine wear, reduced throughput and higher scrap production. Furthermore, conventional approaches for control of vibrations and chatter, such as increasing the stiffness of the tool and reducing the cutting depth or machine speed, are no longer satisfactory and clearly reduce production throughput.

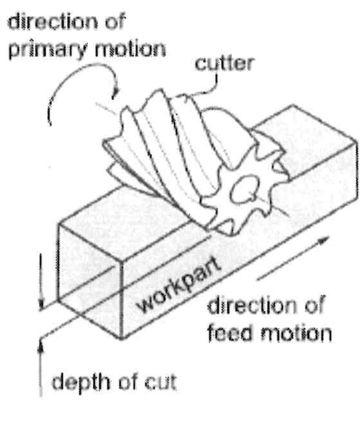

Figure 2.1 – Peripheral milling operation.

2.1.2: Noise

Acoustic noise problems become more and more evident as increased numbers of industrial equipment such as engines, blowers, fans, transformers, and compressors are in use.

Two types of noise exist in the environment, broadband and narrowband. Broadband noise is totally random and distributes its energy more or less evenly across the frequency band. By contrast, narrowband noise concentrates most of its energy at specific frequencies. This noise is related to rotating or repetitive machines, so it is periodic or nearly periodic.

2.2: Introduction to Active Noise and Vibration Control

Active noise control (ANC) involves an electroacoustic or electromechanical system that cancels the primary (unwanted) noise based on the principle of superposition; specifically, an antinoise of equal amplitude and opposite phase is generated and combined with the primary noise, thus resulting in the cancellation of both noises, as shown in Figure 2.2.

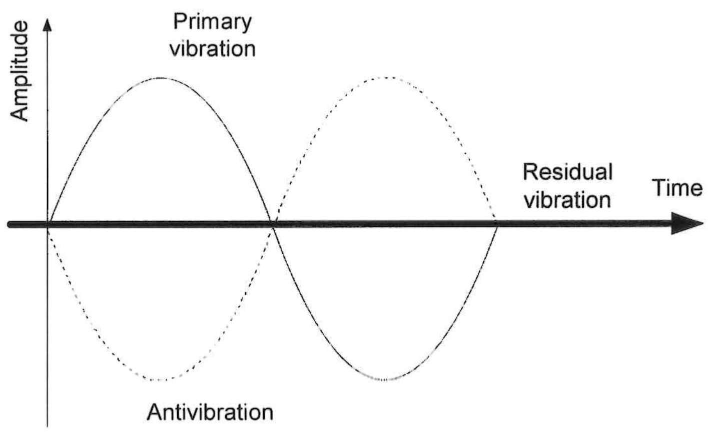

Figure 2.2 – Principle of superposition

The ANC system efficiently attenuates low-frequency noise where passive methods are either ineffective or tend to be very expensive or bulky. ANC is developing rapidly because it permits improvements in noise control, often with potential benefits in size, weight, volume, and cost.

The structures and algorithms developed for acoustic ANC can be equally applied to Active Vibration Control (AVC), which is used to isolate the vibrations from a variety of machines, as well as to stabilize various platforms in the presence of vibration disturbances. As the performance and reliability of active systems continues to improve and the initial cost continues to decline, they may become the preferred solution to a variety of vibration control problems.

2.2.1: Adaptive Transversal Filters

Since the characteristics of the acoustic noise source and the environment are time varying, the frequency content, amplitude, phase, and sound velocity of the undesired noise are non stationary. An ANC system must therefore be adaptive in order to cope with these variations. Adaptive filters adjust their coefficients to minimize an error signal and can be realized as (transversal) finite impulse response FIR, (recursive) infinite impulse response IIR, lattice, and transform-domain filters. The most common form of adaptive filter and the one who will be used in this project is the transversal filter using the least-mean-square (LMS) algorithm.

The LMS algorithm is relatively simple to design and implement, being summarized as follows:

1. Choose parameters and initial conditions: L, μ, w(0) and γ; where L is the order of the filter, μ is the step size, w(0) is the initial weight vector at time n = 0, and γ is the leakage factor. The leakage factor is used to reduce numeric error in the finite-precision implementation [1] and $0<\gamma<1$.

2. Compute adaptive filter output

$$y(n) = \sum_{l=0}^{L-1} w_l(n).x(n-l)$$

3. Compute error signal

$$e(n) = d(n) - y(n)$$

4. Update adaptive weight vector from w(n) to w(n+1) using the LMS algorithm:

$$w_l(n+1) = w_l(n) + \mu . x(n-l) . e(n), \quad l = 0, 1, \ldots, L-1$$

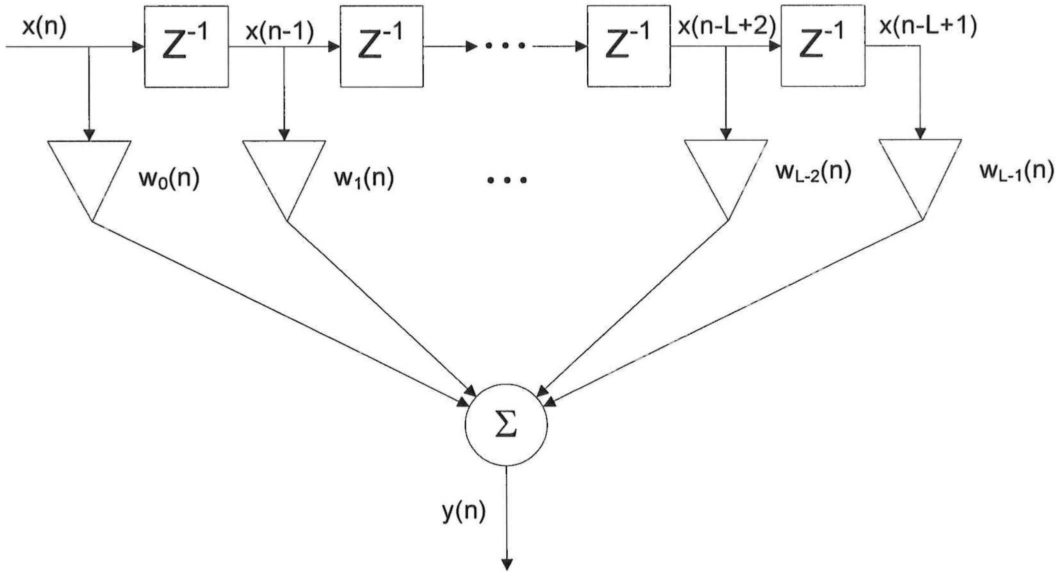

Figure 2.3 – Transversal structure of a FIR filter.

2.2.2: Feedforward AVC Systems

In the feedforward AVC system, the undesired vibration from a primary vibration source is used to drive a secondary source, such as an actuator, to cancel the vibration. The input signal from the reference sensor must be well correlated with the noise from the primary source.

The basic AVC system is described in an adaptive system identification framework illustrated in Figure 2.4, in which an adaptive filter is used to estimate an unknown plant. The primary path consists of the acoustic response from the reference sensor to the error sensor where the noise attenuation is to be realized. If the plant is dynamic, the adaptive algorithm then has the task of continuously tracking time variations of the plant dynamics. The most important difference between Figure 2.4 and the traditional system identification scheme is the use of an acoustic summing junction between d(n) and y(n), instead of the subtraction of electrical signals. However, for consistency we will continue to represent the summing junction by a subtraction; it is really arbitrary anyway because it can be implemented by a sign change of the secondary signal. In this project, the objective of the adaptive filter is to minimize the residual error signal e(n), so that y(n) ≡ d(n).

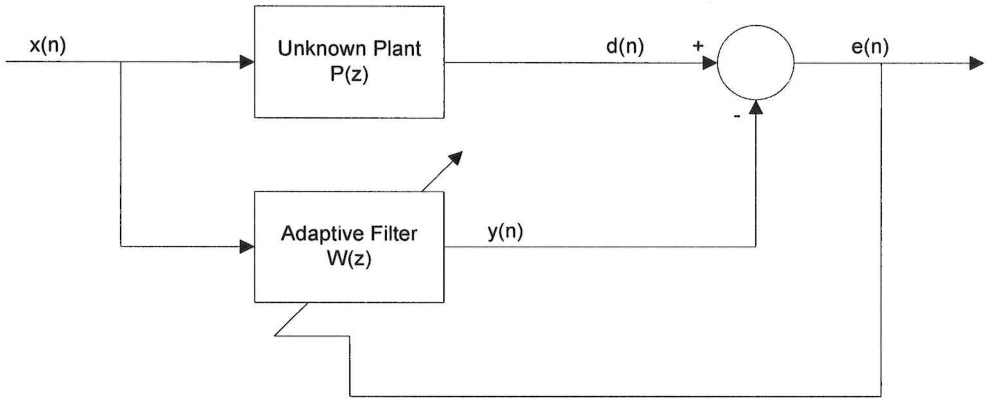

Figure 2.4 - System viewpoint of Feedforward AVC.

2.2.3: Secondary Path effects

The layout of the adaptive filter for AVC has to be extended to be useful in the real ANC systems. This is because it is necessary to compensate the effects of the secondary-path S(z) from y(n) to e(n), which includes the digital-to-analog (D/A) converter, power amplifier, actuator, physical path from actuator to error sensor, error sensor, preamplifier and analog-to-digital (A/D) converter. In general, this secondary path gives a change in amplitude and a phase shift.

For analysis purpose, we shall represent the actual system in Figure 2.4 by the block diagram of Figure 2.5.

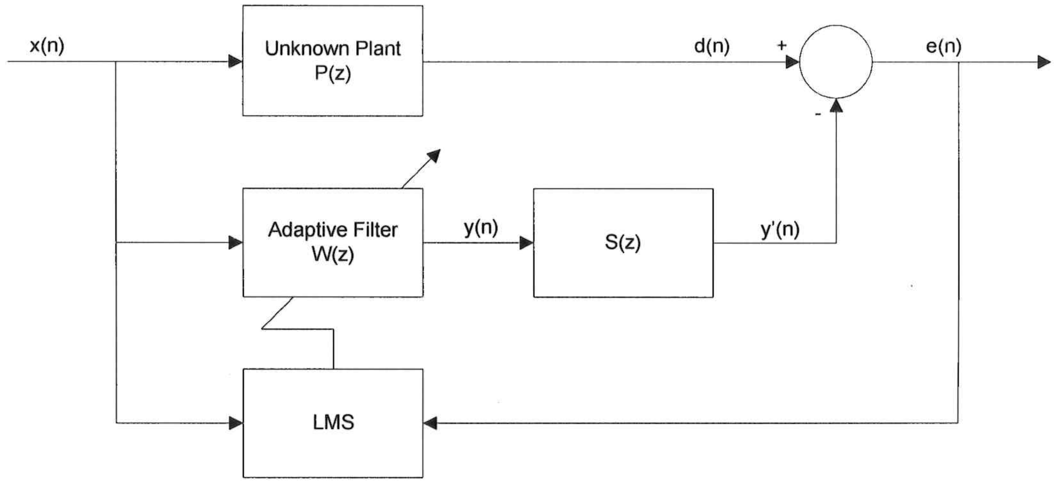

Figure 2.5 - Simplified block diagram of AVC System.

From Figure 2.5, the error signal is

$$E(z) = [P(z) - S(z).W(z)].X(z)$$

The introduction of the secondary-path transfer function into a controller using the standard LMS algorithm shown in Figure 2.5 will generally cause instability [7]. This is because the error signal is not correctly "aligned" in time with the reference signal, due to the presence of S(z). There are a number of possible schemes that can be used to compensate for the effect of S(z). Morgan [8] suggested two approaches to solving this problem. The first solution is to place an inverse filter, 1/S(z), in series with S(z) to remove its effect. The second solution is to place an identical filter in the reference signal path to the weight update of the LMS algorithm, which realizes the so-called filtered-X LMS (FXLMS) algorithm. Since an inverse does not necessarily exist for S(z), the FXLMS algorithm is generally the most effective approach.

The block diagram of ANC system using the FXLMS algorithm is illustrated in Figure 2.6.

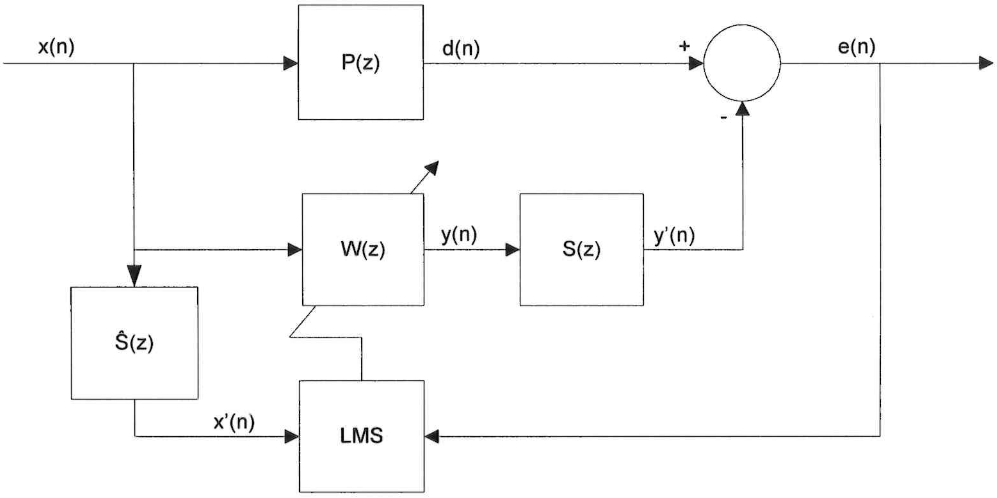

Figure 2.6 - Block diagram of AVC system using the FXLMS algorithm.

With FXLMS, the filtered reference is generated by passing the reference signal through this estimate of the secondary path:

$$x'(n) = \sum_{i=0}^{L-1} \hat{s}_i(n).x(n-i)$$

And the new weight update equation is:

$$w(n) = w(n-1) + \mu.x'(n).e(n)$$

The FXLMS algorithm appears to be very tolerant of errors made in the estimations of S(z) by the filter Ŝ (z). As shown by Morgan [8], within the limit of slow adaptation, the algorithm will converge with nearly 90° of phase error between Ŝ(z) and S(z).

2.2.4: Feedback AVC Systems

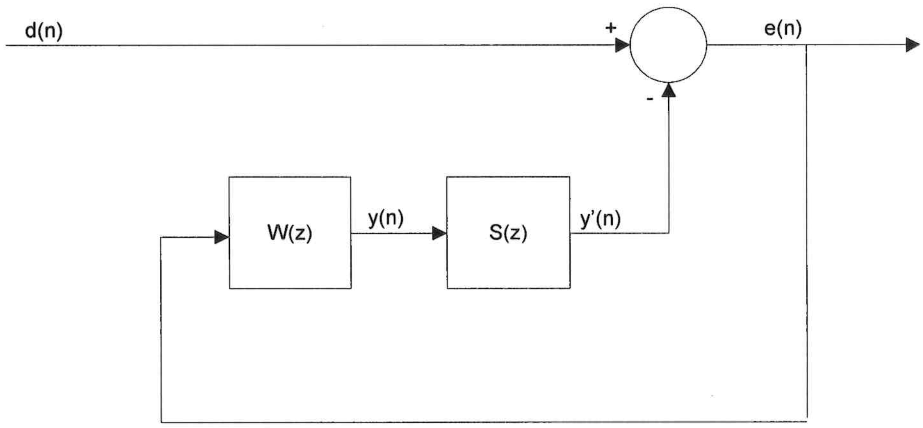

Figure 2.7 – Block diagram of Feedback AVC system

The basic idea of an adaptive feedback AVC is to estimate the undesirable vibration and use it as a reference signal x(n) for the AVC filter. In Figure 2.7, the primary vibration is expressed as D(z) = E(z) + S(z).Y(z), where E(z) is the signal obtained from the error sensor and Y(z) is the secondary signal generated by the adaptive filter. If Ŝ = S(z), we can estimate the primary noise d(n) and use this as a synthesized reference signal x(n). That is

$$X(z) \equiv \hat{D}(z) = E(z) + \hat{S}(z).Y(z)$$

The complete adaptive feedback AVC system using the FXLMS algorithm is illustrated in Figure 2.8, where Ŝ (z) is also required to compensate for the secondary path.

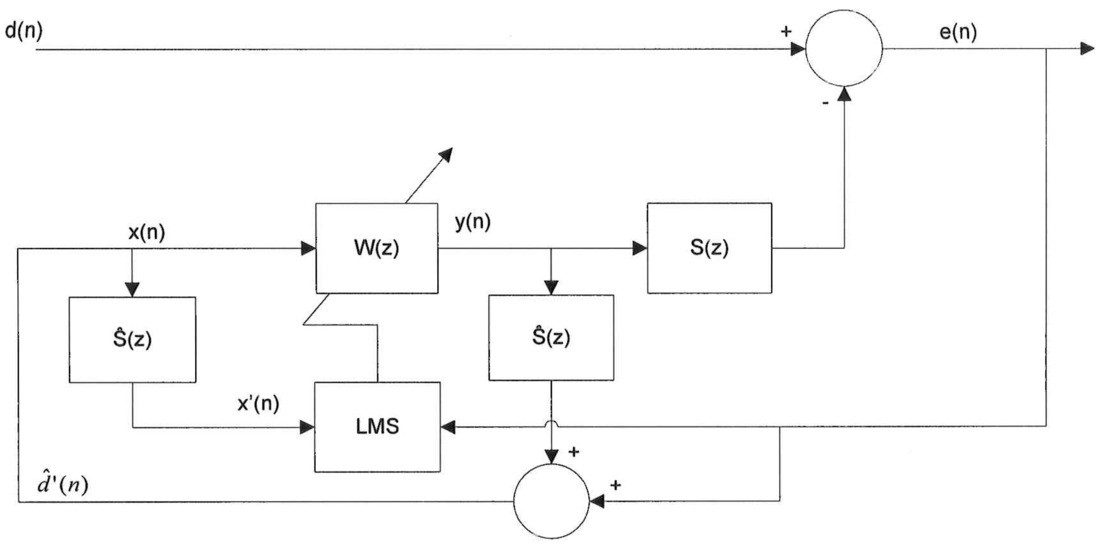

Figure 2.8 - Adaptive Feedback AVC using FXLMS

The reference signal x(n) is synthesized as

$$x(n) \equiv \hat{d}(n) = e(n) + \sum_{m=0}^{M-1} \hat{s}_m . y(n-m)$$

2.2.5: Hybrid AVC Systems

The feedforward ANC systems discussed use two sensors: a reference sensor and an error sensor. The reference sensor measures the primary noise to be canceled while the error sensor monitors the performance of the ANC system. The adaptive feedback ANC system uses only an error sensor and cancels only the predictable noise components of the primary noise. A combination of the feedforward and feedback control structures is called a hybrid AVC system.

The hybrid AVC system using the FIR feedforward AVC and the adaptive feedback AVC is illustrated in Figure 2.9, where the secondary signal y(n) is generated using the outputs of both feedforward AVC filter A(z) and feedback AVC filter C(z). The combined controller W(z) has two reference inputs: x(n) from the reference sensor and d(n), the estimated primary signal. The filtered versions of the reference signals x'(n) and d'(n) are used to adapt the coefficients of the filters A(z) and C(z), respectively.

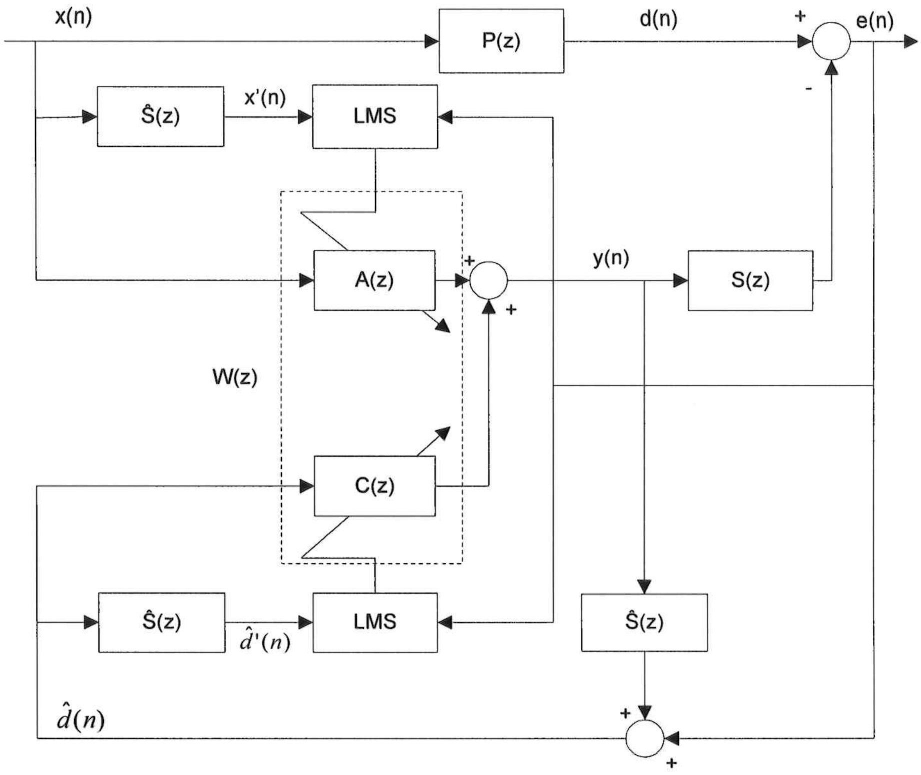

Figure 2.9 - Hybrid AVC system

The advantage of hybrid AVC over other AVC systems is that a lower order filter can be used to achieve the same performance. The hybrid systems also demonstrate an advantage over either the simple feedforward AVC or adaptive feedback AVC system alone when there is significant plant noise.

The disadvantage of this method is the computational complexity, once it is necessary to use two filters for each actuator, while the feedforward needs only one.

Chapter 3: Control Simulation in Matlab

In this project, the control simulation was made using Matlab Simulink. The algorithms codes were written in S-functions blocks, facilitating the user to define inputs and outputs and customize the code according to the need. The complete LMS algorithm written in S-function can be viewed in Appendix A.

During the entire project, two different signals were used to simulate the undesired signal: Sin Wave and Impulse (Figure 3.1). The impulse signal was chosen because its shape is more similar with the real undesired disturbance, making the simulations results closer to the real process. The sin wave was chosen because of its simplicity of simulate the narrow band frequencies.

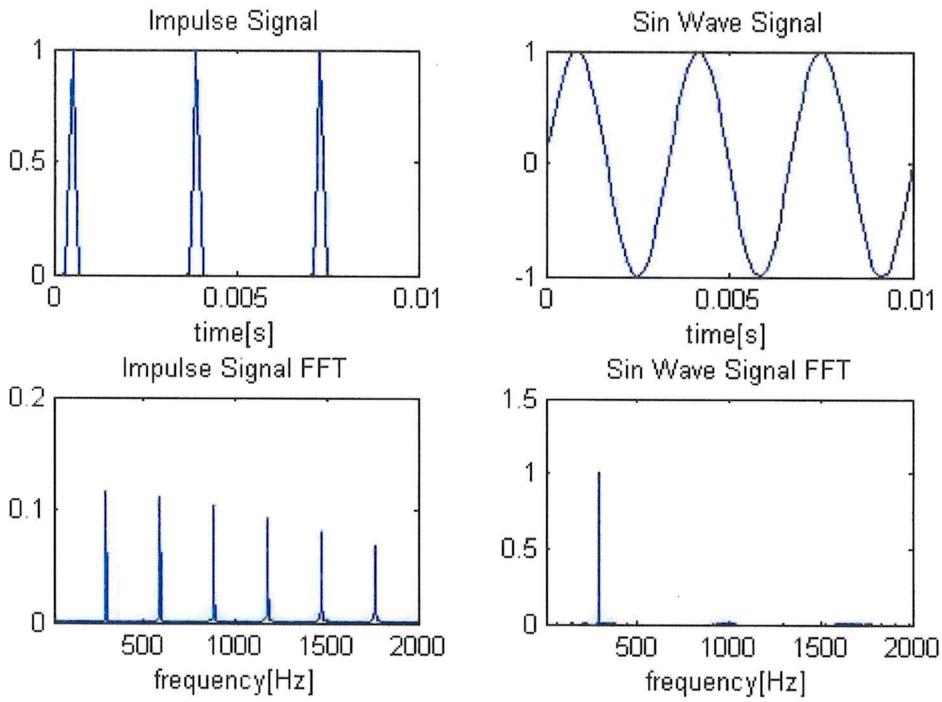

Figure 3.1 – Reference signals

The simulations were done with a 300 Hz frequency for both sine wave and impulse disturbance. Once P(z) and S(z) are unknown in real process, they will be denoted as P(z)= [0 0.9 0.5 0.3 0.1] and S(z)= [0.5 0.4 0.1]. The length of the primary filter order is 40 and the secondary filter order is 4. The value of these lengths were chosen by trial and error, starting with low values and increasing until the length reaches an optimal value. During the simulations, primary filters with size below 30-40 presented an instable control, and over 50 hadn't presented any significant change of the control performance. The same method was used to define the length of the secondary filter. The leakage factor was kept the same for all simulations, with the value of 1e-7.

The results are summarized in the next sections. Since the simulations with sine wave and impulse are very similar, for simplicity, it will be showed the results for impulse signal only for Feedforward AVC simulation. Simulations of Feedback and Hybrid AVC will present only the results for sin wave.

3.1: Feedforward AVC

3.1.1: Sine reference

The figure below shows the layout of Feedforward Control using a sin wave as a reference.

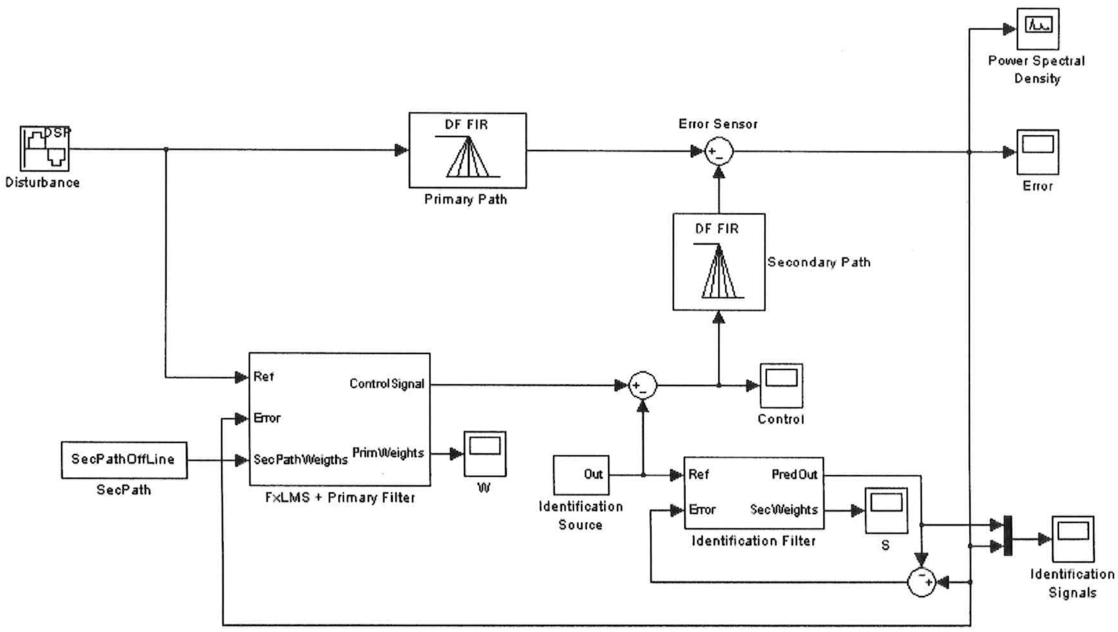

Figure 3.2 – Feedforward AVC layout with sin reference

It is easy to notice the two filters used in this simulation, one responsible for the identification of secondary path, and other responsible for the control. This is a simple scheme of vibration control, simulating the use of only one actuator.

The next figures shows the error and control signal of the simulation, and the weights evolution of secondary path identification. The step size μ used is 1e-3.

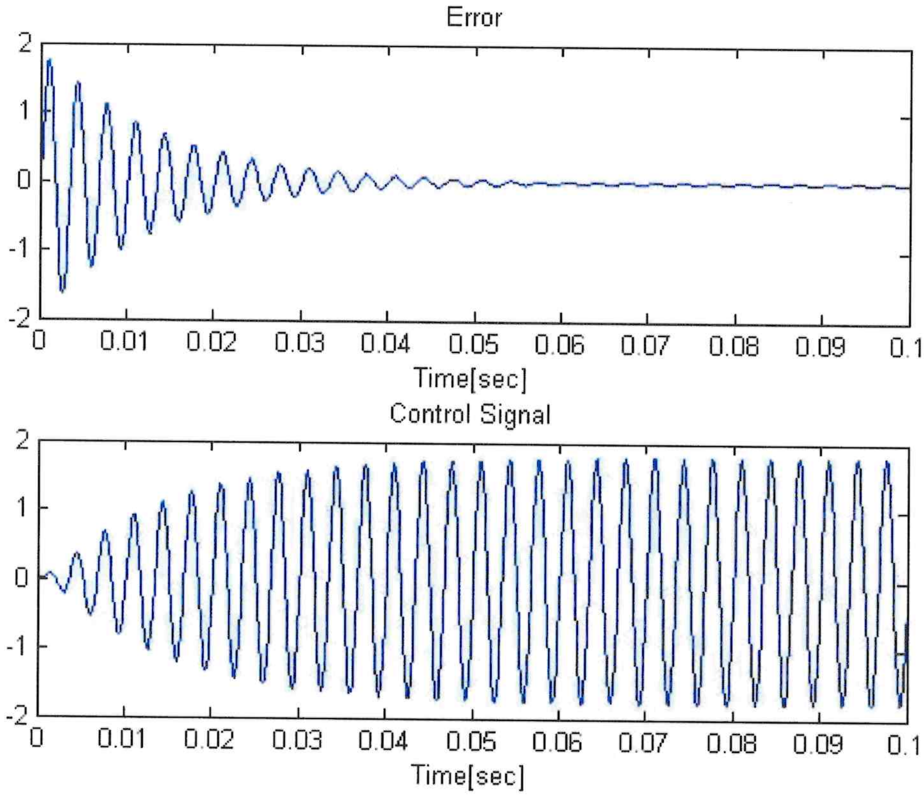

Figure 3.3 – Feedforward AVC simulation results

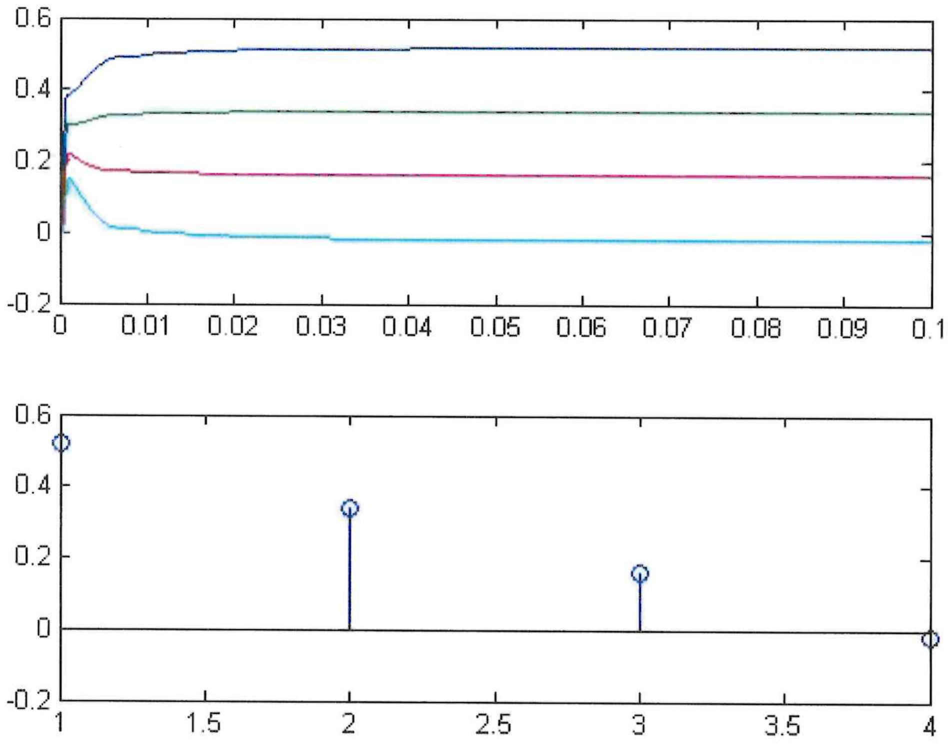

Figure 3.4 – Secondary Path Identification

3.1.2: Impulse reference

Using the same control strategy with sine wave reference, the Feedforward control with impulse signal reference is showed below, as well as the signals from error sensor, control and weights evolution. The step size in this simulation is 1e-1.

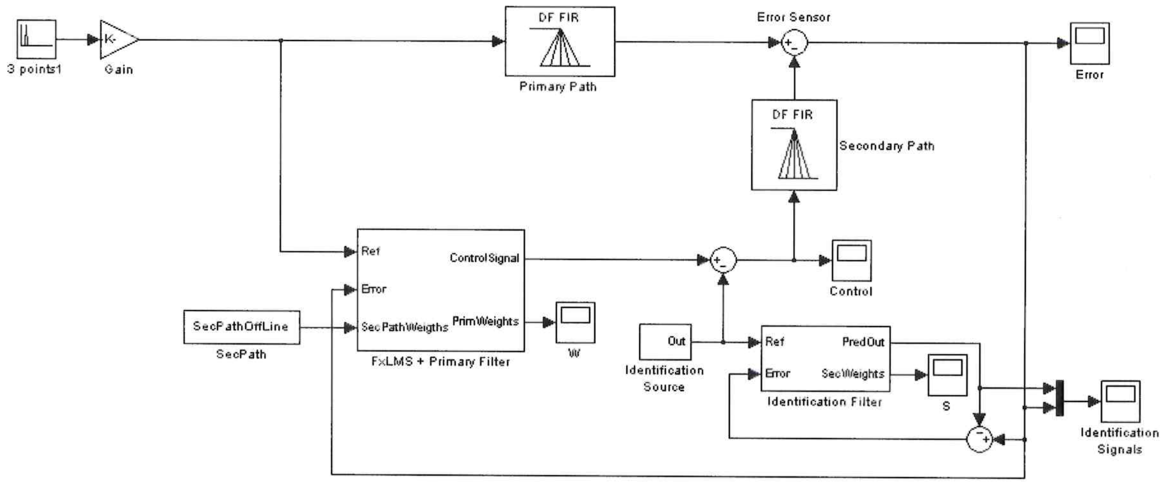

Figure 3.5 – Feedforward AVC layout with impulse reference

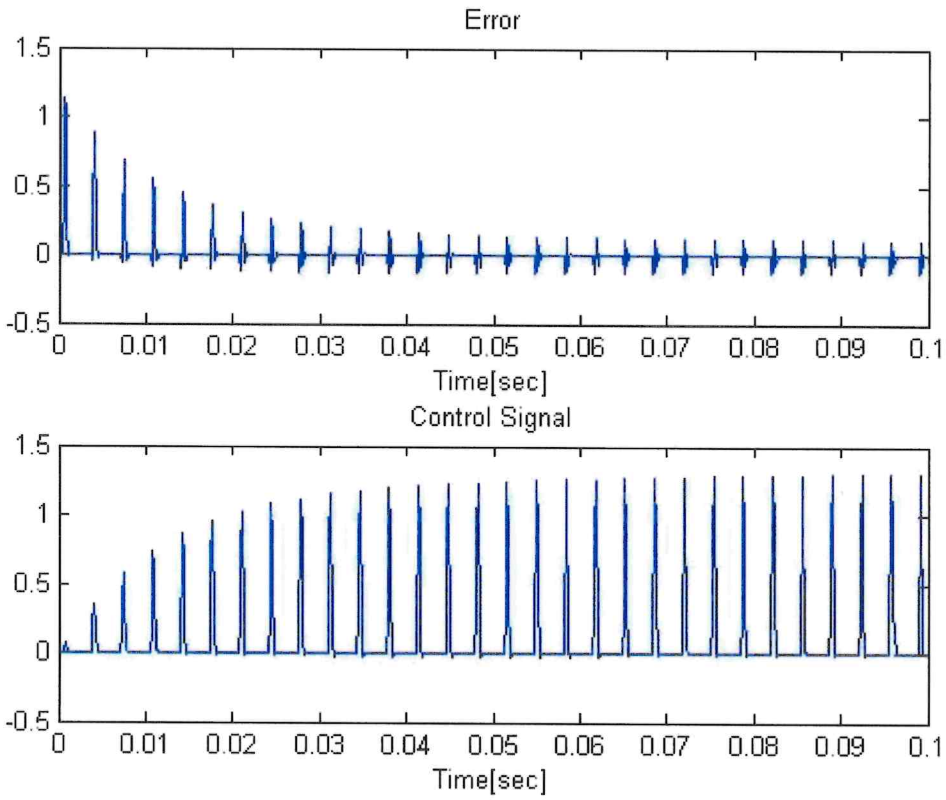

Figure 3.6 – Feedforward AVC simulation results

3.2: Feedback AVC

The layout of the Feedback AVC system is shown in Figure 3.7. It can be noticed that the reference signal is not being directly connected to the control algorithm, but it is being estimated by the error signal.

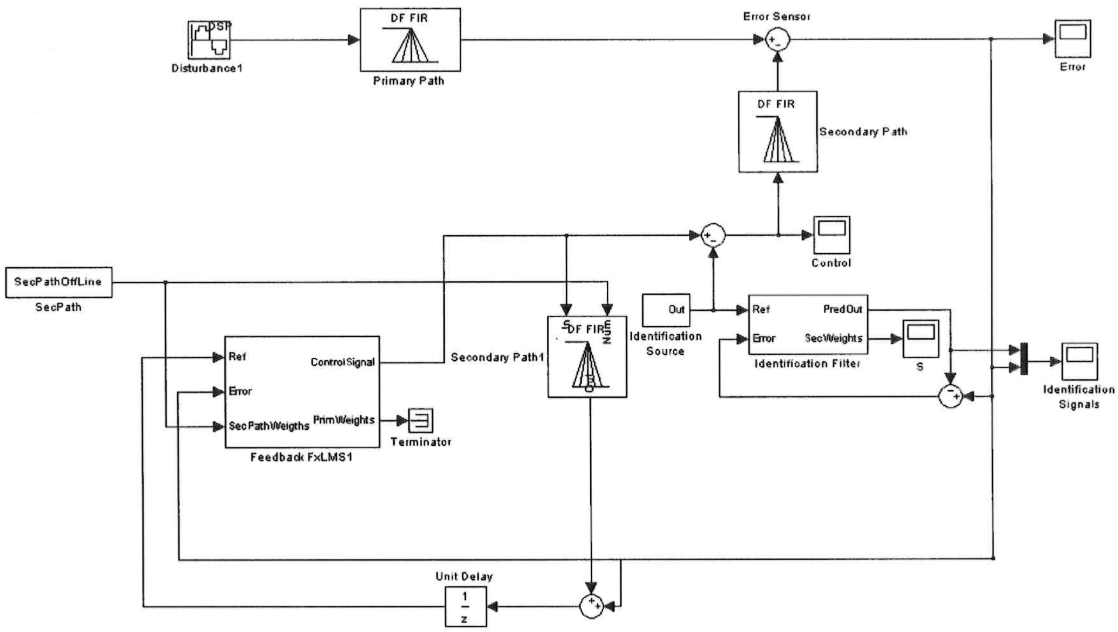

Figure 3.7 – Feedback AVC system layout

This feedback system layout designed in Simulink is different than the feedback layout viewed in figure 2.6. The difference comes from the fact that a delay operation had to be inserted in the block diagram, because the scheme contains an algebraic loop, which cannot be solved in implementation. Figure 3.8 shows the error evolution in feedback control simulation. The step size value is 5e-4.

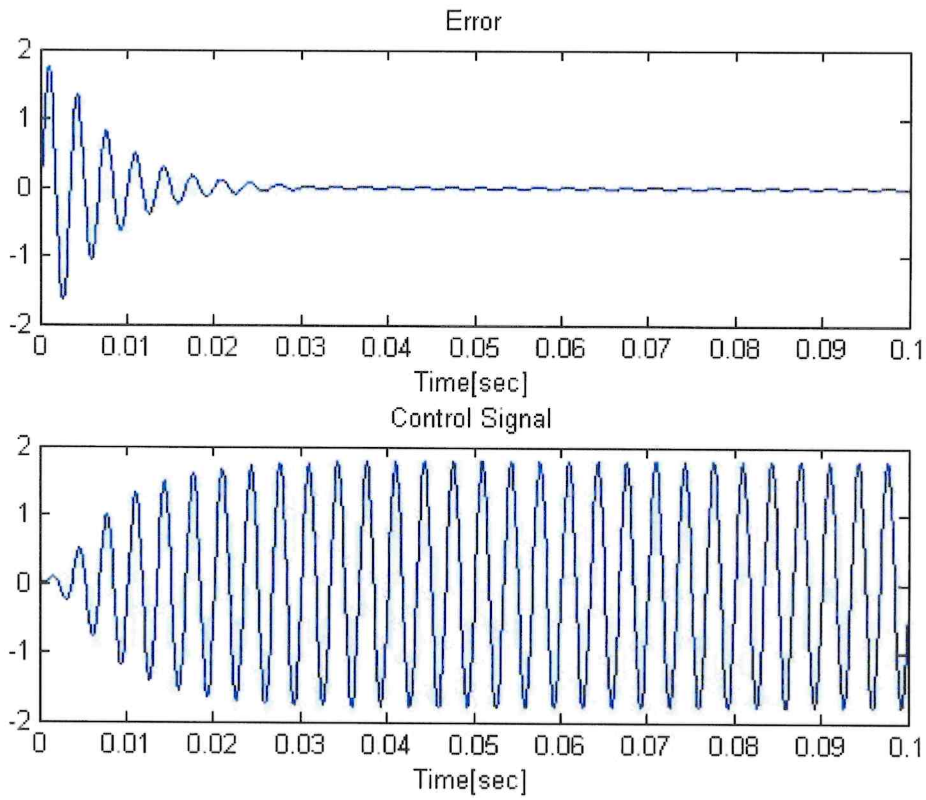

Figure 3.8 – Feedback AVC simulation results

3.3: Hybrid AVC

Figure 3.9 shows the Hybrid AVC system layout. Two filters are being used for control: the feedforward filter, which contains the reference signal directly capted by the primary source, and the feedback filter, which has the estimated reference signal from the error sensor. The error and control signals are viewed in Figure 3.10. The step size value for the feedforward filter is 5e-3 and the value for the feedback filter is 1e-5.

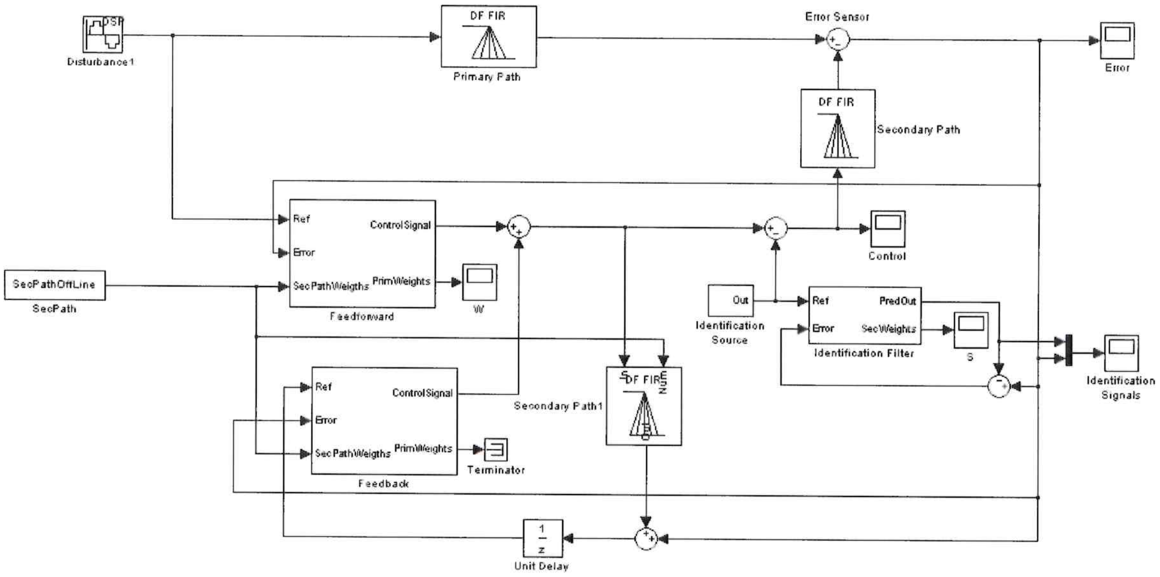

Figure 3.9 - Hybrid AVC system layout

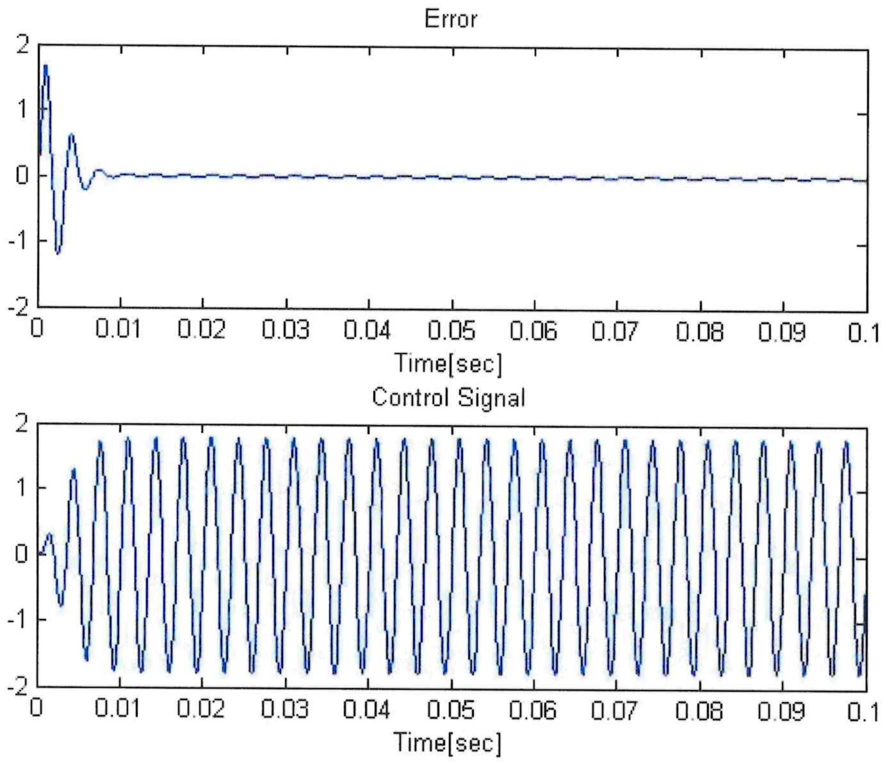

Figure 3.10 – Hybrid Feedback AVC simulation results

Analysing these simulations, all three control approaches (feedforward, feedback and hybrid) presented a satisfactory error reduction without control overload, for both reference signals tested, sin wave and impulse.

However, there are some differences between the three approaches, specially related to the convergence time, which is directly related to the step size μ and the complexity of the system. In this case the feedback AVC is using a bigger step size than the hybrid control, but smaller than the feedforward AVC. In this simulations in Matlab this difference between the parameters is not so relevant, however, in real applications it must be considered the physical limitations of the actuators to avoid the overload, once bigger step size needs a bigger control effort.

Chapter 4: Physical Experimental Set Up

This chapter will present the implementation of AVC system in the real process. First, a brief description of the machine is given, followed by an explanation about the concept of adaptronics. Finally, the graphical user interface (GUI) used in this project will be showed.

4.1: The Wood Machining System

The object of study for this project is an IMA QuadroForm C80/280 (Figure 4.1), a computer numerical control (CNC)-programmable stationary Wood machine system owned by the Institut für Werkzeugmaschinen und Fertigungstechnik (IWF). It is a 7-direction vertical mill with a manual XY table for workpiece positioning and a vertical spindle that moves along the XYZ-axis. In the process, some part of the wood is removed from the woodplate by using a rotating tool. This tool is moved in many different directions to achieve a desired shape. A variety of shapes are available as cutting tools and operations like cutting, planning and drilling are also possible.

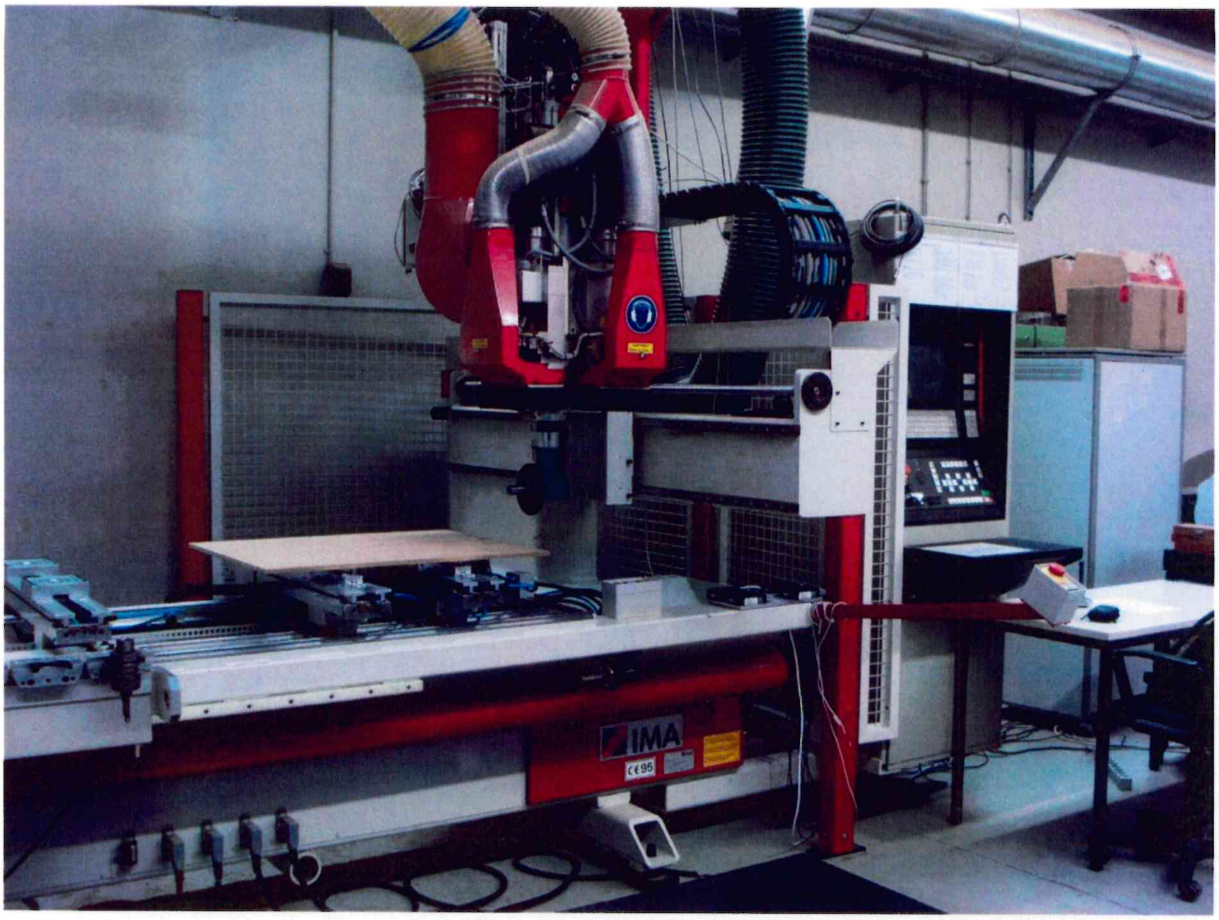

Figure 4.1 - The Stationary Wood-Machine System

The machine offers a working area in XYZ of 2800 x 800 x 75 mm and has a 7,5kW main spindle with a rotational speed of up to 6000 rpm [12]. An 8-position tool changer automatically replaces the tools while the workpiece is being produced. Then manual setup time is minimized and the machine can be operated unattended. A vacuum clamping system with capacity up to $40m^3/h$ is used for holding the woodplate to the machine table. Its advantage is the reduced setting-up time and be well suited for non-magnetic workpieces. Moreover, the machine is equipped with a reliable industrial PC in which the software IMAWOP 2.5 provides a user-friendly and extremely quick programming interface (Figure 4.2). To improve the security level a pressure sensing carpet is mounted near to the machine to prevent the user approximation during the process.

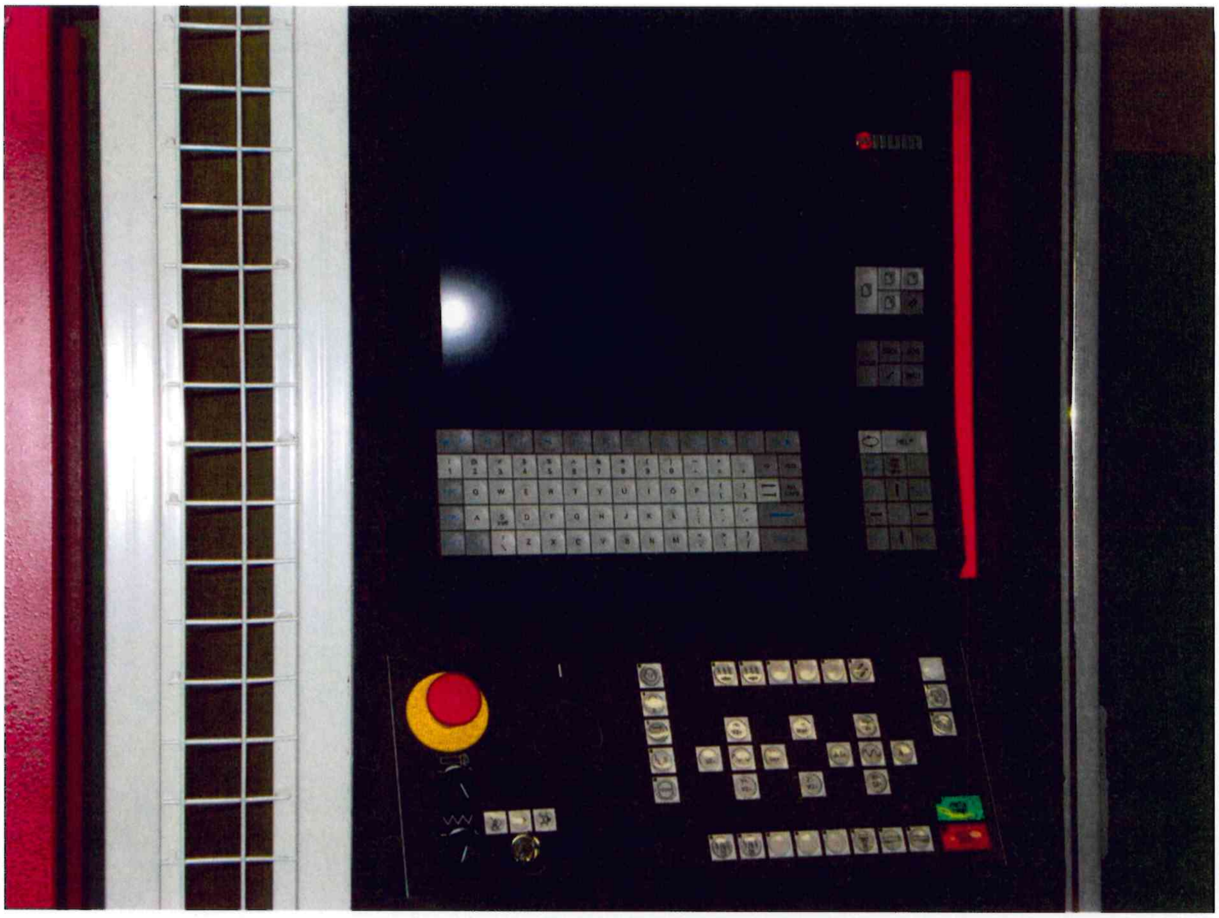

Figure 4.2 - Control Panel

4.2: Adaptronics for machine tools

The physical control system used in this project is the result of one research project about the methodological integration of adaptronical components into the suction blocks on the working table [6]. The adaptronic suction block with integrated actuator can be viewed in Figure 4.3.

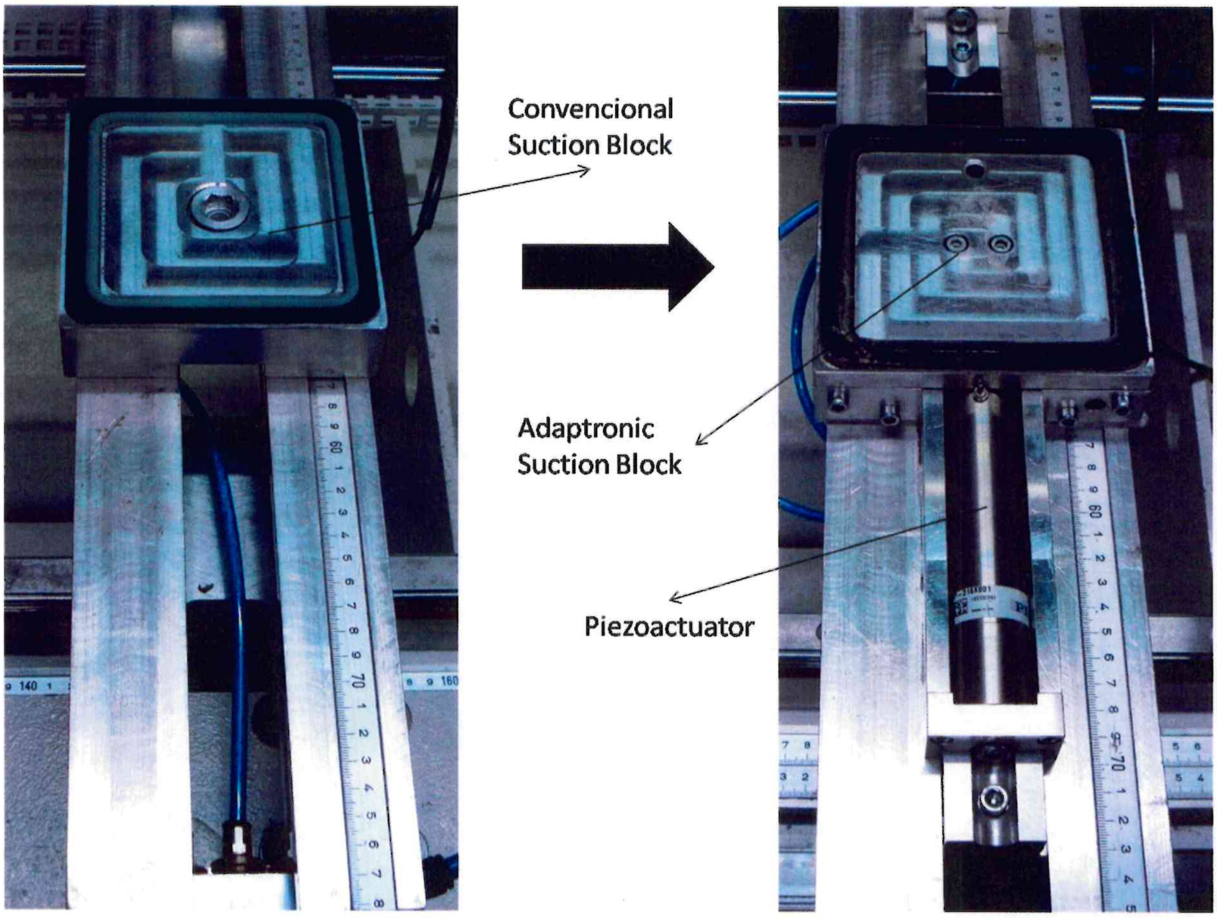

Figure 4.3 - Adaptronic suction block

One definition of an adaptronic system is "the integration of actuators and sensors in the structural components". Although acoustic sensors can also be integrated in the suction block with the actuators, some modal analyses were made to find the best position to place the error sensor. Because of that, the accelerometer used to monitor the vibration of the plate is placed in the middle of the workpiece surface. Figure 4.4 shows the position of the actuators and the error sensor in the plate. The next chapter will show the experiment layouts for shaker and real process.

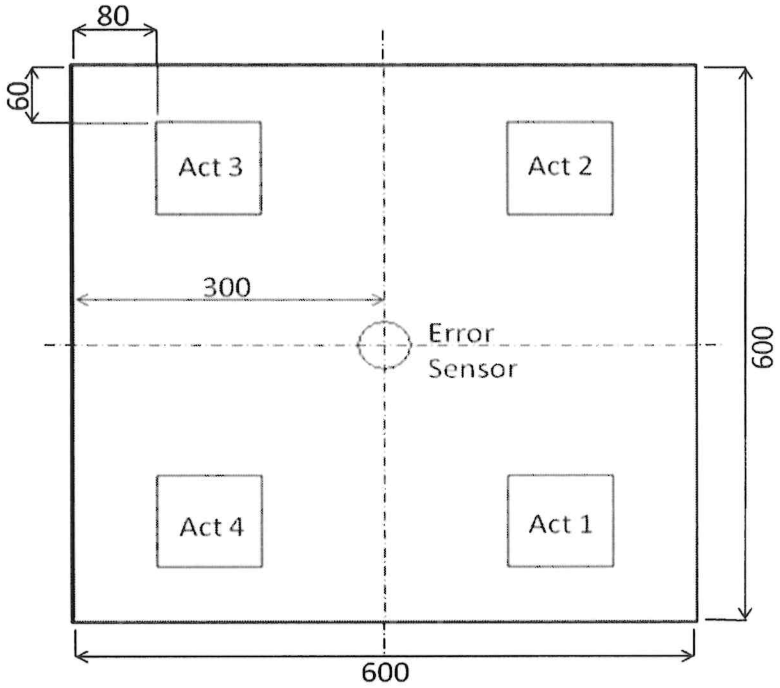

Figure 4.4 – Location of actuators and sensor on the plate surface. Measurements are expressed in millimeters.

4.3: Graphical User Interface

Because of a large number of real time data to be processed in the algorithm and large number of adjustable parameters for the system, a Dspace graphical user interface (GUI) were used to facilitate data monitoring, parameter modification and process control.

The GUI developed in this project is responsible for downloading, initializing and stopping the algorithm, as well as modifying the parameters of the system. Basically, the GUI is composed by two windows: the main window and the identification window.

The main window is composed by a capture settings panel, control bars, numeric inputs and graphics plotter. Thus, controller parameters like step size and forgetting factor can be modified by one click in the numeric box input, and the control can be activated and deactivated also by one click.

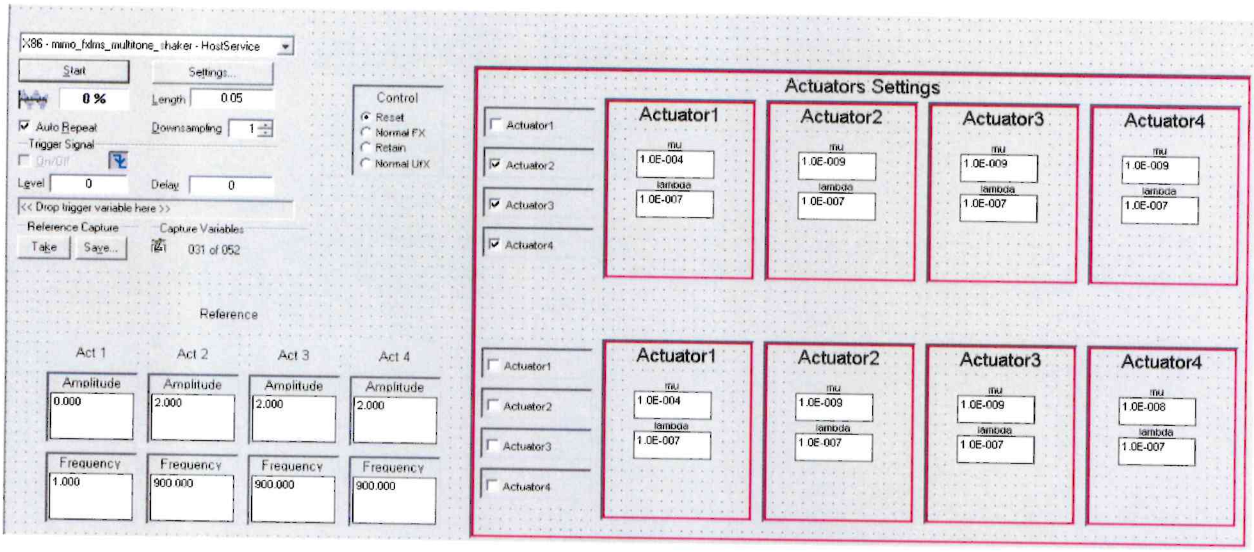

Figure 4.5 – Part of the main window

The identification window is responsible for the estimation of the secondary paths S(z) of the system. The number of identification windows vary according to the number of actuators. Any identification source parameters, such as amplitude and frequency of sinusoidal wave and variance for the white noise can be chosen in this window.

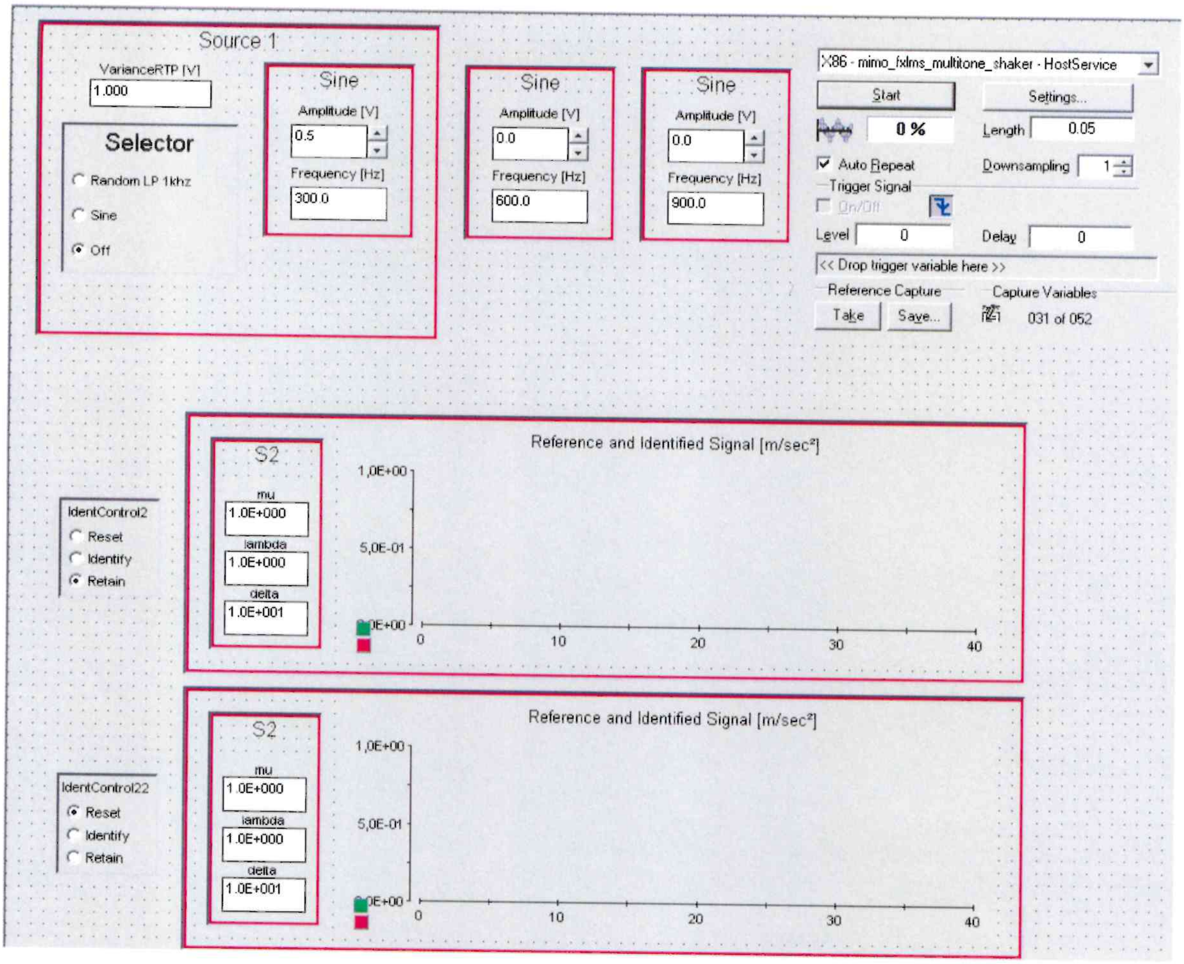

Figure 4.6 – Identification window

Chapter 5: Experimental Results with AVC

The goal of this project is to obtain a good noise reduction in the frequency range until 2000 Hz. The initial idea was to use the Broadband Active Vibration Control, but, because of the behavior of the tool's noise, one different approach was used.

During machining process, the error sensor captured clearly the noises of the tool in specific ranges of frequency: the fundamental frequency, which is the frequency of the tool operation, and its harmonics (Figure 5.1). Due to this behavior, for the reduction of noise in a frequency range until 2000 Hz, the error sensors filters were replaced: instead of a single lowpass filter, it was used six bandpass filters with small range, each one containing the frequencies to reduce (the fundamental 300 Hz and its five first harmonics). Thus, each actuator was responsible to control one, at most two range of frequency.

With this new proposal of control, a lot of alternatives were tested concerning which frequency should be controlled by each actuator. Another important factor mentioned previously is the use of two virtual reference signals, sine wave and impulse. The use of the virtual signals is necessary because in the real process it is not possible to obtain the real disturbance signal, and the impulse and sin wave signals showed to have a good correlation with the real reference.

In order to verify the control algorithm and the configurations proposed for vibration reduction, this chapter presents the results of practical experiments made, first with shaker and then in real process, and finally will be discussed about the vibration control approach with the best noise reduction.

5.1: Experiment using shaker

In the first experiment, a shaker was used to simulate the undesired vibration (figure 5.1). The use of the shaker was necessary to validate the control algorithms before testing in the real process. Unlike the real process, the experiment with shaker is safe and simple, and no extra precautions was necessary, but the results were still relevant to study the performance of the control.

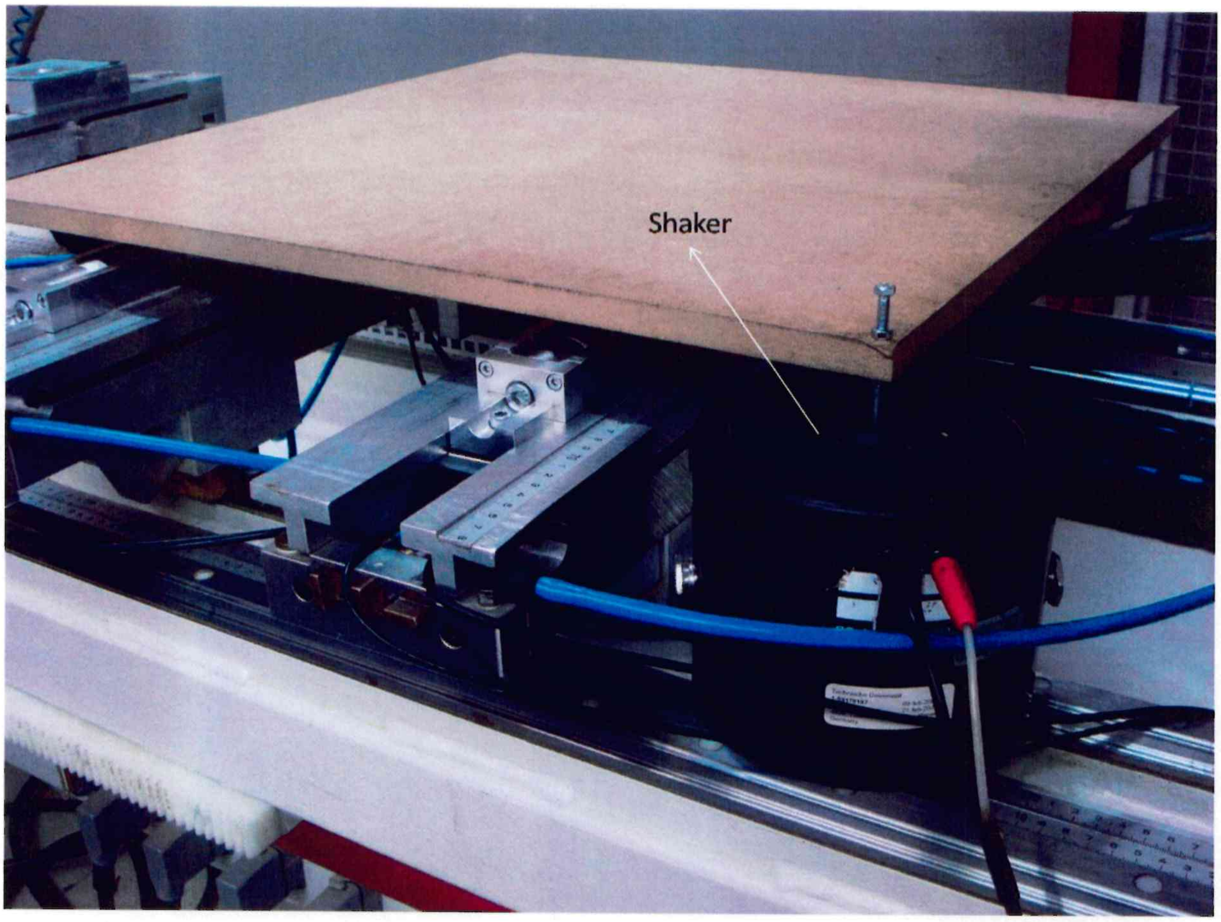

Figure 5.1 - System using shaker

For all of the experiments, were obtained the signals from the accelerometer and the microphone, and the control signal from the actuators. The noise signal from the microphone, as mentioned previously, will be measured only with the purpose of analysis of the noise reduction, therefore not being part of the control system. The microphone signal will be captured by the software of noise and vibration analyser OROS.

The system layout with the shaker can be viewed in the figure below.

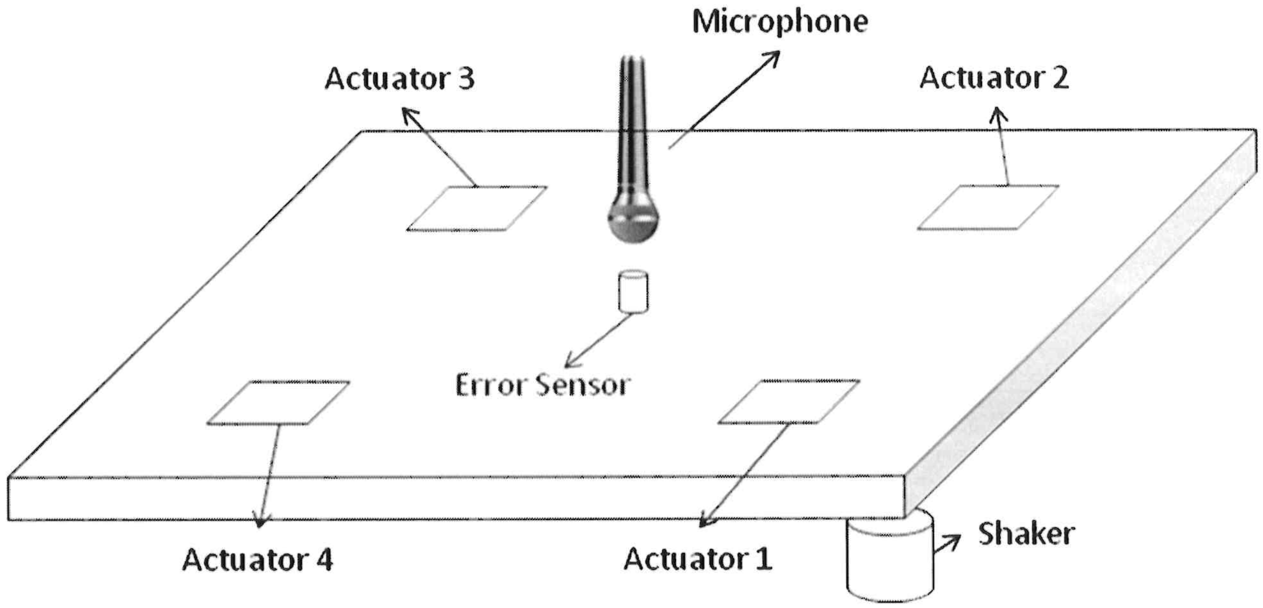

Figure 5.2 - System layout with shaker

The default amplitude of the disturbance signal of all experiments is 1 Volt. This default amplitude is important for the results, because all the experiments have the same conditions, facilitating the choice of the best control strategy. The value of the leakage factor is 1e-7, the same for all experiments too.

5.1.1: Feedforward Control

5.1.1.1: Sine wave as reference

For this first experiment with sin wave as reference, the goal was to achieve the vibration reduction only for the fundamental frequency of disturbance. Thus, for the sin wave signal, was chosen the frequency of 300 Hz, the same rotation frequency of the milling tool. Some different measurements were made using this configuration, and the results can be viewed in the table and figure below.

Exp.	Actuator(s) used	Step size μ	Actuator consumption (V)	Vibration Reduction (%)	Noise Reduction (dB)
1	1	5e-8	1.0	82	22
2	2	5e-8	1.8	82	25
3	3	5e-8	1.5	65	04
4	4	5e-8	1.8	71	08
5	1+3	5e-8, 5e-8	0.9, 0.2	78	22
6	1+2+3	5e-8, 5e-8, 5e-8	0.7, 0.6, 0.2	82	23
7	1+2+3+4	5e-8, 5e-8, 5e-8, 5e-8	0.4, 0.4, 0.1, 0.5	78	17

Table 5.1 – Experiments of AVC using shaker and sin wave reference

Analyzing the table, it can be easily seen that each one presents different results. One important fact to consider is that all measurements had the same value of the step size and the reference gain. Besides, the position of the actuators and the sensor always remained the same.

One reason for this difference between the results is in the relative position between the actuators and the disturbance source (shaker). When the actuator is closer to the undesirable vibration, the performance of the control shows better results, in this case making measurements 1 and 2 have better results than measurements 3 and 4. Now analyzing measurements 1 and 2, they also presented some differences in noise reduction. Even they having practically the same relative position from the shaker, this difference of noise reduction now can be explained by the performance of each actuator, which can vary although they have the same model. The same conclusion can be viewed between measurements 3 and 4. One behavior responsible for this difference of performance analyzed in this project is the hysteresis.

Another important factor to be considered is that the vibration reduction is not necessarily proportional with the noise reduction. Measurements 4 and 5 can explain better this phenomenon: although the difference between the vibration reduction is not so big (71% and 78%), the difference between noise reduction is considerably (08 and 22 dB).

The figures below show the error signal in time and frequency domain, as well as the control signal. They were taken from experiment 7, which although it didn't have the best results for noise and vibration control, it was the one with the smallest consumption of energy by the actuators.

Without Control

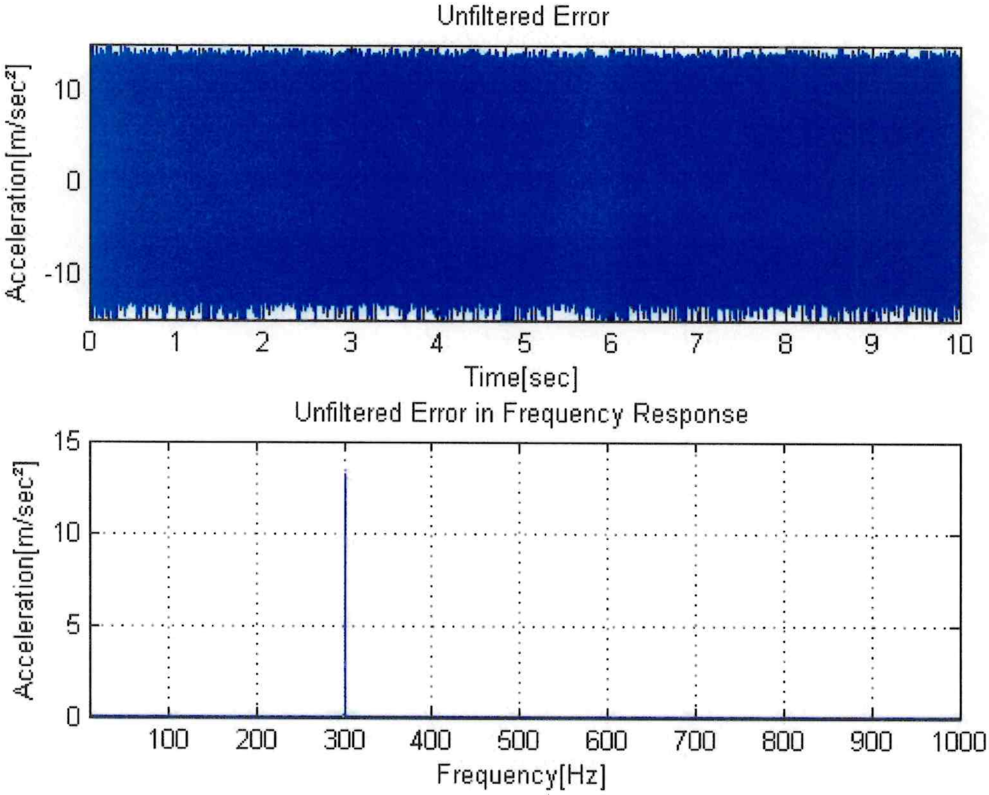

Figure 5.3 - Error signal without control

With Control

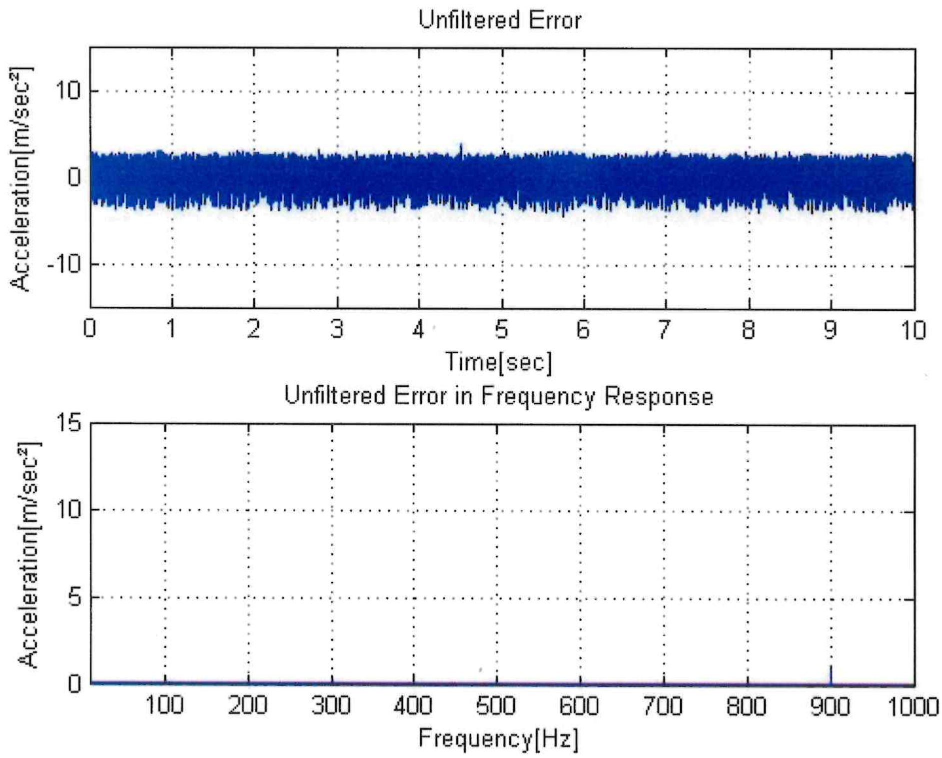

Figure 5.4 - Error signal during control

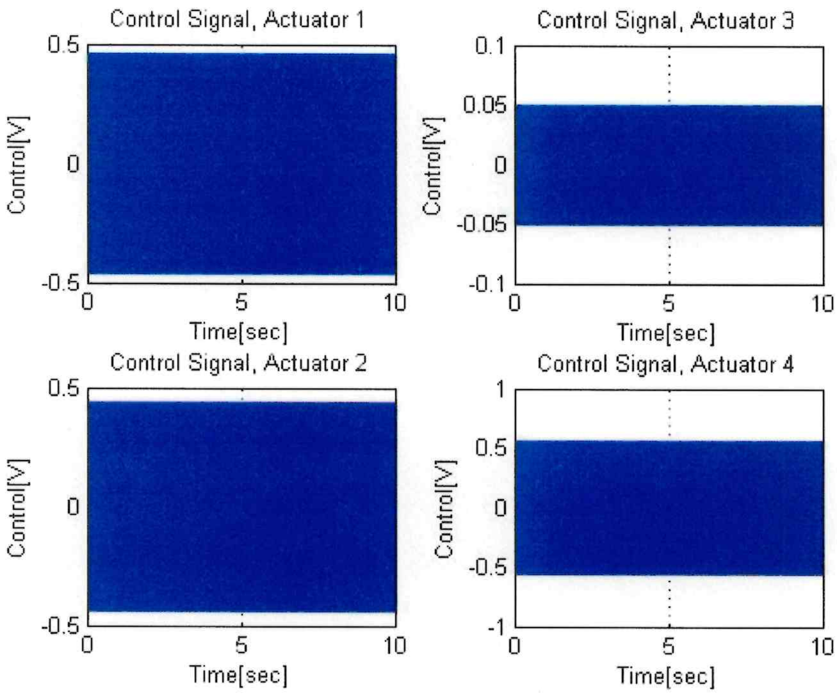

Figure 5.5 - Control signal

Analyzing the control graphics, the vibration control presents a very satisfactory results, eliminating almost completely the vibration source, although the error signal in frequency domain showed an increase of high frequency vibration produced by the actuators. This high frequency noise can also be viewed in the microphone sensor. Figure 5.6 shows the noise reduction in frequency domain, and figure 5.7 shows the noise reduction in 300 Hz during simulation.

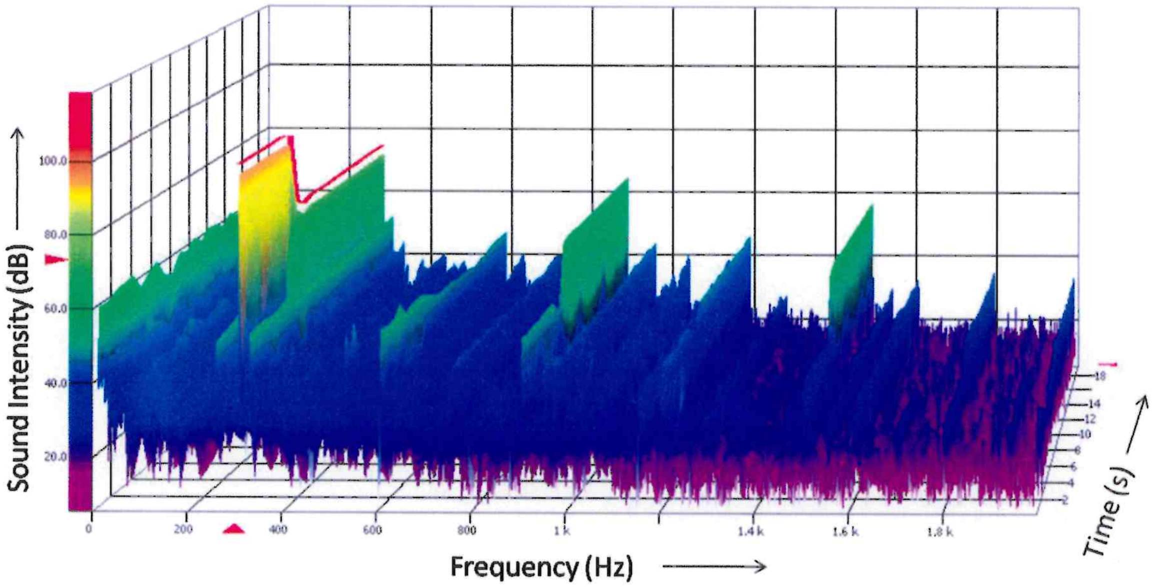

Figure 5.6 - Noise evolution capted by microphone

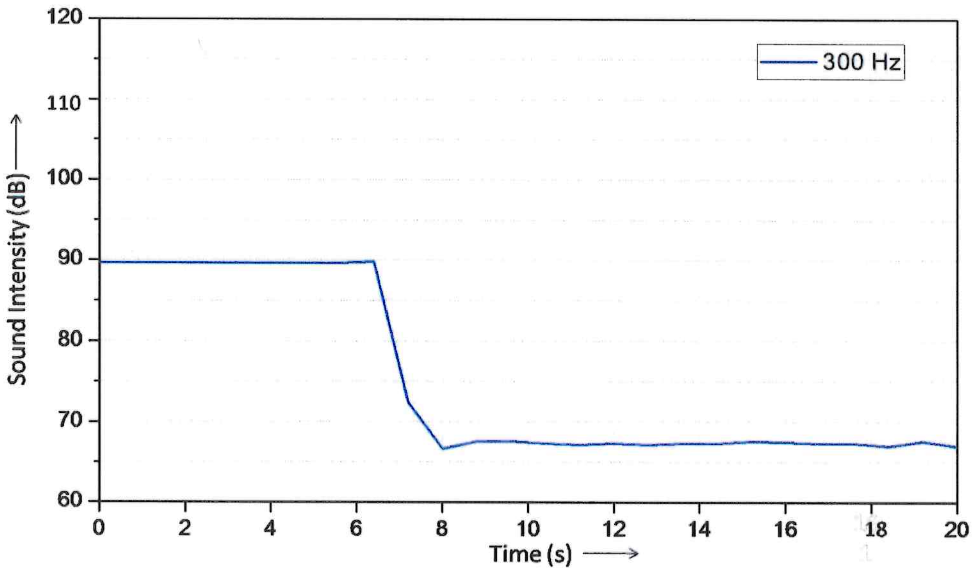

Figure 5.7 – Noise evolution filtered at 300Hz

The measurements from microphone show a noise reduction of over 20 dB in the frequency of 300 Hz. The overall noise reduction is about 17 db, proving that the control is efficient.

5.1.1.2: Impulse as reference

To approach the process disturbance, the reference signal was changed to impulse. Because of technical problems in one actuator, only three actuators were used to control the vibration in this experiment. Figure 5.8 shows the error signal from the accelerometer without control. One interesting observation regarding the impulse signal is that, although the signal amplitude was kept as 1 V, the error signal in frequency domain became different than the impulse signal seen in figure 3.1. This happens because it was used arbitrary secondary path weights in the simulation, which were different than the real secondary path.

Without Control

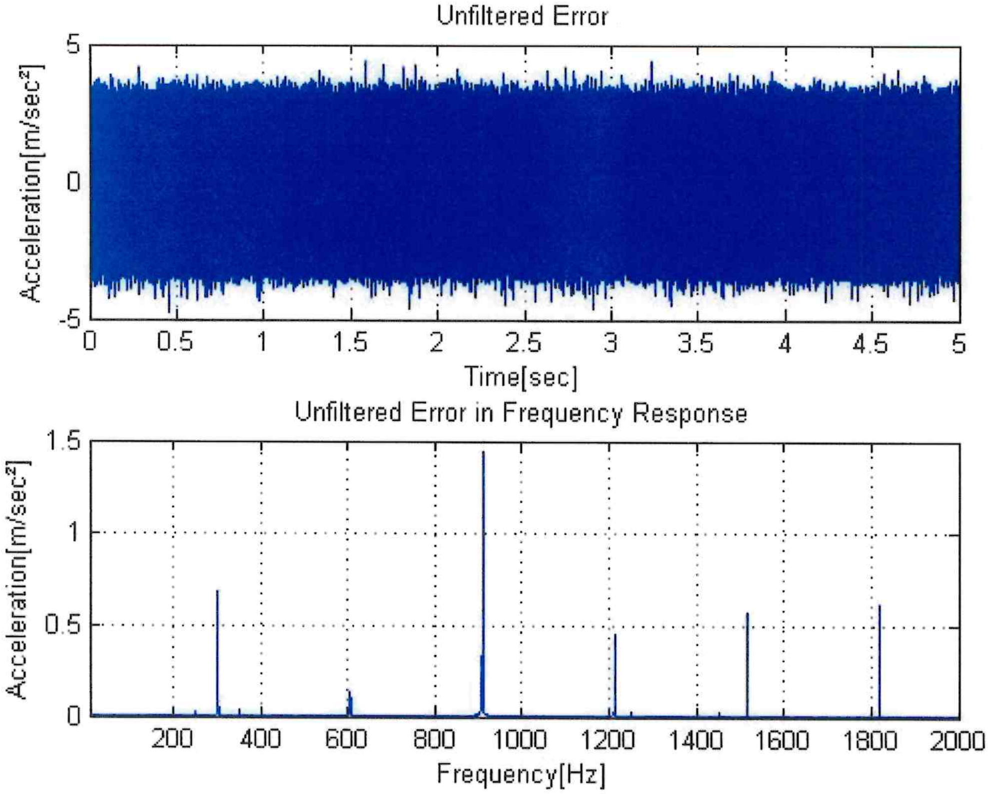

Figure 5.8 - Error signal without control

Analysing the figure with the error in frequency domain, it can be seen the vibration disturbances in different frequencies. Because of that, unlike the control approach used previously with sin wave, the control with impulse reference will try to reduce all low frequencies disturbances.

The next figures show the error and control signal during control. The step size values used in this experiment is showed in the table below.

Actuator	Control filter	Step size value
2	600 Hz	1e-5
	1500 Hz	1e-6
3	300 Hz	1e-5
	1200 Hz	1e-6
4	300 Hz	1e-5
	900 Hz	1e-6

Table 5.2 – Step size values

With Control

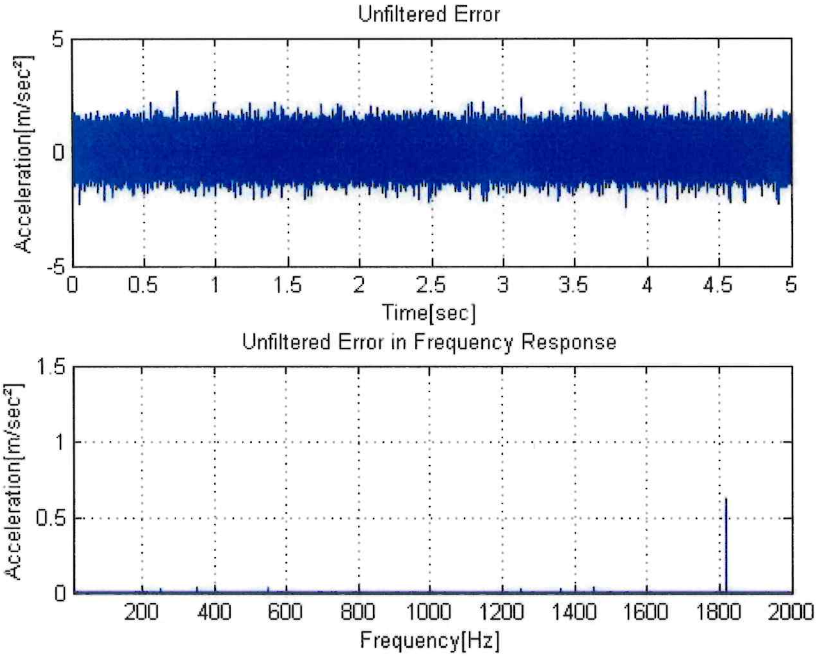

Figure 5.9 - Error signal with control

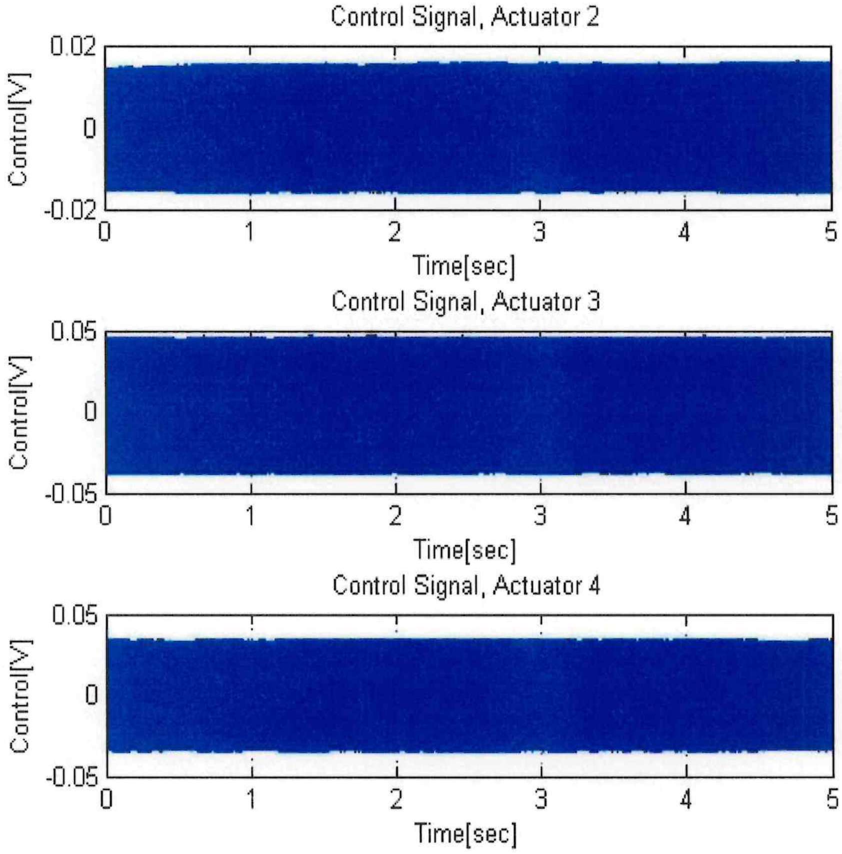

Figure 5.10 - Control signal in time domain

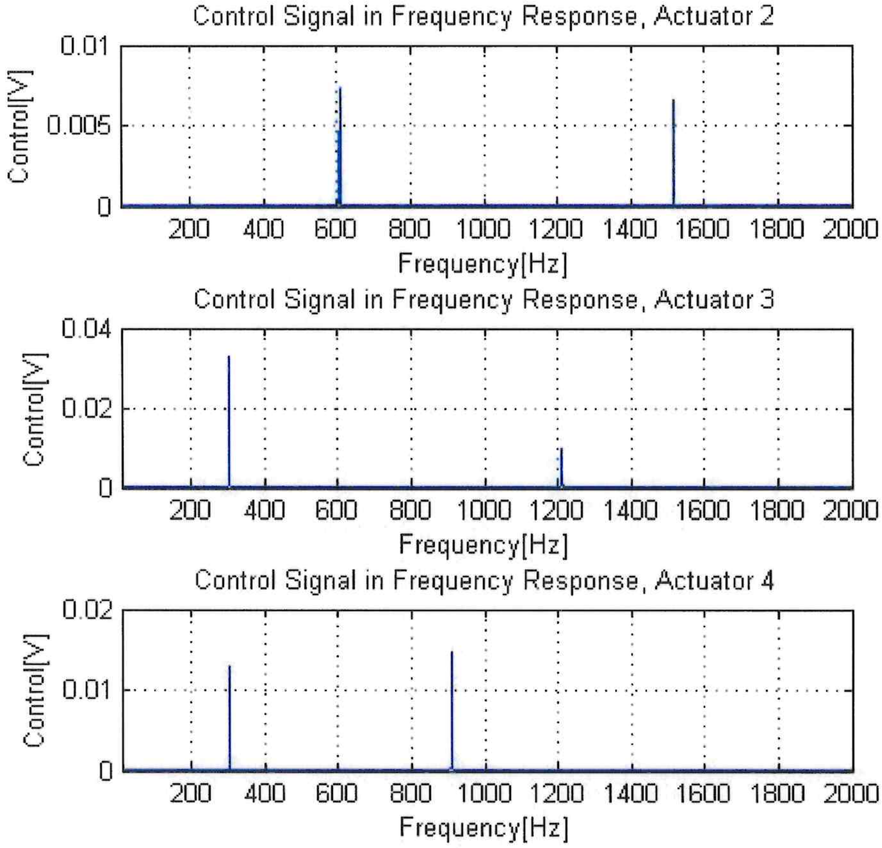

Figure 5.11 - Control signal in frequency domain

The experiments with Impulse reference presented a better control compared to the control with sine wave reference. In this experiment, vibration reduction in all frequencies controlled were achieved (300, 600, 900, 1200 and 1500 Hz), and the high frequency vibrations didn't appear during control. Besides, the control signal is significant smaller than the experiment using sin wave. Figures 5.12 and 5.13 show the noise reduction measured by microphone in frequency domain.

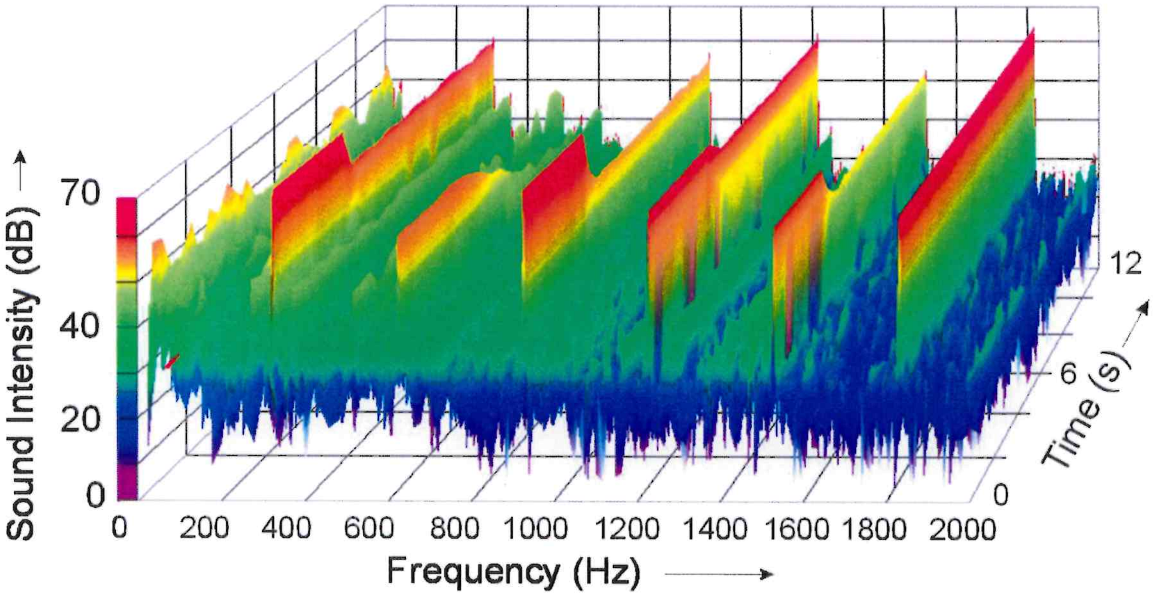

Figure 5.12 - Noise evolution

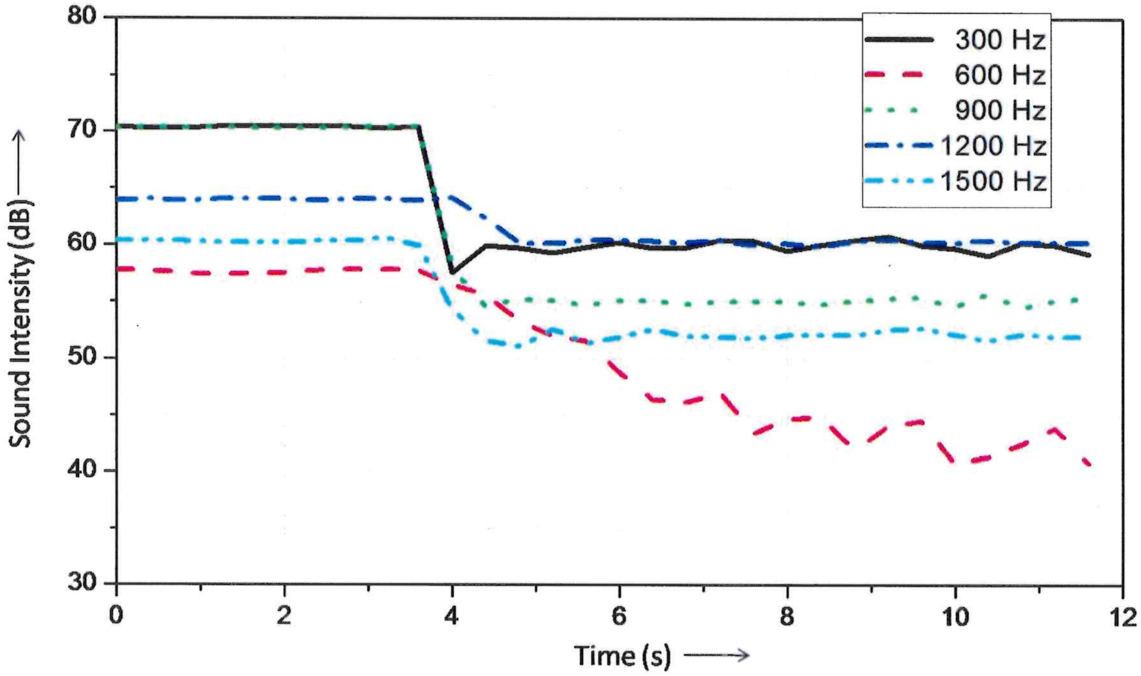

Figure 5.13 – Waterfall graph for noise.

As well as the previous experiment with sin wave, the vibration control using impulse provided also a good noise reduction in all frequencies desired, with an overall noise reduction of about 10 dB.

5.1.2: Feedback Control

Unfortunately, the Feedback control technique implementation was not successful, resulting in an instable control for the experiments made.

Such instability is related to several factors, but the most important is the non linear behavior of the piezo actuator, which tend to have high frequency noises when they are controlling low frequency noises. These high frequency noises turn to enter in the system as an error signal, making the actuators try to control the same noises they produce. This phenomenon ends in a looping, with the actuators trying to control higher frequencies noises until the overload, causing the instability.

Since the Hybrid control also depends on the Feedback control, it wasn't tested.

5.2: Experiment with real process

This section will present the results obtained with the experiments in real time process. For these experiments, the leakage factor was maintained at 1e-7. The system layout in real process is shown in Figure 5.14, and figure 5.15 shows the error evolution measured by the accelerometer during the machining, without control.

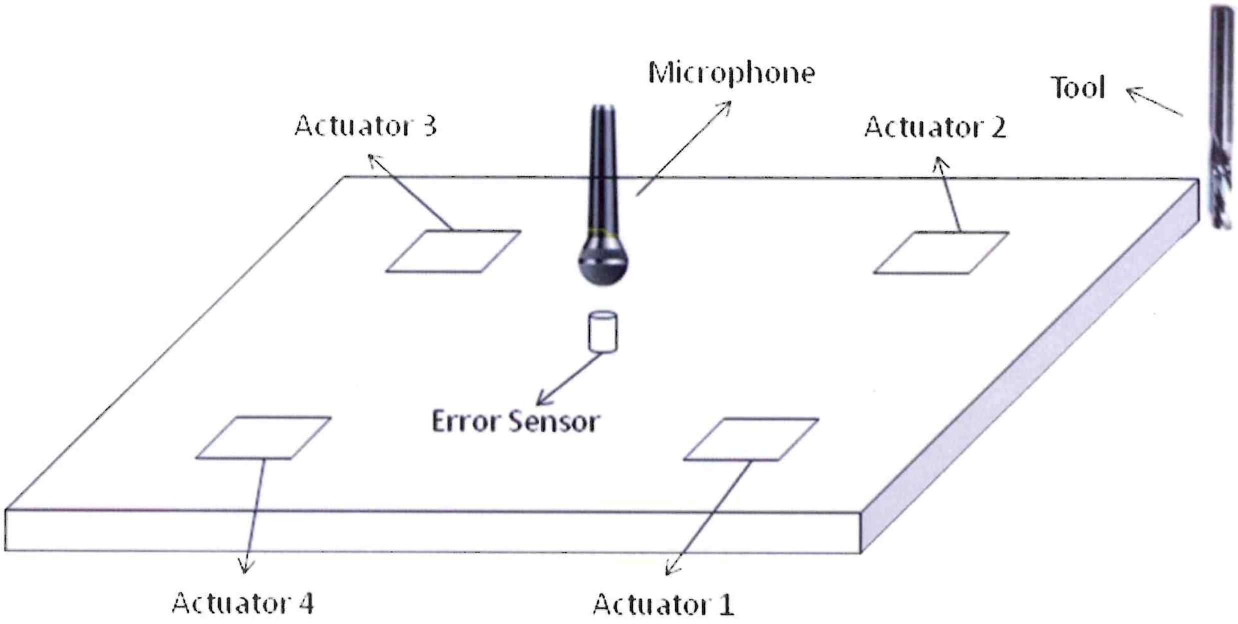

Figure 5.14 - System layout in real process

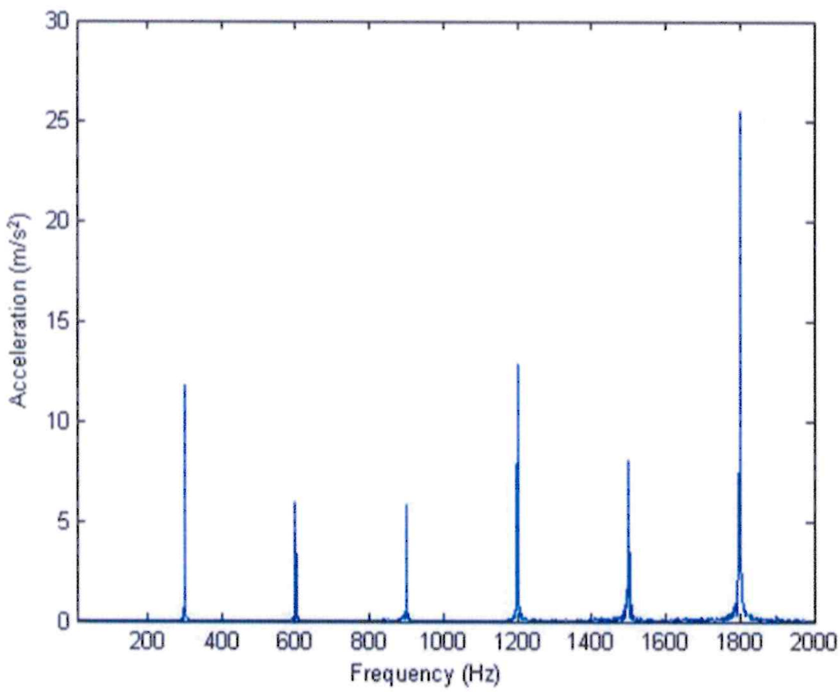

Figure 5.15 – Error evolution from accelerometer

5.2.1: Sine Wave as a Reference

As mentioned in the beginning of this chapter, during machining process, microphone and accelerometer captured noises in different ranges of frequencies. Because of that, the reference signal will be composed of a sum of several sin waves signals, each one having one frequency contained in the noise measured.

Several experiments were tested using this approach, each one having different control characteristics, such as:

- Frequencies of vibration to be controlled, and the choice of the actuators to control each frequency
- Feed speed and cutting depth of machining
- Choice of the step size for each primary filter
- Value of the reference gains of the sin wave signals

The choice of frequencies to be controlled and the actuators responsible for each frequency could be defined in the experiments with shaker, once these experiments were relatively fast, and it was possible to test almost every control possibilities. The initial values of feed speed and cutting depth were chosen according to the limitations of the actuators. The feed speed value started with 20 mm/sec with satisfactory results, however the control performance decreased when the speed increased. The initial cutting depth was 1mm and was increased until 2mm, presenting a good vibration reduction for both cases, although the noise reduction for 1mm was bigger.

For the choice of step size and reference gains, both parameters presented an optimal value, where for higher values than this optimal, the control usually led to instability or actuator's overload. For values smaller than the optimal, the control convergence speed was reduced. The optimal value found for step size is 1e-8, and the optimal reference gains for 300, 600 and 900 Hz are, respectively, 4, 3 and 2.

The next figures will show the experiment with the best result, for both vibration and noise reduction. In this experiment, the control system tried to reduce the vibration in the fundamental frequency (300 Hz) and its two first harmonics. Three actuator were used and each one was responsible for one frequency. The feed speed and cutting depth used in machining was, respectively, 20 mm/sec and 1mm. The step size values used for actuators 2, 3 and 4 are, respectively, 2e-8, 2e-8 and 1e-8.

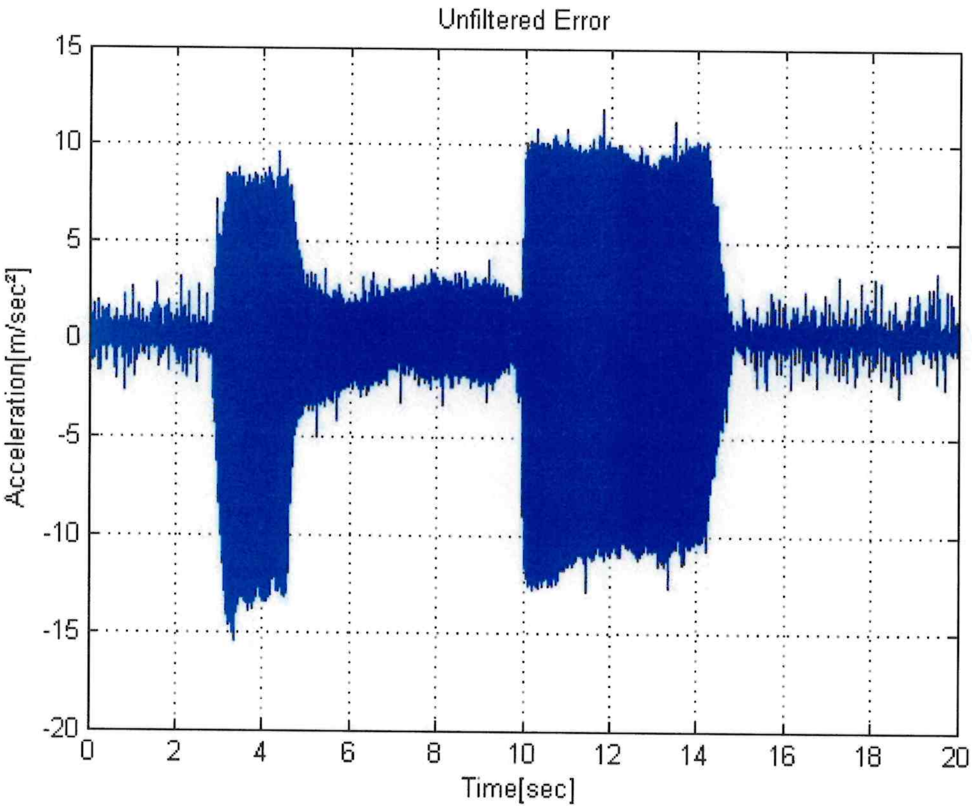

Figure 5.16 - Error evolution from accelerometer

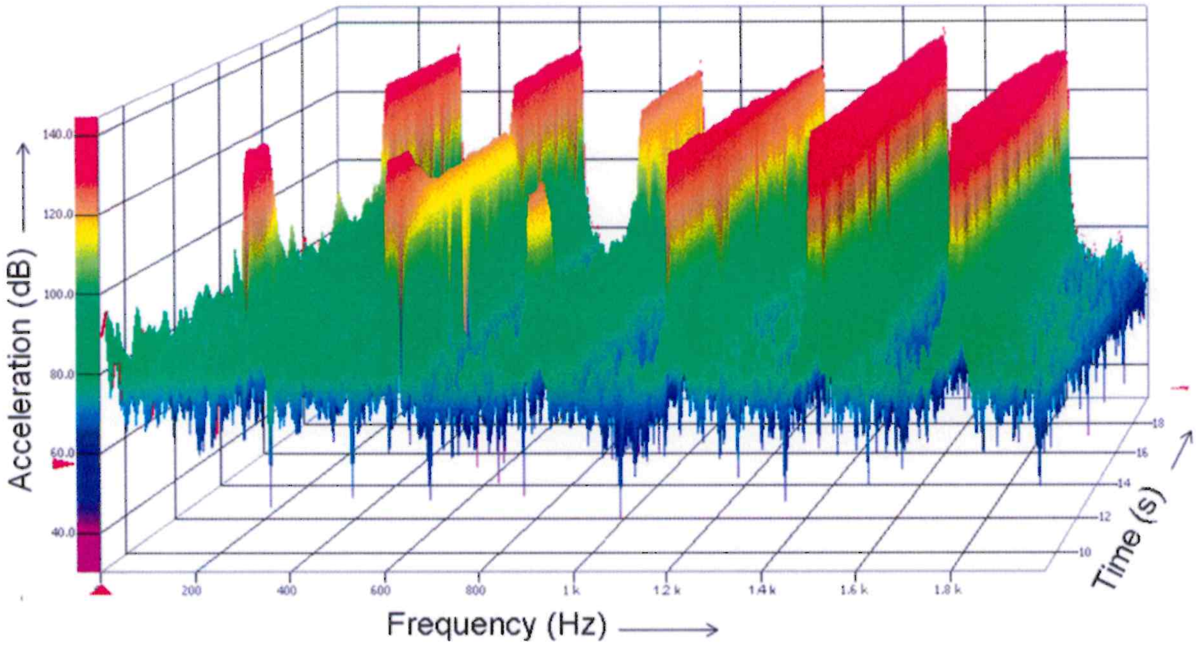

Figure 5.17 - Error evolution in frequency spectrum

Figure 5.16 showed a good performance of the control system, with an overall vibration reduction of over 75% (20dB, 10dB and 30dB reduction for 300Hz, 600Hz and 900Hz respectively).

Figure 5.18 shows the control signal evolution. It can be easily noticed the variation of the control signal, depending of the frequency to be reduced. Finally, figures 5.19 and 5.20 show the noise reduction measured by the microphone.

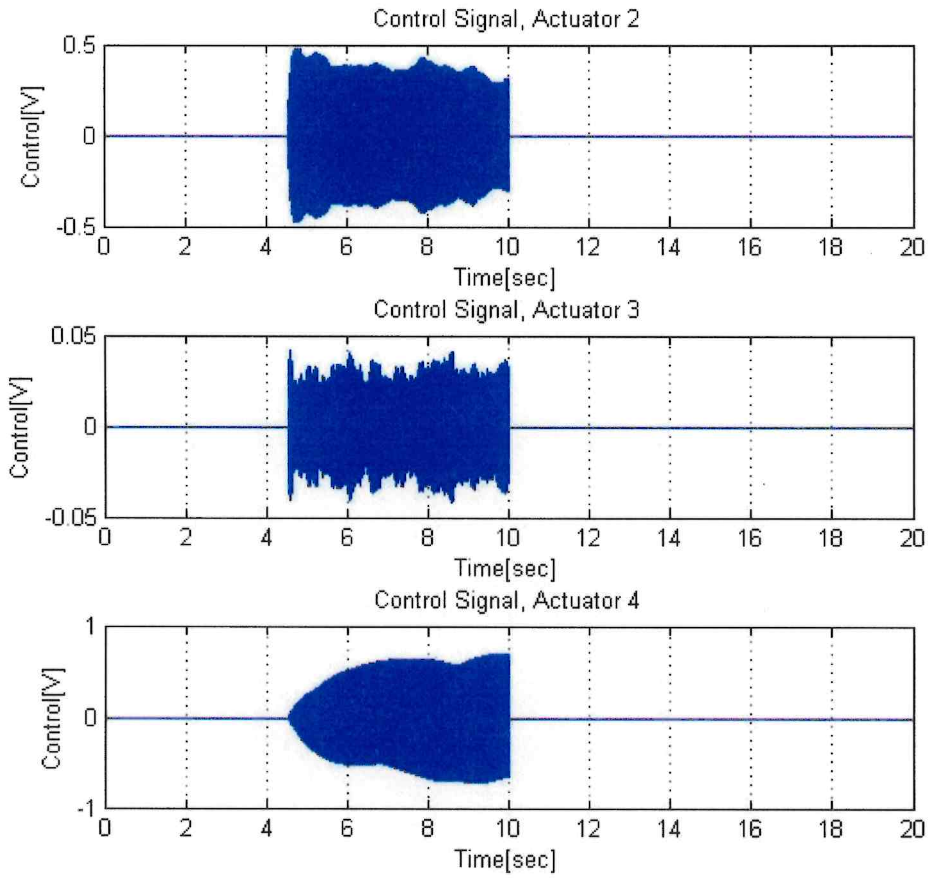

Figure 5.18 - Control signal evolution

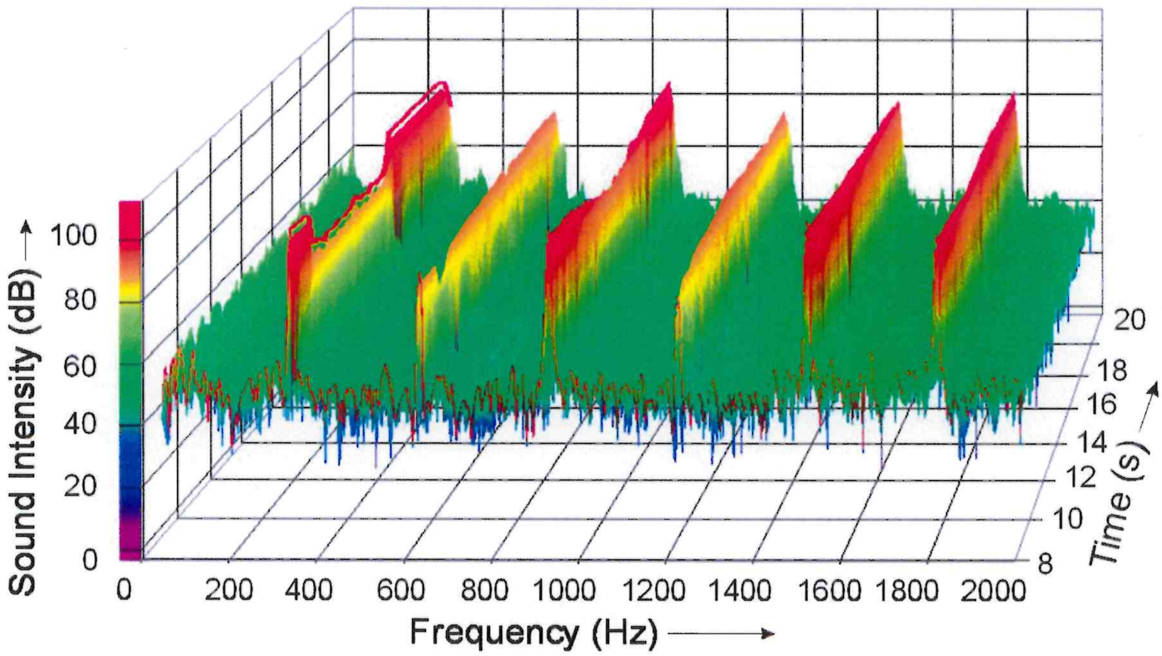

Figure 5.19 - Noise evolution capted by microphone

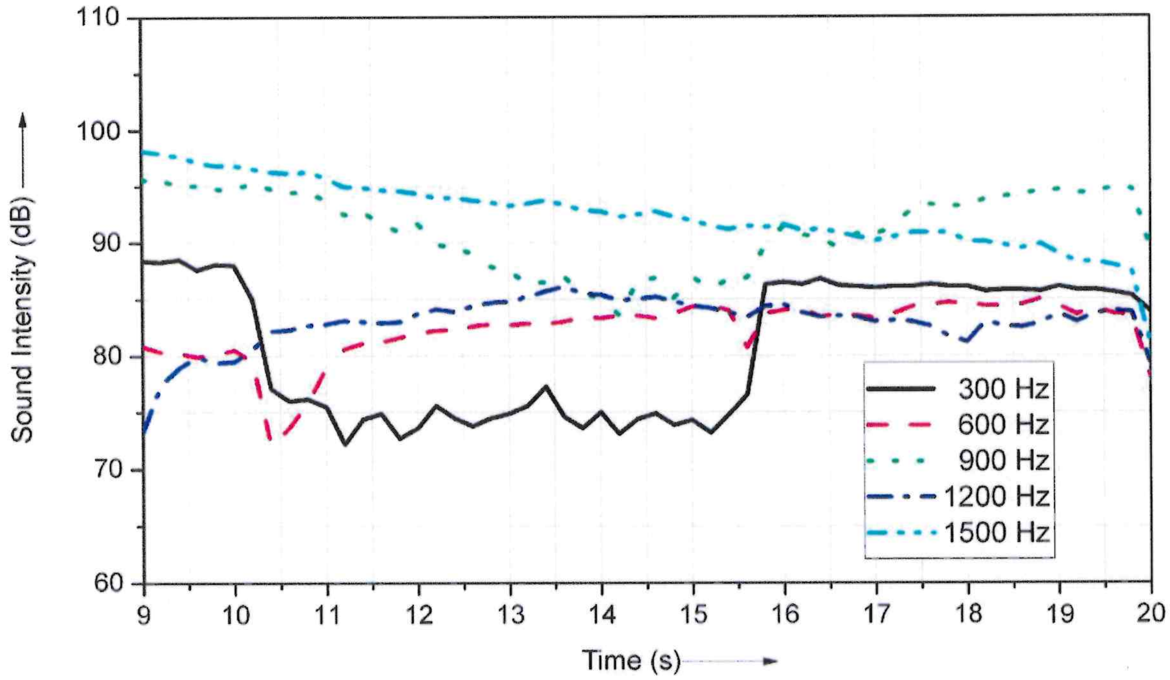

Figure 5.20 - Noise evolution in frequency spectrum

The control strategy using sin wave as a reference in real process proved to be efficient, for both vibration and noise reduction, this last one with an overall reduction of approximately 12 dB.

5.2.2: Impulse as Reference

The control system using the impulse signal as reference presented a worse performance than the previously control system with sin wave as reference.

Analyzing the figure 5.15, each frequency of vibration during machining has a different amplitude. This behavior can be easily compensated changing the reference gains of the sin wave signals, as shown in last section, but with impulse reference, this problem couldn't be solved as well. Because of that, despite the control system's adaptive behavior, the vibration reduction couldn't be achieved.

Figures 5.21 and 5.22 show the error evolution and the control signal during the process. Since the vibration reduction was not satisfactory, the noise signals from microphone were not measured. The step size value used for all filters is 1e-8.

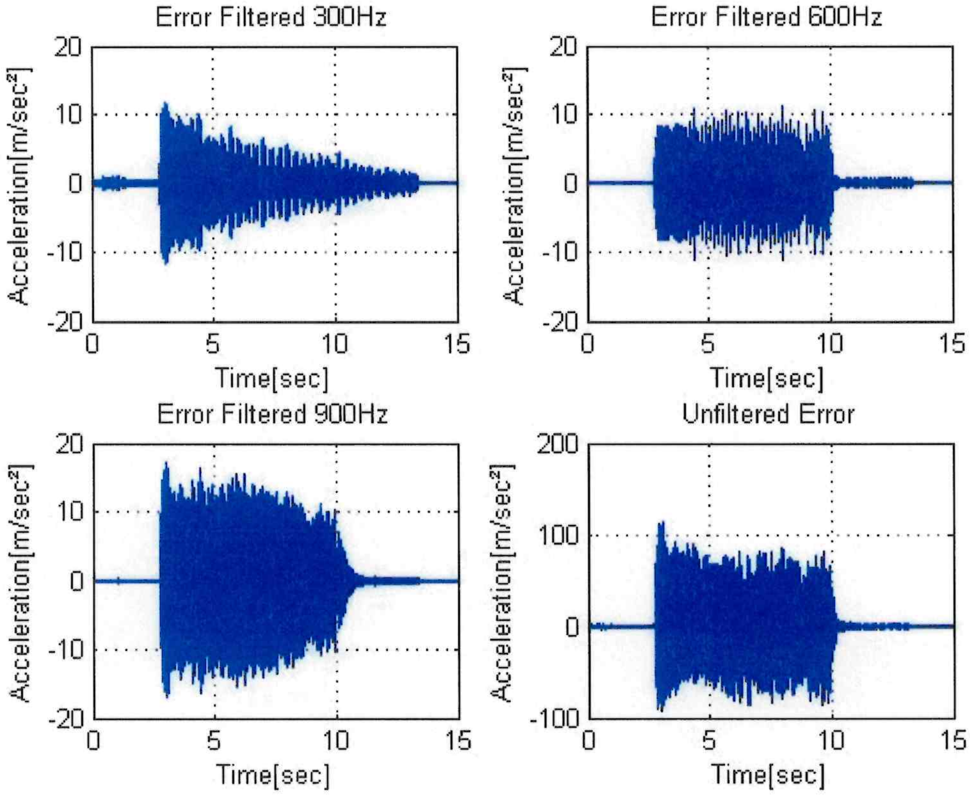

Figure 5.21 - Error signal evolution

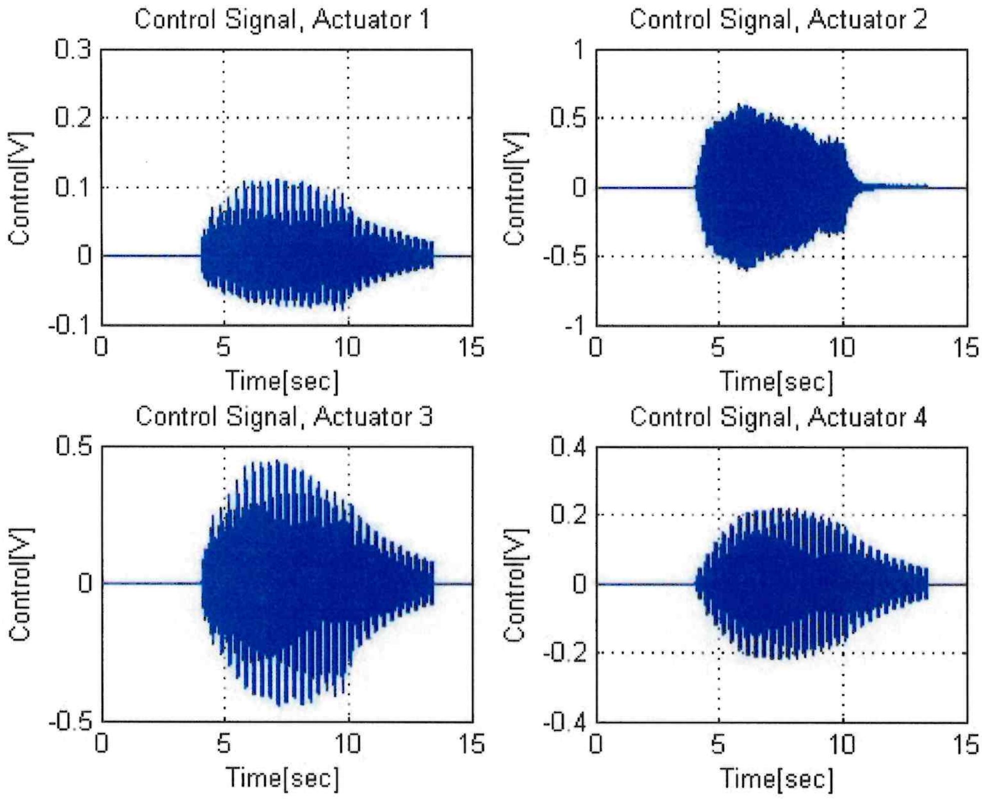

Figure 5.22 - Control signal evolution

5.3: Discussion of results and final considerations

Chapter 5 presented all experiments approaches and layouts used in the wood milling process, showing that the Feedforward control system with sin wave as reference proved to be the most effective for vibration control, and consequently noise reduction.

However, is important to note that even the approach chosen as the best for the problem proposed, there are some factors which can change the performance of the control system.

First, the position of actuators and sensor is extremely important for the performance of the control. Depending on the position of the actuators/suction blocks in relation to the plate, the vibration's propagation changes considerably. Once the position of the actuators are chosen, the sensor error must be located in a place with good variation of amplitude. In this project, before the experiments with AVC in the wood milling process, a model analysis of the wood plate was made aiming to locate the best position of the accelerometer.

Second, once the objective of this project is to verify the influence of the vibration control over the noise reduction during milling process, the position of the microphone sensor, although it isn't part of the control system, is also relevant. It is important to note that in all measurements of noise reduction made in this project, the microphone was positioned close to the plate/machine. Measurements of noise in different positions will certainly show different results.

Finaly, some parameters mentioned in previous sections are as important as the factors listed above, and worth to mention again. They are step size parameter, feed speed, cutting depth of machining and the hysteresis of actuators.

Chapter 6: Conclusions and Future Works

An implementation of active vibration reduction algorithms and structures has been made in this project. Experiments in wood milling process using the control strategies studied proved there is a strong link between vibration control and noise reduction.

From all control strategies tested in the project, the Feedforward Active Vibration Control using sin wave signals as a reference has been proved to be very efficient for both vibration control and noise reduction. Most of the experiments made with this approach could reach a vibration reduction of about 75%, and a noise reduction of about 10 dB, proving that is possible to reduce the undesirable system's noise, thus ensuring a better working environment.

For future works, one interesting study still remains to be done. The Feedback Active Vibration Control proved to be effective in the simulation with Matlab Simulink, however, in practical implementation the performance wasn't satisfactory. A better study of this control strategy and the factors that influence this bad performance, such as the non-linear behavior of the piezoactuators, could help to solve the problem.

References

[1] Kuo, S. M.; Morgan, D. R.; "Active Noise Control Systems: Algorithms and DSP Implementations", John Wiley & Sons, New York, NY, USA, 1996.

[2] Hansen, C. H.; Snyder, S. D.; "Active Control of Noise and Vibration". E & FN Spon, London, UK, 1997.

[3] Kuo, S. M.; Morgan, D. R.; "Active Noise Control: a tutorial review". Proceedings of the IEEE, vol. 87, no. 6, pp. 943-973, 1999.

[4] Fraanje, R.; "Robust and Fast Schemes in Broadband Active Noise and Vibration Control". Ph.D. Thesis, University of Twente, The Netherlands, 2004.

[5] Boersma, G. H.; "Design and implementation of AVI controllers on a SISO setup", Master's thesis, University of Twente, The Netherlands, 2006.

[6] Hoffmeister, H. W.; Schuller, B. C.; Loeis, K.; "Active Vibration Reduction on Clamping Systems for Stationary Machining Centers". Adaptronic Congress 2007, Göttingen, 2007.

[7] Ellitott, S. J.; Nelson, P. A.; "The application of adaptive filtering to the active control of sound and vibration", ISVR, Univ. Southampton, U.K., Tech. Rep. 136, Sept. 1985.

[8] Morgan, D. R.; "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path", IEEE Trans. Acoust., Speech, signal Processing, vol. ASSP-28, pp. 454-467, Aug. 1980.

Appendix A – FxLMS Control Algorithm

```
#define S_FUNCTION_NAME  FxLMS_Control
#define S_FUNCTION_LEVEL 2

#include "LMS.h"    //Here we define the Filter Orders

#include "simstruc.h"
#include <math.h>
#include <stdlib.h>

// Control
#define NORMAL_OPERATION 1
#define RESET 0          // w(k+1)=w_0
#define RETAIN 2         // w(k+1)=w(k)


/*====================*
 * S-function methods *
 *====================*/

/* Function: mdlInitializeSizes
===============================================
 * Abstract:
 *    The sizes information is used by Simulink to determine the S-function
 *    block's characteristics (number of inputs, outputs, states, etc.).
 */
static void mdlInitializeSizes(SimStruct *S)
{
    /* See sfuntmpl_doc.c for more details on the macros below */

    ssSetNumSFcnParams(S, 0);  /* Number of expected parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        /* Return if number of expected != number of actual parameters */
        return;
    }

    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, PrimPathOrder);       //wn: filter weights

    if (!ssSetNumInputPorts(S, 6)) return;
    // void ssSetInputPortWidth(SimStruct *S, int_T port_index, int_T
width)
        ssSetInputPortWidth(S, 0, 1);                  // x: disturbance
        ssSetInputPortDirectFeedThrough(S, 0, 1);
        ssSetInputPortWidth(S, 1, 1);                  // e: prediction error
        ssSetInputPortWidth(S, 2, 2);                  // param: mu, lambda
        ssSetInputPortWidth(S, 3, PrimPathOrder);         // w_0: initial
filter weights
        ssSetInputPortDirectFeedThrough(S, 3, 1);
        ssSetInputPortWidth(S, 4, 1);
        ssSetInputPortWidth(S, 5, SecPathOrder);       // Secondary Path
Filter weights
        ssSetInputPortDirectFeedThrough(S, 5, 1);

    if (!ssSetNumOutputPorts(S, 2)) return;
        ssSetOutputPortWidth(S, 0, 1);                 // y: predicted output
        ssSetOutputPortWidth(S, 1, PrimPathOrder);     // w: filter weights

    ssSetNumSampleTimes(S, 1);
```

```
    ssSetNumRWork(S, PrimPathOrder);
    /* Measurement Set isnt internal states (in the s-function meaning of
it) */
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    ssSetOptions(S, 0);
}


/* Function: mdlInitializeSampleTimes
=========================================
 * Abstract:
 *    This function is used to specify the sample time(s) for your
 *    S-function. You must register the same number of sample times as
 *    specified in ssSetNumSampleTimes.
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, INHERITED_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);

}


#define MDL_INITIALIZE_CONDITIONS   /* Change to #undef to remove function
*/
#if defined(MDL_INITIALIZE_CONDITIONS)
  /* Function: mdlInitializeConditions
========================================
   * Abstract:
   *    In this function, you should initialize the continuous and discrete
   *    states for your S-function block.  The initial states are placed
   *    in the state vector, ssGetContStates(S) or ssGetRealDiscStates(S).
   *    You can also perform any other initialization activities that your
   *    S-function may require. Note, this routine will be called at the
   *    start of simulation and if it is present in an enabled subsystem
   *    configured to reset states, it will be call when the enabled
subsystem
   *    restarts execution to reset the states.
   */
  static void mdlInitializeConditions(SimStruct *S)
  {
  }
#endif /* MDL_INITIALIZE_CONDITIONS */


#define MDL_START  /* Change to #undef to remove function */
#if defined(MDL_START)
  /* Function: mdlStart
=====================================================
   * Abstract:
   *    This function is called once at start of model execution. If you
   *    have states that should be initialized once, this is the place
   *    to do it.
   */
  static void mdlStart(SimStruct *S)
  {
    int i;
    real_T *wn = ssGetRealDiscStates(S);
```

```
    InputRealPtrsType w_0 = ssGetInputPortRealSignalPtrs(S,3);

    for (i=0; i<PrimPathOrder; i++)
        wn[i] = *w_0[i];     //Initialize Filter
  }
#endif /*  MDL_START */


/* Function: mdlOutputs
=======================================================
 * Abstract:
 *    In this function, you compute the outputs of your S-function
 *    block. Generally outputs are placed in the output vector, ssGetY(S).
 */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    int i;
    double y_temp = 0.0;

    real_T *y = ssGetOutputPortRealSignal(S,0);
    real_T *w_out = ssGetOutputPortRealSignal(S,1);
    real_T *wn = ssGetRealDiscStates(S);
    real_T *Measurement_Set_x = ssGetRWork(S);
    InputRealPtrsType u = ssGetInputPortRealSignalPtrs(S,0);

    UNUSED_ARG(tid); /* not used in single tasking mode */

    for (i=PrimPathOrder-1; i>0; i--)
        Measurement_Set_x[i] = Measurement_Set_x[i-1];
    Measurement_Set_x[0] = **u;

    for (i=0; i<PrimPathOrder; i++)
        {
        y_temp += Measurement_Set_x[i] * wn[i];
        w_out[i] = wn[i];
    }

    *y = y_temp;
}


#define MDL_UPDATE  /* Change to #undef to remove function */
#if defined(MDL_UPDATE)
  /* Function: mdlUpdate
=======================================================
   * Abstract:
   *    This function is called once for every major integration time step.
   *    Discrete states are typically updated here, but this function is
useful
   *    for performing any tasks that should only take place once per
   *    integration step.
   */
  static void mdlUpdate(SimStruct *S, int_T tid)
  {
    int i, j;
    double Filtered_Set_x[PrimPathOrder];

    real_T *wn = ssGetRealDiscStates(S);
    real_T *Measurement_Set_x = ssGetRWork(S);
    InputRealPtrsType e = ssGetInputPortRealSignalPtrs(S,1);
    InputRealPtrsType param = ssGetInputPortRealSignalPtrs(S,2);

    InputRealPtrsType w_0 = ssGetInputPortRealSignalPtrs(S,3);
```

```
    InputRealPtrsType control = ssGetInputPortRealSignalPtrs(S,4);
    InputRealPtrsType  secpath  = ssGetInputPortRealSignalPtrs(S,5);

    double mu = *param[0];
    double lambda = *param[1];

    UNUSED_ARG(tid); /* not used in single tasking mode */

    /* xdot=Ax+Bu */

    switch((int)**control)
        {
        case RESET:
            for (i=0; i<PrimPathOrder; i++)
                wn[i] = *w_0[i];
            break;

        case RETAIN: //do nothing
            break;

        default:
            if(**control == NORMAL_OPERATION) //Filterd X
                for(i=0; i<PrimPathOrder; i++)
                    {
                    Filtered_Set_x[i] = 0;
                    for(j=0; ((j<SecPathOrder)&&(i+j<PrimPathOrder)); j++)
                        Filtered_Set_x[i] += (*secpath)[j] *
Measurement_Set_x[i+j];
                    }
            else //Unfiltered X
                for(i=0; i<PrimPathOrder; i++)
                    Filtered_Set_x[i] = Measurement_Set_x[i];

            for (i=0; i<PrimPathOrder; i++)
                wn[i] = (1.0-lambda) * wn[i] + mu * (**e) *
Filtered_Set_x[i];
            break;
    }
}
#endif /* MDL_UPDATE */


#define MDL_DERIVATIVES  /* Change to #undef to remove function */
#if defined(MDL_DERIVATIVES)
  /* Function: mdlDerivatives
=================================================
   * Abstract:
   *    In this function, you compute the S-function block's derivatives.
   *    The derivatives are placed in the derivative vector, ssGetdX(S).
   */
  static void mdlDerivatives(SimStruct *S)
  {
  }
#endif /* MDL_DERIVATIVES */


/* Function: mdlTerminate
=====================================================
 * Abstract:
 *    In this function, you should perform any actions that are necessary
 *    at the termination of a simulation.  For example, if memory was
 *    allocated in mdlStart, this is the place to free it.
 */
static void mdlTerminate(SimStruct *S)
```

```
{
    UNUSED_ARG(S);
}

/*=========================================================*
 * See sfuntmpl_doc.c for the optional S-function methods *
 *=========================================================*/

/*==============================*
 * Required S-function trailer *
 *==============================*/

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```