ADAPTIVE FUZZY SLIDING MODE CONTROL OF A WASHING MACHINE OSCILLATING SYSTEM

By

Can TAMYAMAN
B.S., Electronics & Com. Eng., Yıldız Technical University, 2002

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
The requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University
2005

ACKNOWLEDGEMENTS

I would like to thank my family for their endless love and support.

I would also like to thank my advisor Prof. Kaynak for all his support and understanding.

ABSTRACT

To prevent any mechanical damage, washing machines distribute the load (the laundry) before performing any spins. After some distributive motor movements, there may still be some load that does not lay homogenously on the surface of the tumble. This load may cause undesired oscillations of the system. In cases when the load is not distributed homogenously, the washing machine senses this unbalanced distribution if it is over a threshold value and reduces the spin speed or does not perform the final spin in the washing profile. Because the machine is allowed to perform the final spin if the unbalanced distribution measurements are under a certain limit; some undesired oscillations and overshoots in the displacement of the oscillating system of the washing machine is likely to take place. This shortens the mechanical life of the machine. Furthermore, any undesired movements of the machine may pose a danger in a household. For these reasons the risk of machine displacement should be calculated and tested on every washing machine at the design phase. The machines are either produced with a calculated risk of displacement, or the spin profile is modified to reduce the risk according to the results of the tests at design level. In this thesis, the control of a washing machine suspension system is achieved by using adaptive fuzzy sliding mode control methods. The design and implementations are carried out in MATLAB Simulink environment. In the thesis, first, the methods that are going to be used and their advantages are given. Then, the unbalanced distribution of load and its effects on washing machines are explained. Finally, the behavior of the proposed suspension system is examined in Simulink environment.

ÖZET

Herhangi bir mekanik hasarlanmayı önlemek için, çamaşır makineleri, sıkma işlemi yapmadan önce yükü (çamaşırı) dağıtıcı bazı hareketler yaparlar. Bu hareketlerden sonra bile yükün bir kısmı tambur üzerine homojen olarak dağılmayıp, tahrik grubu dediğimiz mekanik sistemin istenmeyen salınımlar yapmasına sebep olabilir. Bu gibi yükün dengesiz olarak dağıldığı durumlarda, günümüz çamaşır makineleri belli bir limitin üstündeki dengesiz dağılımı algılar ve sıkma devrini otomatik olarak azaltır, ya da yıkama profilinin sonundaki sıkmayı gerçekleştirmez. Ancak dengesiz olarak dağılan yük izin verilen limitlerin altında ise makine belirli bir sıkma profilini gerçekleştirdiğinden, tahrik grubunun istenmeyen salınımlar yapmasının önüne geçilemez. Bu durumda da hem kazan rulman ömürleri ile ilgili problemler ortaya çıkabilir hem de makineler müşteri evinde çalıştırıldıklarında yürüyebilirler. Bu nedenle, tasarım aşamasında yapılan hesaplama ve testlerde öngörülen riske göre de ürün üretilir ya da yazılımda tanımlanan sıkma profili ile oynanarak bu risk azaltılmaya çalışılır. Bu tez çalışmasında bu istenmeyen salınımları kontrol altında tutacak bir çamaşır makinesi süspansiyon sisteminin MATLAB Simulink ortamında uygulaması gerçekleştirilmiştir. Bu tez çalışmasında öncelikle kullanılacak yöntem ile ilgili mevcut çalışmalardan alınan bilgiler, ve bu yöntemin getirileri derlenmiştir. Daha sonra ise çamaşır makinelerinde yüklerin dengesiz olarak dağılması durumu ve etkileri incelenmiştir. Son bölümde ise tasarlanan süspansiyon sisteminin davranışları Simulink ortamında incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SMC	sliding-mode control
MISO	multiple input – single output
FBF	fuzzy basis function
AFSMC	adaptive fuzzy sliding-mode control
UB	unbalanced
RPM	rotations per minute
g	grams
kg	kilograms
mm	millimeters
s	seconds
N	Newtons

1. INTRODUCTION

Most engineers who do not deal with fuzzy applications think that fuzzy logic is an alternative way to conventional control techniques; but fuzzy and conventional control are complementary rather than competitive. This thesis unites the two ideas on the same application. Sliding mode control is the conventional method mentioned on this work.

A washing machine has to drain the free water in the tub and then distribute the wet load to minimize the "unbalanced load" before performing a spin. The term in quotation marks is used for the laundry that is not distributed equally in the tumble surface before the spin. Unbalanced load causes forces that lead the oscillating system to make unwanted displacements. These forces have a negative effect on the lifetime of the machines mechanical equipment and also make the machine move. To prevent these results, some distributive motor movements are held before performing the spin. Then the load that is not distributed equally in the circular surface of the tumble is sensed from the characteristics of tachometer signals. Finally the spin is held in an appropriate speed with the measured unbalanced load.

In this thesis, it is proposed not to reduce the spin speed because of the unbalanced load, but to control the displacement of the oscillating system especially during the spin phase with a controller component that will be assembled between the body and the tub of the machine. This will both increase customer satisfaction and the mechanical life of the system.

As an introduction to theory, in Section 2 a general overview about sliding mode control is given. In Section 3; this time we take a general look at fuzzy sets and systems.

In Section 4 there are two methods mentioned for adaptive fuzzy sliding mode control. In the first method, a fuzzy logic system is utilized to approximate the unknown function f of the nonlinear system. In the second method, two fuzzy logic systems are utilized to approximate the f and g, respectively. In these control schemes, the simple adaptive laws are designed to approximate the nonlinear functions by fuzzy logic systems. In the first method, a robust adaptive law is introduced to reduce the approximation errors between true system functions and fuzzy approximators. In the second method, the control law that is robust to approximation error is also designed.

In Section 5, a brief introduction is made to the washing machine oscillating systems and the unbalanced load problem.

A simulation is presented in Section 6 with the oscillating system of a front-load washing machine.

The results are examined and discussed in the Section 7 that is the conclusion part of this work.

2. AN OVERWIEV OF SLIDING MODE CONTROL

In the last decade, there has been much work on fuzzy sliding mode control (SMC) approaches. SMC has excellent robustness properties with regard to parametric uncertainty. Consider a nonlinear system governed by the differential equation

$$x^n = f(x, \dot{x}, ..., x^{(n-1)}) + g(x, \dot{x}, ..., x^{(n-1)})u + d(t) \tag{2.1}$$

where x $\in R^n$ is the output of the system, u $\in R$ is the input (control vector), and

$$x = (x, \dot{x}, ..., x^{(n-1)})^T \tag{2.2}$$

is the state vector that is assumed to be observable, and f and g are unknown nonlinear functions and d(t) is the time-dependent disturbances with known upper bounds. In order $x^{(n)}$ to be controllable we assume that g>0 for x in being controllability region $U_c \subset R^n$ and it is also assumed that d(t) have upper bound D; that is

$$|d(t)| \leq D \tag{2.3}$$

If the state trajectory e has reached the sliding surface s=0; the system trajectory remains on it while sliding into the origin e=0; independent of model uncertainties, unmodeled frequencies and disturbances.

The control objective is to determine a feedback control u=u(x) such that the state x of the closed-loop system will follow the desired state

$$x_d = (x_d, \dot{x}_d, ..., x_d^{(n-1)})^T \tag{2.4}$$

that is, the tracking error

$$e = x_d - x = (e, \dot{e}, ..., e^{(n-1)})^T \tag{2.5}$$

should converge to zero. Then a sliding surface in the space of the error state can be defined as

$$s(x,t) = -(k_1 e + k_2 \dot{e} + ... + k_{n-1} e^{(n-2)} + e^{(n-1)}) = -k^T e \tag{2.6}$$

where the coefficients $k_1, k_2, .., k_{n-1}$ are the coefficients of a Hurwitzian polynomial

$$\lambda^{(n-1)} + k_{n-1}\lambda^{(n-2)} + ... + k_1 \tag{2.7}$$

If initial error e(0)=0, the tracking problem can be considered as the state error vector *e* remaining on the sliding surface s for all $t \geq 0$.

The process of sliding mode control can be divided into two phases, that is, the approaching phase with

$$s(x; t) \neq 0 \tag{2.8}$$

and the sliding phase with

$$s(x; t) = 0. \tag{2.9}$$

A sufficient condition to guarantee that the trajectory of the error vector e will translate from approaching phase to sliding phase is to select the control strategy such that;

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s| \tag{2.10}$$

Let

$$\dot{s} = -(k_1\dot{e} + k_2\ddot{e} + \ldots + k_{n-1}e^{(n-1)}) + x_d^{(n)} - x^{(n)}$$

$$= -\sum_{i=1}^{n-1} k_i e^{(i)} + x_d^{(n)} - x^{(n)}$$

$$= -\sum_{i=1}^{n-1} k_i e^{(i)} - f(x,t) - g(x,t)u(t) - d(t) + x_d^{(n)} \qquad (2.11)$$

Since f and d are unknown, we can select the control law as

$$u_1 = \frac{1}{\hat{g}(x,t)}\left(\sum_{i=1}^{n-1} k_i e^i - \hat{f}(x,t) - (D+\eta_\Delta)\,\mathrm{sgn}(s) + x_d^{(n)}\right) \qquad (2.12)$$

where

$$D + \eta_\Delta \geq \eta \geq 0 \qquad (2.13)$$

It is obvious that in order to obtain the sliding mode control law, the system function $\hat{f}(x,t), \hat{g}(x,t)$ and switching parameter $(D+\eta_\Delta)$ have to be determined in finalizing the design.

So to simplify this concept, and to clarify what we were doing up to now, let us consider SMC for a single input-single output n^{th} order non-linear system. The differential equation of the system becomes;

$$x^n = f(x,t) + u(t) + d(t) \qquad (2.14)$$

where $u \in R$ is the input of the system. The state vector $x \in R^n$;

$$x = (x, \dot{x}, \ldots\ldots, x^{(n-1)})^T \tag{2.15}$$

and the output $y \in R$ is;

$$y = x^{(n)} \tag{2.16}$$

Here $f(x,t)$ is the unknown nonlinear function and d(t) is the unknown external disturbance. It is assumed that

$$|d(t)| \leq D\ . \tag{2.17}$$

The control objective is to obtain the state x for tracking a desired state

$$x_d = (x_d, \dot{x}_d, \ldots, x_d^{(n-1)})^T \tag{2.18}$$

in the presence of model uncertainties and unknown disturbances.

Let the tracking error be;

$$e = x_d - x = (e, \dot{e}, \ldots, e^{(n-1)})^T \tag{2.19}$$

Then a sliding surface in the space of the error state can be defined as

$$s(x,t) = -(k_1 e + k_2 \dot{e} + \ldots + k_{n-1} e^{(n-2)} + e^{(n-1)}) = -k \cdot e \tag{2.20}$$

where $k = (k_1, k_2, .., k_{n-1}, 1)$ and the coefficients $k_1, k_2, .., k_{n-1}$ are the coefficients of the Hurwitz polynomial

$$\lambda^{n-1} + k_{n-1}\lambda^{n-2} + \ldots + k_1 . \tag{2.21}$$

If the initial error e(0)=0, then the tracking problem can be considered as the state error vector e remaining on the sliding surface

$$s(e; t)=0 \tag{2.22}$$

for all t ≥0.

The process of sliding mode control can be divided into two phases, that is, the approaching phase with s(x, t)≠0 and the sliding phase with s(x, t)=0.

A sufficient condition to guarantee that the trajectory of the error vector *e* will translate from the approaching phase to the sliding phase is to select the control strategy such that

$$s(x,t) \bullet \dot{s}(x,t) \le -\eta |s|,\ \eta > 0 \ \text{(Sliding condition)} \tag{2.23}$$

Corresponding to two phases, two types of control law can be derived separately. In the sliding phase, we have $s = 0$ and $\dot{s} = 0$, then the equivalent control u_{eq} which will force the system dynamics to stay on the sliding surface can be obtained as follows.

Let

$$s = -(k_1 e + k_2 e^{(1)} + \ldots + k_{n-1} e^{(n-1)} + x^{(n)}) + x_d^{(n)}$$

$$= -\left(\sum_{i=1}^{n-1} k_i e^{(i)} + x^{(n)} \right) + x_d^{(n)}$$

$$= \left(\sum_{i=1}^{n-1} k_i e^{(i)} + x^{(n)} \right) + x_d^{(n)} = 0 \tag{2.24}$$

Then;

$$u_{eq} = \sum_{i=1}^{n-1} k_i e^{(i)} + f(x,t) + d(t) + x_d^{(n)} \tag{2.25}$$

In the approaching phase, where s ≠0, in order to satisfy the sliding condition (2.23), a switching-type control term u_{sw} must be added and the resulting sliding mode control law will be

$$u = u_{eq} - u_{sw}\,, \tag{2.26}$$

$$u_{sw} = \eta_{\Delta}\,\mathrm{sgn}(s)\,, \tag{2.27}$$

where $\eta_{\Delta} \geq \eta > 0$.

It is obvious that in order to obtain the sliding mode control law, the system function $f(x,t)$ and switching parameter η_{Δ} have to be determined.

3. A BRIEF INTRODUCTION TO FUZZY LOGIC SYSTEMS

A definitive diagram of a fuzzy logic system configuration is given in (Figure 3.1). In this part, we will discuss general fuzzy logic system and fuzzy approximator.

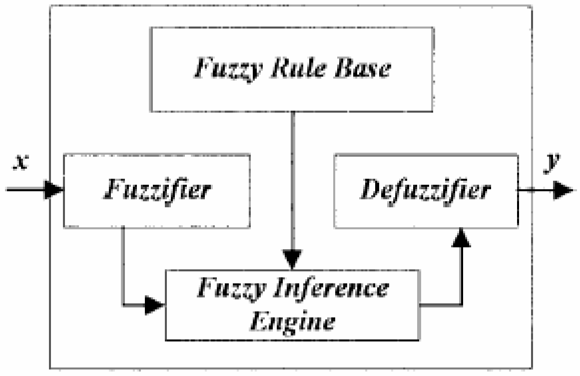

Figure 3.1 – A basic Fuzzy Logic System

Let us first take a look at "knowledge base" which is constructed with fuzzy rules. The knowledge base for the fuzzy logic system comprises a collection of fuzzy IF-THEN rules. To consider the mostly common application, multiple-input single-out (MISO) rules will be used in the formulation of the control law. The MISO IF-THEN rule(s) are of the form

$$R^{(j)}: \textit{If } x_1 \text{ is } A_1^j \text{ and } \ldots \text{ and } x_n \text{ is } A_n^j, \text{ then } y \text{ is } C^j \tag{3.1}$$

where $\underline{x} = (x,\ldots,x_n)^T \in V \subset R^n$ and $y \in W \subset R$ denote the linguistic variables associated with the inputs and output of the fuzzy logic system.

A_i^j and C^j are labels of the fuzzy sets in *V* and *W*, respectively, and *i* denotes the number of input/state of fuzzy logic system, for example, *i = 1,2,…,n;* and *j* denotes the number of rules of the fuzzy logic system, like *j = 1,2,…,M*. Fuzzy rule (3.1) can be implemented using fuzzy implication, which gives

$$A_i^j \times ... \times A_n^j \rightarrow C^j \tag{3.2}$$

which is a fuzzy set defined in the $V \times W$ product space. Based on generalizations of implications in multi-value logic, many fuzzy implication rules have been proposed in the fuzzy logic literature.

The implication rule using t-norm operator is defined as,

$$\mu A_i^j \times ... \times A_n^j \rightarrow C^j(\underline{x}, y) = \mu A_i^j(x_1) * ... * \mu A_n^j(x_n) * \mu C^j(y) \tag{3.3}$$

where * denotes t-norm, which corresponds to the conjunction “min” or “product” in general.

The next step we should take to understand fuzzy systems is understanding “fuzzy inference engine”. The fuzzy inference engine performs a mapping from fuzzy sets in V to fuzzy sets in R, based on the fuzzy IF-THEN rules in fuzzy rule base and the compositional rule of inference.

Let *B* be a fuzzy set in V, then the fuzzy relational equation $B \circ R^j$ where “o” is the supplementary star composition, results in M fuzzy sets. Using the t-norm operator yields

$$\mu B \circ R^j(y) = \sup \underline{x}[\mu B(\underline{x}) * \mu A_1^j \times ... \times A_j^n \rightarrow C^j(\underline{x}, y)] \tag{3.4}$$

In order to combine the M fuzzy sets into one fuzzy set t-norm can be employed, which results in

$$\mu B \circ (R^1, ... R^M)(y) = \mu B \circ R^{(1)}(y) \dotplus ... \dotplus \mu B \circ R^{(M)}(y) \tag{3.5}$$

where $\dotplus$ denotes the s-norm, the most commonly used operation for $\dotplus$ is "max." If we use the product operation and choose * in eqn. (3.3) and (3.4) to be an algebraic product, then the inference is called product inference. Using product inference, (3.4) becomes

$$\mu B \circ R^j(y) = \sup \underline{x} \in V\left[\mu B(\underline{x}) \mu A_1^j(x_1) \ldots \mu A_j^n(x_n) \mu C^j(y)\right] \tag{3.6}$$

3.1 Fuzzifier

The fuzzifier maps a crisp point $\underline{x}$ into a fuzzy set B in V . In general, there are two possible choice of this mapping namely, singleton, or nonsingleton. Here, we use the singleton fuzzifier mapping given by the eqn. (3.7);

$$\mu B(\underline{x}') = \begin{cases} 1 \; for\; \underline{x}' = \underline{x} \\ 0 \quad other \end{cases}, \text{ for } x' \in V \tag{3.7}$$

3.2 Defuzzifier

The defuzzifier maps fuzzy sets in R to a crisp point in R. In general, there are three possible choices of this mapping namely, maximum, center-average, and modified center-average defuzzifier.

We will use the center-average defuzzifier mapping, as given below

$$y = \frac{\sum_{j=1}^{M} y^{-j}(\mu B \circ R^{(j)}(y^{-j}))}{\sum_{j=1}^{M} (\mu B \circ R^{(j)}(y^{-j}))} \tag{3.8}$$

where y^{-j} is the point in R at which $\mu \cdot C^j$ achieves its maximum value (assume that $\mu C^j(\overline{y}^j) = 1$).

3.3 Fuzzy Bases Function

The fuzzy logic system with *center-average defuzzifier* (3.8), *product inference* (3.6), and *singleton fuzzifier* (3.7) is of the following form,

$$y(\underline{x}) = \frac{\sum_{j=1}^{M} y^{-j}\left(\prod_{i=1}^{n} \mu A_i^j(x_i)\right)}{\sum_{j=1}^{M}\left(\prod_{i=1}^{n} \mu A_i^j(x_i)\right)} \tag{3.9}$$

If the terms $\mu \cdot A_i^j(x_i)$'s are fixed and the y^{-j}'s are viewed as adjustable parameters, then (3.9) can be written as;

$$y(\underline{x}) = \theta^T \xi(\underline{x}) \tag{3.10}$$

where $\theta = (y^{-1},..., y^{-M})^T$ is a parameter vector, and $\xi(\underline{x}) = (\xi^1(x),...,\xi^M(\underline{x}))^T$ is a regressive vector with the regressor $\xi^j(\underline{x})$ defined as;

$$\xi(\underline{x}) = \frac{\prod_{i=1}^{n} \mu A_i^j(x_i)}{\sum_{j=1}^{M}\left(\prod_{i=1}^{n} \mu A_i^j(x_i)\right)} \tag{3.11}$$

which are called fuzzy bases functions (FBF's). It is proved that these FBF's are universal approximators. We can fix all the parameters in $\xi^j(\underline{x})$ at the beginning of the FBF expansion design procedure, so that the only free design parameters are adaptive parameter vectors θ_i.

4. ADAPTIVE FUZZY SLIDING MODE CONTROL

We will introduce the two methods from the recent sources about adaptive fuzzy sliding-mode control (AFSMC).

4.1 Method I: Assuming the Accurate Lower Bound of the Control Gain Function is Known

The input of the SMC is previously found as;

$$u_1 = \frac{1}{\hat{g}(x,t)}\left(\sum_{i=1}^{n-1} k_i e^i - \hat{f}(x,t) - (D+\eta_\Delta)\,\mathrm{sgn}(s) + x_d^{(n)}\right) \qquad (4.1)$$

If $f(\underline{x},t)$, $g(\underline{x},t)$ are known, we can easily construct the proper SMC input u_1 given above at (refer to SMC). However, f and g are unknown. To find the solution of this problem, we replace the $f(\underline{x},t)$ by the fuzzy logic system $\hat{f}(\underline{x}|\theta)$, which is in the form of (36) or (37) and consider the term $g^{-1}\,\mathrm{sgn}(s)|F_1|$ in order to reduce the disturbance due to unknown control gain. Where we assume that;

$$0 < \underline{g} < g(\underline{x},t) \qquad (4.2)$$

and

$$g(\underline{x},t) = \underline{g} + \Delta g(\underline{x},t) \qquad (4.3)$$

and g is a known positive constant and $\Delta g(\underline{x},t)$ is an unknown positive function. The resulting control input is as follows:

$$u_1 = \underline{g}^{-1}\left(-\sum_{i=1}^{n-1} c_i \cdot e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h\cdot \mathrm{sgn}(s)\cdot(D+\eta_\Delta)\right) - g^{-1}\,\mathrm{sgn}(s)|F_1| \qquad (4.4)$$

where

$$F_1 = -\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) \tag{4.5}$$

Then;

$$\dot{s} = f(\underline{x},t) - f(\underline{x}|\theta) - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) + d(t)$$

$$+ \Delta g(\underline{x},t) \cdot g^{-1} F_1 - g(\underline{x},t) \underline{g}^{-1} \mathrm{sgn}(s) |F_1| \tag{4.6}$$

4.1.1 Adaptive Law Synthesis

Let the optimal parameter vector of fuzzy logic system θ^*, we can define the minimum approximation error

$$w = f(\underline{x},t) - \hat{f}(\underline{x}|\theta^*) \tag{4.7}$$

So (4.6) can be written as

$$\dot{s} = \hat{f}(\underline{x}|\theta^*) - f(\underline{x}|\theta) + w - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) + d(t)$$

$$+ \Delta g(\underline{x},t) \cdot g^{-1} F_1 - g(\underline{x},t) \underline{g}^{-1} \mathrm{sgn}(s) |F_1| \tag{4.8}$$

If we choose $\hat{f}$ to be the fuzzy logic system in the form (3.10), then (4.8) can be rewritten as

$$\dot{s} = \phi^T \xi(\underline{x}) + w - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) + d(t) + \Delta g(\underline{x},t) \cdot g^{-1} F_1 - g(\underline{x},t) \cdot g^{-1} \cdot \mathrm{sgn}(s) |F_1| \tag{4.9}$$

where $\phi = \theta^* - \theta$ and, $\xi(\underline{x})$ is the fuzzy basis function (3.11). Now consider the Lyapunov candidate

$$V_1 = \frac{1}{2}\left(s^2 + \frac{1}{\gamma_1}\phi^T \phi \right) \tag{4.10}$$

where γ_1 is positive constant. The time derivative of V_1 is

$$\dot{V}_1 = s\dot{s} + \frac{1}{\gamma_1}\phi^T \dot{\phi} \tag{4.11}$$

and

$$\begin{aligned}\dot{V}_1 &= s\phi^T \xi(\underline{x}) + s \cdot w - s \cdot h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) + s \cdot dt + \frac{1}{\gamma_1}\phi^T \dot{\phi} \\ &+ s\Delta g(\underline{x},t) \cdot g^{-1} F_1 - |s| g(\underline{x},t) g^{-1} |F_1| \le \frac{1}{\gamma_1}\phi^T (\gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi}) + s \cdot w \\ &+ s\Delta g(\underline{x},t) \cdot g^{-1} F_1 - |s| g(\underline{x},t) g^{-1} |F_1| - |s| \cdot h \cdot \eta_\Delta \\ &< \frac{1}{\gamma_1}\phi^T (\gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi}) + s \cdot w - |s| \cdot h \cdot \eta_\Delta\end{aligned} \tag{4.12}$$

where $\dot{\phi} = -\dot{\theta}$. Because the term $s \cdot w$ is of the order of the minimum approximation error and from the universal approximation theorem, it is expected that the w should be very small if not equal to zero in the adaptive fuzzy system. Here, we will choose the following adaptive law to consider robustness concerning this term:

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}) \tag{4.13}$$

However, this approach is not complete; thus, the robust control techniques are considered under the assumption that the upper bound of w is known when we work on synthesizing the robust adaptive law.

4.1.2 Robust Adaptive Law Synthesis

Let the control input be

$$u_1 = \underline{g}^{-1}\left(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta + \hat{\rho})\right) - g^{-1}\,\mathrm{sgn}(s)\left|F_{1\gamma}\right| \qquad (4.14)$$

where

$$F_{1\gamma} = -\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta + \hat{\rho}) \qquad (4.15)$$

and estimation of *w,*

$$\hat{\rho} = \rho^* - \tilde{\rho}\,,\ \rho^* = |w|_{\max} \cdot w^* \qquad (4.16)$$

is the upper bound of minimum approximation error of fuzzy approximator. Therefore, the derivative of s can be obtained as follows:

$$\dot{s} = f(\underline{x},t) - \hat{f}(\underline{x}|\theta) - h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta + \hat{\rho}) + d(t) + \Delta g(\underline{x},t) \cdot g^{-1} F_{1\gamma} - g(\underline{x},t)\underline{g}^{-1}\,\mathrm{sgn}(s)\left|F_{1\gamma}\right| \qquad (4.17)$$

Now consider the Lyapunov candidate

$$V_{1\gamma} = \frac{1}{2}\left(s^2 + \frac{1}{\gamma_1}\phi^T\phi + \frac{1}{\gamma_2}\tilde{\rho}^2\right) \qquad (4.18)$$

Applying (4.17) to (4.18) and after straightforward manipulation, the time derivative of V_{1g} can be obtained as follows:

$$
\begin{aligned}
\dot{V}_{1\gamma} &= \frac{1}{\gamma_1}\phi^T(\gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi}) - s \cdot h \cdot \mathrm{sgn}(s) \cdot (D + \eta_\Delta) + s \cdot d(t) \\
&+ s \cdot w - h \cdot |s| \cdot \hat{\rho} + \frac{1}{\gamma_2}\tilde{\rho}\dot{\hat{\rho}} + s\left(\Delta g(\underline{x},t) \cdot g^{-1} \cdot F_{1\gamma} - g(\underline{x},t) \cdot g^{-1} \cdot \mathrm{sgn}(s)\left|F_{1\gamma}\right|\right) \\
&\le \frac{1}{\gamma_1}\phi^T \begin{pmatrix} \gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi}) + sw - h \cdot |s| \cdot \rho^* + h \cdot |s| \cdot (\rho^* - \hat{\rho}) + \frac{1}{\gamma_2}\tilde{\rho}\dot{\hat{\rho}} - h \cdot |s| \cdot \eta_\Delta \\ + s\Delta g(\underline{x},t) g^{-1} \cdot F_{1\gamma} - |s| g(\underline{x},t) \cdot g^{-1} \cdot \left|F_{1\gamma}\right| \end{pmatrix} \\
&\le \frac{1}{\gamma_1}\phi^T\left(\gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi})\right) + sw - h \cdot |s| \cdot \rho^* + h \cdot |s| \cdot (\rho^* - \hat{\rho}) - h \cdot |s| \cdot \eta_\Delta \\
&\le \frac{1}{\gamma_1}\phi^T\left(\gamma_1 \cdot s \cdot \xi(\underline{x}) + \dot{\phi})\right) + \frac{1}{\gamma_2}\tilde{\rho}(\dot{\hat{\rho}} + \gamma_2 \cdot h \cdot |s|) - h \cdot |s| \cdot \eta_\Delta
\end{aligned}
\tag{4.19}
$$

Therefore, the adaptive laws can be chosen as follows:

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}) \tag{4.20}$$

$$\dot{\hat{\rho}} = \gamma_2 \cdot h \cdot |s| \tag{4.21}$$

4.2 Method II: Reducing the Disturbance by Appending another Input which Comes from the Unknown Control Gain

In the first method, we have to know the accurate lower bound of the control gain function $g(\underline{x},t)$. But it is difficult to know this value precisely, so we propose another method that $f(\underline{x},t)$ and $g(\underline{x},t)$ are replaced by the fuzzy logic systems $\hat{f}(\underline{x}|\theta_f)$ and $\hat{g}(\underline{x}|\theta_f)$. We replace the f and g as two fuzzy logic systems $\hat{f}(\underline{x}|\theta_f)$ and $\hat{g}(\underline{x}|\theta_f)$ and append another input u_g in order to reduce the disturbance, which comes from the unknown control gain. The resulting control input is as follows:

$$u_2 = u_f + u_g \tag{4.22}$$

where;

$$u_f = \hat{g}^{-1}(x|\theta_g)\left(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta_f) + x_d^{(n)} - h\cdot \mathrm{sgn}(s)\cdot(D+\eta_\Delta)\right), \quad (4.23)$$

$$u_g = -\Gamma_1 \cdot \mathrm{sgn}(s)|u_f|, \quad (4.24)$$

and

$$\Gamma_1 \geq \frac{|w_g|_{max}}{g(\underline{x},t)|_{min}} \quad (4.25)$$

Then

$$\dot{s} = f(\underline{x},t) - \hat{f}(\underline{x}|\theta_f) + \left(g(\underline{x},t) - \hat{g}(\underline{x}|\theta_g)\right)u_f + g(\underline{x},t)\cdot u_g - h\cdot \mathrm{sgn}(s)\cdot(D+\eta_\Delta) + d(t) \quad (4.26)$$

4.2.1 Adaptive Law Synthesis

Let the optimal parameter vectors of fuzzy logic systems θ_f^*, θ_g^*, we can define the minimum approximation errors

$$w_f = f(\underline{x},t) - \hat{f}(\underline{x}|\theta_f^*),\ w_g = g(\underline{x},t) - \hat{g}(\underline{x}|\theta_g^*) \quad (4.27)$$

So, (4.26) can be written as

$$\dot{s} = \hat{f}(\underline{x}|\theta_f^*) - \hat{f}(\underline{x}|\theta_f) + w_f - h\cdot \mathrm{sgn}(s)\cdot(D+\eta_\Delta) + d(t) + \left(\hat{g}(\underline{x}|\theta_g^*) - \hat{g}(\underline{x}|\theta_g) + w_g\right)u_f + g(\underline{x},t)\cdot u_g \quad (4.28)$$

If we choose $\hat{f}$ and $\hat{g}$ to be the fuzzy logic systems in the form of (3.10), then (4.28) can be rewritten as

$$\dot{s} = \phi_f^T \xi_f(\underline{x}) + w_f + (\phi_g^T \xi_g(\underline{x}) + w_g) u_f + g(\underline{x}, t) \cdot u_g - h \cdot \operatorname{sgn}(s) \cdot (D + \eta_\Delta) + d(t) \tag{4.29}$$

where

$$\phi_f = \theta_f^* - \theta_f, \tag{4.30}$$

$$\phi_g = \theta_g^* - \theta_g, \tag{4.31}$$

and $\xi_f(\underline{x})$ and $\xi_g(\underline{x})$ are the fuzzy basis function (38). Now consider the Lyapunov candidate

$$V_2 = \frac{1}{2}\left(s^2 + \frac{1}{\gamma_3} \phi_f^T \phi_f + \frac{1}{\gamma_4} \phi_g^T \phi_g \right) \tag{4.32}$$

where γ_3 and γ_4 are positive constants. The time derivative of V_2 is

$$\begin{aligned} \dot{V}_2 &= s\dot{s} + \frac{1}{\gamma_3} \phi_f^T \dot{\phi}_f + \frac{1}{\gamma_4} \phi_g^T \dot{\phi}_g \\ &= \frac{1}{\gamma_3} \phi_f^T (s\gamma_3 \xi_f(\underline{x}) + \dot{\phi}_f) + s w_f + \frac{1}{\gamma_4} \phi_g^T (s \gamma_4 \xi_g(\underline{x}) u_f + \dot{\phi}_g) - sh \operatorname{sgn}(s)(D + \eta_\Delta) \\ &+ sd(t) + s w_g u_f - s\Gamma_1 g(\underline{x}, t) \operatorname{sgn}(s) |u_f| \le \frac{1}{\gamma_4} \phi_f^T (\gamma_4 s \xi_f(\underline{x}) + \dot{\phi}_f) + \frac{1}{\gamma_4} \phi_g^T (\gamma_4 s \xi_g(\underline{x}) u_f + \dot{\phi}_g) \\ &+ s w_f - |s| \cdot h \cdot \eta_\Delta \end{aligned} \tag{4.33}$$

where

$$\dot{\phi}_f = -\dot{\theta}_f \, , \; \dot{\phi}_g = -\dot{\theta}_g \; . \qquad (4.34)$$

Since the term $s \cdot w_f$ is of the order of the minimum approximation error, we can choose the adaptive laws

$$\dot{\theta}_f = \gamma_3 \cdot s \cdot \xi_f(\underline{x}) \, , \qquad (4.35)$$

$$\dot{\theta}_g = \gamma_4 \cdot s \cdot \xi_g(\underline{x}) \cdot u_f \qquad (4.36)$$

However, this approach is not complete; thus, we consider the robust control techniques also for the second method.

4.2.2 Robust Control Law

In order to reduce the disturbance coming from w_f, we consider the term $\Gamma_2 \cdot \mathrm{sgn}(s)$ in the control law. The resulting control input

$$u_{2\gamma} = u_f + u_{rob} \qquad (4.37)$$

where

$$u_{rob} = -\Gamma_1 \cdot \mathrm{sgn}(s) \cdot |u_f| - \Gamma_2 \cdot \mathrm{sgn}(s) \cdot \Gamma_2 \geq \frac{|w_f|_{max}}{g(\underline{x},t)|_{min}} \qquad (4.38)$$

then we can find $\dot{s}$ as follows;

$$\dot{s} = \phi_f^T \xi_f(\underline{x}) + w_f + (\phi_g^T \xi_g(\underline{x}) + w_g)u_f + g(\underline{x},t) \cdot u_{rob} - h \cdot \text{sgn}(s) \cdot (D + \eta_\Delta) + d(t) \qquad (4.39)$$

Now consider the Lyapunov candidate V_2 and applying (4.39) to (4.32) and after straightforward manipulation, we obtain the time derivative of V_2 as given below;

$$\begin{aligned}\dot{V}_2 &= \frac{1}{\gamma_3}\phi_f^T\left(s\gamma_3\xi_f(\underline{x}) + \dot{\phi}_f\right) + \frac{1}{\gamma_4}\phi_g^T\left(s\gamma_4\xi_g(\underline{x})u_f + \dot{\phi}_g\right) \\ &- s \cdot h \cdot \text{sgn}(s)(D + \eta_\Delta) + sd(t) + sw_g u_f - s\Gamma_1 g(\underline{x},t)\,\text{sgn}(s)\left|u_f\right| \\ &+ sw_f - s \cdot \Gamma_2 \cdot g(\underline{x},t) \cdot \text{sgn}(s) \le \frac{1}{\gamma_3}\phi_f^T\left(\gamma_3 s\xi_f(\underline{x}) + \dot{\phi}_f\right) + \frac{1}{\gamma_4}\phi_g^T\left(\gamma_4 s\xi_g(\underline{x})u_f + \dot{\phi}_g\right) \\ &- |s| \cdot h \cdot \eta_\Delta\end{aligned} \qquad (4.40)$$

Therefore, we can choose the adaptive laws as

$$\dot{\theta}_f = \gamma_3 \cdot s \cdot \xi_f(\underline{x}), \qquad (4.41)$$

$$\dot{\theta}_g = \gamma_4 \cdot s \cdot \xi_g(\underline{x}) \cdot u_f \qquad (4.42)$$

4.3 Boundary Layer and Fuzzy Switching

The sliding control law that discussed above is given below once again;

$$u_1 = \frac{1}{\hat{g}(x,t)}\left(\sum_{i=1}^{n-1} k_i e^i - \hat{f}(x,t) - (D + \eta_\Delta)\,\text{sgn}(s) + x_d^{(n)}\right) \qquad (4.43)$$

This law function is discontinuous at the sliding surface s(t) and this leads to chattering. This chattering (Figure 4.1(b)) is not desirable because it involves high control activity. It may be used in high frequency dynamics. Defining a boundary layer neighboring the sliding surface would make the control changes continuously and will lead to smoothing, instead of this unwanted the chattering in (Figure 4.1(b)).

$$B(t) = \{x : |s(x,t)| \le \Phi\} \tag{4.44}$$

As we change the control law (4.43) to

$$u = \sum_{i=1}^{n-1} k_i e^i - \hat{f}(x,t) + x_d^{(n)} - (D + \eta_\Delta) sat(s/\Phi) \tag{4.45}$$

where the saturation function $sat(s/\Phi)$ is defined as;

$$sat(s/\Phi) = \begin{cases} -1 & \text{if } s/\Phi \le -1 \\ s/\Phi & \text{if } -1 \le s/\Phi \le 1 \\ 1 & \text{if } s/\Phi > 1 \end{cases} \tag{4.46}$$

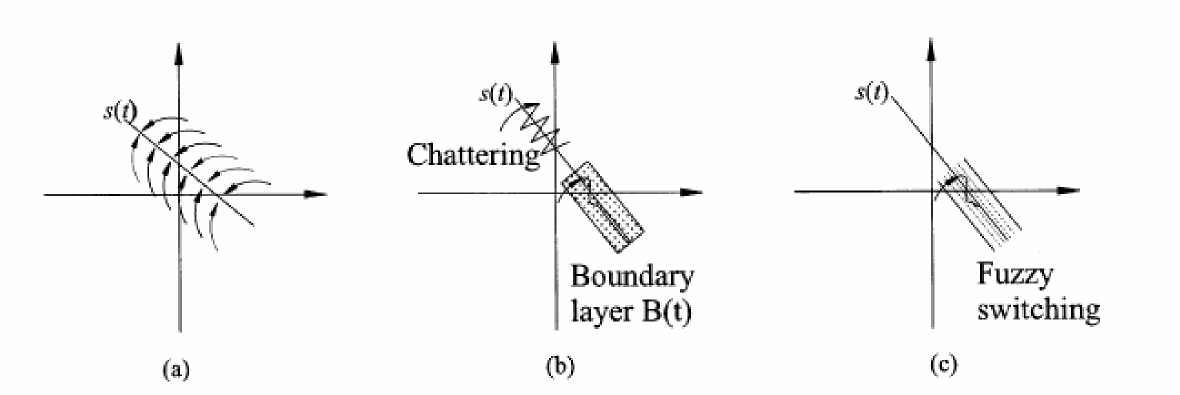

Figure 4.1 – Switching of SMC. (a) Sliding condition in two-dimensional planes. (b) Sliding condition of chatting and boundary layer. (c) Sliding condition of fuzzy switching.

A fuzzy approach for to avoid the chattering at the boundary layer of the SMC can be helpful. We can regard $\hat{h}(s|\theta_h)$ as a series of boundary layer which linked nonlinear smoothly with fuzzy system as we see in (Figure 4.1(c)). The initialization of the switching term can be selected to be boundary layer which is a linear switching and continue to update the switching term $\hat{h}(s|\theta_h)$ to reduce the steady-state error.

5. WASHING MACHINE SPIN PROFILES AND UNBALANCED LOAD PROBLEM

On front-load washing machines, every program has some intermediate spins after washing and rinsing phases, and a final spin with the highest RPM value after the softener phase (the final rinsing phase). Before performing these spins, the machine has to drain the free water in the tub and then distribute the wet load to minimize the “unbalanced load”. This term is used for the laundry which causes the unwanted displacements of system with the central acceleration, because of not being distributed well before the spin.

These forces have a negative effect on the lifetime of the machines mechanical equipment and also make the machine move.

Figure 5.1 - The supporting mill between the wheel and the tumble. Holds the tumble steady inside the tub while it is rotating.

To prevent these unwanted forces, every washing machine has an unbalance-load algorithm buried in each spin profile. This algorithm is made up of three phases, which are

- Distributing the laundry in the machine by making special motor movements
- Sensing the load which is not distributed equally in the circular surface of the tumble (the unbalanced load).

- Performing the spin in an appropriate speed with the measured unbalanced load.

The distribution of the laundry is done by rotating the laundry in a changing rotation speed, such as consequent 52 RPM, 65 RPM and 80 RPM movements for short periods. These speeds can be changed for each model but the distribution motion's mentality is same in almost all benchmarks.

Sensing is performed by reading the tachometer signals at a 100 RPM spin for a period of time and calculating the noise in the tachometer signal. A coefficient is calculated in every control system for the noise caused by the unit grams of unbalanced load. By comparing the measured coefficient by previously decided coefficients for levels of unbalanced load which are embedded in the software, a decision for the spin is made.

The final spin speed is reduced or not performed and the spinning step is skipped. The thresholds differ from model to model for unbalanced load. But for a 5kg capacity washing machine, they are about 400 g of unbalanced load for reducing the spin speed and 600 g for not performing the spin.

Figure 5.2 – Washing machine tumble and the plastic tub

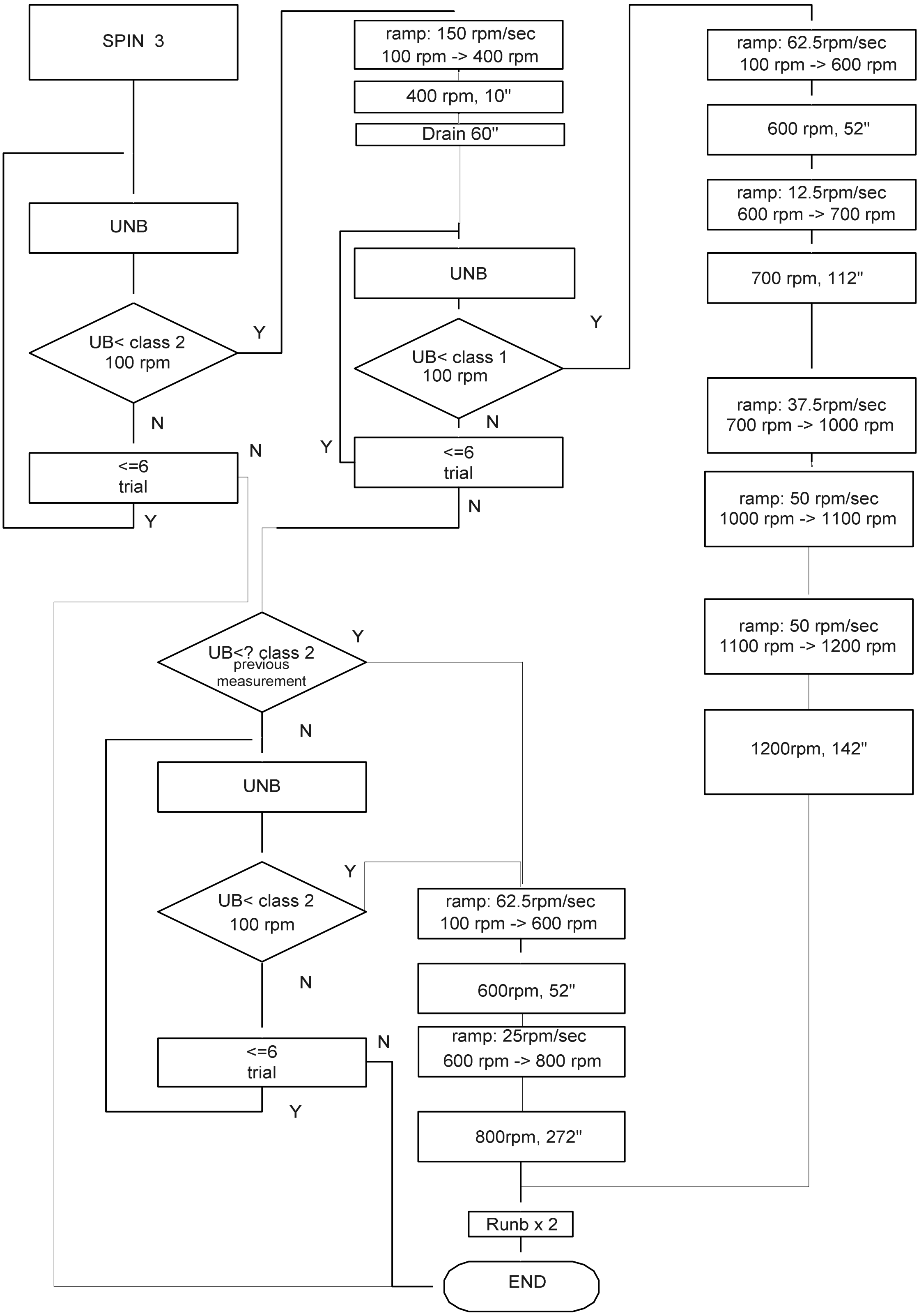

Figure 5.3 – Spin profile of a washing machine with an unbalanced load algorithm. UNB is the sub-profile containing the distributive motions and sensing at 100RPM. RUNB is the final distributive motion to prevent creasing in the laundry.

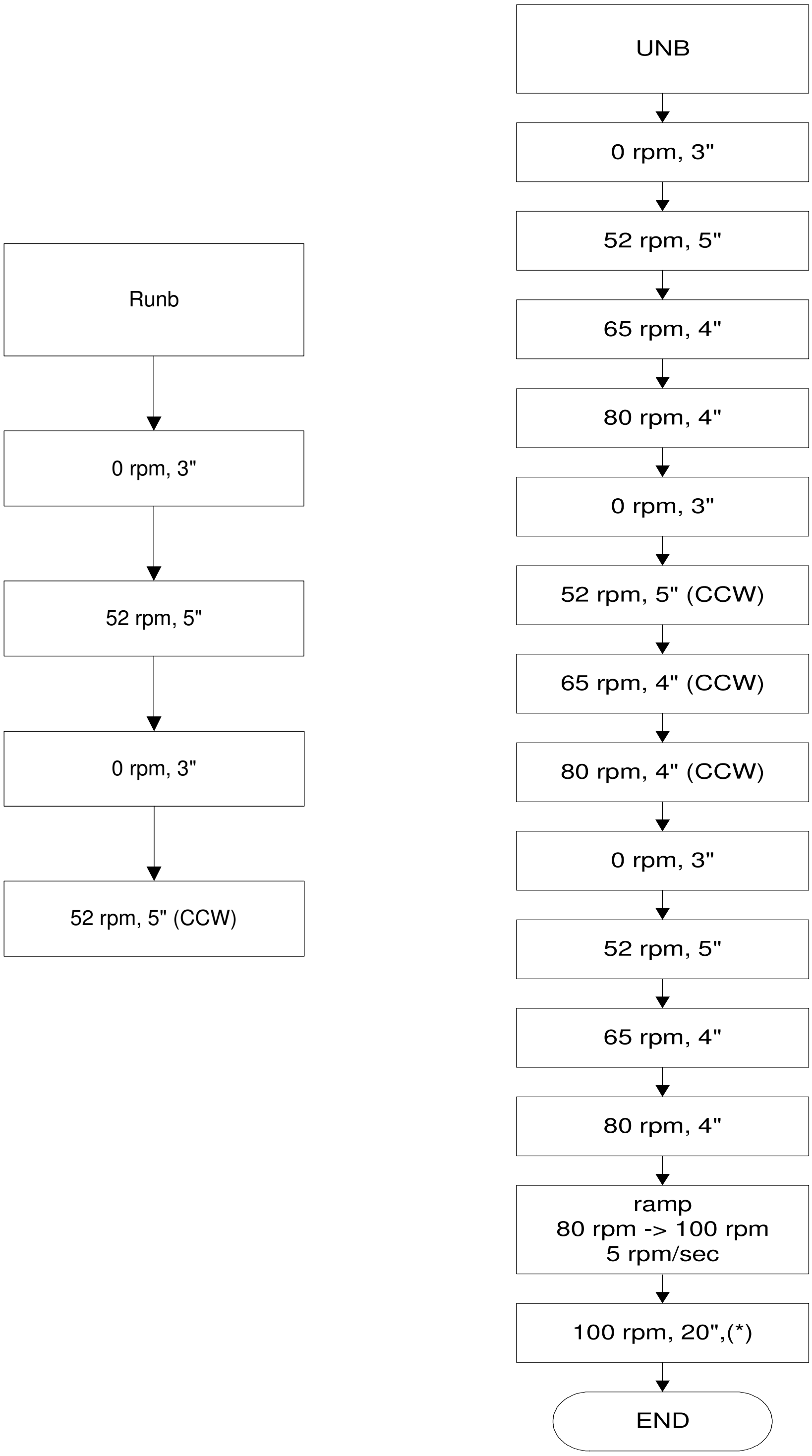

Figure 5.4- Sub-profiles UNB and RUNB

5.1 Calculating the Threshold Unbalance-Load Coefficients

The mechanical designer for the tumble and the tub inform the system engineer about the threshold values which are going to be used in the mechanical system. Then the designer of the system makes some experiments with special metal loads differing from weights 100 grams up to 1 kilogram.

Figure 5.5 – RPM measurement from a washing machine mill in the laboratory.

These metal loads are fixed to the tumble in means of simulating the unbalanced load and the motor is driven at 100 RPMs with the control system's special version which is loaded with special software for sensing this unbalanced load. This special software drives the motor in the same way the control system does in real operation. At 100 RPM the noise on the tachometer signal is measured as a coefficient by the software and the distribution of these coefficients are listed and sorted in the computer. Of course, this test must be repeated in at least 10 machines to be applicable in real life.

Finally the control system designer chooses appropriate coefficients for the defined thresholds and puts these values in the related part of the software.

For example, let us say that our thresholds are 400 g and 650 g, our coefficients would be such as 1900 for 450 g and 3250 for 650 g with this distribution.

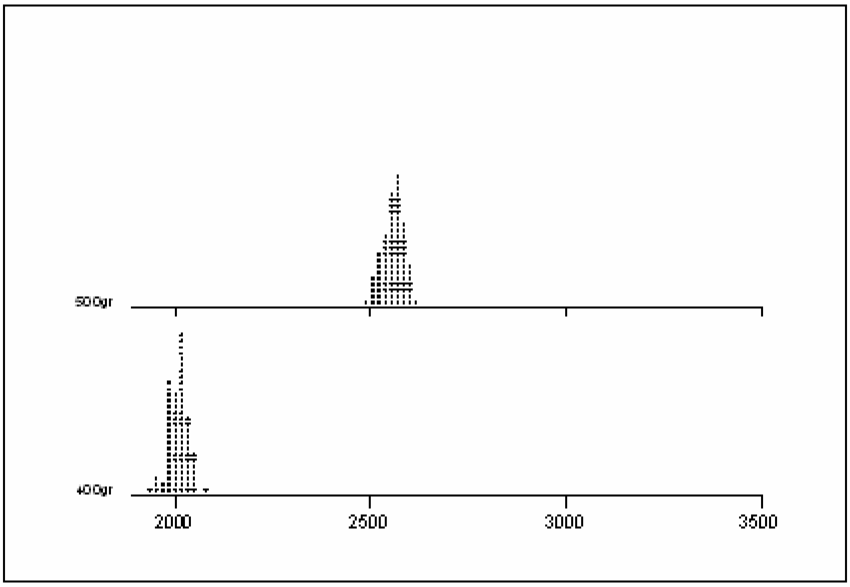

Figure 5.6 – Unbalanced load coefficient distributions for 400g and 500g

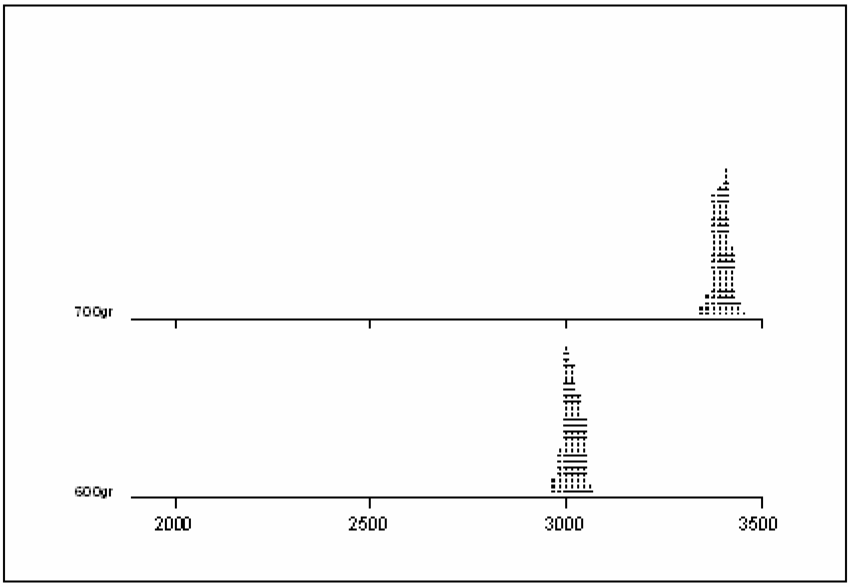

Figure 5.7 – Unbalanced load coefficient distributions for 600g and 700g

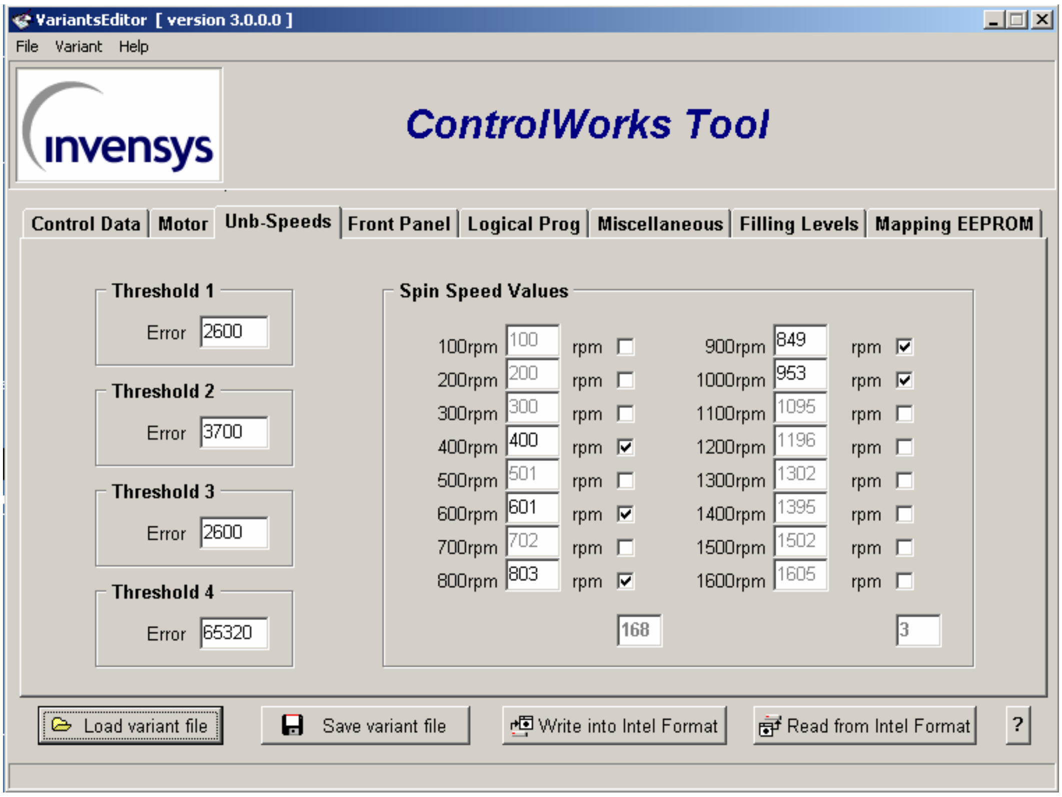

Figure 5.8 – The software editor for creating variant files

The system designer can apply these values to the software by an editor or writing these inside the embedded code, depending on the software structure of the system.

As given in the Figure 5.3 this machine will spin up to 1200 RPM under a threshold of 400 g of unbalanced load. The maximum speed reached can be only 800 RPM if there is a balance load between 400 g and 650 g. None of the spins will be performed at unbalanced loads over 650 g.

6. SIMULATION ON WASHING MACHINE OSCILLATING SYSTEM

In this section we will try to design an imaginary controller (a suspension system) to minimize the displacement of the washing machine's oscillating system. The controller can be designed in two pieces; which are in x-axis and in y- axis.

The purpose of minimizing the displacement is to overcome the negative effects of unbalanced load; such as the "machine walking" in common saying, damaging the coil or the dampers or even the oscillating system itself.

Recent front load washing machines have some special "unbalanced load detection" algorithms but, the algorithms are all based in canceling the spinning step after washing or decreasing the set spin speed value and minimizing the physical effects. Before deciding which spin speed is proper at recent conditions, every washing machine makes trials at a low spin speed (such as 100 rpm) to distribute and measure the unbalanced load. The worst expected condition, unless you do not put a rigid object in the tumble such as a basketball, should be an unbalanced load about 600 grams (in a 5 kg capacity washing machine), and the machine is set to perform the maximum spin at 1000 RPM, with this unbalanced load.

Figure 6.1- Mechanical system of a front-load washing machine

For a 5 kg capacity front loading washing machine which is taken considered in this thesis; the total weight of the oscillating system (the tub, tumble, motor, and the balancing weight on the top) M would be about 39 kg. The system is hanged to the side walls by two identical coils and supported by two dampers from the bottom (Figure 6.4).

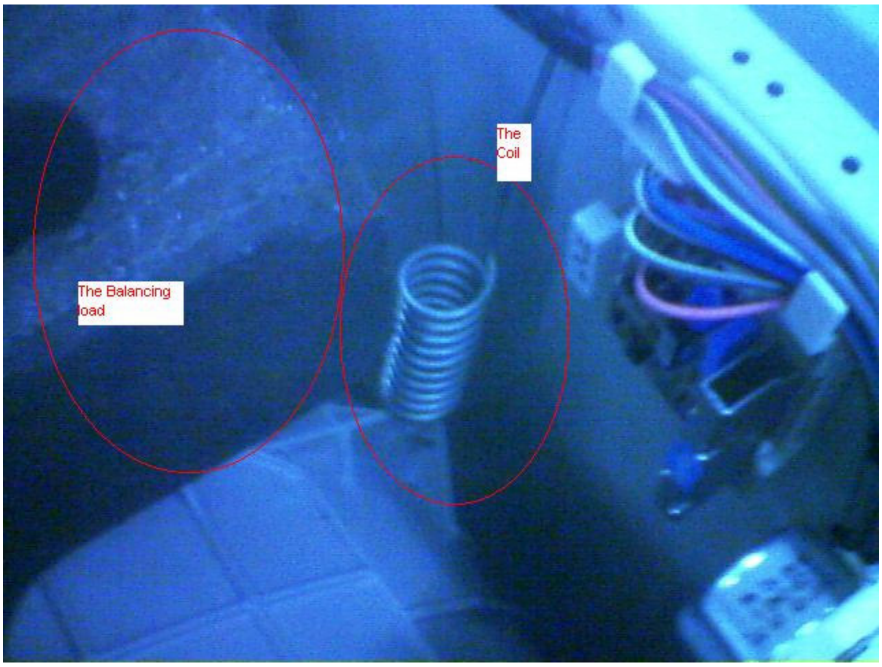

Figure 6.2- The coil and the balancing load of a washing machine

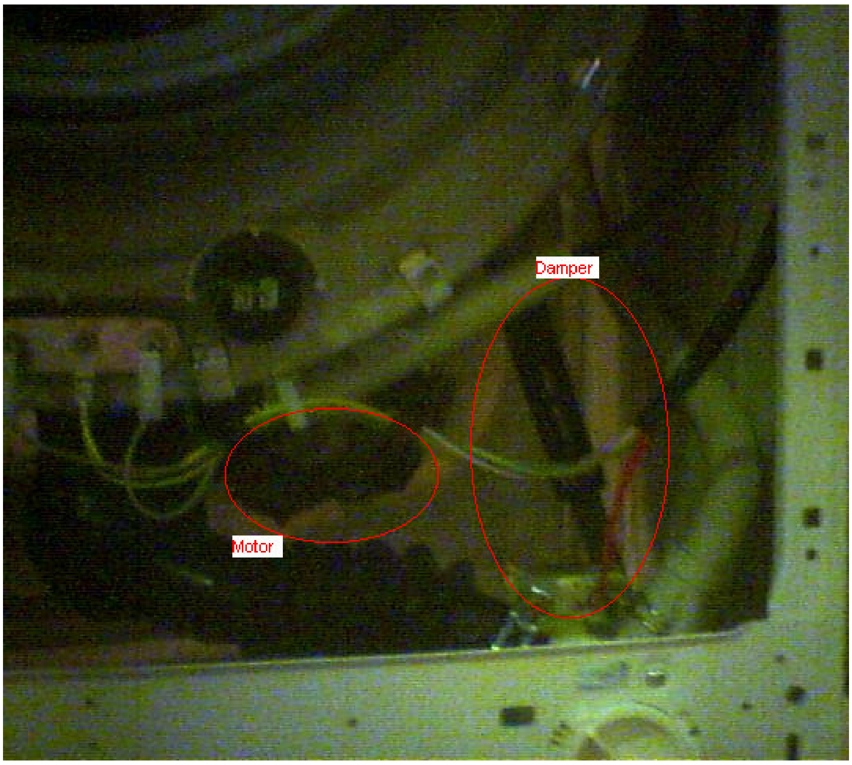

Figure 6.3- The positions of the motor and a damper in a washing machine

The parameters for the system with a spin speed of 1000 rpm are;

M=39 kg, w= 104.6 (1/s)

m=600 g,

k=5.7 N/mm,

c=70 N/(mm/s),

r=228 mm (the radius of the tumble),

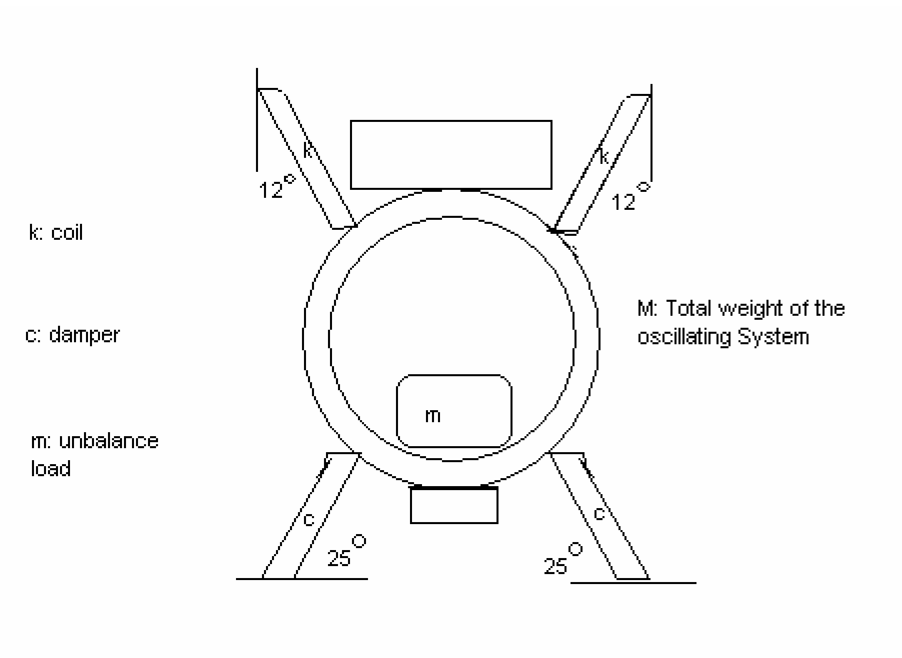

Figure 6.4 –The oscillating system of a front load washing machine. k represents the coils and c represents the supporting dampers. M is the total weight of the system while m is the unbalanced load.

First we will examine the forces which effect the system and the displacement y on y-axis. (Figure 6.5) We can draw the forces on the y-axis for this consideration.

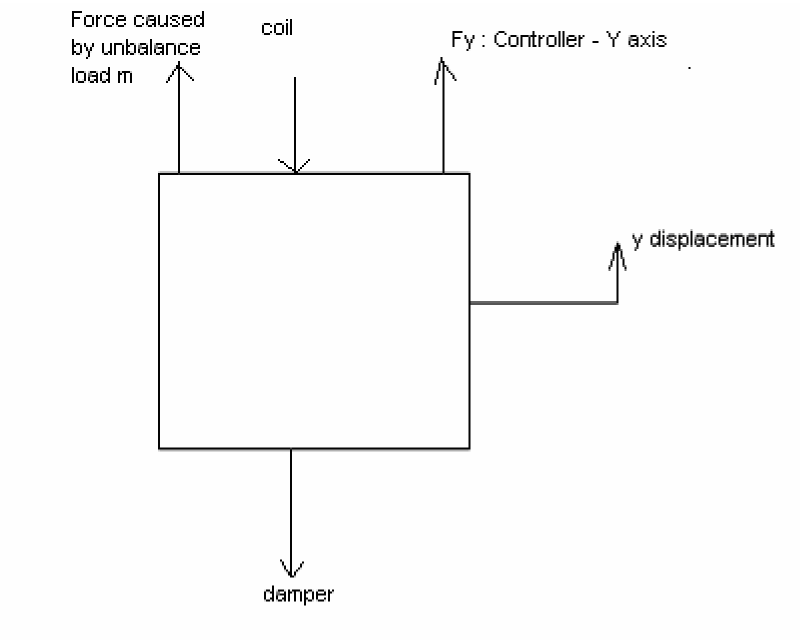

Figure 6.5 – Forces affecting the oscillating system on y-axis.

Y-axis equivalent of the k will be 11.1 N/mm from the coils and the y-equivalent for dampers is 57 Ns/mm

The desired motion on y axis is $y_d = 2\sin(wt)$ [mm] (6.1)

So we can write;

$$M\ddot{y} + c_y\dot{y} + k_y y = mwr^2 \sin(wt) + F_y \quad (6.2)$$

and;

$$\ddot{y} = -\frac{c_y}{M}\dot{y} - \frac{k_y}{M}y + \frac{m}{M}wr^2\sin(wt) + \frac{1}{M}F_y \quad (6.3)$$

$$e = y - y_d = y - 2\sin(wt) \quad (6.4)$$

$$s(y,t) = -(k_1 e + k_2\dot{e}) = -ke \quad (6.5)$$

$s(y,t)\cdot\dot{s}(y,t) \le -n|s|$, n>0 (Sliding condition) (6.6)

$$|d(t)| \le D$$

$$\left|\frac{m}{M} wr^2 \sin(wt)\right| \le D \qquad (6.7)$$

$$F_y = M(ke + \frac{c_y}{M}\dot{y} + \frac{k_y}{M} y - (D+n)\operatorname{sgn}(s) + y_d^{(2)}) \qquad (6.8)$$

So the y-axis behavior of our controller is given above in (6.8).

Now let's take a look at the forces which effect the system and the displacement x on x-axis (Figure 6.6).

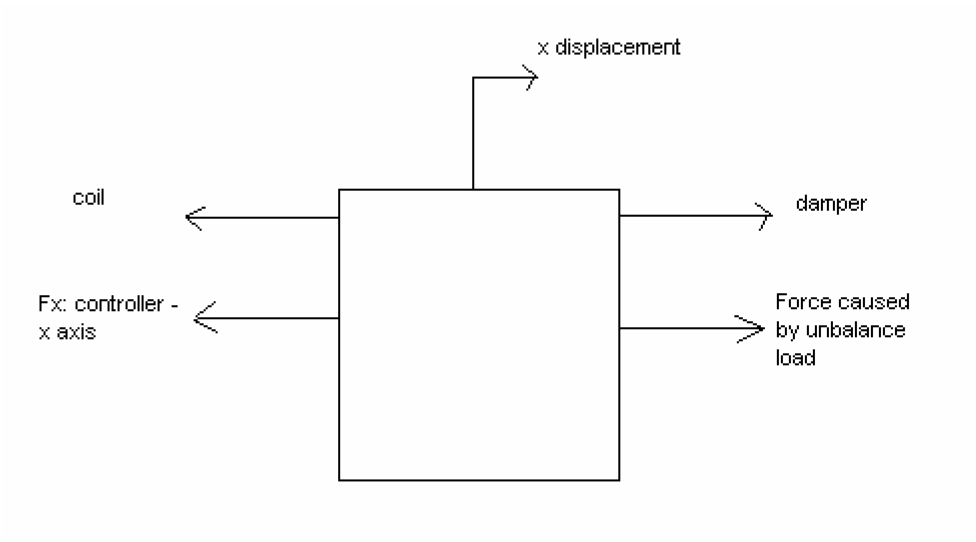

Figure 6.6 – Forces effecting the oscillating system on x-axis.

X-axis equivalent of the k will be 2.4 N/mm from the coils and the x-equivalent for dampers is 128 Ns/mm

The desired motion on x axis is $x_d = 2\cos(wt)$ [mm] (6.9)

So we can write;

$$c_x\dot{x} + k_x x = mwr^2\cos(wt) + F_x \tag{6.10}$$

and;

$$\dot{x} = -\frac{k_x}{c_x}x + \frac{m}{c_x}wr^2\sin(wt) + \frac{1}{c_x}F_x \tag{6.11}$$

$$e = x - x_d = x - 2\cos(wt) \tag{6.12}$$

$$s(x,t) = -(k_1 e + k_2\dot{e}) = -ke \tag{6.13}$$

$$s(x,t)\cdot\dot{s}(x,t) \le -n|s|,\ n>0 \text{ (sliding condition)} \tag{6.14}$$

$$|d(t)| \le D$$

$$\left|\frac{m}{c_x}wr^2\sin(wt)\right| \le D \tag{6.15}$$

$$F_x = c_x(ke + \frac{k_x}{c_x}x - (D+n)\operatorname{sgn}(s) + x_d^{(1)}) \tag{6.16}$$

(6.16) is x-axis behavior of our controller.

Now these values will be implemented in the MATLAB Simulink. The diagram of the system in simulink environment is as given below. We can separately see the x-axis and y-axis behavior of the system while we can see the real motion of the oscillating system on an x-y graph.

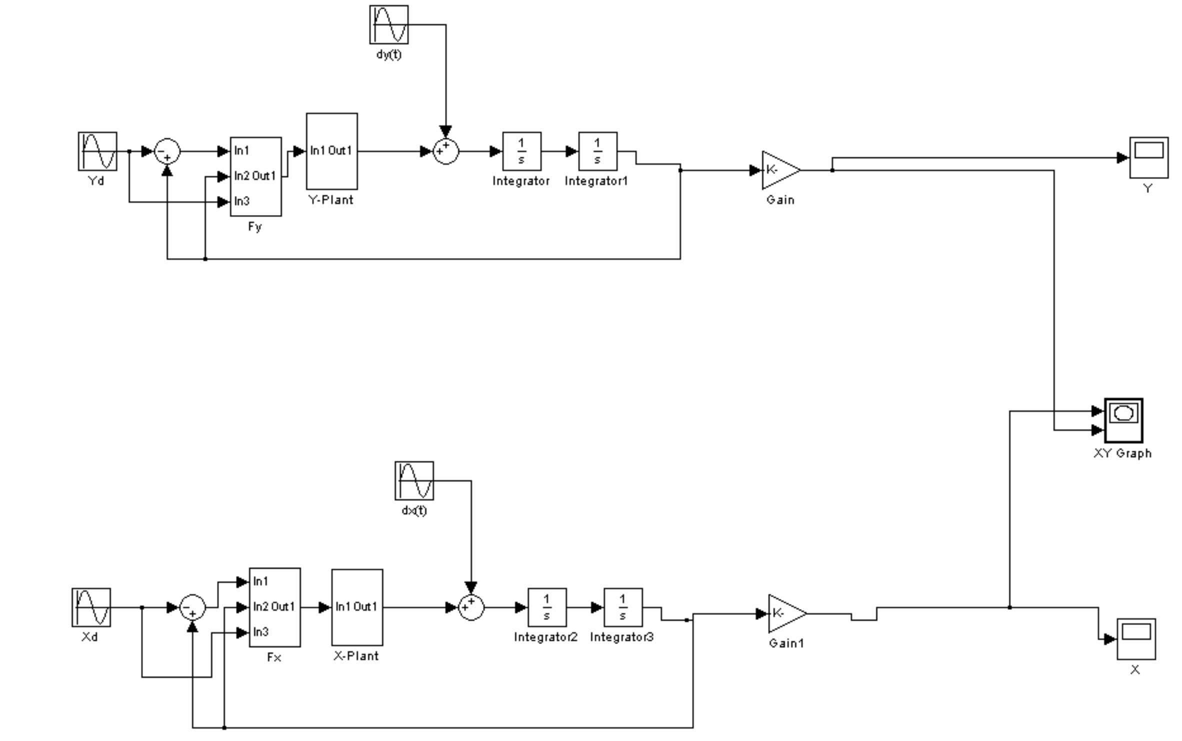

Figure 6.7 – The Simulink diagram of the system with Fuzzy SMC

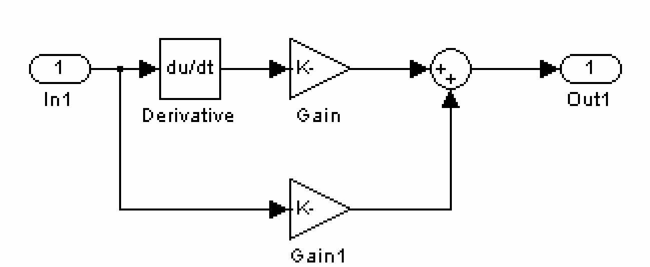

Figure 6.8 – The Simulink diagram for obtaining the displacement on y-axis (The block "Y-Plant" in Figure 6.7)

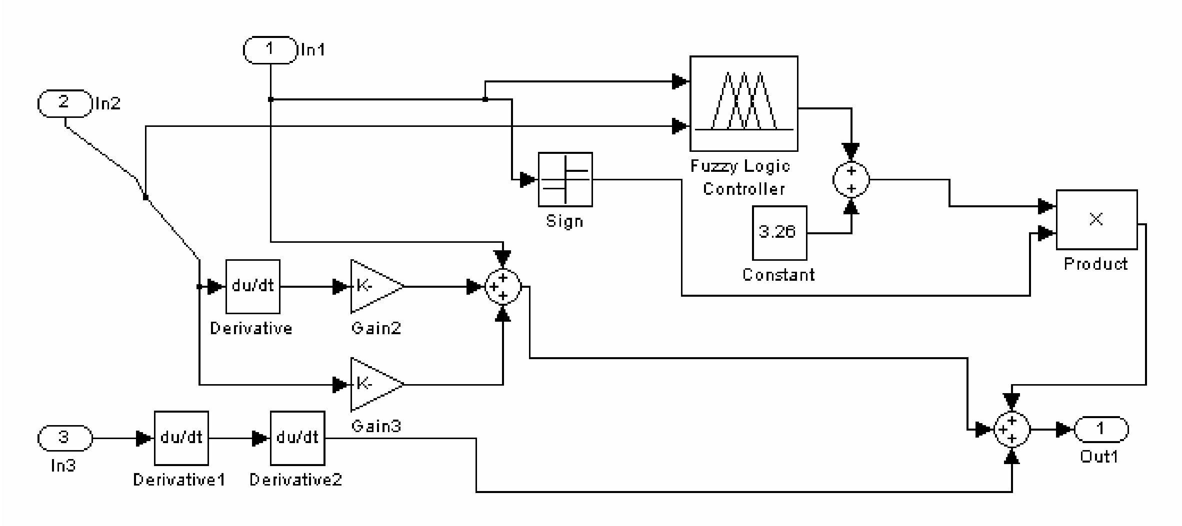

Figure 6.9 – The Simulink diagram of the controller used on y-axis (The block "Fy" in Figure 6.7)

The sub-system details of y-axis are given in Figure 6.8 and Figure 6.9. These are the simulink translations of the controller equation (6.8) which is found using AFSMC methods.

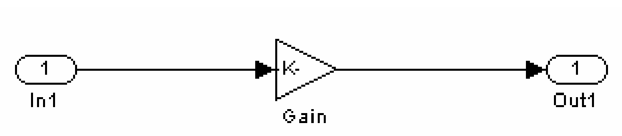

Figure 6.10 – The Simulink diagram for obtaining the displacement on x-axis (The block "X-Plant" in Figure 6.7)

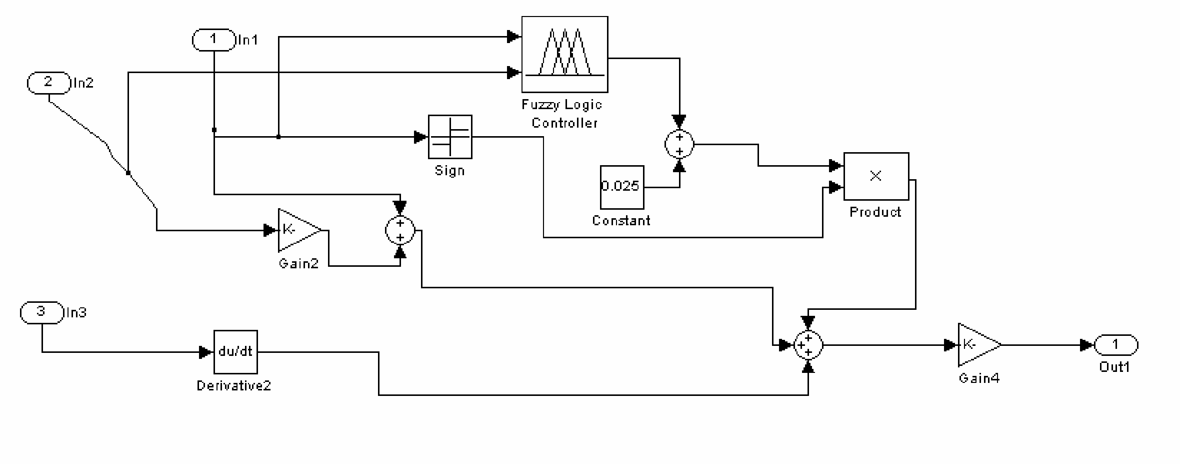

Figure 6.11 – The Simulink diagram of the controller used on x-axis (The block “Fx” in Figure 6.7)

The x-axis sub-system details are given in Figure 6.10 and Figure 6.11 just like the equation (6.16) which is found out using AFSMC method.

The membership functions for controllers on both y-axis and x-axis are given in the same order in Figure 6.12 and Figure 6.13. In Figure 6.14 the fuzzy control surfaces for the both of these controllers are also given.

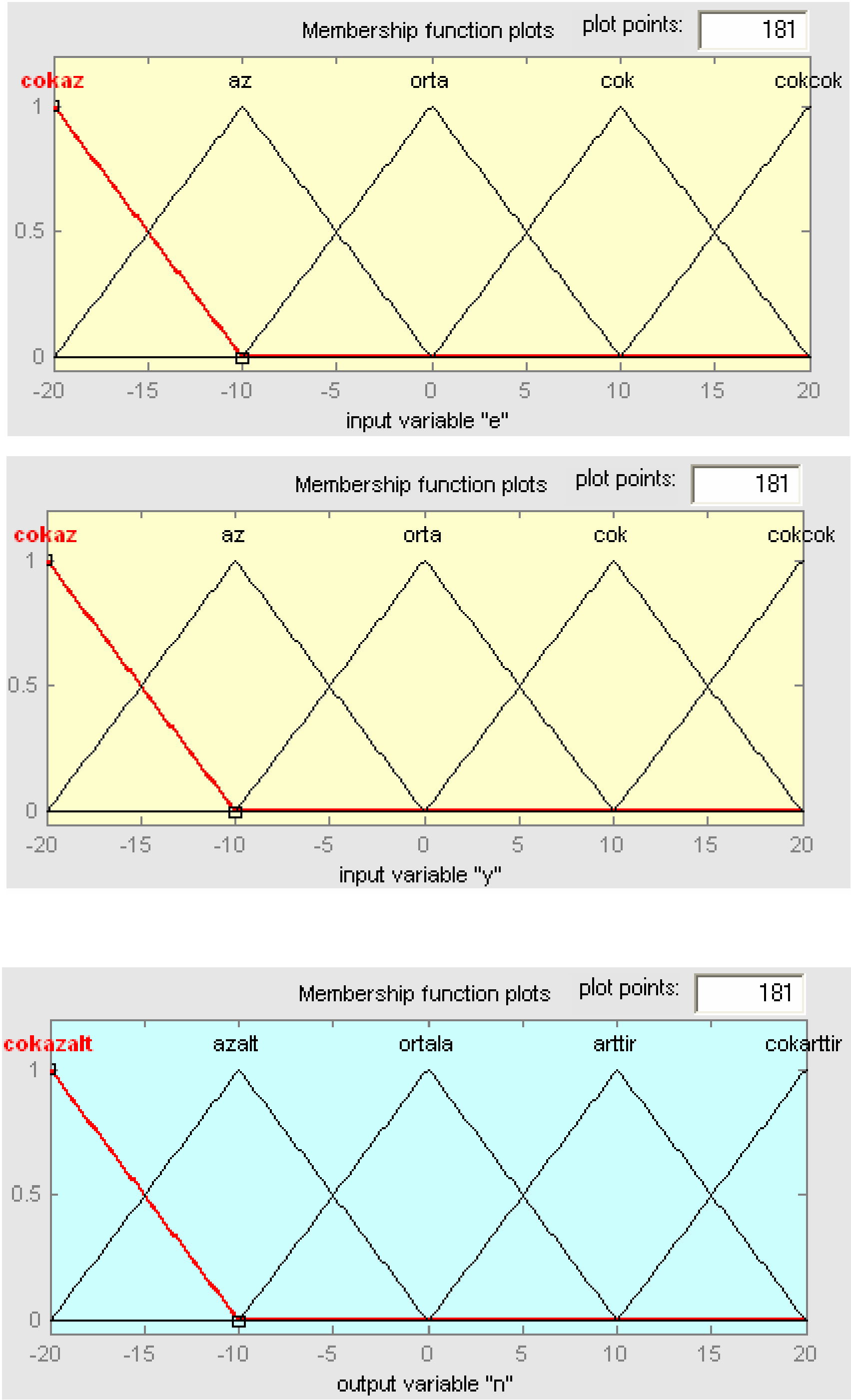

Figure 6.12 – The membership functions for fuzzy controller on y-axis

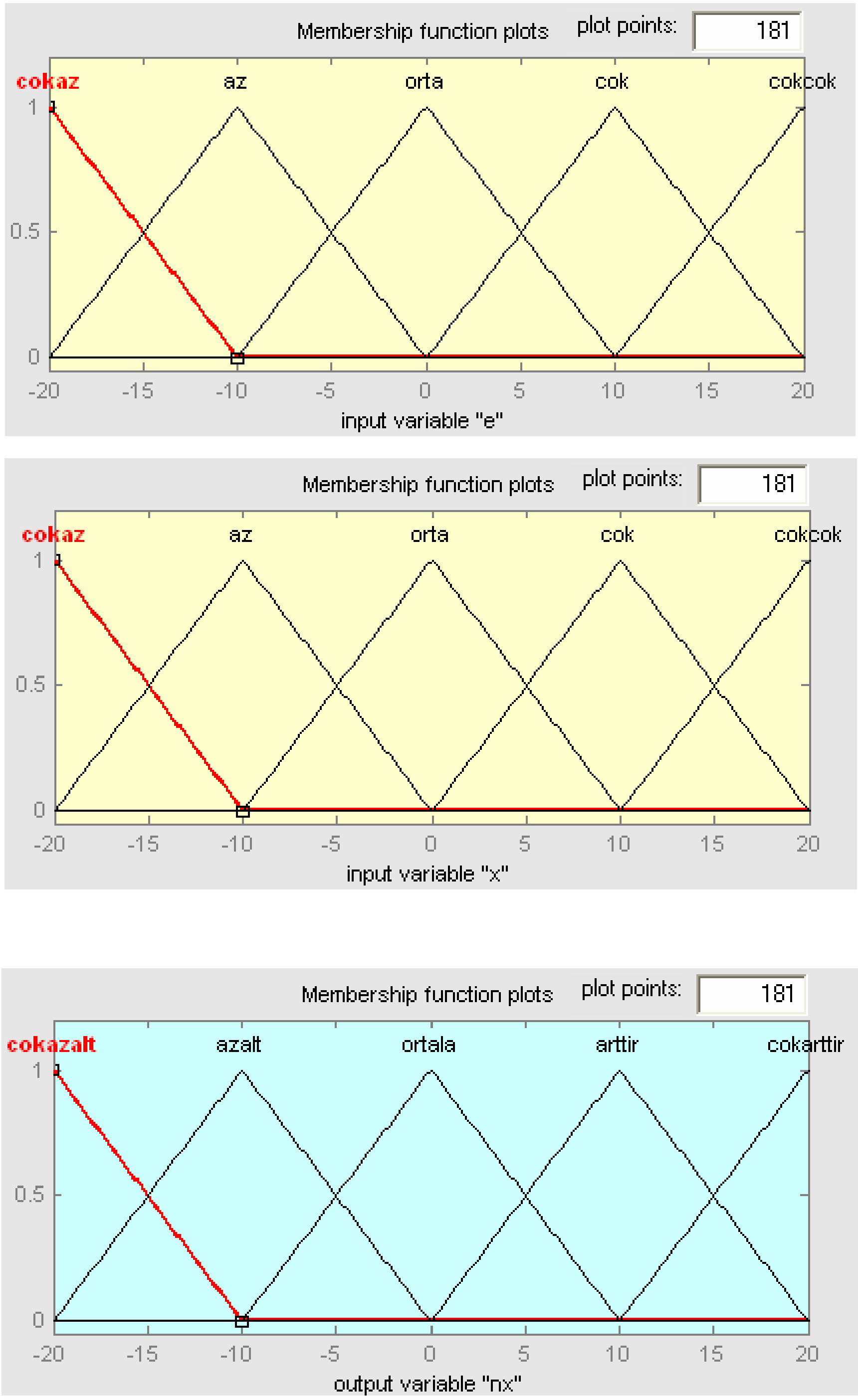

Figure 6.13 – The membership functions for fuzzy controller on x-axis

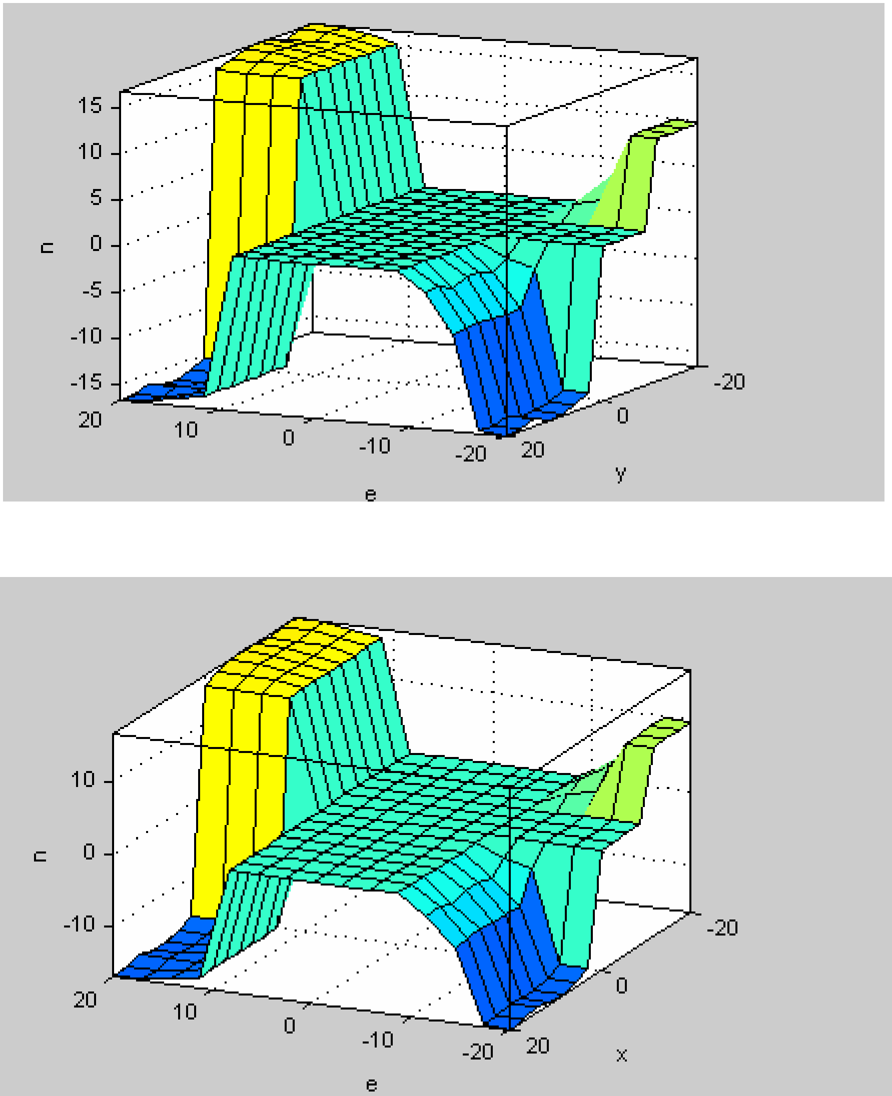

Figure 6.14 – Fuzzy surfaces for switching parameter n for y-axis and the x-axis in the order given above.

Before any controller is applied to the system: x-y displacement of the system can be observed in Figure 6.15. We can see a large overshoot of displacement in the oscillation.

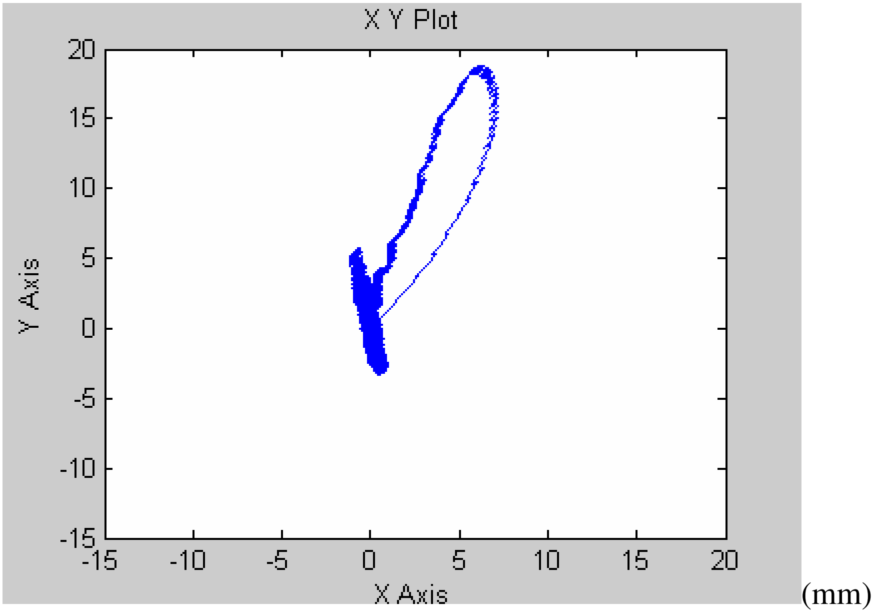

Figure 6.15 – The system behavior before any controller is applied

Just to see the difference and also to make a comparison let us just apply a positive feedback loop to the system and observe the changes in the behavior of the x-y displacement graph (Figure 6.16). We can say that the system's displacement is reduced in between smaller thresholds after a basic closed-loop feedback gain is applied. And now we can say that the oscillation has a character and the overshoots are reduced.

The simulation is continued by the application of the adaptive fuzzy sliding mode control as seen in Figure 6.7. The displacement of the system after the fuzzy sliding mode control is applied is as given in Figure 6.17.

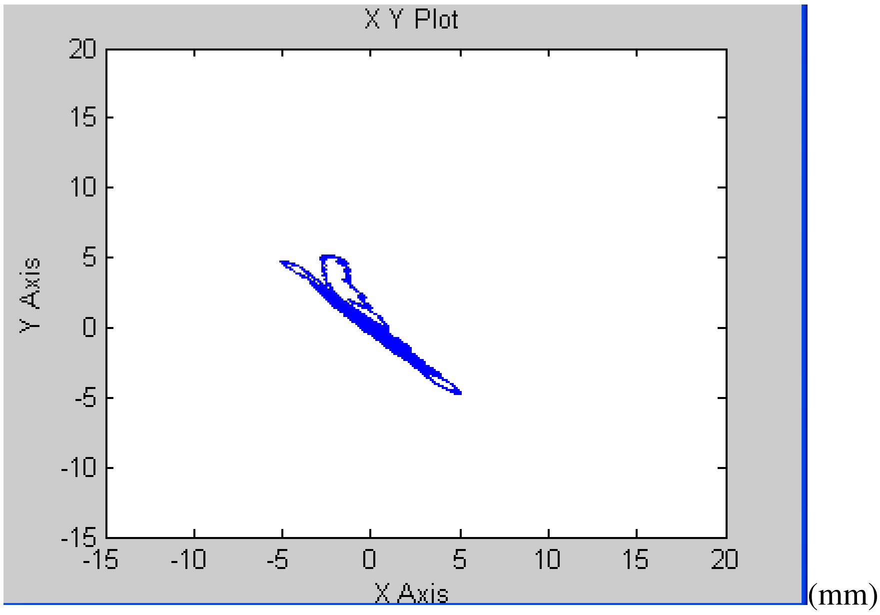

Figure 6.16 – The system behavior after a simple feed-back is applied

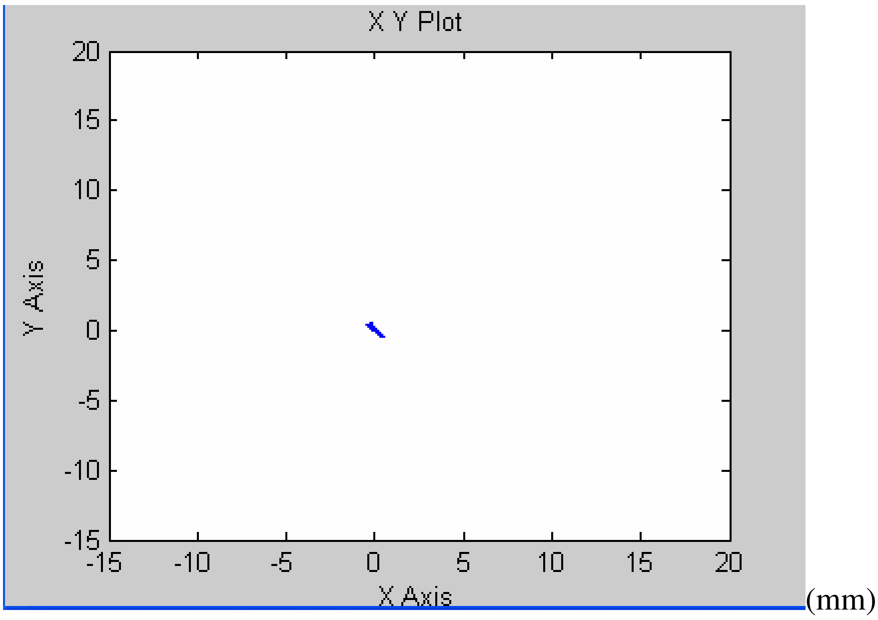

Figure 6.17 – System with the Fuzzy SMC

If we zoom on the x-y plot we can see the displacement is minimized and is much more like the expected x and y displacement functions (Figure 6.18).

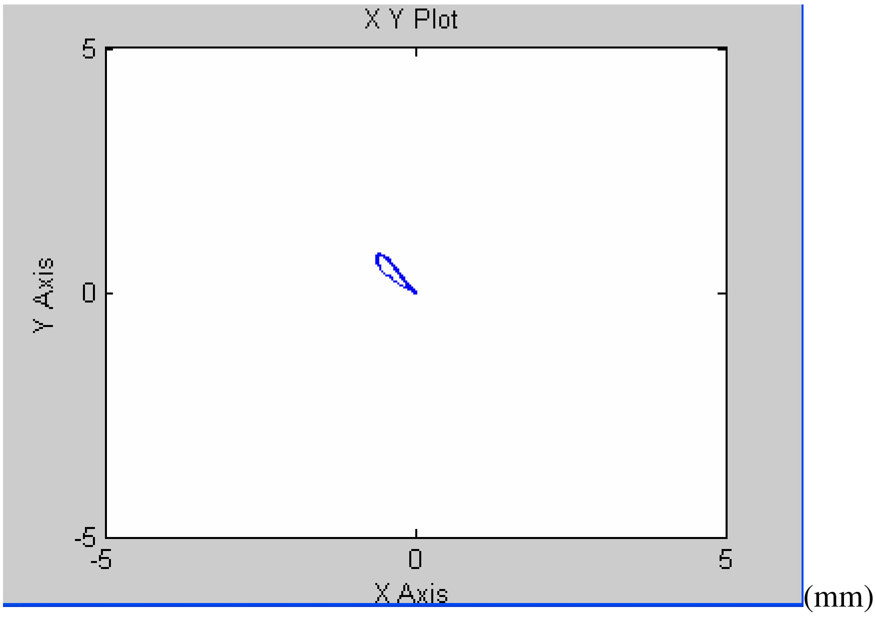

Figure 6.18 – Zoomed in version of Figure 27

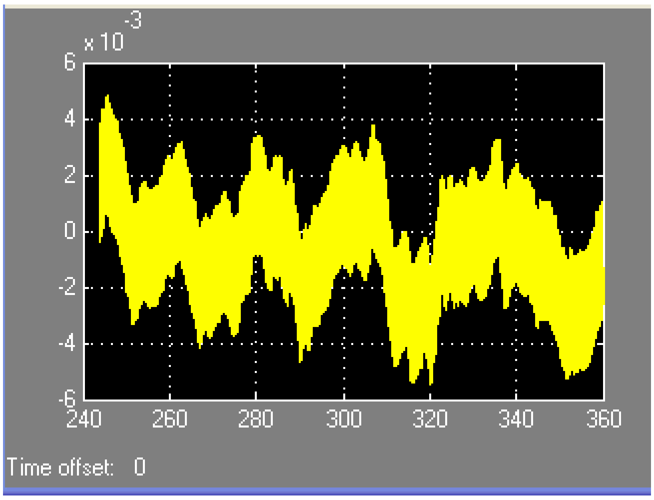

Figure 6.19 - Tracking error y (m)

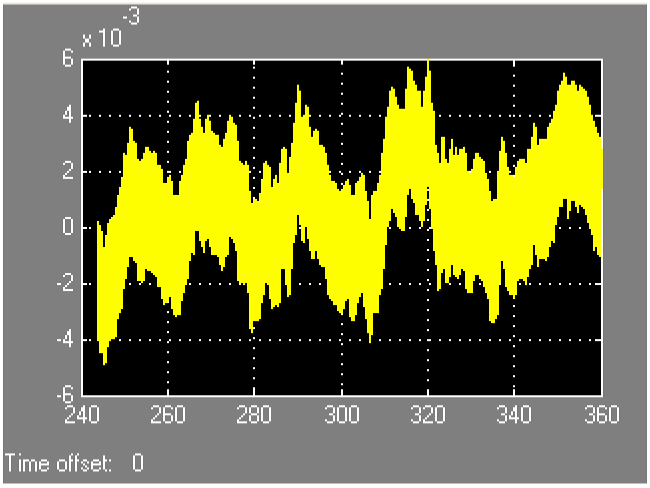

Figure 6.20 - Tracking error x (m)

It is seen the error is in values such -3^{rd} rd power of tens at the end of the 6 minute simulation. Even though there is a chattering noise over both tracking errors of x and y axis's, it may be reduced by a tuning in the fuzzy rules for Fx and Fy controllers.

7. CONCLUSION

In this thesis, adaptive fuzzy sliding mode control and its implementation on washing machine suspension system is discussed. After a brief introduction to all supplementary work, a general overview about sliding mode control is given. Then, a general look is taken at fuzzy sets and systems. Two methods are mentioned for the application adaptive fuzzy sliding mode control in this thesis in Section 4. After, information on front-load washing machines, their spin profiles and unbalanced load distribution algorithms. Finally, simulation studies are held in Section 6 with the oscillating system of a front-load washing machine.

In simulation we have seen that our oscillating system is moving in safer distances than before we have applied the adaptive fuzzy SM controllers on both axises. The comparisons of cases;

- no controller applied
- a single close loop gain applied on each axis
- Application of AFSMC on both axises

has showed us that with AFSMC we can have much better results than conventional control techniques. This means that the risk for decreases in the mechanical system’s life is minimized and also the risk of the machine movement is minimized. This is an important outcome so both the customer will be more satisfied by the product, because it has a lower risk to move; while the reliability of the product is and the product life-time are increased.

Works which are held in the appliance market at R&D level about unbalanced load problem are mostly about measuring the weight of the oscillating system with the dampers with weight sensors and changing the frictional coefficient of the damper. In this thesis it is proposed to add another controller as a different component between the body and the oscillating system to prevent the over shoots in displacement of the oscillating system. There is not any washing machine model sold out yet with any of the features mentioned above.

APPENDIX

The spin profile of the considered machine is given below in Control Works Compact pseudo-code editor.

Step	Instructions	Parameters
1	SET_USER_SELECTION_ON	
2	WAIT_DOOR_LOCKED	
3	SET_UNBALANCE	6
4	PJUMP	6
5	INFO_TIME_PROFILE	
6	CALL	UNB
7	TEST_UNBALANCE	2
8	PJUMP	12
9	PJUMP	34
10	WAIT_MOTOR_STOP	
11	NJUMP	5
12	RAMP	100/400/150
13	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	10
14	WAIT_MOTOR_STOP	
15	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	60
16	SET_UNBALANCE	6
17	PJUMP	19
18	INFO_TIME_PROFILE	
19	CALL	UNB
20	TEST_UNBALANCE	1
21	PJUMP	37
22	PJUMP	25
23	WAIT_MOTOR_STOP	
24	NJUMP	18
25	TEST_UNBALANCE	2

Step	Instructions	Parameters
26	PJUMP	47
27	PJUMP	28
28	SET_UNBALANCE	6
29	PJUMP	31
30	INFO_TIME_PROFILE	
31	CALL	UNB
32	TEST_UNBALANCE	2
33	PJUMP	47
34	PJUMP	54
35	WAIT_MOTOR_STOP	
36	NJUMP	30
37	RAMP	100/600/62,5
38	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	52
39	RAMP	600/700/12,5
40	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	112
41	RAMP	700/1000/37,5
42	RAMP	1000/1100/50
43	RAMP	1100/1200/50
44	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	82
45	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	60
46	PJUMP	51
47	RAMP	100/600/62,5
48	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	52
49	RAMP	600/800/25
50	LAUNCH_T1_AND_WAIT_T1_OR_IF_T2	272
51	WAIT_MOTOR_STOP	
52	CALL	RUNB
53	CALL	RUNB
54	WAIT_MOTOR_STOP	

Step	Instructions	Parameters
55	SET_USER_SELECTION_OFF	
56	EXIT	

The sub-profile UNB is given below;

Step	Instructions	Parameters
1	STOP_AND_SET_DIR_CW	
2	LAUNCH_T1_AND_WAIT_DIR_AND_T1_OR_IF_T2	3
3	SPEED	52
4	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	5
5	SPEED	65
6	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
7	SPEED	80
8	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
9	STOP_AND_SET_DIR_CCW	
10	LAUNCH_T1_AND_WAIT_DIR_AND_T1_OR_IF_T2	3
11	SPEED	52
12	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	5
13	SPEED	65
14	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
15	SPEED	80
16	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
17	STOP_AND_SET_DIR_CW	
18	LAUNCH_T1_AND_WAIT_DIR_AND_T1_OR_IF_T2	3

Step	Instructions	Parameters
19	SPEED	52
20	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	5
21	SPEED	65
22	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
23	SPEED	80
24	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	4
25	CALL	RAMP
26	RETURN	1

The sub-profile RUNB is as given below;

Step	Instructions	Parameters
1	STOP_AND_SET_DIR_CW	
2	LAUNCH_T1_AND_WAIT_DIR_AND_T1_OR_IF_T2	3
3	SPEED	52
4	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	5
5	STOP_AND_SET_DIR_CCW	
6	LAUNCH_T1_AND_WAIT_DIR_AND_T1_OR_IF_T2	3
7	SPEED	52
8	LAUNCH_T1_AND_WAIT_ T1_OR_IF_T2	5
9	RETURN	1

REFERENCES

[1] Yoo B., Ham W., "Adaptive Fuzzy Sliding Mode Control of Nonlinear System", *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, May 1998

[2] Hsu Y., Chen G., Li H., "A Fuzzy Adaptive Variable Structure Controller with Applications to Robot Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 31, no. 3, June 2001

[3] Ramot D., Milo R., Friedman M., Kandel A., "On Fuzzy Correlations", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 31, no. 3, June 2001

[4] Wang J., Chan P., Rad A., "Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching", *Fuzzy Sets and Systems* vol. 122 (2001)

[5] Wang J., Chan P., Rad A., "Indirect adaptive fuzzy sliding mode control: Part II: parameter projection and supervisory control", *Fuzzy Sets and Systems* vol. 122 (2001)

[6] Zhang T., "Stable adaptive fuzzy sliding mode control of interconnected systems", *Fuzzy Sets and Systems* vol. 122 (2001)

[7] Vukovic P., "One-step Ahead Predictive Fuzzy Controller", *Fuzzy Sets and Systems* vol. 122 (2001)

[8] Li T., Shieh M., "Switching-type fuzzy sliding mode control of a cart-pole system", *Mechatronics* vol. 10 (2000)

[9] Liang C., Su J, "A New Approach to Design a Fuzzy Sliding-mode Controller", *Fuzzy Sets and Systems* vol. 139 (2003)

[10] Chen C., Chang M., "Optimal Design of Fuzzy Sliding Mode Control: A Comparative Study", *Fuzzy Sets and Systems* vol. 93 (1998)

[11] Choi B., Kwak S., Kim B., "Design of a Single Input Fuzzy Logic Controller and its Properties", *Fuzzy Sets and Systems* vol. 106 (1999)

[12] Huang S., Lin W., "Adaptive Fuzzy Controller With Sliding Surface for Vehicle Suspension Control", *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, August 2003