Adaptive Control of Harmonic Drive Motors with Parameter Varying Friction using Structurally Dynamic Wavelet Networks

Seyyed Alireza Tadayoni

A Thesis
In The Department of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

March 15, 2006

Your file Votre référence
ISBN: 978-0-494-20764-2
Our file Notre référence
ISBN: 978-0-494-20764-2

Abstract

Adaptive Control of Harmonic Drive Motors with Parameter

Varying Friction using Structurally Dynamic Wavelet Networks

Seyyed Alireza Tadayoni

The many advantages of harmonic drives motors such as compactness, high gear ratio, low backlash, light weight, and high torque capacity has resulted in their wide spread usage in precision control applications. However, the nonlinearities of harmonic drives including hysteresis, kinematic error, position dependent friction, and flexibility make it difficult to develop control systems that achieve precise tracking performance. In this thesis, a new approach for adaptive control of harmonic drive motors is developed using a structurally dynamic wavelet neural network to achieve accurate tracking in the presence of parameter varying friction. Furthermore, a new fuzzy logic approach is proposed for dynamic addition and removal of wavelet nodes that achieves accurate tracking using a minimum number of nodes. Experimental verification of the proposed method indicates that it can achieve precise tracking performance with a significantly smaller number of nodes than existing approaches.

Acknowledgments

This thesis was carried out in the Control and Information System Laboratory (CIS) at Concordia University, Montreal, Canada. I am grateful for having the opportunity to work and learn with the group of people from all corners of the world.

I would like to thank my advisors, Dr. Brandon Gordon and Dr. Wenfang Xie, as they support me financially and intellectually.

To My Wife

&

To those who have brought us where we are...

Table of Contents

List of Figures

List of Tables

Nomenclature

K_b	DC motor voltage constant	V/rps
K_m	DC motor torque constant	Nm/A
R	Motor Resistance	Ω
K_s	Spring showing the stiffness of flexspline	Nm/rad
T_m	Motor side friction torque	N.m
T_{ext}	External torque applied to the motor shaft	N.m
T_l	Load side friction torque	N.m
J_m	Motor side momentum of inertia	$Kg.m^2$
J_l	Load side momentum of inertia	$Kg.m^2$
u	The DC motor voltage input	V
r_g	The gearbox ratio.	-
q_m	The motor side rotational position	Rad
q_l	The load side rotational position	Rad
$\bar{J}_p$	Upper bound for J_p	-
$\bar{C}_i$	Upper bound for C_i	-

1 Introduction

1.1 Motivation

Developed in 1955 primarily for aerospace applications, harmonic drives are high-ratio compact torque transmission systems [1]. Because of their great advantages harmonic drives have captured more and more researchers' attention in the last decades. Harmonic drive gearing earned its reputation as a high performance speed reducer from its ability to provide precise positioning. The high degree of precision has been one of the outstanding advantages of the harmonic drive while others are having high-ratio, high torque transmissibility, compactness, near zero backlash, and concentric geometry. These have given them a key position between control engineers. However, harmonic drives have some disadvantages which make them difficult to be controlled. Harmonic drive transmissions employ a flexible gear for speed reduction, called flexspline. Flexspline is externally toothed, non-rigid (flexible), and thin-walled cylindrical cup which is enclosed by circular spline [3][4][5]. The flexspline makes the transmission stiffness to be lower than what is in conventional transmissions [1][2]. It also adds hysteresis, kinematics error, presliding displacement and friction to the list of harmonic drive's disadvantages which have motivated the control engineers to study the harmonic drives with more details.

All these cause a nonlinear relation between the input and the output torques and the precision positions control of the harmonic drives becomes more challenging to the control engineers.

Another undesirable effect of the flexspline is the friction. Friction in harmonic drives produces nonlinear dynamic behaviours especially at low velocities. When the system enters the

stick-slip regime the friction behaviour becomes much more complicated and much harder to overcome. In this regime many conventional controllers fail to work, fall into limit cycles, or have poor performance. This makes friction one of the challenging issues in the control of harmonic motors especially in low velocities.

In the recent past, several researches have been performed on modeling of harmonic drives starting from [1] which initiated the study of the harmonic drives by Russians. More recently Taghirad & Belanger [2][3][4][5] represent a detailed model for harmonic drive systems including hysteresis, compliance and an overall friction model. Taghirad modeled friction as a combination of linear dampers in different parts of a harmonic drive gear box with no consideration of nonlinear behaviours. Taghirad [6], Hsia[7], Legnani [8], Marilier [9], Chedmail et al. [10] and Seyfferth [33] attempted to model the stiffness, friction and position accuracy of harmonic drive systems.

Gandhi [34][35] also modeled main nonlinear attributes in harmonic drives such as kinematic error, hysteresis, and friction. For the first time he observed that the friction in a harmonic drive is position dependent. This position dependency can be however captured only in low velocities. In high velocities the measured friction is an average value on the whole motor round. Gandhi used Fourier series to model the position dependent friction while he only considered one revolution of the motor side [34][35]. Consequently, he missed addressing the load side friction dependencies.

One of the key notes of precision positioning systems is the slow motion. They are mostly working in the low velocity ranges. Consequently a proper model which captures the position dependency of friction in harmonic drives becomes a key point in designing a proper controller to have an acceptable performance for such systems. However a model which captures

all the nonlinear behaviours of friction in harmonic drives has not been presented yet. In this thesis friction in harmonic drives is studied in more details and a proper model is proposed.

In this thesis, position dependency of friction in harmonic drives is addressed in the cases where the harmonic drives work in the low velocity ranges, unless specified otherwise.

1.2 Friction Modelling

Friction occurs in all mechanical systems. It appears at the physical interface between two surfaces in contact. Friction was studied extensively in classical mechanical engineering. However there has lately been a strong resurgence. Apart from intellectual curiosity, it is driven by strong engineering needs in a wide range of industries particularly for new precise measurement techniques and precision positioning systems.

In the history of science there has been a vast research about friction modelling. Friction force is proportional to load, opposes the motion, and is independent of the contact area—this was all known to Leonardo Da Vinci in 1519. Da Vinci's friction model was rediscovered by Amontons (1699) and developed by Coulomb (1785). Morin (1833) introduced the idea of static friction and Reynolds (1866) introduced the equation of viscous fluid flow, completing the friction model that is most commonly used in engineering: the static + Coulomb + viscous friction model.

So far, about thirty friction models have been presented. Coulomb published the most comprehensive study of friction [20]. Lately Karnopp [18] and Stribeck [21] improved coulomb model in order to take stiction and stricbeck effect into account. Although these models bring a better understanding of the friction, but they are considered as "static" model because they did not consider the velocity dependency of friction.

Dahl (1968) concluded that for small motions, a junction in static friction behaves like a spring [12][19]. Dhal's model was developed for the purpose of simulating control system with friction especially in servo system with ball bearings. Based on the experimental studies made by Rabinowicz [22] and Dahl, the Dahl model was then improved. The new model took the hysteresis into account.

More recent researches on friction modeling reveal that the friction phenomenon demonstrates several nonlinear behaviours that must be considered in the modeling. A dynamic friction model was proposed by C. Canudas de Wit et al. in 1995 [23]. The model, called the LuGre model, captures most of the friction behaviour that has been observed experimentally. This includes the Stribeck effect, hysteresis, spring–like characteristics for stiction, and varying breakaway force. This model was compared with a standard simple kinetic friction model (KFM) by Friedhelm Altpeter et *al.* through a singular perturbation analysis [24]. They concluded that in a unidirectional motion and Fs=Fc (static & columb friction), KFM is sufficient to model the friction. However, if Fs ≠Fc, the LuGre model captures more dynamic properties. The LuGre model was also tested experimentally by Rafael Kelly [25]. C. Canudas de Wit's experimental results validated the LuGre friction model in an adaptive control scheme with friction compensation [26]. In his Ph.D. thesis, Prasanna S. Gandhi uses the LuGre model to identify the friction in harmonic drives [34][35].

In recent years, many other friction models have been proposed. Fitsum A. Tariku et al. [28] proposed two dynamic models for simulation of one-dimensional and two-dimensional stick-slip motion. Ruh-Hua Wu et al. [29] presented a modified Coulomb friction model integrating presliding displacement in the microsliding regime. Milos R. Popovic et al. [30] noticed that most friction models available describe friction only as a function of velocity.

However this is not always true. In many cases, friction is dependent on position; for example, with harmonic drives.

As stated by Milos R. Popovic et al. [30], apart from all the researches performed on the modelling of friction, friction dependency on the position or other system states has not been fully addressed yet. Friction in complex contacts, especially where it depends on the system states other than velocity, does not obey conventional models so much [35]. One of the best techniques to overcome this problem is using general function approximators which can approximate a wide category of functions. In this thesis a new approach to modelling friction in harmonic drives is proposed. This includes approximating friction by a structurally dynamic wavelet network. This idea will be proposed in the next chapters in more details.

1.3 Network Function Approximations

Using networks as a function approximator has become more popular recently. Network function approximators are networks with the function arguments as an input and the function output as the output. They try to reconstruct unknown functions as a summation of some nonlinear functions called nodes. Nodes can be configured and connected together to form layers.

The capability of the networks in function approximation is directly dictated by the number of layers, number of nodes and the type of the nodes. It is more common to use 3 layer networks in function approximation. It has been shown that an MLP (Multi Layer Perceptron) network, with a single hidden layer, can approximate any given continuous function on any compact subset to any degree of accuracy, providing that a sufficient number of hidden layer neurons are used [31][32]. However, in practice, the number of hidden layer neurons required

may be impractically large. So a key choice for a network is the proper type and number of nodes for approximation of a particular function. Common networks are based on a wide class of nodes including simple implicit polynomials to complex nonlinear nodes of ANNs (Artificial Neural Networks), MLPs (Multi Layer Perceptrons), RBNs (Radial Basis Networks) [43][44][45], and WNs (Wavelet Networks) [46][47][48][49]. All node types have good capabilities in approximation of smooth functions. However not all networks work fine for the sharp functions with discontinuities.

Based on the application, function approximation networks work either online or offline. Offline approximations have been widely studied and used. In the offline category the network is first learned and then used for the approximation. In the online approximation, the learning process and using of the network are performed simultaneously. In the control field they are used in conjunction with adaptive controllers and take advantage of the estimators to find the weights of their nodes. Online network approximators still have issues which have not been addressed yet.

Although online versions of RBNs and WNs have been developed and studied [45][52][53] recently; but one of the key notes about these networks is the static configuration of their structure. A structurally dynamic network with the target of decreasing the number of nodes can minimize the number of unknown parameters (minimizing the number of states of the system), and improve the tracking performance and robustness of the controller considerably.

1.4 Structurally Dynamic Network Approximations

Structural adaptation is a method for achieving better performance by dynamically changing the structure of a network or controller. Structurally dynamic networks have not been

fully studied by the researchers mainly because of their great complexity. Sanner and Slotine [49] proposed an algorithm for structural adaptation of a wavelet network in conjunction with an adaptive controller. In the Sanner's algorithm nodes are being selected for adding or removal based on a threshold on the magnitude and the rate of the change of their weights. Nodes whose weights are more than the threshold or increasing in time are kept in the network; and conversely, those nodes whose weights are lower than the threshold and decreasing in time are removed [49]. This kind of structural adaptation method may work fine with the simulations but in real systems that there exist sensor noise and model uncertainties it may easily fail to function and cause large tracking error or even instabilities in the adaptive component of the controller. This causes the system to leave the working range and activate the sliding component of the controller which pushes the system back to its working range. However obviously this means very poor tracking performance.

Sanner's method has three drawbacks: First the structural adaptation is fast; second the network is changing too much in each step; and third finding the best threshold is rather difficult. From the switching control point of view, this method is a fast switch between many too different controllers. With the presence of noise this kind of controller can easily fall into large amplitude chattering or become unstable. However, the instability takes place when the adaptive component of the controller is active and causes the system to leave its working range A_d. Then the sliding component forces the system back to A_d and keeps the system stable. The outcome of this kind of stability is a very poor tracking performance with error bound as large as the whole A_d.

The Sanner's approach drawbacks will be discussed in more details in chapter 5. To conquer them a new fuzzy structural adaptation scheme is proposed which significantly improves

the system performance. The fuzzy structural adaptation algorithm will be further discussed more in chapter 5.

1.5 Adaptive Control

Adaptive control has been an active topic since last decade. It is a control method which deals with parameter uncertainties in a system. Friction compensation is one of the challenging problems in precision positioning application. During last years different friction models have been developed to take advantage of adaptive controllers to predict the unknown parameters in the friction formulations. Friedland [58] studied the simplest model as a Coulomb friction opposing the direction of the motion. Juang [59] introduced an adaptive controller to estimate the coefficients of friction in the LuGre model. His model includes the static and Coulomb frictions as well as the Stribeck effect. Hwang [56] proposed an adaptive controller for the friction function in the form of a Fourier neural network. Canudas [57] studied the adaptive friction compensation of systems with position or velocity dependent friction. He modeled the friction with two different formulations: a polynomial and a nonlinear function in the form of a summation of Gaussian functions. Johnson [37] studied an adaptive controller to compensate friction with the formulation of a viscous friction. He also considered an additional term as an average of position (or other states) dependent friction

Several other researchers have also proposed different adaptive controllers using different formulations for the friction function. But most of their research was concerned with friction in servo motors and simple contacts. However, adaptive controllers which are based on the friction formulation in complex contacts, such as in a harmonic motor drive, have not been fully studied yet.

Other types of controllers have not been used in compensation of friction in motors too much. Taghirad [36] studied an H_∞ torque controller to compensate friction in harmonic drives. Han [38] studied a robust nonlinear H_2 / H_∞ controller for a parallel inverted pendulum with dry friction. Chen [39] proposed a robust friction controller for robotic manipulators. He studied the friction with the formulation of columb+stribeck and an additional term for parametric uncertainties. Cai [40] studied the stick-slip friction in robot manipulators. He proposed a robust controller using the Karnopp [18] friction model. Vega [41] introduced a chattering-free adaptive sliding mode controller for robot arms. He used the LuGre model with a dynamical time base generator gain as the friction model. Bona [42] used a robust hybrid controller for friction compensation in robots using the friction model proposed by Canudas [57]. There are other researches proposing controllers for friction compensation in various systems. But the key common point of them is the use of the simple friction models proposed for simple contacts.

1.6 Thesis Contributions

The main contributions of this thesis are:

1) **Adaptive Control of Harmonic Drive Motors using Structurally Dynamic Wavelet Networks**

   As described before several attempts were made to model the friction in harmonic drives. In this research the friction in harmonic drives is modelled by a structurally dynamic approximation network. A model reference adaptive controller is also proposed to control the system. The controller takes the advantage of an estimator to estimate the unknown parameters of the system, mainly the weights of the friction

approximation network. This is the first time that this approach has been applied for control of harmonic drive motors with parameter varying friction.

2) **A New Fuzzy Logic Algorithm for Dynamically Varying the Structure of the Wavelet Network**

A new fuzzy based algorithm is proposed to dynamically change the structure of an approximation network with the target of decreasing the number of nodes participated in the network. The fuzzy algorithm is composed of two parts which are fully discussed in Chapter 5.

3) **Experimental Verification**

The methods proposed in this thesis are verified on the real system shown in Figure 11. The experimental results are shown and compared with the previous approaches to friction approximation and structurally dynamic networks.

1.7 Thesis Overview

An introduction to the thesis is proposed in chapter 1. In chapter 2 the modeling of the robot and its components are addressed. In Chapter 3 different basis functions which can be used in function approximation networks are discussed. These include Fourier, RBF and Wavelets. An adaptive controller is then proposed in chapter 4 to control the system in a working range, while outside this range a sliding controller will be effective. The stability of the system is then studied in section 4.3. Chapter 5 addresses the structural adaptation process and gives detailed description about the fuzzy algorithm. In chapter 6 the experimental setup is described and the

proposed method is applied to the real system. The experimental apparatus is the manipulator depicted in Figure 11. This manipulator was previously used in [54] and [61]. The experimental results are then presented and different methods are compared. Finally the conclusion and future works are addressed in chapter 7.

2 Problem Statement

Friction is a problem in many applications and a critical problem in precision positioning systems. Because of the zero backlash in harmonic drives, they are widely used in precision positioning devices. However friction in harmonic drives, as in other systems, produces nonlinear dynamic effects, especially at slow velocities when the system enters the stick-slip regime. So in precision positioning applications a more precise model of friction in the components of a harmonic drive is needed. Gandhi [35] showed that the simple columb and viscous friction models are not adequate to accurately model the friction in harmonic drives, and more enhanced models which consider nonlinear position dependency of the friction are required. In this research, the friction in harmonic drives is modeled as the summation of a position dependent Coulomb and a velocity dependent viscose friction. The Coulomb friction function is modeled as a structurally dynamic wavelet network to take care of the nonlinear position dependency; while the velocity dependency of the viscose friction is found to be linear with an acceptable accuracy.

2.1 The Harmonic Drive Gear Box

Harmonic drives are special flexible gear transmission systems that have a non-conventional construction with teeth meshing at two diametrically opposite ends. Because of their unique construction and operation, they have many useful properties. Hence, they are widely used in precise positioning applications including wafer-handling machines in the semiconductor industry, space robots, lens grinding machines, and medical equipment.

However, these drives possess nonlinear transmission attributes that are responsible for transmission performance degradation.

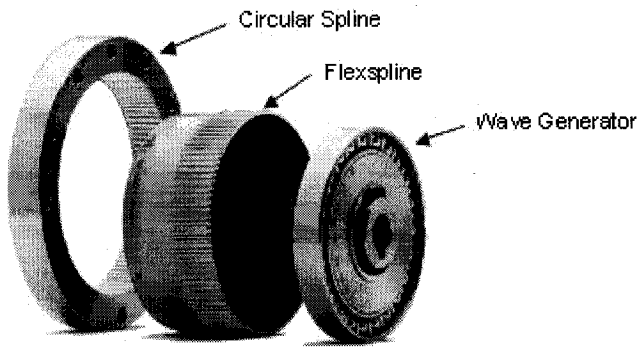

Figure 1. The Harmonic Drive Gear Components

A typical harmonic drive, illustrated in Figure 1, is composed of three components: Circular Spline, Flexspline, and Wave Generator [2][3][4][5][6]. The wave generator is an elliptical cam enclosed in an antifriction ball bearing assembly. It normally functions as the rotating input element. When inserted into the bore of the flexspline, it imparts its elliptical shape to the flexspline, causing the external teeth of the flexspline to engage with the internal teeth of the circular spline at two equally spaced areas 180 degrees apart on their respective circumferences, thus forming a positive gear mesh at these points of engagement. The externally toothed flexspline is a non-rigid or flexible, thin-walled, cylindrical cup which is smaller in circumference and has two less teeth than the circular spline. It is normally the rotating output element but can be utilized as the fixed, non-rotating member when output is through the circular spline. The circular spline is a thick-walled, rigid ring with internal teeth. It normally functions as the fixed or non-rotating member but can, in certain applications, be utilized as a rotating output element as well.

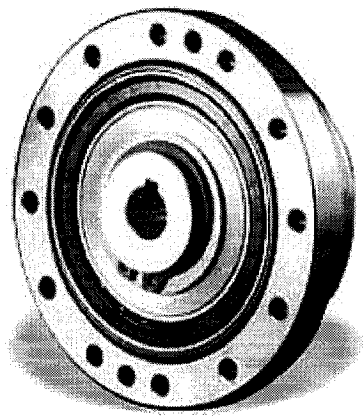

Figure 2. Fully Assembled Harmonic Drive Gear

A fully assembled harmonic drive is shown in Figure 2. In the most common speed reduction configuration, the wave generator is the input port, the flexspline is the output port, and the circular spline is immobile.

Figure 3 illustrates the operation of the harmonic drive in the most common configuration. The teeth on the non-rigid flexspline and the rigid circular spline are in continuous engagement. Since the flexspline has two teeth less than the circular spline, one revolution of the input causes relative motion between the flexspline and the circular spline equal to two teeth. With the circular spline rotationally fixed, the flexspline rotates in the opposite direction to the input.

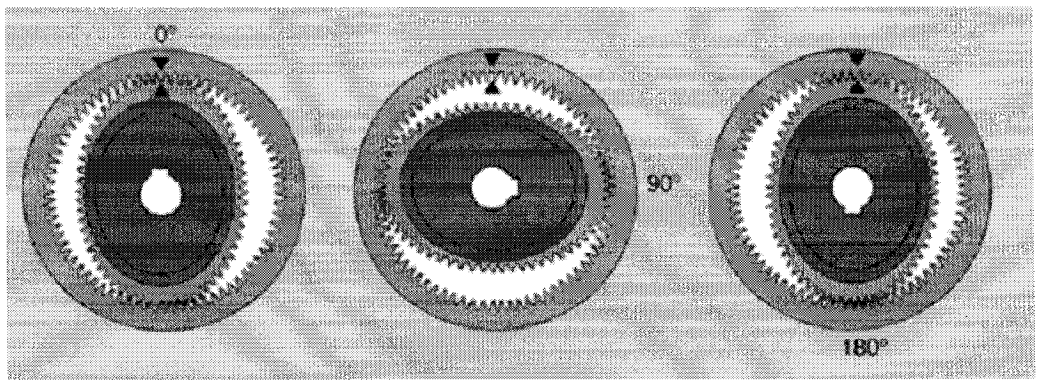

Figure 3. Operation of Harmonic Drive Gear

On the other side of the advantages of a harmonic drive, there are disadvantages which make it a challenging device for control engineers. Some of the drawbacks of harmonic drives are listed here.

2.1.1 Kinematic Error

What harmonic drive literature refers to as kinematic error is the difference between the ideal and the actual output positions. In an ideal gear system, one may expect the gear transmission ratio to be constant and the output position to be proportional to the input position. However, in harmonic drives, a small amplitude of periodic kinematic error exists between the ideal and the actual output position, thereby making the gear ratio dependent on the input position. The error also has a dynamic component [2][3][4].

2.1.2 Flexibility and Presliding Displacement

Flexibility in a harmonic drive results from various compliant elements including the flexspline cup, elliptical ball bearing and gear teeth. Nonlinear interactions of the elliptical ball bearing, the flexspline, and the circular spline with friction at the contact surfaces along with flexibility effects produce a presliding signature. Presliding is the flexible displacement in harmonic drives. In mechanical systems, presliding makes a system's output have hysteresis attributes. A hysteresis curve can be obtained by locking the output port and controlling the input displacement to follow a periodical reference waveform. The displacement, when plotted against the output (generated due to the periodic motion), gives rise to hysteresis. Seyfferth et al. proposed a fairly complex model to capture the hysteresis [33].

2.1.3 Friction and Power Loss

All harmonic drives exhibit power loss during operation. The bulk of energy dissipation can be blamed on the wave generator bearing friction, gear meshing friction, output bearing friction and flexspline structural damping [2][4]. Among them most of the frictional dissipation results from gear meshing [3][4]. Also comparing the ball bearing friction, the wave generator friction is more important than the output bearing friction since it is acting on the high speed/low torque port of transmission, and its effect is magnified by the gear ratio [4]. The wave generator friction is the major part in the motor side friction, and other parts can be neglected with an acceptable accuracy.

2.1.4 Simplified Model

Assuming no friction and external torques on the load side, the hysteresis and kinematic error can be neglected, and the torque applied from the harmonic drive to the motor shaft, T_{ext}, can be simply written as

$$T_{ext} = r_g K_s (r_g q_m - q_l) \qquad (1)$$

where q_l is the load side angular position, q_m is the motor side angular position, r_g is the gear ratio, and K_s is the spring constant for the flexspline flexibility.

2.2 Friction Components

Friction in a harmonic drive consists of two parts: the motor side friction, and the load side friction. The former is the challenging part with nonlinear behaviours. Since the load side is usually equipped with high quality ball bearings, its friction is generally neglected in most

applications. On the other hand, because of being on the "high torque-low velocity" part of the gear box, the error caused by neglecting the load side friction will be more decreased. As a result, most researchers have focused on the motor side friction rather than the load side [2][3][7][9]. However in cases that there exists considerable friction in the load side, there is a great risk of falling in the stick-slip regime of friction, due to the low velocity of the load side.

2.3 DC Motor

The DC motor equipped on our manipulator which is shown in Figure 11 is an RH-5A-5502 harmonic drive motor. The common model of a DC motor is

$$J_m \ddot{q}_m = T_m - \frac{K_m K_b}{R} \dot{q}_m + \frac{K_m}{R} u(t) - T_{ext} \qquad (2)$$

where q_m is the motor shaft angular position, J_m is the motor momentum of inertia, R is the motor resistance, K_m and K_b are motor constants, and u is the motor input. The term T_m is the motor side Coulomb friction and T_{ext} is the external torque applied to the motor shaft. The motor constants K_m and K_b are calculated based on offline experiments and the data provided by the manufacturer. The motor resistance is assumed to be constant while we found that it has 20% noise mostly caused by the contacts in the brushes. This noise in the motor resistance which actually changes the motor constants is considered as a parameter uncertainty.

2.4 Motor Side Friction

Friction in the harmonic drive, as in any other system, produces nonlinear dynamic effects, especially at slow velocities when the system enters the stick-slip regime. As mentioned

before, Gandhi [35] showed that the simple columb and viscous friction models are not adequate to accurately model friction in harmonic drives, and more enhanced models which consider nonlinear speed and position dependency of the friction torque are required. In this research, a structurally dynamic neural network is proposed for estimation of position dependency of harmonic drives, while the velocity dependence is found to be approximated with the viscose friction with an acceptable accuracy.

Yu Kun [61] showed that the static and columb frictions in harmonic drives are very close. He showed that static friction is about 3 percent larger than columb friction. Neglecting the 3% difference, the motor side friction as

$$F_p(\dot{x},x) = -\text{sign}(\dot{x}) f_p(x) \qquad (3)$$

where $f_p(x)$ is the magnitude of the friction function (columb or static). Taghirad [4] also stated the equality of static and columb friction in harmonic drives.

2.5 The Load Side Friction

In order to decrease the friction, the joints in the load side are equipped with high quality ball bearings which result in having near zero friction in compare to motor side friction. Besides, dividing this small friction to the gear ratio ($n = 80$) easily makes the consequents of the load side friction negligible on the motor shaft. This low friction also helps us to neglect the flexibility of the flexspline and hysteresis in the harmonic drive and results in a simpler model.

In the load side we have

$$J_l \ddot{q}_l = K_s(r_g q_m - q_l) \qquad (4)$$

Accumulating all the models result in a nonlinear 4th order model as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{F_m}{J_m} - \dfrac{rK_s}{J_m}(rx_1 - x_3) - \dfrac{K_m K_b}{J_m R} x_2 + \dfrac{K_m}{J_m R} u(t) \\ x_4 \\ \dfrac{F_l}{J_l} + \dfrac{K_s}{J_l}(rx_1 - x_3) \end{bmatrix} \tag{5}$$

$$[x_1 \quad x_2 \quad x_3 \quad x_4]^T = [q_m \quad \dot{q}_m \quad q_l \quad \dot{q}_l]^T \tag{6}$$

where the terms are defined in the nomenclature.

Assuming no friction on the load side the model can be assumed as a 2nd order model. The adaptive controller is designed based on the 2nd order model. Having good performance when applied to the real plant (which is 4th order) shows the stability of the controller to the unmodeled dynamics and uncertainties. So the plant model assumed for the robot manipulator in the adaptive controller is

$$J_p \ddot{x} + c_p \dot{x} - F_p(\dot{x}, x) = u \tag{7}$$

while J_p is the accumulated J (motor side, load side and the gear) on the motor side, c_p is the viscous friction coefficient, u is the system input, and $F_p(\dot{x}, x)$ is the nonlinear friction. Experiments on the harmonic motor show that the friction function is periodical with the interval of 2π, however this is rather obvious as the friction exists on the motor side only. Gandhi [35] also showed this periodic behaviour.

2.6 Approximation of Position Dependent Friction

Based on (3) the estimation of the friction can be written as

$$\hat{F}_p(\dot{x},x) = -\text{sign}(\dot{x})\hat{f}_p(x) \qquad (8)$$

where $\hat{f}_p(x)$ is the magnitude of the Columb friction which is to be adaptively approximated with a function approximation network. The network is mathematically represented as

$$\hat{f}_p(x) = \hat{C}_0 + \sum_{i=1}^{N} \alpha_i \hat{C}_i \Psi_i(x) \qquad (9)$$

where N is the number of nodes in the network, $\hat{f}_p(x)$ is the approximation of $f_p(x)$, $\hat{C}_i$ is the estimate of the weight of each node, $\hat{C}_0$ is the estimate of the bias, and Ψ_i is a basis function (more details of basis functions are addressed in chapter 3). α_i is a parameter for the level of the participation of the node in the network. $\alpha_i = 0$ is a deleted node and $\alpha_i = 1$ is an active node. The graphical structure of the network is shown in Figure 4; however this representation will be described in detail later.

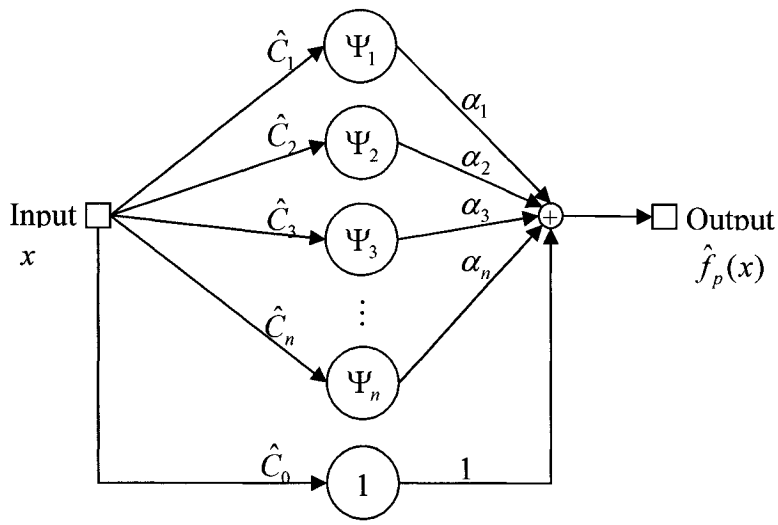

Figure 4. The Network Structure

The controller is a MRAC (Model Reference Adaptive Controller). The target of the MRAC is to follow a reference model with of

$$J_m\ddot{x}_m + c_m\dot{x}_m + k_m x_m = r \qquad (10)$$

where r is the reference input and $c_m / J_m > 0$ and $k_m / J_m > 0$ to satisfy the stability of the reference model.

In this chapter the problem was described and the basics of our new approach were proposed. A 4th order model was proposed for the manipulator which was then simplified to a 2nd order model. A structurally dynamic network is then proposed to estimate the position dependent friction in a harmonic drive. However, there are not any notes about the type of the nodes of this network yet. In the next chapter different node types will be discussed and the most suitable one will be selected for our research.

3 Basis Functions

In signal analysis, there are a number of different ways one can use in order to translate a signal into different forms that are more suitable for different applications. The most popular basis function is the Fourier transform which converts a signal from time versus amplitude to frequency versus amplitude. There are also a number of basis functions that one can map the function onto. Radial Basis Functions (RBF) are also basis functions which synthesize a function based on some radial-symmetric functions localized in time and without any frequency resolution. Wavelet is another category which is almost more capable than Fourier and RBF in function approximation. These basis functions are briefly described in this section.

Fourier, Gaussian Radial Basis Function (GRBF) and Wavelets are basis functions considered in this research. These basis functions are used to approximate the friction function in the harmonic drives. The friction function is assumed to be a function of x only. As previously described, the friction function is periodical with the interval of 2π defined on $[0,2\pi]$.

3.1 Fourier Basis Function

A Fourier series is an expansion of a periodic function $f(x)$ in terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality relationships of the sine and cosine functions. So the periodic functions can be approximated using the Fourier series formulation in a Fourier network. A Fourier Network decomposes the function based on a finite Fourier series approximation and its node is defined as

$$\Psi_i^s(x) = \sin(i\pi x / l)$$
$$\Psi_i^c(x) = \cos(i\pi x / l) \tag{11}$$

where l is the function interval. For the friction function we have $l = 2\pi$ and the node definition reduces to

$$\Psi_i^s(x) = \sin(ix / 2)$$
$$\Psi_i^c(x) = \cos(ix / 2) \tag{12}$$

The adaptable parameter in the Fourier node is the weight of the node.

Fourier transform is a helpful tool for many applications. Its key feature is mapping data into frequency resolution while missing showing when each frequency starts or ends in the time domain. This is because of missing any time resolution in the Fourier series approximation. To combat this problem, mathematicians came up with the short term Fourier transform which can convert a signal to frequency versus time. Unfortunately, this transform also has its shortcomings mostly that it cannot get decent resolutions for both high and low frequencies at the same time. So how can a signal be converted and manipulated while keeping resolution across the entire signal and still be based in time? This is where wavelets come into play. Wavelets are finite windows through which the signal can be viewed. Wavelets can be translated about time in addition to being compressed and widened.

3.2 Radial Basis Function (RBF)

Radial Basis Functions are time localized functions which are widely being used as basis functions in function approximation networks [45][51][53]. Up to now, several different types of RBFs have been introduced to be used in the NN networks for nonlinearities while the Gaussian

RBF (GRBF) is the most common in use. Other types include Logarithmic Radial Basis Function (LRBF), Inverse Multi Quadratic Equation (IMQE), Thin Plate Splines (TPS),...[50]. An RBF node is defined as

$$\Psi_i(x) = \varphi(x - x_i) \tag{13}$$

where x_i is the node center. The GRBF is defined based on the Gaussian function as

$$\varphi(x) = e^{-x^2/2\sigma^2} \tag{14}$$

where the σ is often called the spread of the node.

The parameters in GRBF nodes are the node weight (C_i), the node center (x_i), and the spread (σ). The GRBF node responds only to a small region of the input space where the Gaussian function is centered. The key to a successful implementation of these networks is to find suitable centers (x_i) and suitable spreads (σ) for the Gaussian nodes. In complex and intricate functions this process does not seem to be so straight forward and effortless to do.

3.3 Wavelets

It is well known from Fourier theory that a signal can be expressed as the sum of a, possibly infinite, series of sines and cosines. This sum is also referred to as a Fourier expansion. The major disadvantage of the Fourier expansion however is missing of the time resolution. This means that although we might be able to determine all the frequencies present in a signal, but we do not know when they are present. To overcome this problem in the past decades several solutions have been developed which are more or less able to represent a signal in the time and frequency domain at the same time.

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale [63]. They have advantages over traditional Fourier and RBF methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology [62][49][46]. Interchanges between these fields during the last ten years have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction. Wavelet functions are separated to different families which each family is defined based on a main function called the Mother Wavelet.

The wavelet transform converts a signal from the time domain to two domains of time and frequency. The continuous wavelets transform of a function is defined as [63]

$$T^c\{f\}(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t)\psi(\frac{t-b}{a})dt \qquad (15)$$

where a is the scale, b is the time shift, and ψ is the mother wavelet function. The discrete wavelets transform is defined as [63]

$$T^d\{f\}(m,n) = |a_0|^{-m/2} \int_{-\infty}^{\infty} f(t)\psi(a_0^{-m}t - nb_0)dt \qquad (16)$$

where a_0 and b_0 are constant numbers and m represents the scale and n represents the time shift. Based on these, the formulation of a wavelet node can be defined as

$$\Psi_{(p,k)}(x) = p^{1/2}\varphi(px-k) \qquad p,k \in R \qquad (17)$$

for the continuous form and

$$\Psi_{i=(j,k)}(x) = 2^{j/2}\varphi(2^j x - kx_0) \qquad j,k \in Z \qquad (18)$$

for the discrete form which is based on Multi Resolution Analysis (MRA). The parameters p (in continuous) and j (in discrete) are the frequency scale and k is the time shift (translation). These two parameters which are defined in two different domains give the wavelets transform the capability of having the resolution in both time and frequency. This feature however makes the wavelets one of the most useful tools for function approximation, and prioritizes it over the Fourier series and RBF.

3.4 Mother Functions in Wavelets

Wavelets mother functions, or simply mother wavelets, play a significant role in wavelets networks. They significantly change the capability of the wavelet networks to construct intricate and spiky functions. However not any function can be a candidate of mother wavelets. The mother wavelets must satisfy the following property:

$$\int_{-\infty}^{+\infty} \varphi(x)dx = 0 \tag{19}$$

In orthonormal networks the following properties should also hold

$$\int_{-\infty}^{+\infty} \varphi^2(x)dx = 1 \tag{20}$$

$$< \Psi_i, \Psi_j >= \begin{cases} 0 & i <> j \\ 1 & i = j \end{cases} \tag{21}$$

where $< f, g >$ is the dot product of the functions defined as

$$< f, g >= \int_{-\infty}^{+\infty} f(x).\bar{g}(x)dx \tag{22}$$

where $\bar{g}(x)$ is the complex conjugate of the function $g(x)$.

Different mother wavelet functions are used in the signal analysis field. This includes "Meyer", "Battle-Lemarie", "Haar", "Mexican Hat", "Morlet", "Symlets", "Gauss-n" (n^{th} derivative of the Gaussian function), ... [63]. The mother function wavelet used in this research is the Mexican hat defined as

$$\varphi(x) = C(1-x^2)e^{-x^2/2\sigma^2} \tag{23}$$

where C is such that the L^2 norm of $\varphi(x)$ would be 1.

In this chapter different types of basis functions were discussed. In this research the wavelets are used because of their great capability in approximating intricate functions with less number of nodes. However the experimental results of the other types of basis functions are presented and compared. In the next chapter the adaptive controller will be discussed and the stability of the system will be analyzed.

4 Adaptive Control using Approximation Networks

Adaptive controllers take the advantage of the estimators to have an estimate of the system parameters, and use the estimated values as the best knowledge about the system. Then the system parameters are updated based on the system tracking error through an update law. However this loop results the system to track the reference input and also find the system parameters simultaneously. The tracking error convergence can be guaranteed through the stability analysis while the parameter convergence needs an additional condition called the richness of input or persistent excitation.

Adaptive controllers have proven to be highly effective in control of systems with unknown parameters. As friction is an unknown parameter in our problem, adaptive control is one of the best candidates for controlling this manipulator. For this purpose a model reference adaptive controller is designed to control the system and estimate the unknown nonlinear function using an approximation network simultaneously. As described before, the approximation network constructs the unknown function based on the basis functions reviewed in Chapter 3. The unknown parameters in the network to be estimated are the weights of the nodes. The mathematical formulation of the network is shown in Equation (9).

The system is considered to work in a working the range $A_d \subset A$. The system should be stable in the set A, and accurately controlled in the set A_d. Two controllers are designed for the two cases when the system is inside A_d and when the system is outside A_d. The adaptive controller is effective only in A_d. Outside A_d, a sliding controller is activated which pushes the

system back into A_d. The sliding controller keeps the stability of the system in cases the adaptive controller fail to function or cause large tracking errors. The global structure of the controller is shown in Figure 5.

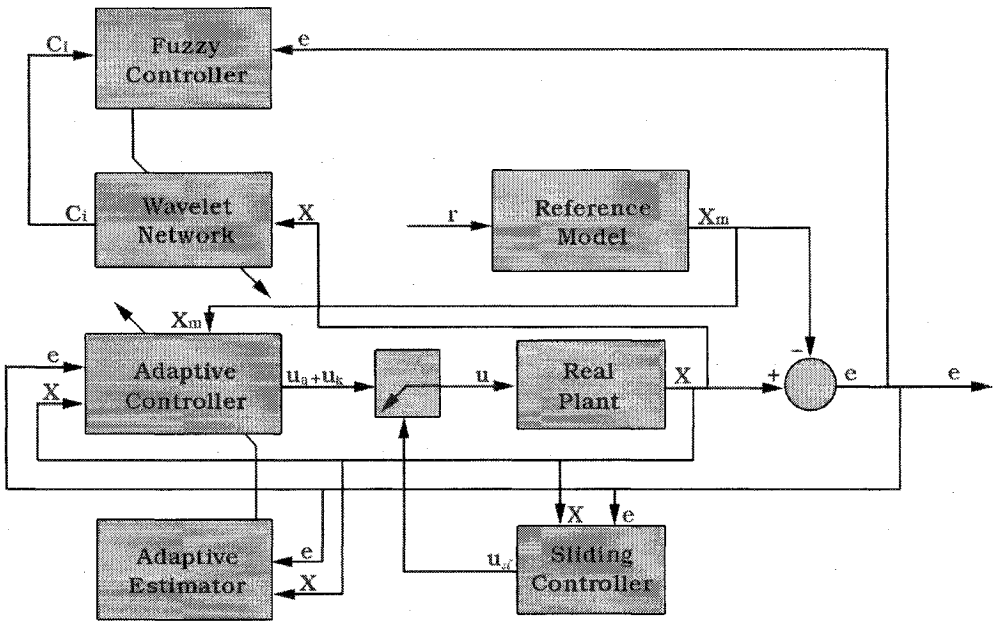

Figure 5. The Fuzzy Controller

4.1 Plant Input

The plant input of the adaptive controller is defined as

$$u = m(u_k + u_a) + \overline{m}u_{sl} \tag{24}$$

$$u_k = -ks \tag{25}$$

$$u_{sl} = -k_{sl}(x,t)\mathrm{sgn}(s) \tag{26}$$

where u_k is the proportional term and u_a is the feed forward adaptive component defined in (28). The term u_{sl} is an additional sliding component which keeps the stability of the system in case the adaptive component fails to function or becomes unstable. The system is supposed to

work in a working range $A_d \subset A$, where A is the set that the system should be stable in. The parameter m is defined such that $m = 1$ in A_d and $m = 0$ outside A_d. So we have

$$m = \begin{cases} 1 & x \in A_d \\ 0 & x \notin A_d \end{cases} \qquad (27)$$
$$\overline{m} = 1 - m$$

The adaptive component is defined as

$$u_a = \hat{J}_p x_r + \hat{c}_p \dot{x} - \hat{F}_p(x, \dot{x}) \qquad (28)$$

while the hat variables are an estimate of the real values and

$$\begin{aligned} x_r &= \ddot{x}_m - \lambda \dot{e} \\ s &= \dot{e} + \lambda e \\ e &= x - x_m \end{aligned} \qquad (29)$$

while k and λ are positive constants, and k_{sl} is a sufficiently large positive constant which satisfies the sliding controller condition. Note that in (28), the sign of $\hat{F}_p$ in the input u is negative. This is to cancel out the friction F_p and lead to having zero friction.

4.2 Adaptation Laws

Based on the adaptation laws, the system unknown parameters are updated in such a way that the system tracking error converges zero. The system unknown parameters include the weights of all nodes in the wavelet approximation network, the system viscous damping, and the system moment of inertia. The common back propagation method is used for the update law and the following are derived:

$$\begin{bmatrix} \dot{\hat{J}}_p \\ \dot{\hat{c}}_p \\ \dot{\hat{C}}_0 \\ \dot{\hat{C}}_i \end{bmatrix} = \begin{bmatrix} \gamma_J & 0 & 0 & 0 \\ 0 & \gamma_c & 0 & 0 \\ 0 & 0 & \gamma_{c_0} & 0 \\ 0 & 0 & 0 & \gamma_{c_i} \end{bmatrix} \begin{bmatrix} \wp\left(-x_r s, \hat{J}_p, \bar{J}_p\right) \\ \wp\left(-\dot{x}s, \hat{c}_p, \bar{c}_p\right) \\ \wp\left(-\text{sign}(\dot{x})s, \hat{C}_0, \bar{C}_0\right) \\ \wp\left(-\Psi_i(x)\,\text{sign}(\dot{x})s, \hat{C}_i, \bar{C}_i\right) \end{bmatrix} \tag{30}$$

where the bar variables are maximums of normal variables which are found based on the physical properties of the system. Conservative estimates can be calculated using the properties of the mother wavelets φ and bounds on the L_2 norm of $f_p(x)$ [49]. The projection function is then defined as

$$\wp(x, y, z) = \begin{cases} x & (|\, y \,| < z) \text{ or } (y \geq z \text{ and } x < 0) \text{ or } (y \leq -z \text{ and } x > 0) \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

4.3 Stability Analysis

Proving the stability of a dynamically structured network has much more issues and is much more complicated than common static networks. However, Sanner and Slotine [49] showed that the stability of the system can be guaranteed regardless of the mechanism of the structural adaptation. In this research a similar proof is proposed which holds regardless of the structural adaptation mechanism. For this purpose the closed loop system dynamics is written as

$$J_p \ddot{x} + c_p \dot{x} - F_p(x, \dot{x}) = m[u_k + u_a] + \overline{m} u_{sl} \tag{32}$$

Outside of the working range we have

$$J_p \ddot{x} + c_p \dot{x} - F_p(x, \dot{x}) = -k_{sl}(x, t)\text{sgn}(s) \tag{33}$$

In this case, outside A_d ($m = 0$), the stability of the system is guaranteed because of the stability of the sliding controller due to the large enough gain k_{sl}. In the working range A_d ($m = 1$), using (9) and (24) to (28), we have

$$J_p\ddot{x} + c_p\dot{x} - F_p(x,\dot{x}) = -ks + \hat{J}_p x_r + \hat{c}_p\dot{x} - \hat{F}_p(x,\dot{x}) \qquad (34)$$

and

$$J_p\dot{s} + ks = \tilde{J}_p x_r + \tilde{c}_p\dot{x} + [\hat{f}_p(x) - f_p(x)]\mathrm{sgn}(\dot{x}) \qquad (35)$$

The term $\hat{f}_p(x) - f_p(x)$ is of great importance in the stability analysis. To analyze this term, $f_p(x)$ is written as

$$f_p(x) = f_p^w(x) + f_0(x) \qquad (36)$$

$$f_p^w(x) = C_0 + \sum_{i=1}^{N} C_i\Psi_i(x) \qquad (37)$$

where the term $f_p^w(x)$ is the discrete wavelet representation of $f_p(x)$ and f_0 is the fixed error introduced to the system by the truncation of the wavelet terms to the discrete points of time shifts and scales in the fully active network. Then we have

$$\hat{f}_p(x) - f_p(x) = \tilde{C}_0 + \sum_{i=1}^{N} \tilde{C}_i\Psi_i - f_{st}(x) - f_0(x) \qquad (38)$$

where f_{st} is the error introduced to the system due to the structural adaptation and relates to those nodes whose $\alpha_i < 1$, i.e. those nodes which are not participating in the network. So we have

$$f_{st}(x) = \sum_{i=1}^{N} (1 - \alpha_i)\hat{C}_i\Psi_i(x) \qquad (39)$$

To prove the stability of the system with the presence of the structural adaptation the following function is considered

$$V = \frac{1}{2} J_p s^2 + \frac{1}{2\gamma_J} \tilde{J}_p^2 + \frac{1}{2\gamma_c} \tilde{c}_p^2 + \frac{1}{2\gamma_C} \tilde{C}_0^2 + \sum_{i=1}^{N} [\frac{1}{2\gamma_{C_i}} \tilde{C}_i^2] \qquad (40)$$

where $\tilde{*} = \hat{*} - *$. Using (30) and (35), the derivative of V along the trajectories of the system would be

$$\begin{aligned} \dot{V} = -ks^2 + \left[\tilde{J}_p x_r + \tilde{c}_p \dot{x} + \tilde{C}_0 + \sum_{i=1}^{N} \tilde{C}_i \Psi_i(x) - f_{err}(x) \right] s + \\ \left[\frac{1}{\gamma_J} \tilde{J}_p \dot{\tilde{J}}_p + \frac{1}{\gamma_c} \tilde{c}_p \dot{\tilde{c}}_p + \frac{1}{\gamma_C} \tilde{C}_0 \dot{\tilde{C}}_0 + \sum_{i=1}^{N} [\frac{1}{\gamma_{C_i}} \tilde{C}_i \dot{\tilde{C}}_i] \right] \end{aligned} \qquad (41)$$

Where

$$f_{err}(x) = f_{st}(x) + f_0(x) \qquad (42)$$

Using (30) we have

$$\begin{aligned} \dot{V} = -ks^2 + f_{err}(x)s + \\ \tilde{J}_p \left(x_r s + \wp\left(-x_r s, \hat{J}_p, \bar{J}_p \right) \right) + \\ \tilde{c}_p \left(\dot{x} s + \wp\left(-\dot{x} s, \hat{c}_p, \bar{c}_p \right) \right) + \\ \tilde{C}_0 \left(\text{sign}(\dot{x}) s + \wp\left(-\text{sign}(\dot{x}) s, \hat{C}_0, \bar{C}_0 \right) \right) + \\ \sum_{i=1}^{N} \left[\tilde{C}_i \left(\Psi_i(x) \,\text{sign}(\dot{x}) s + \wp\left(-\Psi_i(x) \,\text{sign}(\dot{x}) s, \hat{C}_i, \bar{C}_i \right) \right) \right] \end{aligned} \qquad (43)$$

the first term in $\dot{V}$ is always negative. Excluding the second term it is possible to show that all the remaining terms are also negative. For example for the third term we have: if the projection function returns a value, the terms in the parenthesis will cancel. However if the projection function return zero then either $\hat{J}_p > \bar{J}_p$ (so $\tilde{J}_p > 0$) and $-x_r s > 0$ or $\hat{J}_p < -\bar{J}_p$ (so $\tilde{J}_p < 0$) and $-x_r s < 0$. In both cases we have $\tilde{J}_p x_r s < 0$. The same proof is applicable to other terms and the upper bound of the third term to the sixth term is proved to be 0. So we have

$$\dot{V} \leq -ks^2 - f_{err}(x)\mathrm{Sgn}(\dot{x})s \qquad (44)$$

Based on (40), V can be unbounded if any of $s, \tilde{J}_p, \tilde{c}_p, \tilde{C}_0, \tilde{C}_i$ become unbounded. However based on (30) and (31) all the terms $\tilde{J}_p, \tilde{c}_p, \tilde{C}_0, \tilde{C}_i$ are bounded by their maximum values and the only variable which can make V unbounded is s. So for the negative definiteness of the function V, the bound on s would be

$$|s| > \frac{\overline{f_{err}}}{k} \qquad (45)$$

where

$$\overline{f_{err}} = \|f_{err}(x)\|_{\infty} \qquad (46)$$

This shows the function V is decreasing whenever $|s|$ is bigger than some threshold. This shows that the error does not converge to zero. However the bound $|s|$ relates to the value of $\overline{f_{err}}/k$. So the boundedness of the tracking error relates to the boundedness of $\overline{f_{err}}$. $\overline{f_{err}}$ is however bounded because f_0 is bounded due to the boundedness of f_p and boundedness of the wavelet terms; and f_{st} is bounded because of the boundedness of α_i and $\hat{C}_i$. So s is proved to be bounded. The boundedness of s directly results in the boundedness of e because the tracking error is the output of a linear stable first order filter driven by s [49].

Note that this convergence and limit does not relate to the mechanism of the structural adaptation. Of course, a poor choice of structural adaptation mechanism will result in poor a convergence of the tracking error and would be as bad as using no network at all [49]. And in the worse case the bound on the tracking error may be so large that effectively only the action of the sliding component keeps the state of the manipulator bounded.

In this chapter an adaptive controller was proposed which works in conjunction with a sliding controller. The adaptation laws for the adaptive component were presented, and the stability of the system was proved. In the next chapter the structural adaptation will be addressed and its effect on the system performance will be discussed. The new fuzzy based algorithm for the adding/removing of nodes from the network will be then proposed.

5 Fuzzy Logic Structural Adaptation

Structural adaptation is a method for gaining better controller performance by dynamically changing the structure of the controller. It also improves the system behavior in dealing with system uncertainties, unpredicted situations, and noise. The system performance directly depends on the number of unknown parameters that the adaptive controller is estimating. Each unknown parameter adds one state to the system and increases the system dimension. On the other hand having more unknown parameter makes the richness of input condition harder to be satisfied by the reference input which consequently increases the parameter estimation error. This can be a source of transient errors which deteriorates the performance of the tracking. Decreasing the number of nodes decreases the number of unknown parameters and consequently results in a better transient response and a better performance and robustness in response to the unknown dynamics. Large number of unknown parameters in an adaptive controller makes the controller less strong in response to the sensor noise as well. So a method for minimizing the number of nodes to achieve the best structure of a network is of great importance. This method is called the structural adaptation and consists of two processes: "Adding new nodes" and "Removing unnecessary nodes". In the Add process, new nodes are introduced to the network; and in the Remove process, unnecessary nodes are removed. In our representation of the network a node is added to the network by setting its α_i to 1, and is removed by setting its α_i to 0.

Different methods can be used to add/remove nodes to/from a network; however inappropriate adding and removing of nodes can deteriorate the performance such that the

outcome becomes worse than the case of having no networks at all [49]. The performance of the system greatly relies on the way the network structure is being changed. Fast add/removal of nodes, or add/remove of nodes when the system is experiencing severe transient responses can deteriorate the performance such that it became worse than the case that the network was structurally static or even there was no nodes in the network at all [49].

Adding and removing nodes to/from a network imposes large changes in the state of the whole network. Consequent changes in the structure of a network may be a source of causing large errors, chattering or even instability. In experiments, these changes in the structure of the network can effectively cause large impulses to the system and excite the unmodeled high frequency dynamics of the system and cause large tracking errors or large amplitude chattering. However because of the locality of wavelets in both time and frequency, the impact of changing the structure of a wavelet network decreases in regions which are farther from the position that the nodes are inserted or removed.

An important issue is the time between two consequent structure changes. The structural adaptation should be as slow as possible to let the system fit itself into the new structure. Although fast structural adaptation may result faster convergence in achieving the most suitable network, but in real systems transient noises can easily deteriorate the performance and cause large amplitude chattering.

The structural adaptation is continuously applied to the system, i.e., similar to the parameter adaptation process, the structural adaptation process does not necessarily need to be stopped. However, in our method the fuzzy rules automatically stop the structural change when some level of accuracy in the tracking error is achieved.

Sanner and Slotine [49] proposed an algorithm for structural adaptation of a wavelet network in conjunction with an adaptive controller. In the Sanner's algorithm nodes are being selected for adding or removal based on a threshold on the magnitude and the rate of the change of their weights. Nodes whose weights are more than the threshold or increasing in time are kept to the network; and conversely, those nodes whose weights are lower than the threshold and decreasing in time are removed [49]. This kind of structural adaptation method may work fine with the simulations but in real systems that there exist sensor noise and model uncertainties it may easily fail to function and cause large tracking error or even instabilities in the adaptive component of the controller. This causes the system to leave the working range and activate the sliding component of the controller which pushes the system back to its working range. However obviously this means very poor tracking performance.

Sanner's method has three drawbacks: First the structural adaptation is fast; second the network is changing too much in each step; and third finding the best threshold is rather difficult. From the switching control point of view, this method is a fast switch between many too different controllers. With the presence of noise this kind of controller can easily fall into large amplitude chattering or become unstable. However, the instability takes place when the adaptive component of the controller is active and causes the system to leave its working range A_d. Then the sliding component forces the system back to A_d and keeps the system stable. The outcome of this kind of stability is a very poor tracking performance with error bound as large as the whole A_d.

As mentioned before, setting the threshold is not easy and straight forward in the Sanner's method. Using a low threshold causes very few nodes to be participated in the network, while a little increase on the threshold suddenly introduces a lot of nodes.

The proposed method for the structural adaptation is based on some heuristic rules obtained from the system behavior and some notions in the switching control field. The process of structural adaptation is like a switching control problem having many different networks in the controllers, and a switching algorithm to switch between them. So some notions of the switching control field are applicable here. One of the main notions is the dwell time which will be discussed more later.

In this thesis a new method of structural adaptation is proposed which is based on some fuzzy rules. The decision of adding and removing nodes to/from the network is made by a parallel fuzzy controller which is designed to organize the changes in the network structure. The structural adaptation includes two processes: adding new nodes and removing unnecessary nodes. The fuzzy algorithm adds new nodes when a node is needed and deletes unnecessary nodes when they have relatively low effect on the function reconstruction process or cause chattering. The fuzzy controller is designed based on some heuristic phenomena derived from the system behavior and some concepts in the switching control. The fuzzy algorithm consists of two parts: FA1 and FA2.

The main responsibility of FA1 is to decide for adding new nodes to the network. It decides for removing the last node added to the network too. As noted before the performance of the system is very sensitive to the structural adaptation. For example when the system output is chattering or is experiencing severe transient responses with fast changes, adding new nodes can potentially exacerbate the situation. It can magnify the chattering amplitude, or cause the system to leave its working range. But if the tracking error is relatively small and does not have large chattering, the network is capable of accepting a new node without performance deterioration. The new node enhances the system performance by improving the estimation capability of the

wavelet network. FA1 continuously checks the system status for a proper condition to add a new node.

As noted before FA1 decides about removing the last node added to the network too. This takes place in cases that the tracking error becomes large or has chattering. Removing the last node results in a better situation because the network excluding this node had had such an acceptable performance that it had reached that configuration. So the last node is removed by FA1 to enhance the performance in cases that it does not result in a good performance. This node can be added again if FA1 decides to do so.

The inputs of FA1 are: two variables defined based on the tracking error (equation (48)), the last time that a change in the network structure has been made, and the current time. Its only output is the change of the nodes count. If the output is $+1$ then a new node is added, if it's 0 no changes is made and if it is -1 the last node is removed from the network. The overall diagram of FA1 is shown in Figure 6.

At the same time and in parallel to FA1, FA2 decides about removing bad nodes from the network. Bad nodes are those which have a small contribution in reconstruction of the estimated function, or those which have chattering. These nodes deteriorate the performance by causing chattering on the output or decreasing the robustness of the network by being an unnecessary state in the system equation.

FA2 checks each node to see if it is necessary to be in the network or not. Its inputs are the node's weight, the last time that a change in the node's existence has been made, and the current time. Then its output is the change in the node's existence. An output of $+1$ means the node should be added back to the network, 0 means no change should be made, and -1 means the node should be removed. The overall diagram of FA2 is shown in Figure 6.

The fuzzy propositions in FA1 and FA2 can be set such that it results in a cautious structural adaptation, or a reckless one. The definition of these fuzzy propositions also affects the convergence of the structural adaptation process. In real systems, in case of presence of noise and uncertainties, the structural adaptation process may not actually converge to a constant value, but have a limit cycle around a certain point. The width of the limit cycle again depends on the level of cautiousness of the structural adaptation which relates to the parameters set in the fuzzy propositions.

In this research, all the 3 reviewed basis functions of Wavelets, RBF, and Fourier are used as the basis function of the approximation network, and their performances are compared. However the wavelet is found to be the best. This is mainly because of its localization in both time and frequency domain and having both time and frequency resolution at the same time. As described before this greatly localizes the effect of adding a new node to a close neighborhood region. Zhang [46] which proposed a neural wavelet network for function learning showed that a wavelet network is much more capable in function approximation than usual MLP networks and Gaussian networks. Wavelet networks are defined based on different formulations of discrete wavelet transform. Based on (18) our node structure for the wavelets network is

$$\Psi_{i=(j,k)}(x) = 2^{j/2}\varphi(2^j x - kx_0) \qquad j,k \in Z \qquad (47)$$

where the frequency is quantized on m points, and the time shift is quantized on $2^{m+1}+2m$ points on its working range $[0,2\pi]$.

5.1 Adding New Nodes

Having more nodes in a network makes the discrete estimation closer to the continuous one, which consequently improves the accuracy of estimation. Mathematically it decreases f_{st} which causes having less tracking error bound. In this research, a mechanism for adding new nodes to the network is proposed based on some heuristic rules derived from the behavior of the system and some concepts in the switching control field. These rules are grouped together to create a fuzzy algorithm to decide if a node should be added to the network; and conversely, if any node should be removed. The membership functions in the fuzzy rules can be set in such a way to have a cautious structural adaptation or a reckless one. These membership functions are shown in Figure 7 and Figure 8. Small values of A and B cause cautious structural adaptation while large values cause reckless one. Although the stability of the system is guaranteed by the sliding component of the controller, but achieving a good performance is reliant to the performance of the adaptive component which controls the manipulator in the working range. This is because the outcome of relying on the sliding controller to maintain the stability is large bounds of error as large as the whole range of A_d.

Fast switching between two nicely stable controllers can easily result in instabilities [70]. This is the basic idea of definition of dwell-time in the switching control literature. The stability of a switching controller is greatly dependent to the value of the dwell time. A small dwell time (fast switching), may cause instabilities while larger values may stabilize the same controller [70]. Based on this phenomenon some fuzzy rules are defined to prevent the system from having fast changes in the network structure. These rules create a fuzzy algorithm which organizes the structure of the network. The fuzzy algorithm is illustrated in the Figure 6.

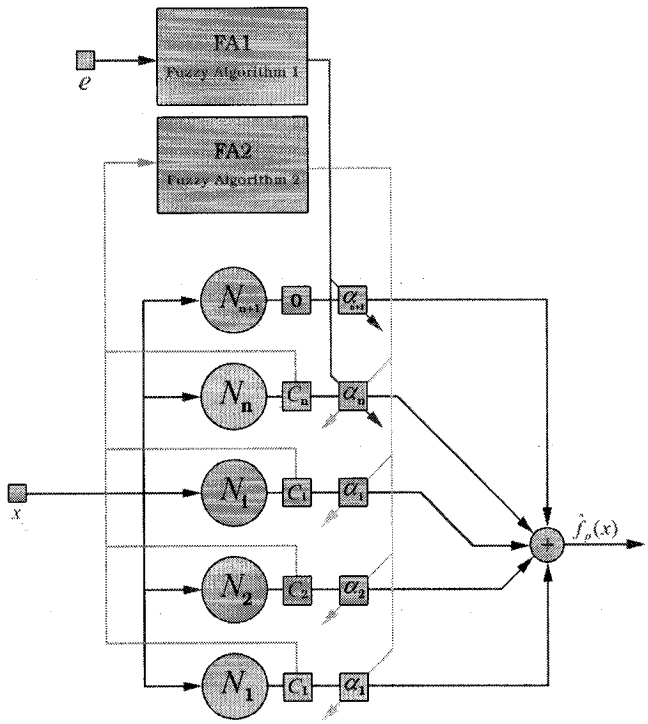

Figure 6. The Fuzzy Controller

where N_i represents a node which can be either wavelets, RBF or Fourier.

Adding a new node takes place by setting its α_i to 1. When a node is added, its weight is set to zero. The weight is then being updated by the update laws. In general, adding new nodes to a network can effectively change the weights of all other nodes. But as wavelets are localized in both space and frequency, the effect of adding a new node to a wavelet network will be less on regions further from the point that the new node is inserted. This phenomenon makes the wavelet networks much more robust to the structural adaptation, while other basis functions (like the Fourier) lack this advantage.

5.2 Fuzzy Algorithm 1 (FA1)

It is found that suddenly introducing a large number of nodes to the network significantly deteriorates the performance. Continuous large structural changes cause large tracking error and large amplitude chattering. Chattering is found to be due to the presence of unmodeled high frequency dynamics in the real plant which are excited in these fast changes. As there may be a large difference between the amplitude of two consequent (neighbor) nodes on the x axis in the transient state, there may be fast jumps in the estimated friction on the consequent points of the x axis. These fast jumps can excite the unmodeled high frequency modes of the system and cause large tracking errors. Based on this, the fuzzy rules are defined based on the following variables

$$\begin{aligned}
&\alpha(t) = |e(t)| \\
&\beta(t) = \delta(t) - \delta^*(t) \\
&\delta(t) = \frac{\int_{\alpha} |e(t)|\, dt}{t - t_{\alpha}} \\
&\delta^*(s) = \frac{1}{\kappa p s + 1}\delta(s) \\
&\kappa = \left[\mathrm{Min}(\frac{\delta}{\delta^*}, 1)\right]^d
\end{aligned} \tag{48}$$

where t_{α} is the time which the last change in the nodes count has taken place, p and d are arbitrary positive numbers. If $\delta < \delta^*$ the output of the filter converges to δ with the time constant of κp where $\kappa = (\delta^* / \delta)^d << 1$; and if $\delta > \delta^*$ the output of the filter converges to δ with the time constant of p.

FA1 is constructed based on 10 rules which are shown in the following table.

If $t-t_\alpha$ is Large^t and			
α \ β	Negative$^\beta$	Zero$^\beta$	Positive$^\beta$
Small$^\alpha$	Δ_N is 0	Δ_N is 0	Δ_N is 0
Medium$^\alpha$	Δ_N is +1	Δ_N is +1	Δ_N is 0
Large$^\alpha$	Δ_N is +1	Δ_N is +1	Δ_N is −1

If $t-t_\alpha$ is Small^t then Δ_N is 0

Table 1. The Fuzzy Engine for Adding New Nodes

where the fuzzy propositions are defined in Figure 7, Figure 8, and Figure 9, and Δ_N represents the number of nodes which should be added to the network. Δ_N is interpreted as

Round(Δ_N) = 0	The number of nodes should not be changed.
Round(Δ_N) = +1	A new node should be added.
Round(Δ_N) = −1	The last node should be removed.

Table 2. Interpretation of Δ_N

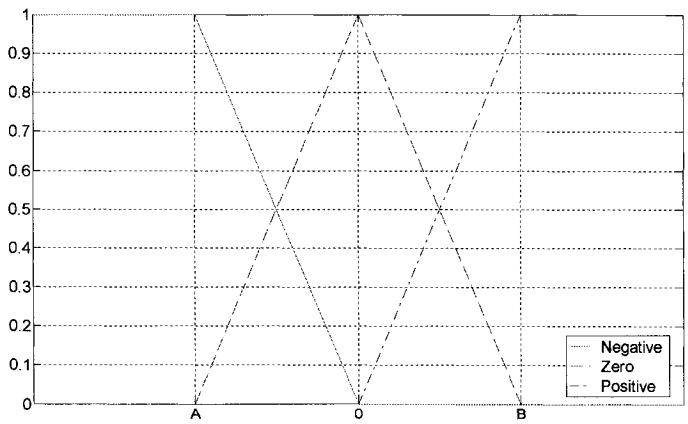

Figure 7. Fuzzy Membership Functions:
"Negative*", "Zero*", "Positive*"

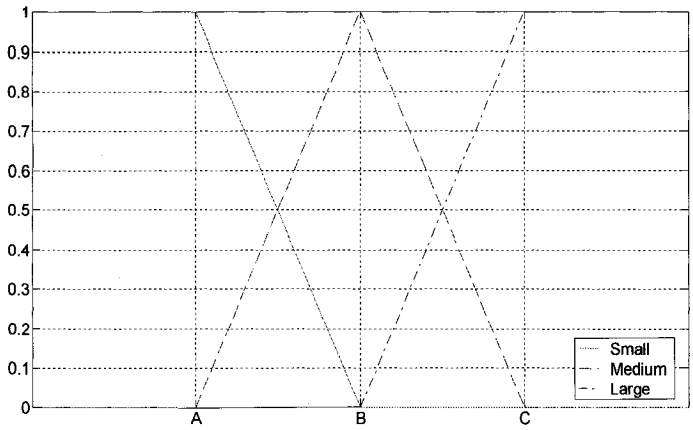

Figure 8. Fuzzy Membership Functions: "Small*", "Medium*", "Large*"

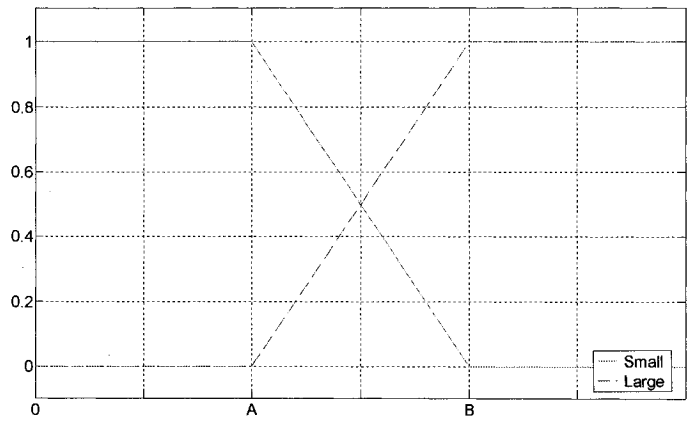

Figure 9. Fuzzy Membership Functions: "Small^{t}" and "Large^{t}"

5.3 Order of Nodes

To get a better function approximation it is much better to spread the nodes almost homogenously on the x axis. By other means the nodes should not be inserted only in some squeezed areas of the x axis. If doing so, the network will have strong approximation capabilities in those areas while having weak capability in other regions. However, if a node is inserted in a region that the original function is flat, the amplitude of that node would gradually converge to zero. This will then result in the removal of that node from the network which will

be discussed later. But as the original function is unknown to the controller, this method gives the new nodes the chance of being placed in almost all points of the space domain. Although this is not a good method for functions having large flat areas but its performance is quite well for common fluctuating functions.

Another issue is the order of frequencies of nodes to be added. Low frequency nodes reconstruct the large scale shape of the function while high frequency nodes reconstruct high resolution (sharp changes) shape. So to obtain a better performance in the function reconstruction process, low frequency nodes must be added first. Based on this idea an algorithm is proposed which sets the order and position of each node in space/frequency domain. This algorithm fills the nodes queue with the low frequency nodes first and the higher frequencies next.

Besides, the low frequency nodes have a wider range on the x axis. So fewer number of low frequency nodes is required to cover the whole space axis ($[0,2\pi]$). Conversely high frequency nodes have a much narrower range and more of them are required to cover the whole x axis range. Based on this, the number of nodes is increased in each frequency level. For example there are 2 node in the first frequency level ($j=0$), 3 nodes in level $j=1$, 5 nodes in $j=2$ and generally 2^j+1 nodes with the frequency of j.

The mentioned algorithm is shown in Figure 10. The variables used in the algorithm are defined as

$$\begin{aligned} j_n &\quad \text{The frequency of node } n \\ k_n &\quad \text{The space shift of node } n \end{aligned} \tag{49}$$

and i, j and n are "integer" variables. The algorithm creates the queue as

$$\begin{bmatrix} k_n \\ j_n \end{bmatrix} = \begin{bmatrix} \frac{0}{1} & \frac{1}{1} & \frac{0}{2} & \frac{1}{2} & \frac{2}{2} & \frac{0}{4} & \frac{1}{4} & \frac{2}{4} & \frac{3}{4} & \frac{4}{4} & \frac{0}{8} & \frac{1}{8} & \frac{2}{8} & \frac{3}{8} & \frac{4}{8} & \cdots \\ 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 & \cdots \end{bmatrix}$$

Table 3. The Nodes Order Algorithm Output

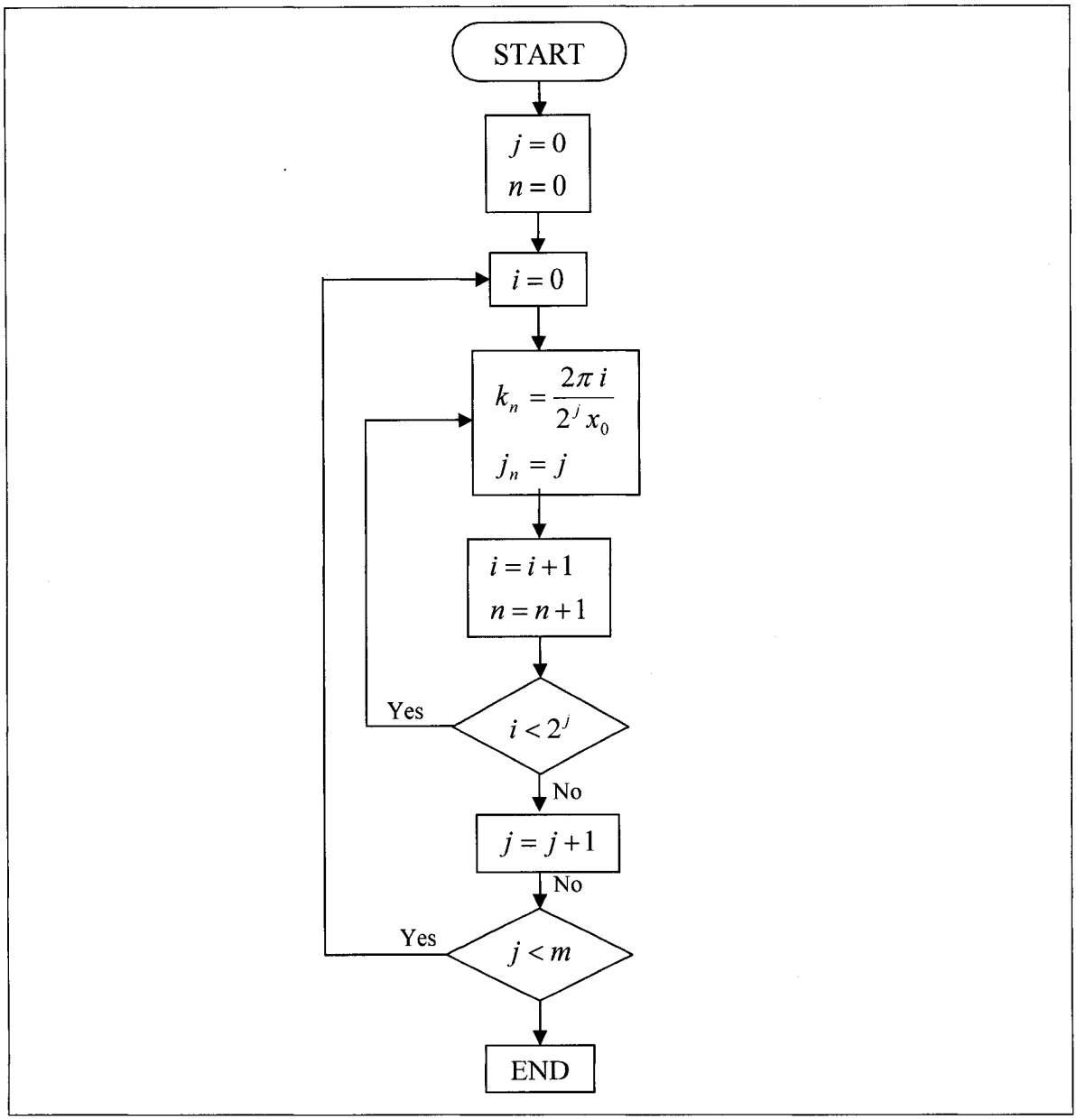

Figure 10. The Algorithm for Generating the Order of Nodes

5.4 Removing a Node

Although having more nodes in a wavelet network results in more precision in theory; but having large number of nodes is not so much desired in practice. Large number of nodes can cause implementation problems such as taking a long time for calculations in real-time controllers, or increasing the round-off error. It can also cause large tracking error and chattering in the presence of noise and uncertainties. So those nodes which are not participating too much in the function reconstruction should be removed from the network. A node is removed by setting its α_i to 0. Based on the proposed method in this research, a node should be removed from the network if any of the following conditions holds:

5.4.1 Condition 1

The last node introduced into the network will be removed if "$\text{Round}(\Delta_N) = -1$". The last node removed from the network can be added again if FA1 states "$\text{Round}(\Delta_N) = +1$".

5.4.2 Condition 2

Another fuzzy algorithm (FA2) is designed to decide if a node should be removed from the network or not. All nodes participating in the network should be examined by FA2 to check if they meet the proper conditions to be in the network. If a node is targeted for removal, the α_i of that node will be set to 0. FA2 is defined based on the following variables

$$\upsilon_i(t) = |C_i(t)| - C_i^m$$

$$\omega_i(t) = \frac{\int_{t_i^\alpha}^{t} |C_i(t) - C_i^*| dt}{t - t_i^\alpha} \qquad (50)$$

$$C_i^* = \frac{1}{\tau s + 1} C_i$$

where t_i^α is the time that the last change in the node's α_i has taken place, τ is an arbitrary positive constant, and C_i^m is a positive number indicating the threshold for magnitude of the weight of each node to be participated of in the network. ω_i indicates the level of chattering of each node. Based on these variables the following rules are designed which compose fuzzy algorithm 2 (FA2) as

If $t - t_i^\alpha$ is Large_i^t and

ω_i \ υ_i	Negative$^\upsilon$	Zero$^\upsilon$	Positive$^\upsilon$
Small$^\omega$	Δ_{α_i} is 0	Δ_{α_i} is +1	Δ_{α_i} is +1
Medium$^\omega$	Δ_{α_i} is −1	Δ_{α_i} is 0	Δ_{α_i} is 0
Large$^\omega$	Δ_{α_i} is −1	Δ_{α_i} is −1	Δ_{α_i} is −1

If $t - t_i^\alpha$ is Small_i^t then Δ_{α_i} is 0

Table 4. The Fuzzy Engine for Removing Nodes

where the fuzzy propositions Small_i^t and Large_i^t are the same as Small* and Large* defined in Figure 7, Figure 8, and Figure 9, and Δ_{α_i} is interpreted as

Round(Δ_{α_i}) = 0	Change nothing.
Round(Δ_{α_i}) = +1	The node should be introduced if it's not in the network.
Round(Δ_{α_i}) = −1	The node should be removed if it's in the network.

Table 5. Interpretation of Δ_{α_i}

where the "Round" is the common "Round(x)" function defined as the closest integer to "x".

In this chapter the structural adaptation was addressed and a new fuzzy based method was proposed. In the next chapter the experimental results of the proposed method will be shown and compared with other methods.

6 Experimental Results

6.1 Experimental Setup

The method proposed in this research has been fully tested and approved on a 5 link 2-DOF manipulator which was also used in [54] and [61]. The key point of this manipulator is two arms driven by two DC motors equipped with two harmonic drive gears. The motors are from HD Systems Inc. model Hi-TDrive RH-5A-5502. The catalogue of this motor is provided in Appendix A. These arms are connected together and form a 5-link system with the target of moving the top joint on a certain predefined path. This robot can be used for many precision positioning purposes like tele-surgery and optoelectronic industries. The robot is shown in Figure 11.

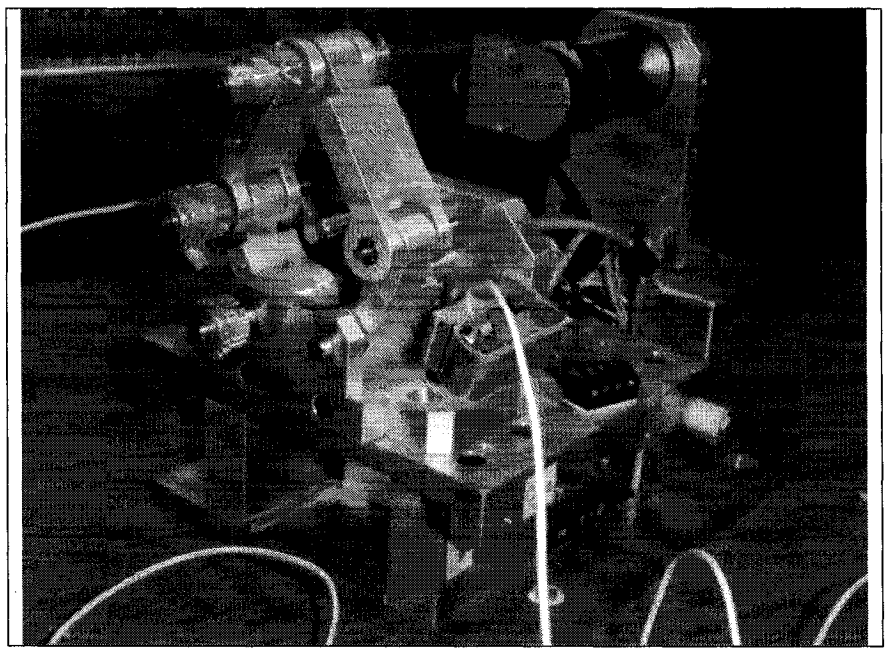

Figure 11. The Robot

Each DC motor is equipped with an encoder coupled to the motor output shaft having 1440 encoder ticks per motor round. These encoders are used as the system feedback for both the position and the velocity; however measuring the velocity with the encoder needs more considerations. The controller is implemented on an Intel 2.4^{GHz} PC, running the Windows 2000, and Venturcom™ RTX™ as the real-time operating system. For the purpose of communication a Quanser™ MultiQ™ PCI interface card having 4 of 16bit D/A, 4 of AB encoder channel, and 64bits DIO is used. The D/A channel is used for the controller output, the encoder channel to measure the position, and the 16bit DI channels for communicating with an external micro controller. The external microcontroller measures the velocity based on the time between each two consequent encoder ticks using the inverse time method. An amplifier with the gain of 3, also from Quanser™ is used to supply the required current for the motor. The maximum voltage of the amplifier is ± 22.76v while the working range for the DC motor is between -10 and +10v. By applying sine waves of different frequencies and measuring the amplifier input and output amplitudes a Bode plot can be obtained as shown in Figure 12. This figure guides us for selecting the sampling period of the controller.

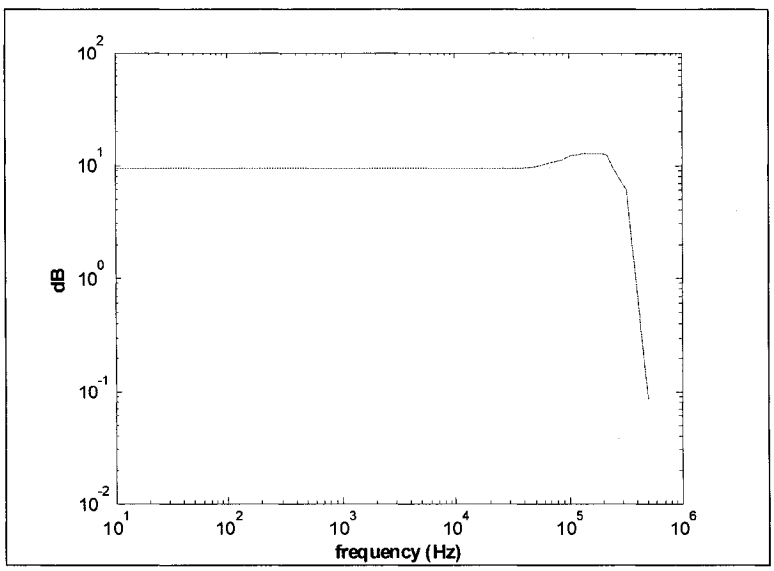

Figure 12. The bode plot of the amplifier

6.2 Implementation Issues

Applying the reviewed theoretical formulations to the experiment needs some additional considerations. Some of these issues are addressed here.

6.2.1 Velocity Measurement Based on the Encoder

Velocity measurement based on encoder turned out to be a critical problem. Neglecting the quantization error caused by the encoder, there still is the problem of measuring the velocity. Measuring the velocity is of great importance because it is directly fed forward to the system input through the viscous friction and through x_r (equation (29)). It also exists in s which plays a significant role in almost all equations. Based on these two reasons, system performance greatly relies on accurate measurement of velocity. As the only sensor of the system is the shaft encoder, an observer is needed to obtain the velocity. Methods of measuring the velocity based on the encoder fall into two main categories:

1. **Finite-Difference [64].**

   Finite-Difference scheme counts the net number of pulses during a fixed interval of time; multiply it by the angle corresponding to successive pulses and divide it by the duration of the interval [64]. References [65][66][67] represent several physical implementations of the finite-difference scheme, with very few variations in the basic principals [64]. This method gives a good precision if the number of encoder counts in two consequent observations is relatively high.

2. **Inverse-Time [64].**

   The inverse-time method deduces velocity as the inter-pulse angle divided by the time between two successive pulses [68][69]. This method is much more precise than the Finite-Difference method if the number of encoder counts between two observations is relatively low.

In large timer intervals both of these two methods may cause large estimation errors, however Belanger [64] designed a velocity and acceleration observer using Kalman filtering which slightly overcomes this problem.

In this research the required method is selected based on the velocity of the arm. In low velocities the Inverse-Time is used which results in a better precision while Finite-Difference is used in fast motions.

6.2.2 Determining the Optimum Motor Speed

The faster the motor rotates, the more change would be in the encoder position in each two consequent timer intervals. To gain a better performance, it is much better to observe the system states in closer points. By other means checking the tracking error and updating the friction function in closer points of the motor shaft result in a much better approximation of the friction function. For this reason the motor speed is selected such that the change in the encoder position in each two consequent timer intervals would be about 2 encoder counts. Now that the number of encoder counts in two consequent timer intervals is small (around 2), the Inverse-Time method would be more suitable for measurement of the velocity.

6.2.3 Calculation's Time

In real time systems the required time for calculation is a critical problem. The more number of nodes participating in the network, the more time is required for the calculations. The calculation consists of three parts: updating the weights of the nodes, calculating the output of the network, and calculation of the fuzzy algorithms. Using the best optimization techniques and buffering results to avoid recalculations a PC with the CPU of P4 2.4GHz is capable of handling the calculation of about 50 nodes in 1000us. So setting the timer interval to 1000us the PC can only handle the maximum of 50 nodes in the network. Needing more nodes for the friction estimator in realtime controllers requires more powerful computers or using distributed networking techniques to distribute the network between two or more computers. This technique is however an active topic these days.

6.2.4 Determining the Timer Interval

Another key parameter is the sampling period. Many issues should be taken into account to select a proper sampling period for the controller. Although lower intervals make the system be closer to the continuous controller, but too low intervals may cause practical problems. Notably the computation time required could be potentially prohibitive for large numbers of nodes. Also the velocity updates are limited by the calculation procedure previously noted above. Considering all these factors a sampling period of 1000us is used for this application using a PC with the P4 2.4GHz CPU.

6.3 Experimental Test Conditions

In this section the experimental results are shown for the wavelet network. However the results are compared with the networks with other types of basis functions (RBF and Fourier). To implement the proposed method some simplifying assumptions are introduced which are listed below. Constant parameters are also listed. The results are then shown in Figure 6.4. It should be noted that as the adaptive component of the controller has a good performance, the system does not leave the working range A_d, and the sliding component of the controller would always be zero.

6.3.1 Assumptions

The experiments are performed by considering some assumptions. Although the real system avoids all these assumptions, but the acceptable results confirm the robustness of the controller to unmodeled dynamics, uncertainties, and noise. The assumptions are

1) The effect of gravity on the arms is zero. However the manipulator is placed in the horizontal plane.
2) The friction in the load side is small in compare to the motor side friction. So the hysteresis and presliding is ignored.
3) The amplifier has a constant gain in the working range of frequencies.
4) Although based on (30) the adaptive controller should find the values of $\hat{J}_p$ and $\hat{c}_p$, but for better convergence of the network and also more precise estimation of the

position dependent component of the friction, and assuming that $\hat{J}_p$ and $\hat{c}_p$ does not change with time and position, they have been estimated offline.

6.3.2 Constant Parameters

The assumed constant parameters in the experiment are listed below. Note that the wavelet functions adaptation gain (γ_i) is considered 10 times more than that of the bias (γ_0). This to impose the estimator to try to estimate the friction function by updating the weights of the wavelet nodes (C_i) rather than changing the bias (C_0).

$$\begin{array}{cc} \lambda = 50 & k = 0.03 \\ \gamma_0 = 1 & \gamma_i = 10 \\ \hat{m}_p = 2.45e-3 & \hat{c}_p = 75e-3 \\ p = 1 & d = 3 \end{array}$$

$$\begin{array}{llllll} \text{For } \alpha & \Rightarrow & A = 2 & B = 7 & C = 12 \\ \text{For } \beta & \Rightarrow & A = 5 & B = 5 & \\ \text{For } \upsilon_i & \Rightarrow & A = 0.01 & B = 0.05 & C = 0.10 \\ \text{For } \omega_i & \Rightarrow & A = 0.05 & B = 0.05 & \end{array} \qquad (51)$$

$$\begin{array}{lll} \text{For Small}^t & \Rightarrow & A = 0.1 \quad B = 0.2 \\ \text{For Small}_i^t & \Rightarrow & A = 0.1 \quad B = 0.2 \end{array}$$

The fuzzy rule base properties are

- The inference method is: "Mamdani".
- The T-norm and S-norms are: "Max" and "Min".
- The fuzzifier is: "Singletone".
- The defuzzifier is: "COG (center of gravity)".

6.3.3 The Reference Input

In order to estimate the position dependent part of the friction function, a proper reference input should be applied. To have a better performance in measuring the friction variation vs. position, the reference input should be such that it does not excite other components of the friction too much. One of the best signals might be the triangular signal at a low velocity. For this purpose the reference input shown in Figure 13 is applied to the system.

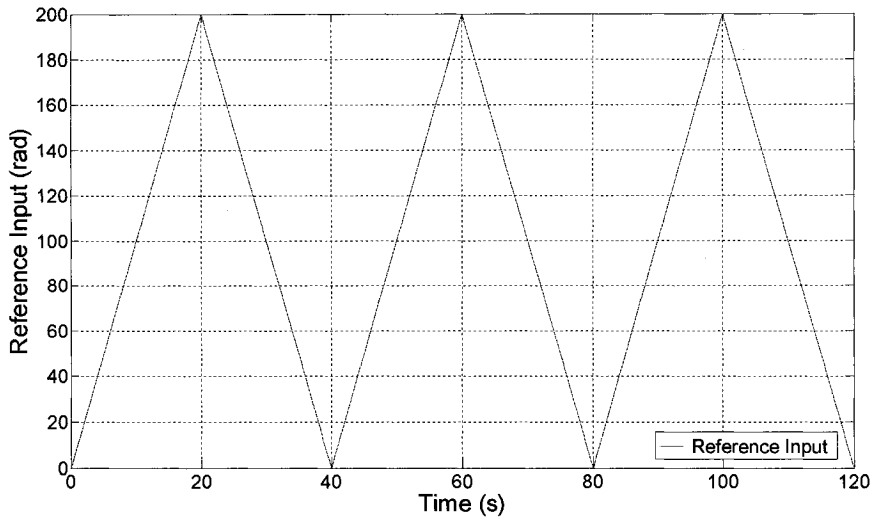

Figure 13. Reference input

6.4 Experimental Results

The reference input shown in Figure 13 is applied to the manipulator and the system response is shown in the following figures. Figure 15 shows the number of nodes participating in the network, while Figure 14 shows the tracking error. The nodes count has had an increase of about 30 nodes in about 45s. After this increase it has started to bounce around 27 and finally it has stopped at 28 nodes.

Different components of the plant input are also shown in Figure 16. It states that the dominant part of the input signal is u_a. The system is found to work in the working range A_d; i.e. $m = 1$ in all the time. The adaptive controller is found to be stable such that the system does

not leave the working range A_d and consequently the sliding component of the controller is always 0.

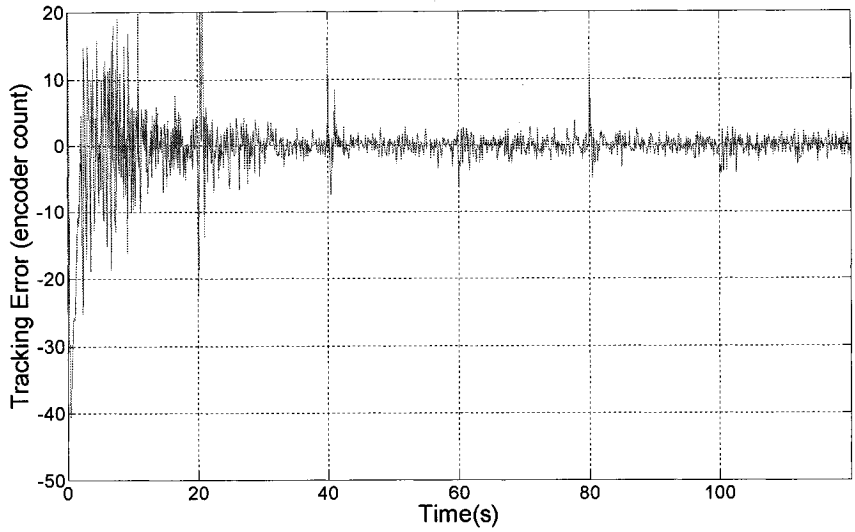

Figure 14. Tracking Error ($PI = 25.82$ - defined in (52))

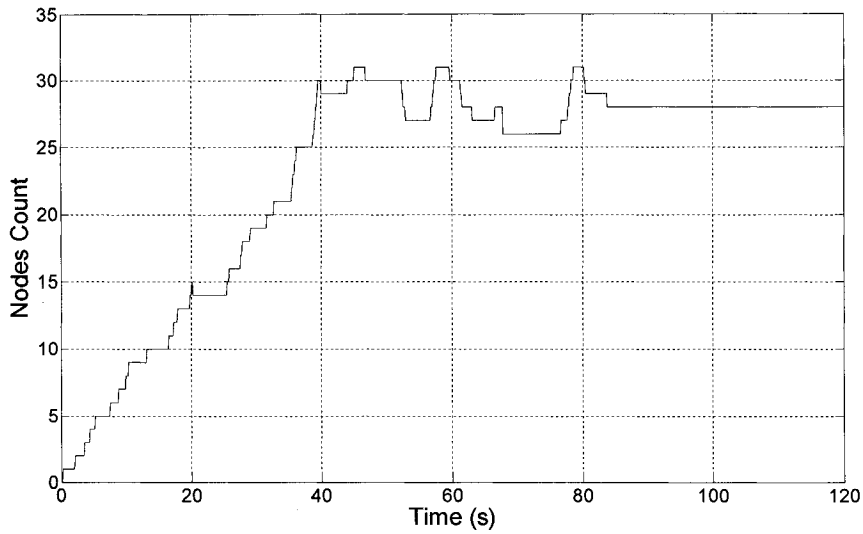

Figure 15. Wavelets Nodes Count

Figure 17 shows the position of the nodes over time. It shows where nodes are inserted and which nodes are removed. Figure 18 to Figure 23 show the histogram of the nodes' position over time. For example at time $t = 100$, there are 1 nodes in the neighborhood of 0, 3 nodes in

the neighborhood of 45 , 5 nodes in the neighborhood of 90 and so on. In Figure 25 to Figure 30 the system response for other types of node configuration are shown. Figure 25 and Figure 26 show the tracking error and nodes count for Fourier, Figure 27 and Figure 28 for RBF, and Figure 29 and Figure 30 for Sanner's method. The former figures clearly show the drawbacks of the Sanner's method.

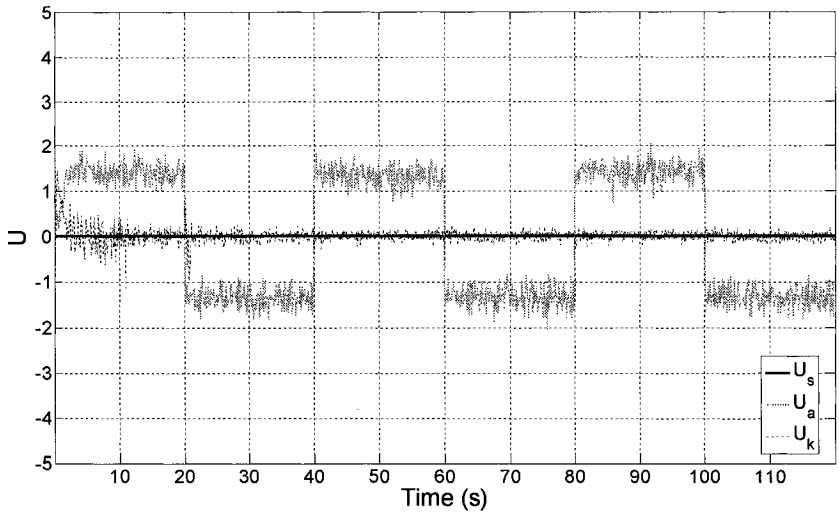

Figure 16. Different parts of u : u_s (solid bold), u_a (dashed bold), u_k (dashed)

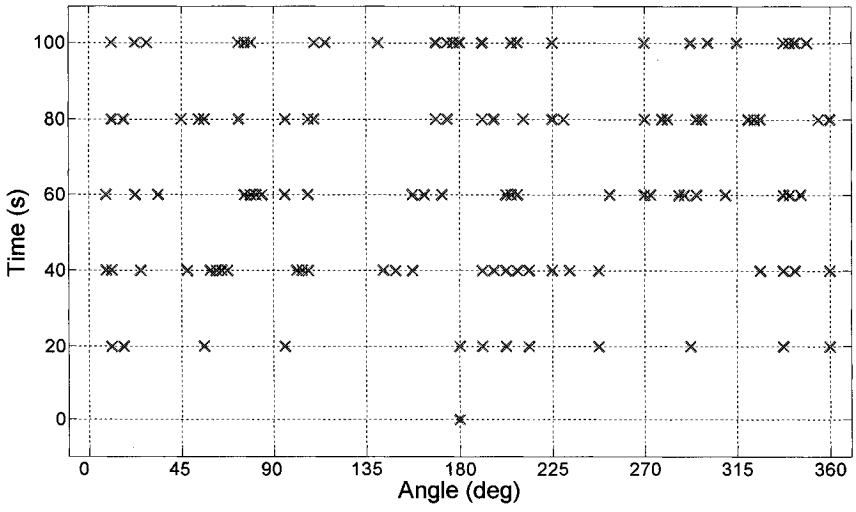

Figure 17. Node Positions

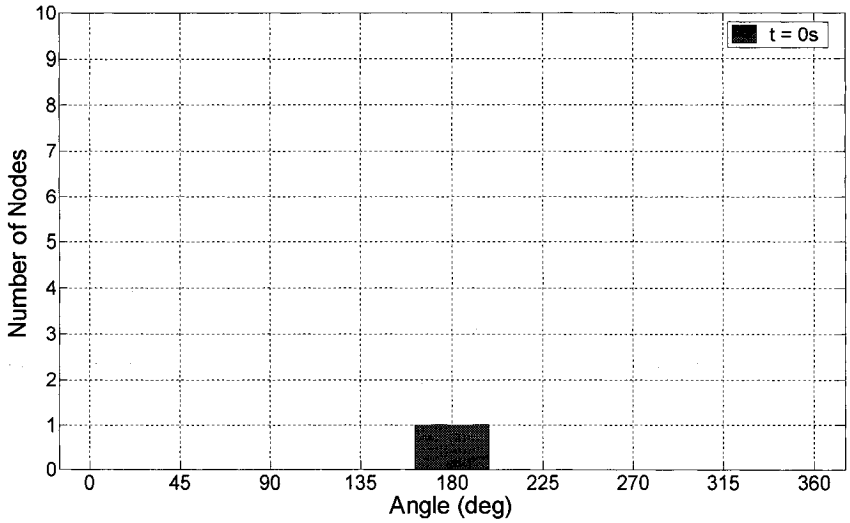

Figure 18. Node Positions Histogram at t = 0s

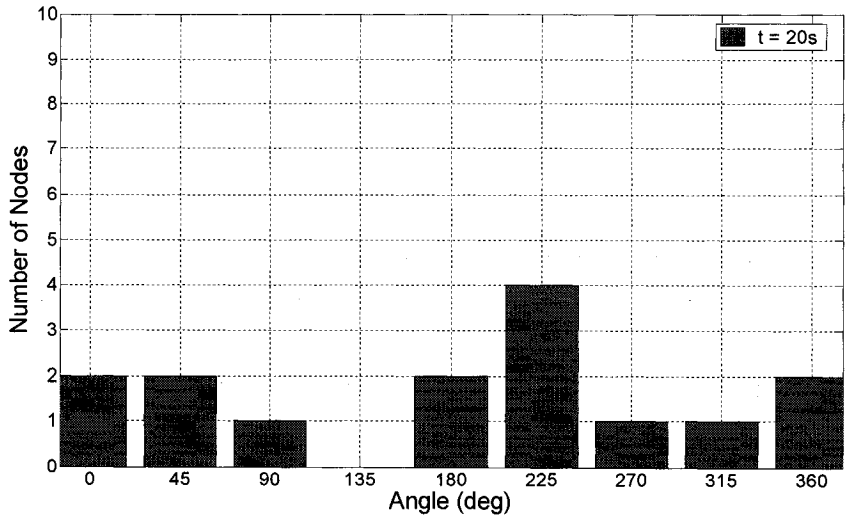

Figure 19. Node Positions Histogram at t = 20s

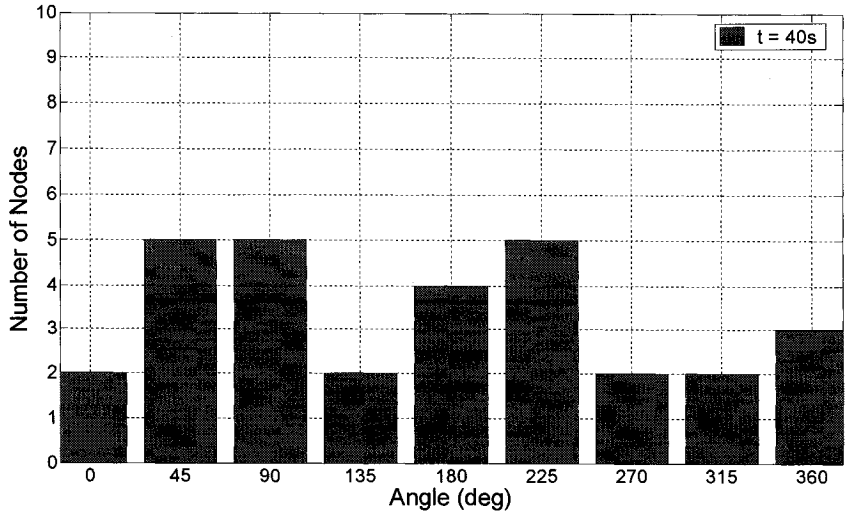

Figure 20. Node Positions Histogram at t = 40s

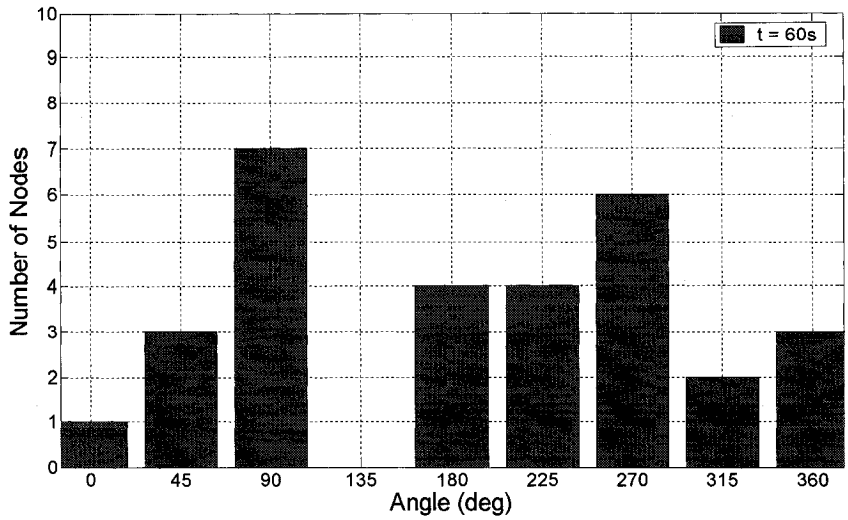

Figure 21. Node Positions Histogram at $t = 60$s

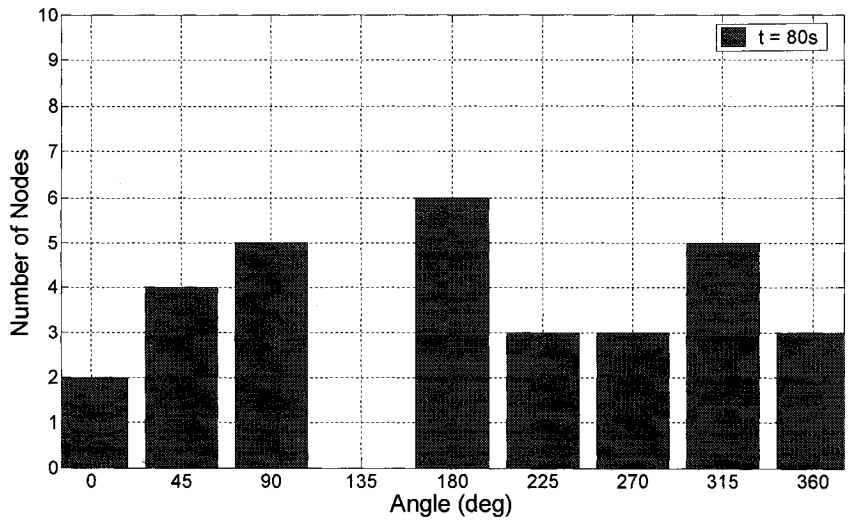

Figure 22. Node Positions Histogram at $t = 80$s

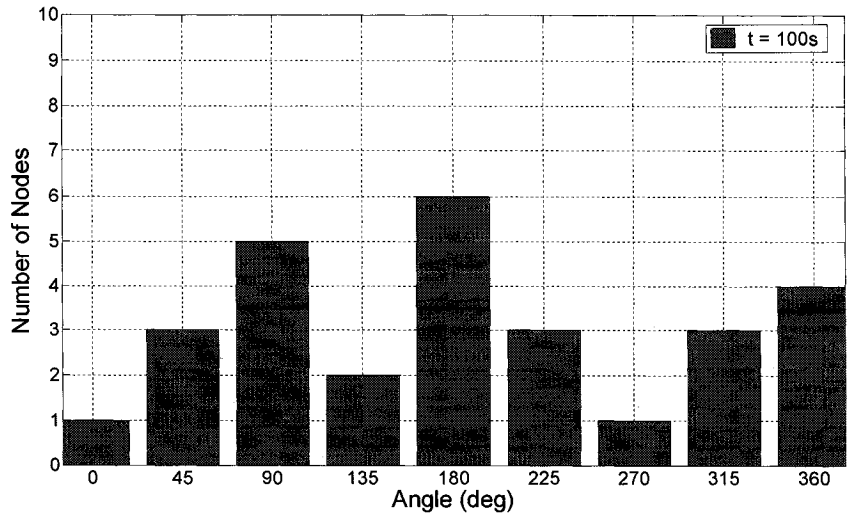

Figure 23. Node Positions Histogram at $t = 100$s

6.5 Comparison

The comparison between different methods and different node configuration is done by defining a performance index as

$$PI = \frac{\int_0^T e^2 dt}{T} \qquad (52)$$

For better comparison of different networks and methods, the performance indices and average nodes count in the last 20 seconds of the experiment are shown in the following table

Method	Performance Index	Average Nodes Count
Our Method – Wavelets	$PI = 25.82$	$N = 28$
Our Method - Fourier	$PI = 90.43$	$N \approx 8$
PI Controller	$PI = 208.79$	----
Our Method - RBF	$PI = 683.6$	$N \approx 45$
Sanner's Method - Wavelets	$PI = 6993.4$	$N \approx 220$

The above table clearly shows that our proposed method with the node configuration of wavelets results in the best performance. However it has the minimum number of nodes after the Fourier network. Although Fourier with about 8 nodes has the minimum number of nodes but this is not because of its good capabilities. It is because of the high frequency of the 9^{th} node. In the Fourier network the frequency of the 9^{th} node is 9 times of the frequency of the 1^{st} node. This frequency is so high that it is almost considered as noise and this node will be removed by FA2 because of chattering or too low weight. This is also verified by the fact that the number of nodes of the Fourier network did not converge to a certain value. This is caused by the similarity of the last couple of nodes to noise which results in consequent removal and add of those nodes to/from the network.

A key point in the success of the Fourier network to achieve a good performance in our experiments is the continuity and smoothness of the friction function. Fourier networks give a good performance in case of periodic and continuous unknown functions. But in case that the unknown function is not continuous or has sharp spikes, theoretically the Fourier network needs many nodes to give acceptable approximation accuracy. However this is practically impossible because of the large frequency of the nodes and consequently similarity to noise.

The PID has the 3rd rank. It has problems when the direction of the motion changes. Each time the sign of the velocity is changed the PI controller imposes a transient response to the system. This is because of the discontinuity of the friction function at the zero velocity. The PI controller is blind to this kind of discontinuity.

The next one is the RBF. The large PI of RBF is because of the time (x axis) locality of an RBF node. When a Fourier node is added to the network, its effect will be applied to the entire x axis; while the effect of adding an RBF network is limited to the neighborhood of its center. Adding an RBF node improves the approximation capability of the network only in its close neighborhood. This means the network does not have any approximation capability in regions having no nodes in the close neighborhood. So the network will have poor tracking performance in these regions.

Finally the Sanner's method on the end of the list has the PI of 6993.4. Looking at the nodes count diagram (Figure 30 and Figure 31) clearly shows the drawbacks of the Sanner's method. The network structure has a very fast and large change between about 180 and 260 nodes36. Comparing the diagrams for the Sanner's method we find:

1. The number of nodes in the network changes very fast.
2. The number of nodes is much higher in compare to other methods.
3. The number of nodes changes too much in each step.

The relation of the nodes count chattering with the Sanner's method is not so clear. Below is a brief step by step description of the cause:

- In the Sanner's method the number of nodes in all frequency levels are the same; for example the number of the nodes with smallest frequency level is equal to the number of nodes with the largest one. However, the range of large scale nodes is much higher than the low scale nodes.
- As the number of low frequency and the number of high frequency nodes are equal, and all the nodes are spread homogenously on the x axis, the time shift between the two adjacent nodes is the same for all frequency levels.
- As there should be a small enough time shift between two adjacent large frequency (small scale) nodes, the distance of two adjacent small frequency (large scale) nodes will be small either.
- Sanner states that the largest scale node should be selected such that it would be nearly constant on the set A. So the effective domain of low frequency nodes is so large that two adjacent nodes can be assumed to be the same. This is because the small time shift between these two can be easily neglected in compare to the node's domain.
- As the adjacent low frequency nodes are nearly the same, the update rate of their weights will also be nearly equal. So their weights will change almost the same.

- As the nodes are selected based on the thresholds on their weights, for two adjacent low frequency nodes, the thresholding condition will be satisfied almost the same time.

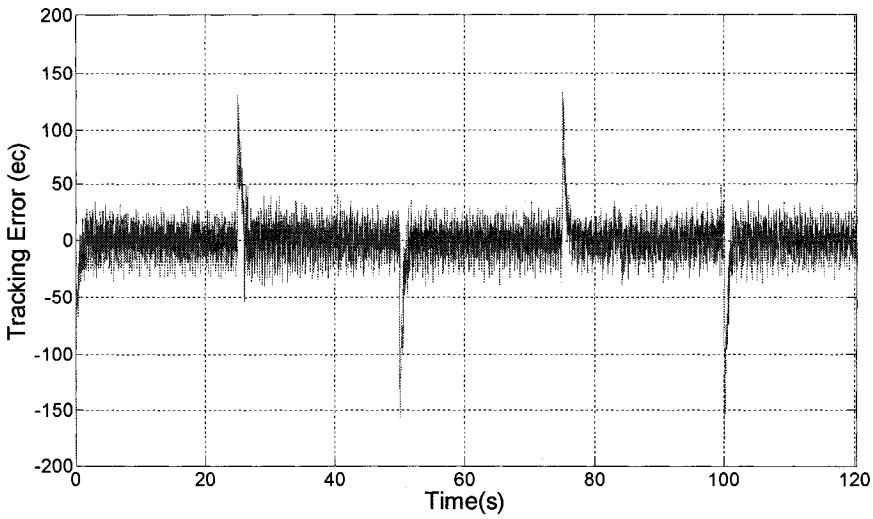

Figure 24. PID Controller (PI = 208.79)

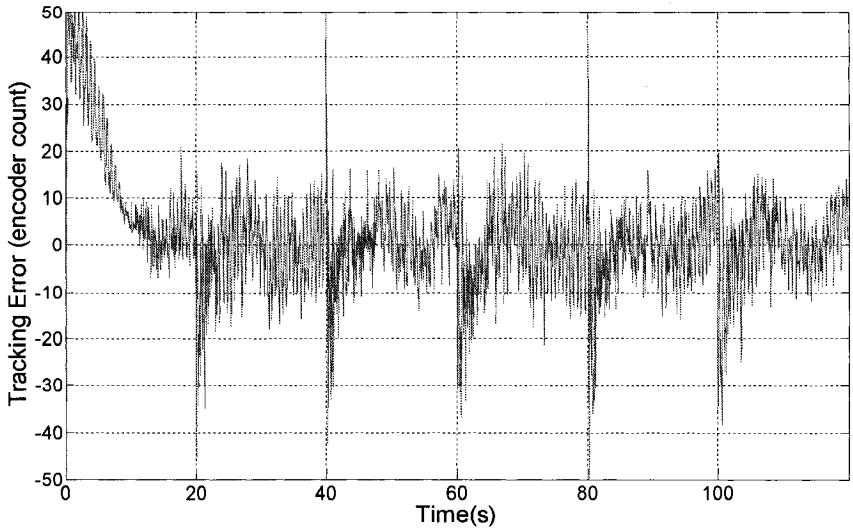

Figure 25. Dynamic Fourier tracking error ($PI = 90.43$)

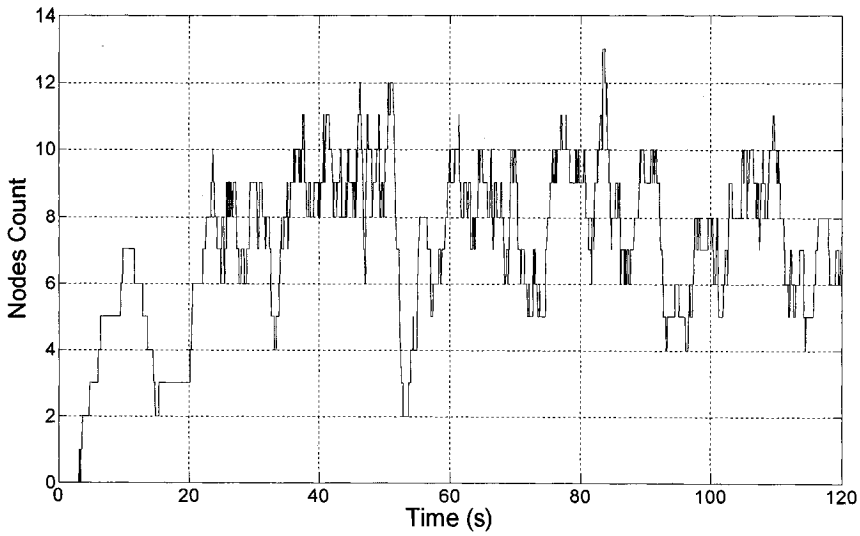

Figure 26. Dynamic Fourier node count

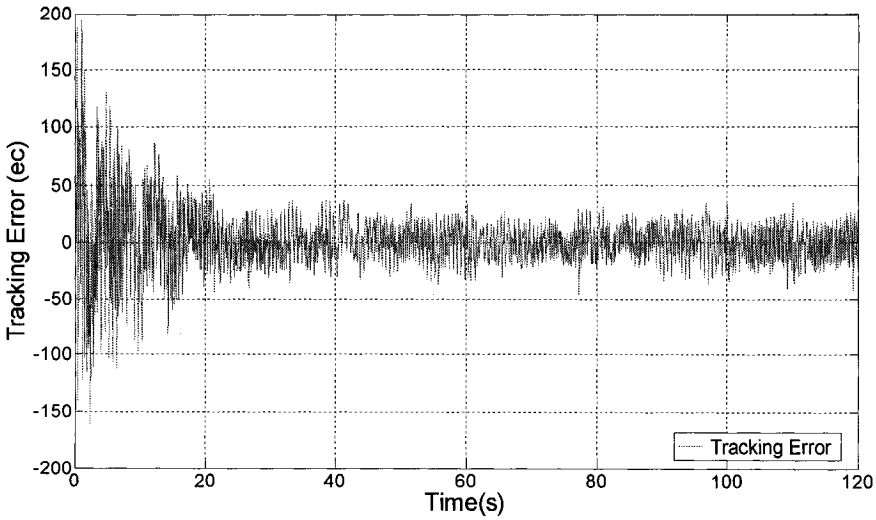

Figure 27. Dynamic RBF tracking error ($PI = 683.6$)

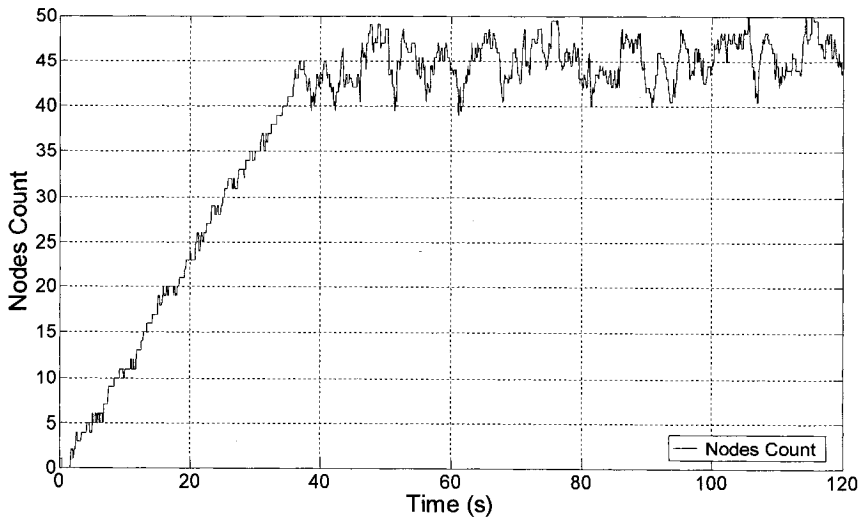

Figure 28. Dynamic RBF node count

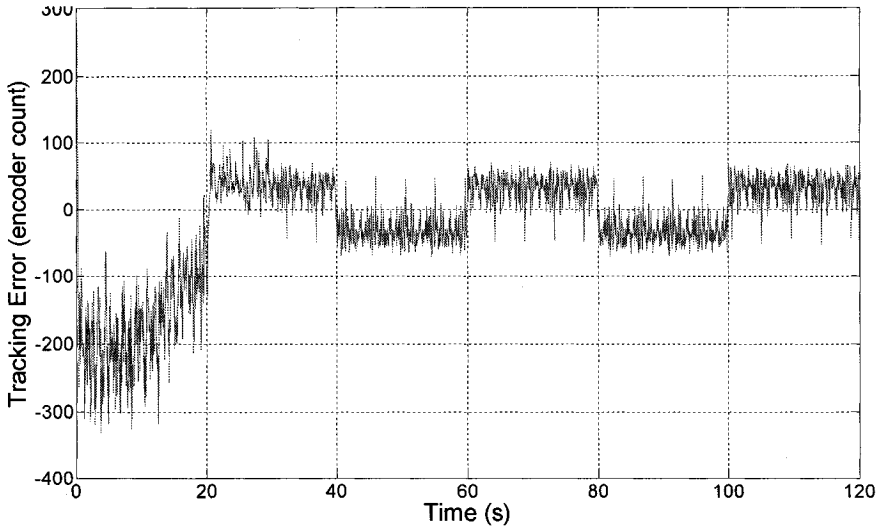

Figure 29. Sanner's method tracking error ($PI = 6993.4$)

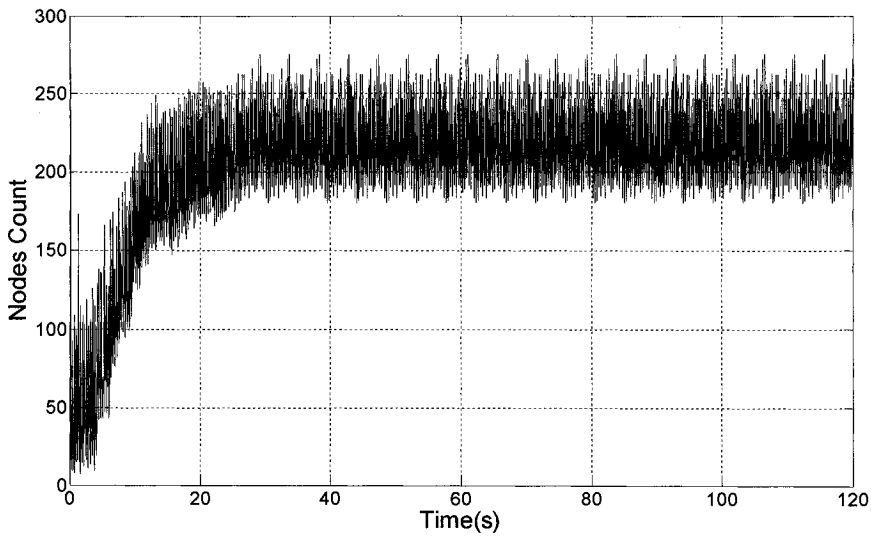

Figure 30. Sanner's method node count

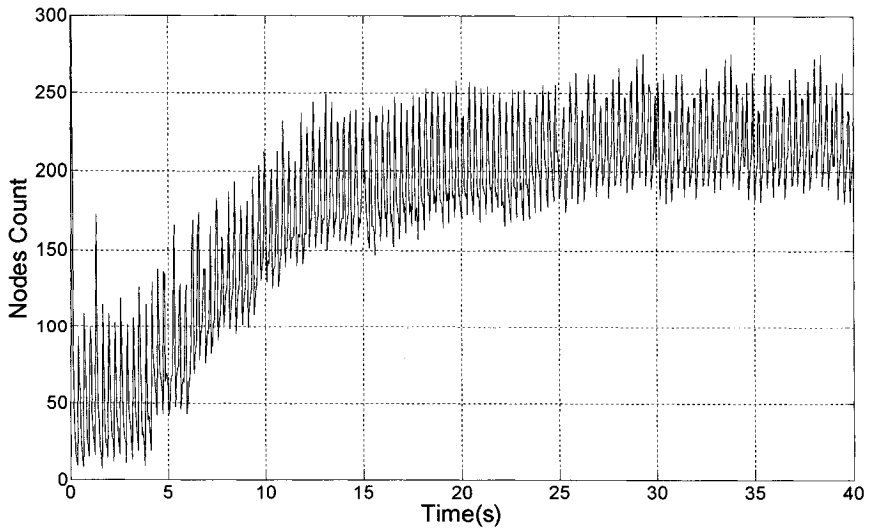

Figure 31. Sanner's method node count, the first 40s

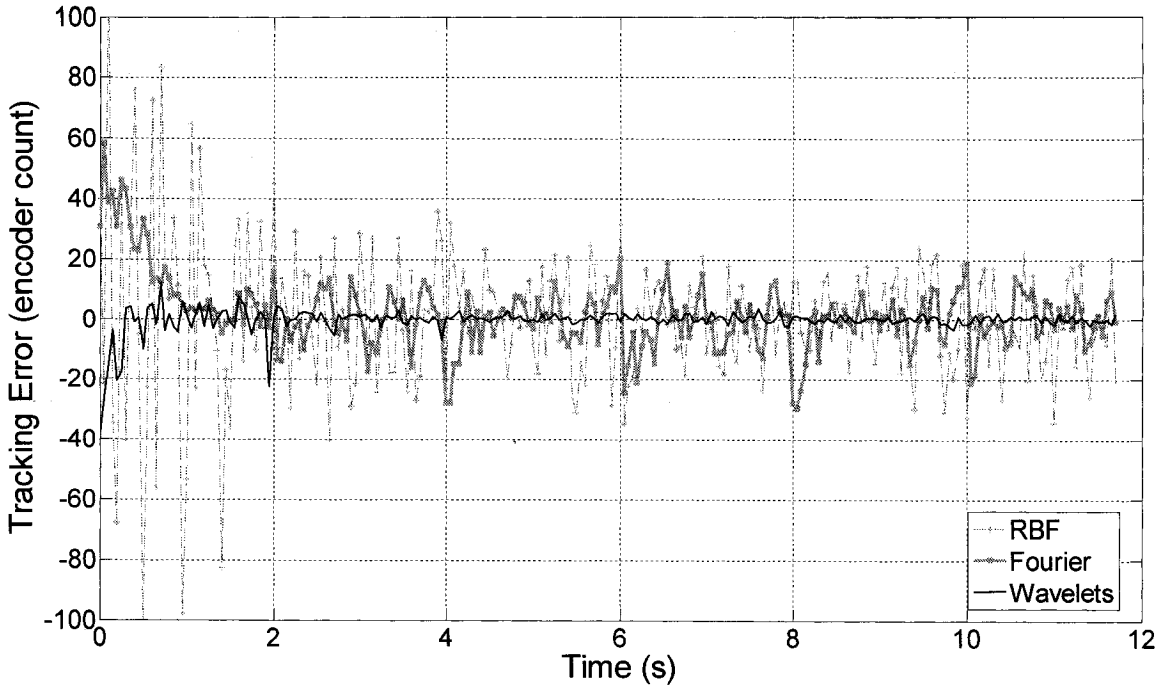

Figure 32. Tracking Error Comparison: RBF, Fourier, Wavelets

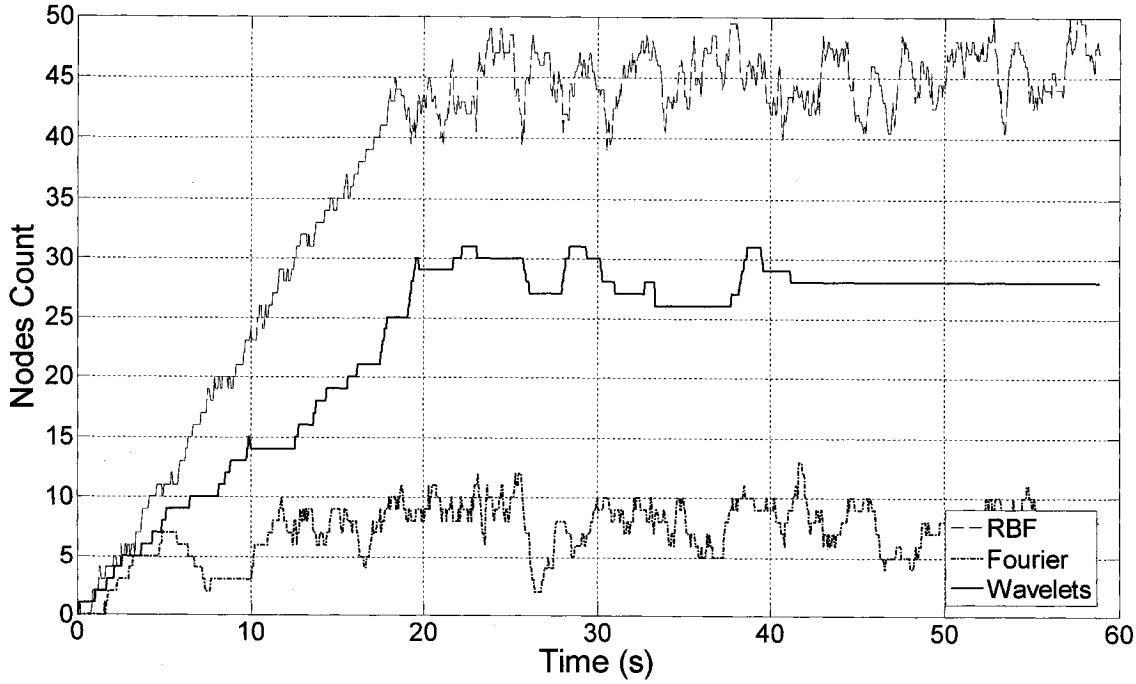

Figure 33. Nodes Count Comparison:
RBF (dot dashed), Fourier (dashed), Wavelets (solid)

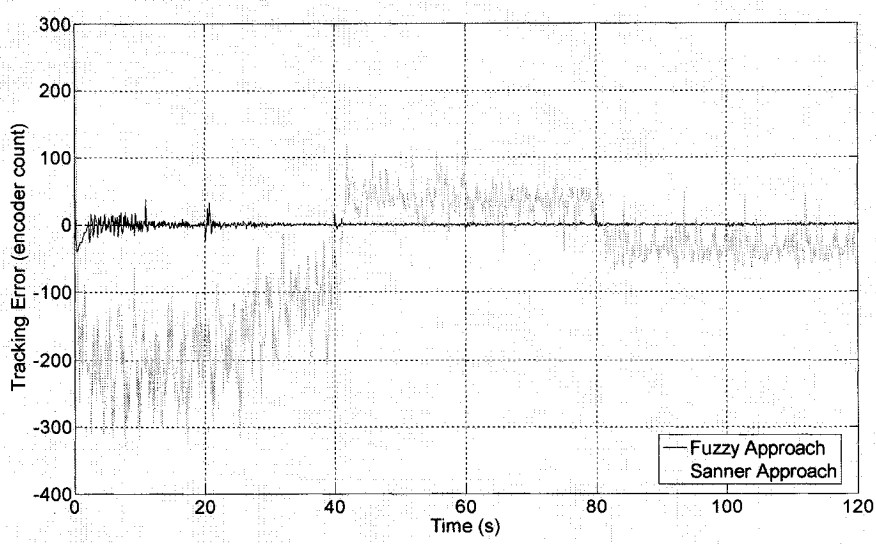

Figure 34. Tracking Error Comparison: Fuzzy Approach and Sanner's Method

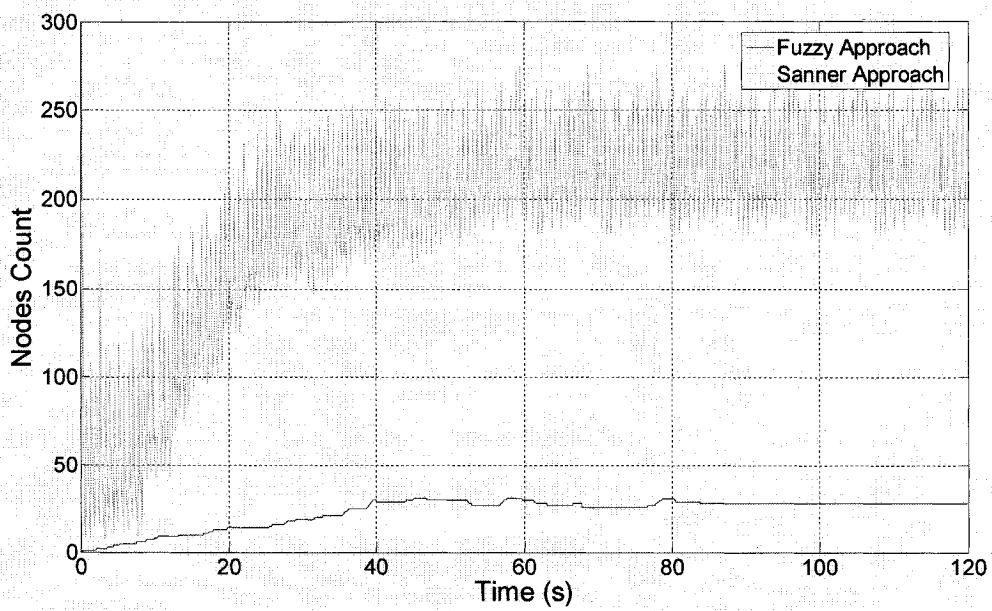

Figure 35. Node Count Comparison: Fuzzy Approach and Sanner's Method

7 Conclusions and Future Work

In order to compensate friction in harmonic drives an adaptive controller is proposed which takes the advantage of approximators as the best estimate of the system. The friction is found to be a function of velocity and position. The velocity dependence is modeled as a viscous friction with acceptable accuracy. However, a structurally dynamic approximation network is proposed which reconstructs the position dependency of friction. Three node types are used in the approximation network and the experimental results of each configuration are proposed. Among the reviewed types the wavelets proved to be the best having the best performance index and the less number of nodes.

To update the structure of the network two parallel fuzzy algorithms are designed. The fuzzy algorithms update the structure of the network with the target of decreasing the number of nodes to improve the system performance and robustness. These fuzzy algorithms decide about the adding and removal of nodes based on some heuristic phenomena derived from the system behavior and some notions in the switching control like dwell-time (Table 1 and Table 4).

This idea significantly improves the tracking performance of harmonic drive motors and makes them a better candidate to be used in precision positioning systems. Comparing the system responses clearly confirms the tracking error improvement. The proposed idea not only has a better performance in compare to the previous method proposed by Sanner, but also it uses a much fewer of nodes.

The main contributions of this thesis are:

1) **Modelling Friction with a Structurally Dynamic Wavelet Network**

   In this research the friction in harmonic drives is modelled by a structurally dynamic approximation network in conjunction with a model reference adaptive controller. Figure 32 shows that the wavelets prove to be the best due to its performance index and the low number of nodes. The wavelet achieves the tracking error limit of about 5 encoder counts which is a very acceptable performance in many precision positioning applications.

2) **A Fuzzy Algorithm to Update the Structure of the Approximation Network**

   In this thesis, for the first time, a fuzzy based algorithm is proposed to dynamically change the structure of an approximation network with the target of decreasing the number of participated nodes. Figure 34 clearly shows the effect of the new algorithm. The number of nodes participating to the network is about 10 times less than Sanner's method while having about 10 times more precision.

3) **The Experimental Verification**

   The proposed methods are applied and verified by the experiments. The experimental results show a good improvement to the tracking error and number of nodes by the new method.

Future Works

To improve the method further, the following areas could be investigated in the future work.

1) The set of rules of the fuzzy algorithms can be improved.

2) The definition of inputs can be changed. Other states of the system can be added to the inputs.

3) A major improvement is an enhancement in the fuzzy algorithm outputs. The outputs can be improved to have a node half in the network and half not. This can be implemented by having the range of $[0,1]$ for α_i, not only strict values of 0 and 1.

4) In this research the velocity dependency of friction is considered to be linear. Another network can estimate the velocity dependency of friction. In our research we missed this because it needs too much computation that could not be done on a single computer.

5) The calculations can be distributed in two or more computers. This makes this idea applicable to systems having complex friction dependency. An example is the aircraft systems which the drag force is a complex function of several system states.

6) Not only friction but also any state dependent complex force in the system can be approximated with this method. The idea can be generalized to estimation of any force dependent to any number of states in a system.

References

[1] N.A. Aliev., "A study of the dynamic behavior of flexible gears in harmonic drives". *Soviet Engineering Research*, 66(6):7--11, 1986.

[2] Taghirad, H., "On the modelling and identification of harmonic drive systems". *CIM Internal Report* , 1995.

[3] Taghirad, H., Jan 29, "On the modelling and identification of harmonic drive systems". *CIM-TR-97-02,* 1997.

[4] Taghirad, H., and Belanger, P. R., "An Experimental Study on Modeling and Identification of Harmonic Drive Systems". *In Proceedings of the 35th Conference on Decision and Control*, pp. 4725–30.

[5] Taghirad, H., and Belanger, P. R., 1998. "Friction compensation and H_∞-based torque control of harmonic drive systems". *IEEE Inretnational Conference on Robotics and Automation.*

[6] Taghirad, H., and Belanger, P., "Modelling and Parameter Identification of Harmonic Drive Systems". *Journal of Dynamic Systems, Measurements, and Control, ASME*, pp. 65–85, 1998.

[7] Hsia, L. "The Analysis and Design of Harmonic Gear Drives". *Proceedings of the 1988 IEEE International Conference on Systems, Man and Cybernetics, Vol 1*, pp. 616–619.

[8] Legnani, G., and Faglia, R., "Harmonic drive transmission: the effect of their elasticity, clearance and irregularity on the dynamic behaviour of an actual scararobot". *Robotica, Vol 10* , pp. 369–375, October 1992.

[9] Marilier, T., and Richard, J., "Nonlinear mechanic and electric behaviour of a robot axis with a harmonic drive gear", Robotics and Computer Integrated Manufacturing, vol 5 (2/3):129-136, 1989.

[10] Chedmail, P., and Martineau, J., 1996. "Characterization of the friction parameters of harmonic drive actuators". *International Conference on Dynamics and Control of Structures in Space,* **1** , pp. 567–581.

[11] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30H7I:1083–1138, 1994.

[12] P.-A. Bliman. Mathematical study of the Dahl's friction model. *European Journal of Mechanics. A/Solids*, 11(6):835–848, 1992.

[13] P.-A. Bliman and M. Sorine. Friction modelling by hysteresis operators. application to Dahl, sticktion and Stribeck effects. In *Proceedings of the Conference "Models of Hysteresis", Trento, Italy*, 1991.

[14] Slocum, A.H., "Precision Machine Design" Prentice Hall, New Jersey, 1992

[15] Moghaddam, M.M., Goldenberg, A.A., "Nonlinear Modelling and Robust H_∞ Based Control of Flexible Joint Robots with Harmonic Drives", Proc. IEEE Conf. Robotics and Automation. April 1997, pp 3130-3135

[16] Popovic, M.R., Gorinevsky, D.M., Goldenberg, A.A., "High-Precision Positioning of a Mechanism with Nonlinear Friction Using a Fuzzy Logic Pulse Controller", IEEE Trans. on Control Systems Technology, Vol. 8, No. 1, pp. 151-158, 2000.

[17] Haessig, D.A., Friedland, B., "On the Modelling and Simulation of Friction" J. of Dynamic Systems, Measurement, and Control, Vol. 113, PP354-362, Sept. 1991.

[18] Karnopp, D. Computer simulation of slipstick friction in mechanical dynamic systems. *Journal of Dynamics Systems, Measurement, and Control*, 107(1):100–103, 1985

[19] Dahl, P. R. A solid friction model. TOR-158(3107-18). The Aerospace Corporation, El-Secundo, California, 1968.

[20] Coulomb, C. A. Théorie des machines simples. *Mémoires de Mathématiques et de Physique de l'Académie des Sciences*, pages 161–331, 1785.

[21] Stribeck, R. The key qualities of sliding and roller bearings. *Zeitschrift des Vereines Seutscher Ingenieure*, 46(28,39):1342–1348,1432–1437, 1902.

[22] Rabinowicz, E. The nature of the static and kinetic coefficients of friction. *Journal of applied physics*, 22(11):1373–1379, 1951.

[23] Carlos Canudas de wit, Olsson, H., Astrom, K.J., Lischinsky, P., "A New Model for Control of Systems with Friction", IEEE Transaction Automatic Control Vol. 40, No. 3, PP 419-425, Mar. 1995.

[24] Altpeter, F., Ghorbel, F.H., Longchamp, R., " Relationship Between Two Friction Models: A Singular Perturbation Approach", Proc. IEEE 37th Conf. Decision and Control, PP 1572-1574, Dec. 1998.

[25] Kelly, R., Llamas, J., " Determination of Viscous and Coulomb Friction by Using Velocity Response to Torque Ramp Inputs", Proc. IEEE International Conf. Robotics and Automation, PP 1740-1745, May 1999.

[26] Carlos Canudas de Wit, and Lischinsky, P., " Adaptive Friction Compensation with Partially Known Dynamic Friction Model", International Journal of Adaptive Control and Signal Processing, Vol. 11, PP 65-80, 1997

[27] Swevers, J., Al-Bender, F., Gansemen, C.G., and Prajogo, T., "An Integrated Friction Model Structure with Improved Presliding Behaviour for Accurate Friction Compensation" IEEE Transaction Automatic Control Vol. 45, No. 4, PP 675-686, Apr. 2000.

[28] Tariku, F.A., Rogers, R.J., "Improved Dynamic Friction Models for Simulation of One-Dimensional and Two-Dimensional Stick-Slip Motion", Journal of Tribology Vol. 123 pp 661-669, Oct. 2001,

[29] Wu, R.H., Tung, P.C., " Studies of Stick-Slip Friction, Presliding Displacement, and Hunting. J. of Dynamic Systems, Measurement, and Control, Vol. 124, PP 111-117, Mar. 2002.

[30] Popovic, M.R., and Goldenberg, A.A., " Modelling of Friction Using Spectral Analysis", IEEE Transaction on Robotics and Automation, Vol.14, No.1, PP 114-122, Feb.1998.

[31] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[32] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximations. *Neural Networks*, 2:359–366, 1989.

[33] Tuttle, T., and Seering, W., "Nonlinear modeling and parameter identification of harmonic drive robotic transmissions". *IEEE International Conference on Robotics and Automation,* 3, pp. 3027–3032, May 21-27 1995.

[34] Presna Gandhi, Fathi Ghorbel, J. D., "Modeling, identification, and compensation of friction in harmonic drives". *Proceeding of the 41st IEEE Conference on Decision and Control*, pp. 160 – 166, Dec 2002.

[35] Gandhi, P. S., "Modelling and control of nonlinear transmission attributes in harmonic drive systems". Ph.D. Thesis, Rice University, Houston, Texas, 2001.

[36] H. Taghirad, "Robust torque control of harmonic drive systems", *PhD thesis, McGill University*, March 1997.

[37] Johnson, C.T.; Lorenz, R.D, "Experimental identification of friction and its compensation in precise, position controlled mechanisms", Industry Applications, IEEE Transactions on, Volume 28, Issue 6, Page(s):1392 – 1398, Nov.-Dec. 1992.

[38] Han, S. I.; Kim, J. S.; Choi, J. W., "Robust Nonlinear H_2 / H_∞ Control for a Parallel Inverted Pendulum with Dry Friction", *JSME International Journal Series C,* vol. 45, no. 1, pp. 194-203, Ingenta, 2002.

[39] Jeng-Shi Chen and Jyh-Ching Juang, "A Robust Adaptive Friction Control Scheme of Robot Manipulators", *ICGST International Journal on Automation, Robotics and Autonomous Systems*, pp 11-19, Vol VI, 2006.

[40] L. Cai and G. Song. *A smooth robust nonlinear controller for robot manipulators with joint stick slip friction.* Proceeding of IEEE International Conference on Robotics and Automation, 449-454, 1993.

[41] Vicente Parra-Vega, "Chattering-free Dynamical TBG Adaptive Sliding Mode Control of Robot Arms with Dynamic Friction for Tracking in Finite-time", *International Conference on Robotics and Automation*, 2001

[42] B. Bona, M. Indri, "Friction Compensation and Robustness Issues in Force/Position Controlled Manipulators", *IEE Proceedings, Control Theory and Applications*, vol. 142, n. 6, pagg. 569-574, 1995

[43] Carlin, M., "Radial basis function networks and nonlinear data modelling". *Proceedings of Neuro-Nimes Neural Networks and their Applications, EC2* , pp. 623–633, 1992.

[44] Mehmetnder Efe, O. K., "Radial basis function neural networks in variable structure control of a class of biochemical processes". *IECON '01, The 27th Annual Conference of the IEEE Industrial Electronics Society,* 1 , pp. 13–18, 2001.

[45] R. Sanner, J. Slotine, "Direct adaptive control using Gaussian networks". *IEEE Trans. Neural Networks*, 3 (6), 1992.

[46] Jun Zhang, Walter, G. M. Y. W. N. W. L., "Wavelet neural networks for function learning". *Signal Processing, IEEE Transactions on Speech, and Signal Processing*, 6 (43), pp. 1485 – 1497, , June 1995.

[47] Zhang, Q., March, "Using wavelet network in nonparametric estimation". *IEEE Transactions on Neural Networks*, 8 (2), pp. 710–718, 1997.

[48] Becerikli, Y., O. Y. K. A., "On a dynamic wavelet network and its modeling application". *Lecture Notes in Computer Science (LNCS),* 2714, pp. 710–718, June, 2003.

[49] R. Sanner, J. J. Slotine, "Structurally dynamic wavelet networks for the adaptive control of uncertain robotic systems". *Proceedings of the 34th IEEE Conference on Decision and Control,* 3 , pp. 2460–2467, Dec 1995.

[50] Renals, Steve & Rohwer, Richard, "Phoneme Classification Experiments Using Radial Basis Functions". *International Joint Conference on Neural Networks, vol.1,Washington DC*, June 18-22, pp.461-467, 1989.

[51] R. Sanner, J. J. Slotine, "Stable Adaptive Control and Recursive Identification Using Radial Gaussian Networks". *Proceedings of the 30h IEEE Conference on Decision and Control,* 3, pp. 2116-2123, 1991.

[52] R. Sanner, J. J. Slotine, "Gaussian Networks for Direct Adaptive Control". *IEEE Transaction on Neural Networks,* 3 , pp. 837–863, Dec 1992.

[53] R. Sanner, J. J. Slotine, "Direct Adaptive Control Using Gaussian Networks". *Proceedings of ACC,* June 1991.

[54] Brandon W. Gordon, "Coordinated Control of Two Macro/Micro Manipulators for Fibre Pigtailing Automation", Master Thesis, Mechanical Engineering Department, Massachusetts Institute of Technology, 1995

[55] M. Cannon, J. J. E. Slotine, Space-frequency localized basis function networks for nonlinear system estimation and control, Neurocomputing 9, March 1995.

[56] Hwang, C.L., "Fourier series neural network-based adaptive variable structure control for servo systems with friction", IEEE Proceedings in Control Theory and Application, Volume 144, Issue 6, pp 559-565, 1997.

[57] C. Canudas de Wit, S.S. Ge, "Adaptive Friction Compensation for Systems with Generalized Velocity/Position Friction Dependency", Proceeding of IEEE Control on Decision and Control, Volume 3, pp2465-2470, 1997.

[58] B. Friedland, Y.-J. Park, "On adaptive friction compensation", Automatic Control, IEEE Transactions on , Vol: 37 , Issue: 10, pp:1609-1612, Oct. 1992.

[59] J. Juang, J.i Chen, "On combining adaptive estimation and robust control for friction compensation", Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on , Volume: 5 , Pages:4396 - 4400 Vol.5, 15-19 June 2004.

[60] J. Zhung, G.G. Walter, "Wavelet Neural Networks for Function Learning", IEEE Transaction on Signal Processing, Vol 43, No. 6, June 1995.

[61] Yu Kun Yang, "Impulse Control of Harmonic Drive Motors for High Precision Positioning Applications", Masters Thesis, March 2004, Mechanical Eng. Dept. Concordia University, Montreal, Canada.

[62] Sheng-Tun Li, Shu-Ching Chen, "Function Approximation using RobustWavelet Neural Networks", 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02), November 04-06, 2002, Washington, DC

[63] Ingrid Daubechies, "Ten Lectures on Wavelets", Publisher: SIAM, May 1, 1992, ISBN:: 0898712742.

[64] P.R. Bélanger, "Estimation of Angular Velocity and Acceleration from Shaft Encoder Measurements", Proceeding of IEEE International Conference on Robotics and Automation, May 1992.

[65] Sinha, N.K., B. Szavados and D.C. di Cenzo, "New high precision digital tachometer", Electron. Lett., Vol 7, pp174-176, 1971.

[66] Hoffman de Visme, G., "Digital processing unit for evaluating angular accelations", Electron. Eng., vol 40, pp 183-188, 1968.

[67] Dunworth, A., "Digital instrumentation for angular velocity and acceleration", IEEE Trans. Instrum. Meas., vol IM-18, pp 132-138, 1969.

[68] Habibullah, B., H. Singh, K.L. Soo and L.C. Ong, "A new digital speed transducer", IEEE Trans. Ind. Elecntron. Contr. Indstr., vol IECI-25, pp 339-343, 1978.

[69] Wallingford, E.E. and J.D. Wilson, "High resolution shaft speed measurements using microcomputer", IEEE Trans. Instr. And Meas., vol IM-26, pp 113-116, 1977.

[70] Pola, G., Polderman, J.W. & Di Benedetto, "Balancing dwell times for switching linear systems", *Proc. 16th Intern. Symposium on Mathematical Theory of Networks and Systems [MTNS-2004]*, Leuven, Belgium, July 5-9, (6 pp.) ISBN 90-5682-517-8

Appendix A

RH-5A-5502 SPECIFICATIONS

Performance Specifications	
DC Voltage (VDC)	12.0
Shaft Speed (rpm)	55
Continuous Current (amps)	0.50
Continuous Torque (In-lbs)	2.60
Output Power (HP)	0.0023
Torque Constant (oz-in / amp)	0.0140
Rotor Inertia (oz-in-sec^2)	0.2240
Motor Type	
DC Construction	Permanent Magnet
Commutation	Brush
Shaft Orientation	In-line; Single-ended
Gearing Options	
Gearing	Gearmotor
Gearhead Model	
Gear Type	Harmonic
Gearbox Ratio (: 1)	80.00
Gearing Efficiency (%)	
Housing / Enclosure	
Units	Metric
Motor Shape	Cylindrical Body
Diameter / Width (inch)	0.79
Length (inch)	3.15
NEMA Frame	
Options	
Extreme Environment	
Other Specifications	
Feedback	Integral Encoder; Integral Tachometer (optional feature)
Features	Servomotor; Brake
Environment	
Operating Temperature (F)	32 to 104
Shock Rating (g)	30.0
Vibration Rating (g)	2.50
Notes	Available in analog & digital control unit, zero backlash, continuous rating