Online Global Learning in Direct Fuzzy Controllers

By

<u>Muhammad Umair Arif</u>

Thesis Supervisor

<u>Dr. Muhammad Bilal Kadri</u>

Pakistan Naval Engineering College, NUST
Department of Electronic and Power Engineering
MS Electrical Controls

May, 2013

PNEC
Department of Electronic and Power Engineering
MS Electrical Controls

Master Thesis Topic

Online Global Learning in Direct Fuzzy Controllers

Presented by

Muhammad Umair Arif

Thesis Supervisor

Dr. Muhammad Bilal Kadri

GEC members

Cdr. Dr. M Farhan PN
Cdr. Dr. S. Sajjad Haider Zaidi PN
Cdr. Dr. Attaullah Memon PN

Acknowledgement

First and foremost, I am obliged to thank Allah Subhana-watala for giving me all the blessings of the world in terms of my family, in terms of health, in terms of opportunities, in terms of the required ability, in terms of the intellect to think and learn, in terms of financial support and in terms of my existence as a whole. If all this was not bestowed upon me, I would never be able to do anything in life. All Praise to Allah Subhana-watala who gave me a supportive advisor like Dr. Muhammad Bilal kadri. His relentless support during my entire work is commendable beyond words. I would finally thank my GEC members for their help and sincere wishes.

Table of Contents

List of Tables

List of Figures

Figure 3.1: Stages of the applied control scheme

Abstract

Controlling non-linear systems in the most intelligent and efficient manner is the biggest and ongoing challenge for control engineers and scientists. Current research revolves around developing methods for control processes which can work with no plant information at all. Implementing online global learning in direct fuzzy controllers is an attempt to pursue this objective. Starting from an empty knowledge free controller, these controllers are able to learn, adapt and tune themselves with the desired results. Online tuning of rule consequents of the fuzzy controllers is discussed on the basis of plant output error i.e., adaptive control scheme was utilized. Modifying the adaptive controllers parameters by using only the qualitative information of the plant i.e., the monotony of input and output along with the adequate no of iterations required for learning was the basis of this work.

Detailed error analysis on linear and non-linear plants concluded that the learning/tuning of the scheme was drastically improved with the correct combination of controller parameters. Finally, this empty knowledge free adaptive fuzzy controller was tested on a real plant in which it was successful in online controlling of water level on a real three tank system.

CHAPTER 1

1. Introduction

1.1 Motivation

Every practical system in this world consists of some complexities and non-linearities. This is the reason why controlling non-linear systems in the most intelligent and efficient manner is the biggest and ongoing challenge for control engineers and scientists. Current research revolves around developing methods for control processes which can work with no plant information at all.

Within the domain of control schemes, Adaptive control scheme is one of the most popularly known. It has revolutionized the process industries because of its abilities to control processes with constraints and nonlinearities.

In mid 70s, a new branch of controls emerged i.e. Fuzzy controls and it has been phenomenal in being a successful control technique used by control engineers worldwide.

Fuzzy controller is primarily based on the concepts of fuzzy set theory and fuzzy logic and proved themselves to be universal approximators. It is their proven ability to approximate any continuous function to any desired accuracy. [1]

Keeping in view the proficiency of the two above mentioned control systems, modern researchers anticipated exceptional results in merging the two schemes and started focusing on the integration of the concepts and techniques from conventional adaptive control into fuzzy control mechanism. Blending of both schemes i.e. fuzzy modelling and adaptive Control has produced phenomenal results in the field of control systems.
The consistently increasing research in this area inspired the author to choose ‘Fuzzy Model based adaptive Control for controlling different nonlinear system’ as thesis work.

1.2 History of fuzzy controllers

Research work [2] done regarding the application of learning techniques to process control was the basis of the concept of Fuzzy control. At that time, conventional/ analytical control theory was unshakable in the areas of learning and adaptive control systems. [3].

This status quo was challenged and eventually taken over because of the increasing application of fuzzy controllers in Japan[3] brought fuzzy control into consideration and now fuzzy control and allied techniques such as self-organizing fuzzy control, neural networks, genetic algorithms and so on, provide an alternate paradigm to the analytic control theory. Fuzzy controls may also employ non analytic approaches to control which are based on decision making approaches derived from artificial intelligence.

There existed a misunderstanding that a controller must be designed by a good analytical treatment only and the similar controller if designed by other techniques such as based on methods of artificial intelligence would essentially be highly complex to meet the desired level of sophistication. Zadeh's paper [4] persuaded the world to use a fuzzy rule-based approach. The controller design based on this paper attracted several researchers by the surprising ease in implementing a fuzzy controller.

Researchers came across many situations where conventional approach to control failed because of the complex systems whose mathematical analysis was unavailable. Moreover, the control theory was applied to some other systems including telecommunication, management, economic systems etc. that were modeled in the form of differential equations instead of deriving their behavior form first principles of Physics. For instance, in the field of telecommunications networks, it is required to assess the performance of the network on regular basis and keep reconfiguring the network to maintain the quality of service metrics at the preferred values. For such systems to work, human interaction is needed. Even if the mathematical laws that describe the input/output behavior of such systems are developed, there still exists the uncertainty that the mathematical law will remain valid for these systems since humans may alter their parameters that will unpredictably change the behavior of the systems.

In addition to this, there are certain systems for which highly complex mathematical models are present and the design of their controllers requires a lot of time. On the other hand, if the controller is being designed on fuzzy logic, the requirement of knowledge of the systems will be less as compared to that for a conventional mathematical based controller.

As an example, consider the process control system. The mathematical equations which depict the behavior of such a process are too complex and almost impossible to derive with a high degree of accuracy. Also is the case with Cement kilns (the first industrial application which used fuzzy control [5]) and Steel making.

Fuzzy control's approach out rightly becomes the best control solution for all the above mentioned systems because they have more in common with human systems. Since Fuzzy control base their working on intelligent rule applications and therefore perform much better than the classical non-linear dynamic systems.

1.3 Description of a fuzzy controller

Fuzzy controllers are simply rule-based controllers for which a designer of the system can provide all the necessary information and no expert operator in terms of mathematical equations is essentially needed to derive the knowledge to be used.

A rule-based control model is capable of producing better controllers than analytic control model. The most favorable properties of a fuzzy logic based controller is its robustness, simple design and easy to implement hardware. This is evident by the recent industrial experience [6].

Consider a simple example of introducing intelligence in a washing machine regarding the “turbidity” of water. One would not be content by just displaying the “turbidity” value to the user of the washing machine because it would be pointless for the user. However, when this new information is understood i.e., Turbidity value indicates how the clothes washed were soiled and how dirty they were, we can interpret it in terms of rules i.e., we automatically adjust the washing time with respect to turbidity value. Therefore this simple knowledge representation (the strength of fuzzy control approach) introduced the associated intelligence for the user.

1.4 Adaptive fuzzy controllers

The first adaptive fuzzy controller was invented and presented by Procyk and Mamdani in 1979 [7]. It was called the linguistic self-organizing controller (SOC), which defined the scope of adaptive control. Their learning algorithm modified the rules which were responsible for poor performance.

An Adaptive approach in fuzzy controllers deals with unpredictable behaviour of systems to be controlled. Therefore when the real implementation is accomplished [8] they are able to perform better than non-adaptive control policies.

Many researchers have contributed different ideas in making the online self tuning adaptive fuzzy controller more robust and effective. In this thesis, one of the most recent approaches outlines by Pomares [9] will be studied.

1.5 Scope of the Work

The aspiration for this work is to study, understand and add value to the emerging approach of online self-organizing fuzzy controllers which do not use any prior knowledge or any offline pre-training. It is assumed here that the plant is unknown and input output data of the plant is made available online.

Pomares[10], proposed an adaptive block which tries to adapt the plant output. This block implements a modified SOC to ascertain the plant's current state and implements an algorithm for updating the rules. It uses basic online information of the tread of plant error i.e., monotonicity of the plant output with respect to the input and updates the rule consequents. Since this block can only ensures relative adaptation therefore it's work is termed as coarse tuning of the rule consequents. Another factor i.e. β or no of epochs, which outlines the controller learning, is very crucial in affecting the online coarse tuning of these rules. For instance, selecting a very small value of β might account for inadequate learning of the controller and too large values may make the convergence very slow.

After coarse tuning of the rules, Pomares implements another block named GL-block (Global Learning Block) for fine tuning of the results. Global learning is done by compiling real time input/output data and the system then learns from this data using controller output error method [11]. The global learning block relies on the adaptation block for initial control action. Thus, it is capable of controlling highly non-linear systems, in a pseudo-optimum way, even when these are time variable.

To the best of author's knowledge, no papers have been published for the identification of the best coarse tuning parameters of the adaptation block of pomares on a non-linear plant. If such a work is done, then the second block i.e., GL block, might become redundant.

This work aims to identify the optimal parameters of Adaptive block (i.e, monotonicity and no of epochs) for best course tuning. With the addition of this, the overall algorithm will eliminate the use of GL-block, making it much more effective.

1.6 Thesis Organization

The thesis is organized as mentioned below:

Chapter 2 details the literature review of the progress in the domain of Online Adaptive Fuzzy Model.

Chapter 3 describes the complete control architecture for online controlling of plants with specific focus on Adaptive learning.

Chapter 4 delineates the overall performance in detail by implementing the control architecture on different plants and different set-points. The simulation results of controller will be presented and discussed.

Chapter 5 finally concludes the thesis work by outlining the working of the proposed scheme, the novelty introduced in the basic idea, its benefits and the recommendations for future.

<u>**CHAPTER 2**</u>

2. Literature Review

2.1 Prologue

The chapter presents a literature review of the studies carried out for thesis work. For the sake of compactness, all of the corresponding material is not brought into the scope of the chapter however some key references will be discussed briefly. A detail list of the reference material is provided in reference section for further study.

2.2 Review of the Research Work

Most recently, control scientists are exploring and extensively working on Online Adaptive schemes using fuzzy models and a significant number of papers have been published. Their approach is to make use of different benefits which are extracted from different variations in the online adaptive scheme.

This chapter outlines the progression of Adaptive fuzzy mechanisms during the course of time. This upstream of technology primarily revolves around the amount of information of available regarding the plant. As it is known, fuzzy models are suitable where the goals and constraints or the physical mechanisms are ambiguous therefore the way a fuzzy controller is designed will vary. It all depends on the dynamics of the system and the knowledge which is available about it. [12].

Broadly speaking, three situations are encountered i.e., plants which either have known internal dynamics or plants with reduced knowledge of internal dynamics or plants with no information at all about it. Adaptive Fuzzy control designers have worked on all these fronts as discussed ahead.

2.3 Plants with known internal dynamics

If the differential equations (i.e., internal dynamics) of the plant and their properties are completely known, then designing stable controllers with this information is an easy task i.e., conventional control techniques are best suited [13], [14].

Its main advantage is that it can result in an optimal controller with satisfactory criteria i.e., a controller with minimum number of rules or a controller with an adequate topology etc. Its design also ensures that their stability can be studied and proved.

However, their disadvantages include a design,

1. That is very time-consuming,
2. That will need to be redefined for various plants
3. That cannot cope with changes in the dynamics of the plant if the designer missed out the possibility of various occurrences.

2.4 Plants with reduced knowledge of internal dynamics

Plants with reduced knowledge of internal dynamics are the most common of all. Reduced knowledge may include the existence of upper bounds or some properties of differential equations. The literature contains a large no of techniques for the design of controllers for such plants.

For instance, existence of certain bounds can help to make assumptions about the plants equations. Liu and Zheng [15] and Wang et al. [16] obtained Lyapunov-based update laws for the parameters of the controller by assuming that the approximation error is bounded.

In all the above cases, only the parameters are updated online while keeping structure of the fuzzy system fixed. Their main advantage is the surety about the stability of the closed-loop system but, the membership functions (MFs), and the rules have to be defined beforehand. This of course means that some knowledge about the plant is fundamental in order to make available a fine definition of the rule base. Without this knowledge the consequential structure would be too complex or too simple and will not be able to handle plant nonlinearities.

A neurofuzzy controller proposed by Gao and Er [17] has the capability to update both the parameters and the structure online. It adds new rules on the basis of completeness and system error and error-reduction-ratio (ERR) concept is used to delete the rules. The advantage of this method is that it places fewer rules in the areas with lower nonlinearities which in turn indicate that uneven rule distribution is formed. Another disadvantage is its high computational cost because of large matrix computations and storage of all previous input/output data.

Some techniques like direct adaptive self-structuring fuzzy controllers [18] add a new MF and new rules to the controller at each instant whenever any input exceeds the range limit of existing MFs. In this way, it places MFs in all those regions which are actually reached by the controller in the input space. Their disadvantage is, the even distribution of the rules in its explored space. This leads to two different cases; if the region contains low non-linearities then a high concentration of MFs are introduced while the regions with many nonlinearities do not have adequate MFs. This in turn will affect the non-linear control performance.

P. A. Phan and T. J. Gale [19] uses the same criteria of Gao and Fr for addition of new MFs. They modified it by proposing a condition that only those MFs will be added which achieved the rule with the maximum firing strength and highest activation degree. The MFs in the input space which are not sufficient to address all the nonlinearities are identified by a large system error. The advantage of this method is reduction of memory requirements as it does not store any past input/output (I/O) data, although it also implies that the decisions made by the algorithm are short-sighted i.e., the algorithm only views happenings at present. Only Gale and Phan were able to prove stability in structure changes. However, all these techniques require certain assumptions about the plant to be made.

2.5 Plants with no knowledge of internal dynamics

This is the most challenging case because of the inability to make assumptions about the equations or the plant because of inadequate relevant information.

The work of this thesis is primarily based on these types of plants therefore this subject will be studied in a little more depth than the previous ones. Control designers have mostly used offline pre-training techniques to control such plants. However, current researchers are trying to work out techniques where controllers automatically tune themselves according to the requirements without any offline work. It will be discussed briefly below.

2.5.1. Offline Techniques

Most offline techniques are based on evolutionary algorithms and automatic design of controllers is utilized. In many cases, the I/O data is made available, and various methods based on pre-training are used.

Offline Parameter learning in the controller proposed by [19] employs evolutionary algorithms with a predefined topology. In [20] and [21], the controller's topology is also learnt, which provide more flexible solutions, as they are independent of the ability of the designer to select the rules.

Several techniques are also presented for controller parameter's offline auto-tuning in batch mode [22]. They generally employ genetic algorithms [23], and neural networks [24]. For instance, plants of certain class of differential equations with known bounds are dealt satisfactory by Wang [25]. This also requires offline pre-training before working in real time. Another approach is MRAC [26] (model reference adaptive control), which also arose from research into how to improve self-organizing controllers by using ideas from conventional control. It employs a reference model, to provide a closed-loop performance feedback for synthesizing and tuning the fuzzy controller.

Their disadvantage is that it is designed as a fixed controller because it is learned offline, and therefore it is incapable of following changes in the plant. Also, there are many situations in which the I/O data used for the training is very difficult and mostly impossible, to extract.

Practical speaking, the unsatisfactory performance is evident from online applications when system behavior changes. The reason is the pre-training parameters are insufficient to cope with online changes. Therefore online techniques were developed.

2.5.2. Online Techniques

Online techniques are usually applied when no I/O data is available. There are several methods for the online adaptation of controller parameters. There are certain methods which tune the output scale factors; others tune the rules and some methods which modify the structure of the controller i.e., the MFs as well. For example in [27], all three parameters are tuned.

2.5.2.1. Online tuning of output scaling factor

In recent times, adapting input or output scale factors of the fuzzy controller has also been utilized. There are many cases in which this is sufficient for controlling a simple plant.

As we tune the gain of a PID controller, similar is the case of tuning the scale factor of a fuzzy system. To improve the control policy in the first iterations where no information is present in the fuzzy controller, online tuning of the output scale factor is used in [28]. However, this addition improves very few control problems.

2.5.2.2. Online tuning of rules

Probably, the initial papers on fuzzy rule-based controller which evolves online surfaced in 2001[29]. In this paper, online data collection is done and rules are updated on the basis of this data.

For simple applications, online tuning of rules is adequate to solve the control problem. In many SOC approaches, algorithms use the monotonicity sign and tune the rules. This however is not very effective in highly nonlinear plants as it is able to only provide a coarse tuning of controller parameters [30][31].

There are certain online algorithms [32] which work primarily on the qualitative knowledge and real time error at the plant output. This style of adaptation of fuzzy controller functions well but it is limited to modifying rule consequents only and is incapable of tuning membership functions.

In [33], tuning of parameters of the controller is done by reinforcement learning. This is done by two fuzzy controllers working as an actor and a critic. The input state used to determine the next action is used by the actor while it is the critic which adds up the reinforcement signal with the information about the action. This produces an internal reinforcement which updates both the fuzzy systems.

2.5.2.3. Online tuning of the controller structure

Cara et. al [34] and [35] presented an online SOC which did not require any offline training. This controller initially has empty rules and a very simple structure. It updates the controller parameters on the basis of data obtained during the plant’s operation. This approach was tested on a nonlinear servo system and gave excellent results [36]. The disadvantage of this approach is the exponential rise of the no of rules with the addition of every new membership function.

Another approach to tune the structure of the controller includes the usage of the error at the controller output and applies gradient–descent technique to find the lowest value of error. This error then updates the structure of the controller [11].

In [10], two auxiliary systems are used. The adaptive block uses plant output error for a coarse tuning of the rules, while Global Learning block utilizes the controller output error for fine tuning of the rule consequents and the structure. We will be applying and studying this approach in our work and trying to modify it.

2.6 Chapter Summary

This literature review summarizes Fuzzy control's evolution from controlling a plant with complete details of internal dynamics towards a plant whose internal dynamics are completely unknown. In this thesis an attempt was made to study the controller which is able to control plants which have unknown internal dynamics.

CHAPTER 3

3. Control Scheme

3.1 Prologue

This chapter discusses the complete control scheme, its architecture and the control flow which enables the fuzzy controller to tune its parameters for desired control action.

3.2 Overview of control scheme

Generally, the system or plant to be controlled is expressed by its difference equations. It can be mathematically represented as:

$$y(k+d) = f(y(k)),\ldots, y(k-p), u(k),\ldots, u(k-q)) \qquad (3.1)$$

Where,

d= Delay of the plant
f= Unknown continuous/ derivable function.
y= Output of the plant
u= Control signal

There is a mandatory condition that the plant to be controlled should have a control policy that can transform the output to preferred value. It means that the output value should always be dependent on the control signal. This can be mathematically explained by considering a function (F) which is continuous with respect to all state variables. Therefore the control signal is given by:

$$u(k) = F(\vec{x}(k)) \qquad (3.2)$$

The states of the controller are given by;

$$\vec{x}(k) = (r(k), y(k), ..., y(k-p), u(k-1), ..., u(k-q)) \tag{3.3}$$

Here, r(k) is the desired output at instant k, This means that we have substituted $y(k+d) = r(k)$ This is able to reach the set point target after d instants of time.

In this algorithm, no information is needed about the mathematical equations of the plant, although it is necessary to know the monotonicity of its output with respect to the control signal and the inputs that have a significant influence on the plant output.

Now let us consider a complete rule-based fuzzy controller [13]. Generally the rules are defined as:

If x_1 is $X_1^{i_1}$ AND x_2 is $X_2^{i_2}$ AND ... AND x_N is $X_N^{i_N}$

$u = R_{i_1 i_2 ... i_N}$

Where,

X_N^j = jth MF of variable x

N = no of input variables

$R_{i_1 i_2 ... i_N}$ = rule consequent

We are using product operator (T-norm) as fuzzy inference method. The defuzzification strategy includes the centroid method with sum–product operator. Triangular membership functions are used. The strength of rule can be calculated using

$$\mu_{i_1 i_2 ... i_N}(\vec{x}) = \mu_{i_1 i_2 ... i_N}(x_1 x_1 .....x_N) = \prod_{m=1}^{N} \mu_{X_m^{i_m}}(x_m) \tag{3.4}$$

Hence, the following equation calculates the output of our fuzzy controller;

$$u(k) = F(\vec{x}(k))$$

$$= \frac{\sum_{i=1}^{n_{rules}} R_i\, \mu_i(\vec{x}(k))}{\sum_{i=1}^{n_{rules}} \mu_i(\vec{x}(k))}$$

$$= \frac{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_N=1}^{n_N} (R_{i_1 i_2 \dots i_N} . \prod_{m=1}^{N} \mu_{X_m^{i_m}}(x_m))}{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_N=1}^{n_N} (\prod_{m=1}^{N} \mu_{X_m^{i_m}}(x_m))} \tag{3.5}$$

Now, with this basic infrastructure of controller in place with no rules and a generalized triangular membership function structure of the controller, we move on towards the two stages of this scheme.

These stages are illustrated by the flowchart in Figure 3.1.

Initially, Fuzzy rule consequents adaptation is done on the basis of output error of the plant. For this, the plant monotonicity is taken into account.

Once this coarse tuning of the rules is complete, then stage two takes charge and fine tunes the system on the basis of controller output error. It does not only update the rule consequents but also modifies rule antecedents i.e., the M.Fs. If we are able to perform a perfect course tuning in the first stage, the second stage would become redundant and unnecessary. This will save computational cost and time.

Further details of control scheme are explained in depth in the architecture of this controller.

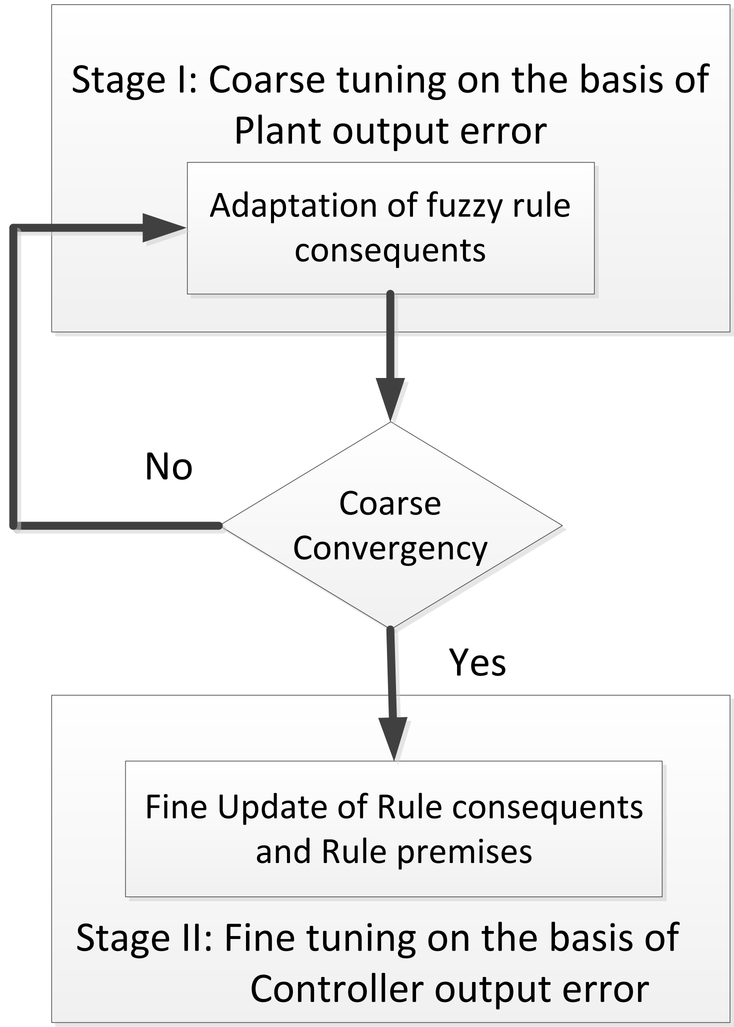

Figure 3.1: Stages of the applied control scheme

3.4 Architecture of the Fuzzy controller

Control architecture of the applied control scheme is depicted in Figure 3.2. The Main Fuzzy controller is represented by the larger block. Self organizing controllers can only be tuned by the knowledge base present inside the controller. If we further analyze the knowledge base, we can see that it contains two major set of factors: the structure of controller identified by the definition of the MFs, and the rule consequents. This main controller now interacts with the two auxiliary blocks: The Adaptation block (A – Block) and the Global Learning block (GL – Block) to find suitable parameters from the evaluation of our control architecture.

By using the monotonicity of the output of the plant as regards the control signal, the coarse tuning of the rule consequents is done by the A-Block. For the fine tuning of the rule consequents and the MFs, GL-Block is employed. The GL-Block does this fine tuning by collecting the I/O data from the progression of the plant. A-Block contributes when starting parameter values are not present i.e., in the preliminary iterations of the control process. And then the GL-Block takes over and fine tunes the control results.

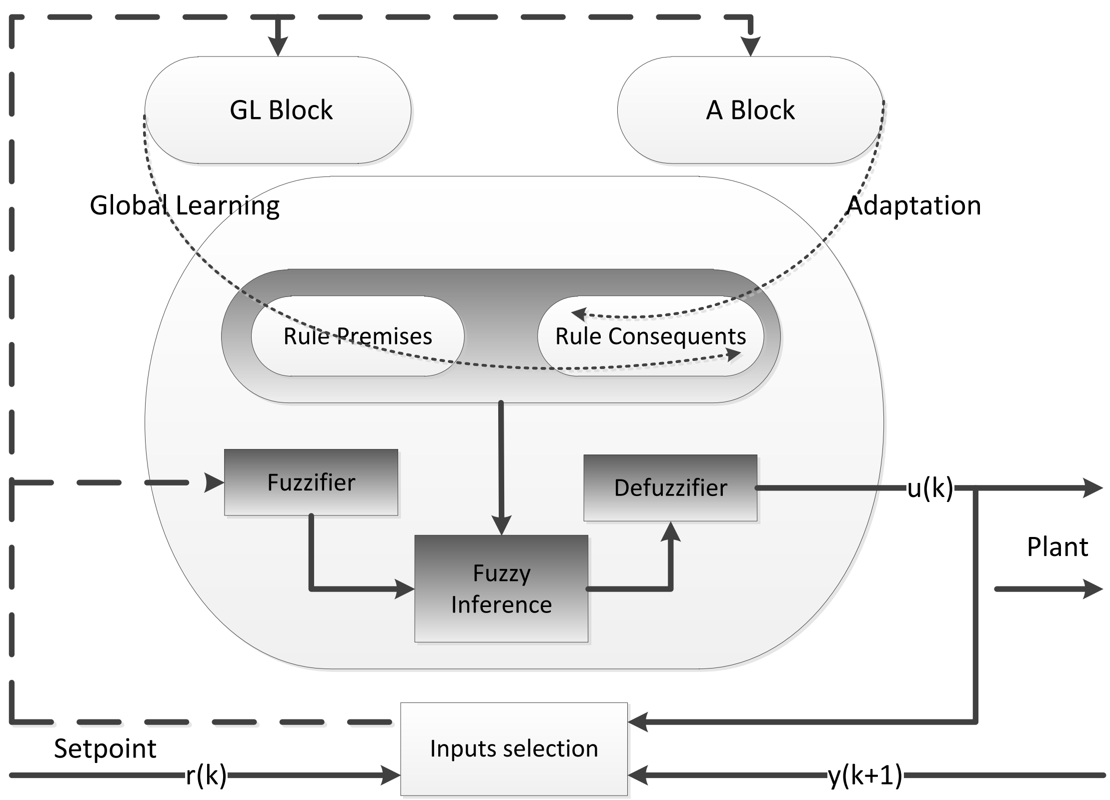

Figure 3.2: Controller Architecture

3.3.1 Adaptive Block

This block works on the information on "monotonicity" of the plant. With known C_0, we can move in the right direction for updating the rule consequents. If the output of the plant increases directly with the control signal then we will need to adjust the rule consequents for the desired control signal. Alternatively, if the C_0 was negative, then the rule consequents for which a control signal is less will need to be adjusted.

It must be remembered that we modify only those rules which affect the fuzzy controller output in the desired manner. Along with this, the modifications are performed based on the activation degree of the rules.

This adaptive block is working on the basis of Specialized Inverse Learning. The parameters of fuzzy controller are adapted by minimizing the deviation between the output y and the reference r. Hence the adaptation is a goal-oriented technique [13] and the process is automatically excited with the correct signal if a typical reference trajectory must be followed. This is depicted in the Figure 3.3

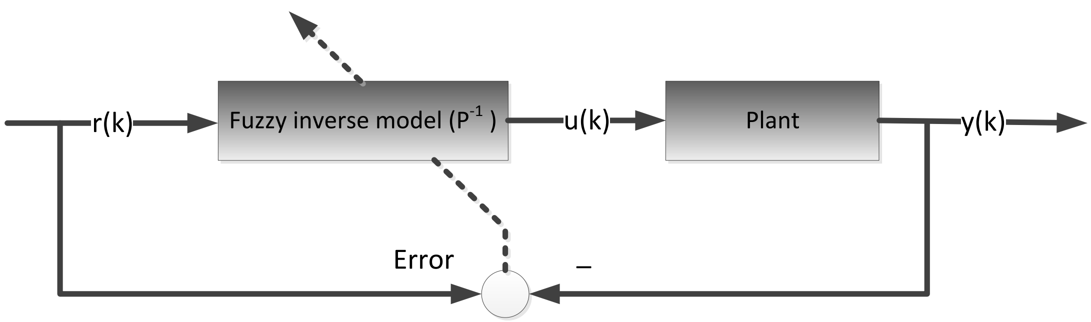

Figure 3.3: Plant output error based inverse learning scheme

Since this is being done on the basis of only qualitative information therefore, this block performs coarse tuning of fuzzy rules. As mentioned in the Figure 3.2, the GL block also updates the rules and structure. Therefore, both the blocks work in harmony to find the optimal rules for the best control action. First, the A block evaluates its results and then it gives way to GL block to fine tune the system.

Hence, this A Block becomes adapts to the fuzzy rules of the main controller in accordance with the expression below:

$$\Delta R_{i_1 i_2 \ldots i_N} = C.\mu_{i_1 i_2 \ldots i_N}(k-d).e_y(k)$$

$$= C.\mu_{i_1 i_2 \ldots i_N}(k-d).(r(k-d)-y(k)) \qquad (3.6)$$

Where,

$$C = C(k) = C_0.\exp(-k/\beta) \qquad (3.7)$$

C_0 represents monotonicity of the plant and β represents the no of epoch and also termed as the forgetting factor. This β ensures that the influence of the A block reduces with time and give way to the GL block.

For minimizing the control error and maximizing the adaptation by the controller, the two most critical factors are C_0 and β in the above equation. We intend to work on them to make sure that setpoint is achieved using this block only.

3.3.2 Global Learning Block

This block uses the controller output error method proposed by [37] to fine tune the controller results. It applies the gradient decent methodology on the controller output error without needing a reference model or a plant model.

If at any instant k, a control signal u(k) is introduces to the controller and after ‘d’ periods of samples the output y(k+d) is observed then we do not attain only the error at the plant output as available information. Infact now we know that if our intended response was r(k)=y(k+d) then the optimum control signal would be accurately u(k). Hence, we get a more precise value for the plant’s true inverse function after each sampling period.

With this inverse plant information, and the error of u(k) as mentioned in Figure 3.4, we update the parameters of the original controller.

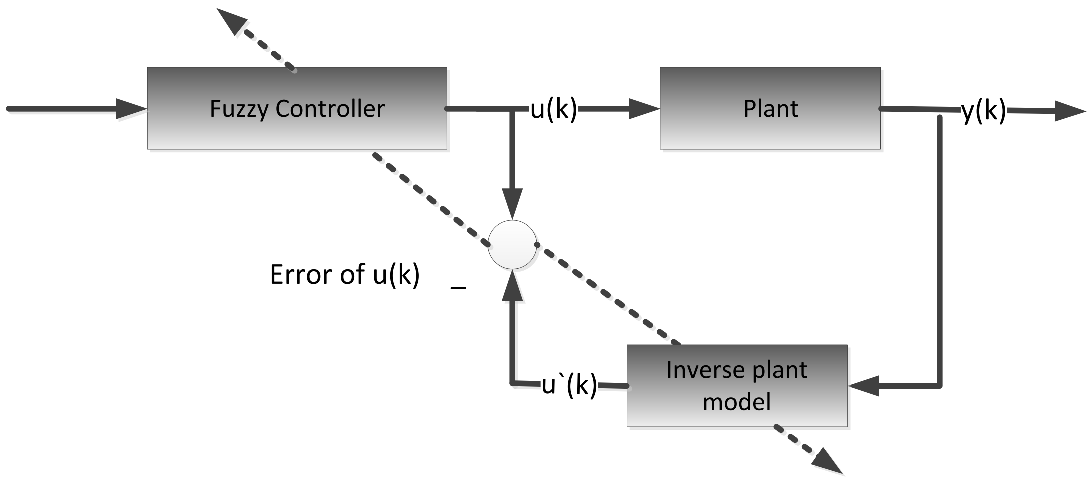

Figure 3.4: Controller output error based inverse learning scheme

Mathematically, at any instant k the control signal that the plant exerts can be represented as

$$u(k) = \hat{F}(\vec{x}(k), \Theta(k)) \qquad (3.8)$$

Where, x(k) is represented by the expression below and that represents the set of parameters that define the controller at instant k (rules plus MFs)

$$\vec{x}(k) = (r(k), y(k), ..., y(k-p), u(k-1), ..., u(k-q) \qquad (3.9)$$

The plant output y(k+d) is obtained after d iterations. Now if y(k+d) is replaced by r(k) , we get the detail that belongs to the actual inverse plant function. Now, the input vector x(k) is represented by

$$\hat{x}(k) = (y(k+d), y(k), ..., y(k-p), u(k-1), ..., u(k-q) \qquad (3.10)$$

To perform the global learning process we must evaluate the output U^ given by the controller for each possible input, thus obtaining an error signal in the output of the controller.

$$\hat{u}(k) = F(\hat{x}(k)) \qquad (3.11)$$

It is important to note that, although u^m is produced by the controller; it is not applied to the plant. Its only purpose is to calculate e(m)

$$e_u(m) = u(m) - \hat{u}(m) \qquad (3.12)$$

Where u(m) and u^(m) are the desired output and that obtained by the current parameters of the main fuzzy controller, respectively.

Hence, the optimization of the parameters of the main fuzzy controller in each iteration is done in the following manner:

$$\Delta\Theta(k) = \eta(k).e_u(k-d).\partial\hat{F}(\hat{x}(k-d);\Theta(k)) \qquad (3.13)$$

Here, n(k) is the learning factor, taken to be < 1

While the rest of the derivatives are calculated using the following expressions:

Derivatives which update the rule consequents;

$$\frac{\partial\hat{F}(\hat{x};\Theta)}{\partial R_{j_1 j_2 \ldots j_N}} = \prod_{m=1}^{N} \mu_{X_m^{j_m}}(\hat{x}_m) \qquad (3.14)$$

While those which update the rule antecedents;

$$\frac{\partial \hat{F}(\hat{x};\Theta)}{\partial c_v^j} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_N=1}^{n_N} \left(R_{i_1 i_2 \ldots i_N} \cdot \frac{\partial \prod_{m=1}^{N} \mu_{X_m^{i_m}}(\hat{x}_m)}{\partial c_v^j} \right) \tag{3.15}$$

Where,

$$\frac{\partial \prod_{m=1}^{N} \mu_{X_m^{i_m}}(\hat{x}_m)}{\partial c_v^j} = \frac{\partial \mu_{X_v^{i_v}}(\hat{x}_v)}{\partial c_v^j} \cdot \prod_{m=1}^{N} \mu_{X_m^{i_m}}(\hat{x}_m) \tag{3.16}$$

And further,

$$\frac{\partial \mu_{X_v^i}(x)}{\partial c_v^j} = \begin{cases} \dfrac{-(c_v^j - x)}{(c_v^j - c_v^{j-1})^2} U(x; c_v^{j-1}, c_v^j) & j = i-1 \\ \dfrac{-(c_v^j - x)}{(c_v^j - c_v^{j-1})^2} U(x; c_v^{j-1}, c_v^j) + \dfrac{(x - c_v^j)}{(c_v^{j+1} - c_v^j)^2} U(x; c_v^j, c_v^{j+1}) & j = i \\ \dfrac{(x - c_v^j)}{(c_v^{j+1} - c_v^j)^2} U(x; c_v^j, c_v^{j+1}) & j = i+1 \end{cases}$$

With the above controller designed, the controller is fine tuned because it acts as a perfect inverse model of the plant.

Nevertheless, without the existence of the adaptation block the global learning block is not workable because it is the adaptation block which initially finds useful data for the controller.

3.4 Measuring Control Performance

Since our work is based on the improvement in the adaptation block by varying its fundamental parameters i.e., C_0 and β, we need to define a standard for measuring the control performance and relating our results to it.

Hence, for measuring the effect of the two factors (C_0 and β) on the overall control performance, we use the standardised RMSE between the plant output and the set point.

$$RMSE = \sqrt{\frac{\sum_{k=1}^{Num\_epochs}(r(k) - y(k+d))^2}{Num\_epochs}} \tag{3.17}$$

3.5 Chapter Summary

We have outlined the detailed control scheme which includes its controller, additional blocks and the way we shall measure the control performance.

CHAPTER 4

4. Simulation Results

4.1 Prologue

In this section it is intended to establish the fact that adaptive learning as outlined by the control scheme can be drastically improved by the way we select the two factors (C_0 and β) which form the core of the adaptive mechanism. Performing simulations by considering various values of β and C_0, the importance of our investigation will be evident.

This analysis is not proved by considering only one non-linear plant, but different simulations have been done considering a linear and a nonlinear plant and then for two different reference signals. And the desired conclusion holds true for all these cases. For additional understanding of the strength of this controller, a linear plant is chosen and three non-linearities to further cross check the results. With this, it is believed that implementing GL block becomes unnecessary.

4.2 Performance of Adaptive block for different B and Co

To implement the controller technique on a non-linear system, we select a system whose difference equation is;

$$y(k+1) = -0.3\sin(y(k)) + u(k) + u^3(k) \qquad (4.1)$$

The above model was also implemented by Pomares[10] as a demonstration of his work. I have performed all simulations on Matlab and Simulink. Figures 4.1 and 4.2 represent the Simulink Model of the adaptive controller as implemented and the plant model used.

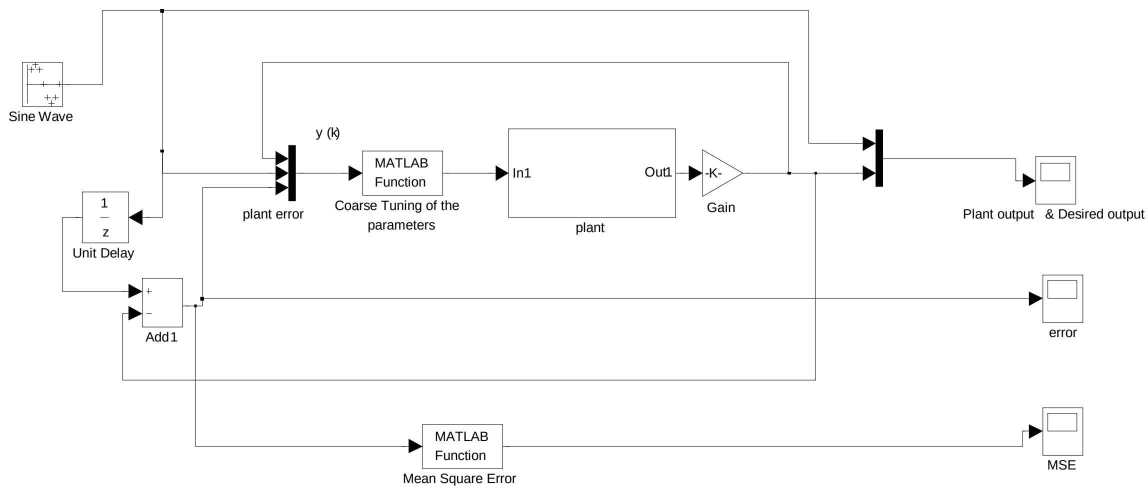

Figure 4.1: Simulink Model - Adaptive controller with Sinusoidal as Set-point

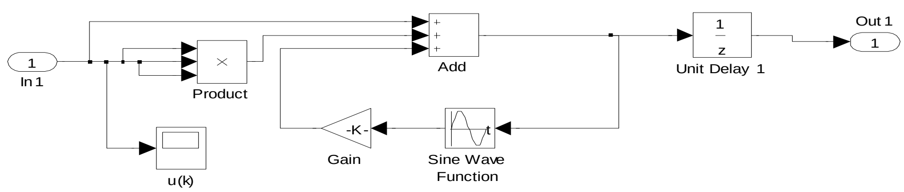

Figure 4.2: Simulink Model - Non-linear plant

A purely sinusoidal signal with amplitude 0.75 and 600 samples per period is used as the desired response from the above non-linear system. Initially, rule consequents of the controller are zero, membership functions are triangular and universe of discourse is from -4 to 4.

Table 4.1 was formed for 70k iterations considering the different values of C_0 and β. If one examines Table 4.1, we can clearly deduce that for every controller there is an operating range with specific values of β and C_0. By operating range, we mean a set of values for which the controller works and fulfills the control action by following the set point.

Here, β should not be too small to give insufficient room to the adaptation block to take up suitable values very large values of β slows down the learning process. C_0 is the scale factor which is necessary to avoid out of range modifications in the fuzzy rules. It also controls the convergence speed. When a controller is learning, how fast it learns depends on the correct rules it adapts. Small values of C_0 indicate very less or no convergence but large values of C_0 means faster convergence. Even higher values of C_0 can provoke undesired oscillations in the rule consequents which can cause instabilities.

In order to explain our results, values of β are selected from lowest of 100 to a maximum of 100,000,000 with steps of x10. Values of C_0 are selected from lowest of 0.001 to 5 with variable steps in which the outputs significantly change. Table 4.1 calculates the RMSE values for different combinations of the β and C_0.

C_0	0.001	0.01	0.1	0.25	0.5	0.75	1	2.5	5
β = 100	0.5289	0.5732	0.5205	0.5079	0.5902	0.6054	0.5979	inf	inf
β =1000	0.4928	*0.3195*	*0.1847*	*0.1714*	*0.1662*	*0.164*	*0.1628*	inf	Inf
β =10,000	*0.3031*	0.0673	0.0285	0.0226	0.0197	0.0185	0.0176	inf	Inf
β =100,000	0.1779	0.0615	0.02	**0.0124**	**0.0086**	**0.0069**	0.0057	inf	Inf
β =1,000,000	0.172	0.0606	0.0192	0.0117	0.0077	0.0059	0.0048	inf	Inf
β =10,000,000	0.1715	0.0605	0.0192	0.0116	0.0077	0.0058	0.0047	inf	Inf
β =100,000,000	0.1715	0.0605	0.0192	0.0116	0.0076	0.0058	0.0047	inf	Inf

Worst results | Unstable | *Poor performance* | Good operating range | **Best results** | negligible effect of Beta

Table 4.1: RMSE of non-linear plant output for different values of β and C_0.

By analyzing Table 4.1 the following points can be deduced;

1. The worst results are obtained for β <= 100 and for all values of C_0. The controller is unable to learn because the forgetting factor i.e., β is too small to let the adaptive block take up suitable values of rule consequents.

2. The Unstable results are obtained for $C_0 > 1$ and for any value of β. The reason is that for the non-linear plant considered in equation 4.1 understands this value to be excessively large and thus cause undesired oscillations.
3. *Poor performance* is for the range of β >100 & β <1000. Here, once again there is inadequate learning because of low values of β.
4. Good operating range contains value low RMSE values. This range resides between β >10,000 and β <100,000. The **Best results** are also within this range with RMSE values the most optimum with fast convergence and adequate learning.
5. Negligible effect of β is marked on those results which do not reduce the RMSE values but slow down the convergence speed because of large values of β.

This analysis is further clarified by drawing graphs highlighting the trends of RMSE for β and C_0. This is verified by the actual simulations on the plant and their outputs are also presented for comparison.

For example, in Figure 4.3, it is evident that for small β, the learning is not adequate because the adaptive controller does not have enough time to fully learn itself. Another important finding is that as β increase more than 10,000, the control performance stabilizes and does not change much. For instance in figure 4.4, the controller fails to tune itself initially for small values of β but for β greater than 10,000 the control performance drastically improves as shown in figure 4.5.

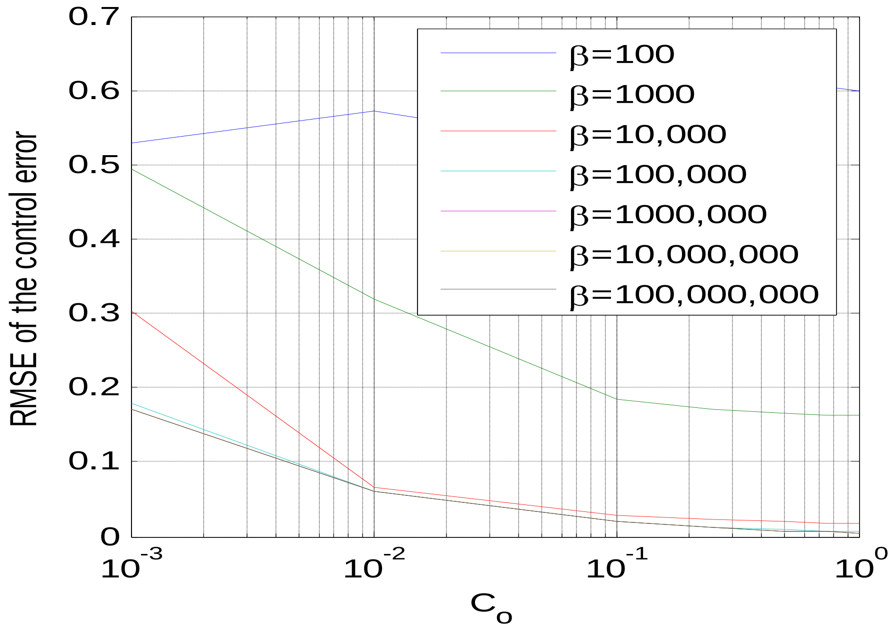

Figure 4.3: Impact of changing β on Control performance of non-linear plant

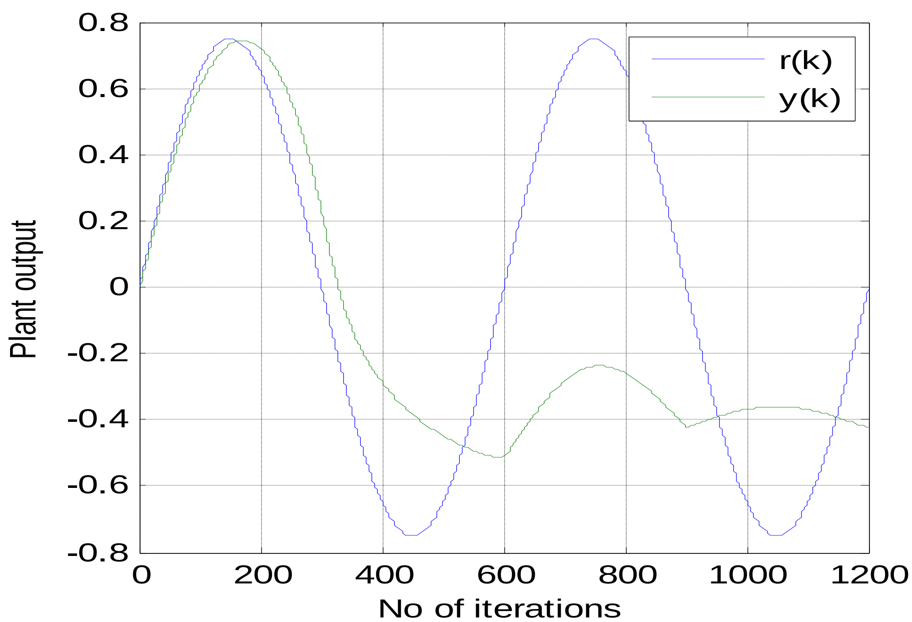

Figure 4.4: Controller evaluation of non-linear plant

for $C_0 = 0.5$ and β = 100

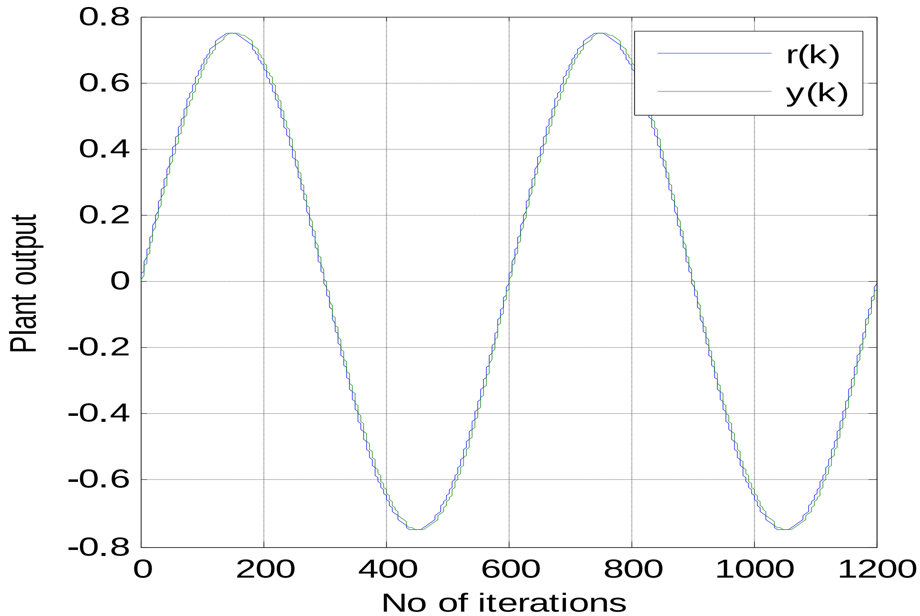

Figure 4.5: Controller evaluation of non-linear plant

for C_0 = 0.5 and β = 1,000,000

Now, when we chalk out the trend of RMSE values in Figure 4.6, Its clear that the impact of C_0 is constant from C_0 =0.1 to 1. The reason is because Co for this specific case limits the desired modifications of rule consequents in this range. Therefore naturally different systems will have different ranges of Co .Increasing Co within this range also improves the RMSE values because it also effects the convergence speed of the controller. Also, for values less than 0.1, the controller is unable to adapt rules as it is way below the range for this plant. For values greater than 1, the controller becomes unstable because the very high values (for this specific plant is > 1) cause undesired oscillations and makes the controller unstable. So the optimal range in this case would be any points between them.

Results of figure 4.7 figure 4.8 serve as evidence to the discussion in the preceding paragraph.

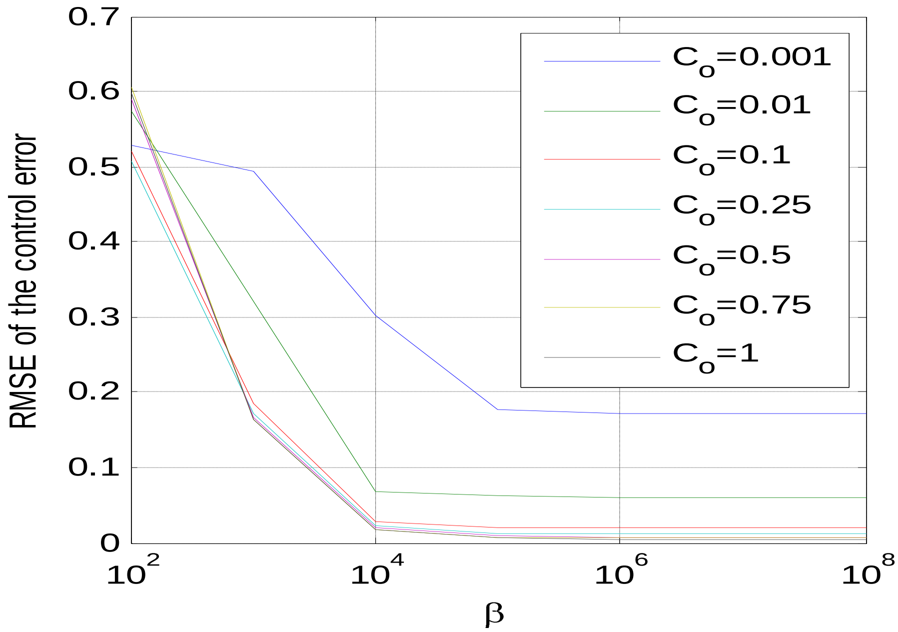

Figure 4.6: Impact of changing C_0 on control performance of non-linear plant

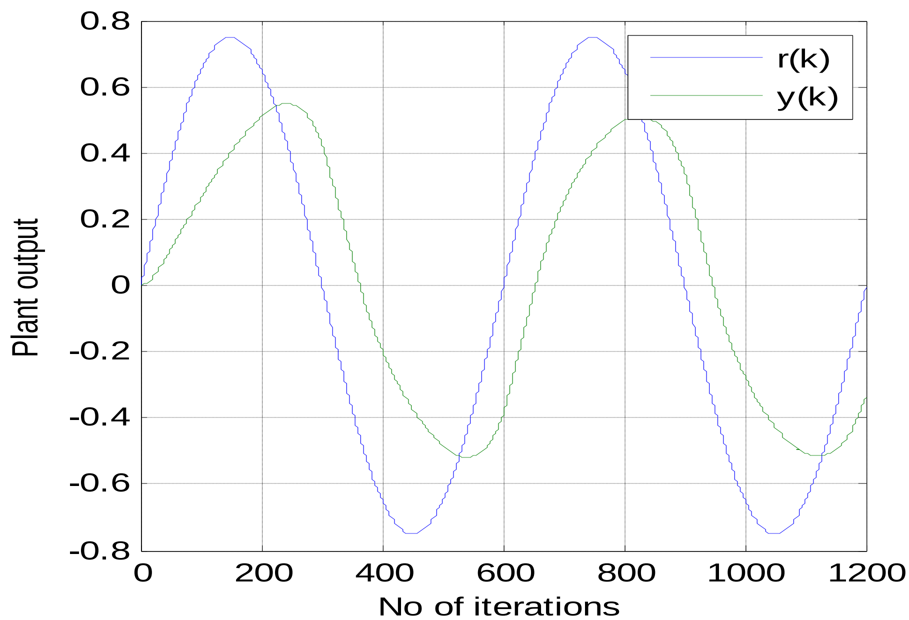

Figure 4.7: Controller evaluation of non-linear plant

for $C_0 = 0.01$ and β = 100,000

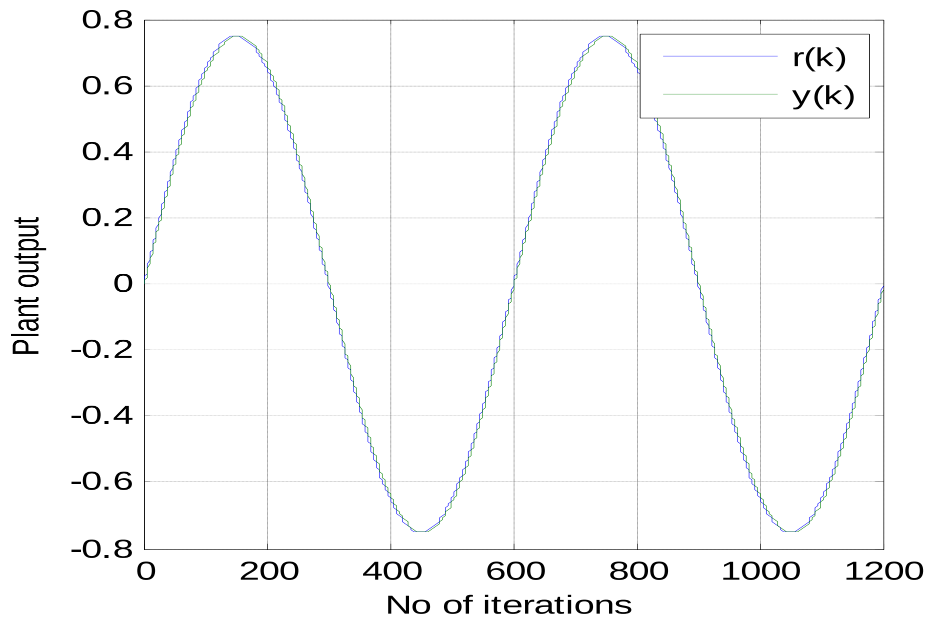

Figure 4.8: Controller evaluation of non-linear plant for C_0 = 0.75 and β = 100,000

Since the results obtained are almost perfect by just using the adaptive block of the controller therefore we do not require a Global learning block to enhance the control performance. This is as per the initial discussion of the scope of this work.

It is important to mention here that this analysis is for the non-linear plant of equation 4.1 only. The optimal ranges ascertained in this analysis do not apply to any other plant. The reason is because all plants have different internal dynamics and parameters which certainly do not coincide with the plant in discussion.

However, what is clear is that all plants can be analyzed in the above step by step procedural manner. Initially by testing the controllers from lowest values of C_0 and β and increasing them until the controller becomes unstable. Then finding best results from within this data would be an easy task.

4.2.1 Comparison of performance for different systems

The proposed analysis technique is well established by the evidences given above. However, to further consolidate its findings a completely different plant is selected and analysed. This new plant is a linear plant of motor, control and gear represented by the following transfer function;

$$T(s) = \frac{0.2083}{s + 1.71} \qquad (4.2)$$

Table 4.2 gives a detailed comparison of the results. As speculated, the table shows that the optimal range for this plant is different than the previous one. The reason is of course for different plants, their aptitude for learning which are effected by β and the rule modification ranges controlled by C_0 will be different.

As can be seen from Table 4.2, the controller for first plant becomes unstable for $C_0 > 1$ while the controller for the second plant improves the results as we increase C_0 from 1 onwards. This shows that the rule consequents can still be adapted and the convergence for this plant has the potential to be further increased.

Furthermore, it is visible from the trends of RMSE in Figure 4.9 and Figure 4.10, this analysis gives us an optimum range near the values of C_0 =0.1 and β =10,000.

However, there is a similarity in the trend of RMSE values of the non-linear and linear plant and that is on two fronts.

1. Increasing the values of B more than 10,000, the effect on the RMSE values become negligible. This means that normally this value of B is sufficient to fully take up desired values of rule consequents.
2. The optimum values of Co generally lie in the range of 0.1 to 1 and ahead of that does not drastically affect the convergence of controllers.

This analysis shows that our online adaptive controller successfully performs coarse tuning for different plants even though the controller contains no knowledge of the new plant. Therefore, again, we do not require a Global learning block to enhance the control performance.

RMSE of output of non-linear

C_0 / β	0.001	0.01	0.1	0.25	0.5	0.75	1	2.5	5
100	0.5289	0.5732	0.5205	0.5079	0.5902	0.6054	0.5979	inf	inf
1000	0.4928	0.3195	0.1847	0.1714	0.1662	0.164	0.1628	inf	inf
10,000	0.3031	0.0673	0.0285	0.0226	0.0197	0.0185	0.0176	inf	inf
100,000	0.1779	0.0615	0.02	0.0124	0.0086	0.0069	0.0057	inf	inf
1,000,000	0.172	0.0606	0.0192	0.0117	0.0077	0.0059	0.0048	inf	inf
10,000,000	0.1715	0.0605	0.0192	0.0116	0.0077	0.0058	0.0047	inf	inf
100,000,000	0.1715	0.0605	0.0192	0.0116	0.0076	0.0058	0.0047	inf	inf

RMSE of output of linear plant

C_0 / β	0.001	0.01	0.1	0.25	0.5	0.75	1	2.5	5
100	0.5304	0.5968	0.717	0.5129	0.7984	0.8622	0.8487	0.6845	0.5375
1000	0.5225	0.4611	0.3188	0.3094	0.3014	0.2996	0.2985	0.2959	0.2861
10,000	0.4705	0.3415	0.1616	0.1219	0.0993	0.0886	0.0819	0.0647	0.0538
100,000	0.4195	0.2682	0.1278	0.0938	0.0741	0.0646	0.0585	0.0428	0.0337
1,000,000	0.415	0.2593	0.1239	0.0905	0.0711	0.0617	0.0558	0.0402	0.0311
10,000,000	0.4146	0.2584	0.1235	0.0902	0.0708	0.0614	0.0555	0.0399	0.0309
100,000,000	0.4145	0.2583	0.1235	0.0902	0.0708	0.0614	0.0555	0.0399	0.0308

Table 4.2: Comparison of performance of non-linear and linear plants

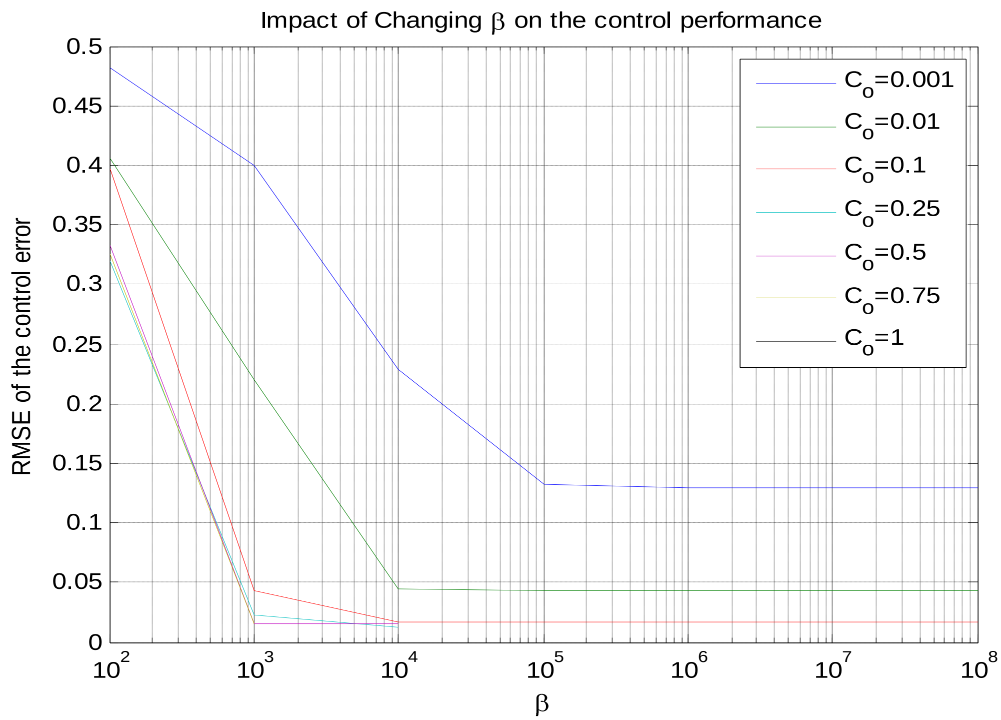

Figure 4.9: Impact of changing β on Control performance of linear Plant

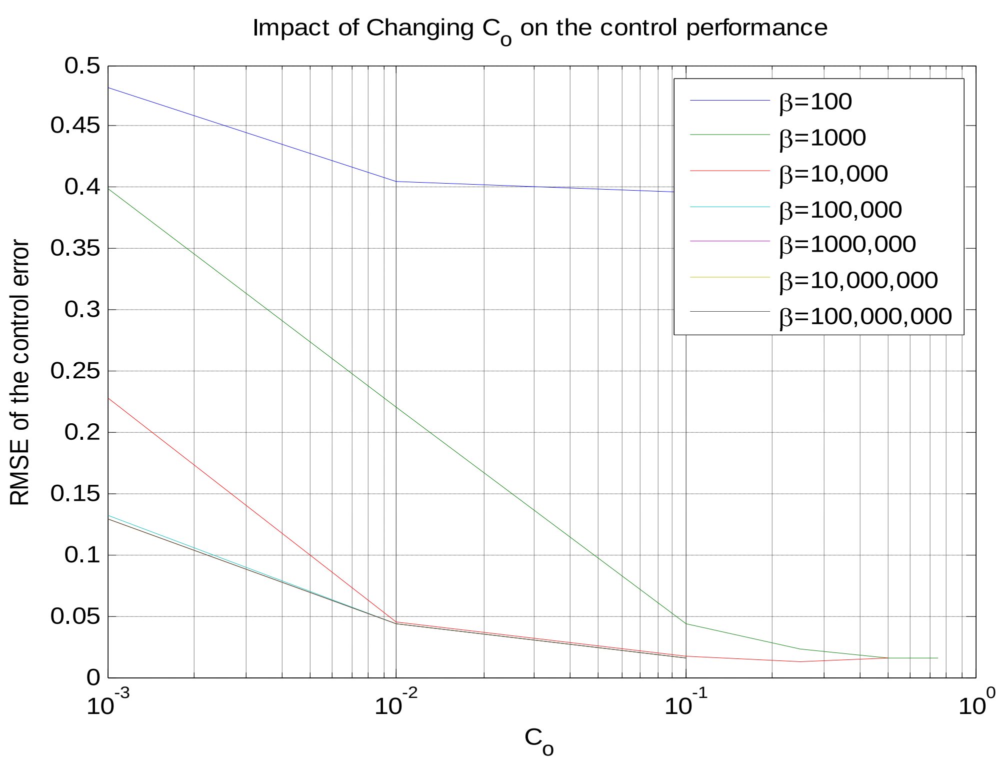

Figure 4.10: Impact of changing C_0 on control performance of linear Plant

4.2.2 Comparison of performance for different reference signals

The adaptive controller was now implemented for a step signal as reference signal. As commented above, we need to make sure that the initial control evaluation is almost perfect so that we do not need to move on to global learning. Figure 4.11 shows the Simulink Model for the implementation of step signal as reference for the non-linear plant. If we consider Figure 4.12, it is self explanatory that the online adaptive controller has not done adequate learning hence applying fine tuning would be useless.

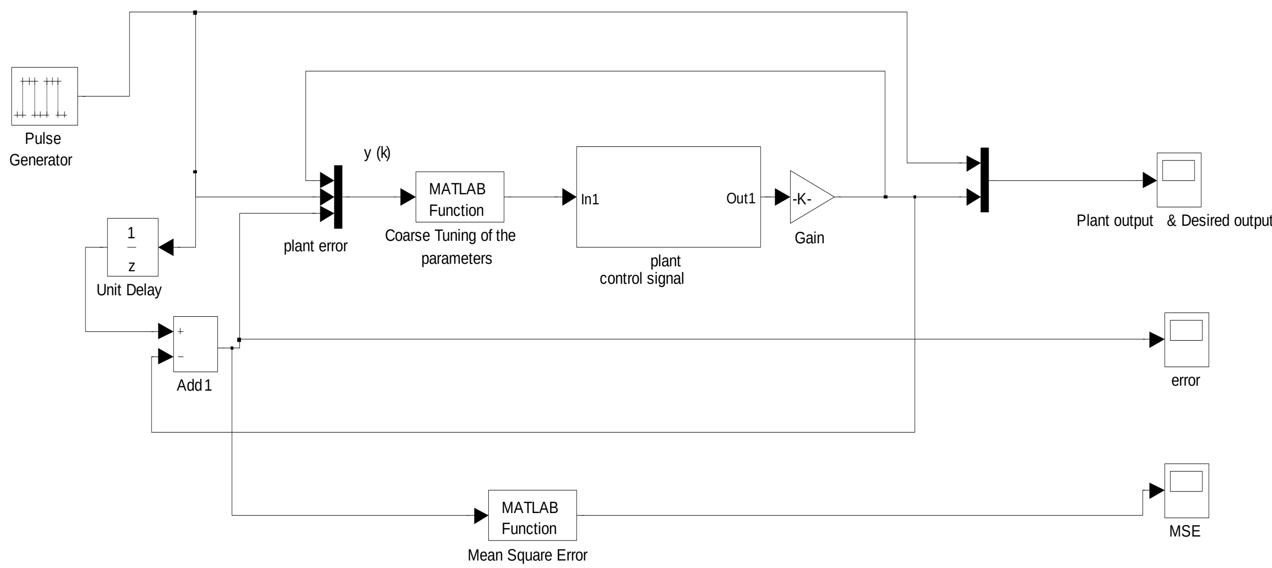

Figure 4.11: Simulink Model - Adaptive controller with Step as Set-point

However, after thorough analysis of the controller on the basis of different values of its parameters (as outlined by Table 4.3), we can easily conclude that for β =10,000 and C_0 = 0.25, excellent results are obtained depicted by Figure 4.13.

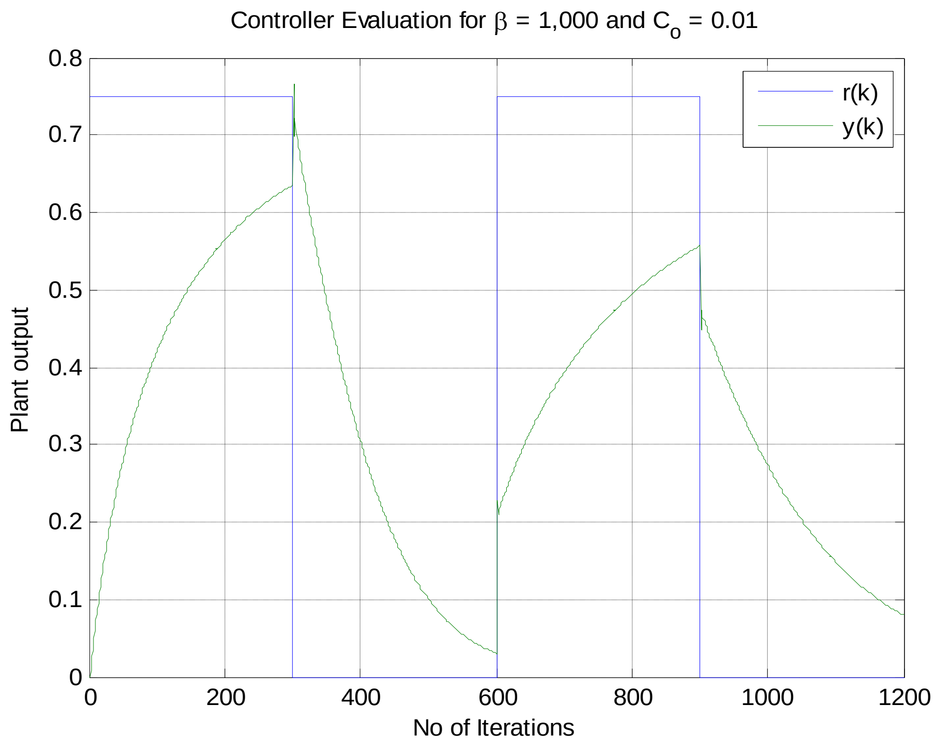

Figure 4.12: Controller evaluation of step signal for C_0 = 0.01 and β = 1,000

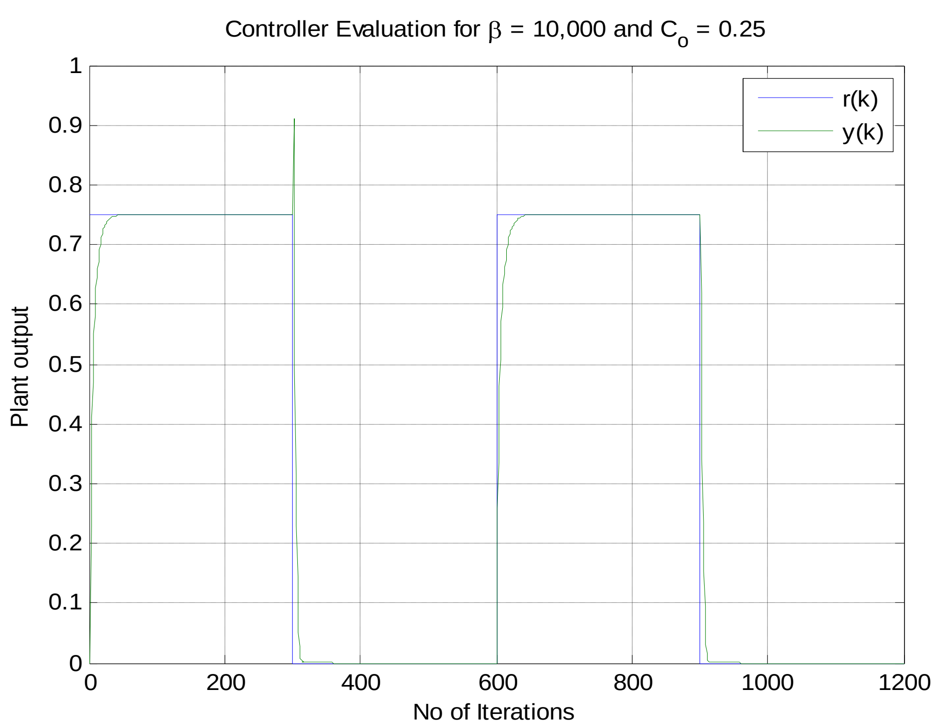

Figure 4.13: Controller evaluation of step signal for C_0 = 0.25 and β = 10,000

RMSE with sinusoidal as reference signal

C_0 / β	0.001	0.01	0.1	0.25	0.5	0.75	1	2.5	5
100	0.5289	0.5732	0.5205	0.5079	0.5902	0.6054	0.5979	inf	inf
1000	0.4928	0.3195	0.1847	0.1714	0.1662	0.164	0.1628	inf	inf
10,000	0.3031	0.0673	0.0285	0.0226	0.0197	0.0185	0.0176	inf	inf
100,000	0.1779	0.0615	0.02	0.0124	0.0086	0.0069	0.0057	inf	inf
1,000,000	0.172	0.0606	0.0192	0.0117	0.0077	0.0059	0.0048	inf	inf
10,000,000	0.1715	0.0605	0.0192	0.0116	0.0077	0.0058	0.0047	inf	inf
100,000,000	0.1715	0.0605	0.0192	0.0116	0.0076	0.0058	0.0047	inf	inf

RMSE with step as reference signal

C_0 / β	0.001	0.01	0.1	0.25	0.5	0.75	1	2.5	5
100	0.4815	0.405	0.3965	0.319	0.3323	0.3257	inf	inf	inf
1000	0.3994	0.2201	0.043	0.0223	0.0154	0.0151	inf	inf	inf
10,000	0.2283	0.0449	0.0164	0.0123	0.0152	inf	inf	inf	inf
100,000	0.1325	0.0432	0.0161	inf	inf	inf	inf	inf	inf
1,000,000	0.1293	0.0431	0.0162	inf	inf	inf	inf	inf	inf
10,000,000	0.1291	0.0431	0.0162	inf	inf	inf	inf	inf	inf
100,000,000	0.1291	0.0431	0.0162	inf	inf	inf	inf	inf	inf

Table 4.3: Comparison of performance of sine and step as reference signals

This analysis shows that if our desired output changes, the controller changes its ranges of operation and successfully adapts to the required need. This finding is important for the designer as to be careful in selecting parameters of the adaptive controller (i.e., β and C_0) for moving ahead and implementing the global learning of controller.

Again, it is pertinent to mention here that the operating ranges of the controller drastically change if it is supposed to follow step signal as the reference signal. It means that the rules adapted do not work for regions of $C_0 > 0.25$ and β > 10,000 and, the best results after which the controller becomes unstable is this extreme limit (figure 4.13). Also, the change in β becomes ineffective for $C_0 < 0.25$ and B > 10,000.

4.3. Performance of Adaptive block with Deadzone as non-linearity

Once again, fuzzy adaptive controller was implemented on the same linear plant i.e, Motor control but with Dead zone with limits of [-0.5 to 0.5] as an added nonlinearity. The Simulink model of the implementation is given in Figure 4.17.

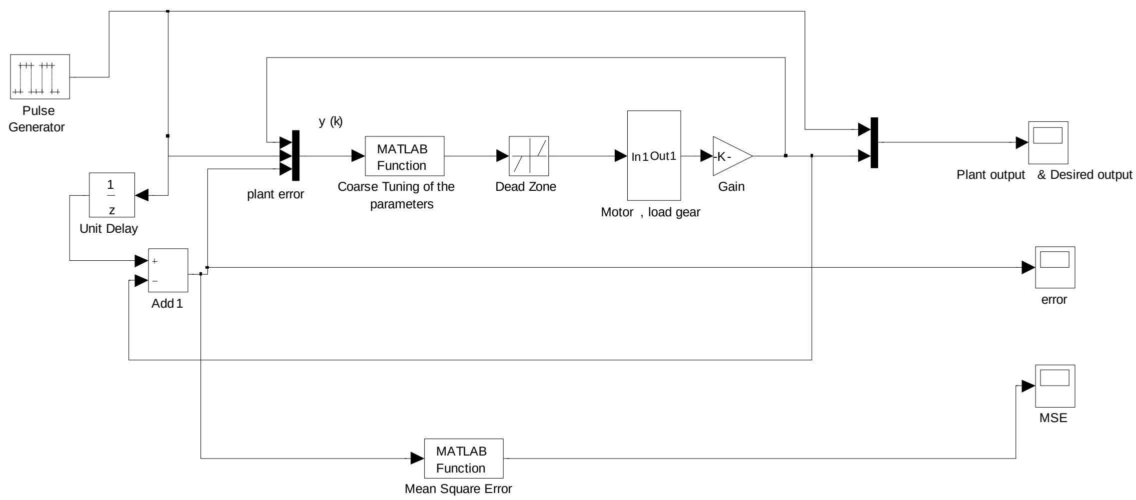

Figure 4.14: Simulink Model - Adaptive controller with Dead zone as Non-linearity

The idea here is to demonstrate the performance of the controller for different non-linear systems. The results for various parameters are visible from the figures 4.18 to 4.21, the controller works relatively very well for both the step signals as well as the sinusoidal signals as reference.

As per our thesis findings, it is evident that for poor selection of C_0 and β, the results of Figure 4.18 and 4.20 show that the control signal is NOT adapted. However, once we select good values of C_0 and β, results improve drastically. And of course, if we perform our proposed detailed analysis, we can pick out the best values of our controller parameters. Here, once again there is no need to perform global learning because of good results.

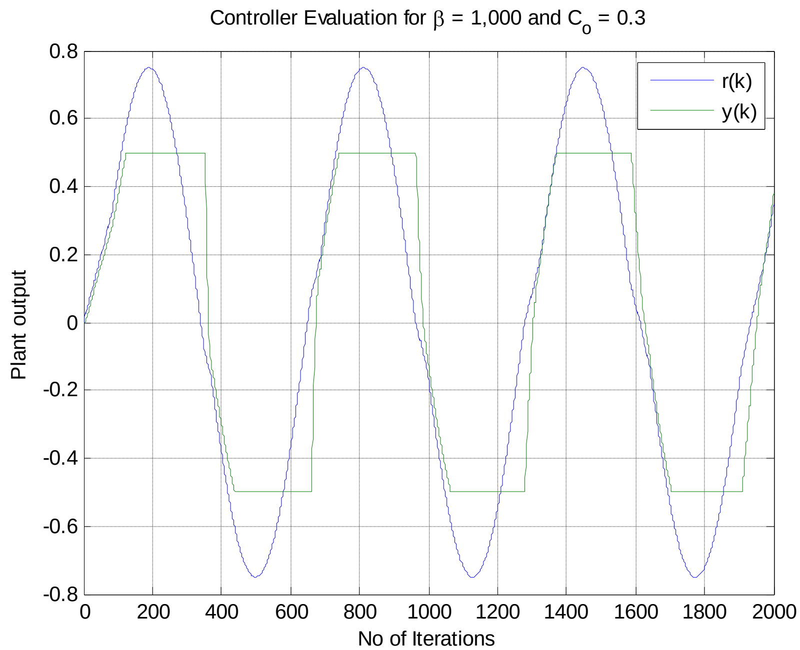

Figure 4.15: Controller evaluation of plant with Dead Zone as non-linearity for $C_0 = 0.3$ and β = 1,000

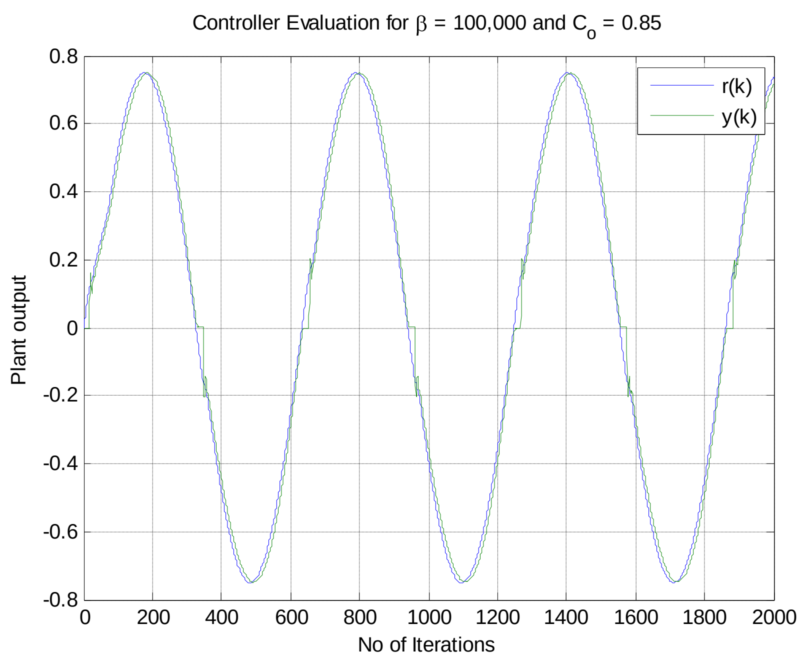

Figure 4.16: Controller evaluation of plant with Dead Zone as non-linearity for $C_0 = 0.85$ and β = 100,000

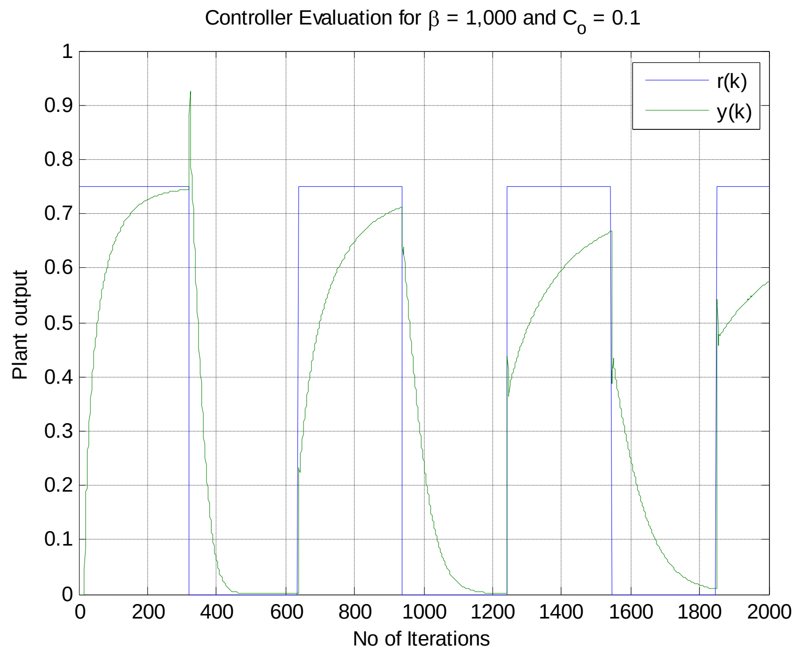

Figure 4.17: Controller evaluation of plant with Dead Zone as non-linearity for $C_0 = 0.1$ and β = 1,000

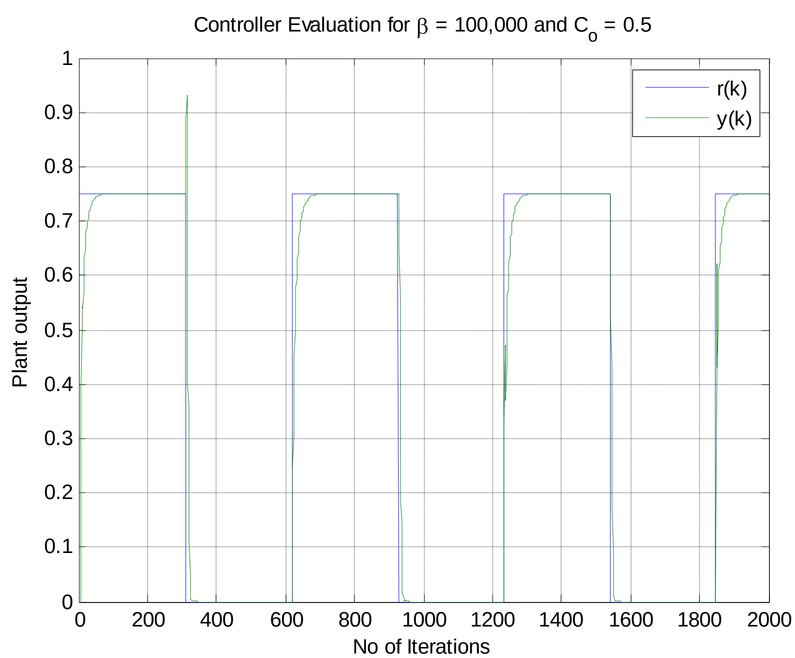

Figure 4.18: Controller evaluation of plant with Dead Zone as non-linearity for $C_0 = 0.5$ and β = 100,000

4.4. Performance of Adaptive block with Backlash as non-linearity

Similarly, applying Backlash with dead-zone of 1 as non-linearity in Simulink Model shown in Figure 4.22 for the test of the capability of our fuzzy controller shows excellent results. We can see in figure 4.24 that a good selection of C_0 and β is able to solve the control problem as compared to a casual selection of parameters of figure 4.23. If we do detailed analysis in this case, we will again be achieving the best results. Once again, Global learning becomes redundant in this case.

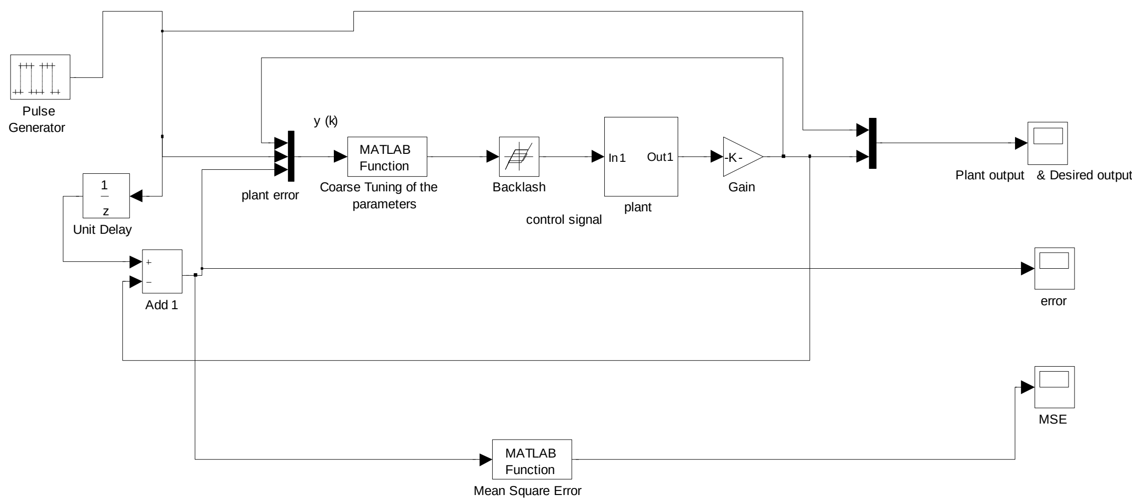

Figure 4.19: Simulink Model - Adaptive controller with Back lash as Non-linearity

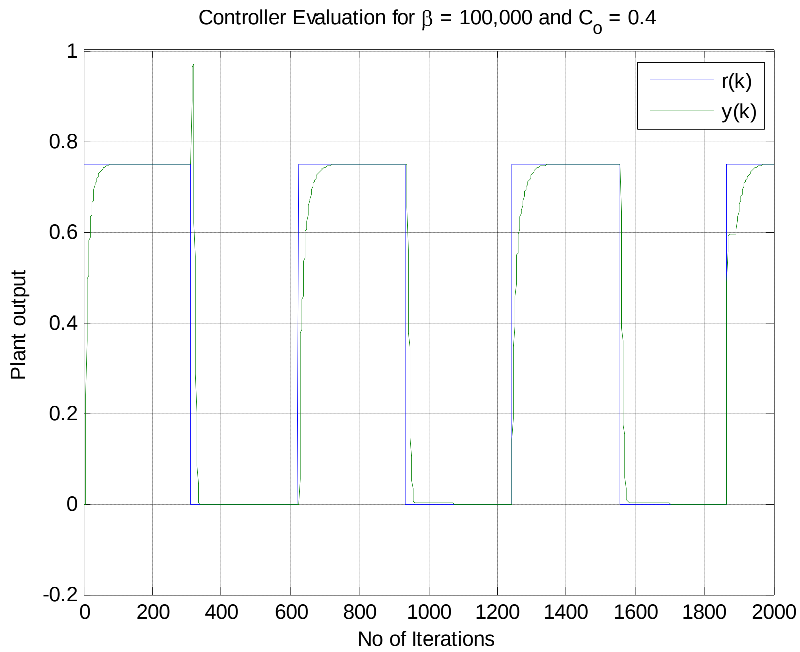

Figure 4.20: Controller evaluation of plant with Back lash as non-linearity for C_0 = 0.4 and β = 100,000

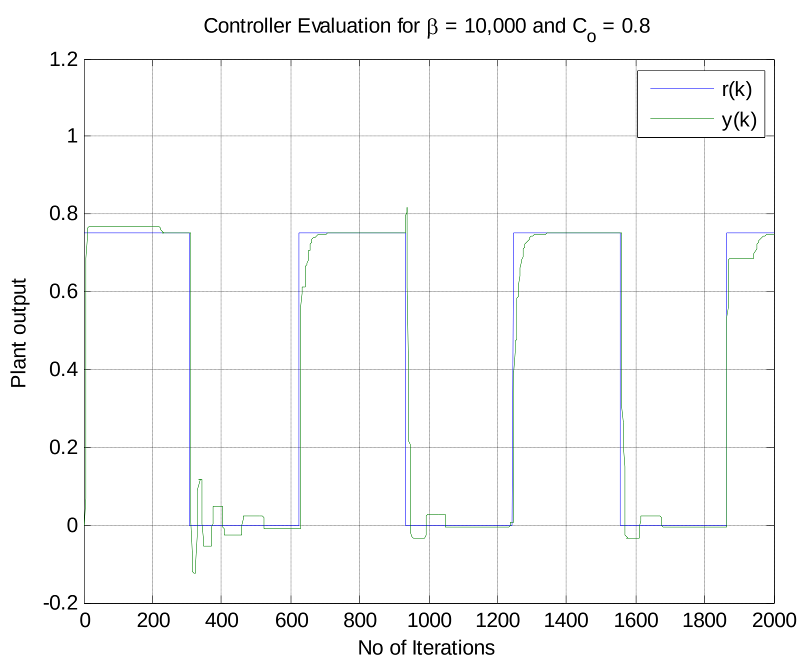

Figure 4.21: Controller evaluation of plant with Back Lash as non-linearity for C_0 = 0.8 and β = 10,000

4.5. Performance of Adaptive block with Saturation as non-linearity

Our fuzzy adaptive controller was implemented on the same linear plant i.e., motor control but with saturation as an added nonlinearity. The Simulink model of the implementation is given in Figure 4.14.

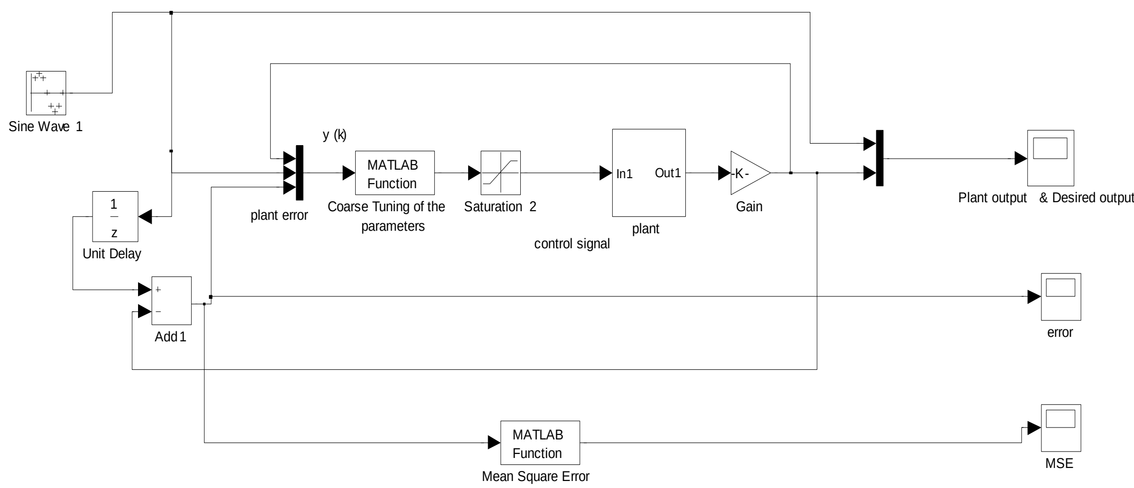

Figure 4.22: Simulink Model - Adaptive controller with Saturation as Non-linearity

This is a unique case in which the controller is unable to completely adapt the desired results. As it can be seen from the figures 4.15 and 4.16, the controller tries to learn but is incapable of doing so with saturation as non-linearity. This is a specific class of non-linearity which the adaptive controller is unable to adapt.

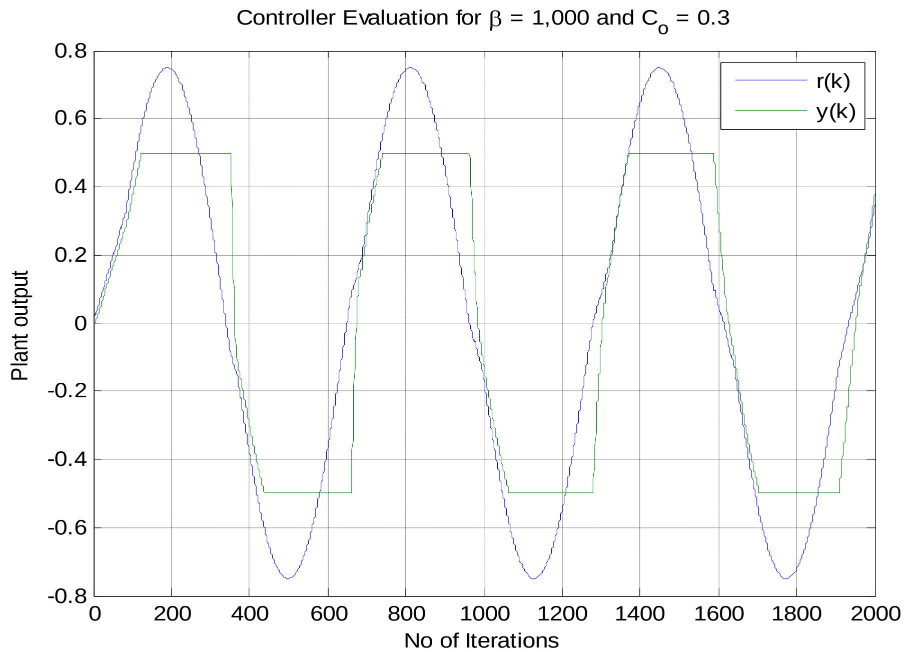

Figure 4.23: Controller evaluation of plant with saturation as non-linearity for $C_0 = 0.3$ and β = 1,000

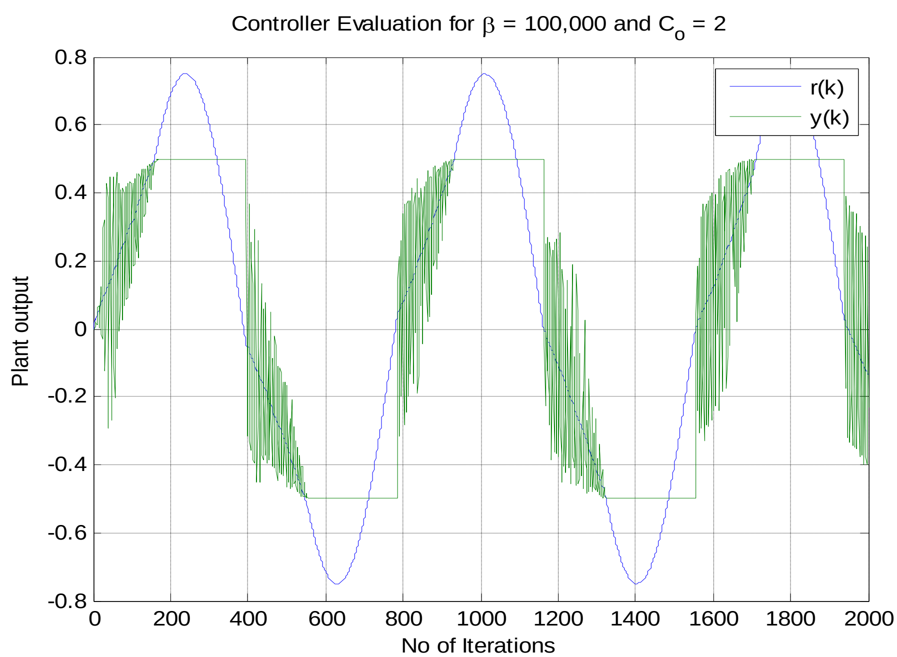

Figure 4.24: Controller evaluation of plant with saturation as non-linearity for $C_0 = 2$ and β = 100,000

4.6 Real time Controlling of water level in three tank system

After thorough review of the adaptive controller's strength on simulations, it is now tested on an actual non-linear plant. The adaptive controller now attempts to control water level on a three tank system in real time. This adaptive controller does not have any information about the plant to be controlled and it is not tuned or pre-trained offline. The actual plant is demonstrated in figure 4.25.

Figure 4.25: Actual plant (Water level control of Three Tank System)

Figure 4.26: Magnified picture of Actual plant (Water level control of Three Tank System)

The details of the real plant are as under:

1. There are total 3 tanks interconnected with valves
2. Each tank is of 10.3cm x 9.2cm x 60cm dimensions
3. The valves used are ball valve of 0.5"
4. The two interconnected valves are completely open.
5. Height in middle tank is measured using pressure transducer.
6. Water pump has a maximum pumping capacity of 36001/h
7. There is one inlet for flow of water inside the tanks represented by white pipe in figure 4.26.
8. There is one outlet for flow of water outside the tanks represented by red pipe in figure 4.26.

With the above plant, the adaptive controller is required to control and maintain the water level in the center tank at 0.5 units for 500 iterations. Each of the iteration corresponds to 1 second, and each unit of height corresponds to a water level of approximately 2.66cm in the center tank. Consequently, the adaptive controller is required to maintain the water level in the middle tank to around 13cm for 500 seconds i.e. for about 8.3 minutes. After this is achieved, the controller needs to shift the water level of the main tank to 0.9 units and maintain it for the next 500 iterations. This means that the controller now pushes the water level of the middle tank from 13cm to 24cm and maintains it to 24cm for another 8.3 minutes. This cycle is repeated for another 16.6 minutes.

Here, it is pertinent to mention that the controller is controlling only the speed of the pump with which it delivers water to the tanks. It does this by sensing the output height of the middle tank. It does not concern itself with the amount of water flowing out of the plant or the amount of water distributed in other tanks. It works on a goal oriented approach by viewing the desired height and generating the required control signal which would trigger the water pump to control/maintain that height.

The simulink model which contains the adaptive controller and controls the plant is depicted in figure 4.27. Apart from the Matlab function which contains the code for adaptive controller, three separate Matlab functions were added. The first Matlab function namely “limits of control signal” was added to ensure that the control output does not saturate the water pump. The second one converts the voltages received from the pressure sensor into their equivalent heights. Finally, the third function converts the heights obtained from the real plant to the units which are being input to the controller as desired results.

The results are then checked by plotting the desired response with respect to the real response generated by the plant.

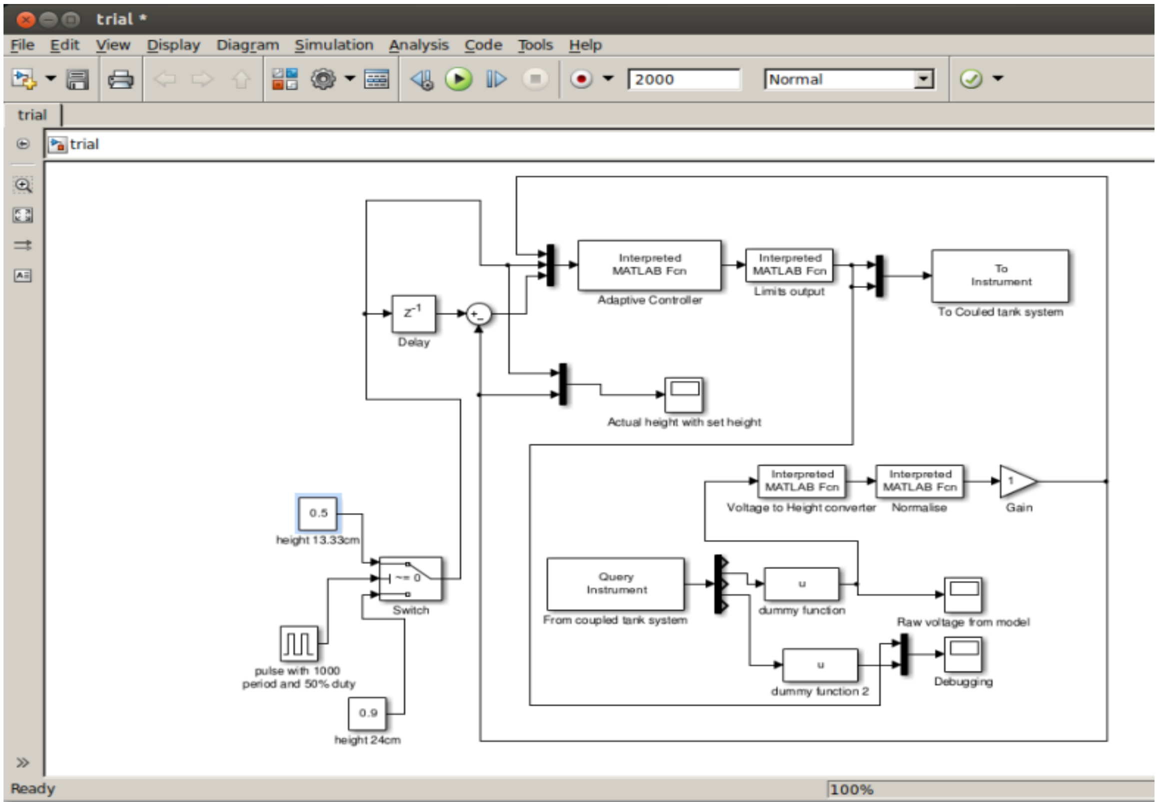

Figure 4.27: Simulink Model which connects to the Real Plant.

After controller evaluation for different C_0 and β, one of the best control results were achieved for β=100,000 and $C_0 = 20$. As can be seen clearly from control performance in figure 4.28, the adaptive controller has an initial offshoot in the initial cycle. This offshoot is because of the inadequate learning of the controller in the beginning but once the controller gathers enough online information about the plant, it learns well and controls the plant as per desired height of 0.5 units. Now, after 500 iterations, the controller senses automatically that the desired height has changed to 0.9 units and thus pushes the plant to act accordingly. As the controller is given more iteration, its results start to improve further and the weights of the adaptive controller have been tuned as per the needs of the three tank system.

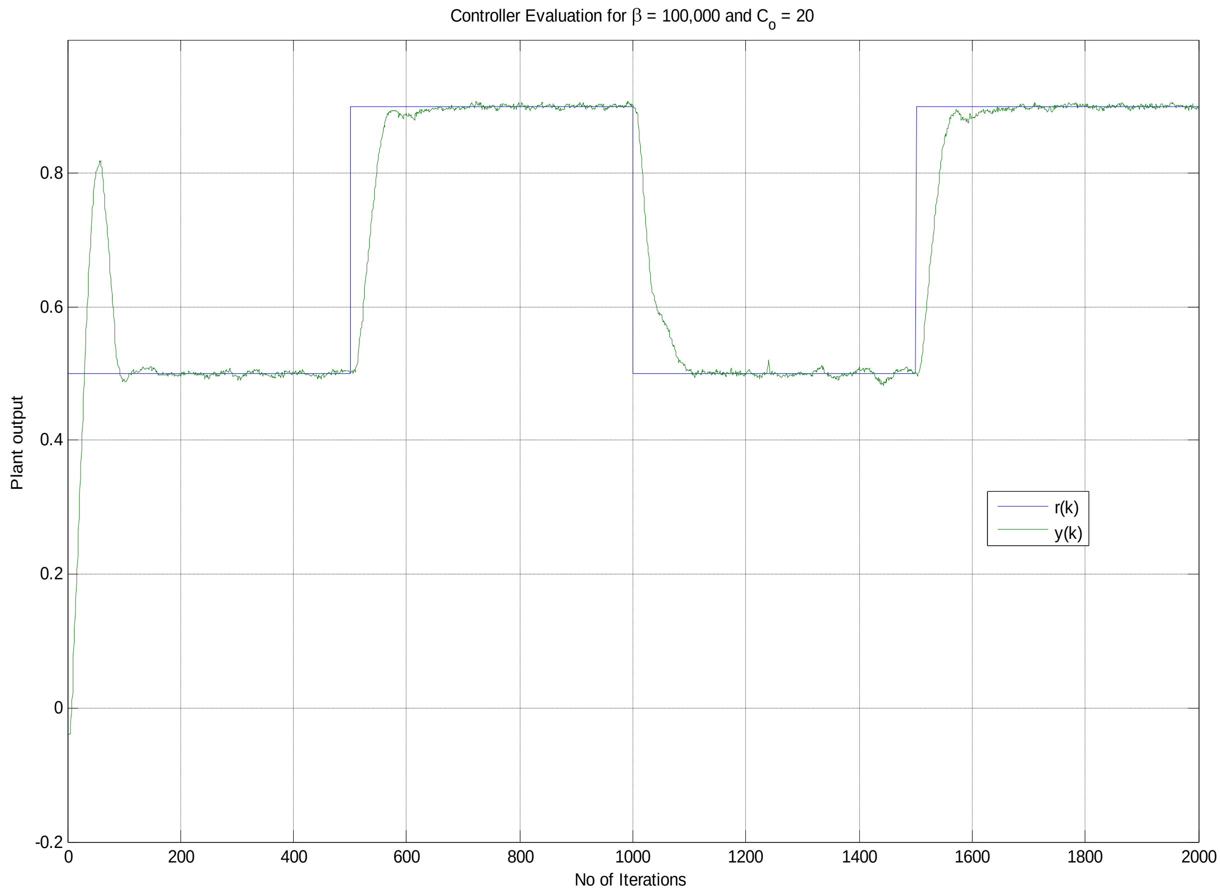

Figure 4.28: Controller evaluation of actual plant
(Water level control of Three Tank System)

4.7 Chapter Summary

This chapter analyzes the performance of the online tuning of fuzzy controller which is implemented by the studied adaptive controller on different linear and non-linear systems as well as considering different set points. Simulation results show that if this analysis is not done in a comprehensive manner, the results of controller can drastically change. The parameters in discussion i.e., C_0 and β effect the controller performance to the extent that it is necessary to ascertain their optimal values. Finally, this adaptive controller was successful in online controlling of water level in a real three tank system which is an excellent proof of the diverse controlling ability of this adaptive fuzzy controller.

CHAPTER 5

5. Future of the Proposed Control Scheme

5.1 Future of the Control Scheme

The aim of this work was to demonstrate that a capable controller can be designed for plants which lack complete information. The suggested modifications were to improve its results but what remains as part of the challenge is the limitations that are encountered with such controllers.

The following limitations are under research for improvement.

1. Such controllers aren't able to demonstrate stability of the closed-loop system.

2. Such controllers are inapplicable to reach certain states of the plant i.e., since we are unaware of the states of the plant, we cannot ascertain which state to enter. In this case, some pre training is needed to ensure that at the start of the operation of the controller these states are not encountered.

3. Such controllers are only applicable to plants in which the output of the plant is directly dependent on the previous control signal. Therefore the sampling period for the controller should be adjustable in such a manner.

These are the future prospects of this type of controllers. This work presents a preliminary structure which aims at developing new controllers which ultimately will solve all the above issues and develop a universal controller.

5.2 Conclusion

Whilst performing a control process, a very challenging subject for researchers of today is to acquire techniques that are capable of working with smallest available information.

These techniques share the inconvenient of pre defining structure of fuzzy controller and similar issues which we have previously addresses.

As per literature review for this work it is evident that online adaptation of rule consequents as well as rule premise for the fuzzy controller without offline pre-training of the plant has been little studied.

This was an attempt to move forward in the direction of these interesting works by identifying an improvement mechanism. For future research, some automatic tuning technique can be designed to find the best nominal values of these online adaptive controller parameters.

As outlined by this study and various simulation results presented along with controlling of real time non-linear plants, it is evident that the proposed modification ensures better controlling and better working of online adaptive controller for the Global learning.

References

[1] J. Castro, "Fuzzy logic controllers are universal approximators," IEEE Trans. Syst., Man, Cybern., vol. 25, no. 4, pp. 629–635, Apr. 1995.

[2] Mamdani, E. H. and Assilian S. "An Experiment with in Linguistic Synthesis with a Fuzzy Logic Controller", Int. J. Man-Machine Studies, vol. 7 pp 1-13, 1975

[3] E.H. Mamdani "Twenty years of Fuzzy Control: Experiences Gained and Lessons Learnt"

[4] Zadeh, L. A "Outline of a new approach to the analysis of a complex system and decision processes", IEEE Trans. SMC, vol. 3, pp 28, 1973

[5] Holmblad, L.P. and Ostergaard, J. J. , "Control of a Cement Kiln by Fuzzy Logic", Fuzzy hformahon and Decision Processes (Ed M.M. Gupta et al) North Holland 1982

[6] Sugeno, M. (Ed) Industrial Application of Fuzzy Control, North-Holland, 1985

[7] T. Procyk and e. Mamdani, "A linguistic self-organizing process controller", Automatica, vol. 15, pp. 15-30, 1979.

[8] R. Ordonez, J. Zumberge, J. t. Spooner, and K. M. Passino, "Adaptive fuzzy control: Experiments and comparative analyzes", IEEE Trans. Fuzzy syst., vol. 5, pp. 167-188, May 1997.

[9] H. Pomares, I. Rojas, F. J. Fernandez, M. Anguita, E. Ros, and A. Prieto, " A new approach for the design of fuzzy controllers in real time", in Proc. 8th Int. Conf. Fuzzy systems, Seoul, Korea, Aug. 1999, pp. 522-526.

[10] H. Pomares, I. Rojas, J. Gonzalez, M. Damas, B. Pino and A. Prieto, " Online Global Learning in direct fuzzy controllers", in IEEE Trans. On Fuzzy Systems, vol. 12, NO. 2, April. 2004.

[11] H. Andersen, A. Lotfi, and A. Tsoi, "A new approach to adaptive fuzzy control: the controller output error method," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 27, no. 4, pp. 686–691, Aug. 1997.

[12] A. Rajapakse, K. Furuta, and S. Kondo, "Evolutionary learning of fuzzy logic controllers and their adaptation through perpetual evolution," IEEE Trans. Fuzzy Syst., vol. 10, no. 3, pp. 309–321, Jun. 2002.

[13] S. Lalouni, D. Rekioua, T. Rekioua, and E. Matagne, "Fuzzy logic control of stand-alone photovoltaic system with battery storage," J. Power Sources, vol. 193, no. 2, pp. 899–907, 2009.

[14] M. Galluzzo and B. Cosenza, "Control of the biodegradation of mixed wastes in a continuous bioreactor by a type-2 fuzzy logic controller," Comput. Chem. Eng., vol. 33, no. 9, pp. 1475–1483, 2009.

[15] Y. Liu andY. Zheng, "Adaptive robust fuzzy control for a class of uncertain chaotic systems," Nonlinear Dyn., vol. 57, no. 3, pp. 431–439, 2009.

[16] W. Wang, Y. Chien, and I. Li, "An on-line robust and adaptive T-S fuzzy neural controller for more general unknown systems," Int. J. Fuzzy Syst., vol. 10, no. 1, pp. 33–43, 2008.

[17] Y. Gao and M. J. Er, "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems," IEEE Trans. Fuzzy Syst., vol. 11, no. 4, pp. 462–477, Aug. 2003.

[18] J. Park, G. Park, S. Kim, and C. Moon, “Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system,” Fuzzy Sets Syst., vol. 153, no. 3, pp. 429–445, 2005.

[19] P. A. Phan and T. J. Gale, “Direct adaptive fuzzy control with a self structuring algorithm,” Fuzzy Sets Syst., vol. 159, no. 8, pp. 871–899,
2008.

[20] C. Li and C. Lee, “Self-organizing neuro-fuzzy system for control of unknown plants,” IEEE Trans. Fuzzy Syst., vol. 11, no. 1, pp. 135–150, Feb. 2003.

[21] C. Juang, “A symbiotic genetic algorithm with local-and-global mapping search for reinforcement fuzzy control,” J. Intell. Fuzzy Syst., vol. 19, no. 2, pp. 103–114, 2008.

[22] M. Mucientes and J. Casillas, “Quick design of fuzzy controllers with good interpretability in mobile robotics,” Fuzzy Systems, IEEE Transactions on, vol. 15, no. 4, pp. 636–651, 2007.

[23] K. Shimojima, T. Fukuda, and Y. Hashegawa, “Self-Tuning modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm,” Journal on Fuzzy Sets and Systems, vol. 71, pp. 295–309, 1995, 3.

[24] H. Mingzhi, W. Jinquan, M. Yongwen, W. Yan, L. Weijiang, and S. Xiaofei, “Control rules of aeration in a submerged biofilm wastewater treatment process using fuzzy neural networks,” Expert Systems with Applications, vol. 36, no. 7, pp. 10 428–10 437, Sep. 2009.

[25] L. X. Wang, Adaptive Fuzzy Systems and Control, Design and Stability Analysis. ,Upper Saddle River, NJ: Prentice Hall, 1994.

[26] I. Skrjanc, S. Blazic and D. Matko, “Direct fuzzy model-reference adaptive control”, Int. J. Intell, Syst, vol. 17, pp 943-963, 2002

[27] I. Rojas, H. Pomares, J. Gonzalez, L. Herrera, A. Guillen, F. Rojas, and O. Valenzuela, “Adaptive fuzzy controller: Application to the control of the temperature of a dynamic room in real time,” Fuzzy Sets Syst., vol. 157, no. 16, pp. 2241–2258, 2006.

[28] I. Rojas, H. Pomares, F. J. Pelayo, M. Anguita, E. Ros, and A. Prieto, “New methodology for the development of adaptive and self-learning fuzzy controllers in real time”, Int. J. Approx. Reason., vol. 21, pp. 109-136, 1999.

[29] P. Angelov, R. Buswell, J. A. Wright, and D. Loveday, “Evolving rule based control,” in EUNITE Symposium, Tenerife (Spain), 2001, pp. 36–41.

[30] Chih-Min Lin and Yi-Jen Mon“Hybrid adaptive fuzzy controllers with application to robotic systems“, Fuzzy Sets and Systems, Vol. 139, Issue 1, pp. 151-165, 2003.

[31] Changchun Hua, Xinping Guan and Guangren Duan. “Variable structure adaptive fuzzy control for a class of nonlinear time-delay systems“, Fuzzy Sets and Systems, In Press, Uncorrected Proof, Available online 19 November 2003.

[32] Y.P. Singh, “A modified self- organizing controller for real-time process control applications”, Fuzzy Sets Syst.., vol. 96, pp. 147-159, 1998.

[33] M. Zarandi, “Reinforcement learning for fuzzy control with linguistic states,” J. Uncertain Syst., vol. 2, no. 1, pp. 54–66, 2008.

[34] A. B. Cara, H. Pomares, and I. Rojas, “A new methodology for the online adaptation of fuzzy Self-Structuring controllers,” Fuzzy Systems, IEEE Transactions on, vol. 19, no. 3, pp. 449–464, 2011.

[35] A. B. Cara, H. Pomares, I. Rojas, Z. Lendek, and R. Babuska, “Online self-evolving fuzzy controller with global learning capabilities,” Evolving Systems, vol. 1, no. 4, pp. 225–239, Oct. 2010.

[36] A. B. Cara, Z. Lendek, R. Babuska, H. Pomares, and I. Rojas, “Online self-organizing adaptive fuzzy controller: Application to a nonlinear servo system,” in Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, 2010, pp. 1–8.