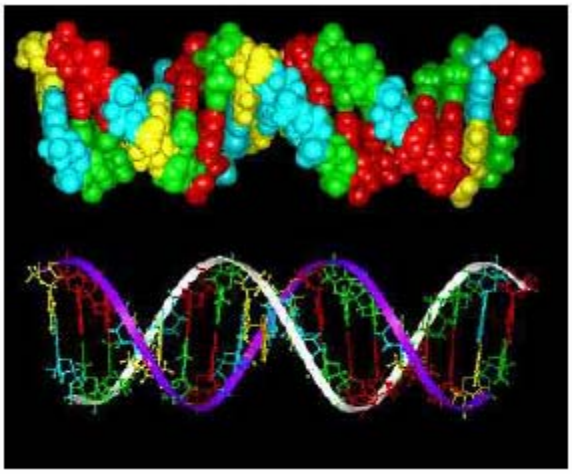
ADAPTIVE CONTROL AND TRACKING OF INTRACELLULAL NETWORKS USING RECURRENT HIGH ORDER NEURAL NETWORKS

By
Papadakis Dimitrios (S.N.: 2001030099)



Advisor: Prof. Manolis Christodoulou
Co-advisors: Prof. M. Zervakis, Assoc. Prof. E. Georgiou

Diploma thesis
Presented to the Department of Electronic and Computer Engineering of
Technical University of Crete, GREECE

Technical University of Crete, GREECE
2006

To my family
Manolis, Georgia, Maria, Katerina

ACKNOWLEDGEMENTS

I would want firstly to thank my professor and supervisor of this work, Manolis A. Christodoulou for the guidance all the duration of this thesis.

At this point I also want to thank my parents for the moral and material support at all the duration of my studies at Technical University of Crete. Special thanks to my sister Katerina for her supporting at many difficulties during this thesis.

Contents

List of Figures

Chapter 1

INTRODUCTION

Recent technological developments have pushed controls engineers to deal with very complex systems that are having uncertain and possibly unknown nonlinearities, operating in highly uncertain environments. Man has two principal objectives in the scientific study of his environment: he wants to understand and to control. The two goals reinforce each other, since deeper understanding permits firmer control, and, on the other hand, systematic application of scientific theories inevitably generates new problems which require further investigation, and so on.

In this project, an adaptive control algorithm based on Recurrent High Order Neural Networks (RHONNs) is used to control and tracking Intracellular networks of the cell.
The close loop signals are uniformly ultimately bounded and the output of the system is proven to follow a desired trajectory of the linear reference system. Simulation results are presented to show the effectiveness of the approach proposed in order to control the two molecular species.

The cell is made up of molecules, like a car is made up from plastic and metal. But a soup of molecules is no more a cell than a heap of plastic and metal is a car. To understand *the functioning and function* of a cell we need to know the relations and interactions of the components that constitute it. If the central dogma of systems biology is that *it is dynamics that determines biological function*, we would argue that *the dynamical manifestation of feedback determines the development and maintenance of biological process.*

Chapter 2

A GENERAL OVERVIEW IN MOLECULAR BIOLOGY

Because molecular biology has a vital role in this work and it is necessary to understand some fundamentals functions and structures of the cell.

The best approach for Genomic information is to consider this as digital because it is represented in the form of sequences of which each element can be one out of a finite number of entities.
To be more specific, DNA, RNA and proteins have been mathematically represented by character strings, in which each character is a letter of an alphabet. In the case of DNA, the alphabet is of size four (4) and consists one of the letters A, T, C and G. In addition to the case of the RNA the alphabet has four elements (letters) A, U, C and G. Finally in the case of proteins, the size of the corresponding alphabet is 20.
The next section presents a more detailed description of the nature of biomolecular sequences.

The nature of biomolecular sequences

DNA

A single strand of DNA is a biomolecule consisting of many linked, smaller components called nucleotides. Each nucleotide is one of four possible types designated by the letters A, T, C, and G and has two distinct ends, the 5' end and the 3' end, so that the 5' end of a nucleotide is linked to the 3' end of another nucleotide by a strong chemical bond, thus forming a long,
one-dimensional chain (backbone) of a specific directionality. Therefore, each DNA single strand is mathematically represented by a character string which, by convention specifies the 5' to 3' direction when read from left to right. Single DNA strands tend to form double helices with other single DNA strands. Thus, a DNA double strand contains two single strands called complementary to each other because each nucleotide of one strand is linked to a nucleotide of the other strand by a chemical bond, so that A is linked to T and vice versa, and C is linked to G and vice versa.

Each such bond is weak compared to the bonds forming the backbone, but together all these bonds create a stable, double helical structure. The two strands run in opposite directions in which we see the sugar-phosphate chemical structure of the DNA backbone created by strong (covalent) bonds, and that each nucleotide is characterized by a base that is attached to it.
The two strands are linked by a set of weak (hydrogen) bonds. In order to have a more stable 3d structure, DNA tend to take helix form.

Because each of the strands of a DNA double strand uniquely determines the other strand, a double-stranded DNA molecule is represented by either of the two character strings read in its 5' to 3' direction. Thus, in the example above, the character strings CATTGCCAGT and ACTGGCAATG can be alternatively used to describe the same DNA double strand, but they specify
two different single strands which are complementary to each other. DNA strands that are complementary to themselves are called self-complementary, or palindromes. For example AATCTAGATT is a palindrome.

DNA molecules store the digital information that constitutes the genetic blueprint of living organisms. This digital information has been

created and reliably stored throughout billions of years of evolution during which some vital regions of DNA sequences have been remarkably preserved, despite striking differences in the body plans of various animals.

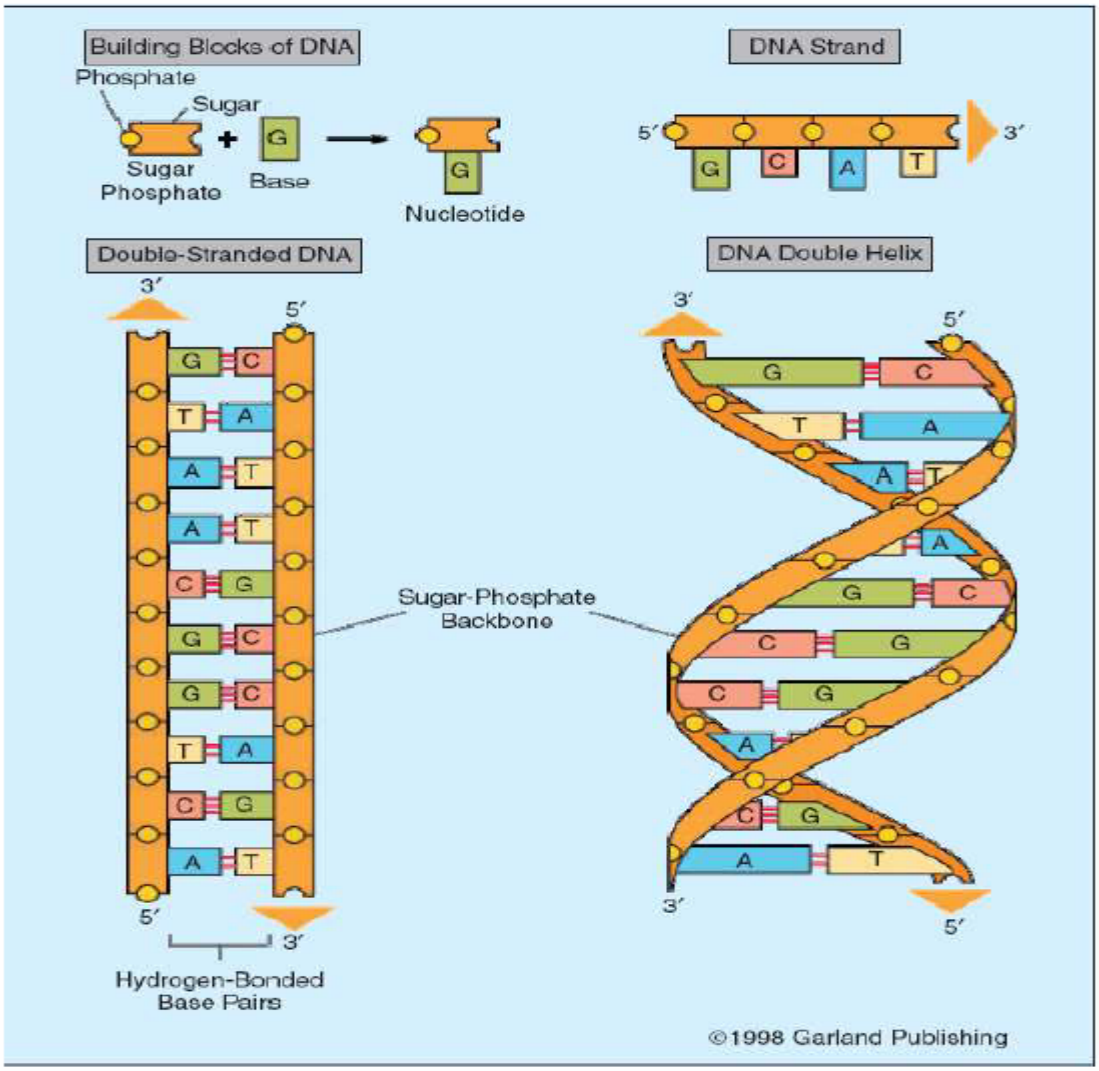


Figure 2.1 DNA and its building blocks

RNA

RNA is very similar in structure to DNA. It contains the base **urasil** instead of the base **thymine**. RNA is single-stranded (this is one major difference with DNA).
There are several types of RNA. M-RNA is copied from genes. Then intron sequences are removed by RNA splicing. The 5' end of m-RNA is capped and a poly-A tail is added to the 3' tail.

T-RNA recognizes and binds the codon and the amino acid in protein translation. Four segments of t-RNA are double-helical. One of these regions forms the anticodon that pairs with the complementary codon in an m-RNA molecule. The other is the site where the amino acid is attached to the t-RNA.

Proteins

A protein is also a biomolecule consisting of many linked smaller components called amino acids. There are twenty possible types of amino acids in proteins, the single strands are connected with strong bonds, one after the other, forming a long one-dimensional chain (backbone) of a specific directionality. Therefore, as in DNA, a character string mathematically represents each protein. The length of a character string representing a protein molecule is relatively small, typically in the hundreds, while the length of a character string representing a DNA molecule in the living cell is typically in the millions, or even hundreds of millions.

Protein molecules tend to fold into complex three-dimensional (3D) structures forming weak bonds between their own atoms, and they are responsible for carrying out nearly all of the essential functions in the living cell by properly binding to other molecules with a number of chemical bonds connecting neighboring atoms. Thus, protein functions are largely determined by their 3D structures because these geometrical shapes often determine whether a protein can bind to another molecule by a process reminiscent of a hand fitting into a glove.

Although we do not yet know how to reliably predict protein 3-D structures from their one-dimensional amino acid sequences, we do know that nearly all proteins in the living cell are uniquely determined by these sequences. Therefore, the amino acid character strings determine the functions of proteins. In fact, protein functions are ultimately determined by the DNA character string because it is the digital information in the DNA nucleotide sequences that determine the amino acid sequences; each protein character string is generated based on information in genes, which are regions in the DNA character strings.

Protein synthesis is governed by the genetic code which maps each of the 64 possible triplets (codons) of DNA characters into one of the 20 possible amino acids.

The things are not simple because the DNA character string composed of regions which code in a protein and other ones serving yet unknown functions! From the above discussion it is clear that the total number of nucleotides in the protein coding area of a gene will be a multiple of three and the area will be bounded by a START codon and it will finished with a STOP codon, and that there will be no other STOP codon in this frame in between. However, given a long nucleotide sequence, it is very difficult to accurately designate where the genes are.

For making this function more understandable and for completeness of our work I am giving the figure of the genetic code:

AAA: K (Lys)	GAA: E (Glu)	TAA: STOP	CAA: Q (Gln)
AAG: K (Lys)	GAG: E (Glu)	TAG: STOP	CAG: Q (Gln)
AAT: N (Asn)	GAT: D (Asp)	TAT: Y (Tyr)	CAT: H (His)
AAC: N (Asn)	GAC: D (Asp)	TAC: Y (Tyr)	CAC: H (His)
AGA: R (Arg)	GGA: G (Gly)	TGA: STOP	CGA: R (Arg)
AGG: R (Arg)	GGG: G (Gly)	TGG: W (Trp)	CGG: R (Arg)
AGT: S (Ser)	GGT: G (Gly)	TGT: C (Cys)	CGT: R (Arg)
AGC: S (Ser)	GGC: G (Gly)	TGC: C (Cys)	CGC: R (Arg)
ATA: I (Ile)	GTA: V (Val)	TTA: L (Leu)	CTA: L (Leu)
ATG: M (Met)/START	GTG: V (Val)	TTG: L (Leu)	CTG: L (Leu)
ATT: I (Ile)	GTT: V (Val)	TTT: F (Phe)	CTT: L (Leu)
ATC: I (Ile)	GTC: V (Val)	TTC: F (Phe)	CTC: L (Leu)
ACA: T (Thr)	GCA: A (Ala)	TCA: S (Ser)	CCA: P (Pro)
ACG: T (Thr)	GCG: A (Ala)	TCG: S (Ser)	CCG: P (Pro)
ACT: T (Thr)	GCT: A (Ala)	TCT: S (Ser)	CCT: P (Pro)
ACC: T (Thr)	GCC: A (Ala)	TCC: S (Ser)	CCC: P (Pro)

Figure 2.2 The Genetic Code

An Introduction to Systems Biology

Because of the sequencing of DNA for a number of genomes, the scientists have the vital opportunity to study the organization and control of genetic pathways. This new phase in the biological revolution, the postgenomic era, is associated with the fields ‘genomics’, transcriptomics, proteomics’ and “metabolomics”(called ‘the omics’ for short).

These fields take us from the DNA sequence of a gene to the structure of the product for which it codes (usually a protein) to the activity of that protein and its function within a cell, the tissue, and ultimatively the organism.

The **two** vital questions that scientists investigate are “What are the functional roles of genes? and “How do genes and/or proteins interact?

Answering these questions has become possible with new high-throughput technologies to take measurements at the molecular level. The identification of interrelationships between groups of genes (with respect to their functional role) and to analyze dynamic interactions among genes (gene networks). The most proteins interact with several other proteins and most of these interactions are the consequence of dynamic and controlled processes.

Genomic Cybernetics

Weaver defined “disorganized complexity” as a problem in which the number of variables is very large and any variable is best described as a random process. At this moment we are at the “molecular level” and the most successful method for representing phenomena at this level derives from statistical considerations.

On the other hand, at the “cellular level” matters are complicated by the fact that organization becomes a fundamental feature of the considered process.

Systems Biology provides a vital interface between cell biology and biotechnological applications. The complexity in the context of biological systems can be defined as

- A property of an encoding
- An attribute of the natural system under consideration
- Our ability to interact with the system and to observe it

Gene Expression and Regulation

Each cell of a multicellular organism holds the genome with the entire genetic material which is represented by a large double-stranded DNA molecule with the famous double-helix structure. The cell is the essential unit of living matter and it takes up chemical substances from its environment and transformed them.

The “central dogma” of biology describes how information, stored in DNA, is transformed into proteins via an intermediate product, called RNA. The transcription is the process by which coding regions of DNA synthesize RNA molecules.

After this, the “translation” is the process of synthesizing proteins using the genetic information in RNA as a template. The most of the proteins are enzymes that carry out the reactions responsible for the cell’s metabolism- the reactions that allow the cell to process nutrients, to built new cellular material, to grow and to divide.

Research conducted in the 1960s showed that most basic cellular processes are dynamic and feedback regulated. There are two types of genes based on their functionality. The two categories are:

- Structural genes which are responsible for coding of proteins and
- Regulatory genes which control the rate at which structural genes are transcribed.

This control of the rate of protein synthesis was the first indication that these processes are most appropriately viewed as dynamical systems.

Although bacteria cells are capable of producing several thousand different proteins, not all are produced at the same time or in the same quantity. The energy consumption for protein synthesis and the relatively short half-life of the RNA molecules are reasons for the cell to control both the types and amounts of each protein.

The central dogma of molecular biology

The “central dogma” of biology describes how information, stored in DNA, is transformed into proteins via an intermediate product, called RNA. The transcription is the process by which coding regions of DNA synthesize RNA molecules.

After this, the “translation” is the process of synthesizing proteins using the genetic information in RNA as a template. The most of the proteins are enzymes that carry out the reactions responsible for the cell’s metabolism- the reactions that allow the cell to process nutrients, to built new cellular material, to grow and to divide.

The next figure is very helpful for understanding this complicated process:

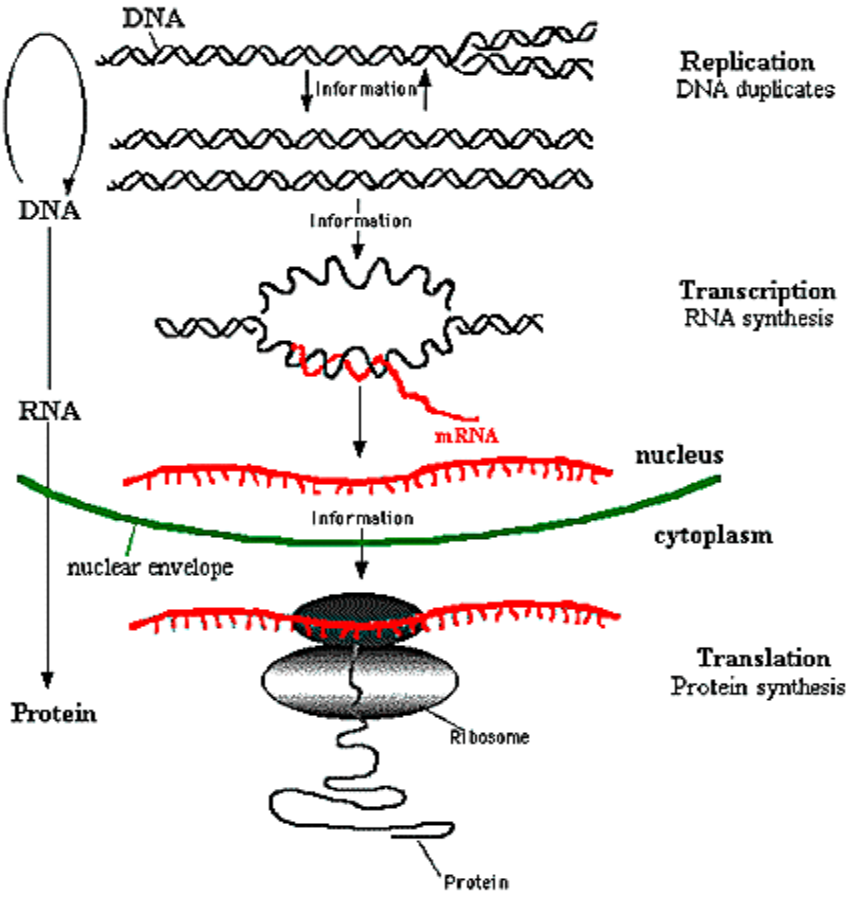


Figure 2.3 The central dogma of Molecular Biology

At this point I would try to explain each stage in a more detailed form:

The major types of proteins, which must work together during the replication of DNA, are illustrated, showing their positions.

When DNA replicates, many different proteins work together to accomplish the following steps:

1. The two parent strands are unwound with the help of DNA helices.
2. Single stranded DNA binding proteins attach to the unwound strands, preventing them from winding back together.
3. The strands are held in position, binding easily to DNA polymerase, which catalyzes the elongation of the leading and lagging strands. (DNA polymerase also checks the accuracy of its own work!).
4. While the DNA polymerase on the leading strand can operate in a continuous fashion, RNA primer is needed repeatedly on the lagging strand to facilitate synthesis of Okazaki fragments. DNA primase, which is one of several polypeptides bound together in a group called primosomes, helps to build the primer.
5. Finally, each new Okazaki fragment is attached to the completed portion of the lagging strand in a reaction catalyzed by DNA ligase.

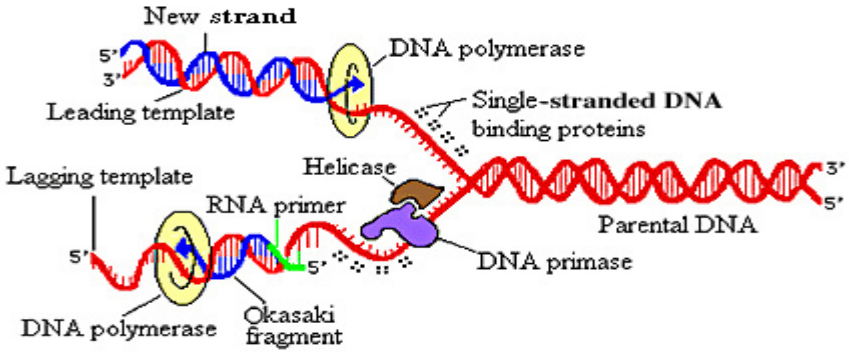


Figure 2.4 The process of collaboration of Proteins at the Replication Fork

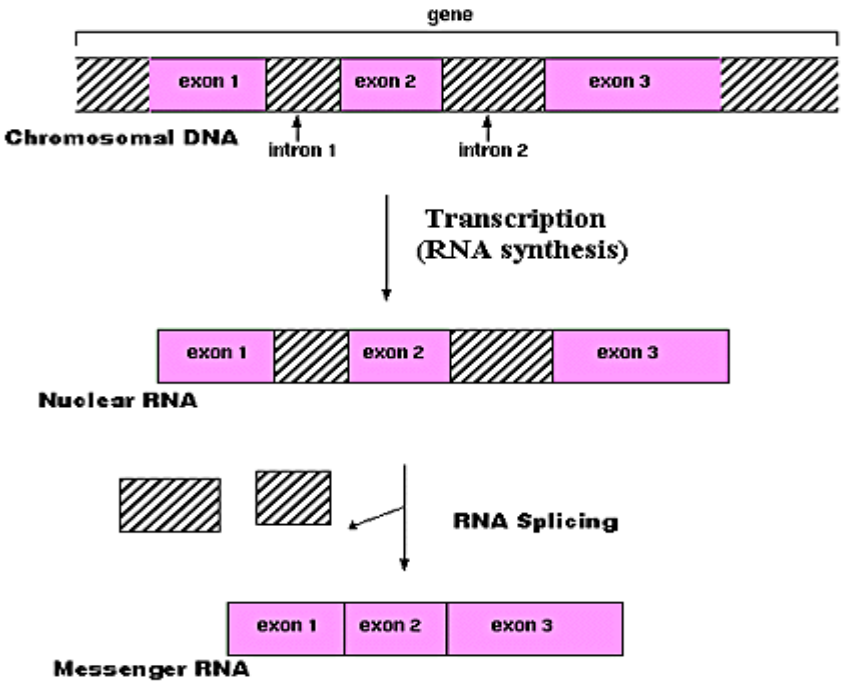
Little information about the RNA synthesis:

The process by which non-coding sequences of base pairs (introns) are subtracted from the coding sequences (exons) of a gene in order to transcribe DNA into messenger RNA (mRNA.)

In chromosomes, DNA acts as a template for the synthesis of RNA in a process called transcription. In most mammalian cells, only 1% of the DNA sequence is copied into a functional RNA (mRNA). Only one part of the DNA is transcribed to produce nuclear RNA, and only a minor portion of the nuclear RNA survives the RNA processing steps.

One of the most important stages in RNA processing is RNA splicing. In many genes, the DNA sequence coding for proteins, or "exons", may be interrupted by stretches of non-coding DNA, called "introns". In the cell nucleus, the DNA that includes all the exons and introns of the gene is first transcribed into a complementary RNA copy called "nuclear RNA," or nRNA. In a second step, introns are removed from nRNA by a process called RNA splicing. The edited sequence is called "messenger RNA," or mRNA.

The mRNA leaves the nucleus and travels to the cytoplasm, where it encounters cellular bodies called ribosomes. The mRNA, which carries the gene's instructions, dictates the production of proteins by the ribosomes.



RNA synthesis and processing

Figure 2.5 The RNA synthesis and processing in a summary form
Information about the protein synthesis:

Process where DNA encodes for the production of amino acids and proteins.

This process can be divided into two parts:

1. Transcription

Before the synthesis of a protein begins, the corresponding RNA molecule is produced by RNA transcription. One strand of the DNA double helix is used as a template by the RNA polymerase to synthesize a messenger RNA (mRNA). This mRNA migrates from the nucleus to the cytoplasm. During this step, mRNA goes through different types of maturation including one called splicing when the non-coding sequences are eliminated. The coding mRNA sequence can be described as a unit of three nucleotides called a codon.

2. Translation

The ribosome binds to the mRNA at the start codon (AUG) that is recognized only by the initiator tRNA. The ribosome proceeds to the elongation phase of protein synthesis. During this stage, complexes, composed of an amino acid linked to tRNA, sequentially bind to the appropriate codon in mRNA by forming complementary base pairs with the tRNA anticodon. The ribosome moves from codon to codon along the mRNA. Amino acids are added one by one, translated into polypeptidic sequences dictated by DNA and represented by mRNA. At the end, a release factor binds to the stop codon, terminating translation and releasing the complete polypeptide from the ribosome. One specific amino acid can correspond to more than one codon. The genetic code is said to be degenerate.

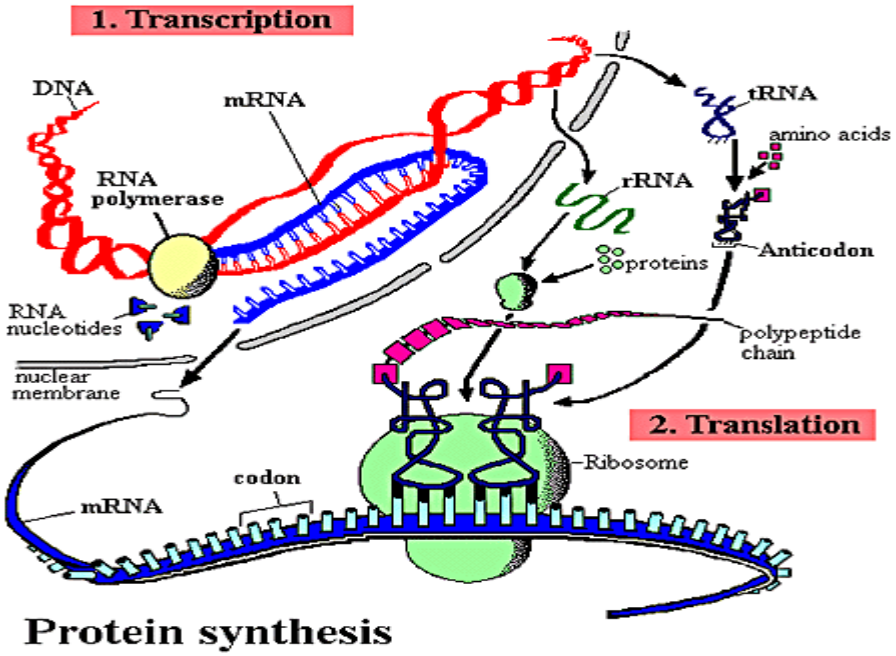


Figure 2.6 The two vital stages of Protein synthesis (Transcription and Translation)

Intra- and Intercellular Dynamics: Cellular Weather Forecasting

To determine how sells act and interact within the context of the organism to generate coherent and functional wholes, we must understand how information is transferred between and within cells. Cell signaling or “signal transduction” is the study of the mechanism that enable the transfer of biological information. Many diseases, such as cancer, involve malfunction of signal transduction pathways.

Bacteria regulate cell metabolism in response to a wide variety of environmental fluctuations, including the heat-shock example above. There must be mechanisms by which the cells receive signals from the environment and transmit them to the specific target to be regulated.

Receptors are proteins that span the membrane, with a site for binding the signaling compound on the outer surface of the receptor results in an activation of an intracellular protein.

In addition to, signal transduction pathways commonly consist of many more cascaded modules between receptor and genome. There can be numerous intermediate steps before the signal transduction process ends, often with a change in the gene expression program of the cell.

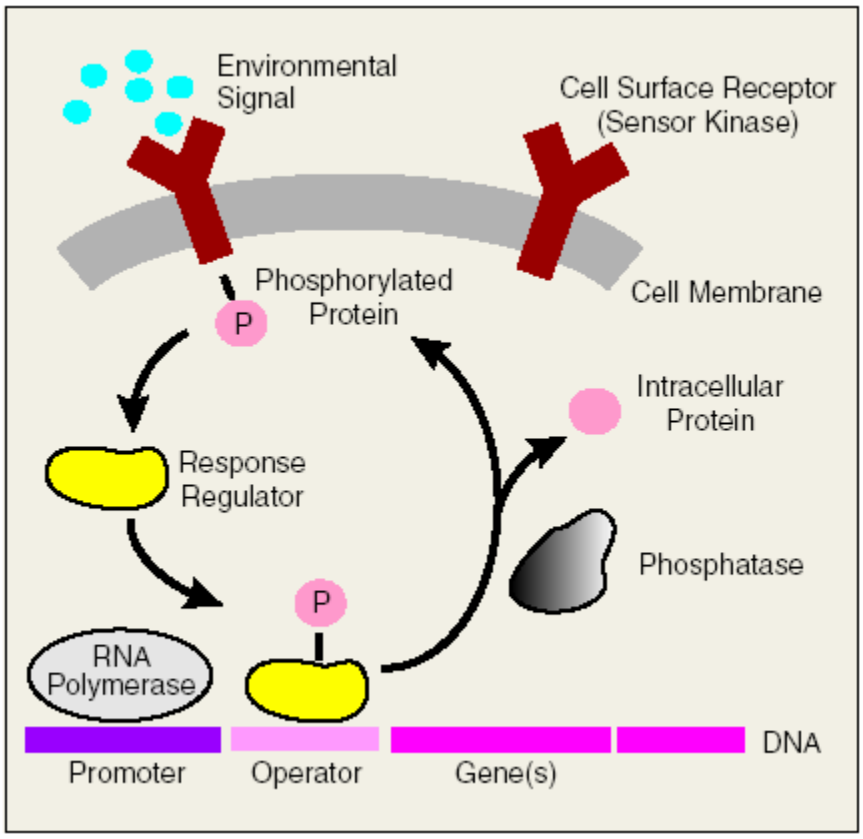


Figure 2.7 Cell signaling (signal transduction).

Intracellular dynamics (gene expression) can be affected by extra-cellular signals. Receptors spanning the cell membrane receive signals and transmit the information to activate intracellular proteins. In this figure, the response regulator binds to the operator region of a gene and prevents the RNA polymerase from transcription of the adjacent gene. A phosphatase ensures that the process is continuous.

As it is shown in Figure 2.7, gene expression can be affected by environmental conditions. The cells have the appropriate mechanisms which regulate a specific target. Receptors, which are proteins, span the cell membrane.

Then another protein, the response regulator, is activated by phosphorylation.
Generally a pathway consists of many more intermediate steps until the signal transduction come to an end.

Epidermal Growth Factor Receptor (EGFR)

EGFR belongs to RTK family and has important role in the cell. EGF is a polypeptide consisting of fifty three amino acids. When EGFR is activated, then cell functions such as cell migration and cell division may happen. EGFR is unbreakably connected to the creation of cancer tumors, that's why many cancer medicines are straight directed to EGF signaling pathway. Furthermore, excessive activation of EGFR on the cancer cell surface is now known to be associated with advanced disease, the development of a metastatic phenotype and a poor prognosis in cancer patients.

EGFR activates the MAPK cascade pathway after it committed various proteins. This activation needs the commitment of the adapter molecules Grb2, Shc, Sos, Gap to the EGF receptor and especially by means of Shc-independent pathway and Shc-dependent pathway. MAPK takes the signal through the cytoplasm to the nucleus where it triggers specific functions, which drive cells into duplication.

Moreover there are and some other kind of receptors that are called internalized receptors and their operation is not clear. When the receptor internalized, then EGFR can commit the same compounds as the surface receptors do.

Chapter 3

HIGH ORDER RECURRENT NEURAL NETWORKS

A general overview in artificial neural networks

Man has two principal objectives in the scientific study of his environment:
He wants to understand and to control. The two goals reinforce each other, since deeper understanding permits firmer control, and, on the other hand, systematic application of scientific theories inevitably generates new problems which require further investigation, and so on.

It might be assumed that a fine-grained descriptive theory of terrestrial phenomena would be required before an adequate theory of control be constructed.
In actuality this is not the case, and indeed, circumstances themselves force us into situations where we must exert regulatory and corrective influences without complete knowledge of basic causes and effects. In connection with the design of experiments, space travel, economics and the study of cancer, scientists encountered processes which are not fully understood. Yet design and control decisions are required.

It is easy to see that in treatment of complex processes, attempts at complete understanding at a basic level may consume so much time and so large a quantity of resources as to impede us in more immediate goals of control.

Artificial Neural Networks have been studied for many years with the hope of achieving human like performance in solving certain problems in speech and image processing. There has been a recent resurgence in the field of neural networks owing to the introduction of new network topologies and training algorithms. The potential benefits of neural networks such as parallel distributed processing, high computation rates, fault tolerance and adaptive capability, have lured

researchers of other fields such us controls, robotics etc. to seek solutions to their complicated problems.
The type of neural network most commonly used in control systems is the *feedforward multilayer neural network*, where no information is fed back during operation. There is, however, feedback information available during training. Typically, supervised learning methods, where the neural network is trained to learn input-output patterns presented to it, are used. Most often, versions of back-propagation (BP) algorithm are used to adjust neural network weights during training. This is generally a slow and very time consuming process, because the algorithm usually takes a long time to converge. However, other optimization methods such us conjugate directions and quasi-Newton have also been implemented [7]. Most often, the individual neuron-activation functions are sigmoid, but also signum or radial basis Gaussian functions are used.

Theoretical studies by several research groups [8], [9], [10], [11], demonstrate that multilayer neural networks with just one hidden layer can approximate any continuous function uniformly over a compact domain, by simply adjusting the synaptic weights, such that a functional of the error between the neural network output and the output of the unknown map, is minimized.

The procedure of training a neural network to represent the forward dynamics of a plant is called *forward modeling*. The neural network model is placed in parallel with the plant and the error between the prediction error- is used as the network training signal. The plant can be single-input single-output or multi-input multi-output, continuous or discrete, linear or non linear.

For the neural network training discrete samples of the plant are often used.
Assume that the plant is described by the nonlinear difference equation:

$$y^p(k+1) = f(y^p(k),...,y^p(k-n+1);u(k),...,u(k-m+1)$$

The system output y^p at time $k+1$ depends on the past n outputs values and the past m values of the input u. An obvious approach for system modeling is to choose the input-output structure of the neural network to be the same as that of the system.

Denoting the output of the network as that of the system. Denoting the output of the network as y^m, we have:

$$y^m(k+1) = f_{apr}(y^p(k),...,y^p(k-n+1);u(k),...,u(k-m+1)).$$

Here, f_{apr} represents the nonlinear input output map of the network, that is, the approximation of *f*. It is clear that the input to the network includes the past values of the real system output, hence, the system has no feedback.
Assuming that after a certain training period the network gives a good representation of the plant that is $y^m \approx y^p$ the training stops and the network output together with its delay values can be fed beck and used as part of the network input. In this way, the network can be used independently of the plant. Such a network model is described by

$$y^m(k+1) = f_{apr}(y^m(k),...,y^m(k-n+1);u(k),...,u(k-m+1)).$$

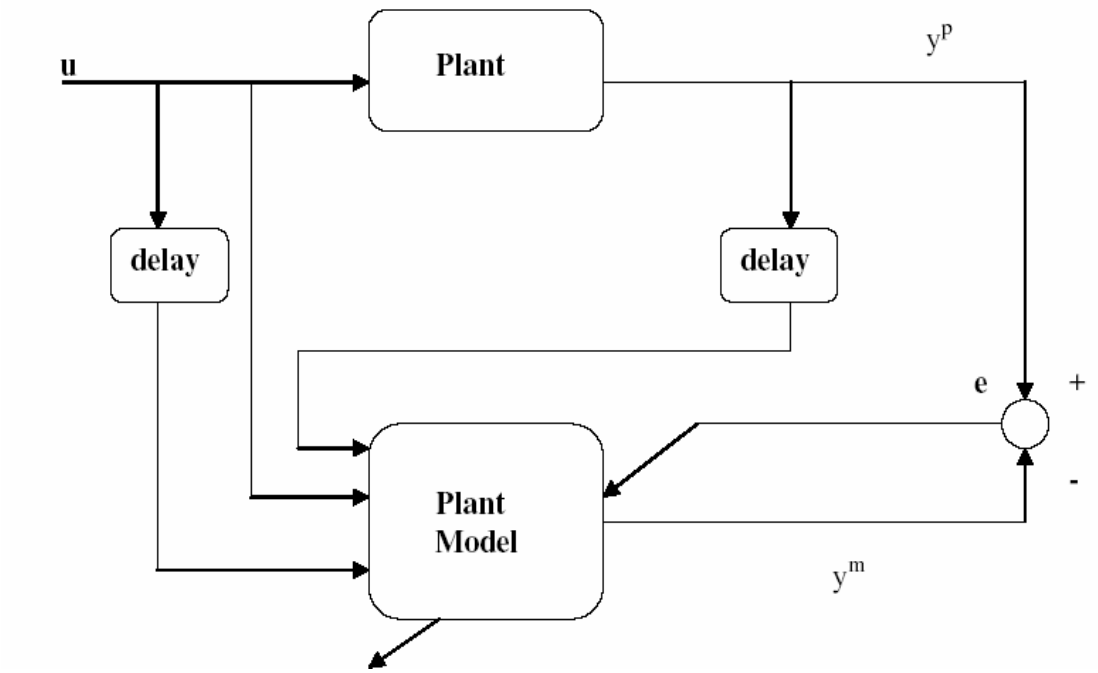


Figure 3.1 Plant Identification with a multi-layer Neural Network

Suppose now that the information about the plant is in the form of an input-output table, which makes the problem for identification look like a typical pattern recognition problem: then, for the training of the

plant model the current and previous inputs to the plant, as well as the previous outputs of the plant should be used again. Other possibilities for the training include the plant states and derivatives of the plant states.
For this reason, if a feedforward multilayer neural network is used and the training is done with BP algorithm, a discrete or discretized continuous plant has to be considered since discrete outputs of the plant model is needed.

This can be illustrated in the Figure 3.1 the arrow that passes through the neural model is indicative of the fact that the output error is used to train the neural network. The number of delays of previous inputs and outputs is unknown: since we have no information about the structure of the plant this number has to be determined experimentally. As far as the training signal is concerned, it has been suggested, [12],[13] that a random signal uniformly distributed over certain ranges should be used.

Instead of training a neural network to identify the forward dynamics of the plant, a neural network can be trained to identify the inverse dynamics of the plant. The neural network's input is the plant's output, and the desired neural network output is the plant's output. The error difference between the actual input of the plant and the output of the neural network is to be minimized and can be used to train the neural network. The desired output of the neural network is the current input to the plant. When modeling the inverse dynamics of the plant with a neural network, the assumption is being made, either implicitly or explicitly, that the neural network can approximate the inverse of the plant well. This, of course, means that the inverse exists and it is unique; if not unique then care should be taken with the ranges of the inputs to the network. It also means that the inverse is stable.

We want a neural network architecture be able to approximate the behavior of a dynamical system in some sense, it is clear that it should contain some form of dynamics, or said with other words, feedback connections.
In the neural network literature, such networks are known as recurrent.

They were originally designed for pattern recognition applications. A static neural network can also be made a dynamic one, by simply connecting the past neural outputs as inputs to the neural network, thus making the neural network a very complicated and highly

nonlinear dynamical system. A more efficient way is to introduce dynamics with the aid of feedforward multilayer neural networks was proposed in [13]. They connect stable linear dynamical systems with static multilayer networks. The connections need not be only serial: parallel, and feedback connections and combinations of the three types are also permitted. Similar to the static multilayer networks, the synaptic weights are adjusted according to a gradient descent rule.

The main problem with the dynamic neural networks that are based on static multilayer networks is that the synaptic weights appear nonlinearly in the mathematical representation that governs their evolution. This leads to a number of significant drawbacks. First, the learning laws that are used, require a high amount of computational time. Second, since the synaptic weights are adjusted to minimize a functional of the approximation error and the weights appear nonlinearly, the functional possesses many local minimum so there is no way to ensure the convergence of the weights to the global minimum. Moreover, due to the highly nonlinear nature of the neural network architecture, basic properties like stability, convergence and robustness, are very difficult to verify. On the other hand the recurrent neural networks possessing a linear-in-the weights property, make the issues of proving stability and convergence feasible and their incorporation into a control loop promising.

The RHONN Model

Recurrent neutral network (RNN) models are characterized by the two way connectivity between units (in our example neurons). This distinguishes them from feed forward neutral networks, where the output of one unit is connected only to units of the next layer.

In the simplest case, the state history of each neuron is governed by a differential equation of the form:

$$\dot{x_i} = -a_i x_i + b_i \sum_j \omega_{ij} y_j,$$

Where x_i is the state of the $i-\text{th neuron}$, a_i, b_i are constants, ω_{ij} is the synaptic weight connecting the $j-th$ input to the $i-th$ neuron and y_j is the $j-th$ input to the above neuron. Each y_j is either an external input or the state of the neuron passed through a sigmoid function (i.e., $y_j = s(x_j)$), where $s(.)$ is the sigmoid nonlinearity.

The dynamic behavior and the stability properties of neural network models of the form $\dot{x_i} = -a_i x_i + b_i \sum_j \omega_{ij} y_j,$ have been studied extensively by various researchers. These studies exhibited encouraging results in application areas such as associative memories, but they also revealed the limitations inherent in such a simple model.

In a recurrent second order neutral network, the input to the neuron is not only a linear combination of the components y_j, but also of their product $y_j y_k$.

Consider a Recurrent High Order Neural Network (RHONN) consisting of n neurons and m inputs. The state of each neuron is followed by a differential equation of the form:

$$\dot{x_i} = -a_i x_i + b_i [\sum_{k=1}^{L} \omega_{ik} \prod_{j \in I_k} y_j^{d_j(k)}],$$

where $\{I_1, I_2, ..., I_L\}$ is a collection of L not-ordered subsets of $\{1, 2, ..., m+n\}$, a_i, b_i are real coefficients, ω_{ik} are the (adjustable) synaptic weights of the neural network and $d_j(k)$ are non-negative integers.

The state of the $i-th$ neuron is again represented by x_i and $y=[y_1, y_2, ..., y_{m+n}]^T$ is the input vector to each neuron defined by:

$$y=[y_1\ y_2\ \ldots y_n\ y_{n+1}\ \ldots y_{n+m}]^T =$$
$$=[s(x_1)\ \mathrm{s}(\mathrm{x}_2)\ \ldots s(x_n)\ u_1\ u_2\ \ldots u_m]^T$$

$$y=\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \\ . \\ . \\ . \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} s(x_1) \\ s(x_2) \\ . \\ . \\ s(x_n) \\ u_1 \\ u_2 \\ . \\ . \\ u_m \end{bmatrix}$$

where $u=[u_1, u_2, ..., u_m]^T$ is the external input vector to the network. The function $s(.)$ is monotone-increasing, differentiable and is represented by sigmoids of the form:

$$s(x)=\frac{a}{1+e^{-\beta x}}-\gamma,$$

where the parameters a, β represent the bound and slope of sigmoid's curvature and γ is a bias constant.

In the case where $a=\beta=1,\ \gamma=0$ we take the logistic function.

If $a=\beta=2,\ \gamma=1$, we take the hyperbolic tangent function.
These two sigmoid activation functions are commonly used in neural network applications.

The L-dimensional vector z is:

$$z = [z_1\ z_2\ \ldots z_L]^T =$$
$$= [\prod_{j \in I_1} y_j^{d_j(1)}\ \prod_{j \in I_2} y_j^{d_j(2)}\ \ldots \prod_{j \in I_L} y_j^{d_j(L)}]$$

$$z = \begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ . \\ z_L \end{bmatrix} = \begin{bmatrix} \prod_{j \in I1} y_j^{d_j(1)} \\ \prod_{j \in I2} y_j^{d_j(2)} \\ . \\ . \\ . \\ \prod_{j \in IL} y_j^{d_j(L)} \end{bmatrix}$$

After this the RHONN model

$$\dot{x_i} = -a_i x_i + b_i [\sum_{k=1}^{L} \omega_{ik} \prod_{j \in I_k} y_j^{d_j(k)}],$$

becomes

$$\dot{x_i} = -a_i x_i + b_i [\sum_{k=1}^{L} \omega_{ik} z_k]$$

Let us consider the adjustable parameter vector as

$$\omega_i = b_i [\omega_{i1}\ \omega_{i2}\ \ldots\ \omega_{iL}]^T,$$

after this the above equation will take the form:

$$\dot{x_i} = -a_i x_i + \omega_i^T z \quad (1)$$

The vectors $\{\omega_i : i = 1,2,...,n\}$ represent the adjustable weights of the network, while the coefficients $\{a_i : i = 1,2,...,n\}$ are part of the underlying network architecture and are fixed during training.

The dynamic behavior of the overall network is described by expressing the equation (1) as a vector notation:

$$\dot{x} = A \cdot x + W^T z$$

where $x = [x_1, x_2, ..., x_n]^T \in R^n,\ W = [\omega_1, \omega_2, ..., \omega_n]^T \in R^{L \times n}$
and $A = diag\{-a_1, -a_2, ..., -a_n\}$ is a $n \times n$ matrix.

Since $a_i > 0,\ \forall\ i = 1,2,...,n$ A is a stability matrix.
The vector z is a function of both the neural network state x and the external input u.

Approximation Properties

Consider now the problem of approximating a general nonlinear dynamical system whose input-output behavior is given by

$$\dot{x} = F(\chi, u) \quad (2)$$

where $\chi \in R^n$ is the system state, $u \in R^m$ is the system input and $F: R^{n+m} \to R^n$ is a smooth vector field defined on a compact set $Y \subset R^{n+m}$

The approximation problem consists of determining whether by allowing enough higher-order connections, there exist weights W, such that

the RHONN model $\dot{x} = A \cdot x + W^T z$
approximates the input-output behavior of an arbitrary dynamical system of the form

$$\dot{x} = F(x, u).$$

In order to have a well-posed problem, we assume that F is continuous
and satisfies a local Lipschitz condition such that (2) has a unique solution
Based on the above assumptions we obtain the following result.

Theorem 2.1.1
Suppose that the system (2) and the model (1) are initially at the same state
$x(0) = \chi(0);$ then for any $\varepsilon > 0$ and any finite T>0, there exists an integer L and a matrix $W^* \in R^{L \times n}$ such that the state $x(t)$ of the RHONN model

$\dot{x}_i = -a_i x_i + \omega_i^T z$ with L high-order connections and weight values $W = W^*$ satisfies

$$\sup_{0 \le t \le T} | x(t) - \chi(t) | \le \varepsilon.$$

Proof:

From (1), the dynamic behavior of the RHONN model is described by

$$\dot{x} = A \cdot x + W^T z(x,u)$$

Adding and subtracting $A\chi$, the equation (2) rewritten as

$$\dot{x} = Ax + G(\chi,u),$$
$$where\ G(\chi,u) = F(\chi,u) - A\chi.$$

Since $x(0) = \chi(0)$, the state error $e = x - \chi$ satisfies the differential equation

$$\dot{e} = Ae + W^T z(x,u) - G(\chi,u),\ \ e(0) = 0. \quad (3)$$

By assumption, $(\chi(t),u(t)) \in Y$ for all $t \in [0,T]$, where Y is a compact subset of R^{n+m}.

Let $$Y_e = \{(\chi,u) \in R^{n+m} : |(\chi,u) - (\chi_y,u_y)| \le \varepsilon,\ (\chi_y,u_y) \in Y\}$$

It can be seen easily that Y_e is also a compact subset of R^{n+m} and $Y \subset Y_e$ In simple words Y_e is larger than Y, where ε is the required degree of approximation. Since z is a continuous function, it satisfies a
Lipschitz condition in Y_e

$$| z(x_1,u) - z(x_2,u) | \le l\, | x_1 - x_2 |$$

The function $W^T z(x,u)$ satisfies the conditions of the Stone-Weierstrass Theorem and can approximate any continuous function over a compact domain, therefore.

From the equations $\dot{x}_i = -a_i x_i + b_i[\sum_{k=1}^{L} \omega_{ik} \prod_{j \in I_k} y_j^{\,d_j(k)}]$, and

$$y = [y_1\ y_2\ \ldots y_n\ y_{n+1}\ \ldots y_{n+m}]^T =$$
$$= [s(x_1)\ \mathrm{s}(\mathrm{x}_2)\ \ldots s(x_n)\ u_1\ u_2\ \ldots u_m]^T$$

It is clear that $z(x,u)$ is in the standard polynomial expansion with the exception that each component of the vector x is preprocessed by a sigmoid function s(.).
The preprocessing of input via a continuous invertible function does not affect the ability of a network to approximate continuous functions: therefore, it can be shown that if L is sufficient large, then there exist weight values $W = W^*$ such that $W^{*T} z(x,u)$ can approximate *G*(*x, u*) to any degree of accuracy, for all (*x, u*) in a compact domain.

There exists $W = W^*$ such that

$$\sup_{(\chi,u)\in Y_e} |W^{*T} z(x,u) - G(\chi,u)| \le \delta \quad (4)$$

where δ is a constant to be designed in the sequel.

The solution of (3) is

$$\begin{aligned} e(t) &= \int_0^t e^{A(t-\tau)}[W^{*T} \cdot z(x(\tau),u(\tau)) - G(x(\tau),u(\tau))]d\tau \\ &= \int_0^t e^{A(t-\tau)}[W^{*T} \cdot z(x(\tau),u(\tau)) - W^{*T} z(\chi(\tau),u(\tau))]d\tau + \\ &= +\int_0^t e^{A(t-\tau)}[W^{*T} \cdot z(\chi(\tau),u(\tau)) - G(\chi(\tau),u(\tau))]d\tau. \end{aligned} \quad (5)$$

Since A is a diagonal stability matrix, there exists a positive constant a such that
$\| e^{At} \| \le e^{-at}$ for all $t \ge 0$. Furthermore, let $L = l \cdot \| W^* \|$.
If we combine this with the definitions of the constants a, L, ε we choose δ in equation (4) as

$$\delta = \frac{\varepsilon a}{2} e^{-\frac{L}{a}} > 0$$

a) Consider the case where $(x(t),u(t)) \in Y_e$ for all $t \in [0,T]$. If we go to the equation (5) and taking norms on both sides and use the definition of δ, the following inequalities hold for all $t \in [0,T]$:

$$
\begin{aligned}
|e(t)| &\le \int_0^t \| e^{A(t-\tau)} \| \cdot | W^{*T} \cdot z(x(\tau),u(\tau)) - W^{*T} z(\chi(\tau),u(\tau)) | \, d\tau + \\
&+ \int_0^t \| e^{A(t-\tau)} \| \cdot | W^{*T} \cdot z(\chi(\tau),u(\tau)) - G(\chi(\tau),u(\tau)) | \, d\tau, \\
&\le \int_0^t e^{-a(t-\tau)} L \, |e(\tau)| \, d\tau + \int_0^t \delta e^{-a(t-\tau)} d\tau, \\
&\le \frac{\varepsilon}{2} e^{-\frac{L}{a}} + L \int_0^t e^{-a(t-\tau)} \, |e(\tau)| \, d\tau.
\end{aligned}
$$

Using the Bellman-Gronwall Lemma [34], we take

$$
\begin{aligned}
|e(t)| &\le \frac{\varepsilon}{2} e^{-\frac{L}{a}} + e^{L} \cdot \int_0^t e^{-a(t-\tau)} d\tau, \\
&\le \frac{\varepsilon}{2}.
\end{aligned}
\qquad (6)
$$

b) Now consider the case where $(x(t),u(t)) \notin Y_e$ for all $t \in [0,T]$.
By the continuity of x(t) ,there exist a T^*, where
$0 < T^* < T,$ such that $(x(T^*),u(T^*)) \in \partial Y_e$, ∂Y_e denotes the boundary of Y_e
If we make the same analysis for $t \in [0,T^*]$ we find that in this interval
$|x(t) - \chi(t)| \le \frac{\varepsilon}{2}$ that it is clearly a contradiction.

After this the inequality (6) holds for all $t \in [0,T]$.

The aforementioned theorem proves that if sufficiently large number of connections is allowed in the RHONN model then it is possible to approximate any dynamical system to any degree of accuracy. This is strictly an existence result: it does not provide any constructive method for obtaining the optimal weights W^*. In what follows we consider the learning problem of adjusting the weights adaptively, such that the RHONN model identifies
general dynamic systems.

Chapter 4

Tracking Problems

The definition of the word **tracking** is the action of forcing the state of the actual system to follow the output of a given stable-dynamical system. In order to achieve this purpose the recurrent high-order-neural-network-based adaptive control algorithm is extended to cover this category of tracking problems.

Complete Matching Case

In our case we investigate the adaptive model reference control problem when the modeling error term is zero and we have the complete model matching.

Under this hypothesis, the unknown system can be written as

$$\dot{x} = -Ax + W^{*}S(x) + W_1^{*}S'(x)u$$

In addition to, we want the unknown system states to follow the states of a stable linear model like as

$$\dot{x}_m = -A_m x_m + B_m u_m$$

From the two previous equations we take the error equation

$$\dot{e} = -Ax + W^{*}S(x) + W_1^{*}S'(x)u + A_m x_m - B_m u_m$$

Where we have defined

$$\overset{\Delta}{e} = x - x_m$$

In the nest step we add and subtract the term Ax_m, the previous equation becomes

$$\dot{e} = -Ax + W^* S(x) + W_1^* S'(x)u + A_m x_m - B_m u_m + Ax_m - Ax_m =$$

$$= -Ae + W^* S(x) + W_1^* S'(x)u + \tilde{A} x_m - B_m u_m$$

Where

$$\tilde{A} = A_m - A$$

We take a function h(e) from R^n to R^+ of class C^2, whose the derivative with respect to time is the following equation

$$\dot{h} = \frac{\partial h}{\partial e}^T \cdot [-Ae + W^* S(x) + W_1^* S'(x)u + A_m x_m - B_m u_m]$$

If we use the W^* and W_1^* the above linear equation can be written as

$$\dot{h} + \frac{\partial h}{\partial e}^T Ae - \frac{\partial h}{\partial e}^T \tilde{A} x_m + \frac{\partial h}{\partial e}^T B_m u_m = \frac{\partial h}{\partial e}^T W^* S(x) + \frac{\partial h}{\partial e}^T W_1^* S'(x)u$$

Define

$$v \overset{\Delta}{=} \frac{\partial h}{\partial x} W S(x) + \frac{\partial h}{\partial x} W_1 S'(x)u - \dot{h} - \frac{\partial h}{\partial x} Ax + \frac{\partial h}{\partial e}^T \tilde{A} x_m - \frac{\partial h}{\partial e}^T B_m u_m$$

Where W and W_1 are estimates of W^* and W_1^* respectively.

The v signal cannot be measured since $\dot{h}$ is unknown.

To deal with this problem, we use the error filtering method, according to which

$$\dot{\xi} + k\xi = v \ ,$$

$$= \frac{\partial h}{\partial e}^T WS(x) + \frac{\partial h}{\partial e}^T W_1 S'(x)u - \dot{h} - \frac{\partial h}{\partial e}^T Ae + \frac{\partial h}{\partial e}^T \tilde{A} x_m - \frac{\partial h}{\partial e}^T B_m u_m$$
$$= -\dot{h} + \frac{\partial h}{\partial e}^T [-Ae + WS(x) + W_1 S'(x)u] + \frac{\partial h}{\partial e}^T [\tilde{A} x_m - B_m u_m] \quad (1)$$

With k a strictly positive constant. To implement the previous equation, we take

$$\dot{\xi} \stackrel{\Delta}{=} \zeta - h \quad (2)$$

If we combine the relations (1) and (2) we take the following result

$$\dot{\zeta} + \kappa\zeta = \kappa h - e^T Ae + e^T WS(x) + e^T W_1 S'(x)u + e^T \tilde{A} x_m - e^T B_m u_m$$
$$h(e) = \frac{1}{2}|e|^2$$
$$L = \frac{1}{2}\xi^2 + \frac{1}{2}tr\{\tilde{W}^T \tilde{W}\} + \frac{1}{2}tr\{\tilde{W}_1^T \tilde{W}_1\} \quad (3)$$
$$\tilde{W} = W - W^*$$
$$\tilde{W}_1 = W_1 - W_1^*$$

with the state $\zeta \in \mathbb{R}$.

We choose h (e) to be $h(e) = \frac{1}{2}|e|^2$ the relation (3) will take the form

$$\dot{\zeta} + \kappa\zeta = \kappa h - e^T Ae + e^T WS(x) + e^T W_1 S'(x)u + e^T \tilde{A} x_m - e^T B_m u_m$$

If we consider the Lyapunov-like function

$$L = \frac{1}{2}\xi^2 + \frac{1}{2}tr\{\tilde{W}^T \tilde{W}\} + \frac{1}{2}tr\{\tilde{W}_1^T \tilde{W}_1\} \quad (4)$$

where

$$\tilde{W} = W - W^*$$

$$\tilde{W}_1 = W_1 - W_1^*$$

If we take the derivative of the relation (4) with respect to time we obtain

$$\dot{L} = \xi\dot{\xi} + tr\{\dot{W}^T\tilde{W}\} + tr\{\dot{W}_1^T\tilde{W}_1\}$$

Continuing

$$\dot{L} = -k\xi^2 + \xi[-x^T W^* S(x) - x^T W_1^* S'(x)u + x^T W S(x)] + \xi[x^T W_1 S'(x)u] + tr\{\dot{W}^T\tilde{W}\} + tr\{\dot{W}_1^T\tilde{W}_1\}$$

or equivalently

$$\dot{L} = -k\xi^2 + \xi x^T\tilde{W}S(x) + x^T\tilde{W}_1 S'(x)u] + tr\{\dot{W}^T\tilde{W}\} + tr\{\dot{W}_1^T\tilde{W}_1\}$$

If we choose

$$\begin{aligned} tr\{\dot{W}^T\tilde{W}\} &= -\xi x^T\tilde{W}S(x) \\ tr\{\dot{W}_1^T\tilde{W}_1\} &= -\xi x^T\tilde{W}_1 S'(x)u \end{aligned} \qquad (5,\ 6)$$

$\dot{L}$ becomes

$$\dot{L} = -k\xi^2 \le 0$$

We can write the relations (5), (6) in terms of elements

$$\dot{\omega}_{ij} = -\xi x_i s_j(x)$$

$$\dot{\omega}_{i1} = -\xi x_i s_i'(x)u_i$$

for all $i, j = 1, 2, ..., n$ and in matrix form as

$$\dot{W} = -\xi x S^T(x)$$

$$\dot{W}_1 = -\xi x' S'(x)U$$

where

$$x' = diag[x_1, x_2, ..., x_n]$$
$$U = diag[u_1, u_2, ..., u_n]$$

One important Lemma for continuing our analysis is the following:

Lemma 4.4.1

Consider the system

$$\dot{x} = -Ax + W^* S(x) + W_1^* S'(x)u$$

$$\dot{x}_m = -A_m x_m + B_m u_m$$

$$\dot{\zeta} = -k\zeta + \kappa h - e^T Ae + e^T WS(x) + e^T W_1 S'(x)u + e^T \tilde{A} x_m - e^T B_m u_m$$
$$\xi = \zeta - h$$
$$h = \frac{1}{2}|e|^2$$
$$e = x - x_m$$

The update laws

$$\dot{W} = -\xi x S^T(x)$$

$$\dot{W}_1 = -\xi x' S'(x)U$$

with $U = diag[u_1, u_2, ..., u_n]$, guarantee the following properties

- $\xi, |e|, W, W_1, \zeta \in L_\infty$
- $|\xi| \in \mathrm{L}_2$
- $\lim_{t\to\infty} \xi(t) = 0, \ \lim_{t\to\infty} \dot{W}(t) = 0, \ \lim_{t\to\infty} \dot{W}_1(t) = 0$

provided that $u \in L_\infty$

If we have understand the lemma 4.4.1 we have the opportunity to take one more step with the theorem 4.4.1:

Theorem 4.4.1

The closed-loop system

$$\dot{x} = -Ax + W^{*} S(x) + W_1^{*} S'(x)u$$

$$\dot{x}_m = -A_m x_m + B_m u_m$$

$$\dot{\zeta} = -k\zeta + \kappa h - e^T Ae + e^T WS(x) + e^T W_1 S'(x)u + e^T \tilde{A} x_m - e^T B_m u_m$$

$$u = -[W_1 S'(x)]^{-1}[WS(x) + \tilde{A} x_m - B_m u_m]$$

$$\xi = \zeta - h$$

$$h = \frac{1}{2} | e |^2$$

$$k = 1$$

together with the update laws

$$\dot{\omega}_{ij} = -\xi e_i s_j(x)$$

$$\dot{\omega}_{i1} = \begin{cases} -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}') = \varepsilon \\ & \text{and } \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) \le 0 \\ 0 & \text{if } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}^*) = \varepsilon \text{ and} \\ & \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) > 0 \\ -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}') = \omega^m \\ & \text{and } \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) \ge 0 \\ 0 & \text{if } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}^*) = \omega^m \text{ and} \\ & \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) < 0 \end{cases}$$

for all $i, j = 1, 2, ..., n$ guarantees that

$\lim_{t\to\infty} |e(t)| = 0$

Direct adaptive tracking (n=m) Complete matching

Actual System: $\dot{x} = f(x) + G(x)u,\ x,u \in \mathbb{R}^n$

Reference System: $\dot{x}_m = -A_m x_m + B_m u_m,\ x_m \in \mathbb{R}^n,\ u_m \in \mathbb{R}^n$

Model: $\dot{x} = -Ax + W^* S(x) + W_1^* S'(x)u$

Tracking error: $e = x - x_m$

Control Law: $u = -[W_1 S'(x)]^{-1}[WS(x) + \tilde{A} x_m - e^T B_m u_m]$

Update Laws: $\dot{\omega}_{ij} = -\xi e_i s_j(x)$

$$\dot{\omega}_{i1} = \begin{cases} -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}') = \varepsilon \\ & \text{and } \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) \le 0 \\ 0 & \text{if } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}^*) = \varepsilon \text{ and} \\ & \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) > 0 \end{cases}$$

$$\dot{\omega}_{i1} = \begin{cases} -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}') = \omega^m \\ & \text{and } \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) \ge 0 \\ 0 & \text{if } \omega_{i1}\,\mathrm{sgn}(\omega_{i1}^*) = \omega^m \text{ and} \\ & \xi e_i s_i'(x) u_i\,\mathrm{sgn}(\omega_{i1}^*) < 0 \end{cases}$$

Filter: $\dot{\zeta} = -k\zeta + \kappa h - e^T A e + e^T W S(x) + e^T W_1 S'(x)u + e^T \tilde{A} x_m - e^T B_m u_m$

$$\tilde{A}_m = A_m - A$$

$$\xi = \zeta - h,\ k = 1$$

$$h = \frac{1}{2}|e|^2$$

Properties: $u, \xi, |e|, \zeta, W \in L_\infty,\ W_1 \in W'$

$$|\xi| \in L_2,\ \lim_{t\to\infty} \dot{W}(t) = 0$$

$$\lim_{t\to\infty} \xi(t) = 0,\ \lim_{t\to\infty} |e(t)| = 0$$

Requirements: $W_1^* \in W'$, $W_1(0) \in W'$

$S(x)$ is an L-dimensional vector with elements,
$s_i(x),\ i = 1,2,...,L$ of the form

$$s_i(x) = \prod_{j \in I_i} [s(x_j)]^{d_j(i)}$$

where $I_i,\ i = 1,2,...,L$ are a collection of L not-ordered subsets of {1,2,…,n} and
d_j are positive integers.

$S'(x)$ is a $L_0 \times m$ matrix with elements $s'_{lk}(x)$ of the form

$$s'_{lk}(x) = \prod_{j \in I_{lk}} [s(x_j)]^{d_j(l,k)}$$

for all $l = 1,2,...,L_0$ and $k = 1,2,...,m$ where I_{lk} are collections of mL_0 not-ordered subsets of
[1,2,…,n} and $d_j(l,k)$ are positive integers.

The $s(x_j)$ is a smooth, monotone increasing function, which is usually represented by sigmoidals of the form

$$s(x_j) = \frac{\mu}{1 + e^{-l_o x_i}} + \lambda$$

for all $j = 1,2,...,n$ with the parameters μ,l_0 to represent
the bound and the slope of sigmoid's curvature and λ a bias constant.

Adaptive Tracking Examples

Example Scalar case

Consider the following first order plant:

$$\dot{x} = ax + bu$$

where a and b are unknown parameters but the sign of b is known. The control objective is to choose an appropriate control law such that all signals in a closed-loop plant are bounded and x tracks the state of x_m of the reference model given by

$$\dot{x}_m = -a_m x_m + b_m r$$

for any bounded piecewise continuous signal *r*(*t*), where *am* > 0, *bm* are
known and *xm*(*t*)*; r*(*t*) are measured at each time *t*. It is assumed that *am; bm* and *r* are chosen so that *xm* represents the desired state response of
the plant.

Control Law

For *x* to track *xm* for any reference input signal *r*(*t*), the
control law should be chosen so that the closed-loop plant transfer function
from the input *r* to output *x* is equal to that of the reference model. We propose the control law

$u = -k^* x + l^* r$
where k^*, l^* are calculated so that

$$\frac{x(s)}{r(s)} = \frac{bl^*}{s - a + bk^*} = \frac{b_m}{s + a_m} = \frac{x_m(s)}{r(s)}$$

Of course,

$$l^* = \frac{b_m}{b},\ k^* = \frac{a_m + a}{b}$$

provided of course that $b \neq 0$ the plant $\dot{x} = ax + bu$ is controllable. The control law guarantees that the transfer function of the closed-loop plant is equal to that of the reference model.

Vector Case: Full Statement Measurement

Consider the n-order plant:

$$\dot{x} = Ax + bu,\ x \in R^n$$

where $A \in R^{n \times n}, B \in R^{n \times q}$ are unknown constant matrices and (*A;B*) is controllable. The control objective is to choose the input vector $u \in R^n$ such that all signals in the closed-loop plant are bounded and the plant state *x*
follows the state $x_m \in R^n$ of a reference model specified by the LTI system

$$\dot{x}_m = A_m x_m + B_m r$$

where $A_m \in R^{n \times n}$ is a stable matrix, $B_m \in R^{n \times q}$ and $r \in R^q$ is a bounded reference input vector. The reference model and input *r* are chosen so that $x_m(t)$ represents a desired trajectory that *x* has to follow.

Control Law

If the matrices *A, B* were known, we could apply the control law

$$u = -K^* x + L^* r$$

and obtain the closed-loop plant

$$\dot{x} = (A - BK^*)x + BL^* r$$

If $K^* \in R^{n\times q}$ and $L^* \in R^{q\times q}$ are chosen to satisfy the algebraic equations

$$A - BK^* = A_m,\ BL^* = B_m$$

then the transfer matrix of the closed-loop plant is the same as that of the reference model and $x(t) \to x_m(t)$ exponentially fast for any bounded reference input signal *r*(*t*).

Adaptive Law

By adding and subtracting the desired input term, namely,

$-B(K^* x + L^* r)$ in the plant equation we take

$$\dot{x} = A_m x + B_m r + B(K^* x - L^* r + u)$$

Following the same procedure as in the scalar example, we can show that the tracking error $e = x - x_m$ and parameter error

$\tilde{K} = K - K^*$ and $\tilde{L} = L - L^*$ satisfy the equation

$$\dot{e} = A_m e + B(-\tilde{K} x + \tilde{L} r)$$

which also depends on the unknown matrix *B*. In the scalar case we manage to get away with the unknown *B* by assuming that its sign is known. An extension of the scalar assumption to the vector case is as follows: Let us assume that L^* is either positive definite or negative definite and $\Gamma^{-1} = L^* \cdot \mathrm{sgn}(l)$

where $l = 1$ if L^* is positive definite and $l = -1$ if L^* is negative definite. Then $B = B_m L^{*-1}$ and the above equation becomes

$$\dot{e} = A_m e + B_m L^{*-1}(-\tilde{K} x + \tilde{L} r)$$

We propose the following Lyapunov function candidate

$$V(e, \tilde{K}, \tilde{L}) = e^T P e + tr\,[\tilde{K}^T \Gamma \tilde{K} + \tilde{L}^T \Gamma \tilde{L}]$$

where $P = P^T > 0$ satisfies the Lyapunov equation

$$PA_m + A_m{}^T P = -Q$$

for some $Q = Q^T > 0$. Then

$$\dot{V} = -e^T Qe + 2e^T PB_m L^{*-1}(-\tilde{K}x + \tilde{L}r) + 2tr\,[\tilde{K}^T \Gamma \dot{\tilde{K}} + \tilde{L}^T \Gamma \dot{\tilde{L}}]$$

Now

$$e^T PB_m L^{*-1} \tilde{K} x = tr\,[x^T \tilde{K}^T \Gamma B_m^T Pe] \cdot \mathrm{sgn}(l) = tr\,[\tilde{K}^T \Gamma B_m^T Pex^T] \mathrm{sgn}(l)$$

and

$$e^T PB_m L^{*-1} \tilde{L} r = tr\,[\tilde{L}^T \Gamma B_m^T Per^T] \cdot \mathrm{sgn}(l)$$

After this we have: $\dot{\tilde{K}} = \dot{K} = B_m^T Pex^T \cdot \mathrm{sgn}(l),\ \dot{\tilde{L}} = \dot{L} = B_m^T Per^T \cdot \mathrm{sgn}(l)$
We have

$$\dot{V} = -e^T Qe$$

Analysis

From the properties of $V, \dot{V}$, we establish as in the scalar case that *K*(*t*), *L*(*t*), *e*(*t*) are bounded and that $e(t) \to 0$ as $t \to \infty$.

Chapter 5

PROBLEM ANALYSIS

Systems theory and cell biology have enjoyed a long relationship that has received renewed interest in recent years in the context of systems biology. The term “systems” in systems biology based on systems theory or dynamic systems theory:

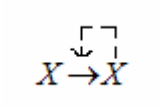
Systems biology is defined through the application of systems and signal-oriented approaches for understanding of inter- and intra-cellular dynamic processes. The aim of the present text is to review the systems and control perspective of dynamic systems. The biologist’s conceptual framework for representing the variables of a biochemical reaction network, and for describing their relationships, and pathway maps. A principal goal of systems biology is to turn these static maps into dynamic models which can provide insight into the temporal evolution of biochemical reaction networks. Towards this end we review the case for differential equation models as a natural representation of casual entailment in pathways. Block diagrams, commonly used in the engineering sciences, are introduced and compared to pathway maps. The stimulus-response representation of a molecular system is a necessary condition for an understanding of dynamic interactions among the components that make up a pathway. Using simple examples, we show biochemical reactions are modeled in the dynamic systems framework and visualized using block-diagrams.

5.1 System Description

Consider a simple monomolecular reaction where chemical species X is transformed. The change in concentration of X at time t depends on the concentration of X at time t in that the rate by which the reaction proceeds is proportional to the concentration at each time instant,

$$\frac{dx(t)}{dt} \propto x(t)$$

with a certain positive rate constant k. A diagrammatic representation of this biochemical process illustrates the fact that chemical species X "feed back" on itself:



A linear mathematical ODE model of the process is given by

$$\frac{d}{dt}x(t) = k \cdot x(t)$$

Here X acts as a substrate being converted and the product. Thee is positive feedback in that the larger the product X, the greater the rate of change by which substrate X is transformed. A simulation of this system reveals the expected unbounded growth in the concentration of X,

$$x(t) = x_0 \cdot e^{kt}$$

where $x_0 = x(t=0)$ denotes the initial condition. With increasing x, the growth rate $\frac{dx}{dt}$ also increases in this system, leading to an unbounded growth.

Continuing, we analyze the autocatalytic reaction

$$X + A \underset{k2}{\overset{k1}{\rightleftarrows}} 2X$$

where for a given X molecule, A facilitates the doubling. A pathway map of this process would be

In pathway maps we use a bar at the end of the arrow to denote an inhibition or negative feedback loop. If A is considered to have a constant concentration, generalizing the law of mass action, we have the differential equation model:

$$\frac{d}{dt}x(t) = k_1 a x(t) - k_2 x^2(t)$$

$$= k_1 a x(t)(1 - \frac{k_2}{a k_1} x(t))$$

In this autocatalytic reaction the 'product' has a strong inhibitory effect on the rate at which X is transformed. For the reason to indicate the internal feedback mechanisms at work in this system we will label the term $(1 - \frac{k_2}{a k_1} x(t))$ as control input variable $u(t)$

$$\frac{d}{dt}x(t) = k_1 a u(t) x(t)$$

If we consider variable x to represent the state of the system, and we write $\frac{dx(t)}{dt} = \dot{x}$ for short, this system becomes at the form of the state-space model, in particular:

$$\dot{x} = f(x) + g(x)u, \quad x(t_0) = x_0$$
$$y = h(x)$$

At this point it very important to point out the state-space model:

The most commonly employed framework to model nonlinear dynamic systems is the state space model

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i,$$
$$y_j = h_j(x) \qquad 1 \le j \le p$$

where $\dot{x}$ is shorthand for the rate of change $\frac{dx(t)}{dt}$ of the n variables summarized in the vector x. At any time t, x, represented though the variables $x_1(t),...,x_n(t)$ defines the state of the system. This system has m inputs and p outputs, the dependence on t is omitted to simplify notation.
A fundamental assumption in above model is that together with some initial condition $x_0 = x(t=0)$ the state completely defines the future behavior of the system.

Control variable u represents some independent stimulus. In cell signaling $u(t)$ would typically model ligands binding to receptors. In many situations we will not be able to observe all state variables directly. Through the response variable y and the mapping h we can capture this situation. The $f, g \text{ and } h$ are mathematical morphisms or mappings, relating the variables on the right-hand side of the equation to rates on the left-hand side. Be careful that $x, u \text{ and } y$ are functions of time, f, g, h do not explicitly depend on time.

This means that we consider only time-invariant systems, dynamic systems where the variables evolve in time but where the system properties remain unchanged.

The state-space model is available to a wide range of systems. Although spatial aspects are not represented explicitly, this would require Partial Differential Equations, for many practical cases it is either possible to assume rapid diffusion and thus ignore it, or different regions of the cell may also be modeled by introducing additional variables to the model. For many intracellular processes it is clear which proteins can be considered “drivers” and which “followers”.
For signal transduction pathways ligands binding to cell surface receptors may be considered the input to the system and gene expression as the output or response to the stimulus $u(t)$. The area of cell signaling [9] is therefore most susceptible to the control perspective on intra-cellular dynamics.

Continuing our analysis:

We can write: $u(x) = \alpha - \beta x$, where the constant α is called the intrinsic growth rate of the population and $\frac{\alpha}{\beta}$ corresponds to the maximum attainable population.
The model we thus obtain is specified by the equation

$$\frac{dx}{dt} = \alpha x(\frac{\alpha/\beta - x}{\alpha/\beta})$$

$$= \alpha x(t) \cdot (1 - \frac{\beta}{\alpha} x(t))$$

This model form is called the logistic growth model and is equivalent to the autocatalytic reaction.
The model describes the real growth rate as a proportion of the intrinsic growth rate. This proportion however decreases with an increased in the population.

For two molecular species we can generalize the control of the system into

$$\dot{x_1} = u_1(x_1, x_2)x_1$$

$$\dot{x_2} = u_2(x_1, x_2)x_2$$

We understand that the vector u is depended on both of the x_1, x_2.

This is equal with the form of matrices:

Actual system:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{pmatrix} x_1 & 0 \\ 0 & x_2 \end{pmatrix} \cdot [u_1 \; u_2]$$

Reference System:

$$\begin{bmatrix} \dot{x}_{m1} \\ \dot{x}_{m2} \end{bmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} u_{m1} \\ u_{m2} \end{bmatrix}$$

Now we give an example if we specify for u_1 and u_2

$$u_1(x_1, x_2) = k_1\alpha - k_2 x_2$$
$$u_2(x_1, x_2) = k_2 x_1 - k_3$$

We take the well known Lotka-Voltera model of two competing populations.
If variables x_1 and x_2 correspond to the chemical species X_1 and X_2, the biochemical representation of this system is

$$X_1 + A \xrightarrow{k_1} 2X_1$$
$$X_1 + X_2 \xrightarrow{k_2} 2X_2$$
$$X_2 \xrightarrow{k_3} B$$

where A is maintained at a constant concentration and B corresponds to the degradation of X_2. The first two reactions are autocatalytic.

The Lotka-Voltera model of competing species give an opportunity to discuss the purpose of mathematical model as a mechanism for illuminating basic principles, while not necessarily describing the details of a particular case.

5.2 Introduction of the method

In this project we implement direct adaptive tracking using RHONN (Recurrent High-Order Neural Networks). RHONNs are used as models of the unknown plant, transforming the original system into a RHONN model which is of known structure, but contains a number of unknown constant-value parameters, known as synaptic weights. RHONNs were introduced in the Chapter 3. As we have already discussed the particular method refers to affine in the control nonlinear dynamical systems possessing unknown non-linearities.

A system affine in the control has the form:

$$\dot{x} = f(x) + G(x) \cdot u$$

We want the unknown system to follow the state of the reference system and in our case is the stable linear model

$$\dot{x}_m = -A_m x_m + B_m u_m$$

in the general form.

To be more specific:

$$\begin{bmatrix} \dot{x}_{m1} \\ \dot{x}_{m2} \end{bmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} u_{m1} \\ u_{m2} \end{bmatrix}$$

Our system is a nonlinear dynamical system and has 2 states according to the state equations.

This system is of the general form:

$$\dot{x} = f(x, u)$$

$$\dot{x}_1 = u_1 \cdot x_1$$

$$\dot{x}_2 = u_2 \cdot x_2$$

where the inputs u1, u2 are depended on x1, x2.

In our case the matrices $f(x)$ and $G(x)$ are:

$$f(x) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and $$G(x) = \begin{pmatrix} x_1 & 0 \\ 0 & x_2 \end{pmatrix}$$

Therefore the true plant can be modeled by the recurrent neural network

$$\dot{x} = -Ax + W^{*} S(x) + W_1^{*} S'(x)u$$

where u is a control input.

The control law is the following:

$$u = -[W_1 S'(x)]^{-1}[WS(x) + \tilde{A} x_m - e^T B_m u_m]$$

There are appropriate update laws of matrices of the weights *W, W1,* which ensure the boundness of the weights estimates and guarantee the stability of our system.

The update laws are the following:

$$\dot{\omega}_{ij} = -\xi e_i s_j(x)$$

and

$$\dot{\omega}_{i1} = \begin{cases} -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1} \operatorname{sgn}(\omega_{i1}') = \varepsilon \\ & \text{and } \xi e_i s_i'(x) u_i \operatorname{sgn}(\omega_{i1}^*) \le 0 \\ 0 & \text{if } \omega_{i1} \operatorname{sgn}(\omega_{i1}^*) = \varepsilon \text{ and} \\ & \xi e_i s_i'(x) u_i \operatorname{sgn}(\omega_{i1}^*) > 0 \\ -\xi e_i s_i'(x) u_i & \text{if } \omega_{i1} \in W' \text{ or } \omega_{i1} \operatorname{sgn}(\omega_{i1}') = \omega^m \\ & \text{and } \xi e_i s_i'(x) u_i \operatorname{sgn}(\omega_{i1}^*) \ge 0 \\ 0 & \text{if } \omega_{i1} \operatorname{sgn}(\omega_{i1}^*) = \omega^m \text{ and} \\ & \xi e_i s_i'(x) u_i \operatorname{sgn}(\omega_{i1}^*) < 0 \end{cases}$$

for all $i, j = 1, 2, ..., n$ guarantees that

$\lim_{t \to \infty} | e(t) | = 0$

where

Filter: $\dot{\zeta} = -k\zeta + \kappa h - e^T A e + e^T W S(x) + e^T W_1 S'(x) u + e^T \tilde{A} x_m - e^T B_m u_m$

$\tilde{A}_m = A_m - A$

$\xi = \zeta - h, \; k = 1$

$h = \frac{1}{2} | e |^2$

Results

We apply the method with the next dimensions of vectors and matrices:

$x \in R^2$

$u \in R^2$

$W \; 2 \times 2$

$W_1 \; 2 \times 2$

$A \; 2 \times 2$

$S(x) \in R^2$

$S'(x) \in R^2$

We choose a relatively small value of the design constant:

$$k = 1$$

We use the control law (5.4) of the method, which gives the dynamic feedback.
Moreover we choose:

$$S(x_i) = S^{'}(x_i)$$

This choice gives a simpler model with simpler adaptive laws and it is crucial for the run-time of the program.

The initial values of the states are

Variable	Initial Value
X1	5 Moles
X2	7 Moles

The initial weights *W,W1* and the matrix *A* are:

$$W_{initial} = \begin{pmatrix} 0.09 & 0.09 \\ 0.9 & 0.9 \end{pmatrix}$$

$$W_{1,initial} = \begin{pmatrix} 0.2 & 0.06 \\ 0.06 & 0.3 \end{pmatrix}$$

$$A = \begin{pmatrix} 1.2 & 0 \\ 0 & 2.6 \end{pmatrix}$$

The parameters of the sigmoid functions were set to:

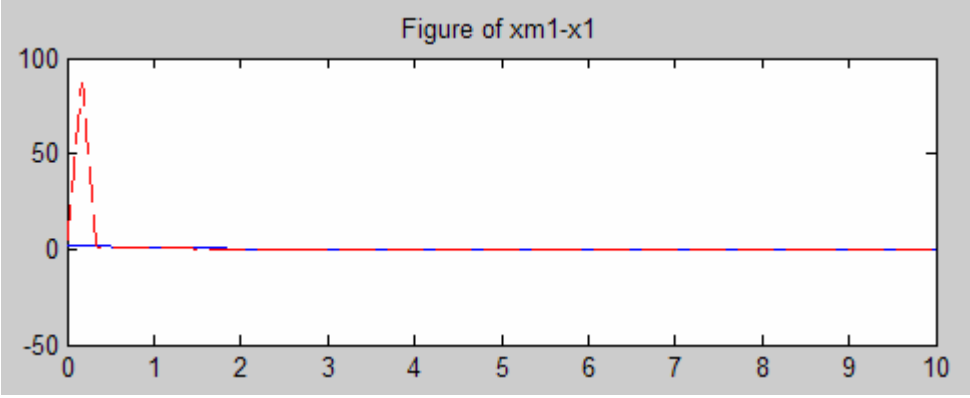
$$\mu = 1$$
$$l_0 = 1$$
$$\lambda = -0.5$$

We have the following simulations for the below cases:

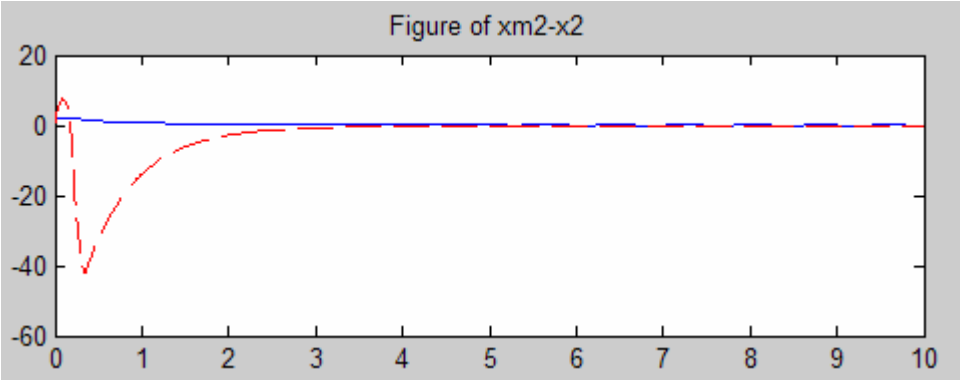
5.3 SIMULATIONS

For $\begin{matrix} u_{m1} = 3\cos(5t)./(5.t+1.) \\ u_{m2} = 5\cos(8t)./(8.t+1.) \end{matrix}$ and initial condition $x_{m0} = [2\ 2]$

We take the following results:



(a)



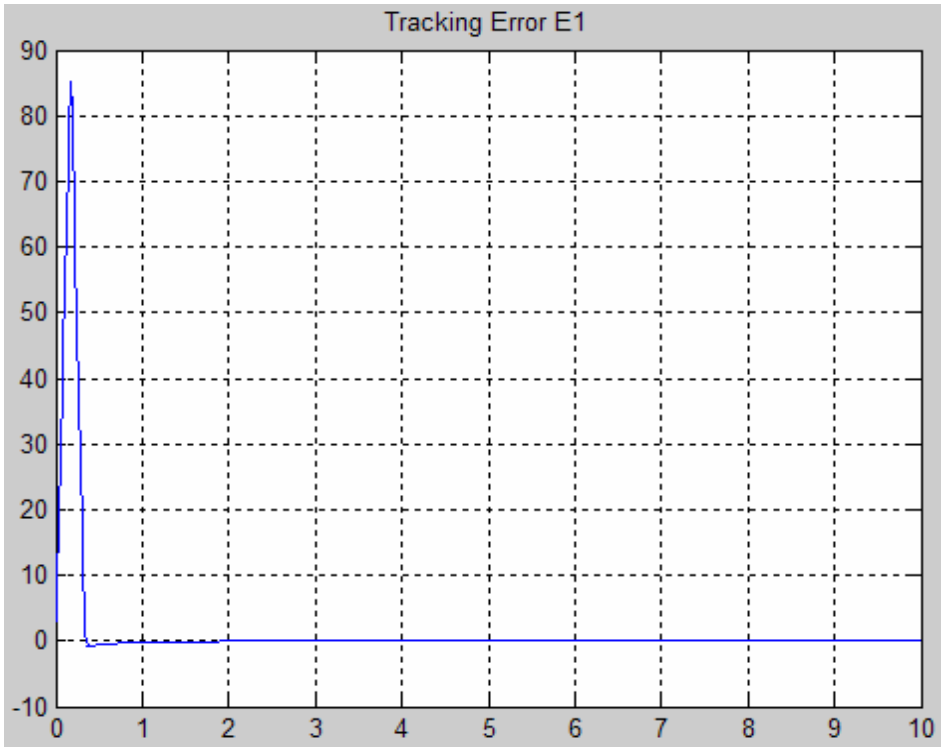
(b)

Figure 5.2.1

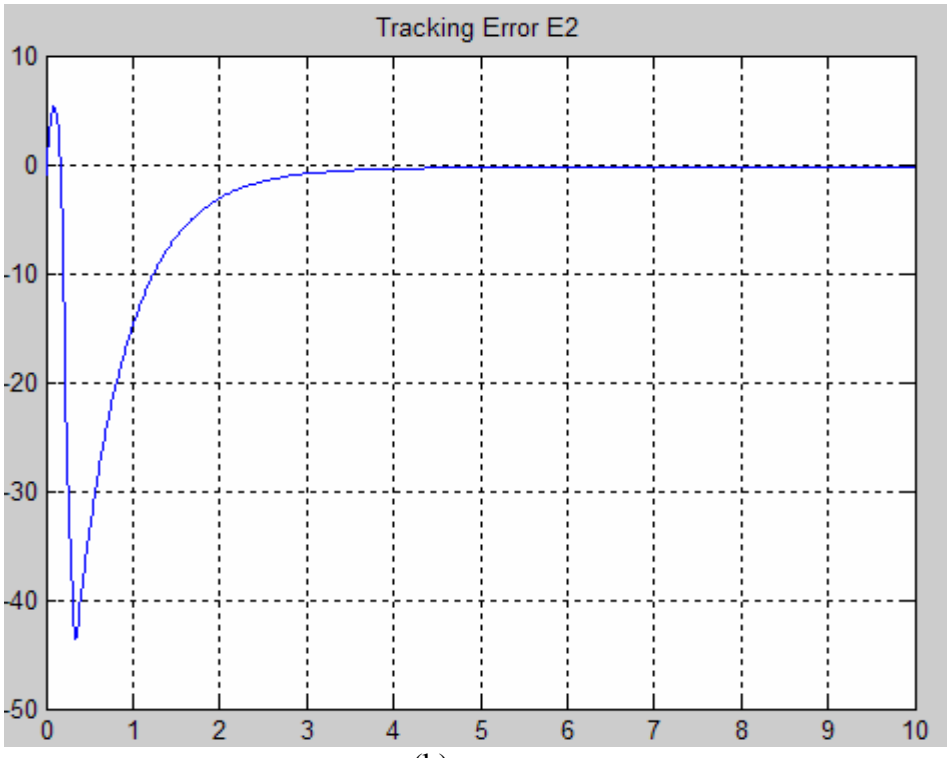
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.1 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.6 sec and
x_2 follows x_{m2} after the time t =2.8 sec. The tracking errors E_1 and E_2 are zero after these times t.

Tracking errors:



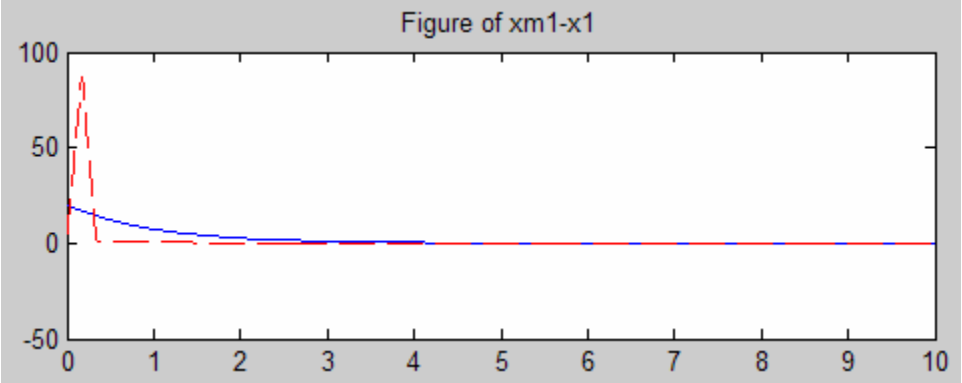
(a)



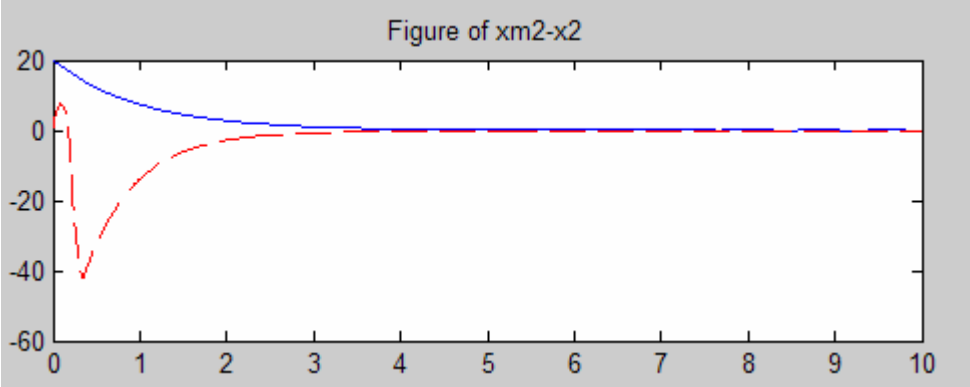
(b)

Figure 5.2.2 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ 20]$



(a)



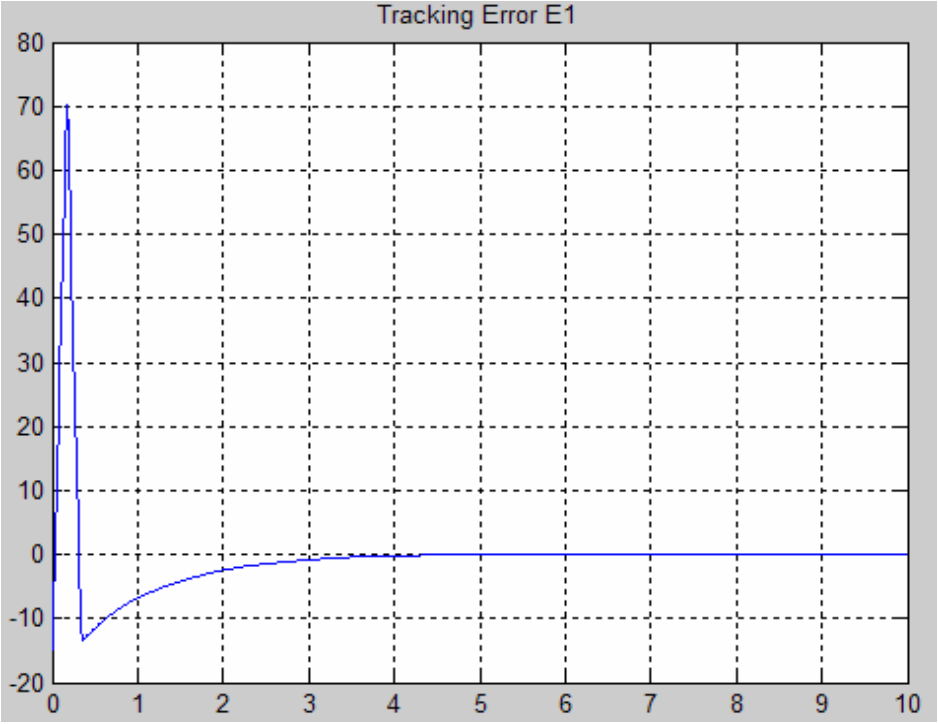
(b)

Figure 5.2.3

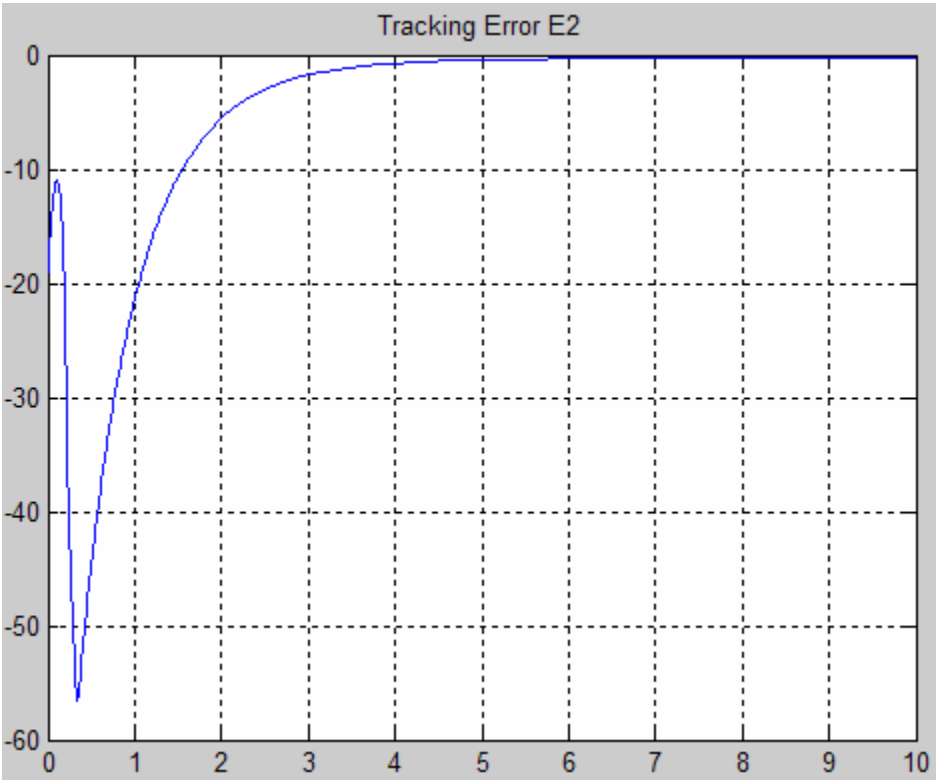
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.3 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=2.1 sec and
x_2 follows x_{m2} after the time t =2.8 sec. The tracking errors E_1 and E_2 are zero after these times t.

Tracking errors:



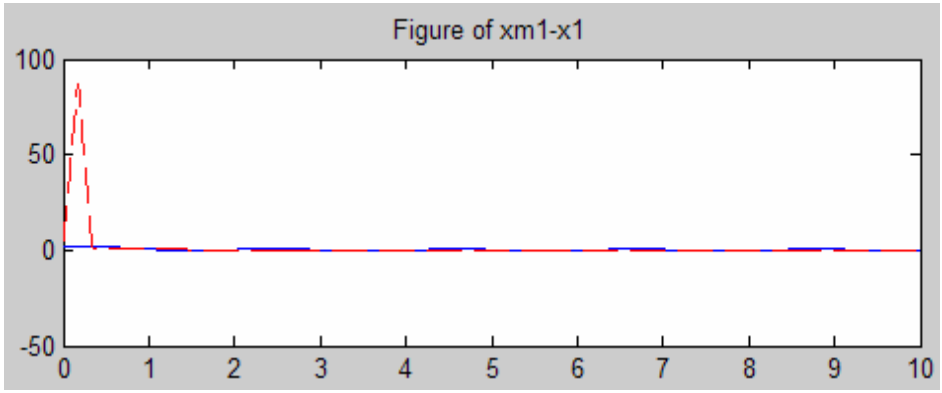
(a)



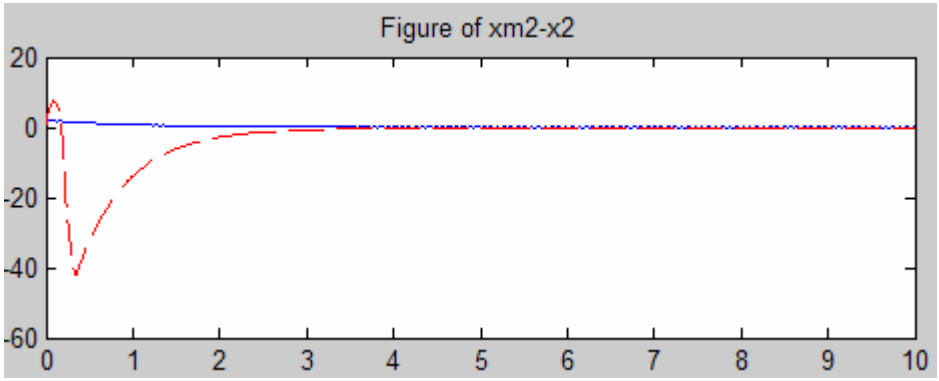
(b)

Figure 5.2.4 Tracking errors E_1 in a) and E_2 in b)

For $u_{m1} = 2\cos(3t)$ $u_{m2} = 5\cos(80t)$ and initial condition $x_{m0} = [2\ 2]$



(a)



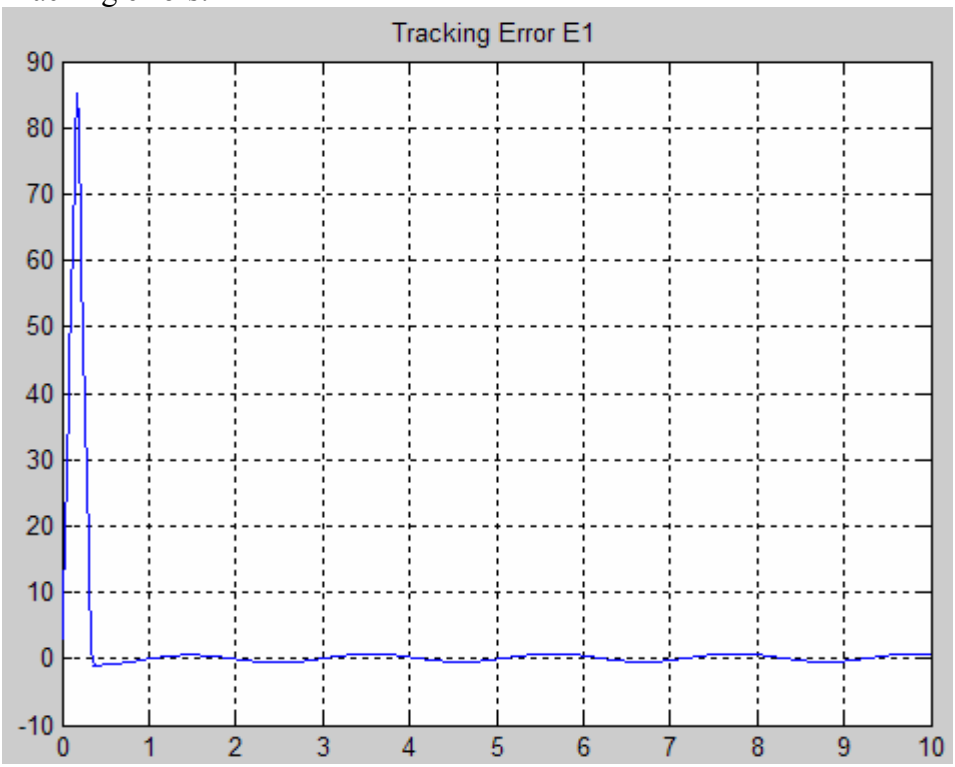
(b)

Figure 5.2.5

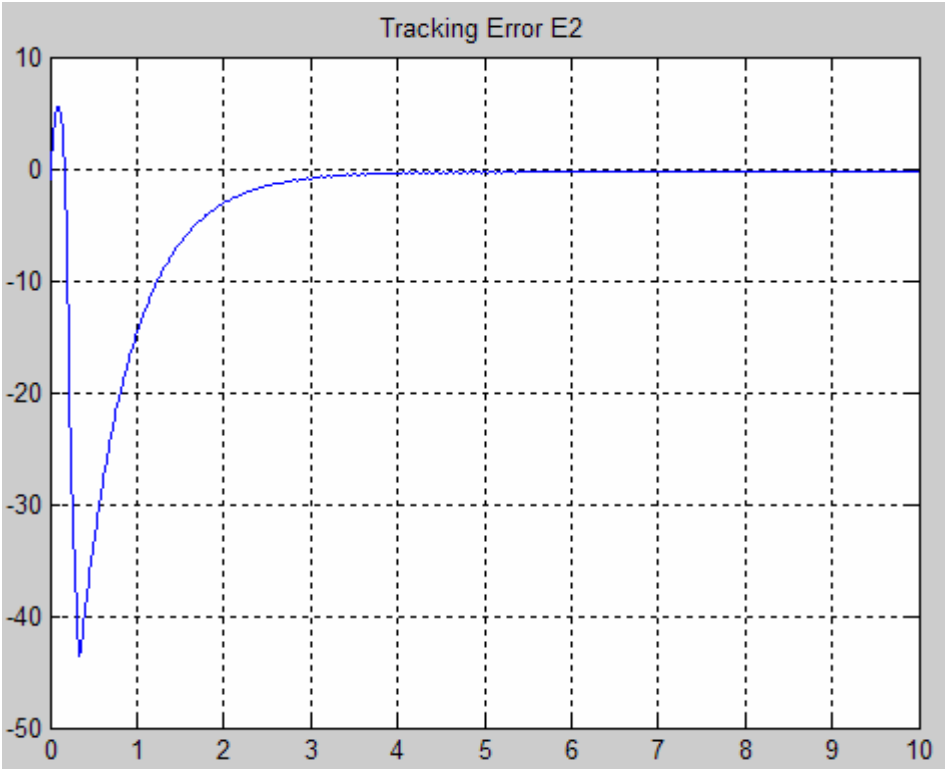
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.5 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.6 sec and
x_2 follows x_{m2} after the time t =2.6 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.6.

Tracking errors:



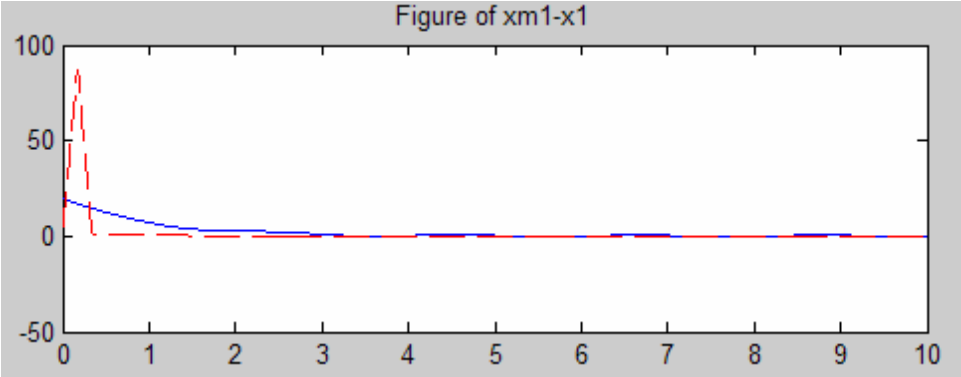
(a)



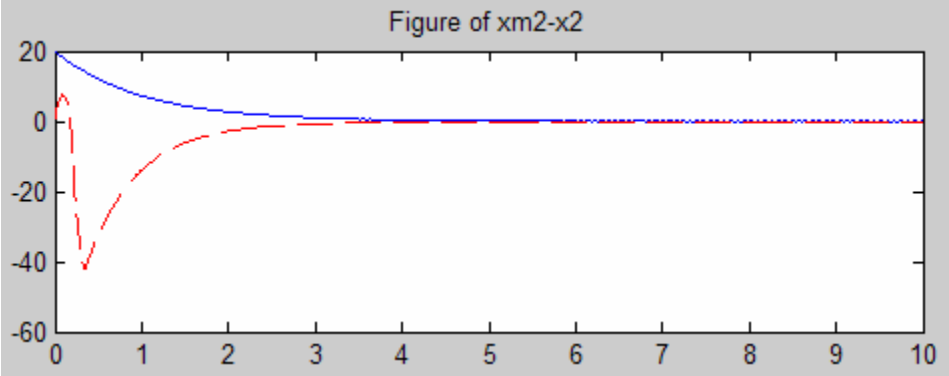
(b)

Figure 5.2.6 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ 20]$



(a)



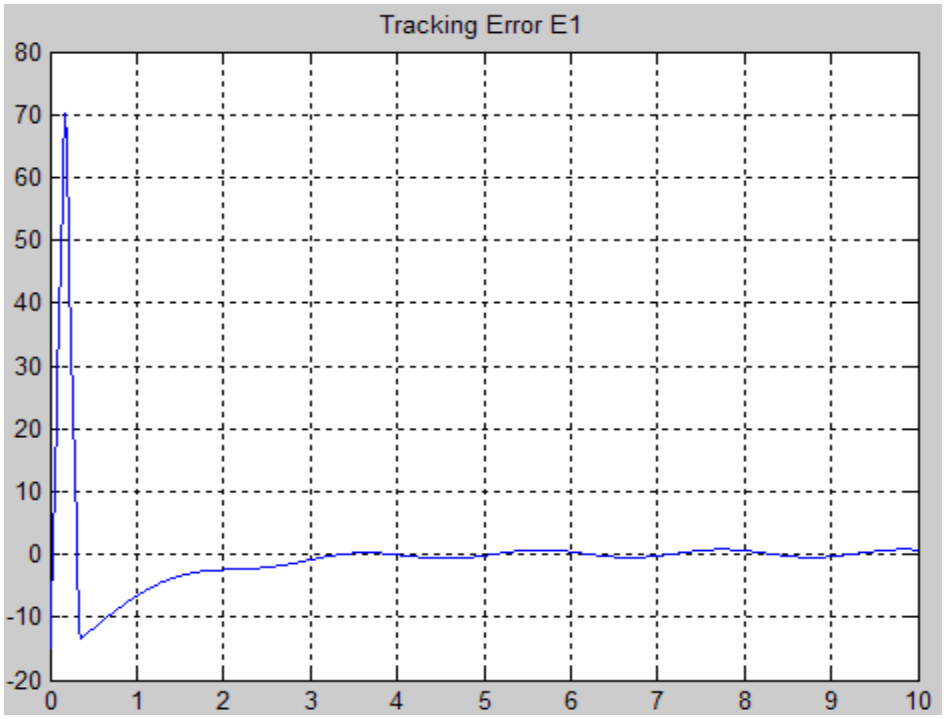
(b)

Figure 5.2.7

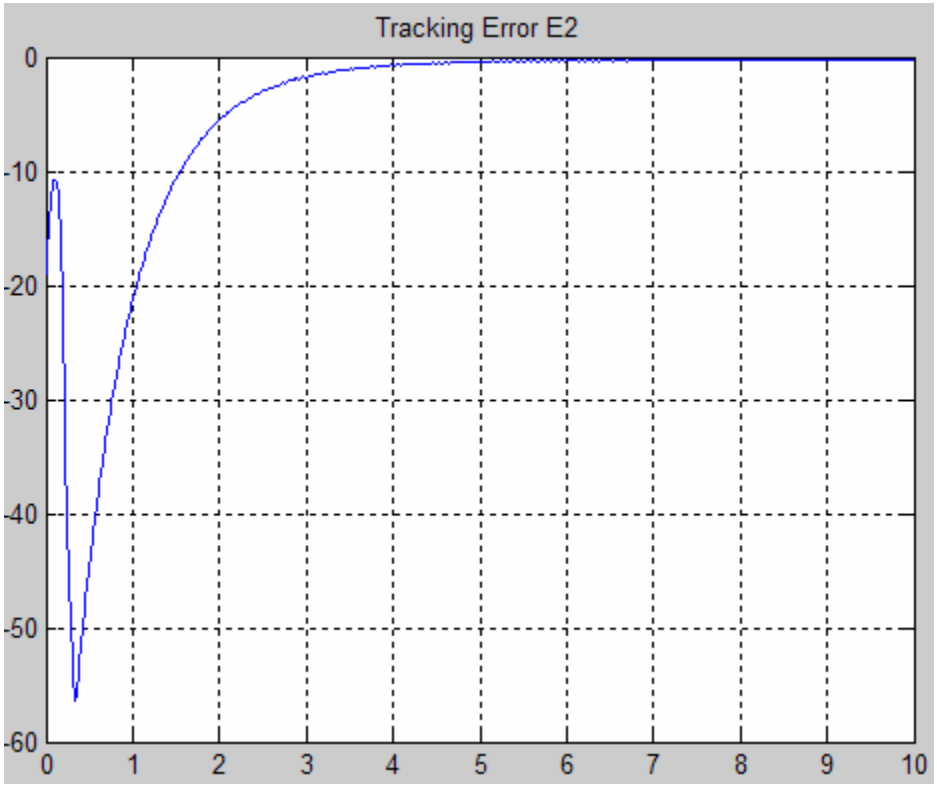
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.7 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=1.6 sec and
x_2 follows x_{m2} after the time t =2.7 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.8.

Tracking errors:



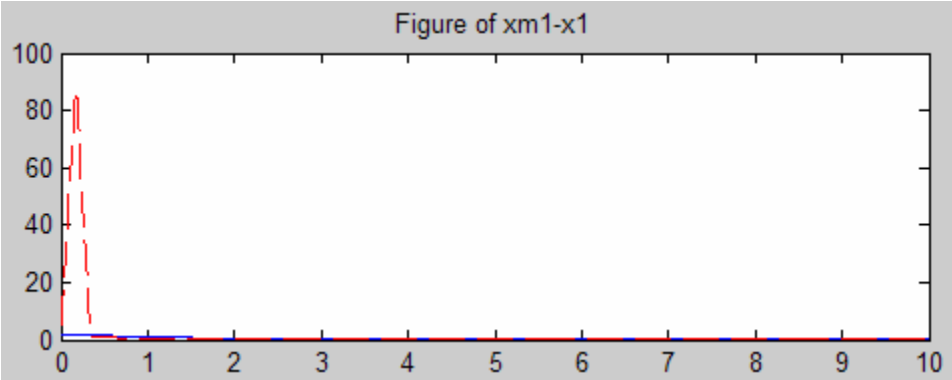
(a)



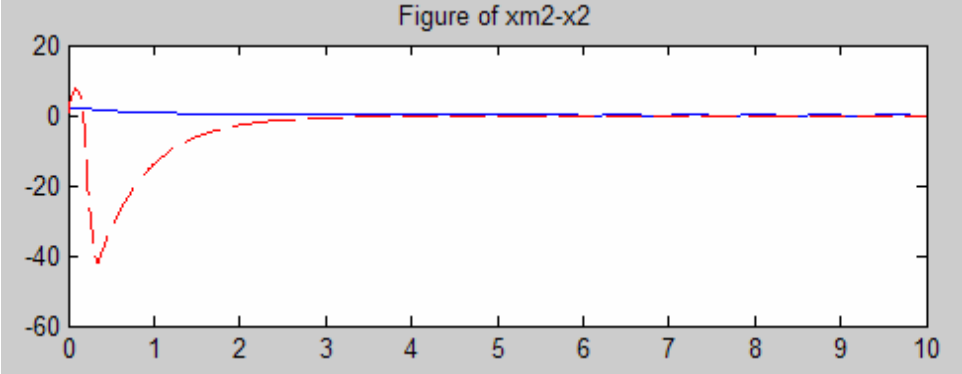
(b)

Figure 5.2.8 Tracking errors E_1 in a) and E_2 in b)

For $u_{m1} = 3(\cos(5t)).\hat{}\,2./(5t+1.)$
$u_{m2} = 5(\cos(8t)).\hat{}5./(8t+1.)$ and initial condition $x_{m0} = [2\ 2]$ we have:



(a)



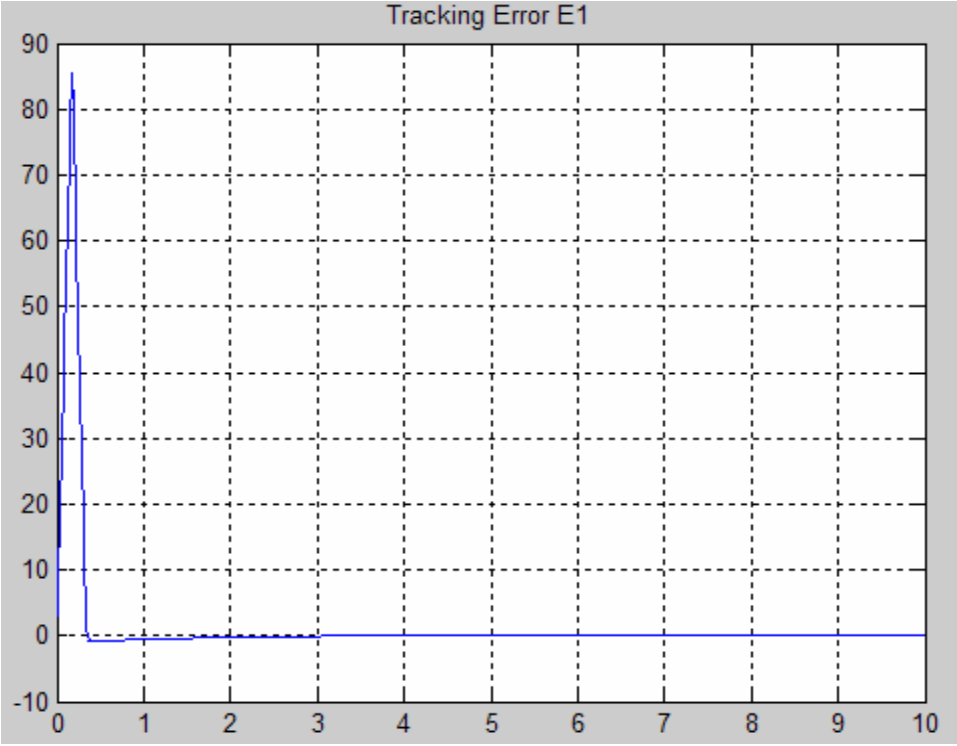
(b)

Figure 5.2.9

[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.9 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.3 sec and
x_2 follows x_{m2} after the time t =2.1 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.10.

Tracking errors:

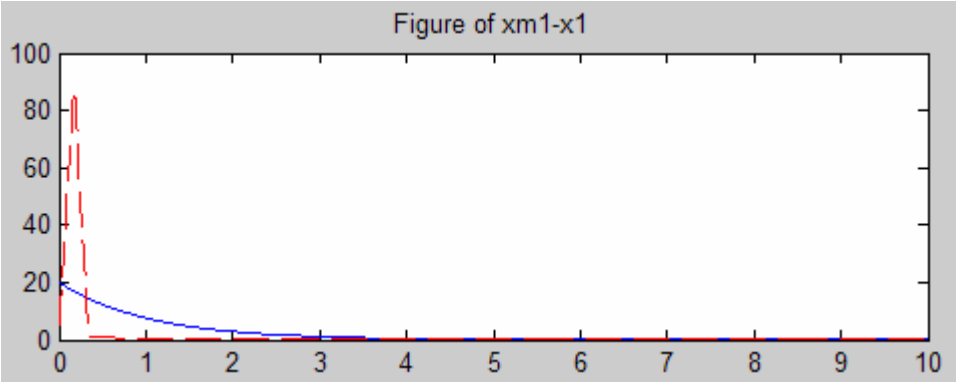


(a)

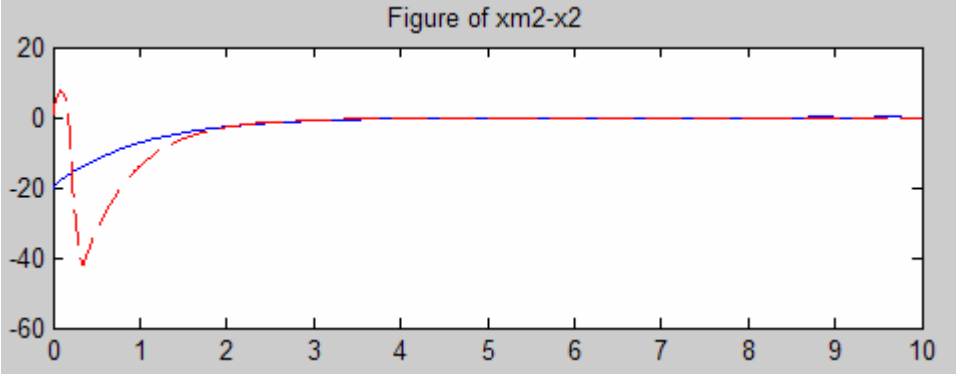
(b)

Figure 5.2.10 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ -20]$



(a)



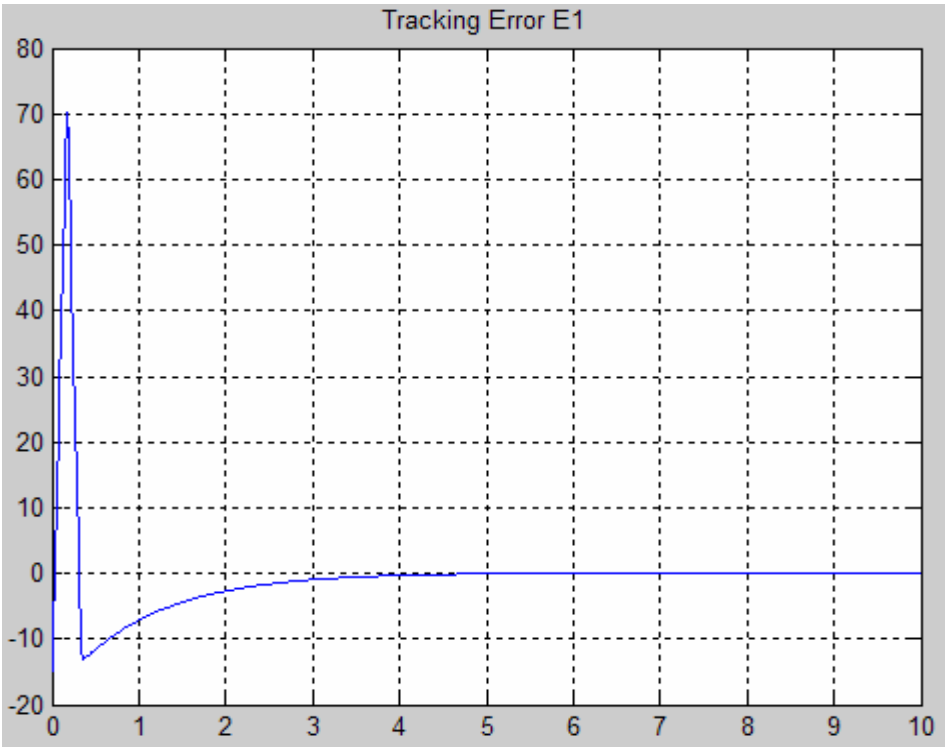
(b)

Figure 5.2.11

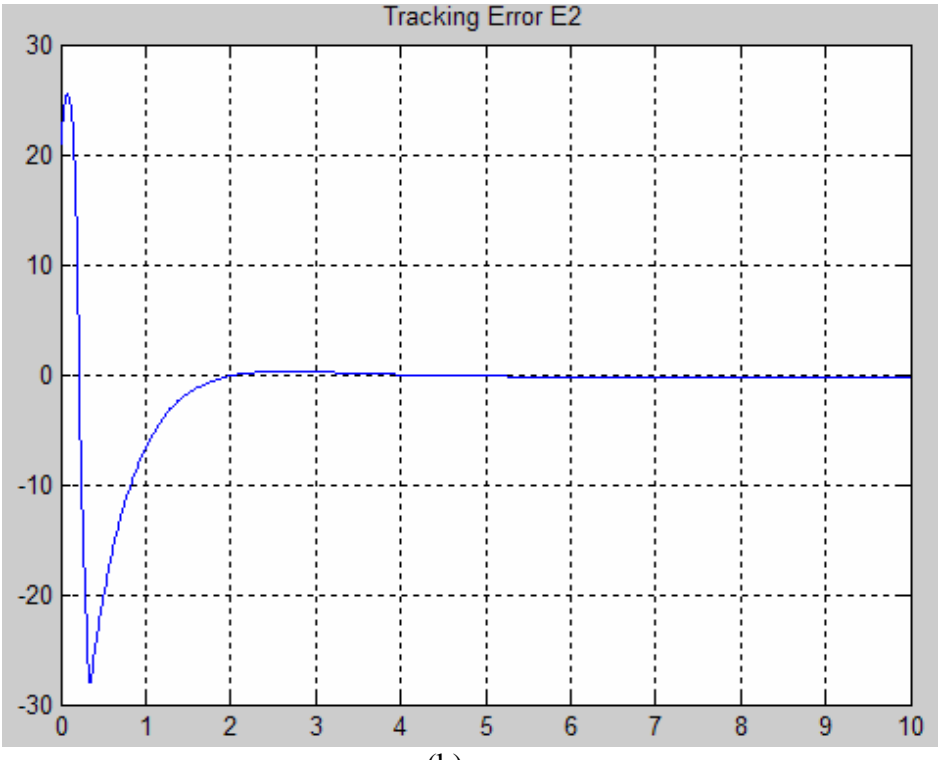
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.11 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=2 sec and
x_2 follows x_{m2} after the time t =1.6 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.12.

Tracking errors:



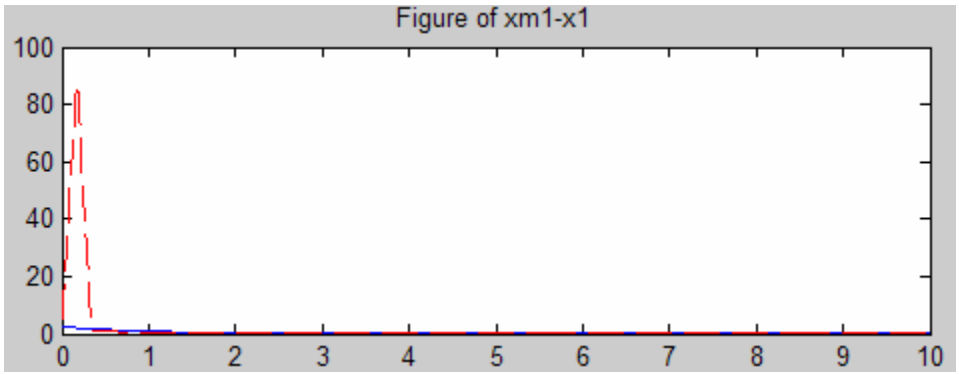
(a)



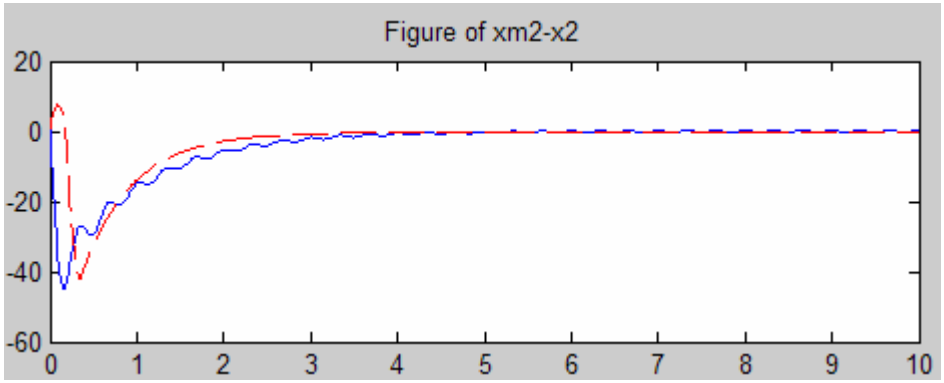
(b)

Figure 5.2.12 Tracking errors E_1 in a) and E_2 in b)

For $u_{m1} = 50\sin c(1000t)$
$u_{m2} = \text{-}500\sin c(6t)$ and initial condition $x_{m0} = [2\ 2]$



(a)



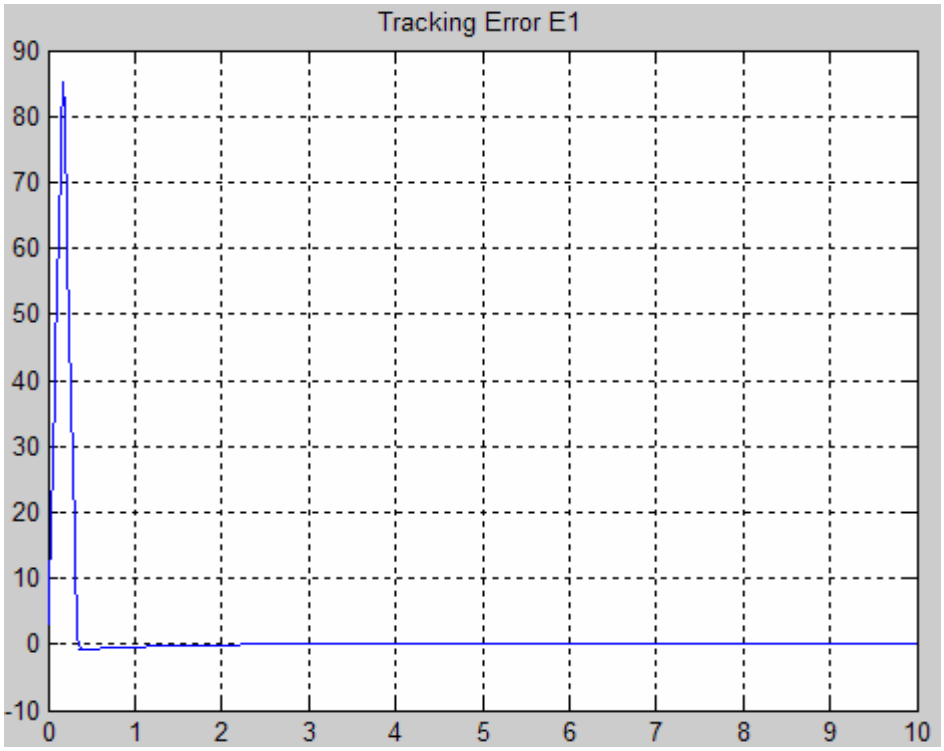
(b)

Figure 5.2.13

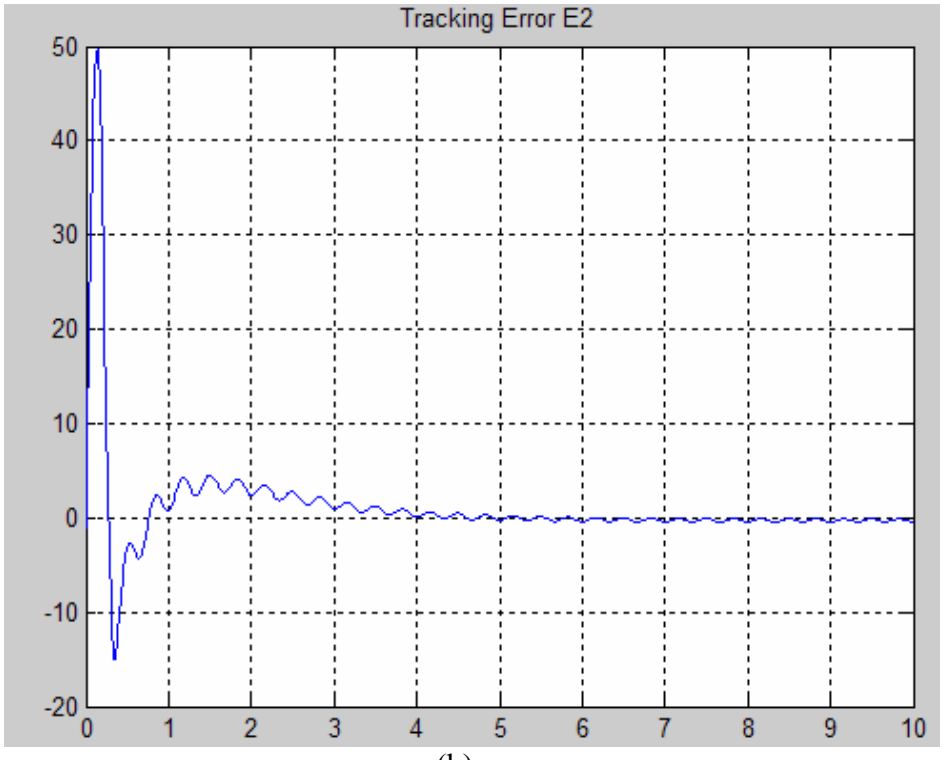
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.13 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.4 sec and
x_2 follows x_{m2} after the time t =1 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.14.

Tracking errors:



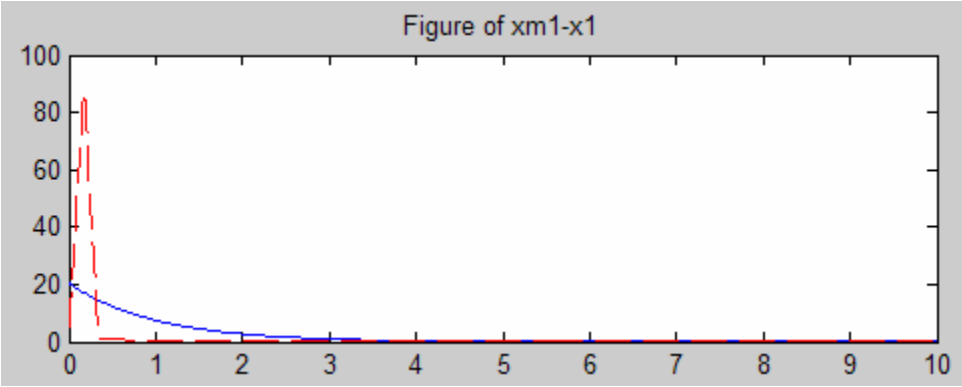
(a)



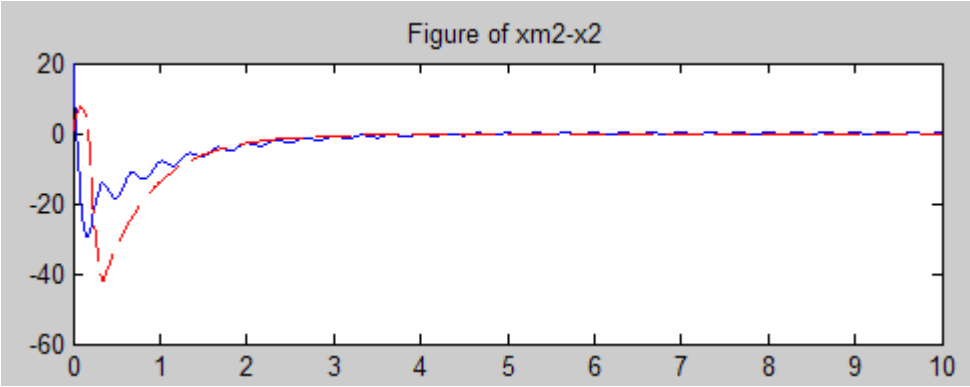
(b)

Figure 5.2.14 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ 20]$



(a)



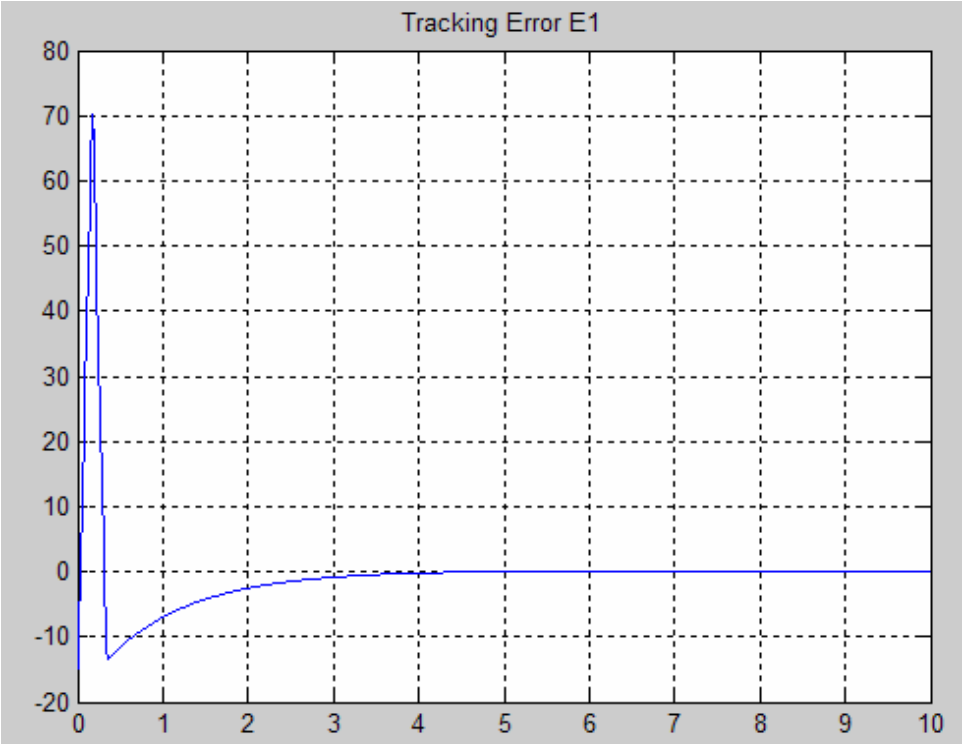
(b)

Figure 5.2.15

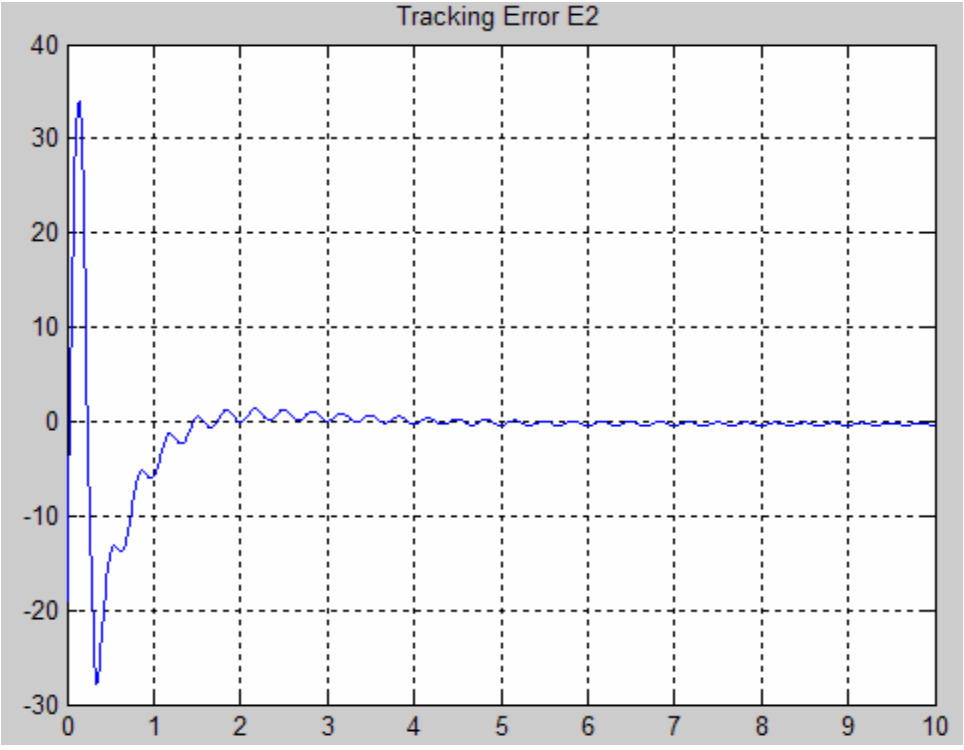
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.15 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t =2 sec and
x_2 follows x_{m2} after the time t =1.6 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.16.

Tracking errors:



(a)

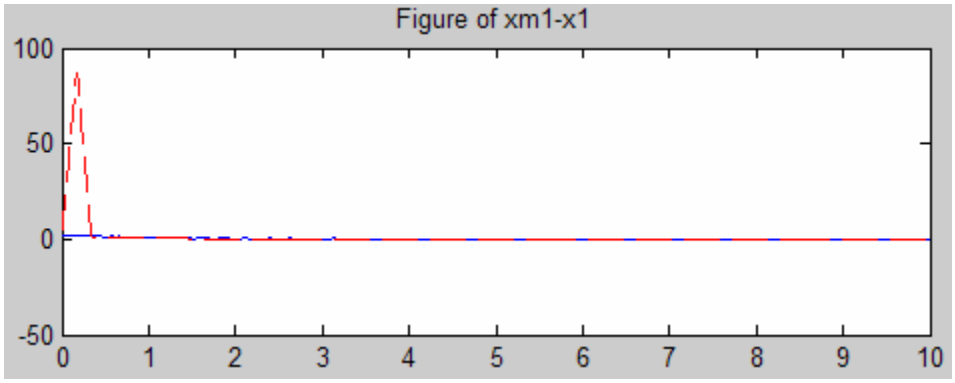


(b)

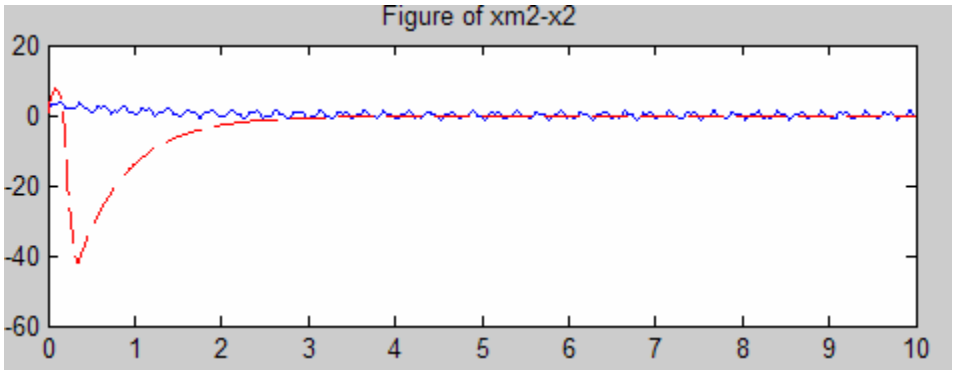
Figure 5.2.16 Tracking errors E_1 in a) and E_2 in b)

For $u_{m1} = 5\sin(25t) + 6\sin(61t)$
$u_{m2} = 22\sin(25t) + 26\sin(61t)$ and initial condition $x_{m0} = [2\ 2]$

We take the following results:



(a)



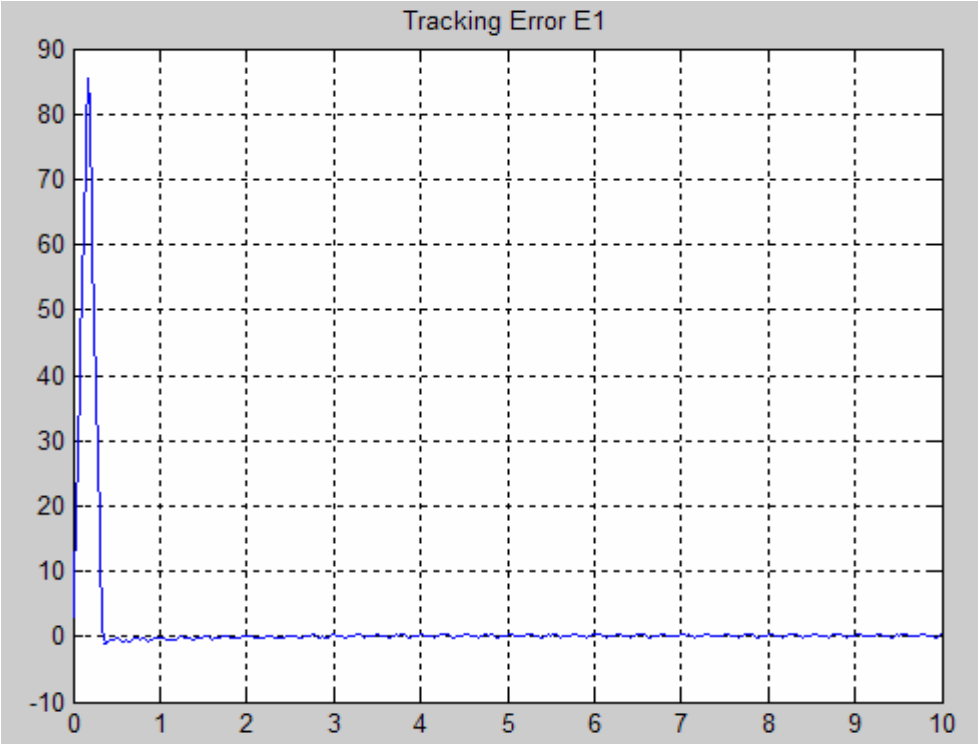
(b)

Figure 5.2.17

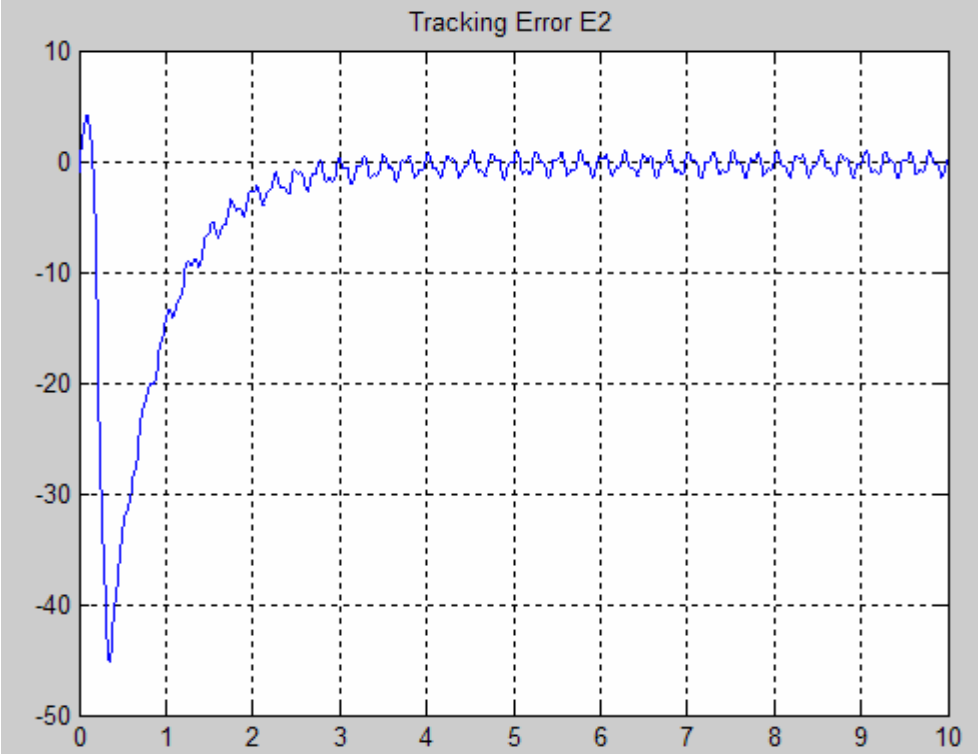
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.17 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.3 sec and
x_2 follows x_{m2} after the time t =2.2 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.18.

Tracking errors:



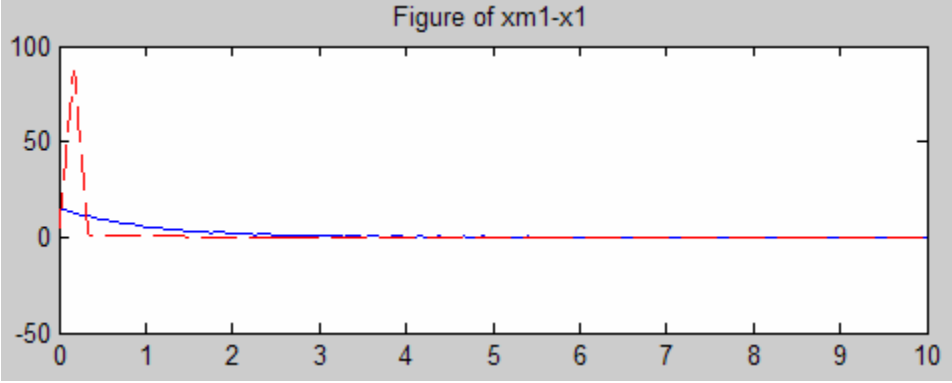
(a)



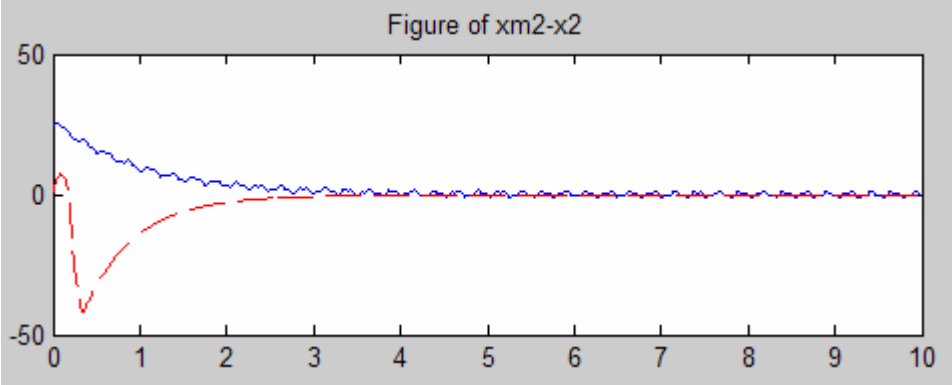
(b)

Figure 5.2.18 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ 20]$



(a)



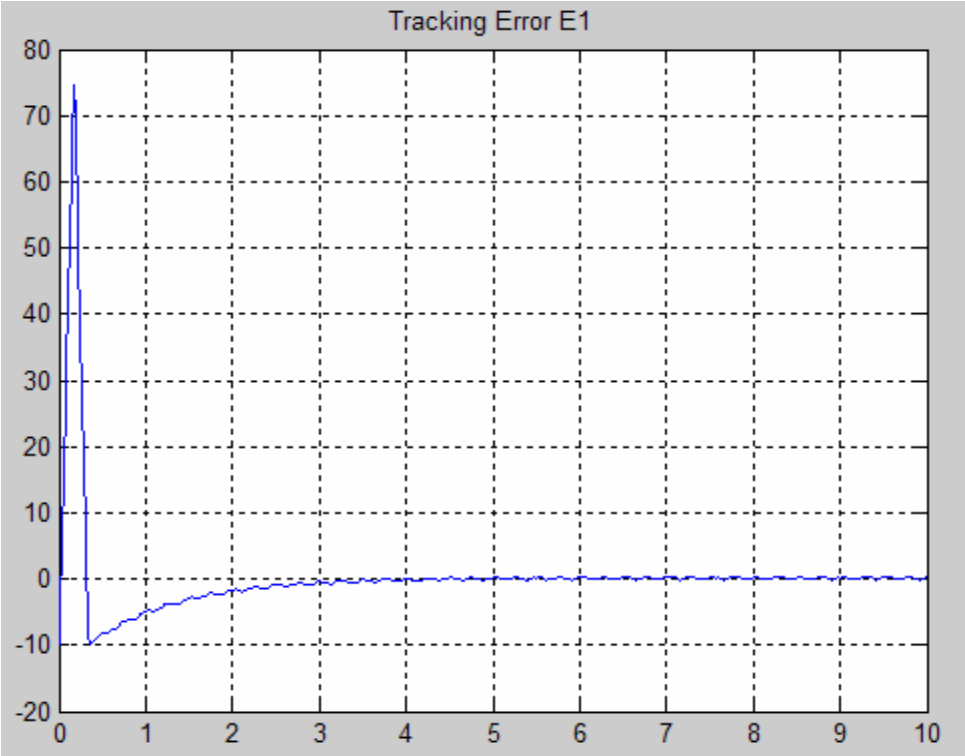
(b)

Figure 5.2.19

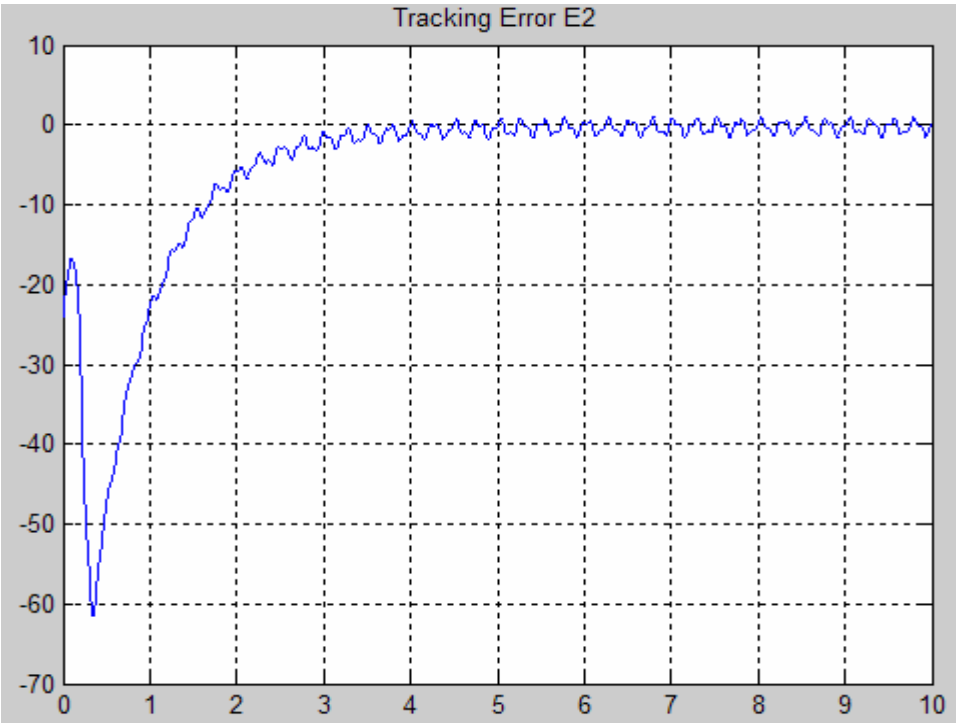
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.19 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=1.6 sec and
x_2 follows x_{m2} after the time t =2.6 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.20.

Tracking errors:



(a)

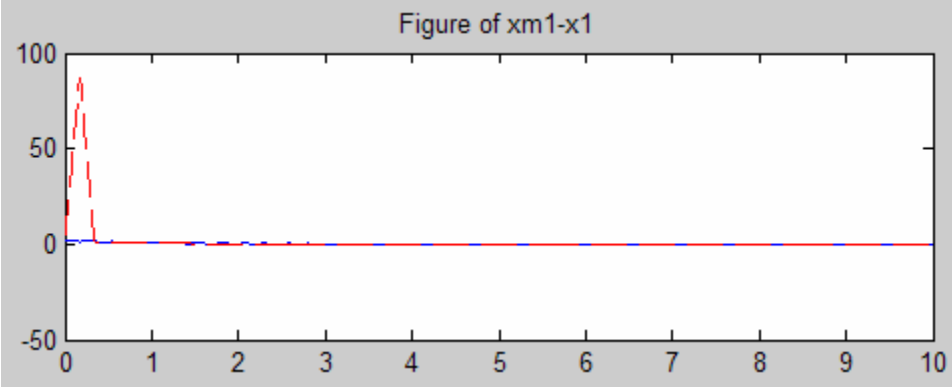


(b)

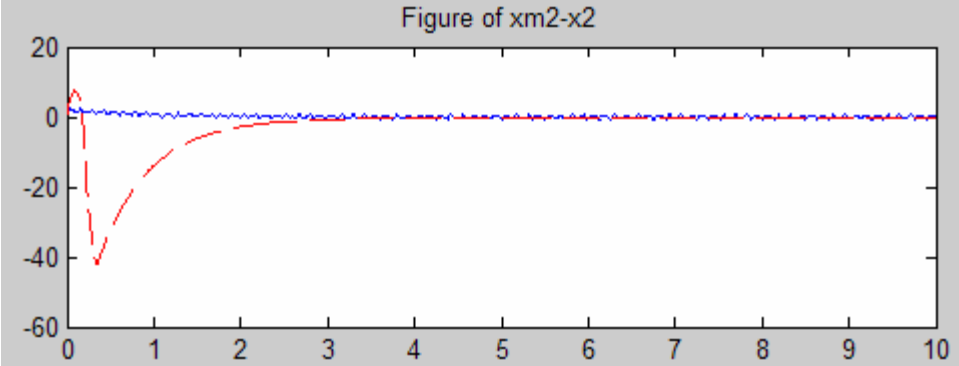
Figure 5.2.20 Tracking errors E_1 in a) and E_2 in b)

For $u_{m1} = 5\cos(25t) + 6\cos(61t)$
$u_{m2} = 22\cos(50t) + 26\sin(90t)$ and initial condition $x_{m0} = [2\ 2]$

We take the following results:



(a)



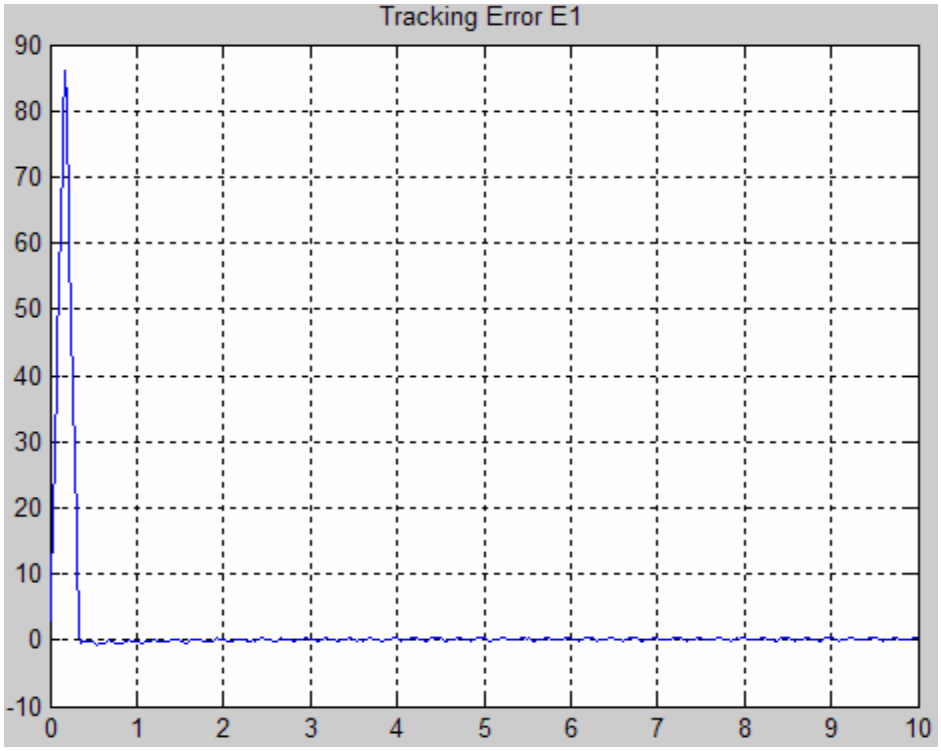
(b)

Figure 5.2.21

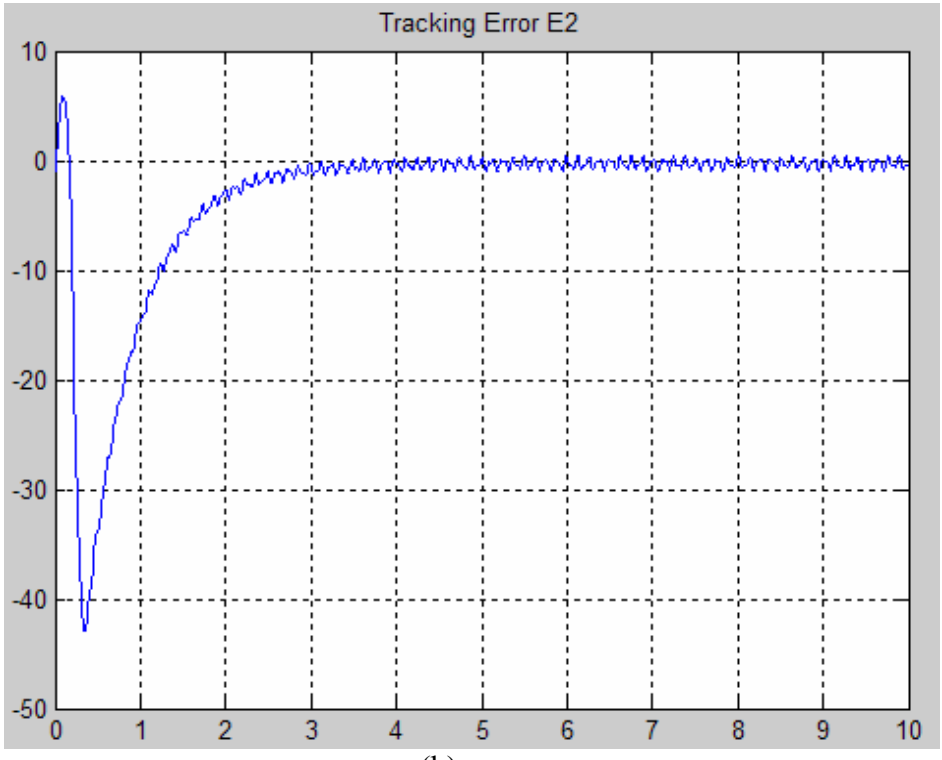
[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.21 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=0.4 sec and
x_2 follows x_{m2} after the time t =2 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.22.

Tracking errors:



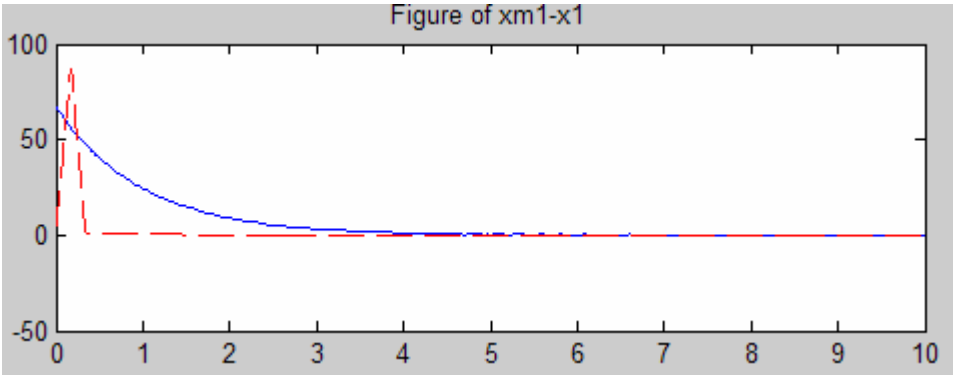
(a)



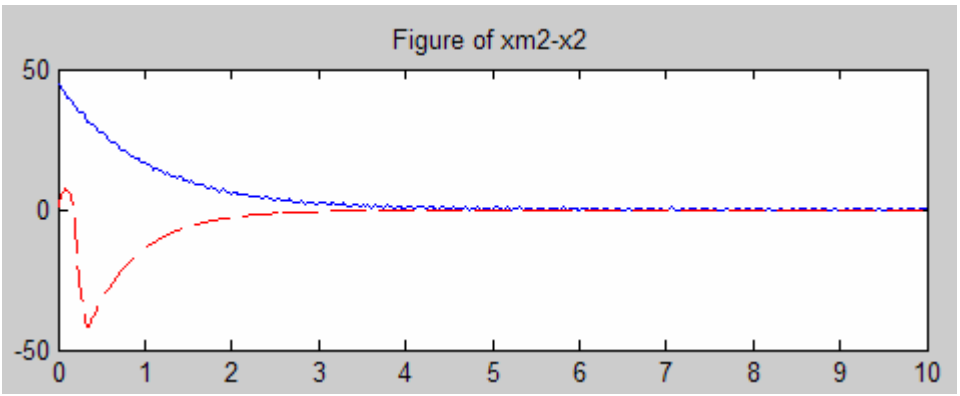
(b)

Figure 5.2.22 Tracking errors E_1 in a) and E_2 in b)

Now, follows an another simulation with different initial condition $x_{m0} = [20\ 20]$



(a)



(b)

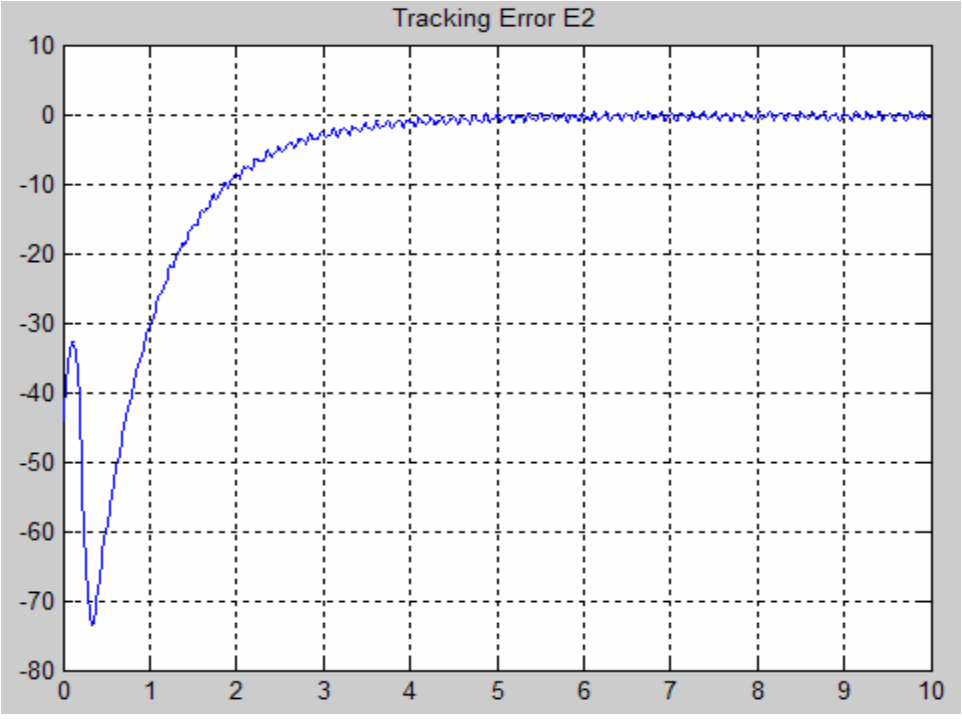
Figure 5.2.23

[In (a) we see the plot of x_{m1}, x_1 and in (b) we see the plot of x_{m2}, x_2]

In figure 5.2.23 we can see that the output of our actual system follows the output of our desired trajectory of our reference system.
The plot of output of the reference system is with the blue line, and the output of our system is with the red line. We can see that x_1 follows x_{m1} after the time t=2.6 sec and
x_2 follows x_{m2} after the time t =2.9 sec. The tracking errors E_1 and E_2 are zero after these times t as we can see in the Figure 5.2.24.

Tracking errors:

(a)



(b)

Figure 5.2.24 Tracking errors E_1 in a) and E_2 in b)

Chapter 6

Conclusions and future Developments

6.1 Final Conclusions

This project provides the model of the adaptive tracking of the dynamic behavior of two molecular species in the intra-cellular environment. This function seems to be very significant for the cell because it is essential in cancer research, because we have the great opportunity, are forced the cancer cells to follow a stable model in which the populations of cancer cells would have eliminated.
A simulation of it thus would be of a great interest for the biologists, since they waste a lot of time for the experiments in the laboratory. A first model of it was constructed by Kholodenko, [48] from the calculation of the kinetics.
Producing data from that model we proved that recurrent high order neural networks provide very excellent approximation capabilities.

When biological data are available, the RHONN methods are capable of the computer simulation of the effects caused by a change in the concentration of one molecular specie.
The biologists elsewhere have to know very complicated and always precise differential equations to describe those effects. In opposite we proved how easy for the RHONN is to approximate systems of the form:

$$\dot{x}_i = f(x_i, u),\ u : \text{external input}$$

Recurrent high order neural networks provide a powerful mathematical tool for the calculation of the appropriate control in order to bring the system to the right dynamical
behavior. In medical words, this means finding the treatment to a disease.

It is questionless, how complicated things are in the cell. The experiments in the cell are also very difficult. One could easily imagine thus how hard is for the biologists to understand only from the chemical experiments the interaction between two or more cells. Control theory provides powerful mathematical tools in order to make these enormous numbers of data useful and easy to handle.
To sum up, technical knowledge coming from control engineers should be used in cooperation with biological knowledge in order to produce significant results in medicine.

6.2 Future Work

The design of adaptive controllers with certain robustness properties with respect to modeling errors or external disturbances can be further improved in order to cover the aspect of this subject. This condition guarantees the existence and uniqueness of solutions of $\dot{x} = -Ax + W^* S(x) + S'(x) W_1^* u + \omega_0(x,u)$ which is necessary according to Theorem 2.1.1 for the actual system. Furthermore, larger values of $k1,k2$ cause larger modeling error, but we can take small $k1,k2$ because the approximation error ε can be considered arbitrarily small, according to Theorem 2.1.1.

Another future extension of my work is the generalization for n molecular species of the cell:

This general control of the system is:

$$\dot{x}_1 = u_1(x_1, x_2, ..., x_n) \cdot x_1$$

$$\dot{x}_2 = u_2(x_1, x_2, ..., x_n) \cdot x_2$$

.

.

.

$$\dot{x}_n = u_n(x_1, x_2, ..., x_n) \cdot x_n$$

and the only think to do this is the considering in the n-dimensionality.

Another significant matter is the use of Recurrent High Order Neural Networks in order to find the appropriate control input for pathways which do not have a desirable behaviour. RHONNs is an effective method and can calculate the control input in order the system obtain the desirable state. This can bring about benefits as the treatment of a disease.
In conclusion, biologists recognize system approaches as necessary to understand the complex mechanisms or interactions among cells. There are various tools and methods from system engineering which can help them handle the experimental data or make simulations. Consequently, the application of Control Theory to biological research is necessary and can lead to important innovations in Biology and Medicine.

“Before you begin a thing, remind yourself that difficulties
and delays quite impossible to foresee are ahead. If you
could see them clearly, naturally you could do a
great deal to get rid of them but you can’t.
You can only see one thing clearly and
that is your goal. Form a mental
vision of that and cling to it
through thick and thin.”

Kathleen Norris

Appendix A

Case 1:

Let us now choose $u_{m1} = \sin(10t)$
$u_{m2} = \sin(20t)$

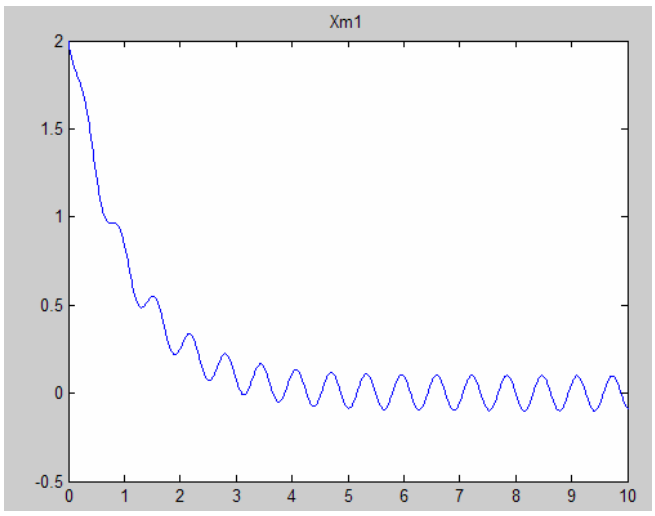
The steady-state response of the plant is given by

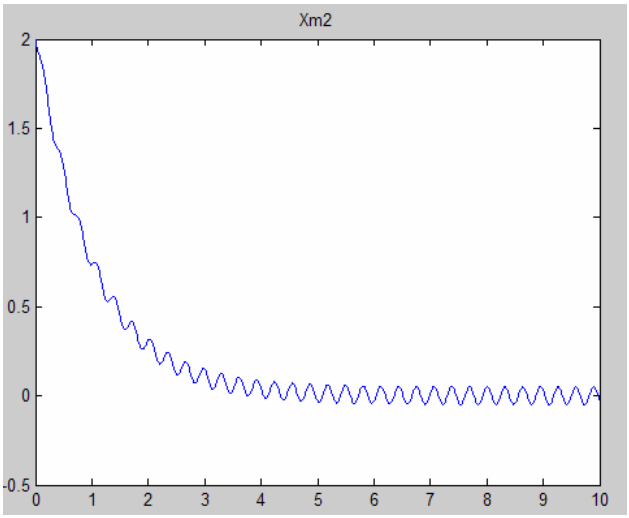
$$y_1 \approx A_1 \cdot \sin(10t + \phi_1)$$
$$y_2 \approx A_2 \cdot \sin(20t + \phi_2)$$

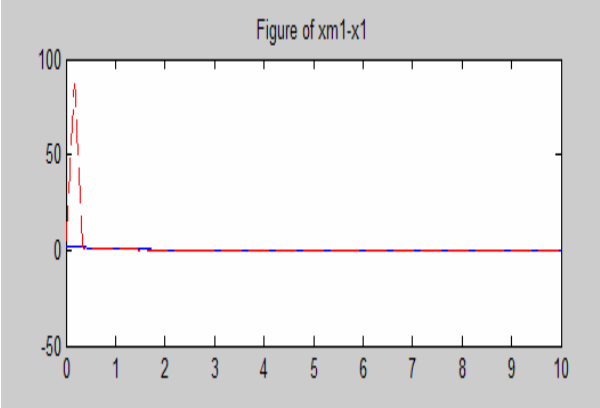
where

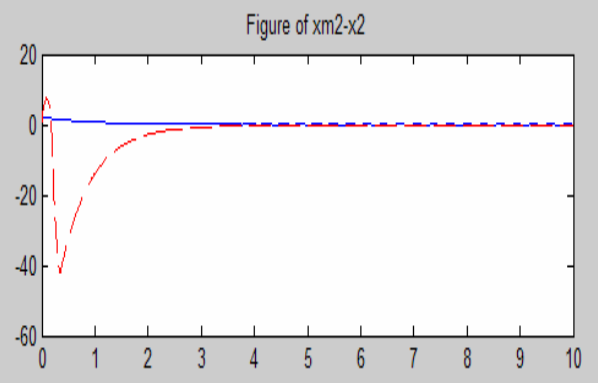
$$A_1 = \frac{|b|}{|j10 + a|} = \frac{|b|}{\sqrt{10^2 + a^2}}, \qquad A_2 = \frac{|b|}{|j20 + a|} = \frac{|b|}{\sqrt{20^2 + a^2}},$$
$$\phi_1 = (\mathrm{sgn}(b) - 1) \cdot 90^o - \tan^{-1}\frac{10}{a} \qquad \phi_2 = (\mathrm{sgn}(b) - 1) \cdot 90^o - \tan^{-1}\frac{20}{a}$$

In our case we take:





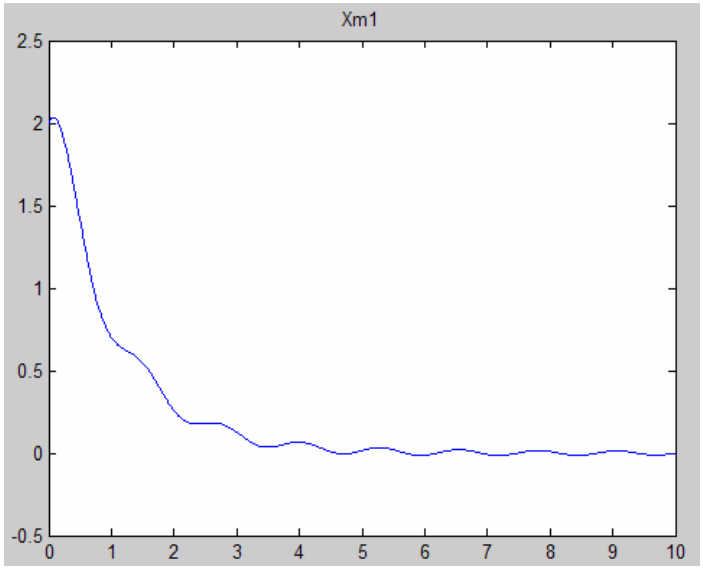


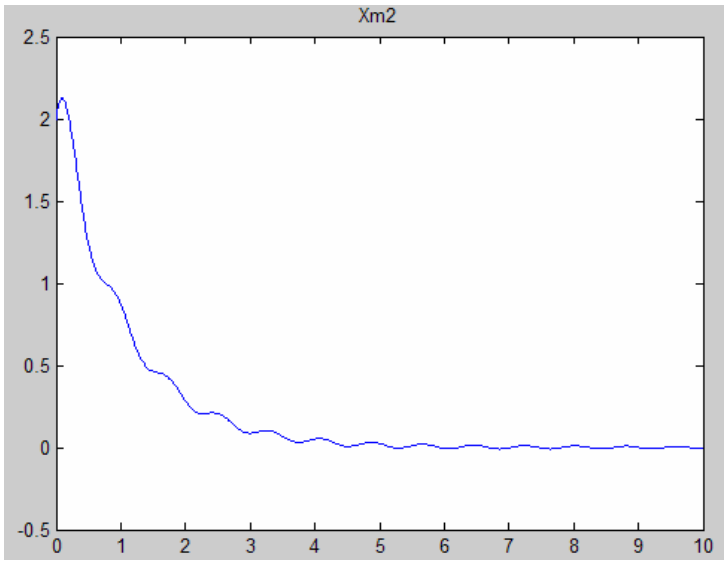


I am giving this solution because it is important to see that these figures are in the simulation results with the blue line, with the red line in the simulations are the results of the X1, X2 which we taking from the model.

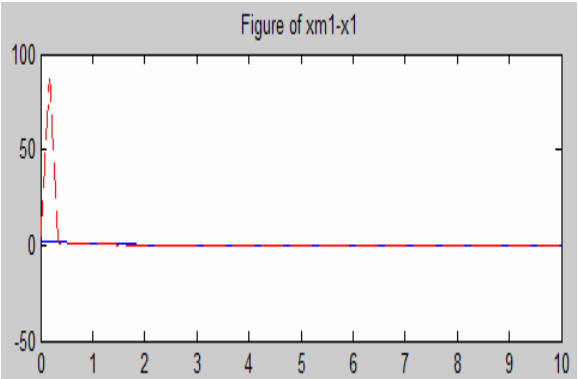
Case 2:

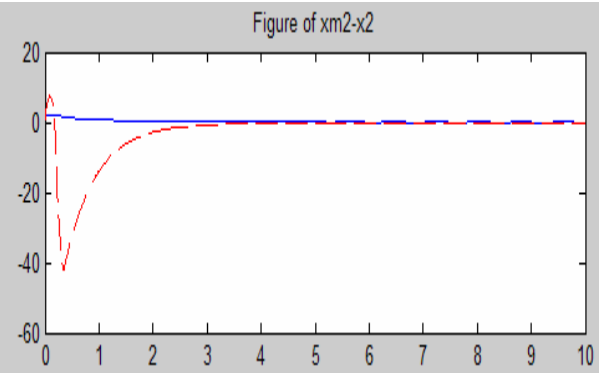
In the case of $u_{m1} = 3\cos(5t)./(5.t+1.)$ $u_{m2} = 5\cos(8t)./(8.t+1.)$ and with the initial condition $x_{m0} = [2\ 2]$





The solution of this linear reference model.





I am giving this solution because it is important to see that these figures are in the simulation results with the blue line, with the red line in the simulations are the results of the X1, X2 which we taking from the model.

BIBLIOGRAPHY

[1] G.A. Rovithakis, and M.A. Christodoulou, Adaptive Control with Recurrent high order Neural Networks, Springer Verlag, New York, USA, 2000.

[2] P. Ioannou, Robust Adaptive Control, University of Southern California

[3] Systems Biology IEEE CSS Magazine Aug. 2003

[4] Rovithakis G.A., Christodoulou M.A (1994) *IEEE Trans. Systems Man Cybernetics*, 24, 400-412

[5] O. Wolkenhauer, M. Ullah, P. Wellstead, K-H. Cho.
The Dynamic Systems Approach to Control and Regulation of Intracellular Networks December 2004

[6] E.D. Sontag. Mathematical Control Theory. Springer, 1998

[7] Hertz J., Krogh A., and Palmer R.G - (1991). *Introduction to the Theory of Neural Computation* (Addison-Wesley Publishing Company)

[8] Cybenko G. (1989) *Mathematics of Control, Signals, and Systems*, 2, 303-314

[9] Funahashi K. (1989) *Neural Networks*, 2, 183-192

[10] Hartman E.J, Keeler J.D., Kowalski J.M. (1990) *Neural Computation*, 2, 201-215

[11] Hornik K.M., Stinchombe M., White H. (1989) *Neural Networks*, 2, 359-366

[12] Hou Z. (1992) *Analysis of Auto Powertrain Dynamics and Modelling Using Neural Networks*, M.Sc. Thesis, Department of Electrical Engineering, University of Notre Dame.

[13] Narendra K.S, Parthasarathy K. (1990) *IEEE Transactions on Neural Networks*, 1, 4-27

[14] J.D. Murray. Mathematical Biology. Springer Verlag, third edition, 2002

[15] G.A. Rovithakis, and M.A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks," *IEEE Trans. Systems, Man, andCybernetics*, vol. 24, No. 3, Mar. 1994, pp. 400-412.

[16] R. Heinrich, B. Neel, and T. Rapoport, “Mathematical Models of protein kinase signal transduction,” *Molecular Cell*, vol. 9,no. 5, pp. 957-970, May 2002.