**MODEL ORDER REDUCTION USING GENETIC ALGORITHM AND ITS CONTROL**

**A MAJOR THESIS**

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

**MASTER OF TECHNOLOGY**

**IN**

**CONTROL & INSTRUMENTATION**

BY

**SUMAN CHAUDHARY**

*Under the guidance of*

**Mr. RAM BHAGAT**

(Assistant Professor)

****

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

**2012**

**CERTIFICATE**

This is to certify that the thesis entitled “**MODEL ORDER REDUCTION USING GENTIC ALGORITHM AND ITS CONTROL” being submitted by Suman Chaudhary in the partial fulfillment for the award of the degree of Master of Technology in Control and Instrumentation of Electrical Engineering Department of Delhi Technological University is a record of bonafide work done by her under my supervision and guidance. It is also certified that this dissertation has not been submitted elsewhere for any other degree.**

**Mr.RAM BHAGAT**

Assistant Professor

Department of Electrical Engineering

Delhi Technological University

**ACKNOWLEDGEMENT**

I take the honour and privilege to express my deep sense of gratitude and sincere thanks to Mr. Ram Bhagat, Assistant Professor, EED, Delhi Technological University, for having given me the opportunity to do the project work under his guidance and encouragement throughout the course of this work. It has been great experience to get the basic training of research under his rich experience, exemplary patience and exceptional capabilities of reducing complexities to simpler form. Inspite of busy schedule, he was available whenever I needed help.

I want to thanks Mr. S.T Nagrajan, Assistant Professor, EED, Delhi Technological University, for his constant encouragement and help, whenever I approached him.

I want to express my sincere thanks to all my teachers, for their support during my M.Tech course work. I profusely thank them for having patience to clear my doubts and channelise my efforts.

I am extremely grateful to Prof.Narendra Kumar, HOD, EED, Delhi Technological University, for providing me the required facilities for carrying the project work.

Lastly I would like to express my thanks to all those who helped me directly or indirectly in the progress and completion of this work.

(SUMAN CHAUDHARY)

ROLL.NO-08/C&I/2010

DELHI TECHNOLOGICAL UNIVERSITY

**ABSTRACT**

This project involves the study of techniques of Reduced Order Modeling (ROM) of Higher Order Systems, a branch of systems and control theory, which aims at studying the dynamical properties of systems and reducing the complexity of the system, while preserving the input output characteristics to the maximum extent possible. The study involved model order reduction of linear time invariant (LTI), single input single output (SISO) systems using unorthodox optimization technique ‘Genetic Algorithm’. The same reduction is previously done for comparison by more conventional techniques ‘Routh Pade approximation’ and ‘Routh approximation’.

“Routh Pade approximation” is a mixed method for finding stable reduced-order models using the Pade approximation technique and the Routh-Hurwitz array. This method guarantees stability of the reduced model when the original system is stable.

“Routh approximation method” is based on an expansion that uses the Routh table of the original transfer function, the method has a number of useful properties: if the original transfer function is stable, then all approximants are stable; the sequence of approximants converge monotonically to the original in terms of ‘‘impulse response” energy; and poles and zeros of the approximants move toward the poles and zeros of the original as the order of the approximation is increased.

In the “Genetic algorithm” based reduction method lower order transfer function are determined by minimizing the integral square error between the transient responses of original and reduced order models using Genetic algorithm in MATLAB. The reduction procedure is simple and computer oriented. It is shown that the algorithm has advantage that the reduced order models retain the steady-state value and stability of the original system.

The problem of controller design for higher order system via reduced model is also investigated. The project describes a technique for designing a stabilizing controller for the stable higher order system via its reduced order model. The method uses the parameterization of all compensators that stabilize a given plant. It is shown that the Compensator, which is obtained from reduced model, not only stabilizes reduced Model but also the higher order system. The developed methods are illustrated with numerical examples.

**TABLE OF CONTENTS**

List of figures

List of tables

List of symbols

CHAPTER 1-INTRODUCTION……………………………………………………………...1

* 1. What is model order reduction………………………………………………………...1
  2. Why model order reduction…………………………………………………………...2
  3. Applications of reduced order modeling………………………………………………3

CHAPTER 2- LITERATURE REVIEW……………………………………………………...4

CHAPTER 3- DESCRIPTION OF ROUTH PADE APPROXIMATION METHOD……….10

3.1 Reduction method……………………………………………………………………...10

3.2 Numerical example for order reduction using Routh Pade approximation…………….12

CHAPTER 4-DESCRIPTION OF ROUTH APPROXIMATION METHOD………………..15

4.1 Alpha-Beta Expansion…………………………………………………………………..15

4.2 Routh convergents………………………………………………………………………19

4.3 Reciprocal transformation………………………………………………………………21

4.4 Routh Approximation…………………………………………………………………...21

4.5 Properties of Routh Approximation……………………………………………………22

4.6 Numerical example for order reduction using Routh Approximation………………….25

CHAPTER 5-MODEL ORDER REDUCTION USING GENETIC ALGORITHM…………29

5.1 Methodology……………………………………………………………………………..30

5.1.1 Initialization………………………………………………………………………...30

5.1.2 Selection…………………………………………………………………………….30

5.1.3 Reproduction………………………………………………………………………..31

5.1.4 Termination………………………………………………………………………….31

5.2 Fitness function…………………………………………………………………………...33

5.3 Description of Genetic algorithm in Optimization toolbox in MATLAB………………..34

5.3.1 Some Genetic algorithm Terminology and their relevance in Optimization toolbox34

5.3.2 Outline of how Genetic algorithm works in Optimization toolbox…………………35

5.4 Numerical example of order reduction using Genetic algorithm…………………………36

5.5 Conclusion………………………………………………………………………………...40

5.6 Plots of various aspects of Genetic algorithm executed…………………………………..41

CHAPTER 6-CONTROL OF HIGHER ORDER SYSTEM VIA ITS REDUCED MODEL…44

6.1 Problem statement……………………………………………………………………….45

6.2 Controller parametrization……………………………………………………………….45

6.3 Controller design………………………………………………………………………...46

6.4 Inference………………………………………………………………………………....48

6.5Numerical example for controller design of higher order system via its reduced model

obtained from Routh Pade Approximation……………………………………………….49

6.6 Numerical example for controller design of higher order system via its reduced model

obtained from Routh Approximation…………………………………………………….52

6.7 Numerical example for controller design of higher order system via its reduced model

obtained from Genetic algorithm based Optimization……………………………………55

6.8 Conclusion……………………………………………………………………………….58

CHAPTER-7 CONCLUSION & SCOPE FOR FUTURE WORK…………………………..59

References………………………………………………………………………………………60

**LIST OF FIGURES**

Fig. 3.2.1- Simulation result of Step Response of reduced order Routh Pade Approximation

and Higher order system ………………………………………………………14

Fig. 4.1.1- Alpha Beta Canonical expansion………………………………………………...16

Fig. 4.6.1- Simulation result of Step Response of reduced order Routh Approximation

and Higher order system…………………………………………………………28

Fig. 5.1- Problem solution using Genetic algorithm………………………………………29

Fig. 5.1.1- Flowchart of Genetic Algorithm………………………………………………...32

Fig. 5.4.1- Optimization Toolbox……………………………………………………………37

Fig. 5.4.2- Simulation result of Step Response of reduced model by Genetic Algorithm

and Higher order system……………………………………………………….39

Fig. 5.5.1- Simulation result of comparison Step Responses of reduced order models

obtained with Higher order system…………………………………………….40

Fig. 5.6.1- Best fitness plot…………………………………………………………………41

Fig. 5.6.2- Best individual plot……………………………………………………………..41

Fig. 5.6.3- Fitness scaling plot……………………………………………………………...42

Fig. 5.6.4- Genealogy plot………………………………………………………………….42

Fig. 5.6.5- Selection plot……………………………………………………………………43

Fig. 5.6.6- Stopping criteria plot……………………………………………………………43

Fig. 6.5.1- Simulation result of Step response of reduced Routh Pade Approx. and Higher

order system with controller ……………………………………………………51

Fig. 6.6.1- Simulation result of Step response of reduced Routh Approx. and Higher order

system with controller …………………………………………………………54

Fig. 6.7.1- Simulation result of Step response reduced model by Genetic algo and Higher

order system with controller…………………………………………………….57

Fig. 6.8.1- Simulation result of comparison Step responses of Higher order system with

Controllers obtained…………………………………………………………….58

**LIST OF TABLES**

Table 4.1.1- Alpha (Routh) Table…………………………………………………………....17

Table 4.1.2- Beta Table………………………………………………………………………18

**LIST OF SYMBOLS**

|  |  |
| --- | --- |
| **Symbols** | **Explanation** |
| H(s)  R(s)  C(s)  β  α  Re(s) | Transfer function of original transfer function  Transfer function of reduced model  Transfer function of controller  Numerator routh convergent parameter  Denominator routh convergent parameter  Set of real rational functions  Numerator of original transfer function  Denominator of original transfer function  Numerator of transfer function of reduced model  Denominator of transfer function of reduced model |

**CHAPTER-1**

**INTRODUCTION**

Every physical system can be described by a set of differential (or difference) equations, referred to as the mathematical model of that particular system. This model can be obtained either from basic physical principles or from a series of experiments. This mathematical modeling of systems often gives differential equations of higher order which are difficult to analyze. A measure of the complexity of the system model is the number of first order equations used in its description, which is often referred to as the ‘***order***’ of the system.

**1.1 What is Model Order Reduction?**

In modeling physical systems, the order of the system gives an idea of the measure of accuracy of the modeling of the system. The higher the order, the more accurate the model can be in describing the physical system. But in several cases, the amount of information contained in a complex model may obfuscate simple, insightful behaviors, which can be better captured and explored by a model with a much lesser order. Thus by approximating a higher order system to a suitable lower order system, we can have a much better understanding of the system.

Hence, the process of Model Order Reduction involves studying the properties of a complex dynamic system in application for reducing its complexity, while preserving (to the maximum possible extent) its input – output behavior. Depending upon the practical application, we may have to preserve certain specific properties of the higher order complex system into the new reduced order model. There are several model reduction techniques which are flexible enough to capture the essential behaviors of the complex system as per the demand of the practical situation. Therefore, the most important issues of a method for a model order reduction technique can be summarized as follows:

* **Accuracy**: The reduced model should retain the characteristics of the original system as close as possible in the interested region of operation.
* **Simplicity**: The computation involved in the algorithm for finding reduced model should be very simple.
* **Stability**: The simplified reduced model should be stable, if the original system, in case the original model is unstable, the unstable modes should be segregated and then incorporated with reduced model of stable part.

**1.2 Why Model Order Reduction**

Reduced order modelling aims at studying the nature and the dynamic response of a complex higher order system, and then copies the essential features of the complex system into the reduced order model.

Some of the reasons for using reduced order models of higher order linear systems are:

* *For better understanding of the complex system* - A system of uncomfortably high-order poses difficulty in its analysis, synthesis, or identification. An obvious method of dealing with such type of system is to approximate them by a low-order systems which reflect the characteristics of original system such as time constant, damping ratio, natural frequency etc.
* *To decrease the computational complexity* - The developments of state space methods and optimal control techniques have made the design of control systems for higher-order multivariable systems quite feasible. When the order of the systems becomes high, special numerical techniques are required to permit the calculation to be done at a reasonable cost on fast digital computer. This saves both time and memory required by computer.
* *To reduce the hardware complexity in realization of the system* - A control systems design for a high-order system is likely to be very complicated and of a high-order itself. This is particularly true for controller based on optimal control theory. Controllers designed on the basis of low-order model will become more reliable, less costly and easy to implement and maintain.

There are several methods prevalent to achieve this model reduction of highly complex systems. Each of these methods has its own significant advantages in generating reduced order models. The Routh approximation method and Routh Pade approximation(Conventional methods for reducing order ) have been dealt with in detail .Genetic algorithm based reduction of higher order is analysed and is compared with respect to conventional methods to get an idea how good reduced models they generate.

The design of suboptimal control via reduced-order model is also reported in this project. The proposed method is to design a controller for the higher order system via reduced model. The controller design is carried out using only reduced model. The method uses the well known technique of parameterization of all compensator that stabilizes a given plant. This technique ensures that, when the higher order system and controller are placed in closed loop, the overall system will be internally stable.

**1.3 Application of reduced order modelling:**

Reduced-order models and reduction techniques have been widely used for the analysis and synthesis of high-order systems. Some of the typical applications are listed below:

(i) Prediction of the transient response sensitivity of high-order systems using low-order model.

(ii)Prediction of the transient response sensitivity of high-order systems using low-order equivalents.

(iii) Control-systems design.

(iv)Adaptive control using low-order models.

(v) Designing reduced-order estimators.

(vi)Sub optimal control derived by simplified models.

**CHAPTER-2**

**LITERATURE REVIEW**

During the last three decades much effort has been made to solve the problem of model reduction for large-scale systems.

The model order reduction technique can be broadly classified as: Classical reduction method and modified reduction method.

The first group is the **classical reduction method (CRM)** which is based upon the classical theories of mathematical approximation or mathematical concept such as the:

**Pade approximation** [1]-

An approach to order reduction is the Pade approximation method in which the orders of the numerator and denominator of the approximant are selected in advance and the coefficients are chosen so that the Taylor series expansions of the approximant and of the transfer function approximated agree in as many terms as possible. Determining the coefficients of the approximant does not entail locating the poles of the original transfer function, but the method [2] can only be applied to single-input, single-output systems. Another serious shortcoming [3] of the Pade approximation method is that the poles of the approximant depend on both the numerator and the denominator of the original transfer function, and hence may result in an unstable approximant for a stable system [4].

**Continued-fraction method** ( Cauer form) [5]-

Basically, a transfer function can be expanded into the Cauer first or second form of continued

fraction expansion. If a control or instrumentation system has a low-pass filter form, then the continued fraction expansion has to be started from the constant terms, which gives the Cauer second form. On the other hand, if a system is of high-pass filter form, then the continued fraction expansion can be performed by starting from highest order terms, which yields the Cauer first form. The following are two generalized approaches of continued fraction expansion derived from the extension of either the Cauer first or second form [6], by which differentcombinations of matching time moments and Markov parameters can be achieved [7,8].The continued-fraction expansion(CFE) method for obtaining reduced order models has the disadvantage that the reduced model may be unstable although the original system isstable [9,11].

**Factor division algorithm [**12]

A useful algorithm for linear-system reduction has been given which was initially conceived as an alternative approach [13] for retaining dominant poles in the reduced model. It avoids calculating the initial time moments of the system and solving the Pade equations. If the system poles are known (usually the case for a useful comparison of time responses), or at least those to be neglected, then factor division is often quicker to use than the method of Shamash [1]. However, if such information is not available, the poles to be neglected (usually of largest magnitude) have to be found before division is carried out.

The algorithm is also seen to be useful in other methods where the reduced denominator is calculated first and initial time moments are to be preserved. Again this avoids solving Pade equations or inverting a continued fraction to obtain the reduced numerator. Finally, it should be noted that Markov parameters [14] as well as time moments can be preserved by simply dividing the factors from the highest powers of s first [15].

**Dominant mode method** [6]-

The fundamental idea of dominant-mode approach is to preserve dominant eigen values and to neglect those eigen values which are farthest from the origin. The reason is that the unretained eigen values are important only at the beginning of the- system response, whereas the retained eigen values are important throughout the whole of the response.

**Time-moment-matching method** [16,17]-

It zeros the transfer function of the error system for specific frequencies. Approximation errors expressed in terms of the norm of the error system is considered.

It can be proved that all the CRM approaches are equivalent to each other.

There are, however two serious drawbacks which limit the applications of CRM:

1) The reduced model obtained by CRM may be unstable although the original system is stable.

2) The low accuracy in the mid- and high-frequency ranges [18].

The second group is a development from CRM. and can be called the **modified reduction** **method (MRM).** The best knownMRMis the **stability-criterion method** **(SCM):** in which the characteristic equation of the reduced model is assigned beforehand to satisfy one of some criteria of stability such as the Routh stability criterion, Hurwitz polynomial [19], Mihailov criterion the stability equation.

**Routh stability criterion** [20,21]

The basic idea underlying this method [22] is to develop the well known “Routh table” for the original system and then to construct the approximant in such a manner that the coefficients of its Routh table agree, up to a given order, with those of the original system. The Routh approximant of a stable system is stable. The Routh approximation method preserves stability.

The general rational function that is the difference between numerator and denominator order not less than one can also be reduced by slight modification in the beta table proposed in Routh approximants of arbitrary order[23].It can be applied to both time and frequency domain[24,25].

**Mihailov stability criterion** [26]-

The brief procedure for getting reduced order polynomials using Mihailov criterion is as follows:

Substituting s = j in polynomial D(s) and separating into real and imaginary parts as

D(j)=R()+jQ()

where is the angular frequency in rad/sec.

If the reduced model is stable, its Mihailov frequency characteristic must intersect k times with abscissa and ordinate alternatively in the same manner as that of the original system. In other words, k roots of R () = 0 and Q () = 0 must be real and positive and alternately distributed along the axis. So, the first k intersecting frequencies 0 ,1 ........., k-1 are kept unchanged and are set to be the roots of R() = 0 and Q () = 0 . By putting the value of computed roots in the polynomial equation the reduced order polynomial is obtained[29]

**Weighted impulse response method** [27]-

This is comparatively new method for model reduction SISOlinear continuous systems. It is based on the weighted impulse response Gramians [28]**.**These Gramians represent the input output behaviour of the system and arc system invariants. Further, two Gramians corresponding to successive r(order of the weighting factor) form a positive definite solution to the Lyapunov equation with the system in controllability canonical form. The model reduction algorithm is based on approximating the weighted performance criteria which is expressed as quadratic form involving weighted impulse-response Gramian. The method proposed leads to a family of stable reduced order models for different values of r***,*** out of which the one corresponding to r**=** 0 matches exactly the norm of the impulse response and its l-1 derivatives and the first lMarkov parameters[8]. Further, it matches the first time moment when singular perturbation technique is used. The models obtained using this technique for small values of rare found to be generally very close to the model obtained by balanced realisation techniques. Although the balanced realization is very powerful compared to the other model reduction techniques, the proposed method can be useful when it is required to match exactly Markov, covariance and steady-state parameters of the original system.

The parameters in the numerator are adjusted as in CRM to improve the degree of accuracy at the low-frequency range. However, the absolute stability of the SCM is achieved only at the cost of a serious loss of accuracy.

In recent years, one of the most promising research fields has been “**Evolutionary Techniques**”[30], an area utilizing analogies with nature or social systems. Evolutionary techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable objective functions.

Mainly this can be categorized as :

**Genetic algorithm**[31]based-It is a procedure to find approximate solutions to search problems through evolutionary biology.

**Particle swarm optimization** (PSO) based[32]- PSO is a population-based stochastic optimization technique,inspired by social behavior of bird flocking or fish schooling[33].

The focus of my project will be Genetic algorithm based model order reduction[34]-[35] as compared to conventional approaches for model order reduction[36].

**Model reduction by Least square method by using gentic algorithm** [37]

The denominator parameters and time delay of the reduced-order model are coded into binary bit strings and searched by the GA, while the numerator parameters are calculated by the Least Square method for each of candidates of the denominator parameters and time delay[38]. Thus, all the best parameters and time delay of the reduced-order model can be obtained by repeating genetic operations of strings.

**A frequency-based model order reduction (MOR) via genetic algorithm (GA)** [39]

GA predicts the elements of the system state matrix [A] defined in a state space representation along with the elements of the [B] and [C] matrices of the reduced order model. As a frequency-based MOR technique, the GA predicts only the elements of the [B] and [C] matrices of the reduced order model while [A] is set in the modal form.

The proposed GA reduction method is based on the minimization of the mean square error criterion and maximization of the fitness function. GA-based method maintains system stability, preserve the exact dominant frequencies, and providing system response with minimum deviation.

**Tree bond graph and genetic algorithm based** [**40**]

This technique describes a unique technique of generating the Genetic design from the tree structured transfer function obtained from Bond Graph. This combines bond graphs for model representation with Genetic programming for exploring different ideas on design space tree structured transfer function result from replacing typical bond graph element with their impedance equivalent specifying impedance lows for Bond Graph multiport. This tree structure is used for model order reduction.

**Combined methods**

For example:

* The denominator of the reduced order model is obtained by the stability equation method and the numerator terms of the lower order transfer function are determined by minimizing the integral square error[38] between the transient responses of original and reduced order models using Genetic algorithm[41]. The algorithm has several advantages, e.g. the reduced order models retain the steady-state value and stability of the original system. The proposed algorithm has also been extended for the order reduction of linear multivariable systems.
* The denominator polynomial of the reduced order model is determined by using the Mihailov criterion [26] while the numerator coefficients are computed by minimizing the integral square error between the original and the reduced system using Genetic Algorithm. This method [42] guarantees stability of the reduced model, if the original high order system is stable.

One of the main objectives of order reduction is to design a controller of lower order which can effectively control the original high order system so that the overall system is of lower order and easy to understand. There are two common approaches for controller design. First approach is to obtain the controller on the basis of reduced order model called process reduction [43, 44](which is presented here in this thesis).

In the second approach, the controller is designed for the original higher order system and then the closed loop response of higher order controller with original system is reduced pertaining to unity feedback called controller reduction [45].Both the approaches have their own advantages and disadvantages. The process reduction approach is computationally simpler as it deals with lower order models and controller but at the same time errors are introduced in the design process as the reduction is carried out at the early stages of design. In the controller reduction approach error propagation is minimized as the design process carried out at the final stages of reduction but the approaches deals with higher order models and thus introduces computational complexity.

**CHAPTER-3**

**DESCRIPTION OF ROUTH PADE APPROXIMATION METHOD**

A mixed method is proposed for finding stable reduced-order models using the Pade approximation technique and the Routh-Hurwitz array. This method guarantees stability of the reduced model when the original system is stable.

The mixed method for deriving stable low-order equivalents of high-order systems, as given in this letter, is computationally easy to program and conceptually simple. It combines the Pade approximation technique and the Routh- Hurwitz array method.

**3.1 Reduction method**:

Let the high-order transfer function H(s)be given by

…. (3.1.1)

= .… (3.1.2)

Where *m < n* and eqn. 2 is the power series expansion of eqn. 3.1.1 about s = 0.

*Step1:*

Form the Routh-Hurwitz stability array for the denominator polynomial in eqn. 1:

….(3.1.3)

The above array is formed by the well-known algorithm

…. (3.1.4)

for i > 3 and 1 <j <[(n- i + 3)/2], where [.] stands for the integral part of the quantity. A polynomial of lower order *k* may be easily constructed with the (n+1-k) th and (n + 2-k)th rows of the above array. Thus a transfer function of reduced order kmay be written as

where the coefficients of the kthorder denominator polynomial are found from eqn. 3.1.3. Rk(s)may be rewritten as:

.…(3.1.5)

where the bcoefficients are now known.

***Step 2****:* For Rk(s)of eqn. (3.1.5 )to be the Pade approximant of H(s),we have

a0 =b0c0

a1=b0c1+ b1c0

…………………………………

ak-1= b0ck-1+ b1ck-2 b0ck-1+ b1ck-2 +….+bk-2ck+ bkc0  ….(3.1.6)

The aj ( j = 0, 1, 2, ..., k - 1) can be found by solving the above kequations.

**3.2 Numerical example for order reduction using Routh Pade approximation**

To illustrate the calculations entailed in the Routh Pade approximation (Conventional method) the fourth order transfer function is taken and has been reduced to second order.

Let us consider the system described by the transfer function :

for which a second order reduced model R (s) is desired.

**Calculation of Routh Pade Approximation**

H(s) = …. (3.2.1)

*Step1*: The Routh Hurwitz array of denominator

1 35 24

10 50

30 24

42

24

The second order reduced polynomial from Routh Hurwitz array

=Rd(s) = 30s2 + 42s + 24

=b2s2 + b1s +b0 ...(3.2.2)

*Step2:* The taylor series expansion of H(s)

= 1 - (13/12)s +(157/144)s2 – (1843/1728)s3+……………

= c0 + c1s + c2s2+c3s3+…………………….. …(3.2.3)

R(s) =

a0 = b0c0

a1=b0c1 + b1c0

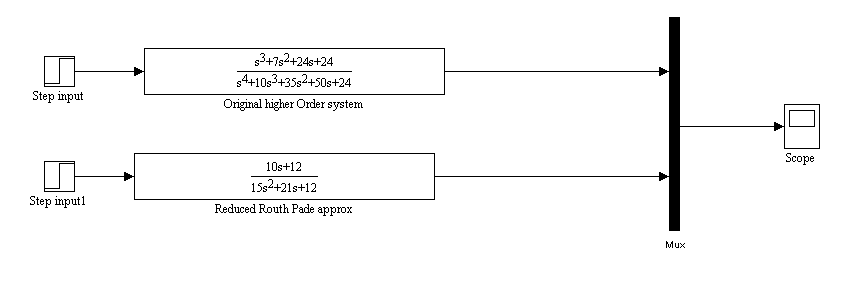
So, a0=24 and a1=20 (From eq 3.2.2 and 3.2.3)

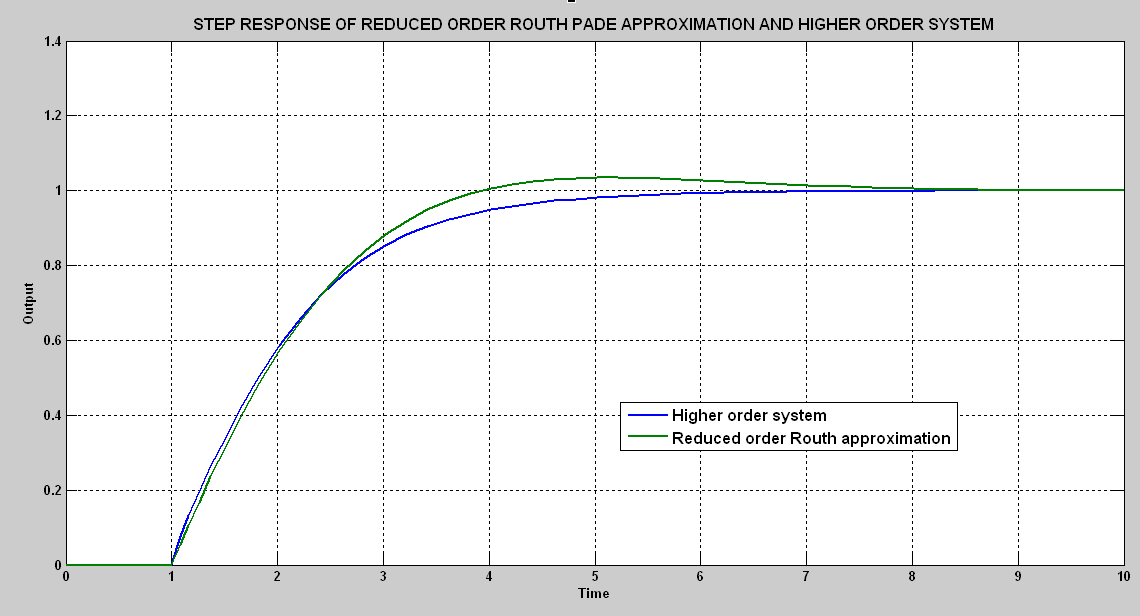
R(s)=(20s+24)/(30s2+42s+24)

Hence, reduced order Routh Pade approximation is:

**R(s)** …. (3.2.4)

**SIMULATION RESULT:**

****

****

**Fig 3.2.1**

**CHAPTER-4**

### DESCRIPTION OF ROUTH APPROXIMATION METHOD

Consider a linear, time-invariant (single-input, single-output) system having the transfer function

H(s) = ...(4.1)

A linear system with loutputs and m inputs can be represented by a matrix of transfer functions of the form (1) with the numerator coefficients bibeing l x mmatrices. Since the denominator of a Routh approximant depends only on the denominator of H(s),a Routh approximant to an l-output, m-input system is computed by computing the Routh approximant for each term in the matrix of transfer functions. The denominator of the Routh approximant only has to be computed once, since it is the same for each term.

#### 4.1 Alpha-Beta Expansion

Atransfer function of the form (4.1)that is asymptotically stable can always be expanded in the following canonical form:

...(4.1.1)

where βi(i = 1,2, . . ,n)are constants and the Fifor i = 2,. . ,nare defined by the continued fraction expansions

For F1(s),definition (4.1.2)is modified slightly; the first term in the continued fraction expansion is 1+ α1sinstead of α1s. The canonical form (4.1.1) is referred as Routh approximations.

Ablock-diagram representation of this canonical expansion is given in Fig. 4.1.1.

**Fig 4.1.1- Alpha Beta Canonical expansion**

OUTPUT

+

+

+

+\_

+

….

+

+

+

+

+

-\_\_--\_\_\_--\_--\_

INPUT

…

+\_

+\_

+

..…

+

\_

\_

\_

-\_\_--\_\_\_--\_--\_

+\_

….

+\_

…

**Table 4.1.1** - **Alpha (Routh) Table**

|  |  |
| --- | --- |
|  | *…* |
| *………* | *….*    *…..*  *…..*  *……*  *…….*  *……….* |

The ncoefficients αiappearing in the alpha-beta expansion can be computed using the algorithm for constructing the Routh table used in stability analysis as shown in alpha table (Table 4.1.1) . The first two rows of the table are formed from the coefficients of the denominator of H(s)where by assumption the entries aj0=aj-11=0 for j>n.The remaining entries are formed by the cross multiplication rule.

:

; i=1,…..(n-1) …(4.1.3)

For n-i odd, the last equation in (4.4) is replaced by

…(4.1.4)

The αi are marginal entries given by

, i=1,2,….,n …(4.1.5)

The use of the Routh table for the development of the canonical form used in the approximations to be developed suggests that the resulting approximations be called “Routh approximations.”

**Table 4.1.2 Beta Table**

|  |  |
| --- | --- |
|  | …  … |
| ……… | …..    …..  …….  ……..  ……… .  ………. |

The coefficients βiappearing in the canonical form can also be obtained by use of a tabular algorithm as shown in Table 4.1.2.The first two rows of the βtable are obtained from the coefficients of the numerator of H(s).The remaining coefficients βientries are computed from entries in the Routh table computed as shown in Table 4.1.1 and the earlier rows of the beta table, using the following recursion:

, i=1,2,….n ….(4.1.6)

And

.…(4.1.7)

i = 1, 2,…, n-2.

The canonical expansion (4.1.1) defines a pair of finite, recursive algorithms: one algorithm (4.1.3)-(4.1.5) for computing nalpha coefficients αi from the denominator of the transfer function, and another algorithm (4.1.6) and (4.1.7) for computing n beta coefficients βifrom both the numerator and denominator of the transfer function. Since H(s)in (4.1.1) is assumed to be asymptotically stable (i.e. all poles a0i, i = 0.1,. . , n in the first column of the alpha (Routh) table are non zero and have the same sign. Consequently, division by zero in (4.1.5) cannot occur, and the alpha-beta expansion, or equivalently, the computations given by (4.1.3)-(4.1.7)can always be performed. Next, the reverse problem of computing the numerator and denominator polynomials of the transfer function from the alpha-beta expansion will be solved by introducing the Routh convergents.

**4.2 Routh Convergents**

The kth Routh convergent Rk(s) for the transfer function H(s) is obtained by truncating the alpha-beta expansion(4.1.1) and arranging the results as a rational function of s. The truncation eliminates those terms in the alpha-beta expansion containing αk+1,…αn,βk+1,….βn and hence depends only on the first k alpha and beta coefficients by the continued fraction expansions for i=2,………,k :

For i= 1, the above definition is modified slightly; the first term in the continued fraction expansion is 1 + α1s instead of α1s. Acomparison to (4.1.2)shows that Gi,kis the result of truncating the last (n– k) terms in the continued fraction expansion of Fi.Utilizing these new functions Gi,k***,*** the kth Routh convergent is equal to

(s)…(s)

.…(4.2.1)

Let **Ak(s)**and **Bk(s)**denote the denominator and numerator, respectively, of the kth Routh convergents, i.e.,

A1(s)=α1s+1

B1(s)=β1

A2(s)=α1α2s2+α2s+1

B2(s)=α2β1s+β2

A3(s)=α1α2α3s3+ α2α3s2+(α1+α3)s+1

B3(s)=α2α3βs2+α3β2s+(β1+β3)

-

-

***-***

More generally as shown in (4.2.1)

....(4.2.2)

, k=1,2,…. ....(4.2.3)

with

In itself, the Routh convergent Rk(s**)**is an approximation to H(s)which tends to preserve high-frequency behavior. For control applications, it is preferable to obtain a low-frequency approximation obtained as explained in next section.

#### 4.3 Reciprocal Transformation

The Routh approximation makes use of a transfer function Ĥ(s)related to H(s)by the following transformation

….(4.3.1)

which is merely the operation of reversing the order of the polynomial coefficients. It is obvious from (4.3.1) that the second application of the reciprocal transformation transforms the reciprocal transfer function back to the original transfer function. The primary property of the reciprocal transformation is that it inverts the non zero poles (or eigen values) and zeros of the original transfer function. For example, if siis a pole (zero) of H(s)then l/si*,* is a pole (zero) of Ĥ(s).

The transfer function Ĥ(s)can be called the "reciprocate" of H(s).

#### 4.4 Routh Approximation

The kth-order Routh approximant of H(s)is defined as the reciprocate of the kth-order Routh convergent of the reciprocate of H(s).Denoting the kth-order Routh approximant by Hk(s)gives

....(4.4.1.)

where,is the kth-order Routh convergent of Ĥ(s).

Computation of the Routh approximation thus entails four steps:

* Apply the reciprocal transformation to H(s)to obtain Ĥ(s) as shown in (4.3.1).
* To compute the alpha-beta expansion of Ĥ(s),which in accordance with (4.1.1) and (4.1.2).

represents the transfer function as a function of sand 2ncoefficients, i.e.,

…,…,

The symbol '' ^" placed above the alpha and beta coefficients signifies that these are computed from the reciprocal transfer function Ĥ(s)and are different from the alpha and beta coefficients of H(s).

* The third step is to compute the kth Routh convergent Rk(s)from the first *k* alpha and beta coefficients of Ĥ(s) according to definition (9).

Truncating the alpha-beta expansion after the first k terms gives, after summing the terms and arranging as a rational function, the kth Routh convergent in the form

….(4.4.2)

Equivalently, (4.4.2) can be computed from using the recursive formulas (4.2.2) and (4.2.3).

* The fourth step is to apply the reciprocal transformation to Rk(s).The resulting transfer function obtained from(4.4.2) is

…..(4.4.3)

and is the desired kth Routh approximant

**4.5 Properties of Routh Approximation**

The Routh approximation has a number of properties that make it useful for the representation of high-order linear systems. These properties include the following:

***Stability Preservation***

If H(s)is the transfer function of an asymptotically stable system then each approximant Hk(s)is asymptotically stable. This property holds because of the very method used in developing the approximation and in fact, is the motivation of this method. It is important to note, however, that Routh approximants of an unstable system could turn out to be stable.

***Impulse Response Energy***

Let h(t)be the impulse response (i.e., the inverse Laplace transform) of H(s),and let the "impulse response energy" be defined by the integral

.…(4.5.1)

assuming, of course, that H(s)is the transfer function of a stable system. The

square root of the impulse response energy ||h|| defines a norm [10], [ 11] of the function h(t)

….(4.5.2)

and it is proved that ||h|| is invariant under a reciprocal transformation ,i.e,

.…(4.5.3)

where and are the parameters of the alpha –beta expansion of the reciprocate Ĥ(s) of H(s*).*

Let hk(t)be the impulse response of the kth-order Routh approximant, i.e., the inverse Laplace transform of Hk(s).Then from (4.18),

….(4.5.4)

Since all the of a stable system are positive, it follows that

0≤||h1||≤||h2||≤……≤|| hn || = ||h|| ....(4.5.5)

hence, the (square root of) impulse-response energies of the Routh approximants converge monotonically to the (square root of) impulse response energy of the original system.

The ratio ||hk||/|| h||provides a useful technique for selecting the order of the approximation to be used. For instance, the criterion might be to pick the smallest k for which the rms value ||hk||is greater than (say) 90 percent of the total.

**Locations of Poles and Zeros**

Another useful property of the Routh approximation is that the poles and zeros of the approximants approach the poles and zeros of the original function as the order of the approximation is increased. For a system having only real poles at

λ1≥ λ2 ………..≥λn ,it has been found that the poles of the approximants "straddle" the actual poles in the following sense:

λi(i+1)≤ λi(i+3)≤………. λi(n) = λi≤………≤ λi(i+2)≤ λi(i)

for i odd

λi(i)≤ λi(i+2)≤……….. λi(n)= λi(i)≤………. λi(i+3)≤ λi(i+1)

for i even

where λi(k)(i = 1,2, . ,k) are the poles of the kth-order Routh approximation *Hk(s)*to the original transfer function. (For all negative real poles, λ1, is the dominant mode located closest to the origin and λ n is furthest from the origin.)

**4.6 Numerical example for order reduction using Routh approximation**

To illustrate the calculations entailed in the Routh approximation (Conventional method) the fourth order transfer function (same as in eq3.2.1) is taken and has been reduced to second order.

Let us consider the system described by the transfer function transfer :

…(4.6.1)

for which a second order reduced model R (s) is desired.

**Calculation of Routh approximants**

H(s) =

*Step1:* Applying the reciprocal transformation to H(s) ,we get

Ĥ(s) =

*Step2*: Compute the alpha table and beta table (From Table 4.1.12 and Table 4.1.2)

Alpha Table for H(s)

|  |  |
| --- | --- |
|  | =24 = 35 |
|  | =30.2 =1      = 8.34437 |

Beta Table for H(s)

|  |  |
| --- | --- |
|  | = 24 = 7    = 24 =1 |
| = 0.7947 | = 2.2    = 0.2053 |

*Step 3:* Routh convergent is calculated .Substituting values of β1,α1,α2,β2 into pair of recursive Equations (4.2.2) and (4.2.3) gives the second Routh convergent of the reciprocal transfer function is

A2(s) = α1 α2s2 + α2s + 1

A2(s) = 0.48 X 1.65563s2 +1.65563s +1

=0.7947024s2 + 1.65563s + 1

B2(s)= α2 β1s+ β1

=1.65563 X 0.48s + 0.7947

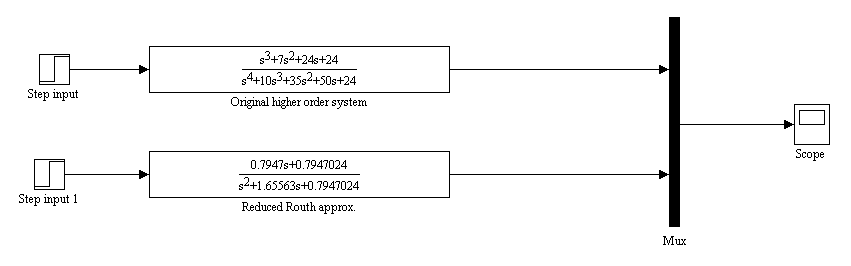
=0.7947024s + 0.7947

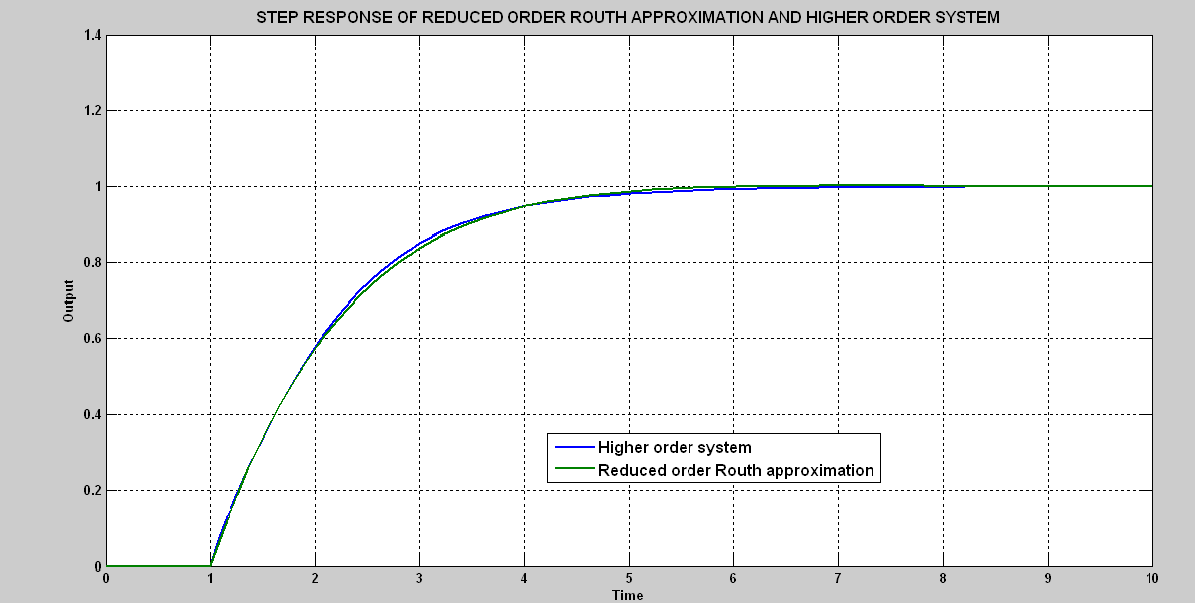
R2(s) = (0.7947024s + 0.7947) /(0.7947024s2 + 1.65563s + 1)

*Step 4:* Routh approximant obtained by reciprocal transformation of R2 is

….(4.6.2)

**SIMULATION RESULT:**

****

****

**Fig 4.6.1**

**CHAPTER-5**

**MODEL ORDER REDUCTION USING GENETIC ALGORITHM**

A **genetic algorithm (GA)** is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic Algorithms is a very recently developed unorthodox search and optimization algorithm. This technique is particularly good at taking larger search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a life time. This method is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population evolves toward an optimal solution.



**Fig 5.1Problem solution using Genetic algorithm**

**5.1 Methodology**

The genetic algorithms works on the principle of Darwinian theory of survival of the fittest.In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encodecandidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolve towards bettersolution. GA performs directed random searches through a given set of alternatives with the aim of finding the best alternative with respect to the given fitness function. The fitness function is defined over the genetic representation and measures the qualityof the represented solution.

The fitness function is always problem dependent.

A typical genetic algorithm requires:

1. a genetic representation of the solution domain,

2. a fitness function to evaluate the solution domain.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions (usually randomly) and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

**5.1.1 Initialization**

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, allowing the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

**5.1.2 Selection**

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-basedprocess, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. From amongst this population ‘parents’ are selected based on their fitness value for reproduction.

**5.1.3 Reproduction**

The next step is to generate a second generation population of solutions .For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. The parents are then used to produce children for the next population by either of the three ways namely elite children, crossover children or mutation children.

***Elite*** children are the individuals in the current generation with the best fitness values. These individuals automatically survive to the next generation. ***Crossover*** children are created by combining the vectors of a pair of parents i.e., a linear combination of the parents generates a new individual for next iteration. ***Mutation*** children are created by introducing random changes, or mutations, to a single parent. These random changes can be in the form of expansion factor or contraction factor over the parent depending on the value of the fitness function.

By producing a "child" solution using the above methods, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. So, these new children become the next generation for polling.

**5.1.4 Termination**

This process continues until the algorithm moves to the minima or one of the stopping criteria is met.

Common terminating conditions are:

• A solution is found that satisfies minimum criteria.

• Fixed number of generations reached.

• Allocated budget (computation time/money) reached.

• The highest ranking solution's fitness is reaching or has reached a plateau such that successive

iterations no longer produce better results.

• Manual inspection

• Combinations of the above.

The individual which has the least fitness value in the final population is taken as the optimal solution.

The flow chart below shows a very simplified version of the algorithm with the stopping criteria as maximum generations.

Gen = 0

Initialize Population

Objective Function Evaluation

YES

Gen > Max. Gen

NO

Reproduction

Crossover

Gen = Gen + 1

Mutation

**Fig 5.1.1Flowchart of Genetic Algorithm**

In a particular iteration of GA, the value of the fitness function is calculated for all the children and a *successful poll* is one in which a child has a better fitness value. This child automatically gets selected for the next generation. In such a case, the mesh span is increased by a certain factor called expansion factor to search for better children in future iterations if possible. An *unsuccessful poll* is one in which none of the children have a better fitness value. This essentially implies that the local minima for the function is near the present individual and so a contraction factor is introduced to reduce the mesh size. In general, the run time of the algorithm increases exponentially with the number of variables in GA.

**5.2 Fitness Function**

Let the given higher order system (HOS), H(s) of order ‘n’ be given by:

…. (5.2.1)

We attempt to find the best second order approximation of this HOS. Let the reduced order model (ROM) be given by:

Reduced Order model R(s) = …. (5.2.2)

where, the coefficients a, b, c, d and e are to be found out using GA.

For this, we create a fitness function and then run it through GA. The GA then mimics the Darwinian theory of survival of the fittest to find the minima of the fitness function over a wide search space.

If x(t) and y(t) denote the step response of the ROM and HOS respectively, then we create a fitness function as follows :

**Fitness function=2**

.… (5.1.3)

The numerator and denominator coefficients of the reduced order model is determined by minimizing Integral square error between the transient part of step response of original system and reduced system using genetic algorithm. The deviation of the lower order system from the original system response is given by the error index ’ISE’ known as the Integral square error, which is our fitness function.

**5.3 Description of Genetic algorithm in Optimization toolbox of MATLAB**

**5.3.1 Some Genetic Algorithm Terminology and their relevance in Optimization toolbox**

* Fitness Functions

The fitness function is the function we want to optimize. For standard optimization algorithms, this is known as the objective function. The toolbox software tries to find the minimum of the fitness function. We can write the fitness function as an M-file and pass it as a function handle input argument to the main genetic algorithm function.

* Individuals

An individual is any point to which we can apply the fitness function. The value of the fitness function for an individual is its score. An individual is sometimes referred to as a genome and the vector entries of an individual as genes.

* Populations and Generations

A population is an array of individuals. For example, if the size of the population is 100 and the number of variables in the fitness function is 3, we represent the population by a 100-by-3 matrix. The same individual can appear more than once in the population.

At each iteration, the genetic algorithm performs a series of computations on the current population to produce a new population. Each successive population is called a new generation.

* Diversity

Diversity refers to the average distance between individuals in a population. A population has high diversity if the average distance is large; otherwise it has low diversity. Diversity is essential to the genetic algorithm because it enables the algorithm to search a larger region of the space.

* Parents and Children

To create the next generation, the genetic algorithm selects certain individuals in the current population, called parents, and uses them to create individuals in the next generation, called children. Typically, the algorithm is more likely to select parents that have better fitness values

**5.3.2 Outline of how genetic algorithm works in Optimization toolbox**

The following outline summarizes how the genetic algorithm works in Optimization toolbox:

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
   1. Scores each member of the current population by computing its fitness value.
   2. Scales the raw fitness scores to convert them into a more usable range of values.
   3. Selects members, called parents, based on their fitness.
   4. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
   5. Produces children from the parents. Children are produced either by making random changes to a single parent i.e *mutation* or by combining the vector entries of a pair of parents i.e *crossover*.
   6. Replaces the current population with the children to form the next generation.
3. The algorithm stops when one of the stopping criteria is met.

**5.4 Numerical example of order reduction using Genetic algorithm**

To illustrate the calculations entailed in the Routh approximation (Conventional method) the fourth order transfer function(same as in eq3.2.1 and eq4.6.1) is taken and has been reduced to second order.

Let us consider the system described by the transfer function transfer :

….(5.4.1)

for which a second order reduced model R (s) is desired

*Step 1*: Let us assume the reduced order response of H(s) be

R(s) = (Ds+E)/(As2+Bs+C)

*Step2:* The fitness function is integral square error between step response of higher order system and reduced order system is

where h(t) is step response of higher order transfer function H(s)and r(t) is step response of reduced order system R(s).(h(t) and r(t) are in time domain)

h(t) =

Let, r(t) =

*Step3*: Finding the minimum of fitness function till the steady state value is reached the unknown parameters of R(s) i.e can be found out by Genetic algorithm in Optimization toolbox of MATLAB.

Options chosen here for minimizing fitness function:

* Population size - 30
* Creation function – Uniform

It creates random initial population with uniform distribution.

* Scaling functions-Rank

It scales the raw scores based on the rank of each individual. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

* Selection-Stochastic uniform

It lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

* Reproduction- Elite count = 2

- Crossover fraction =0.8

* Mutation function-Gaussian

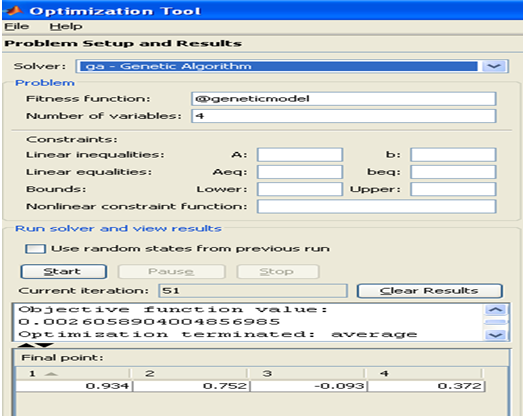
It adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centered on zero.

* Crossover functions-Arithmetic

It creates children that are a random arithmetic mean of two parents, uniformly on the line between the parents

* Stopping criteria-Maximum no of generation = 120

-Stall generation = 50



**Fig 5.4.1 Optimization toolbox** (Reaching final point at the end of 51st iteration)

As seen from the optimization toolbox

Value obtained are:,a=0.752,,b=0.372.

*Step4:* So r(t) can be written as

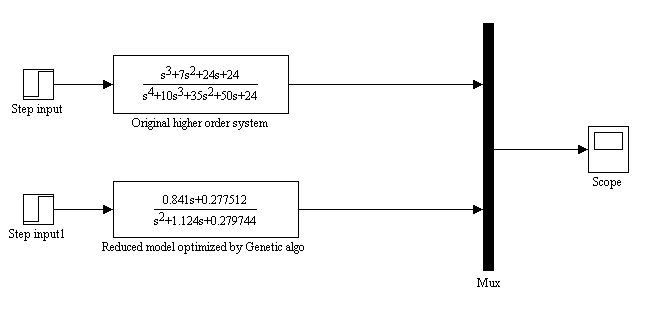
…(5.4.2)

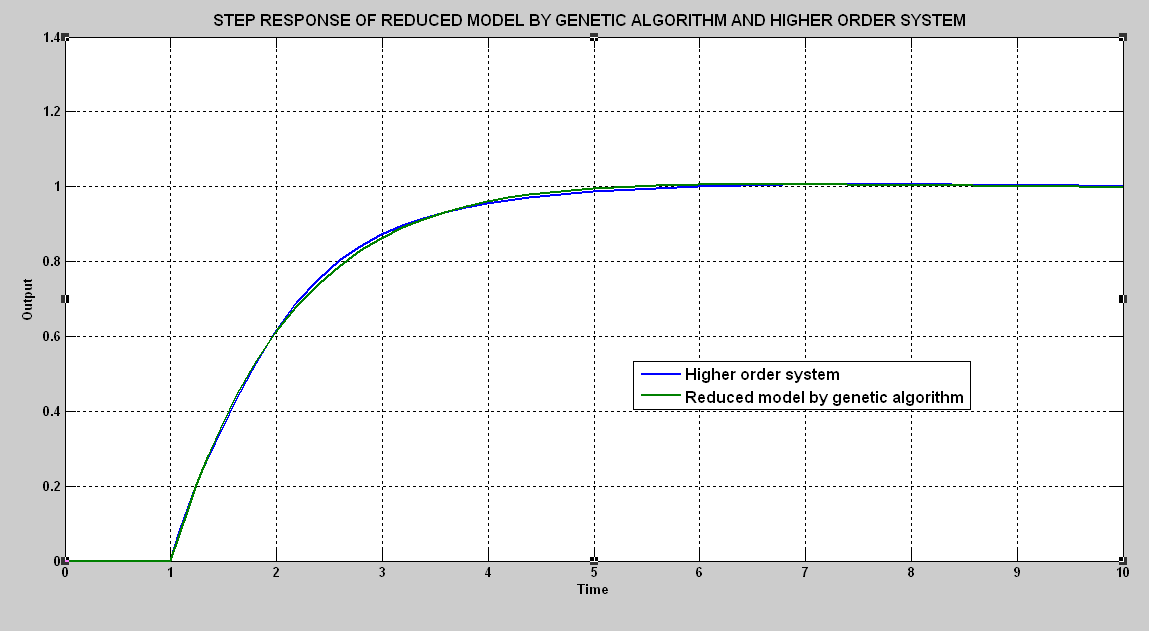
Hence, R(s) =

=

**R(s) =** …(5.4.3)

**SIMULATION RESULT:**

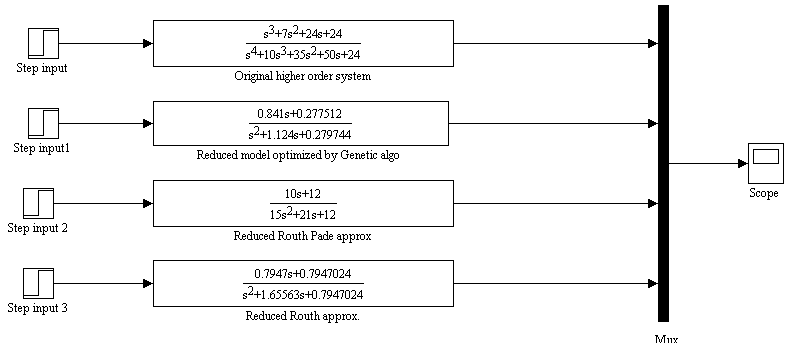
****

****

**Fig.5.4.2**

**5.5 CONCLUSION:**

The step responses of the reduced order approximation obtained from the two conventional methods-Routh Pade and Routh approximation and finally through Genetic algorithm is compared with Original higher order system.

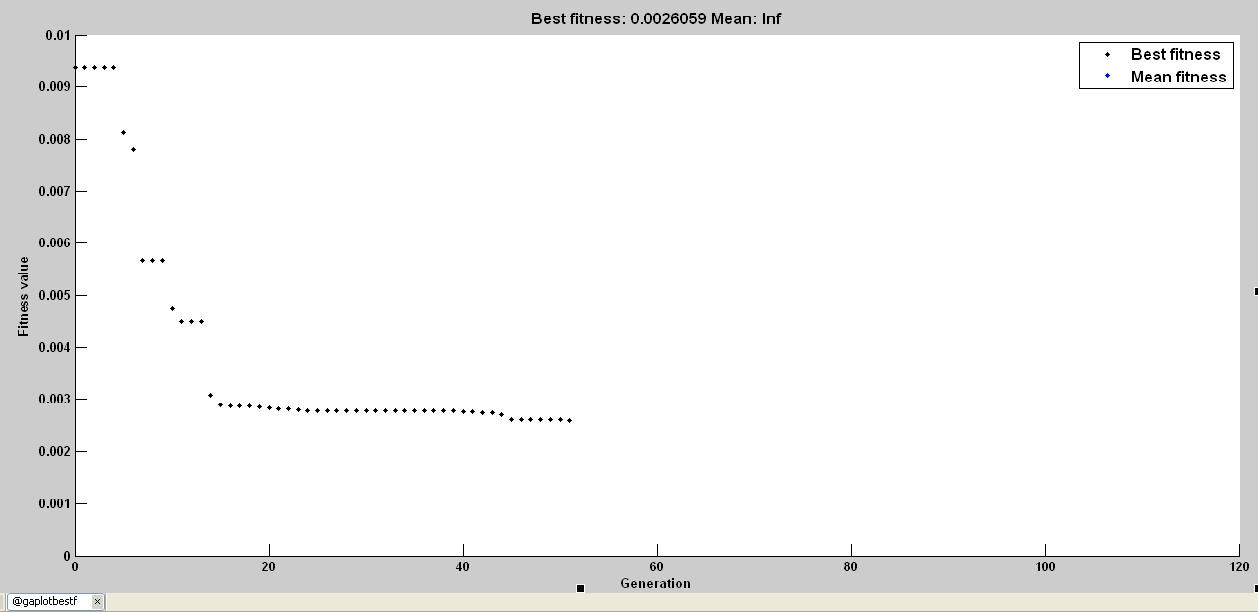


****

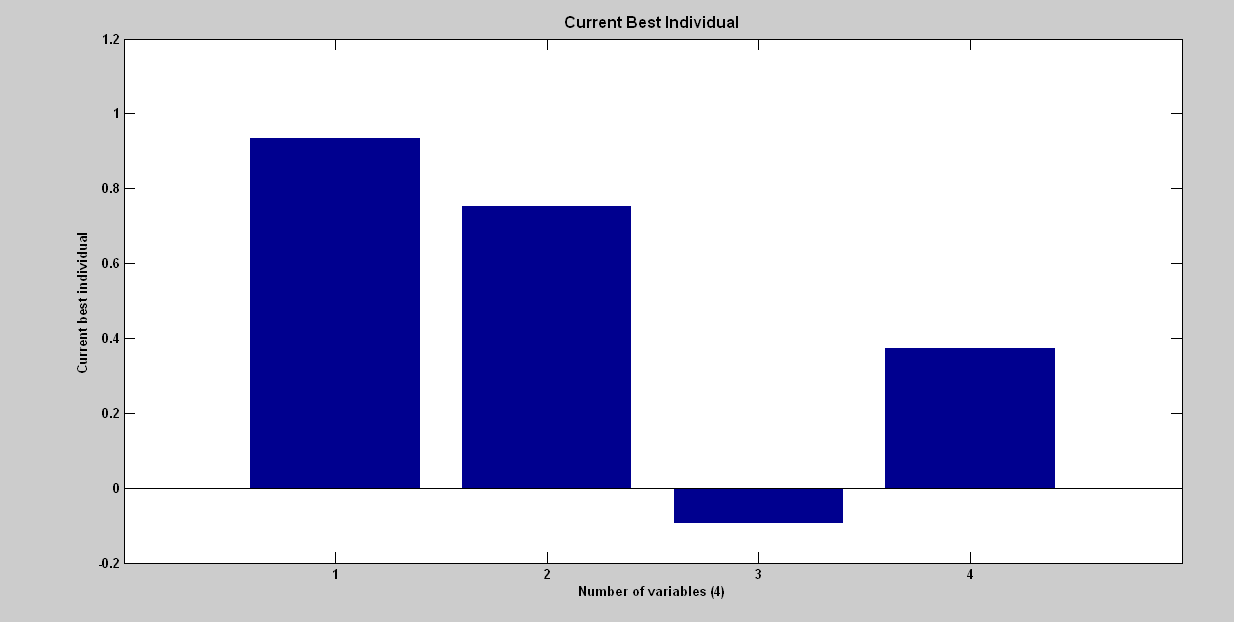
**Fig 5.5.1**

**5.6 PLOTS OF VARIOUS ASPECTS OF GENETIC ALGORITHM EXECUTED:**

* **Best fitness –** It plots the best function value in each generation versus iteration number.

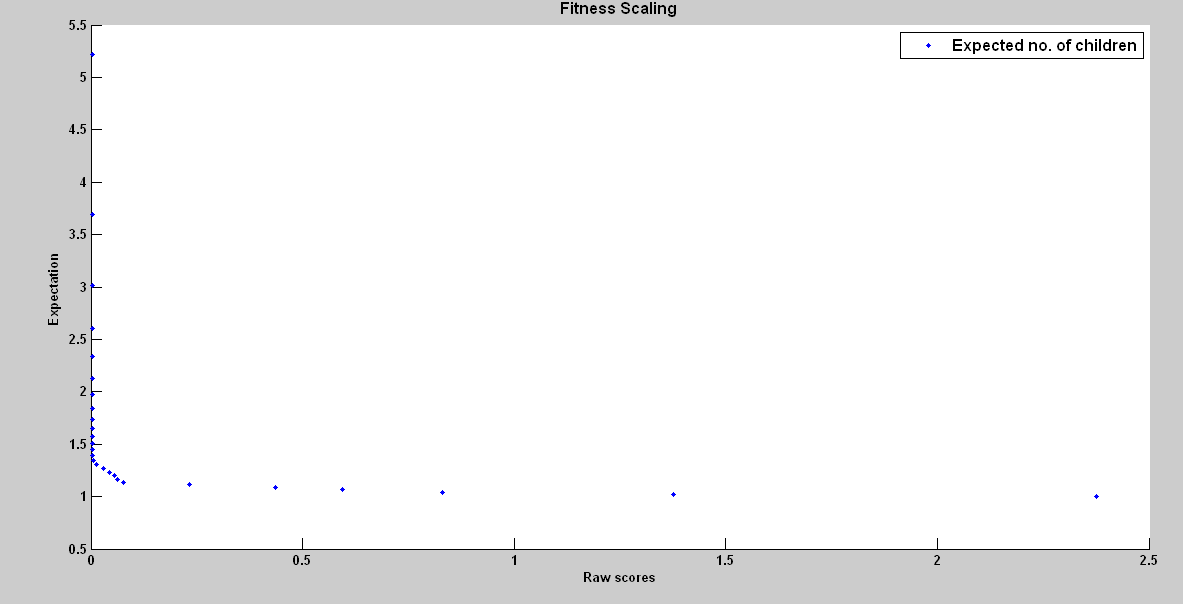
 **Fig. 5.6.1**

* **Best individual** –It plots the vector entries of the individual with the best fitness function value in each generation. The following plot shows the current best individual at final iteration.

\

**Fig. 5.6.2**

* **Fitness scaling-**It plots the expected number of children versus the raw scores at each generation. **Fig5.6.3**

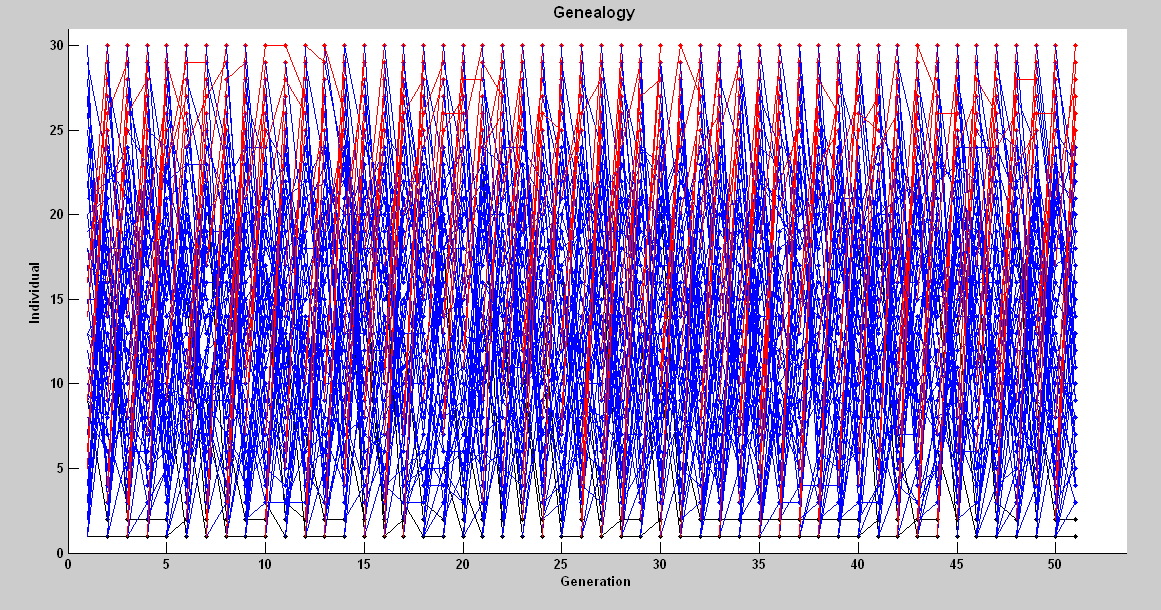


* **Genealogy** –It plots the genealogy (pattern of reproduction) of individuals. Lines from one generation to the next are color-coded as follows:

Red lines indicate mutation children.

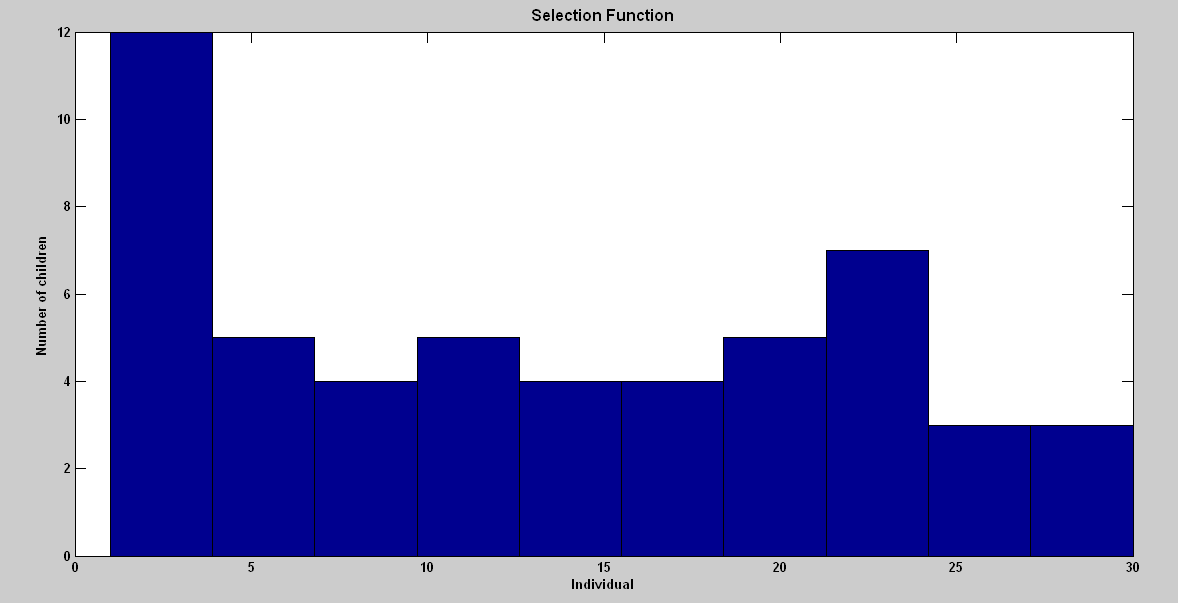
Blue lines indicate crossover children.

Black lines indicate elite individuals. **Fig5.6.4**



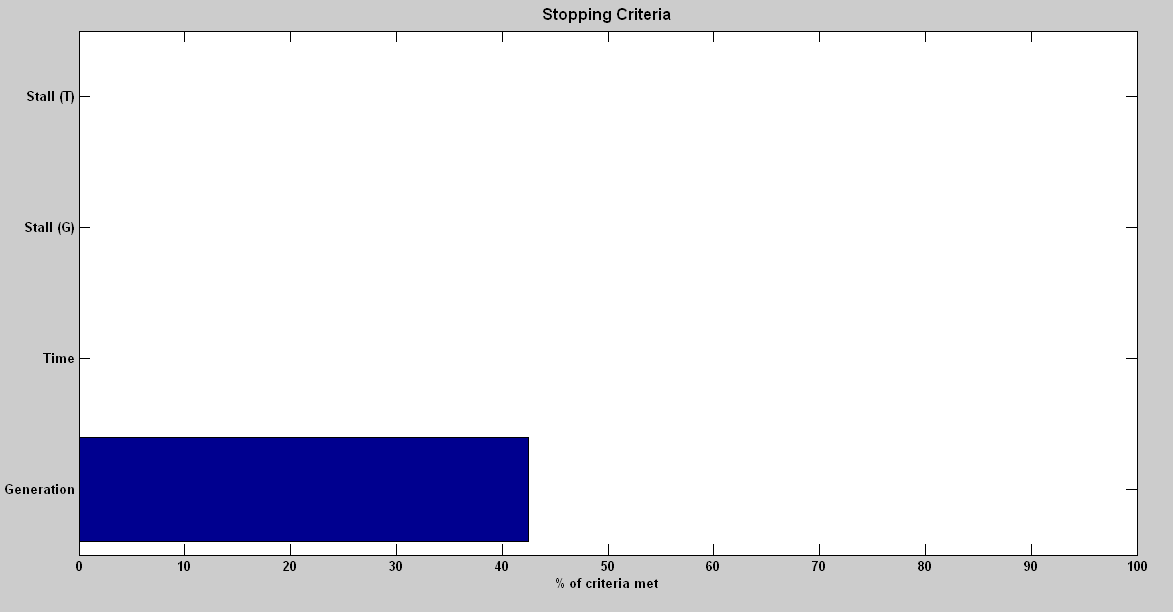
* **Selection** – It plots a histogram of the parents. This shows which parents are contributing to each generation.

**Fig 5.6.5**



* **Stopping**- It plots stopping criteria levels

**Fig5.6.6**

****

**CHAPTER - 6**

**CONTROL OF HIGHER ORDER SYSTEM VIA ITS REDUCED MODEL**

The use of reduced order plant models to design an adequate controller is now an accepted practice among the control engineers, because, the simulation and design of controller of high order system is a difficult problem. The cost and complexity of the controller increases with system’s order. This problem can be overcome if a “good” reduced order model is available for the original higher order system and if it is possible to design a controller using a low order reduced model, which will stabilize the original higher order system when placed in the closed loop.

For any model reduction scheme special care is required to obtain stabilizing controllers from reduced order models. In this project a method is proposed to design a controller for the higher order system via reduced model. The controller design is carried out using only reduced model.

The method uses the well known technique of parameterization of all compensator that stabilizes given plant.

This technique ensures that, when the higher order system and controller are placed in closed loop, the overall system will be internally stable. The chapter is organized as follows: Section 1 describes the problem statement and some mathematical preliminaries. The Section 2and 3 presents the method of controller design using controller parameterization technique. Section 4 is the inference from the procedure described for controller design. Section 5,6 and 7 describes numerical example of controller design via reduced models obtained from the two conventional methods-Routh Pade and Routh approximation and finally through Genetic algorithm.

**6.1 Problem statement**

Let H(s) = ... (6.1.1)

is a stable single-input, single-output (SISO) linear time invariant system of orderwhich represents higher order system and let

R(s) = … (6.1.2)

represents its stable reduced order model of order ‘m’;

where deg np(s) ≤ deg dp(s)and deg nm(s) ≤ deg dm (s) , and there are no pole-zero cancellations.

The problem is to find a controller via reduced model which will stabilize the high order system H(s)and its reduced order model R(s).

The technique of parameterization of all stabilizing controllers for a system is used to design the controller for higher order system via reduced model.

**6.2 Controller Parameterization**

The controller parameterization uses coprime factorization of the plant. The central notion in the coprime factorization approach is that any real rational function can be expressed as a ratio of two transfer function, each of which is BIBO stable and proper, and the two transfer function are coprime. The coprime in this situation means that the transfer functions have no common zeros in the closed, extended right half of the s-plane.

Assume that p(s) ε Re(s)and is expressed as a ratio of rational function as :

**p(s) =** ;where n(s),d(s) ε S.

Let c(s) **ε** Re(s) be a stabilizing controller of p(s)expressed as a ratio of rational functions

C(s) = ; where nc(s),dc(s) **ε** S

The rational functions are selected such that they satisfy Bezout identity,

***nc(s)n(s)* + *dc(s)d(s)* = 1.**

Then the set of all stabilizing controllers of p(s)is given by

**Stab(p(s))** =

...(6.2.1)

where r **ε** S and dc(s)- r n(s) ≠ 0

R(s) is the set of real rational functions in the variable s and S is the subset of R(s)consisting of all rational functions that are bounded at infinity and whose poles all have negative real parts i.e. Sis the set of all proper stable rational functions.

**6.3 Controller design**

In this section, the above controller parameterization is used to derive a controller for higher order system via its reduced order model. As the higher order system is stable we can choose rational functions as:

*n1(s)* = *G(s),d1(s)*= 1, *x1(s)* = *0* and yl(s) = 1,

then they satisfy Bezout identity *x1(s)n1(s)* + *y1(s)d1(s)* = 1. Similarly for reduced order plant,if we choose,

*n2(s)* = *R(s),d2(s)* = 1, *x2(s)*= 0 and y2(s) = 1, then they satisfy Bezout identity .

Sowe can take *G/1* and *R/1* as thefractional factorization of the system and its reduced model respectively. AsGand Rare stable, thereexistsk1and k2(constants) which stabilizes system and model. Let fractional factorization of kland *k2* are k1/1 and k2/1 respectively.

Now from Eq. (6.3),the set of stabilizing controller for the system G(s)is,

...(6.3.1)

Similarly the set of stabilizing controller for the reduced model is,

...(6.3.2)

Given two plants *G(s)* and *R(s),* which are to be stabilized, then the parameterization of the two compensators which individually stabilize those plants can be equated to find the common stabilizing compensator.

If we equate Eq. (6.4) and Eq. (6.5) and solving for r1*,* we get,

...(6.3.3)

**Theorem 1 :**

If dmdp - nmdp + npdmk2 + npdm isHurwitz then there exits an r nth order compensator,

which stabilizes higher order system and itsreduced order model.

**Proof:**

From Eq. **(**6 .6)we have r1as**,**

Now for simplicity we take = 1 ε S. Then,

Finally we get r1as,

…(6.3.4)

Hence if, dmdp - nmdp + npdmk2 + npdm ,is Hurwitz then r1ε S*.* Sofrom Eq. (6.2) and Eq. (6.5), m thorder compensator,

…(6.3.5)

stabilizes both system and model.

**6.4 Inference**

* Ifwe observe Eq. (6.8) it was seen that, for designing a controller for higher order system,only reduced order model and checking Hurwitzness ofcertain polynomial is involved in computation. Hence complexity in obtaining controller for higher order system is greatly reduced and further we get a controller of lower order i.e. of m th order*.*
* The compensator in Eq. (6.8) can be parameterized by one parameter as follows: The range ofk2 for which the feedback system with model is stable can be obtained. Similarly the range of k2 for which the polynomial (dmdp - nmdp + npdmk2 + npdm ), is stable can be obtained. From this actual range of k2 can be obtained for which

stabilizes system and model. In this approach k1 is not required to be computed.

* In case ofdominant pole retention methods of,model order reduction we can factorize the denominator of the plant dp, as dp= dmd n-m, where dm,= denominator ofthe reduced order model which retains dominant poles ofhigher order system. and dn-m= the polynomial which represent remaining poles of the higher order system*.*

If we substitute dp, = dmdn-m in eq(6.7)we get r1 as

... (6.4.1)

Hence if dp-nmdn-m+npk2+np is Hurwitz then r1 ε S. Thus in case ofdominant pole retention method, we are able to design a controller by analyzing a lesser degree polynomial*.*

**6.5 Numerical example for controller design of higher order system via its reduced model obtained from Routh Pade approximation**

The higher order system is given by :

H(s) =

And the reduced model obtained from Routh Pade approximation is:

R(s)

= s3+7s2+24s+24 = s4+10s3+35s2+50s+24

=10s+12 = 15s2+21s+12

Now the range of k2 for which reduced model is stable can be obtained as**:**

1+ k2= 0

15= 0 …(6.5.1)

From Routh-Hurwitz criterion constraints for which reduce model(eq 6.5.1) is stable are:

(12+12) > 0

So,>-2.1,-1 …(6.5.2)

The polynomial **dp- nmdp+ np k2 + np = 0** becomes:

= 0

=> = 0 …(6.5.3)

Using Routh Hurwitz criterion for the polynomial,we get constraints on k2 for which polynomial (6.5.3) is stable as:

* 176+> 0
* > 0
* > 0
* )

> 0

)

)

(

2 > 0

* > 0

Solving above constraints we get :

> -11.7333 ,-4.8994, -1.83865,0.483755,-4.72382,-2.89811,-1 …(6.5.4)

The range of k2 satisfying all the constraints (in 6.5.2 and 6.5.4) : k2 > 0.483755

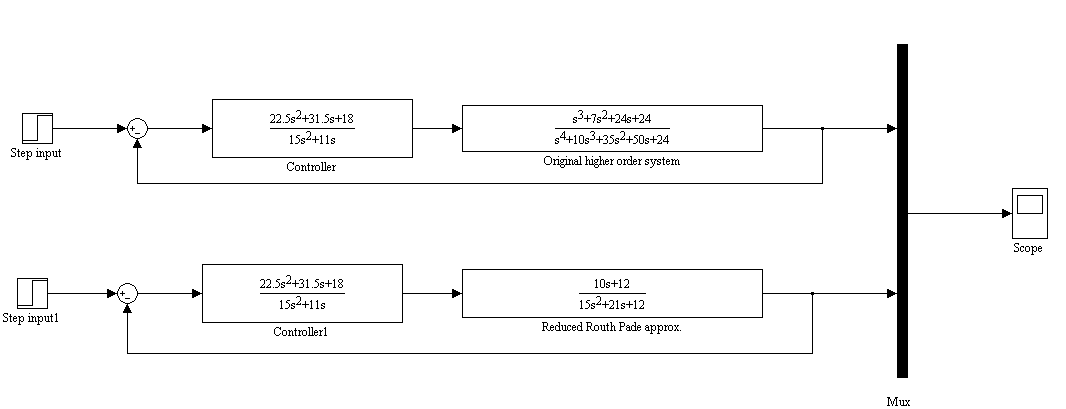
Hence a compensator becomes

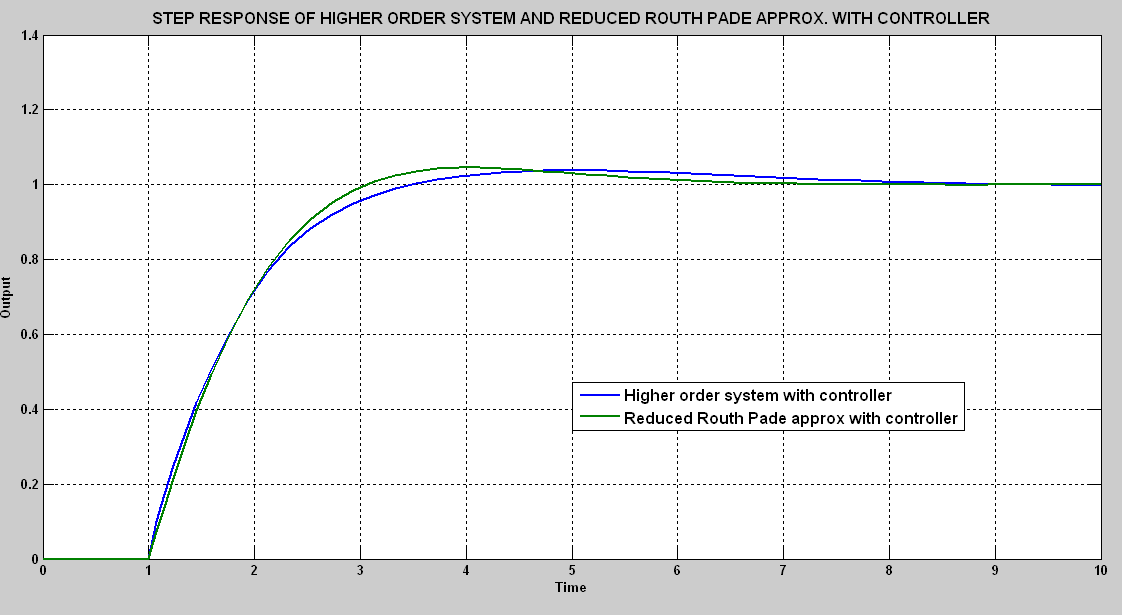
C(s) =

= where k >1.483755 …(6.5.6)

which is stabilizing controller for higher order system and its reduced order model by Routh Pade approximation.

**SIMULATION RESULT:**

****



**Fig6.5.1**

**6.6 Numerical example of controller design of higher order system via its reduced model obtained from Routh approximation**

The higher order system is given by:

H(s) =

And the reduced model obtained from Routh approximation is:

= s3+7s2+24s+24 = s4+10s3+35s2+50s+24

= 0.7947s+0.7947024 = s2 + 1.65563s+ 0.7947024

Now the range of k2 for which reduced model is stable can be obtained as**:**

1+ = 0

s2 + (1.65563+0.7947k2)s+ (0.7947024+ 0.7947024k2) = 0 … (6.6.1)

From Routh-Hurwitz criterion constraints for which reduce model(eq 6.6.1) is stable are:

1.65563+0.7947k2 > 0

0.7947024+ 0.7947024k2 > 0

So,k2 > -1,-2.083339 …(6.6.2)

The polynomial **dp- nmdp+ np k2 + np = 0** becomes:

(s2+1.65563s+0.7947024)(s4+10s3+35s2+50s+24)(0.7947s+0.7947024)(s4+10s3+35s2+50s+24)+( s3+7s2+24s+24)( s2 + 1.65563s+ 0.7947024) k2 + ( s3+7s2+24s+24 )( s2 + 1.65563s+ 0.7947024) = 0

=>

= 0 …(6.6.3)

Using Routh Hurwitz criterion,we get constraints on k2 for which polynomial (eq 6.6.3) is stable as:

* > 0
* > 0

> 0

* [{

+30369.25965

> 0

Thus,k2 >-11.86093,0.3433,0.80966,-1.82777,-11.86092,-1,1.018312,-1.412,-1.2259,-13.21075.

…(6.6.4)

The range of k2 satisfying all the constraints (6.6.2 and 6.6.4) is k2 > 1.018312

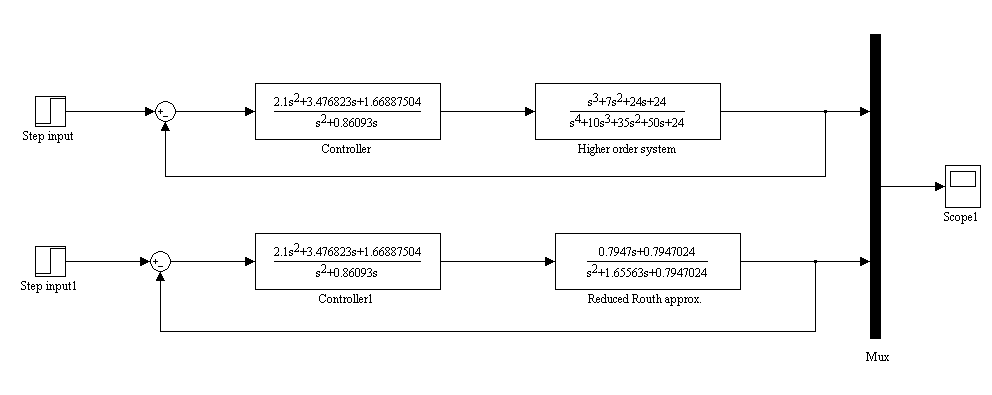
Hence a compensator becomes

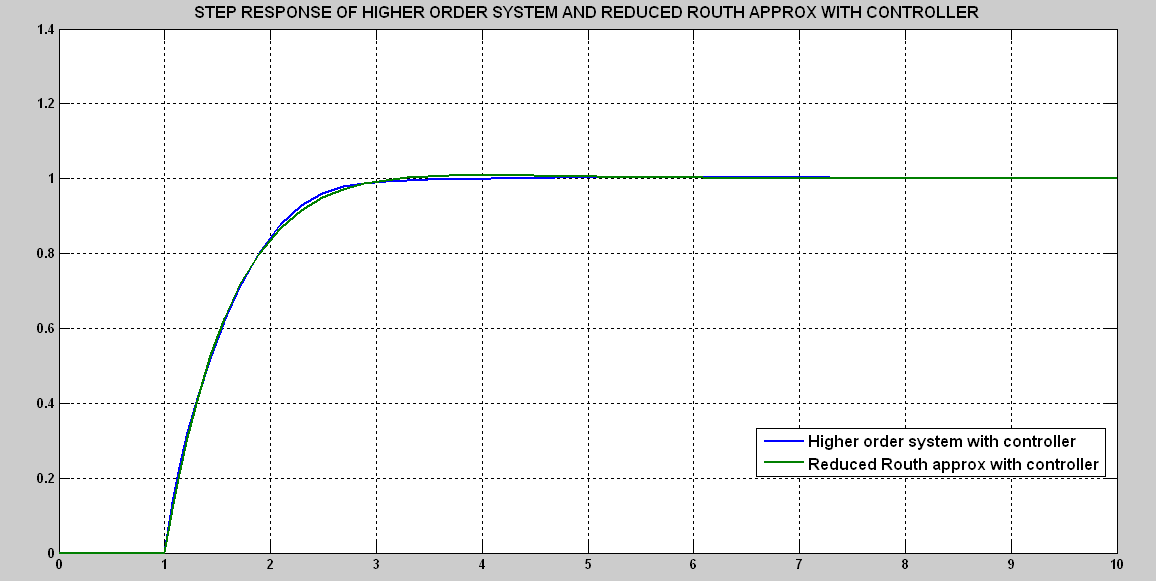
C(s) =

where k >2.018312

which is stabilizing controller for higher order system and its reduced order model by Routh approximation.

**SIMULATION RESULT:**





**Fig.6.6.1**

**6.7 Numerical example for controller design of higher order system via its reduced model obtained from Genetic algorithm based Optimization**

The higher order system is given by :

H(s) =

And the reduced model obtained from Genetic algorithm

R(s) =

= s3+7s2+24s+24 = s4+10s3+35s2+50s+24

=0.841s+0.277512 = 1s2+1.124s+0.279744

Now the range of k2 for which reduced model is stable can be obtained as**:**

1+ k2  = 0

= 0 …(6.7.1)

From Routh-Hurwitz criterion constraints for which reduce model (eq 6.7.1) is stable are:

() > 0

So,> -1.3365,-1.008 …(6.7.2)

The polynomial **dp- nmdp+ np k2 + np = 0** becomes:

-+ = 0

=>= 0 …(6.7.3)

Using Routh Hurwitz criterion for the polynomial(eq 6.7.3),we get constraints on k2 for which polynomial is stable as:

* > 0
* > 0
* > 0

( > 0

2 > 0

* > 0

Thus ,>

The range of satisfying all the constraint is.

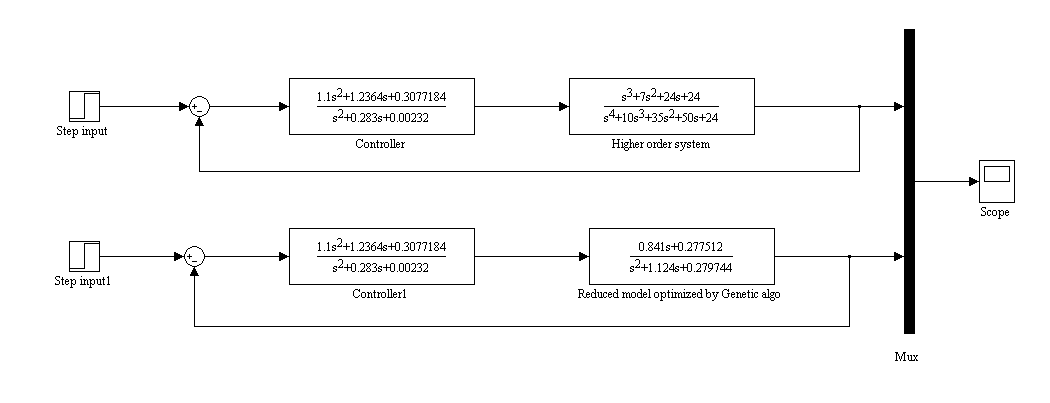
Hence a compensator becomes

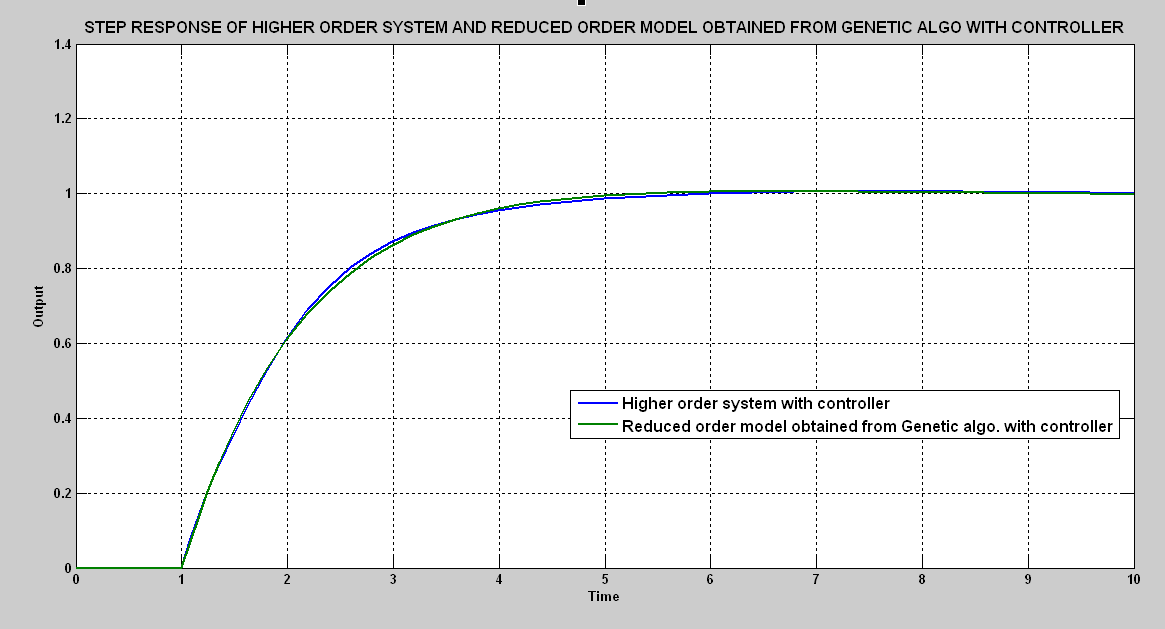
C(s) =

= where k >

which is stabilizing controller for higher order system and its reduced order model Genetic algorithm.

**SIMULATION RESULT:**

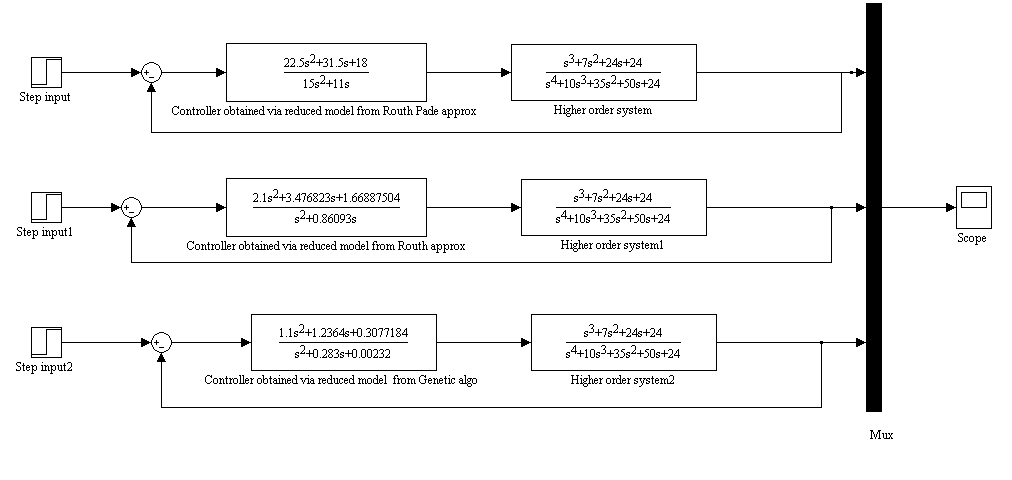
****

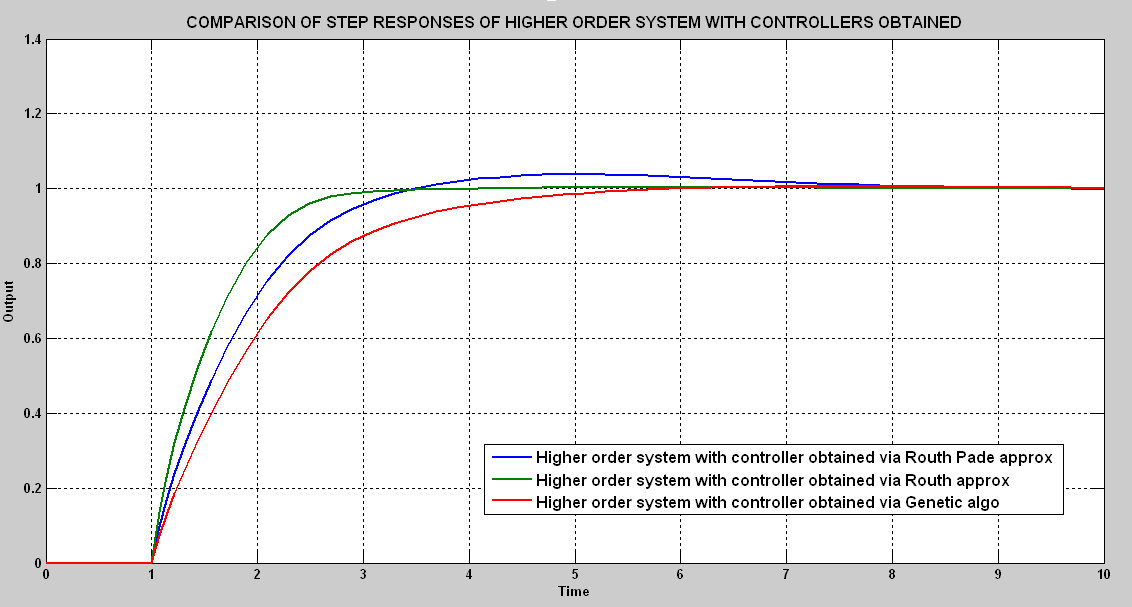


**Fig.5.7.1**

**5.8 CONCLUSION:**

Step responses of original higher order system with controllers designed via reduced models obtained from the two conventional methods-Routh Pade and Routh approximation and finally through Genetic algorithm is compared.





**Fig5.8.1**

**CHAPTER-7**

**CONCLUSION AND FUTURE SCOPE OF WORK**

Second order approximations of fourth order system via conventional methods(Routh Pade and Routh approximation) are compared with Genetic algorithm based optimization.

The controller for higher order system was designed via its reduced order approximations.

It was seen that Genetic algorithm based optimization gives a very good approximation of higher order system in second order transfer function. The controller designed via reduced model obtained through Genetic algorithm based optimization improves performance of the system.

As the procedure of optimization through Genetic algorithm is computer oriented a very high order practical system can be reduced by this algorithm which is comparatively a difficult process in Conventional reduction methods.

Genetic algorithms with adaptive parameters (adaptive genetic algorithms, AGAs) is significant and promising variant of genetic algorithms. The probabilities of crossover and mutation greatly determine the degree of solution accuracy and the convergence speed that genetic algorithms can obtain. Instead of using fixed values of probabilities of crossover and mutation, AGAs utilize the population information in each generation and adaptively adjust the probabilities of crossover and mutation in order to maintain the population diversity as well as to sustain the convergence capacity. In AGA (adaptive genetic algorithm) the adjustment of probabilities of crossover and mutation depends on the fitness values of the solutions. Hence, a more accurate solution can be obtained by the use adaptive genetic algorithms

**REFERENCES**

[1] Y.Shamash, “Model reduction using Routh stability criterion and Pade approximation

technique,” *Int J. Control* ,vol. 21 ,pp 475-484 ,1975.

[2] Y.Shamash, “Stable reduced order models using Pade type approximants,” *IEEE transaction*

*on automatic control*, vol.19,pp 615-619,October ,1974.

[3] C.B Vishwkarma and R.Prasad, “System reduction using pole clustering and Pade

approximation,”*National system conference,*December ,2008.

[4] Kaplan B.Z , “Stable reduced order Pade approximants using stability equation method,”

*Electronics letters* ,vol. 16,no. 9,April, 1980.

[5] K.J Khatwani,R.K.Tiwari and J.S Bajwa, “On Chuang’s continued Fraction expansion

method of model reduction,” *IEEE transaction on automatic control*,vol.25,pp822-824,

August,1980.

[6] Jih-Sheng Lai, “Practical model order reduction,”*IEEE transactions on Industrial electronics*

*,*vol IE-34 no.1*,*February,1987*.*

[7] W.Krajewski,A.Lepschy and U.Viaro, “Model reduction by matching Markov parameters,

Time moments and impulse response energies,” *IEEE transactions on Automatic control,*

vol.40,pp 950-953,May,1995.

[8] Edmund Jonckheere and M.A Chingwo, “Combined sequence of Markov parameters and

moments in linear systems,” *IEEE transaction on automatic control*, vol.34 ,pp379-382,

382,1989.

[9] J.Pal, “System reduction by Cauer continued fraction expansion,”*Electronic letters*,vol.23,

1982.

[10] T.N.Lucas, “Efficient algorithm of reduction by continued fraction and expansion,”

*Electronic letters* vol. 19,no. 23 ,1983.

[11] Pal J, “System reduction by mixed method”,*IEEE proceedings on Automatic control*,vol.

AC-25,pp973-975,1980.

[12] T.N Lucas, “Factor division –A useful algorithm in model reduction,”*IEE proceedings*, vol.

130,pp362-364,November 1983.

[13] C.B Vishwkarma and R.Prasad, “Order reduction using the advantages of differentiation

method and factor division algorithm,”*Indian Journal of Engineering & Material sciences*,

vol. 15,pp 447-451,December 2008.

[14] S.K Mittal,D.Chandra,B.Dwivedi, “A computer aided approach for Routh Pade

Approximation of SISO systems based on multi objective optimization,” *International*

*journal of* *engineering and technology*,vol.2,no.2,April2010.

[15] T.N Lucas, “Linear system reduction by modified factor division method”, *IEE proceedings*

vol. 133,pp293-295,November 1986.

[16]Alessandro Astolfi, “Model reduction by moment matching for linear and Nonlinear

systems,” *IEEE transaction on automatic control*, vol.5 ,pp2332-2336,October ,2010.

[17] Alessandro Astolfi, “Model reduction by moment matching steady state response and

projections ,” *49th IEEE conference* *on decision and control*, pp5344-5349,December,

2010.

[18] H.Xiheng, “FF Pade method of model reduction in frequency domain,” *IEEE transaction on* [19] Khatwani J.K, “On Routh Hurwitz criterion,” *IEEE proceedings*,1981.

[20] Vimal Singh, “Remarks on system simplification using the Routh stability array,”

*Proceedings of the IEEE*,vol 69,pp 662,May 1981.

[21] Krishnmurthy V and Sheshadri V, “Model reduction using Routh stability criterion,”*IEEE*

*proceedings,*1978.

[22] M.F Hutton and Bernard Freidland, “Routh approximation for Reducing order of LTI

systems,”*IEEE transaction on automatic control*, vol.AC-20,pp 329-337,June 1975.

[23] Y.Choo and D Kim , “Routh approximants of arbitrary order,”*Institute of Control*

*Automation and Systems engineering,*Korea,vol.1,No1,June,1999*.*

[24] C.W Merril, “Applied Routh approximation,”*NASA technical paper*,April,1978.

[25] Hseih S and Hwang C, “Model reduction of continous time system using a modified Routh

approximation method,” *IEE proceedings*,vol 136,July,1989.

[26] D. Kranti Kumar ,S.K Sagar,J.P Tiwari, “Model order reduction of Linear interval system

using Mihailov criterion and Routh approximation,”*International Journal of engineering*

*science* *and technology* ,vol.3,pp5593-5598,July 2011.

[27] V.Sreeram,P. Agathokils, “Model reduction of linear continuous systems using weighted

impulse response Grammians,”*IEE proceedings* ,vol.140,pp345-351,September,1993.

[28] T.N Lucas, “Scaled impulse energy approximation for model reduction,” *IEEE transaction*

*on automatic control*, vol.33 ,pp 791-793,1988.

[29] M.Medenica and V.Debeljovick , “Special criteria to the stability criterion to linear systems,

systems,”*Proceedings of the IEEE*,pp2980-2984,October2006.

[30] Panda,S.,Tomar,SK,Prasad “Model reductions of systems by conventional and evolutionary

technique,”*International Journal of computation and Mathematical sciences*,2009.

[31] William.H.Hsu, “Genetic algorithms”,Kansas state university ,USA.

[32] J.Kennedy and R.C Ebehart, “Particle swarm optimization,”*IEEE, Int. conference on Neural*

*networks*,1995.

[33] Z.W Wang and X.Q Liu, “Model order reduction based on particle swarm optimization,”

*IEEE proceedings*,2006 .

[34] Reid.S.Maust and Ali Feliachi, “Reduced order modeling using genetic algorithm,” *IEEE*

*Proceedings ,*pp66-71,1998.

[35] V.S Chellaboina and M.K Ranga, “Reduced order optimal control using genetic algorithm,”

*American control conference*,pp1407-1411,June 2005.

[36] Philip.D.Oliver, “A comparison of reduced order model techniques”,*41st Southeastern*

*symposium on system theory,*March,2009*.*

[37] J.Yanh,T.Haschino,T.Tsuji, “Model reduction with time delay with least squares method wit

with the genetic algorithm,”*IEE proceedings on Control theory Application*,vol.143,pp247-

254,May,1996.

[38] D.K Saini and R Prasad, “Order reduction of linear interval systems using genetic

Algorithm,”*International journal of engineering and Technology*,vol.2,pp316-319,2010.

[39]Zaer .S. Abo-Hammour, Othman M.K Alsmadi, and Adnan M. Al-Smadi , “Frequency

based model order reduction via genetic algorithm approach,”*7th International Workshop* *on*

*System,signal processing and their application,*pp91-94,2011*.*

[40] Moin L,Ali S Ali, “System Response Analysis and Model Order Reduction, “Using.

Conventional Method, Bond Graph Technique and Genetic Programming”,National

University of Sciences and Technology,Karachi,2008.

[41] G. Parmar, R. Prasad, and S. Mukherjee, *“*Order Reduction of Linear Dynamic Systems

using Stability Equation Method and GA,” *International Journal of computer ,information*

*and* *systems science and engineering*, pp 26-32,2007.

[42]C.B Vishwkarma and R.Prasad , “Reduced order modeling using Mihailov criterion and

genetic algorithm,” *National systems conference*,December,2008.

[43] B. Bandyopadhyay, H. Unbehauenu and B. M. Patre, “Control of higer order system via its

reduced model,”*IEEE Trans. Autom. Controls,*vol .9 pp226-229 ,1998.

[44] S. K. Agrawal, Dinesh Chandra & Irfan Ali Khan, “Controller Design Based on ISE

Minimization and Dominant Pole Retention Method,” *International Journal on Computer*

*Science* *and Engineering,* vol. 02, no.7, pp 2242-2245, 2010.

[45] Brian D.O Anderson and Yi Liu, “Controller reduction concepts and approaches,”*IEEE*

*Trans. Automatic Control,*vol. 34.no.8 pp802-812 ,August,1989.