ABSTRACT

The complications of the systems make their analysis a rather difficulty and possibly a non desirable task, mainly due to difficult economic and computational considerations involved. This makes the need for using model, which constitutes a good approximation of the original system.

The purpose of this dissertation is to develop methods, which preserve time-domain as well as frequency-domain characteristics of original higher-order discrete-time systems and their application for the control of discrete-time systems.

The Routh stability method is reported to be simple and powerful one, which carries significance like reduced-order models are stable, provided original system is stable. The overall time and frequency-domain characteristics are retained and it offers computational simplicity.

The Routh approximation method on the other hand, posses not only the stability preservation characteristics of the system, but also many desirable features such as model stability, minimum computation and recursive nature of calculation for reduced model of various orders. However, it does not provide good transient response for purpose. A Modified Routh stability method has been proposed.

Besides these reduced-order modeling techniques, Modified Routh stability method using p-domain transformation has been reported which patches up shortcomings of bilinear transformation and yields stable reduced-order models provided original system is stable.

CHAPTER 1

INTRODUCTION

1.1 GENERAL DESCRIPTION:

The analysis of all physical systems starts by building up of a model. A physical phenomenon has to be adequately modelled so as to be a faithful representation of a reality; all further analysis can be done on the model so that experimentation on the process is no longer required. The advent of the digital computer has meant that relatively complex models can be analyzed.

The mathematical procedure of the system modeling often leads to comprehensive description of a process in the form of high-order differential equations which are difficult to use either for analysis or controller synthesis. Thus, it motivates to find the possibility of some equations of the same characteristics but of lower-order that adequately reflects the dominant characteristics of the system under considerations.

The reasons which prompt to have reduced-order models of higher-order linear system could be:

(a) To have better understanding of the 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.

(b) To reduce 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.

(c) To reduce hardware complexity:

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.

1.2 APPLICATIONS 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.

1.3 CLASSIFICATIONS OF REDUCED ORDER MODELLING:

The order reduction techniques can be broadly classified as:

(i) Time Domain Simplification Techniques:

In time domain reduction techniques, the original and reduced system are expressed in state space from where the order of matrices A_r, B_r, C_r are less than A, B, C and the output Y_r will be a close approximation to $Y_{orginal}$ for specified inputs. The time-domain techniques belong to either of categories:

(a) Perturbation Method: It is based on the approximation of system's structure through neglecting certain interactions within the model which leads to lower-order. The basic benefits from this model are computational and structure realization. However these benefits cannot be at the expense of key system's properties such as stability.

(b) Aggregation: The intuitive notion behind an aggregative model is to combine certain system variable which in effect, involves weighted averaging of the state vector to find an approximate model for a large scale system. It has been applied to both time and frequency domain. It preserves the stability of the systems.

(c) Gramian Technique: It is a balanced realization which is based on parameter matching. The ROM (Reduced Order Model) matches various combination of four types of the system invariant parameters of the full-order system associated with low frequency response, high frequency response, low frequency power spectral density and high frequency power spectral density.

(ii) Frequency Domain Model Reduction:

Frequency domain model reduction may be divided into three groups.

(a) The first group is classical reduction method (CRM) which is based on classical theories of mathematical approximations or mathematical concepts such as continued-fraction expansion and truncation, pade approximations and the time-moments matching these approximations. These methods are algebraic in nature. The problems such as instability, non-minimum phase behavior and low accuracy in the mid and high frequency range of reduced-order model restrain the application of CRM.

(b) The second group is a development of CRM and includes the stability preservation method (SPM) such as Routh approximation. Hurwitz polynomial approximation. Dominant pole retention and stability-equation method. The SPM poses serious drawback of lack of flexibility which, the reduced model does not produce good enough approximation.

(c) The third group includes the mixed methods and known as stability criterion method (SCM) where the denominator of reduced model is derived by one of SPM and numerator parameters are evaluated by a described CRM. The SCM incorporates interest by a described method. This improves the degree of accuracy in the low frequency range.

(1.4) LITRETURE SURVEY:

These dissertations work in based on reduced-order modeling for the analysis of discrete time systems in this regard as in this manner.

AI-Saggaf, U. M. and Franklin, G.F, [1], have studied the robust discrete control system design techniques. A new linear quadratic gaussian /loop transfer recovery procedure for discrete time systems in presented. In this technique, a full-state feedback or an output injection feedback is designed which has the designed loop shape, and than recovered by a realizable linear quadratic gaussian controller. The complexity of the resulting controller is than reduced without causing closed-loop instability.

Badreddin, E. and Mansour. M, [2], have studied the model reduction of discrete-time systems using Schwarz canonical form. It is employed to have stable reduced-order models which are stable if original system is stable. Further, the steady part of system responses of the model to a step input is equal to that of the system.

Chen T-C, and chang, C.Y, [3], have described a method of model reduction of reducing a high-order transfer function to its low-order models based upon the stability-equation method. The transformations of reduced order are obtained directly from the pole-zero patterns of the stability equation of the original transfer function.

Farsi, M., Warwick, K. and Guilandoust K, [4], have suggested a technique which provides stable reduced order models for discrete time systems. In this method Routh stability approach in employed to reduce the order of discrete time systems. Transfer function which employs a new transformation approach i.e. p-domain gives a stable reduced order model if the original system is stable.

Hwang, C. and Hsieh C-S, [5], have described a method of combining the Routh approximation method with the bilinear transformation for deriving stable reduced-order models of a strictly proper Z-transfer function. It is based on applying the bilinear transformation to the (Z+1) $G\eta(Z)$, and then deriving a new bilinear Routh γ–δ canonical expansion for $G\eta(Z)$

Hwang, C., Hwang, J.H., and Guo, T-Y, [6], have described a multipoint Routh γ–δ canonical continued fraction expansion for the transfer function of linear systems. Based on this general form, a multifrequency Routh approximant to the system is derived by selecting the expansion point on the imaginary axis and truncating the resulting continued fraction expansion.

Warwick, K, [7], have suggested a method in which an error polynomial is defined, the coefficients of which indicate the difference at any instant between a system and a model of lower order approximating the system. Also discussed the way in which the error between system and model can considered as being a filtered form of an error input function specified by means of model parameters selection.

Zhang, W.D., Sun, Y.X. and Xu, X.M, [8], proposed the Dahlin controller in the complex – frequency domain in terms of performance and robust stability. The possibility of extending the Dahlin controller to the control of plants is discussed. A new procedure in developed for digital controller.

T.Kangsanant, [9], presented a new algorithm for model reduction of z- transfer function. The technique is based on moment matching. The simple algorithm is obtained with the use of a power series transformation. Reduced order models are obtained by matching the coefficients of the power series in the transformed domain, which is equivalent to matching of time moments of impulse responses. Though this technique produces good results but it does not guarantee stability of reduced transfer function.

Shanti Mishra, Jayanta Pal, [10], have described a mixed method for the reduction of high order discrete time systems. The reduced model denominator is formed by using the stability-equation method and reciprocal transformation in w-domain. The numerator dynamics in w-domain are chosen to fit a number of initial time-moments and Markov parameters of the original system. The reduced model in w-domain is then transformed back to the z domain. The method approximates both low and high magnitude poles to give better matching in both the transient and steady state regions of the step response.

R. Unnikrishnan and A. Gupta, [11], presented a modified approach for reducing the order of discrete time systems. The objective of reduction process is to preserve specific design parameters of the transfer function such as phase margin, gain margin and bandwidth. In this technique the transfer function is first expanded about zero and then about infinity so that steady state response and transient response for a step input are both carefully approximated. The remaining function is then approximated by a lower order transfer function to match the frequency response at specific frequencies.

B.Clapperton, F. Crusca and M. Aldeen, [12], have described a new general bilinear relationship between continuous and discrete Generalised Singular Perturbation (GSP) reduced order models. This result is applied to the problem of deriving discrete analogues of continuous singular perturbation and direct truncation model reduction, and leads to a new definition of discrete “Nyquist” model reduction. Also ’unit circle’ bilinear transformations are used to relate several known facts about continuous and discrete balanced model reduction and incorporate them into a symmetrical, unified framework.

Vivek kumar sehgal, [13], suggested a method which preserves time domain and frequency domain specifications of original discrete time systems with higher order controller and their application for the control of discrete time systems. A new mixed method, improved routh stability method, p- domain transformation have been proposed which patches up the shortcomings of bilinear transformation and yields stable system with reduced order controller.

P. Brehonnet, A.Derrien, P. Vilbe and L.C. Calvez, [14], proposed a novel non iterative method for deriving reduced order models of linear discrete time systems without solving any non linear equation. The resulting error energy is usually relatively close to optimal.

M. Diab, W. Q. Liu, and V. Sreeram, [15], suggested an double–sided weights. A number of properties of the gradient flows associated with the objective function are obtained.optimization technique for model reduction. The objective function being minimized is the impulse energy of the overall system with unity.

Chen-Chien Hsu, Kai-Ming Tse, and Wei-Yen Wang, [16], suggested a framework to automatically generate a reduced-order discrete-time model for the sampled system of a continuous plant preceded by a zero-order hold using an enhanced multiresolutional dynamic genetic algorithm (EMDGA).Chromosomes consisting of the denominator and the numerator parameters of the reduced-order model are coded as a vector with floating point type components and searched by the genetic algorithm. Therefore, a stable optimal reduced-order model satisfying the error range specified can be evolutionarily obtained. Because of the use of the multiresolutional dynamic adaptation algorithm and genetic operators, the convergence rate of the evolution process to search for an optimal reduced order model can be expedited. Another advantage of this approach is that the reduced discrete-time model evolves based **on** samples directly taken from the continuous plant, instead of the exact discrete-time model, so that computation time is saved.

Shih-Lian Cheng and Chyi Hwang,[17], described a differential evolution algorithm (DEA) incorporating a search-space expansion scheme for solving the problem of optimally approximating linear systems The optimal approximate rational model with/without a time delay for a system described by its rational or irrational transfer function is sought such that a frequency-domain L^2-error criterion is minimized. The distinct feature of the proposed model approximation approach is that the search-space expansion scheme can enhance the possibility of converging to a global optimum in the DE search. This feature and the chosen frequency-domain error criterion make the proposed approach quite efficacious for optimally approximating unstable and/or non minimum-phase linear systems.

G. Kotsalis, A.Megretski, M. A. Dahleh, [18], proposed a model reduction algorithm for discrete-time, markov jump linear systems. The main point of the reduction method is the formulation of two generalized dissipation inequalities that in conjunction with a suitably defined storage function enable the derivation of reduced order models that come with a provable a priori upper bound on the stochastic L2 gain of the approximation error.

J. S. Yadav, N. P. Patidar, J. Singhai and S. Panda, [19], described reduction of SISO discrete systems, using a conventional and a bio-inspired evolutionary technique. In this method, the original discrete system is first converted into equivalent continuous system by applying bilinear transformation. The denominator of the equivalent continuous system and its reciprocal are differentiated successively and the reduced denominator of the desired order is obtained by combining the differentiated polynomials. The numerator is obtained by matching the quotients of MCF. Finally, the reduced continuous system is converted back into discrete system using inverse bilinear transformation. In the evolutionary technique method, Differential Evolution (DE) optimization technique is employed to reduce the higher order model. DE method is based on the minimization of the Integral Squared Error (ISE) between the transient responses of original higher order model and the reduced order model pertaining to a unit step input.

(1.5) ORGANISATION DISSERTATION:

This dissertation comprises following significant parts to which follows in this manner.

Chapter 1 deals with general introduction of reduced order modeling and its implication for systems analysis.

Chapter 2 describes reduce order modelling techniques extended by K.Warwick [7] Hwang, C; and Hseih, C.S. [5], Hwang, C; Hwang, J.H; and Guo, T.Y. [6], Zhang, W.D., Sun, Y.X. and Xu, X.M. [8], Shanti Mishra, Jayanta Pal[10], well as J. S. Yadav, N. P.

Patidar, J. Singhai and S. Panda[19] which impart systems if original system is stable.

Chapter 3 elaborates reduced order modelling by modified Routh stability method using P-domain transformation stable state system. This technique incorporates P-domain transformation which resolves the lacunes of bilinear transformation.

Chapter 4 elaborates modified Routh stability for unstable systems.

Chapter 5 Includes summaries of the major parts of the Dissertation, discussion about proposed methods (A new modified Routh Stability method) and constraints imposed on controller design for discrete time systems have been described. Some, observations about the scope of further work in this area have been included.

CHAPTER 2

AN OVERVIEW OF MODEL REDUCTION TECHNIQUES FOR DISCRETE-TIME SYSTEMS

2.1 INTRODUCTION

Reduced order modelling for the analysis of continuous time systems have been attempted by various researchers. Latter it become well known that most of reduced order modelling techniques can be extended for discrete time systems equally by introducing bilinear transformation, other transformations like z = p+1 (6) and z = p/(A+BP) where A and B are continuous and P is a new variable.

Significant reduced order modelling techniques and its application for the analysis of discrete time systems proposed by earlier workers are:

(i) A new approach to reduced order modelling, [7].

(ii) Order reduction of discrete time systems via Bilinear Routh Approximations, [5].

(iii) Multifrequency Routh approximants for linear systems, [6].

(iv) Robust digital control design for process as with dead time; new results, [8].

(v) A mixed method for the reduction of discrete time systems,[10]

(vi) Differential evolution algorithm for model reduction of SISO discrete time systems,[19]

2.2 A NEW APPROACH TO REDUCED ORDER MODELLING:

Warwick, K. [7], proposed a new approach to reduced order modelling by an error polynomial is defined, the coefficients of which indicate the difference of any instant between a system and a model of lower order approximating the system.

Many control technique are relatively simple to implement on a low-order system containing few parameters. High order systems, however generally lead to a much greater amount of effort for their analysis, especially in terms of necessary computation. The methods involve detailed ladder networks which are computationally time consuming and prone to rounding errors. In this paper a new method for obtaining reduce-order models is suggested. It is believed that the approach is computationally efficient and simple to comprehend.

2.2.1 Reduction of discrete-time systems:

It is assumed that a high-order discrete-time transfer function, relating plant input to output, is available, and can be described by the expression

$$G(z) = \frac{d_0 + d_1 z + \dots\dots\dots + d_{n-1} - Z^{n-1}}{e_0 + e_1 z + \dots\dots\dots\dots e_{n-1} z^{n-1} + e_n z^n} \tag{2.1}$$

where the numerator order is given as one less than the denominator. This limitation is merely for explanation purposes; the method of reduction to be described later imposes no such restrictions itself.

The transfer function G (z) can also be written with regard to its power series expansion about $Z = \infty$, i.e.

$$G(z) = \sum_{i=1}^{\infty} g_{-i} z^{-i} \qquad (2.2)$$

the parameters {g_{-i}: i = 1, ……., }being called the Markov parameters.

Similarly, G (z) can be written in terms of its power series expansion about z =1

$$P = z\text{-}1 \qquad (2.3)$$

such that

$$G(p) = \frac{d'_0 + d'_1 p + \text{.........} + d'_{n-1} - p^{n-1}}{e'_0 + e'_1 p + \text{...........} e'_{n-1} p^{n-1} + e'_n z^n} \qquad (2.4)$$

The expansion of G (p) about p = 0 is equivalent to the expansion of G (z) about z =1; hence

$$G(p) = \sum_{i=1}^{\infty} h_i p^i$$

The parameters {h_i: I = 1, ..,} are proportional to the system time moments.

The desired reduced-order model can be described in a similar fashion to G(z). Let this model be

$$R(z) = \frac{a_0 + a_1 z + \text{.........} + a_{k-1} - Z^{k-1}}{b_0 + b_1 z + \text{...........} b_{k-1} z^{k-1} + z^k} \qquad (2.6)$$

in which the index k n {When $k \leq n$, R(z) G(z)}. The two transfer function may now be written as

$$\text{(i)} \qquad G(z) = \frac{D(z)}{E(z)}$$

$$\text{(ii)} \qquad R(z) = \frac{A(z)}{B(z)} \qquad (2.7)$$

This overall problem can now be described as one in which the parameters {a_i : I = 1, .., k-1} and {bi: i=1, …., k} are chosen such that the model transfer function, R(z), is in some these, a good approximation to the system transfer function G(z).

2.2.2 Error polynomial and its implications:

The resultant error between system and model can be described by the equation

$$G\ (z) = R\ (z) + \lambda\ (z) \qquad (2.8)$$

Where λ (z) is a rational transfer function denoting the undesired error

Eqn. 2.8 can, however, also be written, from eqn. 2.7, in the form

$$D\ (z)\ B\ (z) = A\ (z) E\ (z) + \lambda\ (z)\ B\ (z)\ E\ (z)$$

Such that the error polynomial may now be defined as

$$W\ (z) = \lambda\ (z)\ B\ (z)\ E\ (z) \qquad (2.9)$$

Where $W\ (z) = w_0 + w_1 z + \ldots\ldots\ldots.. w_m\ z^m$ (2.10)

in which m = n + K-1.

The usefulness of the {wi: i = 0, 1... m} parameters and their meaning in relations to the error between model and system at any time instant will now be discussed.

If an identical input is provided to both system and model, an error will be apparent between the respective outputs when the transfer functions are not identical. Let this input be u (t) at time t {t = 0, ±1, ±2, ….}, and let the outputs, at time t, be y_s (t) and y_m (t) for system and model, respectively.

Then, the error at time t is defined as:

$$v(t) = y_s(t) = y_m(t) \tag{2.11}$$

This also may be written as

$$v(t) = [G(z) - R(z)]u(t) \tag{2.12}$$

or, by inclusion of eqn. 2.8,

$$v(t) = \lambda(z)\, u(t)$$

Hence using 2.9

$$E(z)\, B(z)\, v(t) = W(z)\, u(t) \tag{2.13}$$

The m roots of the W (z) polynomial are therefore also the zeros of the transfer function relating input to error. It follows that if the system denominator polynomial, E(z), is stable, it is a requirement that the model denominator, B(z), is also stable, to enable the error to tends to zero under steady state conditions, Model stability is not generally achieved with all reduction methods and is shown here to be a limiting factor for a particular model choice.

2.2.3 Error polynomial with a unit step input:

Applying a unit step input to both model and system, {u (t) = 1: t ≥0} and {u(t) = 1 : t ≥0}. Then, if a subsidiary error signal is defined as

$$\bar{v} = \mathrm{W(Z)u(t)} \tag{2.14}$$

this signal is filtered by the polynomial E(z)B(z) to become the error V(t). At time instant t =1, therefore, v(1) = $\mathrm{W_m}$, and its follows that v(2) = $\mathrm{W_m}$,+$\mathrm{W_{m-1}}$, etc. This addition of errors at each time instant can be summarized by

$$\bar{v} = \{\sum_{\mathrm{i=j}}^{\mathrm{m}} \mathrm{w_i : j = m+1\text{-}t, 0 < t \le m+1 : j = 0, t > m+1}\}$$

Under steady-state conditions the error $\bar{v}(t)$ fed through to the E (z)B(Z) filter is thus $\sum_{i=j}^{m} w_i$, i.e. the summation of all the W(z) coefficients. There are, therefore, m+1 = n+k error terms $\{\bar{v}(t) : \mathrm{t = 1, .., m+1}\}$, but only 2k, the number of model parameters to be chosen, degrees of freedom. Hence, while k<n, at least one of the $\bar{v}(t)$ values will be nonzero, i.e. the model response cannot be made to fit exactly that of the system.

2.2.3.1 Illustrative example:

Consider a system of order 3, n =3, and a model of order 2, k =2. The error are built up as shown if Table 2.1.

Table2.1: Example 2.2.3.1

Time instant t	Error signal $\bar{v}(t)$	
0	0	
1	$\mathrm{W_4}$	$\overline{w}_4$

2	W_4+W_3	$\overline{w}_3$
3	$W_4+W_3+W_2$	$\overline{w}_2$
4	$W_4+W_3+W_2+W_1$	$\overline{w}_1$
5	$W_4+W_3+W_2+W_1+W_0$	$\overline{w}_0$
6	$W_4+W_3+W_2+W_1+W_0$	$\overline{w}_0$

Because K = 2, there are 2K = 4 model parameters to be chosen. Thus, there are 4 degrees of freedom with respect to the 5 error coefficients $\{\overline{w} : \mathrm{i} = 0, .. 4\}$.

One further polynomial must be introduced, this being W (p) obtained from W(z) by means of the substitution given in eqn. (2.3). Hence,

$$\mathrm{W(P)} = \hat{\mathrm{W}}_{\mathrm{m}}\mathrm{p}^{\mathrm{m}} + \hat{\mathrm{W}}_{\mathrm{m-1}}\mathrm{P}^{\mathrm{m-1}} + ... + \hat{\mathrm{W}}_1\mathrm{p} + \hat{\mathrm{W}}0$$

Such that $\hat{W}_i$ coefficients can be obtained simply form Pascal's triangle. Referring back to the illustrative example 2.3.1.1:

$$\hat{\mathrm{w}}_0 = \mathrm{w}_0 + \mathrm{w}_1 + \mathrm{w}_2 + \mathrm{w}_3 + \mathrm{w}_4 \qquad = \overline{\mathrm{w}}_0$$

$$\hat{w}_1 = \mathrm{w}_1 + 2\mathrm{w}_2 + 3\mathrm{w}_3 + 4\mathrm{w}_4$$

$$\hat{w}_2 = \mathrm{w}_2 + 3\mathrm{w}_3 + 6\mathrm{w}_4$$

$$\hat{w}_3 = \mathrm{w}_3 + 4\mathrm{w}_4$$

$\hat{w}_4 = \mathrm{w}_4 \qquad\qquad = \overline{\mathrm{w}}_4$

It must be denoted that

(a) $\hat{w}_0 = \overline{w}_0$

(b) $\hat{w}_0 = \overline{w}_m$

2.3 ORDER REDUCTION OF DISCRETE TIME SYSTEMS VIA BILINEAR ROUTH APPROXIMATION:

Hwang, C; and Hseih C.S. [5], proposed a method of combining the Routh approximation method with bilinear transformation for deriving stable reduced order models of a strictly proper Z-transfer function $G_\eta(Z)$. It is based on applying the bilinear transformation of the (z+1) $G_\eta(Z)$ and then deriving a new bilinear Routh $\gamma-\delta$ canonical expansion for $G_\eta(Z)$. The proposed bilinear Routh approximation method has all the advantages of Routh approximation [13], without having initial-value problem posed by the bilinear transformation.

Bilinear Routh approximation does not require the bilinear transformation explicitly, which involves directly in z-plane via single set of computation.

2.3.1 Bilinear Routh $\gamma-\delta$ Expansion:

Consider an asymptotically stable nth-order discrete-time described by the strictly proper z-transfer function:

$$\mathrm{G_n(z)} = \frac{\mathrm{b_1 + b_2 z + ... + b_n Z^{n-1}}}{\mathrm{a_1 + a_2 z + ... + a_{n+1} z^n}} \Delta \frac{\mathrm{N(z)}}{\mathrm{D(z)}}, \mathrm{a_{n+1}} = 1 \tag{2.15}$$

Or

$$G_n(z) = \frac{1}{z+1}\frac{(z+1)N(z)}{D(z)} = \frac{1}{(z+1)}\overline{G}n(z) \qquad (2.16)$$

Then it can be shown that the bilinear-transformed function

$$\overline{G}_n(S) = \overline{G}_n(z)\Big|_z = \left(\frac{1+s}{1-s}\right) \text{ is strictly proper} \qquad (2.17)$$

Hence $\overline{G}_n(s)$ can be expanded into the following Routh $\gamma - \delta$ canonical from [13]:

$$\overline{G}_n(s) = \frac{1}{s}\sum_{i=1}^{n}\delta_i \prod_{j=1}^{i}\hat{w}_n(s)$$

Where $\hat{w}_j(s)$, for j = 2, 3, …………….., n are defined by

$$\hat{w}_j(s) = \cfrac{1}{\gamma\, j/s + \cfrac{1}{\cfrac{\gamma\, j+1}{s} + \cfrac{1}{\ddots\cfrac{\gamma_{n-1}}{s} + \cfrac{1}{\cfrac{\gamma_n}{s}}}}} \qquad (2.19)$$

Application of inverse bilinear transformation [s = (z-1)/(z+1)] to $\overline{G}_n(s)$ results $\overline{G}_n(s)$.

$$\overline{G}_n(z) = \frac{1}{z+1}\overline{G}_n(s) = \frac{1}{z+1}\overline{G}_n(s)\bigg|_{s=\left(\frac{z-1}{z+1}\right)}$$

$$= \frac{1}{z-1}\sum_{i=1}^{n}\delta_i \prod_{j=1}^{i} w_i(z) \qquad (2.20)$$

Where

$$\overline{G}_n(z) = \frac{1}{z+1}\overline{G}_n(s) = \frac{1}{z+1}\overline{G}_n(s)\Big|_{s=\left(\frac{z-1}{z+1}\right)}$$

$$= \cfrac{1}{\gamma\ j\left(\frac{z+1}{z\text{-}1}\right) + \cfrac{1}{\gamma_{\ j+1}\left(\frac{z+1}{z\text{-}1}\right) + \cfrac{1}{\ddots \lambda_{n+1}\left(\frac{z+1}{z-1}\right) + \cfrac{1}{\gamma_n\left(\frac{z+1}{z-1}\right)}}}} \qquad (2.21)$$

γ_i and δ_i coefficients are obtained from the following recursive polynomials U_1(z)

And V_1 (z) as follows:

$$U_1(z) = \lfloor D(z) + z^n D(z^{-1}) \rfloor / 2 \qquad (2.22)$$

$$V_1(z) = \lfloor D(z) + z^n D(z^{-1}) \rfloor / 2 \qquad (2.23)$$

γ_i coefficients are obtained from following recursive algorithm:

Ui+1 = V_j(z)/(z-1) (2.24)

$$\gamma_i = U_i(1)/2[U_{i+1}(1)] \qquad (2.25)$$

V_{i+1}(z) = [U_i(z)$_{-i}$(z+1)U_{i+1}(z)]/(z-1) (2.26)

Once γ_i is calculated, δ_i coefficients in the bilinear Routh expansion can be computed by the recursive algorithm for i=1, 2, ……n.

δ_i = N_i (1) / [2U_{i+1}(1)] (2.27)

N_{i+1}(z) = [N_j(z)-(z+1)U_{j+1}(z)] / (z-1) (2.28)

Where

N1(z) = (z+1) N (z).

2.3.2 Bilinear Routh Approximation in Frequency-Domain:

H_m(z) be the mth order (m<n) model obtained by truncating the bilinear Routh $\gamma - \delta$ expansion [13]. After the first m terms truncation eliminates terms containing

$\gamma_{m+1,}$ ………….. $\gamma_{n,}$, $\delta_{m+1,}$ ……………, δ_n in the bilinear Routh $\gamma - \delta$ expansion and hence H_m(z) depends only on the first m terms γ and δ coefficients.

Then the mth order bilinear Routh approximation H_m (z) is given by

$$\mathrm{H_m(z)} = \frac{1}{\mathrm{z}-1}\sum_{\mathrm{i=1}}^{\mathrm{m}} \delta \mathrm{i} \prod_{\mathrm{j=1}}^{\mathrm{i}} \mathrm{W_{jm}(z)} \tag{2.29}$$

$\mathrm{H_m(z)}$reduces to

$$\mathrm{H_m(z)} = \frac{d_{m.1} + d_{m.2} + ...d_{m.m^{Z^{m-1}}}}{C_{m.1} + C_{m.2^{Z+}....+C_{m.m}} +_1 Z^m} \underset{=}{\Delta} \frac{B_m(z)}{A_m(z)} \tag{2.30}$$

Where,

$$\mathrm{B_m(z)} = \delta_{\mathrm{m}}(\mathrm{z}-1)^{\mathrm{m}-1} + \gamma_{\mathrm{m}}(\mathrm{z}+1)\mathrm{A_{m-1}(z)} \tag{2.31}$$

$$A_m(z) = (z-1)^2 B_{m-2}(z) + \gamma_m (z+1) A_{m-1}(z) \tag{2.32}$$

With

$$B_0\ (z) = 0,\ B_{-1}(z) = 0 \tag{2.33}$$

$$A_0\ (z) = 1,\ A_1(z) = 1\ /\ (z-1) \tag{2.34}$$

Bilinear Routh approximations possess all the advantages of Routh approximation. Aforesaid bilinear Routh approximation [9] preserves the stability characteristics of the original system in reduced models and the evaluation of reduced-order model of higher-order model. It does not suffer from initial-value problem posed by bilinear transformation because; it introduces the concept of bilinear transformation implicitly.

2.4 MULTIFREQUENCY ROUTH APPROXIMATION FOR LINEAR SYSTEMS:

Hwang, C; Hwang, J.H: and Guo, T.Y. [6] suggested multipoint Routh $\gamma - \delta$ canonical continued fraction expansion for continuous time transfer function of a linear system. They have emphasized that the proposed method can be fully extended for the discrete time case by using bilinear transformation. It can also be extended to yield the reduced-order models for multivariable systems (MIMO).

Their work for continuous-time has been reproduced here.

A multifrequency Routh approximation to the system is derived by selecting the expansion. A connection between the stability preservation property of the multifrequency Routh approximants and the expansion points on the imaginary axis is established.

Multifrequency Routh approximation procedure is flexible in deriving stable reduced-order models while fitting frequency responses and retaining the time moments and for markov parameters of impulse response of the systems. He proposed multifrequency Routh approximation approach of model reduction retains the property of original system at selected frequencies. This method finds its application in dealing with analysis and design of communication systems for which response is of more importance.

2.4.1 Multifrequency Routh $\gamma - \delta$ canonical Expansion:

Continuous-time transfer function can be described as

$$\mathrm{G(s)} = \frac{b_0 + b_1 s + ...+ b_{n-1 s^{n-1}}}{a_0 + a_1 s + a_2 s^2 + ...+ a_n S^n} \underset{=}{\Delta} \frac{B\ (s)}{A(S)}$$

The denominator polynomial A(s) can be split into the even and odd parts:

$$\mathrm{A(s) = A_0(S)^2{+}_sA_1(s)^2} \tag{2.36}$$

Where

$$\mathrm{A_0(s^2) =}\ \mathrm{a_0 + a_2 s^2 + a_4 s^2 + \ldots\ldots + a_{2n_0} +_1 s^{2n_0}} \tag{2.37}$$

$$\mathrm{A_0(s^2) =}\quad \mathrm{a_1 + a_2 s^2 + a_5 s^4 + \ldots\ldots + a_{2n_1} +_1 s^{2n_1}} \tag{2.38}$$

Numerator Polynomials B(s) can be split into the even and odd parts:

$$\mathrm{B(s) = B_1(S)^2{+}sB_2(S^2)} \tag{2.39}$$

Where

$$B_1 (s^2) = \quad b_0 + b_2 s^2 \ldots\ldots + b_{2n_1} +_1 s^{2n_0} \tag{2.40}$$

$$B_2 (s^2) = \quad b_1 + b_3 s^2 \ldots\ldots + b_{2n_1} +_1 s^{2n_1} \tag{2.41}$$

Reduced order modeling using multifrequency Routh approximation carries γ i and δ i

γ i Coefficients are evaluated using following recursive formula

$$\gamma\, i = \frac{A_{i-1}(-W^2)}{A_i(-W_i^2)} \tag{2.42}$$

$$A_{i+1}(s^2) = \frac{A_{i-1}(s^2) - \gamma_i A_i(s^2)}{(s^2 + w_i^2)} \tag{2.43}$$

γ i coefficients are evaluated using following recursive formula :

$$\delta\, i = \frac{B_{i-1}(-w^2)}{A_i(-w^2)} \tag{2.44}$$

$$B_{i+1}(s^2) = \frac{B_{i-1}(s^2) - \delta_i A_i(s^2)}{(s^2 + w_1^2)} \tag{2.45}$$

Where w_i are general frequencies which lie on imaginary axis around which expansions have to be carried out.

2.4.2. Stability Condition of Multi Frequency Routh Algorithm:

Stability condition of multifrequency Routh algorithm in term of the expansion coefficients γ_i and expanding points ω_i are set up. The stability condition for the polynomial

$$F_{n-1}(s) = A_i(s^2)+sA_{i+1}(s^2) \tag{2.46}$$

is first determined. Criterion for the stability test of a polynomial by multifrequency Routh algorithm is in this manner.

The frequency points j ω_i involved in the algorithm may be chosen such that $0 < \omega_i \leq \omega_2 \leq K\,\mathrm{K} < \omega_{\mathrm{n\text{-}1}}$. Efficacy of frequencies which lie on imaginary axis around which expansions have to be carried out depend upon the value of γ_i which put restrains that $\gamma_i > 0$

2.4.3 Reduced-order Modelling Procedure:

Reduced order models obtained by the above method [16] are:

$$G_m(s) = \frac{\hat{\mathrm{B}}_{\mathrm{m}}(s)}{\hat{A}(s)} \tag{2.47}$$

Where

$$\hat{\mathrm{B}}(\mathrm{s}) = \delta_{\mathrm{m}}\mathrm{S}^{\mathrm{m}-1} + (\mathrm{S}^2 + \mathrm{w}^2{}_{\mathrm{m}-1})\hat{\mathrm{B}}(\mathrm{S}) + \delta_{\mathrm{m}}\hat{\mathrm{B}}_{\mathrm{m}-1}(\mathrm{S}) \tag{2.48}$$

$$\hat{\mathrm{A}}(\mathrm{s}) = (\mathrm{S}^2 + \mathrm{w}^2{}_{\mathrm{m}-1})\hat{\mathrm{A}}(\mathrm{S}) + \gamma_{\mathrm{m}}\hat{\mathrm{A}}_{\mathrm{m}-1}(\mathrm{S}) \tag{2.49}$$

Where

$$\hat{\mathrm{A}}_1(\mathrm{s}) = \frac{1}{\mathrm{S}}\hat{\mathrm{A}}_0(\mathrm{s}) = 1, \qquad \hat{\mathrm{A}}_1(s) = S + \gamma_1$$

$$\hat{A}_0(s) = 0, \qquad \hat{\mathrm{B}}_1(s) = \delta_1$$

Multifrequency Routh expansion approximation method retains bans pass and high frequency characteristics of original system. Therefore by selecting an appropriate set of expanding frequency points. The stable multifrequency Routh approximation can be retain not only the time moments and /or markow parameters but also the response characteristics of the system. Moreover, performance of multifrequecy Routh approximants can be made to be optimal by searching optimal expanding frequency points. The meet of stability constraints on choosing the ith expanding frequency ω_i can be guaranteed by checking if

$\gamma_i > 0$ and $\mathrm{A_i(s^2) + sA_{i+1}(s^2)}$ is stable.

2.5 ROBUST DIGITAL CONTROLLER DESIGN FOR PROCESS WITH DEAD TIMES: NEW RESULTS:

Zhang, W.D.; sum, Y.X. and Xn, X.M. [8], suggested the Dahlin controller in studied in the complex frequency domain in terms of performance and Robust stability.

2.5.1 Performance and robust stability:

Consider the unity feedback control loop shown in fig. 2.1, where C(s) is the controller and G(s) is the plant. As the Dahlin controller is designed for first order plants with dead times, the discussion of the paper is also limited to this kind of plant. Assume that

$$\mathrm{G(s)} = \frac{\mathrm{Ke}^{-\theta s}}{\mathrm{Ts+1}} \tag{2.50}$$

and the system input is a unit step. The Dahlin controller is designed by specifying the closed loop transfer function T(s) to be a first-order process with dead time equal to that of the plant G(s), i.e.

$$T(s) = \frac{e^{-\theta s}}{\lambda s + 1} \qquad (2.51)$$

Where λ is the time constant of the closed-loop response. One the other hand, the closed-loop transfer function can be described as.

$$T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} \qquad (2.52)$$

This relation can be rearranged to give an expression of the controller C(s)

$$C(s) = \frac{T(s)}{1 - T(s)} \frac{1}{G(s)} - \frac{1}{k} \frac{Ts + 1}{\lambda s + 1 - e^{\theta s}} \qquad (2.53)$$

In the complex-frequency domain, the dead time is an irrational function. Thus C(s) can not be realised exactly. However, in the discrete domain the dead time is approximated by the finite-dimensional function. Suppose that the sampling time is T_s and $\theta = NT_{s.\ by}$ introducing a zero-order hold, we have

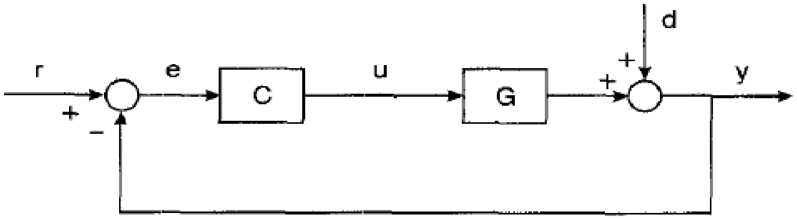

Fig. 2.1 Closed-loop system with unity feed back

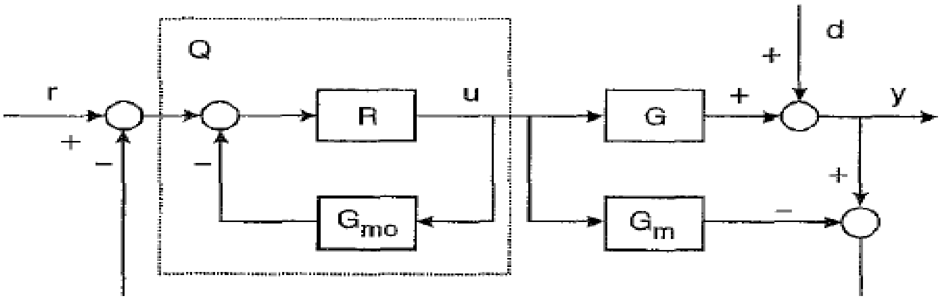

Fig. 2.2: Structure of internal model control

$$T(z)=\frac{(1-e^{-Ts/\lambda})z^{-N-1}}{1-e^{-Ts/\lambda}z^{-1}} \tag{2.54}$$

The corresponding discrete controller can then be obtained directly from

$$c(z)=\frac{(1-e^{-Ts/\lambda})z^{-N-1}}{1-e^{-Ts/\lambda}z^{-1}}\frac{1}{G(s)} \tag{2.55}$$

We will interpret the merits of the Dahlin controller by internal model implementation. In the case of two model-plant mismatch, the classical unity feedback loop show in fig. 2.1 can be converted into the external model control structure shown in Fig. 2.2 with

$$Q(s)=\frac{C(s)}{1+G_m(s)C(s)} \tag{2.56}$$

$$C(s)=\frac{Q(s)}{1-G_m(s)Q(s)} \tag{2.57}$$

Where $G_m(s) = G(s)$ is the model. Therefore

$$T(s)=\frac{C(s)G(s)}{1+C(s)G(s)}=G(s)Q(s) \tag{2.58}$$

The structure shown fig. 2.1 is in fact an open loop.

The performance of the control system design is usually specified as keeping the error between the plant output y and the reference r small, or equivalently, minimizing the effect of the disturbance *d* the plant output *y* normally, we can take

$$Q(s)=\frac{\mathrm{Ts}+1}{\mathrm{K}(\lambda \mathrm{s}+1)} \qquad (2.59)$$

Then the closed-loop system transfer function is

$$\mathrm{T(s)}=\frac{\mathrm{e}^{-\theta\theta}}{\lambda \mathrm{s}+1} \qquad (2.60)$$

When γ tends to zero, the plant output can track the reference input exactly after the delay θ This is in fact the result of optimal control.

Although the Dahlin controller is designed for a nominal plant, it is robust. This can be explained by modern robust theory. We can describe the plant dynamic behavior by a linear time invariant family. Assume that the plant uncertainly profile is L *(w)*, then the family can be represented as

$$\frac{\left|\mathrm{G(jw)}-\mathrm{G_m(jw)}\right|}{\left|\mathrm{G_m(jw)}\right|}\leq \mathrm{L(jw)} \qquad (2.61)$$

From the basic result of modern robust control theory [15], it is know that the closed-loop system is stable if, and only if,

$$\left\|T(jw)L(jw)\right\|\infty<1$$

or equivalently

$$\left\|\frac{1}{\lambda jw+1}e^{-\theta jw}L(w)\right\|_{\infty} < 1 \ \text{ for all w} \qquad (2.62)$$

In other words, by tuning the parameter λ , is obtained for the closed-loop system.

The optimal normal performance is usually defined as min $\int_{0}^{\infty} e^{2}$ dt or min $\left\|W(s)S(s)(s)_{2}^{2}\right\|$, where W(s) is the input weighting function, and S(s) the transfer function form the input r to the error e. Since the system input is a unit step, we can simply let the weight function be 1/s. Observe that

$$\mathrm{S(s)} = \frac{1}{1+\mathrm{C(s)G(s)}} = 1-\mathrm{T(s)} \qquad (2.63)$$

The robust performance can be tested by the sufficient by the sufficient and necessary condition as follows:

$$\left\|\,\|\mathrm{W(s)S(s)}\| + |\mathrm{T(s)T(s)}|\,\right\|_{\infty}$$

$$= \left\|\left|\frac{1}{\mathrm{jw}}[1-\mathrm{G(jw)Q(jw)}]\right| + |\mathrm{G(jw)Q(jw)L(w)}|\right\|$$

$$= \left\|\left|\frac{\lambda}{\lambda \mathrm{jw}+1}\right| + \left|\frac{\lambda}{\lambda \mathrm{jw}+1}\mathrm{e}^{-\theta \mathrm{jw}}\mathrm{L(w)}\right|\right\|_{\infty}$$

< 1 for all w. (2.64)

Hence, if the model is exact, the tunning parameter can be used for optimizing the nomial performance arbitrarily. In the case of a mismatch existing between the model and the plant, performance and stability can always be evaluated by some simple conditions.

2.5.2 Equivalents and extension of the Dahlin controller:

One can find a very large number of digital control algorithms for single-loop systems. Each of them lies to satisfy some commonly accepted criterial.

2.5.2.1 Dead beat control :

Suppose the response of the closed-loop system to a unit sep is required to be unity at every sampling instant after the application of the unit step. If the plant is expressed as a first-order process with dead time, then the desired closed-loop model must have the pulse transfer function.

$$T(z) = Z^{-N-1}$$

The controller can be obtained from the Dahlin controller with the parameter

$$\lambda \rightarrow 0$$

2.5.2.2 Internal model control:

The internal model control provides a framework for the design and tuning of robust controllers. It includes a two-step design procedure which was introduced by Zafirous and Morari. In the first step, $Q_{im}(z)$ is designed so that no offset is produced for the given input, which means tha $Q_{im}(z)$ is equal to:

$$Q_{im}(z) = \frac{1}{G\text{-}(z)} z^{-N-1} \tag{2.66}$$

Where the subscript '_' denotes the delay free part of G(z) with the zeros inside the unit circle. A mismatch between the model and the plant will generate a feedback signal which may cause performance deterioration or instability. Thus, in the second step the filter F (z) is included to take care of the problem. It is of the form

$$\mathrm{F(z)} = \frac{1\text{-}\alpha}{1-\alpha z^{1}} \tag{2.67}$$

Where $\alpha = e^{-Ts/\lambda}, 0 \le \alpha < 1$. For a first-order process with dead time, Q(z) becomes.

Q (z) = Q_{im}(z) F(z)

$$= \frac{(1-e^{-Ts/\lambda})(1-e^{-Ts/Tz^{-1}})}{k(1-e^{-Ts/T})(1-e^{-Ts/\lambda}z^{-1})}$$

and the closed-loop transfer function is given by

$$T(z) = \mathrm{Q}_{\mathrm{im}}(z)F(z)G(s) = \frac{(1-e^{-T_s/\lambda})z^{-N-1}}{(1-e^{-Ts/\lambda}z^{-1})} \tag{2.68}$$

The result us the same as that of the Dahlin controller

2.5.2.3 One-step ahead dynamic matrix-control

Using a step response model, Culter and Ramaker developed the algorithm. They have applied it successfully to the control of chemical processes such as in catalytic cracking units. The scheme is a typical predictive controller. It has three major features in common.

2.5.2.3.3 The predictive law:

The criteria function used for predictive controller design is

$$j = [y\ (k+1)\text{-}w\ (k+1)]^2 \tag{2.69}$$

where the symbol ^ denotes estimation. Let L be the control horizon and L = P. Then the predictive control law can be written as

$$\Delta u = A^{1}\ e \tag{2.70}$$

Where A is the dynamic matrix and e is the error between the reference trajectory and the closed-loop predictive output with zero input. It is easy to find that the result is identical with the Dahlin controller.

As we known, the classic Dahlin controller cannot be used for the control of plants with zeros outside the unit circle and unstable plants. Based on the above discussion, we will develop its extension. Since the expression in the discrete domain is very complex and hard to understand, only the result in the complex frequency domain are given.

2.5.2.3.4 Control of unstable plants:

Consider the following plant

$$G(s) = \frac{1}{T_s} e^{-\theta_s} \tag{2.71}$$

which is a type II system. In order to track the step input, the closed-loop transfer function has to satisfy the following constraints.

$$\lim_{s \to 0} [1 - T(s)] = 0 \tag{2.72}$$

$$\lim_{x \to 0} \frac{d}{ds} [1 - T(s)] = 0 \tag{2.73}$$

For stability, we can take Q(s) = $sQ_1(s)$. The closed-loop system is stable if, and only if, $Q_1(s)$ is stable. Let $Q_1(s)$ be of the form.

$$Q_1(s) = \frac{B(s)}{(\lambda_s + 1)^n}$$

where B(s) is a polynomial. Combining it with eqns. 28 and 29 yield n=2 and B(s) = T[(2λ+θ),+1] Then one can obtain the described closed-loop transfer function

$$T(s) = \frac{(2\lambda + \theta)_s + 1}{(\lambda_s + 1)^2} \qquad (2.74)$$

The result in the controller

$$\mathrm{C(s)} = \frac{\mathrm{T_s[(2\lambda + \theta)_s + 1]}}{(\lambda_s + 1)^2 - [(2\lambda + \theta)_s + 1]e^{-\theta_s}} \qquad (2.75)$$

Remarks:

Most of the model reduction techniques proposed by various investigators for continuous-time systems are equally applicable to discrete-time systems by employing suitable transformation techniques like bilinear transformation, p-domain transformation as well as z=p/(A+Bp) where A and B are constants. Therefore, multifrequncy Routh approximation approach [10] and reduction be extended for discrete-time case.

2.6 A MIXED METHOD FOR THE REDUCTION OF DISCRETE TIME SYSTEMS

The problem of reducing a high order system to a lower order model is considered important in analysis, synthesis and simulation of practical systems. It is known that reduction methods based on power-series expansion, like pade approximation, or continued-fraction may often lead to instability even when the original system is stable. One way of overcoming the instability problem is by using the stability-equation method, where reduced models are guaranteed to be stable when the original system is stable. Often, the bilinear transformation is used to extend continuous-time reduction methods to reduce z-transfer functions in the ‘w’ domain. Such methods suffer from the drawback that due to the nature of the bilinear transformation, the initial value of the step response may not be zero even though the initial value to the step response of the original system is zero. Shanti Mishra, Jayanta Pal [10], proposed a new method combining the advantages of stability-equation and Pade approximation technique that retains the important fast poles while ensuring identical responses at zero time.

2.6.1 THE PROPOSED METHOD

The procedure consists of four steps: (1) the use of bilinear transformation to convert the original z-domain high order system to w-domain, (2) the denominator polynomial is then reduced by using the stability equation method along with reciprocal transformation, (3) the numerator dynamics in w-domain are chosen to fit a number of the initial time-moments and Markov parameters, (4) the reduced order model in z-domain is finally obtained by using the reverse bilinear transformation.

Step (1): Transformation of the system and the model

Let the nth-order system transfer function be given by G (z). By using the bilinear transformation $z = {}^{1+w}/_{1-w,}$ separately on the numerator and the denominator of G (z) = ${}^{N(w)}/_{D(w)}$, the w-domain G(w) can be obtained as :

$$G(w) = \frac{N(w)}{D(w)} = \frac{\sum_{j=1}^{n} a_{j-1^{w^{j-1}}}}{\sum_{j=1}^{n+1} b_{j-1^{w^{j-1}}}} = \frac{a_0 + a_1 w + \ldots + a_{n-1^{w^{n-1}}}}{b_0 + b_1 w + \ldots + b_{n^{w^n}}} \qquad (2.76)$$

In this case, the pole-zero excess of G (w) remains the same as that of G(z). Hence, the step response of G (z) and the reduced model R_r (z) will match exactly at t=0. As detailed below, G(w) can then be reduced to give an r-th order reduced model R_r (z) which may be converted back into the z-domain by using the bilinear reverse transformation separately on the numerator and the denominator of R_r (z).

Let the rth –order reduced model be R_r (z), which, in the w-domain may be written as:

$$R_r(w) = \frac{N_r(w)}{D_r(w)} = \frac{\sum_{j=1}^{r} n_{j^{w^{j-1}}}}{\sum_{j=1}^{r+1} d_{j^{w^{j-1}}}} = \frac{n_1 + n_2 w + \ldots + n_{r^{w^{r-1}}}}{d_1 + d_2 w + \ldots + d_{r+1^{w^r}}} \qquad (2.77)$$

Step (2): Reduction of the denominator

Let D (w) from (1) and D_r (w) to be found out, be given by

$$D(w) = \sum_{j=1}^{n+1} b_{j-1^{w^{j-1}}} \qquad (2.78)$$

$$D_r(w) = \sum_{j=1}^{r+1} d_{j^{w^{j-1}}} \qquad (2.79)$$

For stable G (w), the stability equations of D(w) can be written as (5), (6)

$$D_e(w) = b_n \prod_{i=1}^{m} (w^2 + z_i^2) \qquad (2.80)$$

$$D_o(w) = b_{n\text{-}1} \prod_{i=1}^{m} (w^2 + p_i^2) \qquad (2.81)$$

Where m is the integer part of n/2 and (n-1)/2 for n even and odd respectively and the pole-zero patterns is

$$p_1^2 < p_2^2 < p_3^2 < p_4^2 < \&z_1^2 < z_2^2 < z_3^2 < z_4^2 \ldots$$

The reduced stability equation of order r1 can be obtained by discarding the poles (p_i)

and zeroes (z_i) with larger magnitudes .The results are

$$D_r1(w) = b_n \prod_{i=1}^{r_1/2}(1+w^2/z_i^2)wb_{n-1} \quad \prod_{i=1}^{(r_1-1)/2}(1+w^2/p_i^2) \tag{2.82}$$

The reciprocal polynomial of D (w), $\tilde{D}$ (w) be defined by

$$\tilde{D}(w) = w^n D(\frac{1}{w}) = \sum_{j=1}^{n+1} b_{\; j-1^{w^{n+1-j}}} = b_0 w^n + b_1 w^{n-1} + ... + b_n \tag{2.83}$$

The reciprocal polynomial in Eqn. 2.83 has the property that it inverts the roots of the original polynomial and thus the small magnitude roots of D (w) will become the large magnitude roots of D (w) and vice versa.

It is known that only poles nearest to the origin are retained in Dr_1 (w) and no consideration is given to poles which have large negative real parts. To ensure that $D_r(w)$ also approximates some large magnitude poles of G(w), stability equations similar to Eqns. (2.80,2.81) are constructed for the reciprocal polynomial of Eqn. (2.83) and a reduced polynomial $D_{r2}(w)$ of order r_2 is formed. D_r (w) is then found as

D_r (w) =D_r1 (w) xDr2 (w) (2.84)

Step (3): Reduction of the numerator

Using the power series expansions of G (w) about w=0 and w=∞, the modified time moments (t_i) and the Markov parameters (m_i) are obtained as

$$G(w) = \sum_{i=0}^{\infty} t_i w^i \tag{2.85}$$

$$G(w) = \sum_{i=0}^{\infty} m_i w^{-i-1} \qquad (2.86)$$

The coefficients of Nr (w) of Eqn. (2.77) are determined form the following set of equations with the assumption that R(w) and G(w) have identical first α time-moments (t_0,t_1--) and first β Markov parameters (m_0,m_1--)

Choosing α+β=r, we have,

$$\begin{aligned}
n_1 &= d_1 t_0 \\
n_2 &= d_1 t_1 + d_2 t_0 \\
\vdots \quad & \quad \vdots \qquad \vdots \\
n_\alpha &= d_1 t_{\alpha-2} + \ldots + d_{\alpha-1} t_1 + d_\alpha t_0 \\
n_{r-}\beta + 1 &= d_{r+1^m}\beta - 1 + d_{r^m}\beta - 2 + \ldots + d_{r-}\beta + 3m_1 + d_r - \beta + 2^m{}_0 \\
\vdots \quad & \quad \vdots \qquad \vdots \\
n_{r-1} &= d_{r+1} m_1 + d_r m_0 \\
n_r &= d_{r+1^{m_0}}
\end{aligned}$$

Step (4) To obtain the reduced model Rr(z) from Rr(w)

By using the inverse bilinear transformation $w = \frac{z-1}{z+1}$, separately on the numerator and the denominator of Rr (w) =Nr (w)/Dr(w), the z-domain reduced order model R_r (z) may be obtained. For steady-state matching, the final reduced model is obtained by multiplying the coefficients of the numerator polynomial of R_r (z) by a constant $k = \frac{G(1)}{R(1)}$

2.7 DIFFERENTIAL EVOLUTION ALGORITHM FOR REDUCTION OF SISO DISCRETE TIME SYSTEMS

Reduction of high order systems to lower order systems (LOS) has been an important subject area in control engineering for many years. The conventional methods of reduction, developed so far, are mostly available in continuous domain. However, the high order systems can be reduced in continuous as well as in discrete domain.

J. S. Yadav, N. P. Patidar, J. Singhai and S. Panda [19], suggested two methods of model reduction of Single-input (SISO) discrete system have been presented. The first method which is based on the conventional approach combines the advantages of Modified Cauer Form (MCF) and differentiation of the denominator polynomials. This transformation is accomplished using synthetic division .The denominator of reduced continuous system is derived using differentiation, of both, the original and reciprocal polynomials in *w*-domain and multiplying various derivatives of these two polynomials .The numerator is found by matching the quotients of MCF. After obtaining the Reduced Order Model (ROM) of continuous system is conversion into discrete system is accomplished by using inverse bilinear transformation, separately on numerator and denominator polynomials to give the desired result. Also a steady state correction is applied to match the final values of response of original and reduced systems.

In the second method, Differentional Evolution (DE) is employed for the order reduction where both the numerator and denominator coefficients of low order system are determined by minimizing an ISE error criterion. Differential Evolution (DE) is a branch of evolutionary alogorithms optimization problems .DE uses weighted differences employs a greedy selection process with inherent elitist features. Also it has a minimum number of control parameters, which can be tuned effectively.

2.7.1. STATEMENT OF THE PROBLEM

Given a high order discrete time stable system of order ‘n’ that is described by the z-transfer function:

$$G_0(z) = \frac{N(z)}{D(z)} \frac{a_0 + a_1 z + ... + a_{n-1} z^{n-1}}{b_0 + b_1 z + ... + b_{n-1} z^{n-1} + b_n z^n} \tag{2.89}$$

The objective is to find a reduced rth order model that has a transfer function (r<n):

$$R_0(z) = \frac{N_r(z)}{D_r(z)} = \frac{c_0 + c_1 z + ... + c_{r-1} z^{r-1}}{d_0 + d_1 z + ... + d_{r-1} z^{r-1} + d_r z^r} \tag{2.90}$$

The polynomial D(z) is stable, that is all its zeros reside inside the unit circle $|z| = 1$. Where $a_i(0 \le i \le n\text{-}1)$, $b_i(0 \le i \le n), c_i(0 \le i \le r-1)$, and $d_i(0 \le i \le r)$, are scalar constants.

The numerator order is given as being one less than that of the denominator, as for the original system, The R(z) approximates G$_0$(z) in some sense and retains the important characteristics of G$_0$(z)

2.7.2 REDUCTION BY CONVENTIONAL METHOD

The reduction procedure by conventional method (modified Cauer Form and differentiation) may be described in the following steps:

Step-1

Apply bilinear transformation $z = \frac{1+w}{1-w}$, separately in the numerator and denominator polynomials of Eq. (2.89) using synthetic division. This converts G$_0$(z) into G(w) as:

$$\mathrm{N(w) = N(z)}\Big|\mathrm{z} = \frac{1+\mathrm{w}}{1-\mathrm{w}} = \frac{\mathrm{N(w)}}{(1-\mathrm{w})^{\mathrm{n}-1}} = 0 \tag{2.91}$$

$$\mathrm{D(w) = D(z)}\Big|\mathrm{z} = \frac{1+\mathrm{w}}{1-\mathrm{w}} = \frac{\mathrm{D(w)}}{(1-\mathrm{w})^{\mathrm{n}}} = 0 \tag{2.92}$$

From Eqs. (2.91) and (2.92) we get G(w) = $\frac{\mathrm{N(w)}}{\mathrm{D(w)}}$

This can be expressed as:

$$\mathrm{G(w)} = \frac{\mathrm{a}_{11} + \mathrm{a}_{12}\mathrm{w} + ... + \mathrm{a1}_{\mathrm{n-1}}\mathrm{w}^{\mathrm{n-1}}}{\mathrm{b}_{11} + \mathrm{b}_{12}\mathrm{w} + ... + \mathrm{b1}_{\mathrm{n-1}}\mathrm{w}^{\mathrm{n-1}} + \mathrm{b1}_{\mathrm{n}}\mathrm{w}^{\mathrm{n}}} \qquad (2.93)$$

The reciprocal of D (w) is given as:

$$\tilde{D}(w) = w^n D(1/w) = b_{1n-1} + b_{1n-1}{}^{w} + ... + b_{12}w^{n-1} + b_{11^{w^n}} \qquad (2.94)$$

Step-2

Compute the questions of Modified Cauer form (MCF) h_1, h_2, ……,H_1, H_2…..using Modified Routh Array.

Step-3

Differentiate successively the denominator of the Eq. (2.93), and its reciprocal, the rth order denominators of reduced order models can be obtained by multiplying various combinations of differentiated polynomials. In particular D (w) is differentiated (n-r_1) times and its reciprocal $\tilde{D}(w)$ is differentiated (n-r_2) times. The rth order reduced denominator is obtained as:

$$D_r(w)=D_{r1}(w).D_{r2}(w) \qquad (2.95)$$

Where $\tilde{D}r_2(w)$ is the reciprocal of $Dr_2(w)$ and r = r_1+r_2

Step-4

Match the quotients h1, h2, ……., H1, H2….to find out the numerator N(w) of the ROM as given in .

The ROM G (w) will be obtained as G (w) = $\frac{N(w)}{D(w)}$

Step-5

Apply the inverse bilinear transformation w=$\frac{z-1}{z+1}$ separately in the N (w) and D(w) to convert G(w) in z domain. Thus the rank of G_0 (z) and G (w) will be same. Hence the step response of G_0 (z) and G (w) will match at initial time t=0.

Step-6

Remove steady state error by evaluating k=$\left.\frac{G0(z)}{R(z)}\right|_{z=1}$, and multiply it in the numerator of R (z).

2.7.3. DIFFERENTIAL EVOLUTION (DE)

In conventional mathematical optimization techniques, problem formulation must satisfy mathematical restrictions with advanced computer algorithm requirement, and may suffer from numerical problems. Further, in a complex system consisting of number of controllers, the optimization of several controller parameters using the conventional optimization is very complicated process and sometimes gets struck at local minima resulting in sub-optimal controller parameters. In recent years, one of the most promising research field has been "Heuristics from Nature", an area utilizing analogies with nature or social systems. Application of these heuristic optimization methods a) may find a global optimum, b) can produce a number of alternative solutions, c) no mathematical restrictions on the problem formulation, d) relatively easy to implement and e) numerically robust. Several modern heuristic tools have evolved in the lat two decades that facilitates solving optimization problems that were previously difficult or impossible to solve. These tools include evolutionary computation, simulated annealing, tabu search, genetic algorithm, particle swarm optimization, etc. Among these heuristic, techniques, Genetic Algorithm (GA), particle Swarm Optimization (PSO) and Differential Evolution (DE) techniques appeared as promising algorithms for handling the optimization problems. These techniques are finding popularity within research community as design tools and problem solves because of their versatility and ability to optimize in complex multimodal search applied to non-differentiable objective functions.

Differential evolution (DE) is a stochastic, population-based optimization algorithm introduced by Storn and Price in 1996. DE works with two populations; old generation and new generation of the same population. The size of the population is adjustable by the parameter N_p. The population consists of real valued vectors with dimension D that equals the number of design parameters/control variables. The population is randomly initialized within the initial parameter bounds. The optimization process is conducted by means of three main operations: mutation, crossover and selection. In each generation, individuals of the current population become target vectors. For each target vector, the mutation operation produces a mutant vector, by adding the weighted difference between two randomly chosen vectors to a third vector. The crossover operation generates a new vector, called trial vector, by mixing the parameters of the mutant vector with those of the target vector. If the trial vector obtains a better fitness value than the target vector, then the trial vector replaces the target vector in the next generation. The evolutionary operators are described below.

2.7.3.1. Initialization

In DE, a solution or an individual *i*, in generation G is a multidimensional vector given as:

$$X_i^G = (X_{i,1,\ldots} X_{i,D}) \tag{2.96}$$

$$X_{i,k}^{G} = X_{k\min} + \mathrm{rand(0,1)x(X_{k\max}\text{ - }X_{k\min})} \qquad (2.97)$$

With $i \in [1, N_p], \mathrm{k} \in [1, \mathrm{d}]$

Where, N_p is the population size, D is the solution's dimension i.e. number of control variables and rand (0,1) is a random number uniformly distributed between 0 and 1. Each variable k in a solution vector i in the generation G is initialized within its boundaries $X_{k\min}$ and $X_{k\max}$.

2.7.3.2. Mutation

DE does not use a predefined probability density function to generate perturbing fluctuations. It relies upon the population itself to perturb the vector parameter. Several population members are involved in creating a member of the subsequent population. For every $i \in [1, N_p]$the weighted difference of two randomly chosen population vectors, X_{r2} and X_{r3}, is added to another randomly selected population member, X_{r1}, to build a mutated vector V_i.

$$V_i = X_{r1} + F.(X_{r2} - X_{r3}) \qquad (2.98)$$

With r_1, r_2, $r_3 \in [1, N_p]$ are integers and mutually different, and F>0, is a real constant to control the differential variation $d_i=X_{r2}-X_{r3}$.

2.7.3.3.. Crossover

The crossover function is very important in any evolutionary algorithm. It also should be noted that there are evolutionary algorithms that use mutation as their primary search tool as opposed to crossover operators. In DE, three parents are selected for crossover and the child is a perturbation of one of them whereas in GA, two parents are selected for crossover and the child is a recombination of the parents. The crossover operation in DE can be represented by the following equation

$$U_i(j) = \begin{cases} V_i(j), & \text{if } \mathrm{U_i}(0{,}1) < CR \\ X_i(j), & \text{otherwise} \end{cases} \tag{2.99}$$

2.7.3.4. Selection

In DE algorithm, the target vector $X_{i,G}$ is compared with the trial vector $V_{i,G+1}$ and the one with the better fitness value is admitted to the next generation. The selection operation in DE can be represented by the following equation:

$$X_i G = \begin{cases} U_i G+1, & \text{if } \mathrm{f}(U_{i,G+1}) < f(X_{i,G}) \\ X_i \mathrm{G} & \text{otherwise} \end{cases} \tag{2.100}$$

Where I $\in$ [1,N_p].

CHAPTER 3

MODEL REDUCTION BY MODIFIED ROUTH STABILITY METHOD USING P-DOMAIN TRANSFORMATION.

3.1 INTRODUCTION:

Reduction of high-order system transfer function to low-order Models has been an important subject and in the control engineering environment for many years. The Routh-stability method is employed to reduce the order of discrete time system transfer function. The Routh approximation is well suited to reduce both the denominator and numerator polynomials. Farsi, M; Warwick, K; and Guilandoust M, [4] employed this method to have order reduction of discrete time systems incorporating p-domain transformation which resolves the short comings of bilinear transformation

Unfortunately, algorithms suggested [4], leads to erroneous result, Vivek kumar sehgal [13], suggested Modified Routh stability which resolves lacunae of Routh stability Method. In this chapter Modified Routh stability method has been applied to quite higher order system comparatively. This proposed method presents rectified value of α, β and expansion point m (z=p+m) and leads to correct value.

3.2 MODEL REDUCTION IN THE DISCRETE TIME DOMAIN:

It is assumed that a higher-order transfer-function relating system input to system output can be expressed by

$$G(z) = \frac{D(z)}{E(z)}$$

$$= \frac{d_{11}z^{n-1} + d_{21}z^{n-2} + d_{12}z^{n-3} + \Lambda}{e_{11}z^{n} + e_{21}z^{n-1} + e_{12}z^{n-3} + \Lambda} \qquad (3.1)$$

Where d and e coefficients are the numerator and denominator scalar constants respectively, also numerator order is given as being one less than the denominator.

Assuming that a reduced order model R (z) of order k (k<n) is to be constructed.

It assumes the form of

$$R(z) = \frac{B(z)}{A(z)}$$

$$= \frac{b_{k-1}z^{k-1} + b_{k-2}z^{k-2} + \Lambda + b_0}{a_k z^k + a_{k-1}z^{k-1} + \Lambda + a_0} \qquad (3.2)$$

The overall modeling procedure necessary in this modified Routh stability method incorporates following steps:

STEP 1: Given transfer G (z) is transformed into G (p) which is governed by (z=p+m) transformation where m is a scalar quantity equal to the distance from the farthest pole or zero to the centre of unit circle. This is performed by substituting z=p+m in equation 3.1.The transfer function in z-domain is converted into p-domain with the help of Pascal's triangle.

therefore

$$G(p) = \frac{d'_{11} + d'_{21}p + d'_{12}p^2 + d'_{22}p^3 + \Lambda}{e'_{11} + e'_{21}p + e'_{12}p^2 + e'_{22}p^3 + \Lambda} \qquad (3.3)$$

STEP 2: Routh arrays for the numerator and denominator are constructed by arrangements of parameters contained in Eq. (3.3) (Tables 3.1 and 3.2):

Table 3.1: ROUTH ARRAY FOR E (p)

e'_{11}	e'_{12}	e'_{13}	e'_{14}
e'_{21}	e'_{22}	e'_{23}	e'_{24}
e'_{31}	e'_{32}	e'_{33}	
$e'_{(n-1).1}$	$e'_{(n-1).2}$	…….	
$e'_{n.1}$	…….		
$e'_{(n+1).1}$	……..		

The first two rows of the arrays are obtained from Eqn. (3.3), where as other elements are determined from equation:

$$e'_{(i,j)} = e'_{(j-2),(j+1)} - \lfloor e'_{(j-2).1} e_{(i-1),(J+1)} \rfloor / \lfloor e'_{(i-1).1} \rfloor \qquad (3.4)$$

Table 3.1: ROUTH ARRAY FOR E (p)

d'_{11}	d'_{12}	d'_{13}	d'_{14}
d'_{21}	d'_{22}	d'_{23}	d'_{24}
d'_{31}	d'_{32}	d'_{33}	…..
$d'_{(n-1).1}$	$d'_{(n-1).2}$	…….	

$d'_{(n).1}$ …….

$d'_{(n+1).1}$ ……..

Where

These first two rows of Table (3.2) are determined form Eq. (3.3) and other entries are obtained from equation.

$$d'_{(i,j)} = d'_{(i-2),(j+1)} - \lfloor d'_{(i-2).1} d'_{(i-1),(J+1)} \rfloor / \lfloor d'_{(i-1).1} \rfloor \tag{3.5}$$

for $\mathrm{i} \geq 0$ and $\mathrm{i} \leq \mathrm{j} \leq \{(\mathrm{n} - \mathrm{i} + 3)/2\}$ in which {.} indicates the integer part of the quantity.

STEP 3: Desired values of α_i, β_i and γ can be calculating by using.

$$\alpha_i = \frac{e'i,1}{e'_{(k+1),1}}, \qquad \text{(Where i=1, ………k+1)} \tag{3.6}$$

Where k is the desired order of reduced model

$$\beta_i = \frac{d'_{i,1}}{e_{k,1}}, \qquad \text{(Where i=1, ………k)} \tag{3.7}$$

$$\gamma = \frac{d'k,1}{e'_{(k+1),1}}, \qquad \text{for stable system} \tag{3.8}$$

STEP 4: The model denominator and numerator are calculated using following equations:

$$A(z)=\sum_{i=0}^{k}\alpha_{i+1}(z-m)^{i}$$ (where i=0, 1, 2,……, k) (3.9)

$\alpha_{k+1}=1$

$$B(z)=\sum_{i=0}^{k-1}\beta_{i+1}(z-m)^{i}$$ (where i=0, 1, 2,……, k) (3.10)

$\beta_{k+1}=1$

STEP 5: Desired reduced model assumes following form:

$$R(z)=\gamma\frac{\mathrm{B(z)}}{A(z)}$$

Where γ gain correction factor is is defined as ratio of original system steady-state gain to reduced order model steady state gain.

3.3 Numerical example 3.1:

A SISO 8th order – linear time-invariant system whose transfer – function, relating input to output, in discrete-time domain is represented by:

$$G(z)=\frac{0.2125z^{7}+0.1395z^{6}-0.02625z^{5}+0.019z^{4}-0.0645z^{3}-0.03525z^{2}+0.0055z-0.00075}{z^{8}-0.63075z^{7}-0.4185z^{6}+0.07875z^{5}-0.057z^{4}+0.1935z^{3}+0.09825z^{2}-0.01625z+0.00225}$$

STEP 1: Given G (z) is transformed into G(P) which results

$$\mathrm{G(P)}=\frac{\mathrm{D(P)}}{\mathrm{E(P)}}$$

$$= \frac{0.2125p^{7} + 1.627p^{6} + 5.2732p^{5} + 36.9802p^{4} + 9.9765p^{3} + 5.8605p^{2} + 1.3763p - 0.0675}{p^{8} + 7.3693p^{7} + 23.1662p^{6} + 40.3220p^{5} + 41.9830p^{4} + 26.3068p^{3} + 9.6010p^{2} + 2.0002p + 0.2502}$$

STEP 2: Routh array for numerator Polynomials is obtained by using Table 3.2 and eq. (3.5):

0.2498	6.1777	9.4177	1.6270
2.0108	9.9765	5.2732	0.2125
4.938327	8.76261478	1.6006013	
6.40851719	4.62146326	0.2125	
5.201370242	1.436851332		

Routh array for Denominator polynomials is obtained by using Table (3.1) and Eq. (3.4)

0.2502	9.6010	41.9830	23.1662
2.0002	26.3068	40.3220	7.3693
6.310348	36.939222	22.24439275	
14.598122	33.27116312		
22.55705574			

3.3.1 Reduced Model (1st-Order):

α_1 = 0.125087 α_2=1

β_1=1 γ= 0.1248875

STEP 4: Numerator of the reduced order would be based on Eq. (3.10) and Denominator of the reduced order would be based on Eq. (3.9) :

B(z)=1

A (z) = (z-0.874913)

STEP 5: Desired reduced model will assume the following for after incorporating gain correction factor where gain correction factor is obtained by using Eq. (3.8):

$$R_1(z) = \frac{0.1248875}{z - 0.874913}$$

3.3.2 Reduced Model (2nd –Order):

First two steps will remain same which are described for reduced model (1st order).

STEP 3: Using Eqs. (3.6), (3.7) and (3.8) for the 2nd order model:

α_1 = 0.039649 α_2 = 0.31697 α_3 = 1

β_1 = 0.124229 β_2 = 1

γ = 0.31865

STEP 4: Numerator of the 2nd order reduced model would be based Eq. (3.10):

B (z) = z-0.875771

Denominator of the 2nd order reduced model would be based on Eq. (3.9)

A (z) = z^2-1.68303+0.722679

STEP 5: Desired reduced model will assume the following form $\mathrm{R(z)} = \gamma \dfrac{\mathrm{B(z)}}{\mathrm{A(z)}}$

after incorporating gain correction factor (γ) which is described by Eqn. (3.8):

$$R_2(z) = \frac{0.31865z - 0.27906443}{z^2 - 1.68303z + 0.722679}$$

3.3.3 Reduced Model (3rd –Order):

First two steps would remain same which are described for the 1st order reduced Model and 2nd order reduced model.

STEP 3: Using Eqs. (3.6), (3.7) and (3.8) for the 3rd order reduced model:

α_1 = 0.017139 α_2 = 0.1370176 α_3 = 0.432272 α_4=1

β_1 = 0.0505839 β_2 = 0.40718 β_3=1

γ = 0.338285

STEP 4: Numerator of reduced order model (3rd order) would assume the following form which is generated by Eq. (3.10):

B (z) =z^2-1.59282z+0.6434039

Denominator of reduced order model (3rd order) would assume the following form which is generated by Eq. (3.9):

A (z) =z^3-2.5677288z^2+2.2724752z-0.0.6876074

STEP 5: Desired model will assume the following form where $\mathrm{R(z)} = \gamma \frac{\mathrm{B(z)}}{\mathrm{A(z)}}$

after incorporating gain correction factor which is described by Eq. (3.8):

$$R_3(z) = \frac{0.338285z^2 - 0.538827113z + 0.21765388}{z^3 - 2.5677288z^2 + 2.2724752z - 0.6876074}$$

Table 3.3

DERIVED REDUCED – ORDER MODELS

1. $R_1(z) = \dfrac{0.124625}{z - 0.875}$; Modified Routh stability method (1st –Order reduced MRSM)

2. $R_2(z) = \dfrac{0.324133z - 0.283646}{z^2 - 1.67514z + 0.715748}$; Modified Routh stability method (2nd –Order reduced MRSM)

3. $R_2(z) = \dfrac{0.324133z - 0.283646}{z^2 - 1.67514z + 0.715748}$; Modified Routh stability method (3rd-Order reduced MRSM)

3.4 MATLAB ANALYSIS AND PLOTS (STABLE SYSTEM)

ABOUT MATLAB:

In the few years, matlab has become the most widely used software package in academia and industry for calculation and response from the transfer function. It is very useful and powerful software to find out the higher order transfer function responses.

We write the command in a MATLAB's command window and makes an .m file. Saves the program in .m files and gives the run command, and then MATLAB gives the desired result. Through MATLAB we can find out gain margin, phase margin, poles, zeros, Bode plot, Nyquist Plot etc.

Transfer function of an eight order stable system is considered. Model reduction is carried out and reduced 2nd and 3rd order systems are obtained

3.4.1 Script file(Stable System)

```
  %Script file
   %Purpose:
 %Step response of original 8th order system and reduced 2nd and 3rd
order system

   clc;
   clear all;
   n1=[0.2125 0.1395 -0.02625 0.019 -0.0645 -0.03525 0.0055 -0.00075];
   d1=[1 -0.6307 -0.4185 0.07875 -0.057 0.1935 0.0982 -0.0162 0.00225];
   [z p k]=tf2zp(n1,d1)
   r=ones(1,101)
   k=0:100
   y=filter(n1,d1,r)
   plot(k,.99*y,'-');
   v=[0 100 0 2]
   axis(v);
   title('unit step response');
   xlabel('k');
   ylabel('y(k)');
   hold on;
   n2=[0 0.318672001 -0.279090585];
   d2=[1 -1.682992082 0.72265274];
   r=ones(1,101);
   k=0:100;
   y=filter(n2,d2,r);
   plot(k,0.99*y,'.');
   v=[0 100 0 2];
   axis(v);
   title ('unit step response')
```

```
xlabel('k');
ylabel('y(k)');
hold on;
n3=[0 0.338329368 -0.538907862 0.21768817];
d3=[1 -2.567734622 2.272500791 -0.68762224];
r=ones(1,101);
k=0:100;
y=filter(n3,d3,r);
plot(k,0.99*y,'-.')
v=[0 100 0 2];
axis(v);
title ('unit step response')
xlabel('k');
ylabel('y(k)');
gtext('__original 8th order system')
gtext('...reduced 2nd order system')
gtext('-.-reduced 3rd order system')
```

3.4.2. Results in command window(Stable System)

```
z =

   0.7121

   0.0097 + 0.7378i

   0.0097 - 0.7378i

  -0.9840

  -0.5505

   0.0733 + 0.1069i

   0.0733 - 0.1069i
```

```
p =

   0.8798 + 0.2448i

   0.8798 - 0.2448i

  -0.0539 + 0.6554i

  -0.0539 - 0.6554i

  -0.5872 + 0.0945i

  -0.5872 - 0.0945i

   0.0766 + 0.1085i

   0.0766 - 0.1085i

k =

    0.2125

r =

  Columns 1 through 14

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 15 through 28

     1     1     1     1     1     1     1     1     1     1     1     1     1     1
  Columns 29 through 42

    1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 43 through 56

    1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 57 through 70

    1     1     1     1     1     1     1     1     1     1     1     1     1     1



  Columns 71 through 84

    1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 85 through 98

    1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 99 through 101

    1     1     1

k =

  Columns 1 through 14

    0     1     2     3     4     5     6     7     8     9    10    11    12    13
  Columns 15 through 28

   14    15    16    17    18    19    20    21    22    23    24    25    26    27

  Columns 29 through 42

   28    29    30    31    32    33    34    35    36    37    38    39    40    41

  Columns 43 through 56

   42    43    44    45    46    47    48    49    50    51    52    53    54    55

  Columns 57 through 70

   56    57    58    59    60    61    62    63    64    65    66    67    68    69

  Columns 71 through 84

   70    71    72    73    74    75    76    77    78    79    80    81    82    83

  Columns 85 through 98

   84    85    86    87    88    89    90    91    92    93    94    95    96    97

  Columns 99 through 101

   98    99   100

y =

  Columns 1 through 8
  0.2125    0.4860    0.7212    0.9863    1.1781    1.3306    1.4313    1.4890

Columns 9 through 16

  1.4960    1.4654    1.4057    1.3259    1.2344    1.1406    1.0519    0.9739

Columns 17 through 24

  0.9106    0.8643    0.8357    0.8238    0.8269    0.8422    0.8665    0.8966

Columns 25 through 32

  0.9292    0.9615    0.9912    1.0164    1.0361    1.0497    1.0572    1.0590

Columns 33 through 40

  1.0560    1.0492    1.0397    1.0287    1.0172    1.0062    0.9965    0.9884

Columns 41 through 48

  0.9824    0.9786    0.9768    0.9769    0.9786    0.9814    0.9850    0.9890

Columns 49 through 56

  0.9929    0.9966    0.9998    1.0023    1.0041    1.0051    1.0055    1.0052

Columns 57 through 64

  1.0044    1.0033    1.0020    1.0006    0.9992    0.9980    0.9970    0.9962

Columns 65 through 72
```

```
  0.9957    0.9954    0.9954    0.9956    0.9959    0.9963    0.9968    0.9973

 Columns 73 through 80

  0.9978    0.9982    0.9985    0.9987    0.9989    0.9989    0.9989    0.9988

 Columns 81 through 88

  0.9987    0.9985    0.9983    0.9982    0.9980    0.9979    0.9978    0.9977

 Columns 89 through 96

  0.9977    0.9977    0.9977    0.9977    0.9978    0.9978    0.9979    0.9980

 Columns 97 through 101

  0.9980    0.9981    0.9981    0.9981    0.9981

v =

     0   100     0     2
```

3.4.3. Plots (Stable System)

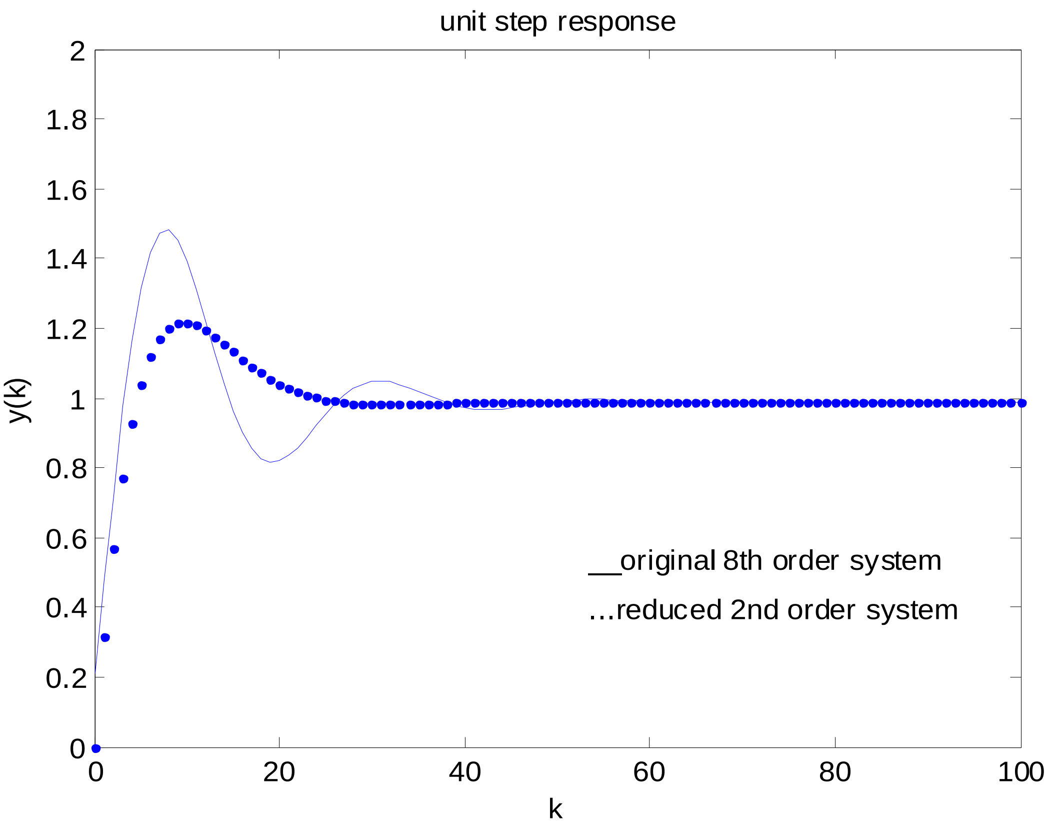

Fig.3.1: Step response of original 8th order system and reduced 2nd order system

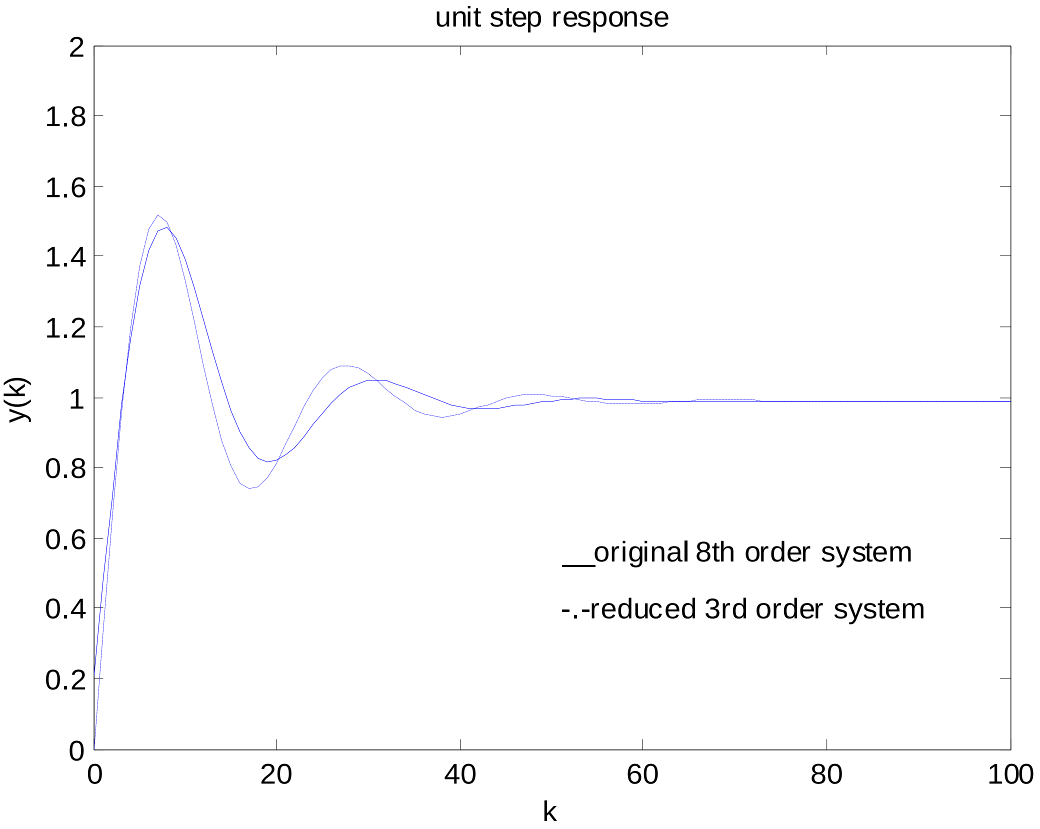

Fig.3.2: Step response of original 8th order system and reduced 3rd order system

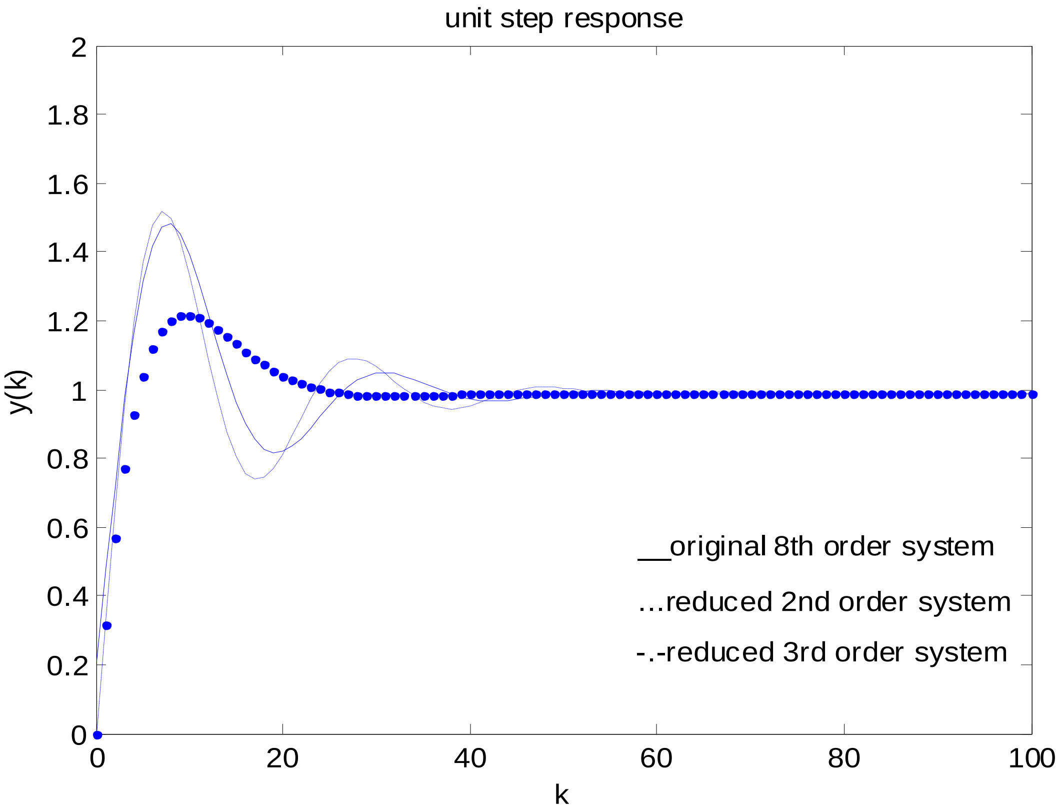

Fig.3.1: Step response of original 8th order system and reduced 2nd and 3rd- order MRSM system

Remarks:

Modified Routh stability method is stability preservation method (SPM) which retains stability in obtained reduced-order model. Modified Routh stability method imparts information categorically that if system whose order is to be reduced is stable, its expansion point m (z=p+m) should be unity.

If original system is oscillatory, then its expansion should be made around the point which lies out side of the unit circle and should be more than unity.

If original system is unstable whose order is to be reduced, it expansion point (z=p+m) should lie out side of unit circle and should be more than furthest pole/zero of original system which lies outside unit circle.

Some of the salient features of the method are:

1. It incorporates the interest of p-domain than bilinear transformation. A P-domain transformation patch up the initial-value problem posed by bilinear transformation and it is quite simpler than bilinear transformation.

2. The overall time and frequency-domain characteristics are closely matching.

3. It offers computational simplicity.

REDUCED ORDER MODELLING OF UNSTABLE SYSTEM BY MODIFIED ROUTH STABILITY METHOD USING P-DOMAIN (PROPOSED METHOD)

4.1 INTRODUCTION:

In order to have reduced order model for unstable system which fins its application for design aspects of controllers extensively, Farsi, M; Warwick, K; and Guilandoust, M. [4], suggested that if system is unstable i.e. It has pole or zero lying outside of unity circle, then expansion should be carried around be point which is furthest pole or zero lying outside of unit circle.

While attempting the work it has been concluded by the author that if expansion is carried around the furthest pole or zero then degree of instability in reduced order model would be less than degree of instability which lies in original system which is undesirable. Thus, to preserve the characteristics of original unstable system, expansion should be carried out around the point which lies out sides of unity circle and should be more than the furthest pole or zero of original system.

The overall modeling procedure involves same steps as described in previous example for stable except (z=p+m) is used where m$\in$ [1, 2].

4.2 NUMERICAL EXAMPLE 4.1:

Considered an unstable 4th order system whose transfer function is given as:

$$G(z)=\frac{z^3-1.55z^2+0.91z-0.323}{z^4-1.2z^3+0.72z^2-1.2z+0.01}$$

STEP 1: Original G (z) has the pole which lies outside of unity circle having further pole

1. Thus according to argument made in introductory part of modified Routh stability method expansion point m (z=p+m) should be more than 1 so

m = 1.1, 1.2, 1.3, 1.4, 1.5, 1.6.

Thus,

$$G(P) = \frac{D(p)}{E(p)}\bigg| z = p + 1.1 = \frac{p^3 + 1.75p^2 + 1.13p + 0.1335}{p^4 + 3.2p^3 + 4.02p^2 + 1.352p - 0.5719}$$

STEP 2: Routh array for numerator polynomials is obtained by using Eq. (3.5)

and Table (3.2), when z=p+1.1:

0.1335	1.75	0
1.13	1	

1.63186

Routh array for denominator polynomial is obtained by using table (3.1) and Eq. (3.4), when z=p+1.1

-0.5719	4.02	1
1.3520	3.2	
5.373609	1	

4.2.1 Reduced Model (2nd –order):

STEP 3: Using Eqs. (3.6), (3.7) and (3.8) for the second order reduced model:

α_1 = -0.10643, α_2 = 0.251599 α_3=1

β_1 = 0.08181 β_2 = 1,

γ = 0.210287

STEP 4: Numerator of reduced order model is based on Eq. (3.10)

B (z) = z-0.91819

Denominator of reduced order model is based on Eq. (3.9):

A (z) = z^2-1.7484z+0.641971

STEP 5: Desired reduced order model will assume the following form after incorporating gain correction factor becomes:

$$R(z) = \frac{0.210287z - 0.193083}{z^2 - 1.7484z + 0.641971}$$

The steps corresponding to other values of m would be same as a step corresponding to m = 1.1

Derived Reduced 2nd –Order Models

$$G(z) = \frac{z^3 - 1.55z^2 + 0.91z - 0.323}{z^4 - 1.2z^3 + 0.72z^2 - 1.2z + 0.01};$$ (Original system)

$$R_2(z) = \frac{0.210287z - 0.193083}{z^2 - 1.7484z + 0.641971};$$ (Reduced 2nd –Order, m=1.1)

$$R_2(z) = \frac{0.26643z - 0.21967}{z^2 - 1.6019z + 0.53262};$$ (Reduced 2nd –Order, m=1.2)

$$R_2(z) = \frac{0.308814z - 0.2395288}{z^2 - 1.46536z + 0.447385};$$ (Reduced 2nd –Order, m=1.3)

$$R_2(z) = \frac{0.341674z - 0.2500507}{z^2 - 1.339523z + 0.379966};$$ (Reduced 2nd –Order, m=1.4)

$$R_2(z) = \frac{0.367784z - 0.2542108}{z^2 - 1.22289z + 0.325833};$$ (Reduced 2nd –Order, m=1.5)

$$R_2(z) = \frac{0.3889821z - 0.2538575}{z^2 - 1.1135924z + 0.2820012};$$ (Reduced 2nd –Order, m=1.6)

4.3 MATLAB ANALYSIS AND PLOTS (UNSTABLE SYSTEM)

4.3.1 Script file(Unstable System)

```
 %Script file
 %Purpose:
%Step response of original 4th order unstable system and reduced 2nd
order MRSM
clc;
clear all;
%for Modified Routh unstable system(z=p+1.1)
n=[1 -1.55 0.91 -0.323];
d=[1 -1.2 0.72 -1.2 0.01];
r=ones(1,101);
k=0:100;
y=filter(n,d,r);
plot(k,0.81*y,'-');
v=[ 0 20 0 10] ;
axis(v);
title('unit step response');
xlabel('k');
ylabel('y(k)');
hold on;
n1=[0 0.210287 -0.193083];
d1=[1 -1.7484 0.641971];
r=ones(1,101);
k=0:100;
y=filter(n1,d1,r);
plot(k,0.81*y,'-.');
v=[ 0 20 0 10] ;
axis(v);
title('unit step response');
xlabel('k');
ylabel('y(k)');
gtext('__original 4th order system');
gtext('-.-2nd order reduced MRSM');
```

4.3.2. Results in command window (Unstable System)

```
z =

   0.9500
   0.3000 + 0.5000i
   0.3000 - 0.5000i
```

```
p =

   1.3317
  -0.0701 + 0.9443i
  -0.0701 - 0.9443i
   0.0084

k =

     1

r =

  Columns 1 through 14

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 15 through 28

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 29 through 42

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 43 through 56

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 57 through 70
```

```
    1    1    1    1    1    1    1    1    1    1    1    1    1    1

  Columns 71 through 84

    1    1    1    1    1    1    1    1    1    1    1    1    1    1

  Columns 85 through 98

    1    1    1    1    1    1    1    1    1    1    1    1    1    1

  Columns 99 through 101

    1    1    1

k =

  Columns 1 through 14

    0    1    2    3    4    5    6    7    8    9   10   11   12   13

  Columns 15 through 28

   14   15   16   17   18   19   20   21   22   23   24   25   26   27

  Columns 29 through 42

   28   29   30   31   32   33   34   35   36   37   38   39   40   41

  Columns 43 through 56

   42   43   44   45   46   47   48   49   50   51   52   53   54   55
```

```
Columns 57 through 70

  56    57    58    59    60    61    62    63    64    65    66    67    68    69

Columns 71 through 84

  70    71    72    73    74    75    76    77    78    79    80    81    82    83

Columns 85 through 98

  84    85    86    87    88    89    90    91    92    93    94    95    96    97

Columns 99 through 101

  98    99   100
```

4.3.3. Plots (Unstable System)

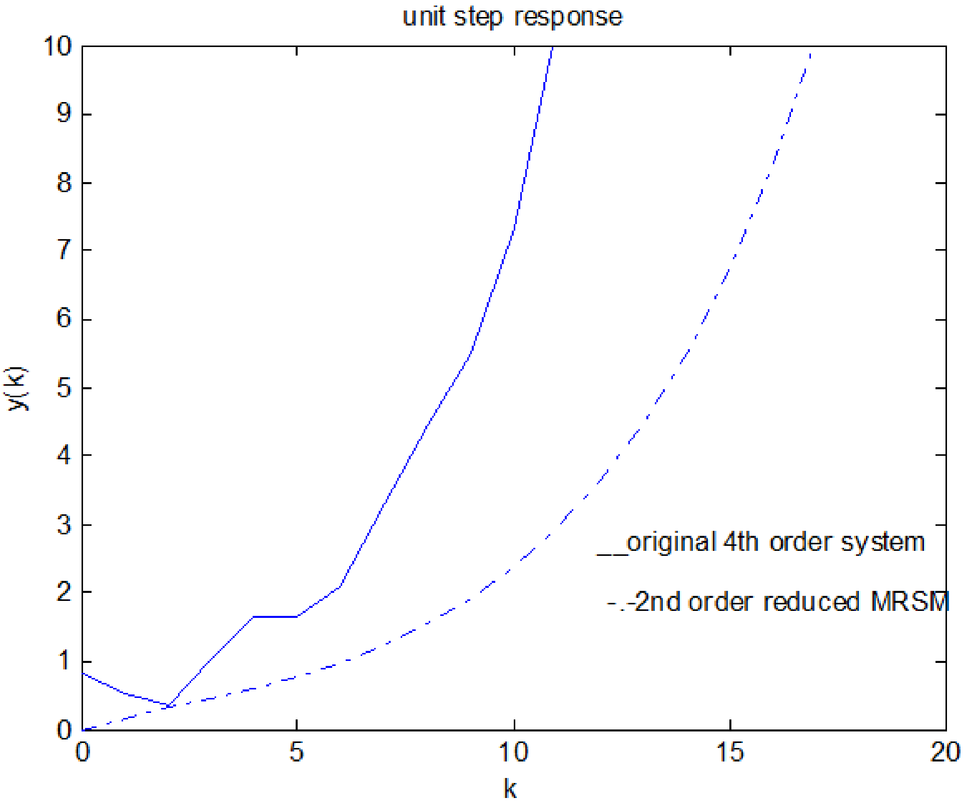

Fig.4.1:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.1)

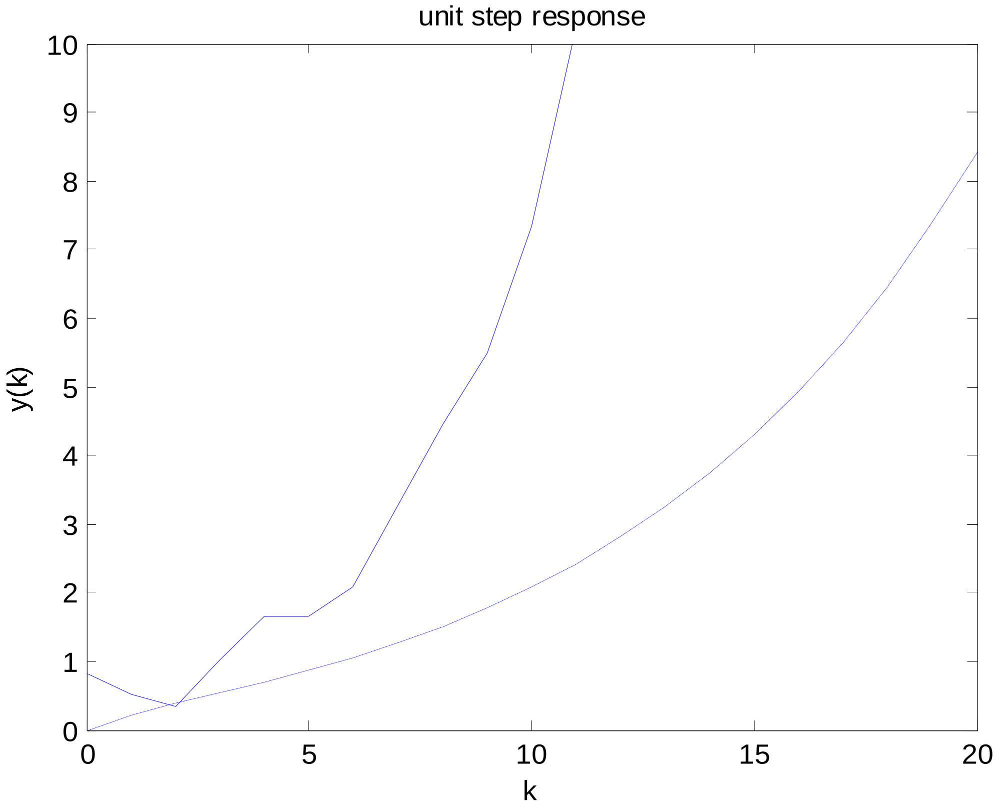

Fig.4.2:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.2)

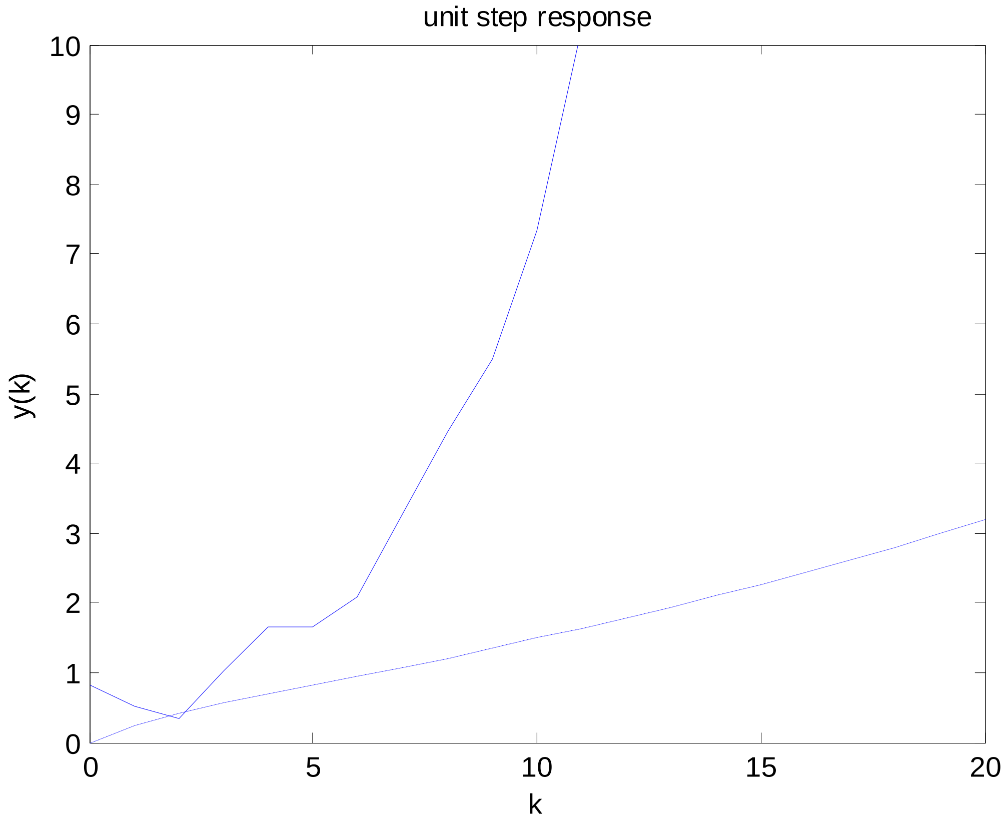

Fig.4.3:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.3)

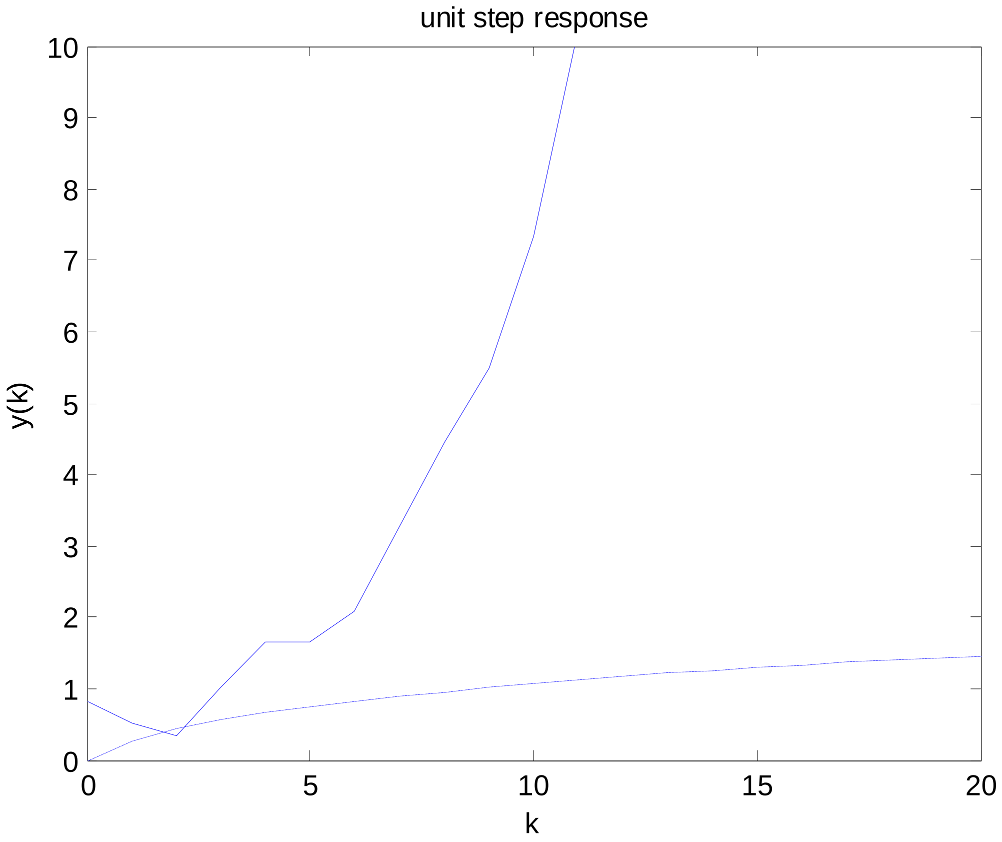

Fig.4.4:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.4)

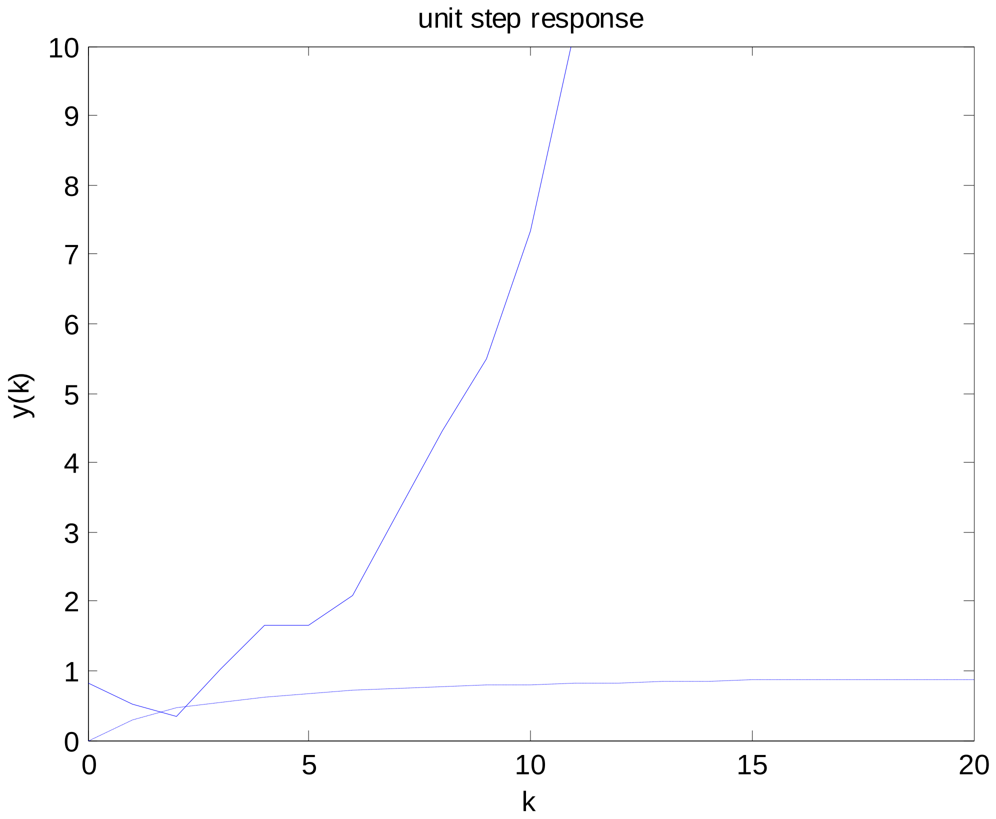

Fig.4.5:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.5)

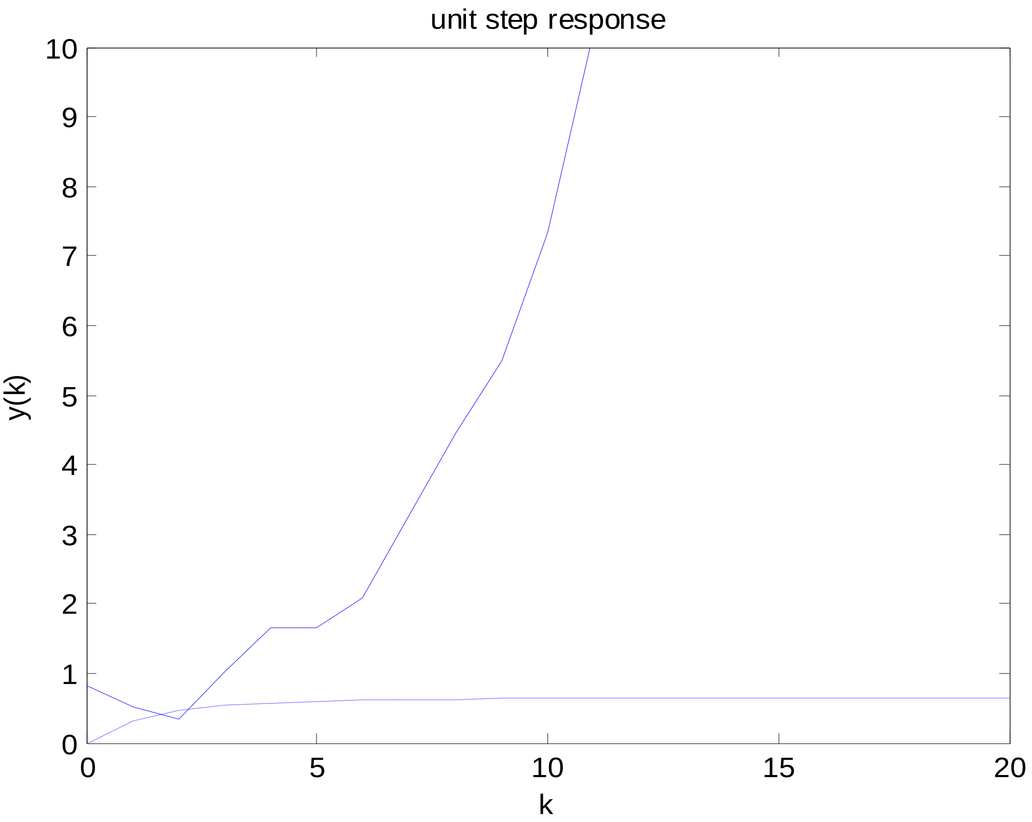

Fig.4.6:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.6)

4.4 NUMERICAL EXAMPLE 4.2:

Considered an unstable 4th order system whose transfer function is given as:

$$G(z) = \frac{z^3 - 2.1z^2 + 1.53z - 0.365}{z^4 - 3.712z^3 + 5.146z^2 - 3.1631z + 0.72624}$$

STEP 1: Original G (z) has the pole which lies outside of unity circle having further pole 1. Thus according to argument made in introductory part of modified Routh stability method expansion point m (z=p+m) should be more than 1 so

m = 1.1, 1.2, 1.3, 1.4, 1.5, 1.6.

Thus,

$$G(P) = \frac{D(p)}{E(p)}\bigg| z = p + 1.1 = \frac{p^3 + 1.2p^2 + 0.54p + 0.1008}{p^4 + 0.688p^3 + 0.1564p^2 + 0.0075p - 0.0031}$$

STEP 2: Routh array for numerator polynomials is obtained by using Eq. (3.5)

and Table (3.2), when z=p+1.1:

	0.1008	1.2	0
0.54	1		
1.01333			

Routh array for denominator polynomial is obtained by using table (3.1) and Eq. (3.4), when z=p+1.1

-0.0031	0.1564	1
0.0075	0.688	
0.44077	1	
0.67098		

4.4.1 Reduced Model (2^{nd} –order):

STEP 3: Using Eqs. (3.6), (3.7) and (3.8) for the second order reduced model:

α_1 = -0.007033, α_2 = 0.01839 α_3=1

β_1 = 0.18666 β_2 = 1,

γ = 1.225128

STEP 4: Numerator of reduced order model is based on Eq. (3.10)

B (z) = z-0.91334

Denominator of reduced order model is based on Eq. (3.9):

A (z) = z^2-2.18161z+1.182738

STEP 5: Desired reduced order model will assume the following form after incorporating gain correction factor becomes:

$$R(z) = \frac{1.225128z - 1.119005}{z^2 - 2.18161z + 1.82738}$$

The steps corresponding to other values of m would be same as a step corresponding to m = 1.1

Derived Reduced 2nd –Order Models

$$G(z)=\frac{z^3-2.1z^2+1.53z-0.365}{z^4-3.712z^3+5.146z^2-3.1631z+0.72624}; \qquad \text{(Original system)}$$

$$R_2(z)=\frac{1.225128z-1.119005}{z^2-2.18161z+1.82738}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.1)}$$

$$R_2(z)=\frac{1.915799z-1.885052}{z^2-0.90048z+0.20947}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.2)}$$

$$R_2(z)=\frac{1.59645z-1.694631}{z^2-2.34149z+1.37046}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.3)}$$

$$R_2(z)=\frac{1.36008z-1.54408}{z^2-2.4487z+1.50376}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.4)}$$

$$R_2(z)=\frac{1.20664z-1.45651}{z^2-2.5619z+1.65047}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.5)}$$

$$R_2(z)=\frac{1.10205z-1.40788}{z^2-2.67778z+1.80979}; \qquad \text{(Reduced }2^{\text{nd}}\text{ –Order, m=1.6)}$$

4.5 MATLAB ANALYSIS AND PLOTS (UNSTABLE SYSTEM)

4.5.1 Script file (Unstable System)

```
 %Script file
 %Purpose:
%Step response of original 4th order unstable system and reduced 2nd order MRSM
clc;
clear all;
%for Modified Routh unstable system(z=p+1.1)
n=[0 1 -2.1 1.53 -0.365];
d=[1 -3.712 5.416 -3.1631 0.72624];
[z p k]=tf2zp(n,d)
r=ones(1,41)
k=0:40
y=filter(n,d,r);
plot(k,y,'-');
v=[ 0 40 -8000 6000] ;
axis(v);
title('unit step response');
xlabel('k');
ylabel('y(k)');
hold on;
n1=[0 1.225128 -1.119025];
d1=[1 -2.18161 1.82738];
g=tf(n,d)
r=ones(1,36);
k=0:35;
y=filter(n1,d1,r);
plot(k,y,'-.');
v=[ 0 40 -8000 6000] ;
axis(v);
title('unit step response');
xlabel('k');
ylabel('y(k)');
gtext('__original 4th order system');
gtext('-.-2nd order reduced MRSM');
```

4.5.2. Results in command window (Unstable System)

```
z =


  0.8000 + 0.3000i
```

```
   0.8000 - 0.3000i

   0.5000

p =

   1.3892 + 0.7783i

   1.3892 - 0.7783i

   0.4668 + 0.2617i

   0.4668 - 0.2617i

k =

     1

r =

  Columns 1 through 14

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 15 through 28

     1     1     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 29 through 41

     1     1     1     1     1     1     1     1     1     1     1     1     1
```

```
k =

  Columns 1 through 14

     0     1     2     3     4     5     6     7     8     9    10    11    12    13

  Columns 15 through 28

    14    15    16    17    18    19    20    21    22    23    24    25    26    27

  Columns 29 through 41

    28    29    30    31    32    33    34    35    36    37    38    39    40
```

4.5.3. Plots(Unstable System)

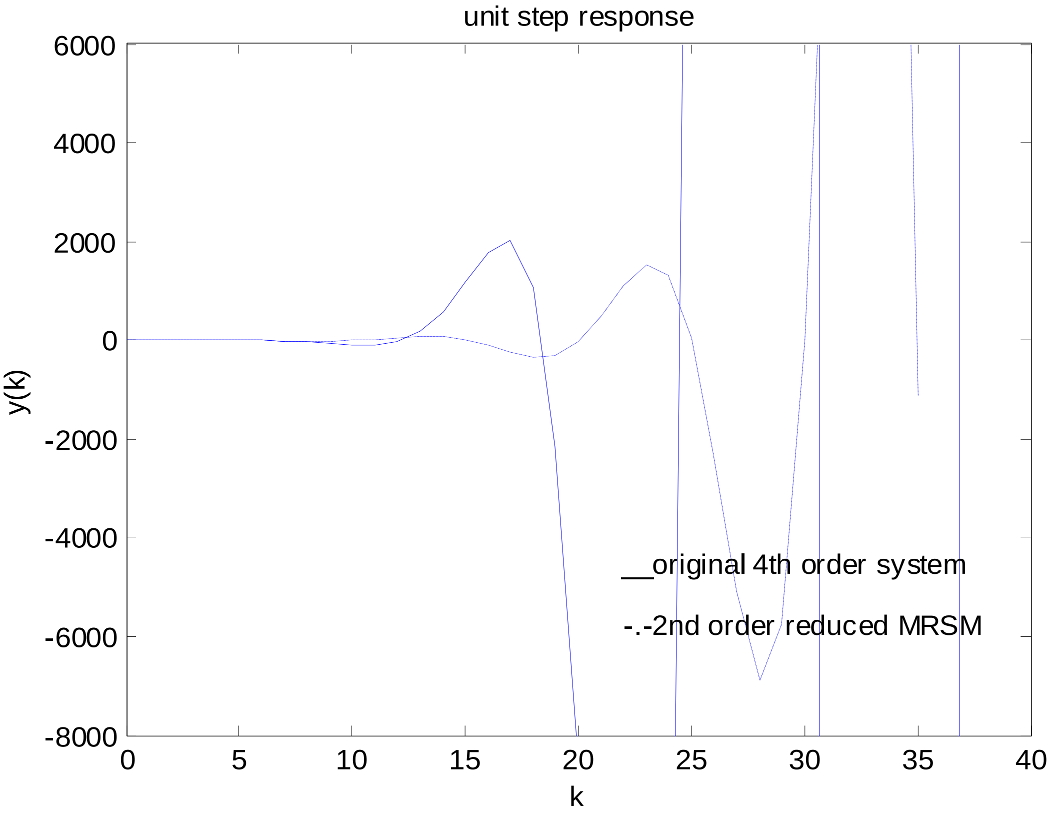

Fig.4.7:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.1)

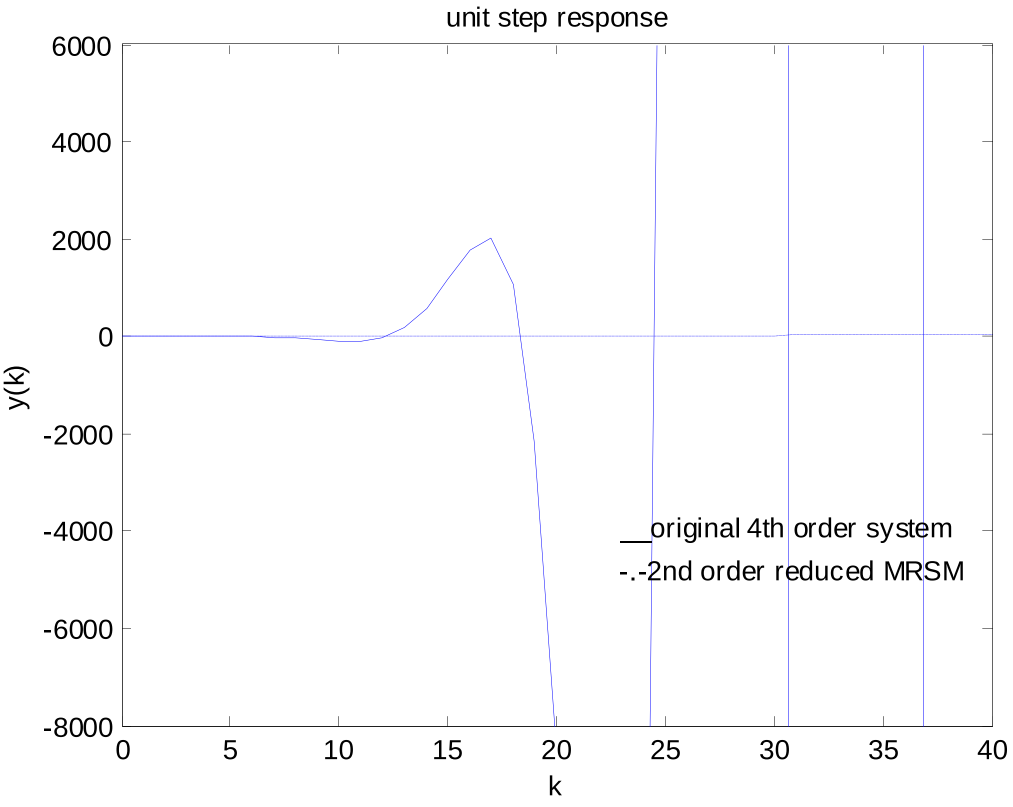

Fig.4.8:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.2)

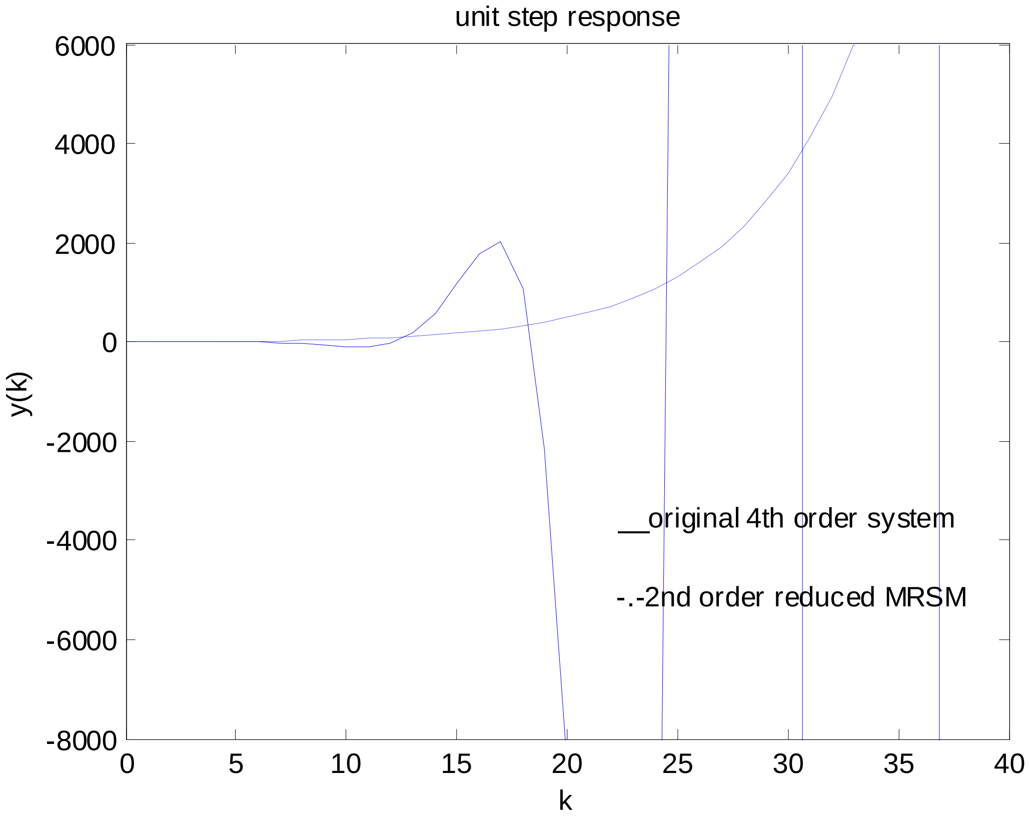

Fig.4.9:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.3)

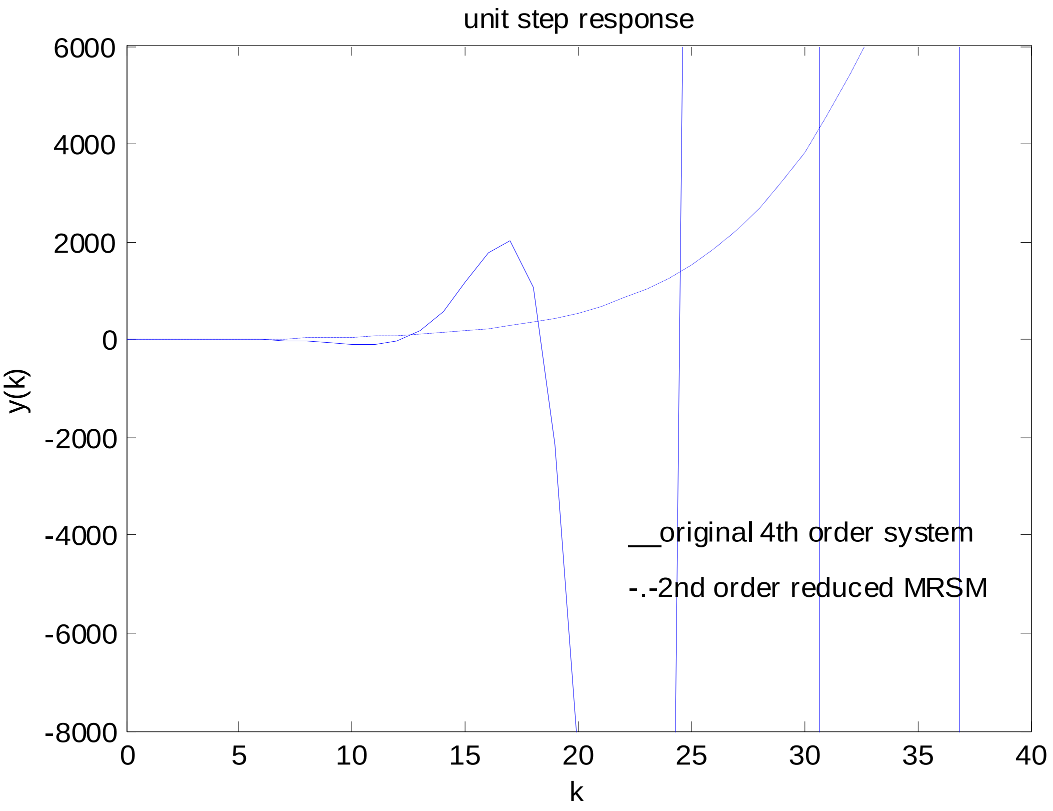

Fig.4.10:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.4)

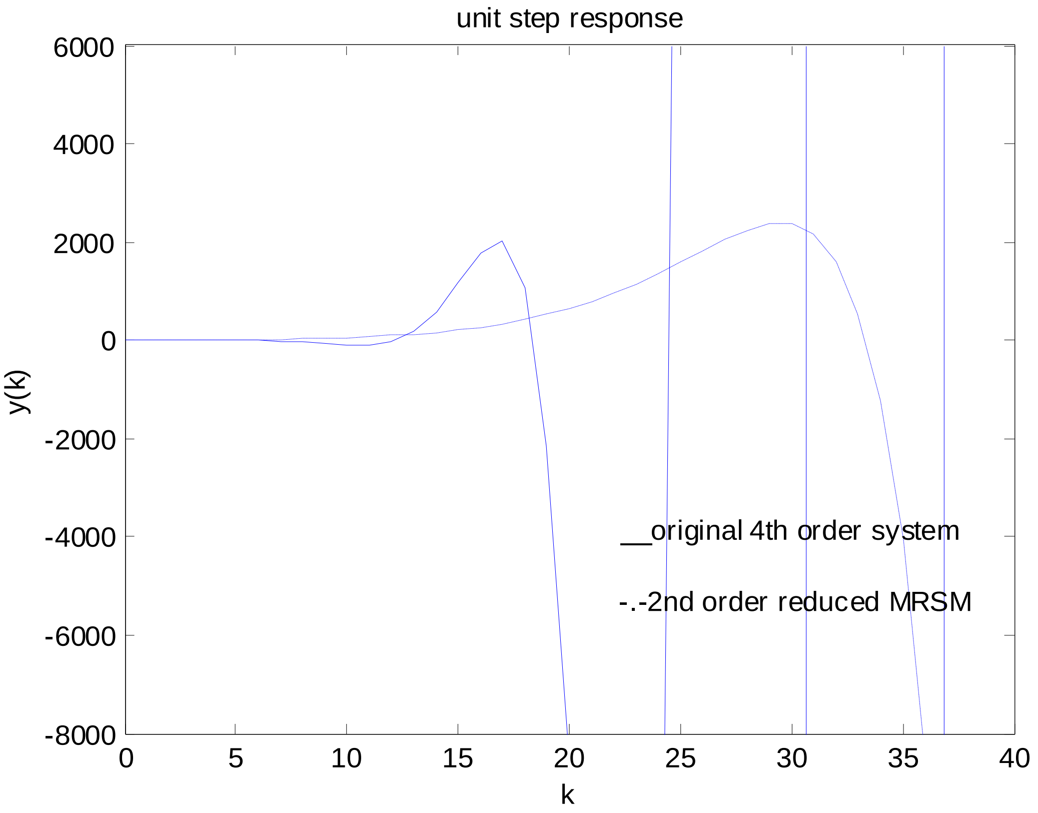

Fig.4.11:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.5)

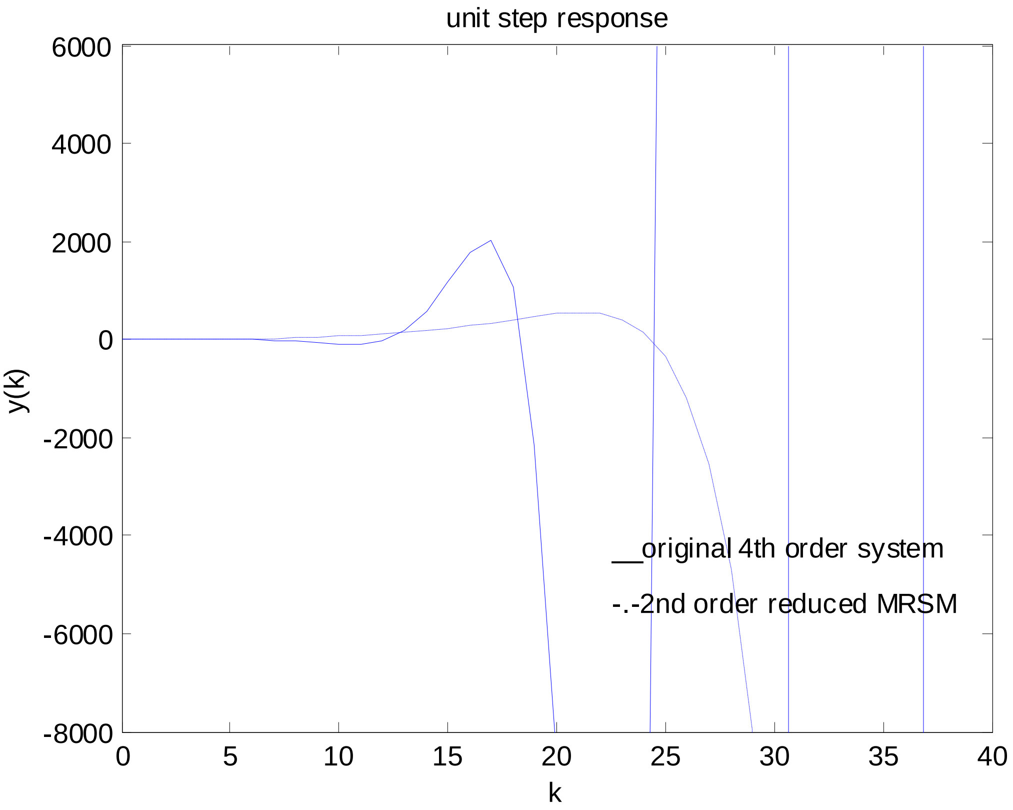

Fig.4.12:Step response of original 4th order system and reduced 2nd-order MRSM(z=p+1.6)

Remarks: In order to reduce the order of unstable system (z=p+m) transformation is adopted where p is greater than 1(one). To preserve the characteristic of original unstable system in reduced-order system author suggest the value of m which should have range [1, 1.5], if m>1.5 then characteristic of reduced unstable system will not match with original system.

CHAPTER 5

CONCLUSION AND SCOPE FOR FURTHER WORK

5.1 CONCLUSION:

The dissertation contains the results of the investigation carried out by the author in the area of reduced order modeling and its applications.

Routh stability method has been discussed which is stability preservation method.

Modified Routh stability method using p-domain transformation for stable, and unstable system has been proposed in this dissertation. Modified Routh stability method which employs a Routh array for reduction of linear time-invariant, discrete-time systems yields stable reduced models, if original is stable. The method is easy to employ and relates simply control engineering problem.

Modified Routh stability method takes the advantages of p-domain, transformation instead of s-domain which poses initial, value problem where p-domain transformation does not. This proposed modified Routh stability method is applicable to order reduction of stable, oscillatory and unstable systems which finds its application for controller design purposes.

5.2 SCOPE FOR FURTHER WORK:

Proposed method utilizes bilinear transformation which poses initial value problem can be resolved using efficient p-domain transformation like $z = Ap/ (A+Bp)$ instead of bilinear transformation. Proposed method retains desired time-domain specification as well as frequency-domain specifications in reduced-order model where practically significant frequencies are preserved implicitly. But explicit incorporation of practically significant frequencies interest can be tired in this method as multifrequency

Routh approximation has. Proposed modified Routh stability method using p-domain transformation can be tired for multivariable systems.

REFERENCES

(1) Al-Saggaf, U.M and Franklin, G.F., Reduced order controller design for discrete-time systems, int. j.Syst. Sci., Vol 22, No.10, PP. 1743 – 1756, 1991

(2) Badreddin, E., and Mansour, H., Model Reduction of discrete-time systems using the Schwarz Canonical form, electron, Lett. Vol. 1, PP. 373-383, 1982.

(3) Chen, T.C., Chang, C.Y., and Han, K.W, Model reduction using stability-equation method and continued fraction method, Int. J. control, Vol 32, P. 81-94, 1980.

(4) Farsi, M., Warwick, K., Guilandoust, M., Stable reduced – order models for discrete-time systems, IEE Proceedings, Vol. 133, Pt. D, NO. 3, PP 137-141, May 1986.

(5) Hwang, C. and Hseih, C.S., order reduction of Discrete time systems via bilinear Routh approximation, Transaction of the ASME journal of dynamic systems, Measurement and control, Vol. 112, pp, 292-297, June 1990.

(6) Hwang, C., Hwang, J.H., and Guo, T-Y, Multifrequency routh approximants for linear systems, IEE Proc. Control Theory Appl.,Vol.142, No.4,P.351-358, July1995

(7) K.Warwick, A new approach to reduced order modeling, IEE Proceedings, Vol 131, Pt.d, No.2 P.74-78, March1984

(8) Zhang, W.D., Sun, Y.X. and Xu, X.M., Robust digital controller design for processes with dead times, IEE Proc. Control Theory Appl.,Vol.145, No.2,P.159-164, March 1998.

(9) T.Kangsanant, Model reduction of discrete time systems via power series transformation, Royal Melbourne institute of technology, Australia, P-247-250

(10) Shanti Mishra, Jayanta Pal, A mixed method for the reduction of discrete time systems, National systems conference, p.580-582, December 2008.

(11) R. Unnikrishnan and A. Gupta, Model reduction for discrete time systems, Electronics letters, Vol. 26, No. 5, P. 282-284, March 1990.

(12) B.Clapperton, F. Crusca and M. Aldeen, Bilinear transformation and generalized singular perturbation model reduction, IEEE proceedings, P-3737-3742, December 1994.

(13) Vivek Kumar sehgal, Reduced order controller design in discrete time domain, T.Sobh et al.(eds.),P-481-486,Springer, 2007

(14) P. Brehonnet, A.Derrien, P. Vilbe and L.C. Calvez, Quasi-optimal model reduction of discrete time systems, Electronic letters, Vol. 32, No. 16, August 1996.

(15) M. Diab, W. Q. Liu, and V. Sreeram, A new approach for frequency weighted L_2 model reduction of discrete time systems, Proceedings of the American control conference Albuquerque, New Mexico, P.3863-3867, June 1997.

(16) Chen-Chien Hsu, Kai-Ming Tse, and Wei-Yen Wang, Discrete time model reduction of sampled systems using an enhanced multiresolutional dynamic genetic algorithm, IEEE proceedings, P.280-285, 2001.

(17) Shih-Lian Cheng and Chyi Hwang, Optimal approximation of linear systems by a differential evolution algorithm, IEEE proceedings, P.698-707, 2001.

(18) G. Kotsalis, A.Megretski, M. A. Dahleh, Model reduction of discrete time

markov jump linear systems, IEEE proceedings, P.454-459, 2006.

(19) J. S. Yadav, N. P. Patidar, J. Singhai and S. Panda, Differential evolution algorithm for model reduction of SISO discrete time systems, IEEE proceedings, P.1053-1058, 2009.