الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جـامـعـــة زيــــان عـاشـــور بـالـجـلـفـــة
Ziane Achour University of Djelfa

كـلـــيـــة الـــعـــلـوم و الـتـكـنـــولـوجـيـا
Faculty of Science and Technology

Department: Electrical Engineering

Order N°: .... / 2022

Defense authorization N° ..../2022

DOCTORAL THESIS

3rd Cycle Doctoral (D-LMD)

Presented by

Raouf ZERROUGUI

With a view to obtaining the doctoral diploma in 3rd Cycle Doctoral (D-LMD)

Branch: Electronics

Specialty: signal image and systems

Topic

INTERVAL SYSTEMS APPROXIMATION

Defended in public, on… /… / 2020, before the jury composed of:

Last and first name	Grade	Institution of affiliation	Designation
Pr OMAR MANSOUR	Professor	UNIVERSITY OF DJELFA	PRESIDENT
Pr AMEL B.H.ADAMOU-MITICHE	Professor	UNIVERSITY OF DJELFA	SUPERVISOR
Pr LAHCÈNE MITICHE	Professor	UNIVERSITY OF DJELFA	EXAMINER
Pr MOHAMED S EGHIR BOUCHERIT	Professor	ENP of 'ALGERS	EXAMINER
Pr MOHAMED TADJINE	Professor	ENP of ALGERS	EXAMINER
Pr MOHAMMED KHAMADJA	Professor	Université of OUM EL BOUAGHI	INVITED

Djelfa University, FST - 2022

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة زيان عاشور بالجلفة
Université Ziane Achour de Djelfa

كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie

Département : Génie Electrique

N° d'Ordre : . …. / 2022

Autorisation de Soutenance N° …………/2022

THESE DE DOCTORAT

Doctorat 3ème Cycle (D-LMD)

Présentée par

Raouf ZERROUGUI

En vue de l'obtention du diplôme de Docteur en 3ème Cycle D-LMD

Filière : Electronique

Spécialité : SIGNAL IMAGE ET SYSTEMES

Thème

APPROXIMATION OF SYSTEMS DEFINED BY INTERVAL

Soutenue publiquement, le 17 /12 /2022, devant le jury composé de :

Nom et Prénom	Grade	Etablissement de rattachement	Désignation
Pr OMAR MANSOUR	Professeur	Université de Djelfa	Président
Pr AMEL B.H.ADAMOU-MITICHE	Professeur	Université de Djelfa	Directeur de thèse
Pr LAHCÈNE MITICHE	Professeur	Université de Djelfa	Examinateur
Pr MOHAMED SEGHIR BOUCHERIT	Professeur	ENP D'ALGER	Examinateur
Pr MOHAMED TADJINE	Professeur	ENP D'ALGER	Examinateur
Pr MOHAMMED KHAMADJA	Professeur	Université de OUM EL BOUAGHI	Invité

Université de Djelfa, FST, 2022

Contents

List of Figures

List of Tables

List of acronyms

- **MOR :** Model Order Reduction.
- **AI:** Artificial Intelligence.
- **ANN :** Artificial Neural Network.
- **GA :** Genetic Algorithm.
- **MIMO :** Multi-Input Multi-Output.
- **SISO :** Single-Input Single-Output.
- **ANN-MOR :** Artificial Neural Network - Model Order Reduction.
- **ANN-P.A :** Artificial Neural Network - Polynomial Approximation.
- **SVD :** Singular Values Decomposition .
- **BLAS :** Basic Linear Algebra Subprograms.
- **ISE :** Integral Square Error.
- **MSE :** Mean Squared Error.
- **SSE :** Steady-State Error.
- **IEEE :** Institute of Electrical and Electronics Engineers.

Abstract

ABSTRACT : This thesis deals with reducing mathematical models of interval systems to obtain a model of reduced dimension. In the first part, we give a general overview of some properties of interval arithmetic; then, we proceed into background and state-of-the-art. Furthermore, we go up to the point of our thesis in chapter II by presenting the developed methods of MOR published in two top journals. The first proposed method is based on a modified Schur approach to reduce interval systems. The second method describes a new artificial intelligence technique based on the artificial neural network to reduce the interval system. This one is used to reduce the degree of the polynomial numerator and denominator individually by permitting them to learn automatically from the initial system. Finally, in chapter IV, a comparison study is presented between current works and the suggested technique with the help of illustrations from literature.

Keywords : Approximation, Model Order Reduction, Interval systems , SVD, Uncertain Systems, Artificial Neural Network

Résumé : Cette thèse traite differentes methodes de réduction de modèles des systèmes définis par intervalles pour obtenir un modèle d'ordre réduit. Dans la première partie, nous donnons un aperçu général sur l'arithmétique d'intervalle, puis nous approfondissons un état de l'art sur le domaine. le chapitre III, contient les méthodes qui nous avons développées et publiées dans deux articles. La première méthode proposée est basée sur l'approche de Schur modifiée pour réduire l'ordre système définis par intervalle. La deuxième méthode décrit une nouvelle technique d'intelligence artificielle basée sur les réseaux de neurones artificiels pour réduire un système défini par intervalles. Celui-ci est utilisé pour réduire le degré polynomial de numérateur et de dénominateur, chacun individuellement, avec l'apprentissage automatique à partir du système initial. dans le chapitre IV Une étude comparative est présentée entre quelque travaux existe dans la littérature et la technique proposée, à l'aide d'illustrations numerique .

Mots clés : Approximation, Réduction d'ordre, Systèmes définis par intervalle, SVD, Systèmes Incertains, Réseau de Neurones Artificiels.

ملخص : تركز هذه الأطروحة على خفض درجة النماذج الرياضية للأنظمة المعرفة بمجالات للحصول على نموذج ذو درجات مخفضة يمكن من خلاله ضمان نفس الأداء مثل الاستقرار . في الجزء الأول ، نعطي لمحة عامة عن بعض خصائص حساب المجالات ، ثم نتعمق في مسألة ماهية تخفيض درجة النموذج الرياضي ، وفي نفس الوقت نقدم بعض الطرق النموذجية لـ MOR في الدراسات السابقة. بالإضافة إلى ذلك ، تقديم الأساليب المطورة لـ MOR ، المنشورة في مجلتين محترمتين. تعتمد الطريقة المقترحة الأولى على نهج Schur المعدل لتقليل الدرجة الأعلى لنظام المعرف بمجالات الزمني الخطي إلى درجة أدنى محافظا على الاستقرار. الطريقة الثانية تصف تقنية ذكاء اصطناعي جديدة تعتمد على الشبكة العصبية الاصطناعية لتقليل نظام المعرف بمجالات. في المطلب الاخير يتم تقديم دراسة مقارنة بين العمل الحالي والتقنية المقترحة باستخدام الرسوم التوضيحية من المراجع.

الكلمات المفتاحية : التقريب , تصغير درجة النموذج , الانظمة المعرفة بمجالات , قيمة التحلل المفرد , الانظمة الغير مؤكدة, الشبكات العصبونية الاصطناعية

Acknowledgments

Almost all of these years have passed in a blink of an eye for my **Ph.D.** Life. Looking back at the impressive and rich experience, i would like to express my sincerest appreciation to everyone who made this possible.

In the first place, i wish to express my gratitude sincerely to my advisors, **Professor Mme Amel B. H. ADAMOU-MITICHE and Professor LAHCENE MITICHE**, for providing me with the extraordinary opportunity to work closely with them for **my Ph.D.** theses. From the outlooks of research subjects to the suitable choice of every word in articles, they do not just care about the tiny detail but, more importantly, consideration and value my interest and opinion. I appreciate it very much that they constantly put the development of their students in the first position under whatever uncertainties. Furthermore, their generous support, unconditional trust, constant patience, and incredible enthusiasm have substantially boosted my academic growth. Last but not least, my appreciation also goes to **Mr.President of the jury and all jury members** who accepted to examine and evaluate my work.

In the second place, my knowledge goes to **my parents** because of you. I came into this globe and have grown up healthy and joyful, Also a considerable knowledge goes **To my dear brothers, sister, and ALL my Family, especially** my daughter **INES**. For providing wholehearted encouragement and support for the successful completion of this work.

To conclude, i take this opportunity to convey my sincere thanks to all the members of the laboratory **(laboratoire modélisation simulation et optimisation des systèmes complexes réels, universite de djelfa**) and **PSG colleagues** of the department of electrical and electronics engineering at **(ziane achour university of djelfa)** and all members of the department of electromechanical & department of electronics at **(university of BBA)**.

– RAOUF ZERROUGUI.

Publications

Portions of the work described in this thesis have also appeared in:

Journals

- R. Zerrougui, A. B. H. Adamou-Mitiche, and L. Mitiche, "**Projection approach for interval systems approximation: An extension to mimo systems,**" IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 2, pp. 499–503, 2022.
- R. Zerrougui, A. B. H. Adamou-Mitiche, and L. Mitiche, "**A novel machine learn- ing algorithm for interval systems approximation based on artificial neural network,**" Journal of Intelligent Manufacturing, vol. feb 5, pp. 1–14, 2022.

Part A

General introduction

A long time ago, the philosophy of science focused on different theories that explained the origin of the universe and queried whether they revealed about the laws of nature. Over time, alternative conceptions of science have emerged, such as the semantic conception of theories, which considers scientific theories as sets of models rather than logical constructs. During the 20th century, understanding these physical systems provided a whole new community of scientists who studied control engineering. In this community, each phenomenon can be associated with a mental image of an object, making it possible to anticipate its behavior and observe the relationships, regularities, and symmetries of these phenomena. This image is written using the mathematical language, equations, transfer functions, algorithms.. etc., However, Modeling errors or component tolerances. This difference is called uncertainty or tolerance. Uncertainty can exist in the parameters, in this case, called structured or parametric, or in the structure of the model, in this case, called unstructured. Uncertainties in parameters may appear in actual system models where the exact value of the parameter is unknown but within an interval. The Systems in this type have uncertainty models called dynamic interval systems. which is a set of parameters mainly represented by intervals of values bounded with two ends [a,b] using the basic concepts of interval arithmetics [1] that were introduced for the first time by Ramon Moore in the 1960s as a mathematical approach for bounding the rounding errors in computation.

The more we fit into specific cases, the more the objective of control engineering becomes more complicated. The system's complexity often affects the mathematical description and makes the simulation difficult. Since then, there has been a strong need to support the understanding of the system and facilitate the modeling process and control by reducing computational costs and memory requirements (in Microprocessors). In addition, we need to create an approximation of a highly dynamic system that reproduces the same properties of the system. In another sense, it is necessary to use model reduction (MOR) [2, 3] to reduce the complexity of analyzing such systems. In short, it is a control system technology that provides an optimized solution with a sufficient low-order representation while preserving as much as possible the behavior and essential properties of the original high-level system [3].

Recently, significant efforts have been made in MOR for linear/discrete dynamical systems [3], and many research papers propose good techniques to solve the MOR problem. Among the best-known methods with good performance, we can compute: the γ - α Routh approximation [4], pade approximation [5], eigen spectrum [6], dominant pole[7], and Time Moments in linear systems and Markov Parameters [3]. Many of these methods develop and apply reduction techniques to the interval systems [2]. For example, the eigenvalues are calculated as the roots of a reduced polynomial. Same techniques compute the MOR through a modified γ - δ table respectively. This idea has become the focus of many researchers in the MOR community. Different studies that use a mixed method have also been developed. For example, in [3], eigen spectrum analysis and factor division algorithm are combined. In Ref [6], the MOR is obtained using the clustering technique and the Pade approximation. Various methods [2] are proposed for the reduction based on the singular value decomposition technique (SVD.) [3]. Other methods involve the Krylov subspace method [2] to find the lower order model for complex systems.

Nowadays, model order reduction approaches based on projections have become the most prominent tool and are widely used in control engineering because they are favored to answer the limitations of more conventional methods. It gained significant interest in the modern control area. Firstly it was initiated by Moore [8] and is performed by Safonov and Chiang [9]. The applicability of the idea has been seen either in a model reduction for standard systems, like controllers, the problem of feedback control, or in speech processing [10], in one-dimensional and bi-dimensional digital filters [11]. Furthermore, it has been carried to singular systems MOR [12]. In this thesis, an extension of the vital tool of the Singular Value Decomposition (SVD) technique [12, 9] is proposed for dynamic interval systems model reduction. This method occurs in state space, and the reduced model is built by discarding the state of most negligible energy that contributes weakly to the original system's overall behavior. The reduced model is generated according to the remaining Hankel singular values [13]. It is compared to other reduced-order models cited in the literature to appreciate its performance. Stability is verified using Kharitonov's theorem [14].

On the other hand, plenty of methods have been developed in the international literature, which contain machine learning algorithms [2] such as fuzzy logic [15, 16, 7], genetic algorithm [12]..., etc. Such modern methods use nature-inspired algorithms to achieve more promising results than conventional methods. Among the best and latest techniques, the artificial neural network (ANN) remains recent. It was already discussed and applied only to discrete systems [17, 18, 19]. This biological-inspired algorithm consists of many units called neurons, connected and arranged in layers that stimulate neurons in the biological processes of the brain [20, 21, 22]. Neural networks are algorithms based on the human brain structure system. which apply sets of mathematical equations used to simulate biological brain processes such as learning, decision making, and memory. For a long time, neural networks have been used to model natural systems for decision making (artificial intelligence) by providing automated and intelligent extraction of knowledge and high precision of inference. Among many advantages offered by neural networks, we can cite the most important, such as the need for less formal statistical training, the ability to learn and link between inputs and outputs in all possible ways until we obtain a highly accurate mathematical approximation. The critical motive for using these neurons is the ability to adapt to any system to be studied [2].

For that reason, we investigate a new idea of an advanced machine learning algorithm for reducing a large-scale interval system based on ANN. This idea benefits interval systems, fixed-parameter systems, and simple polynomials. The numerator and the denominator are reduced using a new ANN polynomial approximation method (ANN -PA) by using two neurons arranged in one hidden layer and the pure-line activation function. The pinpoint of this technique is the input vector formatting way, in which each component of this vector corresponds to the main entry. Our proposed algorithm is valid in generalized interval systems in their different forms: continuous-time and discrete-time. Furthermore, it is extended to Multi-input/multi-output interval and fixed-point interval systems. Therefore, it ensures Stability and better performance matching than the already existing methods. We will show from various applications that the results are always superior to what has been achieved.

This thesis is organized as follows: start with a general introduction, then chapter I contain an overview of some properties of interval arithmetic, and chapter II presents some basic concept about SVD and an introduction to ANN. In the same chapter, we offer some typical methods of MOR in the literature. In the second part, in chapter III, we go directly to the contribution of this thesis by presenting the developed methods of MOR that have been published. The first is a Projection procedure for interval dynamic systems approximation, and the second method is A novel machine learning algorithm for the approximation of interval systems founded on an artificial neural network (ANN). Then, this thesis provides a simulation and interpretation in chapter IV. Finally, a general conclusion is given at the end.

Part B

Background and State-of-the-Art

Chapter I

Interval arithmetics and software

I.1 Introduction

Some numerical uncertainty arises in actual calculations on natural phenomena due to slight differences between measurements and expert estimates values. Scientifically speaking, the objective measures and the expected values can't be equal, which creates a range of values. In other words, we call it an interval of values denoted by [a, b] is the set of real numbers given by $[a,b] = \{x \in R : a \leq x \leq b\}$ [23].

The notion of interval analysis is to calculate with intervals in place of real numbers. While floating-point arithmetic is involved rounding errors and can make inaccurate outcomes, interval arithmetic has the benefit of delivering rigorous bounds for the precise solution. An application is when some parameters are unknown but lie within a specific interval. The algorithm uses interval arithmetic with uncertain pit has appeared several times under different names in history parameters as intervals to produce an interval that bounds all possible results[24].In this chapter, we define the basic concepts of arithmetic and present the basic operations between intervals. The critical point of these definitions is that calculating with intervals is calculating with sets.

I.2 Interval history

Calculating intervals is not an entire discovery throughout the history of science and mathematics. It has appeared under different names throughout the history of science. For example, Archiemades in the past has determined the π value by lower and upper bounds $223/71<\pi<22/7$ in the $3rd$ century BC. Nevertheless, actual calculation with intervals has neither been as famous as other numerical techniques nor completely forgotten in basic calculations on natural phenomena [1].

The birth of modern interval arithmetic marks by the appearance of the book Interval Analysis by Ramon E. Moore.[25] He had the idea in the spring of 1958 and later in 1959. Its advantages were that starting from a simple principle provides a general method for automated error analysis, not just errors due to approximation. In 1956, an independent work suggested by Mieczyslaw Warmus formulae for computing with intervals,[26] though Moore found the first fundamental applications. Over the following years, Karl Nickel [27] studied more useful performances, while improved containment practices about the solution of systems of equations were due to Neumaier Arnold in other works. In 1960s, Eldon Robert Hansen [28] developed an interval extension for a linear equation, and after that, they provided a crucial contribution to global approximation, including their theorem, what is now known as Hansen's method, perhaps the most widely used interval algorithm. [1] Classical methods often have the problem of determining the largest (or smallest) global value. However, they could only find a local optimum but not better. Jon George Rokne and Helmut Ratschek extended branch and bound techniques, which until then had only used integer values by operating intervals to provide applications for continuous values.

I.3 Basic terms and concepts

Recall that the following interval represented by $[\underline{a},\overline{a}]$ is the set of real numbers given by :

$$[\underline{a},\overline{a}] = \{a\in R : \underline{a}\leq x\leq\overline{a}\} \tag{I.1}$$

Although other kinds of intervals (open, half-open) appear throughout mathematics, our work will focus mainly on closed intervals. In this work, the term interval will indicate a closed interval. [1]

I.3.1 Interval arithmetic notation

The brackets “[·]” are used for intervals defined by an upper and a lower bound. The underscores ($\underline{\cdot}$) mean inferior bounds of intervals, and the overscores ($\overline{\cdot}$) represent the superior bounds. For intervals determined by a midpoint and a radius, the brackets “$< \cdot >$” will be used. [1]

I.3.2 Real interval arithmetic

A real interval a is a non-empty set of values :

$$a = \{\underline{a}, \overline{a}\} = \{a \in R : \underline{a} \leq a \leq \overline{a}, \} \tag{I.2}$$

Where a is called the infimum and a is called the supremum. The set of all intervals over IR is denoted by IR where :

$$IR = \{[\underline{a}, \overline{a}] : \underline{a}, \overline{a} \in R, \underline{a} \leq \overline{a}\} \tag{I.3}$$

The midpoint of a :

$$mid(a) = \check{a} = \frac{1}{2}(\underline{a} + \overline{a}) \tag{I.4}$$

And the radius of a :

$$rad(a) = \frac{1}{2}(\overline{a} - \underline{a}) \tag{I.5}$$

Two intervals, A and B, are equal if they are the same sets. Operationally, this happens if their corresponding endpoints are equal: $A = B$ if $\underline{A} = \underline{B}$ and $\overline{A} = \overline{B}$.[1]

I.3.3 Interval operators

The operation on two intervals, such as addition (+) or multiplication (x), such as :

$$[a_{(1)}, a_{(2)}] \times [b_{(1)}, b_{(2)}] = \{a \times b \,|\, a \in [a_{(1)}, a_{(2)}] \wedge b \in [b_{(1)}, b_{(2)}]\}. \tag{I.6}$$

All values of $a \times b$, suppose $\star$ is monotonic in every operand over intervals, which is the possibility for all four basic arithmetic operations (except division when the denominator contains 0). In that case, extreme values occur at the ends of the ranges of operands. Write all the combinations. One way to put it is: $[a_1,a_2]\star[b_1,b_2]=$

$$[\min\{a_1\star b_1, a_1\star b_2, a_2\star b_1, a_2\star b_2\}, \max\{a_1\star b_1, a_1\star b_2, a_2\star b_1, a_2\star b_2\}], \quad \text{(I.7)}$$

Provided that $a\star b$ is defined for all $a\in[a_1,a_2]$ and $y\in[b_1,b_2]$.

For useful applications, this can be facilitated further:

Table I.1: Interval operators

Addition:	$[a_1,a_2]+[b_1,b_2]=[a_1+b_1,a_2+b_2]$.
Subtraction:	$[a_1,a_2]-[b_1,b_2]=[a_1-b_2,a_2-b_1]$.
Multiplication:	$[a_1,a_2]\cdot[b_1,b_2]=[\min\{a_1b_1,a_1b_2,a_2b_1,a_2b_2\}$, $\max\{a_1b_1,a_1b_2,a_2b_1,a_2b_2\}]$.
Division:	$[a_1,a_2]/[b_1,b_2]=[\min\{a_1/b_1,a_1/b_2,a_2/b_1,a_2/b_2\}$, $\max\{a_1/b_1,a_1/b_2,a_2/b_1,a_2/b_2\}]$.

The last case loses useful information about excluding $(1/b_1,1/b_2)$. Thus, it is common to work with $\left[-\infty,\left\{\frac{1}{b_1}\right\}\right]$ and $\left[\left\{\frac{1}{b_2}\right\},\infty\right]$ as separate intervals. More generally, it is sometimes useful to calculate with so-called multi-intervals of the form $\bigcup_i [a_i,b_i]$. when working with discontinuous functions. The corresponding multi-interval arithmetic maintains a set of (usually disjoint) intervals and provides overlapping intervals to unite.

Multiplication of positive intervals often requires only two multiplications. If a_1,b_1 are nonnegative :

$$[a_1,a_2]\cdot[b_1,b_2]=[a_1\cdot b_1,a_2\cdot b_2], \qquad \text{if}, \qquad a_1,b_1\geq 0. \quad \text{(I.8)}$$

The multiplication can be interpreted as the area of a rectangle with varying edges. The result interval covers all possible areas, from the smallest to the largest.

With the help of these definitions, it is already possible to calculate the range of simple functions, such as $f(\alpha,\beta,x)=\alpha\cdot x+\beta$. For example, if $\alpha=[1,2],\beta=[5,7]$ and $x=[2,3]$:

$$f(\alpha,\beta,x))=([1,2]\cdot[2,3])+[5,7]=[1\cdot 2,2\cdot 3]+[5,7]=[7,13]. \quad \text{(I.9)}$$

The addition and subtraction of infinite or semi-infinite intervals are then defined by the following:

$$[\underline{a},\bar{a}]+[-\infty,\overline{b}]=[-\infty,\bar{a}+\overline{b}], \quad \text{(I.10)}$$

$$[\underline{a},\bar{a}]+[\underline{b},\infty]=[\underline{a}+\underline{b},\infty] \quad \text{(I.11)}$$

$$[\underline{a},\bar{a}]+[-\infty,\infty]=[-\infty,\infty] \quad \text{(I.12)}$$

$$[\underline{a},\bar{a}]-[-\infty,\infty]=[-\infty,\infty] \quad \text{(I.13)}$$

$$[\underline{a},\bar{a}]-[-\infty,\overline{b}]=[\underline{x}-\overline{b},\infty] \quad \text{(I.14)}$$

$$[\underline{a},\bar{a}]-[\underline{b},\infty]=[-\infty,\bar{a}-\underline{b}] \quad \text{(I.15)}$$

For further rules for extended interval arithmetic see. [29]

I.3.4 Intersection, union, and interval hull

The intersection of these two intervals a and b is empty if either $\overline{b}<\underline{a}$ or $\overline{a}<\underline{b}$. In this case, we let $\varnothing$ denote the empty set and write : $a\bigcap b=\varnothing$, Indicating that a and b have no points in common. Otherwise, we may define the intersection $a\bigcap b$ as the interval : $a\bigcap b = max\{a,b\},min\{a,b\}$.

In this latter case, the union of a and b is also an interval : $a\bigcup b = min\{a,b\},max\{a,b\}$.

In general, the union of two intervals, defined by : $a\bigcup b = min\{a,b\},max\{a,b\}$.

It is always an interval and can be used in interval computations. We have $a\bigcup b \subseteq a\underline{\bigcup}b$ For any two intervals a and b. [29]

I.3.5 Interval vectors and matrices

An interval vector is defined as a vector with interval components, and IR^n denotes the space of all n-dimensional interval vectors. Besides, a matrix interval is a matrix with interval elements, and the space of all $m\times n$ matrix is represented by IR^{mxn}. A point matrix or point vector has elements all with zero radius; on the other hand, a matrix or vector is called thick when. Arithmetic functions are the same operations that vectors and matrices carry out according to real operations. The comparison of interval vectors $a, b \in IR^n$ is componentwise. For example,$a > b$ means that $a_i > b_i$ for all i. The infinity norm of a vector $b \in IR^n$ is given by :

$$\left|\left|b\right|\right|_{\infty} = max\{|b_i| : i = 1, ..., n\} \tag{I.16}$$

An interval matrix A is called an M-matrix if and only if $A_{ij} \leq 0$ for all $i \neq j$ and $Au > 0$ for some positive vector $u \in IR^n$. If the comparison matrix A, where :

$$A_{ii} = min\{|\alpha| : \alpha \in A_{ik}\} \tag{I.17}$$

$$A_{ii} = -max\{|\alpha| : \alpha \in A_{ik}\} \tag{I.18}$$

Is an M-matrix then A is said to be an H-matrix. [1]

I.4 Kharitonov theorem

Consider the linear, interval system

$$H(s) = \frac{[a_0^-, a_0^+]s^0 + [a_1^-, a_1^+]s^1 + .. + [a_m^-, a_m^+]s^m}{s^n + [c_{n-1}^-, c_{n-1}^+]s^{n-1} ... + [c_1^-, c_1^+]s^1 + [c_0^-, c_0^+]} \tag{I.1}$$

Where, the interval polynomial family :

$$P(s) = \sum_{i=0}^{n} [c_i^-, c_i^+]s^i \tag{I.2}$$

With invariant degree the four associated Kharitonov polynomials [14] are considered as:

$$k_1(s) = c_0^+ + c_1^+ s^1 + c_2^- s^2 + c_3^- s^3 + c_4^+ s^4 + c_5^+ s^5 + \cdots. \tag{I.3}$$

$$k_2(s) = c_0^- + c_1^+ s^1 + c_2^+ s^2 + c_3^- s^3 + c_4^- s^4 + c_5^+ s^5 + \cdots.. \tag{I.4}$$

$$k_3(s) = c_0^- + c_1^- s^1 + c_2^+ s^2 + c_3^+ s^3 + c_4^- s^4 + c_5^- s^5 + \cdots. \tag{I.5}$$

$$k_4(s) = c_0^+ + c_1^- s^1 + c_2^- s^2 + c_3^+ s^3 + c_4^+ s^4 + c_5^- s^5 + \cdots, \tag{I.6}$$

The principles point of Kharitonov's theory is that rather than experimenting with an infinite number of polynomials for stability, it requires only trying four polynomials (I.3 to I.6) (utilizing Routh-Hurwitz and/or any different technique of stability test[30]). Hence, the interval dynamic system is robustly stable if exclusively the four (presented above) associated Kharitonov polynomials of the interval system denominator are stable [14].

Those polynomial are constructed from two different even and odd parts as :

$$k_1(s) = k_{min}^{even} + k_{min}^{odd} \tag{I.7}$$

$$k_2(s) = k_{min}^{even} + k_{max}^{odd} \tag{I.8}$$

$$k_3(s) = k_{max}^{even} + k_{min}^{odd} \tag{I.9}$$

$$k_4(s) = k_{max}^{even} + k_{max}^{odd} \tag{I.10}$$

where,

$$k_{min}^{even}(s) = c_0^- + c_2^+ s^2 + c_4^- s^4 + c_6^+ s^6 + \cdots \tag{I.11}$$

$$k_{max}^{even}(s) = c_0^+ + c_2^- s^2 + c_4^+ s^4 + c_6^- s^6 + \cdots \tag{I.12}$$

$$k_{min}^{odd}(s) = c_1^- s^1 + c_3^+ s^3 + c_5^- s^5 + c_7^+ s^7 + \cdots \tag{I.13}$$

$$k_{max}^{odd}(s) = c_1^+ s^1 + c_3^- s^3 + c_5^+ s^5 + c_7^- s^7 + \cdots \tag{I.14}$$

Theorem : (Robust stability of interval systems). An interval polynomial family $P(s) = \sum_{i=0}^{n} [c_i^-, c_i^+] s^i$, with the invariant degree, is robustly stable if and only if its four kharitonov polynomials are stable

---

The stability of the whole full/reduced systems in the simulation chapter (IV) is confirmed by using the procedure of Kharitonov to denominator polynomial of the interval transfer functions.

I.5 Software for interval arithmetic (INTLAB)

INTLAB (INTerval LABoratory) is an intval arithmetic library using MATLAB and GNU Octave, available on Windows, Linux, and macOS. SM Rump was developed at the Technical University of Hamburg. INTLAB has been utilized to develop additional MATLAB-based libraries, such as VERSOFT [31], and has got solved many hundred-digits challenge issues. [32] INTLAB can be used wherever MATLAB and IEEE 754 algorithms are available. Siegfried Rump's INTLAB Toolbox is freely available from [33], and contains information on installation and use. Version 4 is used here. All routines in INTLAB are M-files, except for one routine, set round, which changes the rounding mode, allowing portability and high speed on different systems.

I.5.1 Interval input and output

Real intervals in INTLAB are stored by the infimum and supremum, whereas complex intervals are stored by the midpoint and radius. However, this is not seen by the user. Intervals may be entered using either representation. For example, the intval a = [1, 1] is entered using infimum and supremum as

```
>>x=infsup(-1,1);
```

but the same interval could be entered using the midpoint and radius as

$>> x = midrad(0,1);$

Since complex intervals are stored as a circular region using the midpoint and radius it is more accurate to input such intervals in this way. For example, the circular region with midpoint at $1+i$ and radius 1 is entered using

$>> y = midrad(1+i,1);$

If a rectangular region is entered using the infimum and supremum then the region is stored with an overestimation as the smallest circular region enclosing it. The infimum is entered as the bottom left point of the region and the supremum, is the top right point. The region with an infimum of 1 + i and a supremum of 2 + 2i is entered as

$>> z = infsup(1+i, 2+2i);$

However, it is stored by the midpoint and radius notation as

$>> midrad(z)$
intval $z =< 1.50000000000000 + 1.50000000000000i, 0.70710678118655 >$

Interval vectors and matrices are entered in a similar way, with the arguments being vectors or matrices of the required size. The function intval provides another way to enter an interval variable or can be used to change a variable to an interval type. This function gives an interval with verified bounds of a real or complex value and is a vital part of verification algorithms. However, care has to be taken when using it. It is widely known that the value 0.1 cannot be expressed in binary floating point arithmetic so intval may be used to give a rigorous bound. Using

$>> x = intval(0.1);$

the variable a will not necessarily contain an interval including 0.1, since 0.1 is converted to binary format before being passed to intval. The result is a thin interval with

$>> rad(x)$
$ans = 0$

Rigorous bounds can be obtained using intval with a string argument, such as

$>> x = intval('0.1')$
intval x=[0.09999999999999,0.10000000000001]

which uses an INTLAB verified conversion to binary. It can be seen that a contains 0.1 since the radius is nonzero.

$>> rad(x)$
$ans = 1.387778780781446e-017$

Using a string argument with more than one value will produce an interval column vector with components that are intervals that enclose the values entered.The structure may be altered to a matrix using reshape or a column vector using the transpose command (’). **[32]**

I.5.2 Matrix and vectors

INTLAB enables basic operations to be performed on real and complex interval scalars, vectors, and matrices. These operations are entered similar to real and complex arithmetic in MATLAB. For example, if the matrix A is entered, then A^2 performs $A*A$ in interval arithmetic, whereas $A.^2$ outcomes in each element of A being squared by operating interval arithmetic. Standard functions such as trigonometric and exponential functions are available and used in the usual MATLAB way.

There are four ways to enter a real interval matrix. The first is the function intval which allows the input of a floating point matrix. The following creates a 2×2 matrix of point intervals.

```
>> A = intval([0.1,2;3,4])
intval A =
0.1000 2.0000
3.0000 4.0000
```

The matrix will not necessarily contain an interval enclosing 0.1, since 0.1 is converted to binary format before intval is called. Using a string argument overcomes this problem. This example creates a vector with interval components that encloses the values entered.

The first component now encloses 0.1. This can be seen since the radius is nonzero.

```
>> A = intval('0.1 2 3 4')
intval A =
0.1000
2.0000
3.0000
4.0000
>> rad(A(1,1))
ans =
1.3878e-017
```

Notice that A is a column vector. All output using a string argument produces a vector of this form. It may be changed to a 2×2 matrix by:

```
>> A = reshape(A,2,2)
intval A =
0.1000 3.0000
2.0000 4.0000
```

The third alternative is to give an interval by its midpoint and radius.

```
>> B = midrad([1,0;0.5,3],1e-4)
intval B =
1.000_ 0.000_
0.500_ 3.0000
```

Finally an interval may be input by its infimum and supremum.

```
>> C = infsup([-1,0;2,4],[-0.9,0;2.4,4.01])
intval C =
-0.9___ 0.0000
2.____ 4.00__
```

I.5.3 Additional INTLAB functions

```
For a, b of type intval:
intersect (a,b) >> is the inter-section of a, b
hull (a,b) >> is the union of a, b
abss (a) >> is the absolute value of a
mig (a) >> is the mignitude of a
in (a,b) >> is included a in b (logical array)
in0 (a,b) >> is included the value a in the interior of b (logical array)[34]
```

I.5.4 Other interval arithmetic software

SLAB : Shin'ichi Oishi develops **SLAB** , a stand-alone MATLAB platform only for interval calculations and the automatic result validation it delivers. It is open for free (under the GNU license) and has been built for MS Windows, Linux, and Mac OS. A short illustration appears in [35].

C-XSC : was created by a group of students associated with Ulrich Kulisch at the University of Karlsruhe and is presently supported by members of that group at the University of Wuppertal. [36]

PROFIL/BIAS and FILIB++ : the widely used class libraries in C++ for interval arithmetic are PROFIL/BIAS and FILIB++. [37]

MPFI: Under IEEE 754 standard for binary floating-point arithmetic. Nathalie Revol et al. have created the multiple precision interval libraries MPFI. [38]

The *INTLAB toolbox* is also available in pylab-python [39] and GNU Octave.

I.6 Conclusion

Interval arithmetic is an excellent approach to meet the requirement for the reliability of calculations. Indeed, it is based on the principle that any analysis returns a guaranteed framework of its result. In addition, it could be implemented efficiently on a computer, and a variety of numerical algorithms have been designed specifically to take advantage of this arithmetic. However, it would be naive to believe that the algorithms used in floating-point arithmetic give convincing results on intervals. Most often, the frames obtained are too pessimistic. On the other hand, algorithms developed for interval arithmetic, most often iterative algorithms are based on a contracting iteration, provide information and results inaccessible by floating: let us recall the problem of the global optimization of a continuous function for which floating-point algorithms return a local optimum. We can also mention the difficulty of solving nonlinear systems in this chapter. Using interval arithmetic, we can not only determine all the solutions but also prove their existence or uniqueness, using the fixed point theorems that become effective with such arithmetic.

Every coin has its flip side; that of interval arithmetic is to turn over sometimes too broad frames of the results. We believe very much in the possibilities offered by the combination of interval arithmetic and multi-precision arithmetic, which combines reliability and precision. Indeed, once we have a contracting iteration, only the accuracy of the calculation (ignoring the exponential complexity) constitutes an obstacle to obtaining results as precise as we want. Our first experiments show that the additional cost due to multiple precision is compensated by saving some unnecessary calculations.

Chapter II

Basic concepts and literature review on Model Order Reduction (MOR)

II.1 Introduction

Model order reduction is an approach consisting of building models with a reduced number of state variables compared to the original model but accounting for the complexity of the system's behavior with good precision. Before the rise of microsystems, the development of model order reduction methods for the simulation of integrated circuits for the macro models used in simulation software. Model order reduction methods are diverse, and their study covers theoretical and practical aspects.

However, electrostatically actuated Micro-electromechanical systems components, and more particularly microswitches, are the components that have most given rise to the development of reduced-order models [15, 40, 41]. the point of interest in model order reduction methods has been motivated by large order dynamical systems linked to the emergence of computer calculation and finite element type numerical resolution methods. However, such methods for high-order non-linear systems have been more particularly developed in the context of fluid mechanics. One of the first contributions concerning this field is [42, 43, 44]. In another sense, it is necessary to reduce the complexity using model order reduction (MOR) to analyze such systems. This chapter is, in brief, a control system technique that gives an optimized solution by a sufficient representation with a lower order that preserves the behavior and essential characteristics of the original high system as closely as possible [3].

Recently, much effort in MOR has been made for linear/discrete dynamical systems, where many researchers propose good techniques to solve the MOR problem. Among the most famous methods with good performance, we can cite the Routh approximation [45] and eigen spectrum [6]. Many of these methods develop and apply reduction techniques to the interval systems. For example, the eigenvalues are calculated as the roots of a reduced polynomial. The same techniques compute the MOR through modified γ and δ tables, respectively. This idea has become the focus of many researchers in the MOR community. Different studies using a mixed-method have also been developed[46]; other methods involve the Krylov subspace method [47]to find the lower-order model for complex systems.

This chapter also aims to describe some basic concepts used in the research design. First, by defining the controllability/observability gramian, the Singular Value Decomposition and the Hankel norms [13, 48]are used in the methodology. After that, we introduce the artificial neural network, an artificial intelligence technique inspired initially from biological systems, applied in computer science that uses previous states of the system to give the machine the ability to learn and have the position to act without being programmed [49].

II.2 Problem statement

Consider the transfer function $H(s)$ of n^{th} order interval system

$$H(s) = \frac{[a_0^-, a_0^+]s^0 + [a_1^-, a_1^+]s^1 + .. + [a_m^-, a_m^+]s^m}{[b_n^-, b_n^+]s^n + [b_{n-1}^-, b_{n-1}^+]s^{n-1} ... + [b_1^-, b_1^+]s^1 + [b_0^-, b_0^+]} \tag{II.1}$$

Such as $[a_i^-, a_i^+]$, $[b_j^-, b_j^+]$ for $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$ are the interval parameters of $H(s)$. The desired reduced order model $R(s)$ is considered as:

$$R(s) = \frac{[e_0^-, e_0^+]s^0 + [e_1^-, e_1^+]s^1 + .. + [e_v^-, e_v^+]s^v}{[d_r^-, d_r^+]s^r + [d_{r-1}^-, d_{r-1}^+]s^{n-1} ... + [d_1^-, d_1^+]s^1 + [d_0^-, d_0^+]} \tag{II.2}$$

Where $[e_i^-, e_i^+]$, $[d_j^-, d_j^+]$ for $i = 1, 2, ..., v$ and $j = 1, 2, ..., r$ are the interval parameters of $R(s)$.

The goal of this work is to obtain a reduced-order model (r<<n) that preserves and reflects the important properties of the original system as much as possible.

II.3 Basic concepts

II.3.1 Controllability and observability gramians

The following standard interval system presented by the state space, n is the order of this system [12, 9]:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t) \end{aligned} \tag{II.1}$$

where

$$A = \begin{bmatrix} [\alpha_{11}^-, \alpha_{11}^+] & [\alpha_{12}^-, \alpha_{12}^+] & . & [\alpha_{1n}^-, \alpha_{1n}^+] \\ [\alpha_{21}^-, a_{21}^+] & [\alpha_{22}^-, \alpha_{22}^+] & . & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ [\alpha_{n1}^-, \alpha_{n1}^+] & \dots & \dots & [\alpha_{nn}^-, \alpha_{nn}^+] \end{bmatrix} \tag{II.2}$$

such as $\left[\alpha_{ij}^-, \alpha_{ij}^+\right]$ are intervals value represent the elements of matrix A , same thing for B,C,D probably are presented by intervals of values.

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t) \\ y &= C_r x_r(t) + D_r u(t) \end{aligned} \tag{II.3}$$

where r is the order of this reduced system and r<n. Assumed that the pair (A, B) is controllable and (C,A) is observable, the controllablity gramian :

$$W_c = \int_0^{t_f} e^{\tau A} B B^T e^{\tau A^T} d\tau \tag{II.4}$$

the observability gramian

$$W_o = \int_0^{t_f} e^{\tau A^T} C^T C e^{\tau A} d\tau \tag{II.5}$$

where $W_o = [W_o^-, W_o^+]$, $W_c = [W_c^-, W_o^+]$ are computed as the solutions to the following Lyapunov equations:

$$A^T W_o A - W_o = -BB^T \tag{II.6}$$

$$A^T W_c A - W_c = -CC^T \tag{II.7}$$

II.3.2 Singular Values Decomposition and Hankel norm

Definition 1: Any complex matrix φ, of rank k, belonging to m by n (m $*$ n) can be broken down into:

$$\varphi = V \Sigma W^H \tag{II.8}$$

where V and W are unitary matrices. Σ represents the spectrum of singular values of φ, which are numbers σ_i, real, positive or equal to zero, such that:

$$\sigma_1 \geqslant \sigma_2 \geqslant ...\sigma_k \geqslant \sigma_{k+1} = \sigma_{k+2} = ... = \sigma_q = 0 \tag{II.9}$$

The superior singular value is a spectral norm. It is denoted $\bar{\sigma}(\varphi)$ and is equal to:

$$\bar{\sigma}(\varphi) = \sup_{x \neq 0} \frac{||Ax||_E}{||x||_E} \tag{II.10}$$

therefore, it has all the properties of a norm induced by a vector norm.

Definition 2 : We call Hankel singular values of a system G, the square roots of the eigenvalues of the product W_c W_o:

$$\sigma_i = [\lambda_i(W_c W_o)]^{1/2} \tag{II.11}$$

They are therefore positive real numbers $\sigma_i > 0$, and their identity i is equal to the dimension of the state vector.

Definition 3 : We call the Hankel norm of G its greatest Hankel singular value:

$$\|G(s)\|_H = \sup_i(\sigma_i) = \sup_i([\lambda_i(W_c W_o)]^{1/2}) \tag{II.12}$$

II.3.3 Introduction to Artificial Neural Network (ANN)

ANN's are strongly connected networks of elementary processors operating in parallel. Each elementary processor calculates an individual output based on the information it receives. Any hierarchical structure of networks is obviously a network. Today, many terms are used in the literature to designate the field of ANN's , such as connectionism or neuromagnetic. However, each of these names must be associated with precise semantics. Thus, ANN's refer only to manipulated models; it is neither a field of research nor a scientific discipline. Connectionism and neuromagnetic are both research fields in their own right, each manipulating models of ANN's but with different objectives. The objective pursued by connectionist engineers and researchers is to improve computing capabilities using models with strongly connected components. For their part, neuromagnetic manipulate models of ANN's to verify their biological theories of the functioning of the central nervous system. [50]

Each artificial neuron is an elemental processor. It obtains a variable number of inputs from upstream neurons. Each input is associated with a weight w, an abbreviation of weight, representing the strength of the connection. Each essential element has a single outcome, which branches out a varying number of downstream neurons. Each link is associated with a single weight.

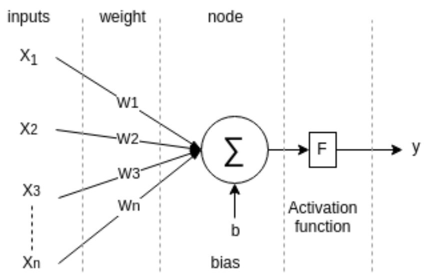


Figure II.1: Single neuron

Figure II.1 shows that The ANN algorithm structure contains several components (inputs $(X_1, X_2, X_3, ..., X_i)$, nodes, edges, biases, activation function, layers), nodes that are called neurons arranged in layers (input layer, hidden layer(s) and output layer). Each connection (edge) between neurons is associated with weight (W_1,W_2,W_3,....,W_n) and the constant (b) is an input to the node. This addition is called a bias node of the ANN ,. Finally, the activation function gives the final output of the neuron F(Q), such as :

$$Q_i = b_i + \sum_{j=1}^{n} W_{ij} * X_j \tag{II.13}$$

$$y_i = F(Q_i) \tag{II.14}$$

Where b_i are constants, W_{ij} is the synaptic weight connecting the j^{th} input to the i^{th} neuron. The links between the neurons describe the model's topology. The neurons are arranged by layer. There is no link between neurons of the same layer, and the connections are only made with the neurons of the downstream layers (fig. II.2). Usually, each neuron in every layer is attached to all founded neurons in the next successive layer and exclusively. Allows us to introduce the notion of direction of information flow (of activation) within a network. Moreover, therefore define the concepts of input neuron and output neuron. The input layer is the set of input neurons by extension, and the output layer is the set of output neurons. The intermediate layers having no contact with the exterior are called hidden layers.

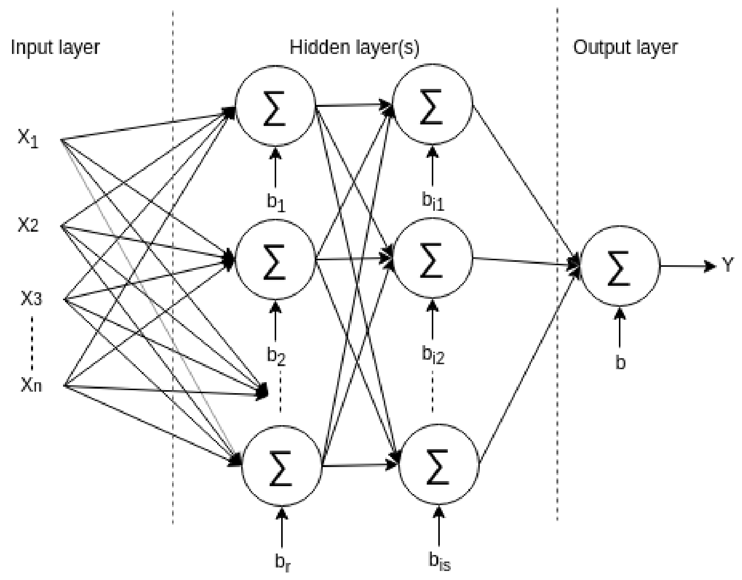


Figure II.2: Definition of the layers of a multilayer network

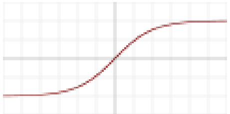
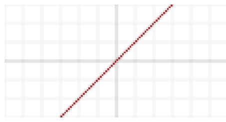
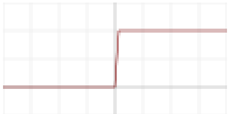
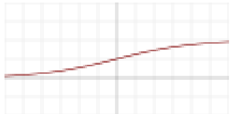
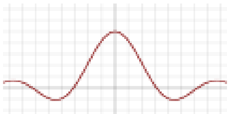
Activation functions

The activation function (or transfer function) is operated to correct the result of the weighted totality of the inputs of a single neuron into outputs. This conversion is carried out by calculating the neuron state introducing a non-linearity in the functioning of the neuron (Cybenko 1989). Bias b plays a role in the threshold. When the result of the weighted sum exceeds this threshold, the argument of the transfer function becomes positive or zero; otherwise, it is considered harmful. Finally, if the result of the weighted sum is:

1. Below the threshold, the neuron is considered non-active
2. Around the threshold, the neuron is considered in the transition phase.
3. Above the threshold, the neuron is considered active.

There are several activation functions that perform some fixed mathematical operation [49], for example :

Table II.1: Some activation functions

Function		Function	Plot
Tanh: $f(x) = tanh(x)$		pure-line : $f(x) = x$	
Binary step : $f(s) = \begin{cases} 0 & for \quad x < 0 \\ 1 & for \quad x >= 0 \end{cases}$		sigmoid: $\sigma(x) = \frac{1}{(1+e^x)}$	
Sinc : $f(s) = \begin{cases} 1 & for \quad x = 0 \\ \frac{sin(x)}{x} & for \quad x \neq 0 \end{cases}$			

Training an ANN

Training is a very important stage in developing a neural network during which the behavior of the network is modified iteratively until the desired behavior is obtained, and this is by adjusting the weights (connection or synapse) of the neurons to a well-defined source of information. Learning also consists of extracting conformity among the data used for training the network, but the essential objective of learning is resolving the problem by forecasting, classification, etc. For an ANN, learning can also be viewed as updating weights (connections) within the network to adjust the network's response to experience and examples. Several types of learning rules can be gathered into two categories: supervised training and unsupervised training[20].

Supervised training is a type of learning; that aims to impose a given operation on the network by forcing the network's outputs to take well-given values (chosen by the operator) and modifying the synaptic weights. The network behaves like a filter whose transfer parameters are adjusted from the presented input-output pairs. The adaptation of the network parameters is carried out using an optimization algorithm. The initiation of the synaptic weights is most often random. Unlike supervised learning, only input values are available in unsupervised learning or adaptive training called "competitive learning." In this case, the examples presented at the input cause the network to self-adapt to generate output values that are close in response to similar input values[20].

II.4 Classical techniques of MOR for standard systems

The idea of reducing a model by balanced truncation is to obtain a reduced-order model by finding and removing those states that are at the same time the least controllable and observable.

II.4.1 Balanced truncation approximation

The idea of reducing a model by balanced truncation is to obtain a reduced-order model by finding and removing those states that are at the same time the least controllable and observable. B. C. Moore [1, 29] was able to propose a similar transformation for such balanced systems, i.e., sets both grammians equal and diagonal

$\hat{P} = \hat{Q} = diag(\sigma_1 \ldots \sigma_n)$

With the diagonal entries in descending order $\sigma_1 \geqslant \sigma_1 \geqslant ..... \geqslant ... \geqslant \sigma_n$. The parameters σ_i are called Hankel singular values (HSV) and are transformation invariant. They are property of the system and depend only on input-output behavior. Hankel singular values can be computed as the square roots of eigenvalues λ_i of the product of P and Q:

$\sigma_i = \sqrt{\lambda_i(P.Q)},\ i = 1, \ldots, n$

It can be shown that the Hankel singular values reflect the contributions of different entries of the state vector to system responses[13] . Hence, to reduce the order of the balanced system $(\tilde{A}, \tilde{B}, \tilde{C})$, it is sufficient to remove (truncate) those state variables (and corresponding blocks in $\tilde{A}, \tilde{B}$, and $\tilde{C}$) related to the smallest Hankel singular values. Balancing and truncation can be done simultaneously by applying a projection to the original non-balanced system, using e. g. square root algorithm [9].

Inputs: A, B, C

Outputs: $\tilde{A}_r, \tilde{B}_r, \tilde{C}_r$the matrices of the reduced, internally balanced system of order r, W_r and V_{r-} mutually orthogonal projection matrices such that

$$\tilde{A}_r = W_r^T A V_r, \tilde{B}_r = W_r^T B \tag{II.1}$$

And

$$\tilde{C}_r = V_r^T C \tag{II.2}$$

1. Calculate the controllability grammian P (see chapter II) and grammian Q's observability by solving the Lyapunov equations.

2. Find the Cholesky factorization of the solutions P and Q:

$$P = L_c L_c^T, Q = L_o L_o^T \tag{II.3}$$

3. Calculate the singular value decomposition of the matrix $L_o^T L_c$:

$$L_o^T L_c = U_1 \Sigma U_2 \tag{II.4}$$

Where $\Sigma \in R^{n\times n}$ is a diagonal matrix of Hankel singular values in descending order.

4. Form the projection matrices $W_r \in R^{n\times r}$ and $V_r \in R^{n\times r}$ as:

$$V_r = L_c U_{2r} \Sigma_r^{-1/2} \tag{II.5}$$

And

$$W_r = L_o U_{1r} \Sigma_r^{-1/2} \tag{II.6}$$

Where U_{1r} and U_{2r} are the first r columns of matrices U_1 and U_2 , and

$$\Sigma_r^{-1/2} = diag(\frac{1}{\sqrt{\sigma_1}}, ..., \frac{1}{\sqrt{\sigma_1}}) \tag{II.7}$$

5. Apply the projection to the system (II.4) to find the order r truncated balanced realization as:

$$\tilde{A}_r = W_r^T A V_r, \tilde{B}_r = W_r^T B, \tilde{C}_r = V_r^T C \tag{II.8}$$

The controllability and observability grammians of the order r reduced system $(\tilde{A}_r, \tilde{B}_r, \tilde{C}_r)$ are diagonal and equal, $\tilde{P}_r = \tilde{Q}_r = diag(\sigma_1, ..., \sigma_r)$. Some alternatives to algorithm 3.1 can be found in [12]
.

The most important characteristic of this algorithm is that it provides a global error bound between the transfer functions of the original and the reduced systems $G(s)$ and $Gr^{BTA}(s)$:

$$\left\| G(s) – Gr^{BTA}(s) \right\|_\infty \leq 2(\sigma_{r+1} + ... + \sigma_n) \tag{II.9}$$

Where the infinity norm $\|.\|_\infty$ denotes the largest magnitude of the difference of transfer functions and $G(s)$ and $Gr^{BTA}(s)$are defined as in

$$G(s) = C^T (sI – A)^{-1} B \tag{II.10}$$

And

$$G_r(s) = C_r^T (sI – A_r)^{-1} B_r \tag{II.11}$$

Respectively. The proof of (II.11) can be found in [51]. The discussion of different error norms is given in. [11]

II.4.2 Singular perturbation approximation

The fact that balanced truncation approximation generally incurs an approximation error in the low-frequency region is undesirable in some practical applications. Hence, an algorithm that produces zero error at zero frequency, called the singular perturbation technique, is obtained as follows. After decomposing the system matrices and the state vector of A as:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, E = [E_1 E_2], x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{II.12}$$

With $A_{11} \in R^{rxr}$, and assuming that A_{22} is invertible, the reduced system is defined by following truncation as [11]:

$$\dot{z} = (A_{11} – A_{12}A_{22}^{-1}A_{21})z + (B_1 – A_{12}A_{22}^{-1}B_2)u \tag{II.13}$$

$$y_r = (C_1 – C_2A_{22}^{-1}A_{21})z – C_2A_{22}^{-1}B_2u(3.8) \tag{II.14}$$

An important property is that the steady state gain matches that of the original system, that is:

$$G_r^{SPA}(0) = G(0) \tag{II.15}$$

It turns out that if we first balance the system, and then truncate , the reduced system fulfills the error bound (see[11] for details):

$$\left\| G(s) – G_r^{SPA}(s) \right\|_\infty \leq 2(\sigma_r + 1 + ... + \sigma_n) \tag{II.16}$$

II.4.3 Hankel norm approximation

The Hankel norm $\|.\|_H$ of $G(s)$ is defined as the maximal Hankel singular value of the system :

$$\|G(s)\| = \sqrt{\lambda_{max}(PQ)} = \sigma_{max} \tag{II.17}$$

The optimal Hankel norm approximation problem is the problem of finding an approximation

$G_r^{HNA}(s)$ of degree $r < n$ such that the Hankel norm of the error

$$\left\| G(s) – G_r^{HNA}(s) \right\|_H \tag{II.18}$$

Is minimized. The lower bound for the above norm is given through [11]:

$$\sigma_{r+1} \leq \left\| G(s) – G_r^{HNA}(s) \right\|_H \tag{II.19}$$

(3.12) Holds for any $G_r(s)$ with exactly r stable poles. Since the infinity norm is never smaller than the Hankel norm [11], also means:

$$\sigma_{r+1} \leq \left\| G(s) – G_r^{HNA}(s) \right\|_\infty \tag{II.20}$$

An algorithm for the construction of an optimal Hankel norm approximation can be found in [11]. An optimal Hankel norm approximation $G_r^{HNA}(s)$ of order r fulfills the following upper bound:

$$\left\| G(s) – G_r^{HNA}(s) \right\|_\infty \leq (\sigma_r + 1 + ... + \sigma_n) \tag{II.21}$$

Which is half the bound for the balanced truncation case.[11]

II.4.4 The Γ-δ routh approximation technique

In this section, an extended technique is presented by modifying the γ-δ table approximation technique [4] that incorporates the concepts of [45]. The developed algorithm consists of the following steps to obtain a reduced order model:

Step 1. Bilinear transformation of higher-order systems To maintain the stability rule, the bilinear transformation of higher-order systems is performed by substituting

$$z = \frac{1+s}{1–s}(7) \tag{II.22}$$

then the n th order higher order system is obtained as

$$T_n(p) = \frac{[c_1^-, c_1^+]s^{n-1} + [c_2^-, c_2^+]s^{n-2} + ... + [c_n^-, c_n^+]}{[d_0^-, d_0^+]s^n + [d_1^-, d_1^+]s^{n-1} + ... + [d_n^-, d_n^+]} = \frac{\sum [c_i^-, c_i^+]s^{n-i}}{\sum [d_j^-, d_j^+]s^{n-j}} = \frac{N_n(s)}{D_n(s)} \tag{II.23}$$

Where (i=1,2,...,n) and (j=0,1,2,..,n) are the interval parameters

Step 2. Formation of γ - table

The first two rows of the γ table are obtained from the coefficient of the denominator of $G_n(s)$, and the remaining entries are filled. Finally, the γ – table is formed as below

$$\begin{bmatrix} [d_n^-, d_n^+] & [d_{n-2}^-, d_{n-2}^+] & [d_{n-4}^-, d_{n-4}^+] & \cdots \\ = [d_{00}^-, d_{00}^+] & = [d_{01}^-, d_{01}^+] & = [d_{02}^-, d_{02}^+] & \cdots \\ [d_{n-1}^-, d_{n-1}^+] & [d_{n-3}^-, d_{n-3}^+] & [d_{n-5}^-, d_{n-5}^+] & \\ = [d_{10}^-, d_{10}^+] & = [d_{11}^-, d_{11}^+] & = [d_{12}^-, d_{12}^+] & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ [d_{n-1,0}^-, d_{n-1,0}^+] & \cdots & \cdots & \cdots \\ = [d_{n,0}^-, d_{n,0}^+] & \cdots & \cdots & \cdots \end{bmatrix} \tag{II.24}$$

Where

$$\gamma_k = \frac{[d_{k-1,0}^-, d_{k-1,0}^+]}{[d_{k,0}^-, d_{k,0}^+]} / k = 1.2.3.... \tag{II.25}$$

And

$$[d_{i,j}^-, d_{i,j}^+] = [d_{i-2,j+1}^-, d_{i-2,j+1}^+] - \frac{[d_{i-2,0}^-, d_{i-2,0}^+][d_{i-2,j+1}^-, d_{i-2,j+1}^+]}{[d_{i-2,0}^-, d_{i-2,0}^+]} \tag{II.26}$$

Step 3. Formation of δ - table

The first two rows of the table are obtained from the coefficient of numerator polynomial of $G_n(p)$, and the remaining entries may be filled by applying the following formula.

Theδ - table is constructed as

$$\begin{bmatrix} [c_n^-, c_n^+] & [c_{n-2}^-, c_{n-2}^+] & [c_{n-4}^-, c_{n-4}^+] & \cdots \\ = [c_{10}^-, c_{10}^+] & = [c_{11}^-, c_{11}^+] & = [c_{12}^-, c_{12}^+] & \cdots \\ [c_{n-1}^-, c_{n-1}^+] & [c_{n-3}^-, c_{n-3}^+] & [c_{n-5}^-, c_{n-5}^+] & \\ = [c_{20}^-, c_{20}^+] & = [c_{21}^-, c_{21}^+] & = [c_{22}^-, c_{22}^+] & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ [c_{n-1,0}^-, c_{n-1,0}^+] & \cdots & \cdots & \cdots \\ = [c_{n,0}^-, c_{n,0}^+] & \cdots & \cdots & \cdots \end{bmatrix} \tag{II.27}$$

Where

$$\delta_k = \frac{[c_{k,0}^-, c_{k,0}^+]}{[d_{k,0}^-, d_{k,0}^+]} \tag{II.28}$$

With k=1.2.3.... and

$$[c_{i,j}^-, c_{i,j}^+] = [c_{i-2,j+1}^-, c_{i-2,j+1}^+] - \frac{[c_{i-2,0}^-, c_{i-2,0}^+][d_{i-2,j+1}^-, d_{i-2,j+1}^+]}{[d_{i-2,0}^-, d_{i-2,0}^+]} \tag{II.29}$$

Step 4. The r th order general bilateral transformed model is evaluated as

$$R_r(s) = \frac{C_r(s)}{D_r(s)} / r = 1, 2, ...(14) \tag{II.30}$$

Where

$$D_r(s) = p^2 D_{r-2}(s) + [\gamma_r^-, \gamma_r^+] D_{r-1}(s) \tag{II.31}$$

$$C_r(s) = [\delta_r^-, \delta_r^+] s^{r-1} + s^2 C_{r-2}(s) + [\gamma_r^-, \gamma_r^+] C_{r-1}(s) \tag{II.32}$$

And

$$D_{-1}(s) = \frac{1}{p}, D_0(s) = 1, C_{-1}(s) = 0, D_0(s) = 0, \tag{II.33}$$

Step 5. Finally, apply inverse bilinear transformation by substituting $s = \frac{z-1}{z+1}$ in Eq. (14), the required reduced order model is obtained as

$$R_{m\gamma\delta}(z) = \frac{C_{m\gamma\delta}(z)}{D_{m\gamma\delta}(z)} \tag{II.34}$$

Where

$$D_{m\gamma\delta}(z) = \left(\frac{z-1}{z+1}\right)^2 D_{r-2}\left(\frac{z-1}{z+1}\right) + [\gamma_r^-, \gamma_r^+] D_{r-1}\left(\frac{z-1}{z+1}\right) \tag{II.35}$$

$$C_{m\gamma\delta}(z) = [\delta_r^-, \delta_r^+]\left(\frac{z-1}{z+1}\right)^{r-1} + \left(\frac{z-1}{z+1}\right)^2 C_{r-2}\left(\frac{z-1}{z+1}\right) + [\gamma_r^-, \gamma_r^+] C_{r-1}\left(\frac{z-1}{z+1}\right) \tag{II.36}$$

Remark : Since the point by point technique for every two successive rows of the γ table is maintained using Dolgin and Zeheb's concept [52], the presented method ensures the stability of the reduced discrete interval model, provided the given higher order system is robustly stable.[52, 4]

II.4.5 Genetic Algorithm

This method presents an algorithm for order reduction of linear interval systems based on minimization of the ISE by genetic algorithm (GA)[53]. Consider a high order linear SISO interval system represented by the transfer function as

$$G(s) = \frac{N(s)}{D(s)} = \frac{[c_1^-, c_1^+] + [c_2^-, c_2^+]s + [c_2^-, c_2^+]s^2 + ... + [c_{n-1}^-, c_{n-1}^+]s^{n-1}}{[d_0^-, d_0^+] + [d_1^-, d_1^+]s + [d_1^-, d_1^+]s^2 .... + [d_n^-, d_n^+]s^n} \tag{II.37}$$

Where $[c_i^-, c_i^+]$ i=1,2,...n-1 and $[d_i^-, d_j^+]$ j=1,2,...n are the interval coefficients of higher order numerator and denominator polynomials respectively. The objective is find a r^{th} order reduced interval system. let corresponding r^{th} order reduced model is

$$G_r(s) = \frac{[c_1^-, c_1^+] + [c_2^-, c_2^+]s + [c_2^-, c_2^+]s^2 + ... + [c_{r-1}^-, c_{r-1}^+]s^{r-1}}{[d_0^-, d_0^+] + [d_1^-, d_1^+]s + [d_1^-, d_1^+]s^2 .... + [d_r^-, d_r^+]s^r} \tag{II.38}$$

Where $[c_i^-, c_i^+]$ i=1,2,...r-1 and $[d_i^-, d_j^+]$ j=1,2,...r are the interval parameters of reduced order numerator and denominator polynomial, respectively.

The denominator and numerator coefficients of the MOR are selected by minimizing the Integral square error between the brief part of the step response of an original system and the reduced system using a genetic algorithm. The deviation of the reduced order system from the initial system response is given by the error-indea 'ISE,' known as the Integral square error, which is given as follows:

$$ISE = \int_0^\infty [g(t) - r(t)]^2 dt \tag{II.39}$$

Where G(t) and R(t) stand for the unit step response of the initial and reduced order systems, respectively.

In this method, GA minimizes the objective function 'ISE' as given. The parameter to be determined is the coefficients of the numerator and denominator of the lower order system.

For minimization, routines from the GA optimization toolbox are operated. For various problems, the same parameters for GA probably do not give the best solution, so these can be changed according to the situation. In Table 2.2, the typical parameters for GA optimization routines are used in the present study.[53, 12, 54, 55]

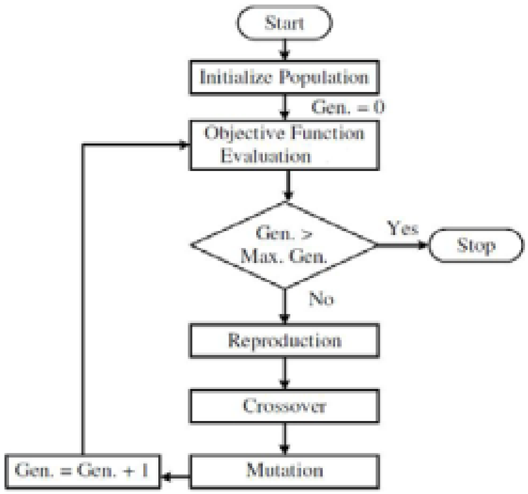


Figure II.1: Flowchart for Genetic algorithm

One more essential point that involves the optimal solution, more or less, is the range for unknowns. For the very first performance of the program, a more comprehensive solution space can be provided. After obtaining the solution, one can shorten the solution space close to the values got in the previous iteration. The computational flow chart of the presented algorithm is demonstrated in Fig. II.3.

Table II.2: Parameters used by the GA

Designation	Value(type)
Nmbr of generations	200
size of Population	100
selection Type	uniform
crossover Type	Arithmetic
mutation Type	uniform
method of Termination	Maximum generation

Note: The proposed algorithm guarantees stability for a stable higher order linear Interval system, and thus any lower order Interval model can be derived with reasonable accuracy.[53, 12, 54, 55]

II.5 Conclusion

Model Order Reduction (MOR) was designed in the area of systems and control theory, which analyzes the properties of dynamical systems in reducing their complexity while maintaining their input-output behavior as much as possible. Numerical mathematicians have also taken up the field, MOR tries to capture the essential features of a structure quickly. This indicates that in the earlier stage of the process, the most fundamental properties of the initial model must already be shown in the smaller approximation. Then, at a specific moment, the procedure of reduction is stopped. At that point, all critical properties of the original model must be captured with acceptable precision. The operation maintains automatically.

Part C

Contribution

Chapter III

Proposed MOR approaches for interval systems

III.1 Introduction

Today, there is a significant effort in the control engineering community to work on complex systems and facilitate the modeling process of these kinds of systems. This is why they seek to create an approximation of a highly dynamic system to reproduce a system with the exact possible properties of the initial system. In another definition, it is required to use MOR to reduce the complexity of analyzing systems (see chapter II). to achieve such results, we discuss two methods in this chapter:

The first is a classical modified approach based on projections. that are widely used in MOR because of their efficacity. This method uses the crucial tool of the Singular Value Decomposition (SVD) [56] that is extended to interval systems model reduction. This method materializes the state-space presentation, and the reduced model is assembled by discarding the state of the most negligible energy that contributes to the original systems' behavior [57]. Furthermore, many methods have been developed in the international literature, containing machine learning algorithms [2].

Therefore, the second idea presented in this chapter is to use ANN implementation to find a reduced model order reduction.. This novel algorithm is valid in generalized fixed / interval systems and their different forms: continuous-time and discrete-time. Furthermore, it is extended to Multi-input/multi-output interval and fixed point interval systems.

In different applications of these two methods (see chapter IV), the results are always better than what has been achieved previously. Our proposed method ensures stability and better performance tuning than existing methods[56, 57].

III.2 Projection approach for MOR interval systems

III.2.1 Description of the methods

Inputs : Consider the linear, interval system

$$H(s) = \frac{[a_0^-, a_0^+]s^0 + [a_1^-, a_1^+]s^1 + .. + [a_m^-, a_m^+]s^m}{s^n + [b_{n-1}^-, b_{n-1}^+]s^{n-1} ... + [b_1^-, b_1^+]s^1 + [b_0^-, b_0^+]} \tag{III.1}$$

Step1 : Transform it into its two state space representations of order n, for continuous-time [58], and discrete-time [48] respectively,

- Using [9], re-write the original continuous-time interval dynamical system as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{III.2}$$

On the base of [48], the original discrete-time interval systems will be described as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{III.3}$$

Where

$$A=\begin{bmatrix} [a_{11}^{-},a_{11}^{+}] & [a_{12}^{-},a_{12}^{+}] & . & [a_{1n}^{-},a_{1n}^{+}] \\ [a_{21}^{-},a_{21}^{+}] & [a_{22}^{-},a_{22}^{+}] & . & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ [a_{n1}^{-},a_{n1}^{+}] & \dots & \dots & [a_{nn}^{-},a_{nn}^{+}] \end{bmatrix} \tag{III.4}$$

With $\left[a_{ij}^{-},a_{ij}^{+}\right]$ Is interval's value representing the elements of the matrix A, B, C and D are built, in accordance with their sizes, in the same way.

Step 2: Compute the controllability G_c and the observability G_o gramians, [59], combined to [60, 9], and [11], as

- For the continuous-time interval systems, [9],

$$G_c=\int_0^{t_f} e^{\tau \mathrm{A}}BB^T e^{\tau A^T}d\tau \tag{III.5}$$

$$G_o=\int_0^{t_f} e^{\tau A^T}C^T Ce^{\tau \mathrm{A}}d\tau \tag{III.6}$$

- For discrete-time interval system,[48],

$$G_c=\sum_{k=0}^{\infty} A^{\mathrm{k}}BB^T(A^{\tau})^k \tag{III.7}$$

$$G_o=\sum_{k=0}^{\infty} (A^{\tau})^k C^T CA^{\mathrm{k}} \tag{III.8}$$

With$G_o=[G_o^-,G_o^+]$, $G_c=[G_c^-,G_o^+]$

Step 3 : Considering [9], one can put G_c G_o into the Schur form, and compute$W_A$and W_D the real orthogonal transformations

$$V_A^T G_c G_o V_A=\begin{bmatrix} \lambda_{A_n} & * & \dots & * \\ 0 & \lambda_{A_{n-1}} & * & \vdots \\ \vdots & \ddots & \ddots & * \\ \vdots & \dots & 0 & \lambda_{A_1} \end{bmatrix} \tag{III.9}$$

$$V_D^T G_c G_o V_D = \begin{bmatrix} \lambda_{D_n} & * & \cdots & * \\ 0 & \lambda_{D_{n-1}} & * & \vdots \\ \vdots & \ddots & \ddots & * \\ \vdots & \dots & 0 & \lambda_{D_1} \end{bmatrix} \tag{III.10}$$

Where λ_{A_i}=λ_{D_i}=λ_i^2 , i = 1,2,.....,n , are the Hankel singular values $\lambda_i = [\lambda_i^-, \lambda_i^+]$ such as $\lambda_1 > \lambda_2 > ... > \lambda_n > 0$ [56, 9].

Step 4 :Determination of the reduced order r of the model

$$[V_A, E, V_D] = SVD(M_w) \tag{III.11}$$

Such as SVD: singular value decomposition [56, 61] , which means which satisfies the following inequality [56, 8]:

$$\sqrt{\sum_{i=1}^{r} \lambda_i^4} >> \sqrt{\sum_{i=r+1}^{n} \lambda_i^4} \tag{III.12}$$

Where r should satisfy the condition of (r$\neq$0 & r$\neq$n) for more efficiency.

Step 5 : Split W_A and W_D [56]. as

$$W_A = \left[\overbrace{W_{d.p}}^{n-r} \;\; \overbrace{W_{g.g}}^{r} \right] \tag{III.13}$$

$$W_D = \left[\overbrace{W_{d.g}}^{r} \;\; \overbrace{W_{g.p}}^{n-r} \right] \tag{III.14}$$

Step 6 : Put $P_g = W_g^T W_{d.g}$and construct the SVD form

$$P_g = U_{p.g} E_{p.g} W_{p.g} \tag{III.15}$$

Step 7 : from chapter II - section 03 compute the projection matrices [56]:

$$S_{g.g} = W_{g.g} U_{p.g} E_{p.g}^{-1/2} \tag{III.16}$$

$$S_{d.g} = W_{d.g} U_{p.g} E_{p.g}^{-1/2} \tag{III.17}$$

Where S$_{g.g}$ and S$_{d.g}$ are matrices of n a r dimension.

Output : From the Step 4 evaluation, the reduced order state-space interval system A_r, B_r, C_r, D_r is constructed as

$$\begin{bmatrix} A_r & B_r \\ C_r & D_r \end{bmatrix} = \begin{bmatrix} S_{g.g}^T A S_{d.g} & S_{g.g}^T B \\ C S_{d.g} & D \end{bmatrix} \tag{III.18}$$

(End of the procedure [56])

III.3 Machine learning algorithm for MOR interval systems

III.3.1 Description of (ANN-PA)

Consider the following polynomial P of degree n:

$$P(x) = a_0 x^0 + a_1 x^1 + a_2 x^2 + ... + a_n x^n \tag{III.1}$$

Where [a_i] for (i=0,1,2,..,n) are parameters of polynomial P of and a is the variable, in this part, we show that high degree polynomials can be approximated by a lower degree polynomials using the ANN algorithm under the supervision of the high degree original polynomial data. The training algorithm carries out the Supervised training process. The suggested technique employs two neurons positioned in one hidden layer with the activation function. F(x)=x. The inputs of the ANN algorithm are arranged in a vector which each element in this vector is related to the main variable x as follows :

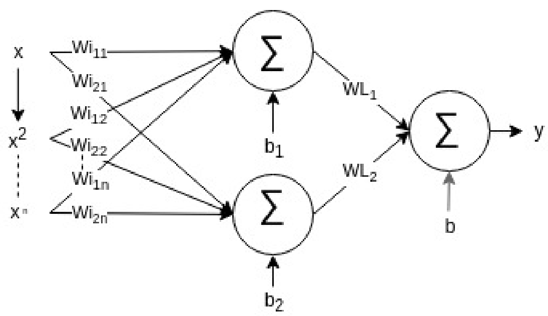


Figure III.1: n^{th}degree (ANN -PA) architecture

The approximation of high degree polynomial by a polynomial of degree n requires the inputs vector $X = [x^1, x^2, x^3, ..., x^n]$, by applying a to the ANN model described above using equation (III.20) that describe the model of neural network, as can be seen in Figure III.1, we obtain the model of ANN , the result of the weight adjustment is giving by [57, 49]:

$$y = f(b + \sum_{k=1}^{2} [WL_K f(b_k + \sum_{m=1}^{n} Wi_{km} x^m)]) \tag{III.2}$$

Such that (x,y) are the input-output of the system respectively,by using equation (III.20), b,$b_i$are constants,Wi_{km} is the synaptic wiegth connecting the m^{th} input to the k^{th} neuron in the hidden layer, WL_k is the synaptic wieght connectiing the k^{th} neuron to the output neuron in the output layer. And f is the activation function. such that $f(x) = x$. We would have, therefore:

$$y = b + \sum_{k=1}^{2} [WL_K (b_k + \sum_{m=1}^{n} Wi_{km} x^m)] \tag{III.3}$$

For k=1,2 the number of neuron, where:

$$y = b + [WL_1(b_1 + \sum_{m=1}^{n} Wi_{1m} x^m)] + [WL_2(b_2 + \sum_{m=1}^{n} Wi_{2m} x^m)] \tag{III.4}$$

Finally, the n^{th} degree polynomial is written as

$$y = b + [WL_1(b_1 + \sum_{m=1}^{n} Wi_{1m} x^m)] + [WL_2(b_2 + \sum_{m=1}^{n} Wi_{2m} x^m)] \tag{III.5}$$

If we fia r=2 so n=2:

$$y = \underbrace{(b + WL_1 b_1 + WL_2 b_2)}_{A} + \underbrace{(WL_1 Wi_{11} + WL_2 Wi_{21})}_{B} x + \underbrace{(WL_1 Wi_{12} + WL_2 Wi_{22})}_{C} x^2 \tag{III.6}$$

Where $y = A + Bx + Cx^2$

Note : ANN-P.A is applicable to interval polynomials as in fixed parameter polynomials using the basic concept of interval arithmetic.[57]

III.3.2 Model order reduction based on ANN procedure (ANN -MOR)

The method used to reduce a system is applied to N(s) and D(s) separately by approximation of each pole and zeros independently. We generate the training data by applying an output data, for example (a = 1 to 100) using i = 0.1 as a step, where data density gives good results and takes more time in the phase of learning, the training data is [x, y] where y is the output of each x (N (x) = y). by applying a to the ANN-P.A under the supervision of the original polynomial, we make training to get the best result where $Y_{ann} = y$. The neurons are arranged especially in this way to obtain a simple algorithm that gives a minimized order with less computational resources during the training, to obtain a fast and exclusive result that can be applied directly in real applications which require quick responses by using the ANN modeling. In order to reduce the system, the application of neural networks to interval systems is presented and described in the following step-by-step scheme[57]:

Input :

The transfer function of a stable n^{th} order interval system is represented by:

$$H(s) = \frac{[a_0^-, a_0^+]s^0 + [a_1^-, a_1^+]s^1 + .. + [a_m^-, a_m^+]s^m}{[b_n^-, b_n^+]s^n + [b_{n-1}^-, b_{n-1}^+]s^{n-1} ... + [b_1^-, b_1^+]s^1 + [b_0^-, b_0^+]} \tag{III.7}$$

Using the ANN-P.A technique presented in the previous section, the MOR diagram is shown in figure III.2 :

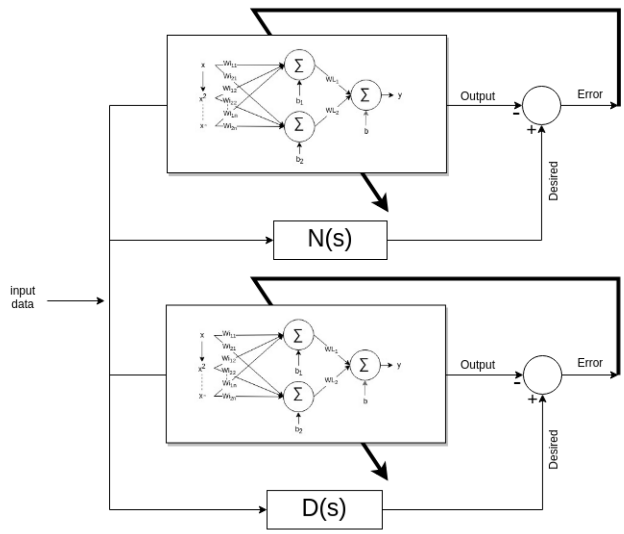


Figure III.2: Model order reduction architecture (ANN -MOR)

Step 1 : Definition of the degree of the approximated numerator (q) and the degree of the approximated denominator (r), where $q <= r$. [57]

Step 2 : Train the algorithm under the supervision (see chapter II) of the denominator and the numerator of the interval transfer function, each separately using the proposed ANN-P.A method.

Step 3 : Extract the results of the algorithm (the two new reduced degree polynomials $N_r(s)$ $D_r(s)$)

and form the reduced model (transfer function) :

$$H_r(s) = \frac{N_r(s)}{D_r(s)}. \tag{III.8}$$

Step 4: After obtaining the ANN algorithm results, the SSE steady-state error is adjusted with the gain correction factor η between the original model and the reduced model following [57, 62].

The output of the algorithm :

$$R_r(s) = \eta \frac{N_r(s)}{D_r(s)} \tag{III.9}$$

(End of procedure)[57]

III.4 Conclusion

This chapter summarizes our published research on MOR, which is firstly an approach for the MOR of interval systems based on the SVD algorithm that is discussed and developed. From the combination of the SVD algorithm and the Schur decomposition, a special projection is built in the state space. The basic idea is to carry out the order reduction on the upper and lower subsystems of the full-order interval system and subsequently compute and plot the responses included between those of the subsystems. The proposed approach gives reduced order models approximating with a small deviation the complete order system, whether in the continuous or frequency domain, and the stability is always ensured. [56]

In the second part of this chapter, since modern heuristic tools [2] have appeared recently in the field of engineering, we introduce a new technique based on one of the advanced methods in intelligence artificial applied to linear interval systems, especially ANN's Algorithm. This proposed algorithm is based on machine learning and designed for model order reduction by approximating the numerator polynomial and the denominator polynomial separately using a simple ANN architecture supervised by the original high degree polynomial. The algorithm contains two neurons organized in one hidden layer and a modified input vector. The goal of the algorithm training is to reduce the error between the original and the reduced polynomial. Finally, the implementation of both approaches provides stability and shows better performance matching. They are applied either for generalized interval systems (continuous time, discrete time) or extended to multi-input/multi-output interval systems and fixed-point intervals. In the next chapter of the thesis, we validate the methods by numerical examples and analysis that are provided and compared to already existing methods.[57]

Chapter IV

Simulation and interpretation

IV.0.1 Introduction

This chapter presents the numerical results obtained through the proposed methods applied to interval systems, and this analytical study focuses on comparing the application of algorithms to systems offered previously in the literature and reduced by previous algorithms; a comparison is made with the results achieved by our algorithms. The most famous and most efficient methods were chosen to carry out this numerical, analytical study. The mean squared error is used to calculate the error's between results. And every result is shown by a plot figure of Step and Frequency, responses of the original system, and reduced models. These results are analyzed and interpreted to show the benefit of the proposed model. Finally, a conclusion is presented.[56, 57]

IV.1 Projection approach for MOR interval systems

In this section, and to prove the efficiency of the approach, two examples of different nature are treated. The first example deals with a digital interval system, and the other shows the generalization of the proposed algorithm to MIMO (2 inputs / 2 outputs) system. In order to validate our approach, the proposed reduced order model is compared to others of the same order obtained from recent works. The time responses (impulse response and step response) and frequency responses (magnitude and phase spectra) are calculated and plotted. The stability of the original system of complete order and its reduced model is verified. In addition, and to quantify and assess the quality of the reduced order model resulting from our algorithm, the MSE is calculated and is compared to that resulting from other reduced models via works cited in references.[56]

IV.1.1 Example 1

Consider the 3^{th} full order interval system system previously investigated by G.Sarawathi [63] presented by the transfer functions :

$$H(s) = \frac{s^2 + [3.3, 6.5]s + [2.7, 10]}{s^3 + [8.5, 8.6]s^2 + [18, 18.2]s + [10.25, 10.76]} \tag{IV.1}$$

The four associated kharitonov polynomials are stable

$$k_1(s) = 10.76 + 18.20s^1 + 8.5s^2 + 1s^3 \tag{IV.2}$$

$$k_2(s) = 10.25 + 18.2s^1 + 8.60s^2 + 1s^3 \tag{IV.3}$$

$$k_3(s) = 10.25 + 18.2s^1 + 8.60s^2 + 1s^3 \tag{IV.4}$$

$$k_4(s) = 10.76 + 18s^1 + 8.50s^2 + 1s^3 \tag{IV.5}$$

By using the proposed procedure of reduction[56], the following second-order model is obtained

$$R(s) = \frac{s + [1.358, 6.471]}{s^2 + [6.568, 8.592]s + [5.15, 6.869]} \tag{IV.6}$$

The four associated kharitonov polynomials are stable

$$k_1(s) = 6.869 + 8.592s^1 + s^2 \tag{IV.7}$$

$$k_2(s) = 5.150 + 8.592s^1 + s^2 \tag{IV.8}$$

$$k_3(s) = 5.150 + 6.568s^1 + s^2 \tag{IV.9}$$

$$k_4(s) = 6.869 + 6.568s^1 + s^2 \tag{IV.10}$$

The following 2^{nd} order model is determined, using the procedure (G.Sarawathi) of reduction proposed in [63] :

$$R(s) = \frac{s + [1.111, 3.773]}{s^2 + [5.667, 5.773]s + [3.8672, 4.4263]} \tag{IV.11}$$

The four associated kharitonov polynomials are stable

$$k_1(s) = 4.426 + 5.773s^1 + s^2 \tag{IV.12}$$

$$k_2(s) = 3.867 + 5.773s^1 + s^2 \tag{IV.13}$$

$$k_3(s) = 3.867 + 5.667s^1 + s^2 \tag{IV.14}$$

$$k_4(s) = 4.426 + 5.667s^1 + s^2 \tag{IV.15}$$

Sum squared error performance:

Table IV.1: Comparison of SSE for reduced order models (Example 1)

Methods	lower limit	upper limit
Proposed method [56](step resp)	2.5483e-08	7.4936e-05
Proposed method [56](impulse resp)	7.8091e-07	0.0011
G.Sarawathi [63] (step resp)	4.8394e-04	0.0052
G.Sarawathi [63] (imupse resp)	0.0062	0.0293

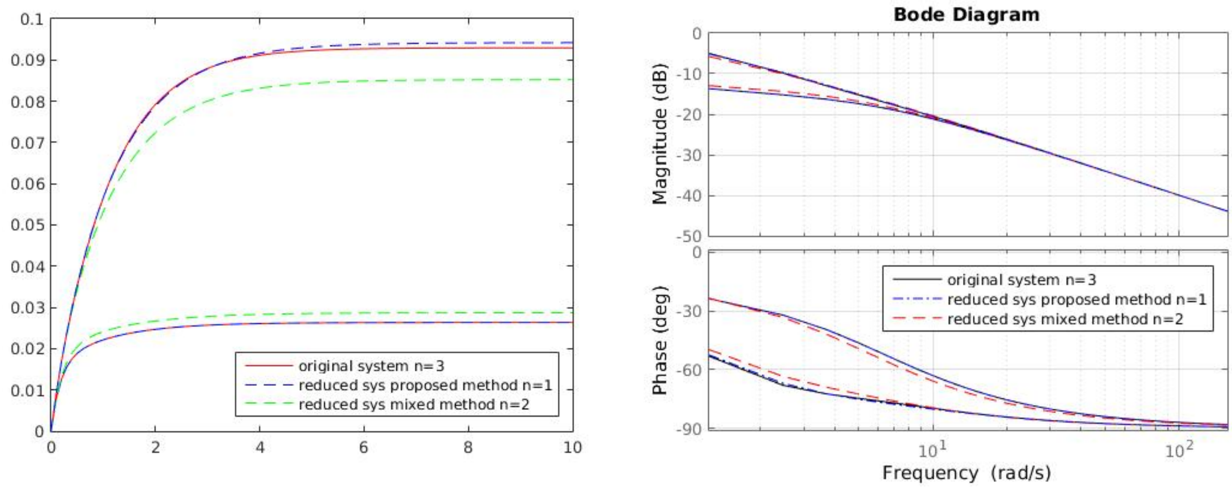


Figure IV.1: Step and frequency responses of original and reduced systems (Example 1)

IV.1.2 Example 2

Consider a 7^{th} full order interval system system previously investigated by DK.Saini [53] presented by the transfer functions :

$$H(s) = \frac{Num}{Den} \tag{IV.16}$$

the parameters of the denominator and the denominator are presented in the following vectors :

Num={[1.9,2.1],[24.7,27.3],[157.7,174.3],[542,599],[930,1028] ,[721.8,797.8],[187.1,206.7]}.

Den={[0.95,1.05],[8.779,9.703],[52.23,57.73],[182.9,202.1],[429,474.2],[572,632.7.5]

,[325.3,359.5],[57.35,63.39]}.

The four associated kharitonov polynomials are stable; the parameters of the kharitonov polynomials are presented in the following vectors.

$$k_1(s) = [0.950, 8.779, 57.729, 202.125, 429.02, 572.47, 359.52, 63.389] \tag{IV.17}$$

$$k_2(s) = [0.95, 9.703, 57.729, 182.875, 429.02, 632.73, 359.52, 57.352] \tag{IV.18}$$

$$k_3(s) = [1.05, 9.703, 52.231, 182.875, 474.18, 632.73, 325.28, 57.352] \tag{IV.19}$$

$$k_4(s) = [1.05, 8.779, 52.231, 202.125, 474.18, 572.47, 325.28, 63.389] \tag{IV.20}$$

By using the proposed procedure of reduction[56], the following 3^{rd}order model is obtained

$$R(s) = \frac{1.756s + 15.44s + 13.38}{s^3 + 2.595s^2 + 11.7s + 4.158} \tag{IV.21}$$

The four associated kharitonov polynomials are stable

$$k_1(s) = 4.157 + 11.703s^1 + 2.595s^2 + s^3 \tag{IV.22}$$

$$k_2(s) = 4.157 + 11.703s^1 + 2.595s^2 + s^3 \tag{IV.23}$$

$$k_3(s) = 4.157 + 11.703s^1 + 2.595s^2 + s^3 \tag{IV.24}$$

$$k_4(s) = 4.157 + 11.703s^1 + 2.595s^2 + s^3 \tag{IV.25}$$

The following 2^{nd} order model is determined, using the Genetic Algorithm (GA) (DK.Saini) of reduction proposed in [53] :

$$R(s) = \frac{[364.7429.7]s + [271.7, 293.2]}{[61.5, 68.99]s^2 + [255.7, 347.1]s + [83.8, 87.67]} \tag{IV.26}$$

The four associated kharitonov polynomials are stable

$$k_1(s) = 87.67 + 347.1s^1 + 61.5s^2 \tag{IV.27}$$

$$k_2(s) = 83.80 + 347.1s^1 + 68.99s^2 \tag{IV.28}$$

$$k_3(s) = 83.80 + 255.7s^1 + 68.99s^2 \tag{IV.29}$$

$$k_4(s) = 87.67 + 255.7s^1 + 61.5s^2 \tag{IV.30}$$

Sum squared error performance :

Table IV.2: Comparison of SSE for reduced order models (Example 2)

Methods	lower limit	upper limit
Proposed method [56](step resp)	0.001353	0.0013525
Proposed method [56](impulse resp)	0.11819	0.11818
GA method [53] (step resp)	0.011972	18.156
GA method [53] (impulse resp)	31.374	37.941

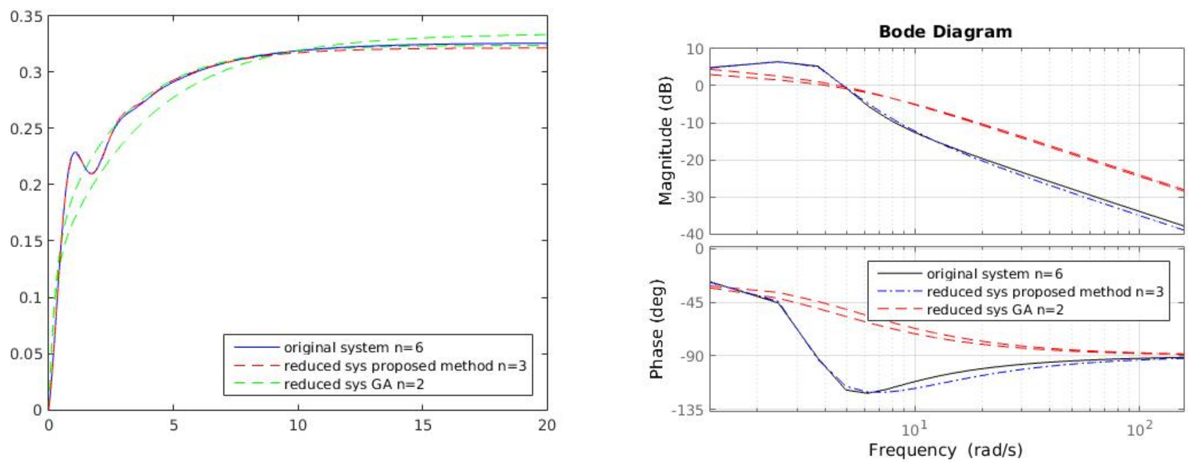


Figure IV.2: Step and frequency responses of original and reduced systems (Example 2)

IV.1.3 Results and discussion

As it was shown, all time and frequency responses show the upper and lower bound interval systems. According to the simulation results in the two examples, we notice that the behavior of the reduced order models calculated via our approach based on the SVD algorithm [56] follows the complete order model faithfully, whether in the time domain , and in the frequency domain (see Fig. IV.1 and Fig. IV.2,) show the superiority of the proposed model, in the time and frequency domains. Concerning the MIMO model, time and frequency responses are traced for the first channel (E1/S1). Same plots and performances are obtained for the other channels. The deviation of the reduced order system from higher-order original system response is given by the Mean Square Error (MSE) criterion, a robust tool [64], given under its two expressions as well for continuous-time systems [54], for discrete-time systems [59]. The superiority of our proposed model over others from cited works is clearly shown in the MSE tables (Table IV.1 for the first example and Tables IV.2 for the second example). The MSE is calculated either for impulse or step responses. This permits enhancement of the superiority of our reduced order models compared to others resulting from recent work. [56]

IV.1.4 Interpretation

The output step and frequency responses of the original full order systems and their reduced interval models constructed from the high-order interval system of Example 1 and Example 2 are plotted in Figs (1 and 2), Figs (4 and 5), respectively. The Poles-zero's are mapped in Figs (3 and 6). All the algorithms presented [56] in this study are applied to stable, high-order systems. The results of these methods give a stable, low-order system, as shown the Figs (3 and 6). The step and frequency responses presented in Fig. IV.1 and Fig. IV.2, show that the behavior of the original system is excellently preserved in the output of the proposed method compared to the previous algorithm GA and the mixed method of G. Sarawathi, and show good agreement, simplicity and effectiveness. As may be seen in the proposed method, the step responses plots are almost identical, and the sum squared error tends to zero, on the other hand, and on the contrary to the method presented, our method shows the power of efficiency on the step responses and the frequency responses at the same time[56].

IV.2 Machine learning algorithm for MOR interval systems

To evaluate the new model order reduction algorithm (ANN -MOR) [57] proposed and described above, different dynamical systems which have been previously treated in the literature are discussed and presented in this section. The deviation (performance matching) of the reduced order system from the higher-order original system response is given by the mean squared error (MSE), which is given by:

$$MSE = \frac{1}{n}\int_{1}^{\infty}[H(s)-R(s))]^2 \tag{IV.1}$$

and for descrit time systems:

$$MSE = \frac{1}{n}\sum_{1}^{\infty}[H(s)-R(s))]^2 \tag{IV.2}$$

Where $H(s)$, are the step responses of the n^{th} order interval system (original system) and the desired reduced order model, respectively.

In order to carry on with the performance analysis, Kharitonov theorem [14] is used to verify the stability of the reduced system using the proposed [57] and existing techniques as discussed in the following examples:

IV.2.1 Example 3 (discrete time system 'fixed coefficients')

Let us consider the all pole 8^{th} system described by the transfer function [18, 17]:

$$H(z) = \frac{N_1(z)}{D_1(z)} \tag{IV.3}$$

Where :

$N_1(z) = 0.4209z^7 + 0.2793z^6 - 0.0526z^5 + 0.038z^4 - 0.1291z^3 - 0.0656z^2 + 0.011z - 0.0015$ $D_1(z) = z^8 - 0.4209z^7 - 0.2793z^6 + 0.0526z^5 - 0.038z^4 + 0.1291z^3 + 0.0656z^2 - 0.011z + 0.0015$

In this example, we only reduce the denominator using the ANN-P.A (case n=2 in the chapter III) and the ANN -MOR [57] steps. The obtained second-order model is given by:

$$R_1(z) = \frac{0.4497z - 0.2897}{z^2 - 1.524z + 0.6836} \tag{IV.4}$$

The following figure shows the variation of MSE during the training operation, validation, and test performance of the ANN training record. This final result shows the best training performance and the number of epochs of the training. [57]

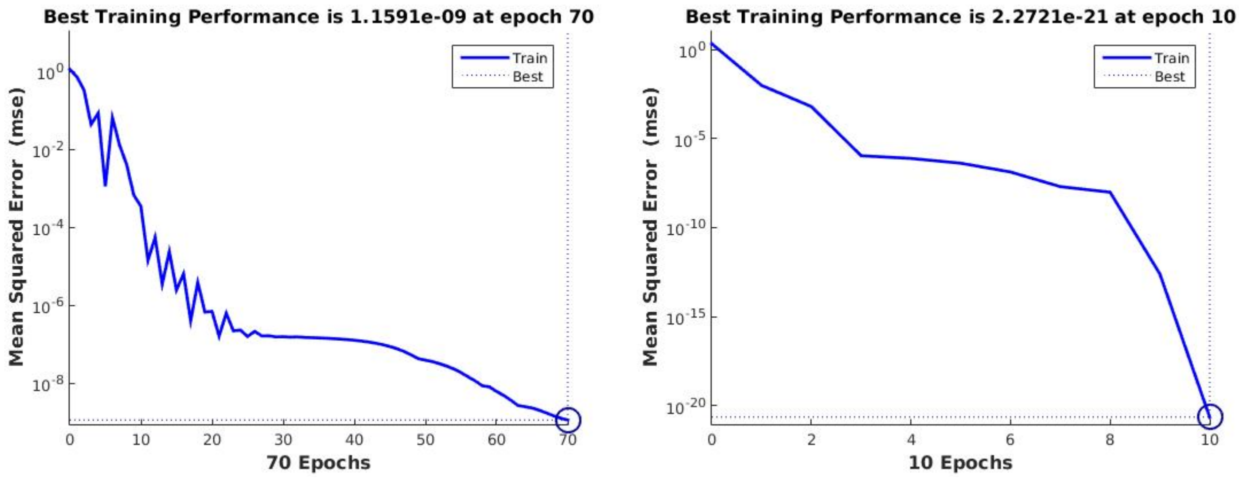


Figure IV.1: Variation of (MSE) values during the ANN training (Example 3)

This figure shows the learning process [57] to find the parameters of the model until the sum of the squares between $N(z)/D(z)$ and $Nr(z)/Dr(z)$ is minimized to $8.786 * 10^{-4}$. Learning is complete when examining additional observations does not reduce the error rate. The second-order reduced model was obtained using the ANN for discrete models method [18, 17]:

$$R_c(z) = \frac{0.52563z - 0.36939}{z^2 - 1.5025z + 0.65849} \tag{IV.5}$$

The final performance analysis and the comparison between the ANN -MOR techniques and the eigenvalue computation based method are given in the following table :

Table IV.3: Comparison of MSE for reduced order models (Example 3)

Methods	MSE	Kharitonov Theorem
Proposed method [57] (step resp)	2.629e-04	stable
Artificial neural network for discrete model [18, 17]	8.786e-04	stable

The following Figures show the comparison of Step / Frequency responses of reduced models and the higher order model

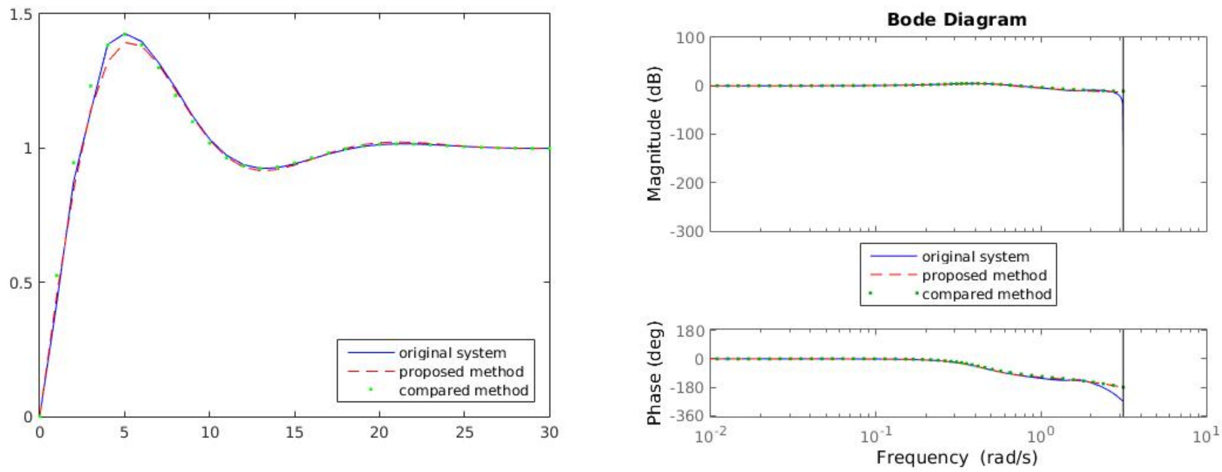


Figure IV.2: Step and frequency responses of original and reduced systems (Example 3)

Analysis

A comparison has been made between the results of the proposed model [57] and the results of mathematical methods proposed in [18, 17], from the figures (IV.4) and comparison table IV.3, we can notice that our method gives excellent results with systems that have fixed parameters and retain the characteristics of the original system ($mse = 2.629e - 04$) compared to the ANN proposed in [18, 17] (mse=8.786e-04) and does not require a long time to produce the results. [57]

IV.2.2 Example 4 (discrete time interval system)

Let us consider the3^{th} discrete interval system described by the transfer function [4]:

$$H(z) = \frac{[1,2]z^2 + [3,4]z + [8,10]}{[6,6]z^3 + [9,9.5]z^2 + [4.9,5]z + [0.8,0.85]} \tag{IV.6}$$

In this example, the numerator is reduced by using the ANN-P.A method [57] (case N°:01) in chapter III, and the denominator by using (case N°:02) in the same chapter. The obtained second-order model is given by:

$$R_2(z) = \frac{[2.421,3.266]z + [6.35,8.565]}{[9.449,9.949]z^2 + [4.891,4.991]z + [0.8,0.85]} \tag{IV.7}$$

As in the previous two examples. The following figures show the variation of the MSE during the training operation:

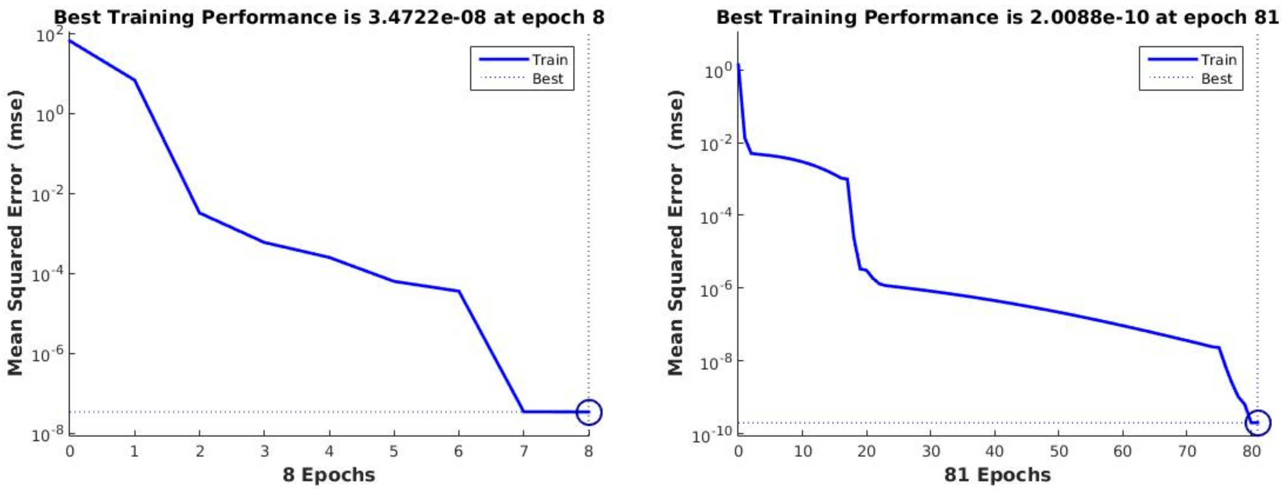


Figure IV.3: Variation of MSE during the ANN training (Example 4).

These two figures show the learning process to find the parameters of the model until the sum of the squares between $D(sz), N(z)$ and $Dr(z), Nr(z)$ is minimized to $3.4722 * 10^{-8}$,$2.0088 * 10^{-10}$ respectively. Learning is complete when examining additional observations does not reduce the error rate. [57]

The2^{nd} order reduced interval model obtained using the modified γ - δ routh approximation method

[4] is:

$$R_c(z) = \frac{[1.07, 0.98]z + [4.5, 6.55]}{[7.44, 10.19]z^2 + [4.5, 7.58]z + [0.15, 2.6]} \quad \text{(IV.8)}$$

The final performance analysis and the comparison between the ANN -MOR [57] technique and the modified γ - δ routh approximation method [4] are given in the following table :

Table IV.4: Comparison of MSE for reduced order models (Example 4).

Methods	Lower limit	Upper limit	Kharitonov Theorem
Proposed method [57] (step resp)	0.08275	0.20836	stable
modified γ - δ routh approximation method [4]	0.17838	0.34271	stable

The following Figures show the comparison between Step/Frequency responses of reduced models and the higher order model :

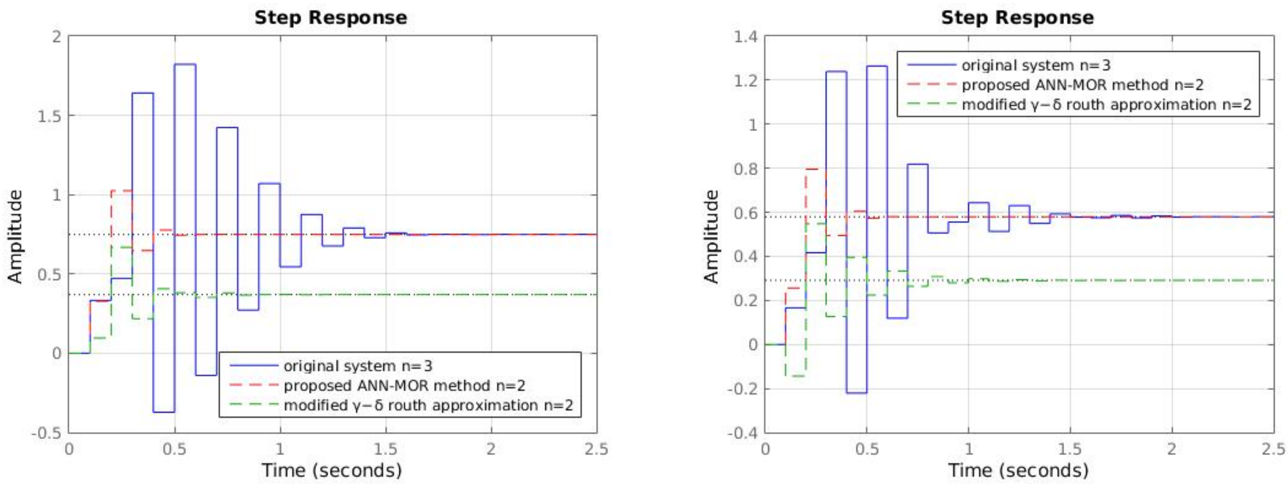


Figure IV.4: Step responses of original and reduced systems (Example 4).

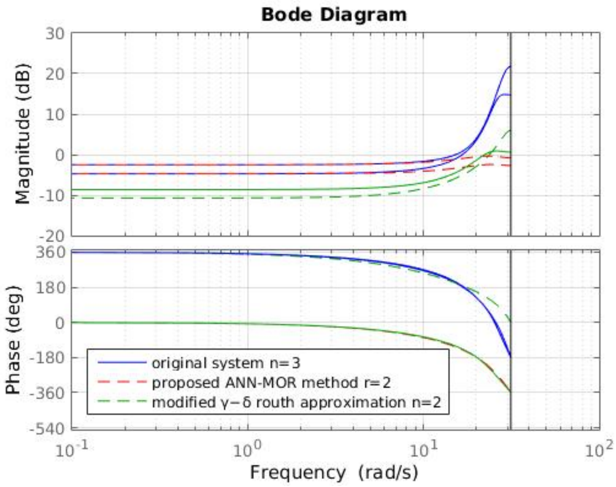


Figure IV.5: Frequency responses of original and reduced systems (Example 4).

Analysis

In the 3rd example , the characteristics of the full order model have been preserved, seen as $mse = [0.0827, 020836]$, while the responses are very close to the full 3rd order model. This can be observed as seen in figures (IV.8 and IV.9) and comparison table IV.5, it can also be noticed that the proposed method does not require much time to produce the results. [57]

IV.2.3 Example 5 (MIMO continuous time interval system)

Let us consider the3^{rd}order two-input/two-output interval system described by the transfer function [65] :

$$G(s) = \frac{1}{D(s)} \begin{bmatrix} [1,2]s^2 + [4,7]s + [11,15] & [1,2]s^2 + [5,7.5]s + [12,15.5] \\ [1,1.5]s^2 + [5.5,7]s + [12.5,15.5] & [1,2.5]s^2 + [5,9.5]s + [12,17] \end{bmatrix} \tag{IV.9}$$

Where $D(s) = [6,6]s^3 + [27,27.5]s^2 + [40.9,42]s + [20.7,21.35]$ In this example, the numerator is reduced using the gain correction factor η as described in the chapter III (step 4), and the denominator is reduced using case n=1 in the same section. The obtained first-order model is:

$$R_3(s) = \frac{\begin{bmatrix} [11,15] & [12.5,15.5] \\ [12,15.5] & [12,17] \end{bmatrix}}{[41.03,42.13]s+[20.7,21.35]} \qquad \text{(IV.10)}$$

As in the previous example, the following figure show the variation of the MSE during the training operation. [57]

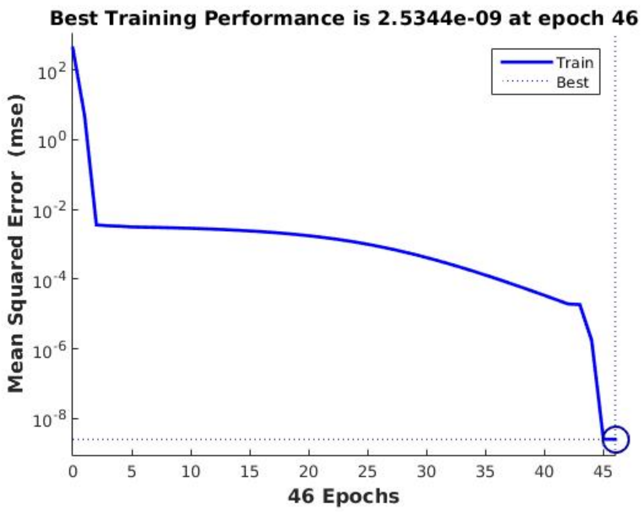


Figure IV.6: Variation of MSE values during the ANN training (Example 5).

This figure shows the learning process [57] to find the parameters of the model until the sum of the squares between $D(s), N(s)$ and $D_r(s), N_r(s)$ is minimized to ,$2.5344 * 10^{-9}$ respectively. Learning is complete when examining additional observations does not reduce the error rate. The 1^{st} order reduced interval model obtained using the gain adjustment algorithm presented in [65] :

$$R_c(s) = \frac{\begin{bmatrix} [0.7748,1.27] & [0.8917,1.368] \\ [0.8529,1.319] & [0.8529,1.417] \end{bmatrix}}{[1,1]s+[1.268,1.534]} \qquad \text{(IV.11)}$$

The final performance analysis and the comparison between the (ANN -MOR) technique and the gain adjustment algorithm [65] are given in the following table :

Table IV.5: Comparison of MSE for reduced order models (Example 5).

Methods	Lower limit	Upper limit	Kharitonov Theorem
Proposed method [57] (ANN -MOR)	$1.0e-03*$ $\begin{bmatrix} 0.45564 & 0.71317 \\ 0.61609 & 0.61609 \end{bmatrix}$	$\begin{bmatrix} 0.00089 & 0.001 \\ 0.00102 & 0.00152 \end{bmatrix}$	stable
modified γ - δrouth approximation method [65]	$\begin{bmatrix} 0.01162 & 0.01593 \\ 0.01443 & 0.01443 \end{bmatrix}$	$\begin{bmatrix} 0.02655 & 0.04070 \\ 0.02928 & 0.02737 \end{bmatrix}$	stable

The following Figures show the comparison between Step/Frequency responses of reduced models and the higher order model .

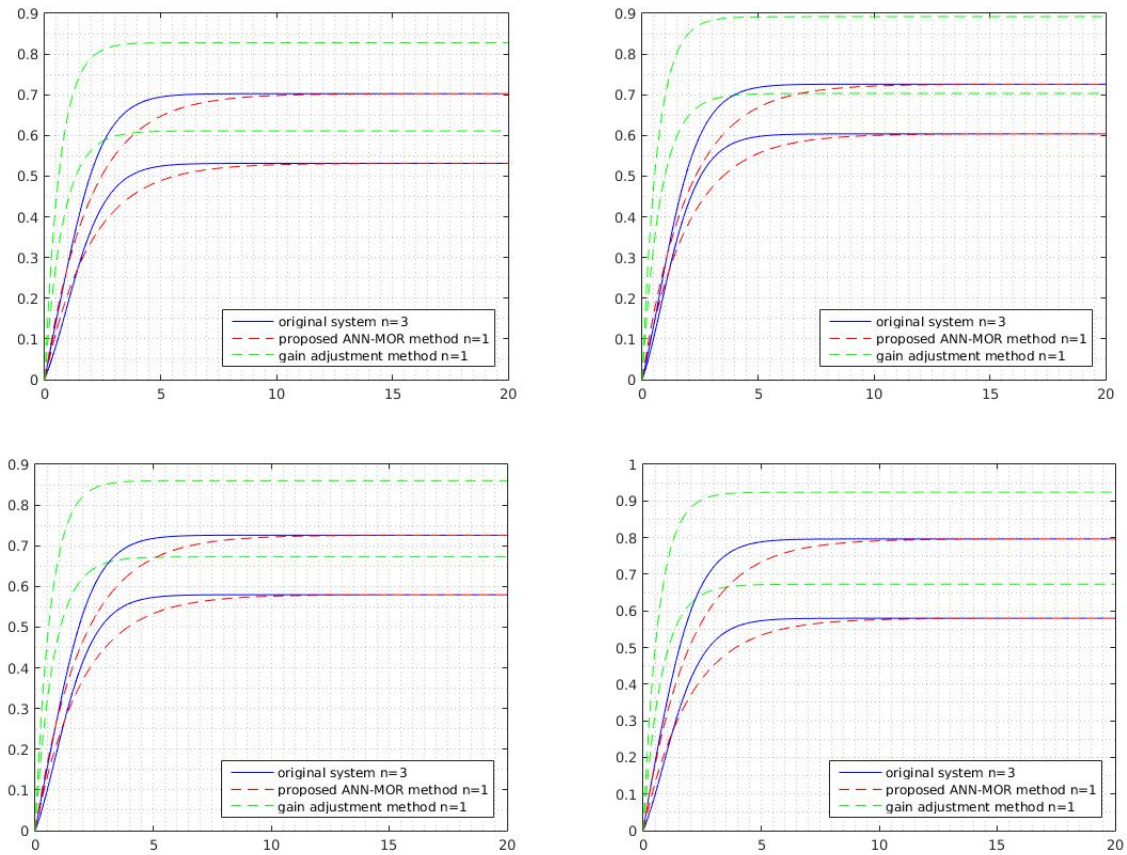


Figure IV.7: Step responses of original and reduced systems (Example 5).

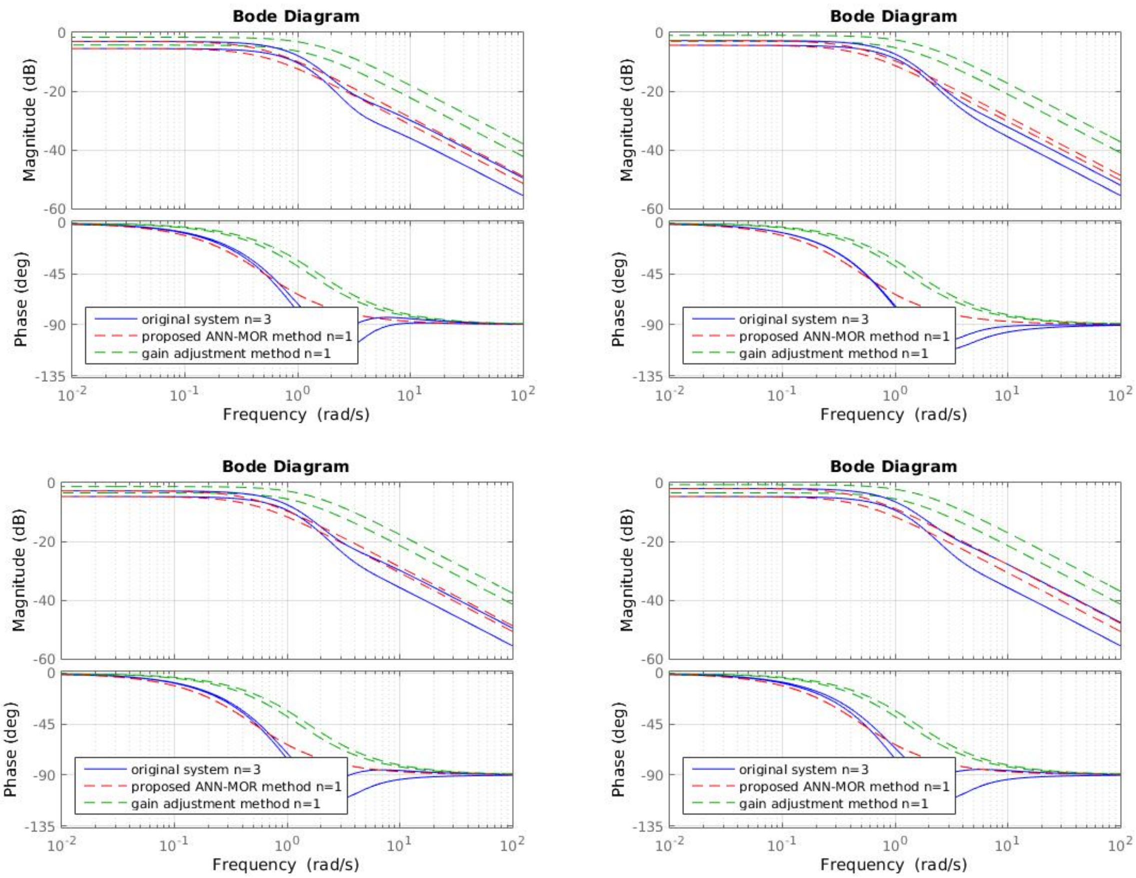


Figure IV.8: frequency responsesof original and reduced systems (Example 5).

Analysis

From figures (IV.11 and IV.12) and comparison table IV.6, it can be seen that our proposed method [57] works best with continuous time interval systems and retains the characteristics of the original system, and does not require much time to produce the results.

IV.2.4 Example 6 (continuous time interval system)

Let us consider the all pole 3^{th} interval system described by the transfer function [57] :

$$H(s) = \frac{s^2 + [3.3, 6.5]s + [2.7, 10]}{s^3 + [8.5, 8.6]s^2 + [18.5, 18.2]s + [10.25, 10.76]} \quad \text{(IV.12)}$$

In this example, the numerator is reduced using the ANN-P.A [57] (case n=1) in the chapter III , and the denominator using (case n=2) in the same section. The obtained second-order model is given by:

$$R_1(s) = \frac{[3.315, 6.519]s + [2.7, 10]}{[8.53, 8.571]s^2 + [18, 18.2]s + [10.25, 10.76]} \tag{IV.13}$$

The following figure shows the variation of MSE during the training operation, validation, and test performance of the ANN training record. This final result shows the best training performance and the number of epochs of the training.

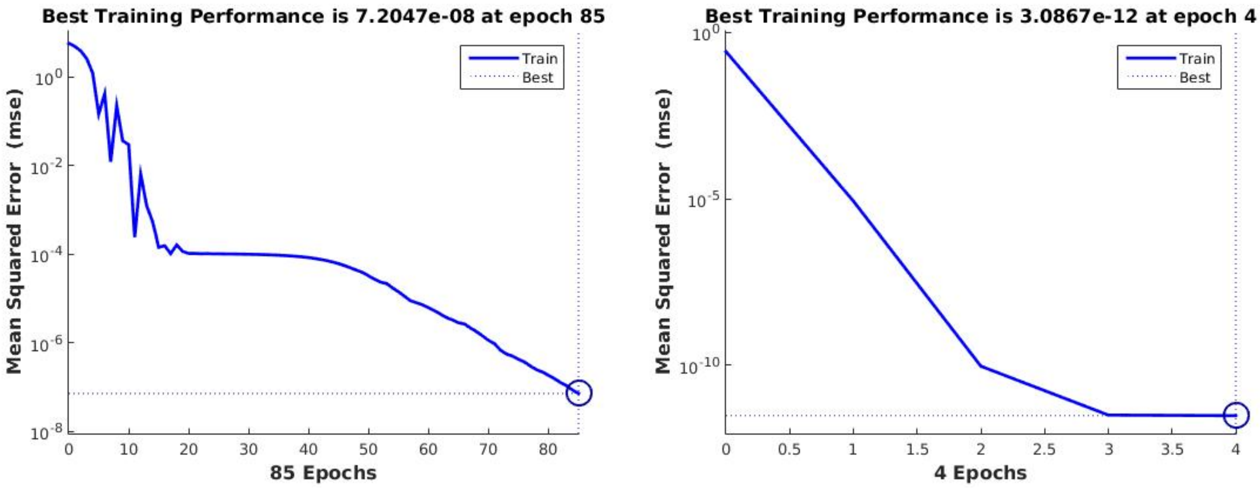


Figure IV.9: Variation of MSE during the ANN training (Example 6).

This figure shows the learning process to find the parameters of the model until the sum of the squares between $D(s), N(s)$ and $D_r(s), N_r(s)$ is minimized to ,$2.5344 * 10^{-9}$, respectively. Learning [57] is complete when examining additional observations does not reduce the error rate. The 2^{nd} order reduced interval model was obtained using the balanced technique presented in [57] :

$$R_c(s) = \frac{s + [1.358, 6.471]}{s^2 + [6.568, 8.592]s + [5.15, 6.869]} \tag{IV.14}$$

The final performance analysis and the comparison between the ANN -MOR techniques and the balanced technique [57] based method are given in the following table:

Table IV.6: Comparison of MSE for reduced order models (Example 6).

Methods	Lower limit	Upper limit	Kharitonov Theorem
Proposed method [57] (ANN -MOR)	0.1762e-07	0.2372e-07	stable
Balanced technique	0.0005e-07	0.1553e-05	stable

The following Figures show the comparison of Step / Frequency responses of reduced models and the higher order model

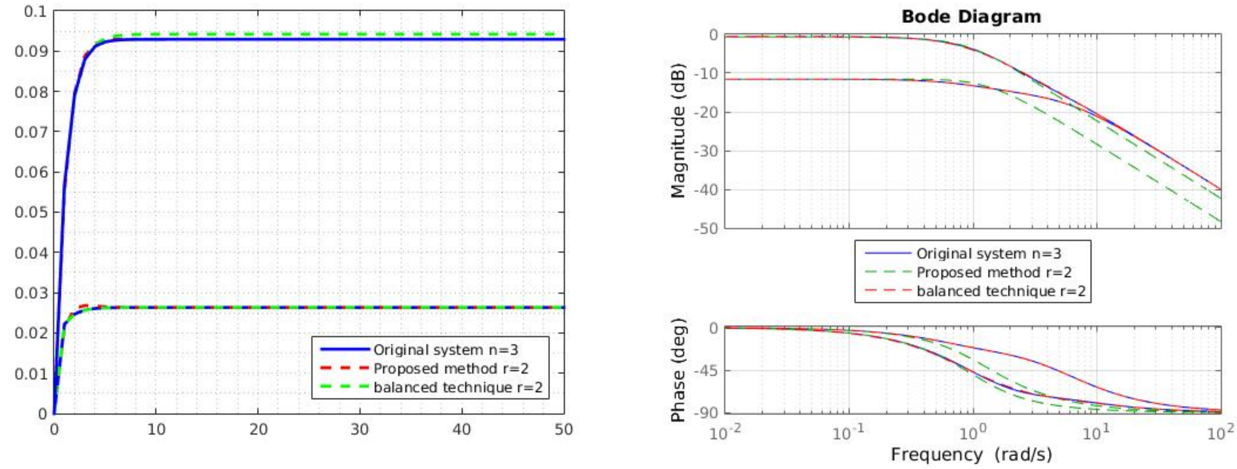


Figure IV.10: Step and frequency responses of original and reduced systems (Example 6).

IV.2.5 Results and discussion

As it was shown, all time and frequency responses show the upper and lower bound fixed / interval systems. According to the simulation results on the five examples, we notice that the behavior of the reduced order models calculated via the ANN-MOR [57] machine learning algorithm follows the full order model faithfully, whether, in the fixed first example continuous Time interval system in example 2, Discrete-time interval system in example 3, continuous / Time MIMO System (example 4), or in Continuous Time / interval systems example 5, (see Fig. IV.14), the application of the new algorithm show the superiority, in the time and frequency domains. Concerning the MIMO model, time and frequency responses are traced. Same plots and performances are obtained for the other channels. The deviation of the reduced order system from the higher-order original system response is given by the Mean Square Error (MSE) criterion, a robust tool [64], given under its five expressions as well. The superiority of our proposed method over others from cited works is clearly shown in the MSE tables. The MSE is calculated either for impulse and step responses, this permits us to enhance the superiority of our reduced order models compared to others resulting from recent work. Stability is always guaranteed. [57]

IV.2.6 Interpretation

Our proposed algorithm is computationally simple and maintains stability using a modern algorithm inspired by biological AI, which is ANN. from the figures and comparison tables it can be noticed that our proposed method [57] works better with all types of systems and retains the characteristics of the original system and does not require a long time to produce the results. In addition, the strategy of learning under the supervision of the initial system and building a new system, MOR, is explained in a very simplified manner and operates with all computing platforms, which is what makes the proposed method has all the good advantages of a modern integrated algorithm that solves a problem in a simple and easy way that we can rely on in our daily scientific life, especially in the areas of simplification and approximation.[57]

IV.3 Conclusion

This chapter presents an analytical study of the proposed model reduction based on two advanced methods [56, 57], projection and artificial neural networks; it is designed to reduce the model order; the algorithm's goal is to minimize the error between the original and reduced interval systems. This analytical study compares the application of the algorithm to systems previously presented in the literature and reduces them by the previous algorithm. The implementation of this method guarantees stability and provides better performance tuning. It applies to both generalized interval systems (continuous time, discrete time) and multiple input/multiple output interval systems, and fixed-point interval systems.

General conclusion

The uncertainty in the parameters of a system and their variation within bounds, which are called interval systems is discussed in this thesis, interval systems, is a set of coefficients especially expressed by intervals of values specified with two bounds [a, b] operating the basic notions of interval arithmetics [1], with all these complications added to the usual systems the objective of control engineering becomes more complicated, since then, this thesis deals with the reduction of this kind of systems to obtain a model of reduced dimension, the two presented and discussed methods [56, 57] in the previous part of this thesis are interpreted and reviewed by two international journals and published after that, the first one is an extension of the important tool of Singular Value Decomposition (SVD) technique[57] proposed for dynamic interval systems model reduction. This method occurs in state space, and the reduced model is built by discarding the state of the smallest energy that contributes weakly to the overall behavior of the original systems. The reduced model is generated according to the remaining Hankel singular values. To appreciate its performance, it is compared to other reduced-order models cited in the literature.

The second method is a new idea of an advanced machine learning algorithm for reducing a large-scale interval system based on ANN. This idea is beneficial for interval systems, fixed-parameter systems, and simple polynomials. The numerator and the denominator are reduced using a new ANN polynomial approximation method (ANN -PA)[56] by using two neurons Placed in one hidden layer. The pinpoint of this procedure is the input vector format mode, in which every component of this vector is connected to the primary entry element. Our proposed algorithm is valid in generalized interval systems in their different forms: continuous-time and discrete-time. Furthermore, it is extended to Multi-input/multi-output interval and fixed-point interval systems. Therefore, it ensures Stability and better performance matching than the already existing methods. Various applications obtain good results that are always superior to what has been achieved in previous studies in the literature.

Bibliography

[1] M. J. C. Ramon E. Moore, R. Baker Kearfott, *Interval Analysis*. 2009.

[2] M. K. Salah, *Machine learning for model order reduction*. 2018.

[3] W. Schilders, "Introduction to Model Order Reduction," pp. 3–32, 2008.

[4] M. K. Sharma and D. Kumar, "Modified gamma-delta Routh approximation method for order reduction of discrete interval systems," *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015*, 2015.

[5] C. B. Vishwakarma and R. Prasad, "Clustering method for reducing order of linear system using pade approximation," *IETE Journal of Research*, vol. 54, no. 5, pp. 326–330, 2008.

[6] A. Jaiswal, P. K. Singh, S. Gangwar, S. Manmatharajan, and D. Kumar, "Order reduction of interval systems using Eigen spectrum and factor division algorithm," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 3, no. PART 1, pp. 363–367, 2014.

[7] N. Gupta and A. Narain, "Reduction of discrete interval systems through fuzzy-C means clustering with dominant pole retention," *2015 Australian Control Conference, AUCC 2015*, vol. 1, pp. 348–353, 2015.

[8] B. C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.

[9] M. G. Safonov and R. Y. Chiang, "A Schur Method for Balanced-Truncation Model Reduction," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 729–733, 1989.

[10] A. B. H. Adamou Mitiche and L. Mitiche, "A very low speech model based on frequency selection-GA approach," *International Journal of Intelligent Engineering Informatics*, vol. 6, no. 3/4, p. 343, 2018.

[11] A. B. H. Adamou-Mitiche and L. Mitiche, "Comparative study of model reduction schemes-application to the digital filters synthesis," *6th International Symposium on Signal Processing and Its Applications, ISSPA 2001 - Proceedings; 6 Tutorials in Communications, Image Processing and Signal Analysis*, vol. 2, pp. 675–678, 2001.

[12] A. B. Adamou-Mitiche and L. Mitiche, "Extended approaches based on Genetic Algorithms (GAs) for MIMO singular systems approximation," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 1, pp. 148–154, 2018.

[13] K. Glover, "All optimal Hankel-norm approximations of linear," *International Journal of Control*, vol. 39, no. 6, pp. 1115–1193, 1984.

[14] M. A. Hitz and E. Kaltofen, "The Kharitonov theorem and its applications in symbolic mathematical computation," *Workshop on Symbolic-Numeric Algebra for Polynomials*, pp. pp. 20—-21, 1997.

[15] R. K. Gautam, N. Singh, N. K. Choudhary, and A. Narain, "Model order reduction using factor division algorithm and fuzzy c-means clustering technique," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 2, pp. 468–475, 2019.

[16] L. Wu, X. Su, P. Shi, and J. Qiu, "Model approximation for discrete-time state-delay systems in the TS fuzzy framework," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 366–378, 2011.

[17] O. M. Alsmadi, Z. S. Abo-Hammour, and A. M. Al-Smadi, "Artificial neural network for discrete model order reduction with substructure preservation," *Applied Mathematical Modelling*, vol. 35, no. 9, pp. 4620–4629, 2011.

[18] A. Adel and K. Salah, "Model order reduction using artificial neural networks," *2016 IEEE International Conference on Electronics, Circuits and Systems, ICECS 2016*, pp. 89–92, 2017.

[19] O. M. Alsmadi, Z. S. Abo-Hammour, and A. M. Al-Smadi, "Artificial neural network for discrete model order reduction with frequency selectivity," *World Scientific Proc. Series on Computer Engineering and Information Science 7; Uncertainty Modeling in Knowledge Engineering and Decision Making - Proceedings of the 10th International FLINS Conf.*, vol. 7, pp. 1011–1016, 2012.

[20] N. Iulian and E. Koskivaara, "A neural network model for prediction : Architecture and training analysis," Tech. Rep. 521, 2003.

[21] K. S. Narendra, "Neural networks and dynamical systems," *International Journal of Approximate Reasoning*, vol. 6, no. 2, pp. 109–131, 1992.

[22] S. K. Jeswal and S. Chakraverty, "ANN Based Solution of Uncertain Linear Systems of Equations," *Neural Processing Letters*, vol. 51, no. 2, pp. 1957–1971, 2020.

[23] S. Workshop and I. Methods, *Summer Workshop on Interval Methods*. 2016.

[24] Y. Hijazi, H. Hagen, C. Hansen, and K. I. Joy, "Why interval arithmetic is so useful," *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, pp. 148–163, 2008.

[25] R. Moore, "Computational functional analysis, (ellis horwood series in mathematics and its applications) by re moore; mj cloud,"

[26] M. WARMUS, "Instytut matematyczny polskiej akademii nauk,"

[27] U. W. Kulisch and W. L. Miranker, "The arithmetic of the digital computer: A new approach," *Siam Review*, vol. 28, no. 1, pp. 1–40, 1986.

[28] E. R. Hansen, "A generalized interval arithmetic," in *International Symposium on Interval Mathematics*, pp. 7–18, Springer, 1975.

[29] R. E. Moore, "Interval analysis: Differential equations.," 2009.

[30] B. Bandyopadhyay, V. Sreeram, and P. Shingare, "Stable gamma delta Routh approximation for interval systems using Kharitonov polynomials," *International Journal of Information and Systems Sciences*, vol. 4, no. 3, pp. 348–361, 2008.

[31] J. Rohn, "Versoft: Guide," tech. rep., Technical report, Technical Report, 2011.

[32] S. M. Rump, "Intlab interval laboratory," in *Developments in reliable computing*, pp. 77–104, Springer, 1999.

[33] "Institute for reliable computing." `http://www.ti3.tu-harburg.de/rump/intlab/index.html`. Accessed: 2022.

[34] N. Revol, "Introduction to the IEEE 1788-2015 Standard for Interval Arithmetic Nathalie," *Springer International Publishing AG 2017*, pp. 1–14, 2017.

[35] S. Oishi, "Slab-a matlab-like numerical tool which can do result verification," in *Proceedings of the Annual Meeting of the Japan Society for Industrial and Applied Mathematics*, pp. 25–25, Japan Society for Industrial and Applied Mathematics, 2002.

[36] R. Klatte, U. Kulisch, A. Wiethoff, and M. Rauch, *C-XSC: A C++ class library for extended scientific computing*. Springer Science & Business Media, 2012.

[37] M. Lerch, G. Tischler, J. W. V. Gudenberg, W. Hofschuster, and W. Krämer, "Filib++, a fast interval library supporting containment computations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 32, no. 2, pp. 299–324, 2006.

[38] N. Revol, "The mpfi library: Towards ieee 1788–2015 compliance," in *International Conference on Parallel Processing and Applied Mathematics*, pp. 353–363, Springer, 2019.

[39] J. Hunter and D. Dale, "The matplotlib users guide," *Matplotlib 0.90. 0 users guide*, 2007.

[40] A. K. Choudhary and S. K. Nagar, "Model order reduction of discrete-time interval system based on Mikhailov stability criterion," *International Journal of Dynamics and Control*, vol. 6, no. 4, pp. 1558–1566, 2018.

[41] N. Kranthi Kumar and J. P. Tiwari., "A New Algorithm for Model Order Reduction of Interval Systems," *Bonfring International Journal of Data Mining*, vol. 3, no. 1, pp. 06–11, 2013.

[42] S. Ubbiali, *Reduced order modeling of nonlinear problems using neural networks*. PhD thesis, 2017.

[43] K. Amoura, P. Wira, and S. Djennoune, "A state-space neural network for modeling dynamical nonlinear systems," *NCTA 2011 - Proceedings of the International Conference on Neural Computation Theory and Applications*, no. May 2016, pp. 369–376, 2011.

[44] J. Hahn and T. F. Edgar, "An improved method for nonlinear model reduction using balancing of empirical gramians," *Computers and Chemical Engineering*, vol. 26, no. 10, pp. 1379–1397, 2002.

[45] D. Parashar, M. K. Sharma, and A. Chandel, "Model reduction of discrete interval systems by Routh approximation method," *2015 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2015*, no. December, pp. 290–293, 2016.

[46] D. K. Kumar and S. K. Nagar, "Mixed method for reducing the order of a linear discrete time interval system," *International Conference on Advances in Computing, Communication and Information Technology (CCIT-2014)*, no. June, 2014.

[47] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Applied numerical mathematics*, vol. 43, no. 1-2, pp. 9–44, 2002.

[48] H. Xu, "An svd-like matrix decomposition and its applications," *Linear algebra and its applications*, vol. 368, pp. 1–24, 2003.

[49] J. E. Dayhoff, *Neural network architectures: an introduction*. Van Nostrand Reinhold Co., 1990.

[50] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the theory of neural computation*. CRC Press, 2018.

[51] R. D. Pal, M. Kumar, and D. Kumar, "Schur balanced approach for frequency weighted model reduction," *SCES 2014: Inspiring Engineering and Systems for Global Sustainability*, no. 5, pp. 0–4, 2014.

[52] Y. Dolgin and E. Zeheb, "On routh-pade model reduction of interval systems," *IEEE transactions on automatic control*, vol. 48, no. 9, pp. 1610–1612, 2003.

[53] D. K. Saini and R. Prasad, “Order reduction of linear interval systems using genetic algorithm,” *International Journal of Engineering and Technology*, vol. 2, no. 5, pp. 316–319, 2010.

[54] A. B. H. Adamou-Mitiche and L. Mitiche, “Multivariable Systems Model Reduction Based on the Dominant Modes and Genetic Algorithm,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1617–1619, 2017.

[55] O. M. Alsmadi and Z. S. Abo-Hammour, “A robust computational technique for model order reduction of two-time-scale discrete systems via genetic algorithms,” *Computational Intelligence and Neuroscience*, vol. 2015, 2015.

[56] R. Zerrougui, A. B. H. Adamou-Mitiche, and L. Mitiche, “A novel machine learning algorithm for interval systems approximation based on artificial neural network.,” *Journal of Intelligent Manufacturing*, vol. feb 5, pp. 1–14, 2022.

[57] R. Zerrougui, A. B. H. Adamou-Mitiche, and L. Mitiche, “Projection approach for interval systems approximation: An extension to mimo systems,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 499–503, 2022.

[58] K. Ganesan, “On some properties of interval matrices,” *a a*, vol. 1, p. 2, 2007.

[59] A. B. H. Adamou-Mitiche and L. Mitiche, “Energetic parameters in approximation of discrete systems: a mimo extension problem,” *International Journal of Operational Research*, vol. 27, no. 1-2, pp. 291–302, 2016.

[60] A. J. Laub, “A Schur Method for Solving Algebraic Riccati Equations,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 913–921, 1979.

[61] J. Siam, “Calculating the singular values and,” *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[62] N. Selvaganesan, “Mixed method of model reduction for uncertain systems,” *Serbian Journal of Electrical Engineering*, vol. 4, no. 1, pp. 1–12, 2007.

[63] G. Saraswathi, K. A. Gopala Rao, and J. Amarnath, “A mixed method for order reduction of interval systems,” *2007 International Conference on Intelligent and Advanced Systems, ICIAS 2007*, pp. 1042–1046, 2007.

[64] A. B. Adamou-Mitiche and L. Mitiche, “Internal projections and l 2 norm for optimal digital filters model reduction: a comparative study,” *Journal of Circuits, Systems and Computers*, vol. 28, no. 01, p. 1930001, 2019.

[65] S. P. Aditya Prasad Padhy, V. P. Singh, “Model reduction of multi-input-multi-output discrete interval systems using gain adjustment,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. Special Issue, pp. 12721–12739, 2018.