CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE CONTROL AUTOMÁTICO

Neural identification of 3D distributed parameter systems

Tesis que presenta

Víctor Hernández Santamaría

Para Obtener el Grado de

Maestro en Ciencias

En la Especialidad de

Control Automático

Directores de Tesis: Dr. Alexander Pozniak Gorbatch y Dr. Jorge Isaac Chairez Oria

México, D.F.

Junio, 2012

Contents

Abstract

The mathematical formulation of many problems in science and engineering can be reduced to a set of partial differential equations (PDE). However there is no general theory known concerning the solvability of all partial differential equations. Such theory is unlikely to exist, given the rich variety of physical and geometric phenomena which can be modeled by PDE. Since neural networks have universal approximation capabilities, therefore it is possible to postulate them as approximate solutions for given differential equations. In this thesis, a differential neural network approach for non-parametric identification of a class of three dimensional (3D) PDE is proposed. Learning laws are derived and practical stability of the identification error is demonstrated via Lyapunov-like analysis. To illustrate the qualitative behavior and efficiency of the suggested methodology, simulation results are presented.

Acknowledgments

- To the Council of Science and Technology, CONACYT, for the scholarship granted during my graduate studies.
- To my mentors, Dr. Alex Poznyak and Dr. Isaac Chairez, for their invaluable help and guidance.
- To my parents, Victor and Gloria, for their love and unconditional support.

Chapter 1

Introduction

1.1 Motivation

In science and engineering, there exists many applications of systems that can be described by partial differential equations (PDE). Due to its importance, many methods have been proposed in the literature for their solution, such as finite difference methods (FDM) [29], [35], [38], finite element methods (FEM) [18], [21], [37], Runge–Kutta methods [42], Splines [9], [20] and predictor–corrector methods [8]. These methods require the discretization of domain into a number of finite elements where the solutions are approximated locally. Although these methods provide a good approximation of the solution, it is required a complete knowledge of the system to discretize the domain via meshing, which can be a challenge in two or more dimension problems. Also, the approximate solution derivatives are discontinuous and can affect the stability of the solution. Furthermore, in order to obtain an accurate solution, it may be necessary to deal with finite meshes, which significantly increase the computational cost. Finally, these methods are well defined for linear systems with well–known structure.

A thoroughly research, shows that other methods can approximate solutions of PDEs. Recent results demonstrate that neural networks techniques can be used effectively to identify a wide class of nonlinear systems, even when the system model is completely unknown. Lagaris and Likas [23] presented a method for solving initial and boundary problems using artificial neural network. They use the collocation method to compute the solution, which assumes the discretization of the domain. They choose a trial function such that by construction satisfies the given boundary conditions. This is obtained by proposing the trial function as a sum of two parts, one with no adjustable parameters that satisfy the boundary conditions and a second term that employs a neural network whose weights are adjusted to deal with a minimization problem. Then network is trained to satisfy the differential equation. Using the same approach, Lagaris et al. in [24] studied partial differential equations where the boundary can be any arbitrary complex geometrical shape. Collocation method is used again and the problem is transformed into a unconstrained optimization problem. In [17], the authors presented a method to solve a a class of first order PDE which appears in input-to-state linearized control systems. They proposed a backpropagation algorithm for training a feedforward neural network and approximate a solution, which was used to design feedback control laws to regulate a class of nonlinear systems. Some useful applications can be found on literature; for example, in [2], the authors developed a multilayer perceptron (MLP) technique to solve a mathematical model of vibration control of flexible mechanical systems. Nevertheless, due to non-linearity and complex boundary conditions, their numerical solutions present major drawbacks like numerical instability. Hybrid methods also can be found in recent literature. Smaoui et al. in [34] analyzed the dynamics of two non-linear partial differential equations known as the Kuramato-Sivashinsky equations and the two dimensional Naiver-Stokes equations using the combination of Karhunen-Loeve decomposition and artificial neural network. In [3], a novel method based on artificial neural networks, minimization techniques and collocation method is presented. It provides an approximate solution to time dependent systems of partial differential equations. In article [39], Tsoulos et al. used a hybrid method utilizing feedforward neural networks (FFNN) by grammatical evolution and a local optimization procedure, in order to solve ordinary and partial differential equations. They used the well stablished evolution technique [27], [40] to evolve the neural network topology along with the network parameters. A different technique, based on radial basis function neural networks (RBFNN) for the resolution of nonlinear Schrodinger equation in hydrogen atom, can be found on [33].

The differential neural network (DNN) approach [30] is a useful tool for the analysis of a variety of problems related to control theory, such as identification, estimation and trajectory tracking. Moreover, these networks have adequate performance in the presence of uncertainty and/or unmodeled dynamics, because its structure incorporates feedback. Therefore, the learning process is reduced to an appropriate design of feedback. A previous related work [5] presents a method developed for the approximation of solutions of a class of partial differential equations in two dimensions. This method proposes the application of differential neural networks for approximation of solutions of partial differential equations with uncertainty. The suggested method proposes the discretization of the domain of PDE by a mesh and use a finite difference method. The solution is then approximated at each node of the mesh using a continuous DNN.

1.2 Results

In this thesis, we present an extended method for approximating the solution of a class of partial differential equations in three dimensions where, in addition, the available information to approximate the solution of the differential equation is restricted. We consider three types of restrictions. First, constraints associated with a dynamic model whose outputs (measured along the entire mesh) are sampled data. These sample–data outputs can be understood as the result of a quantization process applied to a continuous output signal. Then, constraints associated directly to the mesh that divides the domain of the partial differential equation, specifically, an irregular mesh where the available information to approximate the solution is measurable only in some nodes. Finally, we present a case where the two previous constraints are considered simultaneously. As it is shown in [30], a Lyapunov-like method can be a good instrument to generate learning laws and establish error stability conditions. To deal with cases where outputs provide sampled–data, we use some advanced Lyapunov techniques related to descriptor method. We refer the reader to [12], [13], [14] for the corresponding details. In cases where the mesh is irregular, interpolation methods are used to approximate the missing information on the mesh. The method proposed in this work can be used to approximate solutions of partial differential equations in three dimensions, which represent a variety of systems with distributed parameters, such as the three dimensional heat and wave equation.

1.3 Organization

This thesis is organized as follows. In the second section we make a brief discussion about systems with distributed parameters and present the necessary tools to study them in the context of differential neural networks. In the third section, we outline the main problems and present the results of this work. In Section 4, various simulations are presented to illustrate the effectiveness of the proposed methodology and, finally, we present a small comparative table which highlights the main features of various methods for approximating solutions of partial differential equations. In Section 5, a brief conclusion summarizes the objectives achieved. Additionally, in the appendix section, the reader will find the proofs to the main theorems and notation that will facilitate the reading of this work.

Chapter 2

Distributed parameter systems

2.1 Distributed parameter systems and its approximation

As background for the development of non-parametric identifier for approximating the solution of partial differential equations, we must be able to represent them in a form suitable for study within the context of DNN. For this, consider the following set of uncertain partial differential equations

$$
\begin{aligned}
u_t(x,y,z,t) = f\,(& u(x,y,z,t), u_x(x,y,z,t), u_{xx}(x,y,z,t), u_y(x,y,z,t), \\
& u_{yy}(x,y,z,t), u_z(x,y,z,t), u_{zz}(x,y,z,t), u_{xy}(x,y,z,t), \\
& u_{yx}(x,y,z,t), u_{xz}(x,y,z,t), u_{yz}(x,y,z,t)) + \xi(x,y,z,t)
\end{aligned} \tag{2.1}
$$

where $u \in \Re^n$ is defined in the domain

$$G = [0,\ 1]^3 \times [0,\infty) \tag{2.2}$$

this is, $x \in [0,1]$, $y \in [0,1]$, $z \in [0,1]$ and $t \in [0,\infty)$. The boundary (Neumann and Dirichlet) and initial conditions are given by

$$
\begin{aligned}
& u(0,y,z,t) = u_{10} \in \Re^n, \quad u(x,0,z,t) = u_{20} \in \Re^n, \quad u(x,y,0,t) = u_{30} \in \Re^n \\
& u_x(0,y,z,t) = 0 \in \Re^n, \quad u_y(x,0,z,t) = 0 \in \Re^n, \quad u_z(x,y,0,t) = 0 \in \Re^n, \\
& u(x,y,z,0) = c \in \Re^n
\end{aligned} \tag{2.3}
$$

We may use the following notation throughout this document

$$
\begin{aligned}
&u_t(x,y,z,t) = \frac{\partial u(x,y,z,t)}{\partial t}\\
&u_x(x,y,z,t) = \frac{\partial u(x,y,z,t)}{\partial x}, \quad u_{xx}(x,y,z,t) = \frac{\partial^2 u(x,y,z,t)}{\partial x^2}\\
&u_{xy}(x,y,z,t) = \frac{\partial^2 u(x,y,z,t)}{\partial x \partial y}, \quad u_{yx}(x,y,z,t) = \frac{\partial^2 u(x,y,z,t)}{\partial y \partial x}
\end{aligned}
$$

and it follows for all other terms included in the function $f(\cdot)$. It is necessary, when considering the solution of partial differential equations, to introduce the concept of existence and uniqueness. The Cauchy-Kovalevskaya theorem (see, for example, [10], [11]) is basically the only general existence theorem in the subject, and thus should perhaps be regarded as central. This theorem applies to equations of a very general form

$$
\frac{\partial^{n_i} u_i}{\partial t^{n_i}} = f_i\left(t, x, u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x} \ldots\right), \quad i = 1, \ldots, m. \tag{2.4}
$$

where $x = (x_1, \ldots, x_{n-1})$, $u = (u_1, \ldots, u_m)$, and for each $i = 1, \ldots, m$ the function f_i depends on the derivatives of the functions u_j only up to order n_j, is independent of $\partial^{n_j} u_j / \partial t^{n_j}$, and is an analytic function of all its arguments, which covers a great variety of systems. Notwithstanding, in practice, the power series methods proposed by the Cauchy-Kovalevskaya theorem are not so prevalent. Furthermore, the theory for nonlinear partial differential equations is far less unified in its approach, as the various types of nonlinearity must be treated in quite different ways. Some authors (see, for example [11]) rely on functional analysis and "energy" estimates to prove the existence of weak solutions. In a broad sense, a weak solution u is a function which is not continuously differentiable or even continuous, but which is nonetheless deemed to satisfy the equation in some precisely defined sense. Such solutions are very useful because a lot of natural phenomena modeled by partial differential equations (such as those modeled by the equation $u_t + F(u)_x = 0$, which governs one-dimensional fluid dynamics and, in particular, models of formation and propagation of shock waves) do not support sufficiently smooth solutions. We suggest the reader to review the text of Garabedian [15], which makes a precise analysis of the necessary tools for the study of these type of solutions.

Instead of searching for explicit formulas, we will use numerical methods and neural networks techniques to approximate solutions of partial differential equations. When applying numerical methods, it is necessary to introduce a third concept, complementary to that of existence and uniqueness, namely that of a well-posed problem. A problem with a unique solution is said to be well-posed if any small change in the data of the problem leads to a small change in the solution. This consideration makes it possible to find an approximate solution of PDE by means of numerical algorithms. Of the numerical approximation methods available for solving differential equations those employing finite differences are more frequently used and more universally applicable than any other [35]. The essence of all finite difference methods is the replacement, at each of a discrete number of given points, of the partial derivatives by approximations involving the dependent variable evaluated at each given point and at appropriate neighboring ones. This transforms the problem of solving the partial differential equation to one of solving a set of linear algebraic equations. The main difficulty is ensuring that the numerical values obtained are good approximations to the exact solution. In order to solve this problem, let us consider that system (2.1)–(2.3) (fixed in each of the given points) is in a Hilbert space $\mathcal{H}$ with inner norm $\langle \cdot, \cdot \rangle$. Denote by $L_\infty([a,b];\mathcal{H})$ all $\mathcal{H}$-valued functions g such that $\langle g(\cdot), u \rangle$ is Lebesgue measurable for all $u \in \mathcal{H}$ and $\|g\|_\infty := \operatorname{ess\,sup}_{t\in[a,b]} |g(t,\gamma)| < \infty$. Now, let $g(t,\gamma)$ be piecewise continuous in t and satisfy the Lipschitz condition

$$\|g(t,\gamma) - g(t,\eta)\| \leq L\|\gamma - \eta\|$$

$\forall\, \gamma, \eta \in B_r = \{\gamma \in \Re^n \mid \|\gamma - \gamma_0\| \leq r\}, \forall\, t \in [t_0, t_1]$. Then there exists some $\delta > 0$ such that the state equation $\dot{\gamma} = g(t,\gamma)$ has a unique solution over $[t_0, t_0 + \delta]$ (see [19]). The norm used above stands for the Sobolev space defined as in [31] as follows

Definition 1. The Sobolev space consists of all functions (for simplicity, real valued) $f(t)$ defined on G which have p-integrable continuous derivatives $f^{(i)}(t)(i = 1, \ldots, l)$, that is

$$\begin{aligned} S_p^l(G) := \Bigg\{ & f(t) : G \to \Re \mid < \infty \ (i = 1, \ldots, l), \\ \|f\|_{S_p^l(G)} := & \left(\int_{t\in G} |f(t)|^p dt + \sum_{i=1}^{l} \int_{t\in G} \left| f^{(i)}(t) \right|^p dt \right)^{1/p} \Bigg\} \end{aligned} \tag{2.5}$$

where the integral is understood in the Lebesgue sense. More exactly, the Sobolev space is the completion of (2.5).

Remark 1. *The Sobolev space $S_2^l(G)$ of all l times differentiable on G quadratically integrable (in Lebesgue sense) complex functions under inner product*

$$\langle x, y \rangle_{S_p^l} := \sum_{i=0}^{l} \left\langle \frac{d^i}{dt^i} x, \frac{d^i}{dt^i} y \right\rangle_{L_2[a,b]}$$

is a Hilbert space.

Let us consider a function $h_0(\cdot) \in S_2^l(G)$. A classical result of functional analysis, the series expansion (see, for example, [31], [41]), states that for any $h_0(\cdot) \in S_2^l(G)$ the vector representation (if it exists) is

$$\begin{aligned} h_0(\gamma, \alpha^*) &= \sum_{i}^{\infty} \sum_{j}^{\infty} \sum_{k}^{\infty} \alpha_{ijk}^* \phi_{ijk}(\gamma) \\ \alpha_{ijk}^* &= \langle h_0, \phi_{ijk}(\gamma) \rangle_{S_2^l} \end{aligned} \tag{2.6}$$

where $\{\phi_{ijk}(\gamma)\}$ is an orthonormal system of functions that constitutes a basis for $S_2^l(G)$. Similar structures of neural networks to the series expansion (2.6) have been studied in [16], [30]. Based on this, we propose the following NN mathematical structure

$$\begin{aligned} h_0(\gamma, \alpha) &:= \sum_{i=N_1}^{N_2} \sum_{j=M_1}^{M_2} \sum_{k=L_1}^{L_2} \alpha_{ijk} \phi_{ijk}(\gamma) = \Theta^\mathsf{T} W(\gamma) \\ \Theta &= [\alpha_{N_1 M_1 L_1}, \ldots, \alpha_{N_1 M_1 L_2}, \ldots, \alpha_{N_2 M_1 L_1}, \ldots, \alpha_{N_2 M_2 L_2}]^\mathsf{T} \\ W(\gamma) &= [\phi_{N_1 M_1 L_1}, \ldots, \phi_{N_1 M_1 L_2}, \ldots, \phi_{N_2 M_1 L_1}, \ldots, \phi_{N_2 M_2 L_2}]^\mathsf{T} \end{aligned} \tag{2.7}$$

With this NN representation, we claim that any nonlinear function $h_0 \in S_2^l(G)$ can be approximated with an adequate selection of positive integers $N_1, N_2, M_1, M_2, L_1, L_2$. Moreover, the Stone-Weierstrass theorem [6] states conditions that guarantee that the network (2.7) can approximate continuous functions, this is, for any arbitrary positive constant ϵ there are some positive constants $N_1, N_2, M_1, M_2, L_1, L_2$ such that the approximation error satisfies the following

$$\| h_0(\gamma, \alpha^*) - h_0(\gamma, \alpha) \| \leq \epsilon \tag{2.8}$$

2.2 3D approximation of uncertain PDE

The main idea of applying the differential neural network methodology [30] is to use a class of finite difference method for uncertain nonlinear functions. To achieve this, it is necessary to construct a set (called *grid* or *mesh*) that divides the sub–domain $x \in [0,1]$ in N equidistant sections, $y \in [0,1]$ in M equidistant sections and $z \in [0,1]$ in L equidistant sections defined as $\left(x^i, y^j, z^k\right)$ in such way that $x^0 = y^0 = z^0 = 0$ and $x^N = y^M = z^L = 1$. Using the mesh representation, the following definitions can be used

$$
\begin{aligned}
u^{i,j,k}(t) &:= u(x^i, y^j, z^k, t) \\
u_t^{i,j,k}(t) &:= \left.\frac{\partial u(x,y,z,t)}{\partial t}\right|_{x=x^i, y=y^j, z=z^k} \\
u_x^{i,j,k}(t) &:= \left.u_x^{i,j,k}(x,y,z,t)\right|_{x=x^i, y=y^j, z=z^k} \\
u_{xx}^{i,j,k}(t) &:= \left.u_{xx}^{i,j,k}(x,y,z,t)\right|_{x=x^i, y=y^j, z=z^k}
\end{aligned}
$$

and it follows for the other cases (u_y, u_z, u_{yy}, u_{zz}, u_{xy}, u_{xz}, u_{yz}). Using the same representation, it can be applied the finite–difference representation to approximate partial derivatives as

$$
\begin{aligned}
u_x^{i,j,k}(t) &\simeq \frac{u^{i,j,k}(t) - u^{i-1,j,k}(t)}{\Delta x}, \quad & u_{xx}^{i,j,k}(t) &\simeq \frac{u_x^{i,j,k}(t) - u_x^{i-1,j,k}(t)}{\Delta^2 x} \\
u_y^{i,j,k}(t) &\simeq \frac{u^{i,j,k}(t) - u^{i,j-1,k}(t)}{\Delta x}, \quad & u_{yy}^{i,j,k}(t) &\simeq \frac{u_y^{i,j,k}(t) - u_y^{i,j-1,k}(t)}{\Delta^2 y} \\
u_z^{i,j,k}(t) &\simeq \frac{u^{i,j,k}(t) - u^{i,j,k-1}(t)}{\Delta x}, \quad & u_{zz}^{i,j,k}(t) &\simeq \frac{u_z^{i,j,k}(t) - u_z^{i,j,k-1}(t)}{\Delta^2 y} \\
u_{xy}^{i,j,k}(t) &\simeq \frac{u_y^{i,j,k}(t) - u_y^{i-1,j,k}(t)}{\Delta x}, \quad & u_{yz}^{i,j,k}(t) &\simeq \frac{u_z^{i,j,k}(t) - u_z^{i,j-1,k}(t)}{\Delta^2 y} \\
u_{yx}^{i,j,k}(t) &\simeq \frac{u_x^{i,j,k}(t) - u_x^{i,j-1,k}(t)}{\Delta x}, \quad & u_{xz}^{i,j,k}(t) &\simeq \frac{u_z^{i,j,k}(t) - u_z^{i-1,j,k}(t)}{\Delta^2 y}
\end{aligned}
$$

Using the (Δx, Δy, Δz)–approximation, the nonlinear PDE (2.1) can be represented as

$$
\begin{gathered}
u_t^{i,j,k}(t) \simeq \Phi \left(u^{i,j,k}, u^{i,j,k}, u^{i-2,j,k}, u^{i,j-1,k}, u^{i,j-2,k}, u^{i,j,k-1}\right. \\
\left. u^{i,j,k-2}, u^{i.1,j-1,k}, u^{i,j-1,k-1}, u^{i-1,j,k-1}, u^{i-1,j-1,k-1}\right) \\
i = \overline{1,N};\ j = \overline{1,M};\ k = \overline{1,L}
\end{gathered} \tag{2.9}
$$

It is well known that any function sufficiently smooth can be approximated arbitrary closely on a compact set by a finite sum of sigmoid functions [7]. By adding and subtracting the corresponding terms with an adequate selection of a neural network set of activation functions, equation (2.1) can be written as

$$u_t(x,y,z,t) = Au(x,y,z,t) + \sum_{r=1}^{11} \overset{\circ}{V}_r \Omega^r(x,y,z) U^r(x,y,z,t) + \tilde{f}(x,y,z,t) \tag{2.10}$$

where $A \in \Re^{n\times n}$, $\overset{\circ}{V}_r \in \Re^{n\times s_r}$ for $r = \overline{1,11}$ are any constant matrices. This construction reflects the method of approximation of functions described in equation (2.7). The approximation (2.10) and the sets $U^r(x,y,z,t)$ and $\Omega^r(x,y,z,t)$ contain eleven terms corresponding to the eleven in which function $f(\cdot)$ of (2.1) is evaluated. With these eleven elements, we can ensure that the modeling error term $\tilde{f}(x,y,z,t)$, defined as the difference of the function $f(\cdot)$ and the so-called nominal section is bounded for certain given (and known) values of $\overset{\circ}{V}_r$, $r = \overline{1,11}$. This statement obeys a direct application of the Stone-Weierstrass theorem [6]. The term $\Omega^r(x,y,z,t)$ refers to a set of monotonically increasing functions whose elements are given by $\sigma^1(x,y,z) \in \Re^{s_1\times n}$, $\varphi^1(x,y,z) \in \Re^{s_2\times n}$, $\gamma^1(x,y,z) \in \Re^{s_3\times n}$, $\varphi^2(x,y,z) \in \Re^{s_4\times n}$, $\gamma^2(x,y,z) \in \Re^{s_5\times n}$, $\varphi^3(x,y,z) \in \Re^{s_6\times n}$, $\gamma^3(x,y,z) \in \Re^{s_7\times n}$, $\psi^1(x,y,z) \in \Re^{s_8\times n}$, $\psi^2(x,y,z) \in \Re^{s_9\times n}$, $\psi^3(x,y,z) \in \Re^{s_{10}\times n}$ and $\sigma^2(x,y,z) \in \Re^{s_{11}\times n}$, correspondingly. For this application in particular, sigmoid functions [7] are selected as activation functions, which are bounded by positive constants for all x, y, z, i.e.

$$\|\sigma^1(\cdot)\| \le \sigma^{1+}; \quad \|\sigma^2(\cdot)\| \le \sigma^{2+};$$
$$\|\sigma^l(\cdot)\| \le \sigma^{l+}; \quad \|\varphi^l(\cdot)\| \le \varphi^{l+}; \quad \|\psi^l(\cdot)\| \le \psi^{l+}; \quad l = \overline{1,3}$$

Applying the same concept to the Δ–approximation (2.9) of the nonlinear PDE (2.1), we get for each $i \in \overline{1,N}$, $j \in \overline{1,M}$, $k \in \overline{1,L}$ the following relation

$$u_t^{i,j,k}(t) = A^{i,j,k} u^{i,j,k}(t) + \sum_{r=1}^{11} \overset{\circ}{W}{}_r^{i,j,k} \Omega^r(x^i,y^j,z^k) U^{(i,j,k),r}(t) + \tilde{f}^{i,j,k}(t) \tag{2.11}$$

where $\tilde{f}^{i,j,k}(t)$ is the modeling error, and it satisfies the following identity

$$\tilde{f}^{i,j,k}(t) = \Phi\left(u^{i,j,k}, u^{i-1,j,k}, u^{i-2,j,k}, u^{i,j-1,k}, u^{i,j-2,k}, u^{i,j,k-1}\right.$$
$$\left.u^{i,j,k-2}, u^{i.1,j-1,k}, u^{i,j-1,k-1}, u^{i-1,j,k-1}, u^{i-1,j-1,k-1}\right)$$
$$-A^{i,j,k}u^{i,j,k}(t) + \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Omega^r(x^i, y^j, z^k)U^{(i,j,k),r}(t)$$

with $\mathring{W}_r^{i,j,k} \in \Re^{n\times s_r}$ for $r = \overline{1,11}$, any fixed given (and known) matrices that can be considered as the initial values for the weight matrices. Equation (2.11) is obtained by applying the finite difference representation to equation (2.10). The physical interpretation of this new equation gives us an intuitive idea about the construction of the neural network. That is, for each coordinate (i, j, k) we ensure that each of the corresponding partial derivatives in equation (2.1) is estimated and also used for approximating the solution by proper training of the neural network. Finally, it will be assumed that the modeling error satisfy the next assumptions

Assumption 1. The modeling error is absolutely bounded in the domain G, i.e.,

$$\left\|\tilde{f}^{i,j,k}\right\|^2 \le f_1^{i,j,k} \tag{2.12}$$

Assumption 2. The error modeling gradient defined as

$$\nabla_m \tilde{f}(x,y,z,t)\big|_{m=m^i} := \nabla_m \tilde{f}^{i,j,k} \tag{2.13}$$

where m represents the partial derivate by x, y and z, correspondingly, is bounded, i.e.,

$$\left\|\nabla_m \tilde{f}^{i,j,k}\right\|^2 \le f_s^{i,j,k} \tag{2.14}$$

where $f_s^{i,j,k}$ $(s = \overline{1,4})$ are positive constants.

Chapter 3

Neural identification of 3D distributed parameter systems

The data of the problems of technology are invariantly subject to errors of measurement, quantization processes or loss of information. Finite differences methods and neural networks techniques generally give solutions that are as accurate as the data warrant. For the purpose of this paper, we use three types of restriction on the information available for the approximation. In each section, we provide and construct an upper bound for the approximation error and demonstrate its practical stability via Lyapunov-like analysis. At the same time, we derive the learning laws for the suggested neural networks.

3.1 DNN identification for distributed parameter systems with sample–data measurements.

The principal motivation of this section is to study systems of the form

$$\begin{aligned} u_t(x,y,z,t) =& f\left(u(x,y,z,t), u_x(x,y,z,t), u_{xx}(x,y,z,t), u_y(x,y,z,t),\right. \\ & u_{yy}(x,y,z,t), u_z(x,y,z,t), u_{zz}(x,y,z,t), u_{xy}(x,y,z,t), \\ & \left. u_{yx}(x,y,z,t), u_{xz}(x,y,z,t), u_{yz}(x,y,z,t)\right) + \xi(x,y,z,t) \\ \bar{u}^{i,j,k}(t) =& C^{i,j,k} u^{i,j,k}(t_k) \chi^{i,j,k}_{[t_k,t_{k+1})} \end{aligned} \quad (3.1)$$

where $f(\cdot)$ is an unknown nonlinear partial differential equation satisfying conditions stated in chapter 2 and $C^{i,j,k} \in \Re^{q\times n}$ is a given matrix, fixed for each (x^i, y^j, z^k). Boundary and initial conditions as well as the domain are defined like in the previous section. The variable $\bar{u}^{i,j,k}(t)$ describes the real available sample–data measurements, i.e., the stepwise values of $\bar{u}^{i,j,k}(t)$ represents the real measurable output of the system at each fixed point in the mesh. Here

$$\chi^{i,j,k}_{[t_k,t_{k+1})} := \begin{cases} 1 & \text{if } t \in [t_k, t_{k+1}), \\ 0 & \text{otherwise.} \end{cases}$$

denotes the characteristic function of the time interval $[t_k, t_{k+1})$. Based on the DNN–methodology [30], let us consider the following DNN–identifier

$$\begin{gathered} \frac{d}{dt}\hat{u}^{i,j,k}(t) = A^{i,j,k}\hat{u}^{i,j,k}(t) + \sum_{r=1}^{11} W_r^{i,j,k}\Omega^r(x^i, y^j, z^k)\hat{U}^{(i,j,k),r}(t) \\ +L^{i,j,k}\left(\bar{u}^{i,j,k}(t) - C^{i,j,k}\hat{u}^{i,j,k}(t)\right) \\ i = \overline{0,N}; j = \overline{0,M}; k = \overline{0,L} \end{gathered} \tag{3.2}$$

where $\hat{u}^{i,j,k}(t)$ is the estimate of $u^{i,j,k}(t)$ and $L^{i,j,k} \in \Re^{n\times q}$. It is clear that this methodology implies the design of an individual DNN–identifier for each point (x^i, y^j, z^k) in the mesh representation. The collection of these identifiers constitute a DNN–net composed by $N \times M \times L$ connected identifiers working in parallel. Let us introduce the following auxiliary variables

$$\begin{aligned} \tilde{u}^{i,j,k}(t) &:= \hat{u}^{i,j,k}(t) - u^{u,j,k}(t), \quad \tilde{u}_x^{i,j,k}(t) := \hat{u}_x^{i,j,k}(t) - u_x^{i,j,k}(t) \\ \tilde{u}_y^{i,j,k}(t) &:= \hat{u}_y^{i,j,k}(t) - u_y^{i,j,k}(t), \quad \tilde{u}_z^{i,j,k}(t) := \hat{u}_z^{i,j,k}(t) - u_z^{i,j,k}(t) \end{aligned}$$

which define the error between the trajectories produced by the model and the DNN–identifier as well as their derivatives with respect to x, y and z, for each i, j, k. Additionally, consider the variable

$$\Delta u^{i,j,k} := \bar{u}^{i,j,k}(t) - u^{i,j,k}(t) \tag{3.3}$$

which is bounded as

$$\left\|\Delta u^{i,j,k}\right\|^2 \leq \Delta_{1,+}^{i,j,k} \tag{3.4}$$

where $\Delta_{1,+}^{i,j,k}$ is a positive constant.

3.1.1 Learning laws

Let the time–varying matrices $\tilde{W}_r^{i,j,k}(t) \in \Re^n$, $r = \overline{1,11}$ satisfy the following nonlinear matrix differential equations

$$
\begin{aligned}
\dot{W}_r^{i,j,k} = & -\frac{a}{2}\tilde{W}_r^{i,j,k} - K_r^{-1}\Pi_b u_e^{i,j,k}(t)\left[\hat{U}^{(i,j,k),r}(t)\right]^{\mathsf{T}}\left[\Omega^r(x^i,y^j,z^k)\right]^{\mathsf{T}} \\
& - \sum_{l=1}^{3} K_r^{-1} S_l^{i,j,k} u_{e_m}^{i,j,k}(t)\left[\hat{U}^{(i,j,k),r}(t)\right]^{\mathsf{T}}\left[\Omega_m^r(x^i,y^j,z^k)\right]^{\mathsf{T}} \\
& - \frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+61}\Pi_b W_r^{i,j,k}\Omega^r\hat{U}^{(i,j,k),r}(t)\left[\hat{U}^{(i,j,k),r}\right]^{\mathsf{T}}\left[\Omega^r\right]^{\mathsf{T}} \\
& - \frac{1}{2}\sum_{l=1}^{3} K_r^{-1} S_l^{i,j,k}\Lambda_{r+72} S_l^{i,j,k} W_r^{i,j,k}\Omega_m^r\hat{U}^{(i,j,k),r}(t)\left[\hat{U}^{(i,j,k),r}\right]^{\mathsf{T}}\left[\Omega_m^r\right]^{\mathsf{T}} \\
& - \frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+105}\Pi_b W_r^{i,j,k}\Omega^r\hat{U}^{(i,j,k),r}(t)\left[\hat{U}^{(i,j,k),r}\right]^{\mathsf{T}}\left[\Omega^r\right]^{\mathsf{T}}
\end{aligned}
\tag{3.5}
$$

Here, K_r ($r = \overline{1,11}$) are positive definite matrices, $\tilde{W}_r^{i,j,k}(t) := W_r^{i,j,k} - \mathring{W}_r^{i,j,k}$ and $u_e^{i,j,k}(t) := C^{i,j,k}\hat{u}^{i,j,k}(t) - \bar{u}^{i,j,k}(t)$. Matrices $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ ($i = \overline{1,N}; j = \overline{1,M}, k = \overline{1,L}$) are positive definite solutions of the following Riccati matrix inequalities

$$
\begin{aligned}
& \begin{bmatrix} -S_1^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_1^{i,j,k} - Q_{S_1}^{i,j,k} & S_1^{i,j,k}\left[R_{S_1}^{i,j,k}\right]^{1/2} \\ \left[R_{S_1}^{i,j,k}\right]^{1/2} S_1^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\
& R_{S_1}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+12}\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{24} \\
& Q_{S_1}^{i,j,k} := aS_1^{i,j,k}
\end{aligned}
\tag{3.6}
$$

$$
\begin{aligned}
& \begin{bmatrix} -S_2^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_2^{i,j,k} - Q_{S_1}^{i,j,k} & S_2^{i,j,k}\left[R_{S_2}^{i,j,k}\right]^{1/2} \\ \left[R_{S_2}^{i,j,k}\right]^{1/2} S_2^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\
& R_{S_2}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+24}\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{36} \\
& Q_{S_2}^{i,j,k} := aS_2^{i,j,k}
\end{aligned}
\tag{3.7}
$$

$$\begin{bmatrix} -S_3^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_3^{i,j,k} - Q_{S_1}^{i,j,k} & S_3^{i,j,k}\left[R_{S_3}^{i,j,k}\right]^{1/2} \\ \left[R_{S_3}^{i,j,k}\right]^{1/2} S_3^{i,j,k} & I_{n\times n} \end{bmatrix} > 0$$

$$R_{S_3}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+36}\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{48} \tag{3.8}$$

$$Q_{S_3}^{i,j,k} := aS_3^{i,j,k}$$

with $a > 0$ any given constant. Functions $\Omega_m^r(x^i, y^j, z^k)$ (with $m = x, y$ or z) are defined as

$$\Omega_m^r(x^i, y^j, z^k) := \frac{d}{dm}\Omega^r(x, y, z)\big|_{x=x^i, y=y^j, z=z^k} \tag{3.9}$$

3.1.2 Main results

In order to analyze the quality of the DNN–identifier (3.2) with learning laws given by (3.5), let us first present two useful results and a definition needed to state the main results of this work.

Lemma 1. Let a nonnegative function $V(t)$ satisfy the following differential inequality

$$\frac{d}{dt}V_t \le -\alpha V(t) + \beta$$

where $\alpha > 0$ and $\beta > 0$. Then

$$\overline{\lim_{t\to\infty}} V(t) \le \beta/\alpha \tag{3.10}$$

Proof. The proof of this lemma can be found on [32, p.75]. □

The second result is known as the Λ–matrix inequality and it states that

Lemma 2. For any matrices X, $Y \in \Re^{n\times m}$ and any symmetric positive definite matrix $\Lambda \in \Re^{n\times n}$ the following inequalities hold

$$X^{\mathsf{T}}Y + Y^{\mathsf{T}}X \le X^{\mathsf{T}}\Lambda X + Y^{\mathsf{T}}\Lambda^{-1}Y \tag{3.11}$$

and

$$(X+Y)^{\mathsf{T}}(X+Y) \le X^{\mathsf{T}}(I_{n\times n} + \Lambda)X + Y^{\mathsf{T}}\left(I_{n\times n} + \Lambda^{-1}\right)Y \tag{3.12}$$

Proof. See [31, p.213] for a detailed proof. □

Now, consider the following nonlinear system

$$\dot{\eta} = f(t, \eta, u) + g(t, \eta) \tag{3.13}$$

where $\eta \in \Re^n$, $u \in \Re^m$ and $g(\cdot)$ is an external bounded perturbation term such that $\|g(t, x)\| \leq g^+$, where g^+ is a positive constant.

Definition 2. Given $\epsilon > 0$, the system (3.13) the system is said to be ϵ-practically stable around the origin if, for any $t_0 \geq 0$, there exists a $\delta = \delta(\epsilon, t_0) \geq 0$ such that $\eta(t) \in B[0, \epsilon]$, $\forall t \geq t_0$, whenever $\eta_0 = \eta(t_0) \in B[0, \delta]$. If δ is independent of t_0, then the system is said to be uniformly ϵ-practically stable around the origin.

We are now able to formulate the following results

Theorem 1. Consider the nonlinear model (3.1), given by the system of PDEs with uncertainties in the states, and sample–data outputs, with initial and boundary conditions given by (2.3). Suppose that the DNN–identifier is given by (3.2) and its parameters are adjusted by the learning laws (3.5). If there exist positive definite matrices $R_{S_1}^{i,j,k}$, $R_{S_2}^{i,j,k}$ and $R_{S_3}^{i,j,k}$ $(i = \overline{1, N}; j = \overline{1, M}; k = \overline{1, L})$ such that Riccati matrix inequalities (3.6)–(3.8) have a positive definite solutions $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ $(i = \overline{1, N}; j = \overline{1, M}; k = \overline{1, L})$, and if there exist matrices Π_b, Π_c, $P^{i,j,k}$, $L^{i,j,k}$ and parameter ϵ such that

$$W^{i,j,k} = \begin{pmatrix} \mathrm{w}_{11}^{i,j,k} & \mathrm{w}_{12}^{i,j,k} & 0 & 0 & \mathrm{w}_{15}^{i,j,k} \\ \mathrm{w}_{21}^{i,j,k} & \mathrm{w}_{22}^{i,j,k} & 0 & 0 & \mathrm{w}_{25}^{i,j,k} \\ 0 & 0 & \mathrm{w}_{33}^{i,j,k} & 0 & 0 \\ 0 & 0 & 0 & \mathrm{w}_{44}^{i,j,k} & 0 \\ \mathrm{w}_{51}^{i,j,k} & \mathrm{w}_{52}^{i,j,k} & 0 & 0 & \mathrm{w}_{55}^{i,j,k} \end{pmatrix} \tag{3.14}$$

whose elements are given as follows

$$
\begin{aligned}
\mathrm{w}_{11}^{i,j,k} :=& \Pi_b \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k}\right) + \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k}\right)^{\mathsf{T}} \Pi_b + \Pi_b R_{\Pi_b}^{i,j,k} \Pi_b \\
& + Q_{\Pi_b}^{i,j,k} + S_4^{i,j,k} + a P^{i,j,k} \\
\mathrm{w}_{12}^{i,j,k} :=& P^{i,j,k} + \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k}\right)^{\mathsf{T}} \Pi_c - \Pi_b \\
\mathrm{w}_{21}^{i,j,k} :=& P^{i,j,k} + \Pi_c \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k}\right) - \Pi_b \\
\mathrm{w}_{22}^{i,j,k} :=& h^2 R^{i,j,k} - 2\Pi_c + \Pi_c R_{\Pi_c}^{i,j,k} \Pi_c \\
\mathrm{w}_{33}^{i,j,k} :=& -h e^{-ah} R^{i,j,k} \\
\mathrm{w}_{44}^{i,j,k} :=& -e^{-ah} S_4^{i,j,k} \\
.\mathrm{w}_{55}^{i,j,k} :=& -\epsilon I;\ \mathrm{w}_{15}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{51}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{25}^{i,j,k} := -\Pi_c;\ \mathrm{w}_{52}^{i,j,k} := -\Pi_c
\end{aligned}
\tag{3.15}
$$

is negative definite, then the error of identification $\tilde{u}^{i,j,k}(t)$ converges in practical sense to

$$
\overline{\lim_{t\to\infty}} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \left\| \tilde{u}^{i,j,k}(t) \right\|_{S_2^l(G)}^2 \leq \beta/\alpha \tag{3.16}
$$

where $\alpha := a$ and

$$
\begin{aligned}
\beta :=& \varpi_1 \overline{\sum} \sum_{s=1}^{4} f_s^{i,j,k} + \varpi_2 \overline{\sum} \left\| L^{i,j,k} \Delta u^{i,j,k} \right\|^2 + \varpi_3 \overline{\sum} \Delta_{1,+}^{i,j,k} \\
& + \varpi_4 \overline{\sum} \Delta_{2,+}^{i,j,k} + \varpi_5 \overline{\sum} \Delta_{3,+}^{i,j,k} + \varpi_6 \overline{\sum} \Delta_{4,+}^{i,j,k} \\
\varpi_1 :=& \max\left\{\epsilon, \lambda_{\max}\left(\Lambda_{24}^{-1}\right), \lambda_{\max}\left(\Lambda_{32}^{-1}\right), \lambda_{\max}\left(\Lambda_{48}^{-1}\right)\right\} \\
\varpi_2 :=& \lambda_{\max}\left(\Lambda_{60}^{-1}\right) + \lambda_{\max}\left(\Lambda_{61}^{-1}\right) \\
\varpi_3 :=& \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+61}\right) \quad \varpi_4 := \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+72}\right) \\
\varpi_5 :=& \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+83}\right) \quad \varpi_6 := \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+94}\right)
\end{aligned}
\tag{3.17}
$$

Proof. The detailed proof is given in the appendix. □

An immediate consequence of theorem 1 is given in the following corollary.

Corollary 1. The DNN–weight trajectories satisfy the inequality

$$\overline{\lim_{t\to\infty}} \overline{\sum} \left\| \tilde{W}_r^{i,j,k} \right\|^2 \leq K_r^{-1} \frac{\beta}{\alpha}; \quad r = \overline{1, 11} \tag{3.18}$$

That is, the weights remain bounded and are proportional to β/α.

Proof. Straightforward from the proof of theorem 1 □

Evidently, the problem of selecting a set of optimal matrices Π_b, Π_c, $P^{i,j,k}$ and $L^{i,j,k}$ such that $W^{i,j,k} \leq 0$ is a strongly nonlinear problem of mathematical programming. This problem is associated with the resolution of a bilinear matrix inequality (BMI). Our aim is to relax the given nonlinear matrix–constrain with a suitable system of linear matrix inequalities (LMI). To achieve this, select the next matrices as

$$\begin{aligned} \Pi_b = \Pi_c &:= P^{i,j,k} \\ R_{\Pi_b}^{i,j,k} &:= \left[P^{i,j,k}\right]^{-1} \bar{R}_1 \left[P^{i,j,k}\right]^{-1} \\ R_{\Pi_c}^{i,j,k} &:= \left[P^{i,j,k}\right]^{-1} \bar{R}_2 \left[P^{i,j,k}\right]^{-1} \end{aligned} \tag{3.19}$$

where $\bar{R}_1$ and $\bar{R}_2$ are any symmetric positive definite matrices. Then, the elements of $W^{i,j,k}$ can be simplified as following

$$\begin{aligned} \mathrm{w}_{11}^{i,j,k} &:= P^{i,j,k} \left(\tilde{A}^{i,j,k} - L^{i,j,k} C\right) + \left(\tilde{A}^{i,j,k} - L^{i,j,k} C\right) P^{i,j,k} + \tilde{Q}^{i,j,k} \\ \mathrm{w}_{12}^{i,j,k} &:= \left(A^{i,j,k} - L^{i,j,k} C\right)^{\top} P^{i,j,k} \\ \mathrm{w}_{21}^{i,j,k} &:= P^{i,j,k} \left(A^{i,j,k} - L^{i,j,k} C\right) \\ \mathrm{w}_{22}^{i,j,k} &:= h^2 R^{i,j,k} + \bar{R}_2 - 2P^{i,j,k} \\ \mathrm{w}_{33}^{i,j,k} &:= -he^{-ah} R^{i,j,k} \\ \mathrm{w}_{44}^{i,j,k} &:= -e^{-ah} S_4^{i,j,k} \\ .\mathrm{w}_{55}^{i,j,k} &:= -\epsilon I;\ \mathrm{w}_{15}^{i,j,k} := -P^{i,j,k};\ \mathrm{w}_{51}^{i,j,k} := -P^{i,j,k};\ \mathrm{w}_{25}^{i,j,k} := -P^{i,j,k};\ \mathrm{w}_{52}^{i,j,k} := -P^{i,j,k} \end{aligned} \tag{3.20}$$

where $\tilde{A}^{i,j,k} := A^{i,j,k} + \frac{a}{2} I$ and $\tilde{Q}^{i,j,k} := \bar{R}_1 + Q_{\Pi_b}^{i,j,k} + S_4^{i,j,k}$. Using the following additional notation

$$\begin{aligned} X^{i,j,k} &:= P^{i,j,k} \\ Y^{i,j,k} &:= P^{i,j,k} L^{i,j,k} \end{aligned} \tag{3.21}$$

the initial problem of the nonlinear matrix–constraint $W^{i,j,k} \leq 0$ can be reduced to a suitable system of LMIs, defined as

$$W^{i,j,k} = \begin{pmatrix} \Xi^{i,j,k} & \left[A^{i,j,k}\right]^{\top} X^{i,j,k} - C^{\top}\left[Y^{i,j,k}\right]^{\top} & 0 & 0 & -X^{i,j,k} \\ X^{i,j,k}A^{i,j,k} - Y^{i,j,k}C & h^2 R^{i,j,k} - 2X^{i,j,k} + \bar{R}_2 & 0 & 0 & -X^{i,j,k} \\ 0 & 0 & -he^{-ah}R^{i,j,k} & 0 & 0 \\ 0 & 0 & 0 & -e^{-ah}S_4^{i,j,k} & 0 \\ -X^{i,j,k} & -X^{i,j,k} & 0 & 0 & -\epsilon I \end{pmatrix}$$

$$\Xi^{i,j,k} := X^{i,j,k}\tilde{A}^{i,j,k} - Y^{i,j,k}C + \left[\tilde{A}^{i,j,k}\right]^{\top} X^{i,j,k} - C^{\top}\left[Y^{i,j,k}\right]^{\top} + \tilde{Q}^{i,j,k} \tag{3.22}$$

Regarding this considerations, let us present two additional results

Theorem 2. Under assumptions of theorem 1 about solution of the Riccati equations (3.6)–(3.8) and choosing matrices as in (3.19) and (3.21) if there exist a solution $\Upsilon := (\epsilon, X^{i,j,k}, Y^{i,j,k})$ of the simplified LMI (3.22) such that $W^{i,j,k} \leq 0$, then the error of identification $\tilde{u}^{i,j,k}(t)$ converges in practical sense to

$$\overline{\lim_{t\to\infty}} \sum_{i=0}^{N}\sum_{j=0}^{M}\sum_{k=0}^{L} \left\|\tilde{u}^{i,j,k}(t)\right\|^2_{S_2^l(G)} \leq \beta/\alpha \tag{3.23}$$

Moreover, the corresponding gain matrix $L^{i,j,k}$ is given by

$$L^{i,j,k} := Y^{i,j,k}\left[P^{i,j,k}\right]^{-1} \tag{3.24}$$

Proof. The proof of this theorem is based on a linear approximation of the set given by the matrix inequality (3.50). □

The next remark provides additional features about matrices $X^{i,j,k}$ and $Y^{i,j,k}$.

Remark 2. *Assume that the following auxiliary optimization problem*

$$\begin{aligned}&\min_{\Upsilon}\ \operatorname{tr}\left(\left[X^{i,j,k}\right]^{-1}\right)\\&\textit{subject to } W^{i,j,k}\leq 0\\&X^{i,j,k}>0\ \ Y^{i,j,k}>0\end{aligned}\tag{3.25}$$

has an optimal solution

$$\hat{\Upsilon}:=\left(\epsilon,\hat{X}^{i,j,k},\hat{Y}^{i,j,k}\right)\tag{3.26}$$

Then, the ellipsoid defined by the matrix $\hat{X}^{i,j,k}$ approximates a minimal attractive ellipsoid for the error of identification $\tilde{u}^{i,j,k}(t)$. Theory and results on the atractive ellipsoid method can be consulted in [4], [22], [28] and other related papers.

3.2 DNN identification for distributed parameter systems with scattered grid outputs.

Through this section, let us consider the following set of uncertain PDEs

$$\begin{aligned}u_t(x,y,z,t)=&f\left(u(x,y,z,t),u_x(x,y,z,t),u_{xx}(x,y,z,t),u_y(x,y,z,t),\right.\\&u_{yy}(x,y,z,t),u_z(x,y,z,t),u_{zz}(x,y,z,t),u_{xy}(x,y,z,t),\\&\left.u_{yx}(x,y,z,t),u_{xz}(x,y,z,t),u_{yz}(x,y,z,t)\right)+\xi(x,y,z,t)\\u^{i,j,k}(t)=&u^{i^*,j^*,k^*}(t),\text{ for }\ (i^*,j^*,k^*)\in\mathcal{I}\end{aligned}\tag{3.27}$$

where $u\in\Re^n$ in the domain (2.2) and $\mathcal{I}$ is a set of indices of positive integers constituting a 3-tuple indicating a location in the mesh representation. The boundary (Neumann and Dirichlet) and initial conditions are given by (2.3). Equations (3.27) stand that the measurable output of the system is distributed unevenly along the mesh (or grid), i.e., the (Δx, Δy, Δz)–approximation $u^{i,j,k}(t)$ of the system (3.27) is available only for some i,j,k along the grid. The problem that arises is to approximate the missing data and modify the identifier and its learning laws to estimate the upper bound of the identification error given by

$$\overline{\lim_{t\to\infty}}\sum_{i=0}^{N}\sum_{j=0}^{M}\sum_{k=0}^{L}\left\|\hat{u}^{i,j,k}(t)-u^{i,j,k}(t)\right\|^2_{S_2^l(G)}\tag{3.28}$$

and if it is possible to reduce it to the lowest possible value, selecting free parameters participating in the DNN–identifier.

3.2.1 Surface fitting

The scattered nature of available data to approximate the solution of the system (3.27), suggests using a surface fitting scheme to approximate the missing information in the mesh representation. Thus the interpolation problem might be defined as follows: given D data points (x_i, y_i, z_i) and D numbers f_i, $i = 1, 2, \ldots, D$, find a function $f(x, y, z)$ from some class and defined on the whole space (or at leats a region containing the data points) for which $f(x_i, y_i, z_i) = f_i$ for $i = 1, 2, \ldots, D$. It is well documented that there is no universal choice for the solution of the above problem. Depends on the nature of the data and the nature of the modeled phenomenon the choice of the surface fitting technique. For the specific purpose of this work, we chose the triangle-based linear interpolation method to perform the surface fitting regarding the available data. We may refer the reader to [25], [36] to consult the technical details of the interpolation algorithm.

3.2.2 Identifier and learning laws

Based on DNN–methodology [30], consider the following identifier

$$\begin{gathered}\frac{d}{dt}\hat{u}^{i,j,k}(t) = A^{i,j,k}\hat{u}^{i,j,k}(t) + \sum_{r=1}^{11} W_r^{i,j,k}\Omega^r(x^i, y^j, z^k)\hat{U}^{(i,j,k),r}(t) \\ i = \overline{0, N};\, j = \overline{0, M};\, k = \overline{0, L}\end{gathered} \tag{3.29}$$

where $A^{i,j,k} \in \Re^{n\times n}$ is constant matrix to be selected, $\hat{u}^{i,j,k}(t)$ is the estimate of $u^{i,j,k}(t)$. As in the previous section, this methodology implies the design of individual DNN–identifiers for each (x^i, y^j, z^k). Let us introduce the following auxiliary variables

$$\begin{aligned}\tilde{u}^{i,j,k}(t) &:= \hat{u}^{i,j,k}(t) - u^{u,j,k}(t), \quad \tilde{u}_x^{i,j,k}(t) := \hat{u}_x^{i,j,k}(t) - u_x^{i,j,k}(t) \\ \tilde{u}_y^{i,j,k}(t) &:= \hat{u}_y^{i,j,k}(t) - u_y^{i,j,k}(t), \quad \tilde{u}_z^{i,j,k}(t) := \hat{u}_z^{i,j,k}(t) - u_z^{i,j,k}(t)\end{aligned} \tag{3.30}$$

which define the error between the trajectories produced by the model and the DNN–identifier as well as their derivatives with respect to x, y and z, for each i, j, k. Suppose that through triangle-based linear interpolation method, we are able to provide an estimate $u_{\text{int}}^{i,j,k}(t)$ of the missing information along the grid, that satisfies

$$\left\| u_{\text{int}}^{i,j,k}(t) - u^{i,j,k} \right\|^2 \leq \eta_{1,+}^{i,j,k}(\Delta x, \Delta y, \Delta z) \tag{3.31}$$

where $\eta_{1,+}$ is a positive constant which depends directly on the amount of data available for the interpolation algorithm, as well as the maximum separation distance Δ_m for the coordinates x, y and z [25]. Let the time–varying matrices $\tilde{W}_r^{i,j,k}(t) \in \Re^n$, $r = \overline{1, 11}$ satisfy the following nonlinear matrix differential equations

$$\begin{aligned}
\dot{W}_r^{i,j,k} = &- \frac{\alpha}{2}\tilde{W}_r^{i,j,k} - K_r^{-1} P^{i,j,k} \bar{\bar{u}}^{i,j,k}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\
&- \sum_{l=1}^{3} K_r^{-1} S_l^{i,j,k} \bar{\bar{u}}_m^{i,j,k}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_m^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\
&- \frac{1}{2} K_r^{-1} P^{i,j,k} \Lambda_r P^{i,j,k} \tilde{W}_r^{i,j,k} \Omega^r(x^i, y^j, z^k) \hat{U}^{(i,j,k),r}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\
&- \frac{1}{2} \sum_{l=1}^{3} K_r^{-1} S_l^{i,j,k} \Lambda_r S_l^{i,j,k} \tilde{W}_r^{i,j,k} \Omega_m^r \hat{U}^{(i,j,k),r}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_m^r\right)^{\mathsf{T}}
\end{aligned} \tag{3.32}$$

where m represents the partial derivative with respect to x for $l = 1$, with respect to y for $l = 2$ and with respect to z for $l = 3$. Here $\bar{\bar{u}}^{i,j,k}(t) := \hat{u}^{i,j,k}(t) - u_{\text{int}}^{i,j,k}(t)$. Matrices $P^{i,j,k}$, $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ $(i = \overline{1, N}; j = \overline{1, M}, k = \overline{1, L})$ are positive definite solutions of the following Riccati matrix inequalities

$$\begin{bmatrix} -P^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} P^{i,j,k} - Q_P^{i,j,k} & P^{i,j,k}\left[R_P^{i,j,k}\right]^{1/2} \\ \left[R_P^{i,j,k}\right]^{1/2} P^{i,j,k} & I_{n\times n} \end{bmatrix} > 0$$

$$\begin{aligned} R_P^{i,j,k} &:= \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+12} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{12} \\ Q_P^{i,j,k} &:= \sum_{r=1}^{11} \left\|\Omega^r(x^i, y^j, z^k)\right\|_{\Lambda_r^{-1}}^2 + \sum_{r=1}^{11} \left\|\Omega_x^r(x^i, y^j, z^k)\right\|_{\Lambda_{r+12}^{-1}}^2 \\ &+ \sum_{r=1}^{11} \left\|\Omega_y^r(x^i, y^j, z^k)\right\|_{\Lambda_{r+21}^{-1}}^2 + \sum_{r=1}^{11} \left\|\Omega_z^r(x^i, y^j, z^k)\right\|_{\Lambda_{r+36}^{-1}}^2 + \alpha P^{i,j,k} \end{aligned} \tag{3.33}$$

$$\begin{gathered} \begin{bmatrix} -S_1^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_1^{i,j,k} - Q_{S_1}^{i,j,k} & S_1^{i,j,k}\left[R_{S_1}^{i,j,k}\right]^{1/2} \\ \left[R_{S_1}^{i,j,k}\right]^{1/2} S_1^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\ R_{S_1}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+12} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{24} \\ Q_{S_1}^{i,j,k} := \alpha S_1^{i,j,k} \end{gathered} \tag{3.34}$$

$$\begin{gathered} \begin{bmatrix} -S_2^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_2^{i,j,k} - Q_{S_1}^{i,j,k} & S_2^{i,j,k}\left[R_{S_2}^{i,j,k}\right]^{1/2} \\ \left[R_{S_2}^{i,j,k}\right]^{1/2} S_2^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\ R_{S_2}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+24} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{36} \\ Q_{S_2}^{i,j,k} := \alpha S_2^{i,j,k} \end{gathered} \tag{3.35}$$

$$\begin{gathered} \begin{bmatrix} -S_3^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\mathsf{T}} S_3^{i,j,k} - Q_{S_1}^{i,j,k} & S_3^{i,j,k}\left[R_{S_3}^{i,j,k}\right]^{1/2} \\ \left[R_{S_3}^{i,j,k}\right]^{1/2} S_3^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\ R_{S_3}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+36} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{48} \\ Q_{S_3}^{i,j,k} := \alpha S_3^{i,j,k} \end{gathered} \tag{3.36}$$

and $\alpha > 0$ is any constant.

3.2.3 Main result

The main result of this section is summarized in the next theorem

Theorem 3. Consider the nonlinear model (3.27), given by the system of PDEs with uncertainties in the states with initial and boundary conditions given by (2.3). Suppose that there exists an estimate $u_{\text{int}}^{i,j,k}(t)$ of the missing data such that

$$\left\| u_{\text{int}}^{i,j,k}(t) - u^{i,j,k} \right\|^2 \leq \eta_{1,+}^{i,j,k} \tag{3.37}$$

Suppose that the DNN–identifier is given by (3.29) and its parameters are adjusted by the learning laws (3.32). If there exist positive definite matrices $R_P^{i,j,k}$, $R_{S_1}^{i,j,k}$, $R_{S_2}^{i,j,k}$ and $R_{S_3}^{i,j,k}$ $(i = \overline{1,N}; j = \overline{1,M}; k = \overline{1,L})$ such that satisfies Riccati matrix inequalities (3.33)–(3.36) with positive definite solutions $P^{i,j,k}$, $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ $(i = \overline{1,N}; j = \overline{1,M}; k = \overline{1,L})$, then the error of identification $\tilde{u}^{i,j,k}(t)$ converges in practical sense to

$$\overline{\lim_{t\to\infty}} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \left\| \tilde{u}^{i,j,k}(t) \right\|_{S_2^l(G)}^2 \leq \beta/\alpha \tag{3.38}$$

where $\alpha > 0$ and

$$\begin{aligned}
\beta &:= \varpi_1 \overline{\sum} \sum_{s=1}^{4} f_s^{i,j,k} + \varpi_2 \overline{\sum} \eta_{1,+}^{i,j,k} + \varpi_3 \overline{\sum} \eta_{2,+}^{i,j,k} \\
&\quad + \varpi_4 \overline{\sum} \eta_{3,+}^{i,j,k} + \varpi_5 \overline{\sum} \eta_{4,+}^{i,j,k} \\
\varpi_1 &:= \max \left\{ \lambda_{\max}\left(\Lambda_{12}^{-1}\right), \lambda_{\max}\left(\Lambda_{24}^{-1}\right), \lambda_{\max}\left(\Lambda_{32}^{-1}\right), \lambda_{\max}\left(\Lambda_{48}^{-1}\right) \right\} \\
\varpi_2 &:= \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+48}^{-1}\right) \quad \varpi_3 := \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+59}^{-1}\right) \\
\varpi_4 &:= \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+70}^{-1}\right) \quad \varpi_5 := \sum_{r=1}^{11} \lambda_{\max}\left(\Lambda_{r+82}^{-1}\right)
\end{aligned} \tag{3.39}$$

Proof. The detailed proof is given in the appendix. □

3.3 DNN identification for distributed parameter systems with sample–data measurements and scattered grid outputs.

Consider the following set of uncertain PDEs whose outputs are given as follows

$$
\begin{aligned}
u_t(x,y,z,t) =& f\left(u(x,y,z,t), u_x(x,y,z,t), u_{xx}(x,y,z,t), u_y(x,y,z,t),\right.\\
& u_{yy}(x,y,z,t), u_z(x,y,z,t), u_{zz}(x,y,z,t), u_{xy}(x,y,z,t),\\
& \left. u_{yx}(x,y,z,t), u_{xz}(x,y,z,t), u_{yz}(x,y,z,t)\right) + \xi(x,y,z,t)\\
\bar{u}^{i,j,k}(t) =& C^{i^*,j^*,k^*} u^{i^*,j^*,k^*}(t_k) \chi^{i^*,j^*,k^*}_{[t_k,t_{k+1})}, \text{ for } (i^*,j^*,k^*) \in \mathcal{I}
\end{aligned}
\tag{3.40}
$$

where $f(\cdot)$ is an unknown nonlinear partial differential equation and $u \in \Re^n$ is defined in the domain (2.2).The boundary (Neumann and Dirichlet) and initial conditions are given by (2.3). Set $\mathcal{I}$ is defined as in the previous section. The variable $\bar{u}^{i,j,k}(t)$ describes the real available sample–data measurements, i.e., the stepwise values of $\bar{u}^{i,j,k}(t)$ represents the available output, which are measurable only at points spread over the mesh. Here

$$
\chi^{i^*,j^*,k^*}_{[t_k,t_{k+1})} := \begin{cases} 1 & \text{if } t \in [t_k, t_{k+1}), \\ 0 & \text{otherwise.} \end{cases}
\tag{3.41}
$$

denotes the characteristic function of the time interval $[t_k, t_{k+1})$.

Thus, the problem that arises is that, given the sample–time scattered output $\bar{u}^{i,j,k}(t)$, provide an estimate $\bar{\bar{\bar{u}}}^{i,j,k}(t)$ of the missing information along the grid, based on the triangle-based linear interpolation method, that satisfies the following relation

$$
\left\| u^{i,j,k}_{\text{int},t_k} - u^{i,j,k}(t) \right\|^2 \leq \delta^{i,j,k}_{1,+}\left(h, \Delta x, \Delta y, \Delta z\right)
\tag{3.42}
$$

where $h := \max_k |t_{k+1} - t_k|$ and Δm is the is the maximum distance between points at coordinates x, y and z in mesh representation.. Additionally, provide an estimate of the upper bound of the identification error and, if it is possible, reduce it to the lowest possible value selecting free parameters participating in the DNN–identifier.

3.3.1 Identifier and learning laws

Based on the results from previous sections, let us propose the following identifier

$$\begin{gathered}\frac{d}{dt}\hat{u}^{i,j,k}(t) = A^{i,j,k}\hat{u}^{i,j,k}(t) + \sum_{r=1}^{11} W_r^{i,j,k}\Omega^r(x^i, y^j, z^k)\hat{U}^{(i,j,k),r}(t) \\ +L^{i,j,k}\left(\bar{\bar{\bar{u}}}^{i,j,k}(t) - C^{i,j,k}\hat{u}^{i,j,k}(t)\right) \\ i = \overline{0,N}; j = \overline{0,M}; k = \overline{0,L}\end{gathered} \tag{3.43}$$

where

$$\bar{\bar{\bar{u}}}^{i,j,k}(t) := C^{i,j,k}\hat{u}^{i,j,k}(t) - u^{i,j,k}_{\text{int},t_k}(t) \tag{3.44}$$

define the error trajectory between the estimate $\hat{u}^{i,j,k}(t)$ and the data obtained through the interpolation method. Additionally, consider the following quantities

$$\bar{\bar{\bar{u}}}^{i,j,k}_m(t) := \hat{u}^{i,j,k}_m(t) - \tfrac{\partial}{\partial m}u^{i,j,k}_{\text{int},t_k}(t) \tag{3.45}$$

where m represents the partial derivative respect to x, y or z, correspondingly. Let the time–varying matrices $\tilde{W}^{i,j,k}_r(t) \in \Re^n$, $r = \overline{1,11}$ satisfy the following nonlinear matrix differential equations

$$\begin{aligned}\dot{W}^{i,j,k}_r = &-\frac{a}{2}\tilde{W}^{i,j,k}_r - K_r^{-1}\Pi_b\bar{\bar{\bar{u}}}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}}\left(\Omega^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\ &-\sum_{l=1}^{3}K_r^{-1}S_l^{i,j,k}\bar{\bar{\bar{u}}}^{i,j,k}_m(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}}\left(\Omega^r_m(x^i, y^j, z^k)\right)^{\mathsf{T}} \\ &-\frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+61}\Pi_b W^{i,j,k}_r\Omega^r\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}\right)^{\mathsf{T}}(\Omega^r)^{\mathsf{T}} \\ &-\frac{1}{2}\sum_{l=1}^{3}K_r^{-1}S_l^{i,j,k}\Lambda_{r+72}S_l^{i,j,k}W^{i,j,k}_r\Omega^r_m\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}\right)^{\mathsf{T}}(\Omega^r_m)^{\mathsf{T}} \\ &-\frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+105}\Pi_b W^{i,j,k}_r\Omega^r\hat{U}^{(i,j,k),r}(t)\left[\hat{U}^{(i,j,k),r}\right]^{\mathsf{T}}(\Omega^r)^{\mathsf{T}}\end{aligned} \tag{3.46}$$

where m represents the partial derivative with respect to x for $l = 1$, with respect to y for $l = 2$ and with respect to z for $l = 3$. Matrices $S^{i,j,k}_1$, $S^{i,j,k}_2$ and $S^{i,j,k}_3$ $(i = \overline{1,N}; j = \overline{1,M}, k = \overline{1,L})$ are positive definite solutions of the following Riccati matrix inequalities

$$
\begin{aligned}
&\begin{bmatrix} -S_1^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\intercal} S_1^{i,j,k} - Q_{S_1}^{i,j,k} & S_1^{i,j,k}\left[R_{S_1}^{i,j,k}\right]^{1/2} \\ \left[R_{S_1}^{i,j,k}\right]^{1/2} S_1^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\
&R_{S_1}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+12}\left[\mathring{W}_r^{i,j,k}\right]^{\intercal} + \Lambda_{24} \\
&Q_{S_1}^{i,j,k} := aS_1^{i,j,k}
\end{aligned}
\tag{3.47}
$$

$$
\begin{aligned}
&\begin{bmatrix} -S_2^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\intercal} S_2^{i,j,k} - Q_{S_1}^{i,j,k} & S_2^{i,j,k}\left[R_{S_2}^{i,j,k}\right]^{1/2} \\ \left[R_{S_2}^{i,j,k}\right]^{1/2} S_2^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\
&R_{S_2}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+24}\left[\mathring{W}_r^{i,j,k}\right]^{\intercal} + \Lambda_{36} \\
&Q_{S_2}^{i,j,k} := aS_2^{i,j,k}
\end{aligned}
\tag{3.48}
$$

$$
\begin{aligned}
&\begin{bmatrix} -S_3^{i,j,k}A^{i,j,k} - \left[A^{i,j,k}\right]^{\intercal} S_3^{i,j,k} - Q_{S_1}^{i,j,k} & S_3^{i,j,k}\left[R_{S_3}^{i,j,k}\right]^{1/2} \\ \left[R_{S_3}^{i,j,k}\right]^{1/2} S_3^{i,j,k} & I_{n\times n} \end{bmatrix} > 0 \\
&R_{S_3}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+36}\left[\mathring{W}_r^{i,j,k}\right]^{\intercal} + \Lambda_{48} \\
&Q_{S_3}^{i,j,k} := aS_3^{i,j,k}
\end{aligned}
\tag{3.49}
$$

3.3.2 Main result

The main result of this section can be summarized in the following theorem

Theorem 4. Consider the nonlinear model (3.40), given by the system of PDEs with uncertainties in the states, and scattered sample–data outputs, with initial and boundary conditions given by (2.3). Suppose that the DNN–identifier is given by (3.43) and its parameters are adjusted by the learning laws (3.46). If there exist positive definite matrices $R_{S_1}^{i,j,k}$, $R_{S_2}^{i,j,k}$ and $R_{S_3}^{i,j,k}$ $(i = \overline{1,N}; j = \overline{1,M}; k = \overline{1,L})$ such that Riccati matrix inequalities (3.47)–(3.49) have a positive definite solutions $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ $(i = \overline{1,N}; j = \overline{1,M}; k = \overline{1,L})$, and if there exist matrices Π_b, Π_c, $P^{i,j,k}$, $L^{i,j,k}$ and parameter ϵ such that

$$W^{i,j,k} = \begin{pmatrix} \mathrm{w}_{11}^{i,j,k} & \mathrm{w}_{12}^{i,j,k} & 0 & 0 & \mathrm{w}_{15}^{i,j,k} \\ \mathrm{w}_{21}^{i,j,k} & \mathrm{w}_{22}^{i,j,k} & 0 & 0 & \mathrm{w}_{25}^{i,j,k} \\ 0 & 0 & \mathrm{w}_{33}^{i,j,k} & 0 & 0 \\ 0 & 0 & 0 & \mathrm{w}_{44}^{i,j,k} & 0 \\ \mathrm{w}_{51}^{i,j,k} & \mathrm{w}_{52}^{i,j,k} & 0 & 0 & \mathrm{w}_{55}^{i,j,k} \end{pmatrix} \tag{3.50}$$

whose elements are given as follows

$$\begin{aligned} \mathrm{w}_{11}^{i,j,k} :=& \Pi_b \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k} \right) + \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k} \right)^{\mathsf{T}} \Pi_b + \Pi_b R_{\Pi_b}^{i,j,k} \Pi_b \\ &+ Q_{\Pi_b}^{i,j,k} + S_4^{i,j,k} + a P^{i,j,k} \\ \mathrm{w}_{12}^{i,j,k} :=& P^{i,j,k} + \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k} \right)^{\mathsf{T}} \Pi_c - \Pi_b \\ \mathrm{w}_{21}^{i,j,k} :=& P^{i,j,k} + \Pi_c \left(A^{i,j,k} - L^{i,j,k} C^{i,j,k} \right) - \Pi_b \\ \mathrm{w}_{22}^{i,j,k} :=& h^2 R^{i,j,k} - 2\Pi_c + \Pi_c R_{\Pi_c}^{i,j,k} \Pi_c + Q_{\Pi_c}^{i,j,k} \\ \mathrm{w}_{33}^{i,j,k} :=& - h e^{-ah} R^{i,j,k} \\ \mathrm{w}_{44}^{i,j,k} :=& - e^{-ah} S_4^{i,j,k} \\ .\mathrm{w}_{55}^{i,j,k} :=& - \epsilon I;\ \mathrm{w}_{15}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{51}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{25}^{i,j,k} := -\Pi_c;\ \mathrm{w}_{52}^{i,j,k} := -\Pi_c \end{aligned} \tag{3.51}$$

is negative definite, then the error of identification $\tilde{u}^{i,j,k}(t)$ converges in practical sense to

$$\overline{\lim_{t\to\infty}} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \left\| \tilde{u}^{i,j,k}(t) \right\|_{S_2^l(G)}^2 \leq \beta / \alpha \tag{3.52}$$

where $\alpha := a$ and

$$
\begin{aligned}
\beta :=\ & \varpi_1 \overline{\sum} \sum_{s=1}^{4} f_s^{i,j,k} + \varpi_2 \overline{\sum} \left\| L^{i,j,k} \Delta u^{i,j,k} \right\|^2 + \varpi_3 \overline{\sum} \delta_{1,+}^{i,j,k} \left(h, \Delta x, \Delta y, \Delta z \right) \\
& + \varpi_4 \overline{\sum} \delta_{2,+}^{i,j,k} \left(h, \Delta x, \Delta y, \Delta z \right) + \varpi_5 \overline{\sum} \delta_{3,+}^{i,j,k} \left(h, \Delta x, \Delta y, \Delta z \right) + \varpi_6 \overline{\sum} \delta_{4,+}^{i,j,k} \left(h, \Delta x, \Delta y, \Delta z \right)
\end{aligned}
$$

$$
\varpi_1 := \max \left\{ \epsilon, \lambda_{\max} \left(\Lambda_{24}^{-1} \right), \lambda_{\max} \left(\Lambda_{32}^{-1} \right), \lambda_{\max} \left(\Lambda_{48}^{-1} \right) \right\}
$$

$$
\varpi_2 := \lambda_{\max} \left(\Lambda_{60}^{-1} \right) + \lambda_{\max} \left(\Lambda_{61}^{-1} \right)
$$

$$
\varpi_3 := \sum_{r=1}^{11} \lambda_{\max} \left(\Lambda_{r+61} \right) \quad \varpi_4 := \sum_{r=1}^{11} \lambda_{\max} \left(\Lambda_{r+72} \right)
$$

$$
\varpi_5 := \sum_{r=1}^{11} \lambda_{\max} \left(\Lambda_{r+83} \right) \quad \varpi_6 := \sum_{r=1}^{11} \lambda_{\max} \left(\Lambda_{r+94} \right)
$$

Proof. The proof of this theorem does not differ significantly from that made for theorem 1. The only difference lies in the fact that the approximation of the error of identification $\tilde{u}^{i,j,k}(t)$ is as follows

$$
\tilde{u}^{i,j,k}(t) := \bar{\bar{\bar{u}}}^{i,j,k}(t) + \Delta u_{\mathrm{int},t_k}^{i,j,k}
$$

This consideration changes the learning laws of the neural network as in (3.46) and provides a new upper bound for the identification error given by (3.52) □

Chapter 4

Numerical Results

To test the quality of the approximation of the solution of partial differential equations using the proposed neural network methodology, we present below several examples where they qualitatively illustrate the theory presented in the previous sections. At the end, we present a table comparing different techniques for the approximation of solution of partial differential equations using neural networks, which highlight its main features.

4.1 PDE with sample-time outputs

Let us consider the following 3D heat equation with heat source given by

$$\begin{aligned} u_t(t,x,y,z) &= -c_1 u_{xx}(t,x,y,z) - c_2 u_{yy}(t,x,y,z) - c_3 u_{zz}(t,x,y,z) + \xi(t,x,y,z) \\ \xi(t,x,y,z) &= a\sin(t + b_1 x + b_2 y + b_3 z) \end{aligned} \tag{4.1}$$

where $a = 0.1$, $b_1 = 1$, $b_2 = 0.5$, $b_3 = 0.7$ and $c_1 = c_2 = c_3 = 0.01$. This system can be represented (using the FDM) as

$$\begin{aligned} u_t^{i,j,k}(t) =& \frac{1}{\Delta^2}\left[-(c_1+c_2+c_3)u^{i,j,k}(t) + 2c_1 u^{i-1,j,k}(t) + 2c_2 u^{i,j-1,k}(t) \right. \\ & \left. + 2c_3 u^{i,j,k-1}(t) - c_1 u^{i-2,j,k} - c_2 u^{i,j-2,k}(t) - c_3 u^{i,j,k-2}(t)\right] \end{aligned}$$

Boundary and initial conditions were selected as

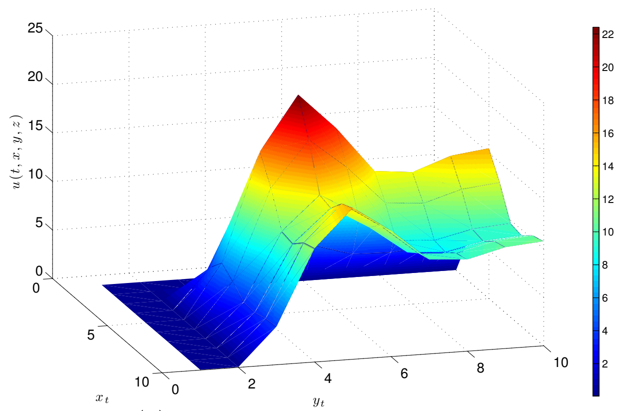
$$\begin{aligned} &u(0,y,z,t) = \mathrm{rand}(1) \quad u(x,0,z,t) = \mathrm{rand}(1) \\ &u(x,y,0,t) = \mathrm{rand}(1) \quad u_x(0,0,0,t) = 0 \quad u(x,y,z,0) = 10 \end{aligned}$$

where rand(1) is a series of random numbers in the interval $[0, 1]$. This condition is provided only to ensure that different initial and boundary conditions give similar approximations of the solution. The sampling time–intervals were selected to $t_k = 0.2s$ and the domain was divided in $10 \times 10 \times 10$ equidistant sections. In figure 4.1 we show a comparison between trajectories produced by the model (4.1) and the output from the identifier proposed after 10 s. In this graph, the z coordinate is hidden. After 10 seconds of simulation, the identifier has approached the solution of partial differential equation (4.1). In this comparison we can see small differences in some regions represented by the color scale. This is because the identifier ensures that the identification error converges to a region and it will remain bounded during the identification process. To clarify this, figure 4.2 shows the difference between the trajectories produced by the equation and the identifier at the time $10s$. The difference is near zero and remains bounded in the domain defined by the equation. Additional information is presented in figure 4.3. In the first graph we can see the comparison between the trajectories produced by the partial differential equation, the sample-time output and the identified output. These information were obtained by measuring directly the node $(6, 8, 9)$ of the mesh representation. We can see that the approximates solution stays close to the solution of the model. Figure 4.13(b) shows the logarithmic quadratic error for the simulation time. Finally, figure 4.4 shows a comparison between the PDE numerical solution and the solution approximated by the identifier. We used the mesh representation because it allows a better comparison node to node. This graph shows the coordinates x, y and z of the grid and the color scale determines the value of the functions $u(t, x, y, z)$ and $\hat{u}(t, x, y, z)$ at time 10s. It can be seen that both grid outputs are nearly identical. One can see that the outputs are almost identical illustrating the high efficiency of the identification algorithm

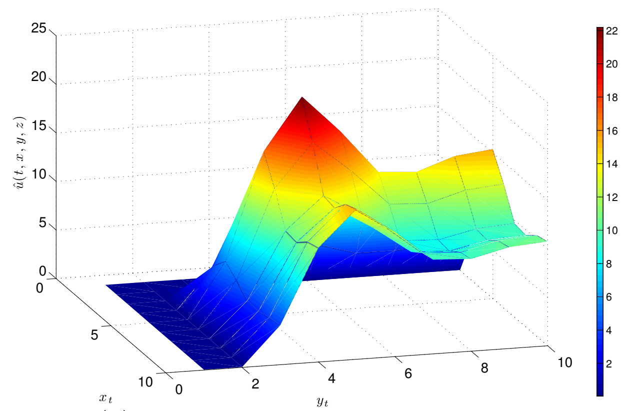
4.2 PDE with scattered outputs

Consider the 3D wave equation defined as

$$u_{tt}(t, x, y, z) = c^2 \nabla^2 u(t, x, y, z) \tag{4.2}$$



(a) PDE solution at time $10s$.



(b) DNN approximation at time $10s$.

Figure 4.1: Comparison between trajectories produced by PDE and the DNN-identifier after $10s$.

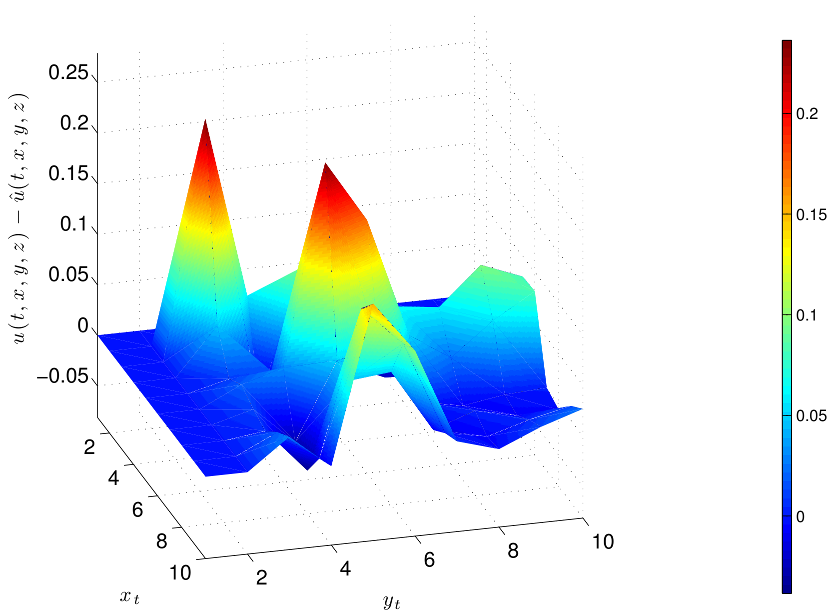
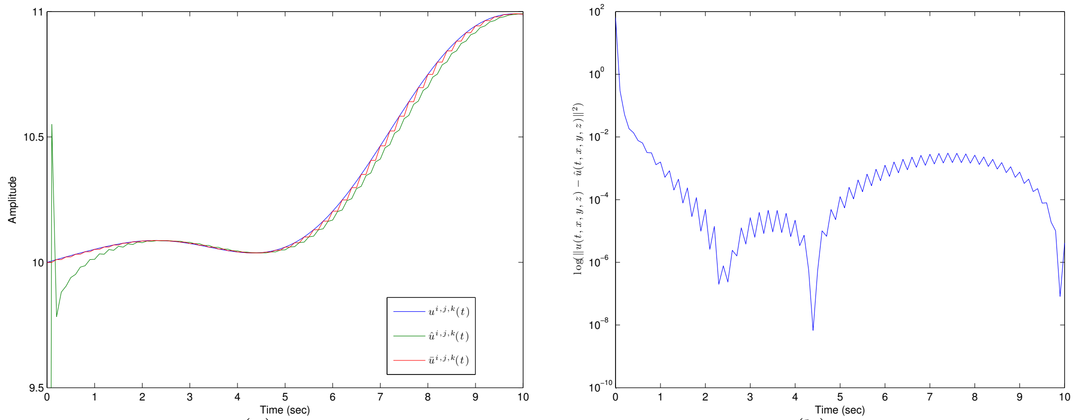
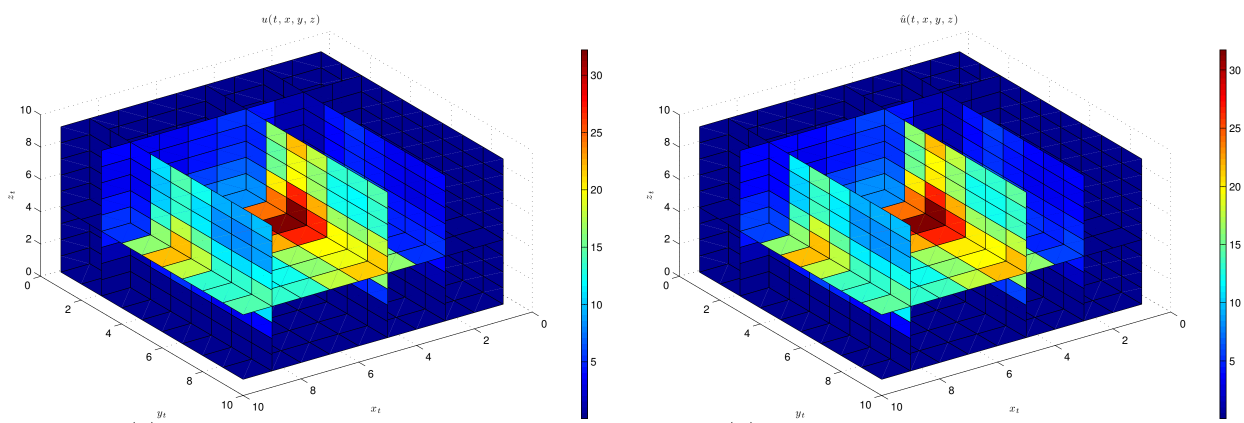


Figure 4.2: Difference between PDE solution and DNN approximation at time $10s$.



(a) (b)

Figure 4.3: (a) Comparison between trajectories produced by PDE solution, sampled-time output and the DNN-identifier output for fixed x, y and z coordinates. (b) Logarithmic quadratic error for fixed x, y and z coordinates.



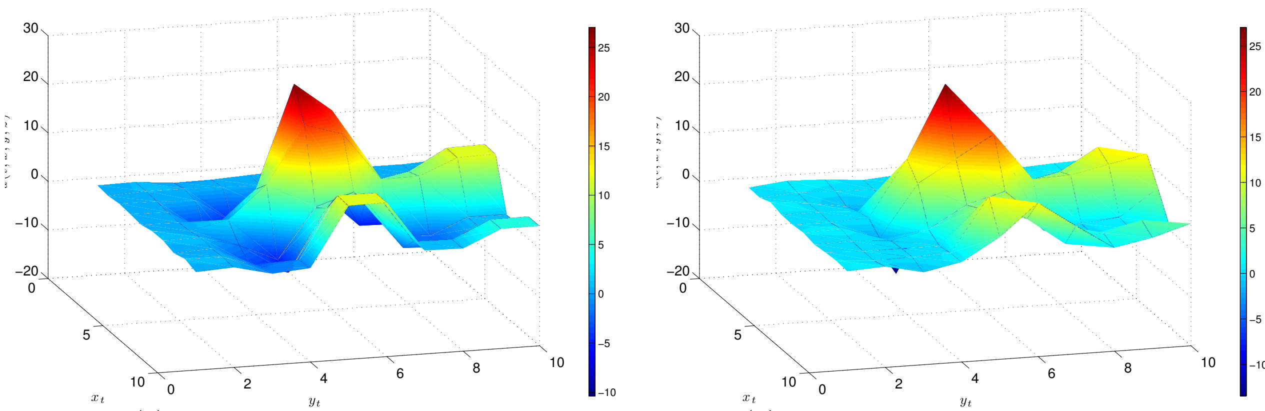
(a) PDE solution at time $10s$. (b) DNN approximation at time $10s$.

Figure 4.4: Comparison between grid outputs produced by PDE and the DNN-identifier after $10s$.

where $c^2 = 0.1$ and ∇^2 is the Laplace operator. The FDM representation of this equation is

$$u_{tt}^{i,j,k}(t) = \frac{1}{\Delta^2}\left[3cu^{i,j,k}(t) - 2c(u^{i-1,j,k}(t) + u^{i,j-1,k}(t) + u^{i,j,k-1}(t)) \right.$$
$$\left. + c(u^{i-2,j,k}(t) + u^{i,j-2,k}(t) + u^{i,j,k-2}(t))\right]$$

The initial and boundary conditions are the same as in (4.1) and the domain was divided in $10 \times 10 \times 10$ equidistant sections. The process to implement this simulation is as follows. To produce outputs in a scattered grid, we have assumed that the measurements are available only in some nodes, i.e., data is available at nodes whose sum of indices i, j, k is even. For this particular simulation, this represents 52% of the total data. Next, we used the *griddata* method of Matlab$^{\text{TM}}$ for interpolation of missing information. This method implements the triangular-based linear surface fitting algorithm and provides an interpolation that satisfies condition (3.31). With these new data, we use the identifier (3.29) with learning laws (3.32) to approximate the solution of (4.2). In figure 4.5 it is shown the comparison between the PDE numerical solution and the DNN-identifier output using the interpolated data for its learning process. Trajectories are quite similar, but the error graph shown in figure 4.6 demonstrate that there are areas where the difference between the model and its approximation is large. These major differences are the result of the interpolation algorithm, because it has approximately fifty percent of the total information for this process. To avoid these results is necessary to measure more mesh nodes in order to obtain a better approximation. In figure 4.7(a) we can see the comparison between trajectories produced by the 3D wave equation, the identifier output and the interpolated trajectory. It can be seen that the trajectories of the equation and interpolation are practically the same, however the output of the identifier is different from these. This is because the learning process uses information from the neighbors in the mesh, and if the interpolation process is deficient in some nodes, errors are propagated through these. In figure 4.7(b) it is shown the logarithmic quadratic error for the node (7,9,7) in the mesh representation. In figure 4.8, we present the comparison between the output grids of PDE and DNN identifier, respectively.



(a) PDE solution at time $10s$.
(b) DNN approximation at time $10s$.

Figure 4.5: Comparison between trajectories produced by PDE and the DNN-identifier after $10s$.

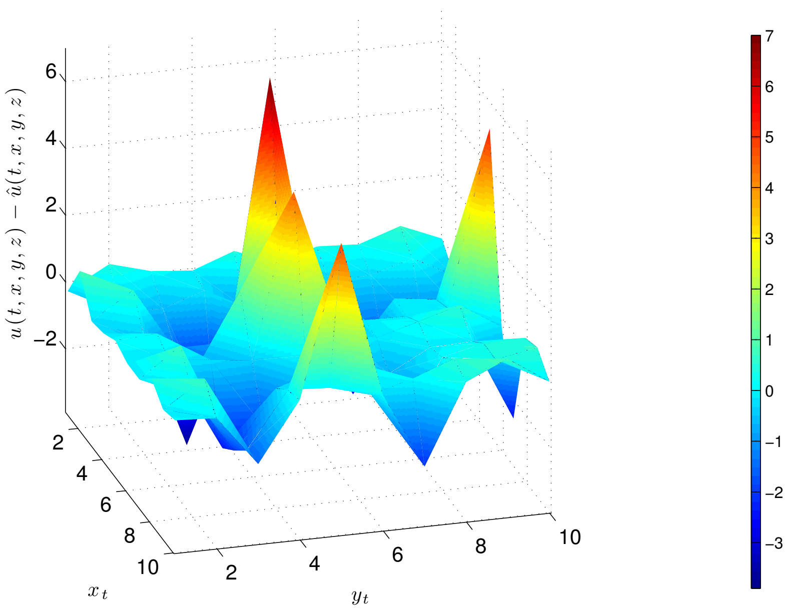
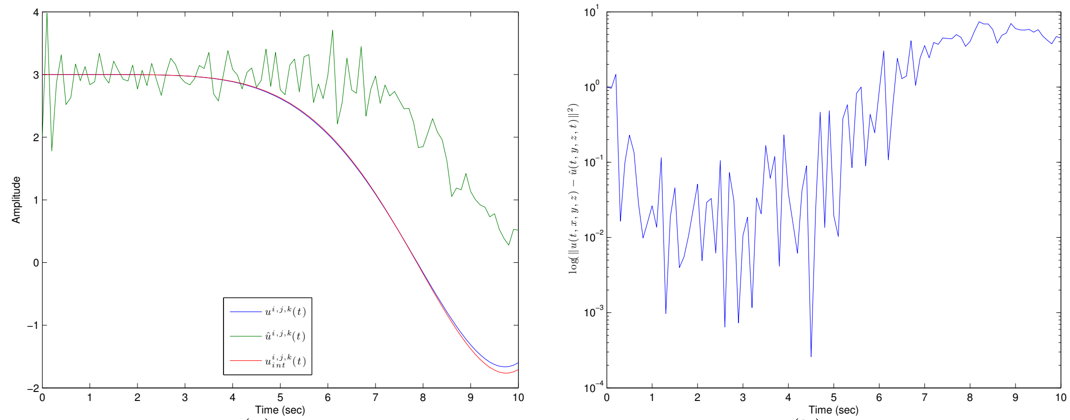
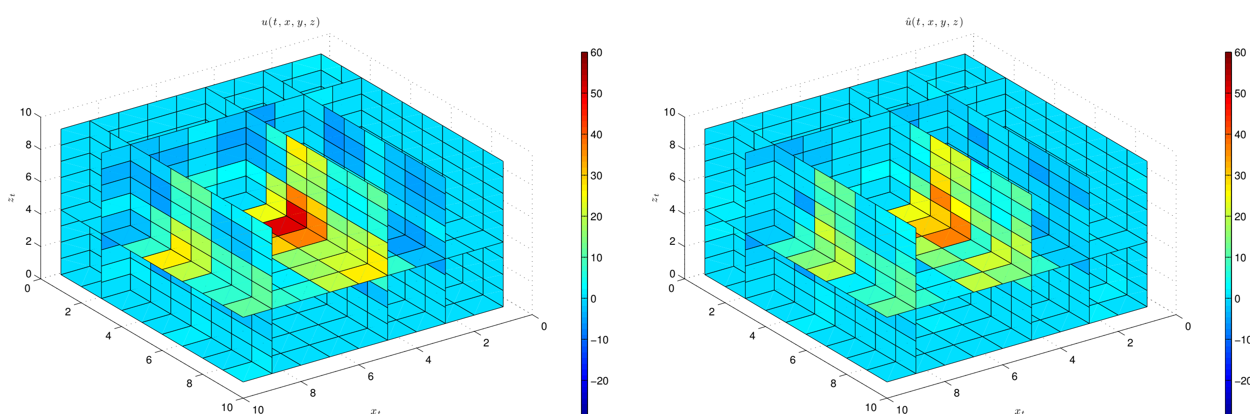


Figure 4.6: Difference between PDE solution and DNN approximation at time $10s$.



(a) (b)

Figure 4.7: (a) Comparison between trajectories produced by PDE solution, interpolation output and the DNN-identifier output for fixed x, y and z coordinates. (b) Logarithmic quadratic error for fixed x, y and z coordinates.



(a) PDE solution at time $10s$. (b) DNN approximation at time $10s$.

Figure 4.8: Comparison between grid outputs produced by PDE and the DNN-identifier after $10s$.

4.3 PDE with sample-time scattered outputs

Consider the 3D wave equation defined as

$$u_{tt}(t,x,y,z) = c^2 \nabla^2 u(t,x,y,z) \tag{4.3}$$

with boundary and initial conditions as in (4.1). Here $c = -0.1$ and ∇^2 is the Laplace operator. The FDM representation of this equation is

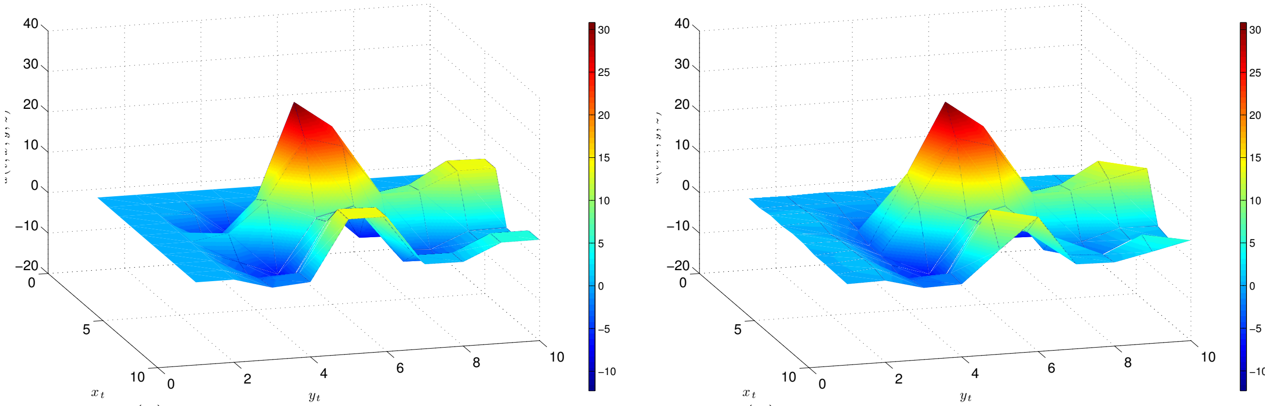
$$\begin{aligned} u_{tt}^{i,j,k}(t) =& \frac{1}{\Delta^2}\left[3cu^{i,j,k}(t) - 2c(u^{i-1,j,k}(t) + u^{i,j-1,k}(t) + u^{i,j,k-1}(t))\right. \\ & \left. + c(u^{i-2,j,k}(t) + u^{i,j-2,k}(t) + u^{i,j,k-2}(t))\right] \end{aligned}$$

Assume that the given output is

$$\bar{u}^{i,j,k}(t) = u^{i^*,j^*,k^*}(t_k)\chi_{[t_k,t_{k+1})}$$

where $t_k = 0.2s$ is the sample time and $\chi_{[t_k,t_{k+1})}$ is the characteristic function of the time interval. Indices i^*, j^* and k^* represent the measurable nodes in the grid representation. For this particular simulation we assume that nodes whose sum of indices is even *or* $(i+j+k)$ mod $(3) = 0$ are available. With this assumption we have 73 percent of all the information to perform interpolation, which will increase its quality, and consequently, the quality of the approximation of the solution of (4.3). Following the same procedure as in the previous section, we provide an interpolation of the missing data through the Matlab™ function *griddata*, which satisfies relation (3.42).

Figure 4.9 shows the comparison between trajectories produced by the mathematical model and the DNN identifier. As in the previous case, the trajectories are very similar, but unlike that case, the error between them is close to zero, except in certain well-identified points (see fig. 4.11). In figure 4.12 are shown a comparison between trajectories involved in the identification process and the logarithmic quadratic error fixed for the node $(8,8,7)$. In figure 4.10, it is shown the comparison between the output grids of PDE and DNN identifier, respectively.



(a) PDE solution at time $10s$. (b) DNN approximation at time $10s$.

Figure 4.9: Comparison between trajectories produced by PDE and the DNN-identifier after $10s$.

4.4 State estimation of PDE

Consider the telegraph equation [26] given by

$$u_{tt} + (\alpha + \beta)u_t + \alpha\beta u = c^2 u_{xx} \tag{4.4}$$

where

$$c^2 = \frac{1}{LC} \qquad \alpha = \frac{G}{C} \qquad \beta = \frac{R}{L}$$

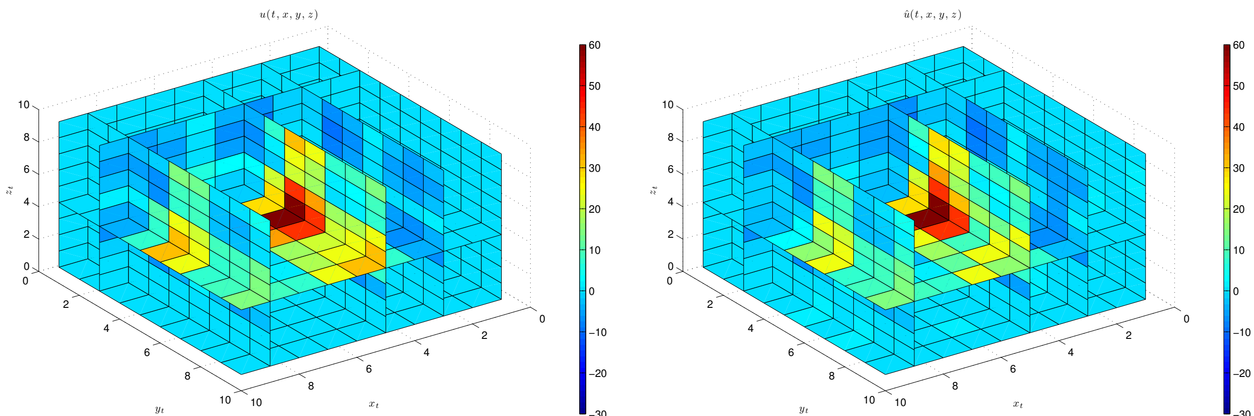
Here, $u(x,t)$ denotes the voltage at position x and time t. C stands for capacitance to ground, L and R are the inductance and resistance of the cable, respectively, and G denotes the conductance to ground. For a case illustrating the ability to estimate unknown states using the identifier proposed in equation (3.2), we use the following benchmark system,

$$\begin{aligned} u_{tt} &= u_{xx} - Au_t - Bu \\ \bar{u}^{i,j,k}(t) &= Cu^{i,j,k}(t_k)\chi^{i,j,k}_{[t_k,t_{k+1})} \end{aligned} \tag{4.5}$$

where

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.6}$$

which represents two transmission lines of which we can measure voltage at the first one. We assume further that the output is only available through a process of sampling with period



(a) PDE solution at time $10s$. (b) DNN approximation at time $10s$.

Figure 4.10: Comparison between grid outputs produced by PDE and the DNN-identifier after $10s$.

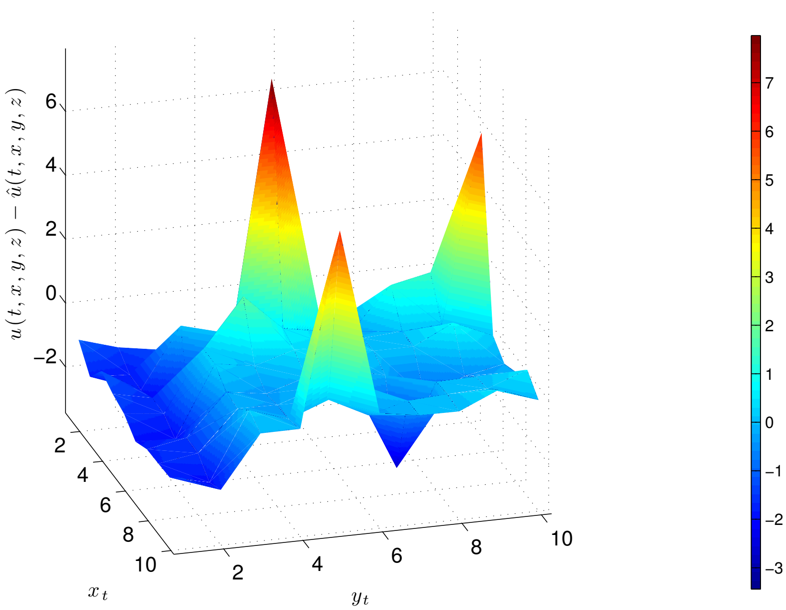
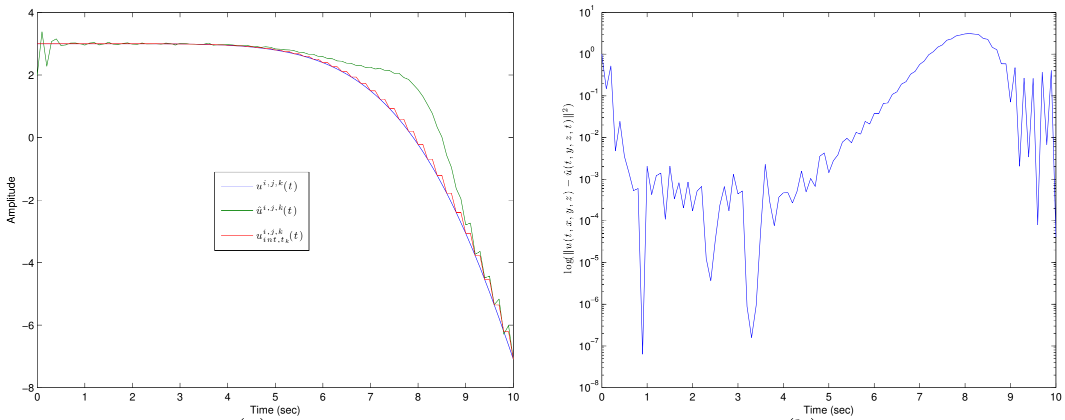


Figure 4.11: Difference between PDE solution and DNN approximation at time $10s$.



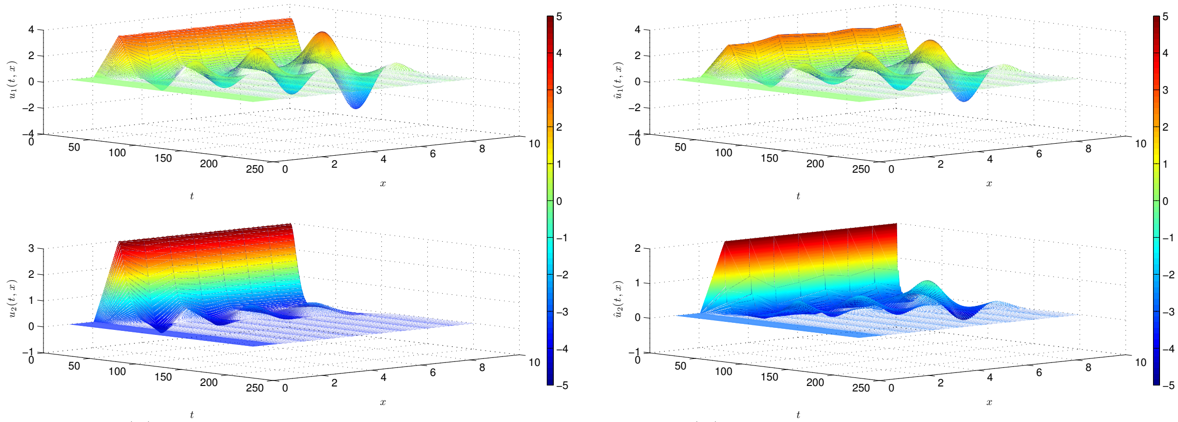
(a) (b)

Figure 4.12: (a) Comparison between trajectories produced by PDE solution, sampled-time scattered output and the DNN-identifier output for fixed x, y and z coordinates. (b) Logarithmic quadratic error for fixed x, y and z coordinates.

$t_k = 0.2s$. In figure 4.13, it is shown a comparison between trajectories produced by the mathematical model and the DNN identifier. The major differences between them are result of the period of learning process. Figure 4.14 allows a better visualization of the process of estimating the trajectories of the system of PDE. In figure 4.14(a), we see a direct comparison between the trajectories produced by the model and the neural network. Moreover, in figure 4.14(b) it is shown the graph of the error between the model and the neural network. With this we can conclude that the estimation process is satisfactory from the known information.

4.5 Comparative chart

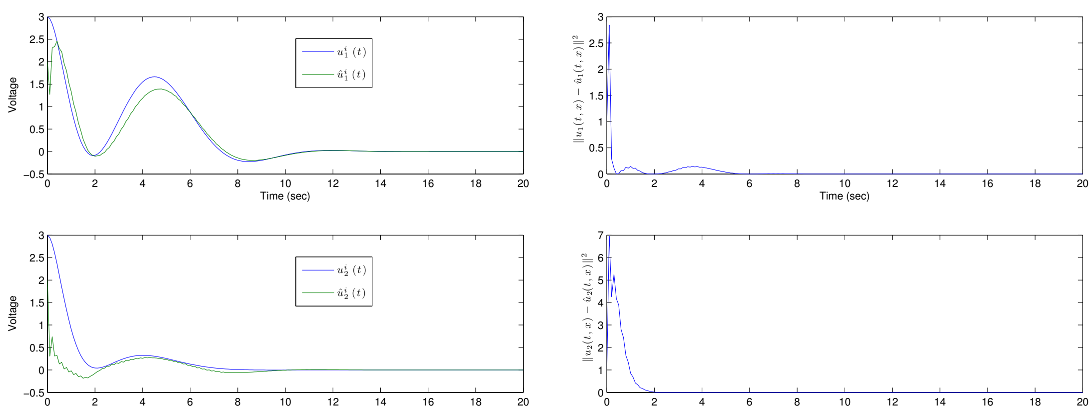
In this section, simulations were developed to illustrate the effectiveness of the proposed algorithms, however, these are not the only ones available. Table 4.1 shows some of the proposals available for approximating solutions of unknown partial differential equations via neural network techniques, highlighting its main characteristics. Thus, the reader will be able to discern and choose the option that best suits its problem.



(a) PDE solution after $20s$.

(b) DNN approximation after $20s$.

Figure 4.13: Comparison between trajectories produced by PDE and the DNN-identifier after $20s$.



(a)

(b)

Figure 4.14: (a)Comparison between trajectories produced by PDE and the DNN-identifier after $20s$ and (b) error trajectory for a fixed x.

Method \ Feature	Technique	Uncertain PDE	Sample-time outputs	Irregular mesh	Complex boundary conditions	State estimation	Accuracy
Chairez et al. [5]	DNN, FDM.	X					High [a]
Lagaris et al. [23],[24]	FFNN, collocation method, trial function.			X	X		High [b]
Beidokhti et al. [3]	MLP, trial function, minimization.				X		High [c]
Shirvany et al. [33]	MLP, RBF, collocation method.			X	X		High [b,d]
Algorithm §3.1	DNN, FDM, descriptor.	X	X			X	Medium to High [a,e]
Algorithm §3.2	DNN, FDM, surface fitting.	X		X			Low to High [a,f]
Algorithm §3.3	DNN, FDM, descriptor, surface fitting.	X	X	X		X	Low to High [a,e,f]

Table 4.1: Comparative chart

[a] Comparing results with FDM approximation solution.
[b] Comparing results with FEM approximation solution.
[c] Comparing results with exact solution.
[d] Comparing results with Runge-Kutta approximation solution.
[e] Accuracy greatly depends on the maximum sample period time.
[f] Accuracy greatly depends on the surface fitting algorithm.

Chapter 5

Conclusions

This thesis presents a new robust adaptive DNN strategy for identifying and approximating the solution of partial differential equations with uncertainty, measured in different ways along a grid that divides the domain of definition of the PDE. We present three different cases for identification. First, the case where information for the identification process is available across the grid and the outputs are sampled over time. Second, the case where information is measurable in some scattered points on the grid. Finally, a combination of both previous cases. For all cases, we demonstrate the practical stability for the approximation process via Lyapunov-like analysis. We provide and construct an upper bound for identification error and, using the same analysis, we derive the learning laws for the suggested neural networks. Additionally, we construct numerical simulations that illustrate the efficiency of the proposed methodology. In the first case, it illustrates the convergence error to a bounded region identification. In the second case, we provide an interpolation of scattered data, however, the quality of the interpolation determines the quality of the approximation of the solution, since small interpolation errors are dispersed over the mesh due to the relationship with its neighbors expressed in learning laws. A third simulation was conducted to illustrate the two previous cases. We note in it that the error converges to a bounded region, with some slight variations due to the interpolation scheme.

Future work includes implementing the identification algorithm for the approximation of real distributed parameter systems as models of distribution of oil and water, where information is available just like in the third case, as well as improve the interpolation algorithms, since the quality identification depends directly on it.

Appendix A

Proofs

A.1 Proof of theorem 1

Proof. Let us consider the following Lyapunov–Krasovskii functional, defined as the composition of NML individual Lyapunov–Krasovskii functionals along the whole space:

$$
\begin{aligned}
V(t) :=& \overline{\sum} \left[\bar{V}^{i,j,k}(t) + \sum_{r=1}^{11} \operatorname{tr} \left\{ \left[\tilde{W}_r^{i,j,k}(t) \right]^\top K_r \tilde{W}_r^{i,j,k}(t) \right\} \right. \\
&+ \int_{t-h}^{t} e^{a(s-t)} \left[\tilde{u}^{i,j,k}(s) \right]^\top S_4^{i,j,k} \tilde{u}^{i,j,k}(s) ds \\
&\left. + h \int_{\theta=-h}^{0} \int_{t+\theta}^{t} e^{a(s-t)} \left[\tilde{u}_t^{i,j,k}(s) \right]^\top R^{i,j,k} \tilde{u}_t^{i,j,k}(s) ds d\theta \right] \\
\bar{V}^{i,j,k}(t) :=& \left\| \tilde{u}^{i,j,k}(t) \right\|_{P^{i,j,k}}^2 + \left\| \tilde{u}_x^{i,j,k}(t) \right\|_{S_1^{i,j,k}}^2 + \left\| \tilde{u}_y^{i,j,k}(t) \right\|_{S_2^{i,j,k}}^2 + \left\| \tilde{u}_z^{i,j,k}(t) \right\|_{S_3^{i,j,k}}^2 \\
\overline{\sum} :=& \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}
\end{aligned} \tag{A.1}
$$

where $S_4^{i,j,k}$ and $R^{i,j,k}$ are positive definite symmetrical matrices, $h := \max_k |t_{k+1} - t_k|$ and $a > 0$. The time derivative $\dot{V}(\cdot)$ of $V(\cdot)$ can be obtained and is given by the following relation

$$
\begin{aligned}
\dot{V}(t) &= 2\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} P^{i,j,k}\frac{d}{dt}\tilde{u}^{i,j,k}(t) + 2\overline{\sum}\left[\tilde{u}_x^{i,j,k}(t)\right]^{\mathsf{T}} S_1^{i,j,k}\frac{d}{dt}\tilde{u}_x^{i,j,k}(t) \\
&+ 2\overline{\sum}\left[\tilde{u}_y^{i,j,k}(t)\right]^{\mathsf{T}} S_2^{i,j,k}\frac{d}{dt}\tilde{u}_y^{i,j,k}(t) + 2\overline{\sum}\left[\tilde{u}_z^{i,j,k}(t)\right]^{\mathsf{T}} S_3^{i,j,k}\frac{d}{dt}\tilde{u}_z^{i,j,k}(t) \\
&+ 2\overline{\sum}\sum_{r=1}^{11}\operatorname{tr}\left\{\left[\tilde{W}_r^{i,j,k}(t)\right]^{\mathsf{T}} K_r\dot{W}_r^{i,j,k}(t)\right\} + \left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} S_4^{i,j,k}\tilde{u}^{i,j,k}(t) \\
&- e^{-ah}\left[\tilde{u}^{i,j,k}(t-h)\right]^{\mathsf{T}} S_4^{i,j,k}\tilde{u}^{i,j,k}(t-h) + h^2\left[\tilde{u}_t^{i,j,k}(t)\right] R^{i,j,k}\tilde{u}_t^{i,j,k}(t) \\
&- a\int_{t-h}^{t} e^{a(s-t)}\left[\tilde{u}^{i,j,k}(s)\right]^{\mathsf{T}} S_4^{i,j,k}\tilde{u}^{i,j,k}(s)ds \\
&- h\int_{t-h}^{t} e^{a(s-t)}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}_t^{i,j,k}(s) \\
&- ah\int_{\theta=-h}^{0}\int_{t+\theta}^{t} e^{a(s-t)}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}^{i,j,k}(s)dsd\theta
\end{aligned}
\tag{A.2}
$$

The tenth term of the time derivative of V(t) can be estimated as

$$
-h\int_{t-h}^{t} e^{a(s-t)}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}_t^{i,j,k}(s) \le -he^{-ah}\int_{t-h}^{t}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}_t^{i,j,k}(s) \tag{A.3}
$$

Applying the Jensen's inequality to the last integrals we get

$$
he^{-ah}\int_{t-h}^{t}\left[\tilde{u}^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}^{i,j,k}(s)ds \ge he^{-ah}\int_{t-h}^{t}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} dsR^{i,j,k}\int_{t-h}^{h}\left[\tilde{u}^{i,j,k}(s)\right]^{\mathsf{T}} ds \tag{A.4}
$$

Now, adding $\alpha V(t)$ to both sides of (A.2) we obtain

$$\begin{aligned}
\dot{V}(t) + \alpha V(t) \leq\ & 2\overline{\sum} \left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} P^{i,j,k} \frac{d}{dt}\tilde{u}^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_x^{i,j,k}(t)\right]^{\mathsf{T}} S_1^{i,j,k} \frac{d}{dt}\tilde{u}_x^{i,j,k}(t) \\
& + 2\overline{\sum} \left[\tilde{u}_y^{i,j,k}(t)\right]^{\mathsf{T}} S_2^{i,j,k} \frac{d}{dt}\tilde{u}_y^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_z^{i,j,k}(t)\right]^{\mathsf{T}} S_3^{i,j,k} \frac{d}{dt}\tilde{u}_z^{i,j,k}(t) \\
& + 2\overline{\sum} \sum_{r=1}^{11} \operatorname{tr}\left\{ \left[\tilde{W}_r^{i,j,k}(t)\right]^{\mathsf{T}} K_r \dot{W}_r^{i,j,k}(t) \right\} + \left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} S_4^{i,j,k} \tilde{u}^{i,j,k}(t) \\
& - e^{-ah} \left[\tilde{u}^{i,j,k}(t-h)\right]^{\mathsf{T}} S_4^{i,j,k} \tilde{u}^{i,j,k}(t-h) + h^2 \left[\tilde{u}_t^{i,j,k}(t)\right] R^{i,j,k} \tilde{u}_t^{i,j,k}(t) \\
& - a \int_{t-h}^{t} e^{a(s-t)} \left[\tilde{u}^{i,j,k}(s)\right]^{\mathsf{T}} S_4^{i,j,k} \tilde{u}^{i,j,k}(s) ds \\
& - he^{-ah} \int_{t-h}^{t} \left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} ds R^{i,j,k} \int_{t-h}^{h} \left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} ds \\
& - ah \int_{\theta=-h}^{0} \int_{t+\theta}^{t} e^{a(s-t)} \left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k} \tilde{u}^{i,j,k}(s) ds d\theta \\
& + \alpha \overline{\sum} \bar{V}^{i,j,k}(t) + \alpha \overline{\sum} \sum_{r=1}^{11} \operatorname{tr}\left\{ \left[\tilde{W}_r^{i,j,k}(t)\right]^{\mathsf{T}} K_r \tilde{W}_r^{i,j,k}(t) \right\} \\
& + \alpha \int_{t-h}^{t} e^{a(s-t)} \left[\tilde{u}^{i,j,k}(s)\right]^{\mathsf{T}} S_4^{i,j,k} \tilde{u}^{i,j,k}(s) ds \\
& + \alpha h \int_{\theta=-h}^{0} \int_{t+\theta}^{t} e^{a(s-t)} \left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} R^{i,j,k} \tilde{u}^{i,j,k}(s) ds d\theta
\end{aligned} \tag{A.5}$$

Let $\alpha := a$. Then, the right hand side of the last inequality can be simplified as

$$\begin{aligned}\dot{V}(t) + \alpha V(t) \leq{}& 2\overline{\sum} \left[\tilde{u}^{i,j,k}(t)\right]^{\top} P^{i,j,k} \frac{d}{dt}\tilde{u}^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_x^{i,j,k}(t)\right]^{\top} S_1^{i,j,k} \frac{d}{dt}\tilde{u}_x^{i,j,k}(t) \\ &+ 2\overline{\sum} \left[\tilde{u}_y^{i,j,k}(t)\right]^{\top} S_2^{i,j,k} \frac{d}{dt}\tilde{u}_y^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_z^{i,j,k}(t)\right]^{\top} S_3^{i,j,k} \frac{d}{dt}\tilde{u}_z^{i,j,k}(t) \\ &+ 2\overline{\sum}\sum_{r=1}^{11} \mathrm{tr}\left\{\left[\tilde{W}_r^{i,j,k}(t)\right]^{\top} K_r \dot{W}_r^{i,j,k}(t)\right\} + \left[\tilde{u}^{i,j,k}(t)\right]^{\top} S_4^{i,j,k}\tilde{u}^{i,j,k}(t) \\ &- e^{-ah}\left[\tilde{u}^{i,j,k}(t-h)\right]^{\top} S_4^{i,j,k}\tilde{u}^{i,j,k}(t-h) + h^2\left[\tilde{u}_t^{i,j,k}(t)\right] R^{i,j,k}\tilde{u}_t^{i,j,k}(t) \\ &- a\int_{t-h}^{t} e^{a(s-t)}\left[\tilde{u}^{i,j,k}(s)\right]^{\top} S_4^{i,j,k}\tilde{u}^{i,j,k}(s)ds \\ &- he^{-ah}\int_{t-h}^{t}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\top} ds R^{i,j,k}\int_{t-h}^{h}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\top} ds \\ &+ a\overline{\sum}\bar{V}^{i,j,k}(t) + a\overline{\sum}\sum_{r=1}^{11}\mathrm{tr}\left\{\left[\tilde{W}_r^{i,j,k}(t)\right]^{\top} K_r\tilde{W}_r^{i,j,k}(t)\right\}\end{aligned} \tag{A.6}$$

Let Π_b and Π_c be some matrices of the suitable dimension. Following the idea of the "descriptor method" for systems with time–delays, we know consider the term

$$\begin{aligned}&2\overline{\sum}\left(\left[\tilde{u}^{i,j,k}(t)\right]^{\top}\Pi_b + \left[\tilde{u}_t^{i,j,k}(t)(t)\right]^{\top}\Pi_c\right) \times \Big[A^{i,j,k}\tilde{u}^{i,j,k}(t) + \tilde{W}_1^{i,j,k}\sigma^1\hat{u}^{i,j,k}(t) + \tilde{W}_2^{i,j,k}\varphi^1\hat{u}^{i-1,j,k}(t) \\ &+ \tilde{W}_3^{i,j,k}\gamma^1\hat{u}^{i-2,j,k}(t) + \tilde{W}_4^{i,j,k}\varphi^2\hat{u}^{i,j-1,k}(t) + \tilde{W}_5^{i,j,k}\gamma^2\hat{u}^{i,j-2,k}(t) + \tilde{W}_6^{i,j,k}\varphi^3\hat{u}^{i,j,k-1}(t) \\ &+ \tilde{W}_7^{i,j,k}\gamma^3\hat{u}^{i,j,k-2}(t) + \tilde{W}_8^{i,j,k}\psi^1\hat{u}^{i-1,j-1,k}(t) + \tilde{W}_9^{i,j,k}\psi^2\hat{u}^{i,j-1,k-1}(t) + \tilde{W}_{10}^{i,j,k}\psi^3\hat{u}^{i-1,j,k-1}(t) \\ &+ \tilde{W}_{11}^{i,j,k}\sigma^2\hat{u}^{i-1,j-1,k-1}(t) + \mathring{W}_1^{i,j,k}\sigma^1\tilde{u}^{i,j,k}(t) + \mathring{W}_2^{i,j,k}\varphi^1\tilde{u}^{i-1,j,k}(t) + \mathring{W}_3^{i,j,k}\gamma^1\tilde{u}^{i-2,j,k}(t) \\ &+ \mathring{W}_4^{i,j,k}\varphi^2\tilde{u}^{i,j-1,k}(t) + \mathring{W}_5^{i,j,k}\gamma^2\tilde{u}^{i,j-2,k}(t) + \mathring{W}_6^{i,j,k}\varphi^3\tilde{u}^{i,j,k-1}(t) + \mathring{W}_7^{i,j,k}\gamma^3\tilde{u}^{i,j,k-2}(t) \\ &+ \mathring{W}_8^{i,j,k}\psi^1\tilde{u}^{i-1,j-1,k}(t) + \mathring{W}_9^{i,j,k}\psi^2\tilde{u}^{i,j-1,k-1}(t) + \mathring{W}_{10}^{i,j,k}\psi^3\tilde{u}^{i-1,j,k-1}(t) + \mathring{W}_{11}^{i,j,k}\sigma^2\tilde{u}^{i-1,j-1,k-1}(t) \\ &-\tilde{f}^{i,j,k}(t) + L^{i,j,k}\left(\bar{u}^{i,j,k}(t) - C\hat{u}^{i,j,k}(t)\right) - \tilde{u}_t^{i,j,k}(t)\Big]\end{aligned} \tag{A.7}$$

Adding equations (A.6) and (A.7), we have that the terms $2\left[\tilde{u}_m^{i,j,k}(t)\right]^{\mathsf{T}} S_l^{i,j,k} \frac{d}{dt}\tilde{u}_m^{i,j,k}(t)$ can be estimated as

$$\begin{aligned}
2\left[\tilde{u}_m^{i,j,k}(t)\right]^{\mathsf{T}} S_l^{i,j,k} \frac{d}{dt}\tilde{u}_m^{i,j,k}(t) \leq{}& \left[\tilde{u}_m^{i,j,k}(t)\right]^{\mathsf{T}} \left(S_l^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right]^{\mathsf{T}} S_l^{i,j,k}\right) \left[\tilde{u}_m^{i,j,k}(t)\right] \\
&+ \left[\tilde{u}_m^{i,j,k}(t)\right]^{\mathsf{T}} S_l^{i,j,k} \left(\sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+b} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{b+12}\right) S_l^{i,j,k} \left[\tilde{u}_m^{i,j,k}(t)\right] \\
&+ \sum_{r=1}^{11} \left[\tilde{U}^{(i,j,k),r}(t)\right] \left\|\Omega_m^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+b}^{-1}} \left[\tilde{U}^{(i,j,k),r}(t)\right] \\
&+ \left[\tilde{f}_m^{i,j,k}(t)\right]^{\mathsf{T}} \Lambda_{b+12}^{-1} \left[\tilde{f}_m^{i,j,k}(t)\right] \\
&+ 2\sum_{r=1}^{11} \left[\tilde{u}_m^{i,j,k}(t)\right]^{\mathsf{T}} S_l^{i,j,k} \tilde{W}_r^{i,j,k} \Omega_m^r(x^i, y^j, z^k) \hat{U}^{(i,j,k),r}(t)
\end{aligned} \tag{A.8}$$

where $l = 1, 2, 3$; m represents the partial derivate, for S_1 it is respect to x, for S_2 with respect to y and for S_3 with respect to z; and $b = 12, 24, 36$, correspondingly. The learning laws can be derived by grouping all terms containing $\tilde{W}_r^{i,j,k}$

$$\begin{aligned}
2\overline{\sum} \sum_{r=1}^{11} \operatorname{tr}\Bigg\{ &\left[\tilde{W}_r^{i,j,k}(t)\right]^{\mathsf{T}} \Bigg(K_r \dot{W}_r^{i,j,k} + \left[\Pi_b \tilde{u}^{i,j,k}(t) + \Pi_c \tilde{u}_t^{i,j,k}(t)\right] \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_m^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\
&S_1^{i,j,k} \tilde{u}_x^{i,j,k}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_x^r(x^i, y^j, z^k)\right)^{\mathsf{T}} + S_2^{i,j,k} \tilde{u}_y^{i,j,k}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_y^r(x^i, y^j, z^k)\right)^{\mathsf{T}} \\
&S_3^{i,j,k} \tilde{u}_z^{i,j,k}(t) \left(\hat{U}^{(i,j,k),r}(t)\right)^{\mathsf{T}} \left(\Omega_z^r(x^i, y^j, z^k)\right)^{\mathsf{T}} + \frac{a}{2} K_r \tilde{W}_r^{i,j,k} \Bigg) \Bigg\}
\end{aligned} \tag{A.9}$$

Now, let approximate $\tilde{u}^{i,j,k}(t)$ as

$$\tilde{u}^{i,j,k}(t) = \hat{u}^{i,j,k}(t) - u^{i,j,k}(t) \pm \bar{u}^{i,j,k}(t) = u_e^{i,j,k}(t) + \Delta u^{i,j,k} \tag{A.10}$$

where $u_e^{i,j,k}(t) := \hat{u}^{i,j,k}(t) - \bar{u}^{i,j,k}(t)$ and $\Delta u^{i,j,k} = \bar{u}^{i,j,k}(t) - u^{i,j,k}(t)$. For the terms containing $\tilde{u}_x^{i,j,k}(t)$ they can be approximated as

$$\tilde{u}_x^{i,j,k}(t) = u_{e_x}^{i,j,k}(t) + \Delta u_x^{i,j,k} \tag{A.11}$$

where

$$
\begin{aligned}
u_{e_x}^{i,j,k}(t) &:= \frac{1}{\Delta x}\left[u_e^{i,j,k}(t) - u_e^{i-1,j,k}(t)\right] \\
\Delta u_x^{i,j,k} &:= \frac{1}{\Delta x}\left[\Delta u^{i,j,k} - \Delta u^{i-1,j,k}\right]
\end{aligned}
\tag{A.12}
$$

Remark 3. *The same is true for* $\tilde{u}_y^{i,j,k}(t)$ *and* $\tilde{u}_z^{i,j,k}(t)$.

Applying the Λ–matrix inequality to terms containing $u_e^{i,j,k}(t)$ and $\Delta u^{i,j,k}$ (and its respective partial derivatives), the learning laws satisfy the following nonlinear matrix differential equations

$$
\begin{aligned}
\dot{W}_r^{i,j,k} =& -\frac{a}{2}\tilde{W}_r^{i,j,k} - K_r^{-1}\Pi_b u_e^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - K_r^{-1}S_1^{i,j,k}u_{e_x}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_x^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - K_r^{-1}S_2^{i,j,k}u_{e_y}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_y^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - K_r^{-1}S_3^{i,j,k}u_{e_z}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_z^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - \frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+61}\Pi_b\tilde{W}_r^{i,j,k}\Omega^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - \frac{1}{2}K_r^{-1}S_1^{i,j,k}\Lambda_{r+72}^{-1}S_1^{i,j,k}\tilde{W}_r^{i,j,k}\Omega_x^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_x^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - \frac{1}{2}K_r^{-1}S_2^{i,j,k}\Lambda_{r+83}^{-1}S_2^{i,j,k}\tilde{W}_r^{i,j,k}\Omega_y^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_y^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - \frac{1}{2}K_r^{-1}S_3^{i,j,k}\Lambda_{r+94}^{-1}S_3^{i,j,k}\tilde{W}_r^{i,j,k}\Omega_z^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_z^r(x^i,y^j,z^k)\right)^{\intercal} \\
& - \frac{1}{2}K_r^{-1}\Pi_b\Lambda_{r+105}\Pi_b\tilde{W}_r^{i,j,k}\Omega^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega^r(x^i,y^j,z^k)\right)^{\intercal}
\end{aligned}
\tag{A.13}
$$

Then, the next inequality is achieved

$$
\begin{aligned}
\dot{V}(t) + aV(t) \leq\ & 2\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} P^{i,j,k}\tilde{u}_t^{i,j,k}(t) + \overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right] I_{\Pi_b}\tilde{u}^{i,j,k}(t) \\
& + \overline{\sum}\left[\tilde{u}_x^{i,j,k}(t)\right] I_1\tilde{u}_x^{i,j,k}(t) + \overline{\sum}\left[\tilde{u}_y^{i,j,k}(t)\right] I_2\tilde{u}_y^{i,j,k}(t) + \overline{\sum}\left[\tilde{u}_z^{i,j,k}(t)\right]^{\mathsf{T}} I_3\tilde{u}_z^{i,j,k}(t) \\
& + \overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}} S_4^{i,j,k}\tilde{u}^{i,j,k}(t) - e^{-ah}\overline{\sum}\left[\tilde{u}^{i,j,k}(t-h)\right]^{\mathsf{T}} S_4^{i,j,k}\tilde{u}^{i,j,k}(t-h) \\
& + h^2\overline{\sum}\left[\tilde{u}_t^{i,j,k}(t)\right]^{\mathsf{T}} R^{i,j,k}\tilde{u}_t^{i,j,k}(t) \\
& - he^{-ah}\overline{\sum}\int_{t-h}^{t}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} dsR^{i,j,k}\int_{t-h}^{t}\left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} ds \\
& - 2\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_b\tilde{u}_t^{i,j,k}(t) - 2\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_b\tilde{f}^{i,j,k}(t) \\
& - 2\overline{\sum}\left[\tilde{u}_t^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_c\tilde{u}_t^{i,j,k}(t) - 2\overline{\sum}\left[\tilde{u}_t^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_c\tilde{f}^{i,j,k}(t) + \Psi \\
& - 2\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_b L^{i,j,k}C\tilde{u}^{i,j,k}(t) - 2\overline{\sum}\left[\tilde{u}_t^{i,j,k}(t)\right]^{\mathsf{T}}\Pi_c L^{i,j,k}C\tilde{u}^{i,j,k}(t)
\end{aligned}
\tag{A.14}
$$

where

$$
\begin{aligned}
\Psi :=\ & \overline{\sum}\left[\tilde{f}_x^{i,j,k}(t)\right]^{\mathsf{T}}\Lambda_{24}^{-1}\tilde{f}_x^{i,j,k}(t) + \overline{\sum}\left[\tilde{f}_y^{i,j,k}(t)\right]^{\mathsf{T}}\Lambda_{32}^{-1}\tilde{f}_y^{i,j,k}(t) \\
& + \overline{\sum}\left[\tilde{f}_z^{i,j,k}(t)\right]^{\mathsf{T}}\Lambda_{48}^{-1}\tilde{f}_z^{i,j,k}(t) + \overline{\sum}\left(L^{i,j,k}\Delta\bar{u}^{i,j,k}(t)\right)^{\mathsf{T}}\Lambda_{60}^{-1}\left(L^{i,j,k}\Delta\bar{u}^{i,j,k}(t)\right) \\
& + \overline{\sum}\left(L^{i,j,k}\Delta\bar{u}^{i,j,k}(t)\right)^{\mathsf{T}}\Lambda_{61}^{-1}\left(L^{i,j,k}\Delta\bar{u}^{i,j,k}(t)\right) + \overline{\sum}\left[\Delta u^{i,j,k}\right]^{\mathsf{T}}\left(\sum_{r=1}^{11}\Pi_b\ \Lambda_{r+61}\ \Pi_b\right)\left[\Delta u^{i,j,k}\right] \\
& + \overline{\sum}\left[\Delta u_x^{i,j,k}\right]^{\mathsf{T}}\left(\sum_{r=1}^{11} S_1^{i,j,k}\ \Lambda_{r+72}\ S_1^{i,j,k}\right)\left[\Delta u_x^{i,j,k}\right] \\
& + \overline{\sum}\left[\Delta u_y^{i,j,k}\right]^{\mathsf{T}}\left(\sum_{r=1}^{11} S_2^{i,j,k}\ \Lambda_{r+83}\ S_2^{i,j,k}\right)\left[\Delta u_y^{i,j,k}\right] \\
& + \overline{\sum}\left[\Delta u_z^{i,j,k}\right]^{\mathsf{T}}\left(\sum_{r=1}^{11} S_3^{i,j,k}\ \Lambda_{r+94}\ S_3^{i,j,k}\right)\left[\Delta u_z^{i,j,k}\right]
\end{aligned}
\tag{A.15}
$$

and

$$
\begin{aligned}
I_{\Pi_b} :=& \Pi_b A^{i,j,k} + \left[A^{i,j,k}\right]^{\mathsf{T}} \Pi_b + \Pi_b R_{\Pi_b}^{i,j,k} \Pi_b + Q_{\Pi_b}^{i,j,k} + aP^{i,j,k} \\
R_{\Pi_b}^{i,j,k} :=& \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_r \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{60} \\
Q_{\Pi_b}^{i,j,k} :=& \sum_{r=1}^{11} \left\|\Omega^r(x^i, y^j, z^k)\right\|^2_{\Lambda_r^{-1}} + \sum_{r=1}^{11} \left\|\Omega_x^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+12}^{-1}} + \sum_{r=1}^{11} \left\|\Omega_y^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+24}^{-1}} \\
&+ \sum_{r=1}^{11} \left\|\Omega_z^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+36}^{-1}} + \sum_{r=1}^{11} \left\|\Omega^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+48}^{-1}} \\
R_{\Pi_c}^{i,j,k} :=& \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+48} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{61}
\end{aligned}
\tag{A.16}
$$

$$
\begin{aligned}
I_1 = \mathrm{Ric}(S_1^{i,j,k}) &:= S_1^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right]^{\mathsf{T}} S_1^{i,j,k} + S_1^{i,j,k} R_{S_1}^{i,j,k} S_1^{i,j,k} + Q_{S_1}^{i,j,k} \\
R_{S_1}^{i,j,k} &:= \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+12} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{24} \\
Q_{S_1}^{i,j,k} &:= aS_1^{i,j,k}
\end{aligned}
\tag{A.17}
$$

$$
\begin{aligned}
I_2 = \mathrm{Ric}(S_2^{i,j,k}) &:= S_2^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right]^{\mathsf{T}} S_2^{i,j,k} + S_2^{i,j,k} R_{S_2}^{i,j,k} S_2^{i,j,k} + Q_{S_2}^{i,j,k} \\
R_{S_2}^{i,j,k} &:= \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+24} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{36} \\
Q_{S_2}^{i,j,k} &:= aS_2^{i,j,k}
\end{aligned}
\tag{A.18}
$$

$$
\begin{aligned}
I_3 = \mathrm{Ric}(S_3^{i,j,k}) &:= S_3^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right]^{\mathsf{T}} S_3^{i,j,k} + S_3^{i,j,k} R_{S_3}^{i,j,k} S_3^{i,j,k} + Q_{S_3}^{i,j,k} \\
R_{S_3}^{i,j,k} &:= \sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+36} \left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}} + \Lambda_{48} \\
Q_{S_3}^{i,j,k} &:= aS_3^{i,j,k}
\end{aligned}
\tag{A.19}
$$

The special class of Riccati equation

$$PA + A^{\mathsf{T}}P + PRP + Q = 0$$

where A, $Q = Q^{\mathsf{T}}$, $R = R^{\mathsf{T}} > 0$ are given matrices of appropriate sizes and $P = P^{\mathsf{T}}$ is variable, may be represented as the following linear matrix inequality (LMI)

$$\begin{bmatrix} -PA - A^{\mathsf{T}}P - Q & PR^{1/2} \\ R^{1/2}P & I_{n\times n} \end{bmatrix} > 0 \tag{A.20}$$

Finally, lets consider the extended vector

$$\eta(t) := \left(\tilde{u}^{i,j,k}(t), \tilde{u}_t^{i,j,k}(t), \int_{t-h}^{t} \left[\tilde{u}_t^{i,j,k}(s)\right]^{\mathsf{T}} ds, \tilde{u}^{i,j,k}(t-h), \tilde{f}^{i,j,k}(t)\right)^{\mathsf{T}} \tag{A.21}$$

therefore, the next inequality can be estimated

$$\dot{V}(t) + aV(t) \leq \overline{\sum} \eta^{\mathsf{T}} W^{i,j,k} \eta + \beta \tag{A.22}$$

where $\beta := \Psi + \epsilon \overline{\sum} \|\tilde{f}^{i,j,k}(t)\|^2$ and

$$W^{i,j,k} := \begin{pmatrix} \mathrm{w}_{11}^{i,j,k} & \mathrm{w}_{12}^{i,j,k} & 0 & 0 & \mathrm{w}_{15}^{i,j,k} \\ \mathrm{w}_{21}^{i,j,k} & \mathrm{w}_{22}^{i,j,k} & 0 & 0 & \mathrm{w}_{25}^{i,j,k} \\ 0 & 0 & \mathrm{w}_{33}^{i,j,k} & 0 & 0 \\ 0 & 0 & 0 & \mathrm{w}_{44}^{i,j,k} & 0 \\ \mathrm{w}_{51}^{i,j,k} & \mathrm{w}_{52}^{i,j,k} & 0 & 0 & \mathrm{w}_{55}^{i,j,k} \end{pmatrix} \tag{A.23}$$

The elements of $W^{i,j,k}$ are given as follows

$$
\begin{aligned}
\mathrm{w}_{11}^{i,j,k} :=& \Pi_b \left(A^{i,j,k} - L^{i,j,k}C\right) + \left(A^{i,j,k} - L^{i,j,k}C\right)^{\mathsf{T}} \Pi_b + \Pi_b R_{\Pi_b}^{i,j,k} \Pi_b \\
& + Q_{\Pi_b}^{i,j,k} + S_4^{i,j,k} + aP^{i,j,k} \\
\mathrm{w}_{12}^{i,j,k} :=& P^{i,j,k} + \left(A^{i,j,k} - L^{i,j,k}C\right)^{\mathsf{T}} \Pi_c - \Pi_b \\
\mathrm{w}_{21}^{i,j,k} :=& P^{i,j,k} + \Pi_c \left(A^{i,j,k} - L^{i,j,k}C\right) - \Pi_b \\
\mathrm{w}_{22}^{i,j,k} :=& h^2 R^{i,j,k} - 2\Pi_c + \Pi_c R_{\Pi_c}^{i,j,k} \Pi_c \\
\mathrm{w}_{33}^{i,j,k} :=& - he^{-ah} R^{i,j,k} \\
\mathrm{w}_{44}^{i,j,k} :=& - e^{-ah} S_4^{i,j,k} \\
.\mathrm{w}_{55}^{i,j,k} :=& - \epsilon I;\ \mathrm{w}_{15}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{51}^{i,j,k} := -\Pi_b;\ \mathrm{w}_{25}^{i,j,k} := -\Pi_c;\ \mathrm{w}_{52}^{i,j,k} := -\Pi_c
\end{aligned}
\tag{A.24}
$$

Hence, by selecting some matrices Π_b and Π_c and the optimal matrices $\hat{P}$ and $\hat{L}$ such that $W \leq 0$, we can conclude that the function $V(t)$ satisfies lemma 1. □

A.2 Proof of theorem 4.

Proof. Let us consider the following Lyapunov functional, defined as the composition of NML individual Lyapunov functios along the whole domain:

$$
\begin{aligned}
V(t) :=& \overline{\sum} \left[\bar{V}^{i,j,k}(t) + \sum_{r=1}^{11} \operatorname{tr} \left\{ \left[\tilde{W}_r^{i,j,k}(t) \right]^{\mathsf{T}} K_r \tilde{W}_r^{i,j,k}(t) \right\} \right] \\
\bar{V}^{i,j,k}(t) :=& \left\| \tilde{u}^{i,j,k}(t) \right\|_{P^{i,j,k}}^2 + \left\| \tilde{u}_x^{i,j,k}(t) \right\|_{S_1^{i,j,k}}^2 + \left\| \tilde{u}_y^{i,j,k}(t) \right\|_{S_2^{i,j,k}}^2 + \left\| \tilde{u}_z^{i,j,k}(t) \right\|_{S_3^{i,j,k}}^2
\end{aligned}
\tag{A.25}
$$

The time derivative $\dot{V}(\cdot)$ can be obtained and is given by the following equation

$$
\begin{aligned}
\dot{V}(t) =& 2\overline{\sum} \left[\tilde{u}^{i,j,k}(t) \right]^{\mathsf{T}} P^{i,j,k} \frac{d}{dt} \tilde{u}^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_x^{i,j,k}(t) \right]^{\mathsf{T}} S_1^{i,j,k} \frac{d}{dt} \tilde{u}_x^{i,j,k}(t) \\
& + 2\overline{\sum} \left[\tilde{u}_y^{i,j,k}(t) \right]^{\mathsf{T}} S_2^{i,j,k} \frac{d}{dt} \tilde{u}_y^{i,j,k}(t) + 2\overline{\sum} \left[\tilde{u}_z^{i,j,k}(t) \right]^{\mathsf{T}} S_3^{i,j,k} \frac{d}{dt} \tilde{u}_z^{i,j,k}(t) \\
& + 2\overline{\sum} \sum_{r=1}^{11} \operatorname{tr} \left\{ \left[\tilde{W}_r^{i,j,k}(t) \right]^{\mathsf{T}} K_r \dot{W}_r^{i,j,k}(t) \right\}
\end{aligned}
\tag{A.26}
$$

Using the Λ–matrix inequality, we can approximate the first term of the time derivative as

$$
\begin{aligned}
2\left[\tilde{u}^{i,j,k}(t)\right]^{\intercal} P^{i,j,k} \frac{d}{dt} \tilde{u}^{i,j,k}(t) \leq & \left[\tilde{u}^{i,j,k}(t)\right]^{\intercal} \left(P^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right] P^{i,j,k}\right) \tilde{u}^{i,j,k}(t) \\
& + \left[\tilde{u}^{i,j,k}(t)\right]^{\intercal} P^{i,j,k} \left(\sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_r \left[\mathring{W}_r^{i,j,k}\right]^{\intercal}\right) P^{i,j,k} \tilde{u}^{i,j,k}(t) \\
& + \sum_{r=1}^{11} \left[\tilde{U}^{(i,j,k),r}(t)\right]^{\intercal} \left\|\Omega^r(x^i, y^j, z^k)\right\|^2_{\Lambda_r^{-1}} \tilde{U}^{(i,j,k),r}(t) \\
& + \left[\tilde{f}^{i,j,k}(t)\right]^{\intercal} \Lambda_{12}^{-1} \tilde{f}^{i,j,k}(t) + \left[\tilde{u}^{i,j,k}(t)\right]^{\intercal} P^{i,j,k} \Lambda_{12} P^{i,j,k} \tilde{u}^{i,j,k}(t) \\
& + 2 \sum_{r=1}^{11} \left[\tilde{u}^{i,j,k}(t)\right]^{\intercal} P^{i,j,k} \tilde{W}_r^{i,j,k} \Omega^r(x^i, y^j, z^k) \hat{U}^{(i,j,k),r}(t)
\end{aligned}
\tag{A.27}
$$

For the terms of the form $2\left[\tilde{u}_m^{i,j,k}(t)\right]^{\intercal} S_l^{i,j,k} \frac{d}{dt} \tilde{u}_m^{i,j,k}(t)$

$$
\begin{aligned}
2\left[\tilde{u}_m^{i,j,k}(t)\right]^{\intercal} S_l^{i,j,k} \frac{d}{dt} \tilde{u}_m^{i,j,k}(t) \leq & \left[\tilde{u}_m^{i,j,k}(t)\right]^{\intercal} \left(S_l^{i,j,k} A^{i,j,k} + \left[A^{i,j,k}\right]^{\intercal} S_l^{i,j,k}\right) \left[\tilde{u}_m^{i,j,k}(t)\right] \\
& + \left[\tilde{u}_m^{i,j,k}(t)\right]^{\intercal} S_l^{i,j,k} \left(\sum_{r=1}^{11} \mathring{W}_r^{i,j,k} \Lambda_{r+b} \left[\mathring{W}_r^{i,j,k}\right]^{\intercal} + \Lambda_{b+12}\right) S_l^{i,j,k} \left[\tilde{u}_m^{i,j,k}(t)\right] \\
& + \sum_{r=1}^{11} \left[\tilde{U}^{(i,j,k),r}(t)\right] \left\|\Omega_m^r(x^i, y^j, z^k)\right\|^2_{\Lambda_{r+b}^{-1}} \left[\tilde{U}^{(i,j,k),r}(t)\right] \\
& + \left[\tilde{f}_m^{i,j,k}(t)\right]^{\intercal} \Lambda_{b+12}^{-1} \left[\tilde{f}_m^{i,j,k}(t)\right] \\
& + 2 \sum_{r=1}^{11} \left[\tilde{u}_m^{i,j,k}(t)\right]^{\intercal} S_l^{i,j,k} \tilde{W}_r^{i,j,k} \Omega_m^r(x^i, y^j, z^k) \hat{U}^{(i,j,k),r}(t)
\end{aligned}
\tag{A.28}
$$

where $l = 1, 2, 3$; m represents the partial derivate, for S_1 it is respect to x, for S_2 with respect to y and for S_3 with respect to z; and $b = 12, 24, 36$, correspondingly. Now, adding

$\alpha V(t)$ to both sides of the resulting inequality we have

$$\begin{aligned}\dot{V}(t)+aV(t)\leq&\overline{\sum}\left[\tilde{u}^{i,j,k}(t)\right]I_1(t)\tilde{u}^{i,j,k}(t)+\overline{\sum}\left[\tilde{u}_x^{i,j,k}(t)\right]I_2(t)\tilde{u}_x^{i,j,k}(t)\\&+\overline{\sum}\left[\tilde{u}_y^{i,j,k}(t)\right]I_3(t)\tilde{u}_y^{i,j,k}(t)+\overline{\sum}\left[\tilde{u}_z^{i,j,k}(t)\right]^{\mathsf{T}}I_4(t)\tilde{u}_z^{i,j,k}(t)\\&+\varpi\overline{\sum}\sum_{s=1}^{4}f_s^{i,j,k}+2\overline{\sum}\sum_{r=1}^{11}\mathrm{tr}\left\{\left[\tilde{W}_r^{i,j,k}\right]^{\mathsf{T}}K_r\dot{W}_r^{i,j,k}\right\}\\&+\alpha\overline{\sum}\sum_{r=1}^{11}\left\{\left[\tilde{W}_r^{i,j,k}\right]^{\mathsf{T}}K_r\tilde{W}_r^{i,j,k}\right\}\end{aligned}\tag{A.29}$$

where

$$\begin{aligned}I_1(t)=\mathrm{Ric}(P^{i,j,k}):=&P^{i,j,k}A^{i,j,k}+\left[A^{i,j,k}\right]^{\mathsf{T}}P^{i,j,k}+P^{i,j,k}R_P^{i,j,k}P^{i,j,k}+Q_P^{i,j,k}\\R_P^{i,j,k}:=&\sum_{r=1}^{11}\mathring{W}_r^{i,j,k}\Lambda_r\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}}+\Lambda_{12}\\Q_P^{i,j,k}:=&\sum_{r=1}^{11}\left\|\Omega^r(x^i,y^j,z^k)\right\|^2_{\Lambda_r^{-1}}+\sum_{r}^{11}\left\|\Omega_x^r(x^i,y^j,z^k)\right\|^2_{\Lambda_{r+12}^{-1}}\\&+\sum_{r=1}^{11}\left\|\Omega_y^r(x^i,y^j,z^k)\right\|^2_{\Lambda_{r+24}^{-1}}+\sum_{r=1}^{11}\left\|\Omega_z^r(x^i,y^j,z^k)\right\|^2_{\Lambda_{r+36}^{-1}}+\alpha P^{i,j,k}\end{aligned}\tag{A.30}$$

$$\begin{aligned}I_2(t)=\mathrm{Ric}(S_1^{i,j,k}):=&S_1^{i,j,k}A^{i,j,k}+\left[A^{i,j,k}\right]^{\mathsf{T}}S_1^{i,j,k}+S_1^{i,j,k}R_{S_1}^{i,j,k}S_1^{i,j,k}+Q_{S_1}^{i,j,k}\\R_{S_1}^{i,j,k}:=&\sum_{r=1}^{11}\mathring{W}_r^{i,j,k}\Lambda_{r+12}\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}}+\Lambda_{24}\\Q_{S_1}^{i,j,k}:=&aS_1^{i,j,k}\end{aligned}\tag{A.31}$$

$$\begin{aligned}I_3(t)=\mathrm{Ric}(S_2^{i,j,k}):=&S_2^{i,j,k}A^{i,j,k}+\left[A^{i,j,k}\right]^{\mathsf{T}}S_2^{i,j,k}+S_2^{i,j,k}R_{S_2}^{i,j,k}S_2^{i,j,k}+Q_{S_2}^{i,j,k}\\R_{S_2}^{i,j,k}:=&\sum_{r=1}^{11}\mathring{W}_r^{i,j,k}\Lambda_{r+24}\left[\mathring{W}_r^{i,j,k}\right]^{\mathsf{T}}+\Lambda_{36}\\Q_{S_2}^{i,j,k}:=&aS_2^{i,j,k}\end{aligned}\tag{A.32}$$

$$I_4(t) = \text{Ric}(S_3^{i,j,k}) := S_3^{i,j,k}A^{i,j,k} + \left[A^{i,j,k}\right]^{\intercal} S_3^{i,j,k} + S_3^{i,j,k}R_{S_3}^{i,j,k}S_3^{i,j,k} + Q_{S_3}^{i,j,k}$$

$$R_{S_3}^{i,j,k} := \sum_{r=1}^{11} \mathring{W}_r^{i,j,k}\Lambda_{r+36}\left[\mathring{W}_r^{i,j,k}\right]^{\intercal} + \Lambda_{48} \tag{A.33}$$

$$Q_{S_3}^{i,j,k} := aS_3^{i,j,k}$$

In order to obtain the learning laws, let us add all terms containing $\tilde{W}_r^{i,j,k}$

$$2\overline{\sum}\sum_{r=1}^{11}\text{tr}\left\{\left[\tilde{W}_r^{i,j,k}(t)\right]^{\intercal}\left(K_r\dot{W}_r^{i,j,k} + P^{i,j,k}\tilde{u}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_m^r(x^i,y^j,z^k)\right)^{\intercal}\right.\right.$$
$$S_1^{i,j,k}\tilde{u}_x^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_x^r(x^i,y^j,z^k)\right)^{\intercal} + S_2^{i,j,k}\tilde{u}_y^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_y^r(x^i,y^j,z^k)\right)^{\intercal}$$
$$\left.\left.S_3^{i,j,k}\tilde{u}_z^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\intercal}\left(\Omega_z^r(x^i,y^j,z^k)\right)^{\intercal} + \frac{\alpha}{2}K_r\tilde{W}_r^{i,j,k}\right)\right\} \tag{A.34}$$

Now, we can approximate $\tilde{u}^{i,j,k}(t)$ as

$$\tilde{u}^{i,j,k}(t) = \hat{u}^{i,j,k}(t) - u^{i,j,k}(t) \pm u_{\text{int}}^{i,j,k} = \bar{\bar{u}}^{i,j,k}(t) + \Delta u_{\text{int}}^{i,j,k} \tag{A.35}$$

where $\bar{\bar{u}}^{i,j,k}(t) := \hat{u}^{i,j,k}(t) - u_{\text{int}}^{i,j,k}$ and $\Delta u_{\text{int}}^{i,j,k} = u_{\text{int}}^{i,j,k} - u^{i,j,k}(t)$. For the other terms, the following applies

$$\tilde{u}_m^{i,j,k}(t) = \bar{\bar{u}}_m^{i,j,k}(t) + \tfrac{\partial}{\partial m}\Delta u_{\text{int}}^{i,j,k} \tag{A.36}$$

Applying the Λ–matrix inequality to terms containing $\bar{\bar{u}}^{i,j,k}(t)$ and $\Delta u_{\text{int}}^{i,j,k}$ (and its respective partial derivatives), the learning laws satisfy the following nonlinear matrix differential equations

$$
\begin{aligned}
\dot{W}_r^{i,j,k} = &- \frac{\alpha}{2}\tilde{W}_r^{i,j,k} - K_r^{-1}P^{i,j,k}\bar{\bar{u}}^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\top}\left(\Omega^r(x^i,y^j,z^k)\right)^{\top} \\
&- \sum_{l=1}^{3} K_r^{-1}S_l^{i,j,k}\bar{\bar{u}}_m^{i,j,k}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\top}\left(\Omega_m^r(x^i,y^j,z^k)\right)^{\top} \\
&- \frac{1}{2}K_r^{-1}P^{i,j,k}\Lambda_r P^{i,j,k}\tilde{W}_r^{i,j,k}\Omega^r(x^i,y^j,z^k)\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\top}\left(\Omega^r(x^i,y^j,z^k)\right)^{\top} \\
&- \frac{1}{2}\sum_{l=1}^{3} K_r^{-1}S_l^{i,j,k}\Lambda_r S_l^{i,j,k}\tilde{W}_r^{i,j,k}\Omega_m^r\hat{U}^{(i,j,k),r}(t)\left(\hat{U}^{(i,j,k),r}(t)\right)^{\top}\left(\Omega_m^r\right)^{\top}
\end{aligned}
\tag{A.37}
$$

If there exist matrices $P^{i,j,k}$, $S_1^{i,j,k}$, $S_2^{i,j,k}$ and $S_3^{i,j,k}$ such that the Riccati equations (A.30)–(A.33) are less than or equal to zero, and taking into account the laws of learning previously obtained, then we obtain the following inequality

$$
\dot{V}(t) = -\alpha V(t) + \beta \tag{A.38}
$$

where

$$
\begin{aligned}
\beta := &\overline{\sum}\sum_{r=1}^{11}\lambda_{\max}\left\{\Lambda_{r+48}^{-1}\right\}\eta_{1,+}^{i,j,k} + \overline{\sum}\sum_{r=1}^{11}\lambda_{\max}\left\{\Lambda_{r+59}^{-1}\right\}\eta_{2,+}^{i,j,k} \\
&+ \overline{\sum}\sum_{r=1}^{11}\lambda_{\max}\left\{\Lambda_{r+70}^{-1}\right\}\eta_{3,+}^{i,j,k} + \overline{\sum}\sum_{r=1}^{11}\lambda_{\max}\left\{\Lambda_{r+82}^{-1}\right\}\eta_{4,+}^{i,j,k} + \varpi\overline{\sum}\sum_{s=1}^{4} f_s^{i,j,k}
\end{aligned}
\tag{A.39}
$$

Thereby, we conclude that the above equation satisfies conditions from lemma 1.

□

Appendix B

Notation

$$
\begin{array}{lll}
U^{(i,j,k),1}(t) = u^{i,j,k}(t) & U^{(i,j,k),2}(t) = u^{i-1,j,k}(t) & U^{(i,j,k),3}(t) = u^{i-2,j,k}(t) \\
U^{(i,j,k),4}(t) = u^{i,j-1,k}(t) & U^{(i,j,k),5}(t) = u^{i,j-2,k}(t) & U^{(i,j,k),6}(t) = u^{i,j,k-1}(t) \\
U^{(i,j,k),7}(t) = u^{i,j,k-2}(t) & U^{(i,j,k),8}(t) = u^{i-1,j-1,k}(t) & U^{(i,j,k),9}(t) = u^{i,j-1,k-1}(t) \\
U^{(i,j,k),10}(t) = u^{i-1,j,k-1}(t) & U^{(i,j,k),11}(t) = u^{i-1,j-1,k-1}(t) &
\end{array}
$$

$$
\begin{array}{lll}
\hat{U}^{(i,j,k),1}(t) = \hat{u}^{i,j,k}(t) & \hat{U}^{(i,j,k),2}(t) = \hat{u}^{i-1,j,k}(t) & \hat{U}^{(i,j,k),3}(t) = \hat{u}^{i-2,j,k}(t) \\
\hat{U}^{(i,j,k),4}(t) = \hat{u}^{i,j-1,k}(t) & \hat{U}^{(i,j,k),5}(t) = \hat{u}^{i,j-2,k}(t) & \hat{U}^{(i,j,k),6}(t) = \hat{u}^{i,j,k-1}(t) \\
\hat{U}^{(i,j,k),7}(t) = \hat{u}^{i,j,k-2}(t) & \hat{U}^{(i,j,k),8}(t) = \hat{u}^{i-1,j-1,k}(t) & \hat{U}^{(i,j,k),9}(t) = \hat{u}^{i,j-1,k-1}(t) \\
\hat{U}^{(i,j,k),10}(t) = \hat{u}^{i-1,j,k-1}(t) & \hat{U}^{(i,j,k),11}(t) = \hat{u}^{i-1,j-1,k-1}(t) &
\end{array}
$$

$$
\begin{array}{ll}
\Omega^1(x^i, y^j, z^k) = \sigma^1(x^i, y^j, z^k) & \Omega^2(x^i, y^j, z^k) = \varphi^1(x^i, y^j, z^k) \\
\Omega^3(x^i, y^j, z^k) = \gamma^1(x^i, y^j, z^k) & \Omega^4(x^i, y^j, z^k) = \varphi^2(x^i, y^j, z^k) \\
\Omega^5(x^i, y^j, z^k) = \gamma^2(x^i, y^j, z^k) & \Omega^6(x^i, y^j, z^k) = \varphi^3(x^i, y^j, z^k) \\
\Omega^7(x^i, y^j, z^k) = \gamma^3(x^i, y^j, z^k) & \Omega^8(x^i, y^j, z^k) = \psi^1(x^i, y^j, z^k) \\
\Omega^9(x^i, y^j, z^k) = \psi^2(x^i, y^j, z^k) & \Omega^{10}(x^i, y^j, z^k) = \psi^3(x^i, y^j, z^k) \\
\Omega^{11}(x^i, y^j, z^k) = \sigma^2(x^i, y^j, z^k) &
\end{array}
$$

Bibliography

[1] Aarts, L.P., Veer, P.V., (2001) *Neural network method for solving partial differential equations*, Neural Processing letters 14, 261–271.

[2] Alli, H., Ucar, Y., Demir, Y., (2003) *The solution of vibration control problems using artificial neural networks*, Journal of the Franklin Institute 340, 307–325.

[3] Beidokhti, R.S., Malek, A., (2009) *Solving initial boundary value problems for system of partial differential equations using neural network and optimization techniques*, Journal of Franklin Institute 346, 898–913.

[4] Blanchini, F., Miani, S., (2008) *Set-theoretic methods in control*, Boston: Birkhäuser.

[5] Chairez, I., Fuentes, R., Poznyak, A., Poznyak, T., Escudero, M., Viana, L. (2010) *DNN–state identification of 2D distributed parameter systems*, in *International Journal of Systems Science*, Vol. 43, No. 2, 296–307.

[6] Cotter, N.E., (1990) *The Stone-Weierstrass theorem and its application to neural networks*, IEEE Transactions on Neural Networks 1, 290–295.

[7] Cybenko, G., (1989) *Approximation of superpositions of a sigmoidal function*, Mathematics of Control, Signals, and Systems 2, 304–314.

[8] Douglas, J., Jones, B.F., (1963) *On predictor–corrector methods for nonlinear parabolic differential equations*, Journal of the Society for Industrial and Applied Mathematics 11, 195–204.

[9] Ei-Hawary, H.M., Mahmoud, S.M., (2003) *Spline collocation methods for solving delay differential equations*, Applied Mathematics and Computation 146, 359–372.

[10] Egorov, Y.V., Shubin, M.A., (1992) *Partial differential equations 1*, New York: Springer-Verlag.

[11] Evans,L.C. (1998) *Partial differential equations*, Rodhe Island: American Mathematical Society.

[12] Fridman, E., (2006) *Descriptor discretized Lyapunov functional method: analysis and design*. IEEE transactions on Automatic Control 51, 890–897.

[13] Fridman, E., (2001) *New Lyapunov-Krasovskii functionals for stability of linear retarded and neutral type systems*, Systems and Control Letters 43, 309–319.

[14] Fridman, E., Niculescu, S.I., (2008) *On complete Lyapunov-Krasovskii functional techniques for uncertain systems with fast-varying delays*, International Journal on Robust and Nonlinear Control 18, 364–374.

[15] Garabedian, P.R., (1986) *Partial differential equations*, New York: Chelsea Publishing Company.

[16] Haykin, S., (1999) *Neural Networks*, New Jersey: Prentice-Hall.

[17] He, S., Reif, K., Unbehauen, R., (2000) *Multilayer neural networks for solving a class of partial differential equations*, Neural networks 13, 385–196

[18] Hughes, T.J.R., (1987) *The finite element method*, New Jersey: Prentice Hall.

[19] Khalil, H., (2002) *Nonlinear systems*, New Jersey: Prentice Hall.

[20] Kumar, M., Gupta, Y., (2010) *Methods for solving singular boundary value problems using splines: a survey*, Applied Mathematics and Computation 32 (2010), 265–278.

[21] Kumar, M., Kumar, P., (2009) *Computational method for finding various solutions for a quasilinear elliptic equation of Kirchhoff type*, Advances in Engineering Software 40, 1104–1111.

[22] Kurzhanski, A., Vályi, I., (1997) *Ellipsoidal Calculus for Estimation and Control*, Boston: Birkhäuser.

[23] Lagaris, I.E., Likas, A., (1998) *Artificial neural networks for solving ordinary and partial differential equations*, IEEE Transactions on Neural Networks 9, 987–1000.

[24] Lagaris, I.E., Likas, A., (2000) *Neural network methods for boundary value problems with irregular boundaries*, IEEE Transactions on Neural Networks 11, 1041–1049.

[25] Lancaster, P., K, Šalkauskas,K., (1986) *Curve and Surface Fitting: an introduction*, London: Academic Press.

[26] O'Neil, P.V., (1999) *Beginning Partial Differential Equations*, New York: Jon Wiley & Sons.

[27] O'Neill, M.O., Ryan, C., (2001) *Grammatical evolution*, IEEE Transactions on Evolutionary Computation 5, 349–358.

[28] Nazin, S.A., Polyak, B.T., Topunov, M.V., (2007) *Rejection of bounded exogenous disturbances by the method of invariant ellipsoids*, Automation and Remote Control 68, 467–486

[29] Petravic, M., Petravic, G.K., Roberts, K.V., (1972) *Automatic production of programmes for solving partial differential equations by finite difference methods*, Computer Physics Communications 4, 82–88.

[30] Poznyak, A., Sanchez, E., Yu, W. (2001) *Differential Neural Networks for Robust Nonlinear Control: identification, state estimation and trajectory tracking*, Singapore: World Scientific.

[31] Poznyak, A. (2008) *Advanced Mathematical Tools for Automatic Control Engineers: deterministic techniques*, Amsterdam: Elsevier.

[32] Sabanovic, A., Fridman, L., Spurgeon, S. (2004) *Variable Structure Systems: from principles to implementation*, London: The Institution of Electrical Engineers.

[33] Shirvany, Y., Hayati, M., Moradian, R., (2009) *Multilayer perceptron neural network with novel unsupervised training method for the solution of partial differential equations*, Applied Soft Computing 9, 20–29.

[34] Smaoui, N., Al-Enezi, S., (2004) *Modeling the dynamics of nonlinear partial differential equations using neural networks*, Journal of Computational and Applied Mathematics 170, 27–58.

[35] Smith, G.D. (1978) *Numerical solution of partial differential equations: finite difference method.* Oxford: Clarendon Press.

[36] Steffensen, J.F., (1950) *Interpolation*, New York: Chelsea Publishing Company.

[37] Thomee, V., (2001) *From finite differences to finite elements: a short history of numerical analysis of partial differential equations*, Journal of Computational and Applied Mathematics 128, 1–54.

[38] Tseng, A.A., Gu, S.X., (1989) *A finite difference scheme with arbitrary mesh system for solving high order partial differential equations*, Computers & structures 31, 319–328.

[39] Tsoulos, I.G., Gavrilis, D., Glavas, E., (2009) *Solving differential equations with constructed neural networks*, Neurocomputing 72, 2385–2391.

[40] Tsoulos, I.G., Gavrilis, D., Glavas, E., (2008) *Neural network construction and training using grammatical evolution*, Neurocomputing 72, 269–277.

[41] Young, N., (1988) *An introduction to Hilbert space*, Cambridge: Cambridge University Press.

[42] Verwer, J.G., (1996) *Explicit Runge–Kutta methods for parabolic partial differential equations*, Applied Numerical Mathematics 22, 359–179.

[43] Xu, X., Zhai, G, (2004) *On practical stability and stabilization of hybrid and switched systems*, Lecture Notes in Computer Science 2993, 615–630.