# 3D Nonparametric Neural Identification 

Rita Q. Fuentes, ${ }^{1}$ Isaac Chairez, ${ }^{2}$ Alexander Poznyak, ${ }^{1}$ and Tatyana Poznyak ${ }^{3}$<br>${ }^{1}$ Automatic Control Department, CINVESTAV-IPN, 07360 México, DF, Mexico<br>${ }^{2}$ Bioprocess Department, UPIBI-IPN, 07360 México, DF, Mexico<br>${ }^{3}$ SEPI, ESIQIE-IPN, 07738 México, DF, Mexico<br>Correspondence should be addressed to Rita Q. Fuentes, rita.q.fuentes.a@gmail.com

Received 23 August 2011; Accepted 2 October 2011
Academic Editor: Haibo He
Copyright © 2012 Rita Q. Fuentes et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.


#### Abstract

This paper presents the state identification study of 3D partial differential equations (PDEs) using the differential neural networks (DNNs) approximation. There are so many physical situations in applied mathematics and engineering that can be described by PDEs; these models possess the disadvantage of having many sources of uncertainties around their mathematical representation. Moreover, to find the exact solutions of those uncertain PDEs is not a trivial task especially if the PDE is described in two or more dimensions. Given the continuous nature and the temporal evolution of these systems, differential neural networks are an attractive option as nonparametric identifiers capable of estimating a 3D distributed model. The adaptive laws for weights ensure the "practical stability" of the DNN trajectories to the parabolic three-dimensional (3D) PDE states. To verify the qualitative behavior of the suggested methodology, here a nonparametric modeling problem for a distributed parameter plant is analyzed.


## 1. Introduction

1.1. 3D Partial Differential Equations. Partial differential equations (PDEs) are of vast importance in applied mathematics, physics, and engineering since so many real physical situations can be modelled by them. The dynamic description of natural phenomenons are usually described by a set of differential equations using mathematical modeling rules [1]. Almost every system described in PDE has already appeared in the one- and two-dimensional situations. Appending a third dimension ascends the dimensional ladder to its ultimate rung, in physical space at least. For instance, linear second-order 3D partial differential equations appear in many problems modeling equilibrium configurations of solid bodies, the three-dimensional wave equation governing vibrations of solids, liquids, gases, and electromagnetic waves, and the three-dimensional heat equation modeling basic spatial diffusion processes. These equations define a state representing rectangular coordinates on $\mathbb{R}^{3}$. There are some basic underlying solution techniques to solve 3D PDEs: separation of variables and Green's functions or fundamental solutions. Unfortunately, the most powerful of the planar
tools, conformal mapping, does not carry over to higher dimensions. In this way, many numerical techniques solving such PDE, for example, the finite difference method (FDM) and the finite element method (FEM), have been developed (see $[2,3]$ ). The principal disadvantage of these methods is that they require the complete mathematical knowledge of the system to define a mesh (domain discretization), where the functions are approximated locally. The construction of a mesh in two or more dimensions is a nontrivial task. Usually, in practice, only low-order approximations are employed resulting in a continuous approximation of the function across the mesh but not in its partial derivatives. The approximation discontinuities of the derivative can adversely affect the stability of the solution. However, all those methods are well defined if the PDE structure is perfectly known. Actually, the most of suitable numerical solutions could be achieved only if the PDE is linear. Nevertheless, there are not so many methods to solve or approximate the PDE solution when its structure (even in a linear case) is uncertain. This paper suggests a different numerical solution for uncertain systems (given by a 3D PDE) based on the Neural Network approach [4].
1.2. Application of Neural Networks to Model PDEs. Recent results show that neural networks techniques seem to be very effective to identify a wide class or systems when we have no complete model information, or even when the plant is considered as a gray box. It is well known that radial basis function neural networks (RBFNNs) and MultiLayer Perceptrons (MLPs) are considered as a powerful tool to approximate nonlinear uncertain functions (see [5]): any continuous function defined on a compact set can be approximated to arbitrary accuracy by such class of neural networks [6]. Since the solutions of interesting PDEs are uniformly continuous and the viable sets that arise in common problems are often compact, neural networks seem like ideal candidates for approximating viability problems (see [7, 8]). Neural networks may provide exact approximation for PDE solutions; however, numerical constraints avoid this possible exactness because it is almost impossible to simulate NN structures with infinite number of nodes (see [7, 9, 10]). The Differential Neural Network (DNN) approach avoids many problems related to global extremum search converting the learning process to an adequate feedback design (see [11, 12]). Lyapunov's stability theory has been used within the neural networks framework (see $[4,11,13]$ ). The contributions given in this paper regard the development of a nonparametric identifier for 3D uncertain systems described by partial differential equations. The method produces an artificial mathematical model in three dimensions that is able to describe the PDEs dynamics. The required numerical algorithm to solve the non-parametric identifier was also developed.

### 2.3D Finite Differences Approximation

The problem requires the proposal of a non-parametric identifier based on DNN in three dimensions. The problem here may be treated within the PDEs framework. Therefore, this section introduces the DNN approximation characteristics to reconstruct the trajectories profiles for a family of 3D PDEs.

Consider the set of uncertain second-order PDEs

$$
\begin{equation*}
u_{t}=f\left(u, u_{x}, u_{x x}, u_{y}, u_{y y}, u_{z}, u_{z z}, u_{x y}, u_{y x}, u_{x z}, u_{y z}\right)+\xi \tag{1}
\end{equation*}
$$

here $u_{h}=u_{h}(x, y, z, t)$, where $h$ represents $t, x, y, z, x x, y y$, $z z, x y, x z, y x, y z$ and $u=u(x, y, z, t)$ has $n$ components $\left(u(x, y, z, t) \in \mathfrak{R}^{n}\right)$ defined in a domain given by $[x, y, z) \in$ $[0,1] \times[0,1] \times[0,1], t \geq 0, \xi=\xi(x, y, z, t) \in \mathfrak{R}^{n}$ is a noise in the state dynamics. This PDE has a set of initial and boundary conditions given by

$$
\begin{array}{ll}
u_{x}(0, y, z, t)=0 \in \mathfrak{R}^{n}, & u(x, 0, z, t)=u_{0} \in \mathfrak{R}^{n}, \\
u(x, y, 0, t)=u_{0} \in \mathfrak{R}^{n}, & u(x, y, z, 0)=c \in \mathfrak{R}^{n} . \tag{2}
\end{array}
$$

In (1), $u_{t}(x, y, z, t)$ stands for $\partial u(x, y, z, t) / \partial t$.
System (1) armed with boundary and initial conditions (2) is driven in a Hilbert space $H$ equipped with an inner product $(\cdot, \cdot)$. Let us consider a vector function $g(t) \in H$ to be a piecewise continuous in $t$. By $L_{\infty}(a, b ; H)$ we denote the set of $H$-valued functions $g$ such that $(g(\cdot), u)$ is Lebesgue measurable for all $u \in H$ and ess $\max _{t \in[a, b]}\|g(\gamma, t)\|<\infty$,
$\gamma \in \mathfrak{R}^{n}$. Suppose that the nonlinear function $g(\gamma, t)$ satisfies the Lipschitz condition $\|g(\gamma, t)-g(\eta, t)\| \leq L\|\gamma-\eta\|$,

$$
\begin{equation*}
\forall \gamma, \eta \in B_{r}\left(\gamma_{0}\right):=\left\{\gamma \in \mathfrak{R}^{n} \mid\left\|\gamma-\gamma_{0}\right\| \leq r, \forall t \in\left[t_{0}, t_{1}\right]\right\}, \tag{3}
\end{equation*}
$$

where $L$ is positive constant and $\|g\|^{2}=(g, g)$ is used just to ensure that there exists some $\delta>0$ such that the state equation $\dot{\gamma}=g(\gamma, t)$ with $\gamma\left(t_{0}\right)=\gamma_{0}$ has a unique solution over $\left[t_{0}, t_{0}+\delta\right]$ (see [14]). The norm defined above stands for the Sobolev space defined in [15] as follows.

Definition 1. Let $\Omega$ be an open set in $\Re^{n}$, and let $\nu \in C^{m}(\Omega)$. Define the norm of $\nu(\gamma)$ as

$$
\begin{equation*}
\|\nu\|_{m, p}:=\sum_{0 \leq|\alpha| \leq m}\left(\int_{\Omega}\left|D^{\alpha} \nu(\gamma)\right|^{p} d \gamma\right)^{1 / p} \tag{4}
\end{equation*}
$$

$\left(1 \leq p<\infty, D^{\alpha} v(\gamma):=\left(\partial^{\alpha} / \partial \gamma^{\alpha}\right) \nu(\gamma)\right)$. This is the Sobolev norm in which the integration is performed in the Lebesgue sense. The completion of the space of function $\nu(\gamma) \in$ $C^{m}(\Omega):\|\nu\|_{m, p}<\infty$ with respect to $\|\cdot\|_{m, p}$ is the Sobolev space $H^{m, p}(\Omega)$. For $p=2$, the Sobolev space is a Hilbert space (see $[14,15]$ ).

Below we will use the norm (4) for the functions $u(\cdot, t)$ for each fixed $t$.
2.1. Numerical Approximation for Uncertain Functions. Now consider a function $h_{0}(\cdot)$ in $H^{m, 2}(\Omega)$. By [16], $h_{0}(\cdot)$ can be rewritten as

$$
\begin{gather*}
h_{0}\left(\gamma, \theta^{*}\right)=\sum_{i} \sum_{j} \sum_{k} \theta_{i j k}^{*} \Psi_{i j k}(\gamma), \\
\theta_{i j k}=\int_{-\infty}^{+\infty} h_{0}(\gamma) \Psi_{i j k}(\gamma) d \gamma, \forall i, j, k \in \mathbb{Z}, \tag{5}
\end{gather*}
$$

where $\left\{\Psi_{i j k}(\gamma)\right\}$ are functions constituting a basis in $H^{m, 2}(\Omega)$. Last expression is referred to as a vector function series expansion of $h_{0}\left(\gamma, \theta^{*}\right)$. Based on this series expansion, an NN takes the following mathematical structure:

$$
\begin{equation*}
h_{0}(\gamma, \theta):=\sum_{i=L_{1}}^{L_{2}} \sum_{j=M_{1}}^{M_{2}} \sum_{k=N_{1}}^{N_{2}} \theta_{i j k} \Psi_{i j k}(\gamma)=\Theta^{T} W(\gamma) \tag{6}
\end{equation*}
$$

that can be used to approximate a nonlinear function $h_{0}\left(\gamma, \theta^{*}\right) \in S$ with an adequate selection of integers $L_{1}, L_{2}$, $M_{1}, M_{2}, N_{1}, N_{2} \in \mathbb{Z}^{+}$, where

$$
\begin{gather*}
\Theta=\left[\theta_{L_{1} M_{1} N_{1}}, \ldots, \theta_{L_{1} M_{1} N_{2}}, \ldots \theta_{L_{2} M_{1} N_{1}}, \ldots \theta_{L_{2} M_{2} N_{2}}\right]^{\top}, \\
W(\gamma)=\left[\Psi_{L_{1} M_{1} N_{1}}, \ldots, \Psi_{L_{1} M_{1} N_{2}}, \ldots \Psi_{L_{2} M_{1} N_{1}}, \ldots \Psi_{L_{2} M_{2} N_{2}}\right]^{\top} . \tag{7}
\end{gather*}
$$

Following the Stone-Weierstrass Theorem [17], if

$$
\begin{equation*}
\epsilon\left(L_{1}, L_{2}, M_{1}, M_{2}, N_{1}, N_{2}\right)=h_{0}\left(\gamma, \theta^{*}\right)-h_{0}(\gamma, \theta) \tag{8}
\end{equation*}
$$

is the NN approximation error, then for any arbitrary positive constant $\varepsilon$ there are some constants $L_{1}, L_{2}, M_{1}, M_{2}, N_{1}$, $N_{2} \in \mathbb{Z}$ such that for all $x \in X \subset \mathfrak{R}$

$$
\begin{equation*}
\left\|\epsilon\left(L_{1}, L_{2}, M_{1}, M_{2}, N_{1}, N_{2}\right)\right\|_{2} \leq \varepsilon . \tag{9}
\end{equation*}
$$



Figure 1: Constructed grid for 3 dimensions.

The main idea behind the application of DNN [11] to approximate the 3D PDEs solution is to use a class of finite-difference methods but for uncertain nonlinear functions. So, it is necessary to construct an interior set (commonly called grid or mesh) that divides the subdomain $x \in[0,1]$ in $L$ equidistant sections, $y \in[0,1]$ in $M$, and $z \in[0,1]$ in $N$ equidistant sections, each one of them (Figure 1) defined as $\left(x^{i}, y^{j}, z^{k}\right)$ in such a way that $x^{0}=y^{0}=z^{0}=0$ and $x^{L}=$ $y^{M}=z^{N}=1$.

Using the mesh description, one can use the next definitions:

$$
\begin{gather*}
u^{i, j, k}(t):=u\left(x^{i}, y^{j}, z^{k}, t\right) \\
u_{t}^{i, j, k}(t):=\left.\frac{\partial u(x, y, z, t)}{\partial t}\right|_{x=x^{i}, y=y^{j}, z=z^{k}}  \tag{10}\\
u_{x}^{i, j, k}(t):=\left.u_{x}(x, y, z, t)\right|_{x=x^{i}, y=y^{j}, z=z^{k}} \\
u_{x x}^{i, j, k}(t):=\left.u_{x x}(x, y, z, t)\right|_{x=x^{i}, y=y^{j}, z=z^{k}}
\end{gather*}
$$

Analogously, we may consider the other cases ( $u_{x x}, u_{y}$, $\left.u_{y y}, u_{z}, u_{z z}, u_{x y}, u_{x z}, u_{y z}\right)$. Using the mesh description and applying the finite-difference representation, one gets

$$
\begin{align*}
& u_{x}^{i, j, k}(t) \simeq \frac{u^{i, j, k}(t)-u^{i-1, j, k}(t)}{\Delta x} \\
& u_{x x}^{i, j, k}(t) \simeq \frac{u_{x}^{i, j, k}(t)-u_{x}^{i-1, j, k}(t)}{\Delta^{2} x} \tag{11}
\end{align*}
$$

and it follows for all cases such that the $(\Delta x, \Delta y, \Delta z)$-approximation of the nonlinear PDE (1) can be represented as

$$
\begin{aligned}
& \dot{u}^{i, j, k}(t)= u_{t}^{i, j, k}(t) \\
& \simeq \Phi\left(u^{i, j, k}, u^{i-1, j, k}, u^{i-2, j, k}, u^{i, j-1, k}\right. \\
& u^{i, j-2, k}, u^{i, j, k-1}, u^{i, j, k-2}, u^{i-1, j-1, k} \\
&\left.u^{i-1, j, k-1}, u^{i, j-1, k-1}, u^{i-1, j-1, k-1}\right) \\
&(i=0, \ldots, L ; j=0, \ldots, M, k=0, \ldots, N) .
\end{aligned}
$$

2.2. 3D Approximation for Uncertain PDE. By simple adding and subtracting the corresponding terms, one can describe (1) as

$$
\begin{align*}
u_{t}= & A u+\stackrel{\circ}{V}_{1} \sigma u+\stackrel{\circ}{V}_{2} \varphi^{1} u_{x}+\stackrel{\circ}{V}_{3} \gamma^{1} u_{x x} \stackrel{\circ}{V}_{4} \varphi^{2} u_{y}+\stackrel{\circ}{V}_{5} \gamma^{2} u_{y y} \\
& +\stackrel{\circ}{V}_{6} \varphi^{3} u_{z}+\stackrel{\circ}{V}_{7} \gamma^{3} u_{z z}+\stackrel{\circ}{V}_{8} \psi^{1} u_{x y} \\
& +\stackrel{\circ}{V}_{9} \psi^{2} u_{y z} \stackrel{\circ}{V}_{10} \psi^{3} u_{x z}+\stackrel{\circ}{V}_{11} \sigma^{2} u_{x y z}+\tilde{f} \tag{13}
\end{align*}
$$

where $u_{t}=u_{t}(x, y, z, t), u=u(x, y, z, t), \sigma=\sigma(x, y, z), u_{x}=$ $u_{x}(x, y, z, t), u_{x x}=u_{x x}(x, y, z, t)$, the same for $\sigma^{2}, \varphi^{i}, \gamma^{i}, \psi^{i}$, $u_{y}, u_{y y}, u_{z}, u_{z z}, u_{x y}, u_{y z}, u_{x z}$, and $u_{x y z}(i=\overline{1,3}), A \in \mathfrak{R}^{n \times n}$, $\stackrel{\circ}{V}_{1} \in \Re^{n \times s_{1}}, \stackrel{\circ}{V}_{2} \in \Re^{n \times s_{2}}, \stackrel{\circ}{V}_{3} \in \Re^{n \times s_{3}}, \tilde{f}=\tilde{f}(x, y, z, t), \stackrel{\circ}{V}_{4} \in$ $\Re^{n \times s_{4}}, \stackrel{\circ}{V}_{5} \in \Re^{n \times s_{5}}, \stackrel{\circ}{V}_{6} \in \Re^{n \times s_{6}}, \stackrel{\circ}{V}_{7} \in \Re^{n \times s_{7}}, \stackrel{\circ}{V}_{8} \in \Re^{n \times s_{8}}$, $\stackrel{\circ}{V}_{9} \in \mathfrak{R}^{n \times s_{9}}, \stackrel{\circ}{V}_{10} \in \mathfrak{R}^{n \times s_{10}}$.

Here $\tilde{f}(x, t) \in \mathfrak{R}^{n}$ represents the modelling error term, $A$ and $\stackrel{\circ}{V}_{k}(k=\overline{1,6})$ any constant matrices and the set of sigmoidal functions have the corresponding size $(\sigma(x, y, z) \in$ $\mathfrak{R}^{s_{1}}, \varphi^{1}(x, y, z) \in \mathfrak{R}^{s_{2}}, \gamma^{1}(x, y, z) \in \mathfrak{R}^{s_{3}}, \varphi^{2}(x, y, z) \in \mathfrak{R}^{s_{4}}$, $\gamma^{2}(x, y, z) \in \mathfrak{R}^{s_{5}}, \psi^{1}(x, y, z) \in \mathfrak{R}^{s_{6}}, \varphi^{3}(x, y, z) \in \mathfrak{R}^{s_{7}}$, $\gamma^{3}(x, y, z) \in \mathfrak{R}^{s_{8}}, \psi^{2}(x, y, z) \in \mathfrak{R}^{s_{9}}, \psi^{3}(x, y, z) \in \mathfrak{R}^{s_{10}}$, and $\left.\sigma^{2}(x, y, z) \in \mathfrak{R}^{s_{11}}\right)$ and are known as the neural network set of activation functions. These functions obey the following sector conditions:

$$
\begin{align*}
& \left\|\sigma^{1}(x, y, z)-\sigma^{1}\left(x^{\prime}, y^{\prime}, z^{\prime}\right)\right\|^{2} \\
& \quad \leq L_{\sigma^{1}}\left(\left\|x-x^{\prime}\right\|^{2}+\left\|y-y^{\prime}\right\|^{2}+\left\|z-z^{\prime}\right\|^{2}\right) \\
& \left\|\sigma^{2}(x, y, z)-\sigma^{2}\left(x^{\prime}, y^{\prime}, z^{\prime}\right)\right\|^{2} \\
& \quad \leq L_{\sigma^{2}}\left(\left\|x-x^{\prime}\right\|^{2}+\left\|y-y^{\prime}\right\|^{2}+\left\|z-z^{\prime}\right\|^{2}\right) \\
& \left\|\phi^{s}(x, y, z)-\phi^{s}\left(x^{\prime}, y^{\prime}, z^{\prime}\right)\right\|^{2} \\
& \quad \leq L_{\phi^{s}}\left(\left\|x-x^{\prime}\right\|^{2}+\left\|y-y^{\prime}\right\|^{2}+\left\|z-z^{\prime}\right\|^{2}\right) \tag{14}
\end{align*}
$$

which are bounded in $x, y$, and $z$, that is,

$$
\begin{gather*}
\|\sigma(\cdot)\|^{2} \leq \sigma^{(l-1)+}, \quad\left\|\varphi^{l}(\cdot)\right\|^{2} \leq \varphi^{l+}  \tag{15}\\
\left\|\gamma^{l}(\cdot)\right\|^{2} \leq \gamma^{l+}, \quad\left\|\psi^{l}(\cdot)\right\| \leq \psi^{l+}, \quad l=1,3 .
\end{gather*}
$$

Following the methodology of DNN [11] and applying the same representation to (12), we get for each $i \in(1, \ldots, L)$, $j \in(1, \ldots, M), k \in(1, \ldots, N)$ the following robust adaptive non-parametric identifier:

$$
\begin{equation*}
u_{t}^{i, j, k}(t)=A^{i, j, k} u^{i, j, k}(t)+\sum_{p=1}^{11} \stackrel{\circ}{W}_{p}^{i, j, k} \phi^{s} U(t)+\tilde{f}^{i, j, k}(t) \tag{16}
\end{equation*}
$$

In the sum we have that $s=\overline{1,3}, \phi$ represents functions $\sigma, \varphi$, $\gamma, \psi$, and $U$ can be taken as the corresponding $u^{i, j, k}, u^{i-1, j, k}$, $u^{i-2, j, k}, u^{i, j-1, k}, u^{i, j-2, k}, u^{i, j, k-1}, u^{i, j, k-2}, u^{i-1, j-1, k}, u^{i, j-1, k-1}$, $u^{i-1, j, k-1}, u^{i-1, j-1, k-1}$. In this equation the term $\tilde{f}^{i, j, k}(t)$, which is usually recognized as the modeling error, satisfies the following identify, and here, it has been omitted the dependence to $x^{i}, y^{j}, z^{k}$ of each sigmoidal function:

$$
\begin{align*}
& \tilde{f}^{i, j, k}(t) \\
& :=\Phi\left(u^{i, j, k}, u^{i-1, j, k}, u^{i-2, j, k}, u^{i, j-1, k}, u^{i, j-2, k}, u^{i, j, k-1}, u^{i, j, k-2}\right. \\
& \left.\quad u^{i-1, j-1, k}, u^{i-1, j, k-1}, u^{i, j-1, k-1}, u^{i-1, j-1, k-1}\right) \\
& \quad-A^{i, j, k} u^{i, j, k}(t)-\sum_{p=1}^{11} \stackrel{\circ}{i}_{p}^{i, j, k} \phi^{s} U(t) \tag{17}
\end{align*}
$$

where $\stackrel{\circ}{W}_{p}^{i} \in \Re^{n \times s_{p}}, p=\overline{1,11}, \phi^{s}, \phi$ represents functions $\sigma$, $\varphi, \gamma, \psi$ and $s=\overline{1,3}, U(t)$ represents the corresponding ( $u^{i, j, k}$, $u^{i-1, j, k}, u^{i-2, j, k}, u^{i, j-1, k}, u^{i, j-2, k}, u^{i, j, k-1}, u^{i, j, k-2}, u^{i-1, j-1, k}$, $\left.u^{i-1, j, k-1}, u^{i, j-1, k-1}, u^{i-1, j-1, k-1}\right)$.

We will assume that the modeling error terms satisfy the following.

Assumption 2. The modeling error is absolutely bounded in $\Omega$ :

$$
\begin{equation*}
\left\|\tilde{f}^{i, j, k}\right\|^{2} \leq f_{1}^{i, j, k} . \tag{18}
\end{equation*}
$$

Assumption 3. The error modeling gradient

$$
\begin{equation*}
\left.\nabla_{s} \tilde{f}(x, y, z, t)\right|_{s=s_{i}}:=\nabla_{s} \tilde{f}^{i, j, k} \tag{19}
\end{equation*}
$$

is bounded as $\left\|\nabla_{s} \tilde{f}^{i, j, k}\right\|^{2} \leq f_{r}^{i, j, k}$, where $s=x, y, z$ and $f_{r}^{i, j, k}(r=\overline{1,3})$ are constants.

## 3. DNN Identification for Distributed Parameters Systems

3.1. DNN Identifier Structure. Based on the DNN methodology [11], consider the DNN identifier

$$
\begin{equation*}
\frac{d}{d t} \widehat{u}^{i, j, k}=A^{i, j, k} \widehat{u}^{i, j, k}+\sum_{p=1}^{11} W_{p}^{i, j, k}(t) \phi^{s} \widehat{U} \tag{20}
\end{equation*}
$$

for all $i=0, \ldots, L ; \hat{u}_{-1}(t)=\hat{u}_{-2}(t)=0$, where $\phi$ represents activation functions $\sigma, \varphi, \gamma$, and $\psi, s=\overline{1,3}, \widehat{U}$ is each one of the states $\widehat{u}^{i, j, k}, \widehat{u}^{i-1, j, k}, \widehat{u}^{i-2, j, k}, \widehat{u}^{i, j-1, k}, \widehat{u}^{i, j-2, k}, \widehat{u}^{i, j, k-1}, \widehat{u}^{i, j, k-1}$, $\widehat{u}^{i, j, k-2}, \widehat{u}^{i-1, j-1, k}, \hat{u}^{i-1, j, k-1}, \widehat{u}^{i-1, j-1, k-1}$, and $A^{i, j, k} \in \Re^{n \times n}$ is a constant matrix to be selected, $\hat{u}^{i, j, k}(t)$ is the estimate of $u^{i, j, k}(t)$. Obviously that proposed methodology implies the designing of individual DNN identifier for each point $x_{i}, y_{j}$, $z_{k}$. The collection of such identifiers will constitute a DNN net containing $N \times M$ connected DNN identifiers working in parallel. Here $\sigma^{1}\left(x^{i}, y^{j}, z^{k}\right), \varphi^{1}\left(x^{i}, y^{j}, z^{k}\right), \varphi^{2}\left(x^{i}, y^{j}, z^{k}\right)$, $\varphi^{3}\left(x^{i}, y^{j}, z^{k}\right), \gamma^{1}\left(x^{i}, y^{j}, z^{k}\right), \gamma^{2}\left(x^{i}, y^{j}, z^{k}\right), \gamma^{3}\left(x^{i}, y^{j}, z^{k}\right), \psi^{1}\left(x^{i}\right.$, $\left.y^{j}, z^{k}\right), \psi^{2}\left(x^{i}, y^{j}, z^{k}\right), \psi^{3}\left(x^{i}, y^{j}, z^{k}\right)$, and $\sigma^{2}\left(x^{i}, y^{j}, z^{k}\right)$ are the NN activation vectors. This means that the applied DNN-approximation significantly simplifies the specification of $\sigma^{1}(\cdot, \cdot), \varphi^{1}(\cdot, \cdot), \varphi^{2}(\cdot, \cdot), \varphi^{3}(\cdot, \cdot), \gamma^{1}(\cdot, \cdot), \gamma^{2}(\cdot, \cdot), \gamma^{3}(\cdot, \cdot)$ and $\psi^{1}(\cdot, \cdot), \psi^{2}(\cdot, \cdot), \psi^{3}(\cdot, \cdot), \sigma^{2}(\cdot, \cdot)$ which now are constant for any $x^{i}, y^{j}, z^{k}$ fixed.
3.2. Learning Laws for Identifier's Weights. For each $i=0$, $\ldots, L, j=0, \ldots, M, k=0, \ldots, N$, define the vector-functions defining 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 $i, j, k$ :

$$
\begin{gather*}
\tilde{u}^{i, j, k}(t):=\widehat{u}^{i, j, k}(t)-u^{i, j, k}(t), \\
\tilde{u}_{s}^{i, j, k}(t):=\hat{u}_{s}^{i, j, k}(t)-u_{s}^{i, j, k}(t),  \tag{21}\\
s=x, y, z .
\end{gather*}
$$

Let $W_{r}^{i, j, k}(t) \in \mathfrak{R}^{n}, r=\overline{1,11}$ be time-variant matrices. These matrices satisfy the following nonlinear matrix differential equations:

$$
\begin{align*}
\dot{W}_{r}^{i, j, k}(t):= & \frac{d}{d t} \bar{W}_{r}^{i, j, k}(t)=-\alpha \widetilde{W}_{r}^{i, j, k}(t) \\
& -K_{r}^{-1} P^{i, j, j, k} \hat{u}^{i, j, k}\left(\hat{U}^{(i, j, k), r}\right)^{\top}\left(\Omega^{r}\left(x^{i}, y^{j}, z^{j}\right)\right)^{\top} \\
& -K_{r}^{-1} S_{1}^{i, j, k} u_{x}^{i, j, k}\left(\hat{U}^{(i, j, k), r}\right)^{\top}\left(\Omega_{x}^{r}\left(x^{i}, y^{j}, z^{j}\right)\right)^{\top} \\
& -K_{r}^{-1} S_{2}^{i, j, k} \sim_{u}^{i, j, k}\left(\hat{U}^{(i, j, k), r}\right)^{\top}\left(\Omega_{y}^{r}\left(x^{i}, y^{j}, z^{j}\right)\right)^{\top} \\
& -K_{r}^{-1} S_{3}^{i, j, k} i_{z}^{i, j, k}\left(\hat{U}^{(i, j, k), r}\right)^{\top}\left(\Omega_{z}^{r}\left(x^{i}, y^{j}, z^{j}\right)\right)^{\top}, \tag{22}
\end{align*}
$$

where $\Omega^{h}\left(x^{i}, y^{j}, z^{j}\right)=S^{l}\left(x^{i}, y^{j}, z^{j}\right)(h=\overline{1,10}, l=\overline{1,3})$ represents the corresponding sigmoidal functions $\sigma^{l}\left(x^{i}, y^{j}, z^{j}\right)$, $\varphi^{l}\left(x^{i}, y^{j}, z^{j}\right), \gamma^{l}\left(x^{i}, y^{j}, z^{j}\right)$, and $\psi^{l}\left(x^{i}, y^{j}, z^{j}\right)$. Here

$$
\begin{array}{r}
\hat{U}^{(i, j, k), 1}(t)=\widehat{u}^{i, j, k}(t), \\
\tilde{U}^{(i, j, k), 3}(t)=\hat{u}^{i,-2, j, k}(t), \\
\tilde{U}^{(i, j), 2}(t)=\hat{u}^{i-1, j, k}(t),, 4 \\
\tilde{U}^{(i, j), 5}(t)=\hat{u}^{i, j-1, k}(t), \\
\tilde{U}^{i, j-2, k}(t), \\
\tilde{U}^{(i, j, j, k), 6}(t)=\hat{u}^{i-1, j-1, k}(t)=\hat{u}^{i-1, j-1, k}(t),  \tag{23}\\
\tilde{U}^{(i, j, k), 9}(t)=\hat{u}^{i-1, j-1, k}(t), \\
\tilde{U}^{(i, j, k), 8}(t)=\hat{u}^{i-1, j-1, k}(t)=\hat{u}^{i-1, j-1, k}(t), \\
\tilde{U}^{(i, j, k), 11}(t)=\hat{u}^{i-1, j-1, k-1}(t)
\end{array}
$$

with positive matrices $K_{r}(r=\overline{1,11})$ and $P^{i, j, k}, S_{1}^{i, j, k}$, and $S_{2}^{i, j, k}$ ( $i=\overline{0, N} ; j=\overline{0, M}$ ) which are positive definite solutions $\left(P^{i, j}>0, S_{1}^{i, j}>0, S_{2}^{i, j}>0\right)$ and $S_{3}^{i, j, k}(i=\overline{0, N} ; j=\overline{0, M}, k=$ $\overline{0, L})$ of the algebraic Riccati equations defined as follows:

$$
\begin{align*}
\operatorname{Ric}\left(P^{i, j, k}\right):= & P^{i, j, k} A^{i, j, k}+\left[A^{i, j, k}\right]^{\top} P^{i, j, k} \\
& +P^{i, j, k} R_{P}^{i, j, k} P^{i, j, k}+Q_{P}^{i, j, k}=0, \\
\operatorname{Ric}\left(S_{1}^{i, j, k}\right):= & S_{1}^{i, j, k} A^{i, j, k}+\left[A^{i, j}\right]^{\top} S_{1}^{i, j, k} \\
& +S_{1}^{i, j, k} R_{S_{1}}^{i, j, k} S_{1}^{i, j, k}+Q_{S_{1}}^{i, j, k}=0, \\
\operatorname{Ric}\left(S_{2}^{i, j, k}\right):= & S_{2}^{i, j, k} A^{i, j, k}+\left[A^{i, j, k}\right]^{\top} S_{2}^{i, j, k}  \tag{24}\\
& +S_{2}^{i, j, k} R_{S_{2}}^{i, j, k} S_{2}^{i, j, k}+Q_{S_{2}}^{i, j, k}=0, \\
\operatorname{Ric}\left(S_{3}^{i, j, k}\right):= & S_{3}^{i, j, k} A^{i, j, k}+\left[A^{i, j, k}\right]^{\top} S_{3}^{i, j, k} \\
& +S_{3}^{i, j, k} R_{S_{3}, j, k}^{i, j, j, k}+Q_{S_{3}}^{i, j, k}=0,
\end{align*}
$$

where each $R_{B}^{i, j, k}$ has the form

$$
\begin{equation*}
R_{B}^{i, j, k}:=\sum_{r=1}^{11} \stackrel{\circ}{W}_{r}^{i, j, k} \Lambda_{r+b}\left(\stackrel{\circ}{W}_{r}^{i, j, k}\right)^{\top}+\Lambda_{b} \tag{25}
\end{equation*}
$$

where $B$ can be $P, S_{1}, S_{2}$, and $S_{3}$ and $b=(7,14,21,28)$.
Matrices $Q_{B}^{i, j, k}$ have the form

$$
\begin{align*}
Q_{B}^{i, j, k}:= & \left\|\Omega_{m}^{1}\left(x^{i}, y^{j}, z^{k}\right)\right\|_{\Lambda_{1}^{-1}}^{2}+\left\|\Omega_{m}^{2}\left(x^{i-1}, y^{j}, z^{k}\right)\right\|_{\Lambda_{2}^{-1}}^{2} \\
& +\left\|\Omega_{m}^{3}\left(x^{i-2}, y^{j}, z^{k}\right)\right\|_{\Lambda_{3}^{-1}}^{2}+\left\|\Omega_{m}^{4}\left(x^{i}, y^{j-1}, z^{k}\right)\right\|_{\Lambda_{4}^{-1}}^{2} \\
& +\left\|\Omega_{m}^{5}\left(x^{i}, y^{j-2}, z^{k}\right)\right\|_{\Lambda_{5}^{-1}}^{2}+\left\|\Omega_{m}^{6}\left(x^{i}, y^{j}, z^{k-1}\right)\right\|_{\Lambda_{6}^{-1}}^{2} \\
& +\left\|\Omega_{m}^{7}\left(x^{i}, y^{j}, z^{k-2}\right)\right\|_{\Lambda_{7}^{-1}}^{2}+\left\|\Omega_{m}^{8}\left(x^{i-1}, y^{j-1}, z^{k}\right)\right\|_{\Lambda_{8}^{-1}}^{2} \\
& +\left\|\Omega_{m}^{9}\left(x^{i}, y^{j-1}, z^{k-1}\right)\right\|_{\Lambda_{9}^{-1}}^{2}+\left\|\Omega_{m}^{10}\left(x^{i-1}, y^{j}, z^{k-1}\right)\right\|_{\Lambda_{10}^{-1}}^{2} \\
& +\left\|\Omega_{m}^{11}\left(x^{i-1}, y^{j-1}, z^{k-1}\right)\right\|_{\Lambda_{11}^{-1}}^{2}+Q_{B}^{i, j, k}, \tag{26}
\end{align*}
$$

where $B$ can be $P, S_{1}, S_{2}$, or $S_{3}$, representing the partial derivative; for $S_{1}$ it is with respect to $x$, for $S_{2}$ with respect to $y$, and for $S_{3}$ with respect to $z$, and $\Lambda_{l}^{-1}(l=\overline{1,46})$, where

$$
\begin{align*}
& \Omega_{x}^{r}\left(x_{i}, y_{j}, z_{j}\right):=\left.\frac{d}{d x} \Omega^{r}(x, y, z)\right|_{x=x_{i}, y=y_{j}, z=z_{k}}  \tag{32}\\
& \Omega_{y}^{r}\left(x_{i}, y_{j}, z_{j}\right):=\left.\frac{d}{d y} \Omega^{r}(x, y, z)\right|_{x=x_{i}, y=y_{j}, z=z_{k}}  \tag{27}\\
& \Omega_{z}^{r}\left(x_{i}, y_{j}, z_{j}\right):=\left.\frac{d}{d z} \Omega^{r}(x, y, z)\right|_{x=x_{i}, y=y_{j}, z=z_{k}} \tag{33}
\end{align*}
$$

Lemma 5. Let a nonnegative function $V(t)$ satisfying the following differential inequality:

$$
\begin{equation*}
\dot{V}(t) \leq-\alpha V(t)+\beta, \tag{31}
\end{equation*}
$$

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

$$
\left[1-\frac{\mu}{\sqrt{V(t)}}\right]_{+} \longrightarrow 0
$$

with $\mu=\sqrt{\beta / \alpha}$ and the function $[\cdot]_{+}$defined as

$$
[z]_{+}:= \begin{cases}z & \text { if } z \geq 0 \\ 0 & \text { if } z<0\end{cases}
$$

Proof. The proof of this lemma can be obtained directly by the application of the Gronwall-Bellman Lemma.

Proposition 6. Given a time interval $T$ and a function $v(\cdot)$ over a continuously differentiable real-valued function $V(z, t)$ satisfying $V(0, t)=0$, for all $t \in T$, is said to be $\varepsilon$-practical Lya-punov-like function over $T$ under $v$ if there exists a constant $\alpha>0$ such that

$$
\begin{equation*}
\dot{V}(z, t) \leq-\alpha V(z, t)+H\left(\omega^{+}\right), \tag{34}
\end{equation*}
$$

with $H$ a bounded nonnegative nonlinear function with upper bound $\mathrm{H}^{+}$. Moreover, the trajectories of $z_{t}$ belong to the zone $\varepsilon:=H^{+} / \alpha$ when $t \rightarrow \infty$. In this proposition $\dot{V}\left(z_{t}, t\right)$ denotes the derivative of $V(z, t)$ along $z_{t}$, that is, $\dot{V}(z, t)=V_{z}(z, t)$. $\left(g\left(z_{t}, v_{t}\right)+\omega_{t}\right)+V_{t}(x, t)$.

Proof. The proof follows directly from Lemma 5.
Definition 7. Given a time interval $T$ and a function $v(\cdot)$ over $T$, nonlinear system (30) is $\varepsilon$-practically stable, $T$ under $v$ if there exists an $\varepsilon$-practical Lyapunov-like function $V(x, t)$ over $T$ under $v$.

## 5. Identification Problem Formulation

The state identification problem for nonlinear systems (13) analyzed in this study, could be now stated as follows.

Problem. For the nonlinear system, given by the vector PDE (20), to study the quality of the DNN identifier supplied with the adjustment (learning) laws (22), estimate the upper bound of the identification error $\delta$ given by

$$
\begin{equation*}
\delta:=\varlimsup_{t \rightarrow \infty} \sum_{i=0}^{L} \sum_{j=0}^{M} \sum_{k=0}^{N}\left\|\hat{u}^{i, j, k}(t)-u^{i, j, k}(t)\right\|_{P^{i, j, k}}^{2} \tag{35}
\end{equation*}
$$

(with $P^{i, j, k}$ from (24)) and, if it is possible, to reduce to its lowest possible value, selecting free parameters participating into the DNN identifier $\left(A, K_{r}, r=\overline{1,11}\right)$.

This implicates that the reduction of the identification error $\delta$ means that the differential neural network has converged to the solution of the 3D PDE; this can be observed in the matching of the DNN to the PDE state.

## 6. Main Result

The main result of this paper is presented in the following theorem.

Theorem 8. Consider the nonlinear model (1) ( $i=0, \ldots, L$; $j=0, \ldots, M, k=0, \ldots, N)$, given by the system of PDEs with uncertainties (perturbations) in the states, under the border conditions (2). Let us also suppose that the DNN-identifier is given by (20) which parameters are adjusted by the learning laws (22) with parameters $\alpha_{m}^{i, j, k}(i=0, \ldots, L ; j=0, \ldots, M ; k=$ $0, \ldots, N)$. If positive definite matrices $Q_{1}^{i, j, k}, Q_{2}^{i, j, k}$, and $Q_{3}^{i, j, k}$ provide the existence of positive solutions $P^{i, j}, S_{1}^{i, j, k}, S_{2}^{i, j, k}$, and


Figure 2: Numerical trajectory produced by the mathematical model described by 3D partial differential equation at time 10 s along the whole domain.

Approximation of 3D differential neural network at time 10


Figure 3: Numerical trajectory produced by the neural network described reproducing the model at time 10 s along the whole domain.


Figure 4: Error of the approximation. Difference between 3D PDE numerical solution and 3D DNN approximation at time 10 s along the whole domain.
$S_{3}^{i, j, k}(i=0, \ldots, L)$ to the Riccati equations (24), then for all $i=0, \ldots, L ; j=0, \ldots, M ; k=0, \ldots, N$ the following $\rho$-upper bound:

$$
\begin{align*}
& \varlimsup_{t \rightarrow \infty} \sum_{i=0}^{L} \sum_{j=0}^{M} \\
& \times \sum_{k=0}^{N}\left[\left\|\hat{u}^{i, j, k}(t)-u^{i, j, k}(t)\right\|_{P^{i, j, k}}^{2}+\left\|\hat{u}_{x}^{i, j, k}(t)-u_{x}^{i, j, k}(t)\right\|_{S_{1}^{i, j, k}}\right. \\
& \left.\quad+\left\|\hat{u}_{y}^{i, j, k}(t)-u_{y}^{i, j, k}(t)\right\|_{S_{2}^{i, j, k}}+\left\|\hat{u}_{z}^{i, j, k}(t)-u_{z}^{i, j, k}(t)\right\|_{S_{3}^{i, j, k}}\right] \\
& \quad \leq \mu \tag{36}
\end{align*}
$$

is ensured with $\mu_{0}:=\sqrt{L M N} \alpha^{-1 / 2}$ and

$$
\begin{align*}
\mu:= & \mu_{0} \sqrt{\max _{i, j, k}} \\
\Psi:= & \omega \sum_{s=1}^{3} f_{s}^{i, j, k} \\
\geq & \left(\tilde{f}^{i, j, k}(t)\right)^{T} \Lambda_{7}^{-1} \tilde{f}^{i, j, k}(t)+\left(\tilde{f}_{x}^{i, j, k}(t)\right)^{T} \Lambda_{14}^{-1} \tilde{f}_{x}^{i, j, k}(t) \\
& +\left(\tilde{f}_{y}^{i, j, k}(t)\right)^{T} \Lambda_{21}^{-1} \tilde{f}_{y}^{i, j, k}(t)+\left(\tilde{f}_{z}^{i, j, k}(t)\right)^{T} \Lambda_{28}^{-1} \tilde{f}_{z}^{i, j, k}(t) \\
\omega:= & \max \left\{\lambda_{\max }\left(\Lambda_{7}^{-1}\right), \lambda_{\max }\left(\Lambda_{14}^{-1}\right), \lambda_{\max }\left(\Lambda_{21}^{-1}\right), \lambda_{\max }\left(\Lambda_{28}^{-1}\right)\right\} \tag{37}
\end{align*}
$$

Moreover, the weights $W_{r, t} r=\overline{1,11}$ remain bounded being proportional to $\mu$, that is, $\left\|W_{r, t}\right\| \leq K_{r} \mu, r=\overline{1,11}$.

Proof. The proof is given in the appendix.

## 7. Simulation Results

7.1. Numerical Example. Below, the numerical smulation shows the qualitative illustration for a benchmark system. Therefore, consider the following three-dimensional PDE described as follows:

$$
\begin{align*}
u_{t}(x, y, z, t)= & -c_{1} u_{x x}(x, y, z, t)-c_{2} u_{y y}(x, y, z, t) \\
& -c_{3} u_{z z}(x, y, z, t)+\xi(x, y, z, t) \tag{38}
\end{align*}
$$

where $c_{1}=c_{2}=c_{3}=0.15$. It is assumed that there is access to discrete measurements of the state $u(x, y, z, t)$ along the whole domain, which is feasible in practice. $\xi(x, y, z, t)$ is a noise in the state dynamics. This model will be used just to generate the data to test the 3D identifier based on DNN. Boundary conditions and initial conditions were selected as follows:

$$
\begin{gather*}
u(0,0,0, t)=\operatorname{rand}(1), \quad u_{x}(0,0,0, t)=0 \\
u_{y}(0,0,0, t)=0, \quad u_{z}(0,0,0, t)=0 \tag{39}
\end{gather*}
$$

The trajectories of the model can be seen in Figure 3 as well as the estimated state, produced by the DNN identifier. The


Figure 5: Comparison between 3D PDE trajectories and 3D DNN approximation for coordinate $z$ at time 10 s.


Figure 6: Index quadratic error for the whole time with coordinates $x, y$, and $z$ fixed.
efficiency of the identification process provided by the suggested DNN algorithm shown in Figure 4.

In Figures 5 and 6 there are shown the trajectories of the PDE and the DNN for the coordinate $z$ and the index error of the PDE and the DNN at 10 seconds for coordinates $x, z$, respectively Figure 2.
7.2. Tumour Growth Example. The mathematical model of the brain tumour growth is presented in this section based on the results of [18]. Here the diffusion coefficient is considered as a constant. Let consider the following three-dimensional


Figure 7: Numerical trajectory produced by the mathematical model described by 3D partial differential equation at time 46 days along the whole domain.
parabolic equation of the tumour growth described as follows:

$$
\begin{align*}
u_{t}(x, y, z, t)= & -P u_{x}(x, y, z, t)-R u_{y}(x, y, z, t) \\
& -S u_{z}(x, y, z, t)+Q u_{x x}(x, y, z, t) \\
& +Q u_{y y}(x, y, z, t)+Q u_{z z}(x, y, z, t)  \tag{40}\\
& +\Gamma-L u(x, y, z, t),
\end{align*}
$$

where $u(x, y, z, t)$ is the growth rate of a brain tumour, $Q$ is the diffusion coefficient, $W=(P, R, S)$ is the drift velocity field, $\Gamma=\Gamma(u)$ is the proliferation coefficient, and $L=L(u)$ is the decay coefficient of cells. It is assumed that there is access to discrete measurements of the state $u(x, y, z, t)$ along the whole domain, which is feasible in practice by PETCT (Positron emission tomography-computed tomography) technology. The domain $0 \leq x \leq 1,0 \leq y \leq 1$, and $0 \leq z \leq 1$ This model will be used just to generate the data to test the 3D identifier based on DNN. Boundary conditions and initial conditions were selected as follows:

$$
\begin{gather*}
u(0,0,0, t)=200 \pm 20 \mu, \quad u_{x}(0,0,0, t)=0  \tag{41}\\
u_{y}(0,0,0, t)=0, \quad u_{z}(0,0,0, t)=0 .
\end{gather*}
$$

The trajectories of the model and the estimate state produced by the DNN identifier can be seen in Figure 7. The dissimilarity between both trajectories depends on the learning period required for adjusting the DNN identifier. The error between trajectories produced by the model and the proposed identifier is close to zero almost for all $x, y, z$ and all $t$ that shows the efficiency of the identification process provided by the suggested DNN algorithm is shown in Figure 8.

In Figures 9 and 10 there are shown the trajectories of the PDE and the DNN for the coordinate $z$ and the index error of the PDE and the DNN at 46 days for coordinates $x$, $z$, respectively.

## 8. Conclusions

The adaptive DNN method proposed here solves the problem of non-parametric identification of nonlinear systems (with uncertainties) given by 3D uncertain PDE. Practical stability for the identification process is demonstrated based on the Lyapunov-like analysis. The upper bound of the identification error is explicitly constructed. Numerical examples demonstrate the estimation efficiency of the suggested methodology.


Figure 8: Error of the approximation. Difference between 3D PDE numerical solution and 3D DNN approximation at time 46 days along the whole domain.


Figure 9: Comparison between 3D PDE trajectories and 3D DNN approximation for coordinate $z$ at time 46 days.

## Appendix

Consider the Lyapunov-like function defined as the composition of NML individual Lyapunov functions $\bar{V}_{i, j, k}(t)$ along the whole space:

$$
\begin{align*}
V(t): & =\sum_{i=0}^{L} \sum_{j=0}^{M} \sum_{k=0}^{N}\left[\bar{V}^{i, j, k}(t)+\sum_{r=1}^{11} \operatorname{tr}\left\{\left[\widetilde{W}_{r}^{i}(t)\right]^{T} K_{r} \widetilde{W}_{r}^{i}(t)\right\}\right], \\
\bar{V}^{i, j, k}(t): & =\left\|\tilde{u}^{i, j, k}(t)\right\|_{p_{i}^{i, j, k}}^{2}+\left\|\tilde{u}_{x}^{i, j, k}(t)\right\|_{S_{1}^{i j, k}}^{2} \\
& +\left\|\tilde{u}_{y}^{i, j, k}(t)\right\|_{S_{2}^{i, j}}^{2}+\left\|\tilde{u}_{z}^{i, j, k}(t)\right\|_{S_{3}^{i, j}, k}^{2} . \tag{A.1}
\end{align*}
$$



Figure 10: Index quadratic error for the whole time with coordinates $x, y$, and $z$ fixed.

The time derivative $\dot{V}(t)$ of $V(t)$ can be obtained so that

$$
\begin{align*}
\dot{V}(t)= & 2 \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left(\tilde{u}^{i, j, k}(t)\right)^{T}(t) P^{i \cdot j, k} \frac{d}{d t} \tilde{u}^{i, j, k}(t) \\
& +2 \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left[\tilde{u}_{x}^{i, j, k}(t)\right]^{T} S_{1}^{i \cdot j, k} \frac{d}{d t} \tilde{u}_{x}^{i \cdot j, k}(t) \\
& +2 \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left[\tilde{u}_{y}^{i, j, k}(t)\right]^{T} S_{2}^{i \cdot j, k} \frac{d}{d t} \tilde{u}_{y}^{i \cdot j, k}(t)  \tag{A.2}\\
& +2 \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left[\tilde{u}_{z}^{i, j, k}(t)\right]^{T} S_{3}^{i \cdot j, k} \frac{d}{d t} \tilde{u}_{z}^{i, j, k}(t) \\
& +2 \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \sum_{r=1}^{3} \operatorname{tr}\left\{\left[\widetilde{W}_{r}^{i, j, k}(t)\right]^{T} K_{r} \dot{W}_{r}^{i, j, k}(t)\right\} .
\end{align*}
$$

Applying the matrix inequality given in [19]

$$
\begin{equation*}
X Y^{T}+Y X^{T} \leq X \Lambda X^{T}+Y \Lambda^{-1} Y^{T} \tag{A.3}
\end{equation*}
$$

valid for any $X, Y \in R^{r \times s}$ and for any $0<\Lambda=\Lambda^{T} \in R^{s \times s}$ to the terms containing $\tilde{f}^{i}(t)$ and their derivatives. We obtain

$$
\begin{align*}
\dot{V}(t) \leq & I_{1}(t)+I_{2}(t)+I_{3}(t)+I_{4}(t) \\
& -2 \alpha \sum_{r=1}^{11} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \operatorname{tr}\left\{\left(\widetilde{W}_{r}^{i, j}(t)\right)^{T} K_{r} \bar{W}_{r}^{i, j, k}(t)\right\} \\
& +2 \alpha \sum_{r=1}^{11} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \operatorname{tr}\left\{\left(\widetilde{W}_{r}^{i, j}(t)\right)^{T} K_{r} \bar{W}_{r}^{i, j, k}(t)\right\}  \tag{A.4}\\
& +2 \sum_{r=1}^{11} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \operatorname{tr}\left\{\left(\widetilde{W}_{r}^{i, j}(t)\right)^{T} K_{r}^{2} \dot{W}_{1}^{i, j, k}(t)\right\} \\
& -\sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \alpha \bar{V}^{i, j, k}(t)+\omega \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \sum_{s=1}^{3} f_{s}^{i, j, k}
\end{align*}
$$

where

$$
\begin{align*}
I_{1}(t) & :=\sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left(\tilde{u}^{i . j, k}(t)\right)^{T} \operatorname{Ric}\left(P^{i, j}\right) \tilde{u}^{i, j, k}(t), \\
I_{2}(t) & :=\sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left(\tilde{u}_{x}^{i . j, k}(t)\right)^{T} \operatorname{Ric}\left(S_{1}^{i, j, k}\right) \tilde{u}_{x}^{i, j, k}(t), \\
I_{3}(t) & :=\sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left(\tilde{u}_{y}^{i . j, k}(t)\right)^{T} \operatorname{Ric}\left(S_{2}^{i, j}\right) \tilde{u}_{y}^{i, j, k}(t), \\
I_{4}(t) & :=\sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L}\left(\tilde{u}_{z}^{i . j, k}(t)\right)^{T} \operatorname{Ric}\left(S_{3}^{i, j}\right) \tilde{u}_{z}^{i, j, k}(t) . \tag{A.5}
\end{align*}
$$

By the Riccati equations, defined in (24), $I_{1}(t)=I_{2}(t)=$ $I_{3}(t)=I_{4}(t)=0$, and in view of the adjust equations of the weights (22), the previous inequality is simplified to

$$
\begin{align*}
\dot{V}(t) \leq & -\alpha \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \bar{V}^{i, j, k}(t) \\
& +2 \alpha \sum_{r=1}^{11} \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} \operatorname{tr}\left\{\left(\widetilde{W}_{r}^{i, j}(t)\right)^{T} K_{r} \bar{W}_{r}^{i, j, k}(t)\right\}  \tag{A.6}\\
& +\Psi \dot{V}(t) \leq-\alpha \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} V^{i, j, k}(t)+\Psi .
\end{align*}
$$

Applying Lemma 5, one has $[1-(\mu / \sqrt{V(t)})]_{+} \rightarrow 0$, which completes the proof.

## References

[1] Y. Pinchover and J. Rubinstein, An Introduction to Partial Differential Equations, Cambridge University Press, 2008.
[2] G. D. Smith, Numerical Solution of Partial Differential Equations: Infinite Difference Methods, Clarendon Press, Oxford, UK, 1978.
[3] T. J. R. Hughes, The Finite Element Method, Prentice Hall, Upper Saddle River, NJ, USA, 1987.
[4] R. Fuentes, A. Poznyak, T. Poznyak, and I. Chairez, "Neural numerical modeling for uncertain distributed parameter system," in Proceedings of the International Joint Conference in Neural Networks, pp. 909-916, Atlanta, Ga, USA, June 2009.
[5] S. Haykin, Neural Networks. A Comprehensive Foundation, IEEE Press, Prentice Hall U.S., New York, NY, USA, 2nd edition, 1999.
[6] G. Cybenko, "Approximation by superposition of sigmoidal activation function," Mathematics of Control, Signals and Systems, vol. 2, pp. 303-314, 1989.
[7] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 9871000, 1998.
[8] M. W.M Dissanayake and N. Phan-Thien, "Neural-networkbased approximations for solving partial differential equations," Communications in Numerical Methods in Engineering, vol. 10, no. 3, pp. 195-201, 1994.
[9] N. Mai-Duy and T. Tran-Cong, "Numerical solution of differential equations using multiquadric radial basis function networks," Neural Networks, vol. 14, no. 2, pp. 185-199, 2001.
[10] S. He, K. Reif, and R. Unbehauen, "Multilayer neural networks for solving a class of partial differential equations," Neural Networks, vol. 13, no. 3, pp. 385-396, 2000.
[11] A. Poznyak, E. Sanchez, and W. Yu, Differential Neural Networks for Robust Nonlinear Control (Identification, Estate Estimation an trajectory Tracking), World Scientific, 2001.
[12] F. L. Lewis, A. Yeşildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," IEEE Transactions on Neural Networks, vol. 7, no. 2, pp. 1-11, 1996.
[13] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," IEEE Transactions on Systems, Man and Cybernetics, vol. 24, no. 3, pp. 400-412, 1994.
[14] H. K. Khalil, Nonlinear Systems, Prentice-Hall, Upper Saddle River, NJ, USA, 2002.
[15] R. A. Adams and J. Fournier, Sobolev Spaces, Academic Press, New York, NY, USA, 2nd edition, 2003.
[16] B. Delyon, A. Juditsky, and A. Benveniste, "Accuracy analysis for wavelet approximations," IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 332-348, 1995.
[17] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," IEEE Transactions on Neural Networks, vol. 1, no. 4, pp. 290-295, 1990.
[18] M. R. Islam and N. Alias, "A case study: 2D Vs 3D partial differential equation toward tumour cell visualisation on multicore parallel computing atmosphere," International Journal for the Advancement of Science and Arts, vol. 10, no. 1, pp. 25-35, 2010.
[19] A. Poznyak, Advanced Mathetical Tools for Automatic Control Engineers, Deterministic Technique, vol. 1, Elsevier, 2008.


## Hindawi

Submit your manuscripts at
http://www.hindawi.com


