Neurocomputing 164 (2015) 56–70

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A CNN-based approach for a class of non-standard hyperbolic partial differential equations modeling distributed parameters (nonlinear) control systems

Daniela Danciu

Department of Automation and Electronics, University of Craiova, A.I. Cuza str., no. 13, RO-200585 Craiova, Romania

ARTICLE INFO

Article history:
Received 13 January 2014
Received in revised form
10 December 2014
Accepted 14 December 2014
Available online 14 March 2015

Keywords:
Distributed parameters control systems
Hyperbolic partial differential equations
Cellular neural networks
Method of lines
Control Lyapunov functionals
Overhead crane

ABSTRACT

The present paper considers a systematic approach within the framework of *Neural Mathematics* for constructing a computational procedure. This procedure aims to solve a class of problems arising from the control of the systems with distributed parameters; these systems are modeled by second-order one-dimensional hyperbolic partial differential equations (hPDEs) with non-standard boundary conditions. The procedure reveals an explicit algorithmic parallelism and is mainly based on the combination of two powerful "tools": a convergent *Method of Lines* (MoL) and the *Cellular Neural Network* (CNN) paradigm. The role of the *Courant-Isaacson-Rees rule* and of the *Riemann invariants* for a correct application of the MoL is emphasized. The procedure is illustrated on a control engineering application – the overhead crane with flexible cable – within a more general context which includes modeling based on the *generalized Hamilton variational principle*, synthesis of a stabilizing controller via the *Control Lyapunov Functional* (CLF), qualitative analysis, numerical solving using the proposed computational procedure, numerical simulations and the evaluation of the performances for the closed loop system. The procedure ensures the convergence of the approximation, preserves the basic properties and the Lyapunov stability of the solution of the initial problem and reduces the systematic errors.

1. Introduction

Beginning with the seminal papers of the 19th century, the importance and study of distributed parameter systems (DPS) increased exponentially mainly due to their applications in both engineering and science. The study of DPS encompasses several areas, such as model development, numerical approximations, control design and experimental implementation. In [1] it is stressed the role of approximations for developing or improving the design methods and computational tools for the distributed parameter control systems (DPCS) encountered in engineering as well as in other areas of science and life. Within this context, taking into account the new concern of using neural computers in solving mathematical problems, a *Neural Mathematics* (NM) approach is desirable.

Neural mathematics (*Neuromathematics*) arises as a new interdisciplinary research direction, being at the confluence of the *Computational Mathematics* and *Neurocomputing* fields. As is described in [2], it aims to develop new methods and algorithms for solving both formalized and non-formalized, or weakly-formalized, problems via the logical basis of the neural networks. Under this vision, a *neural network algorithm* is a computational procedure whose main part can be implemented using a neural network.

For formalized or weakly-formalized problems which display a certain natural parallelism, the solutions can be computed by using a neural network which does not need a "learning" procedure based on experimental data in order to achieve the weights of its interconnections. A short list of problems for *Neural Mathematics* includes mathematical problems or tasks arising from natural sciences and engineering applications [2]: (a) solving high-dimensional systems of linear or nonlinear algebraic equations and inequalities [3,4], (b) solving optimization and systems identification problems [5–7], (c) approximating and extrapolating various functions, (d) solving systems of nonlinear differential [8,9], (e) solving partial differential equations (PDEs) [10–16].

Concerning the solving of PDEs problems by using neural networks, several approaches exist. Most of them consider problems for elliptic or parabolic PDEs. The difficulties in coping with hyperbolic PDEs (hPDEs) are mainly due to the phenomenon of discontinuity propagation; these discontinuities are "ignited" by the "mismatch" of the boundary and of the initial conditions.

E-mail address: ddanciu@automation.ucv.ro

http://dx.doi.org/10.1016/j.neucom.2014.12.092
0925-2312/

Standard literature examples contain either hPDE problems with known solutions in close form or with simple e.g. Dirichlet type boundary conditions (BCs). However, this is not the case of the majority of the problems induced by the DPS of engineering or of other fields of science.

The dynamics of DPS modeled by hPDEs can be simulated by using dynamical neural networks, i.e. recurrent neural networks. Among them, Cellular Neural Networks (CNNs), or more general "cell-based neural networks", are desirable due to the rich dynamical behaviors they cover; in [17] the authors show that the dynamics of CNN models is richer than that of PDEs, "either because they do not approximate any PDE models, or because they have a qualitatively different dynamic behavior"

Worth mentioning is that, in all reported applications of CNNs for solving hPDEs, the main role of CNN structures is to implement the high-dimensional system of ordinary differential equations (ODEs) approximating the initial hPDE problem [10,11,15,16]. Also, these works mainly focus on the one-to-one mapping of the approximating system of ODEs onto the CNN structure as well as on the performances of these implementations for the specific applications. Less attention is devoted to the specific procedure used for deriving the approximating system for the initial hPDE problem. Nevertheless, the role of the computational procedure for obtaining a good approximation is equally important, and moreover, a systematic approach for its construction within the new framework of *Neural Mathematics* is a necessity.

The structure of the paper is as follows. In Section 2 we formulate the problem of the present work. Section 3 introduces the computational procedure step-by-step, together with the main "tools" it considers: a convergent *Method of Lines* (MoL) and the *Cellular Neural Network* paradigm. In Section 4 we consider an engineering application in order to illustrate the entire procedure within a more comprehensive context. More precisely, we consider a DPS described by hPDEs with non-standard BCs for which the mathematical model is deduced. Then, the control is synthesized and the mathematical model of the controlled object with distributed parameters is obtained. Applying next the computational procedure, we derive the approximating system for which the simulation results are discussed. In Section 5 we present some concluding remarks and open directions to be followed.

Notations: Throughout this paper we shall use the standard notations for: (i) PDEs $y_{tt}(\sigma,t):=\partial^2 y/\partial t^2$, $y_\sigma(\sigma,t):=\partial y/\partial\sigma$, $(a(\sigma,t))_\sigma :=(\partial/\partial\sigma)a(\sigma,t)$, (ii) the derivative with respect to time variable $\dot{x}(t):=(d/dt)x(t)$, (iii) the derivative with respect to the space variable $x'(\sigma):=(d/d\sigma)x(\sigma)$. $\mathbb{R}^n$ denotes the real n-space, $\mathbb{R}_+$ denotes the positive real axis, $\mathcal{L}^2(0,1)$ denotes the space of square-integrable functions on the interval $(0,1)$, $D=\mathrm{diag}(d_1,\dots,d_n)\in\mathbb{R}^n$ denotes a diagonal matrix.

2. Problem formulation

Consider the class of, possibly nonlinear, distributed parameter systems modeled by a second-order one-dimensional mixed initial boundary value problem for a hyperbolic partial differential equation with non-standard boundary conditions which may include control signals on one or both boundaries. More specifically, we consider the BCs of types enumerated in [18] and written for the system in the Friedrichs form (7) as follows:

- *Dirichlet BCs*

$$\begin{aligned} &v(0,t)+\beta_1 w(0,t)=\varphi_1(t)\\ &w(0,t)+\beta_2 v(0,t)=\varphi_2(t) \end{aligned} \tag{1}$$

- *BCs "controlled by a system of ODEs which is itself controlled by the BCs"*

$$\begin{aligned} &v(0,t)+\beta_1 w(0,t)=c_1'x(t)-\beta_3\varphi(c_0'x(t))+\varphi_1(t)\\ &w(0,t)+\beta_2 v(0,t)=c_2'x(t)-\beta_4\varphi(c_0'x(t))+\varphi_2(t)\\ &\dot{x}(t)=Ax(t)+b_{11}v(0,t)+b_{12}w(0,t)+b_{21}v(1,t)\\ &\qquad +b_{22}w(1,t)-b_0\varphi(c_0'x(t))+f(t), \quad x(0)=x_0 \end{aligned} \tag{2}$$

- *derivative BCs as considered in [19]*

$$\begin{aligned} &\sum_0^{L_1} a_{j1}\frac{d^j}{dt^j}v(0,t)+\sum_0^{K_2} a_{j2}\frac{d^j}{dt^j}w(0,t)=\varphi_1\\ &\sum_0^{K_1} b_{j1}\frac{d^j}{dt^j}v(1,t)+\sum_0^{L_2} b_{j2}\frac{d^j}{dt^j}w(1,t)=\varphi_2 \end{aligned} \tag{3}$$

- "*BCs described by some nonlinear Voltera operators acting on the boundary of the definition domain of PDE* [20]"

The specific problems of such DPCS, arising either from engineering applications or from other fields of science, include modeling, basic properties of the solution, solution computation, as well as synthesis of the stabilizing controller and evaluation of the performances for the closed loop system. Since for a large number of DPS the analytical solution is unknown or difficult to calculate, a good choice is to use the numerical approximated solutions. It is thus important that the numerical methods/procedures used for their computation ensure the convergence of the approximating solution to the true one and the preservation of the basic properties of the solution, i.e. existence, uniqueness, data dependence, stability in the sense of Lyapunov.

This paper has two main objectives. The first one is to propose a systematic approach within the framework of *Neural Mathematics* for numerically solving the aforementioned class of hPDE problems. This is a formalized problem for NM because one can derive an explicit solution algorithm which, at this time, can be at least partially solved by using the neural networks. It is the case of an explicit algorithmic parallelism, where each step can be considered as a sub-problem for which a specific *Neurocomputing* approach may be developed. The procedure has to ensure the convergence of the approximation as well as the preservation of the basic properties of the solutions of the initial problem. Also, it has to ensure the optimum neural network structure for numerical implementation in order to reduce the systematic errors, storage and computational effort.

The second objective is to illustrate the proposed procedure on an engineering application for DPCS without lossless/distorsionless phenomena and modeled by coupled hPDEs in the normal form of Riemann invariants – the overhead crane with flexible cable, a benchmark for modeling and control theory. This study will give also an overview about the role and the place of the entire procedure for a control engineering application.

3. A Neural Mathematics approach for developing the computational procedure

As it has been clear from the previous sections, the paper's defining technical feature might be described as *deriving a "neural algorithm", based on a convergent Method of Lines and incorporating a cell-based neural network, for applications described by PDEs of hyperbolic type*.

The outcome is in the line of *Neural Mathematics*, a new research area of *Computational Mathematics*, for which we introduce here the main features, as stated in [2]. NM "provides the