Neural Networks 49 (2014) 74–86

Contents lists available at ScienceDirect

Neural Networks

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

An adaptive recurrent neural-network controller using a stabilization matrix and predictive inputs to solve a tracking problem under disturbances[☆]

Michael Fairbank[a,*], Shuhui Li[b], Xingang Fu[b], Eduardo Alonso[a], Donald Wunsch[c]

[a] Department of Computer Science, City University London, London EC1 V0B, UK
[b] Department of Electrical & Computer Engineering, the University of Alabama, Tuscaloosa, AL 35487, USA
[c] Department of Electrical & Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409-0040, USA

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form 28 September 2013
Accepted 29 September 2013

Keywords:
Tracking problem
Stabilization matrix
Recurrent neural networks
Exploding gradients
Vector control

ABSTRACT

We present a recurrent neural-network (RNN) controller designed to solve the tracking problem for control systems. We demonstrate that a major difficulty in training any RNN is the problem of exploding gradients, and we propose a solution to this in the case of tracking problems, by introducing a stabilization matrix and by using carefully constrained context units. This solution allows us to achieve consistently lower training errors, and hence allows us to more easily introduce adaptive capabilities. The resulting RNN is one that has been trained off-line to be rapidly adaptive to changing plant conditions and changing tracking targets.

The case study we use is a renewable-energy generator application; that of producing an efficient controller for a three-phase grid-connected converter. The controller we produce can cope with the random variation of system parameters and fluctuating grid voltages. It produces tracking control with almost instantaneous response to changing reference states, and virtually zero oscillation. This compares very favorably to the classical proportional integrator (PI) controllers, which we show produce a much slower response and settling time. In addition, the RNN we propose exhibits better learning stability and convergence properties, and can exhibit faster adaptation, than has been achieved with adaptive critic designs.

1. Introduction

In this paper, we propose a recurrent neural-network controller to solve the tracking problem. We consider a real-world test problem from electrical power and energy applications, and this forms the motivation for development of the neural-controller presented in this paper. The energy application we consider is that of a three-phase grid-connected dc/ac voltage-source converter, or grid-connected converter (GCC) for short.

A GCC is usually employed to interface between the dc and ac sides of an electric power system. Typical converter configurations containing a GCC include: (1) a dc/dc/ac converter for solar, battery and fuel cell applications (Figueres, Garcerá, Sandia, Gonzalez-Espin, & Rubio, 2009; Wang & Nehrir, 2007), (2) a dc/ac converter for STATCOM applications (Carrasco et al., 2006; Luo et al., 2009), and (3) an ac/dc/ac converter for wind power and HVDC applications (Carrasco et al., 2006; Mullane, Lightbody, & Yacamini, 2005; Pena, Clare, & Asher, 1996; Rabelo, Hofmann, da Silva, de Oliveira, & Silva, 2009; Xu & Wang, 2007).

In all these applications, controlling the GCC efficiently and making it maintain a desired state (a tracking problem) is crucial for the reliability and stability of both the ac and the dc subsystems. The controller must be able to track any reference command variations quickly. For example, these might occur in wind power and photovoltaic applications as a result of sudden variations in the wind speed or solar irradiation levels.

Classically the tracking problem has been addressed using proportional integrator (PI) controllers (Pena et al., 1996; Qiao, Venayagamoorthy, & Harley, 2009). Limitations of these methods are that they can have slow response times to changing reference commands, can take considerable time to settle down from oscillating around the target reference state (Dannehl, Wessels, & Fuchs, 2009), and have difficulty recovering from short-circuit faults in either the generator or the power-grid. Hence neural-network based

☆ This work was supported in part by the U.S. National Science Foundation under Grant EECS 1102038/1102159, the Mary K. Finley Missouri Endowment, and the Missouri S&T Intelligent Systems Center.

* Corresponding author. Tel.: +44 2070404049.
E-mail addresses: michael.fairbank@virgin.net (M. Fairbank), sli@eng.ua.edu (S. Li), xfu@crimson.ua.edu (X. Fu), E.Alonso@city.ac.uk (E. Alonso), dwunsch@mst.edu (D. Wunsch).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.09.010

solutions have been proposed to overcome these difficulties, in this control problem and related ones (Li, Fairbank, Wunsch, & Alonso, 2012; Park, Harley, & Venayagamoorthy, 2004; Qiao, Harley, & Venayagamoorthy, 2008, 2009; Qiao, Venayagamoorthy, & Harley, 2008; Venayagamoorthy, Harley, & Wunsch, 2002, 2003).

These neural-network approaches have mainly been based on Adaptive Critic Designs (ACDs) (Prokhorov & Wunsch, 1997; Wang, Zhang, & Liu, 2009; Werbos, 1992). ACDs use two neural networks: an action network and a critic network. The critic network provides feedback to the action network, allowing the action network to be trained on-line and in real-time, and therefore to be continually learning and adaptive during plant operation. However useful this double network design may be, proving convergence of the two continually learning networks at once is challenging. In fact, just proving the convergence of the critic network on its own is not trivial, since critic learning algorithms generally are not true gradient descent (Barnard, 1993). The general instability in this case is proven by Werbos (1998), and divergence examples of concurrent actor–critic learning exist (Fairbank & Alonso, 2012). In practice, the best course of action is not to allow such a system to be continually autonomously learning while controlling a delicate or critical industrial system. Qiao, Harley et al. (2009), Qiao, Venayagamoorthy et al. (2008) and Venayagamoorthy et al. (2003) overcome this problem by first training the action and critic networks concurrently off-line, and then freezing the action neural network and dispensing with the critic network for on-line operation of the plant. This solution of course neutralizes the adaptive benefits of the ACD architecture. Adaptive behavior is often recreated by using lagged state inputs for the action network (e.g. Venayagamoorthy et al., 2003), effectively creating a time-delay neural network. Modest improvements over PI controllers are made using ACDs, for example, see Qiao, Harley et al. (2009) and Qiao, Venayagamoorthy et al. (2008).

To improve on this situation further, we are using an architecture that uses an action network only, but which is trained off-line through backpropagation through time (BPTT) (Werbos, 1990). This approach has the advantage that the learning algorithm is true gradient descent on the cost-to-go function, and so convergence is assured (assuming a smooth error minimization surface, and a sufficiently small learning rate). Also, since the BPTT algorithm is true gradient descent, learning is guaranteed to find a true local minimum of the training error. In contrast, the ACD learning algorithms used by the aforementioned references are not true gradient descent, and hence the learning progress appears stochastic, and the minimum obtained is often not as low as that obtained by BPTT.

Recent studies show how a single action network can be trained with BPTT to control a GCC under fixed plant behavior (Li et al., 2012). However, for real-life applications, the plant behavior can change; system parameters can exhibit random variations; voltages coming into the system from the power grid can fluctuate; short circuits can occur. Hence the action network needs to become more adaptive than demonstrated by Li et al. (2012).

Adaptive behavior can be enabled by modifying the action network to have neural-context units which respond to the changing behavior of the plant, thus making the action network into a RNN. This design for adaptation is potentially much faster than the adaptation carried out by ACDs, in that the weights of the RNN do not need to change to accommodate adaptation. This is referred to as fixed-weight adaptive behavior by Prokhorov, Feldkamp, and Tyukin (2002), and can produce almost instantaneous adaptation. In contrast, ACD adaptive behavior takes place by retraining the two neural networks involved, and this kind of learning is slow.

A major difficulty with using a RNN for the controller is that because data cycles around the RNN many times, learning gradients may decay rapidly to zero, or alternatively, the learning gradients may rapidly become excessively large, and both of these problems cause difficulties for learning by gradient descent. These problems are known as "vanishing" or "exploding" gradients, respectively, in the RNN literature (Hochreiter & Schmidhuber, 1997). While Hochreiter and Schmidhuber (1997) address the problem of vanishing gradients, our paper attempts to minimize the problem of exploding gradients for the tracking problem domain, through the introduction of a "stabilization matrix", and carefully constrained context units.

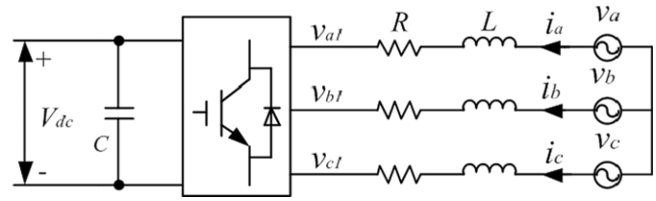

Fig. 1. Grid-connected converter schematic.

The novelties of this paper include: (1) the stabilizing matrix, which is a hand-picked neural weight matrix which represents some pre-learned basic control behavior, allowing the learning algorithm to concentrate on learning the more advanced nuances of behavior and thus to acquire improved solutions than otherwise possible; (2) a theoretical discussion on the importance of handling the problem of exploding gradients in RNNs; and (3) a design for using the predicted as well as the previous inputs that allows the neural network to behave adaptively on-line, despite the training process having taken place entirely off-line.

The rest of the paper is structured as follows: the basic topology of the GCC neural-network vector controller, and how to train it to solve the tracking problem using BPTT, is presented in Section 2. Section 3 shows the stabilization matrix approach, which enhances the neural-network training speed and stability when the system matrix and the control voltage matrix are fixed. Section 4 presents how a RNN is trained to behave adaptively on-line when these matrices vary, which relies upon novel extra context inputs to the neural controller. Simulation experiments are given in Section 5. These include GCC experiments for the neural vector controller, under variable and dynamic conditions, and a comparison to two conventional control methods, showing the advantages of our method. Also an experiment is included that demonstrates how the stabilization-matrix method can be extended to the case of non-invertible matrices. The paper concludes in Section 6 with a summary and a discussion of further work, and Appendix which proves that the method for adaptation which we used is flexible enough to work in a greater variety of applications than just our chosen experiments.

2. Neural-network vector-control architecture

Fig. 1 shows schematics of the GCC, in which a dc-link capacitor appears on the left, and a three-phase voltage source, representing the voltage at the Point of Common Coupling (PCC) of the ac system, appears on the right. In this diagram the capacitor would be connected to the electrical generator (for example the wind turbine, or photovoltaic array) and has a dc voltage represented by V_{dc}, and the three voltages v_a, v_b and v_c would represent the three-phase voltage of the electric power grid.

The power transferred between the grid and the converter includes active power and reactive power. The purpose of the GCC controller is to control the active and reactive power transferred.

The circuit contains 3-phase ac-voltages v_a, v_b, and v_c, with corresponding 3-phase ac-currents i_a, i_b and i_c. By transforming to a rotating frame of reference with axes d and q, as described by Li, Haskew, Hong, and Xu (2011), it is possible to largely eliminate the ac-sinusoidal variations, and to transform these three dimensions