



Adaptive dynamic programming-based design of integrated neural network structure for cooperative control of multiple MIMO nonlinear systems

Nguyen Tan Luy

Industry University of Hochiminh City, Vietnam



ARTICLE INFO

Article history:

Received 9 September 2015

Received in revised form

2 May 2016

Accepted 14 May 2016

Communicated by Weiming Xiang

Available online 25 May 2016

Keywords:

Adaptive dynamic programming

Neural network

Actor-critic

Cooperative multi-agent

Graph theory

ABSTRACT

Solving cooperative problems for multi-agent systems, in which the agent's artificial behaviors are similar to naturally biological behaviors of agents in practice, is a major challenge. The problems become more complex if the controlled agents are multi-input and multi-output (MIMO) nonlinear systems lacking knowledge of internal system dynamics and affected by external disturbances. In this paper, firstly, based on adaptive dynamic programming, three neural networks (NNs) (actor/disturber/critic) of control schemes for two-player games are integrated into the structure with only one NN, known as integrated NN (INN), with the aim of reducing computational complexity and waste of resources. Secondly, an INN weight update law and an online control algorithm, which updates parameters in one iterative step, are designed to find H_∞ optimal cooperative control solutions. With the aid of Lyapunov theory, we prove that the INN weight approximation errors and the cooperative tracking errors are uniformly ultimately bounded (UUB), and the system parameters converge to the approximately optimal values. Finally, two simulation studies, one of which is compared to three-NN structures in existing literature, are carried out to show the effectiveness of the proposed INN structure.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Motivated by works of a flock of birds, colony of ants, and other naturally biological systems, cooperative problems for agents have received much attention in recent years [1–10]. In the cooperative systems, each agent exchanges information with others to do his actions, which not only satisfies his own but also helps his swarm reach the common goals. In practice, the agents are so complex that their models must be presented as a multi-input and multi-output (MIMO) nonlinear system lacking knowledge of internal system dynamics. They are also impacted by external disturbance. Therefore, solving the cooperative problem, in which the agent's artificial behaviors are similar to naturally biological behaviors of practical agents, is always a major challenge.

In the cooperative systems, graph theory is used to establish a communication graph among the agents [1–10]. Depending on the presence of leader nodes in the graph, the cooperative problem can be divided into two types: the cooperative regulator problem and the cooperative tracking problem. In this paper, the latter is focused, where both state trajectories of follower and leader nodes are synchronized [1,2,5–9].

Inspired by the nonlinear approximation abilities, NN structures have been constantly developed for cooperative adaptive control systems [9,6–8,10]. In [6], a method using the distributed NNs with

linearly parametrized structures is proposed to identify uncertainty dynamics directly. Because of dependence on the analytic solution of the Riccati inequality formed by the given matrices, this method is only applied to a specific class of nonlinear systems. In [9], NN structures with a suitable basis set of action functions are selected to design a cooperative robust adaptive control law for linearized dynamic agents tracking the trajectory of the leader node. In [7], NN structures with radial basis functions (RBFs) are used to approximate the uncertain dynamics. The RBF NN weight update laws are based on the upper bound values of the NN weights, which are difficult to determine in practical cooperative applications. In [8], to identify the unknown non-linearity, linear-in-parameter NNs with the basis functions such as sigmoid or Gaussians are chosen. The NN weight update laws are based on the solution of Lyapunov equations derived from the sliding mode control method. In [10], the self-structuring NNs are designed to identify the agent's unknown dynamics directly, where the basic functions are sigmoid in the hidden layer, but the number of hidden nodes is dynamically computed and the sign of input dynamic functions in the system must be known.

Most of the NN-based methods mentioned in the paragraph above do not minimize any long-term performance functions, thus they are not considered as optimal control methods. Recently, an actor-critic NN (ACNN) structure from the adaptive dynamic programming (ADP) [11] has been developed for the cooperative

systems [4,12], where optimal value functions and control policies are approximated by critic and actor NNs, respectively, but all node dynamics in communication graphs are limited to double integrator systems.

To solve cooperative problems of nonlinear systems impacted by external disturbances, ACNNs are developed. In [13–15], ACNNs are used in online algorithms, which synchronously update parameters of NNs, to find the Hamilton–Jacobi–Isaac (HJI) solutions for the non-zero-sum games of the two-player system [14,15] or the multi-player system [13]. However, there are still disadvantages applying the ACNNs, especially when a structure consists of three NNs, namely AC3NN, is exploited [16]. In AC3NN, one NN (critic NN) is used for approximating the optimal value function, the other two (actor NNs) are used for approximating the controller (first player) and the disturber (second player), respectively. Therefore, the greater number of agents in the cooperative system is, the more the number of NNs is tripled. Assume that AC3NN is used for the cooperative multiple MIMO systems which contain a large number of feedback states. Then the number of NN weights and the dimension of NN basis functions representing combinations of the states will significantly increase (see [4,13] for more survey). As a result, besides wasting resources, AC3NN can lead to a significant increase in computational complexity. Another disadvantage of AC3NN is that the algorithms bring two iterative loops, i.e. while the parameters of the disturbance law are updated in an iterative loop, the parameters of the control law must be waited for updating in the other. Consequently, they result in low efficiency even being applied to a single agent [17].

To overcome the disadvantages of using more than one NN, the structures with a single NN (SOLA—Single Online Approximator) are proposed to learn the HJI solution of the affine system [3] or the HJB solution of the MIMO nonlinear system [18]. However, since the SOLA weight update laws require the knowledge of the system dynamics [3] or ignore disturbances [18], SOLA techniques are still restricted. In addition, until now, they have been only applied to the uncooperative problems.

In contrast, the control structures in [17,19] not only use a single NN but also remove the assumption of the known system dynamics [20–22]. In [17], at every loop step, parameters of the control law and the disturbance law must be constantly held in a period of time to collect the sample set for training the NN. In [19], external disturbance of each player is ignored. Furthermore, the NN structures in [17,19] are designed for the uncooperative problem.

Contributions: Comparing with the existing works, the main contribution of this paper include three following aspects:

1. Design an integrated neural network (INN) structure in contrast to the existing work in [4]. To the best of our knowledge, this paper may be the first work that three NNs in the ADP method are integrated into only one for the cooperative problem of the multiple MIMO nonlinear systems. This integration aims to reduce computational complexity and resources.
2. Design an INN weight update law, in which the knowledge of internal system dynamics is not required, and an online H_∞ optimal cooperative control algorithm, in which the INN weight parameters and the parameters of the control and disturbance laws are simultaneously and continuously updated in one iterative step.
3. Prove that, with the aid of Lyapunov theory, cooperative tracking errors of the closed-loop system and the INN weight approximation errors are UUB, and the value function, the H_∞ optimal cooperative control law and the worst disturbance law converge to approximately optimal values.
4. The effectiveness of our method is shown by comparing the simulation results of both INN and existing three-NN structures in [14,21].

The rest of the paper is organized as follows. In Section 2, we introduce a distributed communication graph and derive overall cooperative tracking error dynamics of the MIMO nonlinear systems. In Section 3, we design the INN structure, the weight update law and the control algorithm. Simulation studies are implemented in Section 4. Finally, a brief conclusion is drawn in Section 5.

Notations: \mathbb{R} , \mathbb{R}^n , and $\mathbb{R}^{n \times m}$ are the set of real numbers, the n -dimensional Euclidean space, and the set of all real $n \times m$ matrices, respectively. $\|\cdot\|$ defines the vector or matrix norm in \mathbb{R}^n or $\mathbb{R}^{n \times m}$, respectively. $V_x \triangleq \partial V / \partial x$ denotes the gradient of V in x . The superscript T is used for the transpose. \otimes denotes the Kronecker product with the properties $(X \otimes Y)^T = X^T \otimes Y^T$, $\beta(X \otimes Y) = (\beta X) \otimes Y = X \otimes (\beta Y)$, where X and Y are matrices and β is a scalar. I_n denotes a n -dimensional identity matrix. $\bar{1}_n = [1, \dots, 1]^T \in \mathbb{R}^n$. $\text{diag}(\alpha_i)$ is a diagonal matrix whose the diagonal element i is α_i . $L_2[0, \infty)$ is the Banach space, for $\forall d(t) \in L_2[0, \infty)$, $\int_0^\infty \|d(t)\|^2 dt < \infty$.

2. Graph theory and cooperative tracking error dynamics

2.1. Distributed communication graph theory

Consider N agents in a cooperative system. The distributed communication of the multiple agents can be represented by a directed graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, where the agents are characterized by the set of nodes $\mathcal{V} = \{s_0, \dots, s_N\}$, where s_0 is a leader node. Relationships among the agents are determined by the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ with a connectivity weight matrix $\mathcal{A} = [a_{ij}]$, where $a_{ii} = 0$, $a_{ij} > 0$ for $a_{ij} \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. If the states of the agent s_i is available to s_j , then s_j is a neighbor of s_i . All neighbors of s_i give a neighbor set that is defined by $\mathcal{N}_i = \{j : s_j \in \mathcal{V}, (s_i, s_j) \in \mathcal{E}\}$. Define a graph Laplacian matrix $\mathcal{L} = \mathcal{H} - \mathcal{A} \in \mathbb{R}^{(N+1) \times (N+1)}$, where $\mathcal{H} = \text{diag}(h_i)$, $h_i = \sum_{j \in \mathcal{N}_i} a_{ij}$. Note that row sums of \mathcal{L} are equal to zero.

A directed path is a sequence of ordered edges $(s_i, s_{i+1}) \in \mathcal{E}$, $i = 0, \dots, N-1$. If a directed path from s_i to s_j exists such that $\forall s_i, s_j \in \mathcal{V}$, $s_i \neq s_j$, then the directed graph is strongly connected. The graph is directed spanning tree if the set $\{s_1, \dots, s_N\}$ exits at least one node with a directed path to all other nodes.

The connectivity matrix between the agent i and the leader is defined as

$$Z = \text{diag}[z_1, z_2, \dots, z_N] \quad (1)$$

where $z_i = 1$ if the agent i connects to the leader, and $z_i = 0$, otherwise.

2.2. Node dynamics

Consider N agents are distributed on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$. Agent's node dynamics is a MIMO nonlinear system presented by m ($m \geq 2$) equations

$$\begin{cases} \dot{x}_{il} = f_{il}(\bar{x}_{il}) + g_{il}(\bar{x}_{il})x_{i(l+1)} + k_{il}(\bar{x}_{il})d_{il}, l = 1, \dots, m-1 \\ \dot{x}_{im} = f_{im}(\bar{x}_{im}) + g_{im}(\bar{x}_{im})u_{im} + k_{im}(\bar{x}_{im})d_{im} \end{cases} \quad (2)$$

where for all $i = 1, \dots, N$ and $l = 1, \dots, m$, $\bar{x}_{il} = [x_{i1}^T, \dots, x_{il}^T]^T \in \mathbb{R}^{n_1 + \dots + n_l}$ with $x_{il} \in \mathbb{R}^{n_l}$ being a state vector assumed available for feedback. $u_{im} \in \mathbb{R}^{n_{m+1}}$ is a control input vector. $d_{il} \in \mathbb{R}^{n_l}$ is a unknown bounded disturbance vector such that $d_{il} \in L_2[0, \infty)$. $f_{il}(\bar{x}_{il}) \in \mathbb{R}^{n_l}$, $k_{il}(\bar{x}_{il}) \in \mathbb{R}^{n_l \times n_l}$ and $g_{il}(\bar{x}_{il}) \in \mathbb{R}^{n_l \times n_{l+1}}$ are the smooth nonlinear function vector and matrices, respectively. Furthermore, $f_{il}(\bar{x}_{il})$ denotes unknown internal dynamics of the system.

Download English Version:

<https://daneshyari.com/en/article/4947747>

Download Persian Version:

<https://daneshyari.com/article/4947747>

[Daneshyari.com](https://daneshyari.com)