Robotics and Autonomous Systems 59 (2011) 22–33

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Self-learning fuzzy logic controllers for pursuit–evasion differential games

Sameh F. Desouky*, Howard M. Schwartz

Department of Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada

ARTICLE INFO

Article history:
Received 17 February 2010
Accepted 13 September 2010
Available online 1 October 2010

Keywords:
Differential game
Function approximation
Fuzzy control
Genetic algorithms
Q(λ)-learning
Reinforcement learning

ABSTRACT

This paper addresses the problem of tuning the input and the output parameters of a fuzzy logic controller. The system learns autonomously without supervision or a priori training data. Two novel techniques are proposed. The first technique combines Q(λ)-learning with function approximation (fuzzy inference system) to tune the parameters of a fuzzy logic controller operating in continuous state and action spaces. The second technique combines Q(λ)-learning with genetic algorithms to tune the parameters of a fuzzy logic controller in the discrete state and action spaces. The proposed techniques are applied to different pursuit–evasion differential games. The proposed techniques are compared with the classical control strategy, Q(λ)-learning only, reward-based genetic algorithms learning, and with the technique proposed by Dai et al. (2005) [19] in which a neural network is used as a function approximation for Q-learning. Computer simulations show the usefulness of the proposed techniques.

1. Introduction

Fuzzy logic controllers (FLCs) are currently being used in engineering applications [1,2] especially for plants that are complex and ill-defined [3,4] and plants with high uncertainty in the knowledge about its environment such as autonomous mobile robotic systems [5,6]. However, FLC has a drawback of finding its knowledge base which is based on a tedious and unreliable trial and error process. To overcome this drawback one can use supervised learning [7–11] that needs a teacher or input/output training data. However, in many practical cases the model is totally or partially unknown and it is difficult or expensive and in some cases impossible to get training data. In such cases it is preferable to use reinforcement learning (RL).

RL is a computational approach to learning through interaction with the environment [12,13]. The main advantage of RL is that it does not need either a teacher or a known model. RL is suitable for intelligent robot control especially in the field of autonomous mobile robots [14–18].

1.1. Related work

Limited studies have applied RL alone to solve environmental problems but its use with other learning algorithms has increased. In [19], a RL approach is used to tune the parameters of a FLC. This approach is applied to a single case of one robot following another along a straight line. In [15,20], the authors proposed a hybrid learning approach that combines a neuro-fuzzy system with RL in a two-phase structure applied to an obstacle avoidance mobile robot. In phase 1, supervised learning is used to tune the parameters of a FLC then in phase 2, RL is employed so that the system can re-adapt to a new environment. The limitation in their approach is that if the training data are hard or expensive to obtain then supervised learning cannot be applied. In [21], the authors overcame this limitation by using Q-learning as an expert to obtain training data. Then the training data are used to tune the weights of an artificial neural network controller applied to a mobile robot path planning problem.

In [22], a multi-robot pursuit–evasion game is investigated. The model consists of a combination of aerial and ground vehicles. However, the unmanned vehicles are not learning. They just do the actions they received from a central computer system. In [23], the use of RL in the multi-agent pursuit–evasion problem is discussed. The individual agents learn a particular pursuit strategy. However, the authors do not use a realistic robot model or robot control structure. In [24], RL is used to tune the output parameters of a FLC in a pursuit–evasion game.

A number of articles used the fuzzy inference system (FIS) as a function approximation with Q-learning [25–28] however these works have the following disadvantages: (i) the action space is considered to be discrete and (ii) only the output parameters of the FIS are tuned.

1.2. Paper motivation

The problem assigned in this paper is that we assume that the pursuer/evader does not know its control strategy. It is not told which actions to take so as to be able to optimize its control

* Corresponding author. Tel.: +1 613 520 2600x5725.
E-mail addresses: sameh@sce.carleton.ca, samehfarahat@gmail.com (S.F. Desouky), schwartz@sce.carleton.ca (H.M. Schwartz).

0921-8890/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2010.09.006

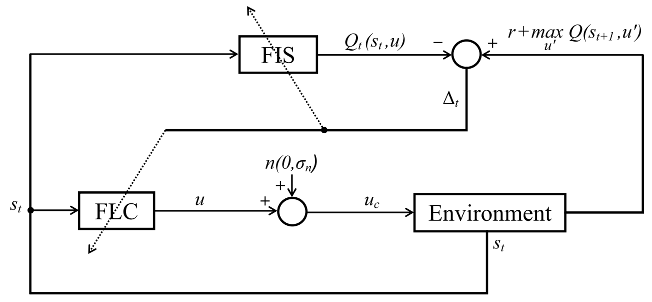

Fig. 1. The proposed QLFIS technique.

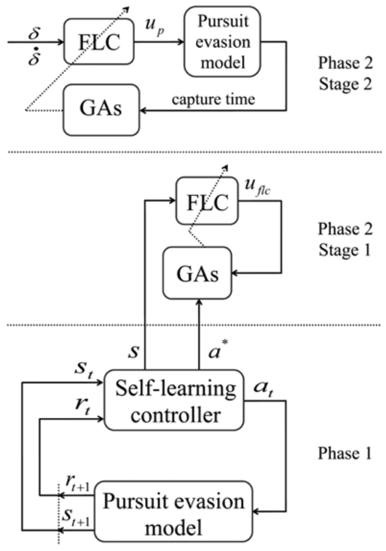

Fig. 2. The proposed QLBGFC technique.

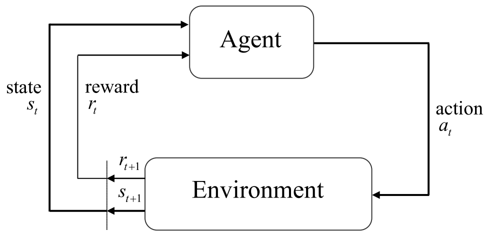

Fig. 3. Agent–environment interaction in RL.

strategy. We assume that we do not even have a simplistic PD controller strategy. The learning goal is to make the pursuer/evader able to self-learn its control strategy. It should do that on-line by interaction with the evader/pursuer.

From several learning techniques we choose RL. RL methods learn without a teacher, without anybody telling them how to solve the problem. RL is related to problems where the learning agent does not know what it must do. It is the most appropriate learning technique for our problem.

However, using RL alone, in most cases, has the limitation in that it is too hard to visit all the state–action pairs. We try to cover most of the state–action space but we cannot cover all the space. In addition, there are hidden states that are not taken into consideration due to the discretization process. Hence RL alone cannot find the optimal strategy.

The proposed Q(λ)-learning based genetic fuzzy controller (QLBGFC) and the proposed Q(λ)-learning fuzzy inference system (QLFIS) are two novel techniques used to solve the limitation in RL. The limitation is that the RL method is designed only for discrete state–action spaces. Since we want to use RL in the robotics domain which is a continuous domain, then we need to use some type of function approximation such as FIS to generalize the discrete state–action space into a continuous state–action space. Therefore, from the RL point of view, a FIS is used as a function approximation to compensate for the limitation in RL. And from the FIS point of view, RL is used to tune the input and/or the output parameters of the fuzzy system especially if the model is partially or completely unknown. Also, in some cases it is hard or expensive to get a priori training data or a teacher to learn from. In this case, the FIS is used as an adaptive controller whose parameters are tuned on-line by RL. Therefore, combining RL and FIS has two objectives; to compensate the limitation in RL and to tune the parameters of the FLC.

In this paper, we design a self-learning FLC using the proposed QLFIS and the proposed QLBGFC. The proposed QLBGFC is used when the state and the action spaces can be discretized in such a way that make the resulting state and action spaces have acceptable dimensions. This can be done, as we will see in our case, if the state and the action values are bounded. If not then the proposed QLFIS will be suitable. In this work and for the comparatively purpose, we will use both of the proposed techniques.

The learning process in the proposed QLFIS is performed simultaneously as shown in Fig. 1. The proposed QLFIS is used *directly* with the continuous state and action spaces. The FIS is used as a function approximation to estimate the optimal action-value function, $Q^*(s, a)$, in the continuous state and action spaces while the Q(λ)-learning is used to tune the input and the output parameters of both the FIS and the FLC.

In the proposed QLBGFC, the learning process is performed sequentially as shown in Fig. 2. The proposed QLBGFC can be considered as *indirect* method of using function approximation. First, in phase 1, the state and the action spaces are discretized and Q(λ)-learning is used to obtain an estimate of the desired training data set, (s, a^*). Then this training data set is used by genetic algorithms (GAs) in phase 2 stage 1 to tune the input and the output parameters of the FLC which are used at the same time to generalize the discrete state and action values over the continuous state and action spaces. Finally in phase 2 stage 2, the FLC is further tuned during the interaction between the pursuer and the evader.

The proposed techniques are applied to two pursuit–evasion games. In the first game, we assume that the pursuer does not know its control strategy whereas in the second game, we increase the complexity of the system by assuming that both the pursuer and the evader do not know their control strategies or the other's control strategy.

The rest of this paper is organized as follows: some basic terminologies for RL, FIS and GAs are reviewed in Section 2, Section 3 and Section 4, respectively. In Section 5, the pursuit–evasion game is described. The proposed QLFIS and the proposed QLBGFC techniques are described in Section 6 and Section 7, respectively. Section 8 presents the computer simulation and the results are discussed in Section 9. Finally, conclusion and future work are discussed in Section 10.

2. Reinforcement learning

Agent–environment interaction in RL is shown in Fig. 3 [12]. It consists mainly of two blocks, an agent which tries to take actions so as to maximize the discounted return, R, and an environment which provides the agent with rewards. The discounted return, R_t, at time t is defined as

$$R_t = \sum_{k=0}^{\tau} \gamma^k r_{t+k+1} \quad (1)$$