ISA Transactions ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

ISA Transactions

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

Research article

Fuzzy Lyapunov Reinforcement Learning for Non Linear Systems

Abhishek Kumar*, Rajneesh Sharma

Division of Instrumentation and Control Engineering, Netaji Subhas Institute of Technology, Dwarka Sector 3, New Delhi 110078, India

ARTICLE INFO

Article history:
Received 23 August 2016
Received in revised form
25 November 2016
Accepted 19 January 2017

Keywords:
Reinforcement learning
Fuzzy Q learning
Fuzzy Lyapunov RL
Inverted pendulum
Rotational/Translational Proof-Mass Actuator

ABSTRACT

We propose a fuzzy reinforcement learning (RL) based controller that generates a stable control action by lyapunov constraining fuzzy linguistic rules. In particular, we attempt at lyapunov constraining the consequent part of fuzzy rules in a fuzzy RL setup. Ours is a first attempt at designing a linguistic RL controller with lyapunov constrained fuzzy consequents to progressively learn a stable optimal policy. The proposed controller does not need system model or desired response and can effectively handle disturbances in continuous state-action space problems. Proposed controller has been employed on the benchmark Inverted Pendulum (IP) and Rotational/Translational Proof-Mass Actuator (RTAC) control problems (with and without disturbances). Simulation results and comparison against a) baseline fuzzy Q learning, b) Lyapunov theory based Actor-Critic, and c) Lyapunov theory based Markov game controller, elucidate stability and viability of the proposed control scheme.

1. Introduction

Real world systems are inherently nonlinear; designing a controller for a truly nonlinear system is quite a challenge. In most of the cases, one requires either partial or full system information for controller design. Conventional controller design techniques are best when one has access to the system model, but it is pretty difficult to construct a flawless model of any system without making idealizing assumptions. System identification techniques [1] may be employed to get an approximate model and a model based control technique [2] could be used. Alternatively, model free control techniques could be used where no system information is available [3].

An important issue in the control of uncertain nonlinear systems is that they exhibit multiple equilibrium states making their stability analysis significantly complex. Several approaches have been proposed in literature to guarantee stability of the designed controller, e. g., Describing function, and Lyapunov's method [2]. Lyapunov's Stability theorem is the most general and widely used one for establishing stability of controllers for nonlinear systems.

In a recent work [5], lyapunov theory has been used for designing a stable RL controller. The authors have proposed a lyapunov theory based Markov game fuzzy controller. The approach seeks to infuse stability in the Markov game based RL control by hybridizing it with a lyapunov theory based action. Lyapunov theory based Markov game controller is tested on three benchmark control problems. The approach, though good, has much higher computational complexity because at each iteration; a game is played between the controller and the disturbances which is solved using linear programming. In contrast, our approach does not involve any linear programming solution, and is conceptually simpler.

In another recent approach [6], authors have proposed a policy iteration based actor critic [8] RL formulation. The technique uses lyapunov theory for generating a stable control policy. The approach is computationally expensive as it uses a two step RL based optimization process: a) value function approximation and b) policy iteration. Furthermore, there are convergence issues when function approximators such as neural network or fuzzy systems are used as it is a difficult proposition to make two function approximators converge in tandem. Our approach, on the other hand, is a model free single step value iteration based RL technique.

Reinforcement learning [8] has evolved as an effective technique for designing self learning, model free, adaptive controllers. An important advantage of RL paradigm is that it puts virtually no condition on the system, i.e., the system could be nonlinear, time varying, stochastic, and even the desired response is not a pre requisite for designing an efficient controller. The controller evolves based on online experiential information gained while it attempts to control an unknown system. However, most of the RL controllers designed so far [4] do not offer any guaranty on stability of the controlled system. Our attempt herein is to address the stability issue in designing of RL based controllers by using lyapunov constrained linguistic rules, in particular, the rule

* Corresponding author.
E-mail addresses: akumar.ju09@gmail.com (A. Kumar), rajneesh496@gmail.com (R. Sharma).

http://dx.doi.org/10.1016/j.isatra.2017.01.026
0019-0578/

consequents. Ours is a first attempt at designing not only a linguistic fuzzy RL control but also a stable one based on the lyapunov theory.

In our earlier works, we sought to infuse stability into RL based controllers by (i) hybridizing RL based control action with a lyapunov theory based action thereby generating a stable hybrid controller [5], and (ii) choosing an action from the controller's action set that satisfies lyapunov's stability condition [7]. Our current work injects two novel ideas into the RL based controller formulation: a) Proposed controller is a linguistic RL controller, i.e., controller discovers an optimal linguistic action and not an optimal discrete action (as in the case of a conventional RL controller), and b) The optimal linguistic action satisfies lyapunov stability condition or we lyapunov constrain the linguistic consequent of a fuzzy rule. Finally, the control action to be applied to the system is derived from this lyapunov constrained linguistic action. Some other key contributions of our approach are: c) it is simple to implement with just one function approximator unlike the actor-critic RL formulation [8] wherein simultaneous tuning of two function approximators is required to achieve an efficient convergence to the optimal solution, d) our approach is scalable to higher dimensional or complex problems as the computational burden does not increase exponentially with the dimensionality of the system.

The approach proposed herein is novel in the above aspects in the overall RL based control paradigm. We have used fuzzy inference system as a generic function approximator to deal with the "curse of dimensionality" issue which is a standard practice in the design of RL based controllers for continuous state-action domains. Of course, any other function approximator such as neural network [8] or support vector machines [9] could be used. Our proposed method, after suitable problem specific modifications, can be used in other applications as well, e.g., for decoupled control of bearingless Induction motors [10] and for countering unbalanced vibrations of bearingless rotor [11].

To test our proposed approach, we employ it on two standard benchmark control problems: a) an Inverted pendulum, and b) Rotational/translational proof-mass-actuator system. We compare performance of the proposed fuzzy lyapunov reinforcement learning control against (i) baseline fuzzy Q learning control (conventional), and some recent lyapunov theory based RL controllers: (ii) lyapunov actor critic RL control and (iii) lyapunov Markov game based control. Rest of the paper is organized as: Section 2 has brief but relevant details on fuzzy Q learning, lyapunov theory based actor critic RL and Markov game based hybrid control. Section 3 gives a detailed presentation of our proposed lyapunov fuzzy RL approach. Section 4 gives details of the systems used to test our proposed approach, i.e., Inverted Pendulum and RTAC with parameters thereof. Section 5 gives simulation results and comparison against fuzzy Q learning and other lyapunov theory based RL controllers and Section 6 concludes the paper.

2. Reinforcement learning

Reinforcement learning is a part of machine learning [4] and has roots in the human learning. In reinforcement learning, an agent searches the space of all possible policies and receives feedback on the results of the selection made. This information must infer a"good" policy or ideally an optimal policy [12]. We can also call RL as action based learning and is highly goal oriented [13]. The terms agent, environment and action used in RL is same as controller, system and control signal in the control engineering literature.

In RL, objective of the agent is to maximize or minimize total accumulated reward or cost incurred in the process of taking actions resulting in a sequence of states or the Markov chain. RL could be model based as in Actor-Critic configuration or model free as in Q-learning and SARSA. Whether model based or model free, all RL techniques belong to what may be called Temporal Difference (TD) methods. Our approach is motivated by the Q learning algorithm which is a model free, off policy incremental learning approach with proven convergence. Basically, RL is a technique to optimize decisions in a sequential decision making problem. For further details on the RL paradigm, we refer the reader to [8].

2.1. Q learning

Q-learning [14] is an off policy learning algorithm and it aims at estimating optimal Q-value Q^*. Q-value is defined as the accumulated reward obtained on taking action '*a*' in state '*s*'. Q-value is also known as state action value and is represented by *Q(s,a)*. In estimating Q^*, the agent learns optimal policy via interaction with the system where policy signifies a mapping from state to action.

$$Q^*(s, a) = E\left[c(s, a) + \lambda \min_{a \in A} Q^*(s', a) \right] \tag{1}$$

where E is the expectation operator, $s' \cong P_{ST}(s, a)$ is successor state and P_{ST} is the state transition probability, $\lambda \in (0, 1]$ is discount factor and signifies relationship between current and subsequent cost, *c(s,a)* is the cost incurred on taking action '*a*' in state '*s*'.

In Q learning the expectation operator is replaced by a single sample, making Q learning a stochastic iterative algorithm. More specifically, the expectation which is based on probability of transition is replaced by a "bootstrapping" of successive samples obtained online while the agent interacts with the system or environment. Q learning update is given by:

$$Q(s^n, a^n) \leftarrow Q(s^n, a^n) + \xi [c(s^n, a^n) + \lambda \min_{a \in A(s^{n+1})} Q(s^{n+1}, a) - Q(s^n, a^n)] \tag{2}$$

where s^n and s^{n+1} are the state at n^{th} and $(n+1)^{th}$ iteration, a^n is the action taken at iteration n, $\xi \in (0, 1]$ is learning rate. This update converges to optimal Q values. Proper decrement in ξ and infinitely large number of visits to each state-action pair is required to get Q^* from Q-values.

2.2. Fuzzy Q learning

Q learning requires one to store all visited state action pairs in the form of a lookup table. This poses a big problem in terms of memory size required to store all Q values for a large state-action space. Another problem arises in the form of computational complexity as the dimensionality of the state space increases. The lookup table based approach becomes infeasible when the state space is continuous. To counter this problem, generalization techniques [8] like neural network and fuzzy inference system (FIS) have been employed wherein the approximation architecture, i.e., neural network or FIS replaces the lookup table. Our approach is motivated by a Q learning implementation through FIS termed fuzzy Q learning. Fuzzy Q learning (FQL) entails rules of the form:

$$\begin{aligned} R_n: \quad & \text{If } s_1^k \text{ is } L_1^n \text{ and } \ldots\ldots \text{ and } s_m^k \text{ is } L_m^n \quad \text{then } a = a_1 \text{ with } q(n, 1) \\ & \text{or} \quad a = a_2 \text{ with } q(n, 2) \\ & \ldots\ldots\ldots \\ & \text{or} \quad a = a_p \text{ with } q(n, p) \end{aligned} \tag{3}$$

Here linguistic term for input variable s_v^k is L_v^n in the rule R_n having membership function $\mu_{L_v^n}$. $s^k = \left\{ s_1^k, s_2^k, \ldots\ldots, s_m^k \right\}$, system state at instant k forms the input vector to FIS. Truth-value of each rule $\mu(s^k)$: $[\mu_1(s^k)\ \mu_2(s^k) \ldots\ldots \mu_N(s^k)]$ for N rules is calculated based

Please cite this article as: Kumar A, Sharma R. Fuzzy Lyapunov Reinforcement Learning for Non Linear Systems. ISA Transactions (2017), http://dx.doi.org/10.1016/j.isatra.2017.01.026