Neurocomputing 000 (2017) 1–12

Contents lists available at ScienceDirect

Neurocomputing

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

Linguistic Lyapunov reinforcement learning control for robotic manipulators

Abhishek Kumar, Rajneesh Sharma*

Instrumentation and Control, Netaji Subhas Institute of Technology, Sector-3, Dwarka, Delhi 110085, India

ARTICLE INFO

Article history:
Received 15 January 2017
Revised 13 May 2017
Accepted 26 June 2017
Available online xxx

Communicated by Dr. Chenguang Yang

Keywords:
Reinforcement learning
Fuzzy Q learning
Linguistic Lyapunov RL
Two link robotic manipulator
SCARA

ABSTRACT

We propose a Lyapunov theory based linguistic reinforcement learning (RL) framework for stable tracking control of robotic manipulators. In particular, we employ Lyapunov theory to constrain fuzzy rule consequents for ensuring stability of the designed controller. Proposed fuzzy RL controller employs Lyapunov theory dictated rules for discovering an optimal yet stable control strategy for robotic manipulators. Furthermore, our proposed linguistic RL controller handles payload variations and external disturbances quite effectively. We validate linguistic Lyapunov RL controller on two benchmark control problems: (i) a standard two-link robotic arm manipulator, and (ii) a two link selective compliance assembly robotic arm (SCARA). Simulation results and comparison against (a) baseline fuzzy Q learning (FQL) controller, and (b) a recently proposed Lyapunov theory based Markov game controller showcases our controller's superior tracking performance and lower computational complexity. Furthermore, our controller exhibits high stability with disturbances and payload variations.

1. Introduction

Controlling robot manipulators is an extremely challenging and complex control problem due to their highly nonlinear and coupled dynamics. Flawless modeling of robotic manipulators is an equally daunting task; forcing the designer to make idealizing assumptions with consequent modeling errors. This has prompted designers to go for model free approaches. RL has emerged as an alternative technique to design high performance controllers for robot manipulators [1]. However most of the RL approaches proposed so for do not guarantee any stability on the designed controller [2]. Lyapunov theory [3,4] offers a powerful platform for designing controllers that are guaranteed to be stable and can handle model uncertainties and parameter variations. Recently, in [3] authors have introduced stability in the RL paradigm by hybridizing Lyapunov theory into the RL action generation mechanism. The approach is excellent but has one drawback of higher computational complexity as one is required to solve one linear program (LP) per iteration to arrive at an optimal solution. Other approaches [5,6] that have tried to infuse stability in the RL domain via Lyapunov theory advocate an actor-critic (AC) formulation. In an AC formulation we, typically have two function approximators; one critic which approximates value function and an actor to approximate optimal policy. A major hurdle here is the simultaneous training of two networks and difficulty in convergence to the optimal solution. In contrast, we seek to guarantee stability of our controller by what may be called a "linguistic Lyapunov control".

In our approach, we use linguistic RL (wherein rule consequents are linguistic rather than crisp). These rules are Lyapunov constrained so as to impart stability to the designed controller. In our previous works, we have attempted at designing stable RL controllers by: (i) generating a hybrid controller by mixing Lyapunov theory based action with RL action [3] and (ii) curtailing the action set of the RL controller so as to satisfy Lyapunov conditions [4]. In this work, we take a conceptually different approach as: (a) we formulate "learning with words" or a linguistic RL approach where we frame linguistic fuzzy RL rules for learning optimal solution, and (b) we impose Lyapunov constraints on these linguistic RL rules or design a Lyapunov linguistic RL controller. This linguistic Lyapunov framework is then carefully fine-tuned for application on robotic manipulators.

Next, we look at some recently proposed soft computing approaches that have used Lyapunov theory for lending stability to the controller. In [7], authors have proposed neural networks based consensus control for multiple robotic manipulators systems and used Lyapunov theory to tune the NN weights for stability. In [8], Lyapunov theory has been used to design an adaptive neural controller for redundant robot manipulators constrained by mobile obstacles. Li et al. [9] propose an RL approach to coordinated manip-

* Corresponding author.
E-mail addresses: rajneesh496@gmail.com, rajneesh496@rediffmail.com (R. Sharma).

http://dx.doi.org/10.1016/j.neucom.2017.06.064
0925-2312/

ulation of multi robots to cope with unknown dynamics of robots and manipulated object. These approaches use NNs (at times two NNs in an actor critic RL formulation) which poses problems in terms of slow convergence and/ or divergence and high computational burden when the system complexity increases. In another approach [10], Lyapunov theory has been used to develop a concurrent learning implementation of model based RL for approximate optimal regulation. The approach relies on availability of a sufficiently accurate system model which is a restrictive pre condition. Our approach, in contrast, proposes a model free linguistic Lyapunov RL controller with no divergence issues and offers scalability to high dimensional non-linear systems.

In implementing RL control on continuous state action space problems such as the manipulator control, use of function approximators becomes a necessity to counter the "curse of dimensionality". We have used fuzzy inference system (FIS) as function approximator as in [3], but our approach has two distinct advantages vis a vis these approaches: (i) we have used only one function approximator unlike the AC based RL approaches [5,6] making it simple and convergence to the optimal solution is fast. In an AC RL setup simultaneous training of two function approximator is required, and (ii) it has lower computational complexity than Markov game based approach [1,3] where a linear program must be solved at each iteration to generate optimal solution. This makes our approach suitable for high dimensional problems as scalability is very easy.

In a recent survey [11], authors have detailed several neural networks (NN) and fuzzy logic based controllers. Some recent NN based RL controllers have also been described that employ two NNs in an actor critic (AC) configuration. However, there are two major concerns in an AC formulation: (i) Simultaneous training and ensuring convergence of actor and critic NNs is difficult and at times tricky, and (ii) severe computational burden due to large number of tunable parameters. In contrast, our proposed approach has fewer tunable parameters and employs a single function approximator leading to faster convergence.

An adaptive fuzzy controller for an exoskeleton robot is presented in [12] wherein fuzzy logic has been employed to approximate nonlinear dynamics of the manipulator. However, our approach is different from this approach [12] in two aspects: (i) in [12], fuzzy logic has been used to approximate the manipulator parameters whereas we use FIS as a function approximator for estimating the linguistic Q function, and (ii) Our approach is an RL approach wherein quadratic error signal is used to modify the q values for optimal trajectory tracking. Moreover our implementation is simpler as we use only one FIS with linguistic consequents.

We test the efficacy of our proposed approach on two benchmark robotic problems: (i) tracking control of a two link robotic manipulator, and (ii) control of a SCARA. The controller is subjected to disturbances and parameter variations as well. Performance comparison has been done against baseline fuzzy Q learning and Lyapunov theory based Markov game controller. Rest of the paper is organized as: theoretical background of reinforcement learning paradigm and fuzzy Q learning is given in Section 2. Section 3 details proposed approach and application of proposed scheme on two link manipulator and SCARA is given in Section 4. In Section 5, we give simulation results and comparison against fuzzy Q learning and Lyapunov theory based Markov game controller which is followed by conclusion and future scope in Section 6.

2. Reinforcement learning and fuzzy Q learning

Reinforcement learning is learning by interacting with the system one intends to control. Typical goal is to maximize an accumulated reward or minimize cost (obtained at each step/interaction). Basically, RL is a family of stochastic iterative algorithms for finding an optimal solution to a sequential decision making problem [13]. RL algorithm can be ; (i) model based wherein a model of the underlying system is learned and is used to generate an optimal solution, and (ii) model free, e.g., Q learning in which model of the plant/system is learned impromptu [13]. Model free RL techniques have gained importance as no model of the system is required and are simple to apply. Next, we describe Q learning (a model free RL approach).

2.1. Q learning

Q learning was proposed by Watkins [13] as a model free technique to optimize unknown systems. It models the sequential decision making task (optimization of controller) as a Markov Decision Process (MDP). MDP is a sequential decision making model in which at time instant k an agent (or controller) applies action $u \in U(s)$ in state $s \in S$ making the environment (or system) move to the next state $s' \in S$, emitting a reward (or cost) c. $U(s)$ and S are set of all possible actions in state s and set of all possible states, respectively. This movement from s to s' is in accordance with state transition probability $p(s,u,s')$. Agent's objective is to optimize policy π: $\pi(s) \rightarrow u$; $u \in U(s)$, so that expected accumulated discounted cost is minimized [3]. In Q learning, we define a Q-value as the expected accumulated cost incurred by agent on taking action u in state s and then continuing on an optimal policy. Q-value, $Q(s,u)$ is given by [3]

$$Q(s,u) = c(s,u) + \xi \sum_{s' \in S} p(s,u,s')V(s') \tag{1}$$

where $V(s') = \min_{u \in U(s')} Q^*(s',u)$ and $\xi \in (0, 1]$ is discount factor that gives relationship between present and future cost.

We can generalize (1) as update rule

$$Q(s^k,u^k) \leftarrow Q(s^k,u^k) + \lambda[c(s^k,u^k) + \xi \min_{u \in U(s')} Q(s^{k+1},u^k) - Q(s^k,u^k)] \tag{2}$$

where s^k and s^{k+1} denotes state at kth and $(k+1)$th iteration, u^k gives action taken at iteration k, $\lambda \in (0, 1]$ is learning rate parameter. Infinitely large visit to each state action pair and proper decrement in λ gives optimal Q-values i.e. Q^*.

2.2. Fuzzy Q learning

In Q learning, we need to store Q values for each visited state action pair; Q value storage becomes infeasible for large state spaces. This problem can be solved using generalization techniques, e.g., neural network (NN) and fuzzy logic have been used for generalization [1,3–6]. We use fuzzy inference system (FIS) to implement Q learning, also known as fuzzy Q learning. Rules formulated in fuzzy Q learning (FQL) have the form [3]:

$$\begin{aligned} R_n: \quad & \text{If } s_1^k \text{ is } L_1^n \text{ and } \ldots\ldots \text{ and } s_m^k \text{ is } L_m^n \text{ then } u = u_1 \text{ with } q(n,1) \\ & \text{or } u = u_2 \text{ with } q(n,2) \\ & \ldots\ldots\ldots \\ & \text{or } u = u_p \text{ with } q(n,p) \end{aligned} \tag{3}$$

In rule R_n linguistic term L_v^n for input variable s_v^k has membership function $\eta_{L_v^n}$. s^k denotes system state $\{s_1^k, s_2^k, \ldots\ldots, s_m^k\}$ at instant k, and acts as input to the FIS. Matching of rule premise to the input vector s^k is used to calculate truth-value of each rule $\mu(s^k)$: $[\mu_1(s^k)\,\mu_2(s^k)\ldots\mu_N(s^k)]$ corresponding to N rules.

In nth rule (R_n) the agent selects minimizing action, i.e., the one with minimum $q(n, u_m)$ value from set $U = \{u_1, u_2, \ldots, u_p\}$:

$$u_n^* = \arg \min_{u_n \in U} q(n, u_n) \tag{4}$$

Please cite this article as: A. Kumar, R. Sharma, Linguistic Lyapunov reinforcement learning control for robotic manipulators, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.06.064