Learning Robot Motion through User Intervention and Policy Search

Aleksandar Jevtić, Adrià Colomé, Guillem Alenyà and Carme Torras
Institut de Robòtica i Informàtica industrial, CSIC-UPC, Barcelona, Spain

ABSTRACT
Reinforcement Learning (RL) through policy search is well suited for robot learning of manipulation tasks but requires a high number of policy updates to produce a satisfactory robot behavior. We propose a method that exploits user intervention as a strategy for policy search updates, which significantly reduces the number of necessary updates and allows users to tailor robot behavior according to their needs. We developed an interface for gesture recognition and hand motion following that allows a user to select and modify a desired manipulation task segment and speed up the learning process through a series of interventions. We implemented the method on a single-arm and dual-arm robot and applied it to a manipulation task on the laboratory tabletop. We compared our method with the conventional policy search strategy in simulation; the results show that the number of necessary policy search updates was significantly reduced by user intervention.

PROPOSED INTERACTIVE LEARNING FRAMEWORK

Robot executes initial trajectory
User modifies initial trajectory
Robot stores new trajectory and reward value
Apply policy search algorithm

RESULTS

ALGORITHMS

Static gesture recognition
Hand motion following

User intervention algorithm and reinforcement learning

Algorithm 1 User intervention
1. set state to "EXECUTE";
2. go to first trajectory point;
3. while not end of trajectory do
4. go to next trajectory point;
5. if hand raised then
6. set state to "FOLLOW";
7. store trajectory cut point;
8. while in "FOLLOW" state do
9. follow user’s hands;
10. if hand raised then
11. set state to "EXECUTE";
12. end if
13. end while
14. find closest trajectory point;
15. end if
16. end while

Algorithm 2 Reinforcement learning of new trajectories
1. Input: User-generated trajectories \( p_q = (x_1, \ldots, x_n) \) for user; cut and connect point indexes \( t_1, \ldots, t_n \) for RL; rewards of user-generated trajectories \( R_q \) for \( q \);
2. Find common unmodified part of the initial trajectory for all robots: \( f_{init} = \min_i \). Start RL for each \( R_q \);
3. Extract segment of trajectory that will be modified by RL, for all trajectories: \( f_{init} = \min_i \). Start RL for each \( R_q \);
4. Use alignment to \( T_2 \) for \( \psi_q \), using trajectory, from the first rollout as reference, to have same number of points in all trajectories;
5. Apply REPS to compute relative importance weight for each rollout: \( w_{n_{max}} = \text{REPS}(\theta) \);
6. Obtain new trajectory by a weighted sum of the time-aligned, resampled trajectories.

REFERENCES