Universität Hamburg

Interactive Reinforcement Learning for HRI

Zhen Deng

13. June 2017

Contents

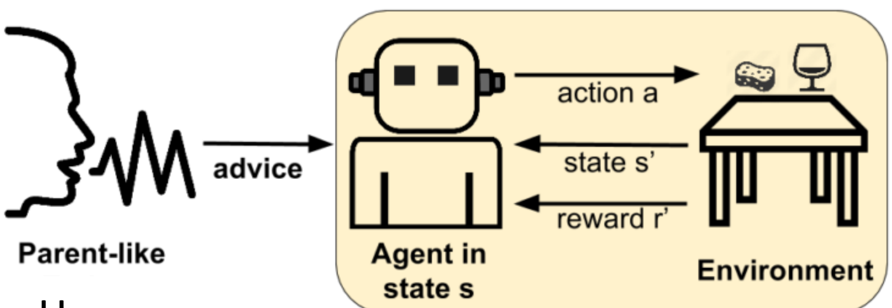
Contents

1. Backgorund: HRI, IRL

How to consider human control in robot learning (RL) algorithm?

Human-robot interaction mechanism

- Cognitive HRI: Human feedback
  - Scalar number: (good, bad ...)
  - Probability (A is better than B)
  - gesture, Speech, Keyboard, ...
- Physical HRI: Human continuous action



Human

Cognitive HRI: Bi-directional multi-model communication and understanding

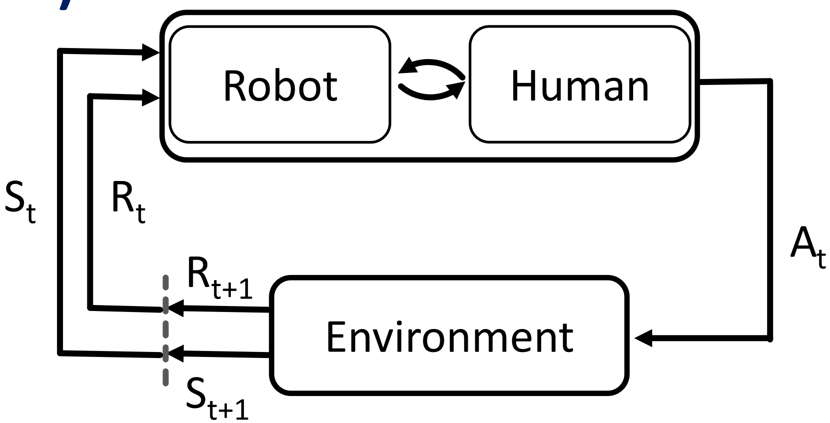
physical HRI: Exchange of the contact force, coordinated operation

1. Backgorund: HRI, IRL

Interactive reinforcement learning (IRL)

The human provides a feedback or advice to guide the exploration of the robot.

- … more sample-efficiency.
- … a coupling of the human and robot



Four techniques:*[Peter Stone, 2012]*

1. Reward shaping: $R^{'}(s,a) = R(s,a) + \beta * H(s,a)$
2. Q-value augmentation: $Q^{'}(s,a) = Q(s,a) + \beta * H(s,a)$
3. Action biasing: $Q^{'}(s,a) = Q(s,a) + \beta * H(s,a)$,Only during action selection
4. Control sharing: $P(a = \mathrm{argmax}[\hat{H}(s,a)]) = \min(\beta,1)$, Otherwise use the RL agent's action selection mechanism

Where, $H(s,a)$ is the shaping function.

Contents

2. Hierarchical Robot Learning

Hierarchical Robot Learning for Physical Collaboration between Humans and Robots

- **Goal**: the robot learns to cooperate with the human for HRC task

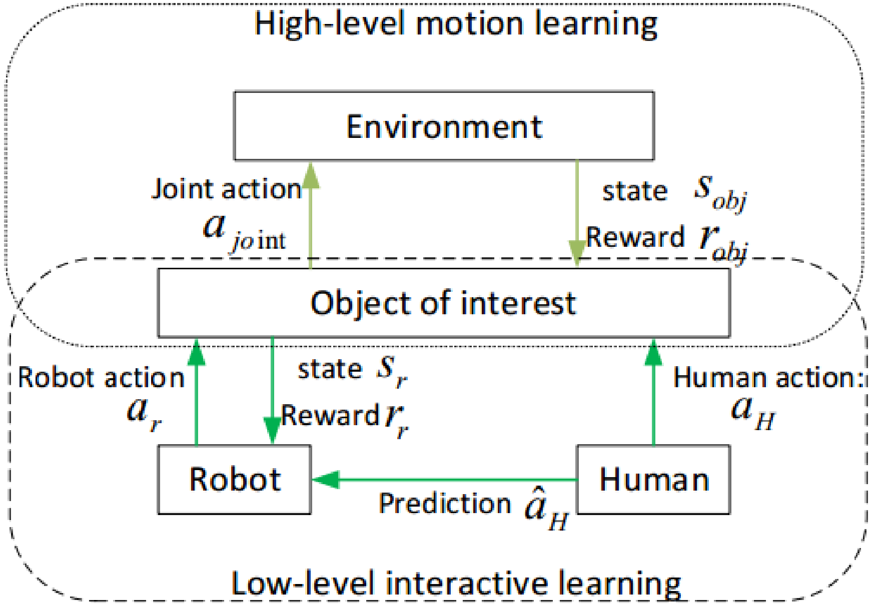
- **Two requirements for the desired robot behavior:**
  1. The human-robot joint action is able to accomplish the task toward the common goal.
  2. The robot should adapt to the human operation capability.

- **Hierarchical robot learning framework**

.... Inspired from Hierarchical reinforcement learning (HRI). *[Ghavamzadeh, 2017]*

High-level → **Motion path of object**

Low-level → **The coordinate behavior**



2. Hierarchical Robot Learning

A. High-level Motion Learning with Policy Search

1. Motion policy is represented by Dynamic Motor Primitive (DMP)
2. Optimize policy parameter by using policy search algorithm

$$\ddot{y} = \alpha_y(\beta_y(y_g - y) - \dot{y}) + \tau^{-1} f(x)$$

$$f(x) = \frac{\sum_i^N \omega_i x \psi_i(x)}{\sum_{i=1}^N \psi_i(x)}$$

PoWER, Expectation-Maximization (EM)-based policy search algorithm, preseted by *[Jan peter, 2009]*

$$\omega' = \omega + \mathbb{E}\{\sum_{t=1}^{T} Q_\pi(s,a,t)W(s,t)\}^{-1}\mathbb{E}\{\sum_{t=1}^{T} Q_\pi(s,a,t)W(s,t)\varepsilon_t\}$$

1. High-level Motion Learning:
Respect (for each episode)
Performing rollout using motion policy $\pi(\omega)$
Collect all the states, actions and rewards $\{s, a, r\}$
Update motion policy parameter ω using Eq. $\boxed{4}$

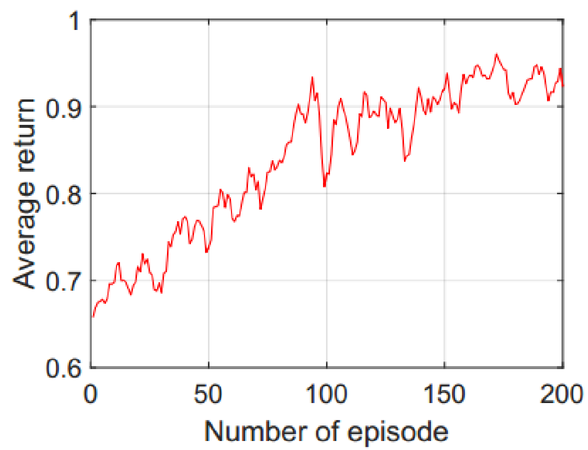


Fig. 3: the return of each episode in high-level motion learning.

2. Hierarchical Robot Learning

B. Low-level Interactive Learning for Active Contributions

1. Policy Evaluation with Function Approximation
2. Human action prediction
3. Prediction-based Action Selection

GP: $p(Q_*|x_*,\vartheta,\mathbb{D}) \sim \mathbf{N}(q_*|\mu(x_*),\Sigma(x_*))$

Update: $$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[\mathbb{E}_{s\sim s_{t+1}}[r(s_{t+1})] + \gamma \underset{a_{t+1}}{\operatorname{argmin}} \mathbb{E}_{s\sim s_{t+1}}[Q(s_{t+1},a_{t+1})] - Q(s_t,a_t)]$$

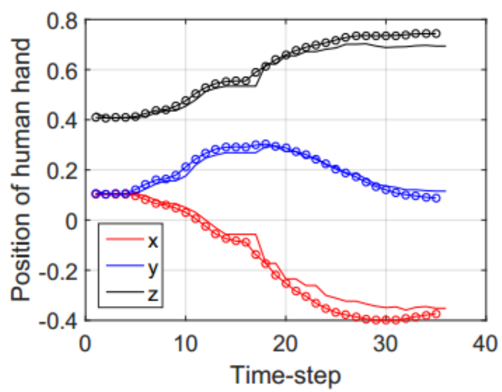


Fig. 4: the comparison of the measured position of human hand by sensor (The solid line) and the predicted position from EKF (the dotted line).

Extended Kalman Filter

$$X_K = \widehat{X_K} + K_k(Z_K - H\cdot\widehat{X_K})$$

$$\begin{cases} X_{k+1} = A\cdot X_K + w \quad \textit{with} \quad A = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \\ Z_{k+1} = H\cdot X_K + \upsilon \quad \textit{with} \quad H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{cases}$$

$\Downarrow$

A constraint optimization problem

$$a_r^* = \underset{a_r}{\operatorname{argmin}}\, Q(s,a_r)$$

$$\text{s.t.} \quad a_r \in [\alpha a_h - \lambda\sigma_r, \quad \alpha a_h + \lambda\sigma_r]$$

2. Hierarchical Robot Learning

- **Contributions:**

1. Motion path is learned as the common goal
2. Don't assume the human as expert
3. Learn to coordinate with the human (human move with a low speed)
4. State prediction improve the sample-efficiency of RL.

- **Shortages:**

1. The performance of POWER largely depends on the value of learning parameter (…be careful!)
2. the human is required to move slowly.
3. The computal complexty cause the delay of control.
4. Still need human demonstration to obtain a prior policy

Algorithm 1 Hireachical Robot Learning with Human Physical Interaction

Inputs: Initial motion policy parameter ω_0, Iteration number H_1, H_2
1. High-level Motion Learning:
  Respect (for each episode)
    Performing rollout using motion policy $\pi(\omega)$
    Collect all the states, actions and rewards $\{s, a, r\}$
    Update motion policy parameter ω using Eq. 4
2. Low-level Interactive Learning:
  Repeat (for each episode)
    Initialize state s and Q-value GP
    Repeat (for each step of episode)
      Predict human action using Eq. 11
      Find optimal robot action a_r using Eq.6
      Collect observation s' and reward r_r
      Update the Q-value $Q(s, a_r)$ using Eq.12
      Set $s \leftarrow s'$, $a_r \leftarrow a'_r$
  Update Q-value GP by fitting to new Dataset $\mathbb{D}$

(a) (b)

Fig. 1: Human and robot collaboratively assemble a toy which includes a support and a tray. The assembled task includes two sub-tasks. Figure a) shows the first sub-task where a robot transports the support to a predefined position. Figure b) shows the second sub-task where a human and a robot cooperate to transport a tray.

Contents

3. Learning from human physical control

Interactive reinforcement learning with human physical control

- **Goal**: learning skills autonomously from the physical interaction with human

Human cognitive action $\xrightarrow{\text{Feeback}}$ The sequential decision-making problem with limited actions, like the game of GO

Human physical action $\xrightarrow{\text{force}}$ The continuous decision-making problem, like robot motor control

- **Two requirements for physical HRI:**
  1. Sample-efficient learning.
  2. A ability to adapt to human behavior and learn from human physical control.

- **Basic idea:**
  1. Using a virtual impedance model to describe the interactive behavior between the human and the robot
  2. Using Model Prediction Control (MPC) to assist the RL algorithm
  3. Utilize human physical action in the RL algorithm

3. Learning from human physical control

A. Virtual Impedance Mode

the transition of states is built by using the virtual Cartesian impedance model:

$$M\ddot{x}(t) + D\dot{x}(t) = a_r(t) + a_h(t)$$

The states pace form of a discrete time system:

$$z(t+\Delta t) = f_z z(t) + f_r a_r(t) + f_h a_h(t)$$

$$z(t) = \begin{bmatrix} x^T \\ \dot{x}^T \end{bmatrix}, f_z = \begin{bmatrix} I_{m\times m} & \Delta t I_{m\times m} \\ I_{m\times m} & -\Delta t M^{-1} C \end{bmatrix}, f_r = f_h = \begin{bmatrix} 0_{m\times m} \\ \Delta t M^{-1} \end{bmatrix}$$

B. Model prediction control

use iteration LQR to implement MPC:

$$\underset{a_t}{\operatorname{argmin}} \sum_{i=t}^{T-1} l(x_t, a_{r,t}, a_{h,t}) + l(x_T)$$

$$s.t. \quad x_{t+1} = f(x_t, a_{r,t}, a_{h,t}), \forall t \in 1, \cdots, T$$

$$Q_{x,t} = l_{x,t} + f_{x,t}^T V_{x,t+1}$$

$$Q_{a,t} = l_{a,t} + f_{a,t}^T V_{x,t+1}$$

$$Q_{xx,t} = l_{xx,t} + f_{x,t}^T V_{xx,t+1} f_{x,t}$$

$$Q_{aa,t} = l_{aa,t} + f_{a,t}^T V_{xx,t+1} f_{a,t}$$

$$Q_{ax,t} = l_{ax,t} + f_{a,t}^T V_{xx,t+1} f_{x,t}$$

$$\boxed{\begin{gathered} g_t = u_t + k_t + K_t(\hat{x}_t - x_t) \\ k_t = -Q_{aa,t}^{-1} Q_a \text{ and } K = -Q_{aa,t}^{-1} Q_{ax,t} \end{gathered}}$$

Near-optimal policy

Iteration LQR recursively computes the first order Tyler expansion of the dynamic model and the second order Tyler expansion of the action-value function

3. Learning from human physical control

B. Interactive learning with human physical action

1. Policy evaluation with function approximation

$$p(Q_*|x_*, \vartheta, \mathbb{D}) \sim \mathbf{N}(q_*|\mu(x_*), \Sigma(x_*))$$

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[r(t+1)] + \gamma \underset{a_{t+1}}{\operatorname{argmin}}(Q(s_{t+1},a_{t+1})) - Q(s_t,a_t)]$$

2. Reward shaping with human physical action

$$r(s,a) = r_t(s,a) - \beta\, r_h(s,a)$$

Task-relevant reward: $r_t(x_t,a_{r,t},a_{h,t}) = (x-x_d)^T Q_1 (x-x_d) + \dot{x}^T Q_2 \dot{x}$

Human-relevant reward: $r_h(s,a) = \eta \dfrac{f_h(t) g(t)}{||g(t)||}$

3. Prediction-based action selection

$$a_r^* = \underset{a_r}{\operatorname{argmin}}\, Q(s,a_r,a_h)$$

$$\text{s.t.} \quad a_r \in [g(t) - a_h - \xi_r, \quad \underline{g(t) - a_h} + \xi_r]$$

Algorithm 1 :HITL robot learning.

```
Requires: M and C: two impedance parameters.Q1, Q2, R1 and R2: the weights in the cost function of MPC. α and λ: the learning rate and discount factor of Q-learning, initialize Q(s,a) arbitrarily.
Perform MPC to find a near-optimal policy until convergence in a simulation environment.
for iteration k = 1 : K do
    Reset the robot state s = s0.
    for time step t = 1 : T do
        Receive current state sT.
        Find a near-optimal impedance action g(st) in Eq.7.
        Measure the force ft exerted by the human.
        Compute optimal robot action a*r using Eq.13.
        Compute the reference velocity ẋr using Eq. 2.
        Compute the reference joint velocity q̇r using Eq.4.
        Receive next state st+1 and reward rt.
    end for
    Update Q(s,a) using Eq.9 and optimize GPs.
end for
```

Combining the model-based planning and human teaching to Guide the exploration of the RL

3. Learning from human physical control

C. Experiments

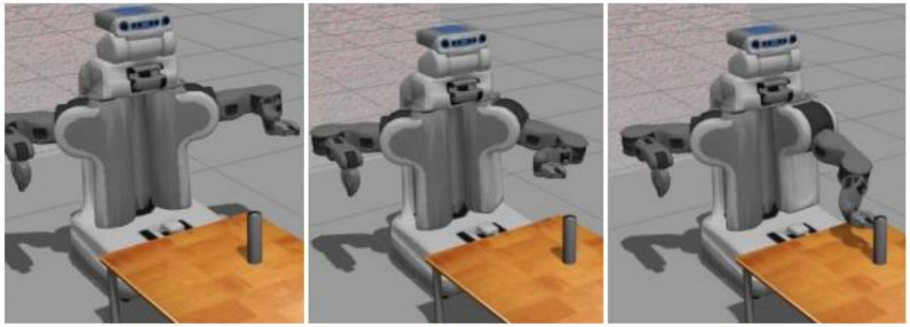
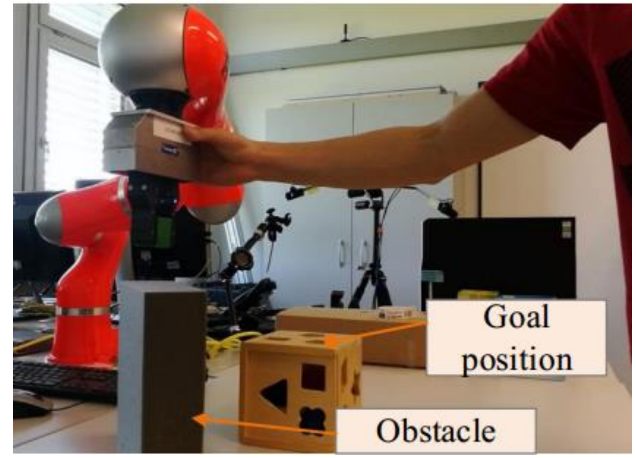


Figure 1: Three consecutive snapshots shows a successful reaching task in Gazebo simulator (a-c).



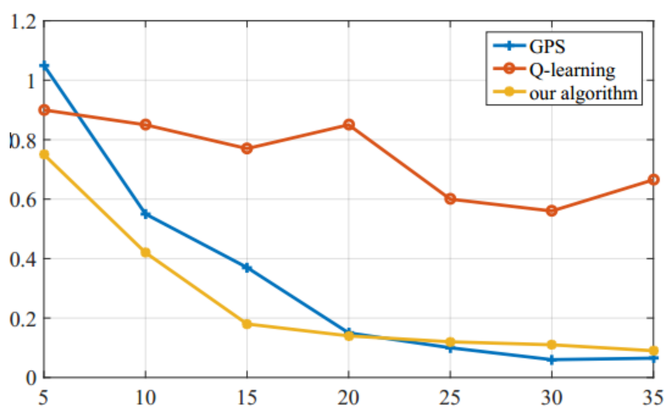


Figure 2: the final normalized distance from the end-effector to the goal posi-
tion of each iteration.

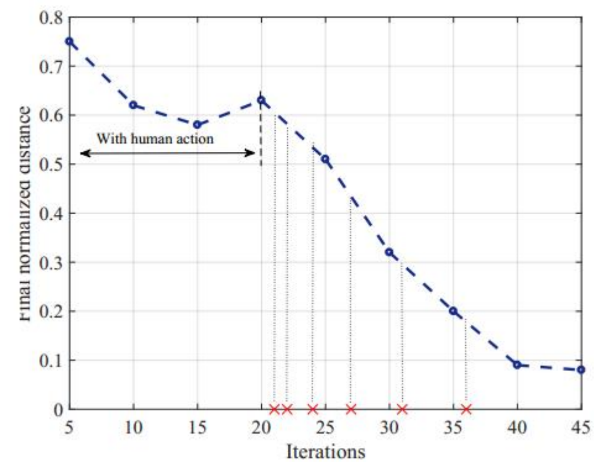


Figure 7: the final normalized distance from the end-effector to the goal posi-
tion of each iteration.

Guide policy search method [*Sergey Levine, 2013*]

Contents

4. Human-in-the-loop robot learning

Human-in-the-loop robot learning with Multi-modal interaction

- **Goal**: learning skills from Multi-modal interaction in Multi-task problem
- **Problem Setting:**
  1. A complex task with a set of sub-task.
  2. Robot learn to obtain a sub-task sequence to decompose the complex task.
  3. Robot learn the underlying policy to instance each select sub-task.
  4. Human may assist robot learning through human speech or human physical control.

The complex task is decomposed under the option framework [Sutton, 1999]

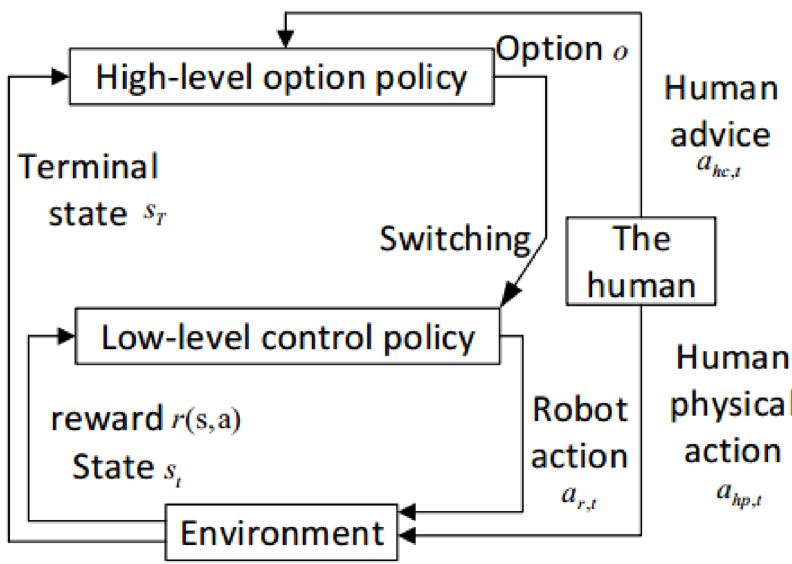
option $o = <\pi_o, I_o, \beta_o>$

High-level → to find a optimal option sequence

$:<o_1, o_2, \ldots, o_n>$

Low-level → optimize the underlying control policy of the select option

$\pi_{o1}, \pi_{o2}, \ldots, \pi_{on}$



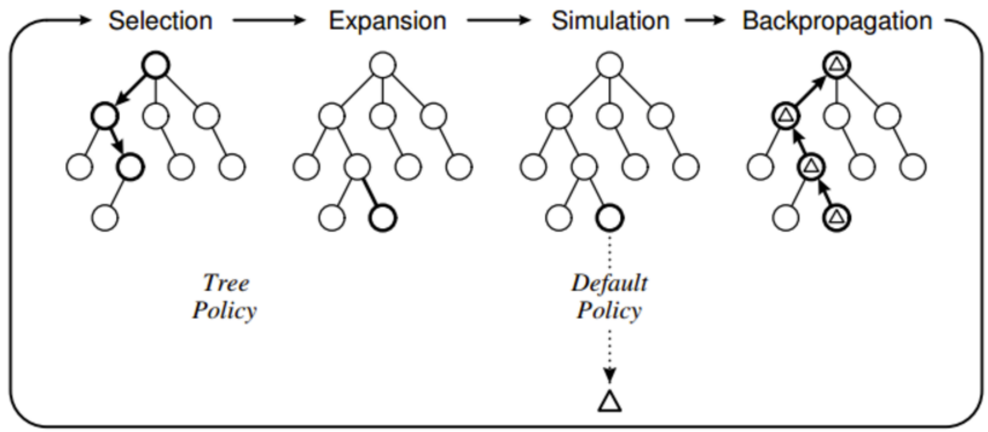
4. Human-in-the-loop robot learning

A. High-level Option learning with MTCS

.. Is a sequential MDP with limited actions

Possible nodes:

1. Pick a object i
2. Place a object i
3. Clear table
4. Go home



a. Without human advice

using the Upper Confidence Bound (UCB) policy

$$UCB = \frac{V_i}{N_i} + c\sqrt{\frac{ln \quad N}{N_i}}$$

b. With human advice

function: SelectAction(n, c)
if human advice h via SAR system **then**
  if *rand* ¡= feedbackProbability **then**
    $o \leftarrow advice$
    $n' \leftarrow None(s, o)$
    if n' in $childNode(n)$ **then Return** o
    else $addChildNode(n, n')$
else
    $o \leftarrow \text{argmin}_o\{\frac{v_i}{n_i} + c\sqrt{\frac{ln \quad n}{n_i}}\}$
Return o

B. Low-level control policy learning

- Is same to the above method introduced in section 3.

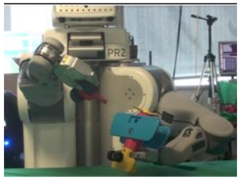
Contents

5. Conclusion

1. to improve the sample-efficient of RL algorithm.

- **Model-based planning method.**

  Continuous setting ← Dynamic programming, like iteration LQR
  Discrete setting ← Monte Carlo Tree Search (MTCS)



GPS [*peter abbeel, 2015*]

Alphago

- **Experience replay**

  reusing its past experiences from a experience memory to speed up the convergence.

Deep Q network (DQN)

- **Human teaching**

  Human provides feedback or advice to guide the exploration of the learning agent

- **Imitation learning / transfer learning**

  1. Represent and generalize the knowledge from human demonstration or other learning method
  2. Then directly give the samples or a prior policy to agent

5. Conclusion

2. to utilize human cognitive or physical action in RL algorithm.

- **By using a shaping function.**

  Human's control objective → a shaping function $$Q^{'}(s,a) = Q(s,a) + \beta * H(s,a)$$

- **But how to learn from the continuous human control is still unknown?**

  1. By taking it as a noisy

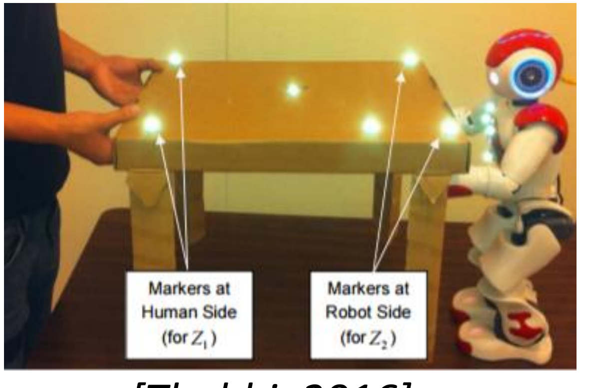
  2. Inverse reinforcement learning

     Record human optimal operation to estimate the value function for RL

  3. By predicting the human state.

     Human future state → human-specific goal

[Ghadirzadeh, 2016]



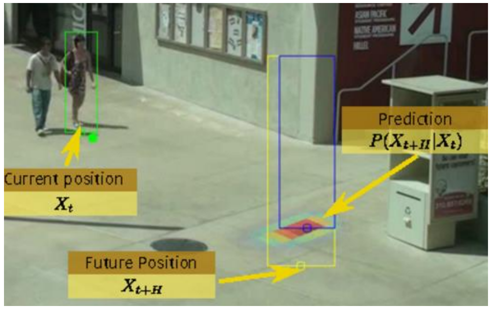
[Thobbi, 2016]

5. Future work

- **Prediction of human**

$$\underline{nextState} \; \underline{=} \; f(currState, \underline{humanAction})$$

Predict & recognize: Human dynamic , human intension , human pose, human action, human preference ….



- **Deep reinforcement learning (DRL)**

Deep → a better approximation ability, like can process complex sensory input
RL → a trail-and-error process, can choose optimal action

- **Human-in-the-loop robot learning**

1. How to achieve the co-adapt and co-learning between humans and robots
2. A risk-aware RL to keep the environment and human safety
3. Sample-efficient Learning from human physical control in continuous setting


Thank you!