# Reinforcement learning for robotic assembly

Gabriel Vallat

November 2, 2022

**Abstract**

**Algorithm 1** Training loop with replay buffer

**Require:** a set of autonomous agent $A$, a *simulator*, a reward function $R(s, a)$
  Initialise an empty replay buffer $B$
  Initialise A
  Initialise the simulator
  $ep \leftarrow 0$
  **while** $ep < ep_{max}$ **do**
    reset *simulator*
    $B \leftarrow$ run episode($A$,$B$,*simulator*)
    $ep \leftarrow ep + 1$
  **end while**

**Algorithm 2** Episode with egoist rewards

**Require:** an agent set $A$ ,a replay buffer $B$, a simulator
  $t \leftarrow 0$
  **while** $t < t_{max}$ **do**
    **for all** $agents \in A$ **do**
      **if** $t > 0$ **then**
        $s' \leftarrow$ simulator state
        Store $s'$ in $T_{agent}$
        Delete oldest $T_{agent}$ if $B_{agent}$ is full
        Store $T_{agent}$ in $B_{agent} \subset B$
      **end if**
      $s \leftarrow$ simulator state
      $a \leftarrow$ agent choose action given and its policy $\pi(s)$
      $r \leftarrow R(s, a)$
      Store $s, a, r$ in $T_{agent}$
      update simulator with $a$
      update $\pi$ with $B_{agent}$
    **end for**
    $ep \leftarrow ep + 1$
  **end while**
  **return** B

**Algorithm 3** Choose action Q-Table

**Require:** a state $s$ and an agent having an internal Q-table $Q$ and a set of action $\mathbb{A}$
**Ensure:** $Q(s) = \vec{v}$, where $v_a \in \vec{v}$ is the expected reward of each action $a \in \mathbb{A}$
  $p \leftarrow softmax(Q(s))$
  $a \leftarrow$ sample $\mathbb{A}$ with probability distribution $p$

**Algorithm 4** Update Q-Table

**Require:** a buffer $B_{agent}$ of transitions $T_t = (s_t, a_t, r_t, s'_t)$ and an agent having a discount factor $\gamma$, a set of action $\mathbb{A}$ and an internal Q-table $Q_\theta$ with parameters $\theta$

**Ensure:** $Q(s) = \vec{v}$, where $v_a \in \vec{v}$ is the expected reward of each action $a \in \mathbb{A}$

$i \leftarrow 0$

**while** $i < n_{batches}$ **do**

Sample a minibatch $b$ from $B_{agent}$

$y \leftarrow r(b) + \gamma max_a(Q_\theta(s'(b)))$

$\theta' \leftarrow \sum_{t \in b}(Q_\theta(s(t) - y(t))\nabla_\theta Q_\theta(s(t))$ ▷ MSE loss with the gradient detached from y

$\theta \leftarrow \theta - \alpha\theta'$

$i \leftarrow i + 1$

**end while**

## References