Using Deep Reinforcement Learning to Simulate Bionic Arm Control

Denis Griaznov

December 2020

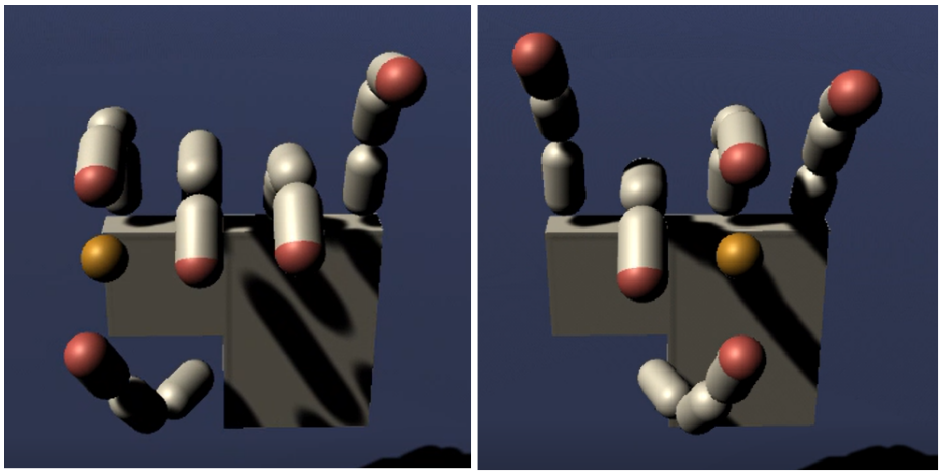
1 Overview

This is a small research aimed at testing the reinforcement learning capabilities for autonomous control of the bionic arm. The main goal in the future is to achieve full autonomy in the control of the bionic arm, so that it adapts to any task in the environment.

Before that, my research was aimed at classifying EMG signals for subsequent control of such a device. But this approach has many disadvantages. First, nothing is automated, machine learning only provides an interface - the relationship between muscle tension and the action of the robot. Secondly, this dependence is often incomprehensible intuitively, especially for people with disabilities. A person can clearly distinguish 3-5 different gestures, but then problems begin. Real living beings, including humans, use an unlimited number of movements depending on the situation and control is carried out continuously.

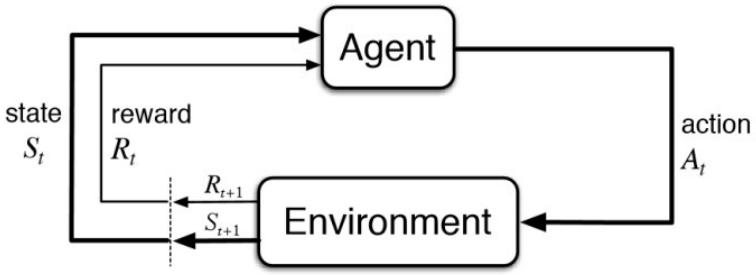
The idea behind the research is to use reinforcement learning algorithms similar to those used in chess and go. In this case, Capturing an object can be thought of as a game, and if successful, the algorithm receives a reward. The position, size and shape of an object can be obtained using sensors such as a binocular camera and lidar. But testing and training the model in real life is quite labor intensive. Therefore research is started with a simulation in a virtual environment that simulates physics. In our case, this is Unity3D. To implement reinforcement learning, the Gym library from OpenAI was chosen. It worked in conjunction with Unity ML Agents via Python.

A model of a hand with 15 degrees of freedom was built. The game consisted in grabbing an item that appeared in one of 3 different positions at random.



2 Reinforcement Learning

The following is a typical reinforcement learning scheme. A hand model acts as an agent. The environment is space with an object. The agent receives a reward after 90 steps. This is where the game ends and a new one begins.



The difficulty in determining the State is that in the future we must somehow obtain all these values using physical sensors.

The Reward can only be determined on the basis of virtual data, but there are difficulties here. It is necessary to correctly interpret intuitive human metrics into mathematical functions.

Action:

A vector of dimension 15 was used as an action. Each value corresponded to the angle of rotation of the finger joint in degrees. Values can range from 0 to 1 for each step. For the lower joint of the thumb, -1 to 1, as it moves in the horizontal plane.

$$\mathbf{A} = \begin{bmatrix} \Theta_{a1} & \Theta_{a2} & \Theta_{a3} & \Theta_{a4} & \Theta_{a5} \end{bmatrix} \in \Re^{15}$$

$$\Theta_{ai} = \begin{bmatrix} \theta^i_{a1} & \theta^i_{a2} & \theta^i_{a3} \end{bmatrix} \in \Re^3$$

where θ^i_{aj} is angle of control rotation of **j** link of **i** finger

State:

The state of the environment is mainly based on the position of the fingers and the position of the target. It is assumed that the position of the fingers can be obtained accurately, the position and size of the target can be obtained using a calibrated binocular camera.

$$S = \begin{bmatrix} \mathbf{r} & \mathbf{s} & d \end{bmatrix} \in \Re^8$$

$$\mathbf{r} = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 \end{bmatrix} \in \Re^5$$

$$r_i = ||r_t - r_{if}||$$

where r_t is vector of target position, r_{if} is vector of end of **i** finger position

$$\mathbf{s} = \begin{bmatrix} s_x & s_y \end{bmatrix} \in \Re^2$$

where s_x and s_x are linear sizes of target

Reward:

Ideally, the reward should be an everywhere differentiable, continuous function that tends to zero if the target is not captured and tends to unity in the opposite case. In fact, it should be a rung that abruptly converts to a unit upon capture. But in reality, the derivative must be different from zero everywhere, so that the agent understands where to move. A function of two terms was chosen - a power-law function for the thumb (since it is less likely to accidentally reach the goal) and an exponent for all the others.

$$\mathbf{R} = \mathbf{R_o} + \mathbf{R_t}$$

$$\mathbf{R} = \lambda \exp(-\prod_{i=0}^{4} r_i) + \frac{1}{(r_5 + \frac{1}{6-\lambda})^2}$$

The λ parameter is responsible for balancing the learning rate of the thumb and upper fingers.

3 Results

3 models were trained with PPO2 policy and different lambda parameter values.

$$\lambda = 1, 2, 3$$

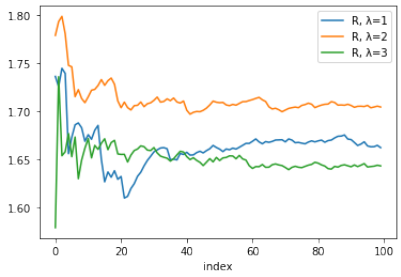
For each model, training was carried out at 100,000 steps.

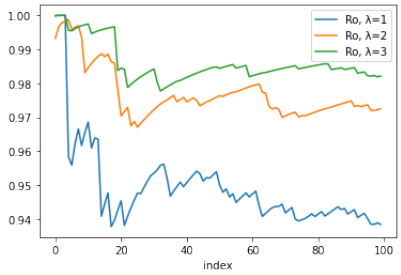
Since the awards were different, it would be incorrect to compare them with each other. Therefore, an independent metric was introduced to determine the accuracy of object capture.

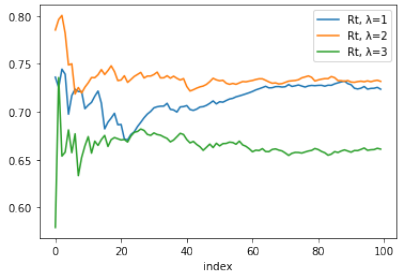
$$\mathbf{R} = \mathbf{R_o} + \mathbf{R_t}$$

$$\mathbf{R} = \exp(-\prod_{i=0}^{4} r_i) + \exp(-r_5)$$

Below are the results for tests on 100 episodes.







It can be seen that the optimal balance is achieved at $\lambda = 2$. The results can be considered satisfactory. Further research can be directed towards developing a sensor system, as well as training a more complex model to capture more objects.