80846 - Report - 6th

GU JUN, 6132230056-4

May 28, 2024

1 AI based controller using reinforcement learning

Problem Statement

In this report, I will using reinforment learning method to design and build a controller for a two-joints robot.

Instead of using Matlab to simulate everything, I will use Isaac-sim to simulate my robot dynamic model and controller.

1.1 Simulink Model

See the Figure 1 below. Besides transport delay blocks, I also used the subsystem block and a clock block to restart q every trail. The length of the trail is 1 second.

1.2 Formulas and Source Codes

This part includes the formulas and source codes for the learning law block

learning law

The feed-forward input is updated by using the control error of the last trial.

$$\tau_{ff2}(t) = \tau_{ff1}(t) + \alpha(\beta(q_d(t) - q_1(t)) + \dot{q}_d(t) - \dot{q}_1(t))$$

The source code is shown below:

```
function tau_ff = updating_law(tau_ff_last, error_last, derror_last)

alpha = 0.5;

beta = 20;
```

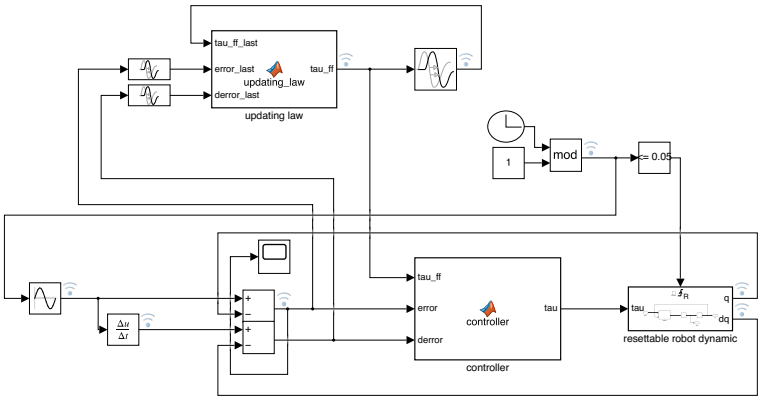

Figure 1: Simulink Model of the iterative learning controller

```
tau_ff = tau_ff_last + alpha * (beta * error_last + derror_last);

end
```

1.3 Simulation Results

As a result, I got the following Figure **??**, where the desired trajectory is *cos* wave, and its frequency is 2π. Because the robot dynamic always restarts in every trail, the error always starts from 1.

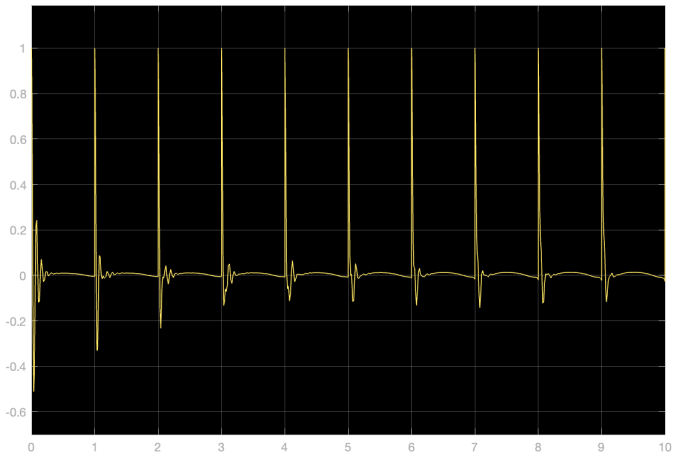

Figure 2: as you can see, the error is decreasing every trail.

2 Explanation

Writing the learning block is easy to do, but how to store and read the data from the last trial is harder. Besides the transport delay block, I also used a clock block and re-settable subsystem block to reset the robot dynamic.

As you can see, the errors decreased in every trial.

2.1 two-joints case

For iterative learning control, it is almost the same as the one-joint case.

However, the result is not good as you can see, maybe it is because parameters are not good enough.