80846 - Report - 1st

GU JUN, 6132230056-4

April 16, 2024

Introduction

The following content will be organized in the following way: Problem Statement, Simulink model, Source codes for RobotDynamics and controller blocks, Simulation results, and Explanation.

Problem Statement

According to the class, we are required to complete two things:
1. Robot dynamics with gravity.
2. PID controller with integration gain.

1 Simulink Model

See the Figure 1 below. The whole system inclues two blocks: RobotDynamics and Controller. And some integration and derivative blocks are used to calculate the error, derror, ierror and dq, q.

2 Formulas and Source Codes

This part inclues the formulas and source codes for RobotDynamics and Controller blocks.

RobotDynamics

The robot dynamics is calculated by the following formula:

$$\ddot{q} = \frac{\tau - m \cdot g \cdot \cos(q)}{I + m \cdot l_g \cdot l_g} \tag{1}$$

where $I = 0.01$, $m = 0.5$, $l_g = 0.2$, $g = 9.8$ accoring to the problem setting.

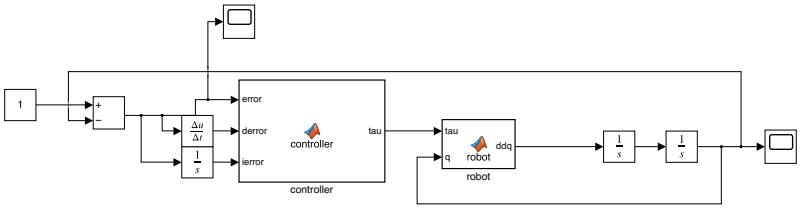

Figure 1: Simulink Model of whole system

The source code is shown below:

```
function ddq = robot(tau, q)

    I = 0.01;

    m = 0.5;

    lg = 0.2;

    g = 9.8;
    % = 0;
    Ibar = I + m * lg * lg;

    tauBar = tau - m * g * lg * cos(q);

    ddq = tauBar / Ibar;
```

Controller

The controller is calculated by the following formula:

$$\tau = K_p \cdot error + K_d \cdot \frac{d}{dt} error + K_i \cdot \int_0^\infty error \tag{2}$$

Because it is not easy to calculate the intgral and derivative in function, I calculate them in simulink model.

The source code is shown below:

```
function tau = controller(error, derror, ierror)

    kp = 8;

    ki = 5;

    kd = 0.7;

    tau = kp * error + kd * derror + ki * ierror;
```

3 Simulation Results

After some tuning of the gains, I got the following Figure 2. The robot can reach the target position in a short time and the error is very small.

4 Explanation

According to the formulas and source codes, most parts are very clear. But there are some points need to be explained.

1. Why the intgrals and derivatives all calculated outside

Because it is not easy to calculate the intgral and derivative in function, you need a environment varible to store the previous values.

2. Why the gains are set to 8, 5, 0.7

I tried many times and found that these gains can make the robot reach the target position in a short time and the error is very small.

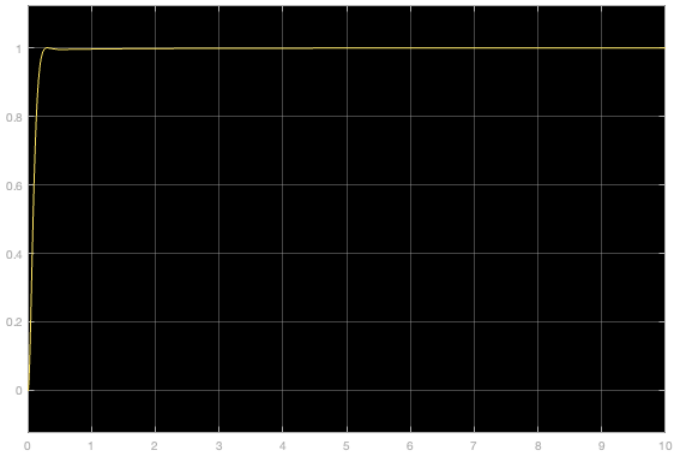

Figure 2: Result Plot, error disapear in 0.3 second.