Sliding Control of a Robot Arm for the Tower of Hanoi Game

Gourav Khadge and Leif Park Jordan

5/8/2015

1. **Introduction and Project Statement**

The Tower of Hanoi game consists of three rods and several disks of decreasing radius. The disks can be stacked on top of one another, but larger disks cannot be placed on top of smaller disks. The goal of the game is to move the stack of disks from the leftmost rod to the rightmost rod. This report will describe an approach used to control a robot arm capable of playing the Tower of Hanoi game. The control system must allow the arm to reach down, pick up a disk, pull it off the rod, and place it on a second rod. The arm must begin and end each move at rest with zero acceleration, it must be able to move the disks in less than 5 seconds, it must stay within 1 mm of a straight line when raising and lowering the disk, and it must be able to adapt to unknown parameters. Specifically, the controller’s estimates of the system’s link inertias, joint/actuator friction coefficients, and end effector load masses can vary from their nominal values. Finally, the maximum torque on each robot arm joint is 3 N-m. The performance of the controller will be verified by programming it to move the robot arm in the pattern shown in Figure 1.

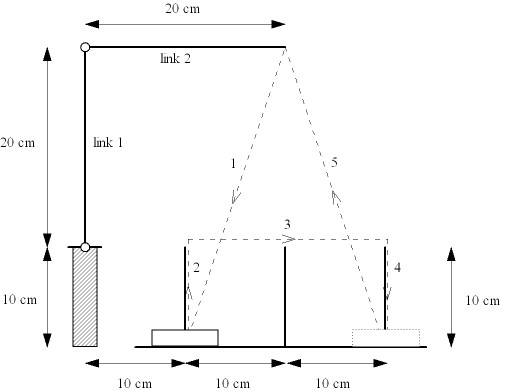


Figure : Target path for Hanoi robot arm.

1. **Design**

The design of the control system consisted of three components – the trajectory generator, the feedback linearized control, and the sliding control. These subsystems were developed one at a time and integrated into the overall controller.

The trajectory generator uses an idealized model of the robot arm to generate a trajectory that results in the arm having zero velocity and acceleration at each “corner” of its motion. This is accomplished by using a 5th order polynomial for each point in the movement:

Where *t* is the target time and *tf* is move time. The lower-order terms are zero because of the requirement that the arm reaches its destination with zero velocity or acceleration. The actual position of the end effector at time *t* is given by

Where *p0* is the initial position for the move and ­*pf* is the final position for the move. The code for this

Once the target trajectory has been generated, the feedback controller is used to ensure that the end effector follows it. The feedback controller is an optimal PD controller designed for the linearized version of the robot arm. We linearize the system by defining the error vectors

So

We can then use these error vectors as state variables in our linearized system:

Where KP and KD are the proportional and derivative feedback gains, respectively. These gains were chosen using MATLAB’s lqr function, with , , and the weighting matricies Q and R chosen to heavily penalize position and velocity errors (making Q a diagonal matrix with large weights in each diagonal entry).

In order to improve the system’s response to unknown parameters, we added a sliding control input. We define the new input to the feedback linearized system as

Where w is the sliding control vector, K is the matrix of proportional and derivative gains determined in the feedback linearization step, and x is the error vector. Choosing a quadratic Lyapunov function

We want P to be a positive definite symmetric matrix that satisfies the Lyapunov equation

Where

Using MATLAB’s lyap function, we can find the value of P for any given KP and KD. We then choose the switching surface to be

Where

To find the sliding control gain vector, we choose U such that

Where . This guarantees global asymptotic stability.

This analysis results in the sliding control block diagram shown in Figure XY.