Lab 2

Jiaming Zhang

* 1. Derivation of Kinematics energy

Joint variable:

Where angles here are both DH angles.

Rotation matrix:

Velocity Jacobian:

Kinematics energy contributed by transition of center of mass:

Angular velocity Jacobian:

Kinematics energy contributed by rotation:

Total kinematics energy:

Substitute all matrixes into the expression of kinematics energy:

Separate the kinematics energy into four items:









To sum up, the whole kinematics energy is:



To simplify, we define some variables:



The whole kinematics energy can also be written as:



Derivation of Potential energy



As the method we use in kinematics energy derivation, we also define some variables:



The whole potential energy can also be written as:



* 1. Dynamic Equations

















We can get inertia matrix, Coriolis matrix and gravitational vector:







State space can be derivate by the matrix we get:



1.3

We have already known the relationship between motor angle and DH angle:



Also, the same as the torque:



Just replace the DH angle and torque by the motor angle and torque in the motion equation:



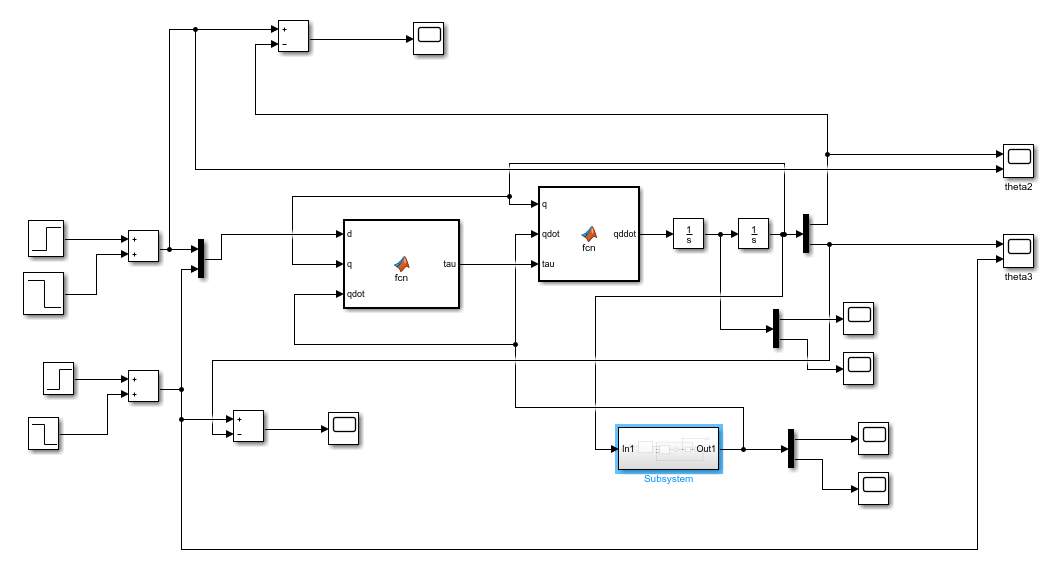
After simplification we can get:



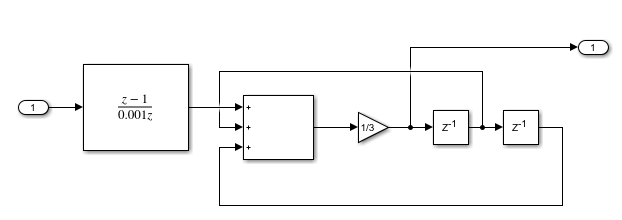


  
2.

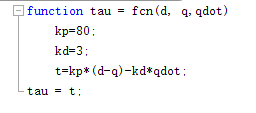
Main model



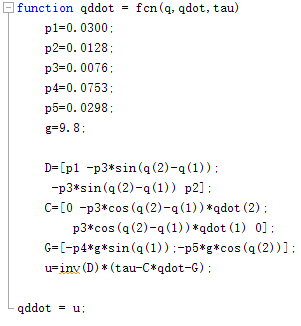
submodel



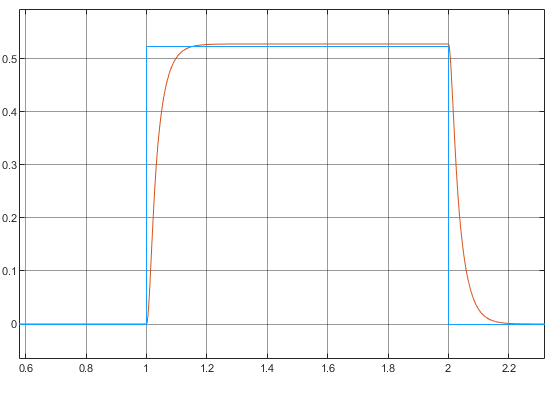
Matlab function 1:



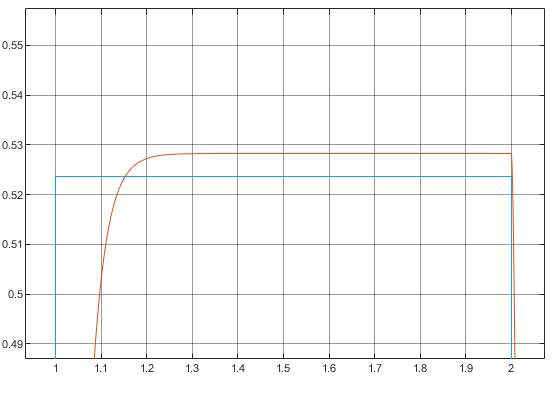
Matlab function 2:



The response of simulation in part 2:

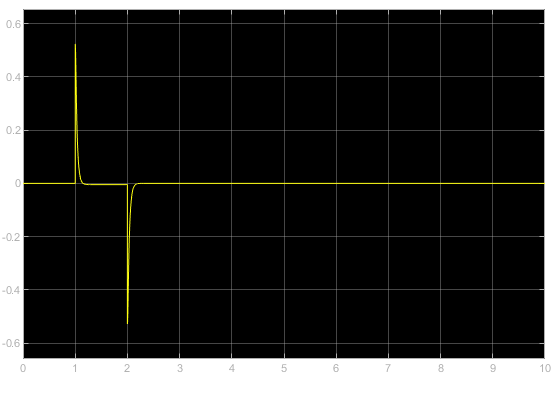


To calculate the overshoot and setting time, we use the image with more details:



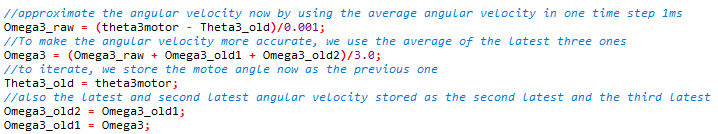
We can find the overshoot is less than 1%, and the setting time is less than 0.3s.

Tracking error



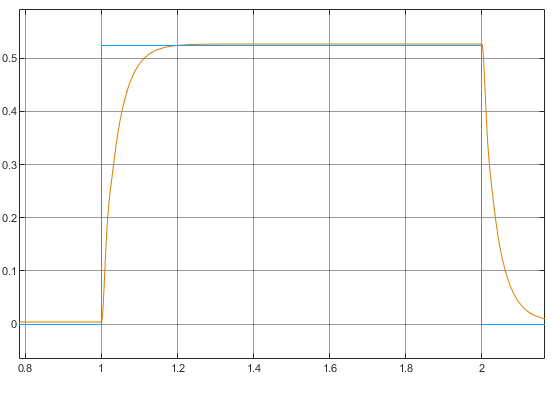
3.

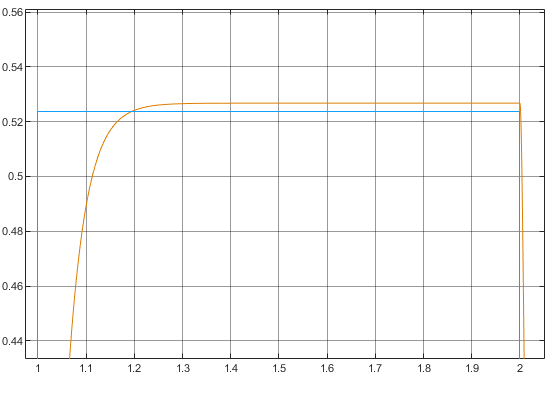
When calculating the velocity, we use the second method, which means we save the disposed velocity as the previous velocity when calculating the latest one, instead of the raw one read from the encoder, which can make the angular velocity calculated more accurate.



4

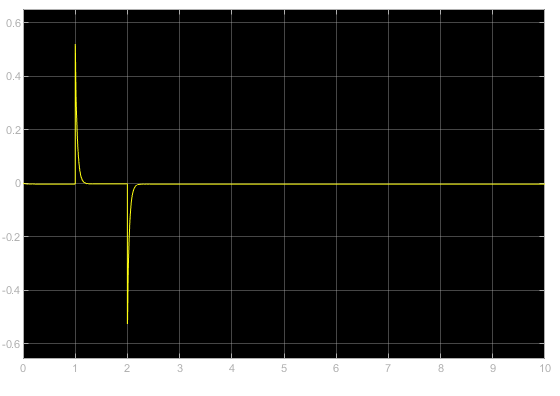
PD control of the robot response:





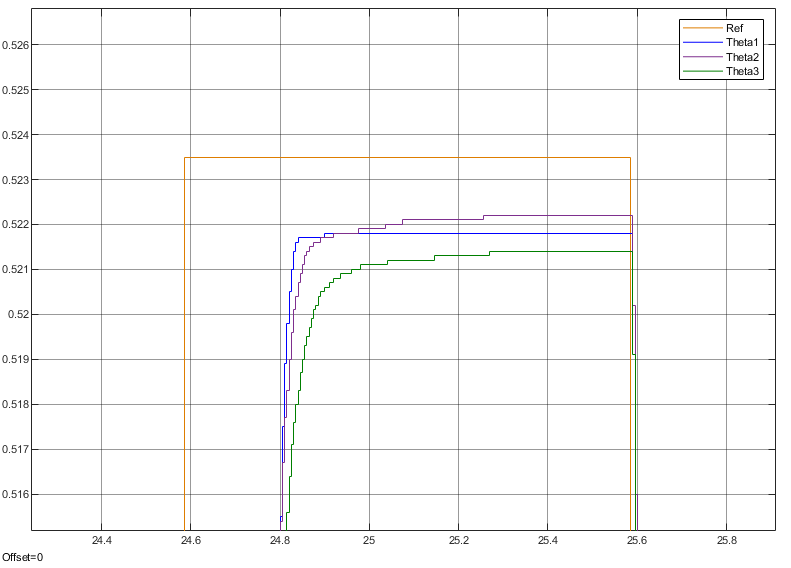
We can find the overshoot is less than 1%, and the setting time is less than 0.3s.

Tracking error



5.

PID control

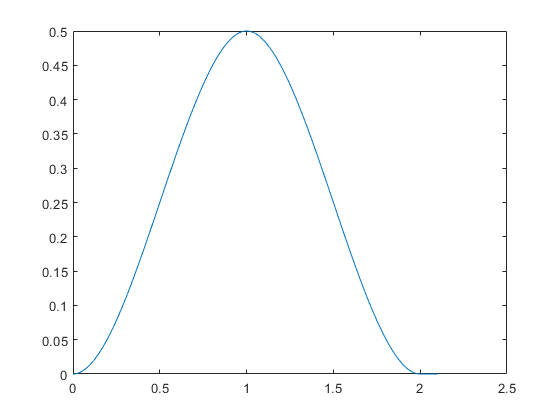


With the integral control, we can see the response is faster, the signal raise fast.

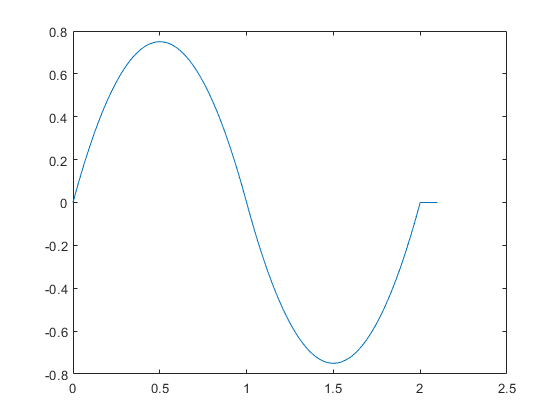
6

With feedforward control:

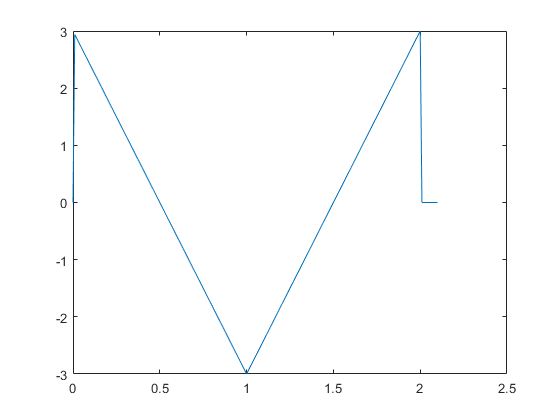
Angle plot:



Angular velocity plot



Accelerate plot:



We can see that the movement is quite smooth, and the move forward and move back process are symmetrical.

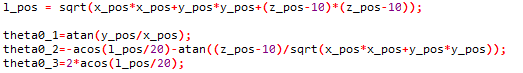
7.

Our trajectory:

We try to draw a circle in the plane vertical to the x-axis, the equation of the trajectory is:



We define another parameter l\_pos to simplify the calculation, and we use inverse kinematics to calculate the motor angle, and use that as an reference to control the robot



PID control response fast and have less state error than PD control. And with feedforward control, the trajectory will be smoother, with fast response.

We have finished all the lab assignment during the lab section. But I am sorry I don’t know whether I have already copied all the file from the computer in the lab.