Problem 1

part 1

The M is shown in matlab code.
M11 is $Iz1 + m3 * (r2 * cos(t2 + t3) + l1 * cos(t2))^2 + Iz3 * cos(t2 + t3)^2 + Iy3 * sin(t2 + t3)^2 + Iz2 * cos(t2)^2 + Iy2 * sin(t2)^2 + m2 * r1^2 * cos(t2)^2$

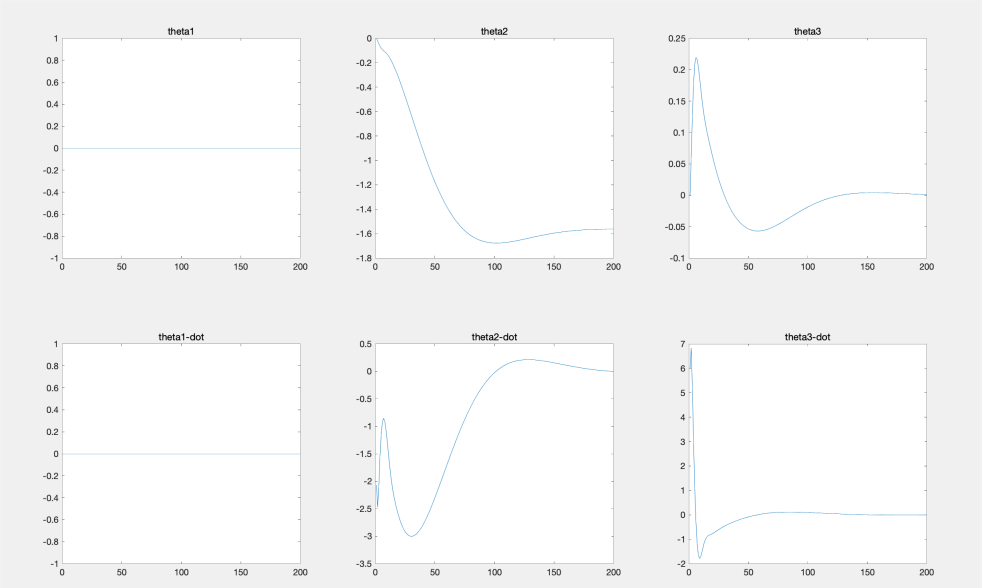
part 2

The C is shown in matlab code.
C21 is $dt1 * ((m3 * sin(2 * t2) * l1^2)/2 + m3 * sin(2 * t2 + t3) * l1 * r2 + (m2 * sin(2 * t2) * r1^2)/2 + (m3 * sin(2 * t2 + 2 * t3) * r2^2)/2 - (Iy3 * sin(2 * t2 + 2 * t3))/2 + (Iz3 * sin(2 * t2 + 2 * t3))/2 - (Iy2 * sin(2 * t2))/2 + (Iz2 * sin(2 * t2))/2)$

part 3

The N is shown in matlab code.
N3 is $-g * m3 * r2 * cos(t2 + t3)$

part 4



To control the arm pointing skywards, I use a simple PD controller to assign the torque.
Parameters assigned to joint 2 is $k_p2 = 3, k_d2 = 1$
Parameters assigned to joint 3 is $k_p3 = 6, k_d3 = 0.1$

Problem2

In inverse kinematic method, we use pseudoinverse method to calculate $\Delta\theta$. $\Delta\theta = (J^T J)^{-1} J^T \vec{e}$
In the field of machine learning, we also use this formula to calculate the weight W for the regression problem, Such that the square error between $W\vec{x}$ and $\vec{y}$ is smallest. The pseudoinverse method is widely discussed in the literature but it often performs poorly because of instability near singularities.Thus, we use damped least squares methods, which can be written as $\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T \vec{e}$
In machine learning field, this is equal to the formula of regression with L2 norm regularization, which is also called Ridge regression. This method can prevent the weight calculated to be too large to overfit the training data set. In the same way, we can regularize the $\Delta\theta$ term so that it is stable around the singularity configuration.