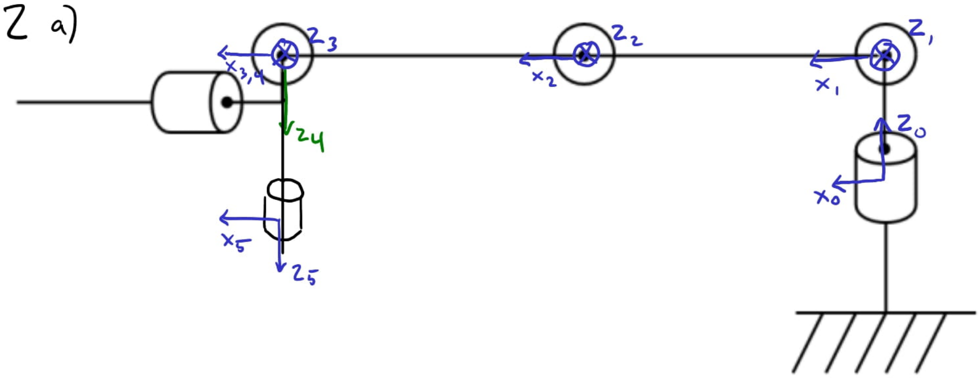
Capstone Robot Report

Eric Jenny, Sidney Nimako
Team 8

December 3, 2021

1 Denavit-Hartenberg Extraction

2 a)



b)

Link	a_i	α_i	d_i	θ_i
1	0	90	56.05	θ_1^*
2	330.30	0	103.55	θ_2^*
3	254.10	0	-73.05	θ_3^*
4	0	90	91.00	θ_4^*
5	0	0	213.75	θ_5^*

$74.05 + 139.70 = 213.75$

$$H_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 56.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 330.30c_2 \\ s_2 & c_2 & 0 & 330.30s_2 \\ 0 & 0 & 1 & 103.55 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & 254.10c_3 \\ s_3 & c_3 & 0 & 254.10s_3 \\ 0 & 0 & 1 & -73.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_4^3 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 91 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_5^4 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 213.75 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_5^0 = \begin{bmatrix} s_1 s_5 + c_{234} c_1 c_5 & c_5 s_1 - c_{234} c_1 s_5 & s_{234} c_1 & a \\ c_{234} c_5 s_1 - c_1 s_5 & -c_1 c_5 - c_{234} s_1 s_5 & s_{234} s_1 & b \\ s_{234} c_5 & -s_{234} s_5 & -c_{234} & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$a = 121.50s_1 + 330.30c_1c_2 + 213.75[c_4(c_1c_2s_3 + c_1c_3s_2) - s_4(c_1s_2s_3 - c_1c_2c_3)] - 254.10[c_1s_2s_3 + c_1c_2c_3]$$
$$b = 330.30c_2s_1 - 121.50c_1 + 213.75[c_4(c_2s_1s_3 + c_3s_1s_2) - s_4(s_1s_2s_3 - c_2c_3s_1)] - 254.10[s_1s_2s_3 + c_2c_3s_1]$$
$$c = 254.10s_{23} - 213.75c_{234} + 330.30s_2 + 56.05$$

ditor - /home/eric/Documents/CMU/16-384/Denavit_Harte...

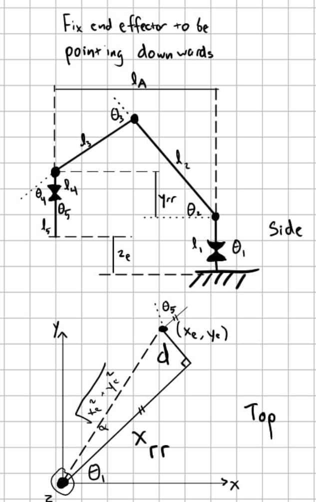
P1.m | Jacobians.m | Denavit_Hartenberg.m

```
syms t1 t2 t3 t4 t5;
dh = [   0   90   56.05   0;
      330.30  0   103.55  0;
      254.10  0   -73.05  0;
         0   90    91.00  0;
         0    0   213.75  0];
t = [t1 t2 t3 t4 t5]*180/pi;
H = eye(4);
format bank
for i = 1:5
    h = DH(dh(i,1), dh(i,2), dh(i,3), t(i));
    disp(h);
    H = H * h;
end
disp(simplify(H));

function H = DH(a, alpha, d, theta)
    H = R_Z(theta) * T(a, 0, d) * R_X(alpha);
end

function R = R_Z(t)
    R = [cosd(t) -sind(t) 0   0;
         sind(t) cosd(t)  0   0;
         0         0      1   0;
         0         0      0   1];
end

function T = T(x, y, z)
     T = [1 0 0 x;
          0 1 0 y;
          0 0 1 z;
          0 0 0 1];
end

function R = R_X(a)
    R = [1     0       0     0;
         0 cosd(a) -sind(a) 0;
         0 sind(a)  cosd(a) 0;
         0     0       0     1];
end
```

Command Window

```
>> Denavit_Hartenberg
[cos(t1), 0,  sin(t1),       0]
[sin(t1), 0, -cos(t1),       0]
[      0, 1,        0, 56.0500]
[      0, 0,        0,       1]

[cos(t2), -sin(t2), 0, 330.3000*cos(t2)]
[sin(t2),  cos(t2), 0, 330.3000*sin(t2)]
[      0,        0, 1,         103.5500]
[      0,        0, 0,                1]

[cos(t3), -sin(t3), 0, 254.1000*cos(t3)]
[sin(t3),  cos(t3), 0, 254.1000*sin(t3)]
[      0,        0, 1,         -73.0500]
[      0,        0, 0,                1]

[cos(t4), 0,  sin(t4),  0]
[sin(t4), 0, -cos(t4),  0]
[      0, 1,        0, 91]
[      0, 0,        0,  1]

[cos(t5), -sin(t5), 0,        0]
[sin(t5),  cos(t5), 0,        0]
[      0,        0, 1, 213.7500]
[      0,        0, 0,        1]

[sin(t1)*sin(t5) + cos(t2 + t3 + t4)*cos(t1)*cos(t5),   cos(t5)*
[cos(t2 + t3 + t4)*cos(t5)*sin(t1) - cos(t1)*sin(t5), - cos(t1)*
[                          sin(t2 + t3 + t4)*cos(t5),
[                                                  0,

>> |
```

2 Analytical Inverse Kinematics



Given $(x_e, y_e, z_e, \theta_e)$ (θ_e is orientation about Z)

$$\text{atan2}(y_e, x_e) = \alpha + \theta_1$$

$$\alpha = \sin^{-1}\left(\frac{d}{\sqrt{x_e^2 + y_e^2}}\right)$$

$$\theta_1 = \text{atan2}(y_e, x_e) - \sin^{-1}\left(\frac{d}{\sqrt{x_e^2 + y_e^2}}\right)$$

θ_2 & θ_3 can be treated as a planor RR arm with

$$y_{rr} = z_e + l_4 + l_5 - l_1$$

$$x_{rr} = \sqrt{x_e^2 + y_e^2 - d^2}$$

$$\theta_2 + \theta_3 + \theta_4 = 0, \quad \theta_4 = -\theta_2 - \theta_3$$

$$\theta_1 + \theta_5 = \theta_e, \quad \theta_5 = \theta_e - \theta_5$$

3 Strategies

Our strategy was to develop an accurate model of the arm in order to allow for the most precise and robust control of it as possible. To that extent we implemented inverse kinematics so we could easily tweak block positions using x-y-z, gravity compensation and torque control to lessen the dependence the arm had on position gains.
We found that as we sped the arm up, it would overshoot our trajectories significantly, especially on theta 1, so we reasoned that this is due to the joint trying to maintain its current velocity because we were commanding torque to zero or to gravity compensation, which would not cause the arm to accelerate. Therefore, we implemented crude static estimates for moments of inertia for the arm about each of the joints, and also commanded the joints to pre-apply the necessary torques to follow the velocity and positions specified by our trajectories. We found this very beneficial when increasing the speeds of our trajectories, as we did not really need to adjust gains in order to make the arm perform faster.
We also decided early on to use cubic spline trajectories exclusively to allow for the smoothest possible motion of the joint. We reasoned that this would be the fastest and most consistent/robust way to move the arm at high speeds. Using only cubic splines also meant our trajectories were simpler to encode, as it was only one trajectory to pick and another trajectory to place. For the remainder of the report the term pick, refers to the location where the blocks are picked up and place, refers to the location where the blocks are placed.
We generated trajectories based on sets of way points:

1. The place location
2. 2 points floating above the place location, one slightly higher than the other
3. 2 points floating above the pick location, one slightly higher than the other
4. The pick location

We also generated timestamps based on two parameters: caution time and travel time. Caution time represented the duration of the slow moving segments of motion, from the floating points to the action points. The travel time is how long it takes to travel from the two highest floating points.

The timestamps were as follows:

0	the place location
caution time * ratio	floating place midpoint
caution time * ratio - caution time	floating place
caution time - caution time + travel time	floating pick
caution time + travel time - caution time + caution time * (1-ratio) + travel time	floating pick midpoint
caution time + caution time * (1-ratio) + travel time - 2 * caution time + travel time	pick location

The durations of segments are:

1. caution time * ratio
2. caution time * (1-ratio)
3. travel time
4. caution time * (1-ratio)
5. caution time * ratio

where ratio represents the percentage of the percentage of the downward path before the secondary point. To achieve the normal 6 level stacking, we tuned the Hebi Robots PID gains, as well as the masses and moments of inertia associated with the links. Once we had a fully functioning 6 level stacking system, we transitioned to a faster time. This consisted primarily of modifying the gains and moments of inertias to function with diminished caution and travel time values.

When optimizing time we discovered better trajectories that used fewer waypoints and adapted both our slower and faster code bases to use these modified points and timings. The waypoints were as follows:

0	pick
caution time	pick high
caution time + travel time / 2	mid point
travel time + caution time	place point

and the movement durations are:

1. caution time
2. travel time / 2
3. travel time

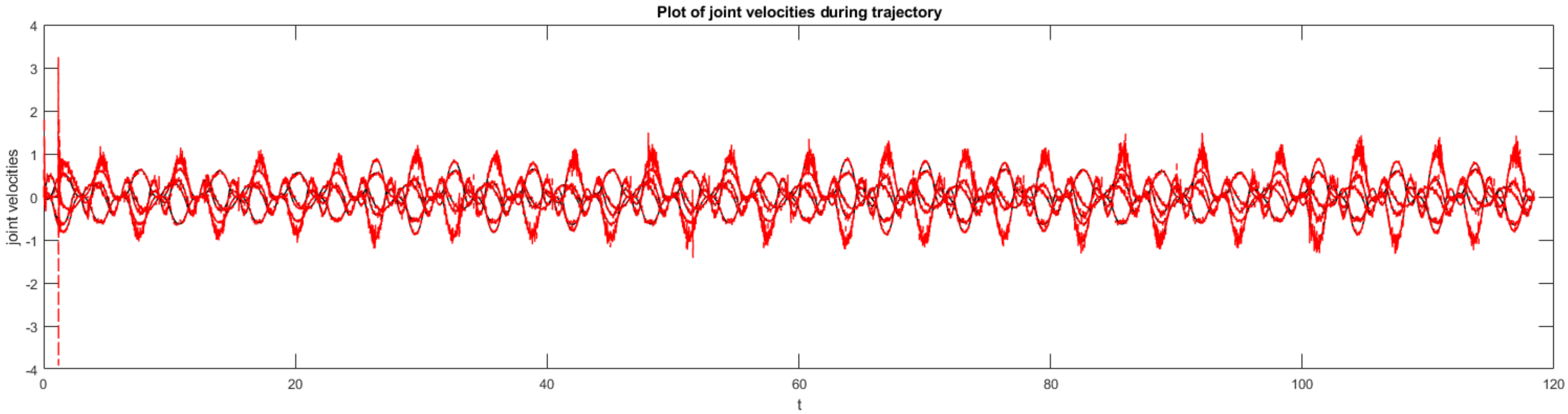
The trajectories for picking and placing where identical in waypoints put had opposite directions and timings.
Another detail we noted, was the large distinction between our gains and those of other groups. We believe that is due to the presence of additional control loops that require larger gains for position to be visible over the torque controls.
We also implemented graphing functionality of trajectories and actual motion that made the process of tuning much simpler and allowed for experimentation with trajectory designs even when not connected to the robot.

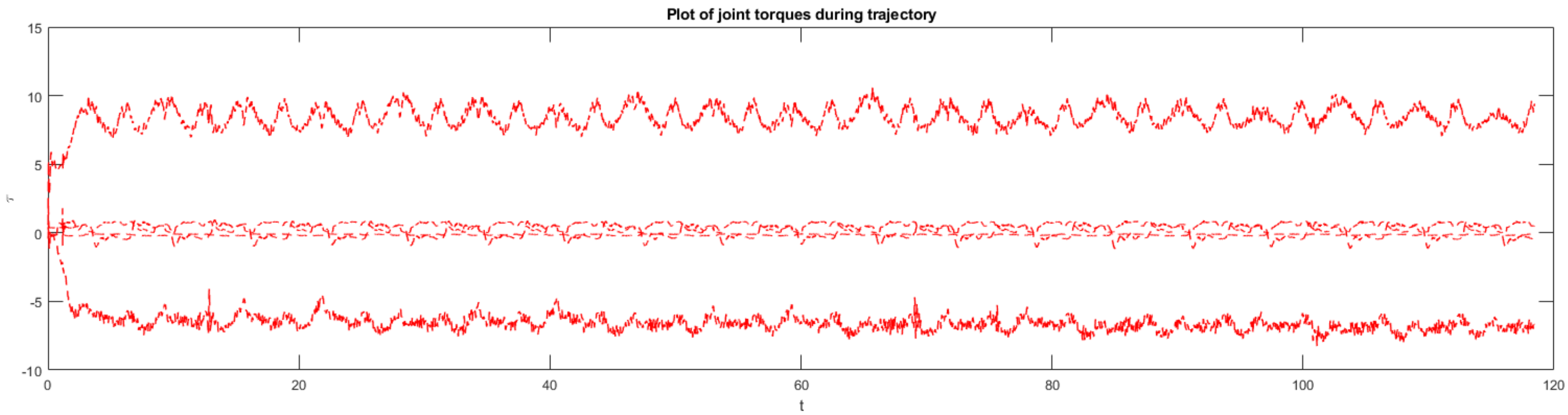
4 Challenges

Our primary challenges were tuning the gains and masses. The robot was very inconsistent in its results from one day to the next, which made determining general values difficult. This presented difficulties based largely on the amount of time that is associated with tuning all of these values and getting time on the robot. Another large source of troubles, especially once we approached higher speeds was the tubing, which we found was very stiff. The tubing for the pneumatic gripper started interfacing with the links and causing additional erratic forces.

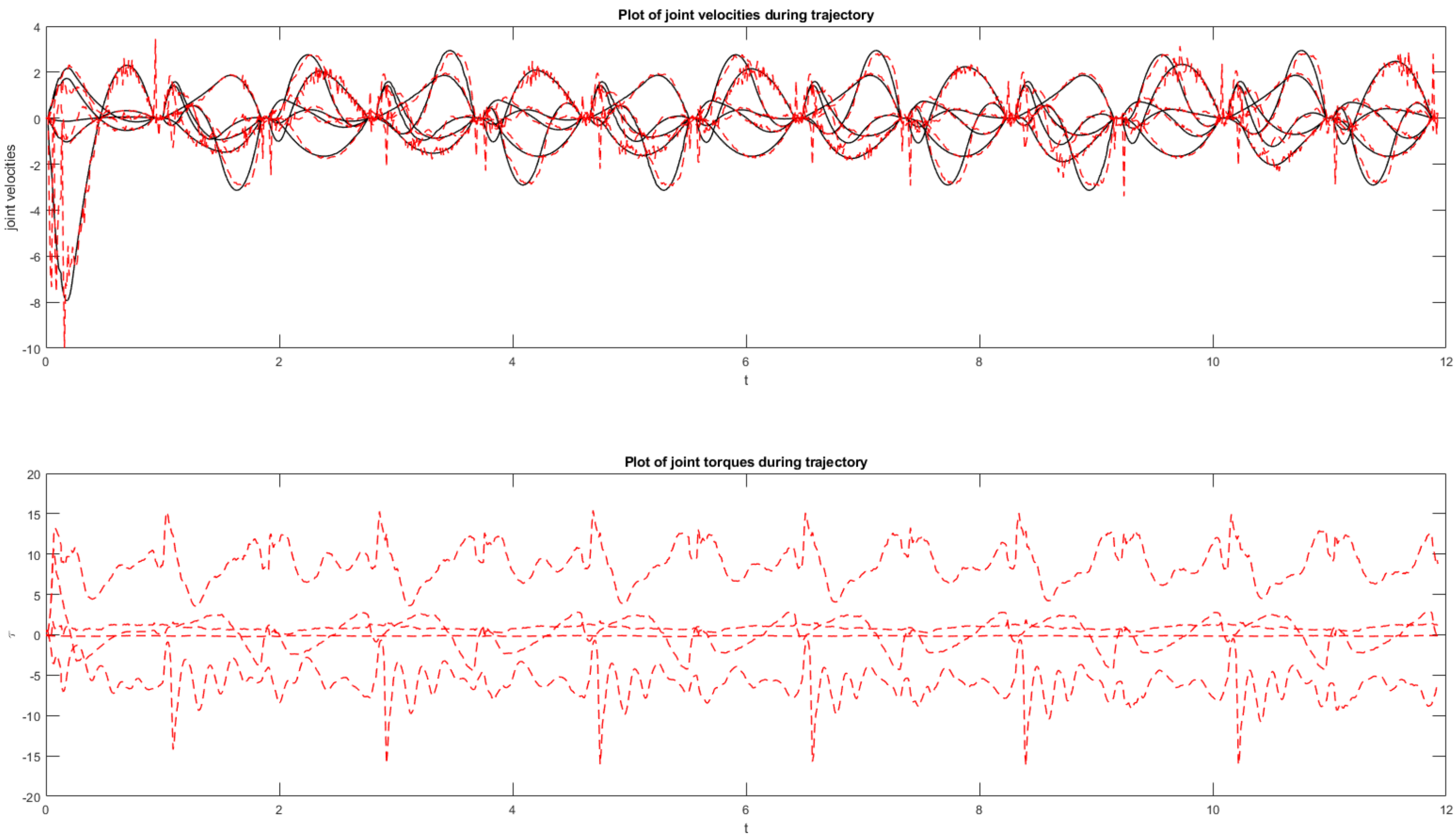
Further tuning was also required when the bracket snapped on the robot, which caused changes to the geometry and therefore the forward and inverse kinematics. We also had issues with the first joint consistently under rotating, which we hypothesize is in part due to the additional forces that result from the pneumatic hose. This is also the believed cause of an issue we had where the placed blocks were slightly under rotated when placed.

5 Plots

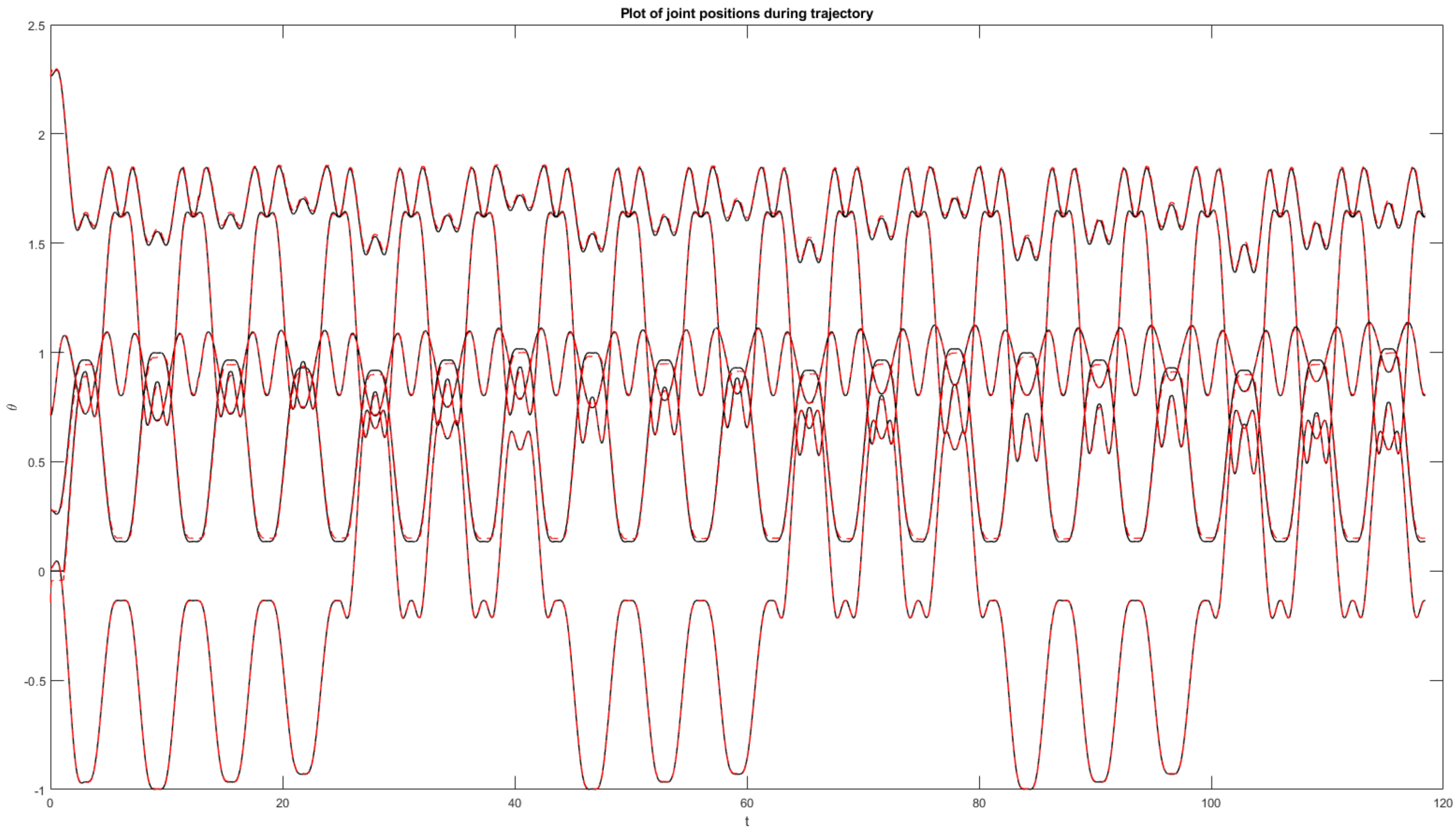




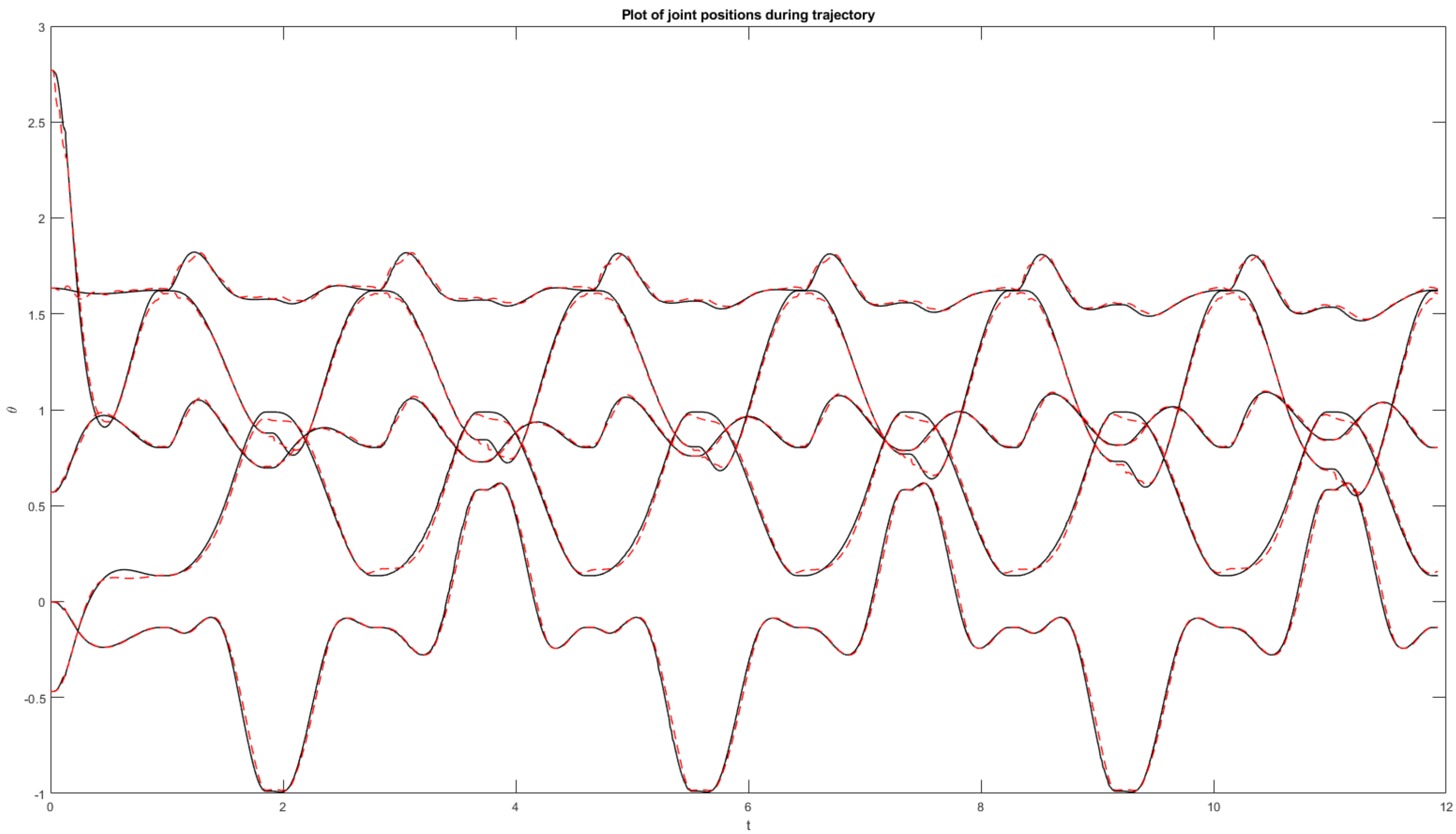
Velocities and Torques for a 6 level 3 block tower



Velocities and Torques for a 6 level 1 block tower (Fast)



Positions for a 6 level 3 block tower



Positions for a 6 level 1 block tower (Fast)