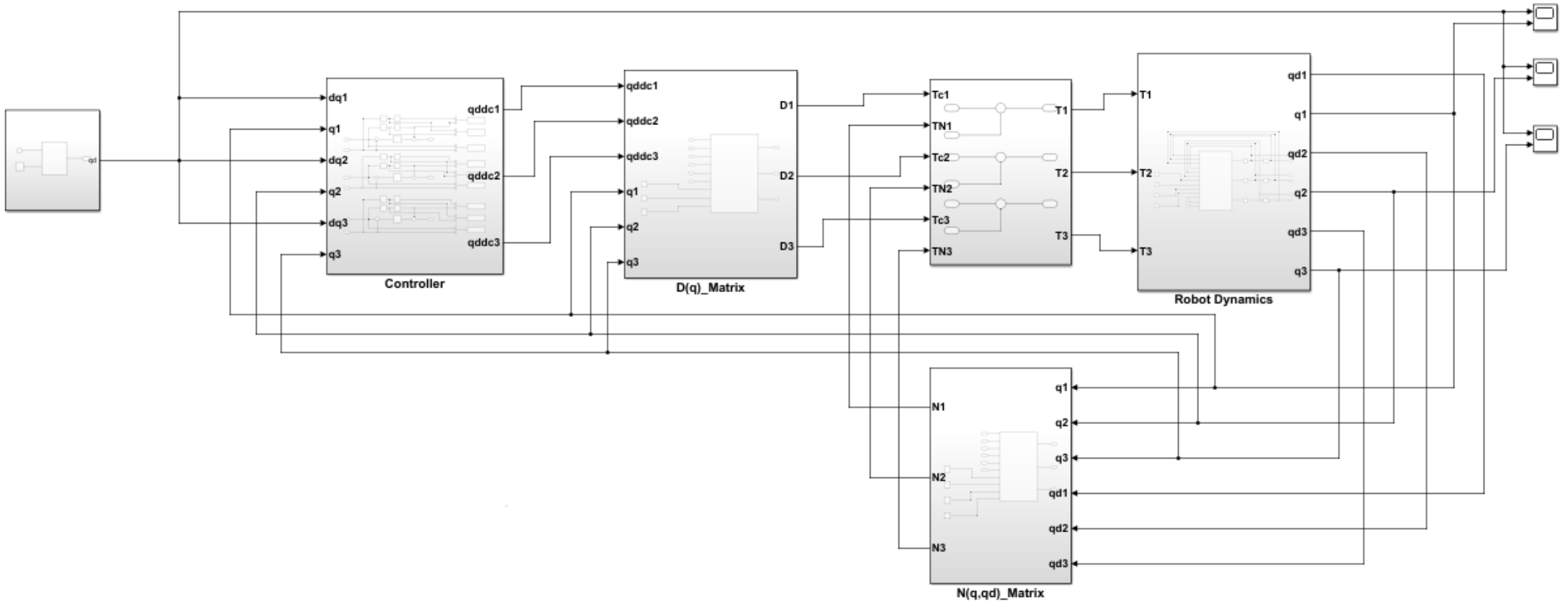
Control of 3-DOF Manipulator

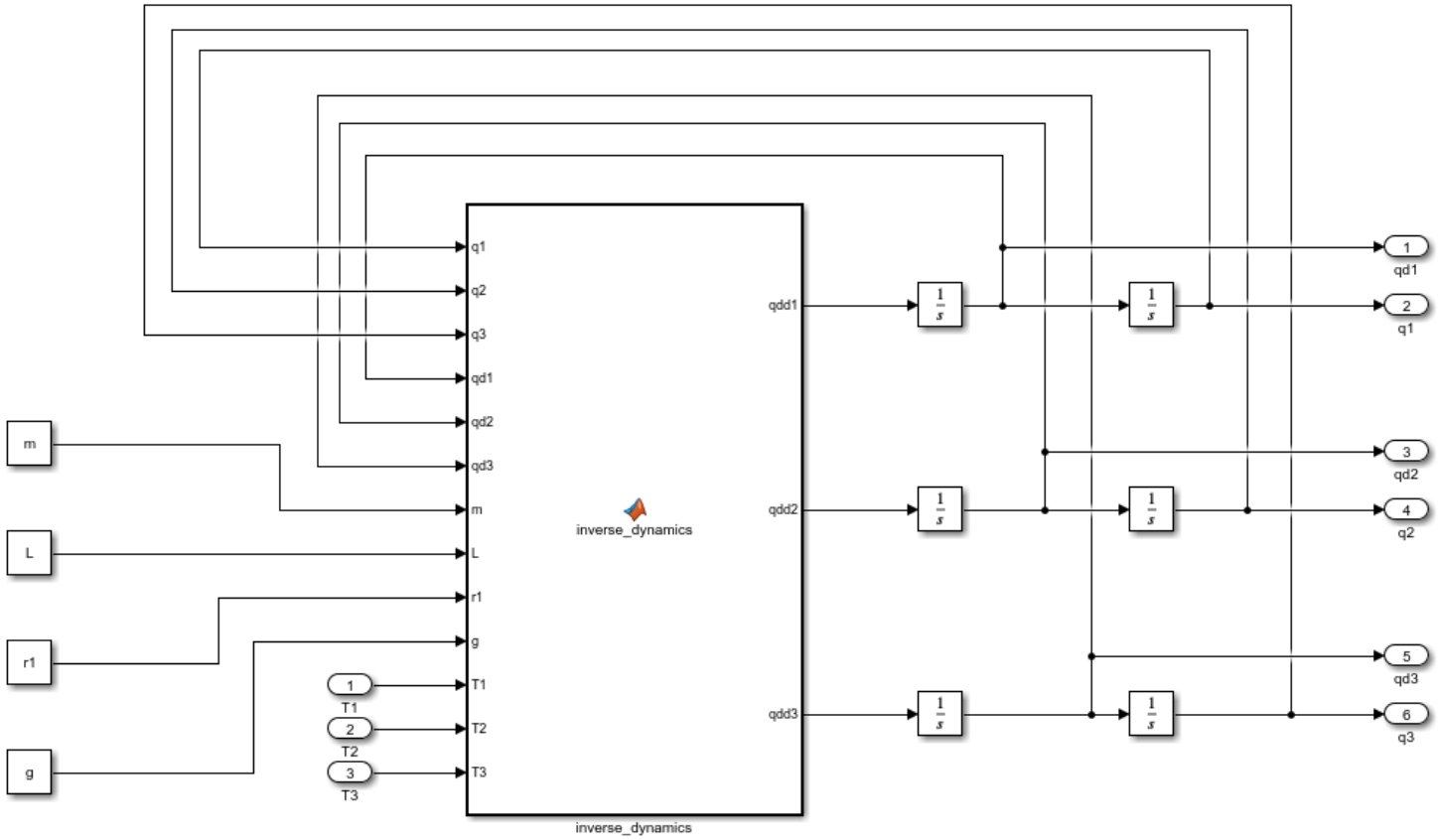
By Computed Torque Control

Simulink Model

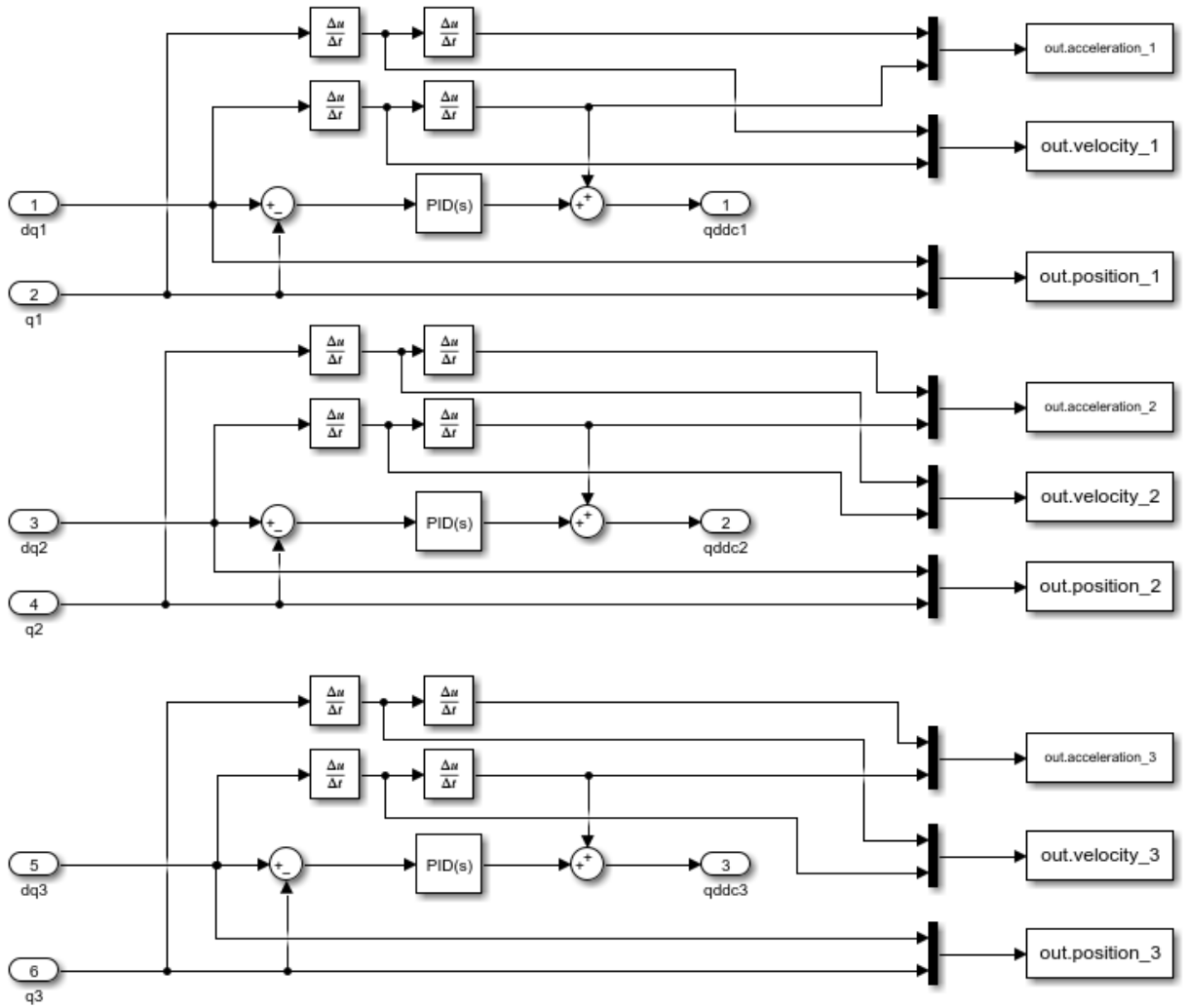
Simulink Model



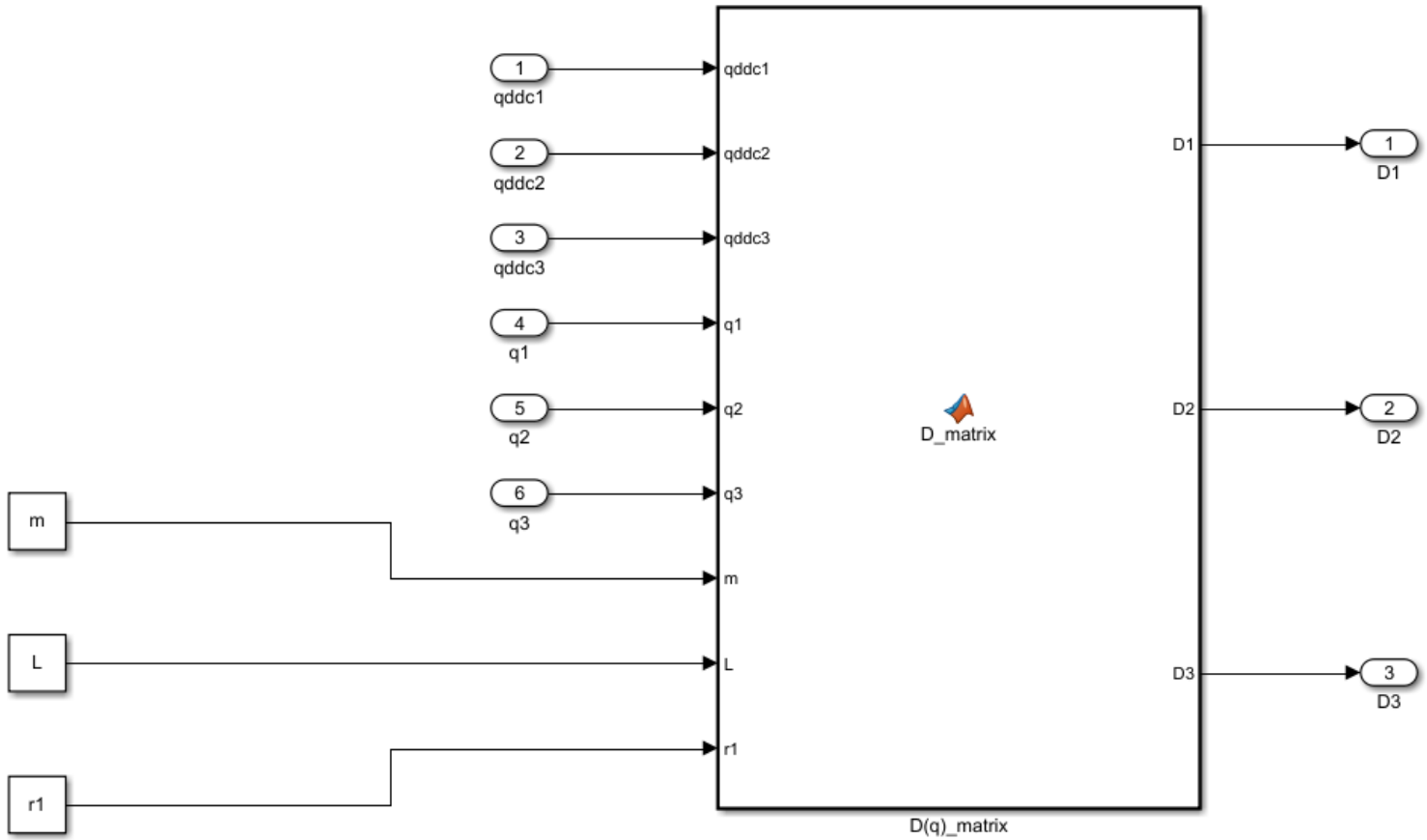
Robot Dynamics



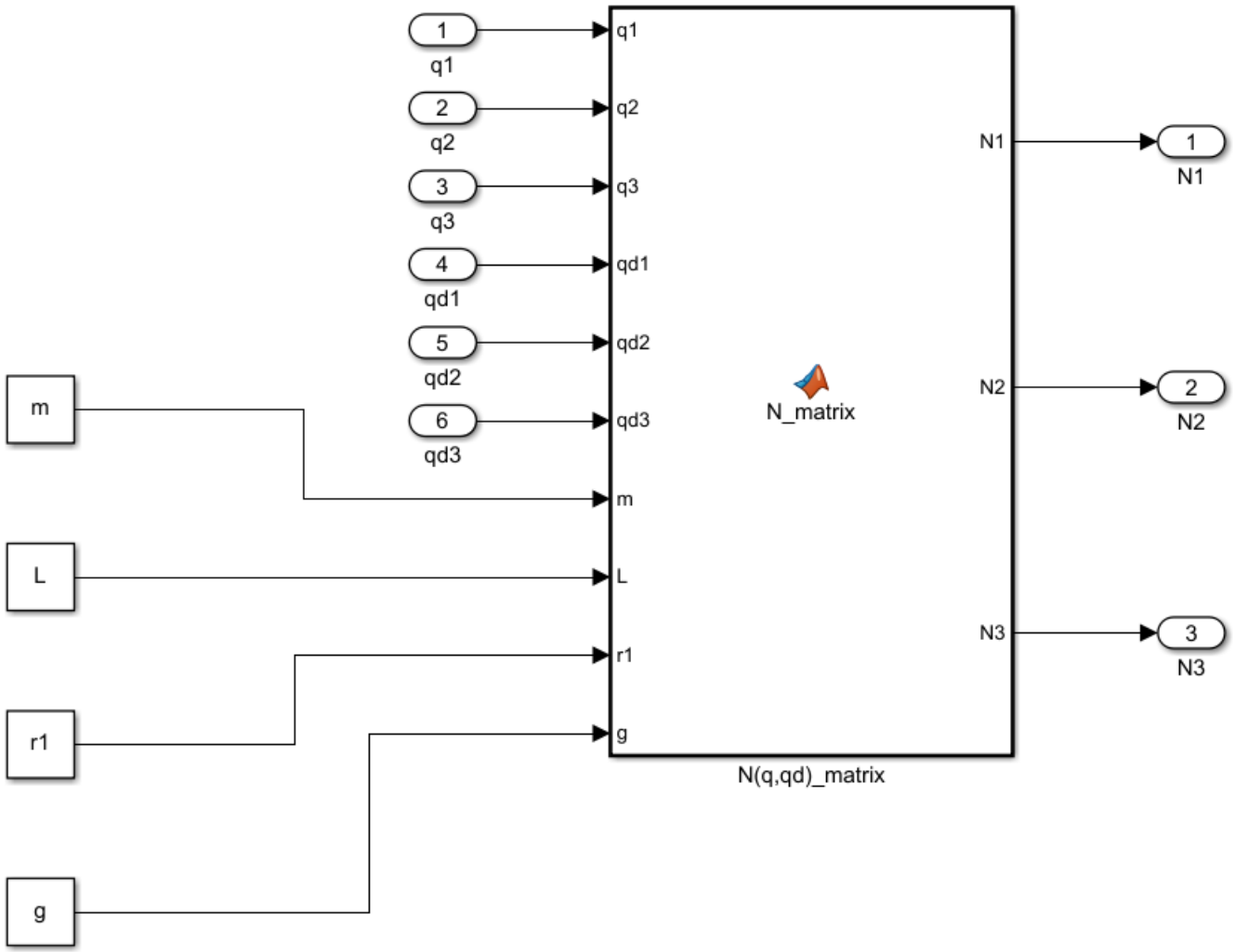
Controller


1
dq1
2
q1
PID(s)
1
qddc1
out.acceleration_1
out.velocity_1
out.position_1
3
dq2
4
q2
PID(s)
2
qddc2
out.acceleration_2
out.velocity_2
out.position_2
5
dq3
6
q3
PID(s)
3
qddc3
out.acceleration_3
out.velocity_3
out.position_3

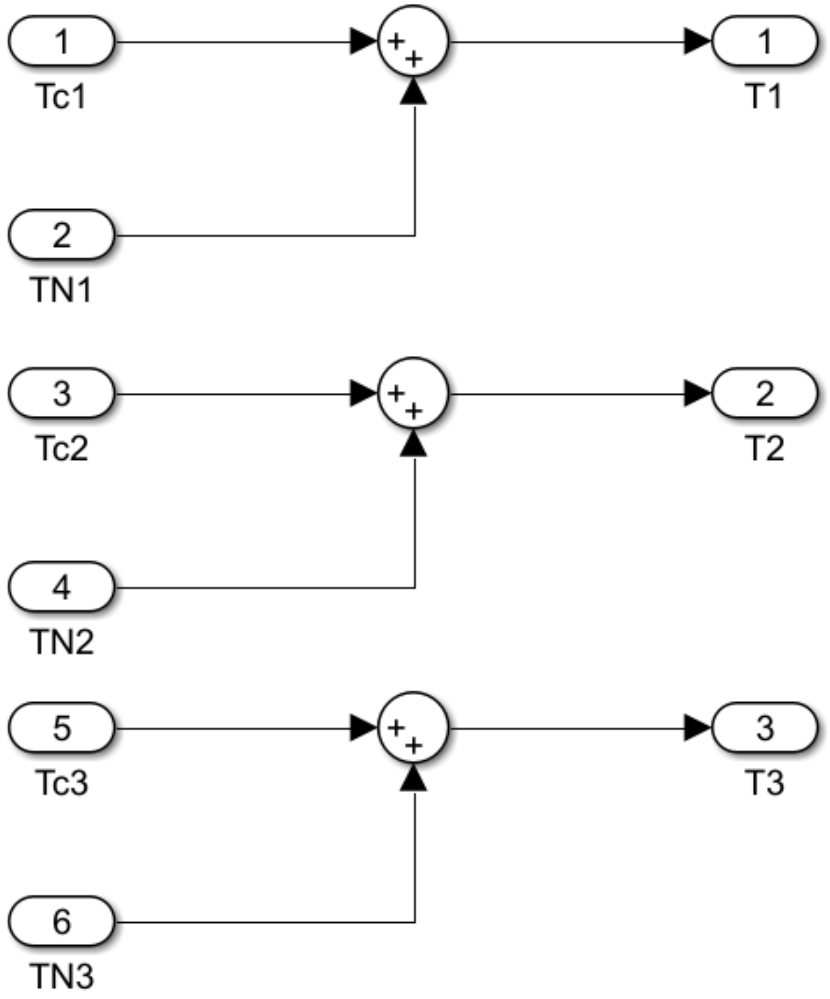
D(q) Matrix


1
qddc1
2
qddc2
3
qddc3
4
q1
5
q2
6
q3
m
L
r1
qddc1
qddc2
qddc3
q1
q2
q3
m
L
r1
D_matrix
D1
D2
D3
1
D1
2
D2
3
D3
D(q)_matrix

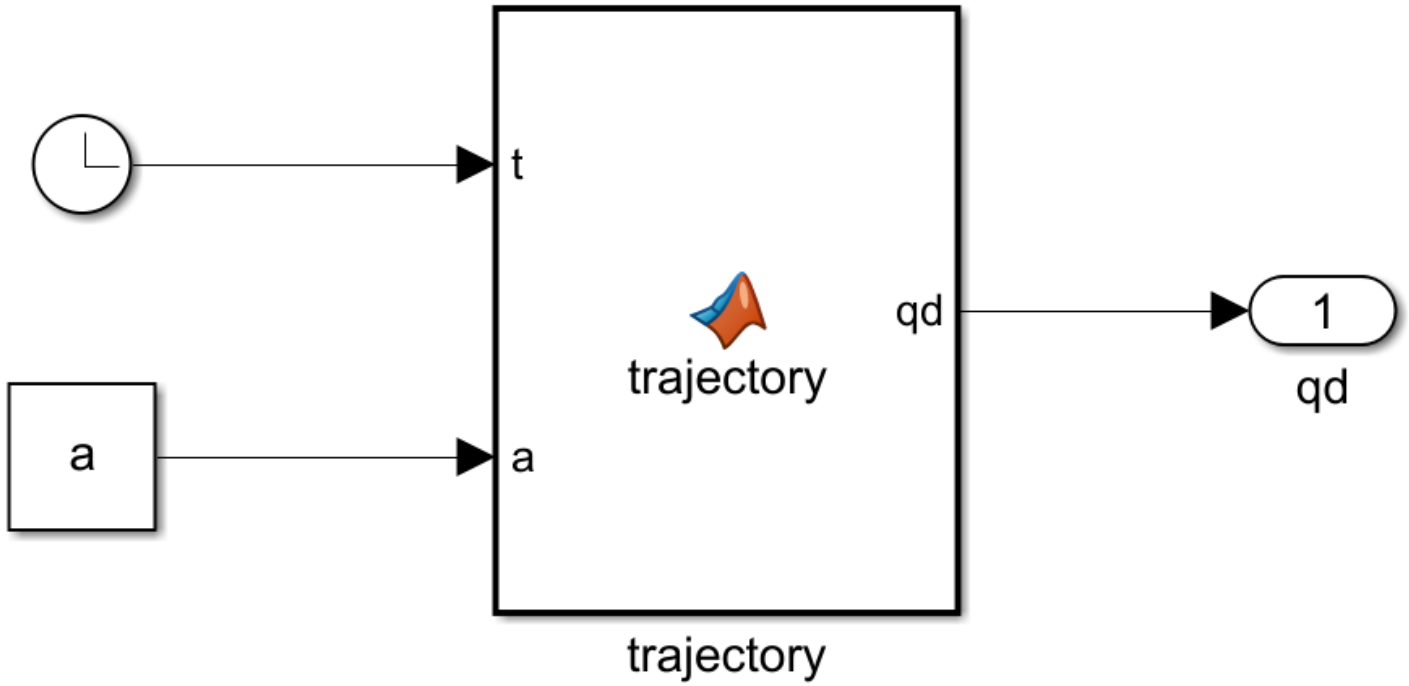
N(q, $\dot{q}$) Matrix



Summation



Trajectory



M File

```
%%
% Written by Mohamed Eid
clear
clc
```

```
%% Parameters
m = [0.1,0.3,0.2];
L = [0.1,0.4,0.3];
r1 = 0.025;
g = 9.81;
```

```
%% Trajectory
d = [0 0 0 0.5 0 0 0 10];
q0 = d(1); v0 = d(2); ac0 = d(3);
q1 = d(4); v1 = d(5); ac1 = d(6);
t0 = d(7); tf = d(8);
M = [ 1 t0 t0^2 t0^3   t0^4    t0^5;
      0 1  2*t0 3*t0^2 4*t0^3  5*t0^4;
      0 0  2    6*t0   12*t0^2 20*t0^3;
      1 tf tf^2 tf^3   tf^4    tf^5;
      0 1  2*tf 3*tf^2 4*tf^3  5*tf^4;
      0 0  2    6*tf   12*tf^2 20*tf^3];
b=[q0; v0; ac0; q1; v1; ac1];
a = M\b;
```

```
%% Simulation
Data = sim('RRR_Robot_Control');
```

M File

```
function qd = trajectory(t, a)
    qd = a(1) + a(2)*t +a(3)*t^2 + a(4)*t^3 +a(5)*t^4 + a(6)*t^5;
end
```

D(*q*) Matrix

```
function [D1,D2,D3] = D_matrix(qddc1, qddc2, qddc3, q1, q2, q3, m, L, r1)
    %Parameters
    m1 = m(1); m2 = m(2); m3 = m(3);
    L1 = L(1); L2 = L(2); L3 = L(3);
    %Equations
    D = [(L2^2*m2)/6 + (L2^2*m3)/2 + (L3^2*m3)/6 + (m1*r1^2)/2 + (L2^2*m2*cos(2*q2))/6
        0, (L2^2*m2)/3 + L2^2*m3 + (L3^2*m3)/3 + L2*L3*m3*cos(q3), (L3*m3*(2*L3 + 3*L2
        0, (L3*m3*(2*L3 + 3*L2*cos(q3)))/6, (L3^2*m3)/3];
    Dc = D*[qddc1;qddc2;qddc3];
    %Output
    D1 = Dc(1); D2 = Dc(2); D3 = Dc(3);
end
```

N(q, $\dot{q}$) Matrix

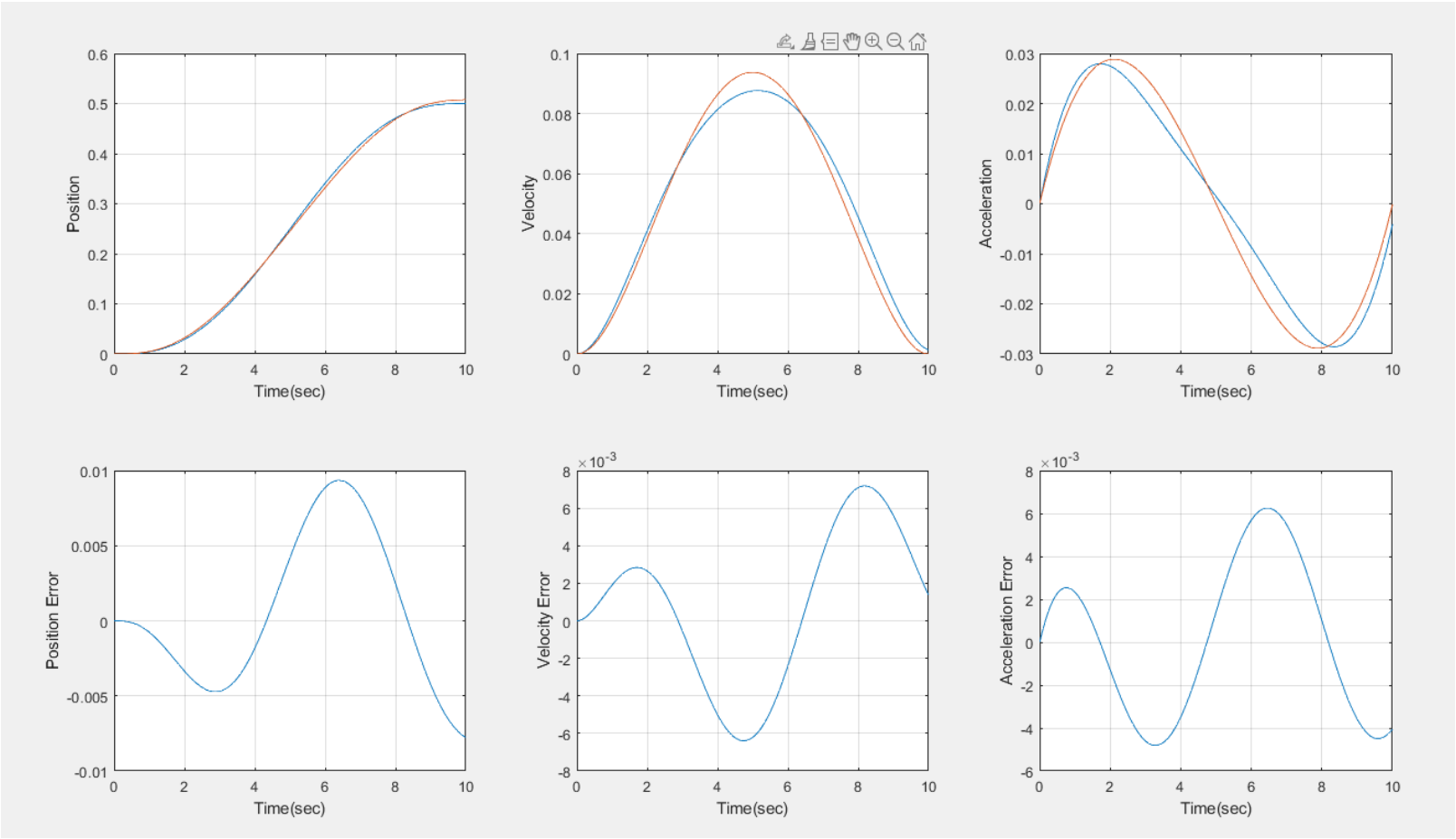
```
function [qdd1,qdd2,qdd3] = inverse_dynamics(q1, q2, q3, qd1, qd2, qd3, m, L, r1, g, T1,
    %Parameters
    m1 = m(1); m2 = m(2); m3 = m(3);
    L1 = L(1); L2 = L(2); L3 = L(3);
    %Output
    qdd1 = (6*T1 + 2*L2^2*m2*qd1*qd2*sin(2*q2) + 6*L2^2*m3*qd1*qd2*sin(2*q2) + 2*L3^2*m3
    qdd2 = (48*L3*T2 - 48*L3*T3 - 72*L2*T3*cos(q3) + 9*L2^2*L3*m3*qd1^2*sin(2*q2 + 2*q3)
    qdd3 = -(24*L3^2*T2*m3 - 72*L2^2*T3*m3 - 24*L2^2*T3*m2 - 24*L3^2*T3*m3 - 3*L2*L3^3*m

end
```

Robot Dynamics

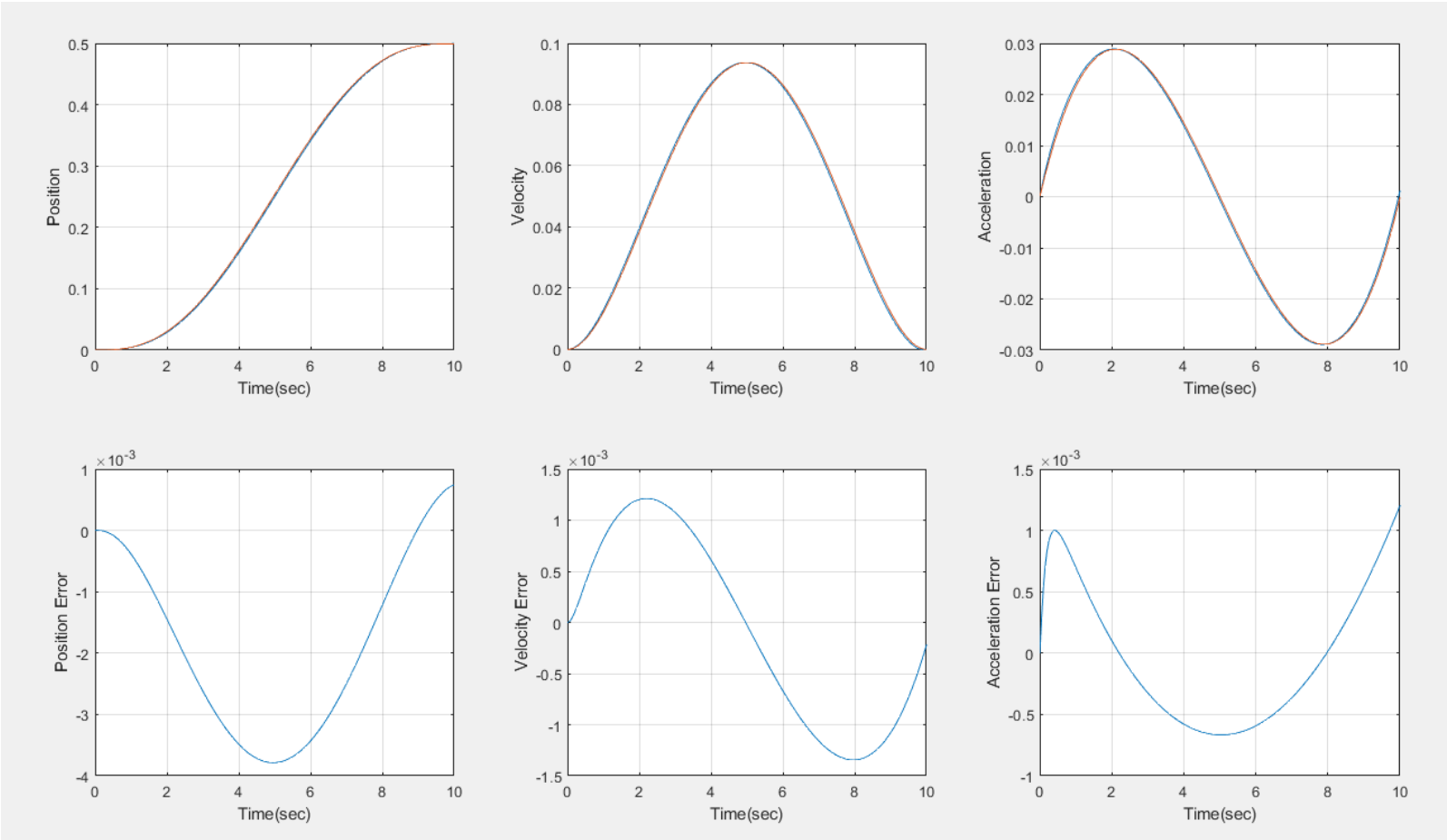
```
function [N1,N2,N3] = N_matrix(q1, q2, q3, qd1, qd2, qd3, m, L, r1, g)
    %Parameters
    m1 = m(1); m2 = m(2); m3 = m(3);
    L1 = L(1); L2 = L(2); L3 = L(3);
    %Equations
    N = [- qd1*qd2*((L2^2*m2*sin(2*q2))/3 + L2^2*m3*sin(2*q2) + (L3^2*m3*sin(2*q2 + 2*q3
        ((L2^2*m2*sin(2*q2))/6 + (L2^2*m3*sin(2*q2))/2 + (L3^2*m3*sin(2*q2 + 2*q3))/6 +
        (L3*m3*(6*g*cos(q2 + q3) + 3*L2*qd1^2*sin(q3) + 6*L2*qd2^2*sin(q3) + 3*L2*qd1^2*
    %Output
    N1 = N(1); N2 = N(2); N3 = N(3);
end
```

Output

Before Tuning



After Tuning



Thank You