

SIMULATION OF A THREE-LINK, SIX-MUSCLE MUSCULO-SKELETAL ARM ACTIVATED BY HILL MUSCLE MODEL

Nafiseh Ebrahimi

Robotics and Motion Laboratory,
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
nafiseh.ebrahimi@utsa.edu

ABSTRACT

In this study human arm modeled with a three-link planar manipulator activated by Hill muscle model. Employing controller, values of force and torques applied to the joints calculated and finally, solving dynamics differential equations, joints and muscle forces trajectories was calculated and presented.

1. INTRODUCTION

In this study, we modeled human arm with a three-link planar manipulator with six muscles to travel on a desired path. For muscles, a nonlinear model named Hill muscle based on the physiological concept of real muscles has been used. We applied the trend based on the method used in [1, 2]. For making the movement of manipulator, a task-space which is the location of end-effector of manipulator, was defined and a simple feedback was applied on that. Joint space also defined as the states of each joint including position and velocity. Muscle space is the third space we are using in this study to address the length and linear velocity of the muscles. It is shown that damping factor of each joint play a significant role in the convergence of the end effector trajectory to the desired path. So, an internal force term besides to the task-space feedback was considered in order to control the damping factors in joint space [1].

In the next section, we formulate the kinematics of Muscle-joint space by which we define the relationship between muscle lengths and geometry of links and joints. In the following, we introduce the kinematic of links to calculate the position of end effector in terms of geometry. Secondly, we consider the modeling of Hill muscle dynamics and after calculation of joint torques. Finally, we apply the joint torques to the dynamics of the three-link manipulator obtained from inverse dynamics in order to calculate the joint states, find, and control the location of manipulator's end-effector.

2. METHODS

The three-link arm manipulator model and equations are exerted from the combination of method and equations presented in [1, 2]. The manipulator used to model the arm is shown in fig.1.

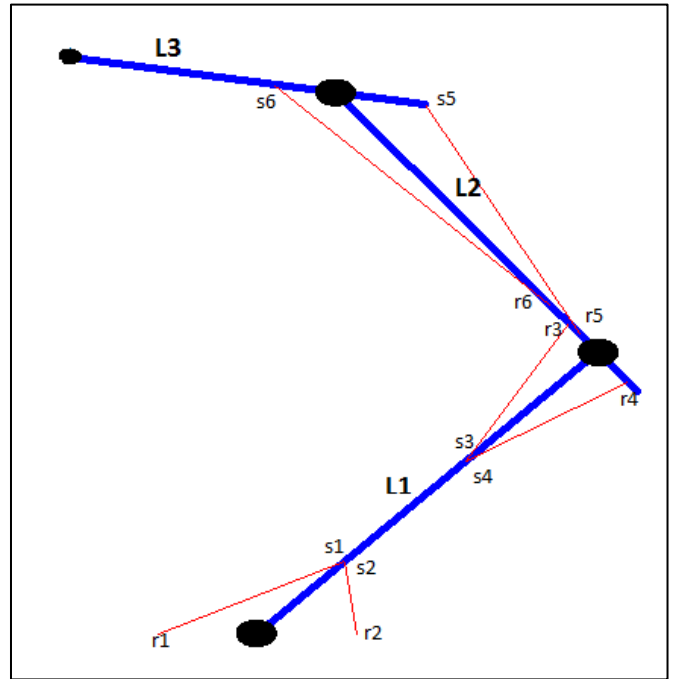


Figure 1. Three-link manipulator model representing arm.

In first step, we need define the kinematics of muscle and joints. Kinematics between muscle space and joint space is expressed in the following equation:

$$l = (l_1, l_2, l_3, l_4, l_5, l_6)^T = \begin{bmatrix} (r_1^2 + s_1^2 + 2r_1s_1 \cos \theta_1)^{1/2} \\ (r_2^2 + s_2^2 - 2r_2s_2 \cos \theta_1)^{1/2} \\ (r_3^2 + s_3^2 + 2r_3s_3 \cos \theta_2)^{1/2} \\ (r_4^2 + s_4^2 - 2r_4s_4 \cos \theta_2)^{1/2} \\ (r_5^2 + s_5^2 - 2r_5s_5 \cos \theta_3)^{1/2} \\ (r_6^2 + s_6^2 - 2r_6s_6 \cos \theta_3)^{1/2} \end{bmatrix} \quad (1)$$

Here, $l \in R^6$ is the length vector representing length of all muscles, $(\theta_1, \theta_2, \theta_3)$ is the angle of all three joints. r_{1-4} and

S_{1-4} are positions at them muscles connect to the bones. Taking derivative of eq. (1) leads to eq. (2).

$$\dot{l} = Q(\theta)\dot{\theta} \quad (2)$$

$\dot{l} \in R^6$ is the vector of linear velocity (contractile) of the muscles while $\dot{\theta} \in R^3$ is the vector of angular velocity of the joints.

$Q(\theta) \in R^{6 \times 3}$ is the Jacobian matrix from the joint space to the muscle space.

$$Q = \begin{bmatrix} q_{11} & q_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{23} & q_{24} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{35} & q_{36} \end{bmatrix}^T$$

$$q_{11} = \frac{-r_1 s_1 \sin \theta_1}{(r_1^2 + s_1^2 + 2r_1 s_1 \cos \theta_1^2)^{1/2}}$$

$$q_{12} = \frac{r_2 s_2 \sin \theta_1}{(r_2^2 + s_2^2 - 2r_2 s_2 \cos \theta_1^2)^{1/2}}$$

$$q_{23} = \frac{-r_3 s_3 \sin \theta_2}{(r_3^2 + s_3^2 + 2r_3 s_3 \cos \theta_2^2)^{1/2}}$$

$$q_{24} = \frac{r_4 s_4 \sin \theta_2}{(r_4^2 + s_4^2 - 2r_4 s_4 \cos \theta_2^2)^{1/2}}$$

$$q_{35} = \frac{-r_5 s_5 \sin \theta_2}{(r_5^2 + s_5^2 + 2r_5 s_5 \cos \theta_3^2)^{1/2}}$$

$$q_{36} = \frac{r_6 s_6 \sin \theta_3}{(r_6^2 + s_6^2 - 2r_6 s_6 \cos \theta_3^2)^{1/2}} \quad (3)$$

In next step we need to find joint torques form the muscle forces. The relation can be expressed by means of virtual work principal.

$$\tau = WF \quad (4)$$

$F \in R^6$ is the vector of muscle forces, $\tau \in R^3$ is the vector of joint torques and $W = Q(\theta)^T \in R^{3 \times 6}$ which is introduced before in eq. (2)

The inverse relationship between the joint torque and the muscle force can be obtained as follows:

$$F = W^+ \tau + (I - W^+ W) k_e \quad (5)$$

While $W^+ = W^T (W W^T)^{-1} \in R^{6 \times 3}$ is pseudoinverse matrix of W and $(I - W^+ W) \in R^{6 \times 6}$ means null space of W^+ , and $k_e \in R^6$ is a voluntary vector that represent the internal force generated by muscles.

After expressing the kinematics between muscle space and joint space, we need to obtain the kinematics of the links. The position of end effector can be calculated as following:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{pmatrix} \quad (6)$$

After taking derivative respect to time, we have:

$$\dot{x} = J(\theta)\dot{\theta} \quad (7)$$

Where $\dot{x} \in R^2$ is the vector of end-effector velocity and $J(\theta) \in R^{3 \times 2}$ is the Jacobian matrix defined from the task space (end-effector position) to joint space as following:

$$J(\theta) = \begin{pmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \end{pmatrix}$$

$$j_{11} = -L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$j_{12} = -L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$j_{13} = -L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$j_{21} = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$j_{22} = L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$j_{23} = L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (8)$$

Where L_{1-3} are the lengths of three links. In the following, we consider the modeling of muscle dynamics based on Hill model. The viscosity of muscle depends on the activation level of muscle. The Hill muscle inspired by some physiological concepts and mathematically expressed as follows:

$$F = \bar{\alpha} - (\bar{\alpha} b + b_0) \dot{l} \quad (9)$$

Where is the output force produced by muscle and applied to the correlated link, and $\bar{\alpha} = \alpha f_0 \geq 0$ ($\bar{\alpha}$ is the control input, f_0 is the maximum output force of the isometric contraction model

and α is the muscle activation level). In this study $\bar{\alpha}$ is considered to keep semipositive. b and b_0 are two damping coefficients. b is the damping coefficient depends on the control input $\bar{\alpha}$, and $b_0 > 0$ is damping coefficient belong to the muscle by its own and independent of $\bar{\alpha}$. Since based on the physiological research, muscle have a springlike behavior, an end-effector position feedback control signal to the muscle input ($\bar{\alpha}$) represent this behavior:

$$\bar{\alpha} = -W^+ J^T K_p \Delta x + (I - W^+ W) k_e \quad (10)$$

Where $J \in R^{2 \times 3}$ is the Jacobian matrix from task space to joint space (defined in eq. (7)), $K_p \in R^{2 \times 2}$ is the position feedback gain, $x \in R^2$ is the position of end-effector and $\Delta x = x - x_d \in R^2$ is the position error from desired position. $(I - W^+ W) k_e \in R^6$ is the null space vector that determine the internal force and $k_e \geq 0$ means a voluntary vector. The first term in eq. (10) represent the springlike force of muscle. Generalizing the Hill model for all muscles in the system, the overall muscle dynamics obtained as following:

$$F = \begin{pmatrix} \bar{\alpha}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \bar{\alpha}_6 \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - \begin{pmatrix} \bar{\alpha}_1 b_1 + b_{01} & & 0 \\ & \ddots & \\ 0 & & \bar{\alpha}_6 b_6 + b_{06} \end{pmatrix} \begin{pmatrix} \dot{i}_1 \\ \vdots \\ \dot{i}_6 \end{pmatrix} \\ = AU - (AB + B_0) \dot{i} \quad (11)$$

Where

$$A = \text{diag}(\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \bar{\alpha}_4, \bar{\alpha}_5, \bar{\alpha}_6)$$

$$U = (1, 1, 1, 1, 1, 1)^T$$

$$B = \text{diag}(b_1, b_2, b_3, b_4, b_5, b_6)$$

$$B_0 = \text{diag}(b_{01}, b_{02}, b_{03}, b_{04}, b_{05}, b_{06})$$

Forces of all muscle in the system are obtainable form eq. (11) which are substituted in eq. (4) to calculate the three torques applied to the three links.

In order to calculate joint trajectory we need joint angles and angular (states of the system) velocity. Inverse dynamics would be exerted to calculate six states (three angles and three angular velocities). Dynamics equations extracted form inverse dynamics used in [2] expressed as follows:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} M(\theta)_{11} & M(\theta)_{12} & M(\theta)_{13} \\ M(\theta)_{21} & M(\theta)_{22} & M(\theta)_{23} \\ M(\theta)_{31} & M(\theta)_{32} & M(\theta)_{33} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{pmatrix} + \begin{pmatrix} C(\theta, \dot{\theta})_1 \\ C(\theta, \dot{\theta})_2 \\ C(\theta, \dot{\theta})_3 \end{pmatrix} \\ + \begin{pmatrix} G(\theta)_1 \\ G(\theta)_2 \\ G(\theta)_3 \end{pmatrix} + \begin{pmatrix} T_{ext}(\theta)_1 \\ T_{ext}(\theta)_2 \\ T_{ext}(\theta)_3 \end{pmatrix} \quad (12)$$

Where,

τ is the vector of joint moment.

$M(\theta)$ is the inertia matrix. (It is symmetric, so only six elements need to be de fined.)

$\ddot{\theta}$ is the vector of angular acceleration.

$C(\theta, \dot{\theta})$ is the vector of centrifugal/Coriolis terms.

$G(\theta)$ is the vector of gravity terms.

T_{ext} is the vector of joint moments because of other external forces besides gravity.

and:

$$M(\theta)_{11} = m_1 d_1^2 + I_1 + m_2 (L_1^2 + d_2^2 + 2L_1 d_2 \cos \theta_2) + I_2 \\ + m_3 [L_1^2 + L_2^2 + d_3^2 + 2L_1 L_2 \cos \theta_2 + 2L_1 d_3 \cos(\theta_2 + \theta_3) \\ + 2L_2 d_3 \cos \theta_3] + I_3$$

$$M(\theta)_{12} = m_2 (d_2^2 + L_1 d_2 \cos \theta_2) + I_2 + m_3 [L_2^2 + d_3^2 \\ + L_1 L_2 \cos \theta_2 + L_1 d_3 \cos(\theta_2 + \theta_3) + 2L_2 d_3 \cos \theta_3] + I_3$$

$$M(\theta)_{13} = m_3 [d_3^2 + L_1 d_3 \cos(\theta_2 + \theta_3) + L_2 d_3 \cos \theta_3] + I_3$$

$$M(\theta)_{21} = m_2 (d_2^2 + L_1 d_2 \cos \theta_2) + I_2 + m_3 [L_2^2 + d_3^2 \\ + L_1 L_2 \cos \theta_2 + 2L_2 d_3 \cos \theta_3 + L_1 d_3 \cos(\theta_2 + \theta_3)] + I_3$$

$$M(\theta)_{22} = m_2 d_2^2 + I_2 + m_3 (L_2^2 + d_3^2 + L_2 d_3 \cos \theta_3) + I_3$$

$$M(\theta)_{23} = m_3 (d_3^2 + L_2 d_3 \cos \theta_3) + I_3$$

$$M(\theta)_{31} = m_3 [d_3^2 + L_1 d_3 \cos(\theta_2 + \theta_3) + L_2 d_3 \cos \theta_3] + I_3$$

$$M(\theta)_{32} = m_3 (d_3^2 + L_2 d_3 \cos \theta_3) + I_3$$

$$M(\theta)_{33} = m_3 d_3^2 + I_3$$

$$C(\theta, \dot{\theta})_1 = -[(m_2 L_1 d_2 + m_3 L_1 L_2) \sin \theta_2 + m_3 L_1 d_3 \sin(\theta_2 + \theta_3)] \\ (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - [m_3 L_1 d_3 \sin(\theta_2 + \theta_3) + m_3 L_2 d_3 \sin \theta_3] \\ (2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2)$$

$$C(\theta, \dot{\theta})_2 = [(m_3 L_1 L_2 + m_2 d_2 L_1) \sin \theta_2 + m_3 d_3 L_1 \sin(\theta_2 + \theta_3)] \dot{\theta}_1^2 \\ - m_3 d_3 L_2 \sin \theta_3 (2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2)$$

$$C(\theta, \dot{\theta})_3 = [m_3 L_1 d_3 \sin(\theta_2 + \theta_3) + m_3 L_2 d_3 \sin \theta_3] \dot{\theta}_1^2 \\ m_3 L_2 d_3 \sin \theta_3 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2)$$

Where, $L_{1-3}, m_{1-3}, I_{1-3}$ are the links' lengths, masses and inertias respectively. Moreover, d_{1-3} are distances from links' center of mass to the proximal joints. Physical parameters of three-link arm model are expressed in table 1.

Table 1. Physical parameters for three links.

Link	L(m)	m (kg)	I(kg m ²)
Link 1	0.31	1.93	0.0141
Link 2	0.27	1.32	0.0120
Link 3	0.15	0.35	0.001

In this three-link manipulator, we do not have any external torque and we neglect the gravity term in eq. (12).

Substituting joint moments in eq. (12) and using ODE45 solver, states of the system (three joint angles and three joint angular velocity) and six muscle forces have been obtained. The Controller's coefficients for calculation of muscle forces and joint moments (we need joint moments to substitute in eq. (12)) have been determined via trial and error.

3. RESULTS

Hand (end-effector) trajectory could be drawn by means of the joint angles calculated via ODE45 in the previous section. Hand trajectory compared to the desired path are depicted in Fig.2.

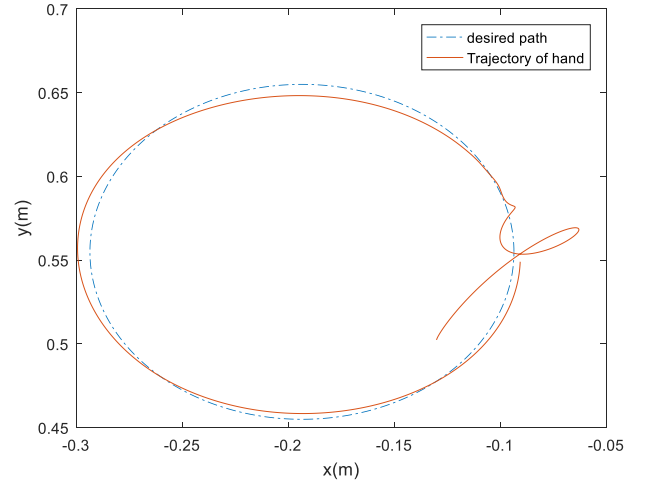
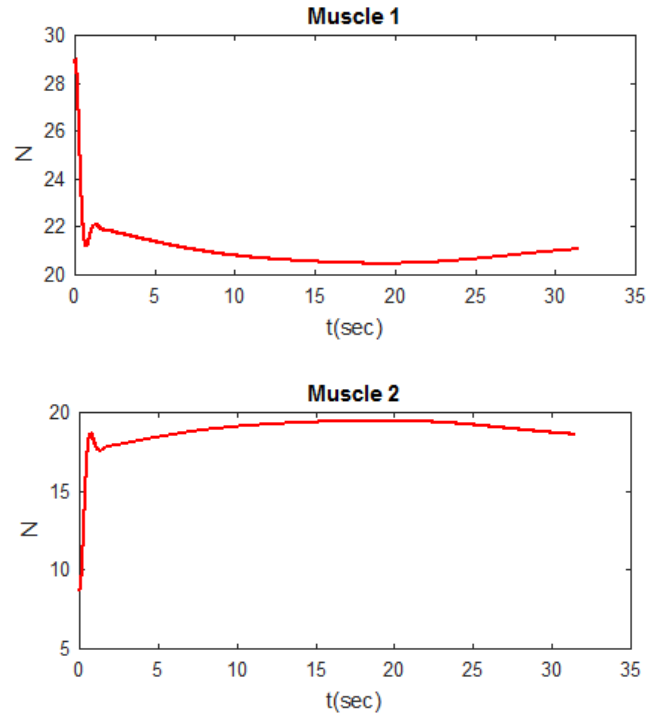


Figure 2. Hand trajectory compared to desired path

Calculated forces for six muscle versus time are presented in figure 3. Clearly, all of the forces are positive and sensible.



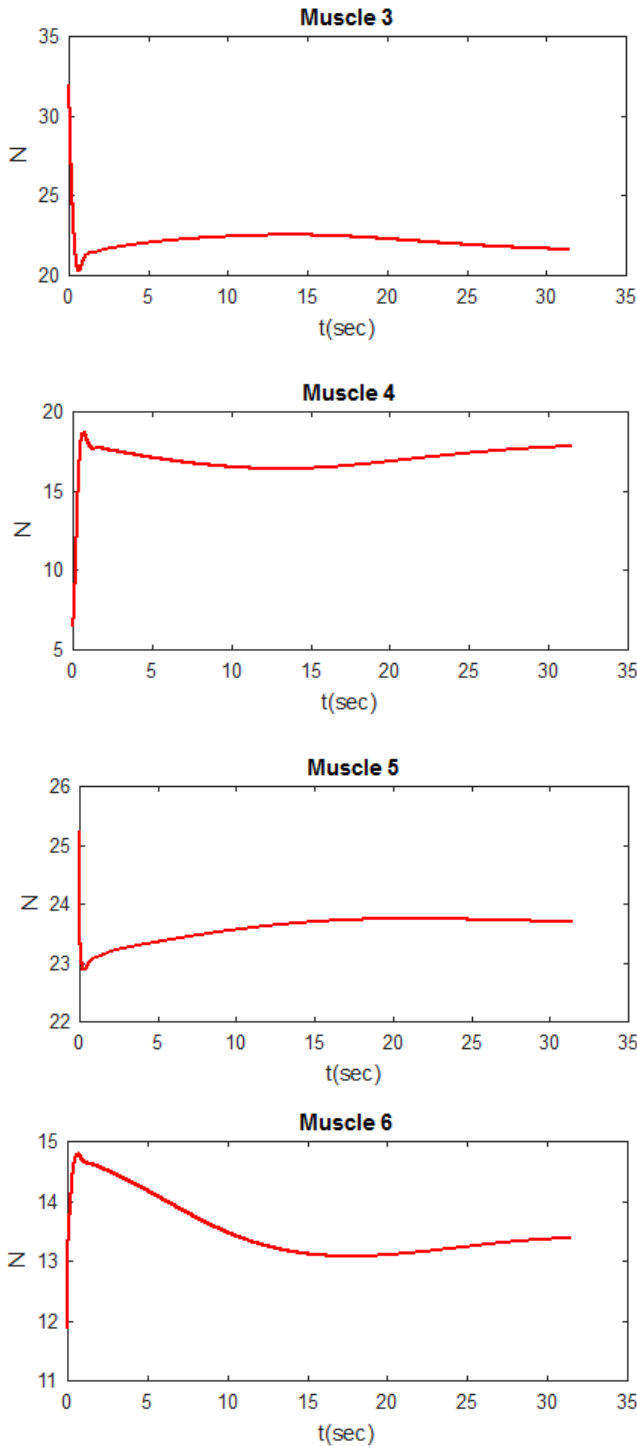


Figure 3. Six muscle force trajectories during desired path tracking

4. DISCUSSION

In this study human arm modeled with a three-link planar mechanism powered by Hill muscle model. Employing controller, values of force and then moments applied to the joints calculated and finally, solving dynamics differential equations, joint trajectory and muscle forces was calculated and drawn.

5. CONCLUSION AND FUTURE WORK

Suggested future work could be application of optimal controller and making comparison between results of current controller and the ones from optimal controllers.

ACKNOWLEDGMENTS

We thank Professor Pranav Bhounsule for assistance with methodology and comments that greatly improved the manuscript.

REFERENCES

- [1] Tahara K., Arimoto S., Kino H., “Sensory-Motor Control Mechanism for Reaching Movements of a Redundant Musculo-Skeletal Arm”, Journal of Field Robotics, Vol. 22, Issue 11, p 639-651, Nov. 2005.
- [2] “Cesari, P., Shiratori, T., Olivato, P., Duarte, M., (2002), “Zatsiorsky, VM (2002) Kinetics of Human Motion. Champaign, Human Kinetics”, “Inverse dynamics for a three-link planar chain”.

APPENDIX

MATLAB Code:

```

Main:
clc;
clear all;
close all
T_time=10*pi; %% total time of simulation
init_cond=[pi/3 pi/3 pi/3]; %%initial angles if shoulder,
elbow and wrist
options = odeset('RelTol',1e-4);
[t,state] = ode45(@odes,[0 T_time],[init_cond(1)
init_cond(2) init_cond(3)...
0 0 0 0 0 0 0],options);

%%%%%%%%%% plot trajectory and deodesired path
%=====
figure(1)

T=0:0.1:T_time;
Xd=-
0.1935+0.10*cos(T*0.2);Yd=0.5549+0.10*sin(T*0.2);%%
desired path
L1=0.31;L2=0.27;L3=0.15;
x3=L1*cos(state(:,1))+L2*cos(state(:,1)+state(:,2))+L3*cos
(state(:,1)+state(:,2)+state(:,3));

```

```

y3=L1*sin(state(:,1))+L2*sin(state(:,1)+state(:,2))+L3*sin(
state(:,1)+state(:,2)+state(:,3));
plot(Xd,Yd,'-',x3,y3)
legend('desired path','Trajectory of hand')
xlabel('x(m)')
ylabel('y(m)')
hold off;
%%%%%%%%%% animation play
%=====
r1=0.055;r2=0.055;
r3=0.03;r4=0.03;
r5=0.035;r6=0.050;
s1=0.08;s2=0.08;
s3=0.12;s4=0.12;
s5=0.22;s6=0.25;
w1=0.04;w2=0.05;
for i=1:35:length(t)
    figure(2)
    L1=0.31;L2=0.27;L3=0.15;
    x0=0;y0=0; %%%shoulder position
    %%% elbow position
    x1=L1*cos(state(i,1));x11=L1*cos(state(i,1))-
w1*cos(state(i,1)+state(i,2));
    y1=L1*sin(state(i,1));y11=L1*sin(state(i,1))-
w1*sin(state(i,1)+state(i,2));
    %%% wrist position
    x2=L1*cos(state(i,1))+L2*cos(state(i,1)+state(i,2));
    x22=L1*cos(state(i,1))+L2*cos(state(i,1)+state(i,2))-
w2*cos(state(i,1)+state(i,2)+state(i,3));
    y22=L1*sin(state(i,1))+L2*sin(state(i,1)+state(i,2))-
w2*sin(state(i,1)+state(i,2)+state(i,3));
    y2=L1*sin(state(i,1))+L2*sin(state(i,1)+state(i,2));
    %%% hand position

x3=L1*cos(state(i,1))+L2*cos(state(i,1)+state(i,2))+L3*cos(sta
te(i,1)+state(i,2)+state(i,3));

y3=L1*sin(state(i,1))+L2*sin(state(i,1)+state(i,2))+L3*sin(sta
te(i,1)+state(i,2)+state(i,3));

    %%% desired path
    T=0:0.1:T_time;
    Xd=-
0.1935+0.10*cos(T*0.2);Yd=0.5549+0.10*sin(T*0.2);
    plot(Xd,Yd,'-')

    hold on

plot([x0,x1,x11,x2,x22,x3],[y0,y1,y11,y2,y22,y3],'b','LineWidt
h',4);
    circle([x0,y0],0.01)
    hold on
    circle([x1,y1],0.01)
    hold on
    circle([x2,y2],0.01)
    hold on

```

```

circle([x3,y3],0.005)
hold on

%%muscle 1
M1_r=[-
r1,0];M1_s=[s1*cos(state(i,1)),s1*sin(state(i,1))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;
%%muscle 2

M1_r=[r2,0];M1_s=[s2*cos(state(i,1)),s2*sin(state(i,1))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;
%%muscle 3
M1_r=[(L1-s3)*cos(state(i,1)),(L1-s3)*sin(state(i,1))];
M1_s=[L1*cos(state(i,1))+r3*cos(state(i,1)+state(i,2)),...
L1*sin(state(i,1))+r3*sin(state(i,1)+state(i,2))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;
%%muscle 4
M1_r=[(L1-s4)*cos(state(i,1)),(L1-s4)*sin(state(i,1))];
M1_s=[x1-r4*cos(state(i,1)+state(i,2)),...
y1-r4*sin(state(i,1)+state(i,2))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;
%%muscle 5
M1_r=[x1+(L2-s5)*cos(state(i,1)+state(i,2)),...
y1+(L2-s5)*sin(state(i,1)+state(i,2))];
M1_s=[x2+r5*cos(state(i,1)+state(i,2)+state(i,3)),...
y2+r5*sin(state(i,1)+state(i,2)+state(i,3))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;
%%muscle 6
M1_r=[x1+(L2-s6)*cos(state(i,1)+state(i,2)),...
y1+(L2-s6)*sin(state(i,1)+state(i,2))];
M1_s=[x2-r6*cos(state(i,1)+state(i,2)+state(i,3)),...
y2-r6*sin(state(i,1)+state(i,2)+state(i,3))];
plot([M1_r(1),M1_s(1)],[M1_r(2),M1_s(2)],'r')
hold on;

xlim([-0.4,0.3])
ylim([-0.05,0.8])
xlabel('x(m)')
ylabel('y(m)')
pause(0.05)
hold off
end
%%%%%%%%%% muscle forces plot
%=====
figure(3)
subplot(3,2,1)
plot(t,state(:,7)./t,'r','LineWidth',1.5);
xlabel('t(sec)');ylabel('N')
title('Muscle 1')
subplot(3,2,2)
plot(t,state(:,8)./t,'r','LineWidth',1.5);

```

```

xlabel('t(sec)');ylabel('N')
title('Muscle 2')
subplot(3,2,3)
plot(t,state(:,9)./t,'r','LineWidth',1.5);
xlabel('t(sec)');ylabel('N')
title('Muscle 3')
subplot(3,2,4)
plot(t,state(:,10)./t,'r','LineWidth',1.5);
xlabel('t(sec)');ylabel('N')
title('Muscle 4')
subplot(3,2,5)
plot(t,state(:,11)./t,'r','LineWidth',1.5);
xlabel('t(sec)');ylabel('N')
title('Muscle 5')
subplot(3,2,6)
plot(t,state(:,12)./t,'r','LineWidth',1.5);
xlabel('t(sec)');ylabel('N')
title('Muscle 6')

```

ODEs:

```

function dx=odes(t,x)
% parameters of hand
%=====
L1=0.31;L2=0.27;L3=0.15;
I1=0.0141;I2=0.0120;I3=0.001;
m1=1.93;m2=1.32;m3=0.35;
d1=0.165;d2=0.135;d3=0.075;
% Insertion of each muscle
%=====
r1=0.055;r2=0.055;
r3=0.03;r4=0.03;
r5=0.035;r6=0.050;
s1=0.08;s2=0.08;
s3=0.12;s4=0.12;
s5=0.22;s6=0.25;
% length of each muscle
%=====
l1=(r1^2+s1^2+2*r1*s1.*cos(x(1))).^(0.5);
l2=(r2^2+s2^2+2*r2*s2.*cos(x(1))).^(0.5);
l3=(r3^2+s3^2+2*r3*s3.*cos(x(2))).^(0.5);
l4=(r4^2+s4^2+2*r4*s4.*cos(x(2))).^(0.5);
l5=(r5^2+s5^2+2*r5*s5.*cos(x(3))).^(0.5);
l6=(r6^2+s6^2+2*r6*s6.*cos(x(3))).^(0.5);
l=[l1 l2 l3 l4 l5 l6];
% Jacobian matrix from the joint space to the muscle
space
%=====
q11=-
r1*s1.*sin(x(1))./(r1^2+s1^2+2*r1*s1.*cos(x(1))).^(0.5);
q12=r2*s2.*sin(x(1))./(r2^2+s2^2+
2*r2*s2.*cos(x(1))).^(0.5);
q23=-
r3*s3.*sin(x(2))./(r3^2+s3^2+2*r3*s3.*cos(x(2))).^(0.5);
q24=r4*s4.*sin(x(2))./(r4^2+s4^2+
2*r4*s4.*cos(x(2))).^(0.5);

```

```

q35=-
r5*s5.*sin(x(2))./(r5^2+s5^2+2*r5*s5.*cos(x(3))).^(0.5);
q36=r6*s6.*sin(x(2))./(r6^2+s6^2+
2*r6*s6.*cos(x(3))).^(0.5);
W=[q11,q12,0,0,0,0;
0,0,q23,q24,0,0;
0,0,0,q35,q36];
Wplus=W/(W*W');

%%% Jacobian matrix from the task space to the joint space
%=====
J(1,1)=-L1.*sin(x(1))-L2.*sin(x(1)+x(2))-
L3.*sin(x(1)+x(2)+x(3));
J(1,2)=-L2.*sin(x(1)+x(2))-L3.*sin(x(1)+x(2)+x(3));
J(1,3)=-L3.*sin(x(1)+x(2)+x(3));
J(2,1)=L1.*cos(x(1))+L2.*cos(x(1)+x(2))+L3.*cos(x(1)+x(2)
)+x(3));
J(2,2)=L2.*cos(x(1)+x(2))+L3.*cos(x(1)+x(2)+x(3));
J(2,3)=L3.*cos(x(1)+x(2)+x(3));

```

```

% desired path and controller parameters*
%
%=====*
Xd=-0.1935+0.10*cos(t*0.2);Yd=0.5549+0.10*sin(t*0.2);
Kp=25*diag(ones(1,2));Kc=20*ones(6,1);

% controller configuration on hill muscle model and end
effector position
%=====
X=L1.*cos(x(1))+L2.*cos(x(1)+x(2))+L3.*cos(x(1)+x(2)+x(
3));
Y=L1.*sin(x(1))+L2.*sin(x(1)+x(2))+L3.*sin(x(1)+x(2)+x(3
));
alpha=-Wplus*J'*Kp*[X-Xd;Y-Yd]+(eye(6,6)-
Wplus*W)*Kc;
alpha(alpha<0)=0;
A=diag(alpha);
U=ones(1,6);
B=diag(ones(1,6)*10);
B0=diag(ones(1,6)*10);
F=A*U-(A*B+B0)*W'*[x(4);x(5);x(6)];
T(1:3)=W*F;
% dynamics equation+ ode's
%=====
M(1,1)=m1*d1^2+I1+m2*(L1^2+d2^2+2*L1*d2.*cos(x(2
)))+I2+...
m3*(L1^2+L2^2+d3^2+2*L1*L2.*cos(x(2))+2*L1*d3*cos(x(
2)+x(3))+2*L2*d3*cos(x(3)))+I3;
M(1,2)=m2*(d2^2+L1*d2*cos(x(2)))+I2+m3*(L2^2+d3^2
+L1*L2*cos(x(2))...
+L1*d3*cos(x(2)+x(3))+2*L2*d3*cos(x(3)))+I3;

```

```

M(1,3)=m3*(d3^2+L1*d3*cos(x(2)+x(3))+L2*d3*cos(x(3
))) + I3;
M(2,1)=m2*(d2^2+L1*d2*cos(x(2)))+I2...

+m3*(L2^2+d3^2+L1*L2*cos(x(2))+2*L2*d3*cos(x(3))+L1*
d3*cos(x(2)+x(3)))+I3;
M(2,2)=m2*d2^2+I2+m3*(L2^2+d3^2+L2*d3*cos(x(3)))
+I3;
M(2,3)=m3*(d3^2+L2*d3*cos(x(3)))+I3;
M(3,1)=m3*(d3^2+L1*d3*cos(x(2)+x(3))+L2*d3*cos(x(3
))) + I3;
M(3,2)=m3*(d3^2+L2*d3*cos(x(3)))+I3;
M(3,3)=m3*d3^2+I3;
C(1)=-
((m2*L1*d2+m3*L1*L2)*sin(x(2))+m3*L1*d3*sin(x(2)+x(3))
)*(2*x(4)*x(5)+x(5)^2)...
-
(m3*L1*d3*sin(x(2)+x(3))+m3*L2*d3*sin(x(3)))*(2*x(4)*x(6
)+2*x(5)*x(6)+x(6)^2);
C(2)=((m3*L1*L2+m2*d2*L1)*sin(x(2))+m3*d3*L1*sin(
x(2)+x(3)))*x(4)^2 ...
-
m3*d3*L2*sin(x(3))*(2*x(4)*x(6)+2*x(5)*x(6)+x(6)^2);
C(3)=(m3*L1*d3*sin(x(3)+x(4))+m3*L2*d3*sin(x(3)))*x
(4)^2 ...
+m3*L2*d3*sin(x(3))*(2*x(4)*x(5)+x(5)^2);

dx(1)=x(4);
dx(2)=x(5);
dx(3)=x(6);
dx(4:6)=M\'(T'-C');
dx(7:12)=F;
dx=dx';
end

```

Circle:

```

function circle(x,r)
% x and y are the coordinates of the center of the circle
% r is the radius of the circle
% 0.01 is the angle step, bigger values will draw the circle
faster but
% you might notice imperfections (not very smooth)
ang=0:0.01:2*pi;
xp=r*cos(ang);
yp=r*sin(ang);
plot(x(1)+xp,x(2)+yp,'k','LineWidth',2);
fill(x(1)+xp,x(2)+yp,'k')
end

```