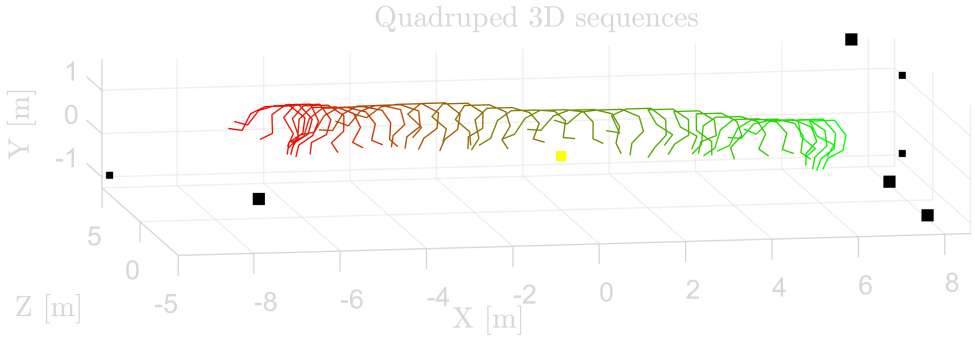
```
%%File: CV-CODE-Project-R2PT-ANIMATION-3link-Quadruped-EXP-2.mlx    :      Robot Arm (2-link) T
% Trajectory Generation of the Equivalent End Effector _EE for a 2-link robot arm
% Source location: QRIS> C:\Users\USER-PC\QRIS\MATLAB Code
% ------------------------------------------------------------------------
% Notes:
% 1. Animation of original 3-link leg quadruped
% 2. Analysis of the configuration changes of the leg that is in contact with the ground level
% 3. Trajectories of J0 & J1 as generated by the front & back legs respectively
% 4. Item (3) above contributes to the forward locomotion of the quadruped
% ------------------------------------------------------------------------------------------------
% ------------------------------------------------------------------------------------------------
clc;clf ; clear all;
clear global
% ------------------------------------------------------------------------
% CODE execution START:
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% C:\Users\USER-PC\QRIS\MATLAB Code\DATA   <---- DATA Folder to access
RXYZ3D=[];
MatrixDATA = dlmread('C:\Users\USER-PC\QRIS\MATLAB Code\DATA\PythonDATA-XYZdataLSQ-3.txt');
RXYZ3D = MatrixDATA;% LSQ-optimised 3D XYZ data [m] <---> [90 x 9] <-- full Quadruped
[rws,cms] = size(MatrixDATA) ;
% ------------------------------------
% [RXYZ3D] = [90x9]--------------->[BL_EE---->Spine------>FL_EE]
% joint order -------------------->[#8 #7 #6 CoM #9 #1 #2 #3 #4]
% order # ----------------------->[ 1  2  3  4  5  6  7  8   9]
global maxConfigs
maxConfigs = 30; % 30 quadruped configurations : XYZ ==> 3 coordinate axes per configuration ==
% --------------------
cnfg=1;
RXYZ=RXYZ3D;
for k=1:maxConfigs
RobotXYZx(cnfg,:) = RXYZ((cnfg*3)-2,:) ;
RobotXYZy(cnfg,:) = RXYZ((cnfg*3)-1,:);
RobotXYZz(cnfg,:) = -1*RXYZ((cnfg*3),:) ;
%RobotXYZz(cnfg,:) = 1*RXYZ((cnfg*3),:) ;
% -------------------------
% joint order = image data order-->[#8 #7 #6 CoM #9 #1 #2 #3 #4]
% order # ----------------------->[ 1  2  3  4  5  6  7  8   9]
% -------------------------
COMx(k) = RXYZ((cnfg*3)-2,4) ;
COMy(k) = RXYZ((cnfg*3)-1,4) ;
COMz(k) = RXYZ((cnfg*3),4) ;
cnfg=cnfg+1;
end
% ------------------------------------
% color CODE vectors ---->
cW = [1 1 1]; % white
cA = [1 0 0]; % red
cB = [0 0 1]; % blue
cC = [0 1 1]; % cyan
cD = [1 0 1]; % magenta
```

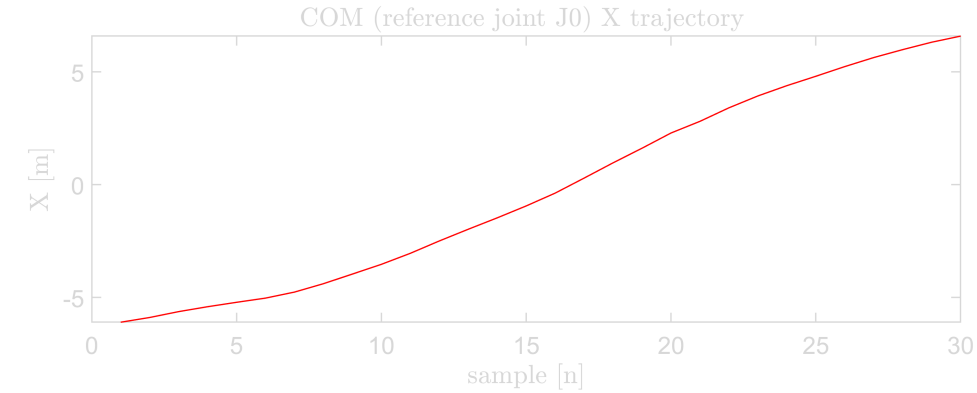
```
cE = [1 1 0]; % yellow
cJ = [0 1 0]; % green
cK = [0 0 0]; % black
% --- hybrids ----------
cF = [0.75 0 0.99]; % purple
cG = [0 0.4 0.3]; % dark green
cH = [0.6 0.98 0]; % light green
cI = [0.99 0.5 0]; % orange
% ------------------------------------------------------------------------------------------
% ------------------------------------------------------------------------------------------
%                        ANIMATION SAVED in File
% ------------------------------------------------------------------------------------------
%                         EXPANSIVE VIEW
% ---------------------------------------------------------------------
% ANIMATION of XY configuration estimates:
%vVideo = VideoWriter('C:\Users\Casey\Matlab2018\Videos\XYZVideoQ8.avi'); % initialise file to
%open(vVideo);
% ---------------------------------------------------------------------
% --------------------------
Naughts = zeros(1,numel(RobotXYZx));
Ones = ones(1,numel(RobotXYZx));
w=1;
cnfg=1;
figure
%view(-32,17) %
view(-62,10) %
% ----------------------------
% set background colour
fig = gcf;
fig.Color = [0 0 0]; % black = [0 0 0]
colordef black
% ----------------------------
hold on
for k=1:30
% ------------------------------------------------------------
% plot camera position
scatter3(0,0,0,26,'yellow','filled','s'); % XZY coordinates order
% ------------------------------------------------------------
% ensure fixed width X axis for comparative frames
scatter3(-7,0,-0.8,'k','filled','s'); % XZY coordinates order
scatter3(7.6,0,-0.8,'k','filled','s'); % XZY coordinates order
scatter3(7.6,-5,-0.8,'k','filled','s'); % XZY coordinates order
scatter3(7.6,5,1.7,'k','filled','s'); % XZY coordinates order
% ------------------------------------------------------------
% look FAR away from sequence
%scatter3(7.6,-15,1.7,'k','filled','s'); % XZY coordinates order
% ------------------------------------------------------------
%IRm = RXYZ((cnfg*3)-2:(cnfg*3),:);
IRm = [RobotXYZx(cnfg,:);RobotXYZy(cnfg,:);RobotXYZz(cnfg,:)];
%IRm = IRm(1:3,:);
IRm(2,:) = -1*IRm(2,:); % flip Y axis for correct up orientation
% joint order = image data order-->[#8 #7 #6 CoM #9 #1 #2 #3 #4]
% order # ----------------------->[ 1  2  3  4  5  6  7  8   9]
```

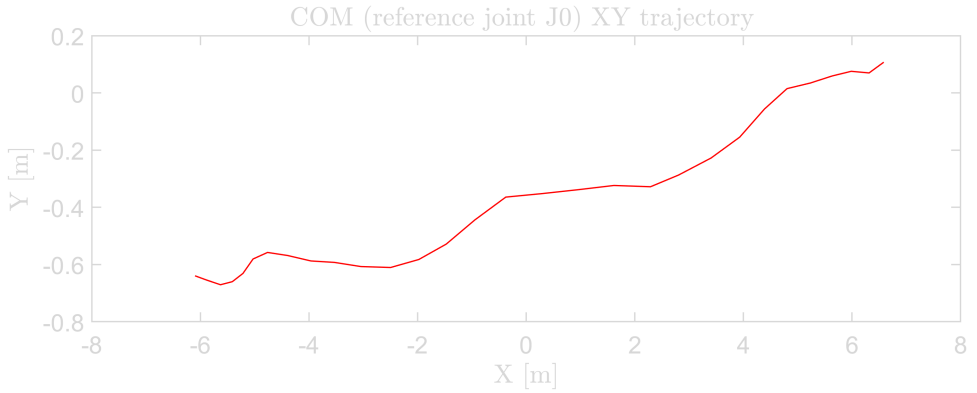
```
%figure;
%colorRED ----->[1 0 0]
crR = linspace(1,0,30);
%colorGREEN --->[0 1 0]
crG = linspace(0,1,30);
%'Color',[crR(p) crG(p) 0]
% ------------------------------------------------------------
% PLOT robot configuration: ACTIVATE next 1 line ------>
%line(IRm(w,:),IRm(w+2,:),IRm(w+1,:), 'Color','r','LineStyle','-'); % LSQ (small wide screen)
%line(IRm(w,:),IRm(w+2,:),IRm(w+1,:), 'Color',[crR(k) crG(k) 0],'LineStyle','-'); % LSQ (small
line(IRm(w,:),-1*IRm(w+2,:),IRm(w+1,:), 'Color',[crR(k) crG(k) 0],'LineStyle','-'); % LSQ (sma
%adjstVEC = ones(1,30);
%line(IRm(w,:),[-8*adjstVEC -1*IRm(w+2,:)],IRm(w+1,:), 'Color',[crR(k) crG(k) 0],'LineStyle','

%view(3);
% ------------------------------------------------------------
%hold on
% PLOT the markers ---------------------------->
for i=1:9 % whole body
%plot3(IRm(w,i),IRm(w+2,i),IRm(w+1,i),'r','Marker','+');
end
% PLOT the markers ---------------------------->
%plot3(IRm(w,4),IRm(w+2,4),IRm(w+1,4),'b','Marker','+'); % reference marker = CoM LSQ
%plot3(IRm(w,6),IRm(w+2,6),IRm(w+1,6),'c','Marker','+'); % reference marker = CoM2 LSQ8
% ------------------------------------------------------------
% plot camera position
%scatter3(0,4,-0.4,26,'y','filled','s'); % XZY coordinates order
% ------------------------------------------------------------
% ensure fixed width X axis for comparative frames
scatter3(-9.75,4,-0.8,12,'k','filled','s'); % XZY coordinates order
scatter3(8.6,4,-0.8,12,'k','filled','s'); % XZY coordinates order
scatter3(8.6,4,1.0,12,'k','filled','s'); % XZY coordinates order
% ------------------------------------------------------------
% Elevate plot from base plane :
%scatter3(8.6,4,-4.0,12,'k','filled','s'); % XZY coordinates order
% ------------------------------------------------------------------------------
% Plot COM points on each plane: % <-----ACTIVATE to also plot reference joint positions
%scatter3(COMx(cnfg),0,-1*COMy(cnfg),'green','filled','s');% X0Y plot COM
%scatter3(COMx(cnfg),COMz(cnfg),0,'red','filled','s');% XZ0 plot COM
nX=10;% ZY plane located at X=nX
%scatter3(nX,COMz(cnfg),-1*COMy(cnfg),'cyan','filled','s');% 0ZY plot COM
% ------------------------------------------------------------------------------
% Trajectory plots projected on all 3 planes:
ZeroDIM=[];
ZeroDIM = Naughts(1:cnfg);
OneDIM=[];
OneDIM = Ones(1:cnfg);
% XY plane trajectory : plot order = [XZY]
%plot3(RobotXYZx(1:cnfg,4),ZeroDIM(1:cnfg),-1*RobotXYZy(1:cnfg,4),'green') %<---ACTIVATE to pl
% XZ plane trajectory : plot order = [XZY]
%plot3(RobotXYZx(1:cnfg,4),RobotXYZz(1:cnfg,4),ZeroDIM(1:cnfg),'red') %<---ACTIVATE to plot re
% ZY plane trajectory : plot order = [XZY]
nX=10;
```

```
%plot3(nX*OneDIM(1:cnfg),RobotXYZz(1:cnfg,4),-1*RobotXYZy(1:cnfg,4),'cyan') %<---ACTIVATE to pl
% ------------------------------------------------------------
xlabel('X [m]', 'Interpreter','latex')
ylabel('Z [m]', 'Interpreter','latex')
zlabel('Y [m]', 'Interpreter','latex')
%title('3D view: moving CoM')
title('Quadruped 3D sequences', 'Interpreter','latex')
%hold off
%axis ij
axis equal
grid on
% -----------------------------
% set background colour
fig = gcf;
fig.Color = [0 0 0]; % black = [0 0 0]
colordef black
% -----------------------------
%axis off
%frame = getframe(gcf); % get frame to SAVE to file
%writeVideo(vVideo,frame); % SAVE this frame to video for kth video frame
cnfg=cnfg+1;
end
% --------------------------
%view(3);% sets the default three-dimensional view, az = -37.5, el = 30.
%view(0,90) ;%view directly above
%view(10,9); % slightly to the right and almost level
%view(0,9); % centered and almost level
view(-10,9); % slightly to the left and almost level
%view(-10,3); % slightly to the left and almost very level
%view(-10,19); % slightly to the left and looking from higher up

hold off
```



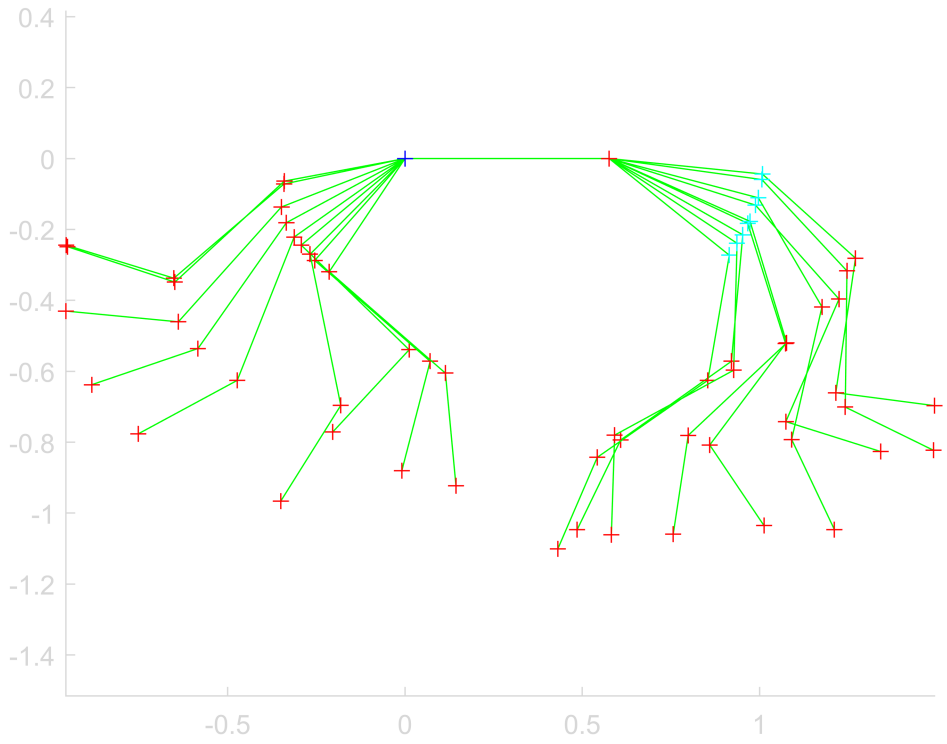

```
%close(vVideo);
%clearvars vVideo frame
% --------------------------------
figure
% ----------------------------
% set background colour
fig = gcf;
fig.Color = [0 0 0]; % black = [0 0 0]
colordef black
% ----------------------------
subplot(2,1,1),plot(COMx,'Color',cA)
title('COM (reference joint J0) X trajectory ', 'Interpreter','latex')
xlabel('sample [n]', 'Interpreter','latex')
ylabel('X [m]', 'Interpreter','latex')

subplot(2,1,2),plot(COMx,COMy,'Color',cA)
title('COM (reference joint J0) XY trajectory ', 'Interpreter','latex')
xlabel('X [m]', 'Interpreter','latex')
ylabel('Y [m]', 'Interpreter','latex')
```



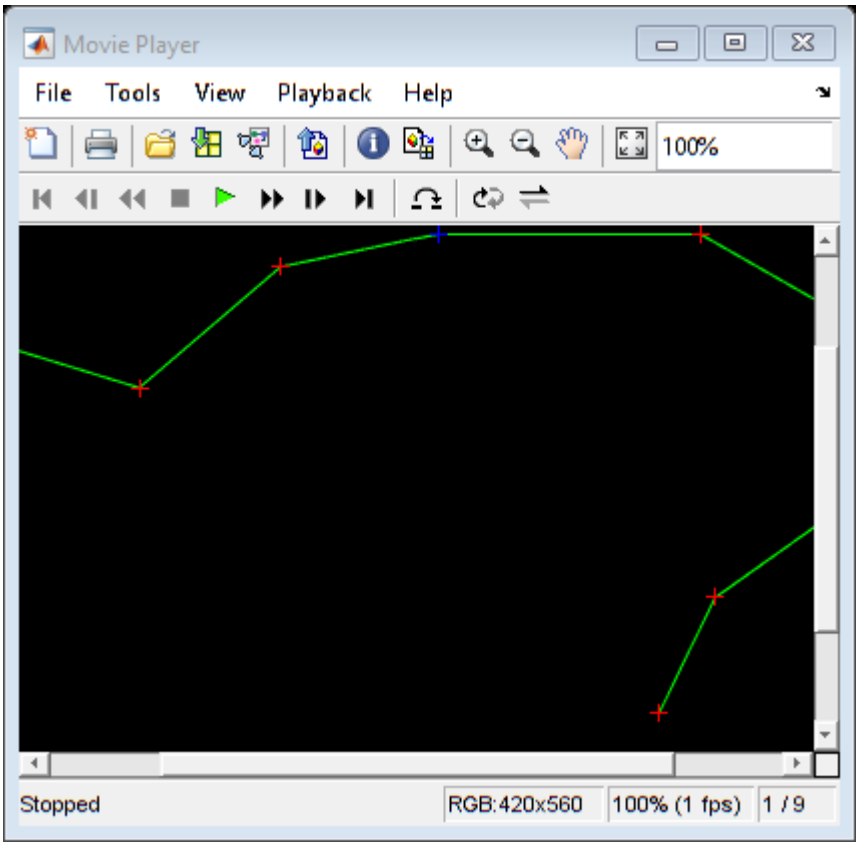



```
% ------------------------------
% ---------------------------------
figure
% ------------------------------
% set background colour
fig = gcf;
fig.Color = [0 0 0]; % black = [0 0 0]
colordef black
% ------------------------------
cnfg = 11; % configuration number out of 30 : gait 2 assumed range -->[11,20]
w=1;
% ----------- set up ANIMATION ---------------------->[]
loops = 1;
movColr(1:loops) = struct('cdata', [],'colormap',[]);
i=1;
% ------------------------------------------------------
for k = 1:9
%IRm = [RobotXYZx(cnfg,:);RobotXYZy(cnfg,:);RobotXYZz(cnfg,:)];
IRm = [RobotXYZx(cnfg,:);RobotXYZy(cnfg,:)];
%IRm = IRm(1:3,:);
IRm(2,:) = -1*IRm(2,:); % flip Y axis for correct up orientation
IRmZEROx = IRm(w,:) - COMx(cnfg)*ones(1,9);
IRmZEROy = IRm(w+1,:) + COMy(cnfg)*ones(1,9);
%line(IRm(w,:),IRm(w+1,:), 'Color',cJ,'LineStyle','-'); % LSQ (small wide screen)
%line(IRmZEROx,IRm(w+1,:), 'Color',cJ,'LineStyle','-'); % LSQ (small wide screen)
line(IRmZEROx,IRmZEROy, 'Color',cJ,'LineStyle','-'); % LSQ (small wide screen)
```

```
hold on
plot(IRmZEROx(1),IRmZEROy(1),'r','Marker','+') % <---- joint 8 marker
plot(IRmZEROx(2),IRmZEROy(2),'r','Marker','+') % <---- joint 7 marker
plot(IRmZEROx(3),IRmZEROy(3),'r','Marker','+') % <---- joint 6 marker
plot(IRmZEROx(4),IRmZEROy(4),'b','Marker','+') % <---- joint COM marker
plot(IRmZEROx(5),IRmZEROy(5),'r','Marker','+') % <---- joint 9 marker
plot(IRmZEROx(6),IRmZEROy(6),'c','Marker','+') % <---- joint 1 marker
plot(IRmZEROx(7),IRmZEROy(7),'r','Marker','+') % <---- joint 2 marker
plot(IRmZEROx(8),IRmZEROy(8),'r','Marker','+') % <---- joint 3 marker
plot(IRmZEROx(9),IRmZEROy(9),'r','Marker','+') % <---- joint 4 marker
axis equal  % <----------------- SET axes equal for plot

cnfg = cnfg + 1;
movColr(i) = getframe(gcf); % <--- store the current frame
i = i + 1;
end
hold off
axis equal  % <----------------- SET axes equal for plot
```




```
fps = 1; % <--- shows a frame every 1 [s]
implay(movColr,fps); % specify the frames per second to show in the animation
```




```
clear movColr
```