Contents

Running Biomechanics Data Set (RBDS) analysis.

Reginaldo K Fukuchi, Mar 2017, reginaldo.fukuchi@ufabc.edu.br

```
% This supplemental material presents a script that exemplifies an
% exploration of the data set contained in the metadata file. In addition, the basic data analysis
% steps taken to calculate the discrete variables and generate the time series curves
% (e.g. angles, torques, powers and GRFs) presented in the manuscript are offered in this script.

% The data set is available at Figshare (DOI: <https://doi.org/10.6084/m9.figshare.4543435.v3>)

% Fukuchi RK, Fukuchi CA and Duarte M (2017). A public data set of running biomechanics
% and the effects of running speed on lower extremity kinematics and kinetics. PeerJ Preprints.

% In addition, it demonstrates plots of angles, moments, powers and ground
% reaction force curves displayed in the manuscript.

% Some of the steps have been reduced to minimize clutter, but the user
% should be able to adapt this code to any given file structure.
% clc, clear all, close all
```

Select the directory where the processed files are located

```
fileDir = uigetdir;
```

Determine what metadata file type to be imported as a table

```
if 0
    T = readtable([fileDir '\RBDSinfo.xlsx'],'FileType','spreadsheet',...
        'ReadVariableNames',true,'ReadRowNames',false,'Sheet','Planilha');
else
    T = readtable([fileDir '\RBDSinfo.txt'],'FileType','text',...
        'ReadVariableNames',true,'ReadRowNames',false);
end
% Removing repeated data
newT = T(1:12:end-11,:);
```

Doing summary statistics using all data

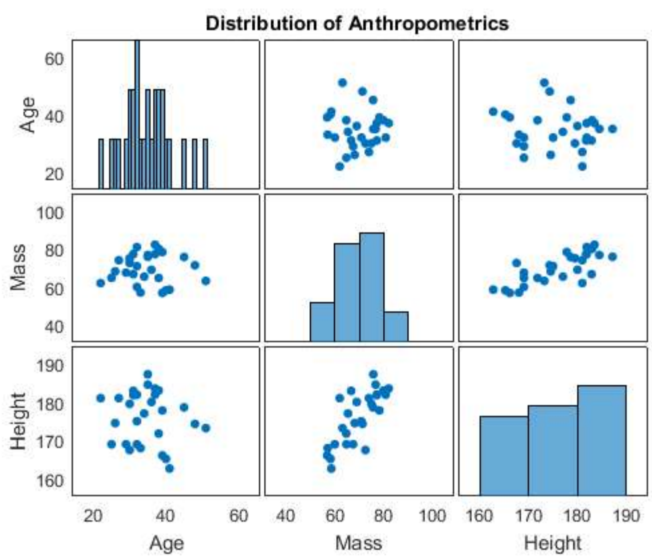
Anthropometric information minimum and maximum

```
anthroStats  = grpstats(newT,[],{'min','max','mean'},'DataVars',{'Age','Mass','Height'})
```

```
anthroStats = 

           GroupCount    min_Age    max_Age    mean_Age    min_Mass    max_Mass    mean_Mass    min_Height    max_Height    mean_Height
           __________    _______    _______    ________    ________    ________    _________    __________    __________    ___________

    All    28            22         51         34.75       56.85       82.15       69.638       162.7         187.2         175.96
```

Plotting the distribution of demographics

```
figure
[S,AX,BigAx,H,HAx] = plotmatrix([newT.Age,newT.Mass,newT.Height]);

title(BigAx,'Distribution of Anthropometrics')

% Axes lables
AX(1,1).YLabel.String = 'Age';
AX(2,1).YLabel.String = 'Mass';
AX(3,1).YLabel.String = 'Height';

AX(3,1).XLabel.String = 'Age';
AX(3,2).XLabel.String = 'Mass';
AX(3,3).XLabel.String = 'Height';
```



Traning habits information

```
runHabitsStats = grpstats(newT,[],{'mean','std'},'DataVars',{'Experience','SessionsPerWk','Pace'})
```

```
runHabitsStats = 

           GroupCount    mean_Experience    std_Experience    mean_SessionsPerWk    std_SessionsPerWk    mean_Pace    std_Pace
           __________    _______________    ______________    __________________    _________________    _________    ________

    All    28            101.5              84.258            3.7143                0.7127               4.0639       0.3771
```

Level of muscle strength

```
strengthStats = grpstats(newT,[],{'min','max'},'DataVars',{'RHIPABD','RHIPEXT','RHIPER','RHIPIR'})
```

```
strengthStats = 

           GroupCount    min_RHIPABD    max_RHIPABD    min_RHIPEXT    max_RHIPEXT    min_RHIPER    max_RHIPER    min_RHIPIR    max_RHIPIR
           __________    ___________    ___________    ___________    ___________    __________    __________    __________    __________

    All    28            16.8           45.767         14.333         33.95          7.7333        13.933        7.725         19.9
```

Level of flexibility

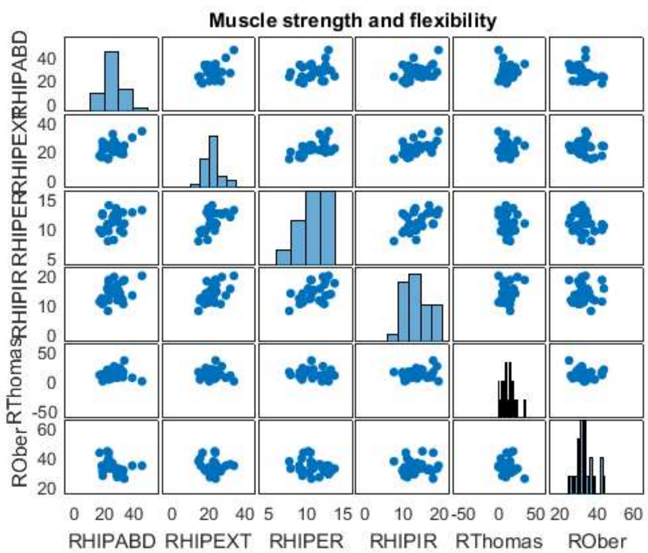
```
flexibStats    = grpstats(newT,[],{'mean','std'},'DataVars',{'RThomas','ROber'})
```

```
flexibStats = 
```

```
         GroupCount    mean_RThomas    std_RThomas    mean_ROber    std_ROber
         __________    ____________    ___________    __________    _________

    All      28           11.607         8.1529         33.179        4.8614
```

Distribution of strength and flexibility levels

```
figure
[S,AX,BigAx,H,HAx] = plotmatrix([newT.RHIPABD,newT.RHIPEXT,newT.RHIPER,...
    newT.RHIPIR,newT.RThomas,newT.ROber]);
title('Muscle strength and flexibility')
% Axes lables
AX(1,1).YLabel.String = 'RHIPABD';
AX(2,1).YLabel.String = 'RHIPEXT';
AX(3,1).YLabel.String = 'RHIPER';
AX(4,1).YLabel.String = 'RHIPIR';
AX(5,1).YLabel.String = 'RThomas';
AX(6,1).YLabel.String = 'ROber';

AX(6,1).XLabel.String = 'RHIPABD';
AX(6,2).XLabel.String = 'RHIPEXT';
AX(6,3).XLabel.String = 'RHIPER';
AX(6,4).XLabel.String = 'RHIPIR';
AX(6,5).XLabel.String = 'RThomas';
AX(6,6).XLabel.String = 'ROber';
```



Frequency of levels of performance

```
tabulate(newT.Level)
```

```
        Value    Count    Percent
  Competitive       25     89.29%
        Elite        3     10.71%
```

Frequency of training volume

```
tabulate(newT.Volume)
```

```
     Value    Count    Percent
  26-35 km        9     32.14%
  36-45 km        7     25.00%
   > 45 km        2      7.14%
    >45 km        9     32.14%
  16-25 km        1      3.57%
```

Frequency of foot strike type across running speeds

2.5 m/s

```
tabulate(newT.RFSI25)
```

```
     Value    Count    Percent
```

```
  Forefoot        5     17.86%
   Midfoot        4     14.29%
  Rearfoot       19     67.86%
```

3.5 m/s

```
tabulate(newT.RFSI35)
```

```
     Value    Count   Percent
  Forefoot        4     14.29%
  Rearfoot       19     67.86%
   Midfoot        5     17.86%
```

4.5 m/s

```
tabulate(newT.RFSI45)
```

```
     Value    Count   Percent
  Forefoot        5     17.86%
  Rearfoot       17     60.71%
   Midfoot        6     21.43%
```

Doing summary statistics organized by group

Comparing demographics between runners participating (Yes) or not (No) in running groups

```
anthroStats  = grpstats(newT,'RunGrp',{'mean','std'},'DataVars',{'Age','Mass','Height'})
```

```
anthroStats =

           RunGrp    GroupCount    mean_Age    std_Age    mean_Mass    std_Mass    mean_Height    std_Height
           ______    __________    ________    _______    _________    ________    ___________    __________

    Yes    'Yes'     17            32          5.3619     67.426       7.5682      174.15         6.7859
    No     'No'      11            39          6.4807     73.055       7.0011      178.76         6.0853
```

Comparing running habits between runners participating (Yes) or not (No) in running groups

```
runHabitsStats = grpstats(newT,'RunGrp',{'mean','std'},'DataVars',{'Experience','SessionsPerWk','Pace'})
```

```
runHabitsStats =

           RunGrp    GroupCount    mean_Experience    std_Experience    mean_SessionsPerWk    std_SessionsPerWk    mean_Pace    std_Pace
           ______    __________    _______________    ______________    __________________    _________________    _________    ________

    Yes    'Yes'     17            77.529             63.904            3.8824                0.69663              4.0975       0.39723
    No     'No'      11            138.55             100.74            3.4545                0.68755              4.0121       0.35583
```

Importing processed data files

```
% Subject 1
xP = importdata([fileDir filesep 'RBDS001processed.txt']);

time = xP.data(:,1);  % time normalized vector

varName = 'RhipAngZ25'; % Hip Sagittal Angle at 2.5 m/s

% Find the column corresponding to the variable based on the file header
iVar = strcmp(varName,xP.colheaders(1,:));

xX = xP.data(:,iVar);

% Calculating global maximum and minimum values using max and min functions
[maxVal,imaxVal] = max(xX);
[minVal,iminVal] = min(xX);
```

Ground reaction forces (GRF) impulse calculation based on the area under the curves

```
grfName = 'RgrfX25'; % GRF in the A-P direction

% Find the column corresponding to the variable based on the file header
iGRF = strcmp(grfName,xP.colheaders);

xGRF = xP.data(:,iGRF);
```

```
% Finding values greater and lower than zero
iGRFgt0 = find(xGRF > 0);
iGRFlt0 = find(xGRF < 0);

% Calculating GRF Impulse from GRF curves using trapz function
impGRFpos = trapz(time(iGRFgt0)/length(time),xGRF(iGRFgt0));
impGRFneg = trapz(time(iGRFlt0)/length(time),xGRF(iGRFlt0));
```

Joint work calculation based on the area under the joint power curves

```
powName = 'RhipPow25'; % Hip joint power

% Find the column corresponding to the variable based on the file header
iPow = strcmp(powName,xP.colheaders);

xPow = xP.data(:,iPow);

% Finding values greater and lower than zero
iPowgt0 = find(xPow > 0);
iPowlt0 = find(xPow < 0);

% Calculating GRF Impulse from GRF curves using trapz function
posPower = trapz(time(iPowgt0)/length(time),xPow(iPowgt0));
negPower = trapz(time(iPowlt0)/length(time),xPow(iPowlt0));
```

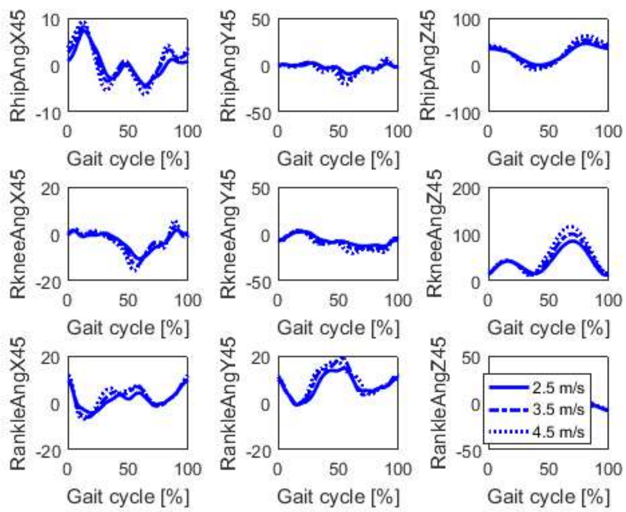
Example of batching processing the data. This can be used to open the processed files and generate plots of angles, moments, powers and GRFs.

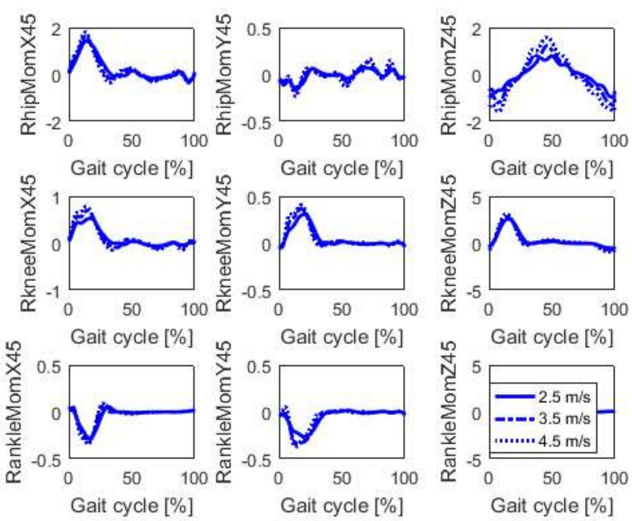
```
nsubjs = 1; % Change this parameter according to the number of subjects to be processed.
```

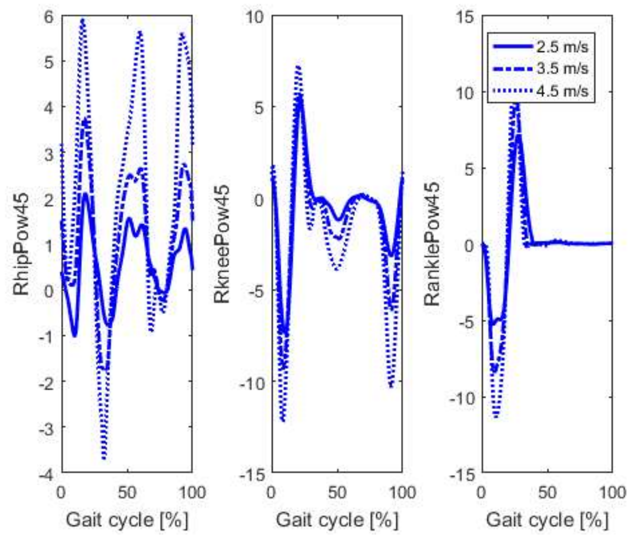
Other parameters

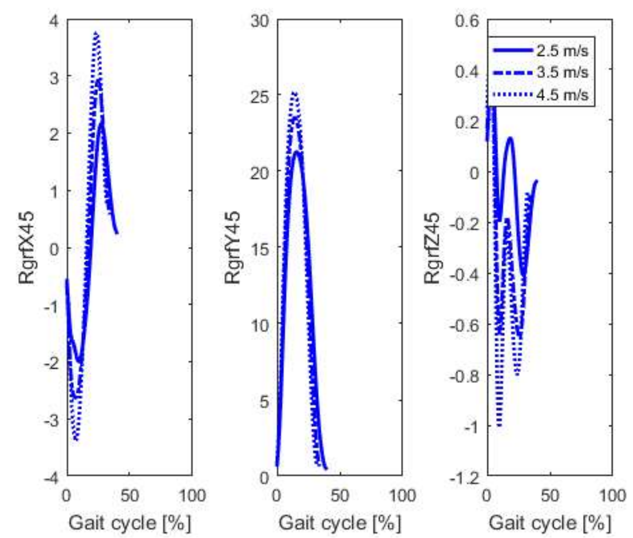
```
speed     = [2.5,3.5,4.5]; % running speeds in m/s
joints    = {'hip','knee','ankle'}; % lower extremity joints
axesXYZ   = {'X','Y','Z'}; % Reference system
side      = {'R','L'}; % Limb side
varType   = {'Ang','Mom','Pow','grf'}; % Biomechanical variable types

% Parameters for plotting data
show   = 1; % control for displaying graphs
cor    = {'b','r'}; symbol = {'-','-.',':'};
ngs = length(speed); % number of different gait speeds

for is = 1:length(side)
    hcurve = [];
    for ivar = 1:length(varType)
        for ij = 1:length(joints)
            for igs = 1:length(speed)
                for xyz = 1:length(axesXYZ)
                    nrows = 3; ncols = 3;

                    step = xyz; stride = ij;

                    if ivar == 3 % Joint power header names
                        % scalar joint power only joints, no xyz
                        xyz = 1; axesXYZ = {''};

                        % Subplot parameters
                        step = ij; stride = 1; nrows = 1; ncols = 3;

                    elseif ivar == 4 % GRF header names
                        % GRF only xyz, no joints
                        ij = 1; joints = {''};

                        % Subplot parameters
                        nrows = 1; ncols = 3; step = xyz; stride = 1;
                    end

                    varName = strcat(side{is},joints{ij},varType{ivar},...
                        axesXYZ{xyz},num2str(speed(igs)*10));

                    xXx = []; %create empty variable

                    for isubj = 1:nsubjs
                        % Import files
                        subLabel = ['RBDS0' num2str(isubj,'%02i')]; % Subject label
                        xP = importdata([fileDir filesep subLabel 'processed.txt']);

                        % Find the column corresponding to the variable based on the header
                        iVar = find(strcmp(varName,xP.colheaders));

                        xX = xP.data(:,iVar);

                        xXx = [xXx xX]; % Concatenate data of different subjects
                    end

                    % Generate the average curves across subjects
                    if show
```

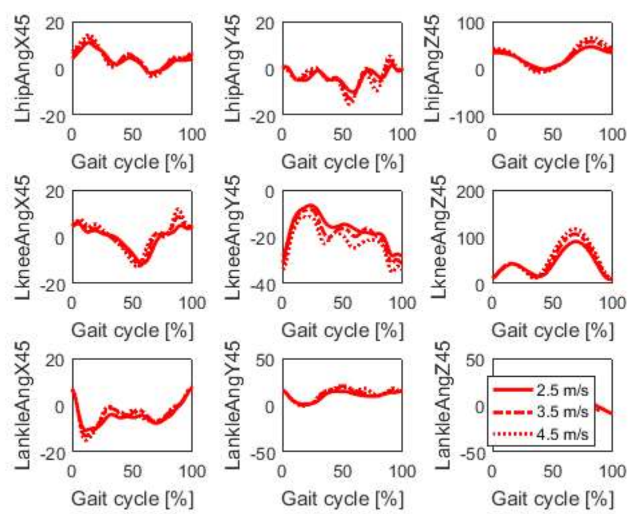
```
                    time = xP.data(:,1); % time normalized vector
                    nvars = length(varType); % Number of variable types

                    figure((nvars*is-nvars)+ivar)
                    subplot(nrows,ncols,(3*stride-3)+step)
                    % Ploting average curve across subjects
                    hcurve = plot(time,mean(xXx,2),...
                        strcat(cor{is},symbol{igs}));
                    set(hcurve,'Linewidth',2)
                    hold on, xlim([0 100])
                    xlabel('Gait cycle [%]'), ylabel(varName)

                    hleg(igs) = hcurve;
                end

                % Update cell arrays
                joints    = {'hip','knee','ankle'}; % lower extremity joints
                axesXYZ   = {'X','Y','Z'}; % Reference system

            end
            % Creating legend for the curves
            legText{igs} = strcat(num2str(speed(igs)),' m/s');

        end
    end
    % Legend of the graphs
    legend(hleg,legText)
  end
end
```

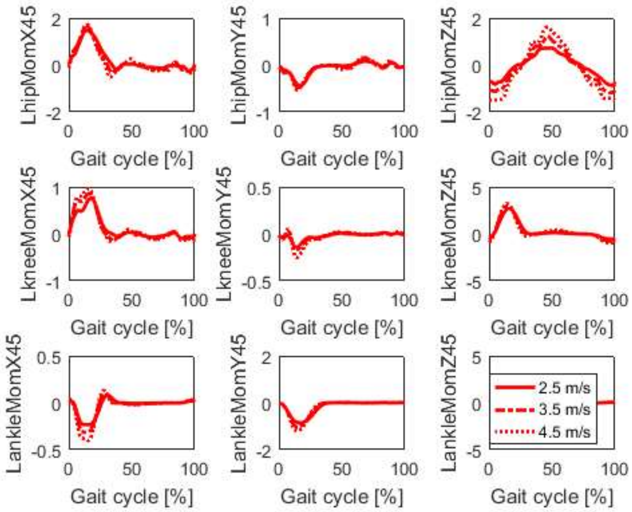


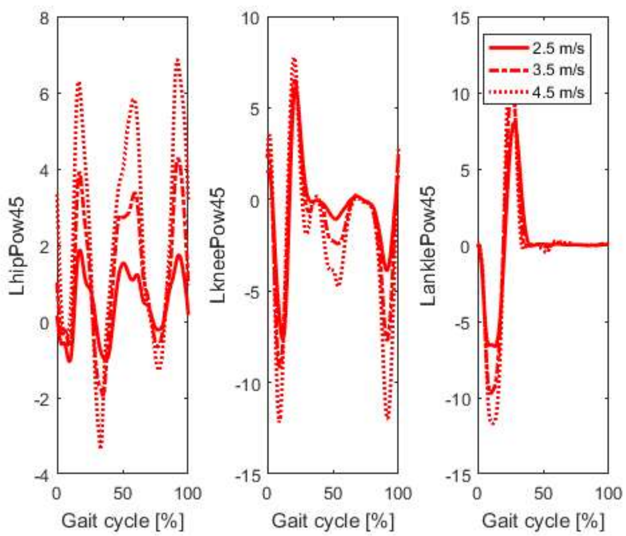


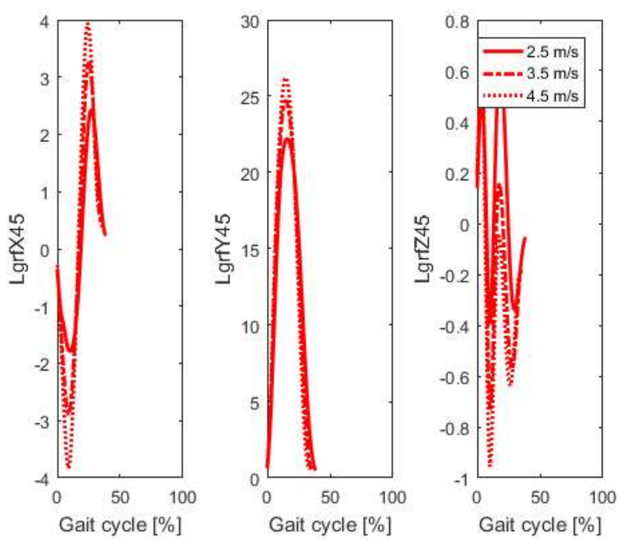

RhipPow45
RkneePow45
RanklePow45
Gait cycle [%]
2.5 m/s
3.5 m/s
4.5 m/s


RgrfX45
RgrfY45
RgrfZ45
Gait cycle [%]
2.5 m/s
3.5 m/s
4.5 m/s


LhipAngX45
LhipAngY45
LhipAngZ45
LkneeAngX45
LkneeAngY45
LkneeAngZ45
LankleAngX45
LankleAngY45
LankleAngZ45
Gait cycle [%]
2.5 m/s
3.5 m/s
4.5 m/s







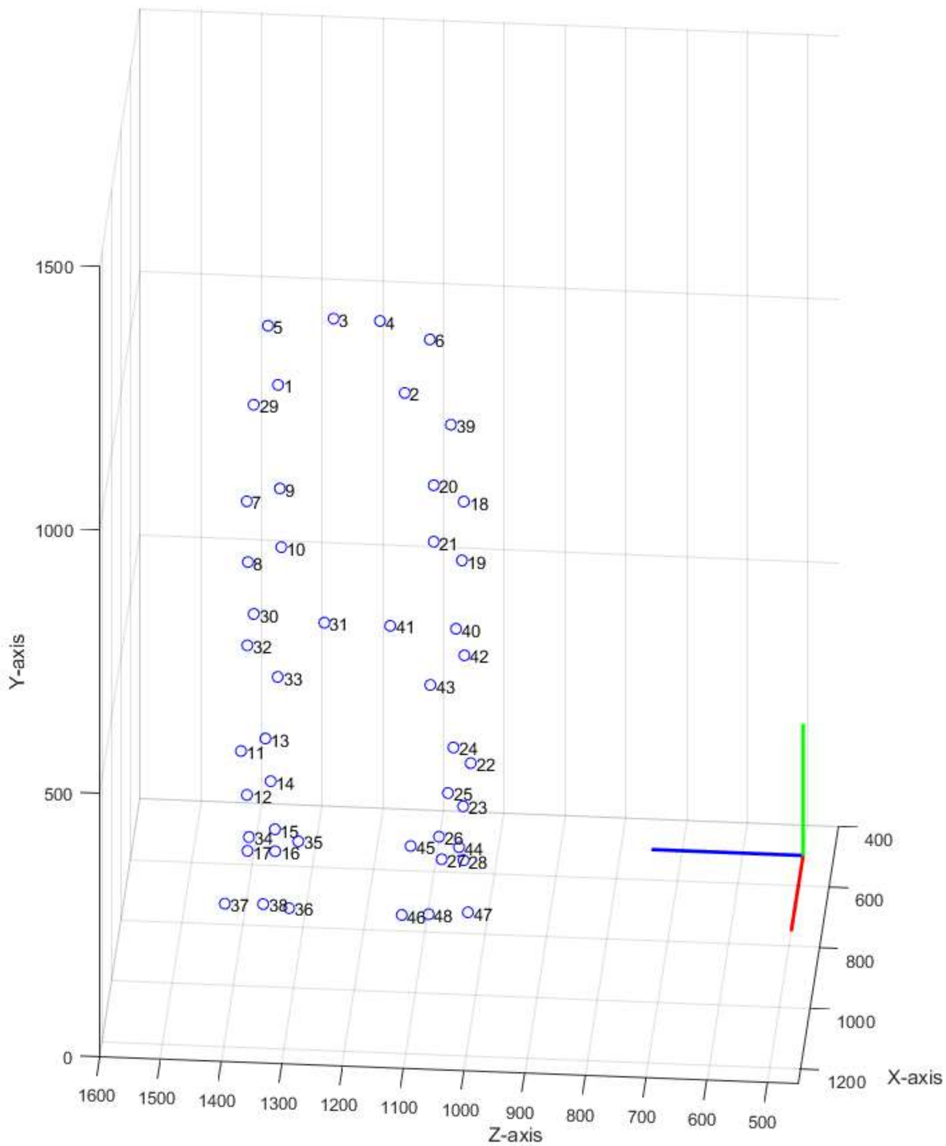
Load and visualize the markers' position during standing calibration trial

```
% Import static trial data
xS = importdata([fileDir filesep 'RBDS001static.txt']);

timeS = xS.data(:,1);

markerLabelS = xS.colheaders(2:end);
```

```
markerLabelS2 = markerLabelS(1:3:end-2);

dataS = xS.data(:,2:end);
```

3D plot of static markers

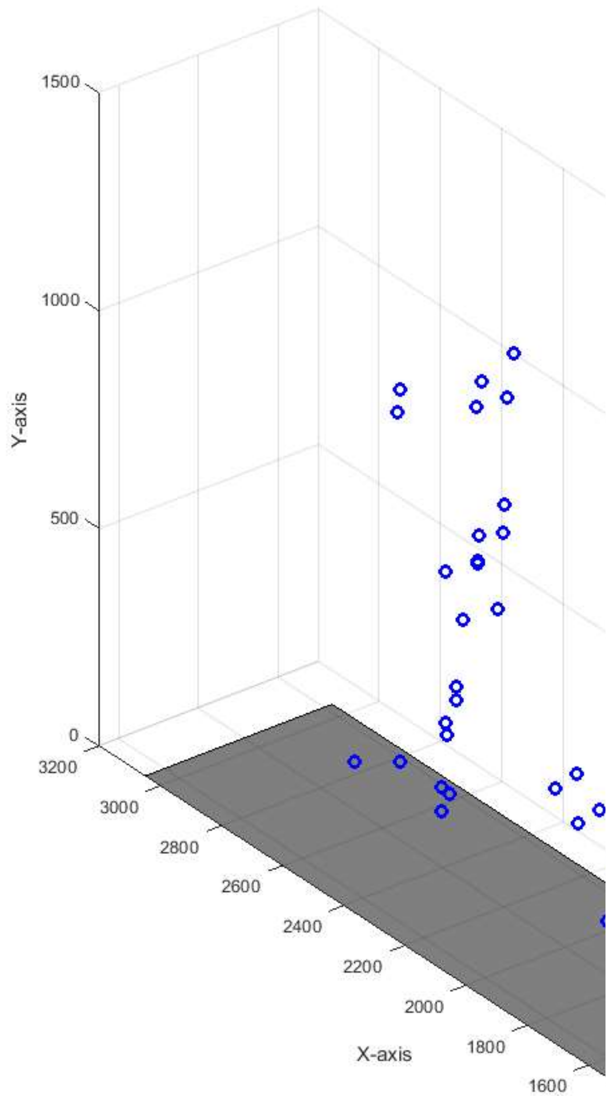
```
figure('units','normalized','outerposition',[0 0 1 1])
subplot(1,2,1)

for i = 1:size(dataS,2)/3
    % Showing standing calibration markers
    h1(i) = plot3(mean(dataS(:,3*i)),mean(dataS(:,3*i-2)),mean(dataS(:,3*i-1)),'bo'); hold on

    % Assigning label to markers
    text(mean(dataS(:,3*i)),mean(dataS(:,3*i-2)),mean(dataS(:,3*i-1)),[' ' num2str(i)])

    leg{i} = [num2str(i) '-' markerLabelS2{i}];
end

% Plotting Lab coordinate system
h2 = plot3([500 500+250],[500 500],[0 0],'b-');
h3 = plot3([500 500],[500 500+250],[0 0],'r-');
h4 = plot3([500 500],[500 500],[0 250],'g-');

set([h2 h3 h4],'Linewidth',2)

xlabel('Z-axis'), ylabel('X-axis'), zlabel('Y-axis')
% view([180 0]) % force figure to be displayed in this view.

axis equal
axis([450 1600 400 1250 0 1500])
grid

% Showing legend of markers
lg = legend(h1,leg);

set(lg,'Position',[0.6629 0.0906 0.1010 0.8261])
set(gca,'CameraPosition',[320.9143 9.7713e+03 5.9311e+03])

% Import markers during running at 3.5 m/s
xD = importdata([fileDir filesep 'RBDS001runT35markers.txt']);

timeD = xD.data(:,1); % time vector

markerLabelD = xD.colheaders(2:end);
markerLabelD2 = markerLabelD(1:3:end-2);

dataD = xD.data(:,2:end);

% Import forces during running at 3.5 m/s
xF = importdata([fileDir filesep 'RBDS001runT35forces.txt']);

timeF = xF.data(:,1);

dataF = xF.data(:,2:end);
```



3D marker trajectories and GRF during treadmill running at 3.5 m/s

```
markers = dataD;
CoPz = dataF(1:2:end,6);
CoPx = dataF(1:2:end,4);
Fz   = dataF(1:2:end,3);
Fy   = dataF(1:2:end,2);
Fx   = dataF(1:2:end,1);

% Treadmill dimensions
widthT  = 486;
lengthT = 1800;

% Position of the geometric center of the treadmill in the lab
centerTposition = [2149 0 976.7];

corner1 = [centerTposition(1)-lengthT/2 0 centerTposition(3)-widthT/2];
corner2 = [corner1(1) corner1(2) corner1(3)+widthT];
corner3 = [corner1(1)+lengthT corner1(2) corner2(3)];
corner4 = [corner3(1) corner3(2) corner1(3)];

n2cm = .75; % Newtons to cm
figure('units','normalized','outerposition',[0 0 1 1])
for i = 1:10:size(markers,1)/2

    % Plotting markers
    hmarkers = plot3(markers(i,3:3:end),markers(i,1:3:end-2),...
        markers(i,2:3:end-1),'bo'); hold on
```

```matlab
    set(hmarkers,'Linewidth',2)

    % Plotting force platform borders
    plot3([corner1(3) corner2(3) corner3(3) corner4(3) corner1(3)],...
        [corner1(1) corner2(1) corner3(1) corner4(1) corner1(1)],...
        [corner1(2) corner2(2) corner3(2) corner4(2) corner1(2)],...
        'k-')


    % Force plate area
    hFP = fill3([corner1(3) corner2(3) corner3(3) corner4(3)],...
        [corner1(1) corner2(1) corner3(1) corner4(1)],...
        [corner1(2) corner2(2) corner3(2) corner4(2)],...
        [0 0 0]);

    hFP.FaceAlpha = 0.5; % Setting transparent filling

    % Plotting GRF vector
    hArrow = plot3([CoPz(i) CoPz(i)+Fz(i)/n2cm],[CoPx(i) CoPx(i)+Fx(i)/n2cm],[0 Fy(i)/n2cm],'k-');

    hArrow.LineWidth = 2;

    xlabel('Z-axis'), ylabel('X-axis'), zlabel('Y-axis')

    grid on
    axis equal, axis([750 1300 1500 3200 0 1500])



    hold off
    pause(0.1)
end
```


1500
1000
500
0
Y-axis
3200
3000
2800
2600
2400
2200
2000
1800
1600
X-axis