ABSTRACT

LUPIANI, ASHLING A. An Investigation into Dynamic Stability Measures to Quantify Human-Prosthesis Interaction. (Under the direction of Dr. He Huang).

A number of measures are used to evaluate human dynamic stability. Of these, the most likely candidates for testing the relationship between a user and their prosthesis are the extrapolated center of mass and margin of stability. Dynamic stability measures are most often calculated using optical motion capture system; however, inertial measurement units present opportunities for flexibility and portability if their limitations can be overcome. Extrapolated center of mass was calculated for one 24-year-old female walker using two different calculations of optical motion capture. Attempts to use inertial measurement units were hindered by drift corresponding to speed.
An Investigation into Dynamic Stability Measures to Quantify Human-Prosthesis Interaction

by
Ashling Ann Lupiani

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Biomedical Engineering

Raleigh, North Carolina
2020

APPROVED BY:

Dr. He Huang
Committee Chair

Dr. Derek Kamper

Dr. Katherine Saul
BIOGRAPHY

Ashling A. Lupiani graduated from Boston University with a Bachelor of Arts in Neuroscience in May 2017. She was involved in numerous outreach activities, including teaching basic neuroscience concepts to high schoolers and training other undergraduates to do the same. As an undergraduate, she worked in the STEPP Lab at Boston University as an Undergraduate Computational Neuroscience Fellow and later as a Visiting Fellow. She joined the Neuromuscular Engineering Research Laboratory at the UNC Chapel Hill & NCSU Joint Department of Biomedical Engineering and is currently pursuing a Master of Science.
ACKNOWLEDGEMENTS

Thank you to all who supported me in this research, including NIH #EB024570 for financial support.
# TABLE OF CONTENTS

LIST OF TABLES .......................................................................................................................... v
LIST OF FIGURES ...................................................................................................................... vi

1. Introduction .............................................................................................................................. 1
   - Step-to-step measures ........................................................................................................ 1
   - Measures based on stochastic dynamics ........................................................................ 2
   - Rotational stability ............................................................................................................ 3
   - Measures based on the inverted pendulum model ............................................................ 4

2. Materials and Methods ........................................................................................................... 6

3. Results ..................................................................................................................................... 11

4. Discussion ................................................................................................................................ 14

5. Conclusion .............................................................................................................................. 15

Appendix A .................................................................................................................................. 20
LIST OF TABLES

Table 1.1 Properties of common measures of stability .............................................................. 5
LIST OF FIGURES

Figure 2.1 Schematic of optical and inertial measurement unit motion capture setup .................. 7
Figure 2.2 Conceptual schematic for extrapolated center of mass ........................................... 9
Figure 3.1 Center of mass and extrapolated center of mass from Visual 3D ............................. 11
Figure 3.2 Center of mass and extrapolated center of mass as calculated manually in MATLAB ............................................................................................................. 12
Figure 3.3 Center of mass and extrapolated center of mass as calculated from MVN data ...... 13
1. Introduction

The interactive relationship between a human user and their prosthesis is difficult to quantify. At the same time, the success of the system depends on this relationship: the user’s ability to manipulate their prosthesis is as important as the capabilities of the prosthesis itself. One way to quantify the relationship between user and prosthesis device is to intentionally disturb the system (Zhang et al., 2015a, 2015b). This allows the determination of how the goals and strategies of the two systems (i.e., human and prosthesis) can differ when disturbed and compare this to how their goals and strategies normally align.

To this end, the use of stability metrics is helpful. However, there is no gold standard to define the balance and stability in human-prosthesis systems. Most commonly used metrics for stability depend on a model of the human body which may not truly represent the prosthesis user. The following is an overview of some of the more common stability measures.

Step-to-step measures

Perhaps the simplest of measures to obtain are those which quantify basic dimensions of steps. Mean stride length, stride width, and stride time, as well as their respective variances, have been variously used to determine whether or not gait is balanced (Beauchet et al., 2009). Less balanced individuals tend to compensate with shorter, wider, and slower strides. While these measures can be completed quickly and easily, they do not correlate with other measures of stability and provide spatially-limited information. It is not always easy to draw conclusions from these measures since changes may be due to an impairment or may be related to a compensation for the impairment.
Measures based on stochastic dynamics

Some measures that attempt to quantify stability do so by modeling gait kinematics as noisy dynamical systems. With this conception of gait, the overall complexity of gait functions can be determined, either mathematical complexity or, more commonly, entropy-based complexity (Costa et al., 2003). Information about temporal correlations can also be found by reconstructing state space representations of gait kinematics (Dingwell & Cusumano, 2000).

One measure which leverages this ideation of gait is maximum Floquet multipliers. These measure orbital stability—the adherence of the gait pattern to an approximating stride-to-stride function. The theoretical justification for this measure is based on a three-link planar bipedal model, and the changes of segment angles within that model. This model produces closed orbits of the in the phase space, and thus can be approximated by a state-space stride-to-stride function. The Floquet multipliers, also known as Lyapunov exponents, are the eigenvalues of the Jacobian of the stride-to-stride function (Hurmuzlu & Basdogan, 1994). However, ‘true’ Floquet multipliers assume exactly periodic motion, which may not hold for humans, especially amputees. Additionally, the calculation of Floquet multipliers requires large amounts of data. Despite these limitations, Floquet multipliers have been found to be a better measure than average stride length and walking speed (Dingwell & Cusumano, 2000).

Due to the limitations of ‘true’ Floquet multipliers, finite-time Lyapunov exponents, also known as local divergence exponents, have been proposed as an alternate measure. The finite-time measure captures local stability, or the response of the system to very small perturbations. It requires less data than true Lyapunov exponents, but it still necessitates a considerable number of continuous steps (Dingwell & Cusumano, 2000). Because local stability assesses noise, finite-time Lyapunov exponents may be impacted by the inherent noisiness of experimental data.
Moreover, local stability has been found to be affected by the differences between treadmill and overground walking (Dingwell et al., 2001).

Able-bodied walkers have been found to have orbital stability but local instability. These types of stability are not well correlated with each other (Dingwell & Hyun, 2007). Both sample entropy and local stability are different between transfemoral amputees and controls (Lamoth et al., 2010). While these measures attempt to quantify stability by taking advantage of the semi-periodic nature of human gait, they are limited by their requirements for large amounts of data. This requirement also precludes the possibility of real-time calculation for Lyapunov exponents and entropy. Furthermore, both kinds of Lyapunov exponents are generally evaluated in a five-dimensional space, which may not be intuitive for non-research populations, such as clinicians, who are more familiar with Cartesian three-dimensional space.

**Rotational stability**

Another method of quantifying stability is by evaluating the rotational motion of the body during gait. In typical able-bodied walking, counterbalancing between the segments serves to minimize the overall rotational variability. Angular momentum undergoes cyclic fluctuations about zero angular momentum in all three dimensions (Herr & Popovic, 2008). It may be possible to estimate angular momentum in real time, as it is based mainly on the position and velocities of segmental centers of mass. The local angular momentum of each segment, which consists of the inertial tensor and the angular velocity of the segment may present difficulties for this application due to the additional level of complexity. Additionally, the relationship between instability and angular momentum is unclear.
Measures based on the inverted pendulum model

The whole-body center of mass (COM), as used for stability estimation, is based on the inverted pendulum model of human walking. In this model, the human body during support is represented by a point mass (with location estimated by a weighted average of the segment center of masses) attached to a ‘leg’ segment which rotates on the ground. The center of gravity is the projection of the center of mass onto the ground. Generally, the goal is thought to be to keep the center of mass within the base of support (defined by the possible limits of the excursion of the center of pressure) or to keep the center of gravity near the center of pressure (MacKinnon & Winter, 1993; D. A. Winter, 1995). However, this does not account for the time in gait when the center of mass is outside of the base of support but moving toward it. Thus, there has been proposed an extrapolated center of mass (XCOM), the projection of the center of mass based on its velocity (Pai & Patton, 1997; Patton et al., 1999). Despite gait asymmetry, the average distance between the center of pressure and the extrapolated center of mass is similar between controls and prosthesis users’ intact limb, though it is slightly higher on the prosthetic side (Hof et al., 2007). Most commonly, the extrapolated center of mass (XCOM) is reported in terms of the margin of stability (MOS), or the distance between the extrapolated center of mass and the boundaries of the base of support. Larger variability in the margin of stability may be indicative of less stable gait (McAndrew Young et al., 2012; Brandt and Huang, 2019). Thus, considering the properties of the measures of stability surveyed here (see Table 1.1), the margin of stability is likely the best measure for widely used real-time calculation. It requires relatively little data and can be intuitively interpreted.
Table 1.1 Properties of common measures of stability

<table>
<thead>
<tr>
<th>Measure</th>
<th>Continuous/Discrete</th>
<th>1D/2D/3D</th>
<th>Real Time?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stride parameters (length, width, time)</td>
<td>Discrete</td>
<td>1D</td>
<td>Yes</td>
</tr>
<tr>
<td>Trunk acceleration RMS</td>
<td>Discrete</td>
<td>2D</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum Floquet multipliers</td>
<td>Discrete</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Finite-time local divergence exponents</td>
<td>Discrete</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Angular momentum</td>
<td>Continuous</td>
<td>3D</td>
<td>Yes</td>
</tr>
<tr>
<td>Center of mass – center of pressure</td>
<td>Continuous</td>
<td>2D</td>
<td>Yes</td>
</tr>
<tr>
<td>Extrapolated center of mass</td>
<td>Continuous</td>
<td>3D</td>
<td>Yes</td>
</tr>
<tr>
<td>Margin of stability</td>
<td>Continuous</td>
<td>2D</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Typically, metrics for dynamic stability are determined using an optical motion capture system, such as Vicon Nexus. This allows for highly accurate readings of the position of each limb and, used in conjunction with force plates, can give very detailed data. However, optical motion capture systems require a complicated setup with limited range that cannot be easily moved. This makes it difficult to obtain metrics in a number of real-life scenarios, such as outside of the lab or in setups that might block the camera’s view of the participant.

As an alternative to optical motion capture, there is inertial measurement unit (IMU)-based motion capture, which has the advantage of being highly mobile. However, it is unclear how accurate IMU-based motion capture would be in determining dynamic stability measures such as extrapolated center of mass. The goal of this thesis is to quantify XCOM with both an optical motion capture system and an IMU-based system to determine if the IMU-based system (MVN, XSens Motion Technologies, Netherlands) or simplifications in the optical motion capture system are accurate enough to be useful when measuring the margin of stability.
2. Materials and Methods

Data was collected on one 24-year-old able-bodied female with IRB approval. Forty-two retroreflective markers were applied according to the Plug-in Gait model used by the camera-based optical motion capture system (VICON, UK), and 17 inertial measurement units (IMUs) were fitted as per XSens MVN instructions (Xsens, Netherland) (see Figure 2.1). The optical motion capture system was calibrated via T-pose in the Vicon software and the MVN IMU system was calibrated with the built-in dynamic calibration. The participant walked for 30 seconds on an instrumented treadmill (Bertec, US) at 1.1 meters per second. The ground reaction force were record at 1000Hz by the force plates on the treadmill. The marker positions were sampled at 100Hz. The IMU data were sampled at 60Hz. Data were recorded simultaneously on the two systems.
Pre-processing for optical motion capture data was completed in Vicon Nexus, including gap filling for missing markers. Data were then exported and processed either in Visual 3D or with a custom MATLAB code for estimation of the center of mass (COM) position. The proprietary Visual 3D software was employed to compute the center of mass because of its common use in gait research, while the MATLAB code was less precise but more transparent. Results from the two methods were not expected to be significantly different.

**Figure 2.1 Schematic of optical and inertial measurement unit motion capture setup.** Front (a) and back (b). Optical motion capture markers are indicated by grey circles, inertial measurement units are indicated by orange lozenges.
MATLAB was also used to compute the center of mass. While the processes of the Visual 3D software were opaque, the estimation of the COM in MATLAB was calculated as follows (see also: Appendix A). Marker locations from the full body Plug-In Gait model were used to represent joint center locations. The location of the center of mass of each segment was estimated using center of mass/segment length ratios from previous work (D. Winter, 2009). These were then weighted according to their approximate mass using segment weight/total body weight from previous work (D. Winter, 2009) to determine the whole-body center of mass. Center of mass velocity was approximated by differentiating center of mass position and filtering with a 4th order lowpass Butterworth filter (10 Hz).

Extrapolated center of mass (XCOM) was then determined for both COM calculations in MATLAB by adjusting center of mass position, obtained either from the Vicon software or MATLAB, by the velocity in relation to $\sqrt{g/l}$, where $g$ is the acceleration due to gravity and $l$ is the length of the ‘leg’ segment in the inverted pendulum model, such that $XCoM = \frac{dCoM}{dt} \sqrt{g/l}$, where $\frac{dCoM}{dt}$ is the change in position of the center of mass over time (see Figure 2.2).
The calculation methods were compared by determining the root mean square error

\[ x_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} |x_n|^2} \]

in the mediolateral direction for center of mass and extrapolated center of mass for both methods, where \( x_n = \text{CoM}_{V3D} - \text{CoM}_{MATLAB} \) or \( x_n = \text{XCoM}_{V3D} - \text{XCoM}_{MATLAB} \).

Because the MVN data is subject to significant drift in rotation and origin location, attempts were made to alleviate drift before calculating extrapolated center of mass. These approaches included rotating segments to align the positive x axis of the global coordinate

---

**Figure 2.2 Conceptual schematic for extrapolated center of mass.** Indicated locations for center of mass (CoM), center of gravity (CoG), center of pressure (CoP), base of support (BoS), and extrapolated center of mass (XCoM).
system with trunk segment and attempting to compensate for origin drift related to the speed of motion.
3. Results

The mediolateral range for center of mass was 0.0619 m for the Visual 3D calculation (Figure 3.1) and 0.0630 m for the MATLAB calculation (Figure 3.2).

The root mean square error values between the Visual 3D system and the MATLAB calculations were 0.0117 m for COM and 0.0131 m for XCOM.

Figure 3.1 Center of mass and extrapolated center of mass from Visual 3D. Grey regions indicate swing phase, left (light) or right (dark). Center of mass (CoM, red solid line) was taken from the Visual 3D model, extrapolated center of mass (XCoM, blue dashed line) was calculated manually in MATLAB from Visual 3D CoM data.
The COM and XCoM values computed from the IMU data were less smooth and tended to drift over time (Figure 1.5).

Figure 3.2 Center of mass and extrapolated center of mass as calculated manually in MATLAB. Grey regions indicate swing phase, left (light) or right (dark). Center of mass (CoM, red solid line) was estimated based on optical motion capture marker locations and extrapolated center of mass (XCoM, blue dashed line) was estimated based on this CoM calculation.

The COM and XCoM values computed from the IMU data were less smooth and tended to drift over time (Figure 1.5).
Figure 3.3 Center of mass and extrapolated center of mass as calculated from MVN data. Grey regions indicate swing phase, left (light) or right (dark). Center of mass (CoM, red solid line) and extrapolated center of mass (XCoM, blue dashed line) are much less smooth and show drift in comparison with values computed from Vicon data.
4. Discussion

Overall, the estimation built on a simple anthropometry-based model was reasonably similar to the estimation from the expensive and opaque Visual 3D software. Both had mediolateral ranges for the center of mass of 6-7 cm and they had similar root mean square error. The extrapolated center of mass from Visual 3D was slightly more consistent in shape with previous work than the center of mass calculated in MATLAB; this may be due to smaller inaccuracies in the MATLAB center of mass estimation being augmented by differentiation.

Previous studies have found a wide array of acceptable center of mass range values, from $0.0329 \pm 0.0129$ m to $0.0699 \pm 0.0134$ m depending on walking speed (Orendurff et al., 2004). The estimations for center of mass from both Visual 3D and MATLAB fell well within that range. The movement of the center of mass and extrapolated center of mass over time also corresponds with previous work in terms of its cyclic motion (Hof et al., 2007).

The generalizability of this study was limited by the single participant and time constraints making it infeasible to sync the Vicon and MVN systems. These systems do have a sync function that should allow time-aligned recording, which future work should utilize. Future work should include more participants and expand the scope to include overground walking.

The drift within the IMU motion capture data was significant. While this may have been exacerbated by treadmill walking, drift has been observed in previous pilots for overground walking as well. The XSens system includes an extra ‘prop’ sensor to record the motion of objects of interest, and it may be possible to use this sensor to define a stationary origin and compensate properly for drift. Future work should explore this possibility.
5. Conclusion

Considering dynamic stability measures in terms of the need for a simple, intuitive metric to evaluate the relationship between a user and their prosthesis, extrapolated center of mass and the related concept of margin of stability present themselves as the best options. Extrapolated center of mass can be found with reasonable accuracy using a simple anthropometry-based model with optical motion capture. At this time, it is not possible to accurately calculate extrapolated center of mass with inertial measure unit motion capture, but future work may present better options for correcting drift.
REFERENCES


Zhang, F., Liu, M., & Huang, H. (2015a) Effects of locomotion mode recognition errors on

Appendix A

% extract segment origin positions and body COM position
% define number of frames & beginning of data
load('vicon_wtdata2')
numFrames = height(dynamicv2);
gender = "F";
body_mass = 57.38784;

% initialize the list of segment origins
origin_list = {'Pelvis', 'L5', 'L3', 'T12', 'T8_Trunk', 'Neck', 'Head', ...
                'RShoulder', 'RUpperArm', 'RForearm', 'RHand', ...
                'LShoulder', 'LUpperArm', 'LForearm', 'LHand', ...
                'RUpperLeg', 'RLowerLeg', 'RFoot', 'RToe', ...
                'LUpperLeg', 'LLowerLeg', 'LFoot', 'LToe'};

% define joint centers within framework
Position.RUpperArm = [dynamicv2.('RSHOX'), dynamicv2.('RSHOY'),
                      dynamicv2.('RSHOZ')];
Position.LUpperArm = [dynamicv2.('LSHOX'), dynamicv2.('LSHOY'),
                      dynamicv2.('LSHOZ')];
Position.RForearm = [dynamicv2.('RELBX'), dynamicv2.('RELBY'),
                     dynamicv2.('RELBZ')];
Position.LForearm = [dynamicv2.('LELBX'), dynamicv2.('LELBY'),
                     dynamicv2.('LELBZ')];
Position.RHand = [(dynamicv2.('RWRAX')+dynamicv2.('RWRBX'))/2, ...
                  (dynamicv2.('RWRAY')+dynamicv2.('RWRBY'))/2, ...
                  (dynamicv2.('RWRAZ')+dynamicv2.('RWRBZ'))/2];
Position.LHand = [(dynamicv2.('LWRAX')+dynamicv2.('LWRBX'))/2, ...
                  (dynamicv2.('LWRAY')+dynamicv2.('LWRBY'))/2, ...
                  (dynamicv2.('LWRAZ')+dynamicv2.('LWRBZ'))/2];
Position.RLowerLeg = [dynamicv2.('RKNEX'), dynamicv2.('RKNEY'),
                     dynamicv2.('RKNEZ')];
Position.LLowerLeg = [dynamicv2.('LKNEX'), dynamicv2.('LKNEY'),
                     dynamicv2.('LKNEZ')];
Position.RFoot = [dynamicv2.('RANKX'), dynamicv2.('RANKY'),
                  dynamicv2.('RANKZ')];
Position.LFoot = [dynamicv2.('LANKX'), dynamicv2.('LANKY'),
                 dynamicv2.('LANKZ')];
Position.RUpperLeg = [(dynamicv2.('RASIX')+dynamicv2.('RPSIX'))/2,...
                       (dynamicv2.('RASIY')+dynamicv2.('RPSIY'))/2,...
                       (dynamicv2.('RASIZ')+dynamicv2.('RPSIZ'))/2];
Position.LUpperLeg = 
[(dynamicv2.('LASIX')+dynamicv2.('LPSIX'))/2,...
 (dynamicv2.('LASIY')+dynamicv2.('LPSIY'))/2,...
 (dynamicv2.('LASIZ')+dynamicv2.('LPSIZ'))/2];
Position.Head = 
[(dynamicv2.('RFHDX')+dynamicv2.('RBHDX')+dynamicv2.('LFHDX')+dynamicv2.('LBHDX'))/4,...
 (dynamicv2.('RFHDY')+dynamicv2.('RBHDY')+dynamicv2.('LFHDY')+dynamicv2.('LBHDY'))/4,...
 (dynamicv2.('RFHDZ')+dynamicv2.('RBHDZ')+dynamicv2.('LFHDZ')+dynamicv2.('LBHDZ'))/4];

%% find position of segment COMs
%initialize list of segments/angles
segment_list_full = {origin_list([9:10 16:18 13:14 20:22]),'Head','T8_Trunk'};
%nSeg = length(segment_list_full);
%for each segment
for x=1:nSeg
 %the first set of segments (not head or trunk) will be the right segments
 if x <= (nSeg-2)/2
  %the corresponding segment type is the same
  segment_type = extractAfter(segment_list_full{x},'R');
  %the second set will be the left segments, final two will be head and trunk
 else
  %find the corresponding segment type
  if x <= nSeg-2
   segment_type = extractAfter(segment_list_full{x},'L');
  else
   segment_type = segment_list_full{x};
  end
 end
%switch for segment types
switch segment_type
 %set UpperArm offset ratio
 case 'UpperArm'
  offset_ratio = 0.576;
 %set Forearm offset ratio
 case 'Forearm'
  offset_ratio = 0.456;
 %set UpperLeg offset ratio
 case 'UpperLeg'
  offset_ratio = 0.456;
end
case 'UpperLeg'
    offset_ratio = 0.382;
%set LowerLeg offset ratio
case 'LowerLeg'
    offset_ratio = 0.444;
%set Foot offset ratio
case 'Foot'
    offset_ratio = 0.42;
end

b = find(ismember(origin_list,segment_list_full{x}));
if ~any([b==5,b==7,b==11,b==15,b==18,b==22]) %not trunk, head, hands, feet
    COM_Position.(segment_list_full{x}) = offset_ratio*...
        Position.(origin_list{b}) + (1-offset_ratio)*...
        Position.(origin_list{b+1});
end

COM_Position.T8_Trunk = [0.5*dynamicv2.'C7X' +
0.5*dynamicv2.'TTENX',...  
0.5*dynamicv2.'C7Y' + 0.5*dynamicv2.'TTENY',...
0.5*dynamicv2.'C7Z' + 0.5*dynamicv2.'TTENZ'];

COM_Position.Head =
[0.5*dynamicv2.'C7X',0.5*dynamicv2.'C7Y',...
0.5*dynamicv2.'C7Z'] + 0.5*Position.Head;

COM_Position.RHand =
[0.5*dynamicv2.'RFINX',0.5*dynamicv2.'RFINY',...
0.5*dynamicv2.'RFINZ'] + 0.5*Position.RHand;

COM_Position.LHand =
[0.5*dynamicv2.'LFINX',0.5*dynamicv2.'LFINY',...
0.5*dynamicv2.'LFINZ'] + 0.5*Position.LHand;

COM_Position.RFoot = [0.5*dynamicv2.'RHEEX' +
0.5*dynamicv2.'RTOEX',...
0.5*dynamicv2.'RHEYE' + 0.5*dynamicv2.'RTOEY',...
0.5*dynamicv2.'RHEE' + 0.5*dynamicv2.'RTOE'];

COM_Position.LFoot = [0.5*dynamicv2.'LHEEX' +
0.5*dynamicv2.'LTOEX',...
0.5*dynamicv2.'LHEEY' + 0.5*dynamicv2.'LTOEY',...
0.5*dynamicv2.'LHEE' + 0.5*dynamicv2.'LTOE'];

% distribute mass
Weighted_COM{:,:,1} = COM_Position.RUpperArm*... %Upper arm 
body_mass*.028+COM_Position.LUpperArm*body_mass*.028;
Weighted_COM{:,:,2} = COM_Position.RForearm*... %Forearm 
body_mass*.016+COM_Position.LForearm*body_mass*.016;
Weighted_COM{:,:,3} = COM_Position.RUpperLeg*... %Upper leg 
body_mass*.1+COM_Position.LUpperLeg*body_mass*.1;
Weighted_COM{:,:,4} = COM_Position.RLowerLeg*... %Lower leg 
body_mass*.0465+COM_Position.LLowerLeg*body_mass*.0465;
Weighted_COM{:,:,5} = COM_Position.Head*body_mass*8.1/100; %Head
Weighted_COM{:,:,6} = COM_Position.T8_Trunk*... %Trunk 
body_mass*50.9/100;
Weighted_COM{:,:,7} = COM_Position.RFoot*... %Foot 
body_mass*.0145+COM_Position.LFoot*body_mass*.0145;

%% find body COM position
Body_COM_Position = 
sum(cell2mat(Weighted_COM),3)/(body_mass*100);

%% find body COM velocity
Body_COM_Velocity = (diff(Body_COM_Position-5)/(1/50)); %+ 0.2;
nanny = find(isnan(Body_COM_Velocity(:,1)));
Body_COM_Velocity(nanny,:) = [];
% filter body COM velocity
[b,a] = butter(4,10/100);
Body_COM_Velocity = filtfilt(b,a,Body_COM_Velocity);

% estimate length ‘L’
L = repmat(1.34*0.802,[length(Body_COM_Velocity) 1]);

% find extrapolated center of mass
XCoM = Body_COM_Position(2:length(Body_COM_Velocity)+1,:)+ 
Body_COM_Velocity./(sqrt(9.81./repmat(L,[1 3])));

% define time vector
t=((1:length(XCoM))/100)';